

Inference Over Bayesian Networks for the Diagnosis and Sensory Enhancement of Mobile Robots

PhD Thesis Dissertation

by

Manuel Castellano Quero

Under the supervision of

Dr. Juan Antonio Fernández Madrigal

Dr. Alfonso José García Cerezo




Mechatronics Engineering Doctoral Programme
Systems Engineering and Automation Department
University of Málaga
2021



UNIVERSIDAD
DE MÁLAGA

AUTOR: Manuel Castellano Quero

 <https://orcid.org/0000-0001-9377-1274>

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional:

<http://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>

Cualquier parte de esta obra se puede reproducir sin autorización pero con el reconocimiento y atribución de los autores.

No se puede hacer uso comercial de la obra y no se puede alterar, transformar o hacer obras derivadas.

Esta Tesis Doctoral está depositada en el Repositorio Institucional de la Universidad de Málaga (RIUMA): riuma.uma.es





DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

D./Dña MANUEL CASTELLANO QUERO

Estudiante del programa de doctorado INGENIERÍA MECATRÓNICA de la Universidad de Málaga, autor/a de la tesis, presentada para la obtención del título de doctor por la Universidad de Málaga, titulada: INFERENCE OVER BAYESIAN NETWORKS FOR THE DIAGNOSIS AND SENSORY ENHANCEMENT OF MOBILE ROBOTS

Realizada bajo la tutorización de D. JUAN ANTONIO FERNÁNDEZ MADRIGAL y dirección de D. ALFONSO JOSÉ GARCÍA CEREZO Y D. JUAN ANTONIO FERNÁNDEZ MADRIGAL

DECLARO QUE:

La tesis presentada es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, conforme al ordenamiento jurídico vigente (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), modificado por la Ley 2/2019, de 1 de marzo.

Igualmente asumo, ante a la Universidad de Málaga y ante cualquier otra instancia, la responsabilidad que pudiera derivarse en caso de plagio de contenidos en la tesis presentada, conforme al ordenamiento jurídico vigente.

En Málaga, a 9 de SEPTIEMBRE de 2021

Fdo.: MANUEL CASTELLANO QUERO Doctorando	Fdo.: JUAN ANTONIO FERNÁNDEZ MADRIGAL Tutor y Director de tesis
Fdo.: ALFONSO JOSÉ GARCÍA CEREZO Director de tesis	



UNIVERSIDAD DE MÁLAGA

DEPARTAMENTO DE
INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

El Dr. D. Juan Antonio Fernández Madrigal y el Dr. D. Alfonso José García Cerezo, directores de la tesis titulada *Inference Over Bayesian Networks for the Diagnosis and Sensory Enhancement of Mobile Robots* realizada por D. Manuel Castellano Quero, certifican su idoneidad para la obtención del título de Doctor en Ingeniería Mecatrónica.

Málaga, 9 de septiembre de 2021.

Dr. D. Juan Antonio Fernández Madrigal

Dr. D. Alfonso José García Cerezo

Contents

Acknowledgments	vi
Abstract	vii
Summary in Spanish (<i>Resumen de la tesis</i>)	viii
Contexto y ámbito de la tesis	viii
Motivación	xi
Contribuciones	xii
Marco de la tesis y actividades de investigación	xvi
Publicaciones	xvii
Estructura y contenido de la tesis	xvii
Conclusiones	xviii
Trabajo futuro	xx
1 Introduction	1
1.1 Context and scope of the thesis	1
1.2 Motivation	4
1.3 Contributions	5
1.4 Framework and research activities	9
1.5 Publications	9
1.6 Outline	10
2 Probabilistic and statistical bases	11
2.1 Introduction	11
2.2 Bayesian networks	12
2.2.1 Representation	12
2.2.2 Inference	17
2.3 Analysis of variance (ANOVA)	21
2.3.1 Basic notions	21
2.3.2 N-way ANOVA	23
3 Influence of abnormal sensory behavior in mobile robots	27
3.1 Introduction	27
3.2 Related research	29
3.3 Study of the impact on the performance of Recursive Bayesian Filters	31
3.3.1 Sensory anomalies and limitations	32
3.3.2 Generic Bayesian networks for filtering range sensors	32
3.3.3 Filter performance measures and problem characterization	35

CONTENTS

3.3.4	Statistical tools	37
3.3.5	Procedure for the analysis and deduction of conclusions	37
3.3.6	Gathering data	41
3.4	Results of the study and discussion	47
3.4.1	Statement of hypotheses	47
3.4.2	Statistical analysis	48
3.4.3	Impact on the operation of mobile robots	50
3.4.4	Validation in a real experiment	51
4	Bayesian networks for sensory diagnosis and recovery in mobile robots	55
4.1	Introduction	55
4.2	Related research	56
4.3	A representation of robotic sensory systems using Bayesian networks	58
4.3.1	The Bayesian sensor. Inference in sensors	58
4.3.2	A Bayesian architecture for robotic sensory systems	61
4.4	An inference algorithm for approximate reasoning in robotic sensory systems	69
4.4.1	Proposed methodology	70
4.4.2	Computational complexity	76
5	Implementation and validation	79
5.1	Introduction	79
5.2	Navigation in environments with human presence	81
5.3	Experimental setup	82
5.4	Evaluation of the inference algorithm	83
5.4.1	Definition of the Bayesian architecture	84
5.4.2	Simulated experiments	91
5.4.3	Real experiments	94
5.5	Implementation for the problem of navigation in human environments	103
5.5.1	Definition of the Bayesian architecture	103
5.5.2	Simulated experiments	109
5.5.3	Real experiments	113
5.5.4	Improving inference efficiency with neural networks	115
6	Final remarks	121
6.1	Conclusions	121
6.2	Future work	123
A	Full report of the results obtained for the study of abnormal sensory behavior	124
A.1	Expected accuracy performance	124
A.2	Expected uncertainty performance	129
A.3	Convergence performance	133
	References	140

Acknowledgments

In the first place, I would like to express my most sincere and profound gratitude to my supervisors, Prof. Juan Antonio Fernández Madrigal and Prof. Alfonso García Cerezo, who have supported me in countless ways throughout this difficult path. Juan Antonio, thanks for sharing your invaluable experience with me, and also for your patience and guidance, which I consider essential contributions to any work. Alfonso, thanks for your trust and advice whenever I needed them. Thanks, again, to both of you. This project would not have been possible without your help.

This thesis has been supported by the Spanish government through the national grant FPU16/02243, by the University of Málaga through its local research program and the International Excellence Campus Andalucía Tech, and by the national research projects DPI2015-65186-R and RTI2018-093421-B-100.

This thesis would not have been possible either without the collaboration of Prof. Holger Voos and the members of the Automation and Robotics group of the University of Luxembourg, who I thank for the amazing three-month research stay and for such an unforgettable experience. Special thanks go to my friend, Manuel Castillo, one of the doctoral candidates of that group, for his incredibly valuable help and for his contributions to this work. I would also like to thank my colleagues at the University of Málaga for all their advices and support. I will always be grateful to Prof. Ana Cruz, for sharing her teaching experience with me and also for all those amazing moments in class. Also, I have to mention my friends Javier, Antonio, Juanma and Curro, and all the members of the MAPIR research group, who have helped me to understand what is truly important in reasearch and life.

The contributions presented in this thesis are the result of nearly four years of work, sacrifice and patience. I would like to thank my parents here, not only for trusting me, but also for being by my side in all the good and bad moments of such a tough process. Undoubtedly, I have made it this far because of them.

Place: University of Málaga

Date: July 16, 2021

Manuel Castellano Quero

Abstract

Mobile robots are nowadays present in countless real-world applications, aiding or substituting human beings in a wide variety of tasks related to scopes as diverse as industrial, military, medical, educational and many others. The use of mobile platforms in all these contexts is revolutionizing their respective fields, overcoming previous limitations and offering new possibilities. However, for a mobile robot to work properly, it is essential that its sensory apparatus provides correct and reliable information, which is often challenging due to the complexity of the physical world and its uncertain nature. To address that, this thesis explores the possibilities of the application of Bayesian networks (BNs) to the problem of sensory diagnosis and enhancement in the context of mobile robotics. Arised from the realm of artificial intelligence, Bayesian networks constitute a rigorous mathematical framework that enables both the integration of heterogeneous sources of information and the reasoning about them while taking their uncertainty into account. The thesis first analyzes different sensory anomalies in mobile robots and the impact of such abnormal behavior on the performance of these platforms. Given the wide variety of existing sensory devices, the analysis is focused on range sensors, since they are essential to many robotic tasks also grounded on probabilistic frameworks such as Bayesian estimators. Specifically, the thesis contributes with a rigorous statistical study of the influence of abnormal range observations on the performance of Bayesian filters, addressing the problem from a generic perspective thanks to the use of BNs. The conclusions obtained serve to illustrate the importance of sensory abnormalities beyond the pervasively studied issue of noisy observations. The treatment of sensory anomalies in mobile robots with Bayesian networks is then addressed. The thesis contributes with a novel modeling paradigm, a so-called Bayesian sensory architecture, that enables the representation of any robotic sensory system, allowing the identification of anomalies and the recovery from them. The main drawback of this proposal is the potentially high computational cost of inference with Bayesian networks, which is addressed with a novel, approximate algorithm that leverages the structure of the proposed model. Both the sensory architecture and the corresponding inference algorithm are implemented for different robotic tasks, and are validated through different sets of both simulated and real experiments. One of the implementations is aimed at analyzing the performance of the proposed algorithm in terms of error and computation time. The results obtained from the experiments show that the cost of inference is significantly reduced, and that the approximate queries produced still serve to perform sensory diagnosis and recovery adequately. Another implementation is proposed for the problem of robotic navigation in human environments. In this case, the experimental results prove that the use of the architecture manages to increase the safety and efficiency of navigation. Lastly, a new inference approach based on the use of feedforward neural networks is implemented and tested for this problem, showing that it is possible to reduce, even more, the cost of inference with Bayesian networks, enabling real time operation.

Keywords: Bayesian networks inference, sensory systems, mobile robotics.

Resumen de la tesis

Contexto y ámbito de la tesis

La Robótica es una disciplina hoy en día considerada como uno de los mayores logros tecnológicos de todos los tiempos. A pesar de su complejidad, este campo interdisciplinario integrado por diversas áreas de la ciencia y la ingeniería surgió en realidad de una idea simple: la imitación y mejora de las capacidades de los seres humanos con el objetivo de ayudarlos o sustituirlos en tareas difíciles, repetitivas o peligrosas. Aunque se considera que la historia de la Robótica comienza en la antigüedad, no fue hasta el siglo veinte cuando comenzó realmente su desarrollo como la disciplina que se conoce en la actualidad. De hecho, el término *robot* apareció por primera vez en una obra teatral de ciencia ficción escrita por el dramaturgo checo Karel Čapek en 1921 [1]. Esta denominación procede del vocablo eslavo *robota*, que significa *esclavo* o *sirviente*. En la actualidad, los robots se emplean en una amplia variedad de contextos y aplicaciones, que no se restringen únicamente a tareas pesadas o peligrosas.

Una de las primeras implementaciones en ámbitos profesionales de la historia fue la instalación del robot industrial *Unimate* en una cadena de montaje de la empresa General Motors en 1961 [2]. Desde entonces, el empleo de estos robots, generalmente conocidos como *brazos manipuladores*, ha revolucionado el campo de la industria, permitiendo operaciones como soldadura, pintura o montaje, entre otras muchas [3]. Hoy en día, los manipuladores no solamente están presentes en entornos industriales, sino también en educativos [4] y médicos [5]. En estos últimos, por ejemplo, dichos robots suelen integrarse como parte de sistemas de mayor complejidad para la asistencia quirúrgica en intervenciones *mínimamente invasivas* (como las cirugías laparoscópicas) [6]. Aunque se emplean para resolver numerosos problemas, los robots industriales no son suficientes para replicar completamente las capacidades de un agente humano, y no son, por supuesto, los únicos que se han desarrollado a lo largo de la historia. Existe, además, una serie de plataformas robóticas conocidas con el nombre de *robots móviles*, ya que todos ellos tienen la habilidad de cambiar su localización en el entorno, y suelen clasificarse en función de dónde pueden operar y cómo. Los orígenes de estos robots están fundamentados principalmente en los logros científicos y tecnológicos producidos entre finales del siglo XIX y principios del XX, especialmente en campos como las Ciencias de la Computación o la Ingeniería de Control [1]. La primera plataforma móvil de la historia, conocida como *Shakey the robot* [2], fue desarrollada en el *Stanford Research Institute* a finales de la década de 1960. Consistía en un robot terrestre de interiores que incluía algunos dispositivos sensoriales (cámaras, parachoques y sensores de rango), además de estar dotado de razonamiento básico sobre sus acciones. Desarrollos como éste condujeron posteriormente a la introducción de muchos tipos de plataformas móviles. En el presente, estos robots no operan únicamente en medios terrestres, sino también aéreos y submarinos [7], [8]. Las aplicaciones de estas plataformas son innumerables, y están presentes en una amplia variedad de contextos. Los robots móviles terrestres, también llamados *Vehículos Terrestres no Tripulados* (UGVs, por sus siglas en inglés) pueden encontrarse, por

ejemplo, en ámbitos industriales como vehículos de transporte [9], en entornos domésticos como asistentes para personas mayores [10], en hospitales y centros de salud como parte de procesos terapéuticos [11], en colegios y universidades como dispositivos educativos [12] e incluso en tareas agrícolas [13]. En lo relativo a las plataformas móviles aéreas, conocidas como *Vehículos Aéreos no Tripulados* (UAVs en inglés), existen numerosas implementaciones en aplicaciones industriales, comerciales y militares [14], [15]. Estos robots se emplean, por ejemplo, en tareas de vigilancia [16], en misiones de búsqueda y rescate [17] o durante inspecciones industriales [18]. Por último, los robots móviles submarinos, llamados *Vehículos Autónomos Subacuáticos* (AUVs en inglés), se encuentran también implementados en aplicaciones muy diversas, algunas de ellas relacionadas con la industria costera [8], la investigación científica marina [19], el uso militar [20], etc. En la actualidad, los robots móviles no solamente se emplean de forma aislada, sino también en configuraciones *multi-robot* [21], [22], siendo implementados incluso en combinación con manipuladores [23] para ampliar sus capacidades y permitir así su uso en situaciones de mayor complejidad.

La mayoría de las aplicaciones robóticas mencionadas anteriormente requieren plataformas que operen con un cierto grado de autonomía, lo cual es particularmente difícil de conseguir para el caso de los robots móviles [1]. A diferencia de los manipuladores, estos robots pueden no contar con información previa sobre el entorno, su espacio de trabajo puede no estar restringido a un área o volumen pequeño y conocido en general, y pueden necesitar interactuar con agentes con comportamiento desconocido (por ejemplo, con personas, otros robots o máquinas, etc.). Además, las plataformas móviles deben incorporar una fuente de alimentación a bordo, por lo que tienen un tiempo de funcionamiento y unas capacidades de cálculo limitadas. Estas son algunas de las razones por las que los robots móviles no son tan comunes como los manipuladores en aplicaciones prácticas, aunque en los últimos años se han producido importantes avances. Para operar de forma autónoma, un robot móvil debe ser capaz de moverse adecuadamente dentro de su entorno, ya que no debe comprometer su integridad ni la de otros agentes presentes; además, debe ser capaz de percibir adecuadamente su entorno y de razonar de forma inteligente con toda esa información, incluida la de su propio estado, ya que esto es imprescindible para interactuar con el mundo físico y, por tanto, para llevar a cabo con éxito la tarea para la que el robot está concebido. Estos problemas han sido ampliamente tratados por la comunidad robótica, especialmente desde la segunda mitad del siglo XX hasta la actualidad [24]. Si bien se dice que algunos de ellos están razonablemente bien resueltos [1], otros siguen representando importantes retos.

Algunos de los problemas mencionados se consideran básicos para una operación autónoma, y se han identificado tradicionalmente en el ámbito de la Robótica Móvil como *navegación, localización y mapeado* [1]. El primero hace referencia a la ya mencionada necesidad de moverse dentro de un entorno determinado de forma segura y eficiente, y representa uno de los problemas considerados razonablemente bien resueltos para muchas situaciones prácticas. En particular, los avances producidos en la Ingeniería de Control y otras disciplinas afines durante el siglo XX han contribuido a dicho éxito [7], aunque la investigación robótica relacionada con la navegación sigue siendo intensa, y han sido numerosas las aportaciones en los últimos años dedicadas a su mejora, por ejemplo, en el ámbito de los entornos complejos en presencia de humanos [25], [26]. Moverse adecuadamente, en cualquier caso, no es suficiente para operar de forma autónoma, ya que un robot debe conocer su ubicación actual y la del objetivo deseado para planificar sus movimientos. La estimación de la *pose* real del robot (es decir, su posición y orientación) se estudia mediante el problema de localización, y se resuelve utilizando información sobre la representación del entorno. Por otro lado, la estimación de dicha información dada la localización del robot se aborda mediante el problema de mapeado. Estos dos problemas están relacionados de tal manera que ninguno de ellos puede ser abordado si no se resuelve previamente alguno de ellos. Por este motivo, también

se han tratado de forma concurrente, desde la perspectiva del conocido problema de *Localización y Mapeado Simultáneos* (SLAM, en inglés) [1], que permite estimar la localización del robot y la representación de su entorno al mismo tiempo en un escenario inicialmente inexplorado. Tratados por separado, ambos problemas de localización y mapeado se consideran resueltos satisfactoriamente en muchas situaciones del mundo real [27], gracias en parte a la incorporación de la probabilidad y la estadística a este campo de investigación a finales de los años 90. El tratamiento de la incertidumbre de la información disponible, implementado en forma de estimadores bayesianos, fue especialmente relevante, como también lo fue el problema de SLAM, que se benefició particularmente de dichos métodos probabilísticos a principios de los dos mil [27]. En la actualidad, el SLAM ha evolucionado significativamente con la mejora de la potencia computacional y la inclusión de dispositivos sensoriales más complejos (por ejemplo, sensores de visión 3D) [28]; sin embargo, todavía existen algunos retos importantes que se siguen estudiando [29].

El rendimiento de cualquier robot móvil en la realización de estas tareas básicas depende en gran medida de sus componentes físicos, es decir, de su aparato sensorial y motor, así como de sus capacidades computacionales [1]. El aparato motor hace referencia a todos aquellos actuadores presentes a bordo del robot que le permiten realizar movimientos, mientras que el sensorial se refiere a todos aquellos dispositivos que son capaces de medir cualquier aspecto relacionado con el estado del robot o el de su entorno. Por último, las capacidades computacionales se refieren a todos los sistemas computerizados disponibles, ya sea a bordo o en remoto, que procesan la información de los sensores del robot y producen los comandos necesarios para sus actuadores. La información sensorial es especialmente crítica para los problemas básicos de la Robótica Móvil. Por ejemplo, los errores que finalmente se obtengan en la localización y el mapeado no sólo dependerán del algoritmo empleado, sino también, y muy especialmente de la calidad de los datos sensoriales que se procesen. Esto también es crucial para muchos otros problemas relacionados con la robótica móvil. Por ejemplo, la detección de obstáculos dinámicos y la predicción de su movimiento [30] se basa principalmente en datos sensoriales de bajo nivel que luego se procesan para obtener representaciones significativas de dichos obstáculos, por ejemplo, en términos de *pose*, identidad, etc.

Considerar la incertidumbre en estos contextos surge como una necesidad natural, ya que todas las magnitudes medibles del mundo físico están sometidas a algún grado de imprevisibilidad [27]. Sin embargo, el mero tratamiento del ruido no basta para representar todos los posibles problemas que afectan a los dispositivos sensoriales. Los robots móviles operan a menudo en escenarios complejos en los que las condiciones ambientales representan un reto desde el punto de vista de la percepción. Dichas condiciones adversas pueden impedir que algunos sensores funcionen de forma nominal, lo que provoca un comportamiento indeseable. Como ejemplo, consideremos el caso de un robot móvil que navega cerca de superficies altamente reflectantes (por ejemplo, espejos) en un entorno de interiores. Todos los sensores de rango de a bordo basados en la radiación infrarroja (por ejemplo, telémetros láser, cámaras, etc.) probablemente fallarían en esta situación, ya que se sabe que las superficies especulares desvían la radiación infrarroja de tal manera que ésta no regresa al sensor, lo que conduce a una falsa detección de espacio libre [31]. Limitaciones sensoriales como la comentada no podrían ser captadas considerando únicamente un modelo de ruido. Por ello, sería deseable que un robot móvil razonara de forma inteligente sobre la existencia de condiciones adversas y que además tuviera la capacidad de superarlas considerando la información disponible a bordo (por ejemplo, datos de otros sensores, conocimiento del entorno o de expertos, etc.). Esta integración de diversas fuentes de conocimiento también tendría que tener en cuenta la incertidumbre intrínseca asociada a la información.

El problema expuesto anteriormente representa el objetivo principal de esta tesis, es decir,

dotar a un robot móvil de la capacidad de diagnosticar condiciones anómalas desde el punto de vista de su aparato sensorial, al tiempo que se mejora su robustez y se trata la incertidumbre de la información implicada en el proceso.

Motivación

El diagnóstico sensorial ha sido abordado desde el punto de vista de la Robótica en los últimos años, así como desde muchas otras perspectivas [32]. De hecho, este campo de investigación se considera dentro del ámbito del diagnóstico de sistemas físicos [33], que representa una disciplina más genérica. La identificación de situaciones anómalas en este contexto se ha resuelto mediante el uso de diferentes metodologías, la mayoría de ellas fundamentadas en técnicas surgidas de la estadística y la inteligencia artificial [34].

Los métodos existentes en el ámbito del diagnóstico sensorial se clasifican tradicionalmente en tres enfoques diferentes, a saber, *basado en modelo* (o *analítico*), *basado en datos* y *basado en conocimiento* [34]. Los métodos pertenecientes al primer enfoque construyen modelos matemáticos del comportamiento de los fallos en un sistema físico, que posteriormente se utilizan para detectar dichas situaciones anómalas. En Robótica, estos métodos aparecen, por ejemplo, en aplicaciones para brazos manipuladores en las que se combina la lógica difusa con la detección de fallos basada en observadores [35]. A diferencia de estos métodos, las técnicas del segundo enfoque se fundamentan principalmente en los datos disponibles para modelar e identificar condiciones anómalas. Las aplicaciones robóticas relacionadas suelen basarse en el análisis de componentes principales, las máquinas de vectores de soporte y los filtros de partículas [36]. Por último, las metodologías basadas en el conocimiento representan un enfoque híbrido que combina aspectos de los anteriores. Estas técnicas también se han aplicado para mejorar la robustez de los sistemas robóticos. Las aplicaciones típicas en este contexto suelen basarse en la representación de conocimiento experto mediante la combinación de la lógica difusa y las redes neuronales artificiales [37].

Aunque las soluciones existentes para el problema del diagnóstico sensorial han demostrado ser útiles, no usan, en general, un marco riguroso que permita una representación adecuada de la incertidumbre y que al mismo tiempo sea capaz de integrar de forma homogénea diversas fuentes de conocimiento. Por ejemplo, las redes neuronales artificiales no ofrecen explicaciones explícitas sobre sus deducciones, la lógica difusa es únicamente adecuada para representar el conocimiento experto, y muchos otros enfoques existentes, como las metodologías ad-hoc o heurísticas, no proporcionan resultados con garantías probabilísticas [38].

Por las razones expuestas, los paradigmas mencionados parecen no cumplir completamente los requisitos para el problema considerado en esta tesis. Afortunadamente, existen otras herramientas que pueden encontrarse en el ámbito de la inteligencia artificial. Desarrolladas en los años 80, las redes bayesianas [39] son un tipo de modelo gráfico probabilístico que permite representar el comportamiento de sistemas complejos, razonar y extraer conclusiones sobre sus componentes, y gestionar correctamente la incertidumbre. Desde su introducción, las redes bayesianas se han aplicado con éxito a una gran diversidad de campos, como la medicina, la economía, la agricultura, la ingeniería y muchos otros [33]. Algunas de estas aplicaciones están relacionadas con el diagnóstico sensorial en el contexto de la Robótica Móvil. Sin embargo, las implementaciones existentes son escasas [38] y no aprovechan del todo la flexibilidad que ofrecen estas redes, en el sentido de que suelen estar restringidas a situaciones muy concretas [40], [41].

Todas las limitaciones relacionadas con el diagnóstico sensorial explicadas anteriormente justifican el desarrollo de esta tesis. El objetivo principal consiste, por tanto, en la construcción de una arquitectura que permita una representación flexible de cualquier sistema sensorial robótico, independientemente de las características concretas de la plataforma móvil que se considere. Este problema también plantea algunos retos adicionales que deben ser abordados. Uno de los inconvenientes del razonamiento con redes bayesianas es su elevado coste computacional [33], que compromete seriamente las capacidades de tiempo real de los robots móviles. Incluso solucionando este problema, la mera definición de dicha representación de los sistemas sensoriales no bastaría para demostrar su utilidad; por lo tanto, la propuesta debería implementarse también en aplicaciones prácticas para evaluar sus beneficios. Entre otros, los retos mencionados resumen los problemas abordados durante el desarrollo de esta tesis.

Contribuciones

Las contribuciones presentadas en esta tesis han surgido de los problemas relacionados con la aplicación de las redes bayesianas para la representación, el diagnóstico y la mejora de los sistemas sensoriales robóticos. Uno de los primeros problemas que se ha abordado es precisamente el estudio del impacto de las anomalías sensoriales en el funcionamiento de los robots móviles, que va más allá del problema ampliamente tratado de las observaciones con ruido [1].

Como se ha explicado anteriormente, los robots móviles modernos suelen navegar en entornos complejos desde el punto de vista de sus sistemas sensoriales, lo que provoca observaciones anómalas (no sólo ruidosas) en determinadas condiciones. La corrupción de datos en estas situaciones no se debe a la imposibilidad de observar el verdadero estado de una magnitud física, sino a las limitaciones intrínsecas de los propios dispositivos sensoriales. Este problema podría afectar a tareas básicas como la localización o el mapeado, lo que podría comprometer el funcionamiento del robot. Dado que estos problemas básicos de la Robótica Móvil se abordan en su mayoría mediante el empleo de marcos probabilísticos como los filtros bayesianos [27], es importante evaluar hasta qué punto las observaciones anómalas afectan a estos filtros y, a su vez, a la propia operación robótica.

El trabajo [31], desarrollado en el ámbito de esta tesis, está precisamente orientado a estudiar el impacto de las observaciones anormales en el rendimiento de los filtros bayesianos. Existen, además, algunas contribuciones en la literatura que también tratan algunos aspectos de este problema. Desde un punto de vista analítico, trabajos como [42], [43] y [44], estudian la convergencia de algunos estimadores bayesianos; sin embargo, ninguno de ellos tiene en cuenta las posibles observaciones anómalas que aquí se tratan. Además, se limitan a implementaciones concretas de estos filtros y no cubren más aspectos de su funcionamiento. Por otro lado, trabajos como [45], [46] y [47], desarrollan estrategias para identificar y corregir las anomalías sensoriales, pero de nuevo carecen de un tratamiento completo de los diferentes aspectos que afectan al rendimiento de los filtros.

En esta tesis se tratan simultáneamente los dos problemas mencionados, es decir, la identificación de los tipos de datos sensoriales anómalos y el estudio de su efecto sobre diferentes aspectos del rendimiento de los estimadores bayesianos. A pesar de que se podrían utilizar muchos sensores para ello, tener en cuenta todos sería abrumador debido a la variedad de posibles efectos anómalos. El estudio se centra únicamente en los sensores de rango, que juegan un papel clave en una gran variedad de tareas en Robótica Móvil [1], abarcando así muchas situaciones. El marco probabilístico utilizado para implementar el problema de filtrado es el de las redes bayesianas dinámicas

[48], consideradas una extensión de las redes bayesianas. Este paradigma permite la representación de estructuras arbitrarias de procesos dinámicos estocásticos y también la inferencia genérica, lo que lo convierte en un candidato ideal para abstraer las implementaciones concretas de estimadores bayesianos como los filtros de Kalman o de partículas [1].

Un tratamiento analítico del problema de filtrado en este contexto sería engorroso, y posiblemente poco práctico, dado el gran número de parámetros que deberían considerarse para cubrir una variedad suficiente de situaciones anómalas (por ejemplo, las condiciones del problema, los parámetros de modelado del sensor, la cantidad y el valor de los datos sensoriales anómalos, etc.). Para resolver esta cuestión, el estudio se lleva a cabo desde un enfoque estadístico riguroso, basado en el análisis de la varianza (ANOVA) [49].

Existe una serie de aspectos que se tienen en cuenta antes de realizar el estudio y obtener conclusiones. Uno de ellos es la definición del problema de filtrado en el contexto de los sensores de rango, que se representa, por simplicidad, mediante un problema de seguimiento de obstáculos unidimensional. A continuación, se recopilan las situaciones más comunes que afectan a los sensores de rango. Una revisión exhaustiva de la literatura sobre estos dispositivos permite resumir los efectos de dichas condiciones anómalas en dos formas de corrupción de datos: las observaciones sesgadas y las perdidas o saturadas [31]. El rendimiento del filtrado en el contexto de este estudio se representa mediante indicadores tales como la precisión esperada del filtro, su incertidumbre esperada y el número mínimo de pasos necesario para la convergencia. El análisis que se lleva a cabo requiere también la definición de un conjunto de factores que potencialmente afecten a estos indicadores. En este caso, los que se consideran relevantes son la posición inicial del obstáculo en el problema de seguimiento, su velocidad y la cantidad de observaciones sesgadas y perdidas.

Con todas estas definiciones, se aplica la metodología estadística de análisis de la varianza a conjuntos de experimentos simulados, diseñados para cubrir una amplia variedad de condiciones en el problema considerado. Los resultados obtenidos proporcionan conclusiones completas y relevantes sobre los efectos de cada factor en el rendimiento del filtrado, que además se validan en un escenario real con un robot móvil. En resumen, los resultados de [31] demuestran que las observaciones anómalas mencionadas pueden tener un impacto importante en el rendimiento en determinadas situaciones, mientras que las condiciones del problema, es decir, la posición y la velocidad del obstáculo, no son relevantes.

El análisis de anomalías sensoriales llevado a cabo en [31] resuelve el primer problema abordado en esta tesis, ya que demuestra la importancia de considerar situaciones anómalas más allá de las observaciones ruidosas en el contexto del aparato sensorial de un robot móvil. Sin embargo, las anomalías tratadas no son las únicas comúnmente presentes en las plataformas móviles, y un mero estudio de las mismas no es suficiente para mejorar la robustez de los sistemas sensoriales robóticos. Por ello, el siguiente problema que debe abordarse está precisamente relacionado con la representación de dichos sistemas, con el fin de capacitar a los robots móviles para un diagnóstico y mejora sensorial inteligentes basado en la integración de diversas fuentes de conocimiento. Este problema representa el núcleo de esta tesis, y se aborda en sus trabajos relacionados ([50], [51], [38], [33] y [52]). Cada una de estas aportaciones, consideradas en orden cronológico, aporta una solución que mejora o complementa las anteriores en algún aspecto del problema.

Este proceso evolutivo comienza con los trabajos [50] y [51]. Su objetivo principal es proporcionar un marco de modelado, basado en redes bayesianas, que permita deducir anomalías sensoriales de bajo nivel y recuperación de las mismas. La definición de esta propuesta se basa,

en realidad, en un trabajo previamente coescrito por el doctorando [53] donde se abordan los fundamentos relativos a la representación de los sistemas sensoriales robóticos. La propuesta allí introducida se fundamenta en un elemento básico, el denominado *sensor bayesiano*, definido como una red bayesiana que pretende representar el comportamiento de un único sensor físico, así como cualquier otra información relevante relacionada con el mismo. En este trabajo se define una representación completa del sistema sensorial de un robot real dotado de dispositivos básicos (por ejemplo, detectores de precipicio, giroscopios, codificadores de ruedas, etc.). Dicha representación se obtiene combinando tantos sensores bayesianos como es necesario, dando lugar a un modelo único y monolítico de todo el sistema. Este modelo se evalúa mediante diferentes experimentos en un entorno real, demostrando su utilidad para el diagnóstico y la mejora sensorial. En los trabajos [50] y [51], esta definición de sensor bayesiano se completa para posibilitar la integración de sensores más complejos, como telémetros bidimensionales y cámaras RGB-D, demostrando, de nuevo, las capacidades del modelo resultante a través de diversos experimentos reales. Asimismo, el trabajo [51] contribuye en este contexto con la definición de un procedimiento para la construcción interactiva de sistemas sensoriales, basado en los dispositivos mencionados. Todos estos trabajos presentan un modelo estático que no integra información temporal de ningún tipo. Este problema se aborda en [38] con la aplicación de filtros temporales a los resultados de inferencia producidos por el modelo propuesto.

Otro de los inconvenientes que presentan estos primeros trabajos es el alto coste computacional de la inferencia con redes bayesianas. Las implementaciones de estas propuestas se basan principalmente en un algoritmo de inferencia exacta, el *jointree* o *junction tree* [54], [55], cuyo coste crece exponencialmente con la complejidad estructural de la red bayesiana empleada. Este y otros muchos problemas relacionados se abordan en [33], uno de los trabajos más relevantes de esta tesis. La primera cuestión tratada allí está relacionada con la definición de sensor bayesiano, que se revisa de nuevo para capturar el comportamiento de los sensores físicos de una forma más natural y también para permitir una representación adecuada para una inferencia más eficiente. Precisamente, una revisión exhaustiva de la literatura sobre algoritmos de inferencia inspira la principal aportación de este trabajo, consistente en un novedoso algoritmo de inferencia que aprovecha las topologías particulares de los sistemas sensoriales. Un modelo monolítico de estos sistemas puede ser potencialmente grande en general, impidiendo incluso inferencia aproximada en situaciones de tiempo real; para solucionarlo, este trabajo propone una nueva arquitectura, un modelo denominado *layered*, que divide estratégicamente el monolítico en diferentes *niveles* o *layers* aislados, en los que se distribuyen los sensores bayesianos existentes.

Sin embargo, aparece un problema que debe ser resuelto antes de la aplicación de dicha partición jerárquica al modelo monolítico del sistema. Las relaciones causales entre los componentes de un sistema sensorial son potencialmente numerosas y complejas. Esto se debe, en parte, a que las dependencias entre dos sensores diferentes pueden ser cíclicas o no cíclicas [33]. En el segundo caso, el comportamiento de un sensor está influenciado por otro y, en el primero, esta influencia es mutua. Dado que las redes bayesianas no están concebidas para codificar estas dependencias cíclicas, su tratamiento podría impedir la inferencia. Este trabajo estudia la distribución de probabilidad conjunta inducida por una red bayesiana cíclica y establece algunas condiciones para su existencia y unicidad. Bajo tales condiciones, se propone una solución aproximada que consiste en convertir las dependencias cíclicas en no cíclicas rompiendo las primeras. A continuación, el trabajo define un procedimiento para la construcción del modelo por niveles, que establece una ordenación jerárquica de los sensores en la que los inferiores explican o afectan al comportamiento de los superiores. Dicha propuesta se basa en el conjunto de relaciones existentes entre los sensores de una plataforma robótica, y también implementa la solución al problema de las dependencias

cíclicas.

Estas soluciones conducen a la introducción del novedoso método mencionado anteriormente, es decir, el algoritmo *layered*, que aplica el *jointree* a cada nivel del modelo *layered* secuencialmente. El método propuesto es en realidad un algoritmo aproximado, ya que emplea inferencia exacta sobre un modelo aproximado. La calidad de los resultados producidos por el algoritmo *layered* se evalúa mediante diferentes experimentos simulados y reales en un contexto robótico concreto. Asimismo, se compara el coste computacional de este método con el del *jointree* exacto y otros métodos aproximados conocidos, utilizando para ello la misma configuración experimental. Los resultados obtenidos demuestran que el método propuesto reduce significativamente el coste de inferencia y que además mejora la robustez del sistema sensorial considerado.

Tanto la arquitectura sensorial como el algoritmo de inferencia presentados en [33], aunque útiles y eficaces, tienen algunas limitaciones. Por ejemplo, las propuestas mencionadas sólo manejan redes bayesianas estáticas y discretas, y no permiten la representación de información sensorial más abstracta. Estas carencias se abordan en el último trabajo de esta tesis [52], que también aborda el problema de implementar la arquitectura propuesta en una aplicación robótica real. En este trabajo se revisa de nuevo la definición de sensor bayesiano para aumentar sus capacidades en cuanto a la representación de sistemas sensoriales genéricos. En particular, dicha definición se extiende para permitir el uso de variables aleatorias tanto continuas como discretas, para lograr un tratamiento más formal de la dinámica sensorial y para permitir la representación simultánea de diferentes niveles de abstracción cognitiva. Para resolver las dos primeras cuestiones, este trabajo redefine el sensor bayesiano como una red bayesiana híbrida y dinámica, y también adapta el modelo de niveles y el algoritmo de inferencia correspondiente para esta nueva propuesta. En el contexto de los sistemas sensoriales, también es necesario manejar diferentes niveles de abstracción, ya que muchos de estos sistemas están integrados por componentes heterogéneos que pueden producir información relacionada con diferentes ontologías al mismo tiempo (por ejemplo, un sistema sensorial que estima simultáneamente la *pose* de un obstáculo en movimiento y lo clasifica en alguna categoría de objetos). Tratar estas fuentes de información en la misma red conduciría a la inducción de dependencias no deseadas y, por tanto, a resultados de inferencia poco fiables o inexactos. Esta situación se resuelve tratando los sensores implicados en redes separadas, lo que se tiene en cuenta en la nueva definición de sensor bayesiano y en el método de inferencia.

Teniendo en cuenta lo anterior, la arquitectura resultante es un modelo tridimensional que incluye un eje de niveles para la representación ordenada de los sensores según sus dependencias mutuas, un eje temporal para la codificación de la dinámica temporal, y un eje cognitivo para el tratamiento simultáneo de diferentes ontologías de información sensorial. La otra aportación relevante de este trabajo es la implementación de la nueva propuesta para una aplicación robótica real. En particular, la metodología propuesta se utiliza para mejorar la robustez de un sistema de detección de peatones y de navegación en entornos con humanos. La arquitectura bayesiana se construye considerando diferentes ontologías de la información disponible sobre los peatones detectados, como su *pose*, velocidad o identidad y también teniendo en cuenta información del entorno, como un mapa de la escena. Esta implementación se pone a prueba en experimentos simulados y reales. Los resultados obtenidos muestran que la incorporación de la arquitectura al sistema existente es capaz de mejorar la robustez de la detección de peatones, así como la eficiencia y seguridad de la navegación.

Finalmente, este trabajo se complementa con una última tarea desarrollada durante la tesis. El coste computacional de la inferencia en la mencionada aplicación de navegación en entornos donde

hay personas no impide la operación en tiempo real, sin embargo, crece significativamente con el número de peatones detectados, limitando dicha operación en determinadas situaciones. Para solucionarlo se utiliza una aproximación basada en redes neuronales artificiales que reducen aún más el coste de inferencia. La propuesta pretende aproximar la inferencia bayesiana en cada nivel del modelo por niveles construyendo una red neuronal *feedforward* [56] que represente dicha inferencia para cada una de las posibles consultas, es decir, la red neuronal es el resultado de “compilar” la red bayesiana, utilizando terminología de Ciencias de la Computación. Estas funciones neuronales se entrenan utilizando conjuntos de datos obtenidos de situaciones simuladas. La solución final se prueba para el mismo problema de detección de peatones, y los resultados muestran que el coste global puede reducirse en varios órdenes de magnitud; además, el error con respecto a la inferencia exacta es, en general, lo suficientemente bajo como para no afectar al rendimiento del sistema.

En este punto ya se han presentado todas las contribuciones de esta tesis, que pueden resumirse en las siguientes:

- Un estudio estadístico riguroso acerca de la influencia de las observaciones sensoriales anormales en el rendimiento de los estimadores bayesianos, en el contexto de la robótica móvil.
- La definición de una arquitectura de modelado, basada en redes bayesianas, concebida para dotar a los robots móviles de detección inteligente de situaciones sensoriales anómalas y la recuperación de las mismas, lo que es posible gracias a la integración flexible de fuentes de conocimiento heterogéneas que permite la metodología propuesta.
- Un nuevo algoritmo de inferencia en redes bayesianas que aprovecha las topologías concretas de los sistemas sensoriales robóticos para reducir significativamente el coste.
- La implementación y validación de la metodología propuesta en aplicaciones robóticas reales.

Marco de la tesis y actividades de investigación

La presente tesis es el resultado de cuatro años de trabajo como estudiante del programa de Doctorado en Ingeniería Mecatrónica de la Universidad de Málaga. El doctorando comenzó su trabajo en el año 2017, en el cual le fue concedida una ayuda de ámbito nacional del Ministerio de Educación del Gobierno de España, perteneciente al programa de *Formación de Profesorado Universitario* (FPU). El autor de la tesis ha desarrollado su labor investigadora en el Departamento de Ingeniería de Sistemas y Automática de la Universidad de Málaga, bajo la supervisión de los catedráticos D. Juan Antonio Fernández Madrigal y D. Alfonso García Cerezo. El trabajo doctoral se ha realizado en el contexto de los proyectos de investigación nacional *FIRST-ROB* y *TRUST-ROB*, ambos coordinados por el catedrático D. Alfonso García Cerezo, centrados en el diseño e implementación de sistemas multi-robot para la asistencia a víctimas humanas en tareas de rescate.

El doctorando ha llevado a cabo, además, una serie de actividades relacionadas con su investigación. En particular, ha asistido a diferentes congresos, tanto nacionales como internacionales, tales como las *XXXIX Jornadas de Automática*, el congreso *Fourth Iberian Robotics Conference* (ROBOT 2019) y el evento *European Robotics Forum* (ERF 2020). El doctorando también ha realizado una estancia de tres meses en la Universidad de Luxemburgo, entre julio y septiembre de 2019, en la que tuvo ocasión de colaborar con otros investigadores en el campo de la navegación con robots móviles. Durante los últimos tres años del período de tesis, el doctorando ha colaborado en tareas docentes en asignaturas de grado como *Control por Computador* (2018-2020),

Programación de Robots (2018-2021) y *Robótica* (2018-2021). La cantidad de experiencia docente adquirida equivale a un total de 18 créditos ECTS (European Credit Transfer System). Asimismo, el doctorando ha asistido a distintos seminarios sobre temas como el empleo de la probabilidad y la estadística en Robótica Móvil o como las aplicaciones del aprendizaje profundo.

Publicaciones

El trabajo desarrollado durante esta tesis ha dado lugar a las siguientes publicaciones (enumeradas por categoría, y en orden cronológico inverso):

- **Revistas indexadas en el JCR**

- Manuel Castellano-Quero, Juan-Antonio Fernández-Madrugal, Alfonso García-Cerezo. *Improving Bayesian inference efficiency for sensory anomaly detection and recovery in mobile robots*. *Expert Systems with Applications*, vol. 163, pp. 113755, 2021, (Q1/T1).
- Manuel Castellano-Quero, Juan-Antonio Fernández-Madrugal, Alfonso García-Cerezo. *Statistical Study of the Performance of Recursive Bayesian Filters with Abnormal Observations from Range Sensors*. *MDPI Sensors*, vol. 20, no. 15, pp. 41-59, 2020, (Q1/T1).

- **Congresos internacionales**

- Manuel Castellano-Quero, Juan-Antonio Fernández-Madrugal, Alfonso García-Cerezo. *Integrating multiple sources of knowledge for the intelligent detection of anomalous sensory data in a mobile robot*. *Fourth Iberian Robotics Conference (ROBOT 2019)*, Porto (Portugal), pp. 159-170, 2019.

- **Congresos nacionales**

- Manuel Castellano-Quero, Juan-Antonio Fernández-Madrugal, Alfonso García-Cerezo. *Hacia la diagnosis y recuperación de sensores robóticos a bajo nivel mediante inferencia en redes bayesianas*. *XXXIX Jornadas de Automática*, Badajoz (Spain), 2018.
- Manuel Castellano-Quero, Juan-Antonio Fernández-Madrugal, Alfonso García-Cerezo. *Interactive Construction of Bayesian Inference Networks for Robust Robot Sensorics*. *XIV Simposio CEA de Control Inteligente (SCI 2018)*, Málaga (Spain), 2018.
- Manuel Castellano-Quero, Iván Fernández-Vega, Juan-Antonio Fernández-Madrugal, Ana Cruz-Martín. *Métodos de inferencia bayesiana empotrados para el diagnóstico y mejora sensoriales de un robot móvil*. *II Jornadas de Computación Empotrada y Reconfigurable (JCER2017)*, Málaga, Spain, pp. 667-676, 2017.

Estructura y contenido de la tesis

La presente tesis doctoral está escrita en inglés, y se organiza en varios capítulos, los cuales presentan una exposición ordenada y exhaustiva de todas las aportaciones relacionadas con el trabajo desarrollado durante la misma. A continuación se detalla el contenido de cada capítulo.

- El **capítulo 1** desarrolla una introducción a la tesis similar al contenido del presente resumen (salvo que éste se encuentra escrito en inglés).

- El **capítulo 2** proporciona una revisión básica de las nociones probabilísticas y estadísticas necesarias para entender la metodología aplicada en los capítulos centrales de la tesis.
- El **capítulo 3** analiza la influencia del comportamiento sensorial anómalo en el funcionamiento de los robots móviles. En particular, se presenta un estudio estadístico riguroso de dicha influencia en el rendimiento de los filtros bayesianos, dada la importancia de marcos probabilísticos como estos en las tareas robóticas básicas.
- El **capítulo 4** presenta la propuesta central de esta tesis, es decir, el uso de redes bayesianas para la representación de sistemas sensoriales robóticos, cuyo objetivo es dotar a los robots móviles de diagnóstico y mejora sensorial inteligente. En el capítulo se define una arquitectura novedosa basada en esta representación, junto con un nuevo algoritmo que aprovecha la estructura del modelo para aumentar la eficiencia de la inferencia.
- El **capítulo 5** aborda la implementación y validación de la metodología propuesta en diferentes aplicaciones robóticas reales. En el capítulo, se evalúa el rendimiento del nuevo algoritmo de inferencia, así como la utilidad de la arquitectura propuesta en dichos contextos reales.
- Por último, el **capítulo 6** extrae conclusiones relevantes de las principales aportaciones de esta tesis, y propone diferentes líneas de investigación para futuros trabajos. Dichas observaciones finales también se exponen a continuación (en español).

Conclusiones

El trabajo presentado en esta tesis ha contribuido a diferentes aspectos tanto de la Robótica como de la Inteligencia Artificial, todos ellos relacionados, en particular, con la aplicación de las redes bayesianas al problema del diagnóstico y la recuperación sensorial en robots móviles. La comprensión de los sistemas sensoriales robóticos y de sus anomalías más comunes ha sido esencial en el proceso de investigación, que precisamente se inició con el estudio del impacto de dichas condiciones anómalas en el rendimiento de los robots móviles. Este estudio representa una de las primeras contribuciones de la tesis.

Llevar a cabo un análisis completo del impacto del comportamiento anómalo de los sensores en los robots móviles sería abrumador, dada la gran variedad de sensores comúnmente empleados en las plataformas móviles. Por ello, un análisis factible debería centrarse únicamente en aquellos dispositivos sensoriales considerados críticos para el funcionamiento básico de los robots. Muchas de las tareas esenciales que realiza un robot móvil (por ejemplo, localización, mapeado, navegación, etc.) dependen del uso de sensores de rango, ya que éstos proporcionan un conocimiento métrico del entorno que es crucial para las tareas mencionadas. La información de rango es empleada por las metodologías que soportan estas operaciones básicas, las cuales, a menudo, están basadas en marcos probabilísticos como los estimadores bayesianos.

El estudio presentado en esta tesis ha tenido como objetivo, precisamente, el análisis de los efectos de las observaciones de rango anómalas en el rendimiento de los filtros bayesianos y, a su vez, en la propia operación robótica. Para ello, el estudio ha abordado la inferencia de los filtros bayesianos desde una perspectiva genérica, utilizando el paradigma de las redes bayesianas dinámicas. Este filtro bayesiano genérico se ha modelado teniendo en cuenta las características de los telémetros robóticos más comunes. También se han analizado sus principales limitaciones, así como aquellos factores que pueden afectar al rendimiento del filtro. Se han diseñado diferentes

experimentos simulados con diversas condiciones, y se han obtenido conclusiones novedosas y relevantes a partir de su utilización con métodos estadísticos rigurosos. Asimismo, estas conclusiones se han validado en una situación real con un robot móvil.

Los resultados obtenidos muestran que los parámetros del problema de seguimiento considerados para el estudio (es decir, la velocidad y la posición inicial del obstáculo) no tienen ninguna relación con el rendimiento de los filtros bayesianos. Por el contrario, el aumento de la cantidad de datos sensoriales anómalos, es decir, las observaciones perdidas y sesgadas, suele afectar negativamente a todas las medidas de rendimiento consideradas. La combinación de ambos tipos de datos anómalos empeora la precisión esperada del filtro, mientras que sólo las observaciones perdidas son capaces de aumentar la incertidumbre del filtrado. Por último, una de las conclusiones que no se esperaba antes de realizar los análisis estadísticos es que la convergencia se ve seriamente afectada por ambos tipos de observaciones anómalas por separado, y que su combinación no conduce a una peor tasa de convergencia en caso de existir alguna de las anomalías mencionadas.

El mero análisis del impacto de los comportamientos sensoriales anómalos no es el único objetivo de esta tesis, ya que también se han abordado estas anomalías, mediante diferentes y novedosas metodologías, con el fin de eliminarlas o mitigarlas. Para ello, la tesis ha presentado un marco basado en redes bayesianas, llamado arquitectura sensorial bayesiana, que permite la representación de cualquier tipo de sistema sensorial para robots móviles y su uso en inferencia inteligente. En esencia, el objetivo de estos modelos es contribuir a un funcionamiento sensorial más robusto y fiable. Esto se consigue codificando el mayor número posible de interacciones entre sensores (tanto cíclicas como no cíclicas), lo que permite detectar posibles situaciones de fallo que impidan a los sensores percibir adecuadamente su entorno y el estado del robot. Además, el proceso de modelado permite la integración inteligente de fuentes de información heterogéneas (datos sensoriales y ambientales, sentido común humano, etc.), incluso considerando aspectos dinámicos y diferentes niveles de abstracción cognitiva. Todo este conocimiento se tiene en cuenta para recuperar el estado correcto de la información sensorial que se representa, es decir, el que debería haberse obtenido en un escenario sin anomalías.

El principal inconveniente de esta propuesta es el elevado coste computacional de los métodos de inferencia exacta y aproximada existentes para redes bayesianas, lo que impide su implementación para aplicaciones con requisitos de tiempo real. Por este motivo, la tesis también ha aportado un nuevo algoritmo que realiza inferencia exacta en un modelo aproximado tridimensional del sistema sensorial robótico, que ha sido definido también para ser utilizado por el método propuesto. Para validar tanto el modelo como el algoritmo de inferencia, se han realizado diferentes estudios simulados y experimentales. Los resultados obtenidos muestran que el método propuesto consigue una considerable reducción del tiempo de inferencia, que es especialmente notable a medida que aumenta el tamaño del modelo. Además, proporciona resultados de inferencia que son útiles en la mayoría de los casos para llevar a cabo detección y recuperación de anomalías sensoriales, tal y como se ha demostrado en los experimentos con el robot real.

El marco de modelado propuesto también se ha implementado en una aplicación robótica real para demostrar su utilidad. En particular, se ha implementado para el problema de la navegación robótica en entornos humanos. El modelo propuesto pretende en este caso mejorar la robustez de un sistema de detección y seguimiento de peatones, con objeto de conseguir una navegación robótica más segura y eficiente en entornos con presencia humana. Esto se ha demostrado con éxito mediante la validación experimental en un conjunto de experimentos tanto simulados como reales. Por último, también se ha desarrollado una propuesta basada en el uso de redes neuronales

feedforward para aumentar aún más la eficiencia de la inferencia. Esta aproximación de inferencia se ha implementado también para el problema de la navegación en entornos humanos, demostrando que es posible realizar los mismos razonamientos que ofrece la arquitectura bayesiana propuesta en tiempo real, incluso cuando el número de peatones detectados es considerable.

Trabajo futuro

Existen diferentes tareas que podrían desarrollarse para futuros trabajos, todas ellas relacionadas con las aportaciones presentadas en la tesis. Uno de estos trabajos consiste en el estudio del impacto del comportamiento sensorial anómalo. Las conclusiones derivadas de dicho estudio presentadas en esta tesis se basan actualmente en un conjunto de factores que pueden ser ampliados. Por ejemplo, esto puede hacerse para incluir una mayor variedad de sensores robóticos, desde dispositivos básicos (giroscopios, codificadores, etc.) hasta otros más complejos (sensores de visión, cámaras térmicas, etc.). Además, podrían considerarse más factores para tener en cuenta una mayor cantidad de parámetros de filtrado, así como los modos de funcionamiento de los robots. Se estudiaría también el impacto de las variaciones en todos los aspectos mencionados en relación con el rendimiento de los estimadores bayesianos. Sin embargo, también podría estudiarse dicho rendimiento en el ámbito de modelos más generales de estimación bayesiana, como los modelos híbridos tales como el Filtro de Kalman Conmutado [57], que también puede ser implementado en el marco proporcionado por las redes bayesianas dinámicas.

En cuanto a la arquitectura sensorial bayesiana propuesta y al correspondiente algoritmo de inferencia, también hay diferentes tareas que pueden abordarse en el futuro. El modelo y el algoritmo propuestos han sido validados en un robot móvil real, sin embargo, deben ser implementados para una amplia variedad de plataformas robóticas (aéreas, terrestres y submarinas) que utilicen dispositivos sensoriales más complejos, con el fin de completar la validación. En cuanto al coste computacional, se ha conseguido mejorar la eficiencia de la inferencia con el modelo propuesto, haciéndolo adecuado incluso para aplicaciones robóticas en tiempo real. Sin embargo, el uso de redes neuronales *feedforward* para ello depende de un procedimiento de entrenamiento previo que no puede generalizarse fácilmente para cada modelo sensorial, por lo que podrían considerarse otros métodos de recopilación de redes bayesianas como el de los *circuitos aritméticos* [58].

También habría que modificar el proceso de modelado para que se realice de forma más automática y autónoma, manteniendo su capacidad de reflejar el conocimiento humano así como el procedente de otras fuentes de información heterogéneas. Para ello, sería interesante explorar los algoritmos existentes para el aprendizaje de parámetros y estructuras en el contexto de las redes bayesianas.

La aplicación de la arquitectura bayesiana propuesta a problemas robóticos concretos también ha tenido éxito; sin embargo, sería necesario estudiar más a fondo algunos aspectos relativos a las implementaciones actuales, que, por supuesto, podrían ampliarse también a una mayor variedad de tareas. Una de las implementaciones propuestas en la tesis está concebida para el problema de la navegación en entornos humanos. Aunque se han realizado varias pruebas simuladas y reales para demostrar la utilidad del enfoque propuesto, estos experimentos deberían realizarse también en escenarios más concurridos y en una mayor variedad de ellos. Por último, también sería de interés estudiar hasta qué punto la integración del conocimiento humano a diferentes niveles de abstracción consigue mejorar la robustez de los algoritmos existentes relacionados con la detección y seguimiento de peatones.

Introduction

1.1 Context and scope of the thesis

Robotics is considered among the most remarkable technological developments of all time. Despite its complexity, this interdisciplinary field involving a number of areas of science and engineering actually arised from a simple idea: the imitation and enhancement of the capabilities of human beings for substituting or aiding them in difficult, repetitive or dangerous tasks. Although the history of robotics is considered to begin in the ancient times, it was not until the twentieth century that the development of this discipline as known these days truly started. In fact, the term *robot* appeared for the first time in a science-fiction play written by the Czech author Karel Čapek in 1921 [1]. This denomination comes from the Slavic word *robota*, and it stands for *slave* or *servant*. Currently, robots are employed in a wide variety of contexts and applications, which do not restrict only to heavy or perilous tasks only.

One of the first successful deployments in professional purposes in history was the installation of the *Unimate* industrial robot at a General Motors assembly line in 1961 [2]. Since then, the use of such robots, usually referred to as *manipulator arms*, has revolutionized industry, allowing operations such as welding, painting or assembly, among many others [3]. Nowadays, manipulators are not only deployed in industrial environments, but also in educational [4] and medical ones [5]. In the latter, for instance, these robots are usually integrated as part of more complex systems in order to assist surgeons during *minimally invasive* interventions (e.g., laparoscopic surgeries) [6]. Although employed to solve many problems, industrial robots do not suffice to completely replicate the capabilities of a human agent, and are not, of course, the only ones developed throughout history. There also exist a number of robotic platforms known with the name of *mobile robots*, since all of them have the ability of changing their location within the environment, and are usually classified depending on where they can operate and how. The origins of these robots are grounded mainly on scientific and technological achievements produced in the mid twentieth century, specially in fields such as computer science or control engineering [1]. The first autonomous mobile platform *Shakey the robot* [2] was developed at the Standford Research Institute in the late 1960s. It consisted in a terrestrial indoor robot that included some sensory devices (cameras, bumpers and rangefinders) and basic reasoning about its actions. Developments like this one lead to the

1.1 Context and scope of the thesis

introduction of many kinds of mobile platforms later on. In the present, these robots do not only operate within terrestrial environments, but also in aerial and submarine ones [7], [8]. The applications of these platforms are countless, being present in a wide variety of contexts. Terrestrial mobile robots, also called *Unmanned Ground Vehicles* (UGVs), can be found, for instance, in industrial settings as automated transportation vehicles [9], in domestic environments as assistants for elder people [10], in hospitals and health centres as part of therapeutical processes [11], in schools and universities as educational devices [12] and even in agricultural tasks [13]. Regarding aerial mobile platforms, known as *Unmanned Aerial Vehicles* (UAVs), there exist numerous implementations related to industrial, commercial and military applications [14], [15]. These robots are used, for instance, in surveillance tasks [16], in search and rescue missions [17] or during industrial inspections [18]. Lastly, submarine mobile robots, called *Autonomous Underwater Vehicles* (AUVs), are also deployed in many different applications, some of them related to the offshore industry [8], to marine scientific research [19], to military purposes [20], etc. Currently, mobile platforms are not only used in isolation, but also in *multi-robot* settings [21], [22], being implemented even in combination with manipulators [23] to expand their capabilities and thus enabling their use in more complex situations.

The majority of robotic applications mentioned above require platforms that operate with a certain degree of autonomy, which is particularly difficult to achieve for the case of mobile platforms [1]. In contrast to manipulators, these robots may not rely on prior information about the environment, their workspace may not be restricted to a small, known area or volume in general, and they may need to interact with agents with unknown behavior (e.g., people, other robots or machines, etc.). Also, mobile platforms must incorporate an on-board power supply, thus having more limited time of operation and computation capabilities. These are some of the reasons behind the fact that mobile robots are not as common as manipulators in practical applications, although important advances have been produced in the recent years, as mentioned. In order to operate autonomously, a mobile robot should be able to properly move within its environment, since it is not supposed to compromise its integrity or the one of other agents present; also, it should be able to adequately perceive its surroundings and to intelligently reason about all that information, including the one about its own state, since this is necessary to interact with the physical world and therefore to successfully complete the task for which the robot is conceived. These problems have been extensively treated by the robotic community, specially from the second half of the twentieth century up to the present [24]. While some of them are said to be reasonably well solved [1], some others still represent important challenges nowadays.

Some of the mentioned problems are considered basic for an autonomous operation, and have been traditionally identified in the scope of mobile robotics as *navigation*, *localization* and *mapping* [1]. The first one refers to the already mentioned necessity of moving within a particular environment in a safe and efficient manner, and it represents one of those problems considered well solved for many practical situations. In particular, the advances produced in control engineering and other related disciplines during the twentieth century have contributed to such successful development [7]. Nevertheless, robotic research related to navigation is still intense, and there have been numerous contributions in the last years dedicated to its enhancement, for instance, in the scope of complex environments in the presence of humans [25], [26]. Unfortunately, moving adequately does not suffice to operate autonomously, since a robot must be aware of its current location and the one of the desired target in order to plan its movements. The estimation of the actual *pose* of the robot (i.e., its position and orientation) is studied by the localization problem, and it is solved by using information about the representation of the environment. On the other hand, the estimation of such information given the location of the robot is addressed by the problem of

1.1 Context and scope of the thesis

mapping. These two problems are related in such a way that none of them can be addressed unless the other is previously solved. For that reason, they have also been treated concurrently, from the perspective of the well-known *Simultaneous Localization and Mapping* (SLAM) problem [1], which allows the estimation of the location of the robot and the representation of its environment at the same time within an initially unexplored scenario. If treated separately, both localization and mapping problems are considered to be satisfactorily solved in many real world situations [27], thanks in part to the incorporation of probability and statistics to this field of research during the late 1990s. The treatment of the uncertainty of the available information, implemented in the form of Bayesian estimators, was particularly relevant, and that was also the case for the SLAM problem, which benefited specially from such probabilistic methods in the early two thousands [27]. Nowadays, SLAM has evolved significantly with the improvement of its computational efficiency and the inclusion of more powerful sensory devices (e.g., computer vision and 3D sensors) [28]; however, there still exist some relevant challenges that are currently under study [29].

The performance of any mobile robot in accomplishing these basic tasks strongly depends on its physical components, namely, its sensory and motor apparatus as well as its computational capabilities [1]. The motor apparatus refers to all those actuators present on-board the robot that enable it to perform movements, while the sensory one refers to all those devices that are able to measure any aspect related to the state of the robot or the one of its environment. Lastly, the computational capabilities refer to all the computer-based systems available, either on-board or remotely, that process all the information from the robot sensors and produce the necessary commands for its actuators. Sensory information is particularly critical for the basic problems of mobile robotics. As an example, the errors finally obtained in localization and mapping will not only depend on the employed algorithm, but also and much more specially, on the quality of the sensory data being processed. This is also crucial to many other problems related to mobile robotics. For example, the detection of dynamic obstacles and the prediction of their motion [30] relies mainly on low-level sensory data that are then processed to obtain meaningful representations of such obstacles, e.g., in terms of pose, identity, etc.

Considering uncertainty in these contexts arises as a natural need, since all the measurable quantities of the physical world are subjected to some degree of unpredictability [27]. However, the mere treatment of noise does not suffice to capture all the possible problems affecting sensory devices. Mobile robots often operate within complex scenarios where the environmental conditions represent a challenge from the point of view of perception. Such adverse conditions could prevent some sensors from working nominally, leading to undesirable behavior. As an example, consider the case of a mobile robot navigating nearby highly specular surfaces (e.g., mirrors) within an indoor scenario. All the on-board range sensors based on infrared radiation (e.g., laser rangefinders, cameras, etc.) would be likely to fail in this situation, since specular surfaces are known to deviate infrared radiation in such a way that it is not reflected back to the sensor, leading to a false detection of free space [31]. Sensory limitations like the one discussed could not be captured by just considering a noise model. Thus, it would be desirable for a mobile robot to intelligently reason about the existence of sensory adverse conditions and also to have the ability of overcoming them by considering the on-board available information (e.g., data from other sensors, environmental or expert knowledge, etc.). Such integration of diverse sources of knowledge would also have to take into account the intrinsic uncertainty associated with the information.

The problem discussed above represents the main aim of this thesis, i.e., to provide a mobile robot with the ability of diagnosing abnormal conditions from the point of view of its sensory apparatus while enhancing its robustness and handling the uncertainty of the information involved

1.2 Motivation

in the process.

The rest of this introductory chapter is organized as follows. In section 1.2, the reasons that motivate the development of the thesis are discussed, taking into account the existing related works as well. In section 1.3, a complete summary of the contributions of this thesis is presented. In section 1.4, the framework under which the thesis has been developed is described along with the research activities carried out by the doctoral candidate. A complete list of the publications produced is then provided in section 1.5. Finally, the thesis outline is detailed in section 1.6.

1.2 Motivation

Sensory diagnosis has been addressed from the point of view of robotics in the recent years, as well as from many other perspectives [32]. In fact, this field of research is considered within the scope of diagnosis of physical systems [33], which represents a more generic discipline. Identifying abnormal situations in this context has been solved by using different methodologies, most of them grounded on techniques arising from statistics and artificial intelligence [34].

Concerning the diagnosis problem, existing methods are traditionally classified into three different approaches, namely, *model-based* (or *analytical*), *data-driven* and *knowledge-based* [34]. The methods belonging to the first approach build mathematical models of the behavior of failures in a physical system, and are then used to detect such anomalous situations. In robotics, these methods appear, for instance, in applications for manipulator arms in which fuzzy logic is combined with observer-based fault detection [35]. In contrast to these methods, data-driven techniques mainly rely on available data to model and identify abnormal conditions. Related robotic applications are commonly based on principal component analysis, support vector machines and particle filters [36]. Lastly, knowledge-based methodologies represent a hybrid approach that combines aspects of the previous ones. These techniques have also been implemented to enhance the robustness of robotic systems. Typical applications in this context are often based on the representation of expert knowledge by combining fuzzy logic and artificial neural networks [37].

