

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/241715081>

Mechatronic system design course for undergraduate programmes

Article in *European Journal of Engineering Education* · August 2011

DOI: 10.1080/03043797.2011.593094

CITATIONS

12

READS

9,119

3 authors:



Ashraf I. Saleem

Michigan Technological University

55 PUBLICATIONS 361 CITATIONS

[SEE PROFILE](#)



Tarek A. Tutunji

Al Hussein Technical University

59 PUBLICATIONS 508 CITATIONS

[SEE PROFILE](#)



Lutfi Al-Sharif

University of Jordan

154 PUBLICATIONS 1,048 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Trajectory tracking of an amplified piezoelectric actuator [View project](#)



Modeling and Control of Piezoelectric Ultrasonic Actuator in Ultrasonically Assisted Machining [View project](#)

Mechatronic System Design Course for Undergraduate Programs

A. Saleem*, T. Tutunji and L. Al-Sharif

Philadelphia University, Amman, Jordan

Technology advancement and human needs have led to the integration among many engineering disciplines. Mechatronics engineering is an integrated discipline that focuses on the design and analysis of complete engineering systems. These systems include mechanical, electrical, computer, and control subsystems. In this paper, the importance of teaching mechatronic system design to undergraduate engineering students is emphasized. The paper offers the collaborative experience in preparing and delivering the course material for two universities in Jordan. A detailed description of such a course is provided and a case study is presented. The case study used is a final year project where students applied a six-stage design procedure that is described in the paper.

Keywords: mechatronics education; design procedure; student projects.

1. Introduction

Mechatronic engineering has gained much interest in recent years due to the advancement of integrated engineering systems (Ebert-Uphoff et al. 2000). Design, analysis, and integration of mechanics with electronics through intelligent algorithms for control applications are fundamental elements in Mechatronics education.

An essential engineering skill that must be taught to undergraduate students is design. This skill is usually built up and taught to students through a combination of problem solving skills, dedicated courses and projects. Nevertheless, constructing (and teaching) engineering design requires much more preparation and skill as compared to the traditional courses. The reasons for the difficulty in teaching such courses are due to the fact that design involves open-ended problems with several possible solutions. So, there is no specific single correct answer. This is usually a challenge to the students and should be provided to them in a well constructed course through practical examples.

There are many engineering schools that offer engineering design courses (Lee, 2009). Admittedly, there are general steps and abstract guidelines that form a common base among all engineering schools. Nevertheless, the teaching methods differ among different universities and among different disciplines. In this paper, an argument will be made that today's engineering need is to prepare the students for an integrated design system approach.

The mechatronic system design course is best suited for mechanical, electrical, and/or control engineering disciplines. It is constructed for senior students that have already finished the following courses: body mechanics, electronics, motors, sensors, programming, and control. In effect, it is a good example of a taught capstone course.

In (Craig, 2001), researchers described an undergraduate program that offers as a core two classes: Mechatronics and Mechatronic System Design. The authors argue that an essential characteristic of a mechatronics engineer is the balance between the modeling/analysis skills and the hardware implementation skills. A block

diagram of a dynamic system investigation was provided and details for the two courses were given. In (Singhose et al. 2009), authors emphasized the importance of hands-on, project-based, design experience. The described course is an introduction to mechatronics course where they discussed the general lectures supported by lab works.

Other researchers discussed their experience in teaching design courses. Jarrah (Jarrah, 2005) proposed a mechatronics design course intended for graduate students and worked in a design lab-like experiments that led to a product. Moreover, details for such a course for undergraduates were provided in (Tutunji et al. 2009) where authors presented, evaluated, and discussed the guidelines for a successful mechatronics project class. Denayer (Denayer, et al. 2003) presented an undergraduate design course by problem-based education.

Capstone courses find applications in many areas of engineering, such as engineering applications in biology and medicine (Goldberg, 2009 and electrical and computer engineering (Saad, 2007) as well as electromagnetic. Ideally the capstone project should relate to real life as much as possible. Some studied the effect of prior industry experience on the success of capstone projects (Gruenther et al. 2009).

In this paper, the experience gained in developing the mechatronic system design course at Philadelphia University and Jordan University will be discussed. This will include the course objectives, teaching methodologies, material, and outcomes.

The importance of this paper derives from the fact that the Mechatronics System Design course is a relatively new course and is not well established as other more traditional courses (e.g., control systems, electrical machines). For this reason, the contents of this course and the method of delivery of such a course are not widely documented. This paper introduces a detailed ‘road-map’ for the recommended course content for this course and the method of delivery. It also contains topics that are not traditionally taught in engineering undergraduate programmes, such as detailed components selection.

The rest of the paper is organized as follows: section 2 presents the course outcomes and topics. Section 3 provides an overview on actuators and sensors and their selection, while controllers, algorithms, and interfacing are presented in section 4. Section 5 explains the system design procedure that is being delivered in the course. A detailed students’ project that is used as a case example is given in section 6. Finally, section 7 concludes this paper.

2. Course Outcomes and Topics

The mechatronics system design course is given to senior students as part of the mechatronics curriculum at Philadelphia University and University of Jordan. The course objective is to prepare the students to work with practical and fully-integrated systems. At the end of the course, students are expected to have the following outcomes:

- Evaluate and select suitable actuators, sensors, controllers, and algorithms.
- Integrate and interface various components and subsystems.
- Follow a well defined design procedure.

