AHMED ITANI

BALL PLATE BALANCING SYSTEM USING IMAGE PROCESIING

NEU 2017

BALL PLATE BALANCING SYSTEM USING IMAGE PROCESSING

A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF NEAR EAST UNIVERSITY

By
Ahmed Itani

In Partial Fulfilment of the Requirements for the Degree of Master of Science in Mechatronic Engineering

NICOSIA, 2017

BALL PLATE BALANCING SYSTEM USING IMAGE PROCESSING

A THESIS SUBMITTED TO THE
GRADUATE SCHOOL OF APPLIED SCIENCES
OF
NEAR EAST UNIVERSITY

By
Ahmed Itani

In Partial Fulfillment of the Requirements for the
Degree of Master of Science
in
Mechatronic Engineering

NICOSIA 2017

AHMED ITANI: BALL PLATE BALANCING SYSTEM USING IMAGE PROCESSING

Approval of Director of Graduate School of Applied Sciences

Prof. Dr. Nadire ÇAVUŞ

We certify this thesis is satisfactory for the award of the degree of Master of Science in Mechatronic Engineering

Examining Committee in Charge:

Assist. Prof. Dr. Kamil Dimililer	Chairperson, Department of Automotive Engineering, NEU
Assist.Prof. Dr. Boran Şekeroğlu	Supervisor, Department of Information Systems Engineering, NEU
Assist.Prof.Dr. Yöney Kırsal Ever	Department of Software Engineering, NEU

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name: Ahmed Itani

Signature:

Date:

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Allah for giving me strength to finish this work. Four semesters passed; I had some good days and other hard days, whenever I was down, Allah was giving me the hope and strength to continue.

My special thanks and heartfelt love would be dedicated to my dearest family for their loyalty and their great confidence in me. I'm greatly indebted to my father who is indeed my inspiration and the man who led me to the treasures of knowledge. I would like to thank my mom for giving me support, encouragement and constant love have sustained me throughout my life.

This thesis would not have been possible without the help, support and patience of my principle supervisor, my deepest gratitude goes to Assist. Prof. Dr. Boran Şekeroğlu for his constant encouragement and guidance. He has walked me through all the stages of the writing of my thesis. Without his consistent and illuminating instruction, this thesis could not have reached its present form.

This research was generously supported by the Department of Mechatronic Engineering of the Near East University. I am grateful to all supporters.

ABSTRACT

This thesis present a ball-plate balancing system using image processing. The aim is to control the position of the ball by tilting the plate. The ball-plate system, with 2 degree of freedom, is using vision feedback as a sensor and DC servo motors as actuators.

This system is an open-loop, nonlinear and unstable system. The target in this thesis is to create a closed-loop controller using PID to calculate the error and to make sure fast communication between sensor and actuator, thus correction of ball position to the center of the plate. The system consists of ball-plate system as plant, two DC servo motors as actuators and a camera as a feedback sensor. Studying the mathematical modeling of this electromechanical system and deriving the transfer function using Lagrange Euler equation, tracking the ball and filtering the frames got from the camera and then designing a PID controller for the system to make sure getting lowest error possible, accuracy and fast system response.

The control system is implemented with ARDUINO to control the servo motors. Image processing and designing the PID controller is done using Processing. In this thesis, a complete design from mathematical modelling to controller design to implementation and testing of results to ensure the designed system is stable and effective.

Keywords: Ball-Plate system; ARDUINO; Processing; PID control system; feedback sensor; image processing

ÖZET

Bu tezde, görüntü işleme teknikleri kullanarak top-plaka dengeleme sistemi sunulmuştur. Plakanın eğimi hesaplanarak, topun pozisyonunu kontrol etmek amaçlanmaktadır. Top-plaka sistemi, sensorden görüntü geribeslemesi alarak ve DC motor ve çalıştırıcı kullanan iki dereceli bir sistemdir.

Sistem, açık döngü, doğrusal olmayan ve kararsız bir sistemdir. Bu tezdeki hedef, PID kullanarak kapalı döngü kontrolörü yaratarak hata hesaplaması yapmak ve sensör ve çalıştırıcılarla hızlı iletişim kurarak topun pozisyonunu plaka üzerinde merkez noktaya gelecek şekilde düzeltmektir. Sistem teçhizat olarak çalıştırıcı konumunda iki DC motor ve geri besleme sensorü olarak da kamera içermektedir. Elektromekanik sistemin matematiksel modellemesi üzerine çalışılmış ve Euler-Lagrange denklemi kullanılarak transfer fonksiyonu türetilmeye çalışılmıştır. Topun kontrolü, kameradan gelen resimlerin filtrelenmesi ve PID kontrolör tasarımı yapılıp, en düşük hata, en yüksek doğruluk derecesi ve hızlı sistem tepkisi elde edilmeye çalışılmıştır.

Kontrol sistemi ARDUINO üzerinden uygulanmıştır. Görüntü İşleme ve PID kontrolör tasarımı gerçekleştirilmiştir. Tezde, matematiksel modellemeden kontrolör tasarımına kadar gerçek zamanlı bir tasarım yapılmış ve test sonuçları ile sistemin kararlılığı ve verimliliği gösterilmiştir.

Anahtar Kelimeler: Top-Plaka Sistemi; ARDUINO; PID kontrol sistemi; Geri-besleme Sensörü; Görüntü İşleme

TABLE OF CONTENTS

CHAPTER 4: HARDWARE PART

CHAPTER 5: SYSTEM DESIGN

CHAPTER 6: CONCLUSION

APPENDICES

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1.Introduction

Digital image processing is now widely popular and growing field where its used in medicine, military, video production, security, tracking objects, and remote sensing. Generally, the examination of moving objects, regardless of whether they are fire fronts, particles, beads, or liquid interfaces, was done physically, more often than not by measuring elements of a picture anticipated on a divider. This manual investigation was dreary and experienced numerous inadequacies, including poor precision and poor repeatability. One of the advantages of machine vision is the ability of sensing moving objects from distance and without contact with the object. However, the data in the image can be easily affected by the disturbance and changes like sunlight and night that occur in the surrounding area of the object being sensed. Such issue can be overcome by using digital image processing and advanced image filtering techniques to get rid of such noises and disturbances. The balancing of objects is a major issue which is still under study and development especially by robots manufacturers. The goal of this thesis is to develop a ball-plate balancing system where the ball is sensed using camera and not a resistive touch screen which is expensive.

The ball-plate system is a nonlinear and unstable open-loop system. This system has 2 degree of freedom (DOF). The objective is to balance the ball in a predefined coordinate or to follow a specific center with falling of the horizontal plate. The ball position will be controlled indirectly by tilting the angle of the plate in a 2D direction, x-axis and y-axis, and its controlled by two DC servo motors. Studying the mathematical modeling of this electromechanical system using Lagrangian-Euler formula of motion is done in order to find the transfer function of the system and thus design a good controller to make sure getting the perfect control of the ball's position.

The vision system (camera) will capture continuous frames that will be processed using Processing in order to get the coordinates of the ball. The servo motors will be controlled

using ARDUINO which controlled by a software developed using Processing which is cable of image processing, designing a controller and with a position feedback.

Characterizing ball-plate system specifications is difficult since the controller is not directly affecting ball and thus the designing of the controller and calculating its numerical value will be difficult. Nowadays the ball and plate system has been controlled by many control methods such as PID control, fuzzy control and so on. However, the Proportional Integral Derivative (PID) controller proved its efficiency in the world of controls and in the industrial field. The feedback controller (PID) will decrease the errors in controlling the ball's position and trajectory. It will also control the overshoot resulted from the jump in the reference signal and thus controlling the large input to the servo motors in order to keep the ball on the plate. In this thesis, an optimal PID controller will be designed with quick rise time, rapid settling, and negligible steady-state error.

1.2.Aim of the Thesis

The aim of this thesis is to study the mathematical modeling of the ball-plate system in order to design a good PID controller with visual position feedback so that the ball can be balanced at the center of the plate or follow a trajectory.

The objectives of the thesis are:

- To apply optimal control method such as proportional integral derivative control to control the ball-plate system for both static and trajectory tracking of the ball.
- To study the mathematical model of the ball-plate system for controller design.
- To develop the image processing system using Processing.
- To develop the servo motors control software using ARDUINO.
- To select the appropriate actuators parameters and mechanical structure for the control of ball-plate system.

CHAPTER 2
THEORETICAL FRAMEWORK

2.1.Mathematical Modeling

In this section, the mathematical modeling of the ball-plate system and derivation of the motion equations will be presented, assuming the following assumptions (Brill, Frank, & Kapola, 2016):

- There is no loss of contact area with the ball
- All frictions are neglected
- The ball is homogenous and symmetric.
- Connection of the motor to tilt the axis is perfectly rigid.
- The ball is not slipping

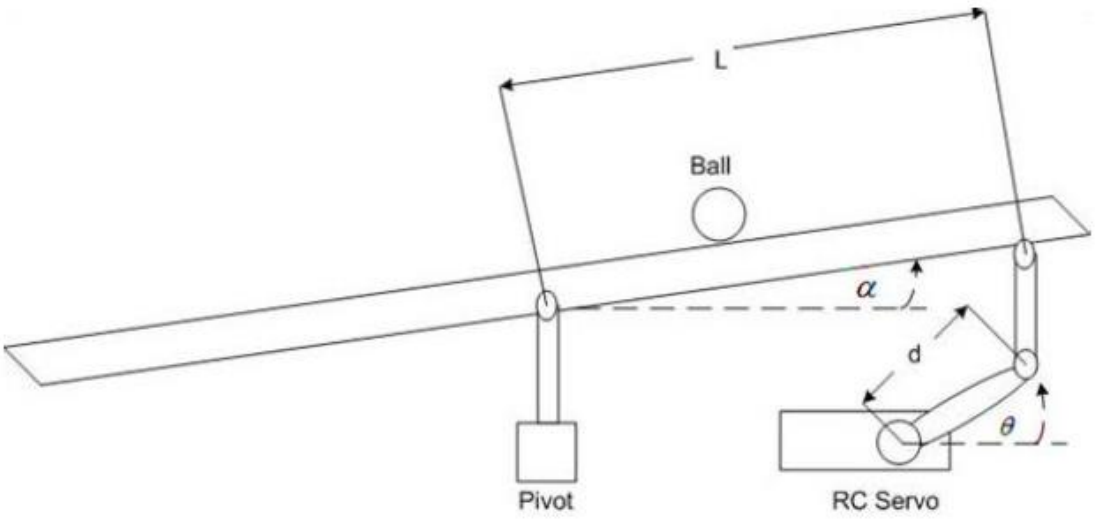


Figure 2.1: Ball and plate system

The dynamical equations of ball-plate system, shown in figure 2.1, will be derived based on Lagrangian formula. The Euler-Lagrange equation of ball-plate system is (Cheng, & Tsai, 2016):

$$\frac{d}{dt}\frac{\partial T}{\partial q_i'} - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i \qquad (2.1)$$

Where

q_i: stands for i-direction coordinate.

T: kinetic energy of the ball-plate system.

V: potential energy of the ball-plate system.

Q: composite force.

The system has 2 degree of freedom. Assume (x_b, y_b) coordinates to be the ball's position on the plate and (α and β) to be the inclination of the plate. Moreover, assume the center of x-y coordinates to be the center of the plate (Buttazo, & Xiao, 2016).

The kinetic energy of the ball consists of translational energy and rotational energy with respect to its center:

$$T_b = \frac{1}{2} m_b (x_b'^2 + y_b'^2) + \frac{1}{2} I_b (\omega_x^2 + \omega_y^2) \quad (2.2)$$

Where

m_b: mass of the ball.

I_b: moment of inertia

x'_b: translational velocity along x-axis.

y'_b: translational velocity along y-axis.

ω_x: ball's rotational velocity along x-axis.

ω_y: ball's rotational velocity along y-axis

r_b: ball radius

The following formulas are the relation between rotational and translational velocities:

$$x_b' = r_b \omega_y \quad (2.3)$$

$$y_b' = r_b \omega_x \quad (2.4)$$

By substituting equation (2.3) and (2.4) into equation (2.2), the following equation result:

$$T_b = \frac{1}{2}\left[m_b (x_b'^2 + y_b'^2) + \frac{I_b}{r_b^2}(x_b'^2 + y_b'^2)\right] = \frac{1}{2}(m_b + \frac{I_b}{r_b^2})(x_b'^2 + y_b'^2) \quad (2.5)$$

The kinetic energy of the plate consists of its rotational energy with respect to its center of mass. The equation is as follow

$$T_p = \frac{1}{2}\left(I_p + I_b\right)(\alpha'^2 + \beta'^2) + \frac{1}{2}m_b(x_b\alpha' + y_b\beta')^2 \quad (2.6)$$

$$T_p = \frac{1}{2}\left(I_p + I_b\right)(\alpha'^2 + \beta'^2) + \frac{1}{2}m_b(x_b^2\alpha'^2 + 2x_b\alpha' y_b\beta' + y_b^2\beta'^2) \quad (2.7)$$

Where

x_b: ball's position on the plate with respect to x-axis.

y_b: ball's position on the plate with respect to y-axis.

α: plate's angle of inclination along x-axis.

β: plate's angle of inclination along y-axis.