Although the existing solutions for the problem of sensory diagnosis have proven to be useful, they are not grounded, in general, on a rigorous framework that allows an adequate representation of uncertainty and that is able at the same time to compactly integrate diverse sources of knowledge. For instance, artificial neural networks do not offer explicit explanations about their deductions, fuzzy logic is mostly suitable to represent expert knowledge only, and many other existing approaches, such as ad-hoc or heuristic methodologies, do not provide results with probabilistic guarantees [38].

For the reasons explained, the mentioned paradigms seem to not completely fulfill the requirements for the problem considered in this thesis. Fortunately, there exist some other tools that can be found within the realm of artificial intelligence. Developed in the 1980s, Bayesian networks [39] are a kind of probabilistic graphical model that enables to represent the behavior of complex systems, to reason and derive conclusions about their components, and to correctly manage uncertainty. Since their introduction, Bayesian networks have been successfully applied to a wide diversity of fields, including medicine, economics, agriculture, engineering and many others [33]. Some of these applications are related to sensory diagnosis in the context of mobile robotics. However, the existing implementations are scarce [38] and do not completely leverage the flexibility offered by Bayesian networks, in the sense that they are usually restricted to very specific situations [40], [41].

1.3 Contributions

All the limitations related to sensory diagnosis explained above justify the development of this thesis. The main aim consists then on the construction of an architecture that enables a flexible representation of any sensory system, regardless of the concrete features of the considered mobile platform. This problem also poses some additional challenges that need to be addressed. One of the drawbacks of reasoning with Bayesian networks is its high computational cost [33], which severely compromises the real-time capabilities of mobile robots. Even solving this issue, the mere definition of such a representation of sensory systems would not suffice to prove its utility; thus, the proposal should also be deployed in practical applications to assess its benefits. Among others, the mentioned challenges summarize the problems tackled during the development of this thesis.

1.3 Contributions

The contributions produced in this thesis have arisen from the problems related to the application of Bayesian networks for the representation, diagnosis and enhancement of robotic sensory systems. One of the first problems to be addressed is precisely the study of the impact of sensory anomalies on the operation of mobile robots, which goes beyond the thoroughly treated problem of noisy measurements in these platforms [1].

As explained above, modern mobile robots often navigate within challenging environments from the perspective of their sensory systems, leading to abnormal (not only noisy) observations under certain conditions. Data corruption in these situations is not due to the impossibility of observing the true state of a physical quantity, but because of intrinsic limitations of the sensory devices themselves. This problem could affect basic tasks such as localization or mapping, thus potentially compromising the robotic operation. Since these basic problems of mobile robotics are mostly tackled by employing probabilistic frameworks such as Bayesian filters [27], it is important to assess to what extent abnormal observations affect these filters, and, in turn, the robotic operation itself.

The work in [31], developed in the scope of this thesis, is precisely aimed at studying the impact of abnormal observations on the performance of Bayesian filters. There exist some contributions in the literature that also treat some aspects of this problem. From an analytical point of view, works such as [42], [43] and [44], study the convergence of some Bayesian estimators, however, none of them take into account the possible abnormal observations discussed here. They also restrict to concrete implementations of these filters and do not cover any further aspects of their performance. On the other hand, works such as [45], [46] and [47], develop strategies to identify and overcome sensory anomalies, but again lack a complete treatment of the different aspects affecting filtering performance.

In this thesis, both of the mentioned problems are treated at once, i.e., the identification of kinds of anomalous sensory data and the study of their effect on different aspects of the performance of Bayesian estimators. Despite many sensors could be used for that, taking into account all of them would be overwhelming, due to the variety of possible abnormal effects. To solve that, the study only focuses on range sensors, since they play a key role in a wide variety of tasks in mobile robotics [1], thus covering many situations. The probabilistic framework used to implement the filtering problem is the one of dynamic Bayesian networks [48], considered an extension of Bayesian networks. This paradigm allows the representation of arbitrary structures of stochastic dynamic processes and also enables for generic inference, which makes it an ideal candidate to abstract from concrete implementations of Bayesian estimators, such as Kalman or Particle filters [1].

1.3 Contributions

An analytical treatment of the filtering problem in this context would be cumbersome, and possibly impractical, given the large number of parameters that should be considered to cover a sufficient variety of abnormal situations (e.g., the conditions of the problem, the sensor modeling parameters, the amount and value of anomalous sensory data, etc.). To solve this issue, the study is carried out from a rigorous statistical approach, based on analysis of variance (ANOVA) [49].

There is a number of aspects that are considered before performing the study and obtaining conclusions. One of them is the definition of the filtering problem in the context of range sensors, which is represented, for the sake of simplicity, by a one-dimensional obstacle tracking problem. Then, the most common situations that affect range sensors are compiled. A thorough review of the literature on these devices allows to summarize the effects of such abnormal conditions in two forms of data corruption, namely, biased and missing or saturated observations [31]. The filtering performance in the context of this study is represented by measures such as the expected accuracy of the filter, its expected uncertainty and the minimum number of steps that lead to convergence. The analysis to be performed also requires the definition of a bunch of factors that are likely to affect these measures. In this case, the ones considered relevant are the initial position of the obstacle in the tracking problem, its speed, and the amount of biased and missing observations.

With all these definitions, the statistical methodology of analysis of variance is then applied to sets of simulated experiments, designed to cover a wide variety of conditions in the considered problem. The obtained results provide complete and relevant conclusions on the effects of each factor on the filtering performance, which are also validated in a real scenario with a mobile robot. In summary, the results in [31] prove that the considered abnormal observations may have an important impact on the performance in certain situations, while the conditions of the problem, i.e., the position and speed of the obstacle, do not have any impact at all.

The analysis of sensory anomalies performed in [31] solves the first problem addressed in this thesis, since it proves the importance of considering abnormal situations beyond noisy observations in the context of the sensory apparatus of a mobile robot. However, the treated anomalies are not the only ones commonly present in mobile platforms, and a mere study of them does not suffice to enhance the robustness of robotic sensory systems. Thus, the next problem to be addressed is precisely related to the representation of such systems, in order to enable mobile robots for an intelligent sensory diagnosis and enhancement, based on the integration of diverse sources of knowledge. This problem represents the core of this thesis, and it is addressed in its related works ([50], [51], [38], [33] and [52]). Each one of these contributions, considered in chronological order, provides a solution that improves or complements previous ones regarding some aspect of the problem.

This evolutionary process begins with the works [50] and [51]. Their main aim is to provide a modeling framework, based on Bayesian networks, that allows the deduction of low-level sensory anomalies and the recovery from them. The definition of this proposal is actually based on a work previously co-authored by the doctoral candidate [53], where the fundamentals regarding the representation of robotic sensory systems are covered. The proposal introduced there is grounded on a basic element, the so-called *Bayesian sensor*, defined as a Bayesian network that aims to represent the behavior of a unique physical sensor, as well as any other relevant information related to it. A complete representation of the sensory system of a real robot endowed with basic devices (e.g., cliff detectors, gyroscopes, wheel encoders, etc.) is then obtained by combining as many Bayesian sensors as necessary, leading to a unique, monolithic model of the whole system. This model is assessed through different experiments in a real setting, proving its usefulness for sensory diagnosis

1.3 Contributions

and enhancement. In the works [50] and [51], this definition of Bayesian sensor is extended to cope with more complex sensors, such as 2-D rangefinders and RGB-D cameras, proving again the capabilities of the resulting model through diverse real experiments. Also, the work [51] contributes in this context with the definition of a procedure for the interactive construction of sensory systems, based on the mentioned devices. All the cited works present a static model that do not integrate temporal information of any kind. This is tackled in [38] with the application of sequential filters to the inference results produced by the proposed model.

One of the drawbacks of these first works is the high computational cost of inference with Bayesian networks. The practical implementations in these proposals mostly rely on an exact inference algorithm, the *jointree* or *junction tree* [54], [55], whose cost grows exponentially with the structural complexity of the Bayesian network employed. This and many other related problems are addressed in [33], one of the most relevant works of this thesis. The first issue treated there is related to the definition of Bayesian sensor, which is revised in order to capture the behavior of physical sensors more naturally and also to allow a representation that is suitable for efficient inference. A thorough review of the literature on inference algorithms inspires the main contribution of this work, consisting on a novel inference algorithm that leverages the particular topologies of sensory systems. A monolithic model of these systems can be potentially large in general, preventing even approximate inference in real-time situations. To solve that, this work proposes a new architecture, a so-called *layered* model, that strategically splits the monolithic one into different, isolated *levels* or *layers*, into which the existing Bayesian sensors are distributed.

There is an issue, however, that must be solved before the application of such hierarchical partitioning to the monolithic model of the system. Causal relationships among the components of a sensory system are potentially numerous and complex. This is due, in part, to the fact that the dependencies between two different sensors can be either cyclic or non-cyclic [33]. In the latter case, the behavior of a sensor is influenced by another one, and, in the former, this influence is mutual. Since Bayesian networks are not conceived to encode these cyclical dependencies, their inclusion could potentially prevent inference. This work studies the joint probability distribution induced by a cyclic Bayesian network and establishes some conditions for its existence and uniqueness. Under such conditions, an approximate solution is proposed, consisting on converting cyclic dependencies into non-cyclic ones by breaking the former. After that, the work defines a procedure for the construction of the layered model, which establishes a hierarchical ordering of sensors in which the lower ones explain or affect the behavior of the higher ones. This is based on the set of existing relationships among sensors in a robotic platform, and also implements the solution to the issue of cyclic dependencies.

These solutions lead to the introduction of the novel method mentioned above, i.e., the *layered* algorithm, which applies the *jointree* to each level of the layered model sequentially. The proposed method is actually an approximate algorithm, since exact inference is performed on an approximate model. The quality of the results produced by the layered algorithm is assessed through different simulated and real experiments in a particular robotic context. Also, the computational cost of this method is compared to the one of the exact *jointree* and other well-known approximate methods, using the same experimental setup for that. The obtained results prove that the proposed method significantly reduces the cost of inference and that also enhances the robustness of the sensory system considered.

Both the sensory architecture and the inference algorithm presented in [33], although useful and effective, have some relevant limitations. For instance, the mentioned proposals only handle

1.3 Contributions

static and discrete Bayesian networks, and do not allow the representation of more abstract sensory information. These shortcomings are addressed in the last work supporting this thesis [52], which also tackles the problem of deploying the proposed architecture in a real robotic application. This work revises again the definition of Bayesian sensor to increase its capabilities concerning the representation of generic sensory systems. In particular, such definition is extended to allow the use of both continuous and discrete random variables, to achieve a more formal treatment of sensory dynamics and to enable the simultaneous representation of different levels of cognitive abstraction. To solve the first two issues, this work re-defines the Bayesian sensor as a hybrid, dynamic Bayesian network, and also adapts the layered model and the corresponding inference algorithm to cope with this new proposal. Handling different levels of abstraction in the context of sensory systems is also necessary, since many of them rely on heterogeneous components that may produce information related to different ontologies at the same time (e.g., a sensory system simultaneously estimating the pose of a moving obstacle and classifying it into some category of objects). Treating these sources of information in the same network would lead to the induction of undesired dependencies, and thus to unreliable or inaccurate inference results. This situation is solved by treating the involved sensors in separate networks, which is taken into account in the new definition of Bayesian sensor and the inference method.

The resulting architecture is then a three-dimensional model including a layered axis for the ordered representation of sensors according to their mutual dependencies, a temporal axis for the encoding of temporal dynamics and a cognitive axis for the simultaneous treatment of different levels of cognitive abstraction. The other relevant contribution of this work is the implementation of the new proposal for a real robotic application. In particular, the proposed methodology is used to improve the robustness of a state-of-the-art pedestrian detection system as well as to enhance navigation in environments with humans. The Bayesian architecture is built by considering different ontologies of the information available about the detected pedestrians, such as their pose, speed or identity and also by taking into account environmental information, such as a map of the scene. This implementation is tested in both simulated and real experiments. The results show that the incorporation of the architecture to the existing system is able to improve the robustness of pedestrian detection, as well as the efficiency and safety of navigation.

Finally, this work is complemented with a last task developed during the thesis. The computational cost of inference in the mentioned application does not prevent real-time operation, however, it grows significantly with the number of detected pedestrians, limiting the operation in certain situations. An approach based on artificial neural networks is used to solve that and thus to reduce even more the cost of inference. The proposal aims to approximate Bayesian inference in each level of the layered model by building a feedforward neural network [56] representing such inference for each of the possible queries. These neural functions are trained using datasets gathered from simulated situations. The final solution is then tested for the same pedestrian detection problem, and the results show that the overall cost can be reduced several orders of magnitude; also, the error with respect to exact inference is generally low enough to not affect the performance of the sensory system.

All the contributions of this thesis have been already discussed, and they can be summarized in the following ones:

- A rigorous statistical study of the influence of abnormal sensory observations on the performance of Bayesian estimators, in the context of mobile robotics.
- The definition of a modeling architecture, based on Bayesian networks, conceived to enable

1.4 Framework and research activities

mobile robots for the intelligent detection of sensory abnormal situations and the recovery from them, which is possible thanks to the flexible integration of heterogeneous sources of knowledge allowed by the proposed methodology.

- A novel Bayesian inference algorithm that leverages the particular topologies of robotic sensory systems to significantly reduce the cost of inference.
- The implementation and validation of the proposed methodology in real robotic applications.

1.4 Framework and research activities

This thesis is the result of four years of work as a student of the mechatronics engineering doctoral programme of the University of Malaga. The doctoral candidate began his work in the year 2017, when he was awarded a national grant from the Spanish government belonging to the *Formación de Profesorado Universitario* (FPU) programme. The author of this thesis developed his work in the Systems Engineering and Automation department of the University of Malaga, under the supervision of Prof. Juan-Antonio Fernández-Madrigal and Prof. Alfonso García-Cerezo. This doctoral work was developed in the context of the national research projects *FIRST-ROB* and *TRUST-ROB*, both coordinated by Prof. Alfonso García-Cerezo, and aimed at the design and implementation of multi-robot systems for assisting human victims in rescue tasks.

The doctoral candidate also carried out a number of activities related to his research. In particular, he attended to several conferences, both national and international, such as the *XXXIX Jornadas de Automática*, the *Fourth Iberian Robotics Conference* (ROBOT 2019) or the *European Robotics Forum* (ERF 2020). The candidate was also invited to the University of Luxembourg for a three-month research stay, from July to September 2019, where he had the opportunity to collaborate with other researchers in the field of navigation with mobile platforms. During the last three years of the thesis period, the candidate collaborated as a lecturer in different undergraduate courses, namely, *Computer Control* (2018-2020), *Robot Programming* (2018-2021) and *Robotics* (2018-2021). The amount of teaching experience acquired is equivalent to a total of 18 ECTS (European Credit Transfer System). The candidate also attended to diverse seminars, covering topics such as the use of probability and statistics in mobile robotics, as well as the theory and applications of deep learning.

1.5 Publications

The work developed during this thesis led to the following publications (listed by category, and in reverse chronological order):

- **Journals indexed in the JCR**

- Manuel Castellano-Quero, Juan-Antonio Fernández-Madrigal, Alfonso García-Cerezo. *Improving Bayesian inference efficiency for sensory anomaly detection and recovery in mobile robots*. Expert Systems with Applications, vol. 163, pp. 113755, 2021, (Q1/T1).
- Manuel Castellano-Quero, Juan-Antonio Fernández-Madrigal, Alfonso García-Cerezo. *Statistical Study of the Performance of Recursive Bayesian Filters with Abnormal Observations from Range Sensors*. MDPI Sensors, vol. 20, no. 15, pp. 41-59, 2020, (Q1/T1).

- **International conferences**

1.6 Outline

- Manuel Castellano-Quero, Juan-Antonio Fernández-Madrigal, Alfonso García-Cerezo. *Integrating multiple sources of knowledge for the intelligent detection of anomalous sensory data in a mobile robot*. Fourth Iberian Robotics Conference (ROBOT 2019), Porto (Portugal), pp. 159-170, 2019.

- **National conferences**

- Manuel Castellano-Quero, Juan-Antonio Fernández-Madrigal, Alfonso García-Cerezo. *Hacia la diagnosis y recuperación de sensores robóticos a bajo nivel mediante inferencia en redes bayesianas*. XXXIX Jornadas de Automática, Badajoz (Spain), 2018.
- Manuel Castellano-Quero, Juan-Antonio Fernández-Madrigal, Alfonso García-Cerezo. *Interactive Construction of Bayesian Inference Networks for Robust Robot Sensorics*. XIV Simposio CEA de Control Inteligente (SCI 2018), Málaga (Spain), 2018.
- Manuel Castellano-Quero, Iván Fernández-Vega, Juan-Antonio Fernández-Madrigal, Ana Cruz-Martín. *Métodos de inferencia bayesiana empotrados para el diagnóstico y mejora sensoriales de un robot móvil*. II Jornadas de Computación Empotrada y Reconfigurable (JCER2017), Málaga, Spain, pp. 667-676, 2017.

1.6 Outline

The rest of this doctoral dissertation is organized in several chapters, which present an ordered and thorough exposition of all the contributions related to the work developed during the thesis. The contents of each chapter are detailed below.

- **Chapter 2** provides a basic review of some probabilistic and statistical notions, necessary to understand the methodology applied in the core chapters of the thesis.
- **Chapter 3** analyzes the influence of abnormal sensory behavior on the operation of mobile robots. In particular, a rigorous statistical study of such influence on the performance of Bayesian filters is presented, given the importance of probabilistic frameworks like these ones in basic robotic tasks.
- **Chapter 4** presents the core proposal of this thesis, i.e., the use of Bayesian networks for the representation of robotic sensory systems, which is aimed at enabling mobile robots for an intelligent sensory diagnosis and enhancement. A novel architecture based on this representation is defined, along with a new algorithm that leverages the model structure to increase the efficiency of inference.
- **Chapter 5** addresses the implementation and validation of the proposed methodology in different real robotic applications. The performance of the new inference algorithm introduced is assessed, as well as the usefulness of the proposed architecture in such real contexts.
- **Chapter 6** draws relevant conclusions from the main contributions of this thesis, and proposes different lines of research for future works.

Probabilistic and statistical bases

This chapter provides the reader with a basic insight on the different probabilistic and statistical methodologies employed throughout the thesis. Although the treatment of these notions is not comprehensive, all their essential aspects are covered without assuming much previous knowledge.

2.1 Introduction

Research in robotics and artificial intelligence is nowadays inconceivable without the support of probability theory and statistics. Many implementations of basic robotic tasks, for instance, are grounded on methods that use these mathematical frameworks as a base paradigm [1]. Also, in the realm of artificial intelligence, there exist numerous methods relying on the mentioned approaches [48]. Most of the contributions of this thesis arise from the application of probabilistic tools from artificial intelligence to different aspects of robotics, while others are obtained thanks to the use of rigorous statistical methods.

This chapter is aimed at providing the reader with the necessary background on the probabilistic and statistical methods employed in the thesis, whose knowledge is essential to fully understand the contributions presented. In particular, the chapter is divided into two clearly different parts, one of them devoted to the introduction of Bayesian networks [39], the core framework of the thesis, and the other to the presentation of the well-known statistical method of analysis of variance (ANOVA) [49]. The latter is applied in chapter 3 to study the impact of abnormal sensory observations on the operation of mobile robots, while the former methodology constitutes the base of chapters 4 and 5, devoted to present the proposals of the thesis related to sensory diagnosis and enhancement in the context of mobile robotics. Although the aim of the thesis is to be self-contained, this chapter is not intended to develop an in-depth treatment of the mentioned notions. For that, the reader could refer to classic texts such as [48] and [58] for the case of Bayesian networks, and to [49] for the case of ANOVA. In the following, a basic knowledge of probability and statistics is assumed. The reader could also refer to [1] for a thorough review.

2.2 Bayesian networks

Bayesian belief networks, or simply, Bayesian networks, are a kind of probabilistic graphical model [48] that serves, on the one hand, to compactly represent multiple sources of knowledge and the dependencies among them and, on the other hand, to rigorously reason about this knowledge with uncertainty, allowing the deduction of new information from available one. This mathematical framework, based on graph and probability theories, arised in the realm of artificial intelligence in the late 1980s [48] from a number of works by Judea Pearl and his colleagues, compiled in the highly influential textbook *Probabilistic Reasoning in Intelligent Systems* [59]. Bayesian networks were then considered a solution to the inconvenients posed by the definition of *expert systems*¹ concerning knowledge representation with uncertainty. Since then, probabilistic graphical models in general, and Bayesian networks in particular, are in common use in a wide diversity of disciplines, including medicine, economics, agriculture and engineering, among many others [33].

In the following, a more detailed review on Bayesian networks is provided. First, a rigorous definition is presented, along with an overview of the possibilities this paradigm offers for the representation of knowledge. Then, the most common algorithms used to perform inference with this framework are covered, focusing specially on those employed in the thesis.

2.2.1 Representation

Formally, a Bayesian network for a set of n random variables $\mathbf{Z} = \{Z_1, Z_2, \dots, Z_n\}$ is a pair (G, Θ) consisting of a directed acyclic graph G over variables \mathbf{Z} , called the *network structure*, and a set of Conditional Probability Distributions (CPDs) Θ for each variable in \mathbf{Z} , called the *network parameterization* [48], [58]. The complete definition of a Bayesian network is a succinct form of modeling a unique joint probability distribution for all the random variables involved, i.e., it represents a way of determining $P(\mathbf{Z})$ by imposing some constraints. On the one hand, the network structure represents random variables through nodes, and the dependencies among them through directed arcs. On the other hand, the network parameterization quantifies such dependencies through the specification of conditional distributions for all the variables. In general, it can be proved that these constraints can only be satisfied by one and only one joint distribution $P(\mathbf{Z})$ [58], as long as the network graph is acyclic [33].

The graphical structure of a Bayesian network can be seen as a representation of the causal relationships existing among the considered random variables. Defining a network in this way, i.e., by encoding cause-effect relationships from the point of view of an expert, is not only possible but also common practice. However, it is important to note that the constraints formally induced by such graphical structure are a set of conditional *independence* statements that are not grounded on the notion of causality [58]. This implies that, in general, two different graphs may encode the same set of independencies even if one of them correctly captures these expert causal perceptions while the other one does not.

The independencies induced by a Bayesian network serve to simplify the general *chain rule* [1] used to express joint distributions as a product of conditional ones. Concerning Bayesian networks, this expression reduces to the following factoring [58]:

¹According to the definition in [48], an expert system is a computerized system conceived to emulate human experts in performing difficult tasks, such as oil-well location or medical diagnosis, for instance.

2.2 Bayesian networks

$$P(\mathbf{Z}) = \prod_{i=1}^n P(Z_i | pa(Z_i)), \quad (2.1)$$

where $pa(Z_i)$ represents the set of parents of node Z_i in graph G , i.e., all the nodes that have an outgoing arc ending at Z_i . Note that the conditional distribution $P(Z_i | pa(Z_i))$ also covers the case $pa(Z_i) = \emptyset$, thus representing a *prior* distribution for those variables Z_i with no parents in G .

Deriving the complete set of independencies that leads to the simplification in equation (2.1) is not trivial in general. Fortunately, there exist a procedure, called the graphical test of *d-separation* [58], that allows to derive all the independencies induced by the network structure. In general, the parameterization may also induce extra independencies, under very particular conditions. A complete treatment of the independencies induced by Bayesian networks is left to the reader interested in this topic, who may refer to [58] for more details.

Bayesian networks are classified depending on the form of the joint probability distribution they represent. This classification can be done partially by focusing on the nature of the random variables involved. In particular, a Bayesian network can be identified as *discrete*, *continuous* or *hybrid* depending on whether it is defined over discrete or continuous random variables only or over a combination of both. In the case of discrete networks, each CPD is defined in tabular form as follows. Let $X \subset \mathbf{Z}$ be one of the network variables and $\mathbf{U} = pa(X)$ the set of its parents. The CPD for variable X is the distribution $P(X|\mathbf{U})$, which is usually defined as a table whose entries represent the concrete *parameters* of such distribution (i.e., its values), expressed for every value x of variable X and every instantiation² \mathbf{u} of parents \mathbf{U} . A simple example of discrete Bayesian network modeling the behavior of a collision sensor is shown in figure 2.1. The joint distribution of this network has the form of a multidimensional *probability mass function* (pmf) over variables A , B and C , expressed as $P(A, B, C) = P(A)P(B)P(C|A, B)$ according to equation (2.1).

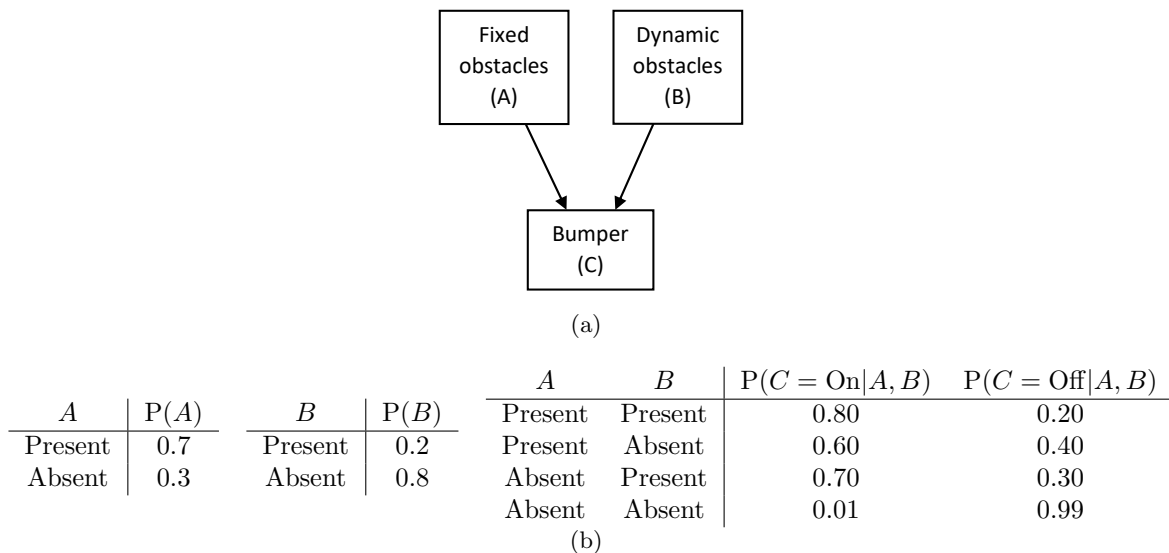


Figure 2.1: Discrete Bayesian network representing the behavior of a collision sensor. (a) Network structure (graph). (b) Network parameterization (CPDs).

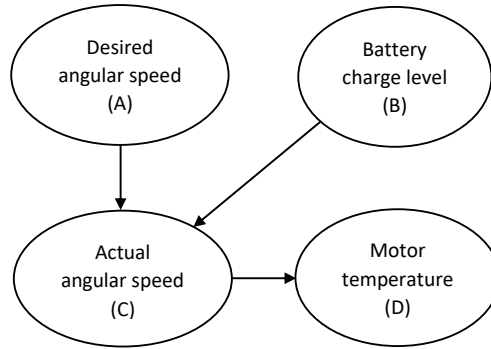
²According to the definition in [58], an instantiation of variables, say, A , B and C , is a propositional sentence of the form $(A = a) \wedge (B = b) \wedge (C = c)$, where a , b and c are values of the mentioned variables, respectively.

2.2 Bayesian networks

When all the random variables in a Bayesian network are continuous, each CPD is expressed as a *probability density function* (pdf), and therefore, the joint distribution of the network has the form of a multidimensional pdf. Although the mentioned distributions could have arbitrary shapes in general, they will always be Gaussian in the scope of this thesis. When all the CPDs of a network are Gaussian, the network is usually referred to as *Gaussian Bayesian network*. Also, all the CPDs in this context will be assumed to be *linear-Gaussian*, according to the following definition [48]. Let $X \subset \mathbf{Z}$ be a continuous random variable in a network where $\mathbf{U} = \{U_1, U_2, \dots, U_n\}$ represents the set of its parents in G . A linear-Gaussian CPD for the corresponding node of variable X , given an instantiation \mathbf{u} of parents \mathbf{U} is the distribution:

$$p(X|\mathbf{u}) = \mathcal{N}(\beta_0 + \beta_1 u_1 + \dots + \beta_n u_n; \sigma^2), \quad (2.2)$$

where $\beta_0, \dots, \beta_n \in \mathbb{R}$ and \mathcal{N} represents a normal distribution whose mean is a linear combination of the values of parents \mathbf{U} and its variance is σ^2 . Note that the lower case notation p is used to denote the shape of distributions over continuous random variables, i.e., pdfs. An example of Gaussian Bayesian network for the behavior of different robotic sensors is shown in figure 2.2. The joint distribution in this case has the shape of a multivariate Gaussian pdf, which, according to the factoring in equation (2.1) is $p(A, B, C, D) = p(A) p(B) p(C|A, B) p(D|C)$.



(a)

$$\begin{aligned}
 p(A) &= \mathcal{N}(0.3; 0.05^2) \text{ (m/s)} \\
 p(B) &= \mathcal{N}(2.2; 0.5^2) \text{ (Ah)} \\
 p(C|A, B) &= \mathcal{N}(0.8A - 0.1B; 0.01^2) \text{ (m/s)} \\
 p(D|C) &= \mathcal{N}(30.5C; 0.5^2) \text{ (}^\circ\text{C)}
 \end{aligned}$$

(b)

Figure 2.2: Gaussian Bayesian network representing the behavior of several robotic sensors. (a) Network structure (graph). (b) Network parameterization (CPDs) with physical units.

Bayesian networks can also be defined over a set \mathbf{Z} including both continuous and discrete random variables, however, the network modeling in this case has some limitations, as explained later on. Analogously to the purely continuous case, there is no restriction of the CPDs to have an arbitrary form. Nevertheless, in the scope of this thesis, they will always be *conditional linear-Gaussian*, according to the definition in [48], as follows. Let $X \subset \mathbf{Z}$ be a continuous random variable and let $\mathbf{U} = \{U_1, U_2, \dots, U_m\}$ and $\mathbf{V} = \{V_1, V_2, \dots, V_k\}$ be the sets of its corresponding discrete and continuous parents, respectively. Then, variable X is said to have a conditional linear-Gaussian CPD if, for every instantiation \mathbf{u} of the discrete parents \mathbf{U} , there exist a set of $k + 1$ coefficients $\beta_{\mathbf{u},0}, \beta_{\mathbf{u},1}, \dots, \beta_{\mathbf{u},k} \in \mathbb{R}$ and a variance $\sigma_{\mathbf{u}}^2 \in \mathbb{R}^+$ such that:

2.2 Bayesian networks

$$p(X|\mathbf{u}, \mathbf{v}) = \mathcal{N}(\beta_{\mathbf{u},0} + \beta_{\mathbf{u},1} v_1 + \dots + \beta_{\mathbf{u},k} v_k; \sigma_{\mathbf{u}}^2). \quad (2.3)$$

This clearly states that variable X would have as many linear-Gaussian CPDs as the number of possible instantiations of its discrete parents. If variable X had no continuous parents, these CPDs would only depend on fixed mean and variance parameters. The limitation commented above arises when defining the CPD in equation (2.3) for the case of discrete variables. In these kind of hybrid networks, called conditional linear-Gaussian networks (CLGs), a discrete variable is not allowed to have any continuous parents. Thus, the CPDs for these variables are defined the same way as the ones for the purely discrete case. The joint probability distribution induced by a CLG network has, in general, the form of a *mixture* of Gaussians, i.e., a weighted average of them [48]. This mixture model has one Gaussian component per instantiation of the discrete variables, and the weight of such component is precisely the probability of that instantiation. An example of a CLG network capturing the behavior of an ultrasonic sensor is depicted in figure 2.3. In this case, the joint distribution has the shape of a mixture of Gaussians, which, according to the simplified chain rule in equation (2.1), has the form $p(A, B, C) = P(A) p(B) p(C|A, B)$.

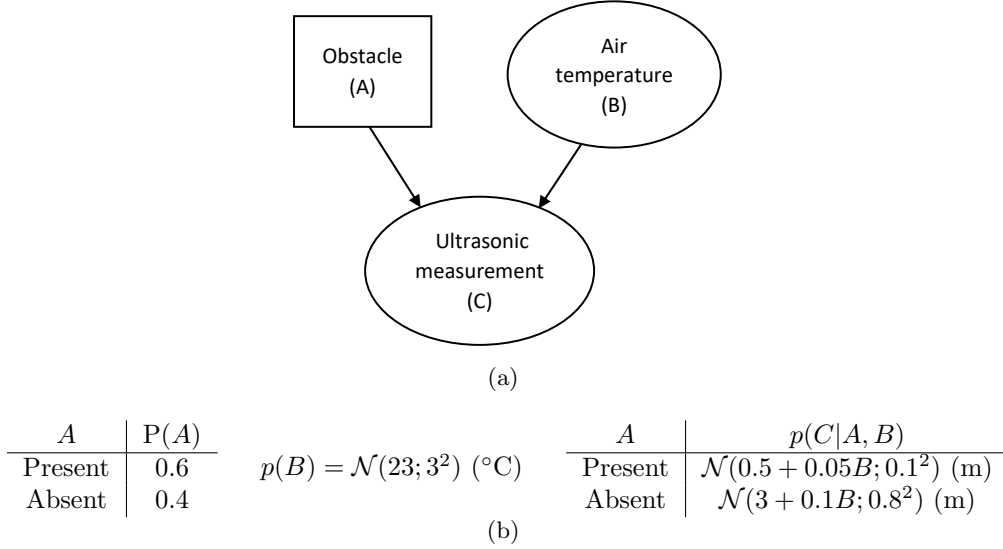


Figure 2.3: CLG Bayesian network representing the behavior of an ultrasonic sensor. Round nodes denote continuous variables and square ones discrete variables. (a) Network structure (graph). (b) Network parameterization (CPDs) with physical units.

Equations (2.2) and (2.3) define CPDs for the case of continuous random variables with one-dimensional support. Both can be extended to the multidimensional case as follows. Let $\vec{X} = (X_1, X_2, \dots, X_n)^t$ be a continuous random vector with support cardinality n , $\mathbf{V} = \{\vec{V}_1, \vec{V}_2, \dots, \vec{V}_k\}$ the set of its corresponding continuous parents (if any) in graph G , and \mathbf{U} the set of discrete parents (if any). Also, let m be the sum of the support cardinalities of all the random vectors in \mathbf{V} , i.e., $m = \sum_{i=1}^k |\vec{V}_i|$. If \mathbf{u} and \mathbf{v} are instantiations of the mentioned set of variables \mathbf{U} and \mathbf{V} , respectively, the form of the CPDs given in (2.2) and (2.3) can be rewritten as:

$$p(\vec{X}|\mathbf{v}) = \mathcal{N}(\vec{\mu} + W \cdot \vec{v}; \Sigma^2) \quad (2.4)$$

and

2.2 Bayesian networks

$$p(\vec{X}|\mathbf{u}, \mathbf{v}) = \mathcal{N}(\vec{\mu}_{\mathbf{u}} + W_{\mathbf{u}} \cdot \vec{v}; \Sigma_{\mathbf{u}}^2) \quad (2.5)$$

respectively, where $\vec{\mu}$ is a n -dimensional column vector of real coefficients, W is a $n \times m$ matrix (also of real coefficients), Σ^2 is a positive semi-definite $n \times n$ covariance matrix and $\vec{v} = (v_1, v_2, \dots, v_m)^t$ is a m -dimensional column vector representing values of variables in \mathbf{V} . Recall that subindex \mathbf{u} in equation (2.5) denotes different $\vec{\mu}$, W and Σ^2 depending on the concrete instantiation \mathbf{u} of the discrete variables.

The models introduced so far only constitute a part of the expressive power offered by the paradigm of Bayesian networks. Another aspect that is usually considered for their classification is the ability of capturing time. In this regard, the previous representations are only applicable to *static* models of reality, but they can be extended to incorporate the temporal evolution of a system, i.e., to become *dynamic*. Formally, and in the context of this thesis, this means that the definition of Bayesian network has to be extended to enable the treatment of discrete-time stochastic dynamic processes [60]. For that, the timeline must be discretized into a set of regularly spaced intervals called *time slices* [48], which represent variables of the system state at different times, and are referred to with integer numbers.

Before formally introducing the notion of dynamic Bayesian network, a previous model has to be defined. A 2-time-slice Bayesian network (2-TBN) [60] is a process whose state variables at a certain time t are $\mathbf{Z}^{(t)} = \{Z_1^{(t)}, Z_2^{(t)}, \dots, Z_n^{(t)}\}$. It is a fragment of a Bayesian network whose structure is defined over the union of state variables at adjacent time slices, i.e., $\mathbf{Z}^{(t-1)} \cup \mathbf{Z}^{(t)}$, and it is only parameterized for those nodes in the graph corresponding to variables $\mathbf{Z}^{(t)}$ (thus, only those nodes are annotated with CPDs). Also, nodes referring to variables $\mathbf{Z}^{(t-1)}$ have no parents. Actually, this network represents a conditional distribution of the form $P(\mathbf{Z}^{(t)}|\mathbf{Z}^{(t-1)})$, usually called *transition model*. Then, a Dynamic Bayesian Network (DBN) [48] can be defined as a pair $(\mathcal{B}_0, \mathcal{B}_{\rightarrow})$ where \mathcal{B}_0 is a Bayesian network over variables $\mathbf{Z}^{(0)}$, called *initial network*, which represents the initial distribution of state variables, and $\mathcal{B}_{\rightarrow}$ is a 2-TBN for the process, also referred to as *transition network*. Note that, given a time span $T \geq 0$, this representation allows for the composition of the initial network \mathcal{B}_0 along with instances of the transition network $\mathcal{B}_{\rightarrow}$ to create an equivalent monolithic Bayesian network over all the variables within such time span. This operation is called *unrolling* of the DBN and it is related to some inference methods [48].

DBNs can also be identified as discrete, continuous or hybrid, depending on the nature of the set of variables over which they are defined. In this thesis, the case of discrete variables having continuous parents in a DBN will not be considered either. The joint distribution induced by a DBN can be expressed in terms of the conditional distributions it is based on, analogously to the case of a static network. Given a time span $T > 0$, the joint distribution of a DBN is defined as follows:

$$P(\mathbf{Z}^{(0:T)}) = \prod_{t=0}^T \prod_{i=1}^n P(Z_i^{(t)} | pa(Z_i^{(t)})), \quad (2.6)$$

where the notation P may refer to different shapes of distributions, depending on the nature of the random variables involved, as in the case of static networks. The conditional distributions in equation (2.6) refer to variables whose corresponding parents may belong to either the same time slice or to the immediately previous one. In the former case, those CPDs define, along with the graphical topology, the *intra-slice* part of the network, while in the latter, the CPDs and the

2.2 Bayesian networks

corresponding topology define the *inter-slice* one. Recall that the mentioned CPDs have the same form as the ones used in static networks, as explained above.

With this definition, the framework of DBNs generalizes the representation of dynamic systems, allowing arbitrary topologies in contrast, for instance, to the well-known Hidden Markov Models (HMMs) and Kalman Filter Models (KFM) [57]. There exist an example of extension of KFM, called *switching* Kalman Filter or switching linear dynamical system (SLDS) [57], aimed at combining different transition models at a time. An implementation of SLDS as a DBN is depicted in figure 2.4 (only the structure is represented). Please refer to [48], [57] for further details on this model and knowledge representation with DBNs in general.

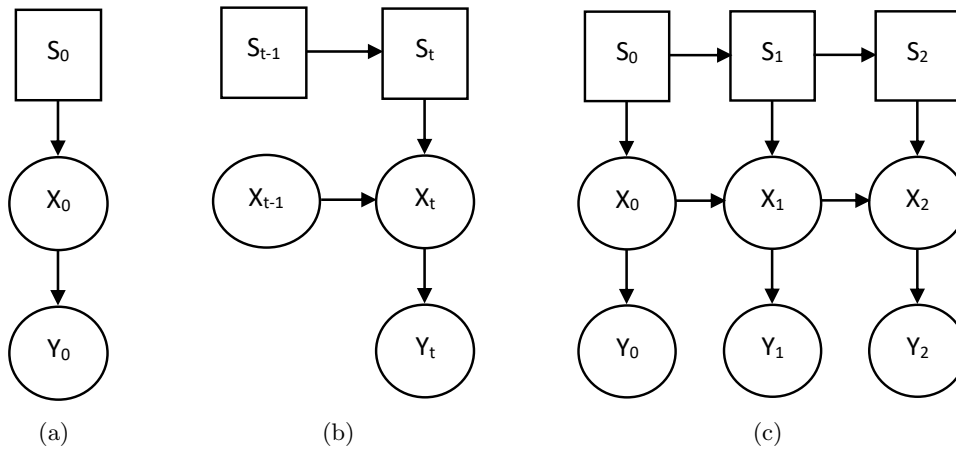


Figure 2.4: Switching linear dynamical system as a DBN (structure only). Here, variables X represent hidden states of the system, variables Y the corresponding observed ones, and variables S the set of indices for the different models considered. Again, round nodes denote continuous variables and squared ones discrete variables. (a) Initial network \mathcal{B}_0 . (b) Transition network $\mathcal{B}_{\rightarrow}$. (c) Unrolled DBN for three time slices.

2.2.2 Inference

Once a Bayesian network is completely defined (i.e, its structure and parameters are specified) it is possible to use it to deduce new information from available one. Considering a Bayesian network defined over a set of random variables \mathbf{Z} , this deductive process formally consists in obtaining the conditional distribution $P(\mathbf{Q}|\mathbf{E})$, where \mathbf{Q} is the set of query variables (the ones of interest) and \mathbf{E} the set of observed variables (also known as *evidence*, which represents the existing knowledge). In this context, $\mathbf{Z} = \mathbf{Q} \cup \mathbf{E}$ and $\mathbf{Q} \cap \mathbf{E} = \emptyset$. There also exist additional queries that can be answered by a Bayesian network [58], but the one described here is the most common and simple, and it will be the one used throughout the thesis. In the particular case of DBNs, this general query usually adopts specific forms. One of them is known as *filtering* [57] and it is, again, the one that will be used in this thesis. If the time slice representing the present in the DBN is denoted as t , the filtering query has the form $P(\mathbf{Q}^{(t)}|\mathbf{E}^{(0:t)})$. This notation aims to highlight the fact that the only instance of the query variables considered is the present one, while, concerning the evidence variables, the instances considered range from the initial to the present time slice. In Bayesian filtering for robotics, this is named *on-line filtering*.

The notion of evidence used above is usually called *certain* or *hard* evidence, since it is assumed that the knowledge it represents is obtained with no uncertainty (this is usually the case

2.2 Bayesian networks

in inference tasks). There exist some other situations, however, in which the available information might not be completely certain, due to some reason. In that context, the evidence is said to be *soft*, and it can be emulated by adding auxiliary variables to the Bayesian network with a suitable parameterization, and then imposing hard evidence on them. This method is known as *virtual evidence*, and it will be the one used in this thesis when necessary. Please refer to [58] for further details on the notion of soft and virtual evidence.

In the context of Bayesian networks, the mentioned queries are always obtained by employing some kind of inference algorithm, since the mere application of basic rules from probability theory could be cumbersome and extremely inefficient in the general case [48]. To illustrate this issue, consider, again, the problem of calculating $P(\mathbf{Q}|\mathbf{E})$. This could be achieved by simply marginalizing the joint distribution $P(\mathbf{Z})$ to obtain both $P(\mathbf{Q}, \mathbf{E})$ and $P(\mathbf{E})$, and finally dividing the former by the latter. The problem with this *naive* approach resides on the necessity of generating the entire joint distribution (specially in the case its definition contains discrete random variables). In a Bayesian network, it is common to find subexpressions in its induced joint that depend on a small number of variables. Building such distribution completely implies the exploration of the entire domain, thus leading to a repeated computation of these expressions exponentially many times in the worst case [48]. To avoid this, the existing inference algorithms perform a more efficient scheduling concerning the computation of these and other expressions, which makes them more suitable for the inference problem in general.

There exist many different methods for inference in Bayesian networks, and all of them can be classified either as *exact* or *approximate* [48], [58], [57]. Exact algorithms provide correct results for the queried distributions at the expense of a generally high computational cost, while approximate ones allow a tradeoff between accuracy and efficiency (many of these algorithms are *any-time*). In general, it can be proved that both exact and approximate inference in graphical models are NP-hard problems, however, they can still be solved effectively for many real-world applications [48]. Despite the wide variety of existing methods, only the ones employed in the thesis will be commented in this chapter. Please refer to [48], [58] for a complete taxonomy and a more in-depth treatment of these algorithms.

Concerning exact methods, the one used in this thesis is the well-known *jointree* algorithm [58], also referred to as *junction tree* and *clique tree* algorithm in the literature. This method was initially developed for discrete and static Bayesian networks in the late 1980s by Lauritzen and Spiegelhalter [54], and was then refined by Jensen and others in the early 1990s [61]. At that time, Lauritzen adapted the method to allow inference in CLG networks [62] and, later on, Murphy developed an algorithm for exact inference in DBNs, known as *interface algorithm*, also based on the traditional jointree [57]. This algorithm has also been extended in many ways (some of them will be discussed in this thesis) and has been used for many tasks beyond probabilistic inference [48].

The *jointree* method, like some others, is grounded on the idea of scheduling local computations related to the joint distribution induced by a Bayesian network, and it is aimed at improving efficiency with respect to the naive approach mentioned before. More concretely, the operations contemplated in this algorithm rely on the manipulation of compact representations of the network CPDs, called *factors* or *potentials*, which are actually functions that encode the full parameterization of such distributions. For instance, a CPD representing the distribution $P(C|A, B)$ of the network in figure 2.1 can be fully captured by a potential, denoted as $\Theta_{C|A,B}$. The particular form of a potential depends on the shape of the distribution it represents. In the fully discrete case, a potential is a tabular function defined over discrete variables that maps instantiations of these

2.2 Bayesian networks

variables to values of the corresponding CPD. Regarding the purely continuous case, a potential is defined as a Gaussian canonical form [48], which is not only able to represent Gaussian distributions but also linear-Gaussian models. When the CPD to be encoded is a conditional linear-Gaussian over both continuous and discrete variables, the potential is again defined as a table. In this case, its entries represent different canonical forms over the continuous variables, whose parameters depend on the particular instantiation of the discrete ones.

Potentials enable to capture the network CPDs and also the intermediate expressions that arise during the inference process. As mentioned before, this process relies on basic operations with potentials. Common ones are product, division, marginalization, normalization and incorporation of evidence, whose implementation vary depending on the concrete kind of potential (see [48] for more details). These representation of potentials and the corresponding operations constitute the basis of many inference algorithms, including the jointree. The main difference among them resides on the way these basic operations are planned. In the case of the jointree, all the operations to be performed are scheduled by using a secondary graphical structure, also called *jointree*, constructed from the Bayesian network of the problem. A *jointree* \mathcal{J} is as an undirected graph (more specifically, a tree) whose nodes represent subsets of variables of the original problem, called *clusters*, and whose arcs represent intersection sets of the variables of the adjacent nodes, called *separators*. This structure will be used to answer probability queries once the algorithm is performed.

The inference process begins by assigning each network potential to some node in the jointree that contains all the variables over which the potential is defined. Auxiliary potentials representing evidence, called *evidence indicators*, are also defined and assigned. Then, all the potentials finally assigned to a node are multiplied among them, and this step is repeated for all the nodes. The result of this operation leads to an *inconsistent* jointree, since it is not ready to answer queries. To solve that, a procedure called *message-passing*, based on fundamental operations with potentials, [58] is performed, finally leading to a *consistent* jointree.

After this process, each node of the jointree is annotated with a product of factors representing a joint distribution over the associated variables and the evidence. In other words, a cluster containing a set of variables \mathbf{X} represents, finally, the joint distribution $P(\mathbf{X}, \mathbf{e})$, where \mathbf{e} is an instantiation of the evidence variables \mathbf{E} . In order to answer to the query $P(\mathbf{Q}|\mathbf{e})$ (with $\mathbf{Q} \subseteq \mathbf{X}$) it would suffice to marginalize and normalize the distribution $P(\mathbf{X}, \mathbf{e})$. Note that the jointree algorithm allows for answering multiple queries at a time, as long as the evidence remains unchanged. There exist several ways of implementing the message-passing procedure. The one used in this thesis is called the *Hugin architecture* [61] and its computational cost is $O(n \exp(w))$ [58], where n is the number of nodes in the jointree and w is the *treewidth*, defined as the size of the largest cluster minus one. Note that this expression represents an upper bound of the computational cost. For instance, if all the variables in the Bayesian network were continuous, the inference could be performed in polynomial time. Particularly, the worst-case cost in this case would be of the form $O(w^3)$ [48], [57]; thus, the exponential character of the cost is given by the presence of discrete random variables. The jointrees corresponding to the networks presented in subsection 2.2.1 are shown in figure 2.5.

The jointree method constitutes the cornerstone of the algorithm proposed in this thesis, which will be addressed in chapter 4. Another inference methods have also been used throughout the thesis, for instance, to compare their performance to the proposed one. In particular, two approximate algorithms have been employed for that, one of them known as *loopy belief propagation* (LBP) [58] and the other one as *Gibbs sampling* [48] for Bayesian networks. The LBP was conceived, in

2.2 Bayesian networks

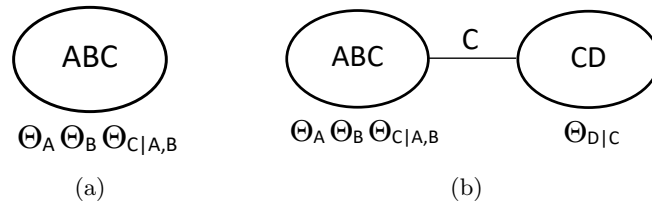


Figure 2.5: Jointrees corresponding to the examples of Bayesian networks introduced in subsection 2.2.1, with an assignment of potentials. (a) Jointree for the Bayesian networks in figures 2.1 and 2.3. (b) Jointree for the Bayesian network in figure 2.2.

its primitive form [63], as an exact inference method for *polytree*³ discrete Bayesian networks. Later on, a proposal was made in [59] to apply the same proposal to networks with loops, which was then tested in several works and even adapted for linear-Gaussian networks [48]. There also exists a version of this algorithm that can be employed for inference with discrete DBNs [57].

The LBP can be viewed as an optimization algorithm that is aimed at obtaining an approximate distribution that minimizes the Kullback-Leibler divergence with respect an exact one induced by a Bayesian network [58]. This method is also based on basic operations with potentials and a message-passing procedure, however, the computations are scheduled by employing the Bayesian network structure directly, and are arranged in iterations. Thus, this algorithm is based on an iterative process that begins with an initialization of messages and ends when some convergence criterion over them has been reached. However, it is not guaranteed that the method converges for every problem in the general case. When it does, it produces an estimation of $P(X, \mathbf{U}|\mathbf{e})$ for every variable X in the network, being \mathbf{U} the corresponding set of its parents. The desired query could be answered in this case by simply looking for a distribution over variables \mathbf{Q} such that $\mathbf{Q} \subseteq X \cup \mathbf{U}$ and finally by marginalizing the mentioned distribution if necessary. The LBP algorithm, like other ones that perform inference by optimization, is any-time, and thus its cost may vary depending, among other factors, on the number of iterations performed.

The other approximate method used in the thesis is the *Gibbs sampling* for Bayesian networks, which belongs to the family of stochastic sampling algorithms and, more specifically, to the sub-family of Markov Chain Monte Carlo (MCMC) methods [58]. These algorithms are based on a fairly different strategy compared to that of the ones introduced above. In particular, they approximate probability distributions by actually estimating the expectation of some function. The algorithms rely, for that, on the definition of some procedure to generate random instantiations for the variables \mathbf{Z} of a Bayesian network, according to its underlying joint $P(\mathbf{Z})$. The instantiations are often called *samples* and the process to obtain them, *simulation* of the Bayesian network. With this definition, the algorithm itself is prepared to perform inference. First, a number of samples is obtained, then, the defined function is evaluated at the instantiation of each sample, and, finally, the target query is answered by calculating the arithmetic average of the obtained values along with some other operations.

The concrete algorithm discussed here was introduced for discrete networks in [64], and it relies on the technique of Gibbs sampling, first defined in [65], for the network simulation. This inference method belongs to the MCMC subfamily, which has some implications in the way the Bayesian network is represented. This algorithm, in particular, encodes the joint distribution $P(\mathbf{Z})$ as the

³A Bayesian network has a *polytree* structure when its corresponding graph does not contain any undirected loops.

2.3 Analysis of variance (ANOVA)

stationary distribution of a *Gibbs chain*, which is a specific kind of Markov chain, i.e., a concrete kind of DBN whose state variables are \mathbf{Z} (please refer to [58] for further details on the concepts related to Markov chains). This is done to allow the simulation procedure even when sampling from the joint $P(\mathbf{Z})$ is difficult or computationally infeasible. Once the entire sample is generated, the algorithm is able to answer queries of the form $P(X|\mathbf{e})$ for any variable $X \notin \mathbf{E}$ in the network. This algorithm, like many other *sample-based* methods, is any-time, since its cost depends, among other factors, on the size of the generated sample.

2.3 Analysis of variance (ANOVA)

Analysis of variance (ANOVA) is a statistical methodology that serves to study the differences existing among several groups in a population, where each group corresponds to a subset of the sample that is obtained under the same conditions. Usually, the variables that explain a specific condition are called factors, whereas the aspect of the population under study is referred to as the dependent variable. The differences among groups are always studied in terms of their means. Thus, ANOVA enables to derive conclusions about the effects that the considered factors might have on the population means. This methodology was first introduced by Ronald Fisher in the early 1920s [49], and it is currently applied to a wide range of fields including social sciences [66], biology [67], physics [68] and engineering [69], among many others. In this thesis, it is employed to study the impact that abnormal observations may have on the performance of Bayesian filters, as discussed in chapter 3.

2.3.1 Basic notions

ANOVA can be seen as a statistical method for hypothesis testing. Depending on the number of factors to be considered for a study, the number and form of such hypothesis may vary. There exist different kinds of statistical tests that can be used to perform ANOVA, however, the most traditional and the one that will be employed in this thesis is the F-test, which relies on the Fisher-Snedecor distribution. Regardless of the number of factors, there are some assumptions that must be met for the validity of the conclusions derived from an F-test [49]:

- The obtained data for a specific group must follow a normal distribution.
- The population variances must be equal for all groups (*homoscedasticity* of variances).
- The observed values must be statistically independent from each other.

It is easy that the previous conditions are not fully satisfied in a real-world situation. Despite that, ANOVA is relatively robust to violations of these assumptions (please refer to [49] for more details on this issue).

As said before, one or more factors may be considered for an analysis. When only one factor is used, the procedure is referred to as *one-way* ANOVA; in general, when there are n factors, it is called *n-way* ANOVA. For the sake of clarity, some basic concepts related to the *one-way* ANOVA will be first introduced and then these will be generalized for the multi-factorial case.

Recall that a factor is a variable that might have some influence on the population. Such behaviour is called *main effect* and it is normally studied for a very reduced set of possible values for that variable, since the study would be really difficult to interpret otherwise. In this thesis, only two values per factor will be used at most, since those could be considered extreme, therefore

2.3 Analysis of variance (ANOVA)

covering a wide range of situations (the general case will be explained anyway). With these notions in mind, the null hypothesis that is tested in every *one-way* ANOVA can be stated. Let A be the only factor in our ANOVA, a the number of groups (thus, possible values) for that factor and μ_i the mean for the i -th group, one of each denoted by using natural numbers from 1 to a . The null hypothesis H_0 to be tested is:

$$H_0 : \mu_1 = \mu_2 = \dots = \mu_i = \dots = \mu_a \quad (2.7)$$

In other words, this hypothesis assumes the equality of all group means. Equivalently, if such hypothesis is not rejected, factor A is said to not have any effect on the population means. In the case that the null hypothesis was rejected, at least one of the equalities in equation (2.7) would not hold and, therefore, factor A would have some effect on the population. In order to either accept or reject any hypothesis, some statistic must be calculated; in this case, it will be the F statistic, which is obtained as explained below.

The methodology of ANOVA relies on the comparison of two different estimates of the population variance. One of them is related to the variability among the group means (usually called variability *between* groups), while the other refers to the variability observed among the individual scores within of each groups (normally referred to as variability *within* groups). Before obtaining the F statistic, some pairs of intermediate parameters associated with those two variabilities have to be determined, namely, the *sum of squares* (SS), the *degrees of freedom* (df) and the *mean square* (MS). Here, only calculation is discussed; please see [49] for more details on meaning. The first parameter is defined as follows:

$$SS_{within} = \sum_{j=1}^a \sum_{i=1}^n (Y_{ij} - \bar{Y}_j)^2 \quad (2.8)$$

where SS_{within} is the sum of squares within groups, a the number of groups, n the number of scores within each group (the design is assumed to be balanced, i.e., all the groups contain the same number of scores), \bar{Y}_j is the *estimated* mean for the j -th group and Y_{ij} is the i -th score for the j -th group. The second parameter is defined as follows:

$$SS_{between} = n \sum_{j=1}^a (\bar{Y}_j - \bar{Y})^2 \quad (2.9)$$

where $SS_{between}$ is the sum of squares between groups and \bar{Y} is the sample mean of the population (therefore, its *estimated* mean). Then, the degrees of freedom within groups is defined as:

$$df_{within} = N - a \quad (2.10)$$

where N is the number of scores in the whole sample. The degrees of freedom between groups can be obtained as follows:

$$df_{between} = a - 1 \quad (2.11)$$

Then, the mean square within is:

$$MS_{within} = \frac{SS_{within}}{df_{within}} \quad (2.12)$$

and, analogously, the mean square between groups:

2.3 Analysis of variance (ANOVA)

$$MS_{between} = \frac{SS_{between}}{df_{between}} \quad (2.13)$$

Finally, the value of the F statistic can be obtained as follows:

$$F = \frac{MS_{between}}{MS_{within}} \quad (2.14)$$

Once the F statistic is calculated, the corresponding F-test to decide whether the null hypothesis should be rejected or accepted can be performed. For that, a *critical region* for a given significance level (it will always be 0.05 in this thesis) must be defined and then the corresponding *p-value* can be calculated by using an appropriate distribution (please refer to [70] for a review on these notions). Recall that the F statistic follows an F-distribution with degrees of freedom $df_{between}$ and df_{within} (numerator and denominator, respectively). In the case that the p-value is equal to the significance level or greater, the null hypothesis will be accepted and it will be rejected otherwise.

At this point, a conclusion on the effect that a certain factor has on the population has just been obtained. However, it is always a good practice to confirm such conclusion, specially in the case that the considered factor has some effect (i.e., in case of rejection of the null hypothesis). This is done by applying some *measure of association strength* to the study. These measures represent the amount of variability of the dependent variable explained by the considered factor. In this thesis, the *omega squared* measure ($\hat{\omega}^2$) will be used, and it can be obtained as follows:

$$\hat{\omega}^2 = \frac{df_{between}(F - 1)}{df_{between}(F - 1) + N} \quad (2.15)$$

There is no strict rule to interpret the value of this parameter. As recommended in [71], the effect will be considered to be weak or negligible when $\hat{\omega}^2 \leq 0.01$, medium or relevant enough when $\hat{\omega}^2 \geq 0.10$, and very strong when $\hat{\omega}^2 \geq 0.25$.

All the basic notions on how to perform a one-way ANOVA have already been introduced. Lastly, it is frequent to summarize these results in the form of table 2.1.

Source	SS	df	MS	F	p-value
A	•	•	•	•	•
Within cells	•	•	•		

Table 2.1: Form of the table normally used to represent one-way ANOVA results. Here, A is the considered factor, the term *cells* refers to individual groups and the symbol “•” denotes some data.

2.3.2 N-way ANOVA

In this section, all the previous notions will be generalized for the multi-factorial case, however, some new notions have to be introduced as well. When more than one factor is considered to explain groups conditions, the effect that one of the factors has in presence of a certain value of another may differ from its average effect. This behaviour is called *interaction* between two factors (also referred to as *effect* produced by the combination of two factors) and it has to be analyzed as well as the effects produced by each factor individually (i.e., the *main* effects). Such analysis is known as *two-way* ANOVA. The concept of interaction can be generalized for the case of more than

2.3 Analysis of variance (ANOVA)

two factors as follows. If three factors are considered, these will have a three-way interaction only if some pair of factors has a different two-way interaction depending on the value of the remaining factor (i.e., such pair has interaction for a certain value but not for some other). In general, an n -tuple of factors will have an n -way interaction if there is some subset of $n - 1$ factors that changes its interaction condition depending on the level of the remaining factor. The analysis of all possible interactions among factors (taken in sets of $n, n - 1, n - 2, \dots, 2$) as well as their main effects is known as n -way ANOVA.

The null hypothesis tested in a one-way ANOVA assumes that the corresponding factor has no effect. In an n -way ANOVA, a null hypothesis per possible group of factors is to be tested, one of each assuming the absence of interaction. For the sake of simplicity, the null hypothesis corresponding to a two-way interaction will be stated first. Let A and B be two different factors, one of each with two different possible values. If $\mu_{A_1B_1}$ denotes the mean of the group in the population with $A = 1$ and $B = 1$, the null hypothesis is:

$$H_0 : \mu_{A_1B_1} - \mu_{A_2B_1} = \mu_{A_1B_2} - \mu_{A_2B_2} \quad (2.16)$$

which, as mentioned before, assumes that there is no two-way interaction, denoted as $A \times B$ interaction in this case. This absence of interaction implies that the difference between group means given one value of some factor (in this example, $B = 1$) is not altered when such value changes ($B = 2$). Analogously, if there was a factor C with two levels as well, the null hypothesis for three-way $A \times B \times C$ interaction would be:

$$H_0 : (\mu_{A_1B_1C_1} - \mu_{A_2B_1C_1}) - (\mu_{A_1B_2C_1} - \mu_{A_2B_2C_1}) = (\mu_{A_1B_1C_2} - \mu_{A_2B_1C_2}) - (\mu_{A_1B_2C_2} - \mu_{A_2B_2C_2}) \quad (2.17)$$

which assumes that the magnitude of two-way interaction at different levels of C is the same. The null hypothesis for the remaining levels of interaction can be formulated similarly.

In order to test all the hypotheses, an F statistic for each one must be calculated, but these operations are more elaborate than they were for the one-way ANOVA. For the sake of clarity, only the necessary parameters for a three-way ANOVA will be discussed, since the ones for higher orders can be formulated analogously. Thus, let again A, B and C be the three factors considered in an analysis. The sum of squares within groups would be:

$$SS_{within} = \sum_{j=1}^a \sum_{k=1}^b \sum_{l=1}^c \sum_{i=1}^n (Y_{ijkl} - \bar{Y}_{jkl})^2 \quad (2.18)$$

where a, b and c are the number of levels for each factor, n the number of scores within each group, Y_{ijkl} the i -th score in a group with $A = j, B = k$ and $C = l$, and \bar{Y}_{jkl} is the estimated mean for that group. Note that this expression could be generalized by adding extra indices and sum operators for the remaining factors. Additionally, the degrees of freedom within groups would be:

$$df_{within} = N - abc \quad (2.19)$$

where N is, again, the number of scores in the whole sample and the product abc could be extended with extra factors for higher orders. The expression for the mean square within groups is the same as the one stated in equation (2.12).

The equations to calculate the *between* parameters should be now introduced, however, these ones are more elaborate and also need previous clarifications. First, the term *between* in an

2.3 Analysis of variance (ANOVA)

n -way analysis refers to each studied effect (either interaction or main effect) and therefore, the corresponding result is different for each one. Second, intermediate parameters have to be estimated for each effect, and their expression is different depending on the level of such effect (main, two-way interaction, three-way interaction, etc.). In this case, the parameters for the A main effect are of the form:

$$\hat{\alpha}_j = \bar{Y}_j - \bar{Y} \quad (2.20)$$

where \bar{Y}_j is the estimated mean of the group formed by all scores with $A = j$, and \bar{Y} is, again, the mean of the whole sample. Now it is possible to state the sum of squares for the A main effect as follows:

$$SS_A = \sum_{j=1}^a \sum_{k=1}^b \sum_{l=1}^c \sum_{i=1}^n \hat{\alpha}_j^2 \quad (2.21)$$

Recall that similar parameters $\hat{\beta}$ and $\hat{\gamma}$ have to be obtained for B and C main effects, as well as their corresponding SS_B and SS_C . Regarding the two-way case, the parameters for the $A \times B$ interaction are of the form:

$$(\hat{\alpha}\hat{\beta})_{jk} = \bar{Y}_{jk} - (\bar{Y} + \hat{\alpha}_j + \hat{\beta}_k) \quad (2.22)$$

where \bar{Y}_{jk} is the estimated mean of the group formed by all scores with $A = j$ and $B = k$. The corresponding sum of squares for this effect is:

$$SS_{A \times B} = \sum_{j=1}^a \sum_{k=1}^b \sum_{l=1}^c \sum_{i=1}^n (\hat{\alpha}\hat{\beta})_{jk}^2 \quad (2.23)$$

Note, again, that similar parameters $(\hat{\alpha}\hat{\gamma})$ and $(\hat{\beta}\hat{\gamma})$ have to be obtained for the $A \times C$ and $B \times C$ interactions, as well as their corresponding $SS_{A \times C}$ and $SS_{B \times C}$. Finally, the parameters for the $A \times B \times C$ interaction are of the form:

$$(\hat{\alpha}\hat{\beta}\hat{\gamma})_{jkl} = \bar{Y}_{jkl} - (\bar{Y} + \hat{\alpha}_j + \hat{\beta}_k + \hat{\gamma}_l + (\hat{\alpha}\hat{\beta})_{jk} + (\hat{\alpha}\hat{\gamma})_{jl} + (\hat{\beta}\hat{\gamma})_{kl}) \quad (2.24)$$

where \bar{Y}_{jkl} is the estimated mean for the individual group (or cell) with $A = j$, $B = k$ and $C = l$. The corresponding sum of squares is:

$$SS_{A \times B \times C} = \sum_{j=1}^a \sum_{k=1}^b \sum_{l=1}^c \sum_{i=1}^n (\hat{\alpha}\hat{\beta}\hat{\gamma})_{jkl}^2 \quad (2.25)$$

Note that the pattern shown by equations (2.20), (2.22) and (2.24) could be repeated to add extra factors for higher order analysis.

