

**INTEGRATING COMPUTER VISION WITH A ROBOT ARM SYSTEM**

**A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED  
SCIENCE OF  
ÇANKAYA UNIVERSITY**

**BY  
ZEAD MOHAMMED YOSIF**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE  
DEGREE OF  
MASTER OF SCIENCE  
IN  
THE DEPARTMENT OF  
COMPUTER ENGINEERING**

**JULY 2014**

Title of the Thesis : **Integrating Computer Vision with a Robot Arm System**


Submitted by **Zead Mohammed YOSIF**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.



Prof. Dr. Taner ALTUNOK  
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.



Assist. Prof. Dr. Murat SARAN  
Head of Department


This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.





Assist. Prof. Dr. Reza HASSANPOUR  
Supervisor

**Examination Date: 17.07.2014**

**Examining Committee Members**

Assist. Prof. Dr. Reza HASSANPOUR (Çankaya Univ.) 

Prof. Dr. Mehmet TOLUN (Aksaray Univ.) 

Assist. Prof. Dr. Abdül Kadir GÖRÜR (Çankaya Univ.) 

## STATEMENT OF NON-PLAGIARISM PAGE

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 conducts, I have fully cited and referenced all martial and results that are not original work.

Name, Last Name : Zead Mohammed YOSIF

Signature :



Date

: 17.07.2014

## **ABSTRACT**

### **INTEGRATING COMPUTER VISION WITH A ROBOT ARM SYSTEM**

YOSIF, Zead Mohammed

M.Sc., Department of Computer Engineering

Supervisor: Assist. Prof. Dr. Reza ZARE HASSANPOUR

Co-Supervisor: Instructor Dr. Sa'ad Ahmed AL-KAZZAZ

July 2014, 68 pages

During last decades, robotic system has been employed in different fields, such as, industrial, civil, military, medical, and many other applications. Vision system is integrated with robot systems to enhance the controlling performance of the robot system. A great deal of features can be computed using the information have been gotten from vision sensors (camera). The extracted information from vision system can be used in the feedback to have the ability to control the robot arm or motion, but the operation of extracting this information from vision system is time consuming.

This thesis addressed the problem of following (tracking) and grasping of moving target (object) with limited velocity in real time by employing the technology of Eye-in-Hand, whereas a camera attached (mounted) to the robot arm end effector. This done by using a predictor (Kalman filter) that estimates the positions of the target in the future, an algorithm was designed to track an object move in different trajectories, within the camera field of view. The Kalman filter uses the measured position of the target as well as previous state estimates to fix the location of the

target object at the next time step, in other word, the Kalman filter is applied to keep observing the object till grasp it.

The employing of vision system information in the feedback control of the robot systems have been the major research in robotics and Mechatronic systems. The utilizing from this information has been proposed to handle stability and reliability issues in vision-based control system.

**Keywords:** Tracking, Kalman Filter, Robot Arm, Visual Servoing, Object Tracking, Object Grasping, Eye-in-Hand.

## ÖZ

### **BİLGİSAYAR GÖRÜŞÜNÜN BİR ROBOT KOLU SİSTEMİ İLE ENTEGRASYONU**

YOSIF, Zead Mohammed

Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı

Tez Yöneticisi: Yrd. Doç. Dr. Reza ZARE HASSANPOUR.

Yardımcı Tez Yöneticisi: Öğr. Gör. Dr. Sa'adAhmed AL-KAZZAZ

Temmuz 2014, 68 sayfa

Son yıllarda robot sistemi çeşitli alanlarda gözlemlendi. Bu alanlar endüstriyel, devlet, askeriye, sağlık ve benzeri alanlardı. Görüş dataları robot sistemini kontrol etme performansını artırmak üzere robota entegre edildi. Görüş sensörleri ( kamera ) kullanılarak ilgili özellikler ve bilgiler hesaplanabilir. Görüş sisteminden alınan bilgi robot kolu efektörünü kontrol etmek üzere geri bildirimde kullanılabilir ancak, bu bilgiyi görüş sisteminden almak fazla zaman alır.

Bu tez, robot kolu efektörüne yerleştirilmiş kamera ile sınırlı hızda hareketli objenin gerçek zamanda Eye-in-Hand teknolojisiyle takibi ve yakalanmasını konu almaktadır. Bu işlem hedef objenin gelecekteki pozisyonları üzerine çıkarımlarda bulunan bir mekanizma ile gerçekleştirilir, obje hareketlerinin çeşitli yörüngelerde takibini sağlayan bir algoritma geliştirilmiştir. Hedef objenin bir sonraki hareketini yakalamak üzere Kalman filtresi objenin yörüngesindeki geçmiş pozisyonlarını ölçer, bir başka deyişle Kalman filtresi objeyi yakalayana kadar takip etmek üzere kullanılır. Robotun hareket kontrol mekanizmasındaki görüş sisteminden gelen geri

bildirim kullanılarak robot ve mekatronik sistemlerde major arařtırmalar yapılmıřtır. Bu arařtırma bilgilerinden yararlanılarak gr bazlı kontrol sistemindeki saęlamlık ve gvenilirlik meseleleri incelenmiřtir.

**Anahtar Kelimeler:** Takip, Kalman Filtresi, Robot Kolu, Grlebilir Geri Beslemeli Denetim, Obje Takibi, Obje Yakalanması, Gz- İinde- El.

## ACKNOWLEDGMENTS

Foremost, I have to express my sincere to my supervisor Assist. Prof. Dr. Reza HASSANPOUR and my co-advisor Dr. Sa'ad Ahmed AL-KAZZAZ for the continuous support of my master study and research.

My sincere thanks also go to Dr. Sa'ad AL-KHAYAT the responsible for robotic laboratory in Mosul University, whose guide and helped me spatially in the robot kinematics topic.

I would like to take this time to thank the staff of Mechatronics Engineering Department for helping me to perform the experimental part of this research. In addition, I would also like to thank my colleagues and my friends for all what have they done for me.

Last, but not least, my gratitude and sincere thanks go to my parents and my family for their unconditional support, kindness, and for having allowed with their efforts, to get to where I am now.

## TABLE OF CONTENTS

STATEMENT OF NON-PLAGIARISM .....	iii
ABSTRACT .....	iv
ÖZ .....	vi
ACKNOWLEDGMENTS.....	viii
TABLE OF CONTENTS .....	ix
LIST OF FIGURES.....	xi
LIST OF TABLE.....	xiii
LIST OF ABBREVIATIONS.....	xiv

### CHAPTERS:

<b>1. INTRODUCTION.....</b>	<b>1</b>
<b>1.1. General Review .....</b>	<b>1</b>
<b>1.2 Thesis Organization .....</b>	<b>3</b>
<b>2. BACKGROUND THEORY .....</b>	<b>4</b>
<b>2.1. Visual Servoing.....</b>	<b>4</b>
<b>2.2. Kalman Filter .....</b>	<b>5</b>
<b>2.2.1. Kalman Filter Stages.....</b>	<b>6</b>
<b>2.2.2. Kalman Filter Components .....</b>	<b>6</b>
<b>2.2.3. Kalman Filter Algorithm .....</b>	<b>8</b>
<b>2.3. Edge Detection.....</b>	<b>10</b>
<b>2.3.1. Morphological Operations .....</b>	<b>11</b>
<b>2.4. Median Filter .....</b>	<b>15</b>
<b>2.5. Robot Kinematics.....</b>	<b>16</b>
<b>2.5.1. Robot Arm Descriptor.....</b>	<b>17</b>
<b>2.5.2. Robot Arm Kinematics .....</b>	<b>17</b>

2.5.3.	Forward Kinematics . . . . .	18
2.5.4.	Inverse Kinematics . . . . .	20
2.5.5.	Base Joint Angle . . . . .	22
2.5.6.	Algebraic Arms' Solution . . . . .	22
2.5.7.	Roll Angle of Wrist. . . . .	24
3.	LITERATURE REVIEW . . . . .	26
3.1.	Eye-to-Hand. . . . .	26
3.2.	Eye-in-Hand. . . . .	29
3.3.	Eye-in-Hand and Eye-to-Hand. . . . .	34
4.	PROPOSED METHOD . . . . .	36
4.1.	Introduction . . . . .	36
4.2.	Proposed Method. . . . .	36
5.	EXPERIMENTAL RESULT . . . . .	43
5.1.	Experiment Framework . . . . .	43
5.2.	Description of Robot Arm . . . . .	44
5.2.1.	Servo Controller Card. . . . .	46
5.3.	Camera and PC. . . . .	48
5.4.	Experiments . . . . .	49
5.4.1.	Stationary Object. . . . .	50
5.4.2.	Moving Object with Linear Trajectory. . . . .	53
5.4.3.	Tracking Moving Object with Random Trajectory. . . . .	58
5.4.4.	Grasping Object Steps. . . . .	60
5.5.	Correction Error and Prediction Error . . . . .	61
5.6.	Comparative with Another Work. . . . .	64
6.	CONCLUSION AND FUTURE WORK . . . . .	67
6.1.	Conclusion . . . . .	67
6.2.	Future Work. . . . .	68
	REFERENCES . . . . .	R1
	APPENDICES A . . . . .	A1
A.	CURRICULUM VITAE . . . . .	A1

## LIST OF FIGURES

### FIGURES

<b>Figure 1</b>	Visual servoing types.....	5
<b>Figure 2</b>	Kalman filter cycle.....	6
<b>Figure 3</b>	Kalman filter block diagram.....	10
<b>Figure 4</b>	Basic logical operation between two images.....	12
<b>Figure 5</b>	Erosion morphological operation.....	13
<b>Figure 6</b>	Dilation morphological operation.....	13
<b>Figure 7</b>	Opening morphological operation.....	14
<b>Figure 8</b>	Closing morphological operation.....	14
<b>Figure 9</b>	Median filter pixel neighborhood.....	15
<b>Figure 10</b>	Coordinate frames of Lynx6 robotic arm.....	17
<b>Figure 11</b>	Link frame.s are attacheded so that frame <sub>i</sub> is fixed to link <sub>i</sub> . .....	18
<b>Figure 12</b>	Top view of lynx6 robot arm.....	22
<b>Figure 13</b>	Plane geometry associated with a 3-link planar robot.....	23
<b>Figure 14</b>	Object orientation.....	25
<b>Figure 15</b>	Gray scale range.....	37
<b>Figure 16</b>	Object detection flow chart.....	39
<b>Figure 17</b>	Object tracking flow chart.....	42
<b>Figure 18</b>	Block diagram of experimental setup.....	43
<b>Figure 19</b>	Lynx6 robot arm.....	44
<b>Figure 20</b>	Frame assignment for the Lynx6 robot arm.....	45
<b>Figure 21</b>	Servo motor PWM control.....	47
<b>Figure 22</b>	SSC-32 controller card.....	47
<b>Figure 23</b>	Webcam.....	48
<b>Figure 24</b>	Camera scope.....	49
<b>Figure 25</b>	Tracking predicted corrected and stationary objects. ....	51

## FIGURES

<b>Figure 26</b>	Tracking predicted, corrected and stationary objects tracking according to the number of frames .....	52
<b>Figure 27</b>	3D robot arm end-effector positions .....	52
<b>Figure 28</b>	Tracking predicted, corrected, and measurement for 1 cm/sec moving object. ....	54
<b>Figure 29</b>	Image snapshot .....	54
<b>Figure 30</b>	Tracking predicted, corrected, and measurement for 2 cm/sec moving object. ....	55
<b>Figure 31</b>	Tracking predicted, corrected, and measurement for 3 cm/sec moving object. ....	56
<b>Figure 32</b>	Tracking predicted, corrected, and measurement for 4 cm/sec moving object. ....	57
<b>Figure 33</b>	Moving object in random trajectory .....	58
<b>Figure 34</b>	Tracking predicted, corrected, and measurement for 4 cm/sec moving object .....	60
<b>Figure 35</b>	Object tracking and grasping steps .....	61
<b>Figure 36</b>	Correction/prediction error .....	63
<b>Figure 37</b>	DENSO robot arm. ....	64
<b>Figure 38</b>	DENSO robot arm and moving object.....	65
<b>Figure 39</b>	Tracking error .....	65
<b>Figure 40</b>	3D robot arm end-effector positions.....	66

## LIST OF TABLES

### TABLES

<b>Table 1</b>	DH Parameters for Lynx6 Robotic Arm .....	19
<b>Table 2</b>	Joint-Motors Robot Arm .....	45
<b>Table 3</b>	SSC-32 Properties .....	48

## LIST OF ABBREVIATIONS

CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
CV	Computer Vision
DH	Denavit Hartenberg
DOF	Degree of Freedom
GPC	Generalized Predictive Control
IAEKF	Iterative Adaptive Extended Kalman Filter
IBVS	Image Based Visual Servoing
IP	Image Processing
KF	Kalman Filter
MO	Morphological Operation
NASA	National Aeronautics and Space Administration
PBVS	Position Based Visual Servoing
PC	Personal Computer
PD	Proportional Derivative
PID	Proportional Integral Derivative
RAM	Random Access Memory
RGB	Red Green Blue
SE	Structure Element
USB	Universal Serial Bus

## CHAPTER 1

### INTRODUCTION

#### 1.1. General Review

Many applications existed today show the need of integrating between robotic systems and computer vision system, these applications based on the idea of controlling robot system motion by depending on the amount of data have been extracted from captured image via camera.