As a total, calculate the kinetic energy of the whole system as followings:

$$T = T_b + T_p = \frac{1}{2}\left(m_b + \frac{I_b}{r_b^2}\right)(x_b'^2 + y_b'^2) + \frac{1}{2}\left(I_p + I_b\right)(\alpha'^2 + \beta'^2)$$

$$+\frac{1}{2}m_b(x_b^2\alpha'^2 + 2x_b\alpha' y_b\beta' + y_b^2\beta'^2) \quad (2.8)$$

The potential energy of the ball can be calculated as:

$$V_b = m_b gh = m_b g(x_b sin\alpha + y_b sin\beta) \quad (2.9)$$

Here the system equations can be derived by

$$L = T_b + T_p - V_b \quad (2.10)$$

Use L to derive system's equations:

$$\frac{\partial T}{\partial \alpha'} = (I_p + I_a)\alpha_x' + m_b x_b(x_b\alpha' + y_b\beta') \quad , \quad \frac{\partial L}{\partial \alpha} = mgcos\alpha \quad (2.11)$$

$$\frac{\partial T}{\partial \beta'} = (I_p + I_a)\beta_x' + m_b y_b(y_b\beta' + x_b\alpha') \quad , \quad \frac{\partial L}{\partial \beta} = mgcos\beta \quad (2.12)$$

$$\frac{\partial T}{\partial x_b'} = (m_b + \frac{I_b}{r_b^2})x_b' \quad , \quad \frac{\partial L}{\partial x_b} = m_b(x_b\alpha' + y_b\beta')\alpha' \quad (2.13)$$

$$\frac{\partial T}{\partial y_b'} = (m_b + \frac{I_b}{r_b^2})y_b' \quad , \quad \frac{\partial L}{\partial y_b} = m_b(x_b\alpha' + y_b\beta')\beta' \quad (2.14)$$

Assume generalized torques as τ_x and τ_y which are exerted torques on the horizontal plate.

Using Euler- Lagrange equation:

$$\frac{d}{dt}\frac{\partial T}{\partial \alpha\prime} - \frac{\partial L}{\partial \alpha} = \left(I_p + I_b\right)\alpha" + m_b x_b^2\alpha" + 2m_b x_b x_b'\alpha' + m_b x_b y_b\alpha" + m_b x_b' y_b\beta' +$$

$$m_b x_b y_b'\beta' - mgcos\alpha = \tau_x \quad (2.15)$$

$$\frac{d}{dt}\frac{\partial T}{\partial \beta\prime} - \frac{\partial L}{\partial \beta} = \left(I_p + I_b\right)\beta" + m_b y_b^2\beta" + 2m_b y_b y_b'\alpha' + m_b x_b y_b\beta" + m_b y_b' x_b\alpha' +$$

$$m_b y_b x_b'\alpha' - mgcos\beta = \tau_y \quad (2.16)$$

$$\frac{d}{dt}\frac{\partial T}{\partial x_b'} - \frac{\partial L}{\partial x_b'} = \left(m_b + \frac{I_b}{r_b^2}\right) x_b^{"} - m_b(x_b\alpha' + y_b\beta')\alpha' + m_b g sin\alpha = 0 \qquad (2.17)$$

$$\frac{d}{dt}\frac{\partial T}{\partial y_b'} - \frac{\partial L}{\partial y_b'} = \left(m_b + \frac{I_b}{r_b^2}\right) y_b^{"} - m_b(y_b\beta' + x_b\alpha')\beta' + m_b g sin\beta = 0 \qquad (2.18)$$

So the non-linear differential equations of motion for the ball-plate-system are as follows:

$$\left(m_b + \frac{I_b}{r_b^2}\right) x_b^{"} - m_b(x_b\alpha'^2 + y_b\alpha'\beta') + m_b g sin\alpha = 0 \qquad (2.19)$$

$$\left(m_b + \frac{I_b}{r_b^2}\right) y_b^{"} - m_b(y_b\beta'^2 + x_b\alpha'\beta') + m_b g sin\beta = 0 \qquad (2.20)$$

$$\tau_x = \left(I_p + I_b + \mathrm{m_b}x_b^2\right)\alpha + 2\mathrm{m_b}\mathrm{x_b}\mathrm{x_b^{'}}\alpha^{'} + \mathrm{m_b}\mathrm{x_b}\mathrm{y_b}\alpha + m_b x_b' y_b\beta' + m_b x_b y_b'\beta' - mgcos\alpha \qquad (2.21)$$

$$\tau_y = \left(I_p + I_b + \mathrm{m_b}y_b^2\right)\beta + 2\mathrm{m_b}\mathrm{y_b}\mathrm{y_b^{'}}\alpha^{'} + \mathrm{m_b}\mathrm{x_b}\mathrm{y_b}\beta + m_b y_b' x_b\alpha' + m_b y_b x_b'\alpha' - mgcos\beta \qquad (2.22)$$

The equation (2.19) and (2.20) shows the relation between plate's state and ball's state which represent the plate's inclination. The equation (2.21) and (2.22) shows the effect of external torque on the plate-ball system.

Where

g: gravitational acceleration

L_p: side length of the plate

r_a: length of the servo arm.

The relationship between α and θ_x is as follows:

$$\alpha = \frac{r_a}{L_p}\theta_x \qquad (2.23)$$

The relationship between β and θ_y is the same as equation (2.23) since the system is symmetrical, so the equation is as follows:

$$\beta = \frac{r_a}{L_p}\theta_y \qquad (2.24)$$

To help simply the equations for modeling, some assumptions will be taken into consideration; so, the friction will be neglected and no slippage between the plate and the ball.

When the angular velocity α' and β' is low, it can be approximated as follows:

$\alpha'\beta' = 0$, $\alpha'^2 = 0$, $\beta'^2 = 0$

By linearizing the equations of motion for $\theta_x = 0$ and $\theta_y = 0$, and after substituting equation (2.23) and (2.24) into motion equations (2.19) and (2.20), the result is as follows:

$$\left(m_b + \frac{I_b}{r_b^2}\right) x_b^{"} + \frac{m_b g r_a}{L_p} \theta_x = 0 \quad (2.25)$$

$$\left(m_b + \frac{I_b}{r_b^2}\right) y_b^{"} + \frac{m_b g r_a}{L_p} \theta_y = 0 \quad (2.26)$$

Since the ball used is a solid sphere ball, then the moment of inertia of the ball is equal to $I_b = \frac{2}{5} m r^2$.

2.1.1. Transfer Function

Taking the Laplace transform of equations (2.25) and (2.26)

$$\left(m_b + \frac{I_b}{r_b^2}\right) X_b(s) + \frac{m_b g r_a}{L_p} \theta_x(s) = 0 \quad (2.27)$$

$$\left(m_b + \frac{I_b}{r_b^2}\right) Y_b(s) + \frac{m_b g r_a}{L_p} \theta_y(s) = 0 \quad (2.28)$$

By linearization equations (2.25) and (2.26), the differential equations for x-axis and y-axis are respectively (Ehsan, 2015). Let's focus on x-axis to find its transfer function P(s). Assuming that θ_x is the input and X_b is the output, so the transfer function is as follows:

$$P(s) = \frac{output}{input} = \frac{X_b(s)}{\theta_x(s)} = -\frac{m_b g r_a}{L_p (m_b + \frac{I_b}{r_b^2}) s^2} \quad (2.29)$$

Table2.1: Ball-plate system parameters

Parameters	Numerical values
m_b	0.0066 [kg]
r_b	0.02 [m]
G	9.81 [m/s^2]
I_b	7.2×10^{-7} [kg.m]
L_p	0.3 [m]
r_a	0.02[m]

2.2.PID Controller Design

A PID controller is proposed to control the system whose transfer function is $P(s) = -\frac{m_b g r_a}{L_p(m_b+\frac{I_b}{r_b^2})s^2}$. The general PID control formula is as follows (Li-Ming, Ming-Tzu, & Yusie, 2013).

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \quad (2.30)$$

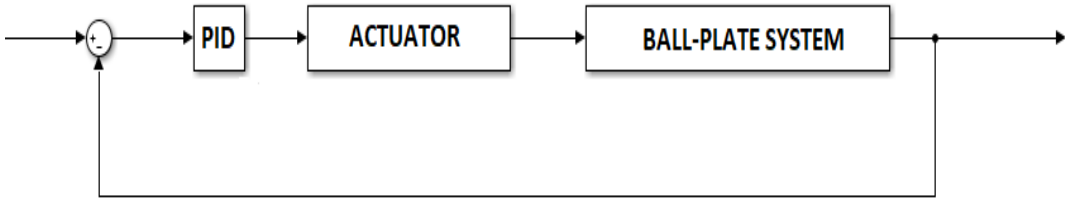


Figure 2.2: Diagram of the closed loop plate-ball system

Figure 2.2 shows the closed loop feedback diagram.

The transfer function of the PID controller in Laplace domain is as follows

$$K(s) = K_p + K_i \frac{1}{s} + K_d s \quad (2.31)$$

Where

Kp: proportional gain that is proportional to the error

Ki: integral gain that interprets for past errors.

Kd: derivative gain that interprets for future errors.

The transfer function of the open loop is given as follows:

$$L(s) = P(s)K(s) \quad (2.32)$$

CHAPTER 3
IMAGE PROCESSING

3.1.Digital Image Processing

Image processing is a mode used to promote frank images received from medical and military applications, satellites, and security cameras. The image is processed and analyzed using several techniques like image enhancement which include brightening, sharpening edge enhancement, etc. Majority of image processing techniques treating with two-dimensional images. Image processing is indicative processing where the input is a picture or video frames and the output is a picture or parameters related to it.

Picture increase is the process of beneficent the type of digital images so that it can be interpreted by human or computers. Image enhancement include many algorithms such as filtering, extraction, correlation, and time compression. There are mainly two methods for image enhancement: first one deals with image in frequency domain and the other one deals with image in spatial domain. The frequency domain is based on Fourier series transformation and the other one is based on processing of individual pixels of the image.

In inferior inequality image, the relative characters combine pending binarization. It will use Power-Law Transformation to reduce the spread of characters before thresholding which raise the disparity of particle and support in preferable image segmentation. The Power-Law Transformation equation is given by $s = cr^{\gamma}$, which r and s are respectively the input and output density, c and γ are positive. The indicator in the power-law equation (γ) is indicate to as gamma correction.

Processing is a high-execution and advanced parlance for solving artistic calculating issues. Processing has a progressing data structure, include built-in marking, simulation, and debugging stuffs which make him as an excellent programming software for teaching and research purposes.

Computerized picture preparing is the innovation of applying various PC calculations to handle advanced pictures. Results of this procedure can be either pictures or an arrangement of agent qualities or properties of the first pictures. Utilizations of advanced picture preparing have been ordinarily found in mechanical autonomy/astute frameworks, therapeutic imaging, remote detecting, photography and criminology (Zhou., 2010).

Flag handling is a teach in electrical designing and in science that arrangements with examination and preparing of simple and computerized signals, and manages putting away, separating, and different operations on signs.

Out of every one of these signs, the field that arrangements with the kind of signs for which the information is a picture and the yield is additionally a picture is done in picture preparing. As it name proposes, it manages the handling on pictures. It can be additionally isolated into simple picture handling and advanced picture preparing.

The expression digital image processing mostly indicates to transformation of a couple-dimensional image by a digital PC. In a floppy condition, it suggests digital processing of any couple-dimensional data. A digital image is a disposition of true numbers demonstrated by a limited number of sting. The standard feature of Digital Image Processing procedures is its variation, repeatability and the conservation of original data accuracy.

3.1.1. RGB to GRAY Conversion

Humans perceive colors through wavelength-sensitive cells called cones. There are three deferent types of cones cells that can detect different electromagnetic radiations. One is sensitive to green light, one to blue light, and one to red light. The combination of these three colors can generate any other color which is also detectable by these three types of cells. the stored color image is called an RGB image. In grayscale images, the total amount of emitted light for each pixel can be differentiated and calculated where the pixels are divided between dark and bright pixels (Ratan, 1999).

Processing RGB images is very complex compared to grayscale images although processing colored imaged can provide better results. There are two ways to convert RGB images to grayscale. First, average method is the simplest method, shown in figure 3.1, where the

average of the three colors (R+G+B/3) is taken. However, since the three colors have different wavelength, the resulting image will be very dark. Consequently, weighted method is better since the conversion depends on the wavelength of each color alone and thus the following equation$[(0.3 \times R) + (0.59 \times G) + (0.11 \times B)]$ is resulted. As a result, green contribute in 59%, red contribute in 30%, and blue contribute in 11%.

a) Original image before gray scale

b) Resulting image after grayscale

Figure 3.1: Image RGB to grayscale

3.1.2. Image Segmentation

Division is one of the opener issues in picture handling. A well-known technique utilized for picture division is thresholding. After thresholding, a paired picture is shaped where all protest pixels had only one dim scale with all foundation pixels take added - by and large the question pixels are "dark" and the foundation is 'candid '. The preferable edge is the one that chooses all the protest pixels and charts them to 'dark'. Different methodologies for the programmed choice of the limit have been suggested. Starting able to be characterized as charting of the dark scale in the double set {0, 1}:

$$S(x,y) = \begin{cases} 0 \; if \; g(x,y) < T(x,y) \\ 1 \; if \; g(x,y) \geq T(x,y) \end{cases} \quad (3.1)$$

which S(x, y) is the estimation of the portioned picture, g(x, y) is the dim scale of the pixel (x, y) and T(x, y) is the limit an incentive at the directions (x, y). In the least difficult state T (x, y) is facilitate free and a consistent for the entire picture. It can be chosen, for example, on the premise of the dim scale histogram. At the point when the histogram has a pair of articulated maxima, which contemplate dark scales of theme (s) and foundation, it is conceivable to choose a solitary limit for the whole picture.

