

Mobile Robot Lab Project to Introduce Engineering Students to Fault Diagnosis in Mechatronic Systems

Jesús M. Gómez-de-Gabriel, Anthony Mandow, Jesús Fernández-Lozano
and Alfonso García-Cerezo

*Universidad de Málaga – Andalucía Tech
Departamento de Ingeniería de Sistemas y Automática, 29071 Málaga, Spain.
Email: jmgomez[@]uma.es*

Abstract - This paper proposes lab work for learning fault detection and diagnosis (FDD) in mechatronic systems. These skills are important for engineering education because FDD is a key capability of competitive processes and products. The intended outcome of the lab work is that students become aware of the importance of faulty conditions and learn to design FDD strategies for a real system. To this end, the paper proposes a lab project where students are requested to develop a discrete event dynamic system (DEDS) diagnosis to cope with two faulty conditions in an autonomous mobile robot task. A sample solution is discussed for LEGO Mindstorms NXT robots with LabVIEW. This innovative practice is relevant to higher education engineering courses related to mechatronics, robotics, or DEDS. Results are also given of the application of this strategy as part of a postgraduate course on fault-tolerant mechatronic systems.

Keywords: control engineering education; discrete event systems; educational courses; electronic engineering education; fault diagnosis; fault tolerant control; mechanical engineering; mechatronics; mobile robots; DEDS diagnosis; FDD learning; LEGO Mindstorms NXT robots ;LabVIEW; autonomous mobile robot task; competitive process; discrete event dynamic system diagnosis..

This document is a self-archiving copy of the **accepted version** of the paper.
Please find the final published version in IEEEXplore: <http://dx.doi.org/10.1109/TE.2014.2358551>

Citation Information:

Gómez-de-Gabriel, J.M.; Mandow, A.; Fernández-Lozano, J.; García-Cerezo, A.,
"Mobile Robot Lab Project to Introduce Engineering Students to Fault Diagnosis in Mechatronic Systems,"
IEEE Transactions on Education, vol.58, no.3, pp.187-193, Aug. 2015
doi: 10.1109/TE.2014.2358551

```
@ARTICLE{Gomez:TE2015,  
author={J. M. G\{o}mez-de-Gabriel and A. Mandow and J. Fern\{a}ndez-Lozano and and A. Garc\{i}a-Cerezo},  
title={Mobile Robot Lab Project to Introduce Engineering Students to Fault Diagnosis in Mechatronic Systems},  
journal={IEEE Transactions on Education},  
year={2015},  
volume={58},  
number={3},  
pages={187-193}  
}
```

Mobile Robot Lab-Work to Introduce Engineering Students to Fault Diagnosis in Mechatronic Systems

Jesús M. Gómez-de-Gabriel, *Member, IEEE*, Anthony Mandow, *Member, IEEE*,
Jesús Fernández-Lozano, and Alfonso García-Cerezo, *Member, IEEE*

Abstract—The paper proposes lab-work for learning fault detection and diagnosis (FDD) in mechatronic systems. These skills are important for engineering education because FDD is a key capability of competitive processes and products. The intended outcome of the lab-work is that students become aware of the importance of faulty conditions and learn to design FDD strategies for a real system. To this end, the paper proposes a lab project where students are requested to develop discrete event dynamic system (DEDS) diagnosis to cope with two faulty conditions in an autonomous mobile robot task. An example solution is discussed for Lego Mindstorms NXT robots with LabVIEW. This innovative practice can be significant for higher education engineering courses related to mechatronics, robotics, or DEDS. Results are offered from the application of this strategy as part of a postgraduate course on fault tolerant mechatronic systems.

Index Terms—Fault detection and diagnosis, project-based learning, mobile robots, engineering education, higher education, LabVIEW, Lego Mindstorms NXT, mechatronics.

I. INTRODUCTION

Fault tolerant techniques contribute to improving robustness and performance in a broad scope of applications [1], which include automated sequential manufacturing systems [2] and safety-critical systems such as automobiles [3] and aircraft [4]. In particular, fault detection and diagnosis (FDD) is an essential part of active fault tolerant control systems [5]. Thus, engineering education in FDD is relevant because it is demanded from industry and is an open research field [6].

Most educational efforts on FDD have been related to monitoring in industrial maintenance [7], where the goal is learning to identify faults that are purposefully induced in lab equipment, such as vibrations in electric motors [8] [9] [10] and hardware errors in electronic circuits [11] [12]. However, not so many works have addressed the design of fault tolerant systems from the multidisciplinary perspective of mechatronics [13]. Tan *et al.* improved lab-work on vibration monitoring by asking students to program a sensor-based intelligent fault diagnoser [14]. Furthermore, the relevance of safety-critical systems for fault tolerance education was put forward by Bertoni *et al.*, who proposed project-based learning with ultra-light unmanned aerial systems [15]. In this sense, Katzourakis *et al.* identified the educational potential of low-cost small-scale vehicles for fault tolerant design [3].