In the medical field, especially the applications that related to perform surgical operations. Whereas, the surgeon can operate by looking at the image that gotten from the camera and move the robot arm to certain positions. In addition, the controlling of unmanned vehicle system that use as exploration the environment that used in military applications. Also positioning, moving some objects, assembling and disassembling mechanical parts at industrial applications, painting, and welding, all these applications and examples refer to the reason of developing this field of science.

The needs for all these applications that are the main part in many fields motivate the scientist to combine between image processing and tracking algorithms with the robot system.

The image processing used to improve the image by remove noise and apply many operations to extract features from the image. The tracking in general refers to the problem of using the information of sensors to determine location, path, and characteristic of target object. The sensor may be radar, infrared, microphone, ultrasound, or camera. The tracking is divided into two main types; passive and active tracking. The passive racking, which store Information in the device (for example, memory card), and analyses this information on a computer. The second type of tracking is active tracking; it is also called real time tracking,

and that's what we concern with. The object tracking refers to many steps have to perform that composed extracting the object of interest from a video and keep tracking its motion, orientation, and prevent occlusion.

Tracking enters many applications in different fields, for example the air space monitoring for tracking aircrafts, video surveillance that used in all government buildings, and prediction.

Many steps should implement to achieve tracking; the first step is extracting information from image frame to detect a target that moves at media stream. The next step is following the target from frame to frame and analysis of object tracks to recognize its behavior. The information extraction is mainly elated to computer vision applications like, people tracking, traffic monitoring, and remote video surveillance.

The combination between robot system, image processing, and tracking algorithm refer to a new branch of science that called visual servoing. The aim of this field is controlling the robot arm motion depending on the information have gotten from vision system.

The visual servoing system is majorly introduced for autonomous robots that work in uncertain or unstructured environments. In general, the system is composed of: a vision system that represented by the camera, a processing or computing unit, and specific image processing algorithms to control the motion of the robot end effectors relative to the object workspace.

A camera mounted on the end effectors of five degrees of freedom robot arm, the information moved to the computer using a USB port, and applying image processing operations to extract features from image frame, KF is used to predict the position of the moving object, the robot arm is commanded using SSC-32 controller via serial port, that generate pulses to move robot arm servo motors with certain angles. A system tested at different speeds of the object also random speed with random trajectory is applied. The system shows the robustness at tracking and grasping moving object with different velocities. The system succeeded in tracking moving objects with speed less than 5 cm/Sec.

Visual servoing systems have been increasingly used in controlling of a robot arm that is depend on visual comprehension of robot-arm and object position, and that consider by a wide range for researchers and scientist to develop this field of science

by the combination between computer vision, image processing and merging them with Robotics and other real time control systems.

## **1.2. Thesis Organization**

A tracking algorithm with an image processing is the main idea that we concerned in at this thesis, these methods applied on the robot arm with eye-in-hand technology.

This thesis has six main chapters, which are organized as follows:

Chapter 2, the background theory for the used algorithm and methods are composed, Kalman filter discussed in this chapter, and that what we related to use as tracking algorithm, edge detection with morphological operations, median filter as an image-processing field. The robotic also presented in this chapter that involved with forward and inverse kinematics.

Chapter 3, related to the methods and algorithms has been used by other researchers that represented by eye-to hand, hand-to-eye, single camera, and stereo vision. The proposed method is introduced in chapter four that included by the algorithm used also the edge detection techniques and the washing of partial windows, and the algorithm for tracking the object to get the object and grasp it. The results of our thesis are involved in chapter five, our method is tested at this chapter, and the object is moved by different speeds to test the system. Finally the future work and conclusions presented in chapter 6.

## CHAPTER 2

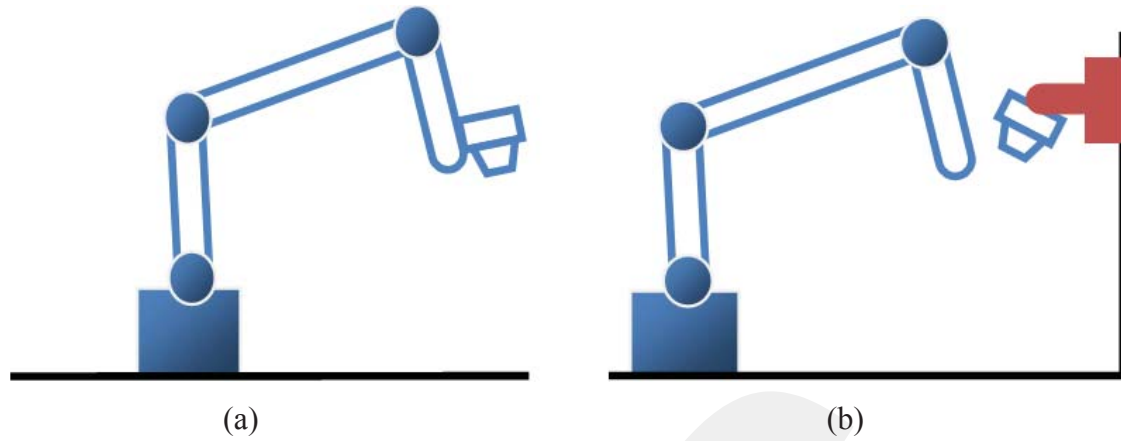
### BACKGROUND THEORY

#### 2.1. Visual Servoing

Visual servoing uses the information that gotten from an image in robot arm manipulator motion control. Visual servoing classified according to camera position and signal camera. The first classification, which based on the position of the camera, has two main approaches the first one is hand-eye or eye-in-hand, and the second approach is eye-to-hand. The first approach that called eye-in-hand whereas the camera is mounted to the robot arm end effector, this approach has more accuracy, mobility and has the ability to explore workspace but it has a limited sight, camera here monitor the object only. The second approach the camera fixed at somewhere around the robot arm manipulator that approve panoramic sight, but low accuracy. Camera here monitors the object and the robot-arm end effectors at the same time (Fig. 1).

The second classification based on the definition of signal error in the feedback control law, which has two methods the first one is the Image Based Visual Servoing and the second one, is Position Based Visual Servoing.

The IBVS measures the error of the signal in the image and commands the robot actuators directly. While in PBVS system, the features that extracted from the image used to estimate the current position of the camera. The error of the position error is calculated in Cartesian space and this error used to enhance the control system [1,2].



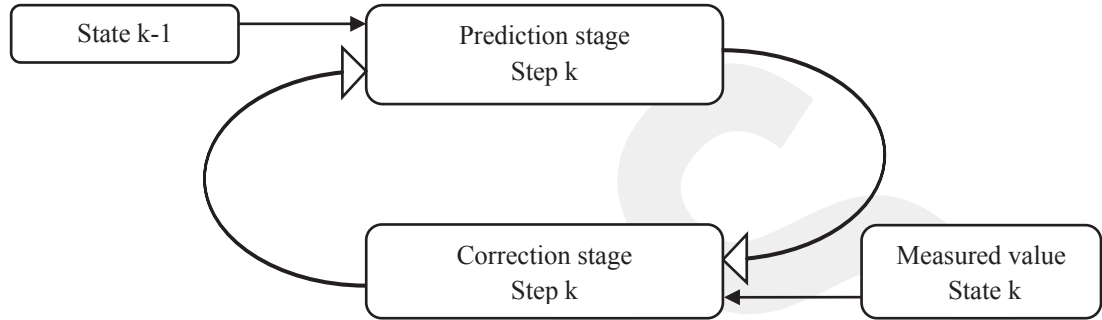
**Figure 1:** Visual servoing types (a) Eye-in-Hand (b) Eye-to-Hand

## 2.2. Kalman Filter

Many researches are supported by NASA especially those which related with tracking, and at that time R.E. Kalman introduced his new method that titled by his name in 1960. His research related to spacecraft tracking and describe a solution that is based on an algorithm that repeated recursively to solve the discrete-data linear filtering problem. From that time and until now, this algorithm has been the subject of a large number of scientists that has a great participation in digital computing. The Kalman filter has been the issue for publishing a large number of researches and many applications based on it, especially in the field that related to the area of self-controlling or assisted navigation. The Kalman filter is composed of a set of mathematical equations that provides efficient computation (recursively) calculations that aim to predict the state of a process with a minimum mean of the squared error. The sturdiness of KF appears at several aspects: the algorithm uses the data of the previous steps in the estimation of the next steps, and it can do so even when the precise nature of the modeled system is unknown [3]. KF enrolled in many applications such as the applications that need to estimate the location, the speed, the temperature, or the voltage. In addition, it has great contributions in tracking different types of moving objects [4]. In the following, a description and some discussion of the basic discrete Kalman filter is presented. The KF used a state space techniques and recursive algorithm to predict the new elements. Kalman filter predict the dynamic system state, the dynamic system can disturbed by noise. To improve the estimated state (values), The Kalman filter uses measured values state to correct the values have predicted in previous.

### 2.2.1. Kalman Filter Stages

Kalman filter mainly separated into two main stages as shown in Fig. 2. The first phase is the prediction stage that estimated by dynamic mode and the second stage is the corrected stage that corrected by observation model (measurement model). The reputation of this procedure leads to minimize the error.



**Figure 2:** Kalman filter cycle

### 2.2.2. Kalman Filter Components

Kalman filter has three components: the state vector, the dynamic model, and observation model. The state vector has the variables of interest that should be estimated, it also describes the dynamic system. The elements of state model could be a position, orientation, or any other variables needed to be estimated. In our thesis the information (elements) of the Kalman is gotten form vision system that represented by image [5].

$$X_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} \quad (2.1)$$

At same time the state vector has two values, the first one is the a priori value that represents the predicted value and the second value is the posterior value that acts corrected value. The dynamic model describes the transformation of the state vector over time. The observation represents the relation among the measurements state. In the linear case, the measurement describes by the following linear system equation.

$$l(t) = H \cdot X(t) + w(t) \quad (2.2)$$

The Kalman filter interested in predicting the state  $x_k$ , which represent a discrete time, controlled process that is described with linear formula:

$$x'_k = \Phi x_{k-1} + Bu_{k-1} \quad (2.3)$$

the measurement values are described by the following formula

$$z_k = Hx_k + v_k \quad (2.4)$$

The state vector  $x'_k$  is defined as a prior state estimate at step-k and  $\hat{X}_k$  is the posteriori prediction at step-k that corrected by measured value  $Z_k$ ,

$$\hat{X}_k = X'_k + K[Z_k - HX'_k] \quad (2.5)$$

After that, the error could be defined as follows:

$$e'_k \equiv x_k - x'_k \quad (2.6)$$

And

$$e_k \equiv x_k - \hat{x}_k \quad (2.7)$$

In the determination of a proper filter gain, a covariance matrix for priori (estimated) errors is calculated as follows

$$P'_k = \Phi P \Phi^T + Q \quad (2.8)$$

Moreover, the posteriori covariance error as flowing:

$$P_k = P'_k - K_k H_k P'_k \quad (2.9)$$

The covariance error Q and measurement noise covariance R values may be constant or updated at each iteration[3]. The matrix  $\Phi$  is n x n matrix that could be constant or changed at each iteration. B matrix related to the optional control. H is an m x n matrix all above matrix might be changed or could be considered as constant.