The degrees of freedom for each effect can be calculated as the product of the number of levels of each factor appearing in such effect minus one. Thus, for the A main effect:

$$df_A = a - 1 \quad (2.26)$$

Analogously, for the $A \times B$ interaction:

$$df_{A \times B} = (a - 1)(b - 1) \quad (2.27)$$

2.3 Analysis of variance (ANOVA)

and for the $AxBxC$ interaction:

$$df_{AxBxC} = (a - 1)(b - 1)(c - 1) \quad (2.28)$$

Note again that the remaining combinations of factors have been omitted, since they are similar and that they can be extended for higher orders.

Every effect considered in an n -way ANOVA has a different F statistic. If the previous sum of squares and degrees of freedom are considered without specifying the concrete effect, i.e., if they are denoted by SS_{effect} and df_{effect} , equations (2.13) and (2.14) can be rewritten in these terms to define both the mean square and the F statistic for the n -way case. As a result:

$$MS_{effect} = \frac{SS_{effect}}{df_{effect}} \quad (2.29)$$

and

$$F_{effect} = \frac{MS_{effect}}{MS_{within}} \quad (2.30)$$

Finally, the null hypothesis for each effect is to be tested by using an F-distribution with df_{effect} and df_{within} degrees of freedom for the numerator and the denominator, respectively. The obtained conclusion about the existence of effect or interaction should be tested with a measure of association strength, like the omega squared value used before. In the n -way case, this parameter is called *partial* omega squared for extrinsic factors (see [49] for more details) and it is different for each effect such that:

$$\hat{\omega}_{partial}^2 = \frac{df_{effect}(F_{effect} - 1)}{df_{effect}(F_{effect} - 1) + N} \quad (2.31)$$

Recall that the same thresholds used in the case of one-way ANOVA to interpret the strength of effects are also applicable here.

All the basic notions on how to perform an n -way ANOVA have already been introduced. Lastly, it is frequent to summarize its results in the form of table 2.2 (a three-way ANOVA table is being showed, for the sake of simplicity).

Source	SS	df	MS	F	p-value
A	•	•	•	•	•
B	•	•	•	•	•
C	•	•	•	•	•
AxB	•	•	•	•	•
AxC	•	•	•	•	•
BxC	•	•	•	•	•
AxBxC	•	•	•	•	•
Within cells	•	•	•		

Table 2.2: Form of the table normally used to represent three-way ANOVA results. Here, A, B and C are the considered factors, the term *cells* refers to individual groups and the symbol “•” denotes some data.

Influence of abnormal sensory behavior in mobile robots

This chapter is aimed at illustrating the importance of sensory abnormalities in the context of mobile robotics, since they represent a problem that goes beyond the thoroughly treated issue of noisy observations. There exist a set of well-known basic tasks in mobile robotics (e.g., localization, mapping, navigation and others) that rely on the information provided by exteroceptive sensors such as rangefinders, the most basic ones in these tasks. The methodologies supporting the mentioned operations are usually based on probabilistic frameworks such as recursive Bayesian filters, which directly tackle the problem of noise but are incapable of capturing abnormal sensory behavior. Given its importance in mobile robotics, this chapter presents a thorough study of the influence of range abnormalities on the performance of Bayesian filters, validating it in a real experiment and also discussing the impact on the robotic operation.

3.1 Introduction

The performance of a mobile robot strongly relies on its own components, generally classified into three different parts. One of them is known as the *motor apparatus*, which refers to all the physical actuators on-board that enable the robot for moving adequately within its environment and also for successfully interacting with this physical world. Another part is constituted by its *sensory apparatus*, which includes all the devices aimed at capturing any aspect related to the state of the robot or the one of its environment. These two parts of the mobile platform are coordinated by another one, represented by its computational capabilities. This refers to all the computer-based systems available, either on-board or remotely, that process all the information from the robot sensors and produce the necessary commands for its actuators.

In this chapter, the attention is focused on potential issues concerning the robot sensory apparatus, since it is essential for an adequate perception of the environment and, in turn, for many

3.1 Introduction

different robotic tasks, including the most basic ones. The wide variety of existing sensory devices for mobile robots can be classified mainly into two groups, namely, *proprioceptive* and *exteroceptive* sensors [1]. The first kind of these devices is intended to measure aspects related to the state of the robot, such as its own position or speed, while the second one aims to capture information about the environment, such as the presence of obstacles, the temperature or lighting conditions, etc. In this last group, the family of *range sensors* is specially relevant. This kind of sensors, also known as *rangefinders*, enables to measure the distance to objects, thus playing a key role in robotic tasks involving recognition of the environment, relative positioning within it, detection of obstacles, and many more [1]. The use of range sensors does not restrict to robotics only, being also present in numerous applications related to industrial manufacturing [72] or autonomous driving [73], among many other fields. Concerning robotics, rangefinders are usually employed in mobile platforms that are required to navigate safely and autonomously. For instance, range sensors are present in mobile robots deployed in industrial, rescue and service tasks [74], [75], [76]. A proper sensory perception is crucial in these contexts; however, it is often challenging for a range sensor to work under real conditions due to the uncertain nature of the environment to be captured and the limitations of the sensor itself.

One of the most well-known shortcomings has to do with the impossibility of getting an exact value of any distance, since all the measurable quantities of the physical world are subjected to some degree of unpredictability. This issue has been extensively treated and it is traditionally addressed by applying *estimation* theory [1]. There exist numerous kinds of estimators depending on the nature of the stochastic process to be considered (please refer to [1] for a more in-depth treatment). In the case of mobile robotics, it is common that the variables that need to be estimated evolve over time, such as the distances measured by a range sensor. The most common dynamic estimators used in robotics are based on Bayesian probability theory, particularly on Recursive Bayesian Estimation (RBE). Among the inference tasks that Bayesian estimation can handle, *filtering* is particularly common in robotics. The concrete methods used for that are employed to solve many different problems, such as localization, navigation and mapping. They are considered essential for a robot to work properly, and the quality of sensory observations is therefore critical for them.

Unfortunately, the impossibility of measuring actual, exact and deterministic distances is not the only issue affecting sensory data from rangefinders. As mentioned before, mobile robots operate in real-world scenarios, where they are exposed to a wide variety of situations that might lead to corrupt sensory data in not fully stochastic ways. These *abnormal* effects, in contrast to *noisy* ones, are often provoked by intrinsic limitations of the sensory apparatus, and related to the measurement principles of physical devices. For instance, a sensor relying on the detection of infrared radiation will not be able to perceive obstacles with transparent or absorbent surfaces, nor operate nominally in conditions of extreme sunlight, leading to saturated or missing observations. Challenging parts of a scene such as corners or columns could also affect ultrasonic sensors, for example, by altering the way that their emitted mechanical waves are reflected, leading to measured distances larger than the actual ones, i.e., biasing them.

This chapter presents a study of the impact of common abnormal range observations on the performance of Bayesian filters [31], given the importance of the latter on the operation of mobile robots, as explained above. There exist already some works in the literature that partially cover the study of filtering performance, such as [42], [43], [44], where the convergence and accuracy of some Bayesian estimators is addressed. From an analytical point of view, these works provide sufficient conditions related to the estimation error and innovation in order to ensure convergence;

3.2 Related research

however, they do not take into account the presence of the abnormal effects that are considered here, which may potentially modify or invalidate the established conditions. They also restrict to the case of a particular filter and do not study any further aspect of the performance. There exist some other works in the literature that address the case of anomalous observations by developing strategies to identify and recover from them, such as [45], [46], [47]; however, these contributions lack a complete analysis of the impact of such sensory data on different aspects of the filtering performance.

The work presented in this chapter aims to address both issues, relying on a thorough statistical study for that. Since the aim is to cover a broad variety of filtering models, the estimation problem is addressed from a generic perspective that allows to abstract from the concrete implementation of any estimator, such as Kalman or Particle filters [77], [78], [79]. For that, the rigorous probabilistic framework provided by dynamic Bayesian networks [80] is employed, since they are capable of representing arbitrary causal relationships among random variables while enabling for generic inference, i.e., they can play the role of any RBF.

The reason why an analytical approach is not advisable for this problem relies on the fact that a large number of parameters have to be considered in order to study a sufficient variety of abnormal situations (e.g., the conditions of the filtering problem, the sensor modelling parameters, the amount and value of anomalous sensory data, etc.). An analytical derivation would be cumbersome under these conditions, and possibly impractical. To solve this issue, from an alternative, statistical approach, the work presented here first analyzes the most common abnormal situations that affect range sensors, defines several parameters that serve to assess the performance of the filters, and also defines the factors (anomalies and system parameters) that are likely to modify such performance. Then, rigorous statistical methodologies are applied to sets of simulated experiments designed to reproduce a wide variety of situations. The obtained results provide complete and relevant conclusions about the effects of dealing with sensory abnormal observations, in a flexible way and without loss of generality. The obtained conclusions are also validated in a real scenario with a mobile robot.

The rest of the chapter is organized as follows. Section 3.2 reviews existing research related to sensory abnormal behavior, Bayesian frameworks for estimation and performance of filters. Section 3.3 describes the design and methodology of the study being carried out about the impact of abnormal range observations on the performance of Bayesian filters, including a description of the procedure followed to obtain simulated data. Finally, section 3.4 provides a complete statistical analysis and experimental validation of the results of the study, both in simulation and in reality.

3.2 Related research

The study of the intrinsic limitations and external abnormal conditions that may affect exteroceptive range sensors has been extensively treated in the literature. However, most of the existing references do not address this issue in isolation; instead, they provide a broader insight ranging from the very physical principles of measurement to concrete applications. One of the first and most complete reviews on sensing technologies in mobile robotics can be found in [81]. More recently, complete classifications of these sensors according to their applications appear in texts such as [82]. Considering the wide variety of existing exteroceptive rangefinders, these classifications could be roughly divided according to two main aspects, namely, the number of spatial dimensions the sensor is able to deal with and the nature of waves it uses (e.g., ultrasonic, electromagnetic, etc.). Measurement principles of most single-direction rangefinders are reported in [1] along with the main limitations they may suffer. Regarding higher dimensional rangefinders, physical working

3.2 Related research

principles are addressed in [83], where the most common abnormal observations they may yield and their causes are also covered, as well as in [1]. Another important aspect to the proposed study concerning sensor modelling is its characterization from a probabilistic perspective, which is tackled in [82] and [1].

Bayesian estimation is a powerful tool for dealing with the noisy nature of the data gathered by these sensors. In general, it can be found within a wide variety of applications in different disciplines such as economics [84], biomedicine [85], physics [86] and engineering [87], among others. This chapter is particularly concerned about its applications in mobile robotics, whose problems have been identified and deeply treated in the literature [88]. The essential tasks that a robot must perform to work properly and autonomously have been addressed successfully in practice from the incorporation of probability theory to robotics in the late 1990s and early 2000s. It is the case of localization [89], navigation [90] and simultaneous localization and mapping (SLAM) [91]. In order to estimate the *pose* (posture) of a robot while navigating within an unknown environment and building a representation of it at the same time, the use of some kind of proprioceptive or exteroceptive sensors is mandatory; exteroceptive range sensors play a role of capital importance in these problems [1].

Under the global denomination of Recursive Bayesian Estimation (RBE), there exist an important variety of concrete implementations of Bayesian estimation depending on the nature of the stochastic process itself and the assumptions made about it. These implementations are usually classified into two broad groups, namely, *parametric* and *non-parametric* filters, depending on whether a known distribution shape for the uncertainties is assumed or not. Developed in the 1960s, the well-known *Kalman Filter* (KF) [77] was the first contribution to the group of parametric filters. Its assumptions consist mainly on the normality of all uncertainties involved and the linearity of the models it represents. As estimator of the state of dynamic systems, it is also referred to as *linear dynamical system* or *state space model* in the literature [57], [48]. Later on, parametric filters allowing the representation of non-linear systems were developed, such as the *Extended Kalman Filter* (EKF) [92], which linearizes such non-linear models while maintaining the assumption of Gaussianity, or the *Unscented Kalman Filter* (UKF) [93], which improves the accuracy of the EKF approximating the original distributions in the non-linear models by using a sampling technique called the *Unscented Transform* [1].

The main limitation of parametric filters relies on the fact that they cannot handle, for instance, uncertainties with multimodal distributions (a mobile robot that estimates that it can be with high probability in one out of several places), and more generally, with non-Gaussian distributions. However, there also have been relevant developments in the scope of non-parametric filters that allow to deal with arbitrary shapes of uncertainty. One of them is the *Histogram Filter* (HF) [89], which is grounded on discrete Bayesian estimation and enables to approximate continuous state spaces (e.g., the so-called *Markov Localization* in mobile robotics). The main drawback is its computational cost, which is usually solved by considering the use of one solution belonging to the family of the *Particle Filters* (PF), the most relevant development in this scope (e.g., *Monte Carlo Localization*). This denomination stands for all those algorithms relying on Monte Carlo simulation methods, which aim to approximate arbitrary distributions by using random observations from them. One of the first concrete implementations of the PF was called *Sequential Importance Sampling* (SIS), which was refined later on with the introduction of *Sequential Importance Resampling* (SIR) [78]. An important drawback of these sampling-based algorithms is their still high computational cost when the dimensionality of the problem is high, which was alleviated by the development of *Rao-Blackwellised Particle Filtering* (RBPF) [79].

3.3 Study of the impact on the performance of Recursive Bayesian Filters

Another area of research related to the study of Bayesian frameworks was producing novel results that would have an important impact on recursive estimation. This area arised from the realm of artificial intelligence in the late 1980s, leading to the framework of Bayesian networks (BNs) [39], which enabled, for the first time, the representation of arbitrarily complex relationships among random variables. Numerous inference algorithms (both exact and approximate) were devised for these models, being one of the most relevant ones the exact *junction tree* or *clique tree* algorithm [54]. However, these models were first conceived for discrete variables and static systems only. The introduction of dynamic Bayesian networks (DBNs) in [80] aimed at incorporating the temporal dimension to such a generic representation tool. This contribution along with the inference methods for both exact and approximate inference over DBNs developed in [57] formed the basis for the connection between reasoning in generalist models and filtering for dynamic systems. For instance, this work showed the relations existing between DBNs and classical models such as Kalman or Particle Filters, presented there as particular cases of the former.

As mentioned before, the main aim of the work presented in this chapter consists in studying the impact of range sensory limitations and abnormalities on the performance of Bayesian filters. There exist related works in the literature that address particular aspects of this issue. On the one hand, some works pursue the identification of abnormal observations and develop solutions to recover from them. This can be seen in [94], where generic anomalous observations are detected and treated for parametric filters. Regarding more specific abnormalities, papers such as [95] and [96] develop robust estimators in the presence of data outliers for parametric and non-parametric filters, respectively. Another common kinds of problematic observations being treated in the literature are the intermittent [45] and biased ones [46]. A work developed in the scope of this thesis [38] also contributes with a solution, based on Bayesian networks, that is able to identify and overcome different kinds of sensory anomalies. On the other hand, and from a more theoretical perspective, there exist analytical approaches that study the optimality, sensitivity and performance of filters in case of modelling errors, such as [97] and [98], while others address their stability and convergence [42], [43], but these works only analyze partial aspects of the filtering performance, such as convergence, doing it without taking into account the effect of possible abnormal observations and also restricting to particular implementations. The work presented here aims to cover a broader variety of filtering models and also to provide a deeper analysis of the performance, using rigorous statistical methods for that.

3.3 Study of the impact on the performance of Recursive Bayesian Filters

This section addresses the two main aspects related to the design and methodology of the study, conceived to assess the impact of abnormal range observations on the performance of Bayesian filters. On the one hand, section 3.3.1 discusses the importance of sensory anomalies in mobile robots and summarizes the most common ones concerning range sensors. Section 3.3.2 states the problem to which the generic Bayesian filter in the form of a DBN will be applied. The filter parameterization is covered as well, along with the inference method to be used. Also, the variables that define the filter performance and the factors that might have some influence on it are introduced in section 3.3.3. All of this constitutes the theoretical aspects to the design of the study. On the other hand, section 3.3.4 presents the statistical methodologies that will be used to derive meaningful and complete conclusions from the study. Lastly, the procedure to be followed to perform it as well as the gathering of performance data is covered in sections 3.3.5 and 3.3.6.

3.3 Study of the impact on the performance of Recursive Bayesian Filters

3.3.1 Sensory anomalies and limitations

Sensory abnormal behavior goes beyond the mere random experiment of obtaining some kind of measurement. As discussed above, noise affecting sensors represents an issue that has been addressed by many different methods including Bayesian recursive estimators, the ones of interest in this case. However, the presence of challenging environmental conditions often has a relevant impact on the behavior of sensory devices, which, due to its own nature, cannot be captured by any of the noise models usually employed in Bayesian filters. Thus, the consequences of such undesirable behavior should be adequately studied, especially regarding the performance of these filters, since they are behind many basic robotic tasks such as localization or mapping. The variety of sensors potentially affected by this problem is wide; for this reason, the attention here will be focused only on exteroceptive rangefinders, one of the most used families of sensors in these and many other robotic tasks. Fortunately, many sensory anomalies are systematic, as long as the same conditions provoking them remain. Studying the concrete effects these anomalies produce on the sensory output is therefore possible, and they will be covered before the analysis of their impact.

According to the literature on range sensors presented in section 3.2, the most common signs of anomalous sensory behaviour appear mainly in the form of biased observations and saturated or missing ones. The first kind is common, for instance, in some ultrasonic rangefinders when placed next to corners or similar surfaces. These sensors rely on the reception of some previously emitted mechanical wave, which would reflect too many times under abnormal conditions before reaching the receptor, thus leading to a detected distance larger than the actual one.

However, this issue does not only affect ultrasonic range sensors, but also the ones relying on infrared radiation. There exist common situations in real environments where mobile robots are placed nearby transparent or highly specular surfaces. As in the case of ultrasonic sensors, these kind of devices usually wait for the reception of a previously emitted pulse of light. This radiation is not sensitive to the case of transparent surfaces, such as windows, therefore ignoring their presence and possibly leading to a larger distance depending on the particular scene behind. Similarly, specular surfaces would deviate this pulse of light towards nearby obstacles, leading again to larger distances than the actual ones depending on the concrete features of the scenario.

The second abnormal issue, i.e., the presence of missing observations, is also common in both ultrasonic and light-based sensors. Under undesirable circumstances, the emitted wave (either mechanical or electromagnetic) could be absorbed or reflected by specific kinds of surfaces in such a way that the receptor is not reached, thus provoking a false indication of free space. There also exist another issue concerning sensors relying on infrared light, related to the presence of external sources of the same radiation. For instance, in conditions of extreme sunlight or heat, the wave emitted by the device would suffer from interferences with the natural radiation, leading again to false indications of free space. In summary, biased and missing observations constitute the two main effects of sensory abnormal behavior affecting range sensors. This knowledge will be used for the design of the study, which is to be discussed in the next sections.

3.3.2 Generic Bayesian networks for filtering range sensors

In order to assess the impact of range abnormal observations on the filtering performance, a problem where rangefinders are used has to be considered. In such a problem, these sensory devices should be employed by Bayesian estimators to access the hidden true distance to some, possibly moving object. This is a common problem in mobile robotics, where the sensor is mounted on-board the robot. Notice that this setting can also fit with many non-robotic applications that

3.3 Study of the impact on the performance of Recursive Bayesian Filters

use rangefinders. For the sake of simplicity, only one-dimensional movement of the obstacle will be considered (along the X axis in this case), since this suffices to cover the common abnormal observations that can occur with a rangefinder. Figure 3.1 shows the conditions of the problem, where x_0 is the initial distance to the obstacle, which moves at a constant speed v in the positive sense of the X axis.

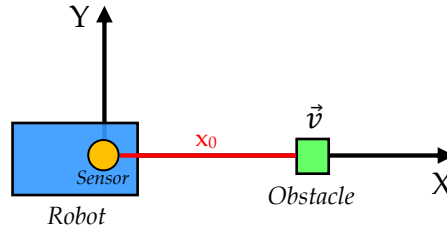


Figure 3.1: General obstacle tracking problem addressed. Here, x_0 represents the initial distance to the obstacle, which moves at a constant speed v in the positive sense of the X axis.

This problem can be solved by using one of the Bayesian estimators reported in Section 3.2, such as the Kalman filter. As explained before, a more generalist approach is pursued and, for this reason, an equivalent estimator in the form of a dynamic Bayesian network will be constructed. For that, two different kinds of variables have to be considered, namely, the ground-truth distance to the obstacle, which is inaccessible (*hidden* variable) and the distance measured by the range sensor, called the *observation*. These variables will be denoted as x_t and z_t , respectively, for a certain time slice t . Since the physical quantities involved in this problem are continuous, the variables used will also be continuous random variables. The model structure corresponds, in this case, to the classical one used in Bayesian estimation for continuous variables, called linear dynamical system (LDS) [48], whose representation in the form of a DBN is depicted in Figure 3.2.

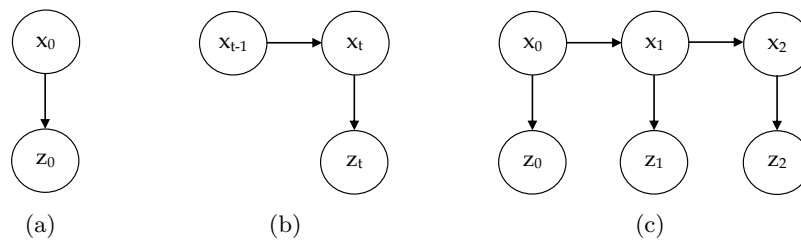


Figure 3.2: Dynamic Bayesian network corresponding to the obstacle tracking problem. Here, variables x represent hidden states (true distances) while variables z represent observations (sensor readings). (a) Initial network \mathcal{B}_0 . (b) Transition network \mathcal{B}_+ . (c) Unrolled DBN for three time slices.

Once the network structure is defined, it is now possible to proceed with its parameterization. In this case, all the variables are continuous and all the corresponding CPDs are assumed to be linear-Gaussian. The LDS model in Figure 3.2 needs three different kinds of CPDs. Firstly, the one for all nodes z is defined, called the *observation* model, i.e., $p(z_t|x_t)$. This CPD encodes the probability distribution of the sensory observation given the true position of the obstacle. In other words, what this CPD represents is the noisy behaviour of the range sensor, which depends on the particular device on use. Such behaviour is often modelled with a truncated normal distribution with the same mean as the true position and some standard deviation depending on the particular sensor, given by the manufacturer in terms of accuracy, which is the error between the measured

3.3 Study of the impact on the performance of Recursive Bayesian Filters

distance and the actual one. The present study aims to cover as much actual sensors as possible. Table 3.1 reports a representative list of commercial rangefinders commonly used in mobile robotics [1]. Images of all those sensors are shown in figure 3.3.

Model Name	Type	Detectable Range	Accuracy
Devantech SRF05 [99]	1D ultrasonic	0.01 to 4 m	4 cm
Sharp GP2Y0A02YK [100]	1D triangulation-based IR	0.2 to 1.5 m	10 cm
Hokuyo URG-04LX-UG01 [101]	2D laser-based	0.06 to 4 m	12 cm
Microsoft Kinect V1 [102], [103]	3D structured-light	0.5 to 4 m	4 cm
SwissRanger SR4000 [104]	3D ToF camera	0.1 to 5 m	10 mm

Table 3.1: Main features of common range sensors in mobile robotics. Here, the accuracy reported is the worst-case error w.r.t the true distance, and the type of sensor includes the number of dimensions and the measurement principle.



Figure 3.3: Images of range sensors commonly used in mobile robots (see table 3.1). (a) Devantech SRF05. (b) Sharp GP2Y0A02YK. (c) Hokuyo URG-04LX-UG01. (d) Microsoft Kinect V1. (e) SwissRanger SR4000.

Based on the accuracy reported for each sensor (table 3.1), its average value is to be used in the simulation framework introduce here, in order to represent most of them. Therefore, the standard deviation for the observation model is $\sigma \approx 6$ cm, i.e., it is being assumed that approximately 68% of the measures will have that error at most. Also, such value has been chosen because $2\sigma \approx 12$ cm, which is the worst accuracy in Table 3.1. This way, all the representative sensors in the table are covered, meaning that 95% of the measures will have that worst error at most. The CPD of the observation model for a given time slice t is:

$$p(z_t|x_t) = \mathcal{N}(x_t, \sigma^2), \quad (3.1)$$

where $\sigma = 0.06$ m (the CPDs will always be parameterized in SI units).

Now the attention is focused on the corresponding CPD for the transition model, that is, $p(x_t|x_{t-1})$. Considering the obstacle tracking problem (figure 3.1), and the lack of any further proprioceptive information on the robot motion¹, the actual distance to the obstacle at a certain time slice t can be expressed in terms of the previous one $t - 1$ with a simple “constant velocity” model:

$$x_t = x_{t-1} + v\Delta t, \quad (3.2)$$

¹In this study, the filters ability to fuse information will not be taken into account.

3.3 Study of the impact on the performance of Recursive Bayesian Filters

where v is the constant speed of the obstacle and Δt is the time span between subsequent slices, also constant. Thus, the CPD for the transition model becomes:

$$p(x_t|x_{t-1}) = \mathcal{N}(x_{t-1} + v\Delta t, \epsilon), \quad (3.3)$$

where ϵ is small because a highly accurate proprioceptive measurement of the speed v is assumed in this model [1].

At this point, the last CPD to be defined corresponds to the prior distribution for the initial state variable x_0 (figure 3.2). Since the actual initial position of the obstacle is assumed to be unknown here, the corresponding CPD must be a normal distribution with a high variance, at least much greater than the variance of the observation model, close to a uniform distribution. The mean has been selected as the average central point of the measurement range for sensors in table 3.1, which is 2 m approximately, and the standard deviation has been defined as 200 times greater than the one for the observations (approximately equal to 12 m). The resulting CPD is:

$$p(x_0) = \mathcal{N}(2, 12^2). \quad (3.4)$$

Once the parameterization of the model is complete, it is possible to perform inference. In the context of dynamic Bayesian networks, there exist different kinds of queries that can be formulated for an inference task (see [57] for a complete review). However, only the one of filtering is considered in the scope of this study. This query consists in calculating the posterior distribution of the current actual position given the whole history of observations, from the initial state up to the present. Such distribution is of the form $p(x_t|\mathbf{z}_{0:t})$, where $\mathbf{z}_{0:t} = \{z_0, z_1, \dots, z_t\}$. The present study is aimed at considering generic filters. For this reason, an inference method called the *interface algorithm* [57] will be employed, since it is able to deal with arbitrary architectures of dynamic Bayesian models.

3.3.3 Filter performance measures and problem characterization

According to [1], there are some important aspects regarding the performance of any kind of Bayesian estimator, namely, how good it is as an approximation to the value of interest, how much uncertainty it has, and how it is expected that it converges to the actual value as more and more observations are gathered. Each of these aspects will be now quantified.

The first one can be defined as the accuracy of the filter, i.e., the error between the predicted value and the actual one. More formally, let μ_t be the actual distance to the obstacle being tracked at time t , and let $\hat{\mu}_t$ be the estimated distance, which corresponds to the mean of the normal distribution represented by the posterior $p(x_t|\mathbf{z}_{0:t})$. The accuracy of the filter e_t at a given time slice t (also called *step* within this scope) is then:

$$e_t = \hat{\mu}_t - \mu_t. \quad (3.5)$$

Note that the value μ_t is non-observable in reality. It can be handled in this simulated statistical study thanks to its nature (please see section 3.4 for further implementation details).

The second aspect to the performance of a Bayesian estimator is its uncertainty, which in this case takes the value of the standard deviation of the normal distribution represented by $p(x_t|\mathbf{z}_{0:t})$. It will be denoted as σ_t , for a given time step t .

3.3 Study of the impact on the performance of Recursive Bayesian Filters

The last aspect to be considered is related to the convergence of estimations to the actual value. Defining a measure that represents convergence is not as straightforward as in the previous cases, and there are several solutions that could be adopted. The term convergence usually refers to the minimum number of steps to be taken in the filtering process such that some desirable behaviour is reached. The characterization of such behaviour is inspired by the time response of dynamical systems [105]. Particularly, it will be considered that a Bayesian estimator converges for a number of steps t^* if the absolute value of the difference between adjacent errors $|e_{t^*} - e_{t^*-1}|$ becomes smaller than a specified threshold and if this still holds for the remaining steps $t \geq t^*$ (note that the full sequence of observations is needed to check that, thus, this must be done offline). The concrete implementation of this measurement as well as the calculation of a proper threshold will be addressed in Section 3.4.

The accuracy and uncertainty have been defined so far as a function of the concrete time step t ; however, the aim is to characterize such performance by using only one value that represents the overall quality of the resulting estimation. For that, the *expected* accuracy and uncertainty of a filter (\bar{e} and $\bar{\sigma}$, respectively) are defined as the values of accuracy and uncertainty that are expected to be achieved when the filtering process has converged (there is no point in considering the case of divergent estimations, since the mentioned values would increase indefinitely). They will be estimated by taking the mean values of accuracy e_t and uncertainty σ_t achieved for the last 10% of time steps in the filtering process.

As a summary, the three measures of performance defined are:

- Expected accuracy of the filter (\bar{e}).
- Expected uncertainty of the filter ($\bar{\sigma}$).
- Minimum number of steps that lead to convergence (t^*).

It is necessary to define a set of factors for the study that might potentially affect these measures of performance. Regarding the context of the problem, a variation in the initial position of the obstacle x_0 or in its speed v might have an impact on some or all the defined measures. Also, the presence of abnormal observations will undoubtedly have an important effect on the performance of estimation, as discussed later on. For these reasons, it will be considered that the factors that are likely to have some kind of impact on the three measures of performance are:

- Initial position x_0 of the obstacle in the tracking problem (figure 3.1).
- Speed v of the obstacle.
- Amount of biased observations (represented as a percentage of the total number of observations).
- Amount of saturated or missing observations (idem).

Determining to which extent these factors or combinations of them change the performance of Bayesian filters is precisely the core of the study. Its concrete implementation will be addressed in sections 3.3.5 and 3.3.6, and its results, in section 3.4.

3.3.4 Statistical tools

The study presented in this chapter relies on statistical tools to analyze the performance of Bayesian filters after carrying out exhaustive simulations. These methodologies are useful to derive conclusions about the different aspects of a certain population, seen as different collections of data obtained under particular conditions. These data are gathered by simulating sequences of observations, i.e., readings from the range sensor, and calculating the corresponding measurements of performance when the filter works on them to estimate the true distance to the obstacle. This is done for as many different conditions as possible combinations of values of the factors mentioned in section 3.3.3 exist. Once gathered, the different groups of data are ordered according to such conditions and then analyzed from a statistical perspective (please refer to section 3.4 for further details). Here, the most suitable tools for the study are described.

One of the best-known descriptive and inferential statistical tools is linear regression [106], which, in this case, serves to model the value of a measurement of performance as a function of the concrete conditions of the simulation, given by specific values of the considered factors. The mentioned model would express the performance as a linear combination of the factors plus an error. Since more than one factor is considered in the study, the concrete methodology would be multiple linear regression. Estimating the parameters of the linear combination is usually solved by applying Least Squares Estimation (LSE) [1], which also provides some measurements of the quality of such estimation. Once these parameters are obtained, they can be interpreted as the relative influence that each factor has on the performance—the higher the absolute value of the parameter, the greater the influence—. However, this is not very reliable since the LSE provides no guarantees on any desirable property of estimators in the general case [1]; thus, this result will only be used as a first approximation and a more in-depth, rigorous analysis of variance (ANOVA) will be then performed (see section 2.3 in chapter 2). This statistical methodology will be adapted for the case of the study presented here. In particular, the considered factors will be the ones previously defined in section 3.3.3, and the dependent variable will correspond to one of the measures of performance stated there. Several n -way ANOVA analyses will be performed in the context of the study, as explained later on. Recall that only two extreme values per factor will be considered, in order to cover a wide range of situations.

3.3.5 Procedure for the analysis and deduction of conclusions

In order to derive meaningful and unambiguous conclusions it is not sufficient to apply only one n -way analysis to the data. A more elaborated procedure based on several applications of ANOVA is required. The main reason is the impossibility of studying the effect of a certain factor or combination of factors in the presence of higher-order interactions involving such factors. To seek both meaningful and concise conclusions, a procedure has been devised, which is formalized in several algorithms, explained in the following.

The results from an n -way ANOVA need to be refined in presence of interactions to interpret effects unambiguously (see algorithm 1). Such refinement can be done by performing different ANOVA analyses, one of each studying a subset of the population restricted to a specific level of some factor (any other but the one of interest). Furthermore, in a higher order analysis this issue should be addressed recursively, since it may arise again in some subset of the population. For this reason, the performance data will always be analyzed taking into account higher orders of interaction first. Recall that the analysis for lower orders is valid as long as the higher ones are proven to not have any interaction. In general, a test of effect will hold as long as the involved factors are a

3.3 Study of the impact on the performance of Recursive Bayesian Filters

Algorithm 1 *testInteractions*(n, s, X, \mathbf{F}, T)

input:

n : level of interaction
 s : significance level
 X : factor of interest
 \mathbf{F} : set of all considered factors
 T : ANOVA table

output:

\mathbf{S} : set of factors involved in an n -way interaction

main:

```
1:  $\mathbf{L} \leftarrow$  set of possible combinations of  $n$  factors from  $\mathbf{F}$  containing  $X$ 
2:  $\mathcal{S} \leftarrow \emptyset$ 
3: for  $i = 1$  to  $|\mathbf{L}|$  do
4:    $p_i \leftarrow$  p-value from interaction test involving factors in  $\mathbf{L}_i \subset \mathbf{L}$  using  $T$ 
5:   if  $p_i < s$  then
6:      $\hat{\omega}^2 \leftarrow$  omega squared value for interaction involving factors in  $\mathbf{L}_i$ 
7:     if  $\hat{\omega}^2 \geq 0.10$  then
8:        $\mathcal{S} \leftarrow \mathcal{S} \cup (\mathbf{L}_i, p_i)$ 
9:     end if
10:  end if
11: end for
12:  $\mathbf{S} \leftarrow$  search for the  $\mathbf{L}_i$  in  $\mathcal{S}$  with the lowest  $p_i$ 
13: return  $\mathbf{S}$ 
```

subset of a valid higher-order test. The greater the number of interactions, the longer the procedure.

Conclusions about all the existing factors can be obtained once the full analysis is complete. These conclusions will always refer to exactly one factor along with a set of restrictions on the others, which would be empty in the case that the conclusion holds for all groups. The union of all conclusions for a factor must cover the entire sample. For instance, in a four-way analysis of a population of values of a performance measure gathered for our problem, using factors A (initial position of the obstacle), B (amount of missing sensory data), C (amount of biased sensory data) and D (speed of the obstacle), each one with two possible levels (low and high), complete sets of conclusions for factors could be like the following ones:

- Factor B (missing data) has effect on the expected uncertainty of the filter.
- Factor C (biased data) has effect on the convergence of the filter given that B takes its low value; factor C (obstacle speed) has no effect given that B takes its high value.
- Factor D has no effect on the expected accuracy performance of the filter given that C takes its low value; factor D has no effect given that B takes its low value and C its high value; factor D has no effect given that both B and C take their high values.

In the first item, only one conclusion suffices to explain the effect for any group in the population. Each conclusion in the second item holds for any combination of levels of factors A and D . The union of the conclusions in the third item also covers all the population groups. It is always a good practice to check the form of the resulting subset of the population expressed by a conclusion. In this study, such conclusion will be accepted only if all of the population subsets are normally distributed (or approximately normal) and will be discarded otherwise (e.g., in case of

3.3 Study of the impact on the performance of Recursive Bayesian Filters

Algorithm 2 *forceInteraction*($n, a, f, X, \mathbf{F}, error, \mathcal{S}$)

input:

n : level of interaction
 a : number of possible interactions
 f : number of attempts
 X : factor of interest
 \mathbf{F} : set of all considered factors
 $error$: boolean value indicating the impossibility of obtaining valid conclusions
 \mathcal{S} : set of pairs of factors (auxiliary variable)

output:

\mathbf{S} : set of factors involved in an n -way interaction
 Z : factor for the study of interaction involving factors in \mathbf{S}
Updates of variables $n, a, f, error$ and \mathcal{S}

subroutines:

generateInteractions(n, X, \mathbf{F}):

$\{(\mathbf{S}_i, Z_i)\} \leftarrow$ set of all possible pairs of n -way interactions $\mathbf{S}_i \subset \mathbf{F}$ and factors $Z_i \in \mathbf{S}_i \setminus \{X\}$
 $\mathcal{S} \leftarrow \{(\mathbf{S}_i, Z_i)\}$

return \mathcal{S}

getInteraction(f, \mathcal{S}):

$(\mathbf{S}, Z) \leftarrow$ search for f -th pair in \mathcal{S}

return (\mathbf{S}, Z)

main:

```
1:  $f \leftarrow f + 1$ 
2: if ( $n = 1$  or  $f = a + 2$ ) and  $n \leq |\mathbf{F}|$  then
3:    $n \leftarrow n + 1$ 
4: else if  $n > |\mathbf{F}|$  then
5:    $error \leftarrow true$ 
6: end if
7: if ( $f = 1$  or  $f = a + 2$ ) and  $error = false$  then
8:    $\mathcal{S} \leftarrow generateInteractions(n, X, \mathbf{F})$ 
9:    $a \leftarrow |\mathcal{S}|$ 
10:   $f \leftarrow 2$ 
11:  if  $a = 0$  then
12:     $error \leftarrow true$ 
13:  end if
14: end if
15: if  $error = false$  then
16:   $(\mathbf{S}, Z) \leftarrow getInteraction(f, \mathcal{S})$ 
17: else
18:   $\mathbf{S} \leftarrow \emptyset$ 
19:   $Z \leftarrow \emptyset$ 
20: end if
21: return  $(\mathbf{S}, Z, n, a, f, error, \mathcal{S})$ 
```

multimodality). In the latter, all the necessary analyses are revisited, from the lower levels, and some non-existent interactions are *forced* so that the partition of the population gets more specific and, hopefully, more normal. It is also taken into account that conclusions should be as concise as possible (see algorithm 2).

3.3 Study of the impact on the performance of Recursive Bayesian Filters

Algorithm 3 ANOVA($s, X, \mathbf{F}, \mathbf{Y}, \mathbf{R}$)

input:

s : significance level

X : factor of interest

\mathbf{F} : set of all considered factors in the population

\mathbf{Y} : population data indexed by levels of factors in \mathbf{F} (i.e., $\mathbf{Y} = \{Y_{11\dots 1}, Y_{11\dots 2}, \dots\}$)

\mathbf{R} : set of restrictions on the entire population

output:

\mathbf{C} : set of conclusions for factor X

main:

```

1:  $\mathbf{C} \leftarrow \emptyset, p \leftarrow \emptyset, \mathcal{S} \leftarrow \emptyset, f \leftarrow 0, a \leftarrow (-2), n \leftarrow |\mathbf{F}|, error \leftarrow false$  (initialization)
2:  $T \leftarrow$  perform an  $n$ -way ANOVA over population  $\mathbf{Y}$  and store its corresponding table
3: while  $error = false$  and conclusions in  $\mathbf{C}$  do not cover the entire population do
4:   if  $n > 1$  then
5:     if  $f = 0$  then
6:        $\mathbf{S} \leftarrow testInteractions(n, s, X, \mathbf{F}, T)$ 
7:     end if
8:     if  $\mathbf{S} = \emptyset$  then
9:        $n \leftarrow n - 1$ 
10:    else
11:      if  $f = 0$  then
12:         $Z \leftarrow$  choose one factor in  $\mathbf{S} \setminus \{X\}$ 
13:      end if
14:       $l \leftarrow$  number of levels of factor  $Z$ 
15:      for  $i = 1$  to  $l$  do
16:         $\mathbf{D} \leftarrow$  subset of the population  $\mathbf{Y}$  verifying  $Z = i$ 
17:         $\mathbf{R} \leftarrow \mathbf{R} \cup \{Z = i\}$ 
18:         $\mathbf{C} \leftarrow \mathbf{C} \cup ANOVA(s, X, \mathbf{S} \setminus \{Z\}, \mathbf{D}, \mathbf{R})$ 
19:         $\mathbf{R} \leftarrow \mathbf{R} \setminus \{Z = i\}$ 
20:      end for
21:      if conclusions in  $\mathbf{C}$  do not cover the entire population then
22:         $(\mathbf{S}, Z, n, a, f, error, \mathcal{S}) \leftarrow forceInteraction(n, a, f, X, \mathbf{F}, error, \mathcal{S})$ 
23:      end if
24:    end if
25:  else
26:    if population is not normal for all levels of factor  $X$  given  $\mathbf{R}$  then
27:       $(\mathbf{S}, Z, n, a, f, error, \mathcal{S}) \leftarrow forceInteraction(n, a, f, X, \mathbf{F}, error, \mathcal{S})$ 
28:    else
29:       $p \leftarrow$  p-value from main effect test for factor  $X$  using  $T$ 
30:       $\hat{\omega}^2 \leftarrow$  omega squared value for main effect  $X$ 
31:      if  $p < s$  and  $\hat{\omega}^2 \geq 0.10$  then
32:         $\mathbf{C} \leftarrow \mathbf{C} \cup \{ \text{Factor } X \text{ has effect given } \mathbf{R} \}$ 
33:      else
34:         $\mathbf{C} \leftarrow \mathbf{C} \cup \{ \text{Factor } X \text{ has no effect given } \mathbf{R} \}$ 
35:      end if
36:    end if
37:  end if
38: end while
39: return  $\mathbf{C}$ 

```

3.3 Study of the impact on the performance of Recursive Bayesian Filters

Considering all of the above, the procedure that is followed in this study can be formally established as described in algorithm 3, which is to be run once per each factor. Since this procedure might be cumbersome, a tree graph that encodes the steps followed by the algorithm is provided, for the sake of clarity. In that graph, nodes represent groups of n factors involved in a potential n -way interaction. In case of no interaction, the graph contains arcs annotated with the factor that will not be considered for the lower order interaction analysis. In case of interaction, one or more arcs are used, each annotated with a specific value of the factor that will be fixed to study the lower interaction or main effect, thus specifying an additional restriction on the population groups. Recall that each of these arcs indicate that a new ANOVA table has been obtained for the studied interaction with the specified restrictions. Finally, the nodes with only one factor indicate that a valid conclusion has been reached on the main effect of the corresponding factor. More complicated cases are also represented, such as rejected conclusions due to multimodal data and forced interactions, providing as well alternative graphs (below the rejected ones) in order to derive the affected conclusions properly.

As an example of this graph, consider the analysis for the population obtained for the expected accuracy performance of the filter, where the four factors mentioned in section 3.3.3 have been used, namely, A (initial position of the obstacle), B (amount of missing range data), C (amount of biased range data) and D (speed of the obstacle). The necessary tree graph for the analysis of factor A is shown in figure 3.4 (see section 3.4.2 for further details on such analysis).

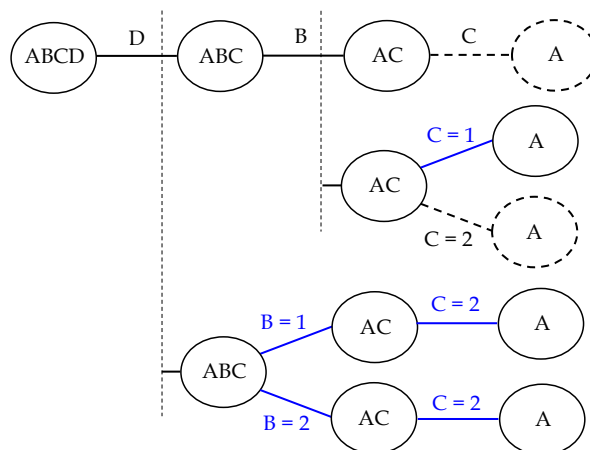


Figure 3.4: Tree graph for the analysis of the effect of the possible values of factor A (initial position of the obstacle) on the expected accuracy performance of the filter. Dashed nodes and arcs correspond to rejected conclusions due to multimodal populations. Arcs in blue denote decisions on the value of factors based on interactions that are forced to get unimodality in the data. Here, “1” and “2” refer to specific levels of the factors.

3.3.6 Gathering data

The statistical study presented in this chapter relies on simulated data in order to reproduce a wide variety of conditions in real environments, to make the number of simulated tests arbitrarily large, and to always have access to the truth state of the system, which in the end is what enables performance measurement and comparison. These simulations are to be performed under the conditions defined by the factors considered in section 3.3.3. Thus, there will be one simulation related to each possible combination of their values, and the data for each performance measure will be divided into different groups according to these conditions. For the reasons given in section 3.3.4, only two levels for each factor will be considered, which covers their entire range. The concrete

3.3 Study of the impact on the performance of Recursive Bayesian Filters

values are provided in table 3.2.

Factor	Meaning	Low Value "1"	High Value "2"
A	Initial distance to obstacle (x_0)	1 m	2 m
B	Amount of missing range data	0%	95%
C	Amount of biased range data	0%	75%
D	Obstacle speed (v)	0 m/s	0.2 m/s

Table 3.2: Factors influencing the performance of Bayesian range filters along with the concrete values that they can take, including both scenario parameters and sensor anomalies. In this study, "1" and "2" will be used to refer to the low and high values of the factors respectively.

The first step in collecting the performance data from the filter consists in simulating sequences of observations from the range sensor obtained under a particular combination of factor values. In this work, 100 time steps will be considered for studying the filter, each of them representing fixed increments of $\Delta t = 100$ ms (that value has been chosen for being a suitable sampling time in robotic applications). Each simulated observation is obtained as a random value drawn from a normal distribution with the same mean as the true distance for the corresponding time step and the standard deviation considered for the observation model, that is, $\sigma = 0.06$ m (see equation (3.1)). This vector is then corrupted, if necessary, with biased and/or missing observations placed at random time steps to simulate the anomalies. In these cases, the distribution of observations may differ from a normal one. To illustrate this with an example, several sequences of random observations have been simulated by using a normal distribution with 1 meter of mean, and some of these observations have been then corrupted with different combinations of anomalies. Figure 3.5 shows a collection of histograms, each one corresponding to a particular sequence. In these simulations, the speed of the obstacle is assumed to be null.

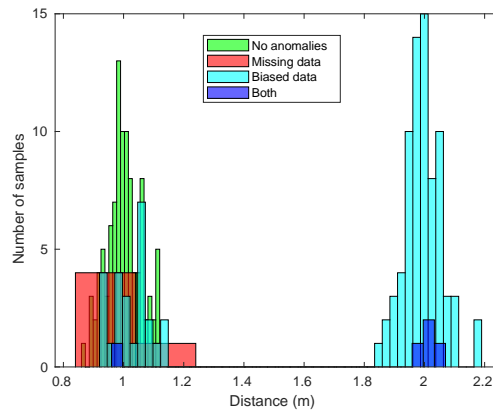


Figure 3.5: Histograms for sequences of range observation data. These sequences have been obtained from a normal distribution with 1 meter of mean; some of them have been corrupted with different combinations of anomalies.

As depicted in figure 3.5, when there is an important amount of biased data, the distribution becomes bi-modal, centering in both the original measurement and in the biased one. When there is a high amount of missing data, the lack of observations modifies the shape of the sampled distribution, but there is no reason to affirm that it is not normal. With the combination of the two anomalies, the mentioned effects are also combined: the bias leads to a multi-modal distribution, which is still locally normal despite the lack of data.

3.3 Study of the impact on the performance of Recursive Bayesian Filters

Algorithm 4 *dataCollection*

output:
 \mathcal{E}_{ABCD} : data concerning filter accuracy, indexed by combinations of values of factors
 \mathcal{U}_{ABCD} : data concerning filter uncertainty, indexed by combinations of values of factors

main:

- 1: $n \leftarrow 500$ (number of experiments for each combination of factors)
- 2: $T \leftarrow 100$ (number of total time steps considered)
- 3: $\Delta t \leftarrow 0.1$ (sampling time in seconds)
- 4: $\sigma \leftarrow 0.06$ (standard deviation in meters considered for the observation model)
- 5: $\mathcal{E}_{ABCD} \leftarrow \emptyset$
- 6: $\mathcal{U}_{ABCD} \leftarrow \emptyset$
- 7: **for each** possible combination ABCD of values of factors **do**
- 8: **for** $i=1$ to n **do**
- 9: $\mathbf{g} \leftarrow \emptyset$ (vector of ground-truth distances)
- 10: $\mathbf{z} \leftarrow \emptyset$ (vector of observations)
- 11: **for** $t=0$ to T **do**
- 12: $x_0 \leftarrow$ initial distance to obstacle (from value of factor A)
- 13: $v \leftarrow$ obstacle speed (from value of factor D)
- 14: $\mathbf{g}(t) \leftarrow x_0 + t \cdot v\Delta t$
- 15: $\mathbf{z}(t) \leftarrow$ random value drawn from distribution $\mathcal{N}(\mathbf{g}(t), \sigma^2)$
- 16: **end for**
- 17: **if** B=2 **then** // *absorption anomaly*
- 18: $\mathbf{z} \leftarrow$ corrupt current vector \mathbf{z} with 95% of empty observations at random positions
- 19: **end if**
- 20: **if** C=2 **then** // *bias anomaly*
- 21: $\mathbf{z} \leftarrow$ corrupt current vector \mathbf{z} by adding 1 m to 75% of non-empty positions randomly
- 22: **end if**
- 23: $\hat{\boldsymbol{\mu}} \leftarrow \emptyset$ (vector of estimated distances)
- 24: $\hat{\boldsymbol{\sigma}} \leftarrow \emptyset$ (vector of standard deviations for estimated distances)
- 25: **for** $t = 1$ to T **do**
- 26: $p(x_t|\mathbf{z}_{0:t}) \leftarrow$ calculate filter posterior by using the interface algorithm (see section 3.3.2)
- 27: $\hat{\boldsymbol{\mu}}(t) \leftarrow$ mean of the normal distribution represented by $p(x_t|\mathbf{z}_{0:t})$
- 28: $\hat{\boldsymbol{\sigma}}(t) \leftarrow$ standard deviation of the normal distribution represented by $p(x_t|\mathbf{z}_{0:t})$
- 29: $\mathcal{E}_{ABCD}(i, t) \leftarrow \hat{\boldsymbol{\mu}}(t) - \mathbf{g}(t)$
- 30: $\mathcal{U}_{ABCD}(i, t) \leftarrow \hat{\boldsymbol{\sigma}}(t)$
- 31: **end for**
- 32: **end for**
- 33: **end for**
- 34: **return** ($\mathcal{E}_{ABCD}, \mathcal{U}_{ABCD}$)

Once the necessary observations are simulated, it is possible to infer posterior distributions of the form $p(x_t|\mathbf{z}_{0:t})$, from $t = 1$ to 100, and measure the accuracy and uncertainty of the filter for each t . The inference task in the filter is performed by applying the *interface* algorithm [57] (the implementation used is the one available in the *Bayes Net Toolbox* (BNT) for Matlab [107]). Since the aim is to generate a reasonable amount of data, this simulated experiment is repeated 500 times for each combination of factor values. Algorithm 4 details the procedure explained in this paragraph.

Once the necessary performance data have been collected, it is possible to synthesize the three measures of performance of interest. Firstly, a threshold for the third one has to be considered, filter convergence (see section 3.3.3). If such value is too high, nearly all tests in the data will

3.3 Study of the impact on the performance of Recursive Bayesian Filters

Algorithm 5 *performanceData*($\mathcal{E}_{ABCD}, \mathcal{U}_{ABCD}$)

input:

\mathcal{E}_{ABCD} : population data for filter accuracy, indexed by combinations of values of factors
 \mathcal{U}_{ABCD} : population data for filter uncertainty (indexed as above)

output:

$\bar{\mathcal{E}}_{ABCD}$: population data for the expected accuracy performance (indexed as above)
 $\bar{\mathcal{U}}_{ABCD}$: population data for the expected uncertainty performance (indexed as above)
 \mathcal{C}_{ABCD} : population data for the convergence performance (indexed as above)

main:

```

1:  $n \leftarrow 500$  (number of tests in each population group)
2:  $C \leftarrow \emptyset$  (set of ordered indices of converging tests)
3:  $\mathbf{f} \leftarrow \emptyset$  (vector of filtered accuracy values)
4:  $\bar{e}_{ABCD} \leftarrow \emptyset$  (temporary population data for expected accuracy)
5:  $\bar{u}_{ABCD} \leftarrow \emptyset$  (temporary population data for expected uncertainty)
6:  $c_{ABCD} \leftarrow \emptyset$  (temporary population data for convergence)
7:  $m \leftarrow 0$ 
8:  $m_s \leftarrow 100$ 
9: for each possible combination ABCD of values of factors do
10:   for  $i=1$  to  $n$  do
11:      $\bar{e}_{ABCD}(i) \leftarrow$  average of accuracy values in vector  $\mathcal{E}_{ABCD}(i)$  for the last 10 time steps
12:      $\bar{u}_{ABCD}(i) \leftarrow$  average of uncertainty values in vector  $\mathcal{U}_{ABCD}(i)$  for the last 10 time steps
13:      $\mathbf{f} \leftarrow$  apply a 5-th order median filter to vector  $\mathcal{E}_{ABCD}(i)$ 
14:     if  $\exists t^* : |\mathbf{f}(t) - \mathbf{f}(t-1)| \leq 0.0038$  ( $\forall t \geq t^*$ ) then
15:        $C \leftarrow C \cup \{i\}$ 
16:        $c_{ABCD}(i) \leftarrow t^*$ 
17:     end if
18:   end for
19:   for  $j = 1$  to  $|C|$  do
20:      $i \leftarrow j$ -th element of  $C$ 
21:      $\bar{\mathcal{E}}_{ABCD}(j) \leftarrow \bar{e}_{ABCD}(i)$ 
22:      $\bar{\mathcal{U}}_{ABCD}(j) \leftarrow \bar{u}_{ABCD}(i)$ 
23:      $\mathcal{C}_{ABCD}(j) \leftarrow c_{ABCD}(i)$ 
24:   end for
25:    $m \leftarrow |C|$ 
26:   if  $m < m_s$  then
27:      $m_s \leftarrow m$ 
28:   end if
29: end for
30: Discard tests randomly such that  $|\bar{\mathcal{E}}_{ABCD}| = |\bar{\mathcal{U}}_{ABCD}| = |\mathcal{C}_{ABCD}| = m_s$  for any ABCD
31: return ( $\bar{\mathcal{E}}_{ABCD}, \bar{\mathcal{U}}_{ABCD}, \mathcal{C}_{ABCD}$ )

```

converge, and, if it is too low, virtually no tests will converge. Under none of these circumstances it is possible to study the filter convergence adequately, thus the proper value must be a tradeoff between the number of converging tests and the usefulness of the resulting data for the study. After some trials, a threshold that allows for at least 45% of converging tests out of a total of 500 for each combination of values of factors has been chosen. Such a threshold corresponds to a maximum difference of 0.0038 m (3.8 mm) between adjacent accuracies.

Algorithm 5 illustrates the procedure to synthesize all the measures of performance according to their definition in section 3.3.3 from the data collected by algorithm 4. All the resulting popu-

3.3 Study of the impact on the performance of Recursive Bayesian Filters

lation groups have to contain the same number of elements, i.e., be balanced (this will be clarified later on). After discarding those tests that are not converging, the 16 groups for each performance (one for each possible combination of factors) finally have 304 elements each.

Once the performance measures are obtained, they have to be validated in order to determine whether the necessary requirements to apply the statistical methodologies used for the study are fulfilled. Recall that for ANOVA it is necessary that the obtained data for each population group is normally distributed, which is also a desirable condition for the Least Squares Estimator [1]. In figures 3.6, 3.7 and 3.8, histograms for the results of expected accuracy, expected uncertainty and convergence are shown, respectively. Recall that there are 16 groups with 304 tests each for each performance.

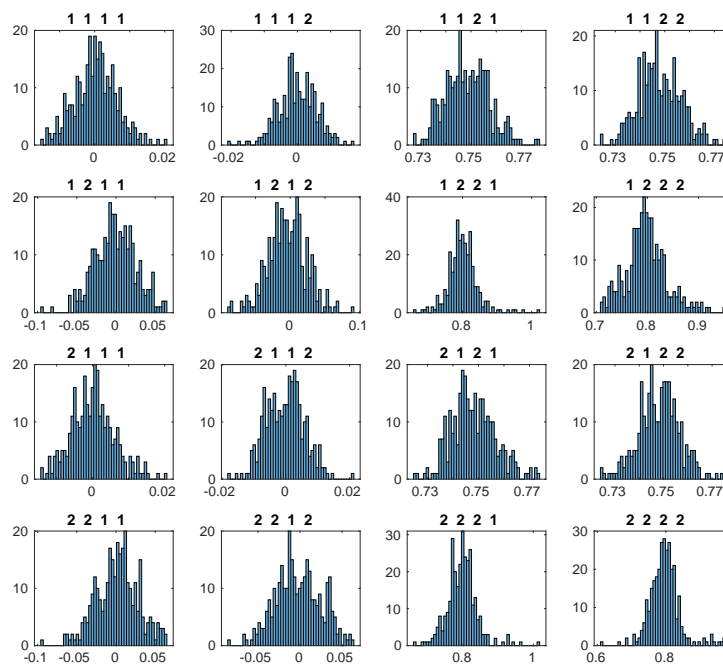


Figure 3.6: Histograms for all the population groups corresponding to the performance measure of the expected accuracy of the filter. Each sequence of four numbers in the figure titles represent the concrete combination of values for factors ABCD (see table 3.2). The horizontal axes represent the expected accuracy in meters, and the vertical ones, the number of tests.

The results in figure 3.6 are to some extent similar to the shape of a normal distribution, but the data in figure 3.7 have very little variation, i.e., they are more similar to a Delta function. This is due to the fact that the estimated uncertainty in a filtering process does not depend on the concrete values of the observations, but on the number of them, among other properties [1]. As a consequence, changes will only be noticeable when the number of observations vary or, at least, when the time step they are acquired is different, from test to test. This issue does not prevent from applying the mentioned statistical methods to these data, as discussed later on. From the results in figure 3.8, it can be noted that groups in this performance measure present a skewed shape. Some de-skewing processes have been applied, but the resulting shapes gets not much better. Fortunately, ANOVA is generally robust to these kind of non-normalities [49].

Another requirement that must be satisfied is the *homoscedasticity* of variances, that is, the variances of all population groups cannot vary across the means of such groups. The mean and

3.3 Study of the impact on the performance of Recursive Bayesian Filters

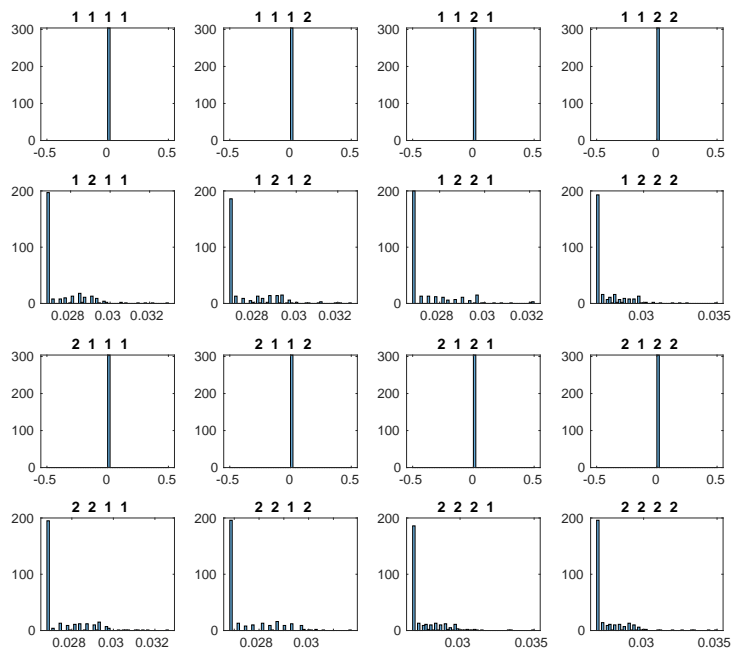


Figure 3.7: Histograms for all the population groups corresponding to the performance measure of the expected uncertainty of the filter. The horizontal axes represent the expected uncertainty in meters, and the vertical ones, the number of tests.

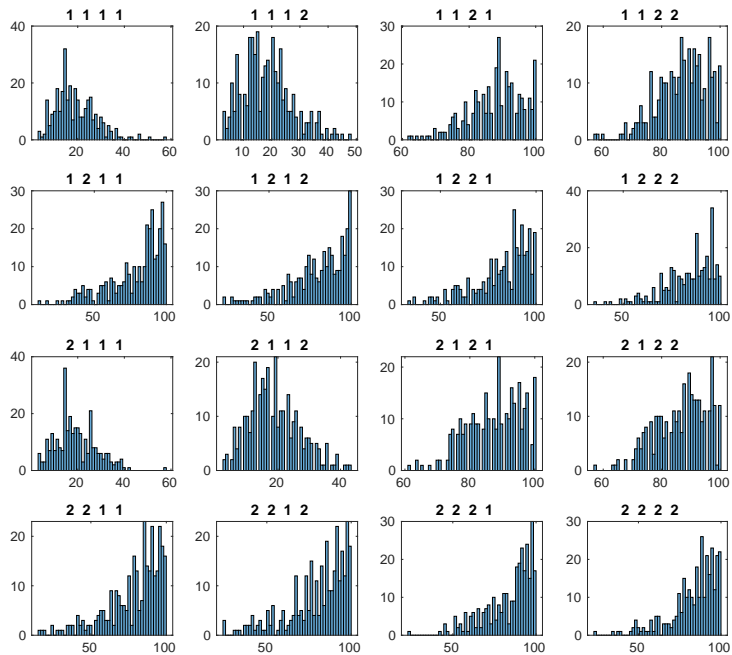


Figure 3.8: Histograms for all the population groups corresponding to the performance measure of the convergence of the filter. The horizontal axes represent the minimum number of steps t^* that lead to convergence, and the vertical ones, the number of tests.

variance have been calculated for each group for all the measurements of performance. The results are depicted in figure 3.9. Note that there are 16 points to each graph, one per group. These results show that the required condition is not strongly violated. ANOVA is also relatively robust to mild violations of this criterion [49].

3.4 Results of the study and discussion

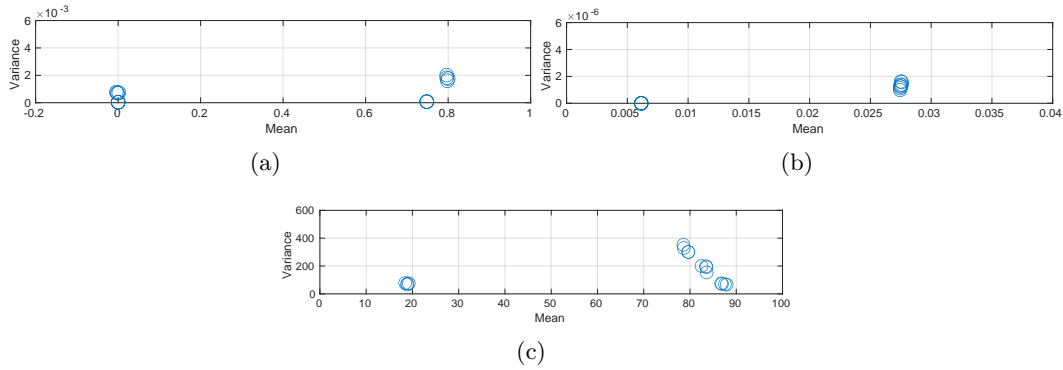


Figure 3.9: Homoscedasticity of population variances for the performance measures of the range filter. (a) Expected accuracy. (b) Expected uncertainty. (c) Convergence.