The authors are grateful to Dr. J. A. Fernández-Madrigal for his helpful cooperation. This work was partially supported by the Spanish CI-CYT project DPI2011-22443. The authors are with Universidad de Málaga, Andalucía Tech, Departamento de Ingeniería de Sistemas y Automática, 29071 Málaga, Spain (e-mail: jmgomez@uma.es; amandow@uma.es; jesus.fernandez@isa.uma.es; ajgarcia@uma.es).

Mobile robots are another significant example of safety-critical autonomous system that should be designed to tolerate failing execution and uncertainty [16] [17]. Precisely, the reliability demanded by challenging applications [18] has fostered a growing research interest in FDD at different operation levels: steering mechanisms [19], anomalous actuator and wheel response [20], collision detection in spite of sensor limitations [21], localization failures due to abrupt wheel slippage [22], multi-robot performance [23], and monitoring of task states and mission goals [24].

Besides, mobile robots have become a core lab system where students can integrate engineering knowledge from different subjects [25]. While industrial-size vehicles can be used for lab demonstrations and capstone courses [26], cost and other practical reasons have favored small-sized systems [27]. Among these, Lego Mindstorms are arguably the most widespread and have been adopted for higher education practices on subjects such as control algorithms [28], data acquisition and real time systems [29], robot localization with the extended Kalman filter [30], and fuzzy logic [31].

This paper proposes a mobile robot lab-work project for learning model-based FDD using discrete event dynamic systems (DEDS) [32] [33]. This project seeks to make students aware of the importance of faulty conditions in mechatronic systems and to provide knowledge for designing a fault diagnoser in a representative application. A project solution example is offered for an autonomous task with a Lego mobile robot and LabVIEW programming. Furthermore, this project-based learning approach has been implemented for a Master in the University of Málaga (UMA), where results have been analyzed.

The paper is organized as follows. Section II discusses the design of the lab-work project. Section III summarizes the essential DEDS-based FDD concepts required for the project. Section IV gives a project solution example. Section V discusses results from the application of the proposed lab-work. The paper is closed by the conclusions.

II. LAB-WORK DESIGN

A. Motivation and intended outcomes

The major motivation for this work is to provide a practical introduction to FDD for engineering students. Since FDD requires insight into the target system, project-based learning is more appropriate for focussing on a single application than independent exercises.

Two major decisions have been taken in the design of the lab-work project. First, different FDD mechanisms are

described in literature [6]. Many technical systems can be considered as hybrid in the sense that they exhibit both continuous time and discrete event phenomena [34]. Thus, DEDS can be used to model many systems for diagnostic purposes [35]. Furthermore, finite state machines (FSM) are a familiar DEDS representation for engineering students. Therefore, an FSM-based approach [33] is suitable for a practical introduction.

Second, since fault tolerance is closely linked to reliability under real world operation conditions [18], lab plants are preferable over computer simulations. In this regard, educational flexible manufacturing systems are an interesting application, but they are expensive and pose the risk of complex sequential control taking on major importance over FDD. Alternatively, compact and inexpensive small-scale mobile robots are appropriate for teaching FDD because they are: *i*) a significant case of autonomous system demanding reliable operation; *ii*) motivating and familiar for students; and *iii*) affordable, if not already acquired for other courses, so working individually or in small groups is possible.

The intended learning outcomes of the lab-work are:

- LO1. The student will become aware of the importance of faulty conditions in the design of reliable mechatronic systems.
- LO2. The student will be able to design, implement and test an FSM-based FDD for a given technical system.

B. Proposed lab-work project

Mobile robot kits allow fast development of simple autonomous tasks such as line following, goal seeking or taking objects out of an area [36]. In the proposed project, students are asked to build a system that will remove three identical objects (e.g., soda cans) from a white area delimited by a black circumference (see Fig. 1). Initially, these objects and the robot are placed randomly inside the area. Each student has to develop an FSM-based navigation strategy and test it to find possible faults. Then, the student will have to draw from FDD theory (see Section III) to design a diagnoser for at least two fault conditions. An example solution for this lab-work is given in Section IV.

As a general rule, the project can be developed through six two-hour lab sessions, even though this may be adapted to the background of the students:

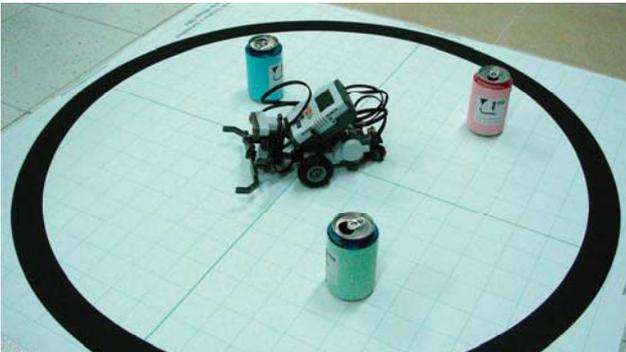


Fig. 1. Lab-work project: the robot has to take three objects outside of a black circumference.