The (n x m)  $K$  is the gain that has minimized the a posteriori covariance error and the formula as following:

$$K_k = \frac{P'_k H_k^T}{H_k P'_k H_k^T + R_k} \quad (2.10)$$

By focusing in the previous equation and observing the effect of the covariance error R, If the R value close to zero the value of K gain will be near to H,

$$\lim_{R \rightarrow 0} k_k = \frac{1}{H} \quad (2.11)$$

On the other hand, if a priori covariance error close to zero also the Kalman gain  $K$  approach to zero

$$\lim_{P'_k \rightarrow 0} k_k = 0 \quad (2.12)$$

In other word if the measured covariance error close to zero that mean the real measured value  $Z_k$  is more trusty, while the estimated measurement matrixes  $H \times k$  is trusted less and less. On the other hand, as the previous prediction error covariance  $P'_k$  approach zeroes. A real measurement  $Z_k$  is less trusted while the estimated real matrix  $H \times k$  is trusted more and more.

Noisy process matrix,  $Q$  measures the variability of the signal that go away from the perfect state, which is defined by the transition matrix. The amount of  $Q$  matrix is important, whereas the large amount of  $Q$  due to the large variance of the signal, therefore, we need to make the filter more modified. On the contrary, the smaller values in  $Q$  may cause the output to be smoother, but the filter adaptability to large changes is not quite high. Therefore, defining the process noise may require some fine-tuning [3].

### 2.2.3. Kalman Filter Algorithm

Kalman filter predicts the elements in the form of feedback control system. The Kalman filter equations have two groups the first one is the time update equation (correction stage), and the second one is the measurement update.

In time update: the equations of this stage have the ability of estimating the values of next step from the current value also, estimation the covariance error therefore it is called predictor stage.

The measurement update equations have the responsibility of incorporating real variables into a priori estimator to achieve an improvement estimator phase that called corrected stage.

$$x'_k = \Phi x_{k-1} + B u_{k-1} \quad (2.13)$$

$$P'_k = \Phi P_{k-1} \Phi^T + Q \quad (2.14)$$

Notice how the value of the state and the covariance start estimation from (k-1). The measurement updates as following:

$$K_k = \frac{P'_k H_k^T}{H_k P'_k H_k^T + R_k} \quad (2.15)$$

$$\hat{X}_k = \Phi_{k,k-1} \hat{X}_{k-1} + K_k [Z_k - H_k \Phi_{k,k-1} \hat{X}_{k-1}] \quad (2.16)$$

$$P_k = P'_k - K_k H_k P'_k \quad (2.17)$$

Where,  $k$  is the current iteration,  $P'_k$  is the covariance error matrix for the estimated error,  $P_{k-1}$  the estimated error in the previous state.

A predicted covariance error matrix  $P_k$ , measures the estimated precision at period iteration  $k$  and modified over time in the filtering procedure. We only need to consider a possible initial value as follows:

1. When the accurate initial state variables are known :  $P_{k=0} = [0]_{4 \times 4}$
2. If the exact initial state variables are unknown:  $P_{k=0} = \sigma I_{4 \times 4}$  where  $\sigma \gg 0$

The best value of filter gain  $K_k$  which makes the state estimation errors minimum and can be obtained as follows:

$$K_k = P'_k H_k^T [H_k P'_k H_k^T + R_k]^{-1} \quad (2.18)$$

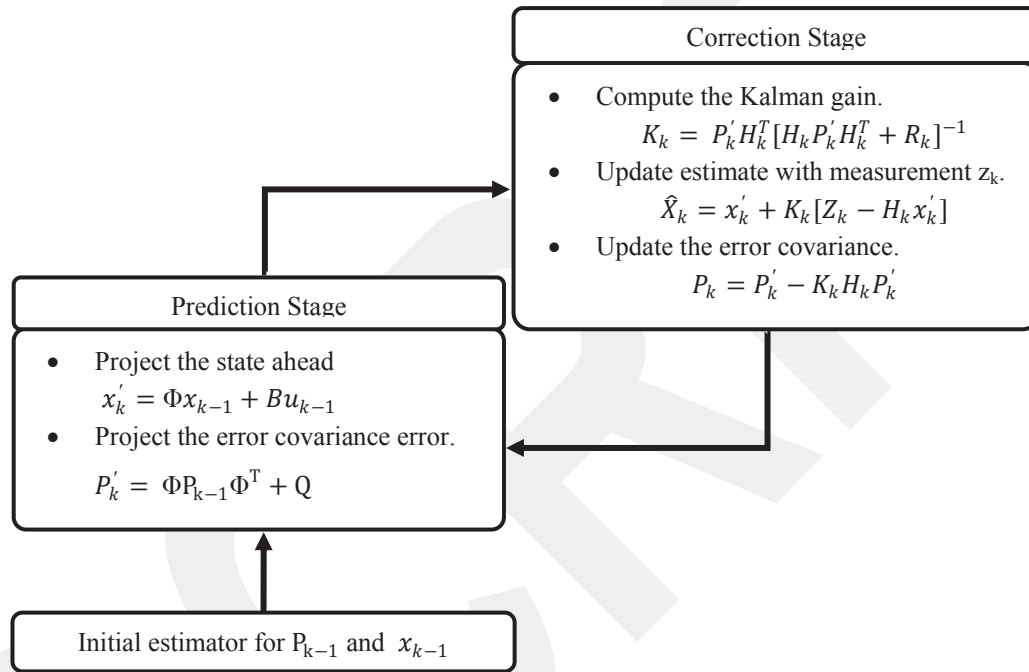
Where  $H_k$  is called an observation matrix, and  $R_k$  is a zero average covariance array of the noise that have been measured, which defines the error of the measuring device. If we can assume that, the measurements are enough accurate, then small values may be used in  $R_k$ . The smoothness of filter reduces the estimated signal error and makes the estimated signal closer to observe signal more and more. On the contrary, the greater value of matrix  $R_k$  means the dependability of the accuracy of the system measurements is reduced also shows smoothness in the performance. The estimation of the state vector  $\hat{x}_k$  from measured values  $Z_k$  is expressed as:

$$\hat{X}_k = \Phi \hat{X}_{k-1} + K_k [Z_k - H_k \Phi \hat{X}_{k-1}] \quad (2.19)$$

Therefore,  $\hat{X}_k$  is updated based on the new elements provided by  $Z_k$ . Before re-using Eq. (2.14) for the next step, the covariance matrix of the predicted error modified as follows:

$$P_k = P'_k - K_k H_k P'_k \quad (2.20)$$

In the next step that falls after  $k + 1$  a new prediction can be supplied performing the same procedure. In the following, a figure shows the process of KF [3]. Fig. 3 display KF block diagram and the equations of each stage cycle.



**Figure 3:** Kalman filter block diagram

### 2.3. Edge Detection

The first step is taking a snapshot using camera to get the digital image. The digital image is consisting of pixels. The pixel defines as the smallest picture element in the image. The picture is in RGB form or in Gray scale form, The RGB form of the image is differ from the grayscale image whereas the difference in image representation, the image has three indices one for each color (Red, Green, Blue), by combining these three colors the other colors can be obtained. In the grey scale image, which we deal with, the image is a two-dimensional function  $(x, y)$ , whereas the x-value and y-value are the coordinates of the image and the value of this image is represented by the amplitude of  $f(x, y)$  [6].

Object detection considered as a basic operation that introduced in CV system because of obtained information from the image is used in tracking algorithm.

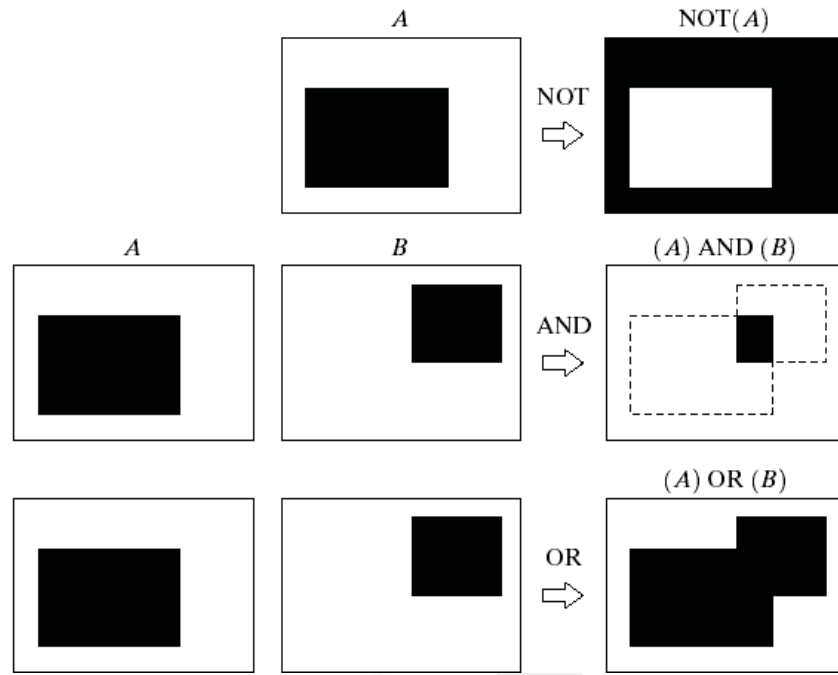
### **2.3.1. Morphological Operations**

Morphological operation (MO) is an advanced form set theory which presented by Matheron as a mechanism for analyzing geometric body of metallic and geologic samples. Serra extended the concept of MO to image analysis. Based on a set theory, MO is a so important theory, whose operation have to be defined by a set of arithmetic operations that gives MO a power for that deal with various problems in IP and CV. MO is advanced from a set of theory.

Closing, opening, dilation, erosion is the main MO operations, theses operations used at detects, modifies, and manipulating the characteristics in the image that based on their shapes.

MO has been represented by a set of algebraically morphological arithmetic operators. The structure element (SE) is a small matrix of pixels with values of zero or one. Morphological operation techniques probe the image with the SE that positioned at all possible locations in the image therefore SE size and shape has an important, crucial task in such type of treatments [7, 8].

Logical operations, although it is simple, provide a powerful complement to the implementation of the image processing algorithms based on morphological operations. AND, OR, and NOT are the basic logic operations used in image processing. The combination of these operations can form any other logic operations. Logic operations are implemented in a pixel basis between corresponding pixels of two images. Figure 4 shows a basic logical operations between two images, the figure shows the two images and the basic logical operations AND, OR, and NOT operations.