3.4 Results of the study and discussion

This section states the main hypotheses that are to be tested with the study, discusses the obtained results from the application of rigorous statistical methods, and presents experimental validation of such results in a real environment with a mobile robot.

3.4.1 Statement of hypotheses

In section 3.3.3, the factors that are likely to have some kind of impact on the measures of performance of the filter were defined. These definitions are here translated into the hypotheses discussed below and, as intuitive affirmations of the behaviour of Bayesian filters, will be confirmed or rejected by the study.

Firstly, it is reasonable to think that, in the context of distance estimation, the conditions of the general tracking problem (i.e., the initial position of the obstacle and its speed) might have some kind of impact on any of the measures of performance. They have a clear influence on the way that the distances under study evolve over time, and therefore could modify the estimation error or even affect the convergence rate.

Observation anomalies must certainly affect the observation model of the filter, since they are not expected nor contemplated in the models of reality it implements. For instance, missing observations produce a lack of data for calculating the filtering estimation given by the posterior distribution $p(x_t | \mathbf{z}_{0:t})$, and force the filter to work only with prior predictions. In this situation, the estimations could diverge in the case that the transition model was not close enough to reality, increasing without limit the estimation error. If the filter does not diverge, these anomalies would increase progressively the uncertainty of the estimate by the injection of the system motion uncertainty at each step. Following the analytical formulation for convergence reported in [42], in that case the Lyapunov function V_k , used for defining the closeness of the filter to convergence, becomes larger, therefore making convergence slower.

In the case of bias anomalies, observations still arrive, but the filter is—unknowingly—using a model that is biased w.r.t reality. That perturbation makes the filter to predict observations farther away from actual ones at each affected step, which has consequences on the error in the estimate. Function V_k is affected by that increased error, getting larger values which, again, would make convergence slower.

3.4 Results of the study and discussion

3.4.2 Statistical analysis

The statistical analysis begins with multiple linear regression. The observed values of each measure of performance Y are estimated in this case as a linear combination of the values of the factors A , B , C and D (recall table 3.2). This method aims to minimize the error between the observed values and the predicted ones \hat{Y} , which is expressed as follows:

$$\hat{Y} = \beta_0 + \beta_1 A + \beta_2 B + \beta_3 C + \beta_4 D \quad (3.6)$$

where $\beta_0, \dots, \beta_4 \in \mathbb{R}$. The results for the three measures of performance of the filter are detailed in table 3.3, along with the quality of their estimation, given by the R^2 statistic [106].

Factor	Parameter	Expected Accuracy	Expected Uncertainty	Convergence
-	β_0	-0.0119	0.0061	34.6577
A	β_1	-0.0001	0.0000	0.1509
B	β_2	0.0253	0.0225	29.6187
C	β_3	1.0316	0.0000	48.3876
D	β_4	-0.0064	0.0000	0.3762
-	R^2	0.9945	0.9943	0.5563

Table 3.3: Multiple linear regression coefficients obtained for the three measures of performance of the range filter and quality of their estimations (R^2). Maximum values are highlighted in bold.

From these results, some interesting conclusions can already be derived by focusing on those parameters with the highest value for each performance. It could be stated, for instance, that factor C (amount of biased range data) is the most relevant for the filter expected accuracy, implying that the greater the amount of these observations, the worse that accuracy, which is pretty intuitive. Factor B (amount of missing range data) seems to have a clear impact on the uncertainty, that is, a greater number of missing observations hinders the reduction of uncertainty in the filtering process. Lastly, factors B and C are estimated to be relevant for the convergence performance, which is also plausible, since a lower amount of available data usually leads to slower convergence rates (and the same holds for an increase in the amount of biased observations).

Although these conclusions are reasonable and expected, the magnitudes of the coefficients also provide information that is not that obvious. For instance, factors B and C are near 2 orders of magnitude more important than the rest in expected accuracy, and the same holds for factor B in uncertainty. In convergence, these factors share their relevance with the influence of the β_0 parameter. This parameter is not related to any factor, but accounts for the importance of those effects that are not explicitly treated in the analysis, i.e., it represents the portion of the performance value that is not explained by the considered factors. This parameter has not a relevant influence on the expected accuracy nor on the uncertainty; however, it is important for convergence, which indicates that there are a number of influences on convergence that are beyond our study of abnormal sensor observations. In this case, the value of β_0 says that, in absence of abnormal observations (represented by factors B and C), the average convergence rate is around approximately 35 steps (see the population groups for convergence in figure 3.8).

Notice that in this regression analysis there are still information that is not elucidated, like the interaction effects among factors. For a more detailed study, the hypothesis testing procedure explained in section 3.3.5 has been applied. Notice, however, that as shown in figure 3.7, all the

3.4 Results of the study and discussion

obtained data for the expected uncertainty of the filter are identical when there are no missing observations (i.e., when factor B takes its low value) for the reasons explained in section 3.3.6; thus, it does not make sense to perform ANOVA in that case, but just conclude that none of the factors have any effect in the expected uncertainty of the filter when $B = 1$.

For the sake of brevity in the following, the attention is focused on the explanation of the final results; the necessary tree graphs for the analyses carried out along with the corresponding ANOVA tables and histograms for obtaining these conclusions are fully reported in appendix A. In short, a total of 12 analyses have been carried out, 4 per each measure of performance of the filter.

Table 3.4 provides a complete summary of the conclusions obtained for each factor. Firstly, factors A and D, which define the parameters of the tracking scenario, i.e., the initial position of the obstacle and its speed, are statistically assessed not to affect any measure of performance, regardless of their values. This is compatible with the results obtained by the multiple linear regression method, and it is plausible, since there is no reason to consider that the concrete values of the gathered distances (or the rate at which they vary) have any undesirable effect on the steady-state performances, providing that they reproduce reality adequately (i.e., they are not obtained under anomalous conditions).

Factor	Expected Accuracy			Expected Uncertainty		Convergence		
A (obstacle position)	C = 1	$\mu(\text{---})$: -6.82×10^{-4} m σ : 0.02, 0.02		B = 1	$\mu(\text{---})$: 0.006 m σ : 0, 0	B = 1	C = 1	$\mu(\text{---})$: 19 steps σ : 8, 8
	C = 2	B = 1	$\mu(\text{---})$: 0.75 m σ : 0.009, 0.009			B = 2	$\mu(\text{---})$: 0.028 m σ : 0.001, 0.001	C = 2
		B = 2	$\mu(\text{---})$: 0.80 m σ : 0.041, 0.044	B = 2	$\mu(\text{---})$: 0.028 m σ : 0.001, 0.001	B = 2	$\mu(\text{---})$: 81 steps σ : 16, 16	
B (% of missing data)	C = 1	$\mu(\text{---})$: 0.003 m σ : 0.006, 0.027		$\mu(\uparrow)$: 0.02 m σ : 0, 0.001		C = 1	$\mu(\uparrow)$: 60 steps σ : 8, 18	
	C = 2	$\mu(\uparrow)$: 0.05 m σ : 0.009, 0.040				C = 2	$\mu(\text{---})$: 85 steps σ : 8, 14	
C (% of biased data)	B = 1	$\mu(\uparrow)$: 0.75 m σ : 0.006, 0.009		B = 1	$\mu(\text{---})$: 0.006 m σ : 0, 0	B = 1	$\mu(\uparrow)$: 68 steps σ : 8, 8	
	B = 2	$\mu(\uparrow)$: 0.80 m σ : 0.027, 0.042		B = 2	$\mu(\text{---})$: 0.028 m σ : 0.0011, 0.0012	B = 2	$\mu(\text{---})$: 81 steps σ : 18, 14	
D (obstacle speed)	C = 1	$\mu(\text{---})$: -6.82×10^{-4} m σ : 0.02, 0.02		B = 1	$\mu(\text{---})$: 0.006 m σ : 0, 0	B = 1	C = 1	$\mu(\text{---})$: 19 steps σ : 8, 8
	C = 2	B = 1	$\mu(\text{---})$: 0.75 m σ : 0.009, 0.009			B = 2	$\mu(\text{---})$: 0.028 m σ : 0.001, 0.001	C = 2
		B = 2	$\mu(\text{---})$: 0.80 m σ : 0.042, 0.042	B = 2	$\mu(\text{---})$: 0.028 m σ : 0.001, 0.001	B = 2	$\mu(\text{---})$: 81 steps σ : 15, 16	

Table 3.4: Summary of the conclusions obtained for the effect that each factor has on the performances of the filter. Again, “1” and “2” stand for the low and high levels of the factors, respectively. See table 3.2 for the numerical values of the factors. Here, μ and σ represent the mean and standard deviation of the factor. The symbol “—” denotes no effect on the mean, which is indicated along with its value, and “ \uparrow ” represents an increase in that value, which is accompanied in this case by the difference between means at each extreme (the high one minus the low one). In each cell, the extreme values of the standard deviation are also reported.

Regarding abnormal observations, missing sensory readings (factor B), usually provoked by the presence of obstacles with transparent or absorbent surfaces or by conditions of extreme lighting, have a negative impact on all the performances in most cases; more concretely, as the occurrences of this anomaly increase ($B = 2$), the performances get worse, but a relevant and not obvious conclusion of this study is that accuracy is affected by the presence of missing readings only if

3.4 Results of the study and discussion

these kind of data occurs along with biased data, although the impact is not very strong.

In the case of the expected uncertainty of the filter, an increase in the percentage of missing readings always leads to a higher uncertainty. As predicted by the linear regression method, only this factor has relation with the uncertainty; the main reason is the fact that the filter uncertainty can be reduced as more observations are available at the time of inference, under certain conditions.

Another non-obvious conclusion on the influence of missing data is that the convergence of the filter is only affected by an increase of missing observations in the case that these readings are *not* combined with any biased sensor readings, otherwise the effect of missing data being negligible. In other words, biased observations produce an influence that “hides” the one of missing data in the convergence of the filter; the very effect of biased readings is sufficient to seriously deteriorate the convergence (see table 3.4).

Biased observations (factor C), which are often provoked by excessive reflections of the waves emitted by sensors on the scene, also have an important and negative effect on the performances with the exception of the filtering uncertainty, which does not depend on the concrete values of the readings but on the number of them, as discussed before. For the case of the expected accuracy of the filter, an increase in the percentage of biased readings always leads to a much worse accuracy, regardless of the remaining conditions. A result that is not so straightforward is that filter convergence is only affected by biased data when these are not combined with missing ones: the very effect of missing observations is strong enough to noticeably worsen the convergence rate, again “hiding” the effects of biased data. In conclusion, once that one of these kinds of anomalous sensory data are present, the effect of the other is negligible in convergence, although biased data has worse effect in the magnitude of convergence.

ANOVA does not provide conclusions about the effects on the standard deviation; in the end, they are considered less relevant than the ones produced on the means of the factors; however, they have been analyzed as well. In this case, variations in the value of factor B (amount of missing readings) always lead to relevant changes on the standard deviation, even when it is proved that there is no effect on the mean. Regarding factor C (amount of biased observations), the differences are not that important in most cases, with the exception of the expected accuracy performance of the filter. Lastly, the remaining factors do not have any noticeable impact on the standard deviation in any case.

3.4.3 Impact on the operation of mobile robots

The abnormal situations studied are plausible in real robotic contexts and they may have a relevant impact on the operation of mobile robots. The study has proved that only factors B and C, which correspond to the amount of abnormal sensory data, have some kind of effect on the steady-state performances of the filter. Such anomalous sensory readings are not infrequent in real scenarios where mobile robots typically operate as discussed in section 3.3.1. For instance, navigation in large corridors may well lead to a high amount of missing sensory data, due to the fact that the maximum detection range of the on-board sensors is systematically exceeded in the longest direction. Unfortunately, this is not the only situation that could lead to the same issue, and there are, in fact, many of them (e.g., navigation under conditions of extreme infrared radiation, navigation nearby highly reflective surfaces, etc.). Biased readings are also common in these kinds of sensors, and are usually due to particular features of the scene (e.g., presence of

3.4 Results of the study and discussion

geometrically challenging surfaces, such as corners, or highly reflective ones as well, etc.). There are also some situations where both kinds of abnormal sensory data can be combined (although not simultaneously). For example, in a scene with a high presence of thin obstacles, such as chair legs, range sensors may produce both biased and missing readings alternately, sometimes due to a high number of reflections and other times due to sudden detections of free space, respectively.

An inadequate value in any of the measures of performance has a negative impact on the operation of a mobile robot. More concretely, essential tasks such as navigation, localization and mapping may result seriously compromised. For instance, an increase in the amount of biased sensory data worsens the expected accuracy of the filter, and, in this situation, the pose of the operating mobile robot could not be estimated properly, biasing it as well. Similarly, a less accurate perception of the scene may affect the mapping of such environment, and this affects, in turn, subsequent navigation, compromising the robotic operation. Abnormal observations such as missing readings have a negative impact on the expected uncertainty: the higher the number of these observations, the higher the filtering uncertainty. In extreme conditions, this may result in useless distance estimations in the scope of an obstacle tracking scenario, or in localization or mapping problems, since an estimation with high uncertainty cannot be considered to solve any of these problems. Finally, the presence of a high amount of either missing or biased sensory readings negatively affects the convergence of the filter. A slow convergence rate could, for instance, limit the maximum navigation speed, since it would not be safe for a robot to operate within the scene relying on highly uncertain or inaccurate distance estimations. In the case that the speed could not be limited, this issue would lead to a poor localization and mapping, due to the low quality of the estimations.

3.4.4 Validation in a real experiment

This section aims to validate the conclusions obtained in section 3.4.2 with a experiment in a real environment. For that, a mobile platform has been employed, the CRUMB robot [108], which is the one that has been used as well throughout the thesis (see chapter 5 for a more complete description). This platform is based on a version of the *Turtlebot-2* that uses a two-wheeled *Kobuki* platform [109]. This mobile robot is endowed, among others, with two range sensors relying on infrared radiation, namely, a *Hokuyo URG-04-LX* 2-D laser [101] and a *Kinect V1* RGB-D camera [102], [103], whose main features were already included in table 3.1. The CRUMB robot is also equipped with an on-board netbook PC with an Intel Celeron N2840 at 2.16 GHz and 2 GB DDR3 that runs Ubuntu 14.04 with ROS [110]. A picture of this robot can be seen in figure 3.10(a).

The experiment takes place in the indoor scenario shown in figure 3.10(b). This setup aims to reproduce the conditions of the general obstacle tracking problem studied in this work (recall Figure 3.1). In this case, the robot moves at a constant speed from point A to B, while facing a static obstacle that is to be detected by the range sensors on board. Only those measurements gathered in the very direction of movement will be considered, which corresponds to the gray chair leg that is closest to the robot.

The CRUMB robot covers in this experiment a distance of 1 m. This has been measured manually in the real scene, as well as the ground-truth distance to the obstacle, which is 2.05 m when the robot is placed at point A and 1.05 m when it is at point B. Also, the measured speed is 0.116 m/s. The obtained sensory measurements from both sensors along with the ground-truth distances are shown in figure 3.11.

3.4 Results of the study and discussion

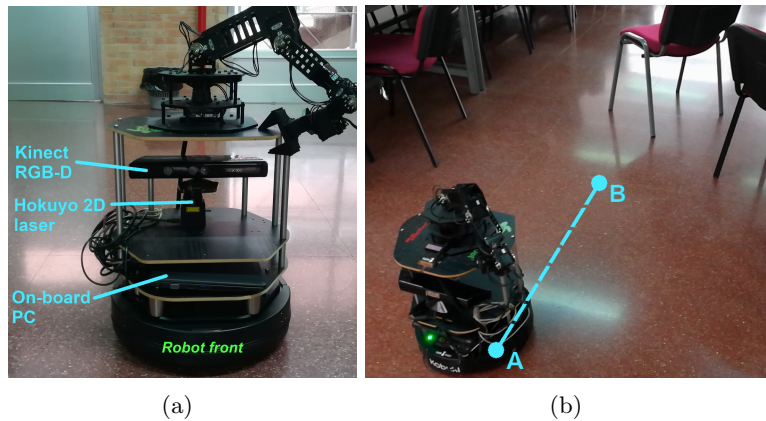


Figure 3.10: Experimental setup. (a) Frontal view of the CRUMB robot with its devices. (b) Indoor scenario used for the experiment. Here, the robot moves at a constant speed from point A to B towards a chair with gray legs placed in front of it, which produces a number of anomalies.

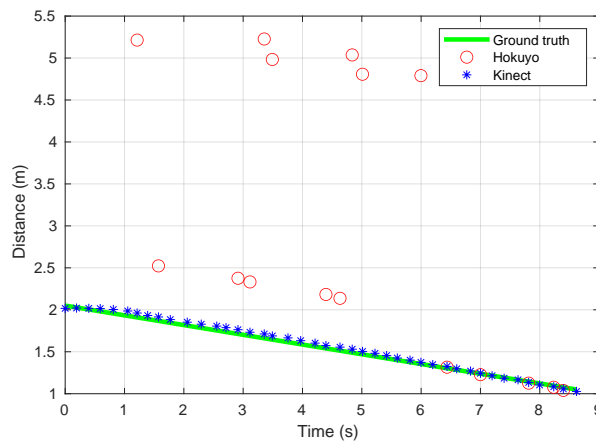


Figure 3.11: Range measurements obtained by the *Hokuyo* and the *Kinect* sensors during the experiment in figure 3.10, along with the ground-truth distances.

The gathered data show that the *Kinect* sensor has worked reasonably well during the experiment and that no anomalies have affected it. In contrast, the *Hokuyo* laser rangefinder has suffered from abnormal conditions up to a point that its observations are rarely correct: the obtained measurements are mostly biased and or missing, corresponding to the extreme position of these factors in the statistical study of section 3.4.2. The obstacle is, in the end, a reflective surface that may have provoked the reflection of the central laser beam over another nearby chair legs (see figure 3.10(b)) leading to a larger distance than the actual one. Also, this beam may have been reflected to an empty area, leading, as a result, to a missing observation. The reason why the *Kinect* sensor is not affected by the same situation is probably due to the fact that its measurement principle, although based on infrared radiation, is different.

The three sources of data present in figure 3.11 are needed for comparing this experiment to the conclusions of the statistical study—the measures of the filtering performance could not be obtained without knowing the ground-truth distances; also, it would not be possible to extract any conclusion on the effects of abnormal conditions on such performance without a fault-free situation.

3.4 Results of the study and discussion

The Bayesian filter has been implemented in the form of a DBN, as explained in section 3.3.2, and the performance measures have been calculated as detailed in section 3.3.3. The parameters of both the observation and transition models of the filter have been modified (equations (3.1) and (3.3), respectively) so that they adapt to the concrete conditions of the real experiment. In particular, the standard deviation of sensory measurements has been set to $\sigma = 0.08$ m, since it represents the average accuracy of both the *Hokuyo* and *Kinect* sensors (see table 3.1). Also, the speed in the transition model has been set to $v = -0.116$ m/s, where the negative sense is due to the fact that it is the robot which moves in this case, and not the obstacle. Also, the value of Δt is not constant and has to be modified in each iteration of the filter. In this case, it has a mean of 0.21 s and a standard deviation of 0.04 s.

Figures 3.12 and 3.13 show the accuracy and uncertainty performance measures over time as well as the convergence achieved in the case of use of the *Kinect* and *Hokuyo* sensors, respectively. Furthermore, the steady-state measures of performance are collected in table 3.5.

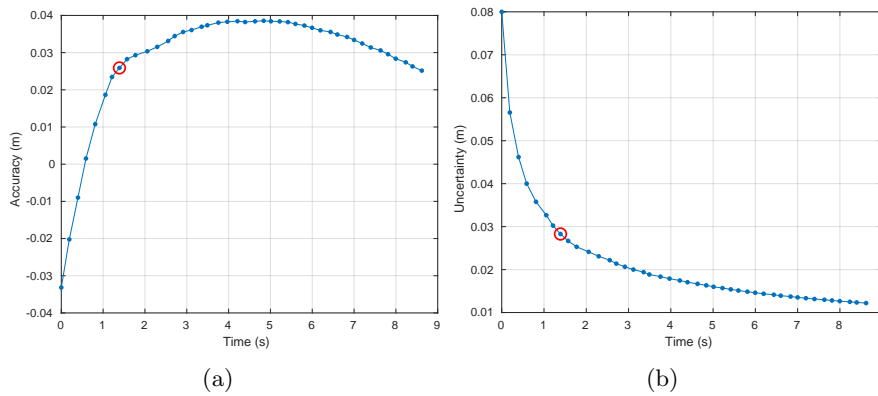


Figure 3.12: Evolution over time of the measures of filtering performance for the case of use of the *Kinect* sensor. The red circle indicates the instant of convergence (after 8 filtering steps). (a) Accuracy. (b) Uncertainty.

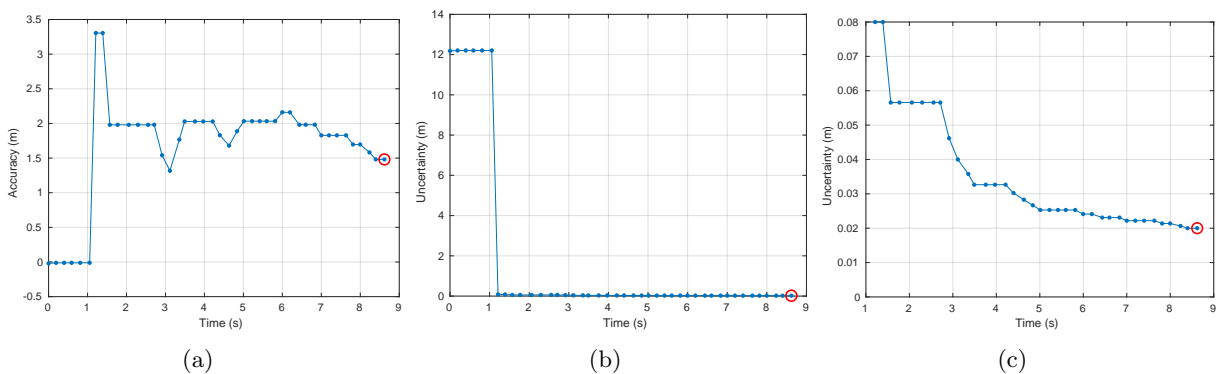


Figure 3.13: Evolution over time of the measures of filtering performance for the *Hokuyo* sensor. The red circle indicates the instant of convergence (43 filtering steps). (a) Accuracy. (b) Uncertainty. (c) Zoomed view (vertically) of the uncertainty.

3.4 Results of the study and discussion

Measure	Kinect	Hokuyo
Expected accuracy	0.0268 m	1.5599 m
Expected uncertainty	0.0124 m	0.0205 m
Steps for convergence	8	43

Table 3.5: Steady-state measures of filtering performance for the sensors used in the real experiment. The expected accuracy and uncertainty were calculated taking into account the last 4 steps of the filter.

With these results, it can be concluded that all the measures of performance are worse for the case of the *Hokuyo* sensor, which was affected by both biased and missing data anomalies, even when considering their evolution over time. These results also allows for the validation of some of the most important conclusions of the study, which were reported in table 3.4. Firstly, the combined presence of anomalies in the experiment with the *Hokuyo* rangefinder leads to a much worse expected accuracy compared to the fault-free situation of the *Kinect* sensor: biased readings are sufficient to deteriorate this performance, regardless of the remaining conditions. Second, the combination of anomalies in the real experiment provokes an increase on the expected uncertainty, which is also compatible with the obtained conclusions, since the sole presence of missing readings is expected to worsen this performance. Finally, the abnormal conditions in the real setup also lead to a much slower convergence, which is again compatible with the statements of the study, since only the presence of one of the anomalies is enough to produce this effect.

Bayesian networks for sensory diagnosis and recovery in mobile robots

This chapter presents the main contributions of the thesis related to the use of Bayesian networks for the representation of robotic sensory systems. In particular, mobile robots often operate in challenging environments from the perspective of their sensory apparatus, which may compromise their performance. The existing paradigms for the detection of abnormal situations are not rigorous either in dealing with uncertainty or in integrating diverse sources of knowledge. In order to overcome these limitations, a novel paradigm based on Bayesian networks is introduced. The proposed model allows for the representation of a variety of robotic sensory systems, enabling mobile robots to identify sensory anomalies and recover from them. The main drawback is the potentially high computational cost of inference with Bayesian networks, which is addressed with a novel, approximate algorithm that leverages the structure of the proposed model. The computational complexity of this algorithm is analyzed, and finally, a method for reducing its cost even more using neural networks is also introduced.

4.1 Introduction

Many of the current applications of mobile robots (like the ones reported in chapter 1) require platforms that operate with a certain degree of autonomy, which is difficult to achieve in general due to the uncertain nature of the physical world. This condition particularly affects the robotic sensory apparatus, as previously discussed in chapter 3. The conclusions obtained there prove that, for a mobile robot to operate within complex environments, it is not sufficient to only extract useful information from noisy sensory data. Those environments are usually governed by a considerable amount of different variables with many interactions among them, thus the robot should also be able to intelligently reason about, identify and overcome abnormal situations that prevent

4.2 Related research

it to achieve an adequate performance, specially from the perspective of its sensory system. At the same time, it is also crucial for a proper robotic operation its efficiency in computation, since many tasks may have real time requirements, depending on the concrete application. For these reasons, a rigorous framework allowing to deal with uncertainty and also to integrate different sources of information intelligently and efficiently is a relevant goal in current robotics research.

As discussed in chapter 1, identifying abnormal situations intelligently has been traditionally treated in robotics by using three different methodologies, namely, *analytical* (or *model-based*), *data-driven*, and *knowledge-based* [32], [34]. In addition to these paradigms, other tools and methodologies such as ad-hoc, heuristics [111] and case-based reasoning [112] have been used for the same purpose. Although all the mentioned approaches have been proven to be useful, they do not meet completely the requirements imposed by the stated problem, since they are not rigorous either in dealing with uncertainty or in integrating different sources of information, or in both.

Recently, Bayesian networks [39] have been considered in the field of fault diagnosis as a powerful representational tool [34]. Based on a rigorous mathematical framework, they are able to manage uncertainty and also to compactly represent the behaviour of complex systems, enabling to reason about them. Their advantages rely on their ability to reason in many directions (not only from the data to conclusions), the possibility of doing inference with different goals without rebuilding the model, and of being hybridized with other methodologies, such as the mentioned fuzzy logic [113] and neural networks [114] paradigms. Their main drawback is the high computational cost of the existing inference algorithms, which is NP-hard in general [48], [58].

This chapter presents the core contribution of this thesis, i.e., a novel sensor anomaly detection framework, based on Bayesian networks, that enables the representation of complex relationships among the sensors in a mobile robot in order to provide more robust, intelligent and reliable operation in real scenarios. The chapter also develops the introduction of a new inference algorithm that uses the proposed model to significantly reduce the computational cost of inference. The algorithm presented is inspired in existing methodologies, commented in section 4.2, but it differs in the way the information is treated to achieve efficiency. Then, one of the last works developed in the thesis is also discussed. Such proposal, based on the algorithm introduced in the chapter, aims to reduce the cost of inference even more by compiling specific probability queries as feedforward neural networks, which serves to illustrate the benefits of hybridizing Bayesian networks with other paradigms, as mentioned above.

The rest of the chapter is organized as follows. Section 4.2 reviews some research related to the applications of Bayesian networks in robotics. Section 4.3 defines the novel model that allows the efficient representation of, and inference in, sensory systems for mobile robots. Finally, section 4.4 sets the theoretical background for the inference algorithm that works on that model, discusses its computational complexity and presents the last proposal developed in the thesis regarding the increase of inference efficiency, grounded on the use of neural networks.

4.2 Related research

Developed in the 1980s, Bayesian networks have been applied to a broad variety of fields and heterogeneous domains such as medicine [115], agriculture [116], economics [117] and engineering [118]. In robotics, Bayesian network inference has been applied to a wide range of tasks, such as localization and mapping [119], navigation [120], manipulation [121], sensory fusion [122], logic

4.2 Related research

programming [123], etc. Also, they have served as a representational tool for sensory systems (not only limited to the scope of robotics); many applications of fault diagnosis using Bayesian networks can be found in industrial processes, energy systems (electrical, thermal, etc.), manufacturing, and network systems [34], [124]. The main aim of these models is to provide more robust sensory state estimation by the intelligent combination of information from the sensors and components available in the system [125].

In the case of mobile robotics, Bayesian networks have been used as a sensor fusion tool that enables to predict low-level data [41] and have served to implement fault tolerance models that detect sensory anomalies taking into account the robot environment [40]. Particularly, the preliminary works related to this thesis [53], [50] and [51], introduced a model based on Bayesian networks to represent the whole sensory system of a mobile robot, enabling the detection of anomalies and the recovery of data thanks to the integration of heterogeneous information coming from very different sources (sensory readings, human commonsense, environmental data, etc.). That proposal was first tested in a real robot restricted to simple devices (bumpers, gyroscopes, cliff detectors, etc.) and later improved to allow the integration of more complex sensors such as rangefinders. This chapter presents the results finally achieved in the thesis regarding sensory diagnosis and recovery with Bayesian networks, mainly supported by the works [33] and [52]. This last proposal consists in a more complete Bayesian model and inference method that serve to improve the efficiency of inference while allowing the representation of both simple and complex devices, sensory temporal dynamics and different levels of cognitive abstraction related to sensory indicators.

As explained in chapter 2, once a Bayesian network model is designed, several inference tasks must be performed; in this case, they would serve to get information for sensory diagnosis and recovery. Inference in the network can be done by simply applying basic probability theory (the Bayes' theorem, the chain rule, etc.), however, this is not feasible in that basic form because the computational cost may grow exponentially with the number of variables in the worst case. Many inference algorithms have been developed in the last decades that address that by the Bayesian network community. They can be classified either in exact or approximate methods (see subsection 2.2.2 in chapter 2).

The mentioned algorithms are intended to perform inference in arbitrary networks. However, each component of a mobile robot sensory system has a particular relationship with the rest of the robot and the environment, which restricts the possible forms of the network. In short, not every sensor is affected by all the others and not every environmental condition has a decisive impact on every sensor. Thus, the attention should be focused on inference algorithms that take into account these constraints. Some proposals related to this are the *edge deletion belief propagation* algorithm [126], [58] and the very notion of *modular Bayesian networks* [127]. The former is an approximate inference method somewhat similar to *loopy belief propagation*: it performs exact inference in an approximate network that is obtained by removing network edges, leading to a more efficient execution (the number of deleted edges is a tradeoff between accuracy and efficiency). On the other hand, the paradigm of modular Bayesian networks aims to reduce the computational cost of inference by splitting the complete (monolithic) model into smaller subnetworks called *modules* that are coordinated to perform inference tasks defined in the global model. This is not an inference algorithm itself, and it has been implemented differently by several authors. For instance, the *virtual linking technique* [128] uses virtual evidence on some nodes similarly to the edge deletion algorithm, and has been applied, for example, to landmark extraction from cell phone data [128], to automated human authentication systems [129], to the diagnosis of networked robots [130], etc. Nevertheless, none of these approaches leverage the particular topologies of sensor inference in a

mobile robot, which is the focus of this thesis.

4.3 A representation of robotic sensory systems using Bayesian networks

This section covers the definition of the basic element that is needed to represent the sensory system of a mobile robot, aiming at efficient inference, using for that Bayesian networks with a restricted form according to the system constraints. This basic element and the most common queries for such a sensory system are presented in subsection 4.3.1, while the complete network architecture is described in subsection 4.3.2. Recall that these models are the result of a proposal that has been continuously evolving throughout different works related to the thesis up to their last version, presented in [33], [52].

4.3.1 The Bayesian sensor. Inference in sensors

The basic element of the proposed framework is a so-called *Bayesian sensor*, which represents a single aspect of the robotic sensory system under study. Such aspect may be one of the quantities measured by the on-board sensors (e.g., distance, speed, temperature), or more elaborated sensory information built upon low-level data (e.g., the presence or category of a detected obstacle). In general, a Bayesian sensor is conceived to capture any sensory indicator about the state of the robot or the one concerning its environment, regardless of the complexity or the level of abstraction.

A Bayesian sensor \mathcal{B} can be formally defined either as a static or a dynamic Bayesian network over a set of variables \mathbf{B}_z , depending on whether it is necessary to encode the temporal dynamics related to some sensory subsystem. In the static case, $\mathcal{B} = (B_g, B_\theta)$, with B_g the graph and B_θ the set of CPDs, while, in the dynamic one, $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_\rightarrow)$, with \mathcal{B}_0 the initial network and \mathcal{B}_\rightarrow the transition network, each one with its corresponding structure and parameterization. Recall also that a Bayesian sensor can be defined over both discrete and continuous random variables, i.e., it can be represented by a *hybrid* Bayesian network (see chapter 2). The general structure of this basic element is depicted in its two possible forms in figure 4.1; as shown, the network structure is similar for both, and it is organized as follows:

- **Ideal sensor node (I):** it encodes the true state of the sensory aspect being represented in the absence of abnormal conditions. This variable will usually correspond to the query set in an inference task ($\mathbf{Q} = I$) in order to recover faulty or unreliable sensory information. The ideal node plays a key role in the definition of the whole sensory system, since the relationships among the different Bayesian sensors of the robot are defined by connecting this node as a parent of nodes belonging to others. More specifically, it should be connected to anomalies and/or virtual subnetworks of other Bayesian sensors, as explained later on. The ideal node also serves to encode the dynamics of the sensory information it represents, as shown in figure 4.1.
- **Real sensor nodes (\mathbf{R}):** this is a set of variables representing the values actually measured -observed- by the related sensory sources available on-board the mobile robot (e.g., physical sensors, algorithms producing sensory knowledge, etc.). Here, $|\mathbf{R}| = n_s$, where n_s is the number of such sources. This set has the role of evidence in an inference task ($\mathbf{R} \subseteq \mathbf{E}$), as long as the measured, available information is absolutely certain, i.e., if it represents *hard* evidence. Notice that the CPDs associated with these nodes (which are of the form $P(R_i|I)$,

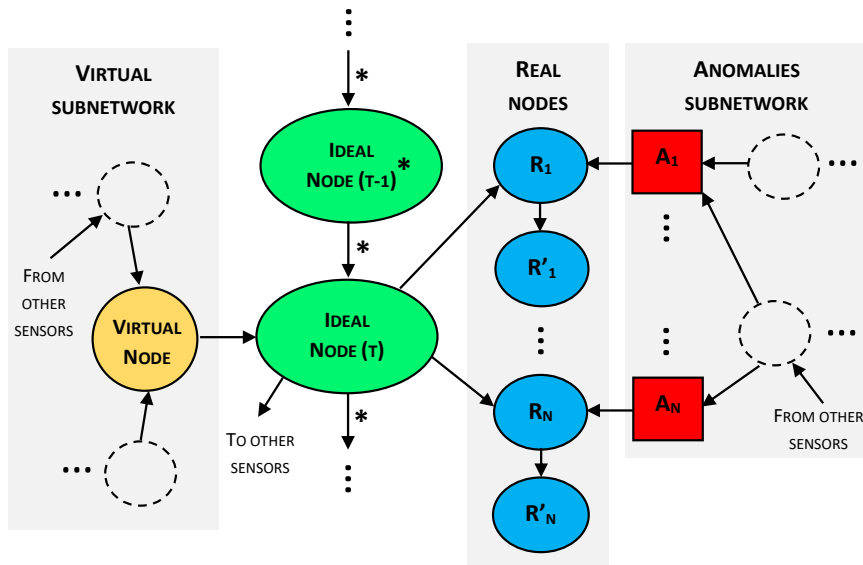


Figure 4.1: Schematic representation of both the dynamic and static forms of the Bayesian sensor defined in this chapter. Here, the complete diagram refers to the transition network $\mathcal{B}_{\rightarrow}$ of the dynamic version. Both the initial network \mathcal{B}_0 and the static sensor can be obtained by removing the node and the arcs marked with an asterisk. Squared nodes represent discrete random variables while the round ones may be either discrete or continuous.

with $R_i \in \mathbf{R}$) could represent observation models of the sensory information being considered (e.g., in the case of physical sensors, the CPDs could encode *noise* models). However, if the available information is uncertain, the set \mathbf{R} is replaced by the set of child nodes \mathbf{R}' (see figure 4.1) as the evidence in an inference task in order to allow the incorporation of *virtual* evidence. For that, the observed values are imposed as hard evidence on these nodes, and the uncertainty associated with them is encoded in their corresponding CPDs, which are of the form $P(R'_i|R_i)$, with $R'_i \in \mathbf{R}'$. These distributions are not to be confused with the models used for variables \mathbf{R} , since the former do not represent the behavior of a sensor, but simply the *degree of belief* in observations gathered from a noisy or possibly unreliable source. Thus, the set \mathbf{R}' is included in the Bayesian sensor only if soft evidence is needed. Recall that the variables of these real nodes are defined over the same support as the ones of the ideal nodes, since they represent the same kind of information.

- **Anomalies subnetwork (\mathcal{A}):** this important subnetwork is used to detect abnormal situations affecting sensors. Defined over a set of variables $\mathbf{A}_{\mathbf{z}}$, it is a pair, $\mathcal{A} = (A_g, A_\theta)$, where A_g represents the subnetwork structure and A_θ its parameterization. A complete subnetwork is required because modeling the anomalies of a certain sensor for intelligent inference may be arbitrarily complex (i.e, modeling a sensor that is affected by many different anomalies with interactions among them). It is possible to integrate information from other sensors (by using the mentioned connection from their ideal nodes), human commonsense, environmental data, etc. Regardless of the complexity of the chosen model, the subset of discrete variables $\mathbf{A} \subseteq \mathbf{A}_{\mathbf{z}}$ highlighted in the figure, called the **anomalies nodes** (never empty, in fact $|\mathbf{A}| = n_s$), represent all the possible undesired situations for the Bayesian sensor, indicating whether there is one of each kind or not (they are always binary). These are connected as parents of the real nodes, since the observable behavior of a sensor is directly influenced by the occurrence of these abnormal situations. In the case that anomalies reasoning is not very complex or there is not enough available information, \mathbf{A} could be the only nodes in the anomalies subnetwork ($\mathbf{A}_{\mathbf{z}} = \mathbf{A}$). In an inference task, one of the variables in the anomalies

4.3 A representation of robotic sensory systems using Bayesian networks

nodes can be queried ($\mathbf{Q} = A_i \subseteq \mathbf{A}$) in order to identify which anomalies are affecting the sensor.

- **Virtual subnetwork** (\mathcal{V}): this is a Bayesian network that is aimed at deducing supplementary information related to the sensory aspect being represented by the ideal node. Such knowledge would be essential to allow the recovery of the correct state of the sensor in case of abnormal situations. This subnetwork is defined similarly to the previous one: $\mathcal{V} = (V_g, V_\theta)$, where V_g is the structure and V_θ the parameterization. \mathbf{V}_z is the complete set of its variables. It receives connections (influences) from ideal nodes of other Bayesian sensors as explained above, and also information from other sources (environmental, etc.) so as to reason about the mentioned correct state of the current Bayesian sensor. There is a special node in this subnetwork called the **virtual node**, $V \subset \mathbf{V}_z$, which summarizes the reasoning described here. It is connected as a parent of the ideal sensor node to enable the rebuilding of this information under anomalies. The virtual node will not be part of the query set in any inference task under normal circumstances, thus it should be used only for model checking and debugging.

In summary, the Bayesian sensor is defined over variables $\mathbf{B}_z = I \cup \mathbf{R} \cup \mathbf{R}' \cup \mathbf{A}_z \cup \mathbf{V}_z$. Note that, in the dynamic version of the sensor, the time slice t would also be taken into account, i.e., the set would be denoted $\mathbf{B}_z^{(t)}$. In this case, only the ideal variable I evolves over time (see figure 4.1). Concerning the structure, $B_g = A_g \cup V_g \cup I_g$, where I_g is a graph over variables $\mathbf{I} = I \cup \mathbf{R} \cup \mathbf{R}' \cup \mathbf{A} \cup \mathbf{V}$ that contains the edges connecting these variables (those that are not included neither in anomalies nor virtuals, see figure 4.1). This graph describes the structure of the static version of the sensor and the one corresponding to the initial network in the dynamic version. In order to obtain the structure of the transition network, extra nodes and arcs related to the ideal sensor node at different time slices should be added to B_g (figure 4.1). Apart from this, note also that certain kind of robotic sensors such as rangefinders, cameras, etc., may need to handle vectorial or matricial data. The proposed Bayesian sensor model can be extended for these cases by considering each single measurement as an individual sensor and copying the same model as many times as necessary, as well as the required interdependencies.

To complete its definition, the Bayesian sensor must also be parameterized by filling the corresponding CPDs. The particular value of each parameter of these distributions can be obtained taking into account expert knowledge (human commonsense), statistical information about the behaviour of a certain sensor (e.g., failure rate, noise model), or any other environmental data (e.g., date and time, the weather, a map of the scene) that is considered useful, even with some automated learning procedure.

A complete model of the sensory system of a mobile robot can be obtained by joining/linking together all the necessary Bayesian sensors, leading to a unique, *monolithic* Bayesian network representing such system. In this context, there are a number of queries that can be answered by the model. In particular, it is critical to know about the anomalies affecting a certain sensor in the system, if any, and also about the ideal value this sensor would have in a fault-free situation. These queries represent the ones that will be employed throughout the thesis, and they have the form expressed below. Recall that the queries are stated for exact inference models (i.e., monolithic Bayesian networks); inference in the novel approximate architecture proposed will be discussed in section 4.4).

In general, a query for a Bayesian sensory system can be defined as a posterior distribution $P(\mathbf{Q}|\mathbf{E})$, where \mathbf{Q} is the set of query variables and \mathbf{E} the set of evidence variables:

$$\mathbf{E} = \bigcup_{i=1}^n \text{real}(\mathcal{B}_i). \quad (4.1)$$

Here, *real* gets the set of real sensor variables, \mathbf{R} or \mathbf{R}' , for Bayesian sensor \mathcal{B}_i in a sensory system with n sensors¹. The mentioned set will always be instantiated in a query by using the data read by the robot. This is denoted as $\mathbf{E} = \mathbf{e}$, where \mathbf{e} is a set of values gathered for variables in \mathbf{E} . The queries for anomalies in a sensor are thus:

$$P(A_i | \mathbf{E} = \mathbf{e}), \quad (4.2)$$

where $A_i \subset \mathbf{A}$ is one of the anomalies nodes of a certain Bayesian sensor. The query for the ideal value of a sensor is:

$$P(I | \mathbf{E} = \mathbf{e}), \quad (4.3)$$

where I represents the ideal node of a certain Bayesian sensor. For the dynamic case, the queries defined in equations (4.2) and (4.3) would be re-formulated in terms of the present time interval (or slice) t ; thus, the query variables would be $A_i^{(t)}$ and $I^{(t)}$ respectively, and the evidence set defined in equation (4.1) would also include all the observations from the initial state to the present. In summary, any query would be of the form $P(\mathbf{Q}^{(t)} | \mathbf{E}^{(0:t)})$ for the dynamic case, which is known as *filtering* query in the context of DBNs (see section 2.2.2 in chapter 2).

In this thesis, query sets with more than one variable will not be considered, thus $|\mathbf{Q}| = 1$. This is because it is not critical to compile the situation of two or more different sensors in the same distribution as long as they can be deduced sequentially. Actually, it is possible to get multiple queries of individual sensors efficiently in the exact *jointree* algorithm, as long as the set of evidences remain with the same values [55], [58].

4.3.2 A Bayesian architecture for robotic sensory systems

The complete model of the sensory system of a robot based on the definitions of subsection 4.3.1 consists of a monolithic Bayesian network with a complex interconnection structure depending on the existing relationships among random variables. Unfortunately, using that model for inference tasks poses, at least, two drawbacks.

One of these issues is related to the necessity of modeling sensory systems with heterogeneous components, which may produce information related to different *ontologies* or *levels of cognitive abstraction* at the same time (e.g., a sensory system simultaneously estimating the pose and the identity of a detected pedestrian). This has important implications on the network structure of the system, since having sensors at different levels of abstraction in the same network might introduce, in general, undesired dependencies among them, potentially leading to inaccurate or unreliable inference results. Thus, the proposed solution is to treat such sensors in separate networks. This has already been contemplated in the definition of the Bayesian sensor, since it counts with a mechanism that allows decoupling from other sensors based on *virtual* evidence. In the case that a sensor needs to be connected to another one belonging to a higher level of cognitive abstraction, it would suffice to replace such connection by asserting the conclusions produced by the former

¹Recall that the choice between \mathbf{R} and \mathbf{R}' depends on whether soft evidence is to be emulated or not, as explained before.

4.3 A representation of robotic sensory systems using Bayesian networks

sensor as virtual evidence in the latter (this procedure will be detailed later on).

This solution would lead to a representation of robotic sensory systems based on as many monolithic subnetworks as the number of necessary levels of cognitive abstraction. However, there is still another issue that has to be addressed. Dividing the representation of the system into different parts would not bound the size of each one of the resulting monolithic models, leading to potentially large networks. Unfortunately, answering queries with an exact inference algorithm in that networks could be very inefficient in the general case. This issue was first addressed in the work [33], developed during the thesis. That proposal consists in an approximate approach, based on a so-called *layered* network, that strategically splits the monolithic models into separate parts in order to improve, even exponentially in the best case, the efficiency of inference.

This chapter presents an evolved version of the proposal in [33] that also enables the treatment of different levels of cognitive abstraction and the incorporation of sensory dynamics, as explained later on. Note that this proposal is not a construction procedure of the architecture from the monolithic one, but a method that builds the former by considering the existing interactions in the complete model and that uses the result to perform inference.

The work [33] also pointed out a further issue related to the proposed representation that has to be solved. Modeling a sensory system accurately implies to compile as much interactions among sensors as possible, and in that process it is common to find mutual relationships in pairs of them, when their behaviors influence each other. These cyclical dependencies are incompatible with Bayesian networks and prevent inference. As an example, consider simple cliff and wheel drop detectors in a mobile robot. Under most conditions, both should be enabled consistently to describe a physically feasible situation (e.g. the robot is about to tip over). Thus, if the cliff is off while the wheel drop is on, that could be considered an anomaly, and the same holds the other way round. Note that only symmetric cyclical dependencies between pairs of sensors will be considered here, and not those cyclical relationships that involve two sensors in a transitive way through a third one, since this might not have physical sense in general.

In the following, the problem posed by cyclic Bayesian networks formed by the defined Bayesian sensors will be addressed. After that, the final version of the proposed sensory architecture will be presented.

Enabling inference in cyclic Bayesian networks

The problem of encoding cyclic dependencies in Bayesian networks has been addressed by some authors. One of the existing approaches in the literature is based on the use of probabilistic models such as dynamic Bayesian networks [131] or stationary Markov processes [132] to approximate the joint probability distribution induced by the cyclic Bayesian network, and then working with that distribution. Here, an alternative, approximate solution based on breaking cyclic dependencies is proposed, which avoids the need of such potentially costly distribution estimation. Notice, however, that both the distribution approximation and the proposed breaking cycles procedure depend on the *existence* of a unique joint distribution for the network. As it will be clarified later on, this may not happen unless the cyclic network verifies some conditions. The reasonings developed in the following are based on some works in the literature that have studied directed cyclic models and the conditions under which their joint distributions exist and are unique [133], [134]. A unique joint distribution should be assured in the proposed Bayesian model by a suitable controlled procedure for the network definition, since a physical sensory system should have that property. For the

4.3 A representation of robotic sensory systems using Bayesian networks

sake of simplicity, the mentioned reasonings, presented below, will be restricted to discrete random variables only; they could be extended to include continuous variables by simply discretizing their supports (although this would be an approximation).

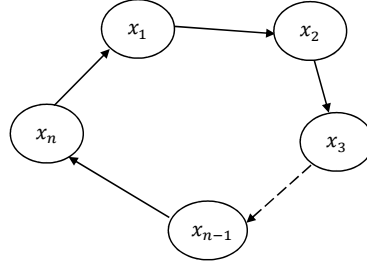


Figure 4.2: Isolated cycle with n random variables.

The simplest generic form of a cyclic dependency is an isolated cyclic network (see figure 4.2). Depending on the concrete parameterization of this model, the whole network may represent a unique joint probability distribution on its variables (nodes), a family of them, or none. The formulation established in [133] poses a linear system of equations to get the marginal distributions of each variable in the cycle. Here, that formulation is extended to allow for discrete random variables with general support cardinalities (non-binary). Then, each variable x_i in the cycle will have a marginal distribution given by equation (4.4):

$$P(x_i = w_{ij}) = \sum_{k=1}^{m_{i-1}} P(x_{i-1} = w_{(i-1)k})P(x_i = w_{ij}|x_{i-1} = w_{(i-1)k}). \quad (4.4)$$

There, w_{ij} represents the j -th possible value for variable x_i , whose support cardinality is $|x_i| = m_i$. This equation expresses a marginal distribution $P(x_i)$ as a result of the marginalization of the joint distribution of two adjacent variables in the cycle $P(x_i, x_{i-1})$. The conditional probabilities $P(x_i|x_{i-1})$ are known (since they are given as inputs in the form of CPDs), while the marginals are not, but a linear system of equations can be formed by applying equation (4.4) to each possible value of each variable in the cycle. Depending on the concrete coefficients, this system of equations may have a unique solution, a family of them or no solution at all; i.e., the marginal distributions may not be unique in general or may not even exist. In the case that they do exist uniquely, as it is assumed here, the joint distributions for adjacent variables are also unique.

The network definition can be controlled in order to satisfy the condition of existence of marginals by checking that the previous linear system is compatible and determined. Unfortunately, even if it is guaranteed that the marginals exist, the joint distribution that represents the whole network, $P(x_1, x_2, \dots, x_i, \dots, x_n)$, may still not be unique or may not even exist under the same conditions. Therefore, it is also mandatory to properly control the network construction for that.

The complete joint distribution for a regular Bayesian network can be obtained by applying the chain rule. In that procedure, all the information from the marginal distributions must be taken into account: the chain rule must be used in as many different ways as the number of variables in the network, n , as equation (4.5) shows:

$$\begin{aligned}
 P(x_1, x_2, x_3, \dots, x_n) &= P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \dots P(x_n|x_1, x_{n-1}), \\
 P(x_2, x_3, \dots, x_n, x_1) &= P(x_2)P(x_3|x_2)P(x_4|x_2, x_3) \dots P(x_1|x_2, x_n), \\
 &\vdots \\
 P(x_n, x_1, x_2, \dots, x_{n-1}) &= P(x_n)P(x_1|x_n)P(x_2|x_1, x_n) \dots P(x_{n-1}|x_n, x_{n-2}).
 \end{aligned} \tag{4.5}$$

There, conditional distributions of the form $P(x_i|x_j, x_k)$ are unknown, while the others, both the marginals and conditioned probabilities given as input, have been determined previously. Note that the chain rule has been simplified by considering the cyclic structure (figure 4.2) through *d-separation* [59]. This is in general a non-linear system of equations, whose analytical solution is not easy to obtain, thus numerical methods should be used (e.g., least-squares minimization [135], [136]). Solving that problem serves to check whether the joint distribution is unique. Recall, again, that the whole process described above could also be applied to the case of cycles formed by both discrete and continuous random variables, by first discretizing the support of the latter. This approach will be considered a good approximation for that case as long as the use of an adequate granularity for the discretization is not prohibitive, which depends on the nature of the particular sensory aspect involved in the cycle.

In summary, solving both the linear and non-linear systems of equations posed before are core steps in properly controlling the network construction in order to obtain a cyclic network with existing and unique marginals and joint. The conditional distributions given as inputs during this construction will be accepted only in the case that the joint distribution is effectively unique (note that, in the case of continuous variables, the original non-discrete CPDs would be accepted as well in this situation).

Now, even having defined a Bayesian network with a unique joint distribution, it is still cyclic. In order to deal with inference without completely losing the dependencies, an approximate solution is proposed here. It consists in removing one of the edges of the cycle and then replicating the resulting sequence a certain number of times. As an example, if there is a cyclic dependency between variables, say, x_1 and x_2 , the proposed equivalent acyclic graph could be, for instance, $x_1 \rightarrow x_2 \rightarrow x'_1 \rightarrow x'_2 \rightarrow x''_1$. In this network, the conditional distributions will be the same as the ones of the cyclic network (i.e. $P(x_i|x_{i-1})$), while the marginal distribution for the root node will be the one obtained in the linear system mentioned above. Consequently, the joint distribution will be an approximation of the original one. Breaking cycles in this way allows the definition of the architecture for Bayesian sensory networks presented in this chapter, enabling inference even with cycles. In such definition, the most recent improvements are considered, namely, the existing levels of cognitive abstraction and the temporal dynamics of sensors.

Consider a list of identifiers $\mathbf{B}_{id} = \{a, b, c, \dots\}$, where each element is associated with a Bayesian sensor (e.g., $\mathcal{B}_a, \mathcal{B}_b, \mathcal{B}_c$, etc.). A dependency involving two sensors a and b can be denoted as a pair (a, b) ; thus, a homogeneous binary relation can be defined as a set of pairs $\mathbf{D} \subseteq \{(a, b) | (a, b) \in \mathbf{B}_{id}^2 : a \neq b\}$. In the case that the dependencies are non-cyclic, \mathbf{D} is an asymmetric homogeneous relation \mathbf{D}_{nc} , where any two elements (a, b) and (c, d) verify $(a \neq d) \vee (b \neq c)$. When the dependencies are cyclic, \mathbf{D} is a symmetric homogeneous relation \mathbf{D}_c , where for any element (a, b) there exist another and unique element (c, d) such that $(a = d) \wedge (b = c)$. Once the set of influences among sensors \mathbf{D}_c and \mathbf{D}_{nc} are clearly defined, each cyclic dependency must be reformulated in terms of two non-cyclic ones that will be appended to \mathbf{D}_{nc} . This is how the equivalent acyclic graph for a cyclic dependency is implemented; a second instance of one of the two Bayesian sensors involved is created and then used to replicate the sequence induced by the

4.3 A representation of robotic sensory systems using Bayesian networks

Algorithm 6 *buildDependencyGraph*($\mathbf{B}_{\text{id}}, \mathbf{C}_{\text{id}}, \mathbf{T}_{\text{id}}, \mathbf{D}_{\text{nc}}, \mathbf{D}_{\text{c}}$)

input:

\mathbf{B}_{id} : list of identifiers of Bayesian sensors

\mathbf{C}_{id} : list of levels of cognitive abstraction associated with Bayesian sensors

\mathbf{T}_{id} : list of temporal dynamics indicators for Bayesian sensors

\mathbf{D}_{nc} : list of non-cyclic dependencies

\mathbf{D}_{c} : list of cyclic dependencies

output:

D_G : annotated dependency graph

\mathbf{B}_{id} : updated list of identifiers of Bayesian sensors

main:

```

1: for each pair of elements of the form  $\{(i, j), (j, i)\} \subset \mathbf{D}_{\text{c}}$  do
2:    $k \leftarrow$  choose one of the sensors, either  $i$  or  $j$ 
3:    $\mathbf{B}_{\text{id}} \leftarrow \mathbf{B}_{\text{id}} \cup k$ 
4:   if  $k$  is a new instance of  $i$  then
5:      $\mathbf{C}_{\text{id}} \leftarrow \mathbf{C}_{\text{id}} \cup c_k$ , with  $c_k = c_i$ 
6:      $\mathbf{T}_{\text{id}} \leftarrow \mathbf{T}_{\text{id}} \cup t_k$ , with  $t_k = t_i$ 
7:      $\mathbf{D}_{\text{nc}} \leftarrow \mathbf{D}_{\text{nc}} \cup \{(i, j), (j, k), (i, k)\}$ 
8:   else if  $k$  is a new instance of  $j$  then
9:      $\mathbf{C}_{\text{id}} \leftarrow \mathbf{C}_{\text{id}} \cup c_k$ , with  $c_k = c_j$ 
10:     $\mathbf{T}_{\text{id}} \leftarrow \mathbf{T}_{\text{id}} \cup t_k$ , with  $t_k = t_j$ 
11:     $\mathbf{D}_{\text{nc}} \leftarrow \mathbf{D}_{\text{nc}} \cup \{(j, i), (j, k), (i, k)\}$ 
12:   end if
13: end for
14:  $D_G = (\mathbf{B}_{\text{id}}, \mathbf{D}_{\text{nc}})$ 
15: Annotate each node  $i$  in  $D_G$  with a pair  $(c_i, t_i)$ , where  $c_i \in \mathbf{C}_{\text{id}}$  and  $t_i \in \mathbf{T}_{\text{id}}$ 
16: return  $D_G$  and  $\mathbf{B}_{\text{id}}$ 

```

cycle, as explained before.

In order to capture the interactions among sensors more compactly while considering levels of abstraction and temporal dynamics, a *dependency graph* D_G is defined from \mathbf{D}_{nc} , which is a directed acyclic graph whose vertices are the set of identifiers \mathbf{B}_{id} , and its edges the set of dependencies in the sensory system, thus, $D_G = (\mathbf{B}_{\text{id}}, \mathbf{D}_{\text{nc}})$. Each node in D_G is also annotated with a pair of elements from two different sets. One of them is $\mathbf{C}_{\text{id}} = \{c_a, c_b, c_c, \dots\}$, which is formed by natural numbers, each one indicating the level of cognitive abstraction to which each sensor represented in \mathbf{B}_{id} belongs, in the same order (e.g, c_a would be the level for sensor \mathcal{B}_a , etc.). These numbers range from the lowest possible level (i.e, zero) to the highest one (i.e., the total number of cognitive levels in the system c_l minus one). The other set is $\mathbf{T}_{\text{id}} = \{t_a, t_b, t_c, \dots\}$, which contains boolean values indicating whether temporal dynamics are encoded for the Bayesian sensors represented in \mathbf{B}_{id} , in the same order (e.g., $t_a = d$ indicates that Bayesian sensor \mathcal{B}_a is dynamic, $t_b = \bar{d}$ indicates that sensor \mathcal{B}_b is not, etc.). A procedure to build this graph given a list of Bayesian sensors, the interactions among them, the levels of abstraction these sensors belong to, and the indicators of their dynamic character is detailed in algorithm 6. For the sake of simplicity, it is assumed there that the assignment of cognitive levels in \mathbf{C}_{id} is defined such that every node in D_G verifies that each one of its children belongs to an equal or higher level of abstraction.

The graph D_G can be viewed in this context as an *abstraction*² of the complete monolithic model of a sensory system, since the former represents the existing interactions among Bayesian sensors with less amount of detail than the latter. The model of abstraction that is considered here is the one that maps sets of nodes and arcs in the original graph to super-nodes and super-arcs in the abstracted one, respectively: every single node in the dependency graph represents one Bayesian sensor, thus, it is a super-node of the set of nodes in the monolithic model that correspond to the components of that sensor (see figure 4.1); similarly, each single arc in D_G represents an influence of one Bayesian sensor over another, therefore, it is a super-arc of the set of arcs in the monolithic model that implement such influence (in general, there may be more than one arc from one sensor to another). For further details on abstraction in graphs, please refer to [137], [138].

The Bayesian sensory architecture

With the notions introduced above, it is now possible to define a Bayesian sensory architecture for the representation of robotic sensory systems. Recall, again, that this model is an extension of the one in [33] that also incorporates the simultaneous treatment of different levels of cognitive abstraction, the encoding of sensory temporal dynamics and the use of both discrete and continuous random variables.

One of the core parts of the proposed architecture, introduced in [33], is based on the partition of monolithic networks in order to improve the efficiency of inference, as mentioned before. This partition is defined by distributing Bayesian sensors into different sets called *layers*, according to their mutual dependencies. Such partition establishes a hierarchical ordering of sensors in which the lower ones explain or affect the behavior of the higher ones. Thus, a layer \mathbf{L}_α is a set of Bayesian sensors $\mathbf{L}_\alpha = \{\mathcal{B}_a, \mathcal{B}_b, \mathcal{B}_c, \dots\}$ that verify some conditions related to the mentioned interactions among sensors. Particularly, influences between sensors in different layers must be represented as in the dependency graph D_G , where any sensor that requires information from others appears as a descendant of them. In order to guarantee that, the construction of this *layered* part of the architecture is based on the *longest path layering* algorithm [139], [140]. The construction procedure proposed here establishes a partition of the nodes (i.e., Bayesian sensors) in D_G such that each of them is assigned to the layer whose number $\alpha \in \mathbb{N}$ is the length of the longest directed path ending at the considered sensor. This algorithm, which can be solved in polynomial time [141], calculates the longest path in a directed acyclic graph for each vertex by following a topological ordering, thus assuring that such path is always obtained from the ancestors of the considered vertex. For further details on how to solve the longest path problem in directed graphs please refer to [142] and [143].

The hierarchical partition of a sensory system into different layers is not the only aspect concerning its definition, since the different levels of abstraction as well as the temporal dynamics associated with the existing sensors also have to be taken into account. For that reason, the complete sensory system of a mobile robot is represented as a pair $\mathcal{S} = (D_G, \mathcal{L})$, where D_G is the annotated dependency graph introduced before (which includes the information about sensory dynamics and levels of abstraction) and $\mathcal{L} = \{\mathbf{L}_0, \mathbf{L}_1, \dots, \mathbf{L}_n\}$ is a set of all the existing layers in the sensory system (which contains the definition of all the Bayesian sensors in it). An example of the construction of this Bayesian sensory architecture will be described at the end of this section.

The proposed modeling has several implications in the way Bayesian sensors interact with each

²Note that the term *abstraction* used in this paragraph does not refer to a level of *cognitive* abstraction.

4.3 A representation of robotic sensory systems using Bayesian networks

other. Every level in \mathcal{L} is isolated from the rest and there may be dependencies among sensors assigned to different levels. For this reason, ideal nodes are allowed to be instantiated (i.e., replicated) for every Bayesian sensor that needs them, since they are not directly available as in the monolithic network. These replicated nodes serve as an *interface* to propagate information through the existing levels in the model, for the purposes of inference. The connection this interface represents is not always of the same nature, and it can be implemented by using replicated nodes in two different ways, as explained below.

The previously defined graph D_G encodes a dependency relationship between two sensors by representing one of them as a parent of the other. When the two sensors belong to the same level of cognitive abstraction, the dependency is said to be *purely layered*, since each sensor would be assigned to a different layer of the ones in \mathcal{L} . However, when the two sensors do not belong to the same level of abstraction, the dependency is said to be *cognitive*, since they are assigned to different layers and also to different levels of cognitive abstraction at the same time. In the first case, the connection is implemented by just replicating the ideal node corresponding to the parent sensor in the layer assigned to the child one. However, in the second case, two real nodes R and R' must be added as descendants of the replicated ideal one (see figure 4.1). This means that the connection between sensors belonging to different levels of cognitive abstraction is implemented by asserting *virtual* evidence in an auxiliary Bayesian sensor created in the higher level.

Note that the definition of Bayesian sensor also contemplates the incorporation of sensory dynamics over discrete time intervals. This is done by using replicated ideal nodes representing ideal variables in the immediately previous time interval (e.g, $t - 1$) as parents of the ones referring to the current interval t (see figure 4.1), which is actually based on the definition of transition network in the context of DBNs.

For the reasons explained above, the proposed sensory architecture is grounded on the propagation of information in three different ways (i.e., layered, cognitive and dynamic). As a result, the architecture could be described as a three-dimensional model whose axes are:

- **Layered axis:** it represents an ordering of sensors according to their mutual dependencies, and thus, to their behavior in the system.
- **Temporal axis:** it describes the evolution of the information represented by a particular sensor over discrete time intervals.
- **Cognitive axis:** it represents different levels of cognitive abstraction (i.e., ontologies).

There is an extra constraint that has to be taken into account before constructing the complete sensory architecture. The inference algorithm developed in the thesis for this model, presented in section 4.4, is conceived to handle CLG Bayesian networks only (see chapter 2), i.e., networks in which discrete nodes are not allowed to have continuous parents. For this reason, the architecture construction procedure is affected every time a sensor defined over continuous variables needs to convey information to another one based on discrete variables. If this happens, the connection between sensors is implemented as in the case of cognitive dependencies (i.e., through virtual evidence) with the exception that the support of the replicated ideal variable would also be discretized.