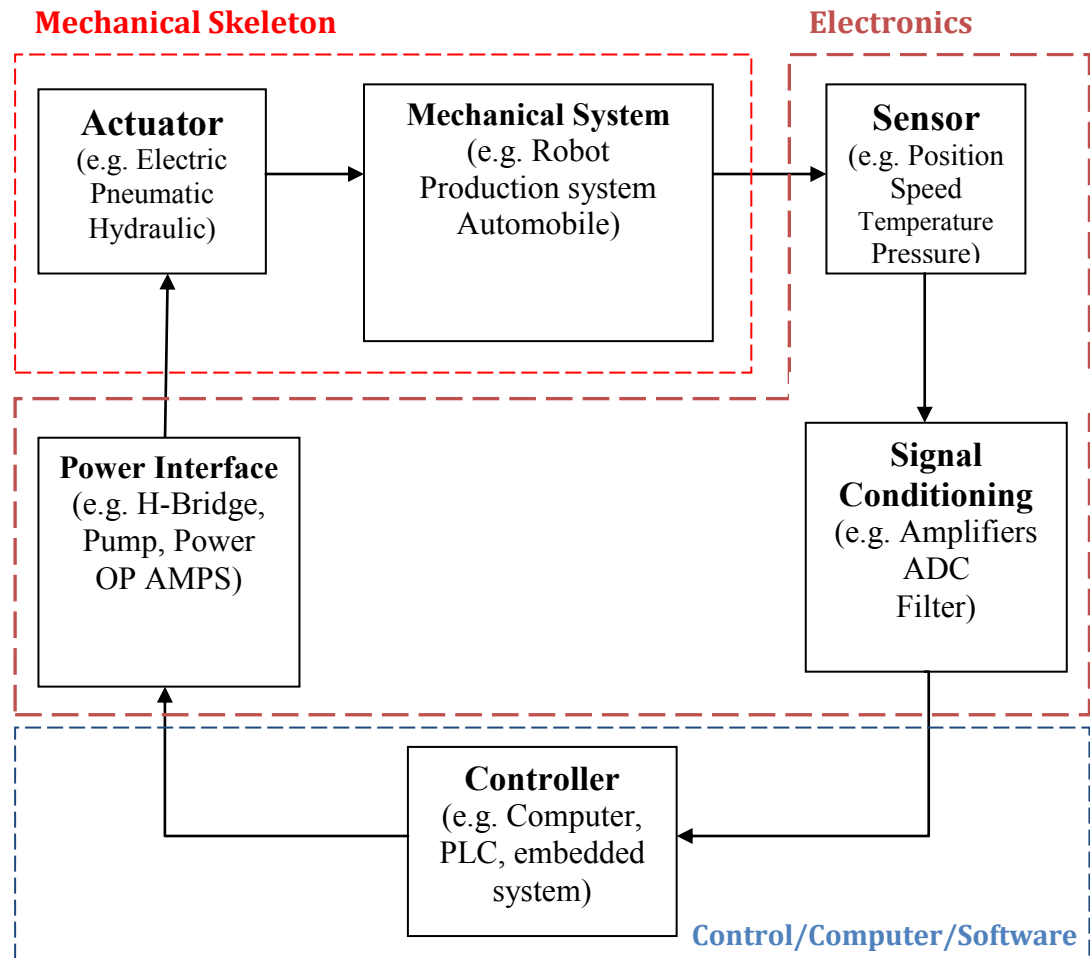


Figure 1. Mechatronic System Block Diagram

Mechatronic systems contain integrated subsystems as shown in Figure 1. These subsystems should be known to the students. They are expected to have taken courses in mechanics, sensors, control theory, microcontroller, electronics, and electrical motors. The main idea of the course is to review and interface the described subsystems in order to design fully-integrated systems that meet specified requirements. The course covers the following main topics:

- *Introduction to mechatronic systems:* overview of mechatronic systems; applications in industry, space, medicine, home appliances and automotive.
- *General engineering principles:* description of a number of general engineering principles; basics of testing and troubleshooting.
- *Electrical actuators:* revision and selection among different motor types.
- *Sensors and transducers:* revision and selection among different sensor types
- *Control systems:* overview and selection of physical controllers and control algorithms.
- *Interconnection and interfacing:* review of drive circuits, conditioning circuits and data acquisition systems.
- *Mechatronic system design procedure:* detailed study of the mechatronic system design stages.

- *Case studies*: Selected case studies to reinforce the principles and the design procedure.

The course is normally delivered through three parallel stages. In the first stage, students are exposed to theoretical material which covers the previous mentioned topics. Parallel to the theoretical lectures, students carry out lab experiments to reinforce the concepts delivered in the class using standard experiments kits such as elevators, production lines, and robot manipulators. All these experiments are carefully designed in order to suit the practical requirements of the course. Finally, students are required to conduct group projects where they must apply design principles and components selection.

Figure 2 shows the lecture time allocation for the listed topics. This is a three credit hour course which has 40 hours of lecture time. The first two topics, introduction and general principles, are mainly used to prepare the student for the course and will not be covered in this paper because they are general terms and can be easily found in the literature. The rest of the topics will be discussed in the following sections.

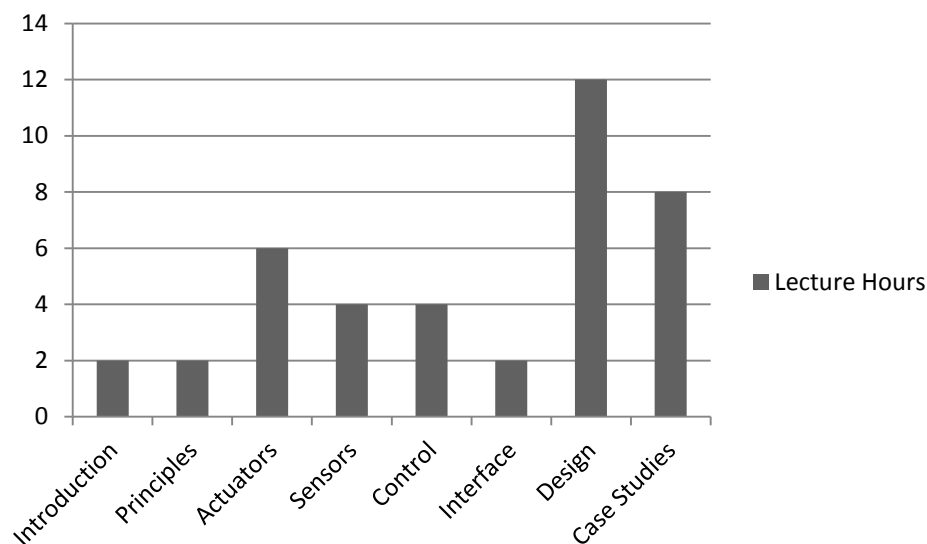


Figure 2. Lecture time allocation

3. Actuators and Sensors

Actuators are the driving force in mechatronic systems and therefore are covered first. Students are reminded of the large range of actuators that are used in practice, whether rotary or linear and whether electrical, electromagnetic, pneumatic or hydraulic. This course however, only covers the rotary electric actuators in detail, leaving the linear electric actuators to exercises and case studies. Pneumatic and hydraulic actuators are taught in a separate dedicated course.

The students are introduced to various factors that influence the selection of rotary electric actuator that include:

- Positioning accuracy

- Power and torque requirements
- Continuous and intermittent duty cycles
- Speed range

Three examples are given to allow the student to gain experience in selecting motors for three mechatronic systems: a conveyor belt using a stepper motor, an elevator using an induction motor and an electric vehicle using a permanent magnet DC motor. The examples aim to link the basic concepts in mechanics with the electrical and mechanical performance of the motor under consideration and the overall system requirements. It also allows the student to gain good experience in dealing with mechanics in terms of balance masses and rotational inertias. These examples prepare the student for dealing with practical systems and make the exercises more enjoyable.

Sensors play an important role in mechatronic systems. They are usually utilized in mechatronic systems to measure (1) system outputs for feedback control; (2) system inputs for feed-forward control; and (3) output signals for system monitoring, diagnosis, evaluation, parameter adjustment, and supervisory control.

Students should be given a brief review of the usage of sensors to measure physical variables such as temperature, pressure, force, speed, and flow. Moreover, students should understand the differences among sensors within the same category in order to appreciate relative strengths and weaknesses for each sensor type.

Students should also comprehend the sensor characteristics such as sensor range, sensitivity, accuracy, precision, linearity, resolution, and frequency response. After reviewing these topics, students should have gained enough understanding in the selection of the appropriate sensors to be used.

4. Control Systems and Interfacing

The design of mechatronic systems involves the choice of the controller. This is arguably the most critical decision in the design process. A vast variety of controllers are available in the market. The following is a list of the four main controller groups is explained to the student:

- *A Programmable Logic Controllers (PLC)* is a user-friendly, microprocessor-based, specialized computer that is used for process control. It contains I/O modules for appropriate sensors/actuator interfaces. It is mainly used in automated manufacturing lines. The PLC is usually used for simple logic operations. It is considered reliable and easy to program (using ladder diagrams, instructions, or function blocks).
- *The Microcontroller* is a computer-on-chip. It is an Integrated Circuit (IC) that contains microprocessor, memory, I/O parts, and sometimes A/D converters. It can be programmed using several languages (such as Assembly or C/C++). It can be used in manufacturing lines, but requires additional hardware. Microcontrollers are mainly used in engineering products such as washing machines and air-conditioners.
- *Digital Signal Processors (DSP)* are specialized microprocessors with advanced architectures (such as multiple buses, parallel processing, hardware multipliers, and fast sampling rate) that are designed to reduce the number of instructions and operations necessary for efficient processing. DSP chips enable developers to implement complex algorithms and perform

computationally efficient and fast algorithms. DSP are preferred over microcontrollers when the need for complex and iterative control algorithms is required. Applications vary from hard disc drives to missile control.