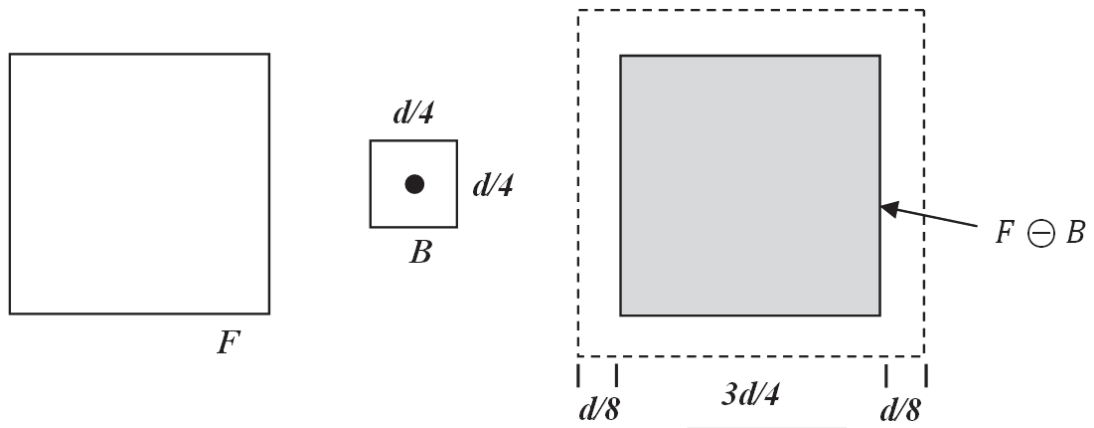


**Figure 4:** Basic logical operation between two images

Dilating image and eroding image considered as the main MO operators because of many other operations based on these two operations. Therefore, these operations are applied to each other. Image opening and Image closing extracted from the performing of dilation and erosion operation simultaneously. Image eroding of the image  $F$  by another SE  $B$ , symbolized with  $F \ominus B$ , also the definition as the following equation:

$$F \ominus B (m, n) = \min\{F(m + s, n + t) - B(s, t)\} \quad (2.21)$$

The symbol inside the circle that acts minus sign refer to the reducing or shrinking of the image therefore the result image is usually less than or the same of the values of the original image that denoted as  $F$ .

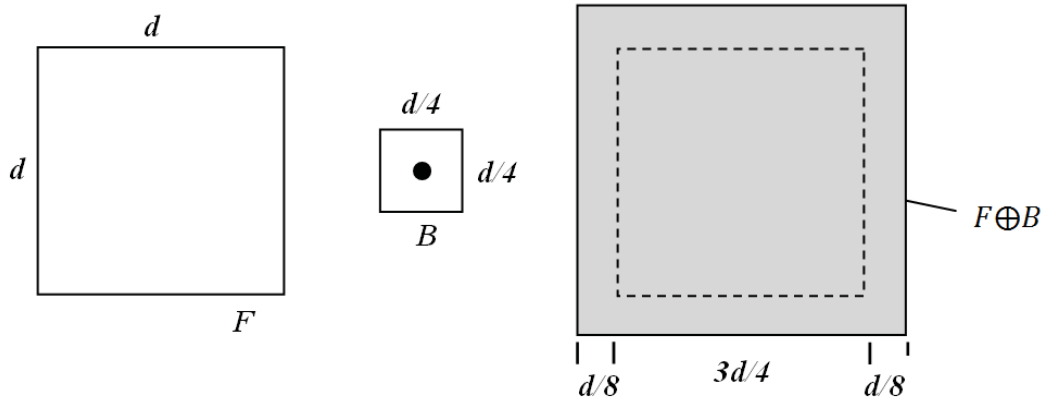


**Figure 5:** Erosion morphological operation

Dilation is opposite of erosion, whereas at the image  $F$  dilates with the SE  $B$ , and symbolized as  $F \oplus B$ , dilation operation is represented as follows:

$$F \oplus B(m, n) = \max \{F(m + s, n + t) - B(s, t)\} \quad (2.22)$$

Fig. 6 shows the dilation MO. The sign inside the circle refer to that the result image I the same original image or greater than the original image.

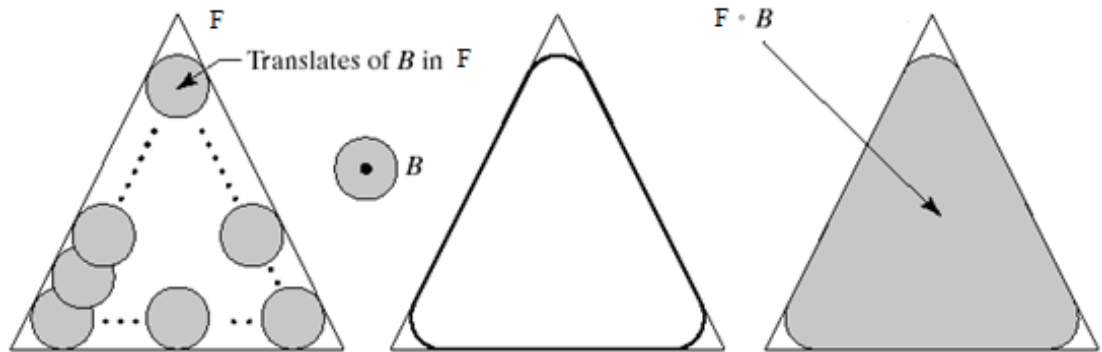


**Figure 6:** Dilation morphological operation

The combination between erosion and dilation produce new other operations, opening and closing is stems from dilation and closing. The same SE this operation called opening operation applies when the SE that denoted by  $B$  and after that the dilation applies the erosion to the image also to the same image. Opening  $F$ -image the  $B$  structuring element, symbolized as  $F \circ B$ , is defined as follows:

$$F \circ B = (F \ominus B) \oplus B \quad (2.23)$$

Gliding the SE along the image contents that represented by pixels along the pixel sequence from beneath and the result is the highest points reached by any portion of the SE. in general the opening operation smoothes the image, and destroy, the narrow holes as shown in Fig 7.

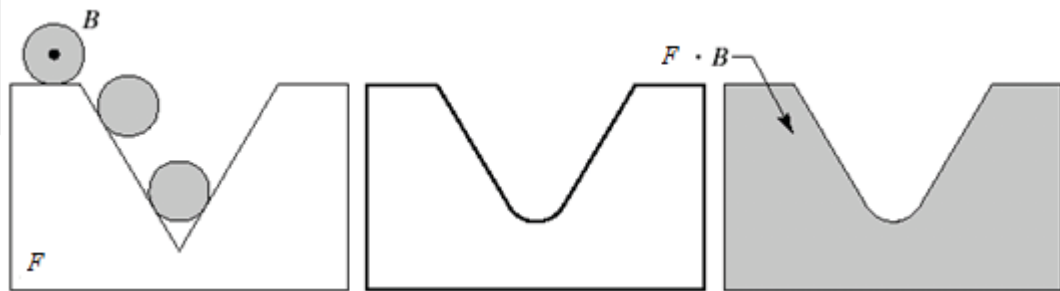


**Figure 7:** Opening morphological operation

The opposite of opening operations is an operation that represented by applying dilation and erosion that applied after dilation by a SE. This operation is called Closing operation the  $F$  image is closed by  $B$  structural element, and symbolized as  $F \cdot B$ , the definition of the closing operation as the following:

$$F \cdot B = (F \oplus B) \ominus B \quad (2.24)$$

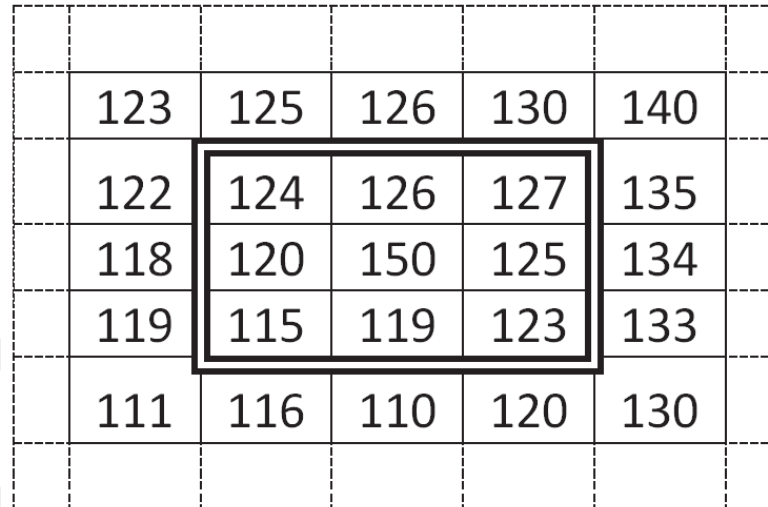
Image closing operation of an image pixel data sequence can be interpreted as sliding or gliding a “flippd-over” version of the SE over. Generally, closing image operation fuses fill up the holes small holes and gaps image [6].



**Figure 8:** Closing morphological operation

## 2.4. Median Filter

Median filter has been widely used filter that used for eliminating image noise and to smooth the image. Median filter considered as an ordering statistically filter and applied in the non-linear sliding window and, median filter is one of the special filters. The name of the filter refers to, the operation of replacing the value of a pixel by the median of the grey levels in the pixel neighborhood, the main value of the pixel is composed in the filter calculations). Applying median filter composed three main steps; the first step is choosing suitable windows. The second step of applying median filter depends on sorts all values of pixels that surrounds neighborhood, and the third step is replacing the value of that falls at the center pixels array of the image window with the ranking result [6]. Fig. 9 explains how the median filter algorithm applied, the 3\*3 neighborhood has values (124, 126, 127, 120, 150, 125, 115, 119, and 123). These values are sorted as the following (115, 119, 120, 123, 124, 125, 126, 127, and 150), the median value is 124. The last step is replacing the central pixel equal to 150 with the median value (124).



	123	125	126	130	140
	122	124	126	127	135
	118	120	150	125	134
	119	115	119	123	133
	111	116	110	120	130

**Figure 9:** Median filter pixel neighborhood

the popularity of median filter due to the excellently noise removing ability , in the consideration of less blurring if it compared to linear filter that has similar size [9].

## 2.5. Robot Kinematics

Robotic manipulators merit specific concern and this due to more than one reason. The first reason is related to that robotic systems enter at many applications, especially the industrial applications. The second reason due to the simplicity formalization of all mechanical robotic systems. The majority of manipulators today, which used in various fields, involved of a set of robust links that combined by joints. The joints provide motion to the manipulator and here the task of motors appears, that support the joint with specific movement. The motors fixed on the manipulator joints, so the mechanism of motion of the arm can be specified by the motors that attached to the manipulator. A gripper also attached to the end effector of the manipulator to provide interaction with the environment. The mechanical structure of the robot arm composed of joints and links, each part of a robot arm, add a special mobility to the arm flexibility, the arm ensures mobility of the robot arm, and the wrist gives a wrist that confers proficiency. The end effector executes the required task of the robot arm [10].

Basic framework that the manipulator is consisted of its series of links and these links can be closed links or opened links. From this viewpoint, the manipulator considered as opened when one part is free, and the robot arm represented as close when the two end effectors are connected as a loop. The mobility of robot arm is achieved by the joints and as the number of joints is increased, the flexibility of the arm becomes larger. There is another classification of the robot arm, this classification depend on the joint physical connection and movement whereas the joint has to be revolute or prismatic joint. The open kinematic, may be revolute or prismatic, and this provides the structure of individual degree-of-freedom that that symbolized as (DOF). The difference between prismatic and revolute joints, the revolute joint provides rotational motion between two links while the prismatic joint provides translation motion between two links. Revolute joints are more popular than prismatic and this due to the flexibility and reliability that provided by revolute joints. In order to execute a specific task the DOF has to be distributed along with the mechanical framework of the manipulator.

### 2.5.1. Robot Arm Descriptor

The human has five DOF, also Lynx6 have the same number of DOF of human in additional to the gripper. In other word Lynx6 robot arm that used in this thesis has five DOF and gripper (5 +1) [11]. The rotation of the shoulder considers as the base of the robot arm, shoulder back as the shoulder of the arm motion, another motion is the motion of the elbow also wrist up and down and its rotation the and forth motion, elbow motion, wrist up and down motion, wrist rotation.