4.3 A representation of robotic sensory systems using Bayesian networks

Algorithm 7 Construction of the Bayesian architecture for a robotic sensory system

input:

\mathbf{B}_{id} : updated list of identifiers of Bayesian sensors (algorithm 6)

D_G : annotated dependency graph (algorithm 6)

output:

\mathcal{S} : sensory system

subroutines:

$parents(G, n)$:

$P \leftarrow$ subset of vertices in graph G that are parents of node n

return P

$cognitiveConnection(D_G, i, j)$:

$C \leftarrow$ boolean value indicating whether nodes i and j in D_G belong to a different cognitive level

return C

$cdConnection(i, j)$:

$C \leftarrow$ boolean value indicating whether Bayesian sensors \mathcal{B}_i and \mathcal{B}_j are based on discrete and continuous variables respectively

return C

$longestPath(G, n)$:

$\alpha \leftarrow$ length of the longest directed path in graph G ending at node n (see [142])

return α

main:

1: $\mathcal{L} \leftarrow \emptyset$ (empty set of layers)

2: **for each** identifier $i \in \mathbf{B}_{\text{id}}$ **do**

3: $P \leftarrow parents(D_G, i)$

4: $\mathcal{B}_i \leftarrow$ design a complete Bayesian sensor (see figure 4.1) including new instances of the ideal nodes from the Bayesian sensors identified by P

5: **for each** node $j \in P$ **do**

6: $CC \leftarrow cognitiveConnection(D_G, i, j)$

7: $CD \leftarrow cdConnection(i, j)$

8: **if** $(CC \vee CD)$ is true **then**

9: add auxiliary Bayesian sensor for virtual evidence in node j of \mathcal{B}_i

10: **if** CD is true **then**

11: discretize the support of variable associated with node j of \mathcal{B}_i

12: **end if**

13: **end if**

14: **end for**

15: $\alpha \leftarrow longestPath(D_G, i)$

16: **if** $\nexists \mathbf{L}_\alpha \in \mathcal{L}$ **then**

17: $\mathbf{L}_\alpha \leftarrow \emptyset$ (empty set of Bayesian sensors)

18: $\mathbf{L}_\alpha \leftarrow \mathbf{L}_\alpha \cup \mathcal{B}_i$

19: $\mathcal{L} \leftarrow \mathcal{L} \cup \mathbf{L}_\alpha$

20: **else**

21: $\mathbf{L}_\alpha \leftarrow \mathbf{L}_\alpha \cup \mathcal{B}_i$

22: replace old layer $\mathbf{L}_\alpha \in \mathcal{L}$ with the current one

23: **end if**

24: **end for**

25: $\mathcal{S} = (D_G, \mathcal{L})$

26: **return** \mathcal{S}

4.4 An inference algorithm for approximate reasoning in robotic sensory systems

Finally, a model of the complete sensory system can be obtained by implementing algorithm 7. This is a construction procedure for the sensory architecture given an updated list of Bayesian sensors and an annotated dependency graph for them (both obtained from the application of algorithm 6). The procedure can be viewed as a *refinement* [137] of the dependency graph, since the user is asked to provide the complete network for each Bayesian sensor in the system (thus, a greater amount of detail is required). Once the algorithm is finished, the resulting model is prepared for approximate inference (more details in section 4.4).

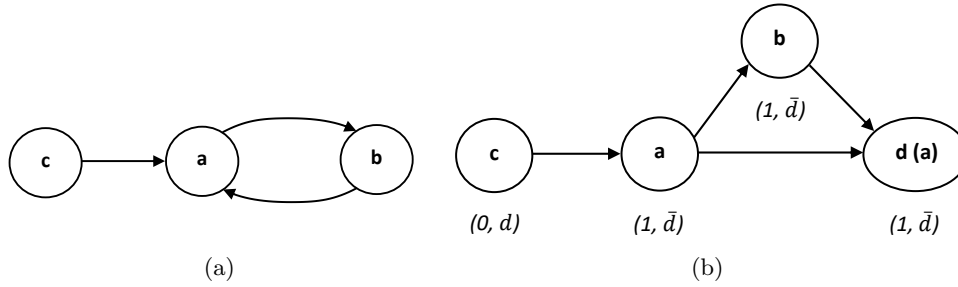


Figure 4.3: Dependencies among Bayesian sensors for the example in the text. (a). Cyclic directed graph form. (b) Annotated dependency graph form. Recall that the sensor labeled as d is a new instance of sensor a .

The construction procedure presented here is now illustrated more clearly with an example. Consider, for instance, a sensory system with three Bayesian sensors \mathcal{B}_a , \mathcal{B}_b and \mathcal{B}_c (for simplicity, they will be referred to as a , b and c). In this example, sensors b and c are defined over continuous random variables, and sensor a over discrete ones. Concerning cognitive levels, the ontologies related to sensor c belong to a low level of cognitive abstraction, while the ones related to sensors a and b belong to a higher level. Regarding temporal dynamics, only sensor c represents information that evolves over time. Also, there is a cyclic dependency between sensors a and b , while sensor c influences sensor a (see figure 4.3(a)). The lists representing all the previous constraints are as follows: $\mathbf{B}_{id} = \{a, b, c\}$, $\mathbf{C}_{id} = \{1, 1, 0\}$, $\mathbf{T}_{id} = \{\bar{d}, \bar{d}, d\}$, $\mathbf{D}_{nc} = \{(c, a)\}$ and $\mathbf{D}_c = \{(a, b), (b, a)\}$. By applying algorithm 6, the set of dependencies is updated so that $\mathbf{D}_{nc} = \{(c, a), (a, b), (b, d), (a, d)\}$ (in this case, a new instance of sensor a named d is created). The resulting annotated dependency graph is shown in figure 4.3(b); the monolithic model of the sensory system is depicted in figure 4.4(a) and the resulting architecture is shown in figure 4.4(b). This theoretical example aims to show all the capabilities of the proposed architecture at once. Please refer to chapter 5 for implementations in real robotic contexts (sections 5.4.1 and 5.5.1).

4.4 An inference algorithm for approximate reasoning in robotic sensory systems

This section covers the introduction of the algorithm proposed for approximate reasoning in robotic sensory systems. This method represents one of the core contributions of the thesis along with the Bayesian sensory architecture defined before. The version of the algorithm presented here is an extension of the one in [33] that also allows inference in models incorporating the most recent features of the Bayesian architecture, i.e., in networks with both continuous and discrete variables, several levels of abstraction and sensory dynamics. The proposed algorithm aims to reduce the computational cost by leveraging the particular structure of these networks, which are an approximate representation of the monolithic model of the whole sensory system. The method described

4.4 An inference algorithm for approximate reasoning in robotic sensory systems

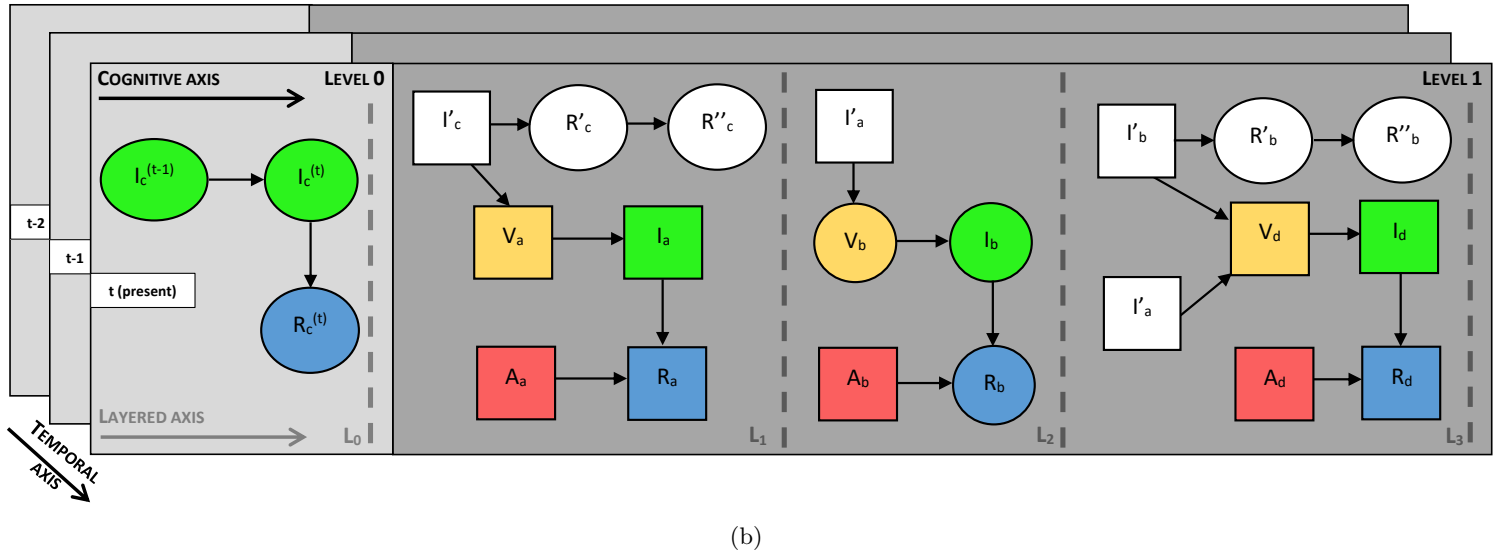
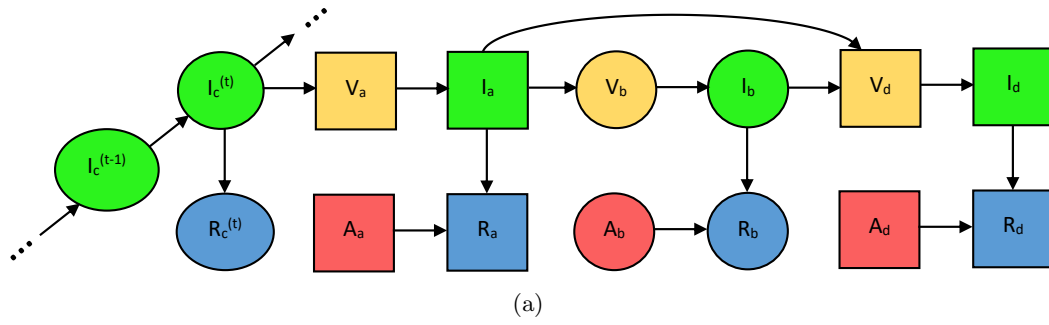


Figure 4.4: Bayesian network representing the sensory system for the example in the text. Colours indicate the kind of node according to the definition of Bayesian sensor (see figure 4.1). Here, replicated ideal nodes and auxiliary Bayesian sensors are in white. Squared nodes represent discrete random variables and round ones represent continuous random variables. The nodes that are not marked with any time interval actually correspond to the current one t . (a) Transition network corresponding to the monolithic form of the sensory system. (b) Three-dimensional approximate model of the system obtained from algorithm 7.

here is conceived to perform exact inference in those approximate models, thus, it will produce approximate results.

4.4.1 Proposed methodology

The inference methodology is based on the propagation of posterior distributions from sensors in lower layers of the architecture (specifically, from their ideal nodes) as prior distributions or virtual evidences in their corresponding instances in higher layers (placed in virtual or anomalies subnetworks). Note that these instances also represent the cognitive and dynamic dimensions in each layer, as explained before. Recall, also, that there are no edges or nodes shared by different layers in the network (each layer is an isolated Bayesian network), therefore these are the only ways of conveying information from one layer to another. Although the mentioned posterior distributions have the form expressed in equation (4.3), i.e., $P(I|\mathbf{E} = \mathbf{e})$, the evidence set \mathbf{E} is built differently, as explained later on. The process is done for all the sensors in the model beginning from the *zero* layer, and propagating the inference sequentially, i.e., inference in a layer cannot be performed until the previous layer is processed. The inference algorithm ends once the desired layer is reached,

4.4 An inference algorithm for approximate reasoning in robotic sensory systems

thus it is not necessary to process all of them. However, in the case that there is any dynamic sensor in the architecture, the whole process is repeated for every time interval t until the desired one t^* is reached, taking into account that all the layers in the architecture must be processed for $t < t^*$. In this case, the obtained posterior distributions for dynamic sensors are stored for their use in future time intervals. Inference in a given layer is done by applying an adapted version of the exact *jointree* algorithm for CLG Bayesian networks [48], [57], although any other inference method could be used here.

As explained in section 4.3.2, the propagation of information (represented in this case by posterior distributions) can be performed in three different ways, and this is done within each layer of the model as follows. First, a posterior distribution corresponding to some ideal node is obtained by processing its corresponding layer. Then, the distribution can be used in a different layer in one out of three different ways depending on the kind of connection existing between the sensors involved. If the dependency is purely *layered*, the posterior is encoded as a prior distribution of the corresponding replicated ideal node. In the case the dependency is *cognitive*, the posterior is encoded as an uncertain observation (i.e., as virtual evidence) of the auxiliary Bayesian sensor created in the target layer (see algorithm 7). Lastly, if the posterior is obtained in a previous time interval, the distribution is encoded as a prior of the corresponding ideal instance in the desired layer for the current time interval.

The proposed inference methodology is presented more formally in algorithm 9 (which uses the subroutine described in algorithm 8). The method solves each layer by applying the exact *jointree* inference algorithm, which, as explained in chapter 2, transforms the input Bayesian network (i.e., the corresponding layer) into a secondary structure called *jointree* that is used to finally perform inference. Recall that a *jointree*, \mathcal{J} , is an undirected graph (more specifically, a tree) whose nodes are mapped to sets of variables of the original problem named *clusters*. In addition, the algorithm uses sets of joint probability distributions, known as *factors* or *potentials*, $\mathcal{F} = \{\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_n\}$, and one of each is assigned to a cluster in the *jointree*. To query for a desired variable or set of variables, \mathbf{Q} , it is enough to marginalize one of these distributions as long as it is defined it over a set of variables containing \mathbf{Q} .

The proposed inference algorithm may also be approximate due to another reason. As explained before, the algorithm propagates the information from layer zero to the desired one, regardless of the following higher layers. In general, it will not consider, for a query related to a particular sensor, the evidences associated with those sensors from which there is no directed path to the sensor of interest. This can be seen more clearly by using the dependency graph, D_G . Consider a sensory system whose dependency graph is the one depicted in figure 4.5.

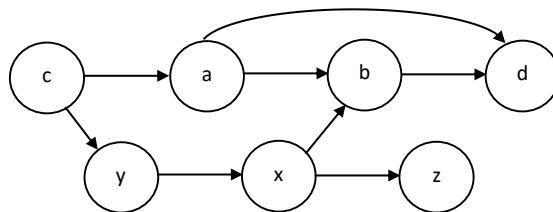


Figure 4.5: Dependency graph for the example about the impact of approximating the evidence set. Here, the graph is not annotated for simplicity, since there are no different cognitive levels or sensory dynamics that need to be represented for the example.



4.4 An inference algorithm for approximate reasoning in robotic sensory systems

Algorithm 8 *processLayer*($\mathbf{L}_i, \mathbf{e}, t, \mathcal{T}^{(t-1)}, \mathcal{T}^{(t)}$)

input:

\mathbf{L}_i : i -th layer from \mathcal{L} in sensory system \mathcal{S}

\mathbf{e} : set of evidences from sensory data

t : current time interval (if necessary)

$\mathcal{T}^{(t-1)}$: set of former posterior distributions for ideal nodes in dynamic sensors (if necessary)

$\mathcal{T}^{(t)}$: equivalent set of posteriors for the current time interval (if necessary)

output:

\mathcal{P} : set of posterior distributions of ideal and anomaly nodes of layer \mathbf{L}_i

$\mathcal{T}^{(t)}$: updated set of current dynamic posteriors

subroutines:

joinTree(\mathcal{N}, \mathbf{e}):

$\mathcal{D} \leftarrow \emptyset$

$(\mathcal{J}, \mathcal{F}) \leftarrow$ apply the exact *jointree* algorithm to network \mathcal{N} with evidences \mathbf{e}

for each ideal node I_j and anomaly node A_k in network \mathcal{N} **do**

$\mathbf{P} \leftarrow$ get distribution $P(I_j|\mathbf{e})$ from an appropriate $\mathbf{F}_i \in \mathcal{F}$

$\mathcal{D} \leftarrow \mathcal{D} \cup (\mathbf{P}, I_j)$

$\mathbf{P} \leftarrow$ get distribution $P(A_k|\mathbf{e})$ from an appropriate $\mathbf{F}_i \in \mathcal{F}$

$\mathcal{D} \leftarrow \mathcal{D} \cup (\mathbf{P}, A_k)$

end for

return \mathcal{D}

updateDynamics($\mathbf{L}_i, \mathcal{P}$):

$\mathbf{L}_i \leftarrow$ update priors of all the ideal nodes of dynamic sensors in layer \mathbf{L}_i using posteriors from \mathcal{P}

return \mathbf{L}_i

storeDynamics($\mathcal{T}^{(t)}, \mathcal{P}$):

$\mathcal{D} \leftarrow$ set of pairs $(\mathbf{P}, I_i^{(t)})$ representing posteriors of dynamic ideal variables from \mathcal{P}

$\mathcal{T}^{(t)} \leftarrow \mathcal{T}^{(t)} \cup \mathcal{D}$

return $\mathcal{T}^{(t)}$

main:

1: $\mathbf{e}_i \leftarrow$ subset of evidences for layer \mathbf{L}_i ($\mathbf{e}_i \subset \mathbf{e}$)

2: **if** $t > 0$ **then**

3: $\mathbf{L}_i \leftarrow$ *updateDynamics*($\mathbf{L}_i, \mathcal{T}^{(t-1)}$)

4: **end if**

5: $\mathcal{P} \leftarrow$ *joinTree*($\mathbf{L}_i, \mathbf{e}_i$)

6: $\mathcal{T}^{(t)} \leftarrow$ *storeDynamics*($\mathcal{T}^{(t)}, \mathcal{P}$)

7: **return** $(\mathcal{P}, \mathcal{T}^{(t)})$

Suppose that an user wants to query about the ideal state of sensor x . In that case, the inference algorithm only considers the evidences coming from sensors c and y , as there are no directed paths from the rest towards x . Similarly, an inference task involving sensor d would take into account the evidences from all sensors except z , since it has not the opportunity to propagate its information through the model towards sensor d .

Summarizing all the above, the proposed inference method provides an approximation given by:

$$P_e(\mathbf{Q}|\mathbf{E}) \approx P_a(\mathbf{Q}|\mathbf{E}_r), \quad (4.6)$$

4.4 An inference algorithm for approximate reasoning in robotic sensory systems

Algorithm 9 Approximate inference in a Bayesian sensory architecture

input:

\mathcal{S} : sensory system (see algorithm 7)
 \mathbf{e} : set of evidences from sensory data
 Q : query variable (ideal or anomaly node)
 L : layer associated with variable Q
 t : current time interval (if necessary)
 $\mathcal{T}^{(t-1)}$: set of former posterior distributions for ideal nodes in dynamic sensors (if necessary)

output:

$P(Q|\mathbf{E}_r = \mathbf{e}_r)$: posterior distribution for variable Q given the *reduced* evidence (see text)
 $\mathcal{T}^{(t)}$: set of current posterior distributions for ideal nodes in dynamic sensors (if necessary)

subroutines:

getDistribution(\mathcal{D}, N):
return \mathbf{P} as the probability distribution in \mathcal{D} associated with node N

main:

```

1:  $\mathcal{T}^{(t)} \leftarrow \emptyset$  (empty set of posteriors)
2:  $\mathcal{L} \leftarrow$  initialize all the priors corresponding to replicated ideal nodes with uniform distributions
3:  $(\mathcal{P}, \mathcal{T}^{(t)}) \leftarrow processLayer(\mathbf{L}_0, \mathbf{e}, t, \mathcal{T}^{(t-1)}, \mathcal{T}^{(t)})$ 
4: for  $i = 1$  to  $L$  do
5:   for each Bayesian sensor  $\mathcal{B}_j \in \mathbf{L}_i$  do
6:      $\mathbf{C} \leftarrow$  set of replicated ideal nodes in  $\mathcal{B}_j$ 
7:     for each node  $c_k \in \mathbf{C}$  do
8:        $\mathbf{P}_k \leftarrow getDistribution(\mathcal{P}, c_k)$ 
9:       if  $c_k$  is part of an auxiliary Bayesian sensor (see algorithm 7) then
10:         assert virtual evidence in that sensor by using  $\mathbf{P}_k$ 
11:       else
12:         replace current CPD for node  $c_k$  by  $\mathbf{P}_k$ 
13:       end if
14:     end for
15:   end for
16:    $(\mathcal{P}, \mathcal{T}^{(t)}) \leftarrow processLayer(\mathbf{L}_i, \mathbf{e}, t, \mathcal{T}^{(t-1)}, \mathcal{T}^{(t)})$ 
17: end for
18:  $P(Q|\mathbf{E}_r = \mathbf{e}_r) \leftarrow getDistribution(\mathcal{P}, Q)$ 
19: return  $(P(Q|\mathbf{E}_r = \mathbf{e}_r), \mathcal{T}^{(t)})$ 

```

where P_e and P_a denote distributions over the exact and the approximate model, respectively, \mathbf{Q} is the query set (equal to only one ideal or anomaly node associated with a certain layer, i.e., $|\mathbf{Q}| = 1$) and \mathbf{E} is the evidence set as defined in equation (4.1). In addition, $\mathbf{E}_r \subseteq \mathbf{E}$ is a reduced set of evidence variables such that:

$$\mathbf{E}_r = \{E | E \in \mathbf{E} \wedge directedPath(\mathbf{Q}, E) : E, \mathbf{Q} \in S_g\}, \quad (4.7)$$

where *directedPath* is true if there is at least one directed path from the variable in \mathbf{Q} towards variable E in the monolithic graph structure S_g (the associated dependency graph D_G could also be used for that). Furthermore, the following still represents an approximation:

$$P_e(\mathbf{Q}|\mathbf{E}_r) \approx P_a(\mathbf{Q}|\mathbf{E}_r). \quad (4.8)$$

In this case, only the impact of using an approximate model is considered. Equations (4.6) and (4.8) could become into equalities under specific conditions, as shown later on.

The approximations in equations (4.6) and (4.8) are stated within the context of a static problem. In the dynamic case, these approximations are valid for a given time interval t , regardless of the others. The format of query used here for inference in dynamic networks is the one of *filtering*, as mentioned before, which accumulates evidences from the initial time interval to the present one. This implies that, for a filtering query in a particular variable, the set of reduced evidences \mathbf{E}_r would lack the same subset of variables as in the static case for all the considered time intervals. As a result, the approximations could be re-formulated for the dynamic case such that:

$$P_e(\mathbf{Q}^{(t)}|\mathbf{E}^{(0:t)}) \approx P_a(\mathbf{Q}^{(t)}|\mathbf{E}_r^{(0:t)}) \quad (4.9)$$

and

$$P_e(\mathbf{Q}^{(t)}|\mathbf{E}_r^{(0:t)}) \approx P_a(\mathbf{Q}^{(t)}|\mathbf{E}_r^{(0:t)}). \quad (4.10)$$

There is also another issue that needs to be considered regarding the approximations stated above. In general, it is not always possible to compare these approximate inferences to an exact distribution defined over a complete monolithic model, for two different reasons. One of them is due to the fact that the monolithic model of a sensory system is not conceived to be a CLG Bayesian network in general, since it may contain discrete variables with continuous parents (this happens in the model of figure 4.4(a)). Exact inference is not possible in such a model, and the existing alternatives address the inference problem by using approximate numerical integration and particle-based methods [48]. The other reason refers to the treatment of different cognitive levels in the same network (which also happens in the model of figure 4.4(a)). As explained before, this might lead to inaccurate inference results, and the solution proposed here to avoid that is to split the model into separate parts through the mechanism of virtual evidence. For both situations, such an splitted monolithic model will be the one considered for the definition of the *exact* distributions referred to in the approximations stated above.

In summary, the proposed method is approximate mainly due to two reasons: it performs inference using an approximate model and it might not take into account the whole set of available evidences. However, the algorithm can be exact under some conditions, which will be explained by using a simple example. Consider the Bayesian network in figure 4.6(a) and its corresponding approximate version in figure 4.6(b). This is a simplistic robotic system with two discrete and static sensors belonging to the same level of cognitive abstraction. The corresponding virtual and anomalies subnetworks have been omitted (since only the ideal and real nodes for each sensor are used), but the results derived below hold anyway.

Firstly, it will be illustrated that the proposed method can be exact if the network structure verifies some conditions. For that, consider the exact ideal distribution corresponding to the sensor in layer one, $P_e(I_1|R_0, R_1)$. Here, probability distributions derived from the monolithic, exact model will be denoted as P_e , while those from the approximate network will be referred to as P_a . If a distribution is not denoted as mentioned, it is because such distribution is equal for both models (CPDs are defined exactly the same where possible). According to the inference process described before, which uses the approximate network, it is possible to get an approximate result. First, $P_a(I_0|R_0)$ is obtained from layer zero and then imposed as the prior distribution of its copy, I'_0 , in layer one. Recall that both R_0 and R_1 must be instantiated to some values, r_0 and r_1 , by using evidence data. In other words:

$$P_a(I'_0) = P_a(I_0|R_0 = r_0). \quad (4.11)$$

4.4 An inference algorithm for approximate reasoning in robotic sensory systems

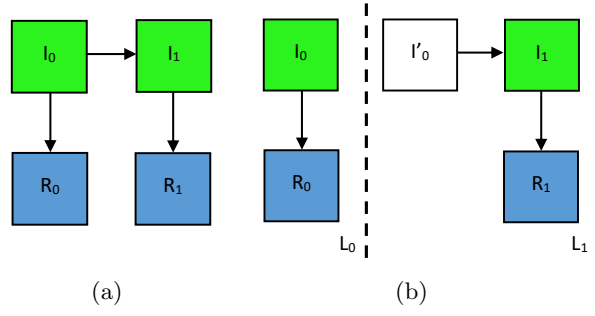


Figure 4.6: Bayesian network for the inference example in the text. Again, squared nodes represent discrete random variables. (a) Monolithic form. (b) Approximate form.

Once this is done, $P_a(I_1|R_1)$ is calculated as an approximation of the desired distribution, $P_e(I_1|R_0, R_1)$. The result can be expressed as follows:

$$P_a(I_1|R_1) = \frac{P_a(I_1, R_1)}{P_a(R_1)} = \frac{\sum_{I'_0} P_a(I'_0)P_a(I_1|I'_0)P(R_1|I_1)}{\sum_{I'_0, I_1} P_a(I'_0)P_a(I_1|I'_0)P(R_1|I_1)}, \quad (4.12)$$

where *d-separation* has been used. Now, substituting $P_a(I_1|I'_0)$ by its exact definition, $P_e(I_1|I_0)$ (both are the same), and $P_a(I'_0)$ by equation (4.11):

$$P_a(I_1|R_1) = \frac{\sum_{I_0} P(I_0)P(R_0|I_0)P(I_1|I_0)P(R_1|I_1)}{\sum_{I_0, I_1} P(I_0)P(R_0|I_0)P(I_1|I_0)P(R_1|I_1)}, \quad (4.13)$$

where $P_a(I_0|R_0)$ has been simplified by $P(R_0|I_0)$ by applying the Bayes' theorem. Equation (4.13) expresses the final result for the approximate inference algorithm. Should an exact algorithm had been applied to the monolithic network, the result would have been:

$$P_e(I_1|R_0, R_1) = \frac{P_e(I_1, R_0, R_1)}{P_e(R_0, R_1)} = \frac{\sum_{I_0} P(I_0)P(R_0|I_0)P(I_1|I_0)P(R_1|I_1)}{\sum_{I_0, I_1} P(I_0)P(R_0|I_0)P(I_1|I_0)P(R_1|I_1)} \quad (4.14)$$

Therefore $P_e(I_1|R_0, R_1) = P_a(I_1|R_1)$.

In this case, the result obtained from the proposed inference algorithm is actually exact. This will happen as long as the exact inference model corresponds to a *polytree* structure [58] and the set of evidence variables \mathbf{E} is the same in both models. In general, the proposed inference algorithm provides approximate results for *multiply connected* networks. As an example, consider again the sensory system in figure 4.4. The exact ideal distribution for sensor d may differ from the approximate one, since the exact model has an undirected loop and therefore it is not a *polytree*.

Now, the network example in figure 4.6 will be used again to illustrate that it is possible to perform exact inference despite the fact that some evidences are ignored. For that, consider the ideal distribution for the sensor in layer zero, $P_e(I_0|R_0, R_1)$. The proposed algorithm allows us to

get an approximation, $P_a(I_0|R_0) \approx P_e(I_0|R_0, R_1)$, in which the possible impact of the observation R_1 has not been considered. However, this approximation would become an equality if variables I_0 and I_1 were independent (i.e., disconnected in the graph structure). This can be proved considering both distributions:

$$P_e(I_0|R_0, R_1) = \frac{P_e(I_0, R_0, R_1)}{P_e(R_0, R_1)} = \frac{P(I_0)P(R_0|I_0) \sum_{I_1} P(I_1|I_0)P(R_1|I_1)}{\sum_{I_0} \left(P(I_0)P(R_0|I_0) \sum_{I_1} P(I_1|I_0)P(R_1|I_1) \right)} \quad (4.15)$$

and

$$P_a(I_0|R_0) = \frac{P_a(I_0, R_0)}{P_a(R_0)} = \frac{P(I_0)P(R_0|I_0)}{\sum_{I_0} P(I_0)P(R_0|I_0)} \quad (4.16)$$

As a result, if $I_0 \perp I_1$, then $P(I_1|I_0) = P(I_1)$ and, by substituting this in equation (4.15), it becomes equation (4.16), therefore $P_a(I_0|R_0) = P_e(I_0|R_0, R_1)$.

4.4.2 Computational complexity

With the previous notions, the computational complexity of the proposed algorithm can now be analyzed. For the exact *jointree* method, this complexity reduces to [58]:

$$O(n \exp(w)), \quad (4.17)$$

where n is the number of clusters (nodes) in the *jointree* and w is the *jointree width*, defined as the size of the largest cluster minus one, also known as the network *treewidth* (since it depends on the particular structure of the corresponding Bayesian network). Note that the complexity is exponential in this parameter, w , and also, that this exponential character of the cost is given by the presence of discrete random variables (see section 2.2.2 in chapter 2).

The proposed inference algorithm applies this exact method once for each layer in the architecture, thus, the complexity for one time interval can be expressed as:

$$O(n_1 \exp(w_1) + n_2 \exp(w_2) + \dots + n_i \exp(w_i) + \dots + n_L \exp(w_L)), \quad (4.18)$$

where L is the number of layers in the architecture. Note that this expression of the cost should be multiplied by the total number of time intervals (from the initial to the present one), however, this is not necessary in the case of the proposed algorithm, for the following reason. Many inference methods for DBNs allow the *online* obtention of filtering queries by considering the complete set of evidences, from the initial time interval to the present one, thus incorporating all those evidences every time that new ones are available. The proposed algorithm, which can also be used for *online* queries, only incorporates evidences related to a particular time interval. This implies that it is not necessary to accumulate all the existing evidences as in the mentioned algorithms, since the proposed one propagates dynamic information across different time intervals by storing certain posterior distributions, as explained before. Thus, the cost expressed in equation (4.18) should be multiplied by the number of time intervals only if the inference algorithm is used to process a sequence of evidences *offline*.

Recall that each parameter in equation (4.18) is different depending on the considered layer. Regarding the relation between (4.17) and (4.18), it can be ensured that $w_i \leq w_m$, where w_m is the treewidth associated with the monolithic Bayesian network model. This is always true for the following reasons. The network treewidth is higher as the number of parents per node increases, and also the presence of undirected loops in the Bayesian network tend to increase the treewidth. Note that transforming an approximate network into a monolithic one never leads to a greater number of parents per node (this also holds for the reverse operation). However, an approximate network eliminates edges established among different layers in the monolithic version. These relations are encoded in the dependency graph. In the case that this graph corresponds to a *multiply connected* structure, the treewidth can be reduced depending on the particular interactions of undirected loops in this graph, since all the edges in it will not be present in the corresponding approximate network.

Boosting inference efficiency with feedforward neural networks

The proposed approximate inference method enables to significantly reduce the computational cost in different situations (see the experimental evaluation developed in chapter 5). However, it is possible to increase the efficiency of this algorithm even more by leveraging certain aspects of its implementation and the form of typical queries in a sensory system. For that, it is important to note that a Bayesian network can be seen as a family of functions, each one corresponding to a particular query $P(\mathbf{Q}|\mathbf{E})$; thus, a BN defined over variables \mathbf{Z} represents a single function for each concrete partition of this set into \mathbf{Q} and \mathbf{E} .

In the proposed sensory architecture, a query always has the form expressed either in equation (4.2) or in (4.3), thus, it is possible to represent it as a function of some parameters (usually, the values of the evidence set, among others). This process is known as *compilation* of Bayesian networks [58], and there exist different methods to implement it. The proposal presented in this thesis relies on the use of feedforward neural networks [56] for that. The strategy to be followed consists in the design of a different neural network for each possible query related to the sensory system. Once trained, these networks could be used to obtain approximations of the queries they represent. This is supposed to considerably reduce the cost of inference, since evaluating a single function should be, in principle, much less demanding than performing all the steps an inference algorithm requires, even for a small network.

The training process could be performed, for instance, by using a monolithic model of the system. In that case, the training dataset would be formed by possible instantiations of the whole evidence set \mathbf{E} as inputs, and by the resulting parameters of the queried distributions as outputs (also called targets). Unfortunately, training sensory queries with this approach poses some important drawbacks. One of them is related to the very obtention of the training dataset. As explained before, inference in potentially large monolithic networks can be extremely inefficient in general, and possibly impractical, which would hinder or prevent the obtention of the necessary output data. The other problem, related to the training process itself, relies on the fact that the cardinality of the evidence set may be arbitrarily high. Such a large set would lead to a neural network with a large amount of inputs, which could be, in turn, really difficult to train.

The proposed Bayesian architecture allows to alleviate these computational requirements for the training process, since a query is always related to only one layer of the architecture, and therefore, it is not necessary to perform inference in the whole monolithic network. Also, only the subset of evidences corresponding to the layer of interest needs to be considered. However, new

4.4 An inference algorithm for approximate reasoning in robotic sensory systems

input data are required for training in this case. Recall that a layer is a Bayesian network including an interface formed by replicated ideal nodes that need to be parameterized. Thus, these CPD parameters have to be included in the training dataset in addition to the evidence values. Note, however, that the incorporation of these parameters makes inference in lower layers unnecessary, which represents an advantage with respect to the monolithic approach.

The method for reducing the cost of inference discussed here has been implemented and tested for a particular sensory system. The obtained results will be presented in chapter 5.

Implementation and validation

The present chapter covers the implementation and validation of the proposed Bayesian sensory architecture and its corresponding inference algorithm in different simulated and real robotic applications. First, the most relevant works developed during the thesis concerning such experimental evaluation are reported. Then, the chapter presents two different implementations of the proposed approach. One of them is devoted to the analysis of the contributed approximate inference algorithm in terms of error and computation time, carried out on both real and simulated experiments. The other implementation is aimed at showing the utility of the Bayesian sensory architecture in the context of robotic navigation in environments with human presence. This proposal is also evaluated through different experiments in simulated and real settings, proving that its use enables for a safer and more efficient navigation. Lastly, a new inference approach based on the use of feedforward neural networks is implemented and tested for this problem, showing that it is possible to significantly reduce the cost of inference with Bayesian networks.

5.1 Introduction

The so-called *Bayesian sensory architecture* defined in chapter 4 for the representation of sensory systems has been implemented and evaluated in different applications related to mobile robotics. Also, the performance of the proposed inference algorithm has been experimentally studied in terms of error and computation time. These contributions have been reported in several works related to the thesis, which describe an evolution of the mentioned proposal whose culmination is presented here.

The first of these works [53] introduces a preliminary version of the proposed architecture that is tested in a real platform for the detection and recovery of basic sensory devices (encoders, bumpers, gyroscopes, etc.). This definition of the architecture is extended in [50] to allow for the treatment of more complex sensory devices such as laser rangefinders and depth cameras, i.e., the definition is revised there to enable the incorporation of vectorial and matricial sensory data. The experiments

5.1 Introduction

carried out for these works (see some of them in figure 5.1) prove the effectiveness of the proposed approach, but only for static situations (i.e., the robot does not move in those settings). The work [38] is the first one evaluating the proposal in dynamic situations, more concretely in a mixed indoor-outdoor scenario. That work also proposes the integration of temporal knowledge through the use of different sequential filters.

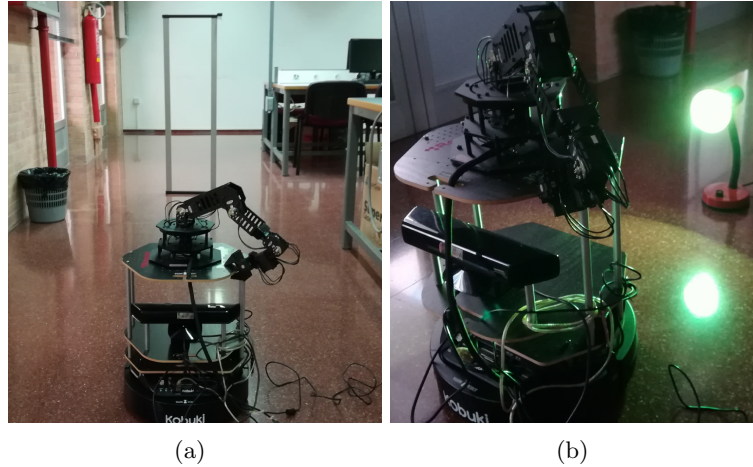


Figure 5.1: Pictures of the real setup used for some of the static experiments carried out in [50]. The platform employed in these tests is the CRUMB robot. The experiments are related to the range sensors it incorporates (see section 5.3). (a) Experiment devised to detect anomalous range observations due to the presence of highly specular surfaces (metal frame in the image). (b) Experiment aimed at the detection of corrupted range data due to intense infrared radiation.

The works mentioned so far implement the model of the robot sensory system by using a monolithic representation, which is then employed to perform inference by relying on classical algorithms. This usually leads to a high computational cost that severely compromises the real-time capabilities of any moving robot. To solve that, the work [33], one of the most relevant to the thesis, introduces a revised version of the architecture and a novel inference method that relies on it for improving efficiency. The work [52] complements this renewed definition of the model to formally include the encoding of sensory dynamics, the reasoning at different levels of cognitive abstraction and the use of both discrete and continuous random variables. Finally, one of the last tasks developed during the thesis is aimed at improving the inference efficiency even more by compiling Bayesian networks queries with feedforward neural networks.

The rest of this chapter is organized as follows. Section 5.2 presents a brief review of the existing works related to robotic navigation in environments with humans, identifying some of their current limitations and discussing possible solutions based on the Bayesian architecture. Section 5.3 describes all the simulated and real environments employed for the validation tasks covered in this chapter. Section 5.4 is dedicated to the evaluation of the proposed inference method in both simulated and real experiments. Finally, section 5.5 covers the implementation of the Bayesian architecture for the problem of navigation in human environments, which is assessed in simulations and real experiments as well. This section provides a solution based on feedforward neural networks to improve the efficiency of inference, whose performance in terms of error and computation time is also discussed.

5.2 Navigation in environments with human presence

Navigation in environments with dynamic obstacles constitutes a key part of countless applications related to service robotics, such as industrial, medical and domestic, among many others [2]. Safety represents a major concern in these applications, since it is crucial to preserve the integrity of the agents (human and robots) involved. The challenges that represent a safe robotic operation have been extensively addressed from the perspective of navigation [144], [25], however, the presence of human agents, usually referred to as pedestrians in this scope, poses some problems related to their detection and tracking that must be solved prior to the integration of mobile service robots in the mentioned applications.

The areas of research dedicated to pedestrian detection and tracking have received numerous contributions in the recent years. Some of the existing solutions tackle this problem from the perspective of computer vision [145], while others also integrate the use of probabilistic methods such as particle filters [146], Markov Decision Processes [147] and Bayesian networks [148]. The problem has also been approached with non-probabilistic methods from machine learning [149], and much more abundantly, using the advances from the field of deep learning [150]. A common concern present in all the mentioned contributions is the problem of occlusion, which prevents or hinders the pedestrian detection in presence of static and/or dynamic obstacles. This has been thoroughly studied for a wide variety of situations in these works, e.g., for both indoor and outdoor settings, known and unknown scenarios, etc.

Another line of research related to pedestrian tracking is the one concerned with the modeling of human intentions, and more specifically, with the prediction of human motion. Again, the existing works are numerous and the problem is addressed from a variety of perspectives (see [30] for a complete taxonomy). Some of them base their solutions on the use of deep neural networks [151], while others are grounded probabilistic methods. Among the latter, it is common the use of Kalman or particle filters [152], [153] to estimate or even learn the pedestrian motion model from existing data. Also, the integration of contextual cues and other expert knowledge through Bayesian networks produce good predictions of the motion trajectory [154], [155].

In summary, the existing works on pedestrian detection and tracking provide a wide variety of solutions, allowing their integration for real navigation tasks. However, they still have some important limitations. First, the majority of existing approaches only tackle the pedestrian occlusion problem when this situation does not persist: they fail in the case of tracking pedestrians that are suffering from *long-term* occlusion. Although there are a few contributions in the literature that address this issue [156], [157], these are conceived for very specific situations and do not allow for a flexible integration of knowledge that would help to achieve generality. In this regard, existing solutions grounded on Bayesian networks, which potentially can benefit from the integration of diverse expert knowledge, are usually defined over a rigid scheme that, again, is only applicable to specific contexts.

To overcome these limitations, the general Bayesian sensory architecture proposed in this thesis can be used to tackle the long-term occlusion problem as well as other issues related to the identity of the pedestrians being tracked. For that, it can rely on fusing the sensory information obtained from a state-of-the-art pedestrian detection and tracking system with other sources of knowledge. All the latest features of the architecture need to be exploited in this case, since the mentioned sensory information is related to different levels of cognitive abstraction and also evolves over time. This implementation has been tested through a set of simulated and real navigation experiments

5.3 Experimental setup

in environments with pedestrians, showing that the proposal enhances the robustness of pedestrian detection and tracking while improving the safety of navigation. The experiments carried out have not been developed in crowded settings yet (left as future work), since a reduced number of detected pedestrians is necessary to properly control and analyze the occurrence of abnormal situations. This facilitates the evaluation of the mentioned capabilities of the architecture, which is, precisely, the main aim of the experiments. Finally, the computation time of inference has been reduced by using feedforward neural networks as explained before.

5.3 Experimental setup

All the validation tasks and experiments described in this chapter have been conceived for a concrete mobile platform, the CRUMB robot [108], which is the one that has been used in all the works related to the thesis. This robot is based on a version of the *Turtlebot-2* that uses a two-wheeled *Kobuki* platform [109]. This platform is equipped with three bumpers placed in the robot front, two magnetic encoders (one for each wheel), three cliff detectors, a tree-axis gyroscope and two wheel drop sensors (one for each wheel). The sensory apparatus has been complemented with two range sensors relying on infrared radiation, namely, a *Hokuyo URG-04-LX* 2-D laser [101] and a *Kinect V1* RGB-D camera [102], [103], whose main features were already included in the study of chapter 3 (table 3.1). The robot is controlled via an on-board netbook PC with an Intel Celeron N2840 at 2.16 GHz and 2 GB DDR3 that runs Ubuntu 14.04 with ROS [110]. A picture of this robot is shown in figure 5.2.

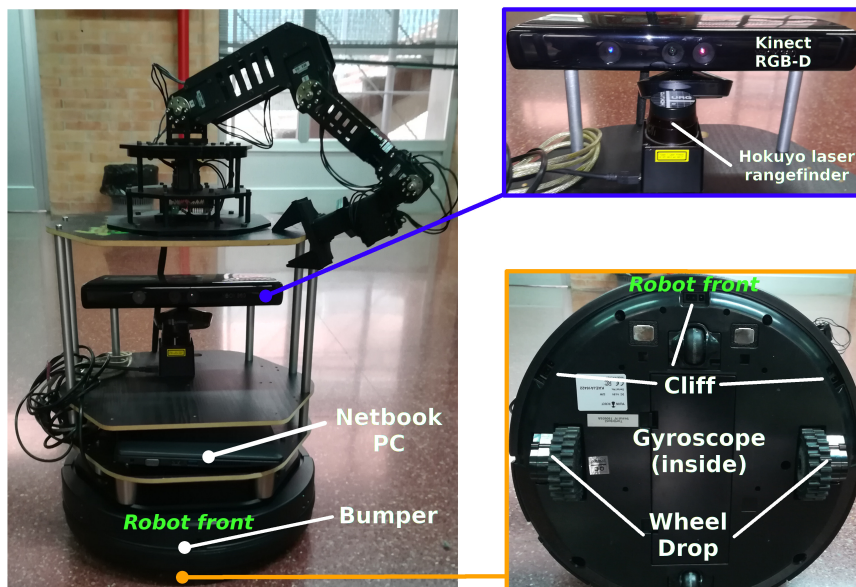


Figure 5.2: Front view of the CRUMB mobile robot, used in this thesis, with its netbook PC and sensors.

The set of physical sensory devices of this robot is the one that has been used for the experimental validation of the proposed inference algorithm (section 5.4). As mentioned before, such validation consists on a set of simulated and real experiments with this platform, all of them relying on the raw, low-level data produced by these sensors (figure 5.2). On the one hand, the real experiments are carried out in a mixed indoor-outdoor scenario where the robot is faced to a variety of sensory abnormal situations (the scene will be showed in section 5.4). On the other

5.4 Evaluation of the inference algorithm

hand, the simulations are based on synthetic data obtained from models of the mentioned sensors incorporating abnormal behavior, implemented in MATLAB (see section 5.4 for more details). For all the experiments, the proposed Bayesian sensory architecture has been implemented in MATLAB by using the *Bayes Net Toolbox* (BNT) from [107].

Concerning the use of this platform for the problem of navigation in human environments, the sensory system considered is more elaborate. Instead of using low-level information directly, it is pre-processed in this case for people detection and tracking. More specifically, range data is employed by an external software, which is based on a ROS package publicly available, developed in the context of the EU FP7 research project known as *Spencer* [158] (from now on, this software will be referred to as the *Spencer system*). This package includes, among other features, a re-implementation of [159] based on an Ada-Boost for people tracking with 2D laser range data and a Nearest-Neighbor Standard Filter (NNSF) for taking the data association decisions required for people tracking. These techniques exhibit a good performance, and do not require a significant computational cost. For a comprehensive guide about this package, please refer to [160].

The real experiments for navigation in human environments have been carried out with the CRUMB robot in a scenario with a pedestrian (see section 5.5). The robot navigates autonomously in that environment thanks to the use of the Model Predictive Control (MPC) approach from [26], which incorporates a unicycle model of the robot to generate collision-free trajectories with safety guarantees. Also, the core of the proposed Bayesian sensory architecture has been implemented in MATLAB by using the BNT. In the experiment, the MPC controller, the Spencer system and the sensory architecture are coordinated within the ROS framework, which is run on board the robot and also on an external, remote PC with an Intel i7-9700K at 3.6 GHz and 32 GB DDR4. Such external hardware is employed to avoid the execution of several nodes with potentially high computational requirements (e.g., the ones related to the MPC controller or the sensory architecture) in a single machine, which could reduce the system performance in certain situations.

Regarding the simulations, a similar setup has been employed. In this case, the CRUMB robot is integrated as part of a simulated environment based on Gazebo [161] (see figure 5.3(a)). The rest of the software used for the experiments is the same as in the real setup; however, an extra module is added for the simulation of human presence and motion. This is also a ROS package, known as *Pedestrian Simulator* or simply *Pedsim*, which enables 2D pedestrian simulation and visualization in real time (see figure 5.3). This package is based on the social model force of [162], and it was also developed during the Spencer project [158]. Only the external PC mentioned before has been used as the hardware for the simulated experiments.

5.4 Evaluation of the inference algorithm

In this section, the performance of the inference algorithm proposed in chapter 4 is assessed. For that, a Bayesian architecture model is defined in subsection 5.4.1, which aims to represent the sensory system constituted by the low-level sensory devices available on board the CRUMB robot, reported in section 5.3. The inference method is then assessed regarding two different aspects. On the one hand, subsection 5.4.2 covers the simulated experiments carried out for the evaluation of error and computation time. On the other hand, subsection 5.4.3 presents real experiments that aim to show the correctness and usefulness of the proposed approach regarding anomaly detection and recovery for the mentioned low-level sensory devices.

5.4 Evaluation of the inference algorithm

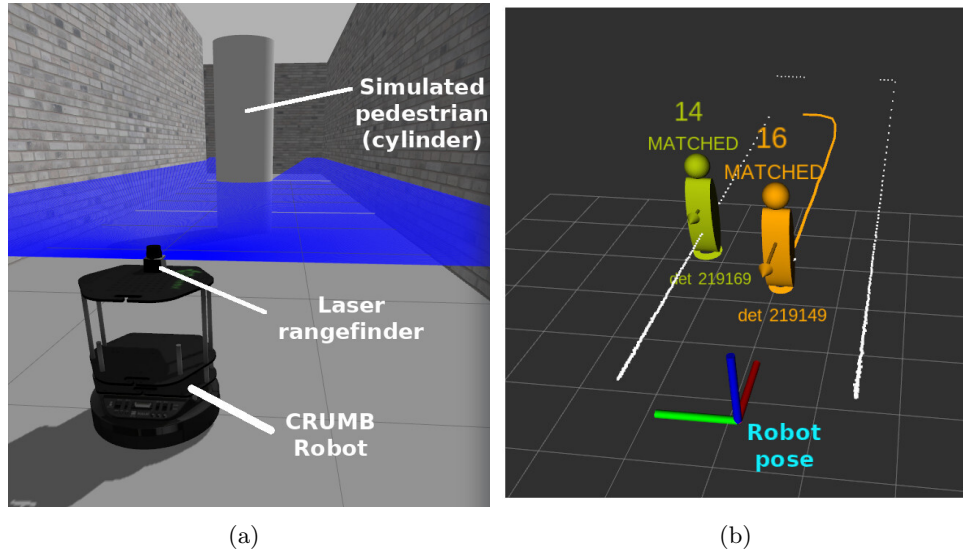


Figure 5.3: Simulated environment for the problem of navigation in human environments. (a) View of the environment in Gazebo, with the CRUMB robot and a simulated pedestrian controlled by *Pedsim*. The blue area corresponds to the field of view of the laser rangefinder. (b) Three-dimensional view of the pedestrians detected by the *Spencer* system. The white lines correspond to the walls of the scene detected by the laser rangefinder. The pedestrian on the right corresponds to the one in figure 5.3(a), while the pedestrian on the left is fictitious (see section 5.5).

5.4.1 Definition of the Bayesian architecture

The Bayesian sensory architecture for this case is defined by following the procedure described in section 4.3.2. Only the key parts of the modeling process are covered here for the sake of brevity. Firstly, it is necessary to identify the different Bayesian sensors to be used in the model and the corresponding sensory information sources (table 5.1). Note that some Bayesian sensors represent elaborate information that can be obtained from raw sensory data by applying some calculations (e.g., the platform linear speed can be obtained from the encoder data by using the direct kinematics model). Once the set of sensors is defined, the dependencies among them can be established, as explained before. A list of dependencies for this model is in table 5.2.

Bayesian sensor	Sensory information source(s)
Cliff detection	Cliff sensor, knowledge about the presence of cliffs in the scene
Wheel drop detection	Wheel drop sensors
Collision detection	Bumper sensors
Platform linear speed	Encoders (calculated)
Platform angular speed (yaw axis)	Gyroscope (z-axis), encoders (calculated)
Distance to obstacles	RGB-D camera, laser, building materials, date, time, lighting

Table 5.1: Bayesian sensors used in this model and their corresponding sources of data.

In this case, all the Bayesian sensors are defined over discrete random variables, thus, the support of the continuous variables involved in the model must be discretized before the definition itself. The reason for this decision of design relies on the fact that the proposed inference algorithm achieves the worst computational cost in the presence of discrete variables, as explained in chapter 4. The sensory system used here represents an extreme case in this sense, since all the variables are discrete. The aim is to assess the reduction of computational cost achieved by the proposed method

5.4 Evaluation of the inference algorithm

Dependencies among sensors
Wheel Drop \leftrightarrow Cliff
Wheel Drop \rightarrow Platform linear speed
Wheel Drop \rightarrow Platform angular speed
Wheel Drop \rightarrow Collision
Collision \leftrightarrow Platform linear speed
Collision \leftrightarrow Platform angular speed
Collision \rightarrow Distance to obstacles
Platform linear speed \rightarrow Cliff
Platform linear speed \rightarrow Platform angular speed
Platform linear speed \rightarrow Distance to obstacles

Table 5.2: List of dependencies among Bayesian sensors. The symbol (\rightarrow) describes a non-cyclic dependency, while the symbol (\leftrightarrow) defines a cyclic dependency.

in such challenging conditions, as well as the errors it produces w.r.t exact inference in monolithic models. Note that there is no need to consider more than one level of cognitive abstraction so far, since all the sensory devices produce low-level information. Also, the possible sensory dynamics are not considered yet.

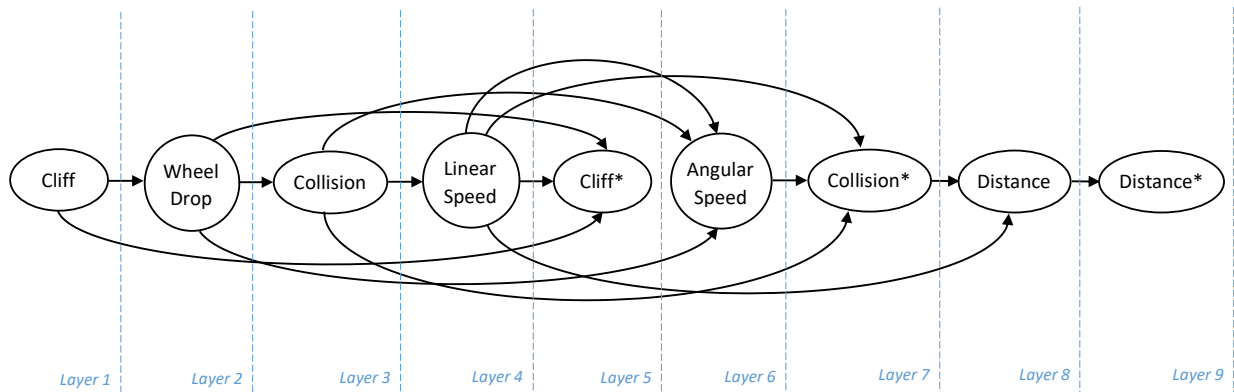


Figure 5.4: Dependency graph for the proposed approximate network, with layer assignment. Asterisks indicate those nodes that are second instances of previous ones, produced due to cycle breaking. Here, the graph is not annotated for simplicity, since there are no different cognitive levels or sensory dynamics that need to be represented.

Once the set of dependencies is available, the next step is to reformulate such relations in terms of non-cyclic dependencies. After that, it is possible to build a dependency graph and then the complete approximate model. The resulting dependency graph for the proposed network is shown in figure 5.4. In the following, a complete description of the models that are part of the Bayesian sensory architecture is provided. Recall, again, that all the random variables are discrete in this case, and thus, all the CPDs are tabular. For the sake of brevity, only some key CPDs will be shown.

The definition of the *zero* layer only contains ideal nodes as parents of their corresponding real nodes; this layer also contains some auxiliary Bayesian sensors that are useful for others but are not relevant in the global model (they will be defined as they appear).

Layer one is entirely dedicated to the lowest instance of the cliff sensor (see figure 5.5). This layer incorporates environmental sources of knowledge and two auxiliary sensors, the robot roll

5.4 Evaluation of the inference algorithm

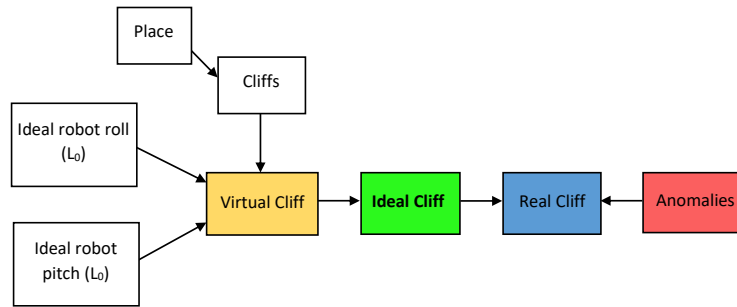
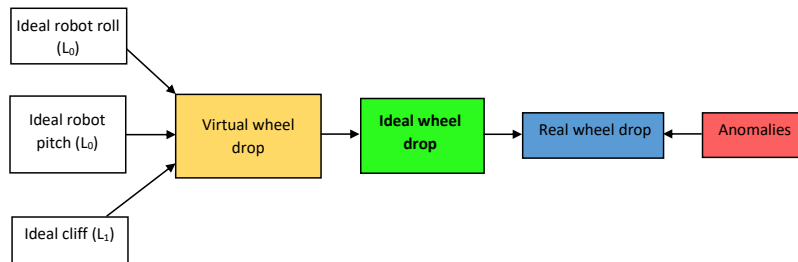


Figure 5.5: Bayesian network corresponding to cliff detection (in layer one). Ideal nodes coming from lower layers are represented in white. Both the real and “place” nodes are evidences. Recall that squared nodes represent discrete random variables.

and pitch angles (calculated using the gyroscope) for the virtual subnetwork. These data are also combined with information about the scene: in a given place, the probability of finding a cliff can be estimated (e.g., when it is known that there are stairs in the environment), since the “place” node is part of the evidence set.



(a)

Anomalies	Real					
	Ideal		Both	Left	Right	None
True	Both		0.10	0.30	0.30	0.30
True	Left		0.30	0.10	0.30	0.30
True	Right		0.30	0.30	0.10	0.30
True	None		0.30	0.30	0.30	0.10
False	Both		0.91	0.03	0.03	0.03
False	Left		0.03	0.91	0.03	0.03
False	Right		0.03	0.03	0.91	0.03
False	None		0.03	0.03	0.03	0.91

(b)

Figure 5.6: (a) Bayesian network corresponding to wheel drop detection (layer two). Again, ideal nodes coming from lower layers are specified. Here, the real node is the only evidence node. (b) Tabular CPD for the real wheel drop node. Here, the entries specify which of the two wheel drop sensors are enabled and whether there are anomalies present or not.

In layer 2 (figure 5.6) the previous cliff sensor and the roll/pitch angles are used in the virtual subnetwork to emulate the behavior of the wheel drop sensor, since the corresponding physical devices are placed in the lowest part of the platform (cliff detectors are situated pointing towards the ground).

5.4 Evaluation of the inference algorithm

Layer 3 (figure 5.7) is devoted to collision detection. In this case, the bumper is emulated by using five new auxiliary Bayesian sensors calculated from raw sensory data coming from the encoders and the laser rangefinder in order to help deducing a collision event. On the one hand, the so-called speed reduction sensors represent the difference between the commanded and actual speed for each wheel (calculated from the encoders). The greater this reduction is, the greater the probability of collision will be. This is valid as long as the robot is in contact with the ground, which is deduced by combining the wheel drop information with the robot angles. On the other hand, the raw distances obtained from the laser rangefinder are used to detect whether there is an obstacle near (or not); this is done by taking the median distance for three different regions (center, left and right), which correspond to the real location of the bumpers.

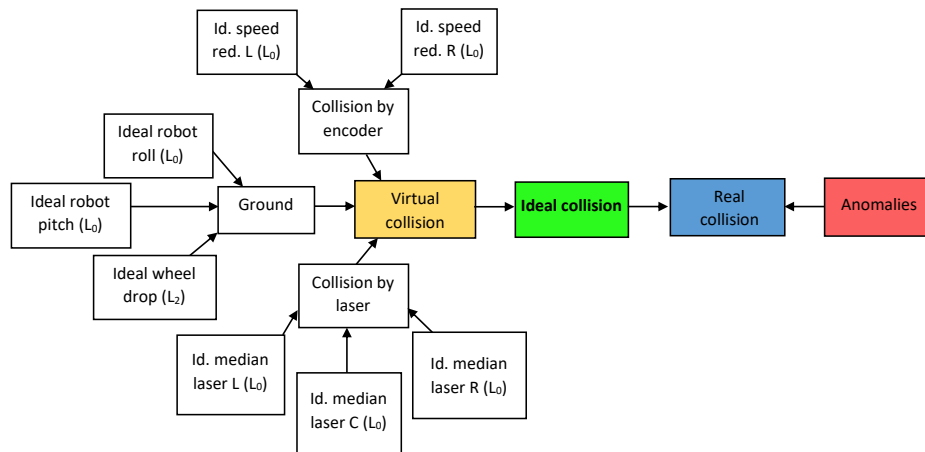
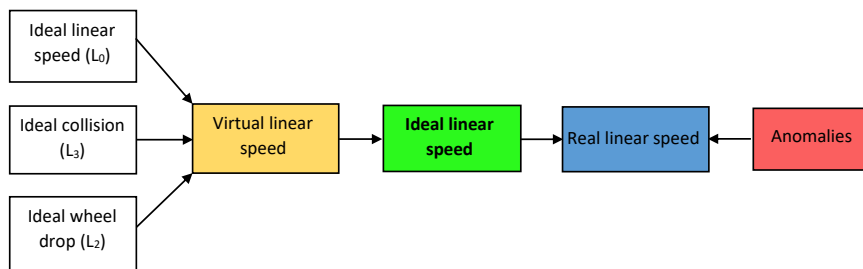


Figure 5.7: Bayesian network corresponding to collision detection (layer three). Here, the real node is the only evidence node.



(a)

Virtual \ Ideal	Ideal				
	High+	Low+	Zero	Low-	High-
High+	0.9900	0.0025	0.0025	0.0025	0.0025
Low+	0.0025	0.9900	0.0025	0.0025	0.0025
Zero	0.0025	0.0025	0.9900	0.0025	0.0025
Low-	0.0025	0.0025	0.0025	0.9900	0.0025
High-	0.0025	0.0025	0.0025	0.0025	0.9900

(b)

Figure 5.8: (a) Bayesian network corresponding to platform linear speed (layer four). Here, the real node is the only evidence node. (b) CPD for the ideal linear speed node. Here, the entries specify the magnitude and the sense of the speed (+ stands for forwards and - for backwards).

5.4 Evaluation of the inference algorithm

Layer 4 has been designed to get a more robust estimation of the platform linear speed (figure 5.8). In this layer, the information about linear speed is re-used from the *zero* layer and combined with collision and wheel drop detection, since these events may affect speed by reducing it or even changing its direction.

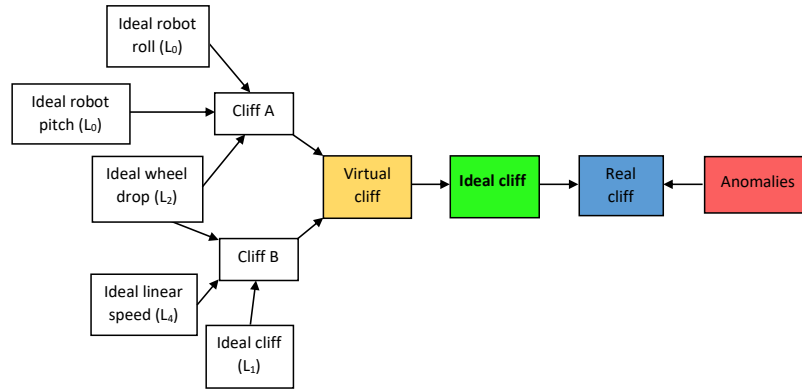


Figure 5.9: Bayesian network corresponding to cliff detection (layer five). Here, the real node is the only evidence node.

Layer 5 (figure 5.9) corresponds to the highest instance of the cliff sensor. In this layer, the information used to emulate the cliff is completed by adding wheel drop detection, linear speed and the previous estimation about the cliff itself, from layer one. In this case, the wheel drop sensor is combined with the robot angles to represent the cliff behavior. Secondly, the belief in the previous cliff estimation should be coherent with the wheel drop detection and the platform linear speed.

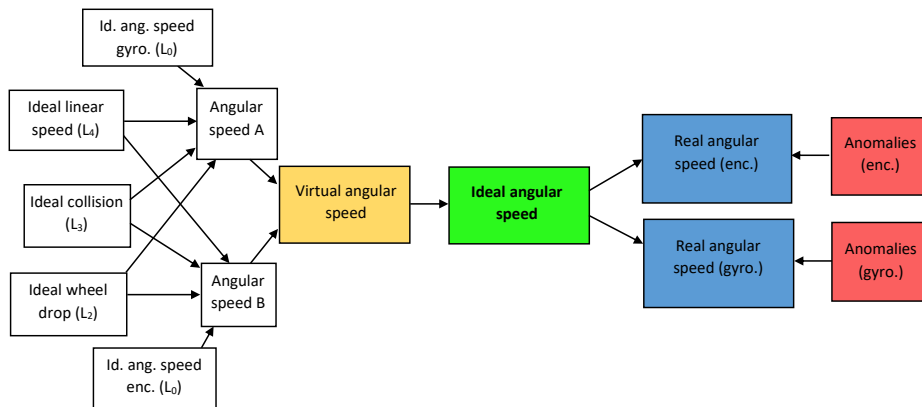


Figure 5.10: Bayesian network corresponding to platform angular speed (layer six). Here, the real nodes are the only evidence nodes.

In layer 6 (figure 5.10), the platform angular speed is estimated in two different ways, one for each physical sensor providing this information (encoders and gyroscope). Both estimations are obtained taking into account previous collision and wheel drop information, as these events may potentially modify the speed. The network design checks data coherency for every possible situation.

