**HACETTEPE UNIVERSITY**
**COMPUTER ENGINEERING DEPARTMENT**

# UNDERGRADUATE PROJECT FINAL REPORT

| Project Name | Report Date |
|---|---|
| Ball Balancing PID System | 01.01.2018 |

| Student Number(s) | Student Name(s) |
|---|---|
| 21427435<br>21327862<br>21591198 | Ufuk Umut ŞENTÜRK<br>Emre DAĞISTAN<br>Hülya Şermin KARAKAŞ |
| **Supervisor(s)** | **Company Representative(s)** |
| Prof. Dr. M. Önder EFE | |

| Project Coordinator | Report Approval |
|---|---|
| Ayça TARHAN<br><br>Date: 01.01.2018 | ☒ Yes ☐ No<br><br>If no, rational of rejection: _____ |

# A. TECHNICAL RESULTS


## ABSTRACT

PID (Proportional, Integral, and Derivative) controller is control mechanism which is widely used industrial systems. It has such a simple, effective algorithm and we wanted to demonstrate this controller's power in the possibly one of the simplest system which is known as 'Ball and Beam' or 'Ball and Plate'. Basic aim is to balance the ball on the plate that can be inclined by servo motors. That can be achieved with different mechanical parts, processors and techniques. However, we try to choose simplest and effective way to achieve this also can be used with such as Image Processing which is one of the Artificial Intelligence field. Image processing is used to detect the ball and camera is used as sensor for this purpose. Two servo motors, which can be controlled by Arduino microcontroller, are used to incline plate along x-axis and y-axis. System implementation is in Python and by using serial port communication between PC and Arduino, this purpose is achieved.


**Keywords:** PID algorithm, Image Processing, Arduino microcontroller, servo motor, Ball and Plate, Python

## I. INTRODUCTION

This project includes PID algorithm, image processing, mechanical design, Arduino concepts. Problem is to balance the ball on the plate. We mainly propose PID algorithm as a solution for this problem. However, it has many sub-solutions. Firstly, image processing was used for detecting ball and calculating position off ball. To incline plate, we used servo motors and Arduino to control servo motors. We explained implementation details, methods, related works, our mechanical design choice, advantages and disadvantages of our decisions in this project.

## II. BACKGROUND

First of all, this project requires basic concepts of PID algorithm, servo motors and Arduino and physics for mechanical design. S. Bennett's papers gives us a lot of information of PID's existing [3]; in 1939, the Taylor Instrument Companies introduced a completely redesigned version of its Fulscope pneumatic controller. In addition to proportional and reset control actions, this new instrument provided an action which the Taylor Instrument Companies called pre-act. In the same year the Foxboro Instrument Company added Hyper-reset to the proportional and reset control actions provided by their Stabilog pneumatic controller. The pre-act and Hyper-reset actions each provide a control action proportional to the derivative of the error signal. Reset provides a control action proportional to the integral of the error signal and hence both controllers offered PID control. The historical development of these controllers is discussed. Other related works has been done mentioned below section.

## III. RELATED WORK

As we mentioned earlier, this controller has been used widely for more than half of century. It has evolved throughout years and a lot of different versions of it. A lot of papers had been written and has been written. Daniel et al. [1] released paper that gives a sight about PID controller design as internal model control in 1986. Woo et al. [2] proposed PID type fuzzy controller. In order to improve further the performance of the transient state and the steady state of the PID type controller, we develop a method to tune the scaling factors of the PID type fuzzy controller on line. Simulation of the PID type fuzzy controller with the self-tuning scaling factors shows a better performance in the transient and steady state response. However, there are more specific and similar projects like ours. Master thesis of Marta Virseda[4] that is called "Modeling and Control of the Ball and Beam Process" explains physical phenomenon into a set of equations. She stated that two different experimental implementation of the Ball and Beam process. The first step consisted of deriving the equations of motion, that is, to do the mathematical modeling of the process. In order to implement this model Modelica has been used. Modelica, which is a powerful language for modeling of physical systems, uses the tool Dymola. Another model was designed also with Modelica but with the help of the extension of the multi body library, which uses a methodology based on object orientation and symbolic manipulation of equations. With this last model it was possible to visualize an animation in real time 3D. In other paper [5] discusses the conception and development of a ball-on-plate balancing system based on mechatronic design principles. They used really high priced parts and touch-pad for the design. Paper named [6] "Design and Implementation of Ball and Beam system using PID Controller" has main idea that model the ball and beam system considering nonlinear factors and coupling effect and to design Proportional Integral Derivative (PID) controller to control the ball position. They used 1-dof and distance sensor for the ball. Xiao et al. [7] tried to implement adaptive embedded control for a Ball and Plate system. They also used camera to detect ball but used two microcontroller which were burden for system. Margus Espersson's [8] paper thesis' main contribution was the testing and implementation of a variety of vision-based algorithms used to localize the beam, the ball on it and calculate the angle of the beam which we should have done. Because it would give us better implementation and testing environment. In another research, Dinesh Bista [9] presents research and design of a Proportional, Integral, and Derivative (PID) controller that uses a microcontroller (Arduino) platform. The research part discusses the structure of a PID algorithm with some motivating work already performed with the Arduino-based PID controller from various fields. The PID parameters for a particular controller are found manually. The role of different PID parameters is discussed with the subsequent comparison between different modes of PID controllers. The designed system can effectively measure the temperature with an error of ± 0.6℃ while a stable 2 temperature control with only slight deviation from the desired value (set point) is achieved.