- 1) A guided introduction to implementing FSM in the target robot system.
- 2) Robot design and building. Students select and integrate sensors and actuators from a kit to achieve the task goal.
- 3) Design, implementation and test of the FSM navigation strategy without considering faults.
- 4) Finding faults in the system and designing a diagnoser.
- 5) Programming the diagnoser and testing solutions.
- 6) Debugging and improvements of the solution.

When the lab-work has been completed, students will write a paper-formatted report with critical insight and will make a presentation with a demo to favor discussion about strategies and implementations.

III. DISCRETE EVENT SYSTEM DIAGNOSIS

This section reviews the basic concepts of a systematic FDD procedure based on DEDS [33]. This methodology allows for modular decomposition of complex systems, so it is also suitable for student lab-work projects with simpler plants. A state of the art in discrete event FDD can be found in [37].

The behavior of a system can be modeled as an FSM:

$$G = (Q, \Sigma, \delta, x_0), \quad (1)$$

where Q represents the state space, Σ is the set of events, δ is the transition function, and x_0 is the initial state. The set of events is defined by $\Sigma = \Sigma_o \cup \Sigma_{uo}$, as it includes both observable Σ_o and unobservable Σ_{uo} events. This distinction is relevant because failure events are not detected by sensors; i.e., $\Sigma_f \subset \Sigma_{uo}$, where $\Sigma_f = \{F_1, \dots, F_m\}$ is the set of possible failure event types.

Then, a diagnoser is defined as a new FSM:

$$G_d = (Q_d, \Sigma_o, \delta_d, q_0), \quad (2)$$

which is built from G . Diagnoser states Q_d are constructed as $q_d = \{(x_1, l_1), \dots, (x_n, l_n)\}$ where x_i are states from Q and l_i are labels. These labels can be $\{N\}$ to mean no failure or $\{F_{i1}, \dots, F_{ik}\}$ to specify that at least one of $k \geq 1$ failure types has happened to reach state x_i .

Thus, the initial diagnoser state is $q_0 = \{(x_0, \{N\})\}$. Assigning labels and defining the transition function δ_d requires a label propagation function LP which labels each state in Q based on past events. As only observable events can be considered in G_d , states in Q with unobservable transitions are collapsed into a single diagnoser state q_d . In order to unequivocally identify a failure condition, all labels in the corresponding q_d state should be failure types.

This approach is illustrated by the example in Fig. 2(a), where $Q = \{0, 1, 2, 3, 4\}$, $\Sigma_o = \{a, b, c\} \in$, and $\Sigma_{uo} = \{\sigma\}$. The transitions in the corresponding diagnoser, shown in Fig. 2(b), are based only on observable events. As a result, reaching diagnoser state $\{(1, N), (3, F_1)\}$ implies that the current G state can be either 1 in a normal condition or 3 with failure types F_1 . In this example, states 1 and 3 have been collapsed into a unique diagnoser state. Furthermore, failure type F_1 is propagated down to $\{(3, F_1)\}$, which is a failure-only state in the diagnoser. Thus, according to the diagnoser, an observable event sequence $[a, c]$ reveals that σ has happened. However,

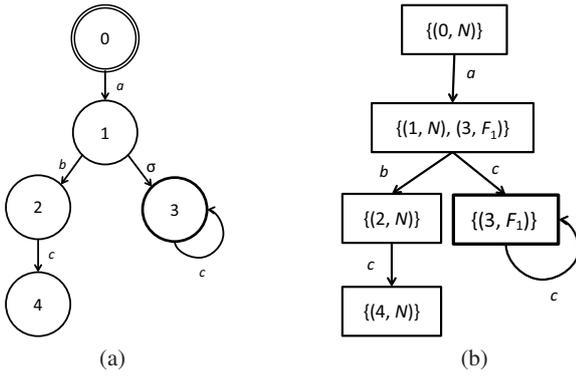


Fig. 2. FDD illustration. a) System model: G ; b) Diagnoser: G_d .

sequence $[a, b, c]$ leads to normal state 4. Failure states in G and G_d are emphasized in Fig. 2.

IV. PROJECT SOLUTION

This section is meant to serve as a guide for educators interested in adopting the proposed approach. A solution example is given of how a student could develop the project proposed in section II-B using a standard Lego NXT differential drive robot configuration (see Fig. 1) and LabVIEW with the Lego NXT Toolkit add-on.

A. Robot building

The first step is designing and building a robot, which can include a front passive fork to hold pushed objects. The Lego sensors considered in this solution are a front sonar, which measures the distances to objects in the range from 4 cm to 140 cm, and an analog light sensor oriented to the floor to detect the black circumference. As for the actuators, two servo motors with optical incremental encoders allow differential drive locomotion and odometric estimations.

B. FSM Navigation Design

A mobile robot navigation strategy can be designed as an FSM [36] [38], as shown in Fig. 3(a). Initially, the system *waits* for the user to press the start button. Then, the robot *searches* for surrounding objects by turning. Once an object has been detected with the sonar, the robot moves in a straight line to approach and *push* the object beyond the black border, which is detected through a threshold in the light sensor reading. Next, *reverse* motion gets the robot back to the bounded area so that it can start a new search. When the counter n of successfully removed objects reaches three, the robot signals its *success* (e.g., with a victory dance) and stops.