Layer 7 (figure 5.11) simply adds the remaining dependencies used for robust collision detection. This part of the model refines the estimation of the bumper state by imposing data coherency

5.4 Evaluation of the inference algorithm

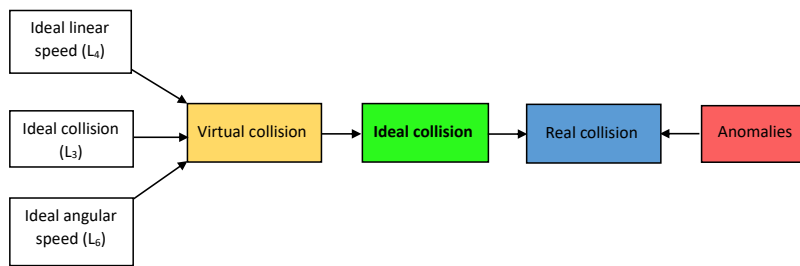
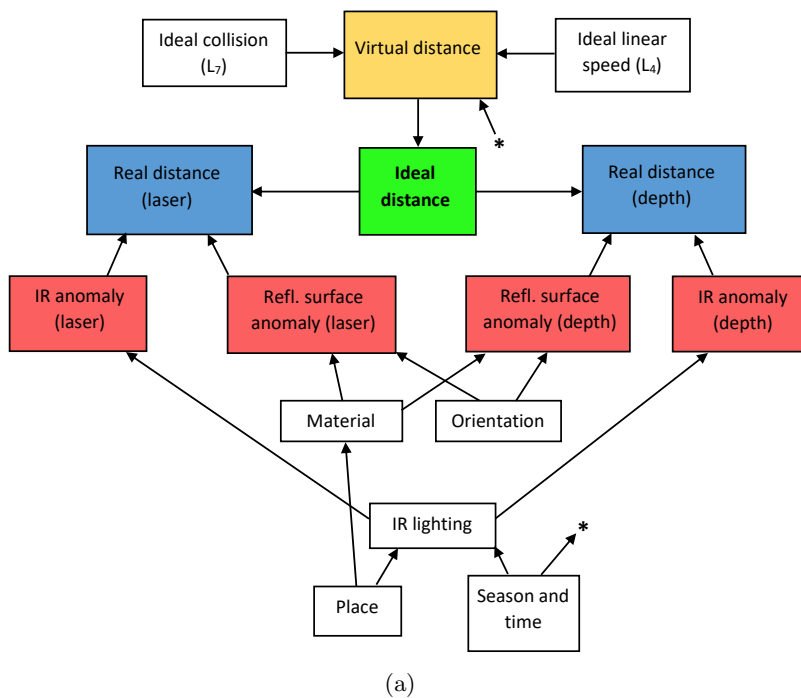


Figure 5.11: Bayesian network corresponding to collision detection (layer seven). Here, the real node is the only evidence node.



(a)

Orientation	Anomaly	
	Material	
Perpendicular	Specular	0.60 0.40
Perpendicular	Medium	0.30 0.70
Perpendicular	Diffuse	0.05 0.95
Crosswise	Specular	0.75 0.25
Crosswise	Medium	0.40 0.60
Crosswise	Diffuse	0.05 0.95

(b)

Figure 5.12: (a) Bayesian network corresponding to one of the beams of the distance sensor (layer eight). The asterisk indicates that the “season” node is connected to the virtual one. Here, the “place” and “season” nodes are evidences, in addition to the real ones. (b) CPD for the reflective surface anomaly node for the depth sensor. Here, the entries specify the orientation of the robot w.r.t the plane of the surface it is pointing to as well as the reflectivity of the material of such surface.

as done in the previous layers. Note that anomalies reasoning only involves a single node (as in the previous layers). This is because the information that enables to predict these anomalies is not available, thus they are deduced from the sensor behavior instead.

5.4 Evaluation of the inference algorithm

One of the most complex parts of the proposed network corresponds to layer 8, devoted to the distance sensor. In this case (figure 5.12), the model is more complex due to the nature of the physical sensors involved (RGB-D camera and laser rangefinder). The virtual subnetwork uses collision information, linear speed and the date and time to estimate the distance to possible obstacles in the scene. As an example, if the robot collides while it is navigating, the belief in a short distance will be increased. Also, if the robot moves according to a reactive paradigm, it will decrease its speed in case nearby obstacles have been previously detected, therefore leading to short distances. Lastly, date and time can be used to predict the amount of mobile obstacles present in the scene by using knowledge about the environment (e.g, people, other robots, etc.).

In this Bayesian sensor, the anomalies subnetwork is also more complex. The two physical sensors involved in this layer are affected by mainly two causes of failure, both related to the measurement principle these sensors use, based on infrared radiation. These anomalies may be provoked by external infrared radiation (e.g., direct sunlight) and also by the presence of reflective surfaces in the scene (e.g., shiny materials, mirrors, etc). When exposed to any of these conditions, the physical sensors simply lose the information of the environment; in other words, they behave as if there were no objects around them. These situations can be detected by considering the robot location in the scene. With a precise knowledge of the environment, it would be possible even to know about the nature of the material the robot is pointing to, as well as its orientation (which is also related to the considered anomaly). Recall that place, time and date are part of the evidence set.

This Bayesian sensor is also different from the others in the sense that it represents vectorial data, in this case a vector of distances gathered at different scan angles. This is treated by simply considering each single measurement as a Bayesian sensor, thus replicating the structure in figure 5.12 as many times as needed to represent all the data (basically, all the nodes should be replicated, except the ones that are related to environmental information and to other layers). In this chapter, each one of these measurements will be referred to as a *beam*.

Finally, an extra layer (figure 5.13) has been added to consider the previous distances in small groups according to their spatial distribution (groups of three beams are used in this case). When the different beams are close enough to each other, the estimated distances can be made more robust assuming they should follow certain patterns (e.g. the three distances should be very similar if the robot is pointing to a wall, while they should change abruptly if it were close to an edge, etc.).

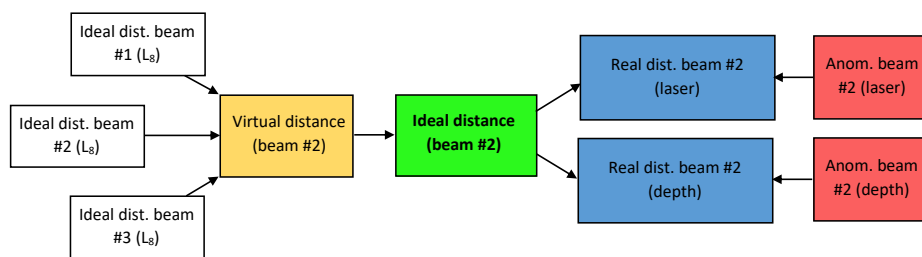


Figure 5.13: Bayesian network corresponding to the beam number two of the distance sensor (layer nine). The remaining beams are obtained similarly. Here, the real nodes are the only evidence nodes.

5.4.2 Simulated experiments

The proposed approximate inference algorithm has been statistically assessed using the model described in subsection 5.4.1 through different simulations, in terms of both its computational cost (execution time) and its error w.r.t exact inference methods (the *jointree* algorithm has also been used as reference). The proposal has also been compared to other approximate inference algorithms: the *loopy belief propagation* (LBP) [163] and the *Gibbs sampling* MCMC method [65] as an example of stochastic sampling algorithm. The former is considered a degenerate case of the *edge deletion belief propagation* in which all network edges are deleted [58]; this is relevant here because that method is in some sense similar to the one proposed. The main difference is that the latter is not iterative (and therefore, it is not *any-time*) and it may also perform inference with a reduced set of evidences.

The approximate network introduced in subsection 5.4.1 as well as its corresponding monolithic version have been implemented in MATLAB by using the *Bayes Net Toolbox* (BNT) [107]. This library provides different inference methods, including the exact *jointree* and the approximate LBP and *Gibbs sampling* algorithms. In this section, all the simulations are carried out by performing inference in these Bayesian networks, executed on a desktop PC with an Intel i7-7700K at 4.20 GHz and 32 GB DDR4.

Regarding errors, there are several methods to measure the difference between two probability distributions. The *Hellinger's distance* [164] is the one used here, as it can handle zero probabilities [165], which are common in the Bayesian networks defined; another advantage is that it is normalized, providing a number between 0 and 1, where 0 means that the two distributions are exactly the same and 1 that they are totally different. The Hellinger's distance H for probability distributions over discrete random variables is defined as follows. Let $P_1(x_i)$ and $P_2(x_i)$ be two discrete probability distributions defined over the same support $x = \{x_1, x_2, \dots, x_n\}$; then,

$$H(P_1, P_2) = \sqrt{\frac{\sum_{i=1}^n \left(\sqrt{P_1(x_i)} - \sqrt{P_2(x_i)} \right)^2}{2}} \quad (5.1)$$

where $0 \leq H(P_1, P_2) \leq 1$.

Experiments for error evaluation

The first test carried out compares the different errors made by the approximate algorithms w.r.t exact inferences. The LBP and *Gibbs* methods have been adjusted such that their execution times are as similar as possible to the ones of the proposed inference method; i.e., the aim is to compare errors when the execution time is bounded by the performance of the proposed algorithm. In this first simulation, all the queries are of the form expressed in equation (4.3), the one for ideal nodes (i.e., the robot requires to know the ideal datum of the sensors in spite of anomalies), and all the layers in the approximate network are considered.

In order to fully analyze the error, all the possible instantiations of the evidence set should be explored¹. However, this comprehensive search is not possible in terms of time, since the considered model has too many evidence nodes. This has been addressed by selecting a representative

¹Note that these instantiations represent many different situations and that some of them may be impossible or very unlikely in reality.

5.4 Evaluation of the inference algorithm

subset of eight of these evidences and three different values for the support of each one, leading to $3^8 = 6561$ tests (note that this search is still comprehensive in the mentioned subset). In addition, several levels of discretization (i.e., variable support sizes) have been considered for some variables, since there are continuous sensors in the system whose measurement ranges must be discretized that way. The test first begins with a low number of discrete states and then they are increased. Three different support sizes are used, thus the probability distribution under study is calculated $3 \cdot 6561 = 19683$ times. Recall that this is done once per each layer in the model; then the Hellinger's distance is used to compare the errors made by the three approximate inference algorithms w.r.t the distributions obtained by the exact *jointree* method. Both the approximate and monolithic networks are built with three beams for the distance sensor.

These parameters are shown in table 5.3 (the LBP method is omitted as its maximum number of iterations is always one). Those groups of variables whose support is changed are compiled in table 5.4. The results for this test are depicted in figure 5.14, where it is shown that the proposed method has the lowest median distance for inference in all the layers.

Layer	Gibbs samples	Variable evidences (real nodes)
0	1	Angular speed (gyro.), roll angle, left speed reduction, left laser
1	5	Cliff
2	25	Wheel Drop
3	100	Bumper
4	150	-
5	350	Cliff
6	350	Angular speed (gyro.)
7	450	Bumper
8	1200	RGB-D camera
9	1200	RGB-D camera

Table 5.3: Configuration parameters for the *any-time* algorithms and variable evidence nodes, per layer. Repeated names represent different real nodes with the same evidence.

Groups of variables	Support size (first, second, third)
Distance to obstacles	9, 16, 22
Angular speed	5, 18, 30
Robot angle	5, 18, 30
Linear speed	5, 18, 30
Speed reduction	3, 16, 28

Table 5.4: Groups of random variables whose support is changed during the first simulation.

It follows from this first study that the *any-time* algorithms would need more computation time in general than the proposed method to achieve a similar approximation quality. This is specially noticeable for the Gibbs sampling, which has the highest median distance for virtually all the layers, but even the LBP algorithm still produces rather inaccurate results, as its best median distance is nearly 0.4. Although the proposed method has been proven to generally produce the best approximation under limited computation time, results may also be somewhat inaccurate under certain conditions (consider, for example, the obtained distances for layers 2 and 3). All in all, the approximations provided by this algorithm do not distort the original probability distributions, thus keeping essential information for sensory diagnosis and recovery.

5.4 Evaluation of the inference algorithm

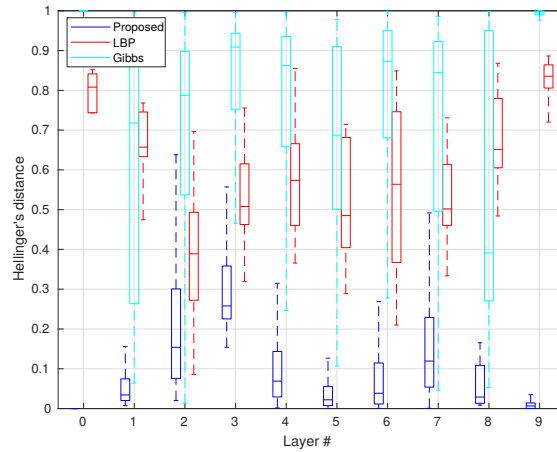


Figure 5.14: Hellinger's distance for the implemented approximate inference algorithms w.r.t exact inference results in the first simulation. This boxplot represents 19683 results per layer.

Experiments for computation time evaluation

The aim of the second simulation is to show the computational cost of each approximate algorithm. In this case, the any-time methods have been manually adjusted so that their error gets similar to the obtained by the proposed algorithm. As in the previous simulation, probability distributions for both ideal and anomaly nodes are calculated for each layer. However, the evidence set is this time fixed for all the tests (by using a representative instantiation) and 26 different support sizes have been used for continuous sensors (note that the cost of inference does not depend on the concrete values of the evidence, but on the cardinality of the network variables). Every inference is repeated 50 times (except for the Gibbs method, as explained later). Parameters for this experiment are shown in tables 5.5 and 5.6 (the network has been built with three distance beams).

Level	LBP iterations	Gibbs samples
0	2	8000
1	4	4000
2	4	4000
3	4	7000
4	6	7000
5	7	7000
6	7	7000
7	6	8500
8	3	8000
9	7	10000

Table 5.5: Configuration parameters for the *any-time* algorithms, per layer, for the computation time tests.

Groups of variables	Support size (from first to last)
Distance to obstacles	10 to 22
Angular speed	5 to 30
Robot angle	5 to 30
Linear speed	5 to 30
Speed reduction	3 to 28

Table 5.6: Groups of random variables whose support size is changed 26 times during the computation time tests.

5.4 Evaluation of the inference algorithm

Gibbs sampling needs a special treatment here. Since it is a stochastic method, the resulting distributions may differ even when the same evidence set is used. That is the reason why they are calculated only once for each test. Moreover, this algorithm is significantly more inefficient compared to the others. This is shown in the example of figure 5.15 (layer 1). Consequently, it is omitted from the results in figure 5.16, in order for the axis limits to focus on the interesting conclusions. Some layers have also been omitted, since the execution times follow similar patterns.

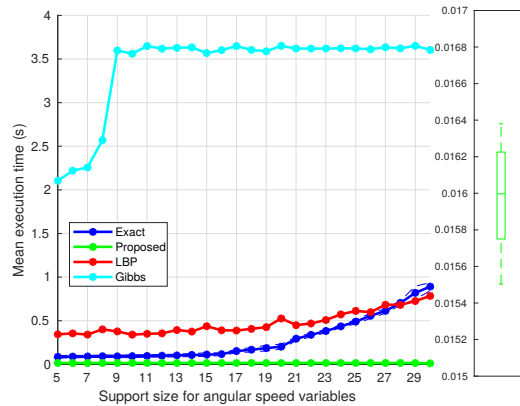


Figure 5.15: Mean execution times for all the inference algorithms used in the experiments. The boxplot summarizes these times only for the proposed method. The query variable is the ideal node belonging to layer 1.

The obtained results show that the proposed inference algorithm is faster than the others in general. One of its advantages is that the computation time is reduced even more when the inference layer is bounded. As explained in chapter 4, this algorithm ends when the layer containing the variables of interest is reached, and this is the case for the results shown in figures 5.16(a), (b) and (c), where the proposed method is always the most advantageous. Regarding the last layer, this method is slightly less efficient than the exact one for the smallest support sizes. However, note that only the support size is changed in this simulation; if the network size was increased as well (e.g., by using more distance beams) the proposed method would be more efficient even for small support sizes, as it will be shown later. On the other hand, the LBP algorithm is generally more efficient than the exact *jointree* when the support size is big enough, but there are some layers where the support size should be increased beyond the tested ones in order to show some advantage (figures 5.16(b) and (d)). In conclusion, the proposed inference algorithm is generally the most advantageous under the specified conditions (bounded error).

5.4.3 Real experiments

In the following, the performance of the proposed approximate network is assessed in a real environment with the CRUMB robot. First, the computation time and the error made by the proposed algorithm are compared to the ones related to the other inference methods considered in this chapter. After that, the capabilities of the proposed approach regarding sensory anomaly detection and recovery are illustrated.

Comparisons in different applications

The performance of the proposed inference method has been evaluated in comparison with that of the other algorithms in a real experiment with the CRUMB robot. As in the previous simulations,

5.4 Evaluation of the inference algorithm

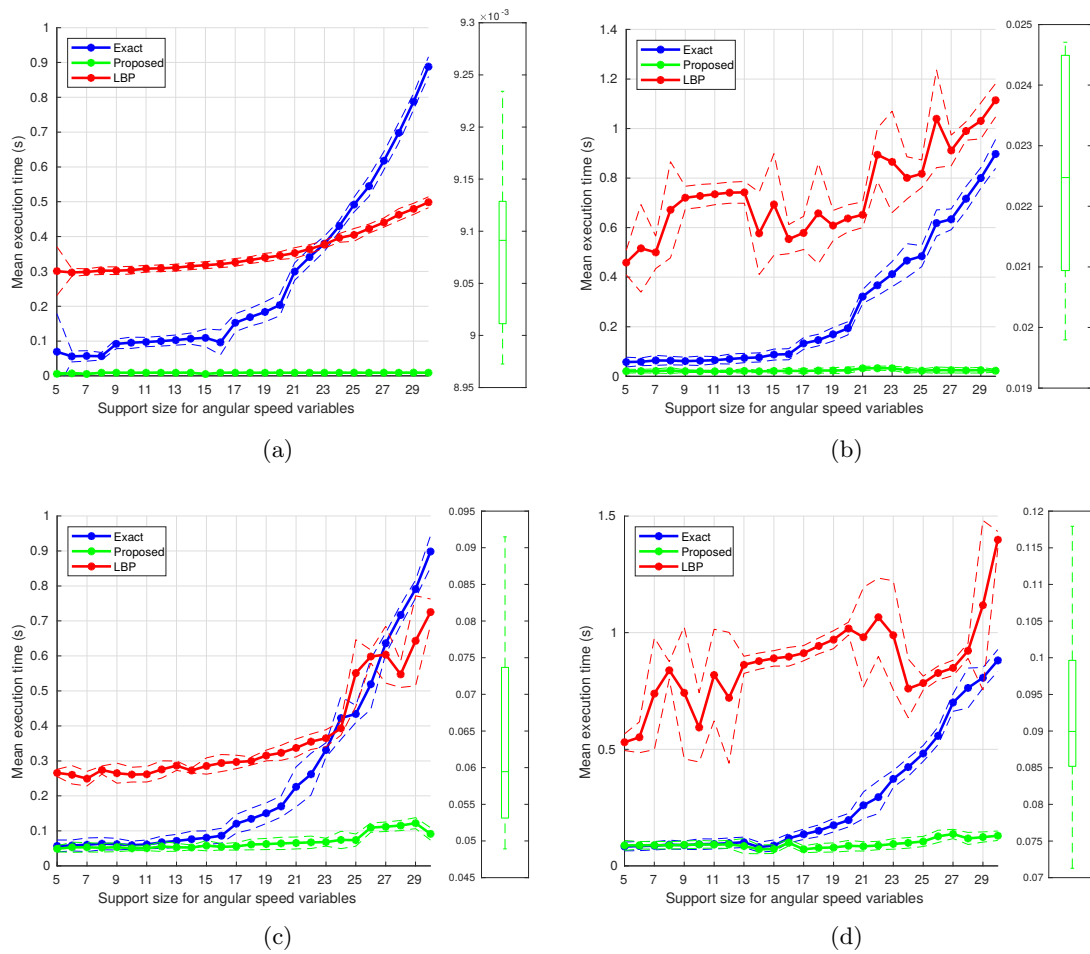


Figure 5.16: Mean execution times for inference in different levels and nodes. Boxplots summarize these times only for the proposed method. Dashed lines correspond to $\pm 2\sigma$, where σ is the standard deviation for each test. (a) Layer 0 (ideal node). (b) Layer 4 (anomaly node). (c) Layer 8 (ideal node). (d) Layer 9 (ideal node).

different support sizes are considered for the variables, but also different network sizes. The aim of this experiment is to analyze the efficiency and error produced by the different inference methods under several configurations of the network that could be integrated in robotic applications.

More specifically, four configurations have been considered, one for each possible robotic application (see table 5.7). The first one is designed to check the proposed sensory model through isolated cognitive reasonings in real time navigation; for that reason, a limited number of distance beams and possible values for each one have been used. The second configuration is devised to robotic SLAM [1]. In this case, the network size and the distance resolution are increased taking into account that the computational cost should not limit the operation of the robot. The third configuration (designed only for map building) reduces the distance resolution to allow a greater number of distance beams, thus leading to a more detailed description of the environment. Finally, the fourth configuration could also be suitable for the previous applications, but the time requirements are relaxed in this case to explore the computational limitations of the proposed model.

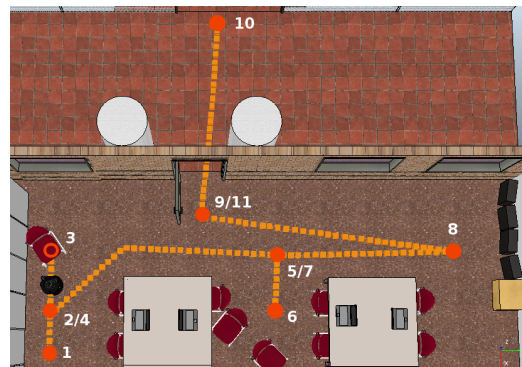
The experiments have been carried out in a mixed indoor-outdoor scenario (see figure 5.17). The CRUMB robot follows a predefined path along which there are obstacles or areas that are diffi-

5.4 Evaluation of the inference algorithm

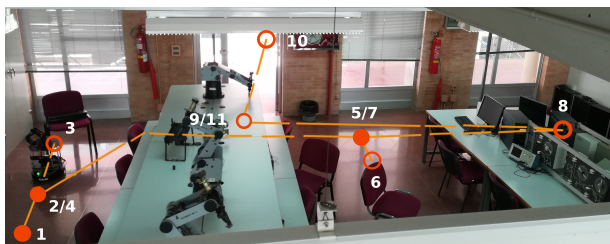
Configuration	Distance beams	Support size
1: <i>reasoning</i>	10	5
2: <i>SLAM</i>	20	31
3: <i>mapping</i>	50	21
4: <i>multipurpose</i>	60	31

Table 5.7: Parameters for the networks used in the different configurations (with names). The specified support size is related to the distance variables (the one used for the remaining continuous sensors is the lowest appearing in table 5.6). Recall that the size of the network depends on the number of distance beams used.

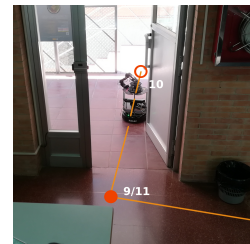
cult to perceive due to the nature of its sensors, thus leading to wrong or incomplete measurements. All these sensory data have been sampled and stored, which amounts to 1048 sample points. Then, these data are used as evidences for the four network configurations described above. Since some sensors are affected by anomalies, their ideal state is inferred as well as their anomalies for each point of the dataset, i.e., 1048 times. These sensors are the bumper, cliff and wheel drop detectors, the gyroscope and the ranging sensors (table 5.8 contains the ideal and anomalies nodes inferred during the experiment). A more detailed description of the detected anomalies is presented later on.



(a)



(b)



(c)

Figure 5.17: Path followed by the robot during the real experiments, with waypoints (from 1 to 11). A video is available at http://babel.isa.uma.es/_fordownloading/thesisMCQ_videoInference.mp4. (a) Virtual 3D representation of the whole environment. (b) Indoor view of the real scene. (c) Outdoor view.

The inference methods have been used as follows. Firstly, the LBP algorithm has been adjusted to achieve the best tradeoff between accuracy and efficiency, taking into account that numerical underflow must be avoided (since there is a maximum number of iterations beyond which the resulting distributions do not sum to one). This leads to 5 iterations as the highest possible value that allows for correct results in the whole network. Secondly, the Gibbs sampling method is omitted here, since it has been proven to be extremely inefficient and inaccurate. The first com-

5.4 Evaluation of the inference algorithm

Query nodes	Layer
Distance to obstacles	9
Anomalies for distance sensors	8
Collision detection	7
Platform angular speed	6
Wheel drop detection	2
Cliff detection	1

Table 5.8: Inferred nodes during the experiment with real data and their corresponding layer in the approximate network model. Query variables always correspond to both ideal and anomalies nodes, except for the first two entries, which represent multiple ideal and multiple anomalies variables, respectively.

parative focuses on the computational cost of each algorithm, where all the ideal and anomalies nodes indicated before have to be inferred for each point of the experiment (figure 5.18).

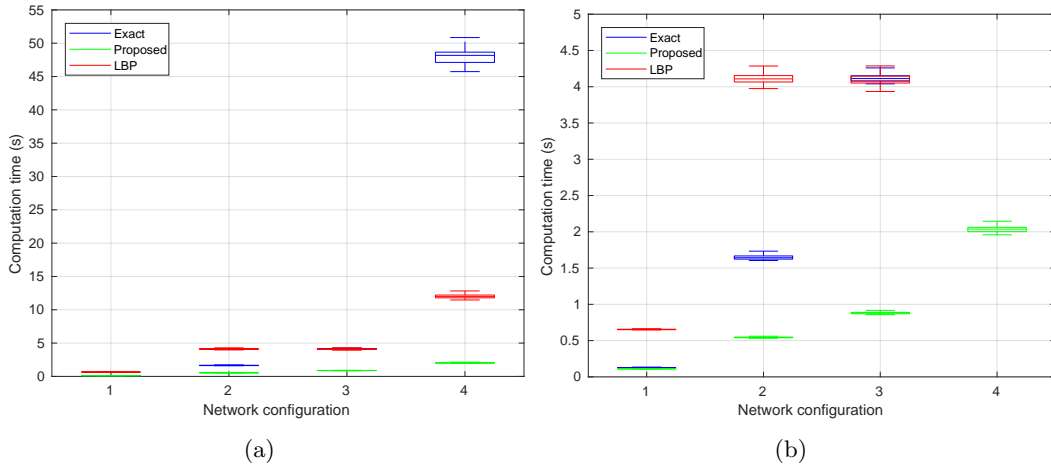


Figure 5.18: Computation time for the exact *jointree*, LBP and proposed inference algorithms. (a) Boxplot representing 1048 inferences for the different network configurations. (b) Zoomed view of the shortest times.

These results show that the proposed algorithm achieves the best efficiency for all the different configurations. However, it is not worth using any approximate algorithm for the *reasoning* configuration since the execution times achieved by the exact method are virtually the lowest ones. Additionally, the LBP algorithm could be considered for the *multipurpose* configuration, although the proposed method is still much faster. In summary, the *reasoning* configuration could be considered suitable for real time applications, while the others are somewhat limiting. However, the use of the proposed method reduces considerably the computational cost, specially for the *multipurpose* configuration, where the other methods show an excessive cost that avoids their application to online navigation.

The analysis performed in this section also includes the error made by the approximate algorithms (except Gibbs) by using the Hellinger's distance w.r.t the exact *jointree* algorithm. The comparative has been done by taking the difference between the H distance obtained by the proposed method and the LBP, i.e., $\Delta H = H_{proposed} - H_{LBP}$ for each obtained distribution, thus, a negative value for ΔH means that the proposed method produces a better approximation. The analysis focuses here on those ideal and anomalies nodes of interest that have been inferred during the experiment, and shows the error comparative taking into account all the network configurations

5.4 Evaluation of the inference algorithm

for each mentioned node. In this case, all the possible distance nodes (beams) have been inferred for each application, thus leading to $1048 \cdot (10+20+50+60) = 146720$ inferences (the same holds for the corresponding anomalies nodes); for each one of the remaining sensors and their anomalies, $1048 \cdot 4 = 4192$ inferences have been obtained. The results of this study are depicted in figures 5.19, 5.20 and 5.21.

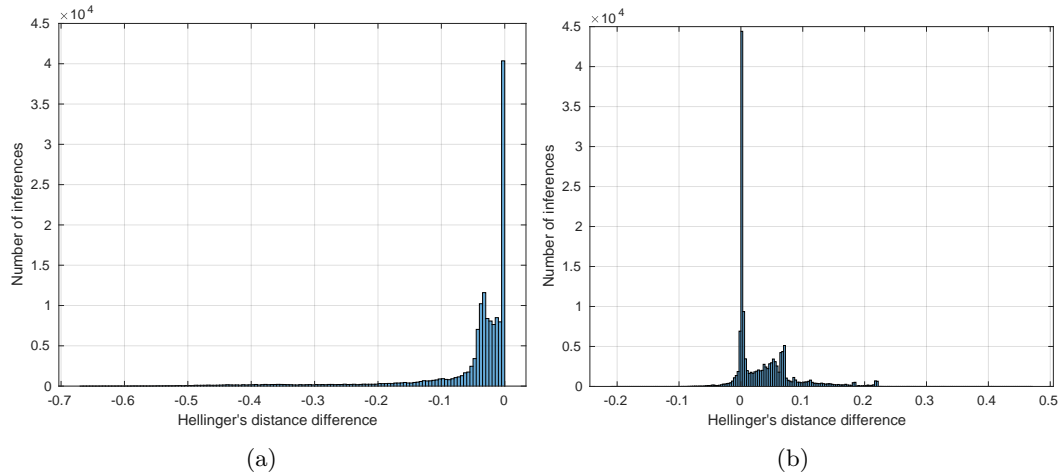


Figure 5.19: Hellinger's distance difference for inference over different nodes (I). All the network configurations have been used. (a) Distance beams. (b) Anomalies for distance beams.

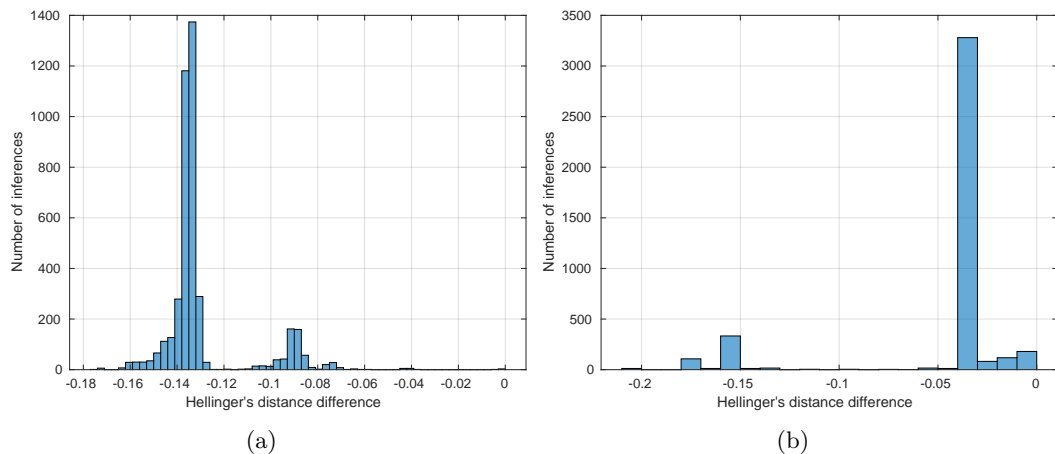


Figure 5.20: Hellinger's distance difference for inference over different nodes (II). (a) Collision detection. (b) Platform angular speed.

It follows from this second study that the proposed inference method produces better quality approximations than the LBP for most of the inferred variables. There are, however, two situations where the LBP algorithm is still more accurate (figures 5.19(b) and 5.21(a)), although the approximations given by the proposed method do not strongly differ from the exact results: the worst case difference is around 0.2 (note that the LBP has been adjusted such that the error for these two cases is extremely low). All in all, the use of the proposed method is justified since it is always more efficient than the LBP in terms of computational cost (figure 5.18).

5.4 Evaluation of the inference algorithm

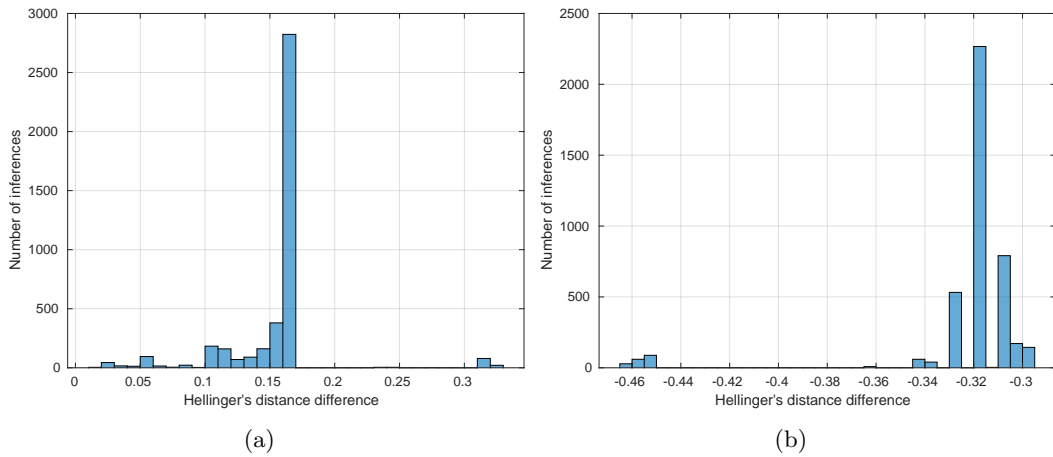


Figure 5.21: Hellinger's distance difference for inference over different nodes (III). (a) Wheel drop detection. (b) Cliff detection.

Anomaly detection and sensor recovery in a real environment

The following real experiment shows the capabilities of the proposed sensory model for anomaly detection and sensor recovery in the mentioned real setting. In this experiment, the robot follows a predefined trajectory while being faced to different adverse situations from the perspective of its sensory apparatus (both the scene and the path followed by the robot are depicted in figure 5.17).

The experiment has been designed in order to illustrate the behavior of some sensors when they are involved in adverse conditions. As mentioned above, the experiment has 1048 points sampled along the trajectory, and most of the physical sensors produce anomalous measurements in some part of it. Table 5.9 summarizes all the affected sensors, the causes that explain their behavior, and the effects on own measurements.

Affected sensor	Causes of anomaly	Effects
Laser rangefinder	Lighting conditions, presence of reflective surfaces	Surface/object not detected
RGB-D camera	"	"
Gyroscope	Electromagnetic interferences	Noisy measurements
Cliff detector	"	False positives
Wheel drop detector	Mechanical issues	"

Table 5.9: Physical sensors affected by adverse conditions in the real experiment. The cause for each anomalous behavior is listed, as well as the effects on measurements.

Despite all the above, the proposed sensory model is not only able to recognize anomalous situations, but also allows the robot to recover suitable measurements, i.e., close to those that should be obtained in a fault-free scenario. In order to illustrate these capabilities, some of the results obtained from the assessment of the proposed model over the sensors in table 5.9 are shown. All the results commented below have been obtained by using the proposed inference algorithm.

As a first example, the laser rangefinder is influenced in this experiment by the presence of reflective surfaces, which prevent the sensor to perceive nearby obstacles (see the results in figure 5.22). The probability of anomalies due to reflective surfaces is represented along the whole path for one of the laser beams on the right. When the robot is situated close to these surfaces (e.g.,

5.4 Evaluation of the inference algorithm

the legs of some chairs, metallic parts of the wall, shiny computer cases, etc.) some measurements are partially lost, i.e., the surfaces are not completely detected. In the highlighted point of the trajectory, the represented beam of laser rangefinder is not able to notice the presence of the nearby computer cases. However, the sensory network enables to identify this anomaly properly, as well as to provide an adequate distance for this obstacle (encoded in the corresponding posterior distribution). Recall that the whole model is defined with discrete variables, so it deals with discretized ranges of distances instead of numerical values (for instance, the state “1” means no obstacle detected, and the remaining correspond to increasing ranges of non-zero distances). In this case, the state “5” has the highest probability, which means that there must be an obstacle in a range of 0.99 to 1.33 meters (as 31 possible ranges of distance have been used). All in all, this recovery is possible as long as there is any set of information sources that allow to deduct it. In this particular situation, the RGB-D camera is able to detect the problematic part of the surface properly.

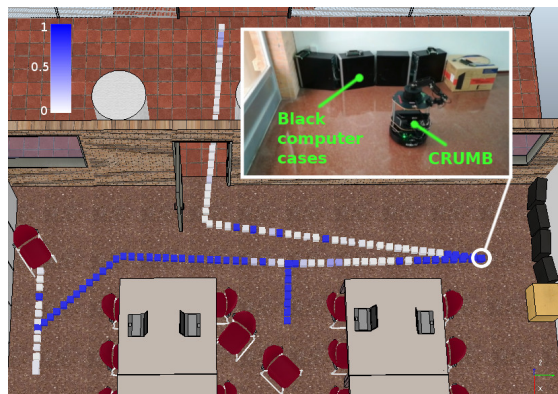


Figure 5.22: Anomaly detection and recovery for the laser rangefinder. Probability of anomalies due to reflective surfaces along the trajectory. A color legend is shown (top left) as well as a view of the real robot in a point of interest (top right).

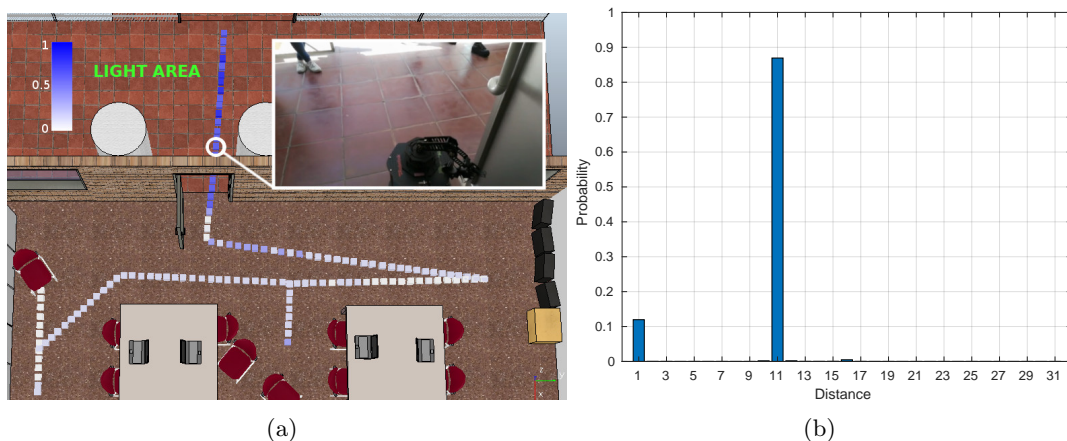


Figure 5.23: Anomaly detection and recovery for the RGB-D camera. (a) Probability of anomalies due to intense IR radiation (along the trajectory), color legend and a view of the real robot in a point of interest. (b) Discrete posterior distribution for the ideal distance node (see figure 5.13), corresponding to the point of interest.

Another sensor that is also affected during the experiment is the RGB-D camera. Lighting conditions are known to be critical for these kind of sensors, as they may be importantly corrupted by

5.4 Evaluation of the inference algorithm

the presence of undesirable infrared radiation. This situation takes place in the outdoor part of our experiment (see results in figure 5.23), where the infrared radiation coming from direct sunlight is intense enough to provoke measurement loss (specially for the central beam). In spite of that, the laser rangefinder is still working nominally under these circumstances, thus allowing the system to provide a correct distance range. This anomaly identification is possible because of the knowledge of the environment and the location of the robot, i.e., it is known that the robot is outdoors and that the test is done in a typical summer morning with high temperatures that may possibly lead to intense IR radiation. In this case, the state named as “11” is assigned the highest probability, so there must be an obstacle in a range of 3.0 to 3.33 meters even when the sensor does not detect it.

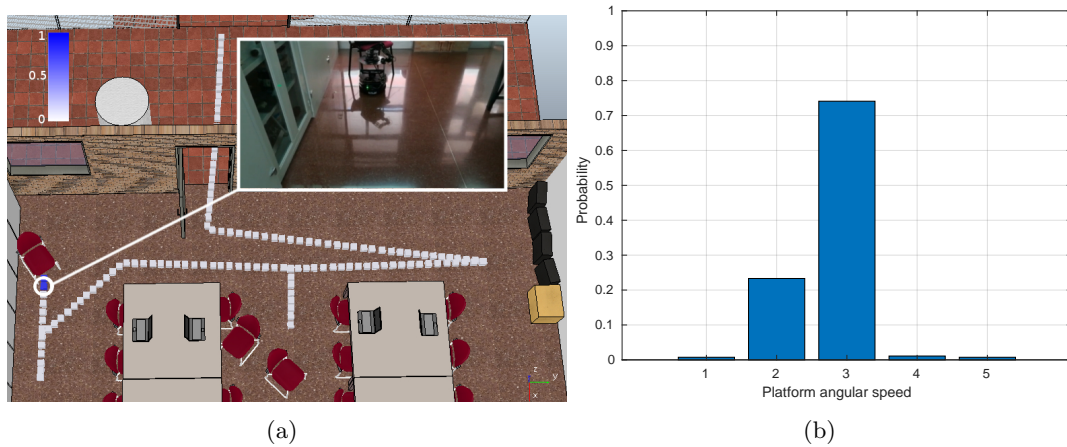


Figure 5.24: Anomaly detection and recovery for the gyroscope sensor. (a) Probability of anomalies (along the trajectory), color legend and real view in a point of interest. (b) Discrete posterior distribution for the platform ideal speed node (see figure 5.10), corresponding to the point of interest. Here, the states evolve linearly from “1” (high clockwise angular speed) to “5” (high counterclockwise speed), where “3” corresponds to very low speed.

The CRUMB robot is also endowed with a 3-axis gyroscope that produces noisy outputs while the robot is navigating around point 3 (see figure 5.17). It is difficult to identify the cause of this behavior, however, it is likely that some electromagnetic interference has affected nearby circuits leading to unstable sensory output. Anyway, this issue is again detected and corrected by the proposed sensory model, as seen in the results shown in figure 5.24. In this part of the experiment, the platform angular speed is very close to zero, as the robot is only moving forwards, however, the noisy data provides a much greater value. Since this is not coherent with the overall state of the sensory system, the probability of anomalies is high. In addition, the posterior distribution for the ideal platform speed in figure 5.24 assigns the highest probability to the state “3”, which means that the speed must be close to zero or very low.

Cliff detectors also show anomalous behavior during the experiment (in this case, they sometimes produce false positives). Again, it is difficult to identify the cause of this anomaly. As the measurement principle of these sensors relies on infrared radiation, they may be possibly affected by the presence of nearby hot surfaces. This seems unlikely, however, since the anomalous behavior does not follow a predictable pattern. Therefore, it is more likely that some external electromagnetic interference has provoked this anomalous response.

The results shown in figure 5.25 prove that the proposed sensory model identifies the previous failure adequately and also recovers the healthy state. This is possible thanks to the available

5.4 Evaluation of the inference algorithm

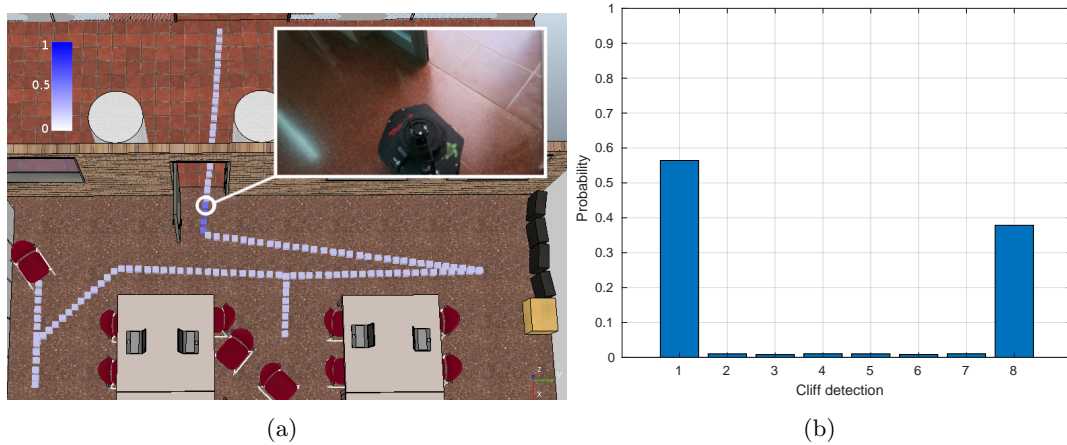


Figure 5.25: Anomaly detection and recovery for the cliff detection sensor. (a) Probability of anomalies (along the trajectory), color legend and real view in a point of interest. (b) Discrete posterior distribution for the ideal cliff node (see figure 5.9), corresponding to the point of interest. The state representation for this sensor is explained in the text.

information of the environment, since it is known that there are no cliffs along the whole path. The definition of the discrete states for this sensor is as follows. Since there are three binary cliff detectors, three bits are needed to indicate whether they are activated or not individually, thus one decimal number representing these bits is used, with an offset of one (e.g., the state labeled as “8” actually corresponds to “7”, meaning the binary number 111). As depicted in figure 5.25(b), the probability distribution for the ideal cliff detector assigns the maximum probability to the state “1”, which is the one that indicates no presence of cliffs.

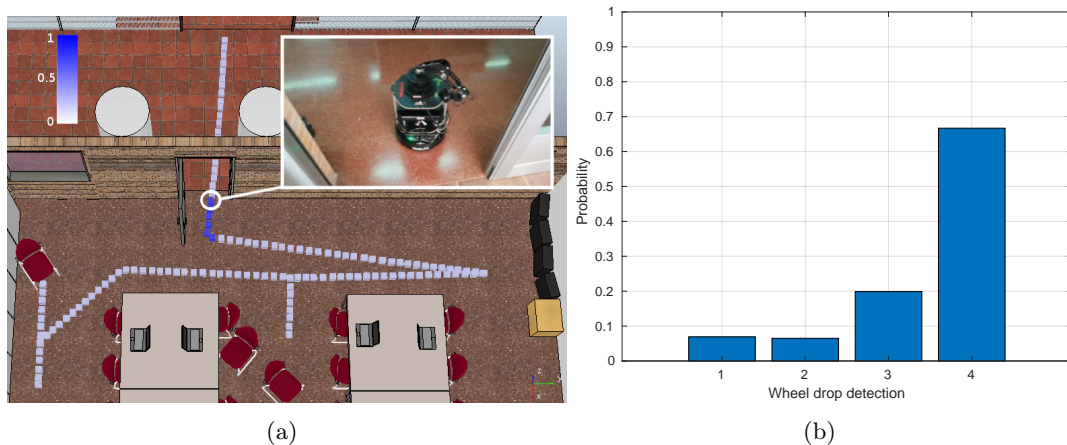


Figure 5.26: Anomaly detection and recovery for the wheel drop detection sensor. (a) Probability of anomalies (along the trajectory), color legend and real view in a point of interest. (b) Discrete posterior distribution for the ideal wheel drop node (see figure 5.6), corresponding to the point of interest. The state representation for this sensor is also explained in the text.

Lastly, the anomalous behavior of the wheel drop detection sensors is analyzed. Such sensors are actually two switches placed inside the mobile platform (see figure 5.2). They serve to indicate whether the wheels are making contact with the ground; in the case they do not, it can be concluded that the robot has been lifted or it simply tipped over. In the real experiment, one of

5.5 Implementation for the problem of navigation in human environments

these switches gets stuck near the end of the trajectory, leading to a false positive. This is properly detected by the sensory system as depicted in figure 5.26. Here, the states are defined such that “1” corresponds to drop detection for both wheels, “2” to left wheel drop, “3” to right wheel drop, and finally, “4” to none of them. Figure 5.26(b) shows how the correct state (i.e., the one named as “4”) gets the maximum probability in the ideal wheel drop distribution.

In conclusion, all the experiments shown in this section prove that the proposed approach manages to significantly reduce the cost of inference while allowing sensory anomaly detection and recovery. This is possible despite the fact that the results of such inference are approximate.

5.5 Implementation for the problem of navigation in human environments

This section covers the implementation of the proposed Bayesian sensory architecture for the problem of navigation in environments with human presence. First, the definition of the architecture is developed in subsection 5.5.1. After that, the set of simulated and real experiments carried out for the validation of the proposed approach are presented in subsections 5.5.2 and 5.5.3 respectively. Lastly, subsection 5.5.4 covers the use of feedforward neural networks for improving the efficiency of inference concerning this robotic application. A set of experiments to assess the error and computation time of this proposal is also presented.

5.5.1 Definition of the Bayesian architecture

The definition of the Bayesian architecture for the problem of navigation in human environments is also carried out by following the procedure detailed in section 4.3.2 (chapter 4). As explained before, all the features offered by the Bayesian sensory model will be necessary in this case, since there is sensory information evolving over time and also belonging to different levels of cognitive abstraction. The complete modeling process is summarized in the following, paying special attention to the most relevant parts, for the sake of brevity. A list with the necessary Bayesian sensors along with the knowledge they rely on is compiled in table 5.10.

Bayesian sensor	Sensory information source(s)
Pose and velocity predictor	Pose and speed of pedestrians (Spencer system)
Age sensor	Age of the detected pedestrian (external computer vision system)
Ghost pedestrian sensor	Map of the scene and localization of the pedestrian
Situation sensor	Social knowledge, pose and age of the detected pedestrian
Pose and velocity estimator	Pose and speed of pedestrians (Spencer system)
Long-term occlusion sensor	Geometry of occlusion zones and their persistence over time
Distance sensor	Pose of several pedestrians of interest
Difference of orientation sensor	(idem)
Identity sensor	Identity of the detected pedestrian (Spencer system)

Table 5.10: Bayesian sensors used in this model and their corresponding sources of data.

The Bayesian sensory architecture is built upon the nine Bayesian sensors listed in table 5.10, and, in this particular case, there are no cyclic dependencies among them. Taking into account the existing relations among these sensors, a dependency graph as the one shown in figure 5.27 can be defined. Note that the model is instantiated identically for each detected pedestrian in the

5.5 Implementation for the problem of navigation in human environments

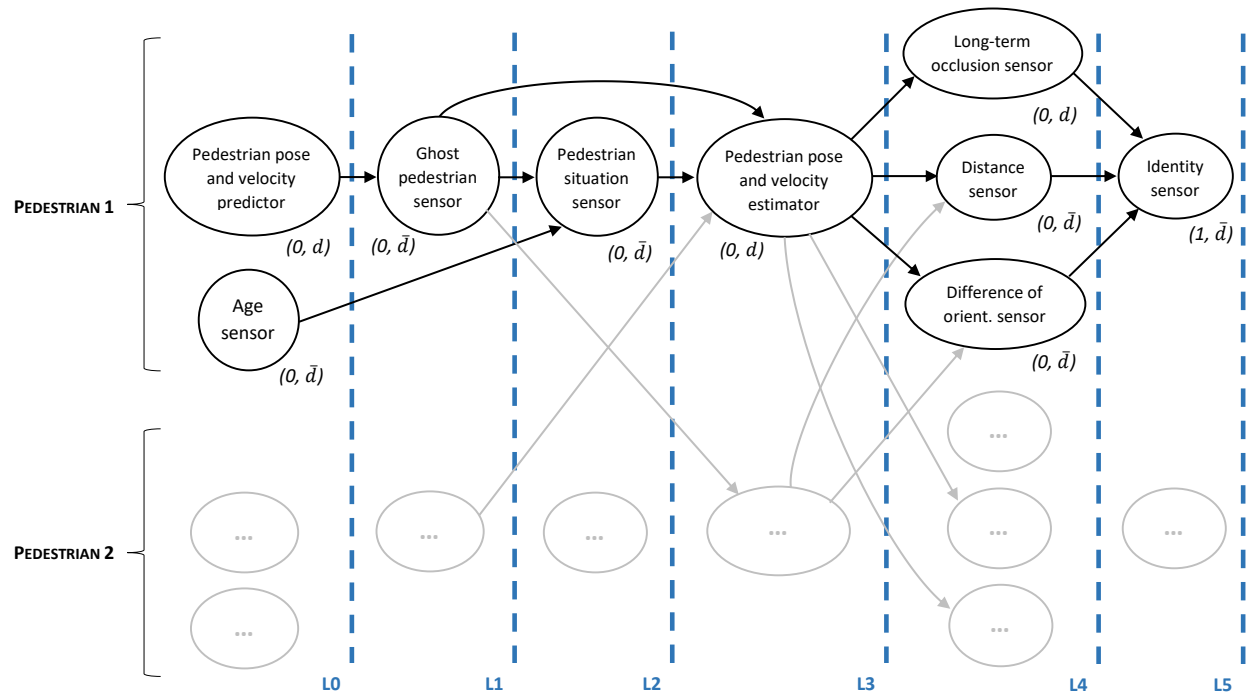


Figure 5.27: Annotated dependency graph for the proposed Bayesian architecture for each detected pedestrian, with layer assignment. Recall that the annotations in the graph indicate the level of cognitive abstraction and the dynamic character of each sensor, respectively. The interactions existing between different pedestrians are also shown.

system, and that these instances exchange information with each other. Figure 5.27 represents these connections only between two different pedestrians, since they can be applied analogously to every pair of detected pedestrians.

In the following, a complete description of the proposed Bayesian sensory model is provided. Some of the Bayesian sensors used rely, in turn, on auxiliary sensors that have not been included in the model definition for the sake of simplicity. All of them will be described as they appear within each layer of the architecture. Also, only some key CPDs will be shown, for the sake of brevity.

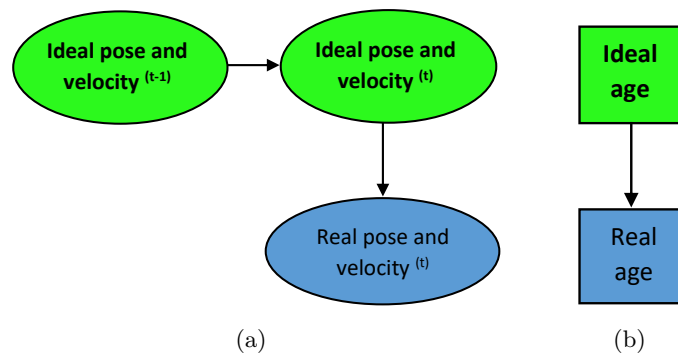


Figure 5.28: Bayesian sensors assigned to layer *zero* in the architecture. Recall that squared nodes represent discrete random variables and that round ones represent continuous variables. (a) Pedestrian pose and velocity predictor. (b) Age sensor.

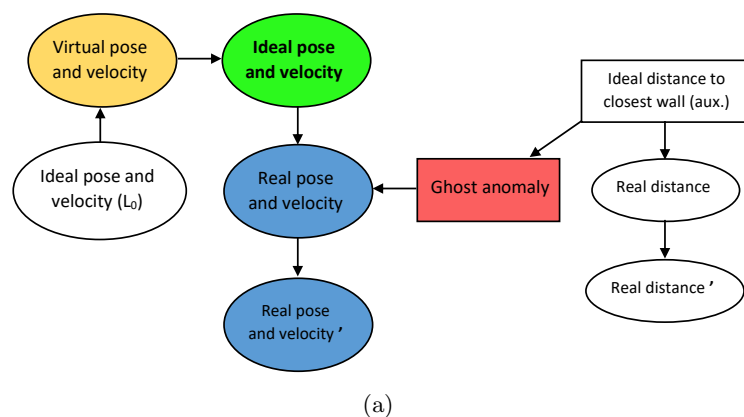
Layer *zero* (figure 5.28) contains two different Bayesian sensors. One of them is a dynamic sen-



5.5 Implementation for the problem of navigation in human environments

sensor (figure 5.28(a)), defined over continuous random variables, that serves to predict the pose and velocity of a pedestrian that was detected by the Spencer system in a previous iteration but not in the current one for some reason. In the case that the pedestrian is not missing, this sensor can also be used as a filtered estimation of the state by incorporating the information from Spencer as evidence (thus, the evidence set would be empty in the previous case). The prediction is based on a simple constant velocity model over six variables, namely, the three describing the two-dimensional pose of a pedestrian (i.e., x , y , θ) and their derivatives (i.e., v_x , v_y , ω , respectively), defined with respect to a global, fixed frame. This model will be improved in subsequent layers, as explained later on.

The other sensor in this layer is a so-called *age* sensor (figure 5.28(b)), which represents the age of a detected pedestrian using discrete binary variables for that (the possible values considered here are *young* and *elder*). This commonsense knowledge will be useful in subsequent layers in order to estimate the speed at which a detected pedestrian is most likely to move. This sensory information should come from an external source, for instance, from a computer vision system, with some uncertainty. Note that the age of a pedestrian could be considered at a higher level of cognitive abstraction; however, the sensor is assigned here to a low cognitive level since the information it represents only serves to modify the belief in the expected speed of a pedestrian, as mentioned before.



Ideal distance	$p(\text{Real Distance} \text{Ideal Distance})$
Too short	$\mathcal{N}(0; w^2/4)$ (m)
Normal	$\mathcal{N}(w; w^2/4)$ (m)

Figure 5.29: (a) Bayesian network corresponding to the ghost pedestrian sensor (layer one). Replicated ideal nodes from lower layers are specified, as well as auxiliary sensors. Again, squared nodes represent discrete variables and round ones continuous variables. (b) CPD for the real distance node of the auxiliary Bayesian sensor, with physical units. Here, w refers to the typical distance that a pedestrian usually keeps from nearby walls.

Layer one (figure 5.29) contains a unique Bayesian sensor, dedicated to the detection of non-existent, “ghost” pedestrians that are sometimes produced by the Spencer tracker. These fictitious pedestrians, if any, usually appear nearby walls or other boundaries. In order to include such an environmental knowledge, the anomalies subnetwork of this sensor is based on an auxiliary one that serves to determine whether the distance of a given pedestrian to the closest wall is too short. To obtain that, the auxiliary sensor counts with the pose estimation for the considered pedestrian from layer zero and with a map of the scene. In case of anomaly, i.e., in case the distance is

5.5 Implementation for the problem of navigation in human environments

actually too short, the recovered pose from the ideal node will have a highly inflated uncertainty, thus severely deteriorating its value. This is done in order that such a fictitious pedestrian can be eliminated from the system, since the implemented version of the architecture ignores detected pedestrians if their uncertainty is too high.

Layer 2 (figure 5.30) also contains only one sensor, which serves to reason about the situation of a given pedestrian within the scene. More specifically, this sensor aims to encode a more abstract version of the pose of a considered pedestrian, while simultaneously capturing the age. This information is represented as a discrete value indicating the *kind* of zone the pedestrian is situated along with the age, which will be useful to improve the pose and velocity estimation in subsequent layers. It is common that the map of a scene, specially in structured environments, is constituted by recognizable parts such as hallways, end of hallways, corners, etc. These ones are precisely the kind of zones considered in this sensor. Combining them with the age of a pedestrian, it would be possible to define a suitable motion prediction model according to this information, considered social knowledge. For instance, a pedestrian is more likely to move forward in a hallway, and also, is more likely to turn while being around a corner.

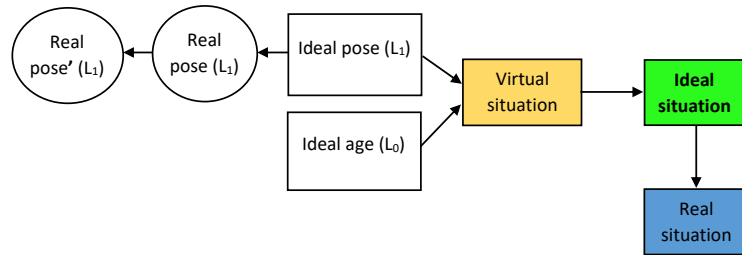


Figure 5.30: Bayesian network corresponding to the situation sensor (layer two). Again, replicated ideal nodes and auxiliary Bayesian sensors are specified.

Taking all the above into account, there will be as many possible values for the ideal sensor variable as the number of possible combinations of kinds of zones and ages. To encode the information, the sensor uses a replicated version of the age sensor from layer zero, and also discretizes the support of the continuous pose obtained from layer one, by defining different zones. Recall, again, that the use of more abstract information does not imply that the sensor must correspond to a high cognitive level, as long as such information is only employed for reasonings involving low-level data.

Layer 3 (figure 5.31) represents one of the most complex networks in the architecture. It contains a sensor that serves to recover a useful estimation of the pose and velocity of a pedestrian in case of severe occlusion. Also, this network can be used to improve the estimation of these state variables if there are no occlusions.

The anomaly subnetwork in this case is dedicated to the detection of occlusion events, encoding three different causes for that: occlusion when the pedestrian gets out of the field of view of the rangefinder, occlusion produced by a fixed obstacle (e.g., the scene itself) and occlusion provoked by other pedestrians (i.e., by dynamic obstacles). For the first two situations, the Bayesian sensor relies on two different discretizations of the pose of the considered pedestrian, obtained from layer one, along with environmental knowledge. Each one of these discretizations serves to determine whether the pedestrian is situated within one of these occlusion zones, defined either by the field of view of the rangefinder or by the scene itself, respectively. For the case of occlusion provoked by dynamic obstacles, the sensor incorporates an auxiliary one representing the difference between

5.5 Implementation for the problem of navigation in human environments

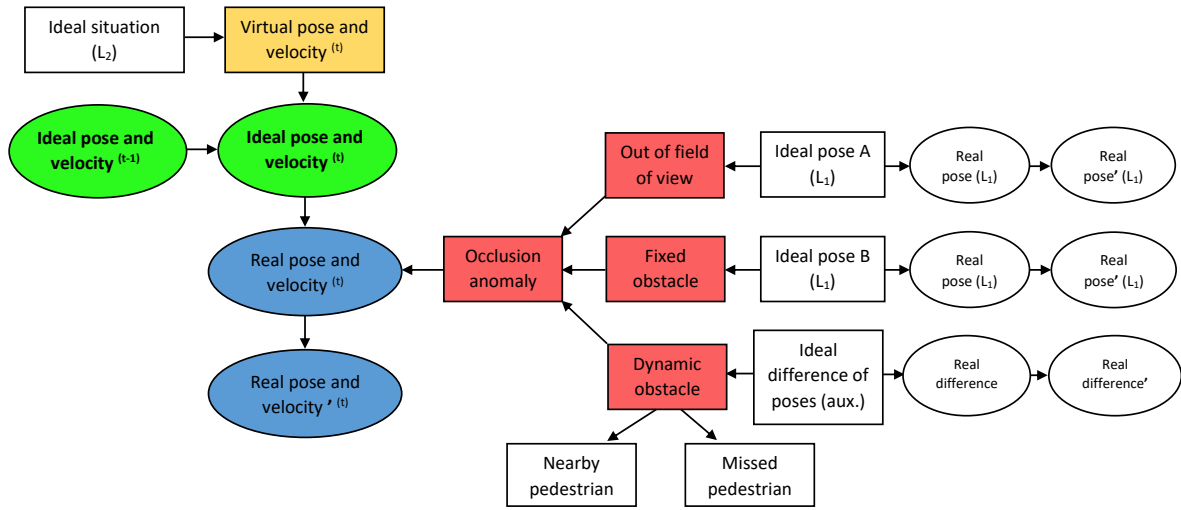


Figure 5.31: Bayesian network corresponding to the pedestrian pose and velocity estimator (layer three). Again, replicated ideal nodes and auxiliary Bayesian sensors are specified.

the estimated pose of two different pedestrians, being one of them the pedestrian of interest, and the other one, a pedestrian chosen for being the most likely to be occluding the former. This difference is also discretized into different zones in order to define in which ones would be the pedestrian occluded. This last part of the anomaly subnetwork also includes two extra discrete variables representing flags provided by the Spencer system, which are useful to determine the occlusion situation.

The rest of the network is dedicated to the recovery of the pose and velocity of the considered pedestrian, as mentioned above. For that, it incorporates the information from the situation given by the corresponding sensor in layer two. As explained before, this information serves to use an adequate motion model depending on the age of the pedestrian and the location in the scene. All these models are encoded in the CPD corresponding to the ideal node of the current time interval t . Each one corresponds to the mean of a multivariate Gaussian distribution, which is a constant velocity model of the form:

$$\begin{aligned}
 x_t &= x_{t-1} + v_{x_{t-1}} \Delta t \\
 y_t &= y_{t-1} + v_{y_{t-1}} \Delta t \\
 \theta_t &= \theta_{t-1} + \omega_{t-1} \Delta t \\
 v_{x_t} &= V \cos(\theta_{t-1}) \\
 v_{y_t} &= V \sin(\theta_{t-1}) \\
 \omega_t &= \Omega,
 \end{aligned} \tag{5.2}$$

where V is a constant linear speed, Ω is a constant angular speed and Δt is the elapsed time between intervals t and $t - 1$. Thus, each one of the mentioned models only differs from the others in its constants V and Ω , leading to a different motion depending on the situation of the pedestrian. The complete CPD is then a list of Gaussian distributions with different means depending on the value of the situation sensor (see figure 5.31). The covariance matrices for the Gaussian distributions in the CPD are all diagonal, only adding uncertainty depending on the amount of change of the corresponding variable over time. Recall, again, that all these state variables (i.e., pose and velocities) are defined over a global, fixed reference frame. Note, also, that the constant velocity

5.5 Implementation for the problem of navigation in human environments

model for pedestrians defined in equation (5.2) represents a nonlinear function of the previous state. In order to encode this in a CLG Bayesian network, the CPDs are transformed by using first-order Taylor series linearization, which involves the intermediate calculation of a Jacobian matrix of the model (please refer to [1] and [48] for further details).