- *Personal Computers (PC)* are used when extensive signal processing and in-depth analysis is required. Advantages include superior graphical and software flexibility. However, the cost is high and therefore is not suitable for a large number of products.

Students are introduced to different systems and are asked to consider what the most suitable controller for such a system is. In many cases there is no one single correct answer. The student is encouraged to consider all the factors that might influence the decision in terms of space, processing power, environment, cost of final product, programming language, safety criticality of the application, required time to market, reliability and number of products to be produced.

Another critical decision that the student must make is the type of control algorithm to use. It is important to highlight to the student that the decisions as to the controller used and the controller algorithm are interdependent. There are many controller algorithms that can be used for mechatronic systems. The following is a list of three that are explained to the student:

- *On-Off Control*: This is the simplest method of control. The control action has three possible outputs: on, off, or no change. This method is usually used for slow acting operations (such as refrigeration unit). The advantage is its ease of design and low cost. However, it cannot vary the controlled variable with precision. The student is introduced to simple examples such as a domestic heating system the exact temperature of which is not critical. This is an ideal example of a simple on/off controller application.
- *Proportional-Integral-Derivative (PID)*: Control. This is the most commonly used controller algorithm. It is applied in automated manufacturing and mechatronic products. In its simplest mode, only the proportional part is used. The error between the desired and measured values of the controlled variable is calculated, multiplied by a gain and applied to the system under control. An integral mode is usually added to minimize the steady state error while a derivative mode is added to minimize the transient overshoot (H. Ang, et al. 2005).
- *Intelligent Control*: When the system to be controlled does not have a well-defined mathematical model because of high nonlinearities or missing information, fuzzy or neural controllers are applied. Fuzzy controllers are used when general fuzzy control rules can be gathered from an expert. Neural network controllers are used when experimental I/O data is available from the system

An important skill that student should master within the area of mechatronics system design is interfacing the controller to the physical system. Students usually take the function of interfacing for granted without understanding its criticality and importance.

Within this course, students learn the importance of switching (whether electromechanical or electronic) concepts. These include:

- Voltage and current rating limitations of electronic devices (e.g., microcontrollers).
- The use of contactors and relays in switching loads.
- The use of the transistor as a switch or a current amplifier.
- The use of drivers and H-bridges.
- The use of opto-isolators to separate circuits in order to prevent noise and damage.

5. Design Procedure

Knowledge of sensors, actuators, controllers, and interfacing prepares students for the design of mechatronic systems. The following is a summary of the mechatronic system design procedure and stages that are covered in this course.

(1) User and System Requirements Analysis

Typically, the Mechatronic system design lifecycle starts with the user requirements analysis stage in order to acquire the system's specifications and requirements. Critical benefits such as enhanced quality of work, reduction in support and training costs and improved user satisfaction can be achieved.

Next, the system requirements can be determined. System requirements analysis can be a challenging phase, because all of the major customers and their interests are brought into the process of determining requirements. The quality of the final product is highly dependent on the effectiveness of the requirements identification process. These requirements form the basis for all future work on the project, from design and development to testing and commissioning. It is of the highest importance that the system designer creates a complete and accurate representation of all requirements that the system must accommodate.

(2) Conceptual Design

A conceptual design embodies the structure of the system, as perceived by the user. This structure is a formulation of the conceptual ideas to meet the requirement specifications which are the results from the previous phase. The input and goals of the conceptual design process form the requirement specifications to be met. *It is always more expensive to improve a bad conceptual design than a bad detailed design, therefore this process is crucial and essential.*

This process generates solutions without detailed design parameters, and acts as a blueprint for the subsequent design and implementation stages. New requirements may arise within the process, which should be considered. After generating concepts and meeting all the requirements, the optimal concept and solution can be then selected, if accepted by the customer. By the end of this stage, system block diagram and operation flow chart should be available.

(3) Mechanical, software, electronics, and interface design

The mechanical design is crucial since it forms the skeleton of mechatronic systems. The design can be supported by Computer Aided Design (CAD) tools which are widely available in industry. During the mechanical design process, control system should be evaluated and designed synergistically with the mechanical design.

If optimal mechanical design is achieved, the designer should move to select the actuators and sensors that can meet the demand specifications. This should then be followed by selecting the drive and conditioning circuits in order to interface the system components.

A mechatronic system is an integrated engineering design, which can be complex to design and optimize since it covers several engineering domains, as shown in Figure 1. Optimizing each part separately might not result in the best design. Therefore, it is essential to establish the optimal tradeoffs between the mechanical and electronic subsystems. This is done through synergistic design where the system design is divided into three parallel tracks (see Figure 3):

- The design of the mechanical frame/machine.
- The design of the electronic system.
- The design of the software/controller.

During the design, the interface among these tracks should be evaluated continuously in order to optimize the system performance. This usually results in design modifications of the individual tracks.

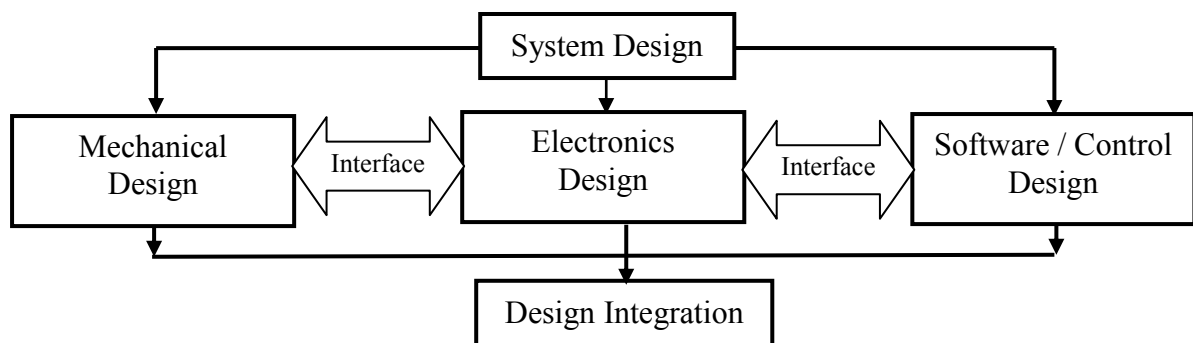


Figure 3. Synergistic Design

(4) System Modeling and Simulation

Mechanical, electrical and electronic components should be included in a mixed system model of the plant. Initially, such a model should be fairly simple. A linear time-invariant model with one input and one output is often adequate at the initial modeling stage, even for relatively complex mechatronic systems.

The model parameters should be determined based on the designed mechanical components and the selected actuators and sensors. The designer has the freedom to modify these values, increase the number of inputs/outputs used, and include nonlinearities in the subsequent design iterations.

Once a plant model is available, simulation is used to decide on the design specifications of the mechatronic system based on the specification of requirements. Specifications can be made in a variety of forms such as rise time, bandwidth, gain margin, and pole/zero locations. If the simulation results do not satisfy the design specifications, the designer should revisit stage 3.