C. Navigation Strategy Implementation

The FSM model in Fig. 3(a) can be easily translated to a programming language to implement a state-based navigation strategy. In LabVIEW, FSMs can be programmed systematically [39] as a main *while loop* that contains a *case structure* with a *case* for every state. Another *while loop* within the state checks for transition events. Moreover, the *case structure*

includes the selection of the next state, which is passed to the next iteration through a *shift register*.

Fig. 4(a) presents an example of this systematic LabVIEW implementation, where an additional *shift register* is used to keep the count of objects to be cleared. In particular, the figure shows the *case* for state 2 (*push*) of Fig. 3(a), where the vehicle starts moving forward until the border is detected by the light sensor (*border reached* event). Then, motion is stopped and state 3 is issued as the next state. The remaining states can be programmed by adding *cases* with a similar structure.

D. Fault Detection and Diagnosis

Next, students have to analyze the behavior of their robots and are asked to find at least two possible faults. In this example, the following faults have been considered:

- *Object is lost* (F_1), which can happen due to a wrong approaching trajectory or the object slipping out of the fork. In the lab, this fault usually occurs spontaneously but it can also be provoked by removing the object before the robot takes it out of the perimeter.
- *Border is missed* (F_2), which can be caused by an error in the light sensor or by illumination conditions. This fault can be provoked in the lab by placing a white sheet of paper over the border line.

Thus, students have to extend the original FSM with two new states to account for these faults, as shown in Fig. 3(b). F_1 and F_2 are reached through unobservable events σ_1 and σ_2 , respectively. However, once these states have been reached, observable events can be identified as an evidence of the occurrence of σ_1 and σ_2 . In this solution example, events f , based on the sonar reading, and g , which uses circle diameter D as a threshold of traveled distance, are observable evidences. The transitions caused by observable events have been added to Fig. 3(b) as dotted lines.

As described in section III, a diagnoser is built from the extended FSM (see Fig. 3(c)). In this diagnoser, an event sequence $[a, b]$ leads to a state $\{(2, N), (5, F_1), (6, F_2)\}$ where the system could be either in normal state $(2, N)$ or in failure states $(5, F_1)$ or $(6, F_2)$. This makes use of events f and g , which serve to isolate fault states $(5, F_1)$ and $(6, F_2)$ in the diagnoser.

E. FDD Implementation

In order to obtain a navigation strategy with FDD, the navigation program developed in section IV-C is combined with the diagnoser, which implies adding fault diagnosis states in Fig. 3(c). As a result, the implementation of some states (i.e., 0, 1, 3, and 4) remains unchanged. States 5 and 6 are programmed as new cases that display the corresponding diagnosis result. As for state 2, the original code is extended with the transitions to the fault states, as shown in Fig. 4(b). The inner while loop includes checking for events f and g . As g is based on traveled distance, the wheel counter is now reset when entering the state. Moreover, the selection of the next state is made from transition event conditions.

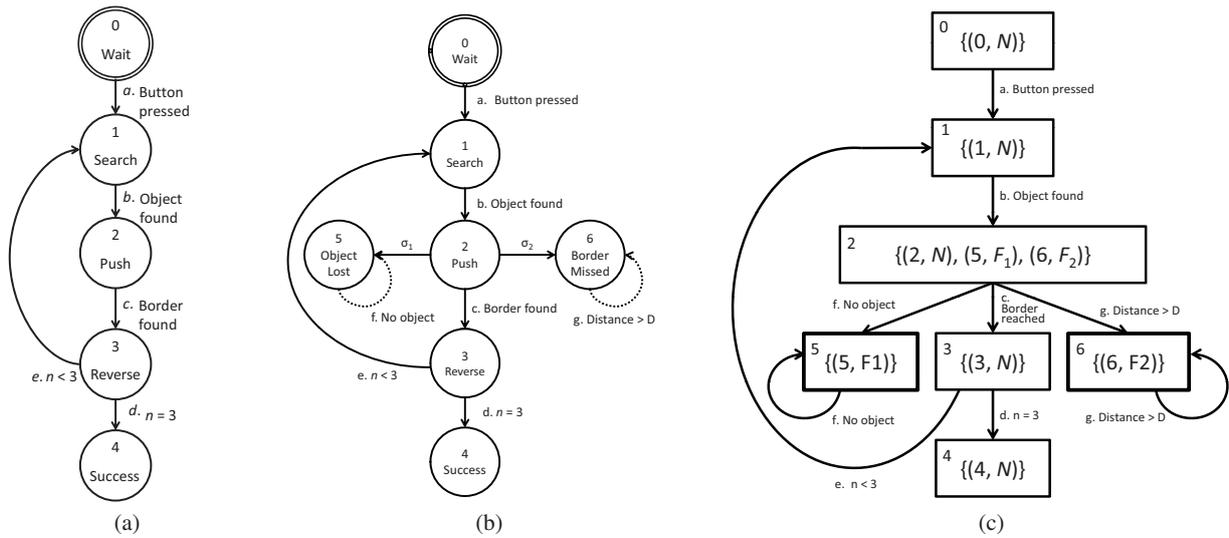


Fig. 3. FDD for the lab-work problem: a) Navigation FSM without considering faults; b) Navigation FSM with faults; c) Navigation with fault diagnoser.