A technique which depends on this thought and uses a relationship foundation to choose the best edge, is portrayed underneath. Occasionally dark level histograms have just a single greatest. This must be brought on, e.g., by inhomogeneous light of different areas of the picture. In this situation, it is difficult to choose a solitary verge esteem for the whole picture and a neighborhood binarization method (portrayed beneath) have be connected. Universal strategies to take care of the issue of binarization of in homogeneously lit up pictures, be that as it may, are not accessible.

Division of pictures includes once in a while not just the segregation amongst items and the foundation, additionally partition between various districts.

The issues of picture division and gathering stay extraordinary difficulties for PC vision. Since the season of the Gestalt development in brain research (Wertheimer., 1938)], it has been realized that perceptual gathering assumes a capable part in human visual per-1 caption. An extensive variety of computational vision issues could on a basic level make great utilization of fragmented pictures, were such divisions dependably and effectively

calculable. For example, middle level vision issues, for example, stereo and movement estimation require a fitting district of support for correspondence operations. Spatially non-uniform districts of support can be distinguished utilizing division procedures. More elevated amount issues, for example, acknowledgment and picture ordering can likewise make utilization of division brings about coordinating, to address issues, for example, figure-ground detachment and acknowledgment by parts.

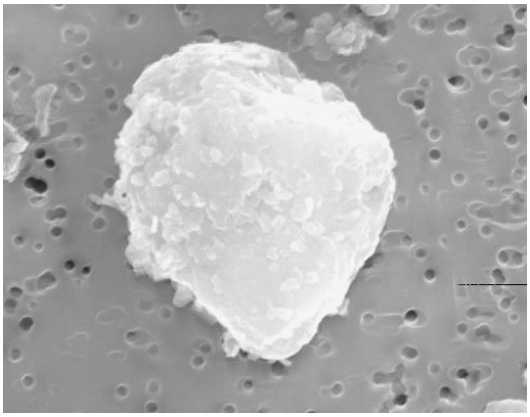
While the previous couple of years have seen extensive improvement in eigenvector-based strategies for picture division (Weiss., 1999), these techniques are too ease back to be in any way functional for some applications. Interestingly, the strategy portrayed in this paper has been utilized as a part of extensive scale picture database applications as depicted in (Rattan et al., 1999). While there are different ways to deal with picture division that are very productive, these techniques by and large neglect to catch perceptually critical non-neighborhood properties of a picture as talked about beneath. The division method created here both catches certain perceptually essential non-neighborhood picture qualities and is computationally 2 effectives – running in O (n log n) time for n picture pixels and with low consistent variables, and can keep running practically speaking at video rates.

3.1.2.1. Global Thresholding using a Correlation Criterion

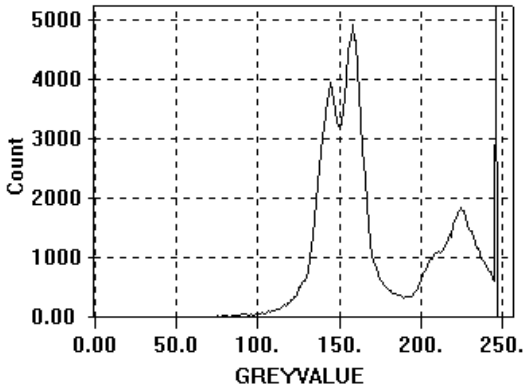
Regarding the verge trouble, lease g symbolizes the potential leaden amount in the main picture. Those amounts are described by the beneath- and above-verge instrumentation $\mu_0(T)\&\mu_1(T)$of the main picture, presented with:

$$\mu_0(T) = \sum_{g=0}^{T} gp_g(\sum_{g=0}^{T} p_g)^{-1} \; and \; \mu_1(T) = \sum_{g=T+1}^{n} gp_g(\sum_{g=T+1}^{n} p_g)^{-1} \quad (3.2)$$

Which g = {0, 1, n} all gray amounts with T (0 <T <n) is the verge scale. The eventuality compilation pg of gray amounts g is assumed by p f N g g= wherever N is the full amount of pixels in the picture and f g is the amount of pixels processing gray amount g.



a) Subordinate electron SEM picture of a municipal powder element



b) Gray scale histogram

Figure 3.2: The producer of stratify the connection standard setup

The difference represented by

$$V_x = E_{xx} - (E_x)^2, V_y(T) = E_{yy}(T) - \left(E_y(T)\right)^2 \quad (3.3)$$

Actually, Ex,Exx & Vx are separated of the verge T that they are acquired from the main image. The connection degree given by

$$P_{xy}(\mathrm{T}) = \frac{E_{xy}(T) - E_x E_y(T)}{\sqrt{V_x V_y(T)}} \quad (3.4)$$

It is today a task from the verge scale. That optimum amount of T coincides to the amount that make the most of the connection through the true and the second amount pictures. This amount is set up by repetition. The effect of stratifying the mechanism to a subordinate electron SEM picture of a civilian powder character. As visible from the model, the

connection procedure elected a significate verge, where figure 3.2 shows the producer of stratify the connection standard setup.

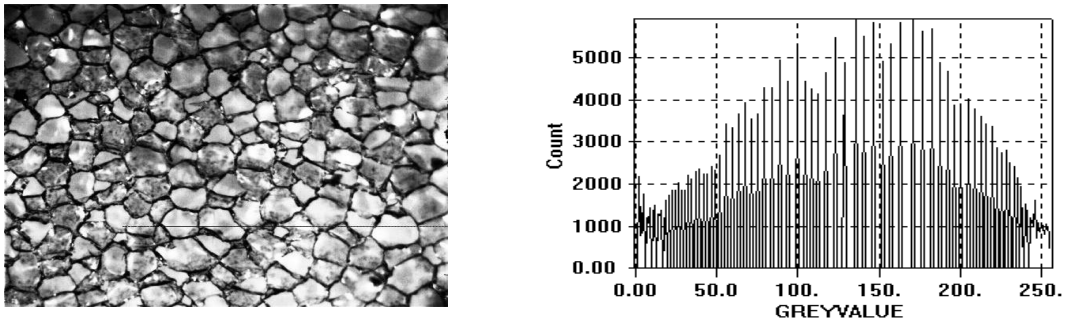
3.1.2.2. Local Binarization using Discrete Convolution

This mode of binarization is established on the implementation of the separated complication refinement mechanism that create a changed picture that is able to be simple threshold using 1 as the verge amount.

For resolution creating near a proper mathematical amount of stricture *p*, the grade of connection among the main and the involute pictures is applied. It is elaborated as tracks:

$$r(p) = \frac{Cov(f,g(p))}{\sqrt{Var.Var(g(p))}} \quad (3.5)$$

Which *f* and *g*(*p*) are the main gray scale image and the image involute by means of amount for the limit, correspondingly. Figure 3.3 shows the four stages of image binarization.



a) Grayscale image b) The gray level histogram

c) Global thresholding d) Local thresholding

Figure 3.3: The 4 steps to the image binarization

3.1.3. Background Subtraction

Digital image background subtraction have been used widely in computer vision applications and other image processing related fields. Background subtraction is the technique used to subtract data from two images in which the pixels values in the two images is subtract. Image subtraction is good for detecting differences in a series of images in order to detect and track an object such as cars, humans, etc. It compares the previous frame from the current frame by using different techniques such as basic motion detection, Gaussian mixture model and Kernel density estimation. Most background subtraction techniques label every pixel in the frame and delete those who have the same value from the background image. Mainly object motion detection begins with image thresholding in order to segment the object from the background. First a background image is captured and then taking frames at time t in order to subtract them. The simplest technique, is to get the value of each pixel in the frame and subtract it from its corresponding pixel value in the background image.

The equation of image subtraction is as follows

$$P\big(I(t)\big) = P\big(F(t)\big) - P(B) \tag{3.6}$$

Where

I(t): result image at time t

F(t): frame image obtained at time t

B: background image

The results of this equation will present a picture where it will show the intensity where the pixel values have changed in the two consecutive frames.

a) Original image

b) Result of background subtraction

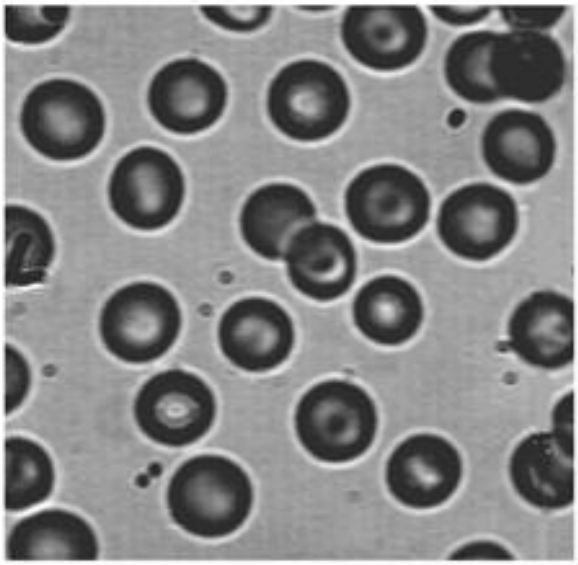
Figure 3.4: Image background subtraction

An example of image background subtraction is shown in figure 3.4, where after doing this subtraction, the people in the image were detected.

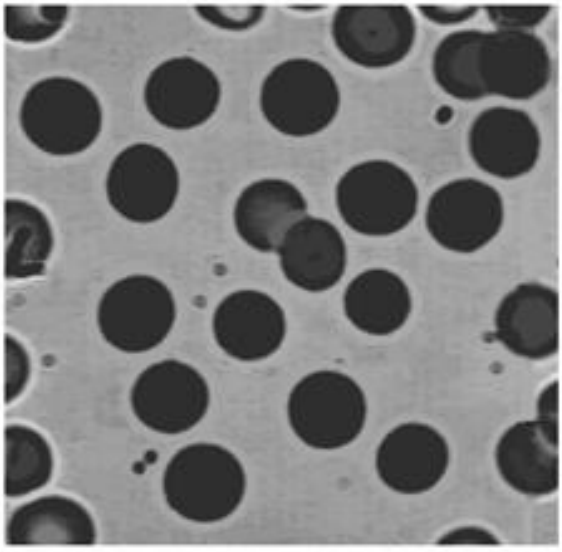
3.1.4. Image Holes Filling

A hole is background region that is subset of an object. During image processing of a frame, a frame maybe produced with some missing regions or pixels, so we use the hole fill method to close and fill this missing data. During segmentation of an image some holes are added to the image, using hole fill technique in order to eliminate these extra produced holes. The hole fill method do a flood-fill operation on a binary image. It changes the connected

background pixels whose value is 0 to a foreground pixels whose value is 1 till it reaches the object boundaries. Figure 3.5 shows as example of image hole filling, where the first image show the red blood cells before filling the holes and the second image we could determine the red blood cells more precisely after filling small holes.



a) Original image



b) Image after filling holes

Figure 3.5: Image holes filling

CHAPTER 4
HARDWARE PART

4.1.Arduino (UNO)

Arduino is an open-source based on a hardware and software. It has been used in thousands of educational and professional projects. It's a microcontroller based on the Atmel ATmega328 which can operate at 5V. The microcontroller can be programmed easily by connecting it to a computer via USB cable and using Arduino software (IDE) which is built based on Java (Ladyada, 2014). This board offers digital and analog inputs and outputs as well as PWM output. Figure 4.1 shows third generation of Arduino UNO.

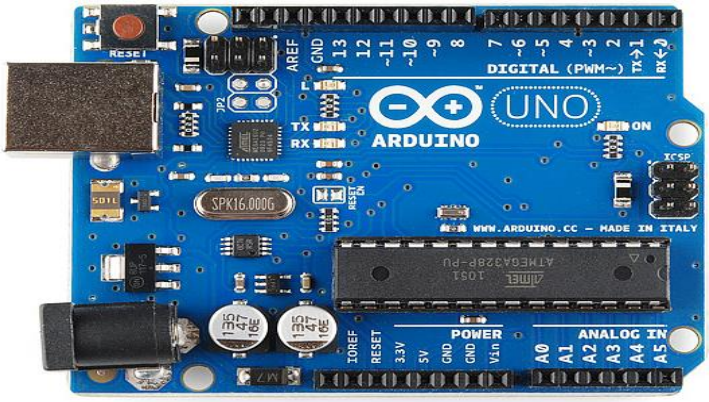


Figure 4.1: Arduino UNO

4.2.Servo Motor

4.2.1. PWM

Pulse-width modulation (PWM) is a technique used to control the amplitude of digital signal in order to control AC/DC motors or even used in communication field. It controls the power of the voltage components by cycling the on and off phases of a signal quickly and thus varying the value of duty cycle. As a motor this input will be a constant voltage since the

square waves are varying quickly. PWM contribute in minimizing the loss in power in the circuit. A square waveform of the PWM duty cycle can be shown in figure 4.2.