The simulation can be divided into three parts: mechanical, electronic, and system. Mechanical simulation is used to test the kinematics and dynamics variables, electronic simulation is used to test circuit functionality and compatibility, and the

system simulation is normally used to test the system's response for different inputs (open and closed loop cases). For the mechanical simulation, students are advised to use software tools such as ProEngineer and SolidWorks. For the electronic and system simulation, they are advised to use Protus and Matlab respectively.

(5) Prototyping and testing

Once the model is verified by simulation, the physical system prototype should be assembled and tested in the lab. If the prototype does not satisfy the design specifications, at this stage the designer should check to see whether some modifications to the plant could yield the required results. In case the plant modifications could realize the desired performance, the design process should stop here. Otherwise the designer should reconsider the control system design.

6. Case Study

A Robotic Parking Garage project is a final year project conducted in Philadelphia University. The students followed the design procedure described in this paper and the output was a working prototype. Robotic parking is a method of automatically parking and retrieving cars, using a computerized system. After brainstorming the following requirements were set:

- A room to determine all the parameters of desired car such as length, width, height, weight and security parameters, provided an indicator to interface between driver and the system. All car dimensions have been scaled down with a ratio 20:1, and hence the dimensions were as follows:
 - Maximum height= 100 mm
 - Maximum width= 110 mm
 - Maximum Length= 220 mm
 - Maximum weight= 1 kg
- Appropriate measures must be provided to deny access cars exceeding the capacity of the system.
- Two gates, if all vehicle parameters are appropriate and if a free space exist, one of the gates will be opened to enter the desired vehicle to the system; otherwise the other gate will be opened.
- The robot shall be capable of handling a maximum weight of vehicle, at appropriate speed in order to have a maximum car parking time of 2 minutes and retrieving time of 10 minutes.

Many concepts for the system design emerged. After deeply analyzing and consulting the system requirements, best concept was selected and adopted. The system consisted of four main subsystems:

- (1) *Satisfaction room*: where all car dimensions and parameters are measured via sensors and fed back to the main computer.
- (2) *Gates*: which rout cars either to the park or to the exit. The gats are actuated by motors.
- (3) *Logging system*: where the customer fingerprint is scanned and entered to a database that is linked to the car parking position.

(4) *Robot arm*: which automatically parks and retrieves cars. The robot is a 3 DOF:

- Up-down motion to move from one floor to another.
- Rotational motion to move between rooms.
- In-out motion to park and retrieve cars.

Figures 4 and 5 show the block diagram and the flow chart explaining the main subsystems and their connection as well as the operation sequence of the system.

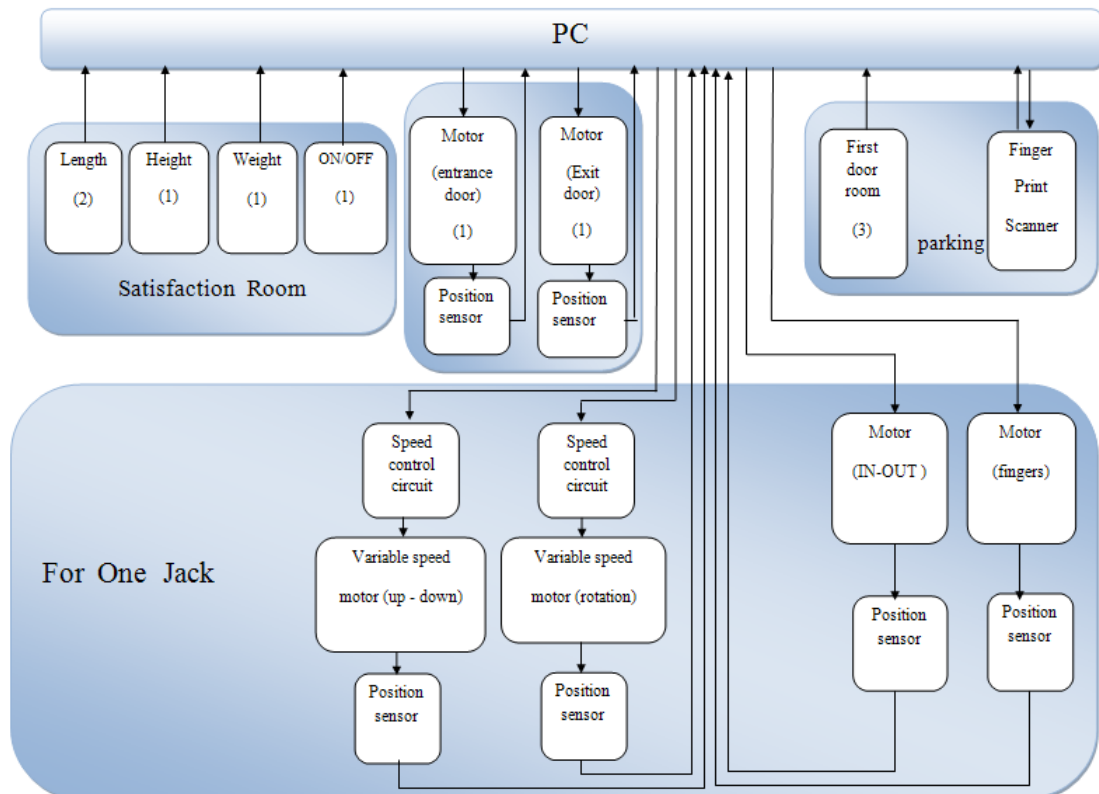


Figure 4. System block diagram

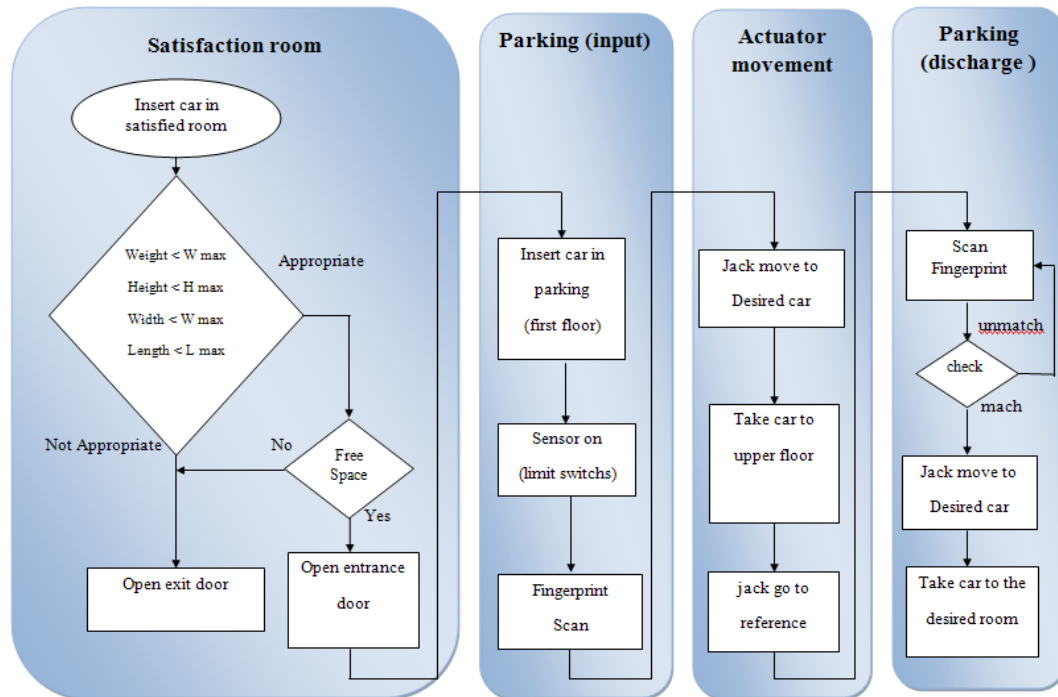


Figure 5. Operational flow chart

After deciding on the concept, students started with the mechanical design. A CAD model of the satisfaction room, robot arm, and building structure were developed as shown in Figures 6, 7, and 8. Next, students worked on component selections as shown in Table 1.