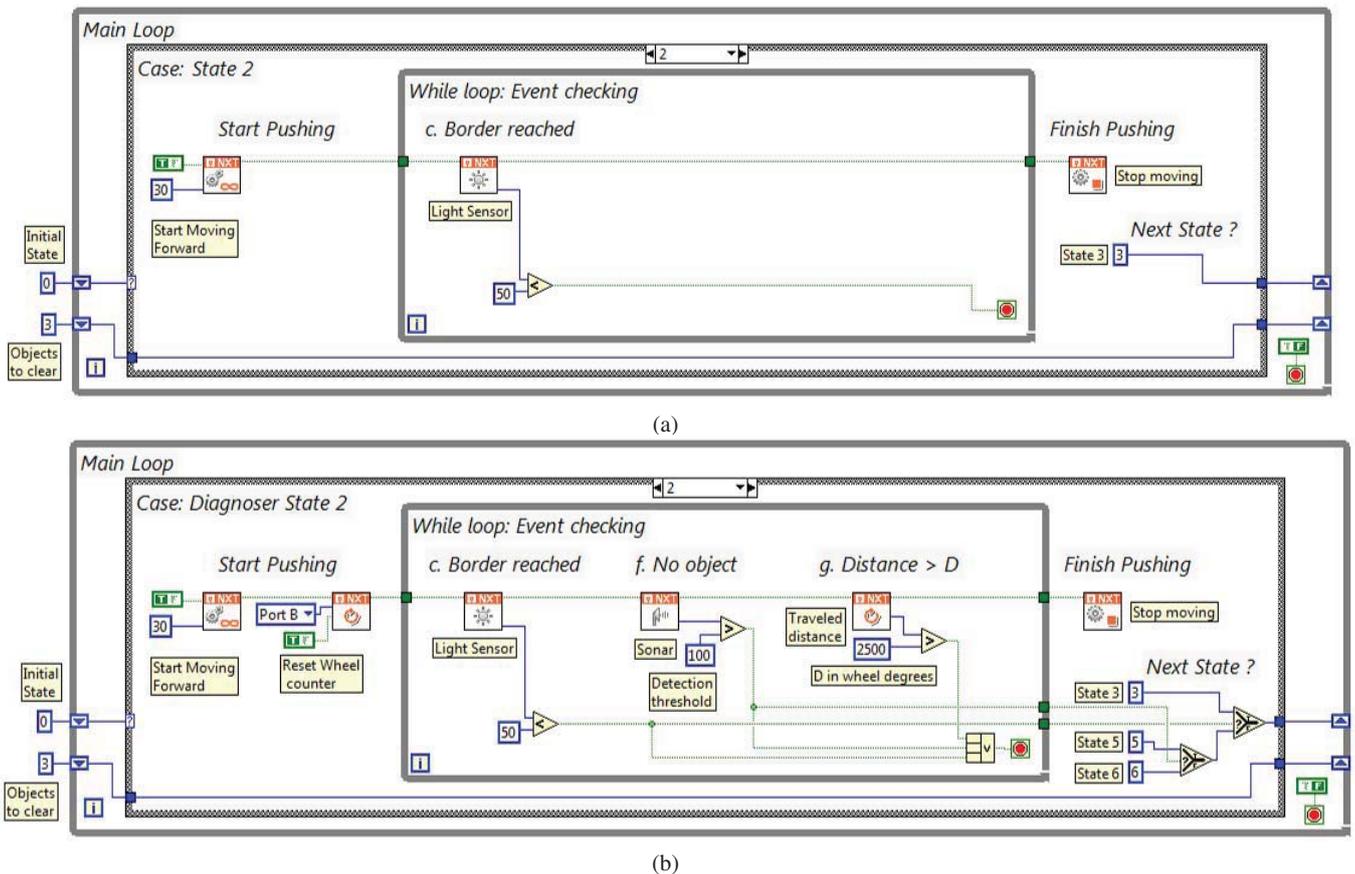


Fig. 4. Two FSM program examples in LabVIEW: a) Implementation of navigation state 2; b) state 2 extended with an FDD solution.

V. APPLICATION & ASSESSMENT

Starting on academic year 2010-11, the proposed lab-work has been applied in UMA's Master in Mechatronics Engineering, a three-semester programme where the second semester is for elective courses. In particular, the lab-work project has been included as part of an elective course entitled

Fault Tolerant Mechatronic Systems (FTMS). The choice of electives also includes *Mobile Robotics (MR)*.