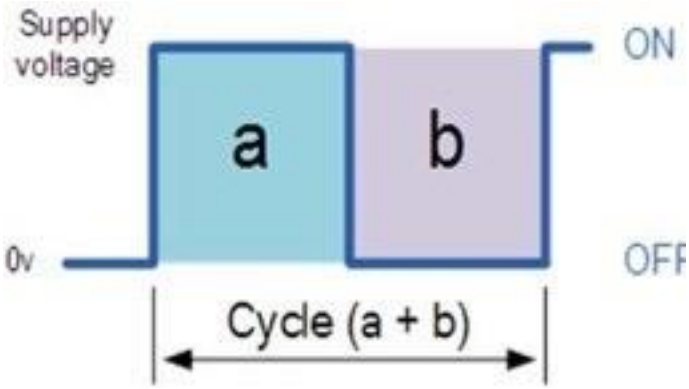


Figure 4.2: PWM (duty cycle)

For better comprehension of PWM these diagrammatic portrayals can be utilized. Figure 4.3 show the waveforms gotten as yield at various voltage requirements.

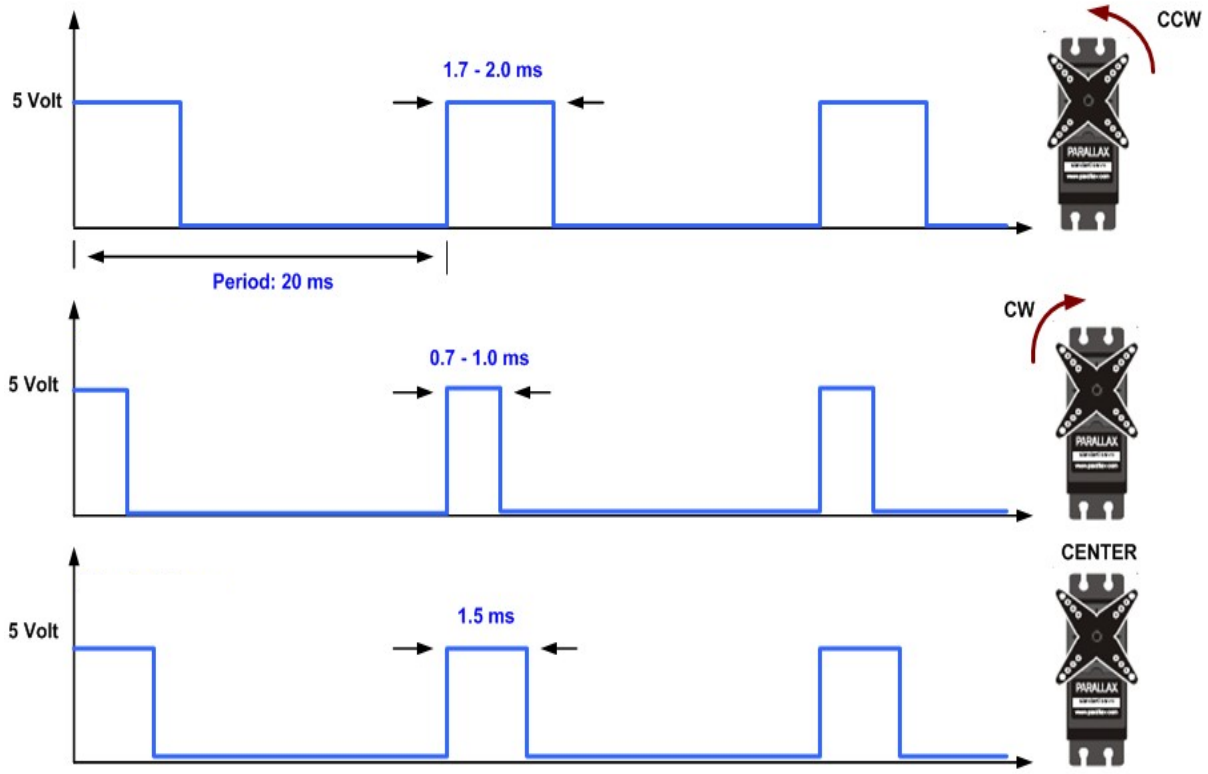


Figure 4.3: Servo motor PWM timing diagram

4.2.2. Servo Motor (MG 90S) Specifications

Its position is defined by the width of duty cycle of the PWM pulses arriving from Arduino. Shown in Figure 4.4.

This will redirect the mechanical structure towards the target.

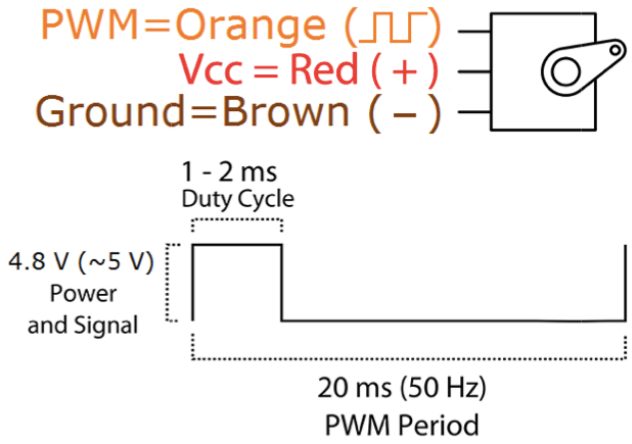


Figure 4.4: PWM Period in Servo Motor

Table 4.1: Servo motor specifications

Weight	13.4 g
Dimensions	22.5×12×35.5 mm
Stall torque	1.8-2.2 kgf.cm
Speed	0.1s/60 degree- 0.08s/60 degree
Voltage	4.8 V – 6.0 V
Dead band width	5µs

Figure 4.5: Servo motor (MG90S)

4.3.Camera

A4Tech PK900H webcam has been chosen because it has ability to recording at 30 fps for high resolutions and 60 fps for low resolution. It is a 1920×1080p full HD camera and the image resolution is up to 16 megapixel

Figure 4.6: A4TECH PK900H webcam

CHAPTER 5
SYSTEM DESIGN

5.1.Aim of the System

The ball-plate system is a platform carrying a camera has two degree of freedom. Two DC servo motors will control the inclination of the plate in order to balance the ball in its predefined coordinates or follow a specific patern. Figure 5.1 shows the full system platform.

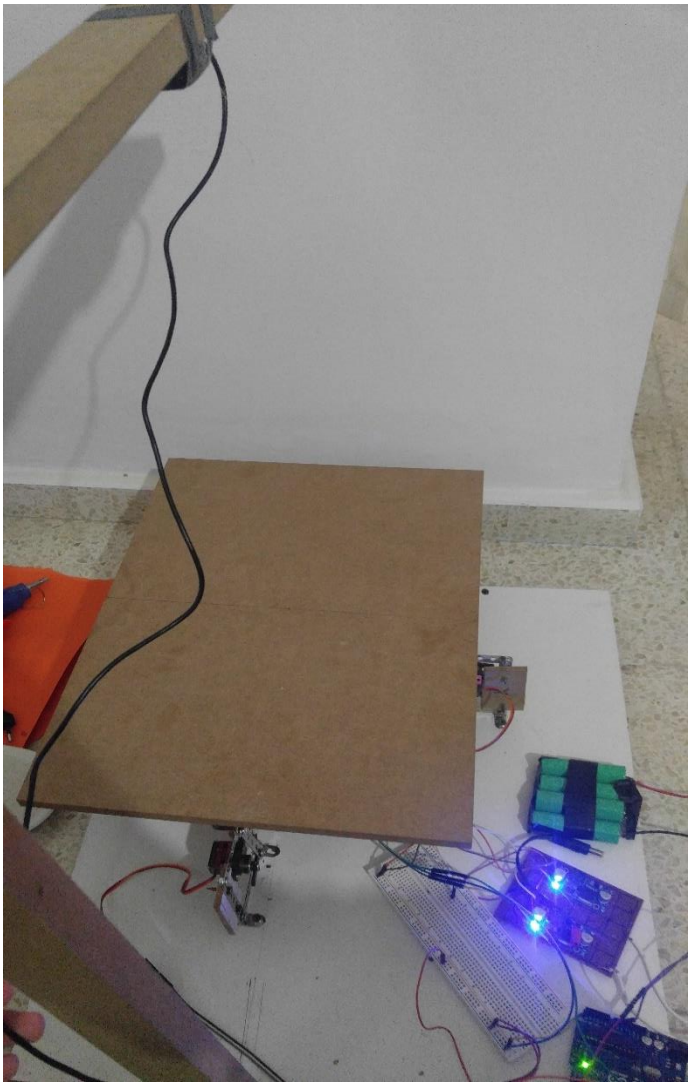


Figure 5.1: Ball-plate system

5.2.System Overview

As showen in Figure 5.2, the camera send a live streem video to the PC in order to process the video frames and get the current position of the ball and compare it to the reference coordinates. Then, it calculates the error values to be sent to the arduino that will control the angle od servo motors and thus the inclination of the plate in order to deliver the ball to its target position coordinates.

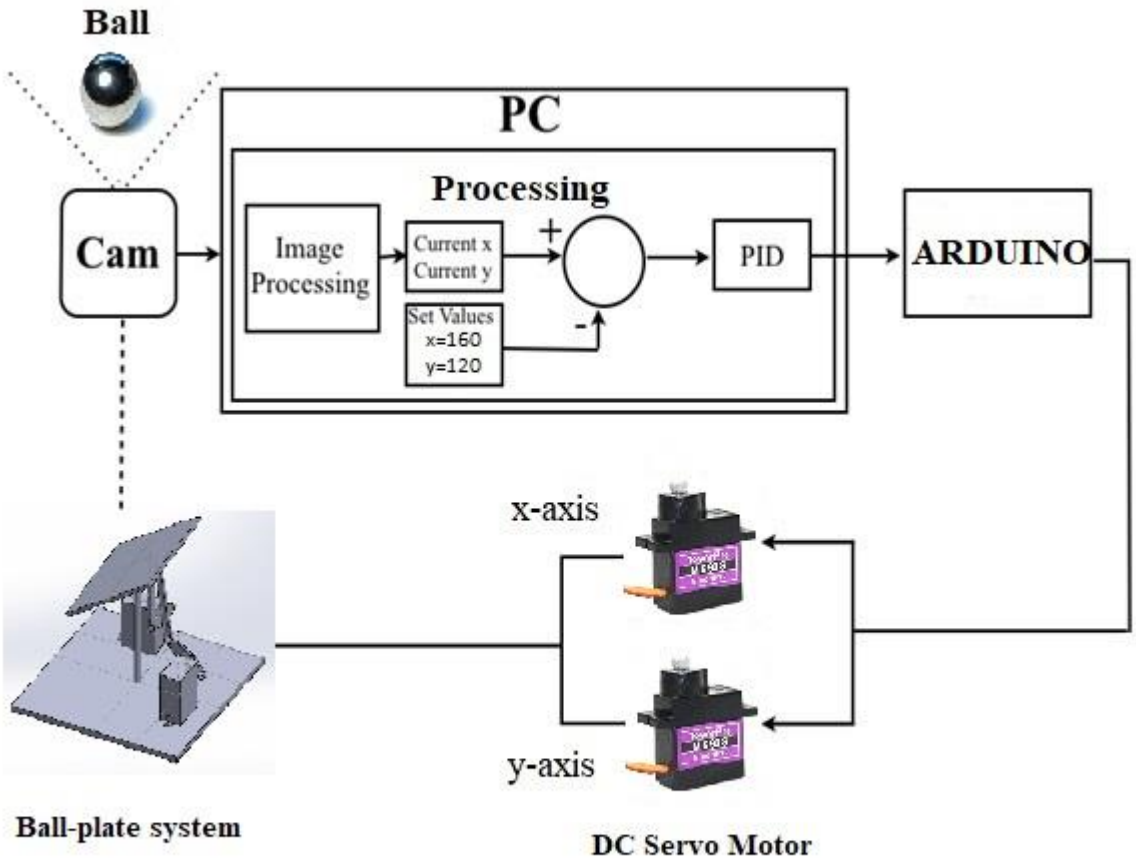


Figure 5.2: Block diagram

5.3.Image Processing

Processing will get frames from the video delivered by the camera, with 320×240 pixels at a rate of 30 frame per second.

Image procdessing in this system will capture the ball which is silver and subtract the backround which is in light brown.