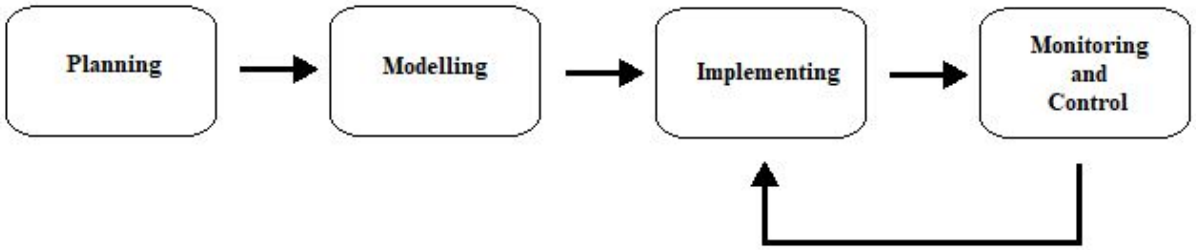
## IV. METHOD

**Planning** : Planning schedule, risks, resources. This phase marked a clear assignment, responsibility for project team members.

**Modelling** : Mathematical and physical modelling of project.

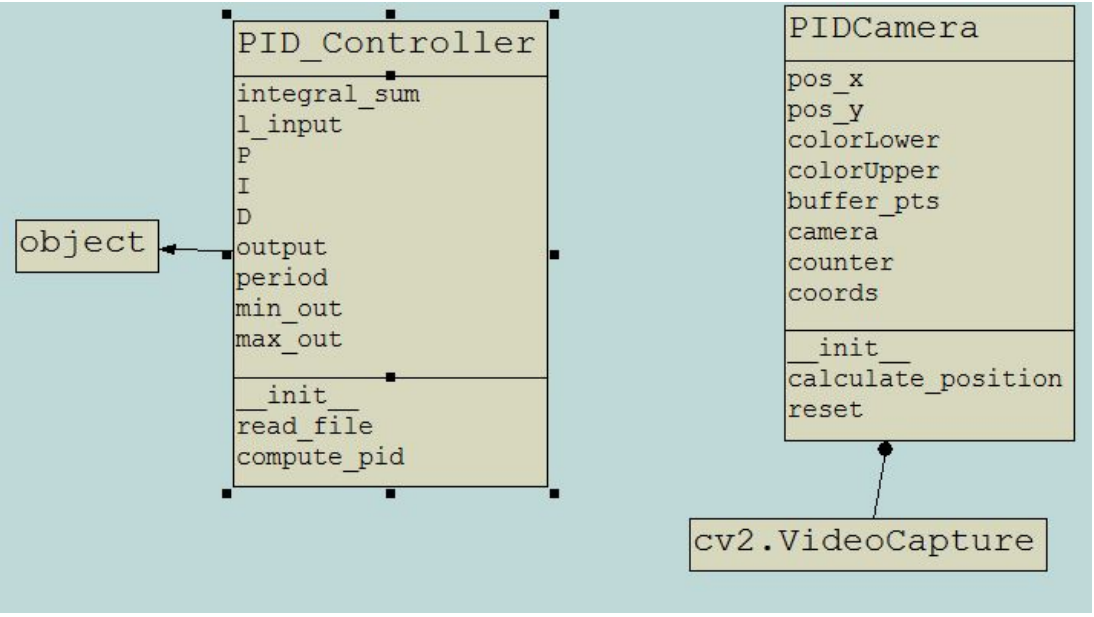
**Implementing** : Implementing required algorithm to modelled structure.