To evaluate intended outcome *LO1*, awareness of the importance of faulty conditions has been surveyed for students of the MR course in academic year 2012-13. Of the total of 13 MR students, nine took also FTMS in the same or in

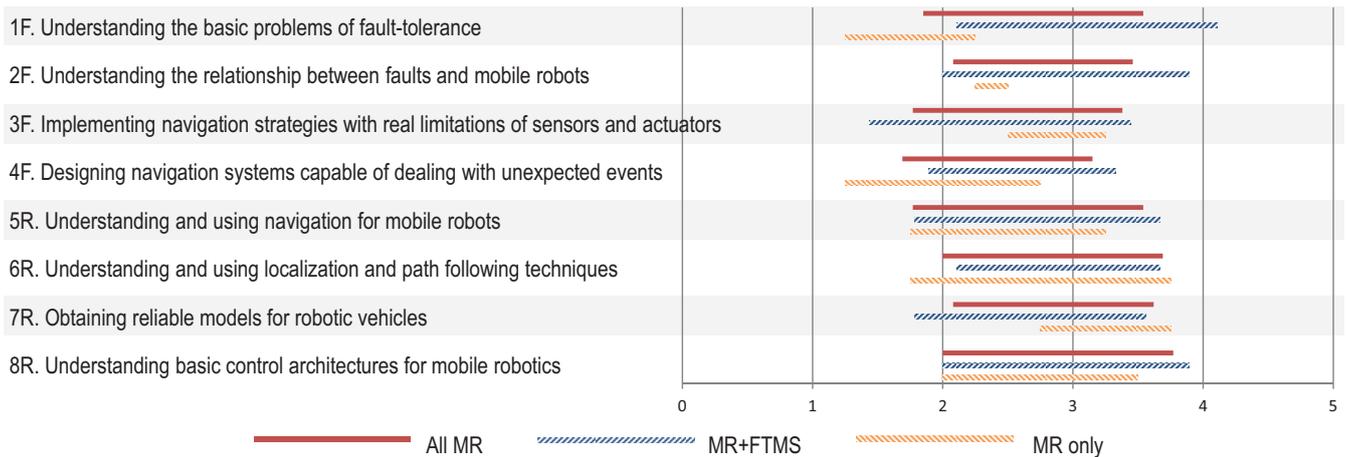


Fig. 5. Average awareness increment in MR students, with a comparison between the group who developed the proposed lab-work project (MR+FTMS) and the group who did not (MR only).

previous years, which allows a comparison. By the end of each course, students evaluated their perceived competences before and after the course with a score between zero and five. The questionnaires referred to issues related to faults (1F to 4F) and also to mobile robotics (5R to 8R). The results of this survey are summarized in Fig. 5, where a bar is shown between the averaged “before” and “after” values, so the bar length represents perceived improvement. First of all, FTMS students acknowledge a substantially greater improvement in 1F, 2F and 3F than those who only took MR, which supports *LO1*. Besides, results for 7R show that students who took both courses are more aware of their initial lack of competences regarding reliable robotic models. Interestingly, with the exception of 6R, MR students improve mobile robotics competences when learning FDD concepts. All in all, this comparison shows that students who developed the FDD project acknowledged not only more awareness on faults but a deeper insight into mobile robotics.

As for the measurement of *LO2*, the FTMS syllabus establishes that individual student lab-work is evaluated according to the following criteria:

- *Project report*. Design and justification of the FSMs for navigation and FDD. Clarity and technical quality of the document.
- *Presentation and discussion*. Content, structure, and delivery of the presentation. Answers to questions posed by the teacher and other students.
- *Software structure and readability*. Effective implemen-

tation of designed FSMs. Proper use of the LabVIEW language and resources.

- *FDD performance*. Response of the implementation to provoked faults.
- *Creativity for extra functionalities*, such as an additional number of faults, uniqueness of the adopted approach, actions based on the diagnosis, etc.

Teachers use these criteria to assess FDD knowledge, as well as key skills such as problem solving or communication of scientific results. Table I summarizes average scores (in a range from 0 to 10) from a total of 20 projects in three academic years. All students completed the project in the scheduled time with a solution that diagnosed at least one fault, with most of them (over 80%) covering two faults. The average lab-work score has been 8.75, with variance of 1.126.

VI. CONCLUSIONS

This paper has addressed education in FDD, which is relevant for higher education engineering students because it is required in reliable industrial processes and products. However, education papers on FDD for safety-critical and other mechatronic systems are virtually non-existent.

Small-scale mobile robots are proposed as an effective equipment to introduce practical skills in FDD. This choice obeys to mobile robots being: *i*) a significant case of autonomous system where dealing with failing execution and uncertainty is essential; *ii*) motivating and familiar for students; and *iii*) affordable, if not already acquired for other courses. Moreover, as many systems can be modeled as DEDS for diagnostic purposes, FMS-based techniques can be a suitable and accessible introduction to FDD for engineering students of different backgrounds.