Figure 5.3: Original image

Create a grayscale image and red component image from from the original image. Then subrtact the grayscale image from the red component image as shown in figure 5.4

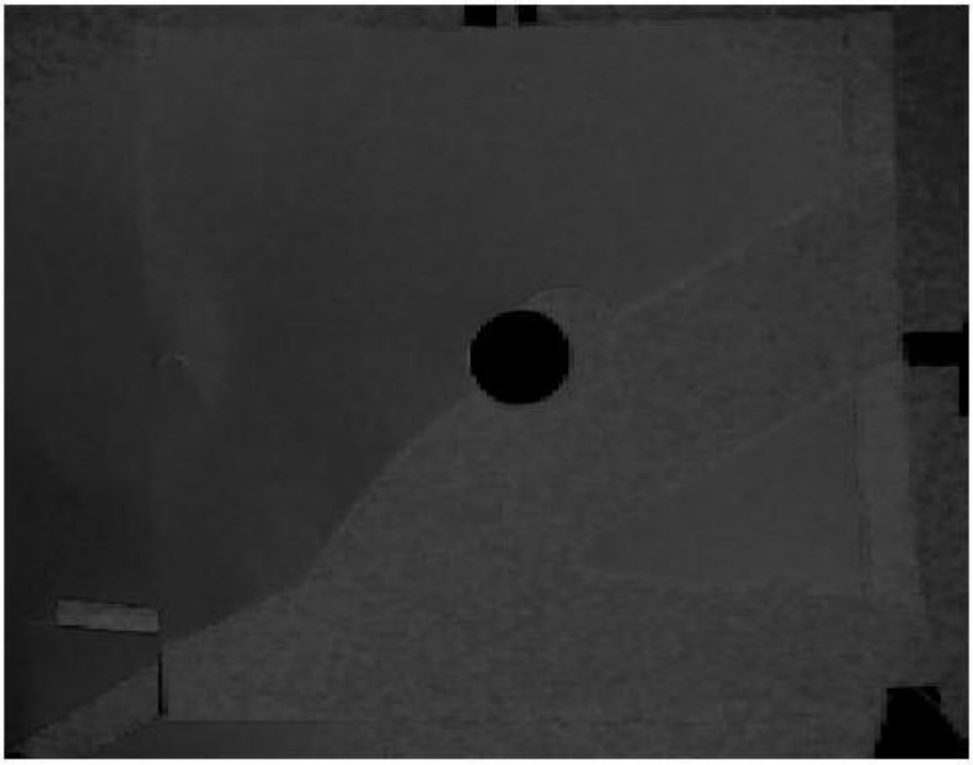


Figure 5.4: Subtraction result

Convert the subtraction image result into a binary image with threshold equal to 0.05 as shown in figure 5.5.

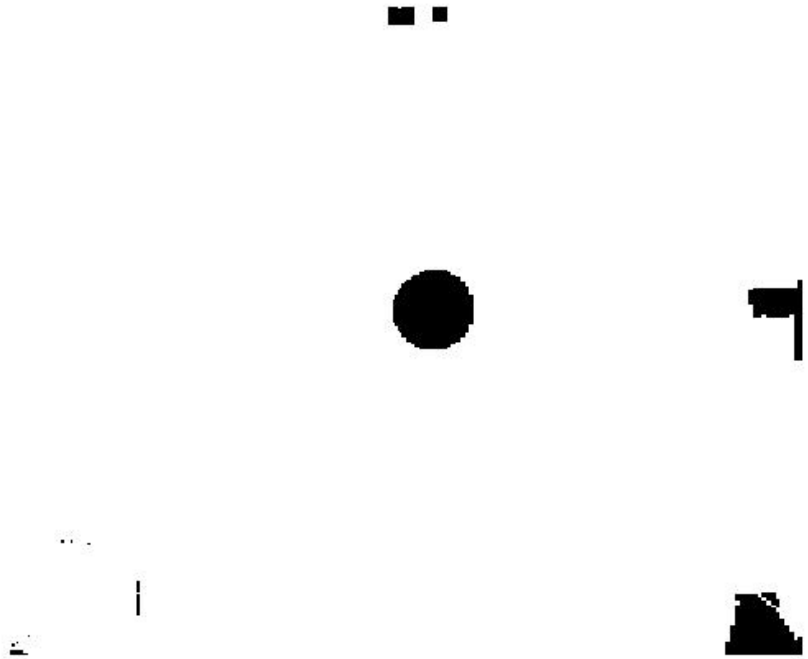


Figure 5.5: Binary image

Figure 5.6 showing the result after filling the small holes.

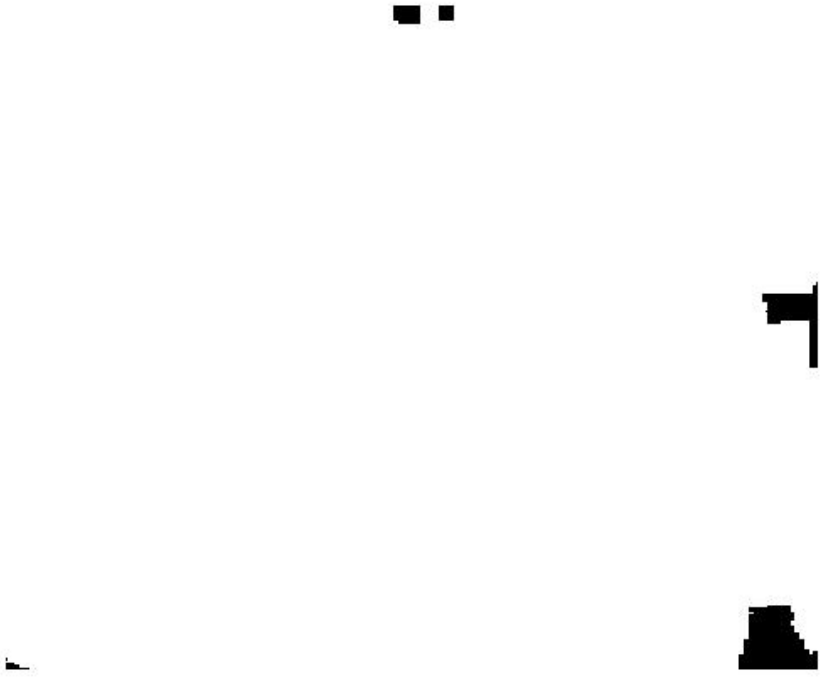


Figure 5.6: image after filling small holes

Figure 5.7 shows the result of subtraction of figure 5.6 from figure 5.5 in order to determine the position of the ball.

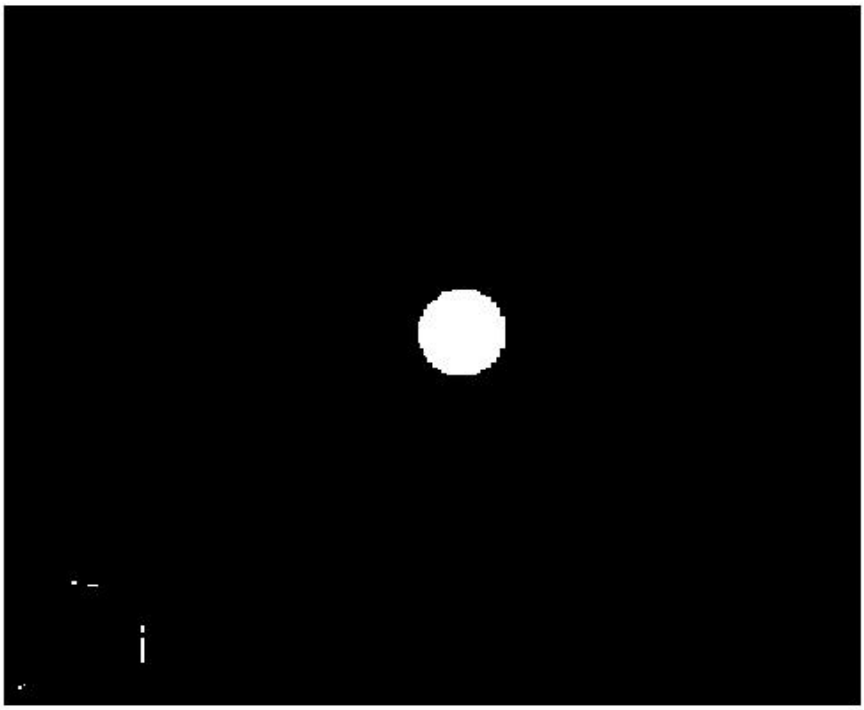


Figure 5.7: subtraction result

Removing small objects those less then 500px from resulting image as shown in figure 5.8.

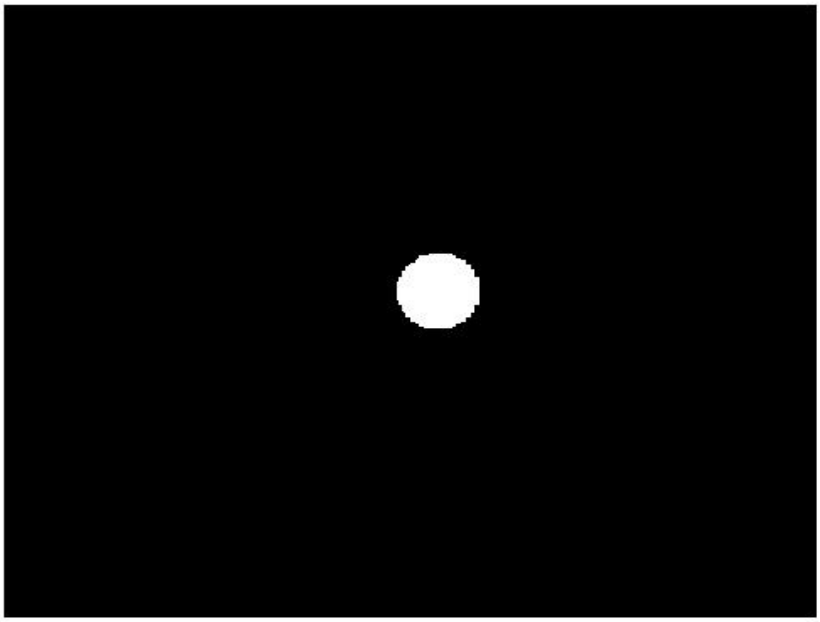


Figure 5.8: remove objects less than 500px.

Labeling the components in the image and set properties on the image.

Display the coordinates (X,Y) of the centroid on the original image as shown in figure 5.9.

Figure 5.9: Label Connected Components in 2-D Binary Image

5.4.Control System

5.4.1. Relation between servo angle and plate angle

The relation between servo angle and plate angle is done in order to make sure getting the best response from the system. This equation is used in the Processing code to control the position of the ball.

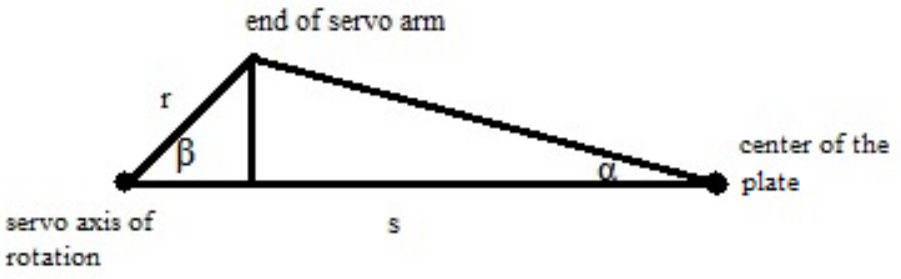


Figure 5.10: Relation between servo angle and plate angle

Using figure 5.10 which represent system drawing, the relation is calculated as follows

$$r.sin\beta = (s - r.cos\beta)tan\alpha \quad (5.1)$$

$$r.sin\beta + (r.tan\alpha)cos\beta = s.tan\alpha \quad (5.2)$$

Where

r: servo arm length

s: distance between plate center and servo center

α: angle of plate

β: angle of servo

For simple numerical calculations, assume:

a=r

b=r.tanα

c=s.tanα

x=β

Substitute the following assumptions in equation 5.2 as follows;

$$a.sinx + b.cosx = c \quad (5.3)$$

Solving equation 5.3 using some trigonometric formulas, the following equations is resulted:

$$\sin(x + \alpha) = \frac{c}{\sqrt{a^2+b^2}} \quad (5.4)$$

$$x + \alpha = sin^{-1}\left(\frac{c}{\sqrt{a^2+b^2}}\right) \quad (5.5)$$

$$x + tan^{-1}\left(\frac{b}{a}\right) = sin^{-1}\left(\frac{c}{\sqrt{a^2+b^2}}\right) \quad (5.6)$$

$$x = sin^{-1}\left(\frac{c}{\sqrt{a^2+b^2}}\right) - tan^{-1}\left(\frac{b}{a}\right) \quad (5.7)$$

As a result

$$\beta = sin^{-1}\left(\frac{s.tan\alpha}{\sqrt{r^2+(r.tan\alpha)^2}}\right) - tan^{-1}(\frac{r.tan\alpha}{r}) \quad (5.8)$$

$$\beta = sin^{-1}\left(\frac{s.tan\alpha}{\sqrt{r^2+(r.tan\alpha)^2}}\right) - \alpha \quad (5.9)$$

5.4.2. PID Controller

A PD controller using trail and error. Kp and Kd are increased to reach the desird proportional gain. First Kd was set to 0 and Kp increased to reach desired proportional gain (Dorf, 2008).

Then Kd is increased to overcome overshooting of the ball-plate system. The results of tuning are shown in table 5.1.

Table 5.1: Tuning of Kp and Kd of the PID system

Proportional gain (Kp)	Derivative gain (Kd)	Percentage of error
1	0	32%
0	1	38%
1	1	15%
1	1.5	7%

5.5.Mechanical structure

The ball-plate system is constructed with wood, fiber and iron. A 40 cm ×30 cm ×1.0cm wood fixed to a pivot in the middle of the plate. The system is built using wood to ensure getting weightless system and also to get frictionless movement of the ball. Each servo motor is connected through an extended arm that will tilt the plate in the X and Y direction. The camera is located on the top of the plate with support at a height of 60 cm and positioned to get the center of the frame perpendicular to the center of the plate. Figure 5.11 show the pivot of the system at the center of the plate and figure 5.12 show the installation of the DC servo motor.