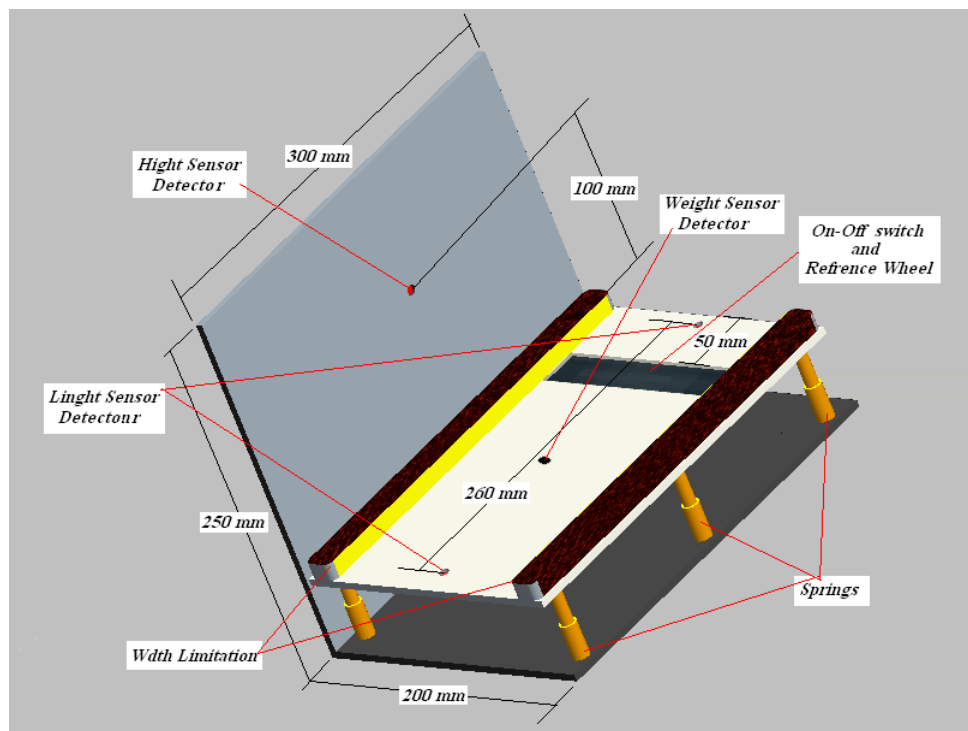


Figure 6. Satisfaction room CAD model

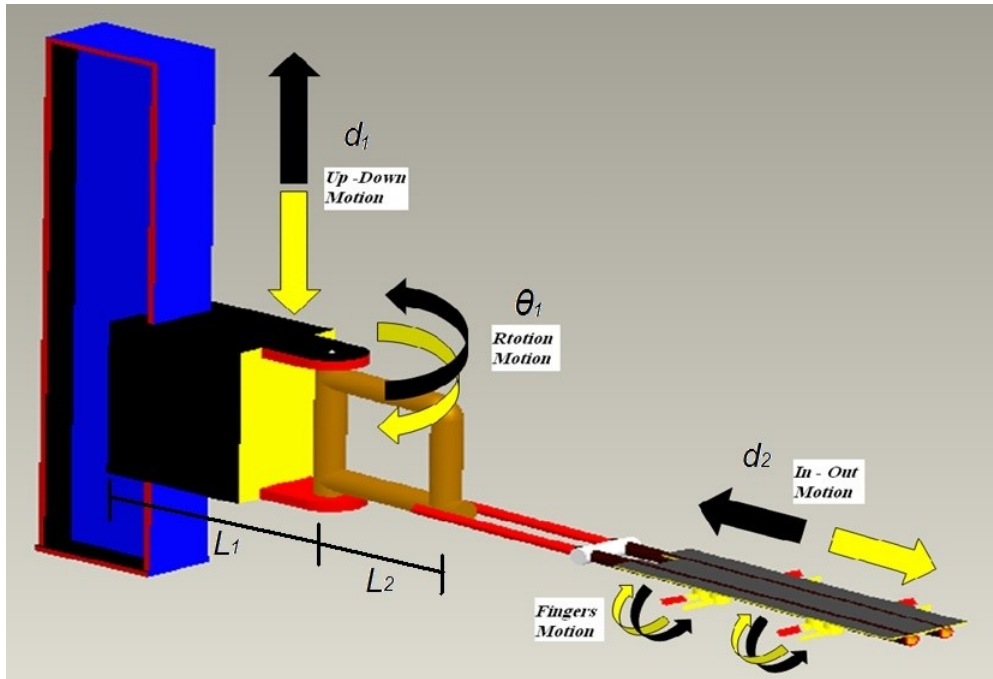


Figure 7. Robot arm CAD model

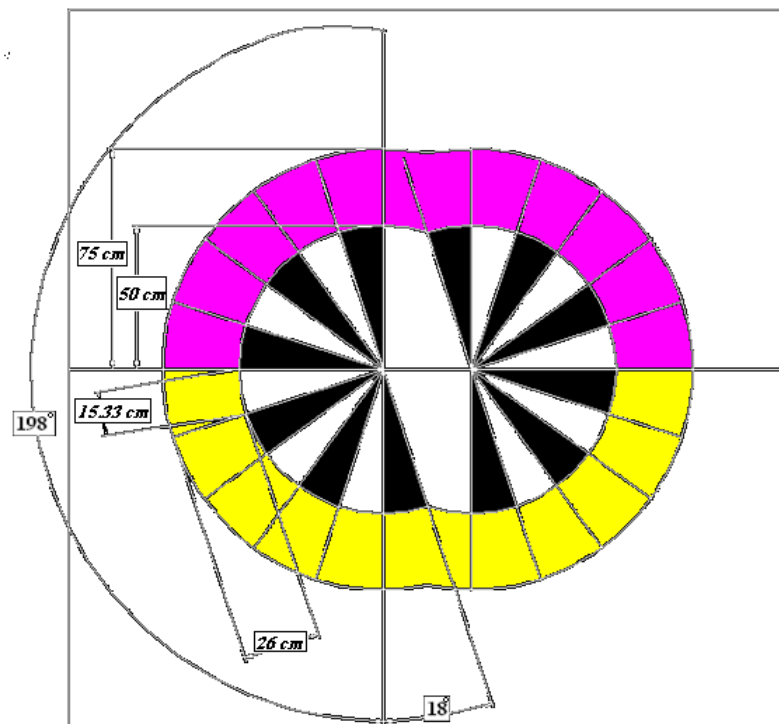


Figure 8. Building structure CAD model

Table 1. Selected actuators and sensors

Subsystem	Actuators	Sensors
Satisfaction room	NA	<ul style="list-style-type: none"> Three infra red sensors

		for height and length measurements. <ul style="list-style-type: none"> • Six springs and limit switch for the weight measurement.
Gates	Two DC motors; one for each gates.	Four limit switches; two for each gate.
Logging system	NA	Fingerprint reader
Robot arm	<ul style="list-style-type: none"> • Three DC motors for the joints. • One DC for the end effector 	<ul style="list-style-type: none"> • Linear encoder for the up-down motion. • Rotary encoder for the rotational motion. • Two limit switches for the in-out motion • Two limit switches for the end effector motion

Once all components were selected, students started looking at different interface circuits to be used. Those included power conversion, power drive, and conditioning circuits. All circuits were designed, tested in simulation, and implemented in the lab.