Let us talk about these joints; Joint<sub>1</sub> represents the base and its motion axis is  $z_1$ . Joint<sub>1</sub> gives a rotational angle  $\theta_1$  that rotates about  $z_0$  axis in  $x_0y_0$  plane. Joint<sub>2</sub> identifies the shoulder and its axis is vertical to Joint<sub>1</sub>, and this provides an angular motion  $\theta_2$  in  $x_1z_1$  plane. Joint<sub>3</sub> and Joint<sub>4</sub> define the Elbow and the Wrist respectively. The  $z$  axes of the wrist and elbow, the elbow goes parallel to Joint<sub>2</sub> this give  $\theta_3$  and  $\theta_4$  angular motions in  $x_2y_2$  and  $x_3y_3$  planes respectively. Finally, Joint<sub>5</sub> identifies the grip's roll.  $Z_5$  axis is perpendicular to  $z_4$  axis and it gives  $\theta_5$  angular motion in  $x_4y_4$  plane. All joints are viewed graphically in Figure 10.

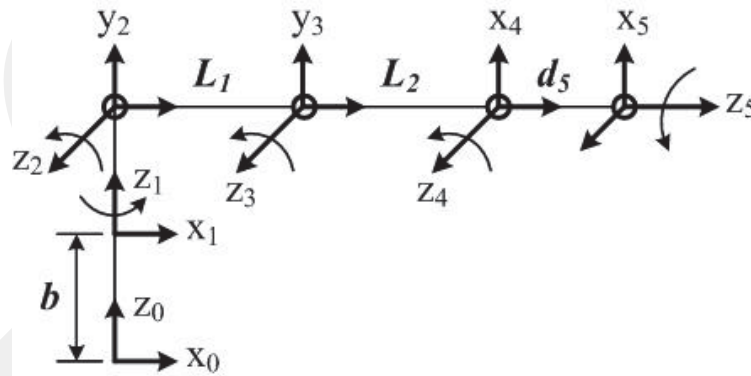


Figure 10: Coordinate frames of Lynx6 robotic arm

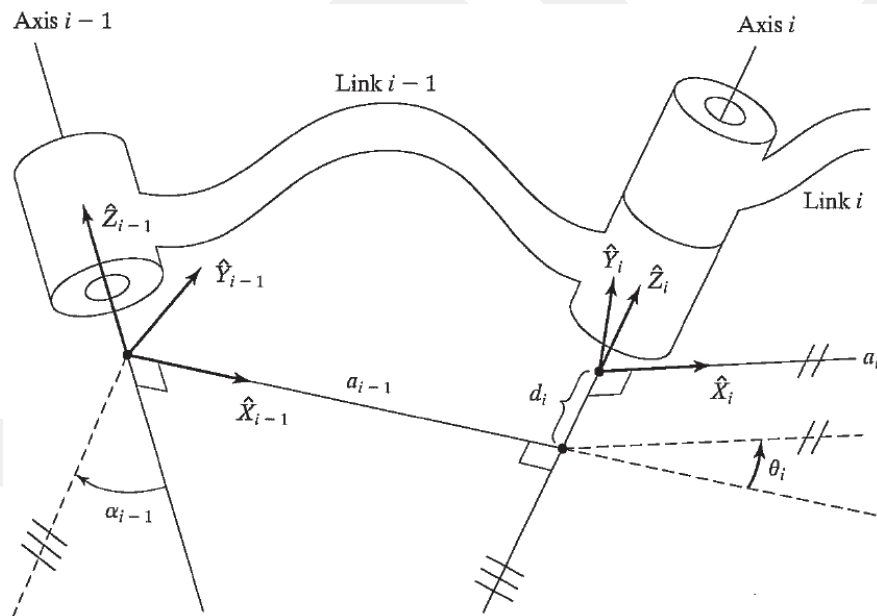
### 2.5.2. Robot Arm Kinematics

The relation that given between the joint motion of the robot arm and the motion that produces a motion of the manipulator is known as the kinematics. The orientation and the position of the robot-arm end effector the motion of gripper are described by kinematics, the description is in the term of joint angles for the robot arm joints. The

kinematic is classified into two categories, the first one is the forward kinematics and the second one is the inverse kinematics.

### 2.5.3. Forward Kinematics

Forward kinematics of the robot manipulator describes the arrangement of the robot arm end effector. The end effectors are determined by tool fixed on it and could be gripper or camera, also the kinematics provides the relative configurations of each pair of adjacent robot arm links. Open-chain manipulator is considered and each link is connected by revolute joints to another link. Whereas, the number of joints starts with 1 and up to  $n$ , the addressing of links starting from the robot arm base, and numbering the links such that joint <sub>$i$</sub>  connects links  $i - 1$  and  $i$ . The base of the robot arm is denoted as link<sub>0</sub> and link <sub>$n$</sub>  is fixed to the robot arm end effectors (Fig. 11).



**Figure 11:** Link frames are attached so that frame <sub>$i$</sub>  is fixed to link <sub>$i$</sub> .

The description of the robot arm kinematics is represented by Denavit–Hartenberg (DH). This method employs four parameters that act the most common method for describing the robot arm kinematics. These parameters as the following:  $a_{i-1}$ ,  $\alpha_{i-1}$ ,  $d_i$  and  $\theta_i$ . Whereas  $a_{i-1}$  is the link length,  $\alpha_{i-1}$  is link twist,  $d_i$  is link offset, and  $\theta_i$  is joint angles. A coordinate frame followed to each joint that produce the determination of DH parameters.  $Z_i$  axis of coordinate frame is referring along the rotary direction of

the joint. The transformation from joint<sub>i</sub> to joint<sub>i-1</sub> is obtained by the matrix transformation [12].

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.25)$$

The D-H parameters of Lynx6 robot arm is defined in the assigned frames in Table1. We can take an example Frame<sub>5</sub> consider is the gripper frame that attached end effectors at joint<sub>5</sub> (see Fig 10).

**Table 1:** DH Parameters for Lynx6 Robotic Arm

Frame	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_{i-1}$
1	0	0	b	$\theta_1$
2	90	0	0	$\theta_2$
3	0	$L_1$	0	$\theta_3$
4	0	$L_2$	0	$90+\theta_4$
5	90	0	$d_5$	$\theta_5$

The parameters substituted to get the following transformation matrices. The matrices  $T_1$  to  $T_5$  can be obtained as shown bellow. For example,  $T_1$  shows the transformation between Frame<sub>0</sub> and Frame<sub>1</sub> (note:  $c_i$  is  $\cos\theta_i$  and  $s_i$  is  $\sin\theta_i$ , etc..).

$$T_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & b \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_2 = \begin{bmatrix} c_1 & -s_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_3 = \begin{bmatrix} c_3 & -s_3 & 0 & L_1 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.26)$$

$$T_4 = \begin{bmatrix} c_4 & -s_4 & 0 & L_2 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_5 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & -d_5 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

When the joint angle of the robot arm end effectors is given and calculation is applied to get the Cartesian coordinates, these calculations called Forward Kinematics. The equation of forward kinematics equations produced from the transformation matrixes which shown above. In addition, final forward kinematics equations of robot arm are gotten by multiplying these five matrices defined as  $T_g$  (according to base). The ordination of the end effectors is positioned at the first three columns in this matrix, and the position of the end effector in the last column. The joint angle representation of the position and orientation of the end effector can be calculated according to the following.

$$T_g = T_1 T_2 T_3 T_4 T_5 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.27$$

$$\begin{aligned} n_x &= -C_1 C_5 S_{234} + S_1 S_5 \\ n_y &= -S_1 C_5 S_{234} - C_1 S_5 \\ n_z &= C_5 C_{234} \\ o_x &= C_1 S_5 S_{234} + S_1 C_5 \\ o_y &= S_1 S_5 S_{234} - C_1 C_5 \\ o_z &= -S_5 S_{234} \\ a_x &= C_1 C_{234} \\ a_y &= S_1 C_{234} \\ a_z &= S_{234} \\ p_x &= C_1 (D_5 C_{234} + L_2 C_{23} + L_1 C_2) \\ p_y &= S_1 (D_5 S_{234} + L_2 C_{23} + L_1 C_2) \\ p_z &= D_5 S_{234} + L_2 S_{23} + L_1 S_2 + B \end{aligned}$$

#### 2.5.4. Inverse Kinematic

The inverse kinematic analysis describes the angles of the joints for desired orientation and position in Cartesian space, in other word we have the Cartesian coordinates (x, y, and z) and would like to get the joint space (the angle of each joint). The solution is more difficult and complex than what presented in forward kinematics because there isn't individual solution. Solving the inverse kinematics problem in general for the planer and special mechanism that is a very interested area of scientists and researcher.

The solution exists if the desired position and orientation is within the manipulator's workspace. The workspace of the robot arm is the section of the environment, whereas the robot arm end effector can reach. The structure and the size of the robot

arm determine the volume and the shape of workspace in addition to the limitation of the mechanical joints. Another possible problem appeared that addressed by multiple solutions at the solving of kinematics equations. A three revolute joints planar robot arm has a large workspace in the plane (large joint ranges and given "good" link lengths), because it is flexible enough to reach to any position in the interior of its workspace with any orientation. Multiple solutions are a new problem has to be solved, and the system should have the ability to choose one solution. The closest solution has been chosen is depending on the criteria that a decision based on.

Because of the non-linearity of the system and equations, the solution misses the form of applying the general algorithm that can be employed to solve a set of nonlinear equations. when all the joints are variable, The motion of joints are composed of the position and the orientation that can be determined by a sequence of steps (algorithm) ,at that time, the system could be considered as solvable [30]. The proposed solution of the manipulator strategies are divided into two phases: the first solution closed form and numerical solutions. the numerical solution is slow and this due iterative nature of the numeric solution. The closed form means a solution method depend on the analytic expressions or on the solution of a polynomial of degree 4 or less, such that non-iterative computations suffice to get the solution. Within the type of closed form solution, it is including two types of solutions: the first solution is the algebraic solution and the second one the geometric solution. These distinctions are somewhat foggy: Any geometric methods brought to be applied by means of algebraic expressions, so the two methods are similar. The difference of two methods is an approach only.

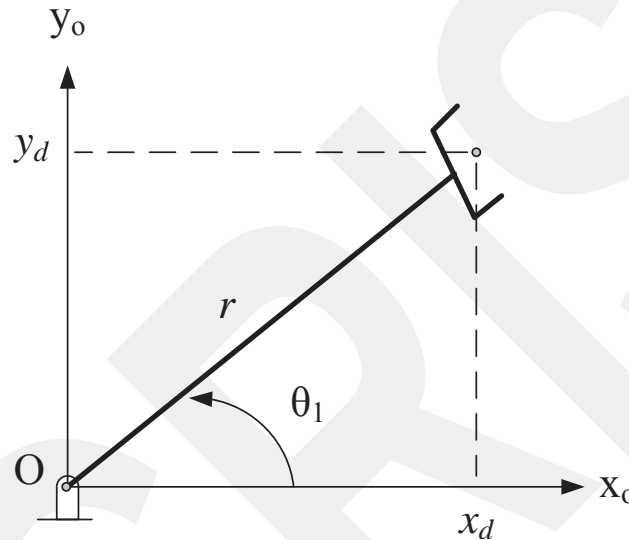
In the geometric solution, the user describes the desired target position of the gripper in Cartesian space as  $(x_d, y_d, z_d)$ . The length  $b, L_1, L_2$  and  $d_5$  are corresponding to the base height, shoulder length, forearm length and gripper length; respectively, and are constant [13]. The angles  $\theta_1, \theta_2, \theta_3, \theta_4,$  and  $\theta_5$  correspond to shoulder rotation, shoulder back and forth motion, and elbow motion, wrist up and down, and wrist's roll; respectively. These angles are updated as the describe the orientation and the position in space change.