A project based on a simple robot task has been proposed. The goal is that students develop their own finite state machine navigation algorithm and incorporate DEDS FDD to cope with two types of faulty conditions. This type of project can be completely developed through several practice sessions. A solution example has been detailed for Lego Mindstorms NXT

TABLE I
AVERAGE EVALUATION SCORES FOR STUDENT LAB-WORK.

	Score	Weight	Weighted score
Project report	9.04	20%	1.81
Presentation & discussion	8.89	30%	2.67
Software	9.48	20%	1.90
FDD performance	8.51	20%	1.70
Extra functionalities	6.83	10%	0.68

robots with LabVIEW, a graphical programming language that is widely used in industry and in higher education curricula.

This lab-work project pursues two major learning outcomes: making students aware of the importance of faulty conditions and learning to design FDD strategies for a real system. These outcomes have been measured after the application of the proposed lab-work for three years in a Master course of the University of Málaga. The achievement of the first learning outcome has been evaluated through a survey to Mobile Robotics students, of which only a portion had taken the proposed project. A comparison of both groups has shown that the FDD project students not only acknowledged more awareness on faults but also a deeper insight into mobile robotics. The second learning outcome has been evidenced through evaluation criteria of student projects according to the syllabus. In the future, it will be interesting to test the lab-work project in courses for under-graduate students.