The system has two main parts to control the satisfaction room and two barriers and the robot. A computer was selected as a controller and two control algorithms were used: (1) On/Off control for the gates and the gripper, and (2) PID control for the robot joints. All control algorithms were verified using simulation tools.

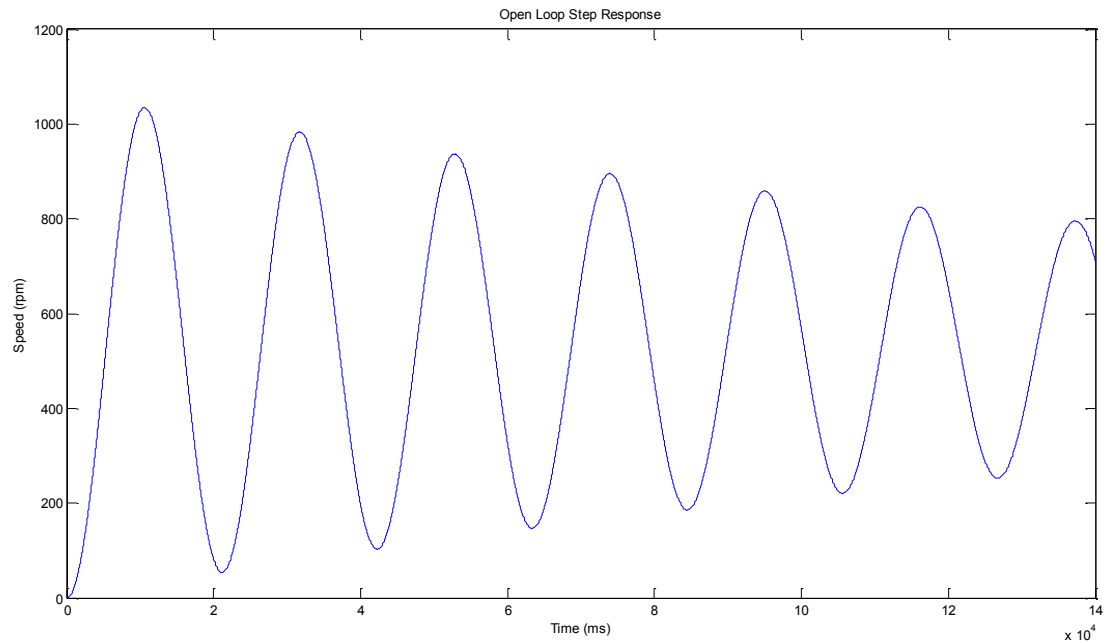
In order to verify the design, system modeling and simulation is performed. First, Denavit-Hartenberg (DH) parameters for the robot arm shown in Figure 7 are created in Table 2. Forward and inverse kinematic simulation is then performed using Matlab.

Table 2: Robot DH Parameters

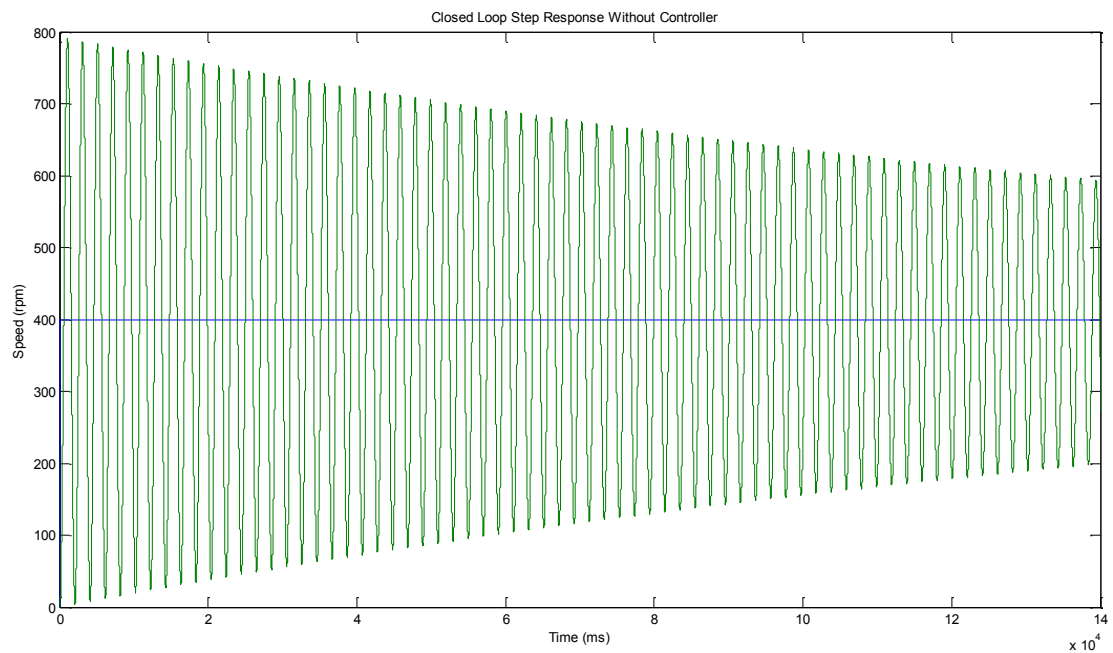
<i>Joint #</i>	<i>α_{i-1}</i>	<i>a_{i-1}</i>	<i>d_i</i>	<i>θ_i</i>
1	0	0	d_1	0
2	0	L_1	0	θ_2
3	90°	0	$L_2 + d_2$	0

Where L_1 and L_2 are the link length, d_1 and d_2 is the linear deployment of joints 1 and 3, and θ_2 is joint 2 angle. Open and closed loop response of each robot joint is simulated in order to optimize the joint controller. Figure 9 shows the open loop and closed loop response of the system. Proportional-Integral-Derivative (PID) controller is used in the closed loop control.