Layer 4 (figure 5.32) contains three different Bayesian sensors. One of them is the long-term occlusion sensor (figure 5.32(a)), which is employed to determine whether any of the occlusions events defined before persist over time. For that, this sensor incorporates an auxiliary one that discretizes the recovered pose from layer three. This serves to determine whether a given pedestrian is situated within any of the mentioned occlusion regions. Note that the ideal node of the auxiliary sensor is defined over a discrete binary random variable, which takes a *true* value when the occlusion situation occurs. The more this situation persists over time, the greater will be the belief in a long-term occlusion. The other two sensors in this layer, i.e., the distance and difference of orientation sensors (figures 5.32 (b) and (c), respectively), rely on the recovered pose from the layer three of two different pedestrians, the one considered and the one having the most similar pose compared to the former. These sensors serve to provide a discrete distance and difference of orientation between these pedestrians, in order to determine whether their poses are actually similar. This information will be used by the identity sensor, as explained later on.

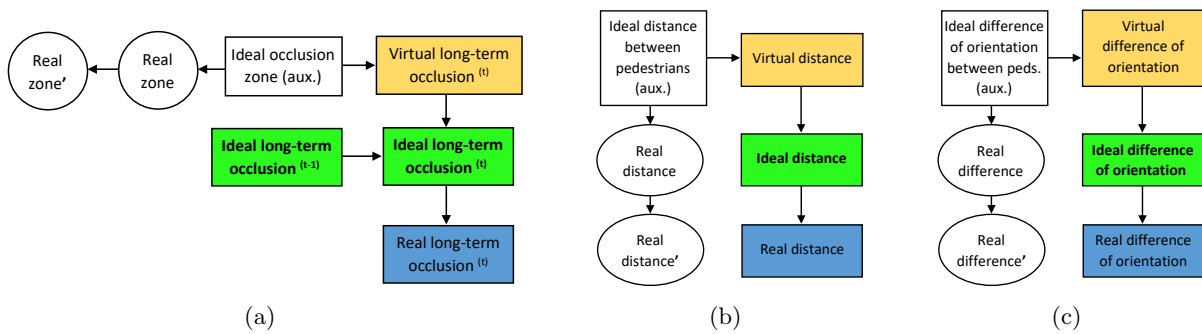
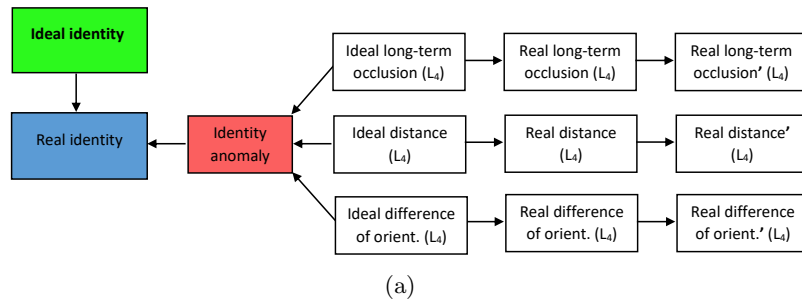


Figure 5.32: Bayesian sensors assigned to layer four in the architecture. (a) Long-term occlusion sensor. (b) Distance sensor. (c) Difference of orientation sensor.

Finally, layer 5 (figure 5.33) contains the identity sensor, which serves to recover the correct identity of a given pedestrian in case it has been confused with another one. This adverse situation occurs when a pedestrian re-appears after suffering a long-term occlusion event. In this case, the Spencer system would assign a new identity, leading to the presence of two pedestrians, the one being maintained by the sensory model during the occlusion period and the one just recovered by the tracker. The identity sensor allows to determine whether this is the case, relying on the information provided by the sensors in layer four. Thanks to the use of the identity sensor, the implemented version of the architecture is able to recover the correct identity of the affected pedestrian while ignoring the other one. Note that this sensor is considered here to belong to a higher level of cognitive abstraction, since its ideal node is defined over a discrete random variable encoding a reasoning about the identity of a pedestrian. In other words, low-level information of pose and distance is being used in this case to produce high-level information about identity, in contrast to the case of sensors in lower layers.

5.5 Implementation for the problem of navigation in human environments



Long-term occlusion	Distance	Difference of orient.	Anomaly	
			True	False
True	Short	Low	0.99	0.01
True	Long	Low	0.05	0.95
True	Short	High	0.05	0.95
True	Long	High	0.01	0.99
False	Short	Low	0.05	0.95
False	Long	Low	0.001	0.999
False	Short	High	0.01	0.99
False	Long	High	0.001	0.999

Figure 5.33: (a) Bayesian network corresponding to the identity sensor (layer five). (b) Tabular CPD for the identity anomaly node.

5.5.2 Simulated experiments

The proposed Bayesian architecture model for the problem of navigation in human environments has been tested in several simulations. As reported in section 5.3, the experiments described in this subsection have been carried out in a simulated environment based on Gazebo. This environment includes an scenario where the CRUMB robot navigates surrounded by two pedestrians (see figure 5.34). For the simulated tests, three kinds of experiments have been carried out. In all of them, the pedestrians incessantly follow a cyclical path between points A and D (see figure 5.34), and they are not aware of the presence of the robot. The pedestrian simulator (i.e., *Pedsim*) allows the emulation of age by setting one of its parameters, which has been modified in order that each pedestrian moves at a different pace (two different ages are considered, namely, *young* and *elder*). Each experiment or set of experiments is performed in the same conditions twice, one of them using the Spencer system only and the other one incorporating the proposed Bayesian sensory architecture for anomalies detection and recovery.

The first kind of experiment aims to prove that the use of the proposed Bayesian architecture serves to increase the safety of navigation, even under adverse conditions. In the experiment, the robot tries to maintain its pose around point B while avoiding approaching pedestrians. The safety of navigation has been assessed in this case through two measures: the distance to the closest pedestrian d , and the inverse time to collision $TTC^{-1} = \dot{d}/d$, which are commonly used in the literature related to robotic navigation [26]. In general, large and negative values of TTC^{-1} indicate high risk of collision, while values close to zero correspond to safe navigation. The results of 5 minutes of simulation are shown in figure 5.35, where only the data concerning the moments when anomalies occur are compared.

The most common anomaly in this case is the occlusion produced when the pedestrian leaves

5.5 Implementation for the problem of navigation in human environments

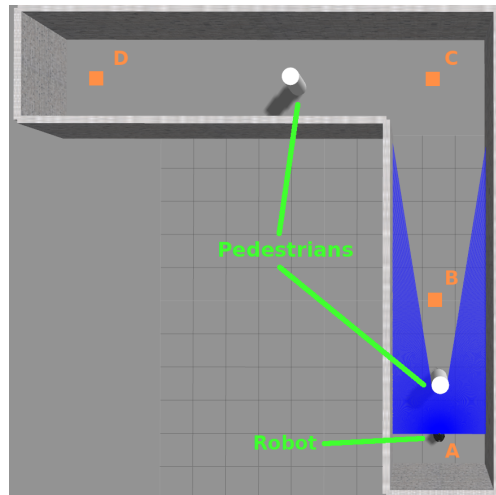


Figure 5.34: Top view of the controlled scenario used for the simulated experiments, with some points highlighted. The blue area corresponds to the field of view of the laser rangefinder.

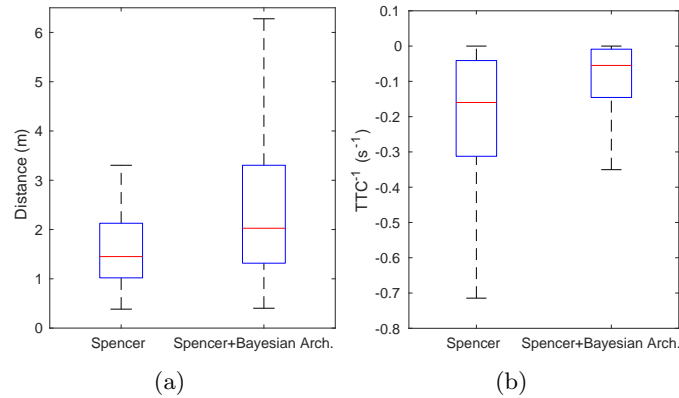


Figure 5.35: Comparative boxplots for the measures of safety considered in the first simulated experiment. (a) Distance to the closest pedestrian. (b) Inverse time to collision.

the horizontal field of view of the robot, which is of 180 degrees. The proposed Bayesian architecture manages to recover an estimated pose of the missed pedestrians (see figure 5.36), which has a direct impact on the safety of navigation: the median distance to the closest pedestrian increases from 1.45 to 2.03 meters, and the median TTC^{-1} from -0.16 to $-0.05 s^{-1}$. Also, the robot collides with the pedestrians a total of 7 times during the experiment carried out only with the Spencer system; when the proposed architecture is integrated, the test is collision free.

The second simulated experiment illustrates the effect of the presence of ghost pedestrians on the efficiency of navigation. The measure chosen to assess such efficiency is the time that the robot takes to go from point A to C. The test has been prepared so that none of the pedestrians are nearby the robot along such trajectory, and the experiments have been repeated 20 times for each configuration (i.e, with and without the integration of the Bayesian architecture). When the Spencer system is used without the proposed approach, some non-existent pedestrians appear, leading to avoidance maneuvering that is not actually needed (see figure 5.37). As a result, the total navigation time increases.

5.5 Implementation for the problem of navigation in human environments

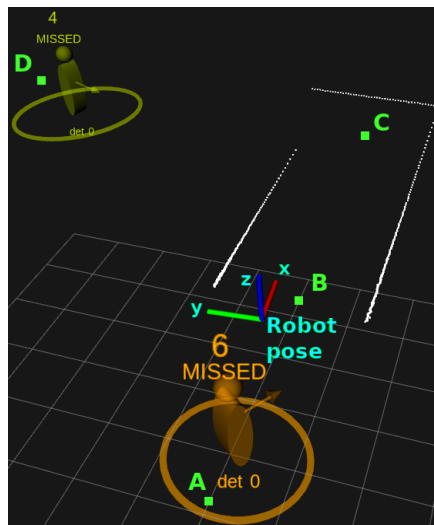


Figure 5.36: Three dimensional view of the pedestrian tracking system using the Bayesian sensory architecture during the first experiment. The robot is pointing towards the positive sense of the X axis. Pedestrian 6 is recovered despite being placed behind the robot, and pedestrian 4 despite being occluded by the walls of the scene, detected by the 2D laser (white lines).

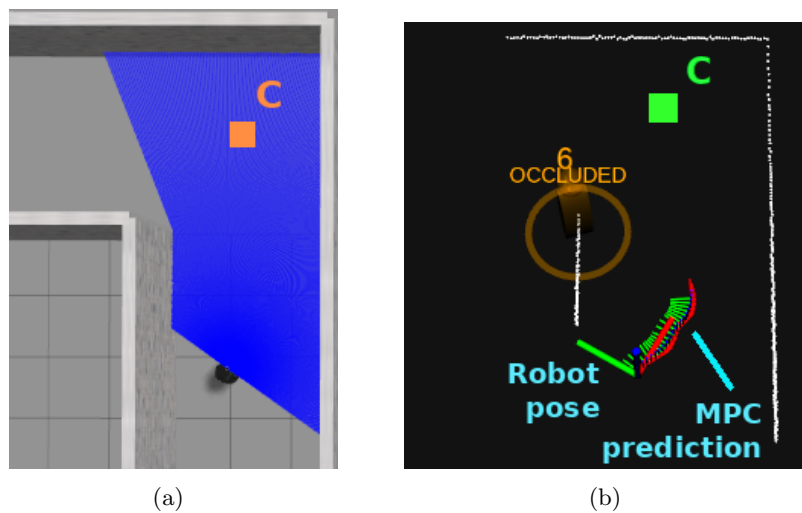


Figure 5.37: Snapshot of the simulated environment during an avoidance maneuvering due to the presence of a ghost pedestrian (second experiment). (a) Top view of the simulation in Gazebo. (b) Top view of the pedestrian tracking system when it is not using the Bayesian architecture. Here, the MPC controller predicts the maneuvering action to be performed in order to avoid the fictitious pedestrian.

The impact of navigating around fictitious pedestrians has been studied through the linear correlation between the time that the robot is nearby them and the total navigation time. The same linear correlation has been calculated as well for the number of such anomalous detections. The coefficient of determination R^2 is of 0.9234 for the first fit and of 0.5385 for the second, thus, the total time needed to complete the trajectory is strongly correlated with the amount of time the robot remains nearby ghost pedestrians, rather than with the number of them. The results for this experiment are shown in figure 5.38 (the second linear fit is omitted for being less relevant). In this case, the application of the proposed Bayesian architecture manages to reduce the median navigation time from 26.62 to 23.14 seconds.

5.5 Implementation for the problem of navigation in human environments

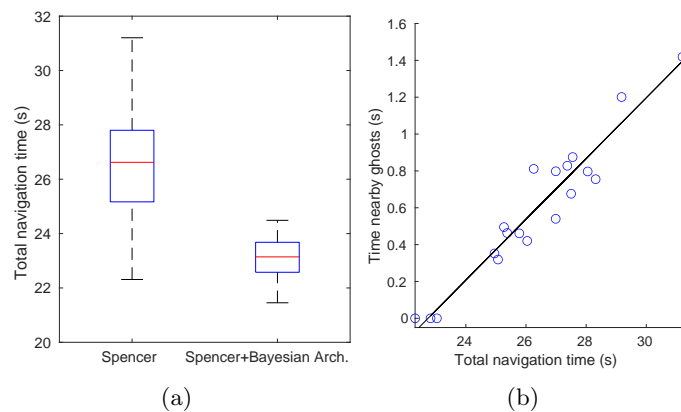


Figure 5.38: Results for the second simulated experiment. (a) Comparative boxplots for the total navigation time. (b) Scatter plot of the time nearby ghost pedestrians versus the total navigation time, with a linear fit.

The third and last simulated experiments aim to assess the utility of the proposed Bayesian architecture in the case of identity anomalies. For that, the robot tracks the pose of one of the pedestrians throughout the scene until the identity is missed or confused with another one. In this case, the total tracking time is measured, and four different configurations are used for the tests by combining the age of the tracked pedestrian (young or elder) and the availability of the sensory architecture (present or absent). The experiment has been repeated a total of 10 times for each configuration. The tracking time results are depicted in figure 5.39. As shown, the proposed approach increases the median tracking time from 20.68 to 73.19 seconds for the young pedestrian and from 42.46 to 84.92 seconds for the elder one.

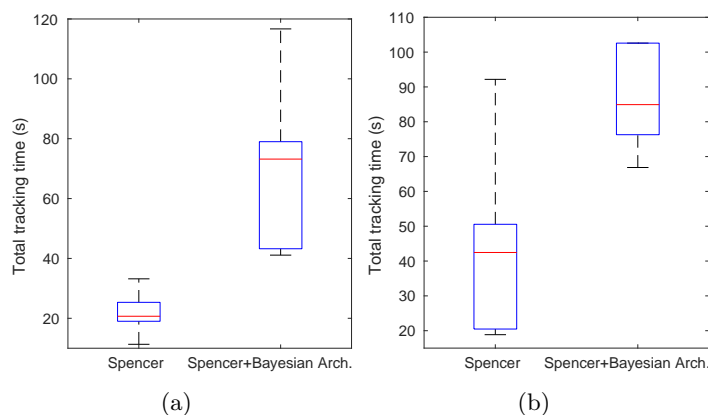


Figure 5.39: Comparative boxplots for the total tracking time of pedestrians in the third simulated experiment. (a) Results for the young pedestrian. (b) Results for the elder pedestrian.

These results prove that the use of the proposed Bayesian architecture enhances the overall robustness of the pedestrian detection system, since the total tracking time achieved without identity anomalies is much higher. To reinforce this idea, it is also interesting to analyze the kind and amount of adverse events being overcome during these tests. Table 5.11 shows, for each experimental setup and kind of occlusion anomaly, the mean percentage of time dedicated to recover from those situations, which are the ones behind the majority of identity problems. Recall that a pedestrian can be occluded when leaving the field of view of the robot or due to the presence of

5.5 Implementation for the problem of navigation in human environments

either an static or dynamic obstacle.

Pedestrian	Bayesian Arch.	Occlusion anomaly			Total
		FOV	Static	Dynamic	
Young	Present	10.98	11.50	6.65	29.13
	Absent	2.42	5.38	6.63	14.43
Elder	Present	4.58	0.66	1.23	6.47
	Absent	1.27	0.00	2.00	3.27

Table 5.11: Average percentage of time for recovering occlusion anomalies.

The obtained results show that the amount of abnormal situations recovered during the tests using the proposed Bayesian architecture is double in general. In other words, this means that the proposed approach is still robust despite a greater amount of anomalies occurring during a much longer period of time. In particular, this is true for the occlusion anomalies related to the field of view of the robot and the static obstacles, which tend to last longer in the case the architecture is employed. In contrast, the Spencer system in isolation recovers reasonably well from the brief occlusions produced by dynamic obstacles (i.e., by other pedestrians).

5.5.3 Real experiments

The real experiments presented in this subsection have been conducted in situations similar to the ones analyzed in the simulations. They are intended both to demonstrate the strong correlation between simulated results and the possible real achievements of the proposed solution and to show its potential when implemented in a real robot. Recall that these experiments are not developed in crowded environments, since their aim is to properly assess the capabilities of the proposed approach, as explained in section 5.2.

In this case, two different experiments have been performed, both in the real scenario shown in figure 5.40. As commented in section 5.3, the software and hardware employed is the same as the one used in the simulations, with the exception of the netbook available on board the CRUMB robot. Each test in these real experiments is repeated with and without incorporating the Bayesian architecture to the Spencer system, and the obtained results are compared, as in the simulated case.

The first experiment aims to validate the simulated results related to the efficiency of navigation in presence of ghost, non-existent pedestrians. For that, the robot is ordered to go from point A to C (see figure 5.40) in the absence of actual people. During this trajectory, some fictitious pedestrians are detected by the Spencer system, as in the simulated case. The impact of navigating around such false detections of human presence is assessed by measuring the time that the robot is placed nearby ghost pedestrians and the total navigation time. The obtained results are compiled in table 5.12. As expected, they prove that the incorporation of the Bayesian architecture manages to reduce the navigation time, since it is conceived to notice false detections of pedestrians, allowing the robot to ignore them. The results also show the correlation between the total navigation time and the one that the robot is influenced by the presence of these pedestrians. In other words, they prove that, the longer the robot is situated around fictitious pedestrians, the more likely it is that the navigation time increases. When this abnormal situation takes place in the real setting, it can be observed how the robot performs unnecessary avoidance maneuvering in the case the Bayesian sensory architecture is not used.

5.5 Implementation for the problem of navigation in human environments

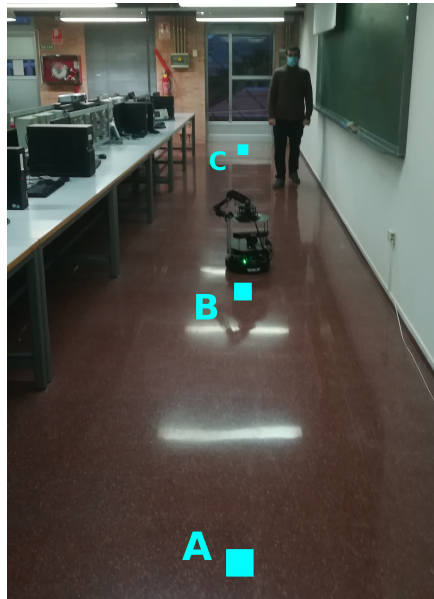


Figure 5.40: Image of the setup used for the real experiments, with some points highlighted. The corridor shown is 13 meters long. A video of these experiments is available at http://babel.isa.uma.es/_fordownloading/thesisMCQ_videoPedestrians.mp4. Recall that the pedestrian appearing here is only present during the second experiment.

Experiment	Measure	Spencer	Spencer+Arch.
First	Total navigation time	29.1 s	26 s
	Time nearby ghosts	1.41 s	0.18 s
Second	Number of collisions	4	0
	Time with occlusions	53.88 s	89.12 s

Table 5.12: Comparative results for the real experiments.

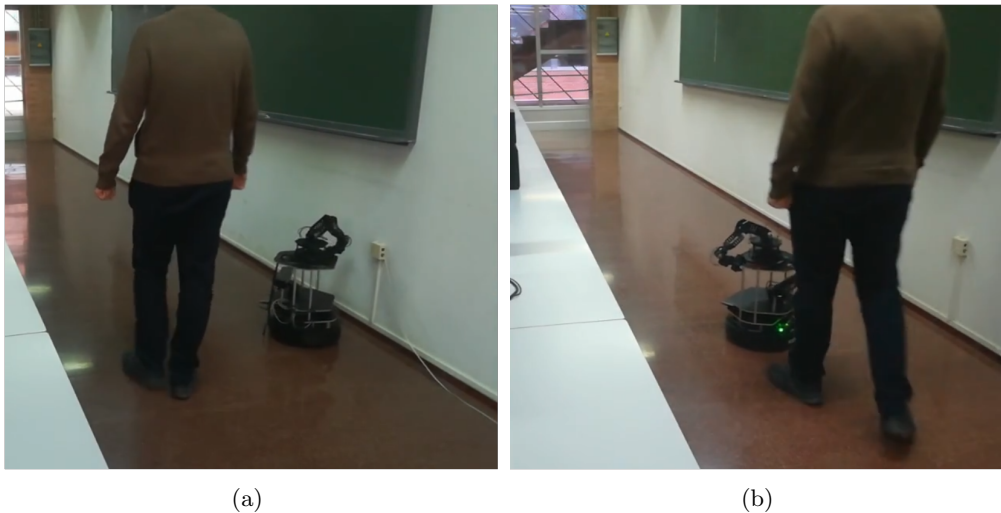


Figure 5.41: Snapshot of an instant recorded during the second experiment. Here, the pedestrian is walking from point A to C while being situated out of the field of view of the robot. (a) The robot manages to avoid the pedestrian thanks to the use of the Bayesian sensory architecture. (b) The pedestrian collides with the robot, which is only relying on the detections provided by the Spencer system.

5.5 Implementation for the problem of navigation in human environments

The second and last experiment is intended to validate the simulated results concerning the safety of navigation. In this test, the robot tries to maintain its pose around point B while avoiding a pedestrian that follows a cyclical path between points A and C during 3 minutes. Here, the safety of navigation is assessed by counting the times the robot collides with the pedestrian. Also, the total time of the test with occlusion anomalies is measured. The obtained results (table 5.12) prove that the use of the Bayesian architecture serves to enhance the safety of navigation, since the test is collision free when the proposed approach is incorporated. As in the simulated case, the period of time dedicated to recover from anomalous situations is much longer when the architecture is used, since it is prepared to handle long-term occlusions. This demonstrates, again, the robustness of the proposed approach, which enables to improve safety even under abnormal conditions lasting much longer. In fact, it is observed in the real setting that only the use of the Bayesian architecture enables the robot not to collide with the pedestrian when situated out of its field of view (see figure 5.41).

5.5.4 Improving inference efficiency with neural networks

This subsection describes one of the last tasks carried out during the thesis period, that is, the use of feedforward neural networks to improve the cost of the proposed approximate inference algorithm over Bayesian networks. This proposal has been implemented for the problem of navigation in environments with human presence, and it has been tested through different simulations. In the following, a basic knowledge of neural networks is assumed, including classical models and training algorithms. Readers may refer to [166] for a review on these notions.

As discussed in section 4.4.2 of chapter 4, a feedforward neural network can be trained to represent any of the typical queries related to a Bayesian network (in this case, it will correspond to a particular layer of the Bayesian architecture). For that, there are some aspects that need to be taken into account. One of them consists in the obtention of an adequate dataset for the training process. In this case, inputs will correspond to available evidences in each layer and also to CPD parameters of the replicated or auxiliar variables present, since both are necessary to fully define the Bayesian network and the corresponding query. Outputs will simply correspond to parameters of the distributions that need to be queried, such as the mean and variance of a Gaussian density, in the case of continuous variables, or the set of probabilities defined by a mass function, in the case of discrete variables. The training datasets for each neural network have been obtained by sampling the space of possible values for the inputs and then calculating the corresponding outputs. This process represents a simulation of the different Bayesian networks defined in the architecture, and it has been implemented in MATLAB by using the BNT toolbox as well. Note, however, that the mentioned simulation process may not be computationally feasible if the number of inputs is high enough. To solve that, the attention is focused only on those input values representing common situations concerning navigation in human environments. The rest will be generalized by the neural network itself.

Another aspect that has to be considered before training is the definition of each neural network in terms of number of hidden layers and neurons as well as the shape of the activation functions employed. There are no strict rules to choose an appropriate number of neurons for each hidden layer or the number of them, however, the output activation function must fit the shape of the output data, which represent parameters of some pdf or pmf, as mentioned before. Here, purely linear activation functions are used for outputs referring to queries involving continuous variables; in the discrete case, a logistic sigmoid function is employed if the query variable has a discrete

5.5 Implementation for the problem of navigation in human environments

binary support (since its image is defined between zero and one) and a softmax function is used if the support is non-binary (since its shape is the one that best fits the form of a pmf). Regarding hidden layers, the chosen functions are always hyperbolic tangents. Figure 5.42 shows the shapes of the mentioned activation functions.

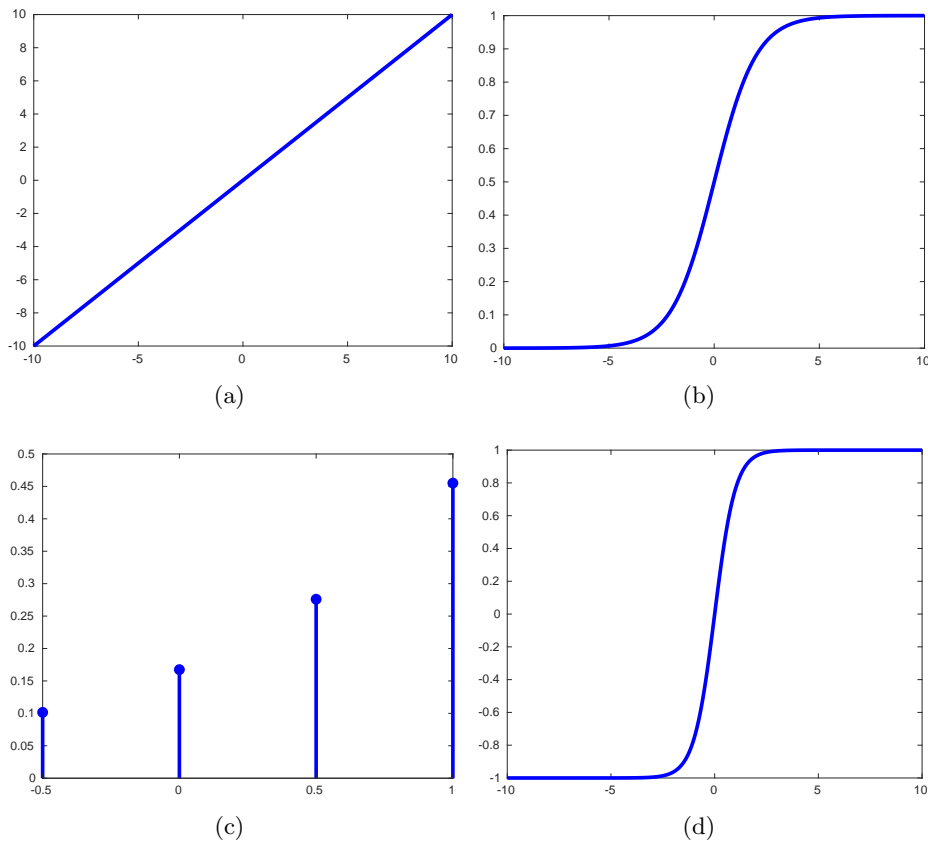


Figure 5.42: Shapes of the activation functions used for the definition of the neural network models employed here. (a) Purely linear. (b) Logistic sigmoid. (c) Softmax (output corresponding to a four-dimensional input vector). (d) Hyperbolic tangent.

The last aspect to be taken into account for the training process is the choice of an appropriate learning algorithm and a hardware platform to run it (here, this refers either to a CPU or to a GPU). Again, there is no general rule for such a decision, thus, it has been made based on the experience acquired throughout different trials. The most suitable algorithms for this case are the well-known *Levenberg-Marquardt* (LM) [167] and the *Scaled Conjugate Gradient* (SCG) [168]. The hardware employed for training is the same desktop PC used for the experiments described in this chapter, which has an Intel i7-9700K CPU. Also, those networks with higher computational requirements have been trained with an NVIDIA Tesla V-100 GPU². The queries considered essential for the problem of navigation in human environments (and thus, the ones considered here) are listed in table 5.13. Also, the configuration of all the neural network models designed to approximate such queries is compiled in table 5.14.

Note that some queries are represented by more than one neural network (see table 5.14). This is the case for layer 0 (corresponding to the pose and velocity prediction), layer 2 (dedicated to

²This hardware is part of an NVIDIA DGX Station.

5.5 Implementation for the problem of navigation in human environments

Layer	Query variable (Q)
0	Ideal pose and velocity (current time interval)
1	Ghost anomaly
2a	Ideal situation (<i>young</i> pedestrian)
2b	Ideal situation (<i>elder</i> pedestrian)
2c	Ideal situation (unknown age)
3a	Out of field of view (anomaly node)
3b	Fixed obstacle (anomaly node)
3c	Dynamic obstacle (anomaly node)
3d	Occlusion anomaly
3e	Ideal pose and velocity (current time interval)
4a	Ideal long-term occlusion
4b	Ideal distance
4c	Ideal difference of orientation
5	Identity anomaly

Table 5.13: Query variables employed for the problem of navigation in human environments. Each variable corresponds to a layer of the proposed Bayesian sensory architecture defined in section 5.5.1. Layer numbering has been completed with letters to denote different queries for the same Bayesian network.

Layer	Neural networks designed			
	# Inputs	# Hidden layers and neurons	# Outputs	Training algorithm
0	7	1 x 21	6	LM (CPU)
	16	1 x 48	21	LM (CPU)
1	2	2 x 25	1	SCG (CPU)
2a	5	2 x 25	20	SCG (CPU)
2b	5	2 x 25	20	SCG (CPU)
2c	5	2 x 25	20	SCG (CPU)
3a	2	2 x 25	1	SCG (CPU)
3b	12	2 x 25	1	SCG (CPU)
3c	13	4 x 15	1	SCG (CPU)
3d	3	2 x 25	1	SCG (CPU)
3e	35	1 x 595	6	SCG (GPU)
	30	1 x 180	1	SCG (GPU)
	30	1 x 180	1	SCG (GPU)
	30	1 x 180	1	SCG (GPU)
4a	3	2 x 25	1	SCG (CPU)
4b	2	2 x 25	1	SCG (CPU)
4c	2	2 x 25	1	SCG (CPU)
5	3	2 x 25	1	SCG (CPU)

Table 5.14: Configuration parameters for all the feedforward neural networks designed, each one representing a query in table 5.13. Again, layer numbering has been completed with letters to denote different queries for the same layer in the Bayesian network. Here, the number of hidden layers and neurons corresponding to each neural network is expressed in the form $A \times B$, which means that the network has A layers with B neurons each. The training algorithm employed for each network is specified along with the hardware platform used (see text).

the situation sensor) and layer 3e (conceived for the improved estimation of pose and velocity). The main reason for using several networks is simply to reduce the computational requirements for the training process. For instance, the two networks designed for layer 0 represent the mean and covariance of the prediction, respectively. In the case of layer 2, three equal networks have been defined, two of them for a different age of the detected pedestrian (i.e., *young* and *elder*), and the last one for the case in which the age is not known. Here, only these three possibilities are considered, for the sake of simplicity. Finally, layer 3e is represented by four different networks. The first one refers to the estimation mean, and the other three, to some elements of the covari-

5.5 Implementation for the problem of navigation in human environments

ance. Such matrix has not been fully represented, since there are numerous values in it that do not vary much across the different tests performed for both the simulated and real experiments carried out. These values have been ignored in order to reduce the network output size, and thus, the complexity of its training.

Once all the necessary neural networks have been defined and trained, it is possible to assess the performance of this new inference approach. For that, new validation datasets have been used, all of them obtained the same way as the ones for training. Then, two measures of performance have been calculated, namely, the error made w.r.t. to the approximate inference algorithm proposed in this thesis and the computation time. As in previous sections, the error has been obtained by using the Hellinger's distance (see equation 5.1). However, that definition is only valid for probability distributions over discrete variables. In this context, there are also some queries involving continuous variables, which will always be distributed normally. The Hellinger's distance between two multivariate normal distributions $P \sim \mathcal{N}(\mu_1, \Sigma_1)$ and $Q \sim \mathcal{N}(\mu_2, \Sigma_2)$ is [169]:

$$H(P, Q) = \sqrt{1 - \frac{\det(\Sigma_1)^{1/4} \det(\Sigma_2)^{1/4}}{\det\left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{1/2}} \exp\left(-\frac{1}{8}(\mu_1 - \mu_2)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} (\mu_1 - \mu_2)\right)} \quad (5.3)$$

where $0 \leq H(P, Q) \leq 1$.

The obtained results for the error performance are shown in figure 5.43. They prove that most of the neural networks defined produce a reasonably good estimation of the desired queries. Nevertheless, some of them produce a significant but moderate error (see figure 5.43(b)) that can be improved by simply increasing the training time (i.e., none of the parameters established for these networks have to be modified). In general, the obtained error is not high enough to distort the original distributions queried, and thus, these approximations can be employed for the task of inference in the proposed navigation problem.

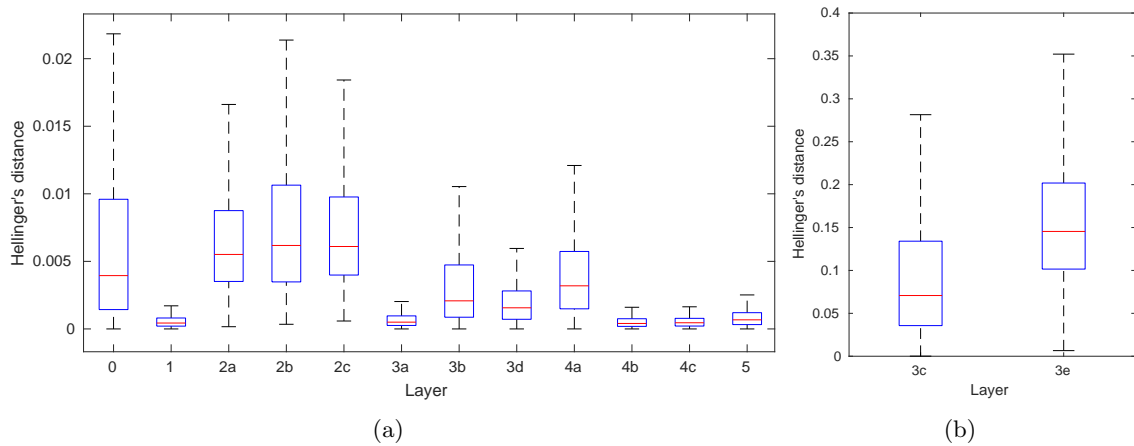


Figure 5.43: Hellinger's distance for the distributions produced by the defined neural networks w.r.t the ones obtained by the approximate algorithm introduced in this thesis, calculated for each layer of the Bayesian architecture. These boxplots represent 9000 inferences per layer. (a) View of the lowest errors. (b) View of the highest errors.

A comparative study of the computational efficiency has also been carried out. Table 5.15 shows the time needed by each approach, (i.e., by the proposed algorithm and by the implemented

5.5 Implementation for the problem of navigation in human environments

Layer	Single query time		
	<i>BN inference</i>	<i>NN inference</i>	<i>NN/BN ratio (means)</i>
0	$0.002 \pm 8.440 \cdot 10^{-4}$	$1.024 \cdot 10^{-4} \pm 1.930 \cdot 10^{-5}$	0.051
1	0.004 ± 0.001	$3.777 \cdot 10^{-5} \pm 2.421 \cdot 10^{-6}$	0.009
2a	0.030 ± 0.008	$3.549 \cdot 10^{-5} \pm 1.684 \cdot 10^{-6}$	0.001
2b	0.031 ± 0.009	$3.534 \cdot 10^{-5} \pm 1.628 \cdot 10^{-6}$	0.001
2c	0.030 ± 0.007	$3.580 \cdot 10^{-5} \pm 4.089 \cdot 10^{-6}$	0.001
3a	0.004 ± 0.002	$3.447 \cdot 10^{-5} \pm 1.831 \cdot 10^{-6}$	0.009
3b	$0.007 \pm 4.266 \cdot 10^{-4}$	$3.436 \cdot 10^{-5} \pm 1.447 \cdot 10^{-6}$	0.005
3c	0.015 ± 0.006	$3.548 \cdot 10^{-5} \pm 1.723 \cdot 10^{-6}$	0.002
3d	$9.762 \cdot 10^{-4} \pm 5.719 \cdot 10^{-4}$	$3.885 \cdot 10^{-5} \pm 3.506 \cdot 10^{-5}$	0.040
3e	0.013 ± 0.003	$5.379 \cdot 10^{-4} \pm 9.212 \cdot 10^{-4}$	0.041
4a	$9.252 \cdot 10^{-4} \pm 3.833 \cdot 10^{-4}$	$2.612 \cdot 10^{-5} \pm 1.902 \cdot 10^{-6}$	0.028
4b	$0.004 \pm 9.560 \cdot 10^{-4}$	$3.701 \cdot 10^{-5} \pm 1.006 \cdot 10^{-4}$	0.009
4c	0.004 ± 0.001	$3.856 \cdot 10^{-5} \pm 7.677 \cdot 10^{-6}$	0.009
5	$9.352 \cdot 10^{-4} \pm 3.447 \cdot 10^{-4}$	$3.466 \cdot 10^{-5} \pm 1.863 \cdot 10^{-6}$	0.037

Table 5.15: Single query time per layer achieved by inference with Bayesian networks and by inference with neural networks. The results are expressed in seconds, in the form $\mu \pm 2\sigma$, where μ and σ are the mean and standard deviation, respectively, of 9000 tests carried out per layer and inference method. The ratio is calculated for mean values only.

neural networks) to obtain a single query. This experiment has been repeated 9000 times for each layer in the architecture and inference approach. The results show that, in general, the execution time achieved by using neural networks is several orders of magnitude lower compared to the one corresponding to inference with Bayesian networks.

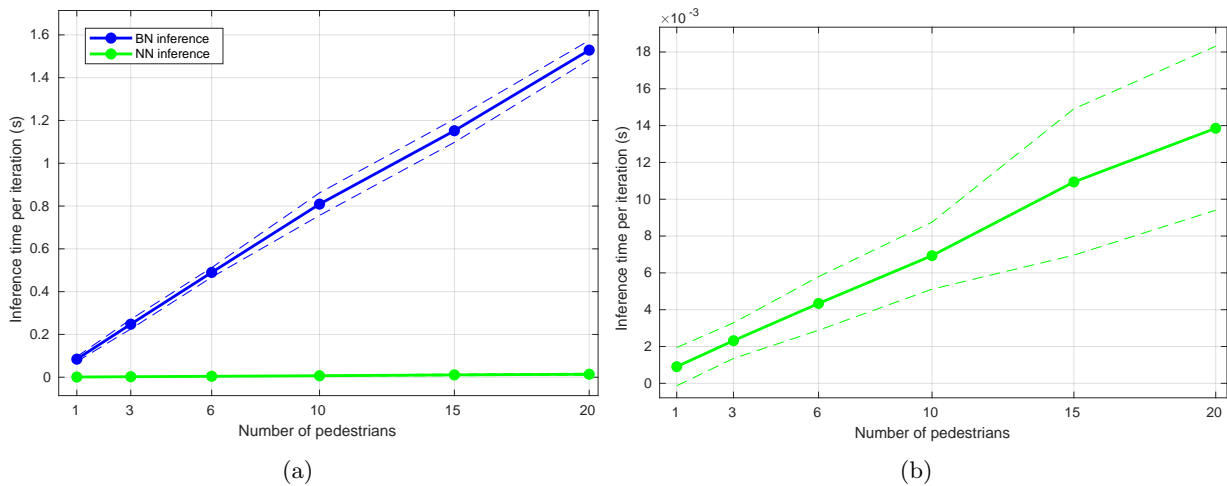


Figure 5.44: Mean inference time per sensory sampling iteration achieved by inference with Bayesian networks and by inference with neural networks, as a function of the number of detected pedestrians. Dashed lines correspond to $\pm 2\sigma$, where σ is the standard deviation for each test, containing 5000 samples. (a) Comparative results for both inference approaches. (b) Zoomed view of the results for inference with neural networks.

Finally, another study regarding computational cost has also been performed. As explained in chapter 4, the proposed inference algorithm computes queries for each sensory sampling iteration taking all the existing layers into account. Also, in the current navigation problem, the Bayesian architecture is completely replicated for each detected pedestrian. Thus, it would be interesting to analyze the cost of inference per sampling iteration for both approaches, considering an increasing

5.5 Implementation for the problem of navigation in human environments

number of *actual* pedestrians. Recall that, in the proposed algorithm, each layer is inferred the same times as the number of detected pedestrians. Several sequences of 5000 iterations have been simulated in these conditions, each one for a different number of detected pedestrians, up to 20 of them. The obtained results are shown in figure 5.44. Both inference approaches achieve a computational cost that is approximately linear with the number of processed pedestrians. These results also prove that the use of feedforward neural networks does not only increases the efficiency for a single layer, but that also enables for real time inference with a reasonable amount of pedestrians. In contrast, direct inference over Bayesian networks only achieves reasonable execution times for a reduced number of them. Recall that the time of inference in these experiments has been measured by using the same desktop PC as in the training process.

Final remarks

6.1 Conclusions

The work presented in this thesis has contributed to different aspects of both robotics and artificial intelligence, all of them related, in particular, to the application of Bayesian networks to the problem of sensory diagnosis and recovery in mobile robots. Understanding robotic sensory systems and their common anomalies has been essential to the research process, which precisely began with the study of the impact of such abnormal conditions on the performance of common tasks of mobile robots. This study represents one of the first contributions of the thesis.

A complete analysis of the impact of sensory abnormal behavior in mobile robots would be overwhelming, given the wide variety of sensors commonly deployed in mobile platforms. For that reason, a feasible analysis should only focus on those sensory devices considered critical for basic robotic operation. Many of the essential tasks performed by a mobile robot (e.g., localization, mapping, navigation, etc.) rely on the use of range sensors, since they provide metrical knowledge of the environment that is crucial for the mentioned tasks. Range information is employed by the methodologies supporting these basic operations, which are often grounded on probabilistic frameworks such as Bayesian estimators.

The study presented in this thesis was precisely aimed at analyzing the effects of abnormal range observations on the performance of Bayesian filters, and, in turn, on the robotic operation itself. For that, the study has addressed Bayesian filtering inference from a generalist perspective, by using the paradigm of Dynamic Bayesian Networks. This generic Bayesian filter has been modeled by taking into account the features of the most common robotic rangefinders. Their main limitations have also been analyzed, as well as those factors that are likely to affect the filter performance. Different simulated experiments with diverse conditions have been designed, and novel and relevant conclusions have been obtained from their use with rigorous statistical methods. Also, these conclusions have been validated in a real situation with a mobile robot.

In particular, the obtained results show that the parameters of the tracking problem considered for the study (i.e., the speed and initial position of the obstacle) do not have any relation with the

6.1 Conclusions

performance of Bayesian filters. In contrast, the increase of the amount of abnormal sensory data, i.e., missing and biased observations, generally affects all the considered measures of performance negatively. The combination of both kinds of anomalous data worsens the expected accuracy of the filter, while only missing observations are capable of increasing the filtering uncertainty. Lastly, one of the conclusions that was not expected before conducting the statistical analyses is that the convergence performance is seriously affected by both kinds of anomalous observations separately, and that their combination does not lead to a worse convergence rate in case of an already deteriorated situation.

A mere analysis of the impact of abnormal sensory behavior is not the only aim of this thesis; these anomalies have also been addressed by different, novel methodologies in order to eliminate or mitigate them. For that, the thesis has introduced a Bayesian network-based framework, a so-called Bayesian sensory architecture, that allows the representation of any kind of sensory system for mobile robots and its use in intelligent inference. Essentially, the aim of these models is to contribute to a more robust and reliable sensory operation. This is achieved by encoding as many interactions among sensors as possible (both cyclic and non-cyclic), leading to the detection of possible faulty situations that prevent sensors to perceive their environment and the state of the robot properly. Additionally, the modelling process allows the intelligent integration of heterogeneous sources of information (sensory and environmental data, human commonsense, etc.) even considering dynamic aspects and different levels of cognitive abstraction. All this knowledge is then taken into account to recover the correct state of the sensory information being represented, i.e., the one that should have been obtained in a fault-free scenario.

The main drawback of this proposal is the high computational cost of the existing exact and approximate inference methods over Bayesian networks, which prevents its implementation for applications with real time requirements. For this reason, the thesis has also contributed with a new algorithm that performs exact inference in a three-dimensional approximate model of the robotic sensory system, which has been defined as well to be used by the proposed method. In order to validate both the model and the inference algorithm, different simulated and experimental studies have been carried out. The obtained results show that the proposed method achieves a considerable reduction of the computation time, which is especially noticeable as the size of the model increases. In addition, it provides inference results that are useful in most cases to perform sensory anomaly detection and recovery, as shown in the experiments with the real robot.

The proposed modeling framework has also been deployed in a real robotic application in order to show its utility. In particular, it has been implemented for the problem of robotic navigation in environments with human presence. The proposed model aims in this case to improve the robustness of a state-of-the-art pedestrian detection and tracking system, in order to achieve a more safe and efficient robotic navigation in these environments. This has been successfully proved through experimental validation in a set of both simulated and real experiments. Lastly, a proposal based on the use of feedforward neural networks to increase the efficiency of inference even more has also been introduced. This inference approach has also been implemented for the problem of navigation in human environments, showing that it is possible to perform the same reasonings offered by the proposed Bayesian architecture in real time, even when the number of detected pedestrians is considerable.

6.2 Future work

There exist different tasks that could be developed for future works, all of them related to the contributions presented in the thesis. One of these works consists in the study of the impact of abnormal sensory behavior. The conclusions derived from such a study presented in this dissertation currently rely on a set of factors that can be expanded. For instance, this can be done to include a wider variety of robotic sensors, ranging from basic devices (gyroscopes, encoders, etc.) to more complex ones (vision sensors, thermal cameras, etc.). Also, more factors could be considered to take into account a greater amount of filtering parameters as well as modes of robotic operation. The impact of variations on all of the mentioned aspects would be studied regarding the performance of Bayesian estimators. However, it could also be possible to study the performance in the scope of more general models of Bayesian estimation, such as hybrid models like the Switching Kalman Filter [57], which can also be implemented within the framework provided by Dynamic Bayesian Networks.

Regarding the proposed Bayesian sensory architecture and the corresponding inference algorithm, there are also different tasks that can be addressed in the future. The proposed model and algorithm have been validated in a real mobile robot, however, they should be implemented for a wide variety of robotic platforms (aerial, terrestrial and submarine) that use more complex sensory devices, in order to complete their validation. Regarding computational cost, the efficiency of inference with the proposed model has been successfully improved, making it suitable even for real time robotic applications. However, the use of feedforward neural networks for that relies on a previous training procedure that cannot be easily generalized for every sensory model, and thus, another methods for compiling Bayesian network queries such as *arithmetic circuits* [58] could be studied. The modeling process should also be modified so that it is done more automatically and autonomously, keeping its ability to reflect human knowledge as well as other heterogeneous sources of information. For that, it would be interesting to explore the existing algorithms for parameter and structure learning in the context of Bayesian networks.

The application of the proposed Bayesian architecture to concrete robotic problems has also been successful; however, some aspects should be studied more thoroughly concerning the current implementations, which, of course, could also be expanded to a wider variety of tasks. One of the implementations proposed in the thesis is conceived for the problem of navigation in human environments. Although several simulated and real tests have been performed to prove the utility of the proposed approach, these experiments should also be carried out in more crowded scenarios and in a wider variety of them, for instance. Finally, it would be also interesting to study to what extent the integration of human knowledge at different levels of abstraction manages to enhance the robustness of existing algorithms related to pedestrian detection and tracking.

Full report of the results obtained for the study of abnormal sensory behavior

This appendix contains a full report of all the tree graphs, ANOVA tables and population histograms that have been produced for the study of abnormal sensory behavior developed in chapter 3. In particular, these results correspond to the procedure explained in section 3.3.5 and constitute the basis of the derived conclusions, presented in section 3.4.2. The results are organized here by each measure of performance and each factor. For the sake of brevity, those ANOVA tables leading to invalid, multimodal conclusions have been omitted.

A.1 Expected accuracy performance

First, this report focuses the attention on the expected accuracy performance. There is a common first step to all the analyses consisting in the obtention of a four-way ANOVA for the population data. This is shown in table A.1.

The tree graph for the case of the expected accuracy population with factor A is depicted in figure A.1 and the ANOVA tables used during the process are collected in figure A.2. Taking into account these results, a complete set of conclusions can be stated for factor A (again, the reduced notation is used for the values of factors shown in table 3.2):

- Factor A has no effect on the expected accuracy of the filter given that $C = 1$.
- Factor A has no effect given that $B = 1$ and $C = 2$.
- Factor A has no effect given that $B = 2$ and $C = 2$.

A.1 Expected accuracy performance

Source	SS	df	MS	F	p-Value
A	0.0000	1	0.0000	0.0237	0.8775
B	0.7041	1	0.7041	1067.5699	0.0000
C	727.8939	1	727.8939	1,103,603.2161	0.0000
D	0.0020	1	0.0020	3.0079	0.0829
AxB	0.0000	1	0.0000	0.0005	0.9822
AxC	0.0020	1	0.0020	3.0027	0.0832
AxD	0.0001	1	0.0001	0.1927	0.6607
BxC	0.7904	1	0.7904	1198.4044	0.0000
BxD	0.0013	1	0.0013	1.9418	0.1635
CxD	0.0005	1	0.0005	0.7927	0.3733
AxBxC	0.0020	1	0.0020	2.9629	0.0853
AxBxD	0.0000	1	0.0000	0.0264	0.8709
AxCxD	0.0002	1	0.0002	0.3474	0.5556
BxCxD	0.0005	1	0.0005	0.6824	0.4088
AxBxCxD	0.0001	1	0.0001	0.0855	0.7700
Within cells	3.1976	4848	0.0007		

Table A.1: Four-way ANOVA table for the expected accuracy performance.

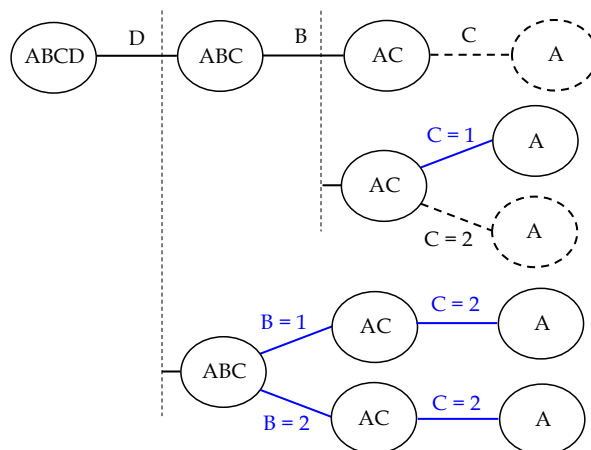


Figure A.1: Tree graph for the analysis of the effect of factor A on the expected accuracy performance of the filter. Dashed nodes and arcs correspond to rejected conclusions due to multimodal populations. Arcs in blue denote decisions on the value of factors based on interactions that are forced to get unimodality in the data.

This summarizes into the fact that the initial position of the obstacle has no relevant influence on the accuracy, regardless of the value of the remaining factors. Here the omega squared measure has not been used because none of the effects nor interactions were considered important. Note as well that all the interactions considered in figure A.1 have been forced in order to get unimodality. In figure A.3, histograms of the population data are shown for this performance at different levels of factor A, according to the conclusions previously stated. Recall that ANOVA only studies the differences among the groups means; in this case it is shown that such difference is barely noticeable. A secondary effect that can be pointed out here is the fact that an increase in the percentage of missing observations (factor B) leads to a higher variance in the expected accuracy that can be obtained by the filter (comparing figures A.3 (b) and A.3 (c)), which can be of importance in a practical range sensing application.

Now the attention is focused on the case of factor B (the anomaly of missing range measurements

A.1 Expected accuracy performance

Source	SS	df	MS	F	p-value
A	0.0008	1	0.0008	1.2462	0.2643
Within cells	3.1976	4848	0.0007		

(a)

Source	SS	df	MS	F	p-value
A	0.00001	1	5.8277e-6	0.0088	0.9251
Within cells	3.1976	4848	0.0007		

(b)

Source	SS	df	MS	F	p-value
A	0.00212	1	0.00212	3.2146	0.073
Within cells	3.1976	4848	0.0007		

(c)

Figure A.2: ANOVA tables for the analysis of the effect of factor A (initial distance of the object to the sensor) on the expected accuracy of the filter. (a) One-way ANOVA for factor A given $C = 1$. (b) One-way ANOVA for factor A given $B = 1$ and $C = 2$. (c) One-way ANOVA for factor A given $B = 2$ and $C = 2$.

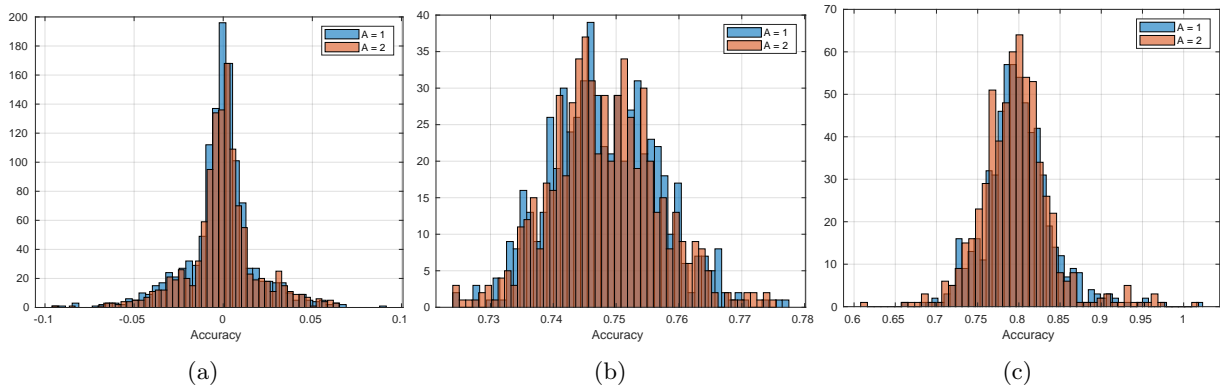


Figure A.3: Histograms for the conclusions about the effect of factor A on the expected accuracy performance, represented for the two levels of such factor with additional restrictions on the population. (a) Factor $C = 1$. (b) Factor $B = 1$ and factor $C = 2$. (c) Factor $B = 2$ and factor $C = 2$.

due, for instance, to absorptions on particular surfaces). The tree graph for this analysis is shown in figure A.4 and the corresponding ANOVA tables appear in figure A.5. This time, only two conclusions are needed to explain the data:

- Factor B has no effect on the expected accuracy of the filter given that $C = 1$.
- Factor B has effect given that $C = 2$.

In this case, positive effects and interactions are being considered; in order to confirm them, the necessary omega squared values have been calculated. For the $B \times C$ interaction (see table A.1) it is $\hat{\omega}^2 = 0.1975$, and, for the B main effect with $C = 2$, $\hat{\omega}^2 = 0.3175$, thus, the ANOVA results can be accepted. The above conclusions imply that the amount of missing observations in a range sensor has effect on the filter accuracy only in the case that the amount of biased observations is high. This can be viewed in the histograms shown in figure A.6 for this factor and its restrictions. More specifically, the percentage of missing observations has no effect when there are no biased

A.1 Expected accuracy performance

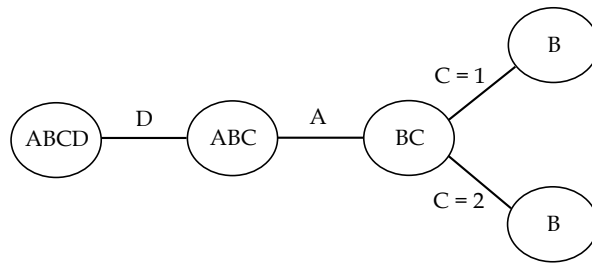


Figure A.4: Tree graph for the analysis of the effect of factor B on the expected accuracy performance.

Source	SS	df	MS	F	p-value
B	0.00125	1	0.00125	1.8901	0.1693
Within cells	3.1976	4848	0.0007		

(a)

Source	SS	df	MS	F	p-value
B	1.4933	1	1.4933	2.2641e+3	0
Within cells	3.1976	4848	0.0007		

(b)

Figure A.5: ANOVA tables for the analysis of the effect of factor B (amount of missing observations) on the expected accuracy. (a) One-way ANOVA for factor B given $C = 1$. (b) One-way ANOVA for factor B given $C = 2$.

observations, although the variance in this performance increases noticeably when it is higher (figure A.6 (a)). On the other hand, the presence of missing observations leads to a worse expected accuracy and also increases the variance of the expected accuracy of the filter (figure A.6 (b)).

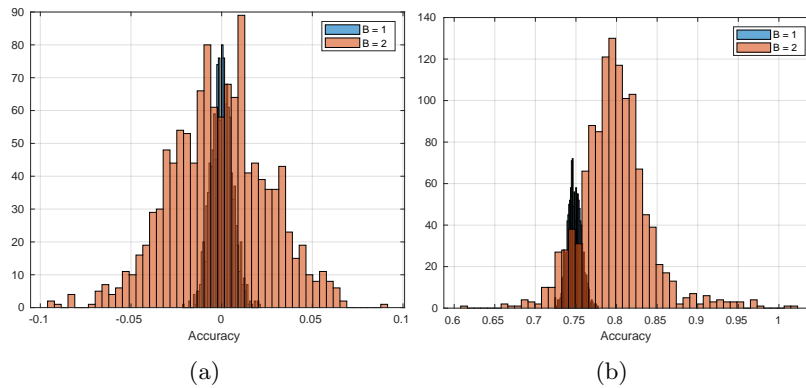


Figure A.6: Histograms for the conclusions about the effect of factor B on the expected accuracy performance, represented for the two levels of such factor with additional restrictions on the population. (a) Factor $C = 1$. (b) Factor $C = 2$.

Regarding factor C (amount of biased observations, produced for instance by reflections), the tree graph is depicted in figure A.7 and the ANOVA tables employed in the process, in figure A.8. The obtained conclusions for this factor are:

- Factor C has effect on the expected accuracy of the filter given that $B = 1$.
- Factor C has effect given that $B = 2$.

A.1 Expected accuracy performance

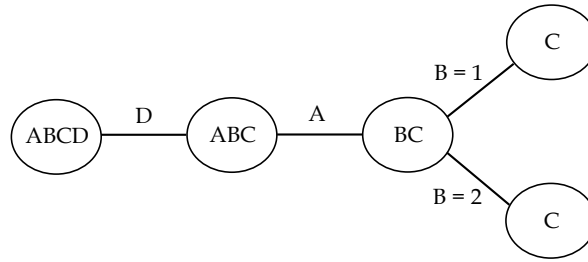


Figure A.7: Tree graph for the analysis of the effect of factor C on the expected accuracy performance.

Source	SS	df	MS	F	p-value
C	340.356	1	340.356	5.1603e+5	0
Within cells	3.1976	4848	0.0007		

(a)

Source	SS	df	MS	F	p-value
C	388.328	1	388.328	5.8877e+5	0
Within cells	3.1976	4848	0.0007		

(b)

Figure A.8: ANOVA tables for the analysis of the effect of factor C (amount of biased observations) on the expected accuracy. (a) One-way ANOVA for factor C given B = 1. (b) One-way ANOVA for factor C given B = 2.

The interaction in the four-way ANOVA that leads to these conclusions is again BxC (see table A.1), and an omega squared value was provided for it before. In this case, omega squared values are also necessary for assessing both main effects of factor C; these are $\hat{\omega}^2 = 0.9907$ for the case B = 1 and $\hat{\omega}^2 = 0.9918$ for the case B = 2, thus confirming the strength of the effects. As predicted by the linear regression method, the percentage of biased observations has a high influence on the filter accuracy, regardless of the value of the remaining factors. This can be noticed in the histograms of figure A.9. When the presence of biased observations is high, the accuracy is much worse in general. The only influence of factor B consists in the increase of the variance of this performance measure (this can be noticed comparing figures A.9 (a) and (b), although it is only a secondary effect).

Lastly, the analysis of factor D is addressed (speed of the obstacle). The corresponding tree graph is shown in figure A.10 and the ANOVA tables generated during the process appear in figure A.11. Given these results, the complete set of conclusions for factor D are:

- Factor D has no effect on the expected accuracy of the filter given that C = 1.
- Factor D has no effect given that B = 1 and C = 2.
- Factor D has no effect given that B = 2 and C = 2.

In this case there is no need to check any omega squared value, since there are no interactions nor main effects. Also, note that all the interactions appearing in figure A.10 have been forced by us to get unimodal data. These conclusions allows to ensure that the speed of the obstacle has no relevant influence on the filter accuracy, regardless of the value of the remaining factors. This can be seen in the histograms of figure A.12, where the only difference relies on the fact that the accuracy mean and variance increase when factors B and C take their highest values; this has nothing to do with the impact of factor D.

A.2 Expected uncertainty performance

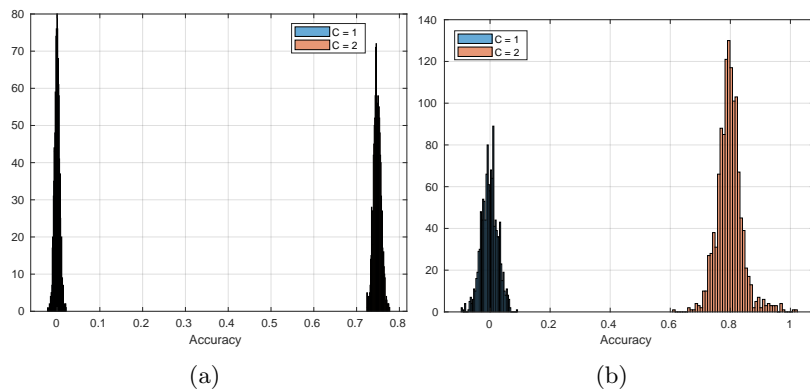


Figure A.9: Histograms for the conclusions about the effect of factor C on the expected accuracy performance, represented for the two levels of such factor with additional restrictions on the population. (a) Factor B = 1. (b) Factor B = 2.

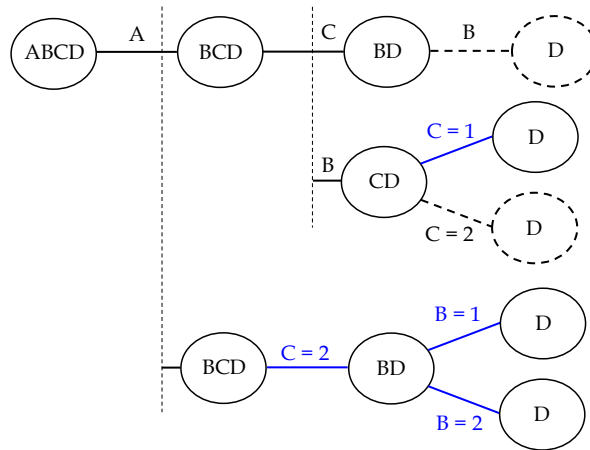


Figure A.10: Tree graph for the analysis of the effect of factor D on the expected accuracy performance. Dashed nodes and arcs correspond to rejected conclusions due to multimodal population. Arcs in blue denote decisions on the value of factors based on forced interactions.

A.2 Expected uncertainty performance

The report continues now with the analysis of the expected uncertainty performance. Following the same procedure as before, the first step consists in performing a four-way ANOVA for the population data in this case (see table A.2), which will be the basis for the subsequent analyses.

The analysis begins by addressing the case of factor A (initial position of the obstacle). The corresponding tree graph is depicted in figure A.13 and the only ANOVA needed for this factor is shown in table A.3. In this case, a special situation arises. As shown in figure 3.7, all the obtained data for the performance are identical when there are no missing observations (i.e., when factor B takes its low value) for the reasons explained in section 3.3.6. Under these circumstances, it does not make sense to perform any ANOVA; it is simply concluded that none of the factors have any effect when $B = 1$, since no change in the population distribution takes place. Taking this into account and the obtained results for the case $B \neq 1$, the complete set of conclusions for factor A is:

- Factor A has no effect on the expected uncertainty of the filter given that $B = 1$.