Figure 5.11: Pivot of the system

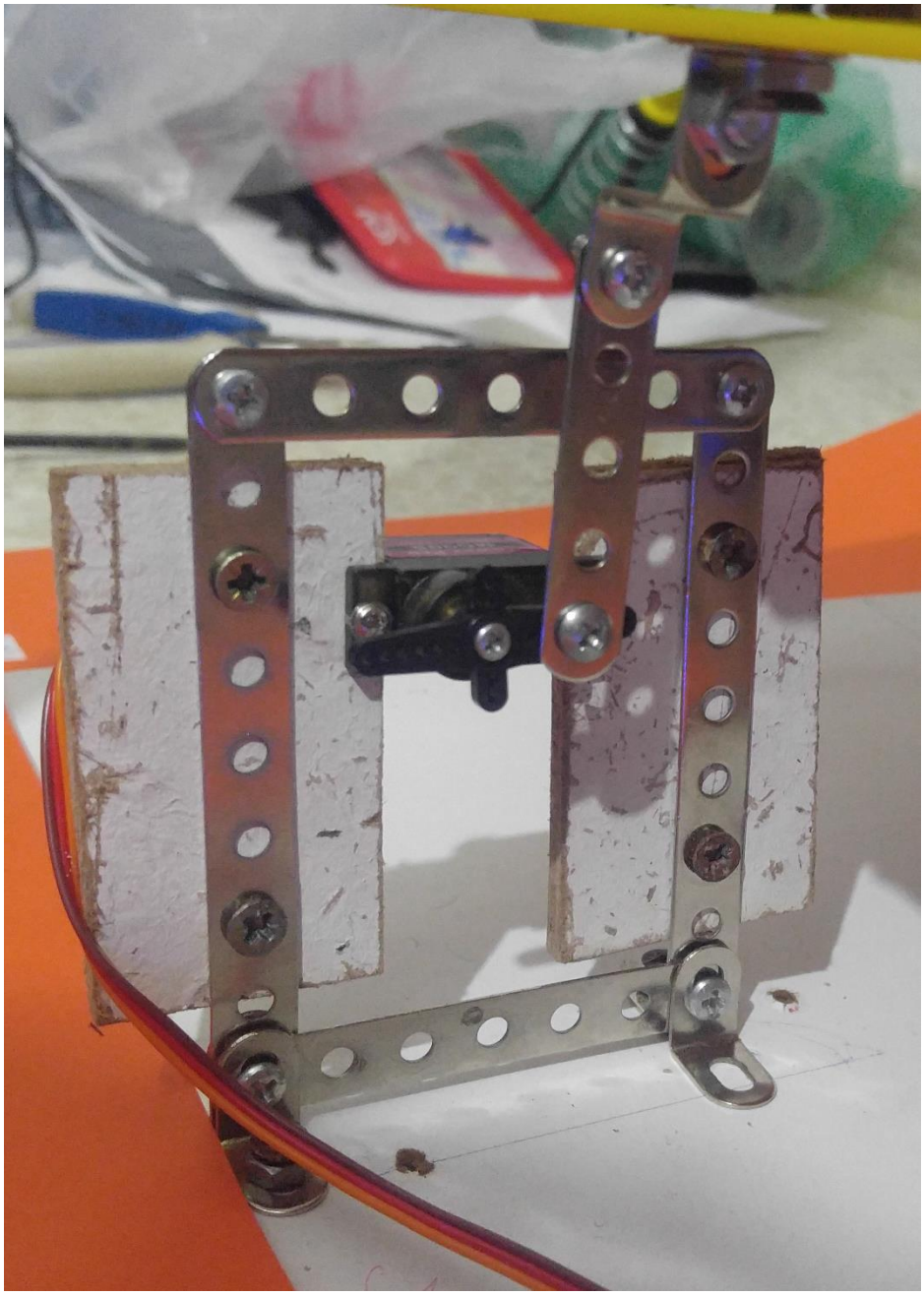


Figure 5.12: Installation of X and Y servo motors.

5.6.Experimental results

After platform construction and doing some experiments and simulations on MATLAB, we have got 8 frames per second only which made the system efficency very low (40%) since whenever the ball is slightly fast, MATLAB couldnot track it and thus resulted in falling of the ball. After this first experiment, a second experiment was done using Processing language which is based on Java and we have got 30 to 32 frames per second and that made

the system very efficient (90%) and it was able to balance the ball at its specidfied coordinates even when the ball is very fast.

The PID controller designed performed good during system testing and the ball kept balanced at any coordinate defined on the plate and not only the center. MATLAB was used to simulate the PID controller effect on the step response. Figure 5.13 shows the step response of the ball-plate system before adding a PD controller to the system and figure 5.14 shows the step response after adding a PD controller to the system.

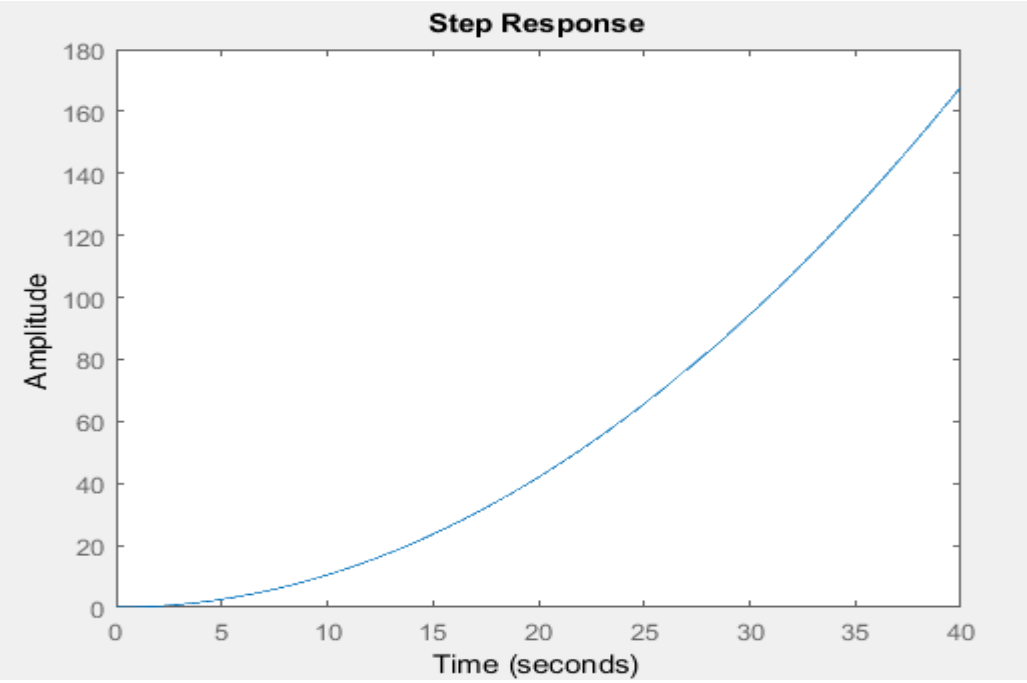


Figure 5.13: Ball-plate system step response without PD controller

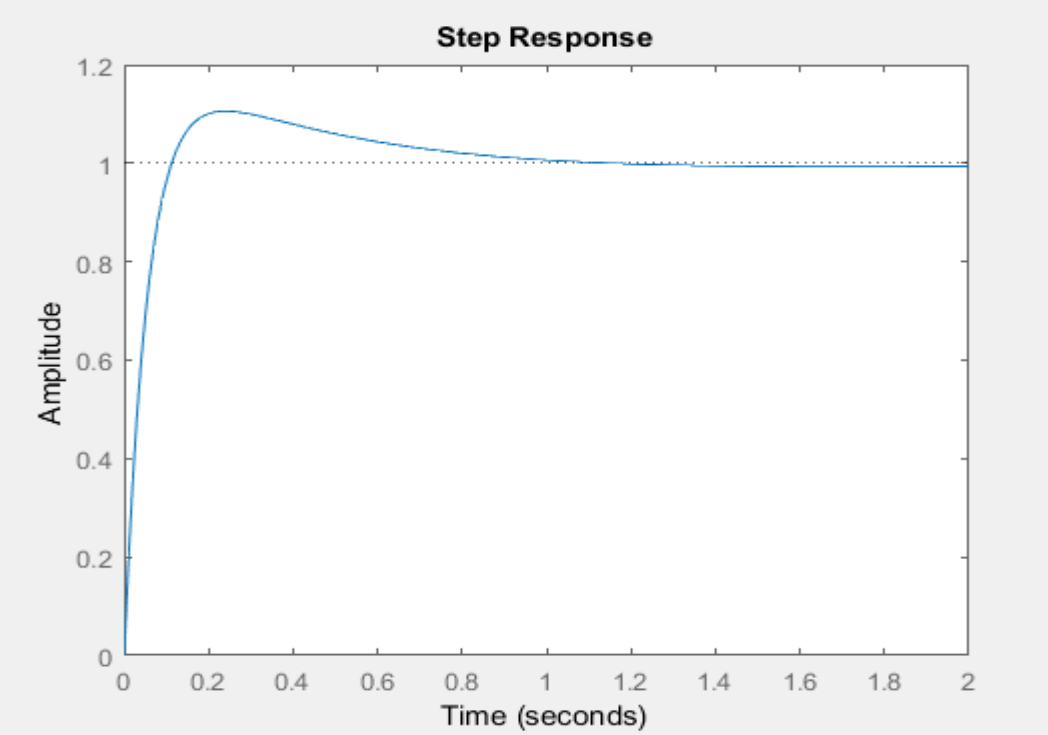


Figure 5.14: Ball-plate system step response with PD controller

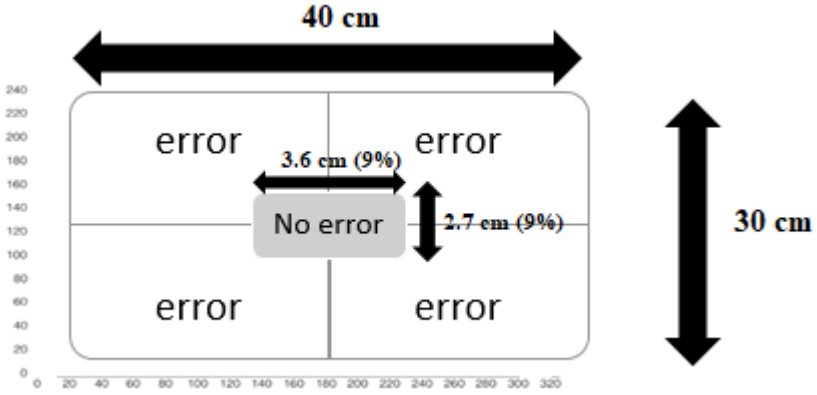


Figure 5.15: no error area threshold 9%

Table 5.2: no error threshold area 9%

Proportional gain	Derivative gain	% of error
0.5	0.1	over shot
0.5	0.3	25%
0.7	0.3	19%
0.9	1.1	7.1%
1.3	1.5	16.7%

The lowest percentage of error is 7.1% with Kp=0.9, Kd=1.1 and threshold percentage equal to 9%; however, after experiments, the region of no error found to be big, since the ball coudn't stop exactly at the center of the plate. Another experiment is done with threshold is equal to 5% and the results are shown in figure 5.16 and table 5.3.

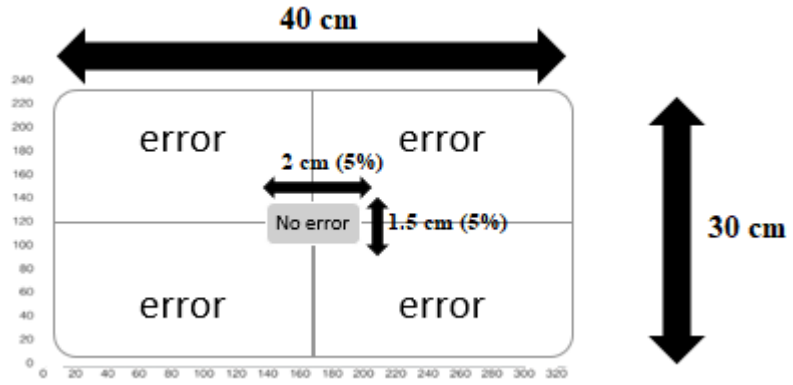


Figure 5.16: no error area threshold 5%

Table 5.3: no error area threshold 5%

Proportional gain	Derivative gain	% of error
0.5	0.1	over shot
0.5	0.3	25%
0.9	1	16%
1	1.5	8.2%
1.3	1.5	16.7%

Although the percentage of error increased a little bit up to 8.2% after doing the second experiment; however, the ball can now center at the centroid of the plate with Kp=1, Kd=1.5 and threshold equal to 5% and thus got a smaller size of no error area.

After various real time experiments, some data was exported from the processing to Excel to analyze the system response. Figure 5.17 shows expected traget value of X vs actual value of X and the response of the plate angel.