### 2.5.5. Base Joint Angle

The top view of the robot arms on the  $x_o, y_o$  plan gives a radial distance (Fig. 12). The radial description distance,  $r$ , from the base is related to the desired target position coordinates  $x_d$  and  $y_d$  by:

$$r = \sqrt{x_d^2 + y_d^2} \quad (2.28)$$

$$\theta_1 = A \tan 2(x_d, y_d) \quad (2.29)$$



**Figure 12:** Top view of lynx6 robot arm

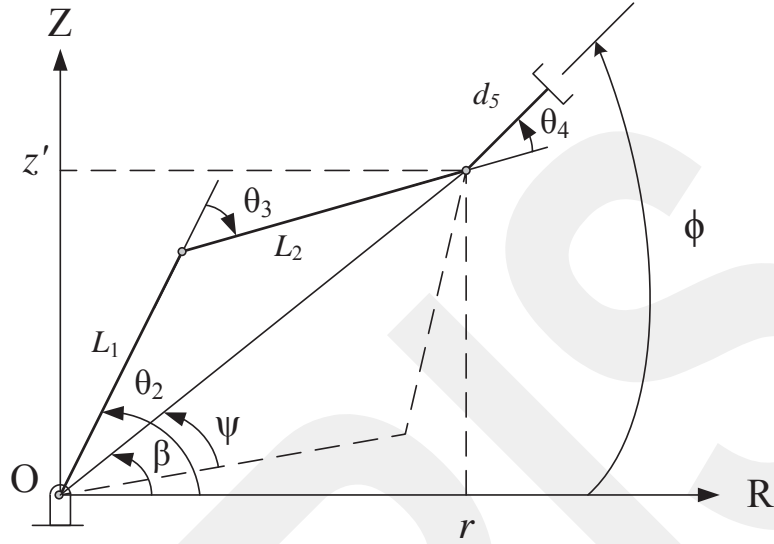
### 2.5.6. Algebraic Arms' Solution

The concentration will be on the inverse kinematics for the wrist of the arm in the absence of consideration the gripper into account. This refers to that the robot arm can be specified as a 2R planar manipulator on a rotating base. Because of the working with a planar manipulator, the description of goal point can be easily specified by specifying three numbers:  $r$ ,  $z_d$  and  $\phi$ , whereas  $\phi$  is the orientation of link<sub>3</sub> in the plane (relative to the +R axis). In addition, by either keeping  $\phi$  fixed in position mode or keeping the wrist fixed relative to the rest of the arm, the inverse kinematic equations can be solved in closed form as shown in the case of a fixed  $\phi$  as shown in Fig. 13.

In order to obtain the  $\theta_2$  and  $\theta_3$ , the distance between the points  $(x_d, y_d)$  have to be calculated from the origin point and the height  $z'$  above the shoulder joint.

$$Z' = Z_d - b \quad 2.30$$

Fig.13 shows the triangle formed by  $L_1$ ,  $L_2$ , and the line joining the origin of the frame o with the origin of frame<sub>4</sub>. The dashed lines are the other possible order of the triangle that would lead to the same position of the frame<sub>4</sub>.



**Figure 13:** Plane geometry associated with a 3-link planar robot.

By applying cosines law in the consideration of the solid triangle, we can apply for  $\theta_3$ :

$$r^2 + z'^2 = L_1^2 + L_2^2 - 2L_1L_2 \cos (180 + \theta_3) \quad 2.31$$

Since

$$\cos(180+\theta_3) = -\cos(\theta_3) \quad 2.32$$

We have

$$C_3 = \frac{r^2 + z'^2 - L_1^2 - L_2^2}{2L_1L_2} \quad 2.33$$

The distance to the destination point ( $\sqrt{r^2 + z'^2}$ ) must be less than or equal to the sum of the link lengths,  $L_1+L_2$  [14]. A solution exists, this equation is solved for that value of  $\theta_3$  which lies between 0 and -180 degrees since only for these values dose the triangle in Figure 3.4 exist. The other possible solution (the one indicated by the dashed line triangle) may be found by symmetry to be  $\theta'_3 = \theta_3$ .

The solution for  $\theta_2$  we'll find expressions for angles  $\psi$  and  $\beta$  as referred in Fig. 12. At first,  $\beta$  may be in any quadrant depending on the signs of  $r$  and  $z'$ . Therefore, we must use a two arguments arctangent:

$$\beta = \text{Atan2}(z', r) \quad 2.34$$

The law of cosines is applied to find  $\psi$ :

$$\cos \psi = \frac{r^2 + z'^2 + L_1^2 - L_2^2}{2L_1 \sqrt{r^2 + z'^2}} \quad 2.35$$

Where the arccosine must be solved so that  $0 \leq \psi \leq 180^\circ$  in order that the geometry which lead to the above equation is preserved. Then we have:

$$\theta_2 = \beta \pm \psi \quad 2.36$$

Where the plus sign is used if  $\theta_3 < 0$  and the minus sign if  $\theta_3 > 0$ .

Finally, it can solve for the sum of  $\theta_2$  through  $\theta_4$  as:

$$\theta_2 + \theta_3 + \theta_4 = \phi. \quad 2.37$$

From this,  $\theta_4$  can be solved because angles  $\theta_2$  and  $\theta_3$  are previously obtained.

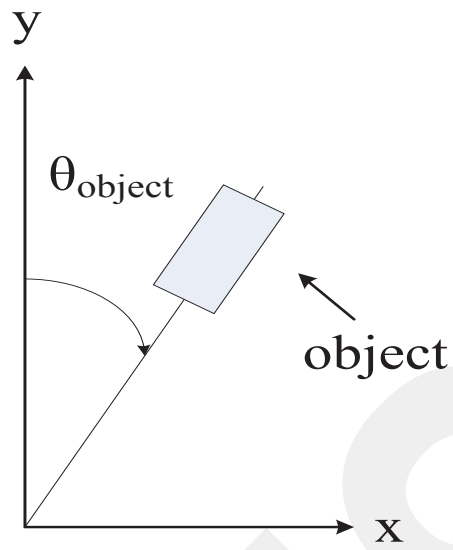
### 2.5.7. Roll Angle of Wrist

Many applications need to know the orientation as well as the position. When the base rotates, the wrist rolls angle ( $\theta_5$ ) changes. Therefore; to get a constant roll angle for the gripper in the case when  $\phi = -90^\circ$ , the value of  $\theta_5$  can be calculated as:

$$\theta_5 = \theta_1 \quad 2.38$$

Now, if the object has orientation does not same as the orientation of the robot arm end effectors, at this case the equation is modified according to the Fig. 14.

$$\theta_5 = \theta_1 + \theta_{object} \quad 2.39$$



**Figure 14:** Object orientation

## CHAPTER 3

### LITERATURE REVIEW

Using vision sensors in controlling the motion of robotics systems have many applications and become a field of research for scientists. The controlling strategies of robot arm classified in to Eye-in-Hand, Eye-to-Hand, and the combination between them.

#### 3.1. Eye-to-Hand

Many researchers used Eye-to-Hand technology, where the camera is mounted on a place in the workspace, the camera here is fixed and not moving with the robot arm and monitor the object and robot arm end effector. In [15] Peter et al. Present tracking non stationary object by achieving an interaction between real time vision systems, the system is able to track moving object by using eye-to- hand stereo vision, the benefit of using two cameras is to get 3-D information. The speed of the system is approximately equal to the speed of the human arm, the tracking and grasping an object is stable because of the using of vision information in feedback control.

The pipelined parallel image processing computer is the main part of the system, the aim is to perform image processing and it is a that has the ability of process 256x256x8 bit images at frame rate speed, the host computer is used to filter and predict the position in 3-D and a robot arm. Tracking algorithm is based on optic-flow because of optic-flow has the ability to track the object in three dimensions, also imaging processing a high computation consuming that slow down speed and using the PIPE image processing engine is suitable to achieve real time application. A real time robot arm controlling is presented to achieve Tracing the object using information has gotten from vision system. The noise is filtered by the control system filter, predict the position and perform kinematics to convert from Cartesian

coordinate into joint coordinate to move the arm into a new position. The system is robust enough to pick up moving objects and grasp it. The using of optical flow that affected by the illumination also the using of another computer for processing the images is costly represents the main drawbacks of this system. Other works are used Eye-to-Hand but with single camera is introduced.

P. J. Sequeira et al. present a new type of unstandardize eye-to-hand visual servoing based on inverse fuzzy modeling is introduced in [16]. The inverse fuzzy control scheme is applied to a robotic manipulator performing visual servoing for random positioning in the robot workspace. The visible sensors on the interested area give information in a big quantity in order to recognize the robots that the objects are moving in. The benefits of the visible system look clear essentially for the robots that are running on the unstable areas. While, choosing the unstable areas, makes the visibility plays very important rule in allowing the flexibility or the softness in order to support the working area. The model of camera that is constructed on the robot is directed using fuzzy techniques in order to get the controller that is able to govern the system. Moreover, using the inverse fuzzy model techniques allow deriving the inverse robot camera model to obtain the connections and the end effector speeds in a straightforward method. A real time controller is chosen in order to apply the proposed inverse fuzzy model directly. Image feature variations and the connection speeds are addressed by getting the inverse fuzzy model the supports a new approach to the classic jacobian model. The mentioned approach is separated than the model of robot kinematics or than the standardizations of the applied camera. In addition, the approach prevents the specifications of the online Jacobian inverting. The obtained model is suitable for the place of the robot to be under camera observation. Moreover, the inverse fuzzy model controller is applied on parts sited out of the interested area because of the input and output prosperities, which are mainly, used for models identification issues.

In addition, many problems for industrial robots that have a large number of degree of freedom (DOF) also the environment surrounding the robots are not fixed and it has complicated tasks and duties. A new approach introduced by Pomares J. in order to obtain a high performance also to improve the system maintainability and traceability [17]. Three degrees of freedom manipulator that uses external camera with eye to hand technique.

This paper has described a new direct visual servoing system with chaos compensation. The control of trajectory based on information extracted from an image. An implementation of the dynamic visual servoing system was proposed. The division of using this design, control schemes and physical properties of the controlled system are decoupled, allowing us the reusability of the generated code. A redundancy resolution method has been described at the acceleration level to combine the concepts of perceptibility and manipulability. To get a repeated motion of joints at the time of the target following tracking of a repetitive image trajectory, a chaos compensation system has been included. Different velocities are introduced in this paper presents.

In [16] and [17] , the researchers introduced a an estimation of the target, while another approach that introduce the prediction of the pose of robot arm determine the orientation and position the end effector or camera in respect to an object's coordinate frame using the information extracted from the image, this idea discussed by Farrokh J. et al [18]. Using of 2-D and 3-D correspondence between geometric features and their projections on the image plane is introduced as a solution the solution to pose prediction problem. There are many geometric features that have a high level introduced to solve the problem like lines and conics, have been introduced, point features are typically used to predict the end effector of the robot arm due to their ease of availability in many objects. In robot visual sevoing, the control error can be computed in both the image space, also in Cartesian space, or using both of them both at the same time. Various ways that depend on KF of pose estimation have been discussed. The IAEKF algorithm is proposed to estimate the pose. Many methods have been compared to their performance under different experimental conditions. To obtain superior performance the mechanisms of noise adaptation and iterative-measurement linearization can be integrated within a novel IAEKF algorithm if it compared with other KF-based methods. Many experiments are tested and show the robustness of IAKEF, also IAEKF can improve the pose estimation performance in the presence of erroneous a priori statistics, nonlinear and fast-tracking trajectories and measurement function, erroneous pose initialization, and slow sampling rates.

Danica et al [19] present the combination between two previous cases, the pose estimation and the target estimations. Whereas a combination between grasping simulator and real-time visual servoing is introduced. The main idea is finding the pose position of the object and plan trajectories, depending on a simulator that has Model-based grasp planning that called GraspIt. This simulator is a real time simulator that able to grasp objects in a stable manner and generate trajectory of the moving object. This information used visual control system and then compares the practical grasping and the trajectory with planned ones. The merging of visual control with grasping simulator creates a method that expands the conception of autonomous robot system control.