**Monitoring and Control** : Collecting progress updates, test results and discussing by team members on each stage. According to these results, review implementing method to improve the performance of work.

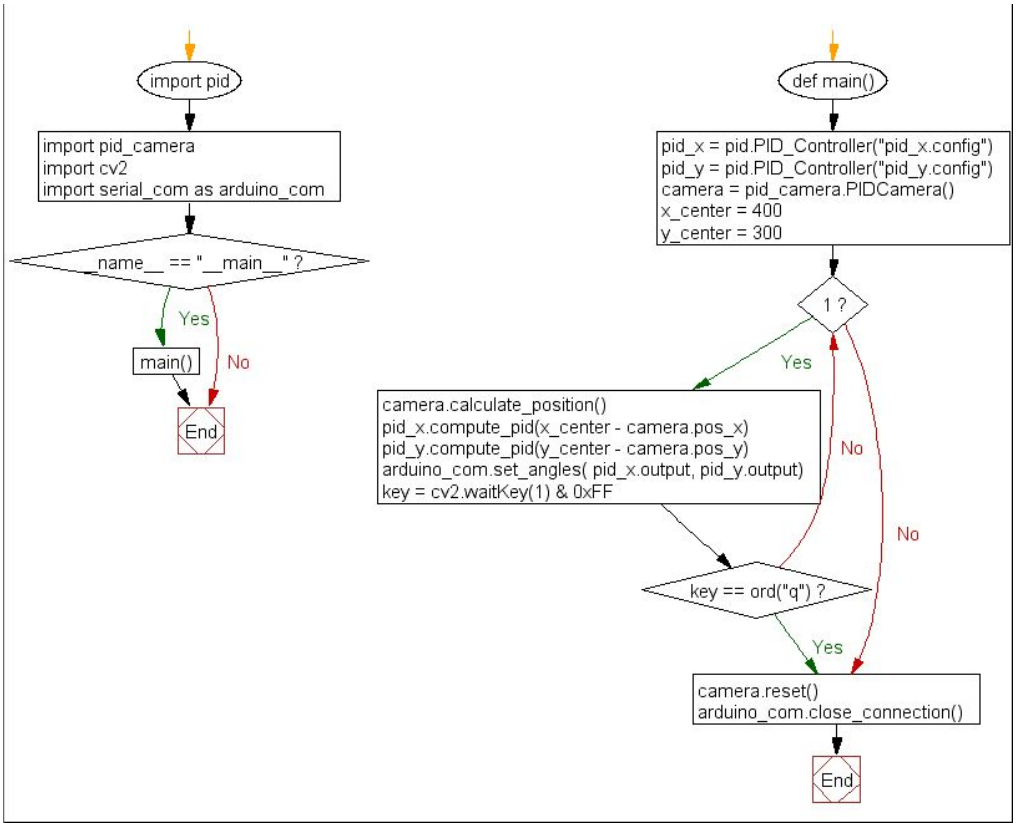



## V. TECHNICAL DESIGN AND CONFIGURATION

In this project, system is designed according to OOP and modularity rules. It has 5 basic parts that are explained below section. Those modules structure remains same even programming language of the project changes which is necessary for other development environments. For instance, if we want import Android environment this project and use that environment as backend, we just change some names and variables but structure remains same. To run program, any PC that runs python 3.x, Arduino IDE and any camera that can connects via USB or can be embedded are necessary. In the below you can see the UML diagrams of two modules;




Above class diagrams also shows structure of pid and pid camera modules. Below flowchart displays main module of the project.