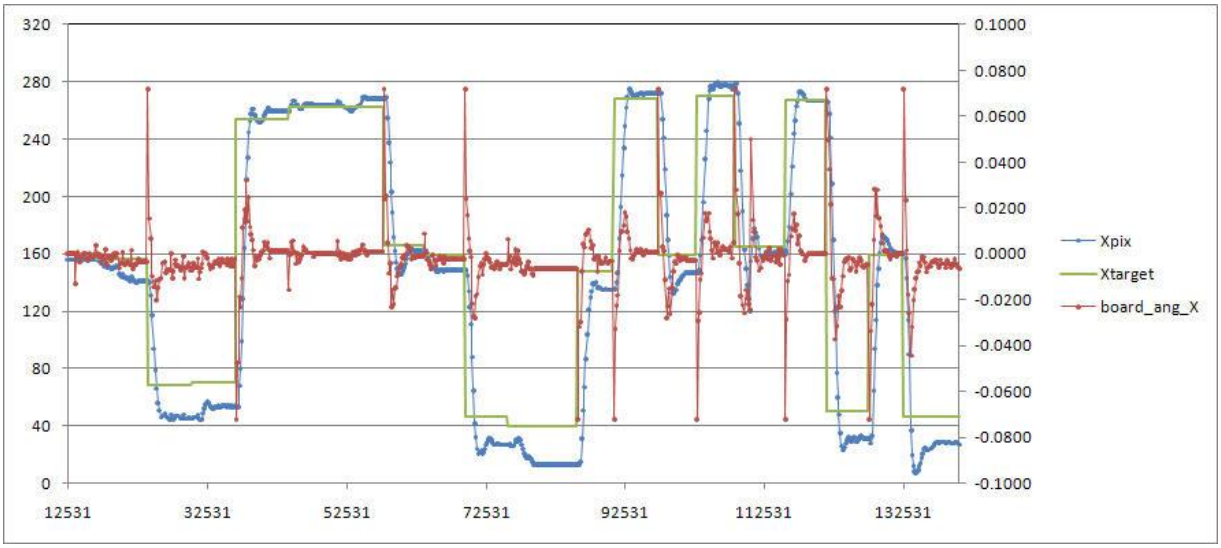


Figure 5.17: Graph showing traget value vs actual value of X and response of plate angle.

CHAPTER 6
CONCLUSION

6.1. Conclusion

In this thesis, the aim was to control the position of the ball on the plate for the center of the plate or tracking a given trajectory. Instead of using electrical sensors such as resistive touch panel which is more expensive than camera, so we have used vision sensing to track the position of the ball on the plate This system consists mainly of two parts, one is image processing in order to track the ball's trajectory and the other one is to build a perfect PID controller which controls the servo motors of the ball-plate system. We had problems in this system such as removing noise from frames captures, and reject disturbance in order to maintain a static ball position balancing.

The ball-plate system transfer function was derived after full modeling of this electromechanical system. The experiment was to drop a ball on the plate and waiting for the system response to balance the ball in the center of the plate. The plate was made of wood to ensure smooth slipping of the ball without friction. Balancing experiments were done, following the mouse, and positioning the ball at the center of the plate.

REFERENCES

Ali, E., & Aphiratsakun, N. (2015, December). AU ball on plate balancing robot. *IEEE International Conference on Robotics and Biomimetics* (pp. 2031-2034).

Brill, A., Frank, J. A., & Kapila, V. (2016, July). Using inertial and visual sensing from a mounted smartphone to stabilize a ball and beam test-bed. In American Control Conference (pp. 1335-1340).

Canny, J. F. (1983). Finding edges and lines in images, *Master's thesis, MIT. AI Lab*. TR-720.

Canny, J. F. (1986). A computational approach to edge detection, *IEEE Transaction on Pattern Analysis and Machine Intelligence*,8, 679-714.

Cheng, C., and Tsai, C. (2016). Visual Servo Control for Balancing a Ball-Plate System. *International Journal of Mechanical Engineering and Robotics Research*, 5, 28.

Dorf,R. (2008). Modern Control System.*Upper Saddle River: Pearson Education ,Inc*.

Dusek, F., Honc, D., and Sharma, R. (2017). Modeling of Ball and Plate System Based on First Principle Model and Optimal Control. *21st International Conference on Process Control* (pp. 216-221).

Frank, J., Gomez, J., and Vikram, K. (2015). Using tablets in the vision-based control of ball and beam test-bed. *Proceedings of 12th International Conference of informatics in control, Automation and Robotics.* France.

Ichihara, H. and Mochizuki, S.(2013) I-PD controller design bsaed on Genaralized KYP Lemma for ball and plate system, *European Control Conference* (pp. 2855-2860).

Kocaoglu, S., and Kuscu, H. (2013). Design and control of PID controlled Ball and Beam System. *Unitech. Int. Science Conference* (pp. 41-46).

Ladyada: (2014).Arduino Tips, Tricks, and Techniques. *New York: Adafruit Industries*.

Li-Ming, C., Ming-Tzu, H., and Yusie, R. (2013). Visual Servoing Tracking Control of a Ball and Plate System: Design, Implementation and Experimental Validation. *International Journal of Advanced Robotic Systems.*

Liu, A., Peng, C., and Chang, S. (1997). Wavelet analysis of satellite images for coastal watch*IEEE Journal of Oceanic Engineering* (Vol.22, pp. 9–17).

Mohsin, M., T. (2015). Development of a Ball and Plate System.*122nd ASEE Anual Conference & Exposition.*Washignton.

Ratan, A.L., and Lozano-Perez, T. (1999). A framework for learning query concepts in image classification *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp 423-431).

Roberts, L. (1965). Machine Perception of 3-D Solids*Optical and Electro-optical Information Processing*, MIT Press.

Urquhart, R. (1982). Graph theoretical clustering based on limited neighborhood sets. *Pattern Recognition* (pp 173-187).

Wertheimer, M., & Ellis, W.B. (1938). Laws of organization in perceptual forms (partial translation) *A Sourcebook of Gestalt Psychology* (pp 71-88).

Weiss, Y. (1999*).* Segmentation using Eigenvectors: A Unifying View. *Proceedings of the International Conference on Computer Vision* (pp 975-982).

Xiao, J., & Buttazzo, G. (2016). Adaptive embedded control for a ball and plate system. *The Eighth International Conference on Adaptive and Self-Adaptive Systems and Applications.*

Yu, W. (2009). Nonlinear PD regulation for ball and beam system. *International Journal of Electrical Engineering Education,* 46, 59-73.

Zhou,H.,Wu,J., and Zhang, J. (2010). Digital Image Processing.*Denmark: Ventus Publishing ApS*.

APPENDICES

APPENDIX 1

PROCESSING CODE

```
/*
Ball-Plate Balance System Using Image Processing
Near East University
Submitted to Assist.Prof.Dr. Boran Şekeroğlu
by Ahmed Itani
NICOSIA, 2017
*/

import processing.serial.*;
import cc.arduino.*;
import JMyron.*;

Arduino arduino;
JMyron I;

float servoX_ang; // in rad
float servoY_ang; // in rad
int servoX_ref=90; //initial position
int servoY_ref=90;
int servoX_value=servoX_ref;
int servoY_value=servoY_ref;

int sizeX_pix=320; //size of video
int sizeY_pix=240;
```

```
//system dimentions
float sizeX_m=0.37; //visible area (meters)
float sizeY_m=0.27;
float plate_center_servo_center_X=0.205; //distance form plate center to servo center
float plate_center_servo_center_Y=0.155;
float arm_length_X=0.01; //length of servo arm
float arm_length_Y=0.01;

//plate angle
float plate_angX; // in rad
float plate_angY;// in rad
float plate_max_angX=atan(arm_length_X/plate_center_servo_center_X);
float plate_max_angY=atan(arm_length_Y/plate_center_servo_center_Y);

//PID/////////////////////////////////////////////////////////////////////////////
//PID variables
int error_X, error_Y;
int previous_error_X, previous_error_Y;
float integral_X, integral_Y;
float derivative_X, derivative_Y;
float output_X, output_Y;
//PID values calibrated using trial and error
float Kp_X = 1;
float Ki_X = 0;
float Kd_X = 1.5;
float Ko_X = 4250;

float Kp_Y = 1;
float Ki_Y = 0;
float Kd_Y = 1.5;
float Ko_Y = 4750;
```

```
// using current and previous glob values for calculation errors
//current frame glob
float t;
int Xpix;
int Ypix;
float X0;
float Y0;
//1st frame previous glob
float t1;
int X1pix;
int Y1pix;
float X1;
float Y1;
//2nd frame previous glob
float t2;
int X2pix;
int Y2pix;
float X2;
float Y2;

//target position
int X_target=160;
int Y_target=120;

boolean auto=false;
boolean help=false;
boolean mouse_follow=false;

PrintWriter output;
////////////////////////////////////////////////////////////////////////////////
```

```
void setup() {
  size(sizeX_pix,sizeY_pix);

  I=new JMyron();
  I.start(sizeX_pix,sizeY_pix);
  I.findGlobs(1);

  PFont font;
  font=loadFont("Monospaced.bolditalic-10.vlw");
  textFont(font);

  arduino=new Arduino(this, Arduino.list()[0], 57600);//connection between processing and
arduino
}

////////////////////////////////////////////////////////////////////////////////
void draw(){
  background(0);

  I.trackColor(0,0,0,200);

  I.update();
  int[] img=I.image();

  loadPixels(); //show video on the screen
  I.imageCopy(pixels);
  updatePixels();

  noFill();
  int[][]g;
```

```
//track glob box
g=I.globBoxes();
stroke(255,0,0);

if(g.length>0){
  int[] b =g[0]; //first glob

  noFill();
  rect(b[0], b[1], b[2], b[3]); //draw bounding box around the ball
}

//calculations of coordinates in pixels and meters
t2=t1;
t1=t;
t=millis();// time of current glob

X2pix=X1pix;
Y2pix=Y2pix;
X2=X1;
Y2=Y1;
X1pix=Xpix;
Y1pix=Ypix;
X1=X0;
Y1=Y0;

g=I.globCenters(); //track centroid

if(g.length>0){
  int[]c=g[0];
  Xpix=c[0];
  Ypix=c[1];
```

```
 }
 else{//predict the coordinate of the ball
 Xpix = int(X1pix + float(X1pix - X2pix) * (t - t1) / (t1 - t2));
 Ypix = int(Y1pix + float(Y1pix - Y2pix) * (t - t1) / (t1 - t2));
 }

float fXpix=float(Xpix);
float fYpix=float(Ypix);

X0=((fXpix/sizeX_pix)*sizeX_m)-(sizeX_m/2); //from center of plate
Y0=((fYpix/sizeY_pix)*sizeY_m)-(sizeY_m/2);

//mouse follower
if (mouse_follow==true){
  X_target=mouseX;
  Y_target=mouseY;
}
if(auto==true){
//ServoX
 error_X = X_target - Xpix;
 integral_X = integral_X + (error_X * (t - t1)/1000);
 derivative_X = (error_X - previous_error_X) / ((t - t1)/1000);
 output_X = (Kp_X*error_X) + (Ki_X*integral_X) + (Kd_X*derivative_X);
 previous_error_X = error_X;

 plate_angX = constrain((-output_X / Ko_X), -plate_max_angX, plate_max_angX);
 servoX_ang = asin((plate_center_servo_center_X * tan(plate_angX)) /
sqrt(pow(arm_length_X, 2) + pow(arm_length_X * tan(plate_angX), 2))) - plate_angX;
 servoX_value = int(map(servoX_ang, -PI/2, PI/2, (servoX_ref-90),(90+servoX_ref)));

 //ServoY
```

```
 error_Y = Y_target - Ypix;
 integral_Y = integral_Y + (error_Y * (t - t1)/1000);
 derivative_Y = (error_Y - previous_error_Y) / ((t - t1)/1000);

 output_Y = (Kp_Y*error_Y) + (Ki_Y*integral_Y) + (Kd_Y*derivative_Y);
 previous_error_Y = error_Y;

 plate_angY = constrain((-output_Y / Ko_Y), -plate_max_angY, plate_max_angY);
 servoY_ang = - asin((plate_center_servo_center_Y * tan(plate_angY)) /
sqrt(pow(arm_length_Y, 2) + pow(arm_length_Y * tan(plate_angY), 2))) - plate_angY;
 servoY_value = int(map(servoY_ang, -PI/2, PI/2,  (servoY_ref-90),(90+servoY_ref)));

 arduino.analogWrite(9,servoX_value);
 arduino.analogWrite(10, servoY_value);
}

 //display data on video////////////////////////////////////////////////////////
 fill(255,0,0);

String s= frameRate+
        "\nt="+int(t)+
        "\n="+Xpix+", "+Ypix+
        "\nservoX_value = "+servoX_value+
        "\nServoY_value = "+servoY_value;

 text(s,5,15);

 //show button menu
 if (help == true) {
   String helpstring = "SPACE       = set servos flat"+
                  "\nC         = camera settings"+
```

```
                    "\nclick     = reset target position"+
                    "\nENTER     = toggle autobalancing"+
                    "\nM          = toggle mouse following"+
                    "\nY          = end program"+
                    "\nH          = toggle help";
    text(helpstring, 75, 15);
  //}
  line(X_target, Y_target-5, X_target, Y_target+5);
  line(X_target-5, Y_target, X_target+5, Y_target);
  text("("+X_target+","+Y_target+")", X_target+5, Y_target+10);

}
}
void keyPressed(){
  if(key == ' ') {
  //set servos to level
  servoX_value = servoX_ref;
  servoY_value = servoY_ref;
  arduino.analogWrite(9, servoX_value);
  arduino.analogWrite(10, servoY_value);
  }else if (key == 'C' || key == 'c'){
  //camera settings
  I.settings();
  }else if (key == 'H' || key == 'h'){
  //toggle Help display
  help = !help;
  }else if (key == ENTER){
  //toggle the autobalancing
  auto = !auto;

  previous_error_X = 0;
```

```
  integral_X = 0;
  previous_error_Y = 0;
  integral_Y = 0;
  } else if (key == 'M' || key == 'm'){
    //toggle the shape following
    mouse_follow = !mouse_follow;
    previous_error_X = 0;
    integral_X = 0;
    previous_error_Y = 0;
    integral_Y = 0;
  } else if (key == 'Y' || key == 'y'){
  output.flush(); // Write the remaining data
  output.close(); // Finish the file

  arduino.analogWrite(9, servoX_ref);
  arduino.analogWrite(10, servoY_ref);

  exit();
  }
}

void mousePressed() {
  //reset target position
  X_target = mouseX;
  Y_target = mouseY;
}
public void stop(){
  I.stop();
  super.stop();
}
```

APPENDIX 2
ARDUINO

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.**This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.
- **IOREF.** This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USBto-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with theanalogWrite() function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.