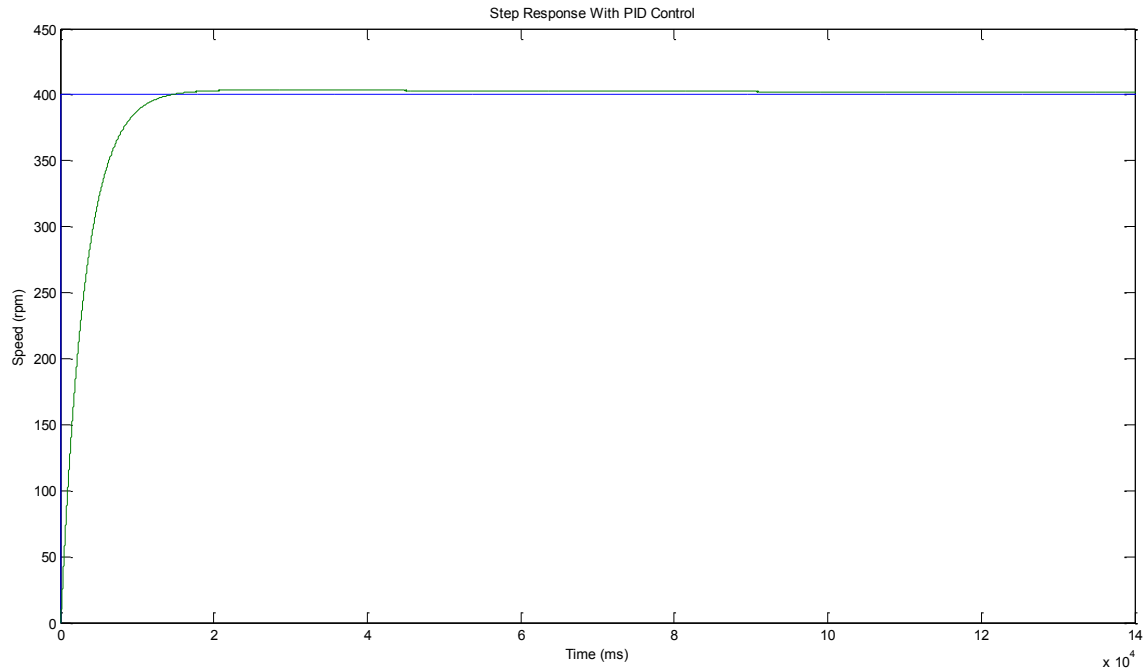
As shown in Figure 9-a, the robot joint speed oscillate around the set point (400 rpm) with very long settling time. After feeding the speed back to the PID controller, the system response improved and quickly settled around the set point. Figure 9-c shows the system response after tuning the PID controller where the overshoot, rise time, and settling time are reduced.



(a)



(b)



(c)

Figure 9: Robot joint step response (a) Open loop, (b) Closed loop without controller, and (c) Closed loop with PID controller.

After verifying the system design through computer simulation, all components for the system were acquired in order to assemble/integrate the system. Figures 10, 11 and 12 show the implemented prototype. The prototype was then tested for further improvements to satisfy the system and customer requirements.