### **3.2. Eye-in-Hand**

Another approach is introduced, where the camera is attached to the robot-arm end effector of the robot arm, the camera is moving with the object also.

A genetic algorithm and a local search technique of the genetic algorithm to visual recognition in dynamic introduced by Hidekazu S. et al in [20] and used them at real time visual servoing. The camera is fixed on industrial robot arm (Eye-in-Hand), when the camera moves all stationary images move relative to movement of the arm, so tracking motion should be faster than object, here the object is fish swims in the pool and should be picked by a robot arm. The first challenge is the distraction that prevents catching fish among several fishes the second problem is the accuracy that should be enough to place the net under the object (fish) using. A global search feature of genetic algorithm and local search technique of the genetic algorithm is shows the effectiveness to track and recognize the target fish.

Vincenzo et al [21] that presented a new solution that deal with catching thrown ball, whereas a single camera is fixed on the robot arm end effector, introduced the more complex idea. The main idea of using single camera is reducing the computational cost in compare with using two cameras that lead to process two images at the same time. Many researches are introduced to solve the problem of catching the ball, but most of them used two cameras, therefore some improvement in the controller and in the prediction algorithm introduced to solve the problem in 3-D. The proposed method related to Resolved-Velocity image based visual servoing. An iterative nonlinear optimization algorithm is used the prediction of the trajectory for the

thrown ball, which employs 2-D visual measurements together with a ballistic ball motion model.

To initialize the nonlinear optimization algorithm a linear estimation algorithm that based on a first initial collection of ball observations and on a simplified ball, motion equation is employed, resulting in a significant speedup of the proposed method. In order to prove the results given by the proposed estimator, a comparison with the measurements given by an OptiTrack motion-capture system has been provided. In both theory and experimental results on a common industrial robotic set-up the efficiency of the introduced approach has been demonstrated. The experiments tested the system in about 300 throws and the success rate of catching the ball is about 87.5%.

In [22] Andrew et al introduced a closed-loop law to perform visual servoing that robust a global class of the error in image processing. The feature error is used to compute the statistical measure of confidence in the control law level, M-estimator and least median square are employed. A rigid, flexible single link manipulator used eye-in-hand by Luca et al [23] to track moving objects based on an image based visual servoing that used to compute the feedback control directly from the visual information, without knowing any previous information about physical parameter about manipulator. The feedback controller splits into two separate designs a fast feedback that used to stabilize the oscillatory dynamics and slow nonlinear feedback to ensure the actual image features close reach the prescribed desired ones. Simulation focused on the modelling of two links manipulator while experimental results are tested on the single link manipulator.

Hesheng et al [24] introduce a novel approach for the IBSV of robot arms with Eye-in-Hand camera where both point and line feature are considered. A new adaptive controller introduced in the paper that based on IBSV of both point and line features using the uncalibrated camera (Eye-in-Hand) a new algorithm is developed for online simultaneous estimation of the unknown camera parameters and 3-D structure feature. These characteristics of the system confirmed a good performance.

Heng et al used an optical flow based motion tracking algorithm, the system can track user motion in real time without needing for model training before using the

system. The motion tracking achieved 90% precision for background and shows accepted robustness for cluttered background and changing the distance between the camera and the user [25], but still have the drawback of illumination effects.

In [26] Graziano C. et al introduce a method to predict the displacement of camera in eye-in-hand visual servoing. This done by estimating the position of objects using Euclidean structure and then estimate the camera displacement from this estimate to the desired position. The control is computed on the basis of this information. This refers to there is no exact value, there is an estimated value and there is an estimation error that effect to the stability and to the achievement of the visual servoing algorithm. The estimated accurate scaled euclidian reconstruction of the object is created in the first steps of visual servoing by using suitable essential matrix al algorithm and homograph matrix algorithm, and the displacement of the camera of this reconstruction.

In [27], a motion of flow based visual servoing method is proposed by Jorge et al as a method for tracking trajectories, that depend on correction the behavior in the image and 3D space. This method allows tracking of required trajectory between the primary and end configurations that generated by time interval. This way averts the use of information in depending on time and that to ensure right target following, in other hand, this due to increase the limitation systems that depend on visual servoing that have, so far, been proposed. The range of the application of this method was the different manipulation tasks that required great precision. In this application, it does not necessary to obtain given features in the image from the initial features, but a tracking of the desired trajectory between them, fulfilling the desired spatial restrictions is also necessary for the correct development of the task. The system's different parameters, which allow the regulating of the desired precision and speed during the tracking, have also been presented.

Indeazno et al introduced seven degrees of freedom controller. The design and the implementation of a distributed proportional-derivative controller used the Takagi-Sugeno fuzzy framework [35]. One of the advantages of this approach is leading to a considerable reduction in the computational cost if it compared to model-based approaches, the other advantage is existing learning approaches to model inverse kinematics. The structure of the controller is so simple; this reason makes it attractive

in industrial implementations where PD/PID type schemes are in common use. While the initial values of PD gain are learned with the help of model-based controller, an online adaptation scheme has been proposed that is capable of compensating for local uncertainties associated with the system and its environment.

Many accurately experiments have been performed to prove that visual servoing tasks such as reaching a static target and tracking of a moving target can be achieved using the proposed distributed PD controller adaptive scheme can dynamically tune the controller parameters during visual servoing, so as to improve its initial performance based on parameters obtained while mimicking the model-based controller. The proposed control scheme is applied and assessed in real-time experiments using an uncalibrated Eye-in-Hand robotic system with a 7-DOF Power Cube robot manipulator.

A new method that is based on using visual information in the feedback control to solve the problem of tracking trajectory for eye-in-hand system. Changchun H. et al consider the uncertainty; disturbance and unknown camera parameter in this paper in [28]. Visual information used to control Six degrees of freedom manipulator. The coordinates of the image can be extracted from the image by extracting the feature of the image for the target. The main idea is trying to find a controller that ensures both the errors and their derivatives asymptotically close to zero, even if the robot manipulator dynamics and the camera parameters are not available.

Gavin Paul et al presented a system for autonomous exploration to build a map of unknown 3-D complex steel bridge structure using six DOF stereophonic robot manipulator [29]. The proposed algorithm is a trade-off between the expected gain information available to the environment from the viewpoint of remote sensing and manipulator joint angle changes the required to put the sensor in right point of view, and then gets a collision free path through the exploration of safe area in the past and fuse the information that has been collected from multiple perspective to achieve a detailed 3-D map. The integrated system has the ability to explore the environment are not known to the efficiently generate a high quality map. Strategies are presented for calculating the efficiency gain information from a certain perspective, identifying and reach ability from the point of view given in the work space, and avoiding the collision with all of the obstacles in the workspace. A successful mining algorithm is

used to choose a new perspective and safe for the transfer of the entire manipulator in order to explore the environment in least possible number of iterations. The results shows that the system is a valid solution find a solution of the maintenance of the bridge under complex environment .

Another Eye-in-Hand that passed on estimation the pose of the end effector of the robot arm are introduced by Jacques [30], that used six degrees of freedom manipulator to follow unknown three dimension profile. The Kinematic model of the robot is used to reconstruct the displacement along the profile and to allow. Two control methods are used the first one is PI controller and the second one is the generalized predictive control. The visual servoing shows better results with generalized predictive control in simulation stage, in real experiment also generalized predictive control is used because it is suitable for prediction property of the algorithm. The robot follows the profile without prior knowledge. The experimental results show an accurate following of an unknown 3-D profile can be performed at a velocity 5 cm/Sec, the increasing in sampling rate from camera improve the accuracy.

Farrokh [31] and R. Gans et al [1] make a combination between the estimation of end effector pose and the target estimation, whereas Farrokh [31] introduced an inclusive comparison between Position Based Visual Servoing and Image Based Visual Servoing. The comparison includes system stability, robustness, sensitivity and dynamic performance in the Cartesian and image spaces. The camera, target, and robot modeling errors are considered in the comparison. The camera is mounted on the six DOF robot arm manipulator end effector (Eye-in-Hand), the paper deal with stationary objects and use extended Kalman filter for pose estimation

Large motion cannot command by the IBVS and PBVS, but it is close to stable manner when there is no error modeling. The modeling error effect clearly to dynamic stability of two methods. While the two methods shows a comparable sensitivity to the camera and target modeling error. The system Cartesian and image trajectories and time to coverage are affected by the camera, target, and robot modeling errors.

A hybrid switched-system visual servo method that utilizes both image-based visual servoing and position-based visual servoing control laws are presented by Nicholas R. Gans et al [15]. The difference between IBVS and PBVS is in IBVS an error signal is measured in the image, and is mapped directly to actuator commands while in PBVS systems, features are detected in an image and used to estimate the current camera position. The new strategy of new system in which system control is partitioned along the time axis rather than along specific dimensions of the state space, a position error is then computed in the Cartesian task space, and this error is used by the control system. This strategy shows that the stability achieved the pose space and image space simultaneously. The most important thing that is possible to specify neighborhoods for the image error and pose error that the state can never leave. This insures that no feature ever leaves the image, and the robot never moves beyond a specified distance to the goal pose.

### **3.3. Eye-in-Hand and Eye-to-Hand**

Merging between the two categories eye-in-hand and eye-to-hand introduced by Abdul Muis [32] and Vincenzo [20]. Whereas in [32] Abdul Muis et al introduced two kinds of visual servoing at the same time, where the camera is eye to hand form main robot and eye in hand to the second robot. A position-based look and move visual servoing are depended with polyhedral artificial features. Two-robot arm is used, the first robot is 5 DOF manipulator and the second robot is 6 DOF manipulator a camera is fixed on the second robot end effector, by this way the drawbacks of an Eye-in-Hand and Eye-to-Hand are resolved.

In [20] Vincenzo et al present a position- based visual servoing in two robot arm , and two cameras, one of them attached at work space Eye-to-Hand and the other fixed on the arm is Eye-in-Hand. The idea is based on real-time prediction of pose target using Kalman filter.

Only optimal image features used to ensure high prediction precision and reduce calculation cost. The characteristic of using Eye-in-Hand is to get high accuracy and the ability to explore the workspace, but it has limited sight, while Eye-to-Hand provide panoramic sight of the workspace but lower accuracy.

KF used here to predict trajectory when occlusion occurs KF has many facilities over other prediction methods, for example the ability of changing the measurement set during operation, also allows to set up dynamic windowing technique that reduce the time consumed by image processing. The experimental results show that shows a good result in performance.

GCPRIS

## CHAPTER 4

### PROPOSED METHOD

#### 4.1. Introduction

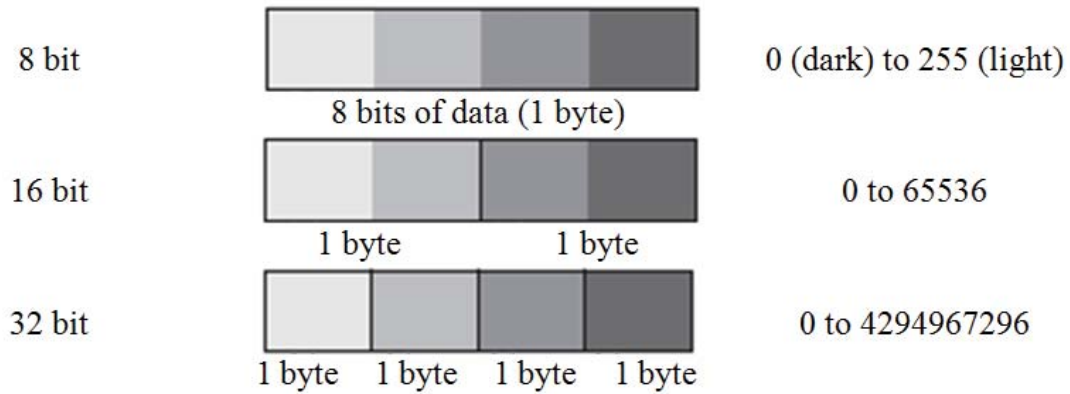
Trying to build control systems based on vision as good as possible is the main goal at this chapter, in the last years several new visual servoing techniques have been proposed. These methods are presented together with some experimentation to evaluate their performance. Inoue describes the term of visual servoing in the early of the 1970s, and specifies the visual information that used in feedback control and due to increase the accuracy of the system [33]. Hill and Park presented the term of visual servoing in 1979 [34]. Prior to the introduction of this term, the less specific term visual have been performed in the development of visual servoing control systems [36].