REFERENCES

- [1] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual Reviews in Control*, vol. 32, no. 2, pp. 229–252, 2008.
- [2] M. M. Abdelhameed and H. Darabi, "Neural network based design of fault-tolerant controllers for automated sequential manufacturing systems," *Mechatronics*, vol. 19, no. 5, pp. 705 – 714, 2009.
- [3] D. I. Katzourakis, I. Papaefstathiou, and M. G. Lagoudakis, "An open-source scaled automobile platform for fault-tolerant electronic stability control," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 9, pp. 2303–2314, 2010.
- [4] J. Cieslak, D. Henry, and A. Zolghadri, "Fault tolerant flight control: From theory to piloted flight simulator experiments," *IET Control Theory and Applications*, vol. 4, no. 8, pp. 1451–1464, 2010.
- [5] J. Jiang and X. Yu, "Fault-tolerant control systems: A comparative study between active and passive approaches," *Annual Reviews in Control*, vol. 36, no. 1, pp. 60–72, 2012.
- [6] R. Isermann, *Fault-Diagnosis Applications. Model-Based condition Monitoring: Actuators, Drives, Machinery, Plants, Sensors, and Fault-tolerant Systems*. Springer, 2011.
- [7] J. Antonino-Daviu, J. Pons-Llinares, and V. Climente-Alarcon, "Educational experiences in electric machine fault diagnosis teaching," in *IEEE Global Engineering Education Conference*, Berlin, Germany, 2013, pp. 1070–1075.
- [8] V. F. Pires, J. F. Martins, and T. G. Amaral, "Development of an experimental system for teaching induction motors with fault detection and diagnosis capabilities," *Computer Applications in Engineering Education*, vol. 20, no. 4, pp. 611–618, 2012.
- [9] J. R. R. Ruiz, A. Garcia Espinosa, L. Romeral, and J. A. Ortega, "An introduction to fault diagnosis of permanent magnet synchronous machines in master's degree courses," *Computer Applications in Engineering Education*, vol. 21, no. 2, pp. 349–359, 2013.
- [10] R. J. Romero-Troncoso, A. Garcia-Perez, E. Cabal-Yepez, and R. A. Osornio-Rios, "Experimental system for teaching induction motor faults during the startup transient and steady state," *Computer Applications in Engineering Education*, vol. 22, no. 1, pp. 33–38, 2014.
- [11] P. Record, "Teaching the art of fault diagnosis in electronics by a virtual learning environment," *IEEE Transactions on Education*, vol. 48, no. 3, pp. 375–381, 2005.
- [12] D. Fay, S. Campbell, G. Miller, and D. Connors, "Teaching fault tolerant FPGA design for aerospace applications," in *IEEE International Conference on Microelectronic Systems Education: Educating Systems Designers for the Global Economy and a Secure World*, 2007, pp. 61–62.
- [13] S. Das, S. A. Yost, and M. Krishnan, "A 10-year mechatronics curriculum development initiative: Relevance, content, and results: Part I," *IEEE Transactions on Education*, vol. 53, no. 2, pp. 194–201, 2010.
- [14] K. K. Tan, K. N. Wang, and K. Z. Tang, "Mechatronic experiment on remote vibration monitoring and fault diagnosis via the internet," *International Journal of Engineering Education*, vol. 19, no. 3, pp. 503–511, 2003.
- [15] G. Bertoni, P. Castaldi, and M. E. Penati, "Postgraduate education on fault diagnosis and control reconfiguration in aerospace," in *IFAC Symposium on Advances in Control Education*, vol. 9, no. Part 1, Nizhny Novgorod, Russian Federation, 2012, pp. 306–311.
- [16] C. Ferrell, "Failure recognition and fault tolerance of an autonomous robot," *Adaptive Behaviour*, vol. 2, no. 4, pp. 375–398, 1994.
- [17] O. Pettersson, "Execution monitoring in robotics: A survey," *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 73 – 88, 2005.
- [18] J. Carlson and R. R. Murphy, "How UGVs physically fail in the field," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 423–437, 2005.
- [19] S. A. Arogeti, D. Wang, C. Low, and M. Yu, "Fault detection isolation and estimation in a vehicle steering system," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 12, pp. 4810–4820, 2012.
- [20] F. Baghernezhad and K. Khorasani, "A robust fault detection scheme with an application to mobile robots by using adaptive thresholds generated with locally linear models," in *Proceedings of the IEEE Symposium on Computational Intelligence in Control and Automation - IEEE Symposium Series on Computational Intelligence*, 2013, pp. 9–16.
- [21] J. Alonso, L. Magdalena, S. Guillaume, M. Sotelo, L. Bergasa, M. Ocaña, and R. Flores, "Knowledge-based intelligent diagnosis of ground robot collision with non detectable obstacles," *Journal of Intelligent and Robotic Systems*, vol. 48, pp. 539–566, 2007.
- [22] P. Sundvall and P. Jensfelt, "Fault detection for mobile robots using redundant positioning systems," in *IEEE International Conference on Robotics and Automation*, vol. 2006, 2006, pp. 3781–3786.
- [23] M. J. Daigle, X. D. Koutsoukos, and G. Biswas, "Distributed diagnosis in formations of mobile robots," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 353–369, 2007.
- [24] B. Durand, K. Godary-Dejean, L. Lapierre, and D. Crestani, "Global methodology in control architecture to improve mobile robot reliability," in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, 2010, pp. 1018–1023.
- [25] S. Jung, "Experiences in developing an experimental robotics course program for undergraduate education," *IEEE Transactions on Education*, vol. 56, no. 1, pp. 129–136, 2013.
- [26] P. J. Navarro, C. Fernandez, and P. Sanchez, "Industrial-like vehicle platforms for postgraduate laboratory courses on robotics," *IEEE Transactions on Education*, vol. 56, no. 1, pp. 34–41, 2013.
- [27] C. M. Ionescu, E. Fabregas, S. M. Cristescu, S. Dormido, and R. De Keyser, "A remote laboratory as an innovative educational tool for practicing control engineering concepts," *IEEE Transactions on Education*, vol. 56, no. 4, pp. 436–442, 2013.
- [28] Y. Kim, "Control systems lab using a LEGO mindstorms NXT motor system," *IEEE Transactions on Education*, vol. 54, no. 3, pp. 452–461, 2011.
- [29] A. Cruz-Martín, J. Fernández-Madrigrá, C. Galindo, J. González-Jiménez, C. Stockmans-Daou, and J. L. Blanco-Claraco, "A LEGO mindstorms NXT approach for teaching at data acquisition, control systems engineering and real-time systems undergraduate courses," *Computers and Education*, vol. 59, no. 3, pp. 974–988, 2012.
- [30] M. Pinto, A. P. Moreira, and A. Matos, "Localization of mobile robots using an extended Kalman filter in a LEGO NXT," *IEEE Transactions on Education*, vol. 55, no. 1, pp. 135–144, 2012.
- [31] D. Zaldivar, E. Cuevas, M. Pérez-Cisneros, J. Sossa, J. Rodríguez, and E. Palafox, "An educational fuzzy-based control platform using lego robots," *International Journal of Electrical Engineering Education*, vol. 50, no. 2, pp. 157–171, 2013.
- [32] F. Lin, "Diagnosability of discrete event systems and its applications," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 4, no. 2, pp. 197–212, 1994.
- [33] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Failure diagnosis using discrete-event models," *IEEE Transactions on Control Systems Technology*, vol. 4, no. 2, pp. 105 – 124, 1996.
- [34] K. Janschek, *Mechatronic Systems Design. Methods, Models, Concepts*. Springer, 2012.
- [35] S. H. Zad, R. H. Kwong, and W. M. Wonham, "Fault diagnosis in discrete-event systems: Framework and model reduction," *IEEE Transactions on Automatic Control*, vol. 48, no. 7, pp. 1199–1212, 2003.
- [36] J. M. Gómez-de-Gabriel, A. Mandow, J. Fernández-Lozano, and A. García-Cerezo, "Using Lego NXT mobile robots with LabVIEW for undergraduate courses on mechatronics," *IEEE Transactions on Education*, vol. 54, no. 1, pp. 41–47, 2011.
- [37] E. Gascard and Z. Simeu-Abazi, "Modular modelling for the diagnostic of complex discrete-event systems," *IEEE Transactions on Automation Science and Engineering*, 2013.
- [38] R. R. Murphy, *Introduction to AI Robotics*. The MIT Press, 2000.

- [39] R. Bitter, T. Mohiuddin, and M. Nawrocki, *LabVIEW Advanced Programming Techniques*, 2nd ed. CRC Press LLC, Boca Raton, 2006, ch. 3.