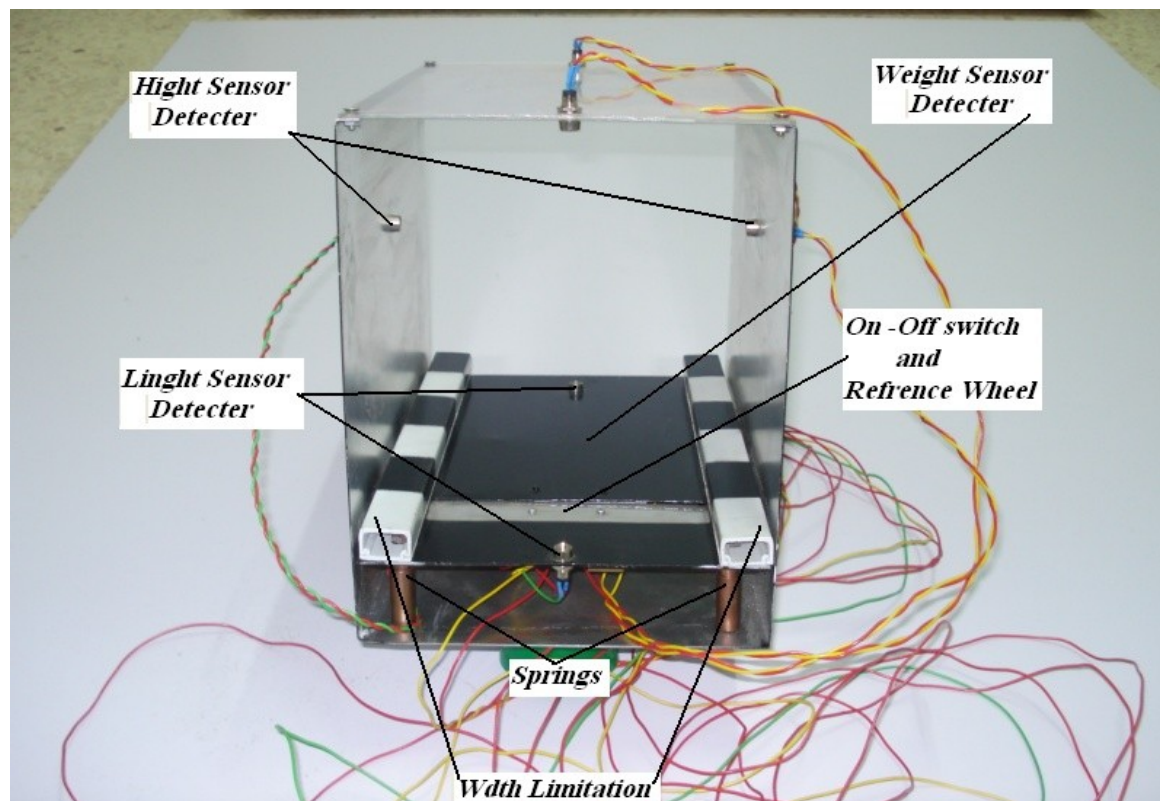


Figure 10. The implemented satisfaction room

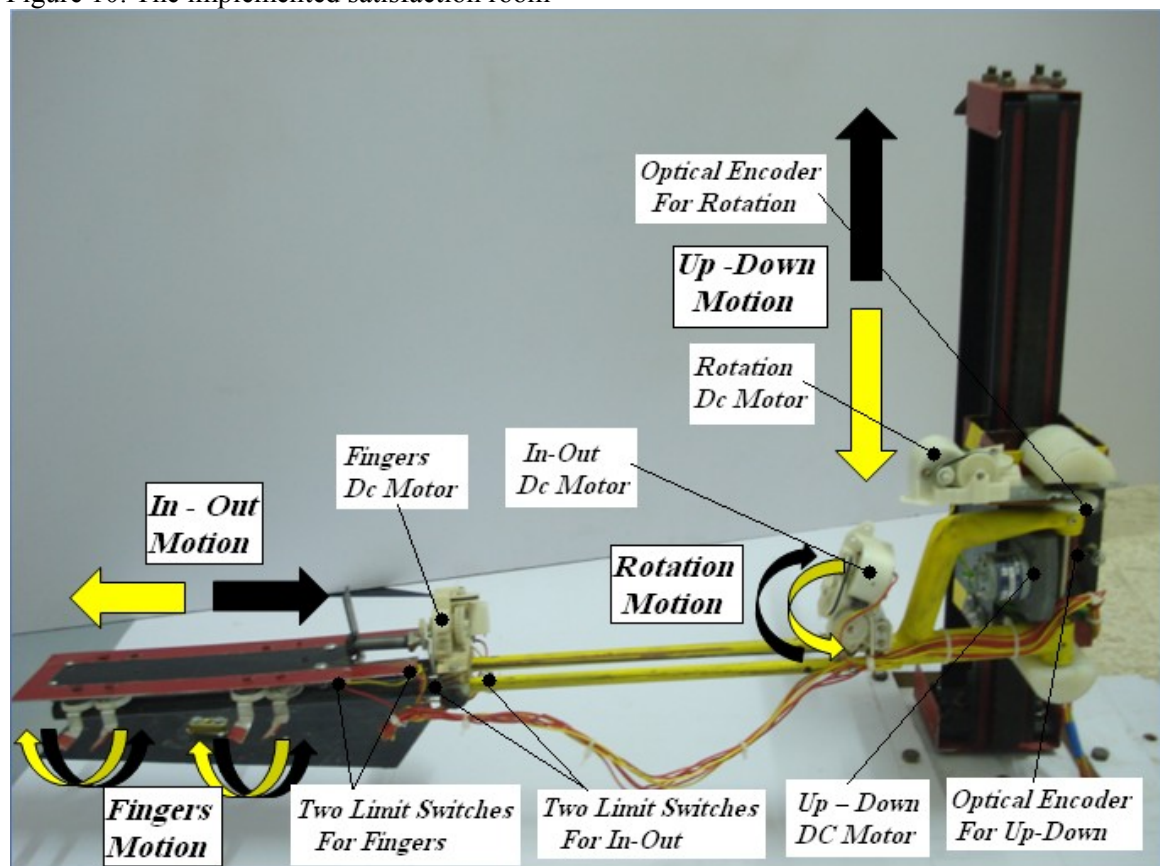


Figure 11. The implemented robot arm

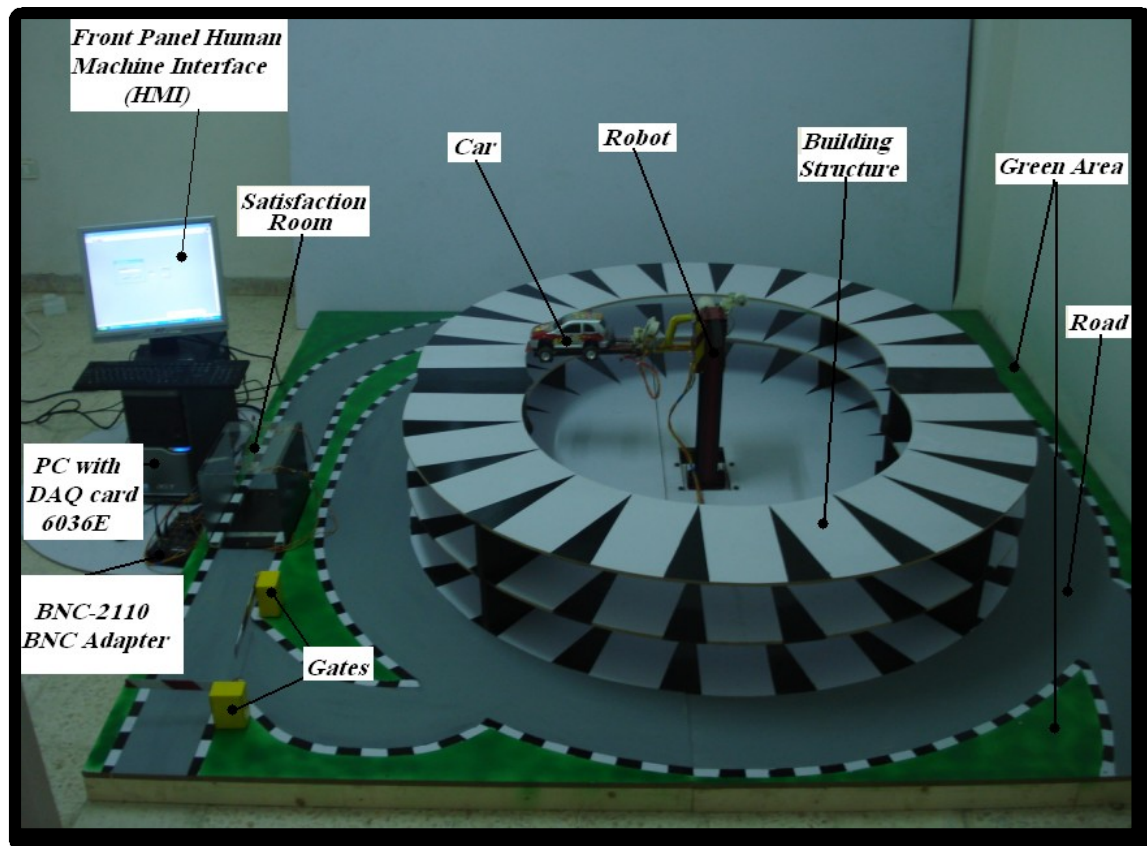


Figure 12. The overall system prototype

7. Conclusions

The mechatronics system design course is one of the most important courses that is studied by mechatronics undergraduate students. It acts as a capstone course that integrates all the knowledge and skills that the student has acquired in the early years, and in many cases prepares the student for the final year graduation project.

This paper has presented an overview of the course contents and the appropriate delivery method by the use of case studies and examples. It is argued that such a design course requires a high level of integration between the various topics and skills and thus requires a special approach in its content and delivery. It should also encourage the student to use practical examples in order to address and solve real life problems.

Prior to discussing the design process itself, selection criteria and methodology for the main components were reviewed. The correct selection procedures of the transducers and actuators for a mechatronics system were reviewed. This is followed by a topic that is rarely discussed in conventional literature: the selection of the physical implementation of the controller and the selection of the suitable control algorithm.. The use of an on/off simple controller for certain applications is emphasized to the students to encourage them to think of simple solutions first where appropriate, before embarking on the more complicated options.

A systemic approach to the design of a mechatronic system was formulated. The steps to be followed in such a design process were detailed. A robotic garage case study was discussed in some detail in order to illustrate how the students applied

the design methodology their final year project. It is argued that such a design process can be applied to any mechatronic system.

Acknowledgment

The authors would like to thank Philadelphia University for hosting and financing the student project. The authors would also like to thank the students who worked on the robotic parking garage projects; **Mohammad I. Al-Qadi and Mohammad S. Al-Zukari.**

References

Ang, K. H. , Chong, G., and Li, Y. 2005. PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, Volume 13, Issue 4, Page(s):559 – 576.

Craig, K., 2001. Is Anything Really New in Mechatronics Education? *IEEE Robotics & Automation Magazine*, pp. 12-19.

Denayer, I., Thael, K., Sloten, J., and Gobin, R., 2003. Teaching a structured approach to the design process for undergraduate engineering students by problem-based education, *European Journal of Engineering Education*, Volume 28, Issue 2, pages 203 – 214.

Ebert-Uphoff, I., Gardner, J. F., Murray, W. R., and Perez, R. 2000. Preparing for the Next Century: The State of Mechatronics Education. *IEEE/ASME Transaction on Mechatronics*, VOL. 5, NO. 2.

Goldberg, J. R., 2009. Design verification in capstone design courses. *IEEE Engineering in Biology and Medicine Magazine*, Jan/Feb.

Gruenther, K., Bailey, R., Wilson, J., Plucker, C., and Hashmi, H. 2009. The influence of prior industry experience and multidisciplinary teamwork on student design learning in a capstone design course” *Design Studies* 30, 721-736.

Jarrah, M. A., 2005. Teaching Mechatronics Design Course for Engineers. *IEEE Workshop on Advanced Robotics and Its Social Impacts*, 2005.

Lee, N., 2009. Project methods as the vehicle for learning in undergraduate design education: a typology, *Design Studies* Vol 30 No. 5.

Saad, A., 2007. Senior capstone design experiences for ABET accredited undergraduate electrical and computer engineering education, *Proceedings of the IEEE Southeastcon*, Volumes 1 and 2 Pages: 294-299

Singhose, W., Vaughan, J., and Mayor, 2009. Use of Design Competitions in Mechatronics Education” *Proceedings of IEEE International Conference on Mechatronics*, Malaga, Spain.

Tutunji, T., Saleem, A., and Rabbo, S. A., 2009. An Undergraduate Mechatronics Project Class at Philadelphia University, Jordan: Methodology and Experience” *IEEE Transactions on Education*, Vol. 52, No. 3.