Our input is frames that is acquired from camera. *PID_Camera* object processes this frame and calculate the ball position. Python backend calculates PID value which is servo motors' angles by using ball position for that input and those angles our output which can be seen as servo motors reaction. So far, we can validate by examining mechanical infrastructure of the system and printing output and input values. If camera cannot process any ball position, output would be stay as its last position. If something which has same or similar color with ball comes on the plate, camera tries to process ball or other objects which gives us wrong outputs and system wouldn't balance the ball. However, mechanical plate does not incline so much, that is why ball can't stay on the plate with so much kinetic energy. We should have buy more powerful and bigger servo motors. Other inputs are P, I, D parameters are tuned manually by using Ziegler-Nichols method.

| Parameter Increase | Rise time | Overshoot | Settling Time | Steady-state error |
|---|---|---|---|---|
| Kp | ↓ | ↑ | Small Change | ↓ |
| Ki | ↓ | ↑ | ↑ | Great reduce |
| Kd | Small Change | ↓ | ↓ | Small Change |

Figure 1. Parameter Characteristics on fan's reponse [10]

Above Figure 1. shows us that effects of increasing PID parameters on the fan system. In our case, expected behaviours would not be different from this.

# VI. PROJECT IMPLEMENTATION

Project PC backend is written in Python and Arduino backend is written in Arduino language. It has basically 5 module;

- PID implementation
- Camera implementation
- Serial port implementation for Arduino
- Arduino implementation
- Wrapper module which acts as main module

## PID Module

PID module has class called *PID_Controller* which computes PID value for given error. Object initialization has been done by reading configuration file for each servo motors. It has *P*, *I*, *D* which stand for proportional, integral, derivative fields, *l_input* that keeps previous computed PID value to compute derivative part, *integral_sum* to compute accumulated errors for integral, *min_out* and *max_out* fields to keep angle values of servo motors between those two values, *period* field is used for integral and derivative calculation which stands for period between last 2 calculations.

To compute correct value for PID; proportional, integral and derivative parts are computed separately. Proportional part is computed by amplify error with some P parameter which is proportional gain. Integral part is basically square area calculation. Integral gain, error and period values are multiplied with each other and are added to integral_sum which is accumulated integral value. Derivative part calculates slope between last error and current error and multiply with derivative gain. Thus, sum of those three values gives us PID value that corrects deviation of the system. If result is greater than max_out that it assigns to max_out and it assigns to min_out if it is smaller than min_out. Also, to prevent windup which refers to the situation in a PID feedback controller where a large change in set point occurs (say a positive change) and the integral terms accumulates a significant error during the rise (windup), thus overshooting and continuing to increase as this accumulated error is unwound (offset by errors in the other direction), we added wind-up guard that adjusts integral value to stay between values 20 and -20.

## Camera Module

For camera module, cv2 module which is OpenCV for python is used. Simple image processing technique is used to detecting ball with recognizable color and by creating circle center of the ball can be calculated.

## Serial Port Module

This module is just small Python script which uses serial module of Python. Firstly, port name and baud for connection are declared, after that connection is opened. It has

two functions which are *set_angles* and *close_connection.* First function takes 2 parameters which are PID value for x-axis and y-axis are added to 90 later to adjust angle. While sending those two values some delimiters are used between them, because it makes system faster. Servo motors reacted 3 seconds later to command that is sent by that module when we had not used those delimiters. Second function just calls *close* function of serial module and it closes connection.

## Arduino

Arduino is programmed with a C/C++ 'dialect' and written in Arduino IDE. It has below fields;

```
#include <Servo.h>

const int SERVO_BALANCE_X = 90;
const int SERVO_BALANCE_Y = 90;

const int SERVO1_PIN = 9;
const int SERVO2_PIN = 11;
//:::::::::::::::::::::::::::::::\\

Servo servo1;
Servo servo2;

String inData;  // Buffer for getting data
int servo1_angle = 0;
int servo2_angle = 0;
```

SERVO_BALANCE_X and SERVO_BALANCE_Y stands for initial angles of servo motors. SERVO1_PIN and SERVO2_PIN stands for pin numbers that servo motors are attach to Arduino card. Then, servo motors variables declared.

```
void setup()
{
  Serial.begin(115200);

  servo1.attach(SERVO1_PIN);
  servo2.attach(SERVO2_PIN);

  servo1.write(SERVO_BALANCE_X);
  servo2.write(SERVO_BALANCE_Y);
}
```

Above function is invoked when microcontroller starts and just once while microcontroller runs. It opens connection via baud number, attaches servo motors via their pin number on the card and writes initial angles to the servo motors.

```
void loop()
{
  if (Serial.available() > 0) {

    // Reading angle of the servo1
    inData = Serial.readStringUntil(':');
    servo1_angle = inData.toInt();

    // Reading angle of the servo2
    inData = Serial.readStringUntil('$');
    servo2_angle = inData.toInt();

    // Writing mapped angles to servos
    servo1.write(servo1_angle);
    servo2.write(servo2_angle);
  }

}
```

Above function invokes always in loop structure while microcontroller runs. Firsty, it receives angles as bytes with delimiters, which are ‘:’ and ‘$’ in this case, and parses by their delimiters and writes those values to servo motors.