MISO (Master In Slave Out) - The Slave line for sending data to the master,

MOSI (Master Out Slave In) - The Master line for sending data to the peripherals,

SCK (Serial Clock) - The clock pulses which synchronize data transmission generated by the master

and one line specific for every device:

SS (Slave Select) - the pin on each device that the master can use to enable and disable specific devices.

- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analogReference().
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

AREF:

This is the analog voltage reference. It can be used in stead of the standard 5V reference for the top end of the analog spectrum.

IOREF:

This is a voltage corresponding to the i/o of that board, for example an Uno would supply 5v to this pin, but a Due would supply 3.3v

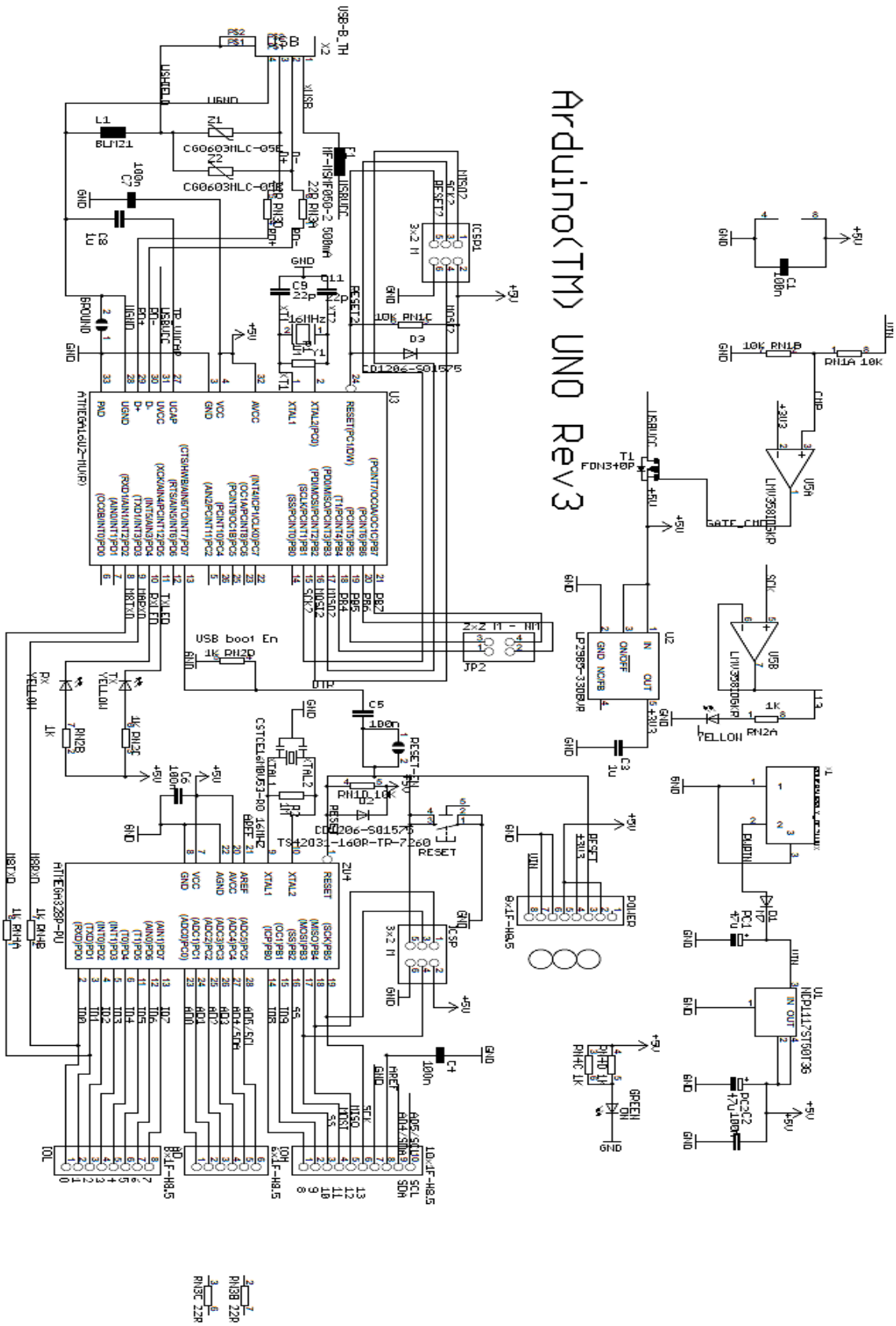
Programming

The Syrduino Uno can be programmed with the Arduino software .

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol .

USB Over current Protection

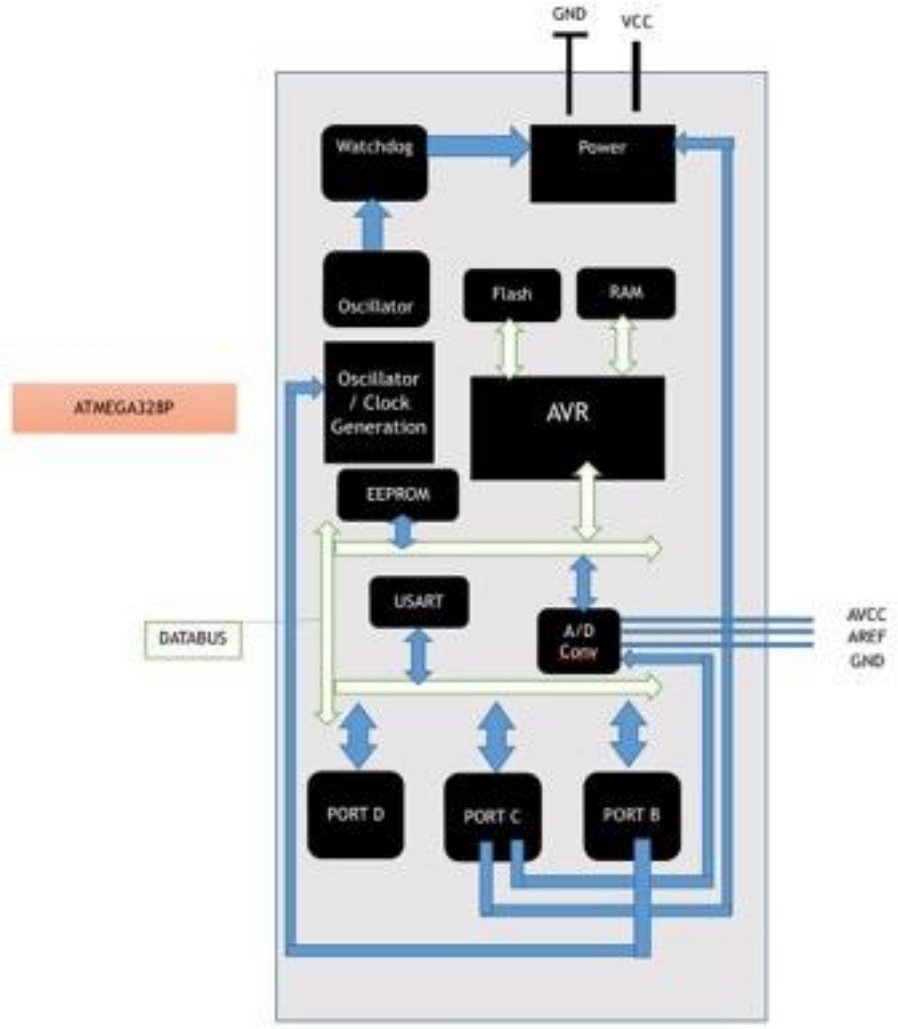
The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.


Arduino(TM) UNO Rev3

APPENDIX 3

ATMEGA328P

Microcontroller ATMEGA328P block diagram



Features

• High Performance, Low Power AVR® 8-Bit Microcontroller

– Up to 20 MIPS Throughput at 20 MHz

High Endurance Non-volatile Memory Segments

–32K Bytes of In-System Self-Programmable Flash program memory (ATmega328P).

–1K Bytes EEPROM (ATmega328P).

–2K Bytes Internal SRAM (ATmega328P).

– Write/Erase Cycles: 10,000 Flash/100,000 EEPROM. Data retention: 20 years at 85°C/100 years at 25°C .

• Peripheral Features

– Two 8-bit Timer/Counters with Separate Pre scaler and Compare Mode. – One 16-bit Timer/Counter with Separate Pre scaler, Compare Mode, and Capture Mode .

– Real Time Counter with Separate Oscillator.

– Six PWM Channels.

– 8-channel 10-bit ADC.

– Programmable Serial USART.

– Programmable Watchdog Timer with Separate On-chip Oscillator. – On-chip Analog Comparator.

• Operating Voltage:

– 1.8 - 5.5V for ATmega328P.

• Temperature Range: – - 40°C to 85°C.

• Speed Grade:

– 0 - 20 MHz @ 1.8 - 5.5V.

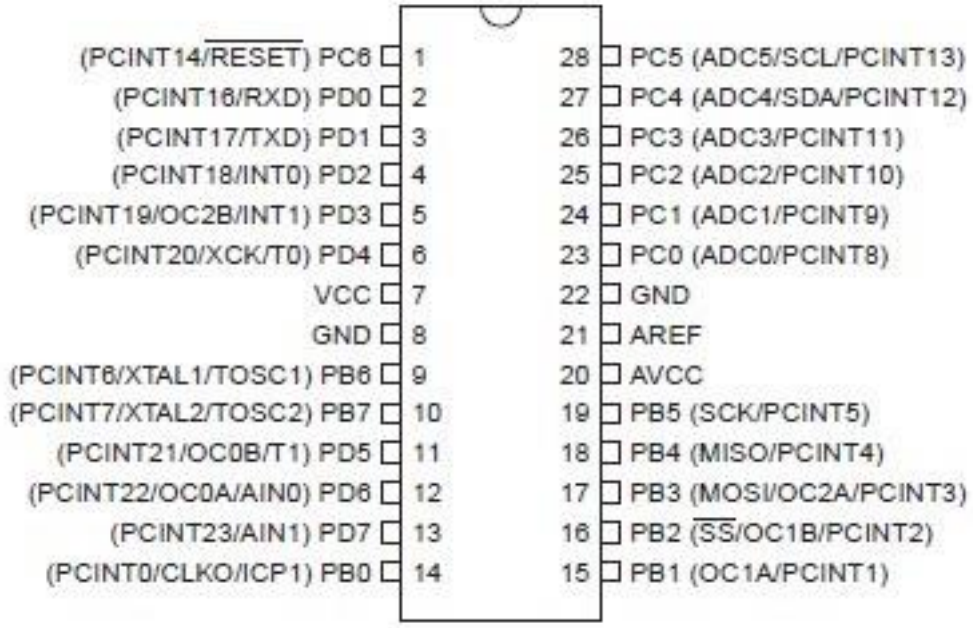
• Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega328P:

– Active Mode: 0.2 mA.

– Power-down Mode: 0.1 μA.

– Power-save Mode: 0.75 μA.

Pin Descriptions :



VCC:

Digital supply voltage.

GND: Ground.

Port B (PB7..PB0):

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega328P.

Port C (PC7..PC0) Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit. The Port C pins are tri-stated when a reset
condition becomes active, even if the clock is not running. Port C also serves the functions of various specific features of the ATmega32

Port D (PD7..PD0) is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.
Port D also serves the functions of various special features of the ATmega32

RESET:

Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length. Shorter pulses are not guaranteed to generate a reset.

XTAL1:

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2:

Output from the inverting Oscillator amplifier.

AVCC:

AVCC is the supply voltage pin for the A/D Converter. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.
AREF: AREF is the analog reference pin for the A/D Converter.