A.2 Expected uncertainty performance

Source	SS	df	MS	F	p-value
D	0.00227	1	0.00227	3.4444	0.0635
Within cells	3.1976	4848	0.0007		

(a)

Source	SS	df	MS	F	p-value
D	1.2609e-5	1	1.2609e-5	0.0191	0.8900
Within cells	3.1976	4848	0.0007		

(b)

Source	SS	df	MS	F	p-value
D	0.0003	1	0.0003	0.4980	0.4808
Within cells	3.1976	4848	0.0007		

(c)

Figure A.11: ANOVA tables for the analysis of the effect of factor D (speed of the obstacle) on the expected accuracy. (a) One-way ANOVA for factor D given $C = 1$. (b) One-way ANOVA for factor D given $B = 1$ and $C = 2$. (c) One-way ANOVA for factor D given $B = 2$ and $C = 2$.

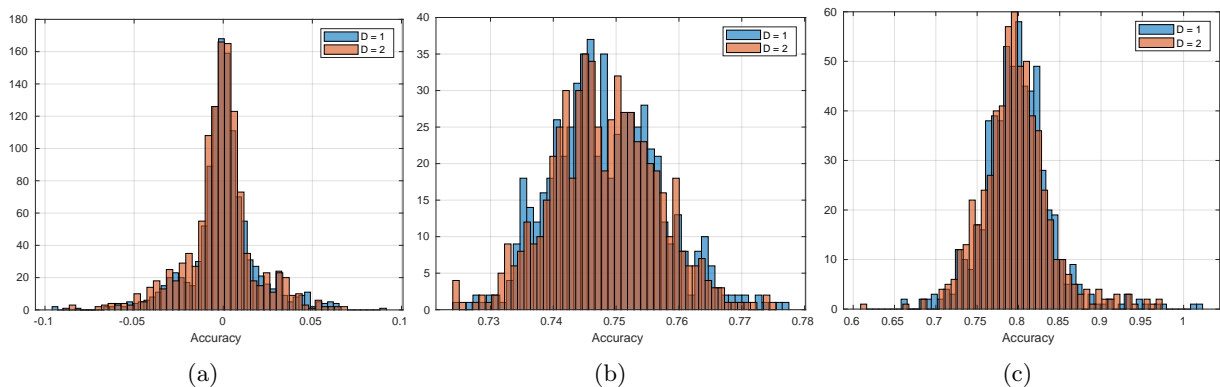


Figure A.12: Histograms for the conclusions about the effect of factor D on the expected accuracy performance, represented for the two levels of such factor with additional restrictions on the population. (a) Factor $C = 1$. (b) Factor $B = 1$ and factor $C = 2$. (c) Factor $B = 2$ and factor $C = 2$.

- Factor A has no effect given that $B = 2$.

These results imply that the initial position of the obstacle does not have any influence on the uncertainty of the filter, as shown in the histograms of figure A.14. We will omit, from now on, all the histograms related to the case $B = 1$, since they are all identical.

For factor B (amount of missing observations), the resulting tree graph is shown in figure A.15. In this case no extra ANOVA tables are necessary, since factor B has effect with strength $\hat{\omega}^2 = 0.9943$, which can be derived directly from table A.2. This result implies that the amount of missing observations from the sensor has an important impact on the expected uncertainty of the filter. As explained before, the number of available observations in a filtering process influence the uncertainty of the posterior distribution. In this case, an increase in the percentage of missing observations lead to a greater uncertainty, as shown in the histograms of figure A.16, regardless of the value of the rest of the factors.

A.2 Expected uncertainty performance

Source	SS	df	MS	F	p-value
A	1.3281e-7	1	1.3281e-7	0.2015	0.6535
B	0.5550	1	0.5550	842141.6088	0.0000
C	2.5223e-8	1	2.5223e-8	0.0383	0.8449
D	7.8373e-10	1	7.8373e-10	0.0012	0.9725
AxB	1.3281e-7	1	1.3281e-7	0.2015	0.6535
AxC	9.9627e-7	1	9.9627e-7	1.5118	0.2189
AxD	1.7472e-6	1	1.7472e-6	2.6514	0.1035
BxC	2.5223e-8	1	2.5223e-8	0.0383	0.8449
BxD	7.8373e-10	1	7.8373e-10	0.0012	0.9725
CxD	6.3747e-8	1	6.3747e-8	0.0967	0.7558
AxBxC	9.9627e-7	1	9.9627e-7	1.5118	0.2189
AxBxD	1.7472e-6	1	1.7472e-6	2.6514	0.1035
AxCxD	1.1712e-7	1	1.1712e-7	0.1777	0.6733
BxCxD	6.3747e-8	1	6.3747e-8	0.0967	0.7558
AxBxCxD	1.1712e-7	1	1.1712e-7	0.1777	0.6733
Within cells	0.00319	4848	6.5899e-7		

Table A.2: Four-way ANOVA table for the expected uncertainty performance.

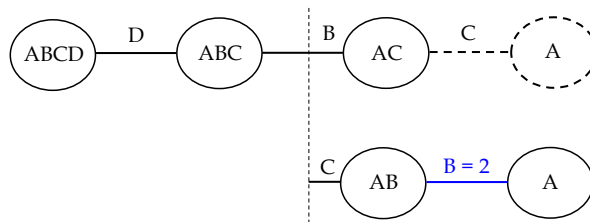


Figure A.13: Tree graph for the analysis of the effect of factor A on the expected uncertainty performance. The dashed node and arc correspond to a rejected conclusion due to multimodal population. The arc in blue represents a decision on the value of factor B based on a forced interaction.

Source	SS	df	MS	F	p-value
A	2.6562e-7	1	2.6562e-7	0.4031	0.5255
Within cells	0.00319	4848	6.5899e-7		

Table A.3: One-way ANOVA table for factor A given B = 2.

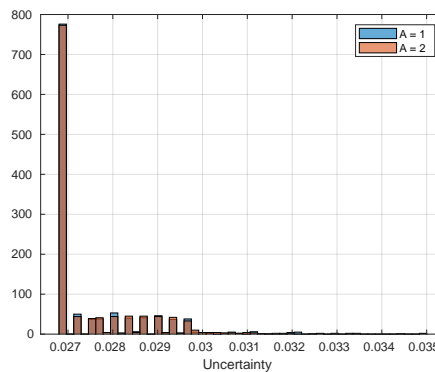


Figure A.14: Histograms for the conclusion of the effect of factor A on the expected uncertainty performance when B = 2, represented for the two levels of factor A.

A.2 Expected uncertainty performance

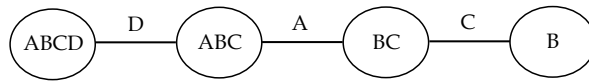


Figure A.15: Tree graph for the analysis of the effect of factor B on the expected uncertainty performance.

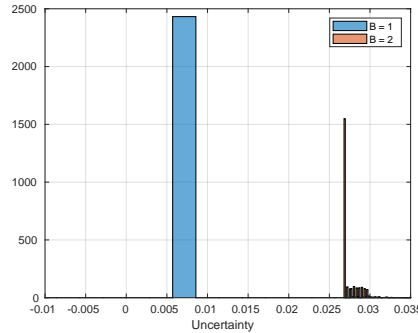


Figure A.16: Histograms for the conclusion of the effect of factor B on the expected uncertainty performance, represented for the two levels of the factor.

The attention is now focused on the analysis of factor C (amount of biased observations). The corresponding tree graph is depicted in figure A.17 and the necessary ANOVA is shown in table A.4. With these results, the conclusions derived for this factor are:

- Factor C has no effect on the expected uncertainty of the filter given that $B = 1$.
- Factor C has no effect given that $B = 2$.

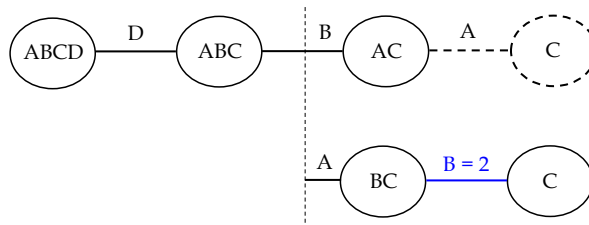


Figure A.17: Tree graph for the analysis of the effect of factor C on the expected uncertainty performance. The dashed node and arc correspond to a rejected conclusion due to multimodal population. The arc in blue represents a decision on the value of factor B based on a forced interaction.

This means that the abnormality of biased observations in the sensor (e.g., reflections) does not modify the filter uncertainty in any case, which can be seen in the histograms of figure A.18. This is due to the fact that uncertainty does not depend on the concrete values of the observations but on the number of them, as we discussed before.

Source	SS	df	MS	F	p-value
C	5.0446e-8	1	5.0446e-8	0.0766	0.7820
Within cells	0.00319	4848	6.5899e-7		

Table A.4: One-way ANOVA table for factor C given $B = 2$.

A.3 Convergence performance

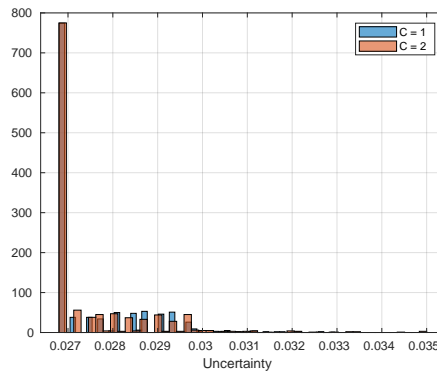


Figure A.18: Histograms for the conclusion of the effect of factor C on the expected uncertainty performance when $B = 2$, represented for the two levels of factor C.

Lastly, the effect of factor D (speed of the obstacle) is analyzed on this performance. The procedure that has been followed is shown in the tree graph of figure A.19, and the necessary ANOVA, in table A.5. Taking into account these results, the complete set of conclusions is:

- Factor D has no effect on the expected uncertainty of the filter given that $B = 1$.
- Factor D has no effect given that $B = 2$.

These conclusions allow us to assure that the speed of the obstacle has no relevant influence on the filter uncertainty, regardless of the value of the remaining factors. This behaviour can be observed in the histograms of figure A.20.

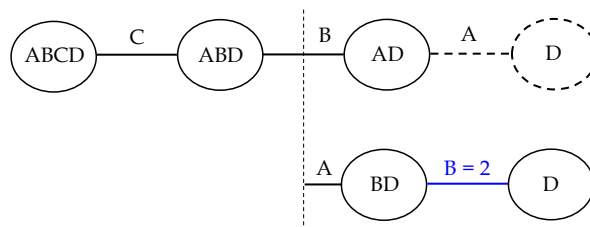


Figure A.19: Tree graph for the analysis of the effect of factor D on the expected uncertainty performance. The dashed node and arc correspond to a rejected conclusion due to multimodal population. The arc in blue represents a decision on the value of factor B based on a forced interaction.

Source	SS	df	MS	F	p-value
D	1.5675e-9	1	1.5675e-9	0.0024	0.9611
Within cells	0.00319	4848	6.5899e-7		

Table A.5: One-way ANOVA table for factor D given $B = 2$.

A.3 Convergence performance

According to the established procedure, the first step is to perform a four-way ANOVA for the data corresponding to the performance of convergence of the filter. These results are shown in table A.6.

A.3 Convergence performance

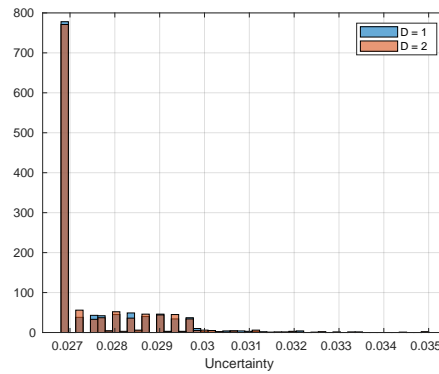


Figure A.20: Histograms for the conclusion of the effect of factor D on the expected uncertainty performance when $B = 2$, represented for the two levels of factor D.

Source	SS	df	MS	F	p-value
A	27.6910	1	27.6910	0.1702	0.6800
B	962747.0726	1	962747.0726	5917.3969	0.0000
C	1601490.7650	1	1601490.7650	9843.3501	0.0000
D	6.8851	1	6.8851	0.0423	0.8370
AxB	17.6499	1	17.6499	0.1085	0.7419
AxC	3.7469	1	3.7469	0.0230	0.8794
AxD	11.3538	1	11.3538	0.0698	0.7917
BxC	1255927.0726	1	1255927.0726	7719.3888	0.0000
BxD	672.7963	1	672.7963	4.1353	0.0421
CxD	61.5150	1	61.5150	0.3781	0.5387
AxBxC	78.7749	1	78.7749	0.4842	0.4866
AxBxD	35.0676	1	35.0676	0.2155	0.6425
AxCxD	17.1713	1	17.1713	0.1055	0.7453
BxCxD	0.0742	1	0.0742	0.0005	0.9830
AxBxCxD	20.9213	1	20.9213	0.1286	0.7199
Within cells	788758.6	4848	162.7		

Table A.6: Four-way ANOVA table for the convergence performance.

As before, the procedure begins with the analysis of the effect of factor A (initial position of the obstacle) on the filter convergence. The steps that have been followed are encoded in the tree graph of figure A.21 and the necessary ANOVA tables, in figure A.22. In this case, the complete set of conclusions is:

- Factor A has no effect on the convergence of the filter given that $B = 1$ and $C = 1$.
- Factor A has no effect given that $B = 1$ and $C = 2$.
- Factor A has no effect given that $B = 2$.

In order to complete this analysis correctly for the ANOVA assumptions, it has been necessary to force some interactions as indicated in figure A.21. With these results, it can be assured that the initial position of the obstacle has no influence on the convergence of the filter, regardless of the values of the remaining factors. Such behaviour can be observed in the histograms of figure A.23, where the difference in the mean is due to changes in the values of factors B and C, as discussed later on.

A.3 Convergence performance

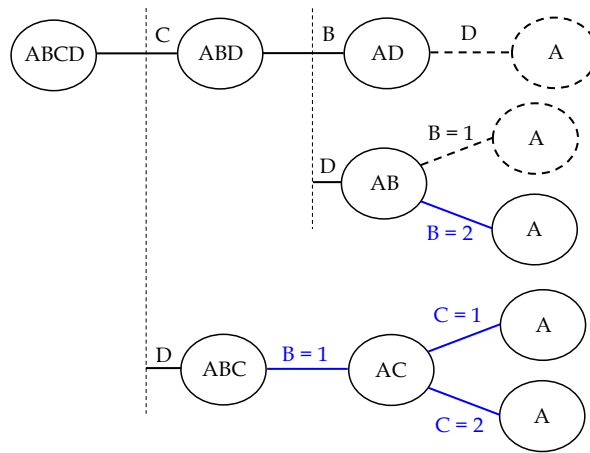


Figure A.21: Tree graph for the analysis of the effect of factor A on the convergence performance. Dashed nodes and arcs correspond to rejected conclusions due to multimodal population. Arcs in blue denote decisions on the value of factors based on forced interactions.

Source	SS	df	MS	F	p-value
A	35.2377	1	35.2377	0.2166	0.6417
Within cells	788758.6	4848	162.7		
(a)					
Source	SS	df	MS	F	p-value
A	23.7664	1	23.7664	0.1461	0.7023
Within cells	788758.6	4848	162.7		
(b)					
Source	SS	df	MS	F	p-value
A	44.778	1	44.778	0.2752	0.5999
Within cells	788758.6	4848	162.7		
(c)					

Figure A.22: ANOVA tables for the analysis of the effect of factor A (initial position of the obstacle) on the convergence. (a) One-way ANOVA for factor A given B = 1 and C = 1. (b) One-way ANOVA for factor A given B = 1 and C = 2. (c) One-way ANOVA for factor A given B = 2.

Now the case of factor B (amount of missing observations) is addressed. The corresponding analysis is encoded in the tree graph of figure A.24, and the ANOVA tables generated during the process are collected in figure A.25. These results lead to the following conclusions:

- Factor B has effect on the convergence of the filter given that C = 1.
- Factor B has no effect given that C = 2.

In this analysis, different omega squared measures have been used. For interaction BxC (see table A.6), $\hat{\omega}^2 = 0.6134$, thus it is considered relevant. For the case of main effect B with C = 1, $\hat{\omega}^2 = 0.7362$, which is also considered very relevant. However, as opposed to the result of ANOVA table in figure A.25 (b), main effect B with C = 2 is not considered relevant enough, because it has $\hat{\omega}^2 = 0.0119$. With these results we can affirm that the amount of missing observations in the sensor has an important impact on the convergence of the filter only when the number of biased

A.3 Convergence performance

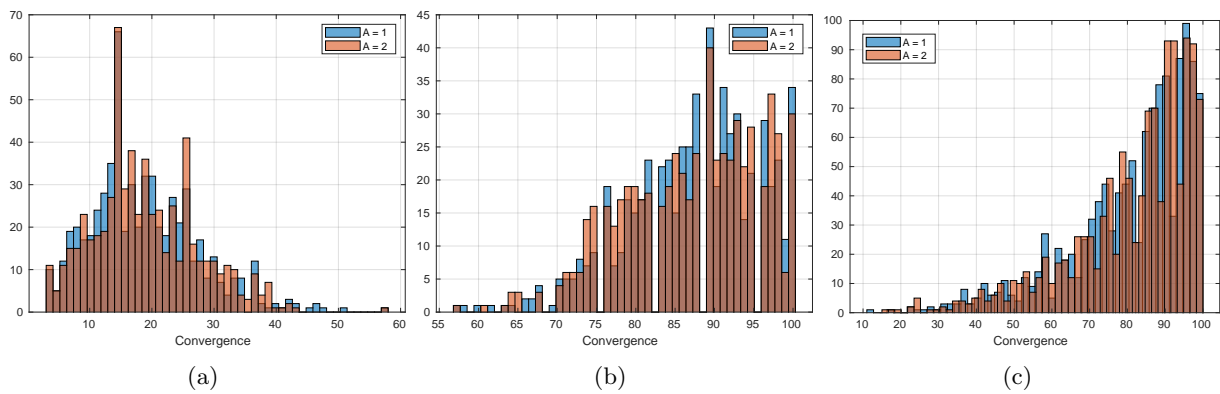


Figure A.23: Histograms for the conclusions about the effect of factor A on the convergence performance, represented for the two levels of such factor with additional restrictions on the population. (a) Factor B = 1 and factor C = 1. (b) Factor B = 1 and factor C = 2. (c) Factor B = 2.

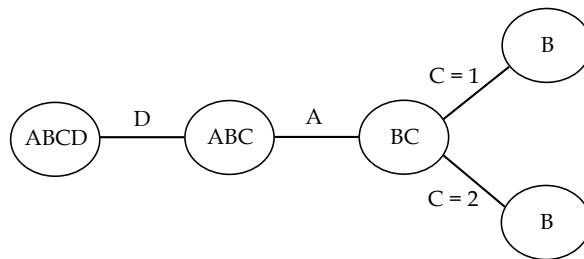


Figure A.24: Tree graph for the analysis of the effect of factor B on the convergence performance.

Source	SS	df	MS	F	p-value
B	2.2089e+6	1	2.2089e+6	1.3577e+4	0
Within cells	788758.6	4848	162.7		

(a)

Source	SS	df	MS	F	p-value
B	9728	1	9728	59.7919	1.2737e-14
Within cells	788758.6	4848	162.7		

(b)

Figure A.25: ANOVA tables for the analysis of the effect of factor B (amount of missing observations) on the convergence. (a) One-way ANOVA for factor B given C = 1. (b) One-way ANOVA for factor B given C = 2.

sensory observations is negligible. More specifically, an increase in the number of missing observations leads to a much slower convergence, as shown in the histograms of figure A.26 (a). This effect, however, nearly vanishes in the presence of biased observations (see histograms in figure A.26 (b)). There, the increase of missing observations only leads to a higher variance, but only as a secondary effect.

The effect of factor C (amount of biased observations) on the convergence performance is now discussed. The resulting tree graph for this case is depicted in figure A.27 and the generated ANOVA tables during the process are shown in figure A.28. The complete set of conclusions for this situation is:

A.3 Convergence performance

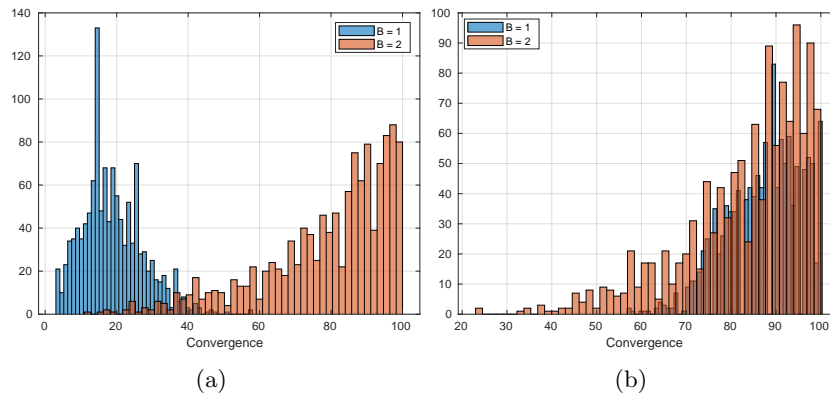


Figure A.26: Histograms for the conclusions about the effect of factor B on the convergence performance, represented for the two levels of such factor with additional restrictions on the population. (a) Factor C = 1. (b) Factor C = 2.

- Factor C has effect on the convergence of the filter given that B = 1.
- Factor C has no effect given that B = 2.

This analysis is very similar to the previous one. Omega squared values for both main effects of factor C indicated in the conclusions are $\hat{\omega}^2 = 0.7828$ and $\hat{\omega}^2 = 0.0129$ respectively, thus, the presence of biased observations in the sensor is relevant to the convergence of the filter only when there are no missing observations too. Also, in this case the increase of biased readings leads to a much slower convergence; the effect disappears when the presence of missing observations is high. These behaviours can be seen in the histograms of figures A.29 (a) and (b), respectively.

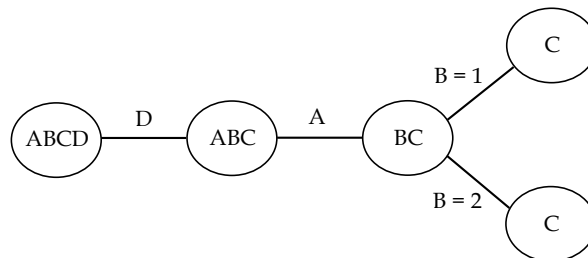


Figure A.27: Tree graph for the analysis of the effect of factor C on the convergence performance.

Finally, the report is focused on the analysis of the effect of factor D (speed of the obstacle). The process being followed is encoded in the tree graph of figure A.30 and the used ANOVA tables are shown in figure A.31. In this case, the complete set of conclusions is:

- Factor D has no effect on the convergence of the filter given that B = 1 and C = 1.
- Factor D has no effect given that B = 1 and C = 2.
- Factor D has no effect given that B = 2.

All the interactions referenced in figure A.30 have been forced. Note also that the BxD interaction has been considered irrelevant (see table A.6) because its associated omega squared value is $\hat{\omega}^2 = 6.4417e-4$. The obtained conclusions allows to affirm that the speed of the obstacle has no influence on the number of steps that lead to convergence in a filtering process, regardless

A.3 Convergence performance

Source	SS	df	MS	F	p-value
C	2.8469e+6	1	2.8469e+6	1.7498e+4	0
Within cells	788758.6	4848	162.7		

(a)

Source	SS	df	MS	F	p-value
C	10486.2	1	10486.2	64.4522	1.4626e-10
Within cells	788758.6	4848	162.7		

(b)

Figure A.28: ANOVA tables for the analysis of the effect of factor C (amount of biased observations) on the convergence. (a) One-way ANOVA for factor C given B = 1. (b) One-way ANOVA for factor C given B = 2.

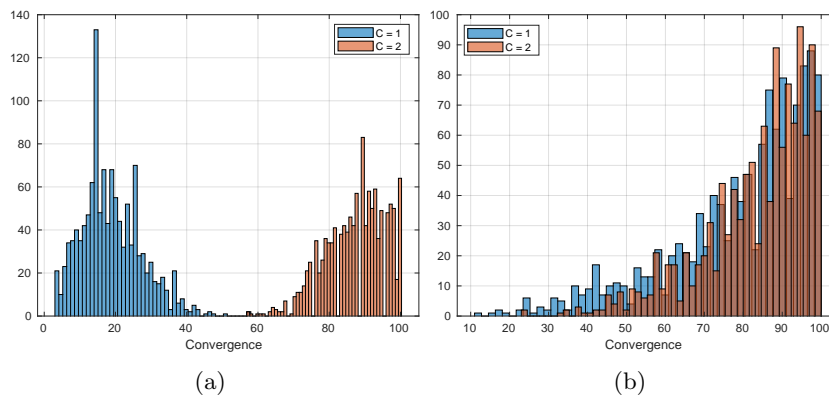


Figure A.29: Histograms for the conclusions about the effect of factor C on the convergence performance, represented for the two levels of such factor with additional restrictions on the population. (a) Factor B = 1. (b) Factor B = 2.

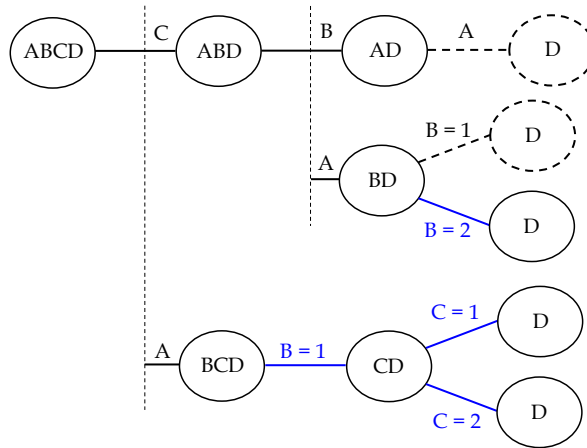


Figure A.30: Tree graph for the analysis of the effect of factor D on the convergence performance. Dashed nodes and arcs correspond to rejected conclusions due to multimodal population. Arcs in blue denote decisions on the value of factors based on forced interactions.

of the value of the remaining factors. This can be seen in the histograms of figure A.32, where the changes in the population mean are only due to the effect of factors B and C, as discussed before.

A.3 Convergence performance

Source	SS	df	MS	F	p-value
D	57.7508	1	57.7508	0.3550	0.5513
Within cells	788758.6	4848	162.7		

(a)

Source	SS	df	MS	F	p-value
D	246.961	1	246.961	1.5179	0.2180
Within cells	788758.6	4848	162.7		

(b)

Source	SS	df	MS	F	p-value
D	407.901	1	407.901	2.5071	0.1134
Within cells	788758.6	4848	162.7		

(c)

Figure A.31: ANOVA tables for the analysis of the effect of factor D (speed of the obstacle) on the convergence. (a) One-way ANOVA for factor D given B = 1 and C = 1. (b) One-way ANOVA for factor D given B = 1 and C = 2. (c) One-way ANOVA for factor D given B = 2.

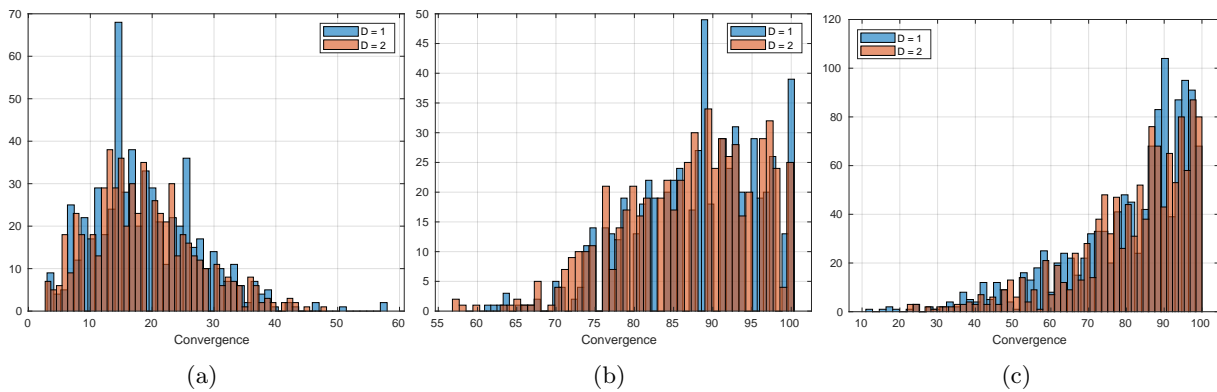


Figure A.32: Histograms for the conclusions about the effect of factor D on the convergence performance, represented for the two levels of such factor with additional restrictions on the population. (a) Factor B = 1 and factor C = 1. (b) Factor B = 1 and factor C = 2. (c) Factor B = 2.

References

- [1] J.-A. Fernández-Madrigal and J. L. Blanco Claraco, *Simultaneous localization and mapping for mobile robots: introduction and methods*, J. Gammon and H. Abelbeck, Eds. Hershey, Pennsylvania: IGI Global, 2013.
- [2] IFR, “International Federation of Robotics,” 2021. [Online]. Available: <https://ifr.org/>
- [3] V. Villani, F. Pini, F. Leali, C. Secchi, and C. Fantuzzi, “Survey on Human-Robot Interaction for Robot Programming in Industrial Applications,” *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 66–71, 2018.
- [4] M. Garduno-Aparicio, J. Rodriguez-Resendiz, G. Macias-Bobadilla, and S. Thenozhi, “A Multidisciplinary Industrial Robot Approach for Teaching Mechatronics-Related Courses,” *IEEE Transactions on Education*, vol. 61, no. 1, pp. 55–62, feb 2018.
- [5] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, “Continuum Robots for Medical Applications: A Survey,” *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, dec 2015.
- [6] E. Bauzano, B. Estebanez, I. Garcia-Morales, and V. F. Munoz, “Collaborative Human–Robot System for HALS Suture Procedures,” *IEEE Systems Journal*, vol. 10, no. 3, pp. 957–966, sep 2016.
- [7] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots, Second Edition*, 2011.
- [8] G. Griffiths, Ed., *Technology and Applications of Autonomous Underwater Vehicles*. CRC Press, 2002.
- [9] M. Hashimoto, F. Oba, and S. Zenitani, “Coordinative object-transportation by multiple industrial mobile robots using coupler with mechanical compliance,” in *Proceedings of IECON '93 - 19th Annual Conference of IEEE Industrial Electronics*. IEEE, pp. 1577–1582.
- [10] M. Luperto, J. Monroy, J. R. Ruiz-Sarmiento, F.-A. Moreno, N. Basilico, J. Gonzalez-Jimenez, and N. A. Borghese, “Towards Long-Term Deployment of a Mobile Robot for at-Home Ambient Assisted Living of the Elderly,” in *2019 European Conference on Mobile Robots (ECMR)*. IEEE, sep 2019, pp. 1–6.
- [11] M. Kyrarini, F. Lygerakis, A. Rajavenkatanarayanan, C. Sevastopoulos, H. R. Nambiappan, K. K. Chaitanya, A. R. Babu, J. Mathew, and F. Makedon, “A Survey of Robots in Healthcare,” *Technologies*, vol. 9, no. 1, p. 8, jan 2021.
- [12] A. Cruz-Martín, J. Fernández-Madrigal, C. Galindo, J. González-Jiménez, C. Stockmans-Daou, and J. Blanco-Claraco, “A LEGO Mindstorms NXT approach for teaching at Data Acquisition, Control Systems Engineering and Real-Time Systems undergraduate courses,” *Computers & Education*, vol. 59, no. 3, pp. 974–988, nov 2012. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0360131512000838>
- [13] S. S. H. Hajjaj and K. S. M. Sahari, “Review of Research in the Area of Agriculture Mobile Robots,” 2014, pp. 107–117. [Online]. Available: http://link.springer.com/10.1007/978-981-4585-42-2_{-}13
- [14] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, “Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges,” *IEEE Access*, vol. 7, pp. 48 572–48 634, 2019. [Online]. Available: <https://ieeexplore.ieee.org/document/8682048/>
- [15] H. Kang, J. Joung, J. Kim, J. Kang, and Y. S. Cho, “Protect Your Sky: A Survey of Counter Unmanned Aerial Vehicle Systems,” *IEEE Access*, vol. 8, pp. 168 671–168 710, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/9194729/>
- [16] G. Ding, Q. Wu, L. Zhang, Y. Lin, T. A. Tsiftsis, and Y.-D. Yao, “An Amateur Drone Surveillance System Based on the Cognitive Internet of Things,” *IEEE Communications Magazine*, vol. 56, no. 1, pp. 29–35, jan 2018. [Online]. Available: <https://ieeexplore.ieee.org/document/8255734/>

- [17] S. Waharte and N. Trigoni, "Supporting Search and Rescue Operations with UAVs," in *2010 International Conference on Emerging Security Technologies*. IEEE, sep 2010, pp. 142–147. [Online]. Available: <http://ieeexplore.ieee.org/document/5600072/>
- [18] D. Galar, U. Kumar, and D. Seneviratne, *Robots, Drones, UAVs and UGVs for Operation and Maintenance*. CRC Press, 2020. [Online]. Available: <https://www.taylorfrancis.com/books/9780429839184>
- [19] A. Pascoal, P. Oliveira, C. Silvestre, L. Sebastiao, M. Rufino, V. Barroso, J. Gomes, G. Ayela, P. Coince, M. Cardew, A. Ryan, H. Braithwaite, N. Cardew, J. Trepte, N. Seube, J. Champeau, P. Dhaussy, V. Sauce, R. Moitie, R. Santos, F. Cardigos, M. Brussieux, and P. Dando, "Robotic ocean vehicles for marine science applications: the European ASIMOV project," in *OCEANS 2000 MTS/IEEE Conference and Exhibition. Conference Proceedings (Cat. No.00CH37158)*, vol. 1. IEEE, pp. 409–415. [Online]. Available: <http://ieeexplore.ieee.org/document/881293/>
- [20] D. Terracciano, L. Bazzarello, A. Caiti, R. Costanzi, and V. Manzari, "Marine Robots for Underwater Surveillance," *Current Robotics Reports*, vol. 1, no. 4, pp. 159–167, dec 2020. [Online]. Available: <http://link.springer.com/10.1007/s43154-020-00028-z>
- [21] Y. Hirata, K. Kosuge, H. Asama, H. Kaetsu, and K. Kawabata, "Coordinated transportation of a single object by multiple mobile robots without position information of each robot," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, vol. 3. IEEE, pp. 2024–2029. [Online]. Available: <http://ieeexplore.ieee.org/document/895268/>
- [22] E. H. C. Harik, F. Guerin, F. Guinand, J.-F. Brethe, and H. Pelvillain, "UAV-UGV cooperation for objects transportation in an industrial area," in *2015 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, mar 2015, pp. 547–552. [Online]. Available: <https://ieeexplore.ieee.org/document/7125156>
- [23] Z. Feng, G. Hu, Y. Sun, and J. Soon, "An overview of collaborative robotic manipulation in multi-robot systems," *Annual Reviews in Control*, vol. 49, pp. 113–127, 2020. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1367578820300043>
- [24] A. A. Panchpor, S. Shue, and J. M. Conrad, "A survey of methods for mobile robot localization and mapping in dynamic indoor environments," in *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*. IEEE, jan 2018, pp. 138–144. [Online]. Available: <http://ieeexplore.ieee.org/document/8316333/>
- [25] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 856–892, jun 2020.
- [26] M. Castillo-Lopez, P. Ludivig, S. A. Sajadi-Alamdari, J. L. Sanchez-Lopez, M. A. Olivares-Mendez, and H. Voos, "A Real-Time Approach for Chance-Constrained Motion Planning With Dynamic Obstacles," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3620–3625, apr 2020.
- [27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT, 2005.
- [28] S. Zhang, L. Zheng, and W. Tao, "Survey and Evaluation of RGB-D SLAM," *IEEE Access*, vol. 9, pp. 21 367–21 387, 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9330596/>
- [29] R. Gomez-Ojeda, F.-A. Moreno, D. Zuniga-Noel, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 734–746, jun 2019. [Online]. Available: <https://ieeexplore.ieee.org/document/8680013/>
- [30] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrilu, and K. O. Arras, "Human motion trajectory prediction: a survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, jul 2020.
- [31] M. Castellano-Quero, J.-A. Fernández-Madrigal, and A.-J. García-Cerezo, "Statistical Study of the Performance of Recursive Bayesian Filters with Abnormal Observations from Range Sensors," *Sensors*, vol. 20, no. 15, p. 4159, jul 2020. [Online]. Available: <https://www.mdpi.com/1424-8220/20/15/4159>
- [32] O. Pettersson, "Execution monitoring in robotics: a survey," *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 73–88, 2005.
- [33] M. Castellano-Quero, J.-A. Fernández-Madrigal, and A. García-Cerezo, "Improving Bayesian inference efficiency for sensory anomaly detection and recovery in mobile robots," *Expert Systems with Applications*, vol. 163, p. 113755, jan 2021.
- [34] B. Cai, L. Huang, and M. Xie, "Bayesian networks in fault diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2227–2240, 2017.
- [35] H. Schneider and P. M. Frank, "Observer-based supervision and fault detection in robots using nonlinear and fuzzy logic residual evaluation," *IEEE Transactions on Control Systems Technology*, vol. 4, no. 3, pp. 274–282, 1996.

- [36] V. Verma, G. Gordon, R. Simmons, and S. Thrun, "Real-time fault diagnosis," *IEEE Robotics and Automation Magazine*, vol. 11, no. 2, pp. 56–66, 2004.
- [37] O. Petterson, L. Karlson, and A. Saffiotti, "Model-free execution monitoring in behavior-based mobile robotics," in *Conference on advanced Robotics (ICAR), Coimbra, Portugal*, 2003, pp. 864–869.
- [38] M. Castellano-Quero, J.-A. Fernández-Madrigal, and A. J. García-Cerezo, "Integrating multiple sources of knowledge for the intelligent detection of anomalous sensory data in a mobile robot," in *Robot 2019: Fourth Iberian Robotics Conference*, 2020, pp. 159–170.
- [39] J. Pearl, "Bayesian networks: a model of self-activated memory for evidential reasoning," in *Proceedings of the 7th Conference of the Cognitive Science Society*, 1985, pp. 329–334.
- [40] A. Rezaee, A. A. Raie, A. Nadi, and S. S. Ghidary, "Sensor fault tolerance method by using a Bayesian network for robot behavior," *Advanced Robotics*, vol. 25, no. 16, pp. 2039–2064, 2011.
- [41] J. Walls, A. Esterline, and A. Homaifar, "Sensor fusion using fuzzy integral and diverse Bayesian networks," in *IFAC Proceedings Volumes*, vol. 41, no. 2, 2008, pp. 10 421–10 426.
- [42] M. Boutayeb, H. Rafaralahy, and M. Darouach, "Convergence analysis of the extended Kalman filter used as an observer for nonlinear deterministic discrete-time systems," *IEEE Transactions on Automatic Control*, vol. 42, no. 4, pp. 581–586, apr 1997. [Online]. Available: <http://ieeexplore.ieee.org/document/566674/>
- [43] S. Wang, W. Wang, B. Chen, and C. K. Tse, "Convergence analysis of nonlinear Kalman filters with novel innovation-based method," *Neurocomputing*, vol. 289, pp. 188–194, may 2018. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0925231218301309>
- [44] T. Liu, Y. Wei, W. Yin, Y. Wang, and Q. Liang, "State estimation for nonlinear discrete-time fractional systems: A Bayesian perspective," *Signal Processing*, vol. 165, pp. 250–261, dec 2019. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0165168419302452>
- [45] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman Filtering With Intermittent Observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, sep 2004. [Online]. Available: <http://ieeexplore.ieee.org/document/1333199/>
- [46] E. Fertig, S.-J. Baek, B. Hunt, E. Ott, I. Szunyogh, J. Aravéquia, E. Kalnay, H. Li, and J. Liu, "Observation bias correction with an ensemble Kalman filter," *Tellus A: Dynamic Meteorology and Oceanography*, vol. 61, no. 2, pp. 210–226, jan 2009. [Online]. Available: <https://www.tandfonline.com/doi/full/10.1111/j.1600-0870.2008.00378.x>
- [47] V. Pathuri Bhuvana, C. Preissl, A. M. Tonello, and M. Huemer, "Multi-Sensor Information Filtering With Information-Based Sensor Selection and Outlier Rejection," *IEEE Sensors Journal*, vol. 18, no. 6, pp. 2442–2452, mar 2018. [Online]. Available: <http://ieeexplore.ieee.org/document/8245799/>
- [48] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, Massachusetts: The MIT Press, 2009.
- [49] S. Maxwell and H. Delaney, *Designing Experiments and Analyzing Data: A Model Comparison Perspective*, 2nd ed., D. Reigert, Ed. New Jersey: Lawrence Erlbaum Associates, 2004.
- [50] M. Castellano-Quero, J.-A. Fernández-Madrigal, and A.-J. García-Cerezo, "Hacia la diagnosis y fusion de sensores roboticos a bajo nivel mediante inferencia en redes bayesianas," in *XXXIX Jornadas de Automática, Badajoz (Spain)*, 2018, pp. 233–240.
- [51] M. Castellano-Quero, J.-A. Fernández-Madrigal, and A. García-Cerezo, "Interactive Construction of Bayesian Inference Networks for Robust Robot Sensorics," in *XIV Simposio CEA de Control Inteligente (SCI 2018)*, Málaga (Spain), 2018.
- [52] M. Castellano-Quero, M. Castillo-Lopez, V. Arévalo-Espejo, J.-A. Fernández-Madrigal, H. Voos, and A. García-Cerezo, "A Bayesian Sensory Architecture for Robust Pedestrian Detection and Safe Navigation in Human Environments," 2021.
- [53] M. Castellano-Quero, I. Fernández-Vega, J.-A. Fernández-Madrigal, and A. Cruz-Martín, "Metodos de inferencia bayesiana empotrados para el diagnostico y mejora sensoriales de un robot movil," in *II Jornadas de computación Empotrada y Reconfigurable*, Málaga (Spain), 2017, pp. 667–676.
- [54] S. L. Lauritzen and D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 50, no. 2, pp. 157–224, 1988.
- [55] C. Huang and A. Darwiche, "Inference in belief networks: a procedural guide," *International Journal of Approximate Reasoning*, vol. 15, no. 3, pp. 225–263, 1996. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0888613X96000692>



- [56] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and Intelligent Laboratory Systems*, vol. 39, no. 1, pp. 43–62, nov 1997. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0169743997000610>
- [57] K. P. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning," Ph.D. dissertation, University of California, Berkeley, 2002.
- [58] A. Darwiche, *Modeling and reasoning with Bayesian networks*. New York: Cambridge University Press, 2009, vol. 9780521884.
- [59] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. Elsevier, 1988. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/C20090276094>
- [60] U. N. Lerner, "Hybrid Bayesian Networks for Reasoning About Complex Systems," Ph.D. dissertation, Stanford University, 2003.
- [61] F. Jensen, S. Lauritzen, and K. Olsen, "Bayesian updating in recursive graphical models by local computation," *Computational Statistics Quarterly*, vol. 4, pp. 269–282, 1990.
- [62] S. L. Lauritzen, "Propagation of Probabilities, Means, and Variances in Mixed Graphical Association Models," *Journal of the American Statistical Association*, vol. 87, no. 420, p. 1098, dec 1992.
- [63] J. H. Kim and J. Pearl, "A computational model for causal and diagnostic reasoning in inference systems," in *Eighth international joint conference on Artificial intelligence*, 1983, pp. 190–193.
- [64] J. Pearl, "Evidential reasoning using stochastic simulation of causal models," *Artificial Intelligence*, vol. 32, no. 2, pp. 245–257, may 1987.
- [65] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721–741, 1984.
- [66] N. Yang, "The (Mis)use of One-Way ANOVA Testing in Sociology Research Papers and Their Implications," Ph.D. dissertation, Portland State University, aug 2019. [Online]. Available: <https://archives.pdx.edu/ds/psu/29373>
- [67] D. Fraiman and R. Fraiman, "An ANOVA approach for statistical comparisons of brain networks," *Scientific Reports*, vol. 8, no. 1, p. 4746, dec 2018. [Online]. Available: <http://www.nature.com/articles/s41598-018-23152-5>
- [68] L. Lista, *Statistical Methods for Data Analysis in Particle Physics*, ser. Lecture Notes in Physics. Cham: Springer International Publishing, 2016, vol. 909. [Online]. Available: <http://link.springer.com/10.1007/978-3-319-20176-4>
- [69] Y. W. Ng, J. C. Ren Jie, and S. Kamaruddin, "Analysis of Shop Floor Performance through Discrete Event Simulation: A Case Study," *Journal of Industrial Engineering*, vol. 2014, pp. 1–10, sep 2014. [Online]. Available: <https://www.hindawi.com/journals/jie/2014/878906/>
- [70] R. S. Witte and J. S. Witte, *Statistics*, 11th ed. Wiley, 2017.
- [71] J. Cohen, *Statistical power analysis for the behavioral sciences*, 2nd ed. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1988.
- [72] B. Behroozpour, P. A. M. Sandborn, M. C. Wu, and B. E. Boser, "Lidar System Architectures and Circuits," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 135–142, oct 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/8067701/>
- [73] F. de Ponte Müller, "Survey on Ranging Sensors and Cooperative Techniques for Relative Positioning of Vehicles," *Sensors*, vol. 17, no. 2, p. 271, jan 2017. [Online]. Available: <http://www.mdpi.com/1424-8220/17/2/271>
- [74] V. Kuts, T. Otto, T. Tähemaa, K. Bukhari, and T. Patariaia, "Adaptive Industrial Robots Using Machine Vision," in *Volume 2: Advanced Manufacturing*. American Society of Mechanical Engineers, nov 2018. [Online]. Available: <https://asmedigitalcollection.asme.org/IMECE/proceedings/IMECE2018/52019/Pittsburgh,Pennsylvania,USA/276346>
- [75] A. Garcia-Cerezo, A. Mandow, J. L. Martinez, J. Gomez-de Gabriel, J. Morales, A. Cruz, A. Reina, and J. Seron, "Development of ALACRANE: A Mobile Robotic Assistance for Exploration and Rescue Missions," in *2007 IEEE International Workshop on Safety, Security and Rescue Robotics*. IEEE, sep 2007, pp. 1–6. [Online]. Available: <http://ieeexplore.ieee.org/document/4381269/>
- [76] M. Luperto, J. Monroy, F.-Á. Moreno, J.-R. Ruiz-Sarmiento, N. Basílico, J. González, and N. A. Borghese, "A Multi-Actor Framework Centered Around an Assitive Mobile Robot for Elderly People Living Alone," in *IEEE International Conference on Intelligent Robots - Workshop on Robots for Assisted Living (IROS)*, 2018. [Online]. Available: <http://mapir.isa.uma.es/mapirwebsite/index.php/mapir-downloads/papers/286>

- [77] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering, Transactions of the ASME*, 1960.
- [78] D. B. Rubin, "The Calculation of Posterior Distributions by Data Augmentation: Comment: A Noniterative Sampling/Importance Resampling Alternative to the Data Augmentation Algorithm for Creating a Few Imputations When Fractions of Missing Information Are Modest: The SIR," *Journal of the American Statistical Association*, vol. 82, no. 398, pp. 543–546, jun 1987. [Online]. Available: <https://www.jstor.org/stable/2289460?origin=crossref>
- [79] A. Doucet, N. De Freitas, K. P. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic Bayesian networks," in *Sixteenth Conference on Uncertainty in Artificial Intelligence*. IEEE, 2000, pp. 176–183.
- [80] T. Dean and K. Kanazawa, "A model for reasoning about persistence and causation," *Artificial Intelligence*, vol. 93, no. 1-2, pp. 1–27, 1989.
- [81] H. Everett, *Sensors for Mobile Robots*. Wellesley, Massachusetts: A K Peters, Ltd, 1995.
- [82] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots, Second Edition*. The MIT Press, 2011.
- [83] R. B. Fisher and K. Konolige, "Range Sensors," in *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 521–542. [Online]. Available: http://link.springer.com/10.1007/978-3-540-30301-5_{_}23
- [84] G. Lombardo and P. McAdam, "Financial market frictions in a model of the Euro area," *Economic Modelling*, vol. 29, no. 6, pp. 2460–2485, nov 2012. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0264999312001976>
- [85] A. Alshareef, J. S. Giudice, J. Forman, D. F. Shedd, T. Wu, K. A. Reynier, and M. B. Panzer, "Application of trilateration and Kalman filtering algorithms to track dynamic brain deformation using sonomicrometry," *Biomedical Signal Processing and Control*, vol. 56, p. 101691, feb 2020. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1746809419302721>
- [86] U. von Toussaint, "Bayesian inference in physics," *Reviews of Modern Physics*, vol. 83, no. 3, pp. 943–999, sep 2011. [Online]. Available: <https://link.aps.org/doi/10.1103/RevModPhys.83.943>
- [87] M. Lin, J. Yoon, and B. Kim, "Self-Driving Car Location Estimation Based on a Particle-Aided Unscented Kalman Filter," *Sensors*, vol. 20, no. 9, p. 2544, apr 2020. [Online]. Available: <https://www.mdpi.com/1424-8220/20/9/2544>
- [88] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. [Online]. Available: <http://link.springer.com/10.1007/978-3-540-30301-5>
- [89] D. Fox, W. Burgard, and S. Thrun, "Markov Localization for Mobile Robots in Dynamic Environments," *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, nov 1999. [Online]. Available: <https://jair.org/index.php/jair/article/view/10246>
- [90] N. Bergman, *Recursive Bayesian Estimation - Navigation and Tracking Applications*, 1999.
- [91] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, jun 2001. [Online]. Available: <http://ieeexplore.ieee.org/document/938381/>
- [92] R. C. Smith and P. Cheeseman, "On the Representation and Estimation of Spatial Uncertainty," *The International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.
- [93] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *International Symposium Aerospace/Defense Sensing, Simulation and Controls*, I. Kadar, Ed., jul 1997, pp. 182–193. [Online]. Available: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.280797>
- [94] B. Gao, G. Hu, X. Zhu, and Y. Zhong, "A Robust Cubature Kalman Filter with Abnormal Observations Identification Using the Mahalanobis Distance Criterion for Vehicular INS/GNSS Integration," *Sensors*, vol. 19, no. 23, p. 5149, nov 2019. [Online]. Available: <https://www.mdpi.com/1424-8220/19/23/5149>
- [95] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "An outlier-robust Kalman filter," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, may 2011, pp. 1551–1558. [Online]. Available: <http://ieeexplore.ieee.org/document/5979605/>
- [96] R. Kumar, D. Castanon, E. Ermis, and V. Saligrama, "A new algorithm for outlier rejection in particle filters," in *2010 13th International Conference on Information Fusion*. IEEE, jul 2010, pp. 1–7. [Online]. Available: <http://ieeexplore.ieee.org/document/5712014/>

- [97] D.-J. Jwo and T.-S. Cho, "A practical note on evaluating Kalman filter performance optimality and degradation," *Applied Mathematics and Computation*, vol. 193, no. 2, pp. 482–505, nov 2007. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0096300307004316>
- [98] C. Jiang and S.-B. Zhang, "A Novel Adaptively-Robust Strategy Based on the Mahalanobis Distance for GPS/INS Integrated Navigation Systems," *Sensors*, vol. 18, no. 3, p. 695, feb 2018. [Online]. Available: <http://www.mdpi.com/1424-8220/18/3/695>
- [99] Devantech, "Devantech Limited corporate website," 2020. [Online]. Available: <http://www.robot-electronics.co.uk/htm/srf05tech.htm>
- [100] Sharp, "Sharp Corporation website," 2020. [Online]. Available: <http://global.sharp/products/device/lineup/selection/opto/haca/diagram.html>
- [101] Hokuyo, "Hokuyo Automatic Co. Ltd. corporate website," 2020. [Online]. Available: <https://www.hokuyo-aut.jp/search/single.php?serial=166>
- [102] J.-R. Ruiz-Sarmiento, C. Galindo, and J. González, "Experimental Study of the Performance of the Kinect Range Camera for Mobile Robotics," University of Malaga, Malaga (Spain), Tech. Rep., 2013. [Online]. Available: <http://mapir.isa.uma.es/cgalindo/papers/tech-kinect.pdf>
- [103] K. Khoshelham and S. O. Elberink, "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, feb 2012. [Online]. Available: <http://www.mdpi.com/1424-8220/12/2/1437>
- [104] Mesa Imaging, "SR4000 Product Specification," Mesa Imaging AG, Tech. Rep., 2020. [Online]. Available: https://www.inf.u-szeged.hu/sites/default/files/ipcglab/docs/vision/SR4000_{_}Data_{_}Sheet.pdf
- [105] N. S. Nise, *Control Systems Engineering*, 7th ed. Wiley, 2015.
- [106] J. Cohen and P. Cohen, *Applied multiple regression/correlation analysis for the behavioral sciences*. New Jersey: Lawrence Erlbaum Associates, 1983.
- [107] K. P. Murphy, "The Bayes Net Toolbox for Matlab," *Computing Science and Statistics*, 2001.
- [108] J.-A. Fernández-Madrigal and A. Cruz-Martín, "The CRUMB Project website," 2020. [Online]. Available: <https://babel.isa.uma.es/crumb/>
- [109] Open Source Robotics Foundation, "Turtlebot website," 2020. [Online]. Available: <https://www.turtlebot.com/turtlebot2/>
- [110] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, 2009.
- [111] C. Zhu, W. Q. Wang, H. Chen, and H. C. So, "Impaired sensor diagnosis, beamforming, and DOA estimation with difference co-array processing," *IEEE Sensors Journal*, vol. 15, no. 7, pp. 3773–3780, 2015.
- [112] X. Berjaga, Á. Pallares, and J. Melendez, "A framework for case-based diagnosis of batch processes in the principal components space," in *ETFA 2009 - IEEE Conference on Emerging Technologies and Factory Automation*, 2009.
- [113] O. Osoba, S. Mitaim, and B. Kosko, "Bayesian inference with adaptive fuzzy priors and likelihoods," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 5, pp. 1183–1197, 2011.
- [114] D. G. Giovanis, I. Papaioannou, D. Straub, and V. Papadopoulos, "Bayesian updating with subset simulation using artificial neural networks," *Computer Methods in Applied Mechanics and Engineering*, vol. 319, pp. 124–145, 2017.
- [115] J. Vila-Frances, J. Sanchis, E. Soria-Olivas, A. Jose-Serrano, M. Martinez-Sober, C. Bonanad, and S. Ventura, "Expert system for predicting unstable angina based on Bayesian networks," *Expert Systems with Applications*, vol. 40, no. 12, pp. 5004–5010, 2013.
- [116] B. Drury, J. Valverde-Rebaza, M. F. Moura, and A. de Andrade Lopes, "A survey of the applications of Bayesian networks in agriculture," *Engineering Applications of Artificial Intelligence*, vol. 65, pp. 29–42, 2017.
- [117] A. Azar and K. Mostafae Dolatabad, "A method for modelling operational risk with fuzzy cognitive maps and Bayesian belief networks," *Expert Systems with Applications*, vol. 115, pp. 607–617, 2019.
- [118] H. S. Sousa, F. Prieto-Castrillo, J. C. Matos, J. M. Branco, and P. B. Lourenço, "Combination of expert decision and learned based Bayesian networks for multi-scale mechanical analysis of timber elements," *Expert Systems with Applications*, vol. 93, pp. 156–168, 2018.
- [119] H. Zhou and S. Sakane, "Sensor planning for mobile robot localization - A hierarchical approach using a Bayesian network and a particle filter," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 481–487, 2008.



- [120] H. J. Min, "Navigation of a mobile robot using behavior network with Bayesian inference," in *IEEE International Conference on Mechatronics and Automation, ICMA 2005*, 2005, pp. 1479–1484.
- [121] D. Song, C. H. Ek, K. Huebner, and D. Kragic, "Task-based robot grasp planning using probabilistic inference," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 546–561, 2015.
- [122] E. Besada-Portas, J. A. López-Orozeo, and J. M. De La Cruz, "Unified fusion system based on Bayesian networks for autonomous mobile robots," in *Proceedings of the 5th International Conference on Information Fusion, FUSION 2002*, vol. 2, 2002, pp. 873–880.
- [123] P. Bessiere, "Probability as an alternative to logic for rational sensory-motor reasoning and decision," in *Springer Tracts in Advanced Robotics*, 2008, vol. 46, pp. 3–18.
- [124] H. B. Jun and D. Kim, "A Bayesian network-based approach for fault analysis," *Expert Systems with Applications*, vol. 81, pp. 332–348, 2017.
- [125] O. J. Mengshoel, A. Darwiche, and S. Uckun, "Sensor validation using Bayesian networks," in *9th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2008.
- [126] A. Choi and A. Darwiche, "An edge deletion semantics for belief propagation and its practical impact on approximation quality," in *Proceedings of the National Conference on Artificial Intelligence*. AAAI Press, Menlo Park, CA, 2006.
- [127] P. de Oude, "Modular Bayesian networks: reasoning, verification and model inaccuracies," Ph.D. dissertation, University of Amsterdam, 2010.
- [128] K. S. Hwang and S. B. Cho, "Landmark detection from mobile life log using a modular Bayesian network model," *Expert Systems with Applications*, vol. 36, no. 10, pp. 12065–12076, 2009.
- [129] S. Eastwood and S. Yanushkevich, "Risk profiler in automated human authentication," in *IEEE SSCI 2014 - IEEE Symposium Series on Computational Intelligence - CIES 2014: 2014 IEEE Symposium on Computational Intelligence for Engineering Solutions, Proceedings*, 2014, pp. 140–147.
- [130] I. Sassi, A. Gouin, and J. M. Thiriet, "Distributed to embedded Bayesian network for diagnosis of a networked robot," in *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications, CIVEMSA 2017 - Proceedings*, 2017, pp. 19–23.
- [131] K. Li, Y. Ren, D. Fan, L. Liu, Z. Wang, and Z. Ma, "Enhance GO methodology for reliability analysis of the closed-loop system using cyclic Bayesian networks," *Mechanical Systems and Signal Processing*, vol. 113, pp. 237–252, 2018.
- [132] M. Kłopotek, "Cyclic bayesian networks - Markov process approach," in *Systemy i technologie informacyjne*, 2006.
- [133] A. L. Tulupyev and S. I. Nikolenko, "Directed cycles in bayesian belief networks: probabilistic semantics and consistency checking complexity," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3789 LNAI, 2005, pp. 214–223.
- [134] R. K. Niven, B. R. Noack, E. Kaiser, L. Cattafesta, L. Cordier, and M. Abel, "Bayesian cyclic networks, mutual information and reduced-order Bayesian inference," in *AIP Conference Proceedings*, vol. 1757, 2016.
- [135] T. F. Coleman and Y. Li, "On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds," *Mathematical Programming*, vol. 67, no. 1-3, pp. 189–224, 1994.
- [136] —, "An interior trust region approach for nonlinear minimization subject to bounds," *SIAM Journal on Optimization*, vol. 6, no. 2, pp. 418–445, 1996.
- [137] J. A. Fernández-Madrigal and J. González, "Multihierarchical graph search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 103–113, 2002.
- [138] C. Galindo, J.-A. Fernández-Madrigal, and J. González, *Multiple abstraction hierarchies for mobile robot operation in large environments*, 2007.
- [139] P. Eades and K. Sugiyama, "How to draw a directed graph," *Journal of Information Processing*, vol. 13, no. 4, pp. 424–437, 1990.
- [140] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall PTR, 1998.
- [141] P. Healy and N. S. Nikolov, "How to layer a directed acyclic graph," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2265 LNCS, 2002, pp. 16–30.
- [142] S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, *Algorithms*, 2006.

- [143] R. Sedgewick and K. Wayne, *Algorithms*, 2014.
- [144] E. Ferrera, J. Capitán, A. R. Castaño, and P. J. Marrón, “Decentralized safe conflict resolution for multiple robots in dense scenarios,” *Robotics and Autonomous Systems*, vol. 91, pp. 179–193, may 2017.
- [145] H. Possegger, T. Mauthner, P. M. Roth, and H. Bischof, “Occlusion Geodesics for Online Multi-object Tracking,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, OH, USA: IEEE, jun 2014, pp. 1306–1313.
- [146] F. Beck and M. Bader, “Map Based Human Motion Prediction for People Tracking,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, nov 2019, pp. 1–7.
- [147] M. Dimitrievski, P. Veelaert, and W. Philips, “Behavioral Pedestrian Tracking Using a Camera and LiDAR Sensors on a Moving Vehicle,” *Sensors*, vol. 19, no. 2, p. 391, jan 2019.
- [148] P. Peng, Y. Tian, Y. Wang, J. Li, and T. Huang, “Robust multiple cameras pedestrian detection with multi-view Bayesian network,” *Pattern Recognition*, vol. 48, no. 5, pp. 1760–1772, may 2015.
- [149] X. Zhang, H.-M. Hu, F. Jiang, and B. Li, “Pedestrian detection based on hierarchical co-occurrence model for occlusion handling,” *Neurocomputing*, vol. 168, pp. 861–870, nov 2015.
- [150] A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, “Computer vision and deep learning techniques for pedestrian detection and tracking: A survey,” *Neurocomputing*, vol. 300, pp. 17–33, jul 2018.
- [151] W.-C. Ma, D.-A. Huang, N. Lee, and K. M. Kitani, “Forecasting Interactive Dynamics of Pedestrians with Fictitious Play,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017, pp. 4636–4644.
- [152] S. Kim, S. J. Guy, W. Liu, D. Wilkie, R. W. Lau, M. C. Lin, and D. Manocha, “BRVO: Predicting pedestrian trajectories using velocity-space reasoning,” *The International Journal of Robotics Research*, vol. 34, no. 2, pp. 201–217, feb 2015.
- [153] S. Hoermann, D. Stumper, and K. Dietmayer, “Probabilistic long-term prediction for autonomous vehicles,” in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, jun 2017, pp. 237–243.
- [154] G. Agamennoni, J. I. Nieto, and E. M. Nebot, “Estimation of Multivehicle Dynamics by Considering Contextual Information,” *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 855–870, aug 2012.
- [155] C. Blaiotta, “Learning Generative Socially Aware Models of Pedestrian Motion,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3433–3440, oct 2019.
- [156] M. Luber, G. Diego Tipaldi, and K. O. Arras, “Place-dependent people tracking,” *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 280–293, mar 2011.
- [157] A. Abrantes, J. Marques, and J. Lemos, “Long term tracking using Bayesian networks,” in *Proceedings. International Conference on Image Processing*, vol. 1. IEEE, 2002, pp. 609–612.
- [158] “Official Web Page of the EU FP7 research project: Social Situation-aware Perception and action for cognitive robots (SPENCER).” [Online]. Available: <http://spencer.eu/>
- [159] K. O. Arras, O. M. Mozos, and W. Burgard, “Using Boosted Features for the Detection of People in 2D Range Data,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, apr 2007, pp. 3402–3407. [Online]. Available: <http://ieeexplore.ieee.org/document/4209616/>
- [160] T. Linder and K. O. Arras, “People Detection, Tracking and Visualization Using ROS on a Mobile Service Robot,” 2016, pp. 187–213. [Online]. Available: http://link.springer.com/10.1007/978-3-319-26054-9_{_}8
- [161] M. Aguilar-Moreno, A. Cruz-Martín, and J.-A. Fernández-Madriral, “Modelado cinemático y simulación realista del manipulador móvil Turtlebot-2 + Widow-X en ROS,” in *XXXIX Jornadas de Automática, Badajoz (Spain)*, 2018, pp. 262–269.
- [162] D. Helbing and P. Molnár, “Social force model for pedestrian dynamics,” *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, may 1995. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevE.51.4282>
- [163] K. P. Murphy, Y. Weiss, and M. I. Jordan, “Loopy belief propagation for approximate inference: an empirical study,” in *UAI 99: Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, 1999, pp. 467–475.
- [164] G. Kokolakis and P. Nanopoulos, “Bayesian multivariate micro-aggregation under the Hellinger’s distance criterion,” *Research in Official Statistics*, vol. 4, no. 1, pp. 117–126, 2001.
- [165] C. Yuan and M. J. Druzdzel, “Importance sampling algorithms for Bayesian networks: principles and performance,” *Mathematical and Computer Modelling*, vol. 43, no. 9-10, pp. 1189–1207, 2006.
- [166] J. Chapmann, *Neural Networks: Introduction to Artificial Neurons, Backpropagation Algorithms and Multilayer Feedforward Networks*, 2017.

- [167] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963. [Online]. Available: <http://epubs.siam.org/doi/10.1137/0111030>
- [168] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525–533, jan 1993. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0893608005800565>
- [169] L. Pardo, *Statistical Inference Based on Divergence Measures*. New York: Chapman and Hall/CRC, 2006.