## Main Module

This module is wrapper module to run Python side of the system. First, *PID_Controller* objects are created via config files which contains field values for object. Then, *PID_Camera* object is created and center of x-axis and y-axis are initialised. Then below part comes;

```
while 1:

    camera.calculate_position()
    pid_x.compute_pid( x_center - camera.pos_x )
    pid_y.compute_pid( y_center - camera.pos_y )

    arduino_com.set_angles( pid_x.output, pid_y.output )

    key = cv2.waitKey(1) & 0xFF
    if key == ord("q"):
        break
```

In a while loop camera object calculates position of ball, then error is calculated by subtracting desired position with current position. That error sends to compute_pid which is method of PID_Controller to calculate pid values and output angles are sent to Arduino via serial port communication.

## VALIDATION AND RESULTS

According to our tests, we successfully implement the PID module to ball balancing system. (You could see the results on Figure 2) In addition, the implementation of the P and PD systems successfully implemented. We got the best results with PD system. We added windup guard to balance I values. Without windup guard, the system overshooted.

When we are setting the P I and D values, we consider the parameters on Figure one which predicts the behaviors of the balancing system.

As a result, the system is working without 100% accuracy. The reason of the noise is the friction of the ball and plate, imperfect P I D values, delay of the system and the low power of the servo motors.

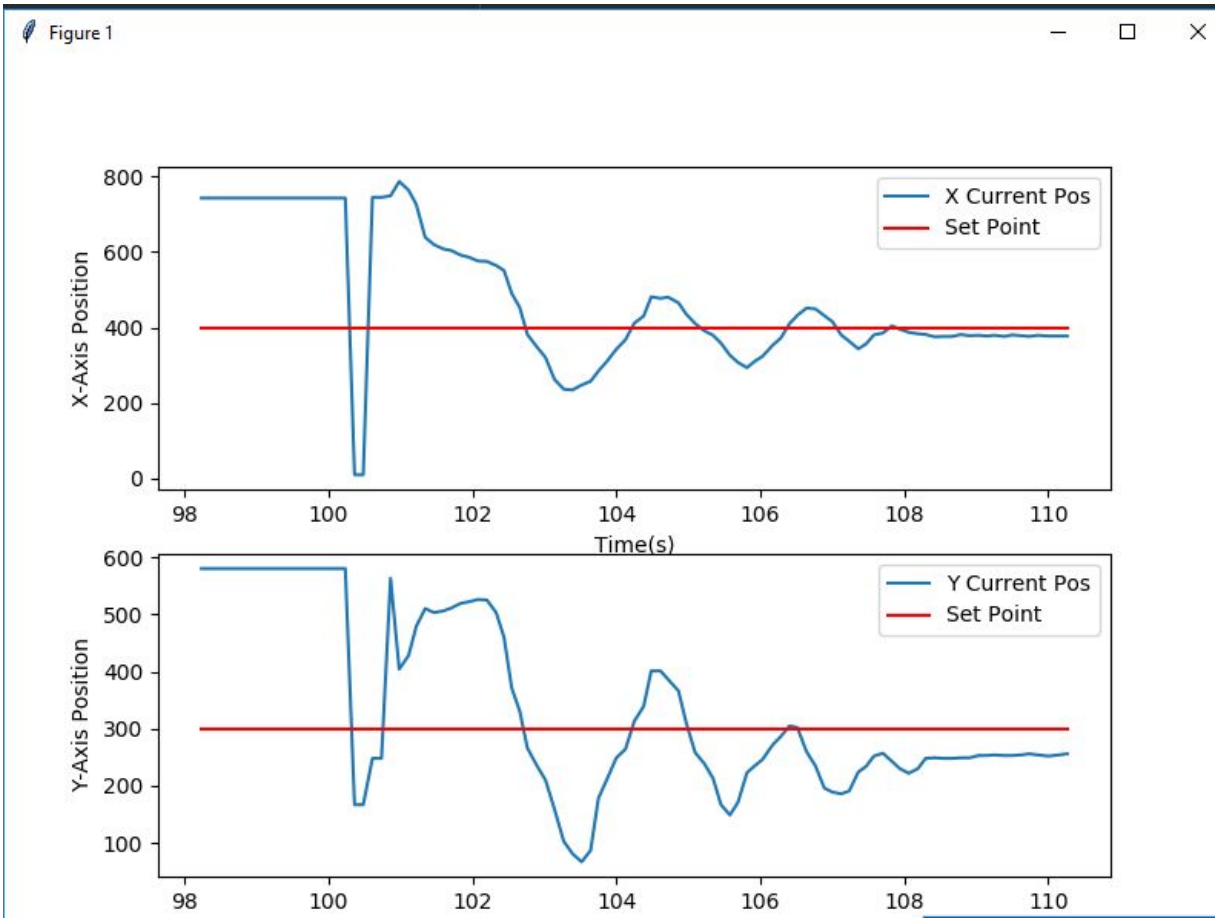



**Figure 2. Test results of the PID system**

## CONTRIBUTION(S) TO INDUSTRY AND ECONOMY

PID loops are one of the simplest yet most effective means to achieve that control on almost anything measurable and regulable e.g pH, temperature, pressure, flow rate, speed, level, position.

A PID controller improves process efficiency, operability, and compliance for sustainable manufacturing. The main benefit of any PID loop is that a designer can "set it and forget it" while still maintaining a well-regulated system. If PID didn't already exist, factory automation would be very limited.

## INNOVATIVE ASPECTS

Most of ball balancing systems use touchscreen. This situation restricts the user to use conductive balls. In our project a camera used for tracking ball so any type of ball can be used.

## REFERENCES


1. http://pubs.acs.org/doi/abs/10.1021/i200032a041?journalCode=iepdaw
2. http://www.sciencedirect.com/science/article/pii/S0165011498001596
3. http://ieeexplore.ieee.org/abstract/document/248006/?reload=true
4. Marta Virseda, Modeling and Control of the Ball and Beam Process, 2004
5. Shorya Awtar, C. Bernard, A. Master. Mechatronic design of a ball-on-plate balancing system.
6. Taifur Ali, Ahmed A.M.Design and Implementation of Ball and Beam system using PID Controller.
7. Jun Xiao, Giorgio Buttazzo. Adaptive Embedded Control for a Ball and Plate System. 2016.
8. Margus Espersson. Vision Algorithms for ball on beam and plate. 2010.
9. Dinesh Bista. Understanding and Design of an Arduino-based PID Controller.2016
10. https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2012/fas57_nyp7/Site/pidcontroller.html

# B. PROJECT RESULTS

## I. CHANGES TO PROJECT PLAN

From the beginning of the project, we didn't change much things. The budget of the project has changed because of the unpredictable product prices.

## II. PROJECT MILESTONES AND OBJECTIVES

| Milestone # | Primary Objective | Due Date | Project Deliverable (if any) | Milestone Achieved? |
|---|---|---|---|---|
| 1 | Detecting the position of the ball by using OpenCV | Nov 3 | -- | **Yes** |
| 2 | Implementing the PID System and mechanics to arduino | Dec 1 | Prototype of the project | Yes |
| 3 | Converting the working principle of the project computer to arduino | Dec 29 | Final project delivery | Yes |

## III. PROJECT PRACTICES AND MEASURES

| Task # | Task Description | Responsibility | Start Date | Finish Date | Success Criteria | Task Succeeded? |
|---|---|---|---|---|---|---|
| 1.1 | Mathematical modelling of system | *Hülya Şermin *Ufuk Şentürk | 16.10.2017 | 03.11.2017 | Observing the mathematical models of the system by graphs | Yes |
| 1.2 | Visual modelling of system | *Emre Dağıstan | 16.10.2017 | 23.10.2017 | Completing the visual models of the project | Yes |
| 2.1 | Learning computer vision (OpenCV) methods | *Hülya Şermin | 16.10.2017 | 03.11.2017 | Being able to use Computer Vision methods | Yes |
| 2.2 | Learning the concepts of the servo motors | *Emre Dağıstan | 16.10.2017 | 03.11.2017 | Being able to use servo motors by arduino programming | Yes |
| 3.1 | Connecting servo motors to arduino | *Emre Dağıstan | 03.11.2017 | 13.11.2017 | Completely connecting servo motors with X and Y axis movements. | Yes |
| 3.2 | Arduino programming | | | | | |
| 3.2.1 | Linking the position values coming from camera to arduino | *Hülya Şermin | 03.11.2017 | 13.11.2017 | Arduino should understand the position of the ball with minimum delay | Yes |
| 3.2.2 | PID system programming | *Ufuk Şentürk | 03.11.2017 | 01.12.2017 | The servo motors should be able to balance the ball by the values coming from camera | Yes |
| 3.2.3 | Using of servo motors by arduino programming | *Emre Dağıstan *Hülya Şermin | 13.11.2017 | 20.12.2017 | The servo motors should make their job by the calculations of the PID system | Yes |

| | | | | | | |
|---|---|---|---|---|---|---|
| 3.3 | Physical montage of the project | *Emre Dağıstan | 20.12.2017 | 01.12.2017 | The physical montage of the project should be completed | Yes |
| 4.1 | Performance and optimization savings | *Ufuk Şentürk | 01.12.2017 | 29.12.2017 | The optimization saving which will observe during project process should improve | Yes |

| Team Member | Task # Under Responsibility | Description of the Work Done |
|---|---|---|
| Hülya Şermin Karakaş | Performance and optimization savings | Optimizing the P I D values and the computer vision codes and increasing the performance of the system. |
| Ufuk Umut Şentürk | Performance and optimization savings | Plotting ball and reference positions to analyze its performance, adding wind-up guard to optimize. |
| Emre Dağıstan | Physical montage of the project | The physical montage of the project should be completed |

## IV. PROJECT BUDGET

| Item # | Description of Income | Date of Income | Planned Amount | Actual Amount | Amount Difference |
|---|---|---|---|---|---|
| 1 | Sponsor (AGE Elektrik Enerjisi Toptan Satış A.Ş.) | 22.12.2017 | 200 □ | 200 □ | --- |

| Item # | Description of Expense | Date of Expense | Planned Amount | Actual Amount | Amount Difference |
|---|---|---|---|---|---|
| 1 | Arduino Mega | 21.10.2017 | 85 □ | 45 □ | 40 □ |
| 2 | Touchscreen | 21.10.2017 | 60 □ | -- | 60 □ |
| 3 | Servo Motor x 2 | 21.10.2017 | 30 □ | 20 □ | 10 □ |
| 4 | Camera | 21.10.2017 | 30 □ | 25 □ | 5 □ |
| 5 | Platform | 21.10.2017 | 10 | 55□ | -45 □ |
| 6 | Jumper | 21.10.2017 | 10 □ | 10 □ | -- |
| 7 | Mafsal | 21.10.2017 | -- | 45 □ | -45 □ |

| Overall Balance | Planned Amount | Actual Amount | Amount Difference |
|---|---|---|---|
| Income | 200 □ | 200 □ | -- |
| Expense | 245 □ | 200 □ | 45 □ |
| **Total** | | -- | |

## V. PROJECT RISKS

| Risk Item # | Description | Probability | Effect | Did It Happen? | How did you handle its occurrence if happened? (Plan-B) |
|---|---|---|---|---|---|
| 1 | Miscommunication between team members | Medium | Medium | No | We meet right after the previous delivery and make a plan of the new tasks' process to handle that risk. |

| | | | | | |
|---|---|---|---|---|---|
| 2 | Lateness of the income | Low | High | No | To handle this risk, we are planning to gather money two day before the expense day. |
| 3 | Delay of the reaching time of position values which comes from the camera | High | High | No | To handle this risk, we are planning to make performance optimizations. In the worst case, we are planning to make project by using touchscreen. |
| 4 | Final delivery takes longer than expected | Low | High | No | To handle this risk, we will try to finish tasks two days before delivery date. |
| 5 | Specification breakdown of a team member | Medium | Medium | No | To handle this risk, we started to plan and manage the software at the very beginning of the project. |

## VI. SELF EVALUATION

The team was successful and completed the task in the expected times. In addition, for the all team members, project process was very informative and innovative. We had some difficulties but with the support of the team member, we had overcome them easily.

## VII. LESSONS LEARNED

We gain technical experience about computer vision techniques, PID control systems, Arduino programming and communicating those with each other. In addition, each team member had a teamwork experience before, but it was all small projects. With this project, we gained much professional project development teamwork experience. Also, we experienced the project management processes like reporting, budget planning etc.