In this thesis a camera mounted and fixed on the end effector of the robot arm is introduced, it also called eye-in-hand, this technique has the ability of exploring workspace, track the object, and prevent occlusion form occurring. There are many steps applied to achieve tracking. The first step is getting a snapshot using a camera, many detections operations are applied to detect the object, and Kalman filter is used to track the object and grasp it. A partial windowing technology used to minimize the image size of the detection stage.

#### 4.2. Proposed Method

Gray scale image consists of  $x$  and  $y$  special coordinates and their values are respective to intensity value, the intensity value depends on the depth value, where pixel value is equal  $2^k$ . For the 8-bit digital image, value of pixel can be a number between (0 and 255), where 0 is black and 255 is white [6]. The values in between

acts different levels of gray. The size of image increase respectively to the number of bits. The 8-bit gray scale is used because it is less size in comparison to others, see Fig. 15.



**Figure 15:** Gray scale range

The first step is taking a workspace snapshot , the next step is converting the image from RGB format to gray scale format according to the following formula[36].

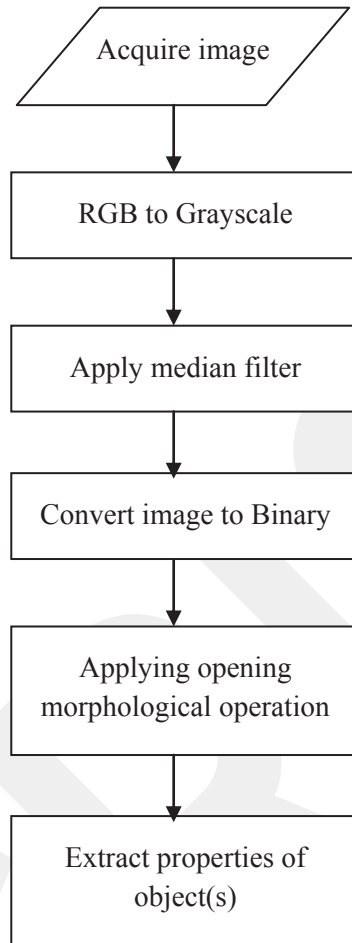
$$Gray_{image} = 0.299 R + 0.587 G + 0.114 B \quad (4.1)$$

Many steps have applied to the gray scale image, which aim to extract the target object out of the background, after converting the image to a gray scale a median filter is applied to the image whereas the intensity value is assigned to the central pixel in the filtered image. Median filter does not reduce the brightness difference across steps because the available value is only those present in the neighboring region, another advantage; the median filter does not shift the boundaries [36].

After applying the median filter, an image is converted to the binary form, this done by applying a comparison of each pixel in the gray image with a threshold value. Zero value, which acts black color, replaces pixel value of the gray scale image which is greater than the threshold value, while pixel value replaced with one value , that represent white color , if a pixel value is less than the threshold value. The threshold value is between zero and one, if the threshold is 0.5 that mean it will compare a pixel falling between white and black in other word it compare with gray color. Choosing the best threshold value is done by adjusting its value until getting best result for many objects with different features. A powerful operation is used for

extracting feature from the image that called morphological operation. These operations are widely applied on binary images to extract image components to describe the region of the shapes also to remove unwanted objects that presented in a frame. Dilation and erosion operations are primitive operations and widely used. Erosion lead to reducing the size of image features, while dilation is the inverse of erosion that rebuild image features [36]. There are two other operations based on dilation and erosion operations the first one is opening and the second one is closing. The erosion MO when followed by a dilation MO this called opening, in contrary the operation called closing when the dilation MO followed by erosion. Opening morphological operation is used in this thesis to detect the object that would like to be tracked. Morphological operations used in many real time applications such as video surveillance and tracking [37]. After applying MO, there is a tool called region properties that extracts the properties of the detected objects, these property can be an area of the objects, bounding box that surround the objects, orientation of the objects, or center coordinates of the object. The center coordinate is used to know the position of the object; also, orientation is important to rotate the gripper of robot arm by the angle of the object. After this step, the detection stage that detects objects in the workspace is completed; Fig. 16 explains the main steps applied to the image to detect the object.

The processing of the image and control strategy computational are time consuming that affect the information flow and cause delays in robot performance. For this reason, the prediction of the moving object can be used to avoid such problems. A monocular (single) camera mounted and attached to robot arm is used to track the object and to overcome the modeling of target. The benefit of using monocular camera in depending on image based visual servoing control that does not need to the geometric model to determine the kinematic relationship between the target and the robot. Another challenge in addition to track moving object is to grasp this object without stopping the motion



**Figure 16:** Object detection flow chart

The coordinates of x and y is in the pixel format and the robot arm kinematics in centimeter format, therefore a ratio should be founded to convert pixel to centimeter. This ratio can be calculated by measure the object with known dimension and apply this ratio to other objects.

A Kalman filter is a proposed tracking algorithm that used for trajectory estimation and position prediction of the object that moves in real time. After detection stage the information like center, and orientation, etc of the object be available. The state vector include object center in two dimensional in other word this vector consist of two values x and y, the estimated vector include estimated x and y that mean we will expect the location of the object.

$$X_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} \quad (4.2)$$

The initializing Kalman filter is needed to initialize the covariance matrix of the estimated noise (P), measurement noise (R), transformation matrix from measure to state a matrix (H), and state transformation matrix ( $\phi$ ). The robot arm should go to estimated location, but at first there are no predicted values, therefore the first predicted value is equal to the current location (the measured values), also others equations should be initialized at first iteration [36]. At the second iteration robot arm goes to predicted location and compute other parameters that related to prediction stage, after the arm be in estimated location the correction stage start its role by correcting the estimated values with measured values. The Following equations are Kalman correction/prediction stages:

Prediction stage:

$$x'_k = \Phi x_{k-1} + B u_{k-1} \quad (4.3)$$

$$P'_k = \Phi P_{k-1} \Phi^T + Q \quad (4.4)$$

Correction stage:

$$K_k = P'_k H_k^T [H_k P'_k H_k^T + R_k]^{-1} \quad (4.5)$$

$$\hat{X}_k = x'_k + K_k [Z_k - H_k x'_k] \quad (4.6)$$

$$P_k = P'_k - K_k H_k P'_k \quad (4.7)$$

For first iteration the prediction equations as following:

$$x'_k = \Phi x_k + B u_{k-1} \quad (4.8)$$

$$P'_k = \Phi P_k \Phi^T + Q \quad (4.9)$$

The procedure of robot arm motion as following, the predicted location of the object that founded by Kalman equations compared with the current location of the robot arm and finds the difference between them:

$$\Delta x = x_{current\_robot} - x_{predicted\_object} \quad (4.10)$$

$$\Delta y = y_{current\_robot} - y_{predicted\_object} \quad (4.11)$$

After that, the robot arm should move by the difference after multiplying by gain ( $G_{gian}$ ) to achieve smooth motion, this gain is between zero and one. At same time, the difference of object orientation and robot arm orientation is founded to rotate the arm by computed angle.

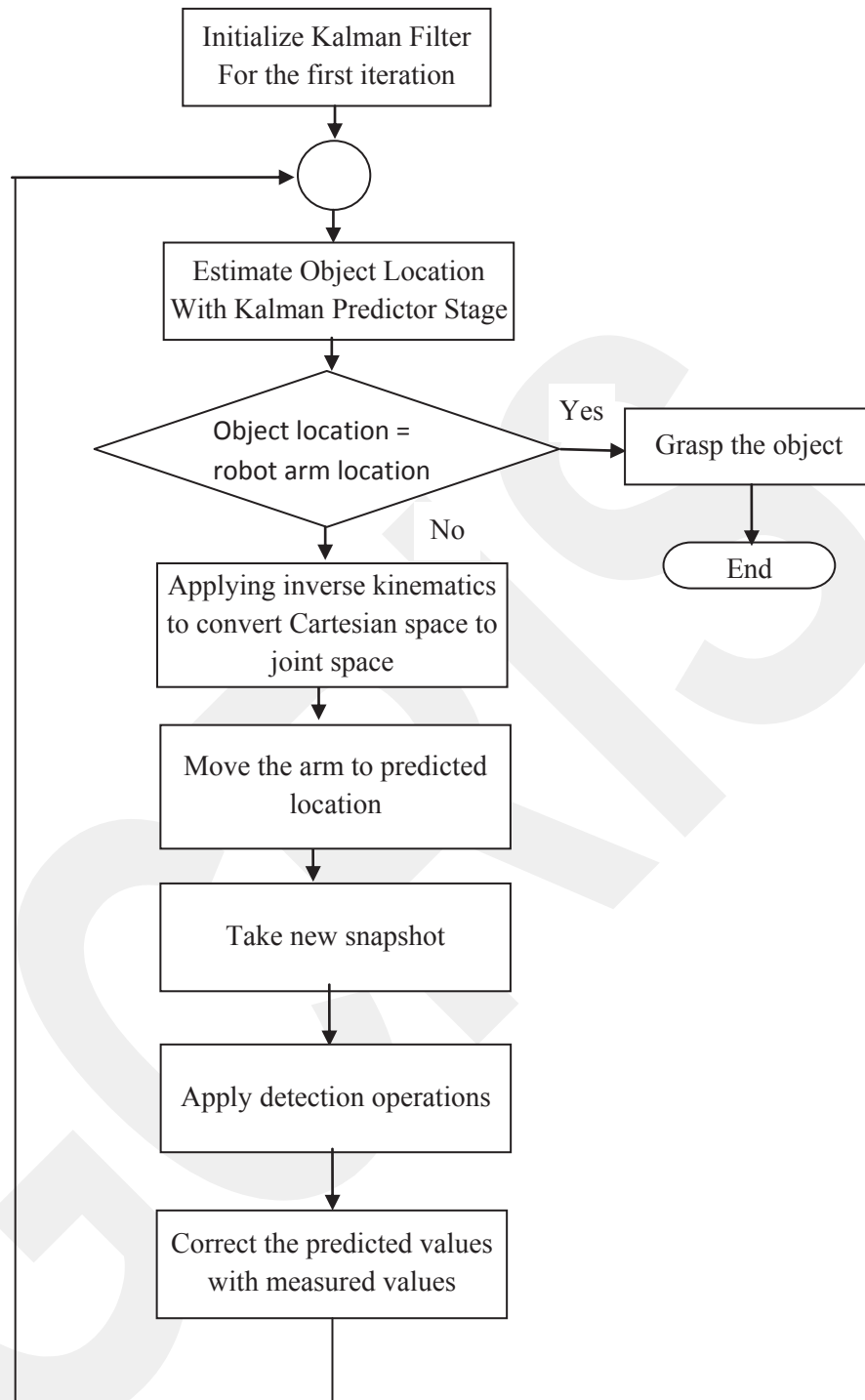
$$\Delta_{orientation} = robot_{orientation} - object_{orientation} \quad (4.12)$$

Also this difference multiplied by gain.

$$x_{robo\ t_{new}} = x_{robot\ previous} + \Delta x * G_{gian} \quad (4.13)$$

$$y_{robo\ t_{new}} = y_{robot\ previous} + \Delta y * G_{gian} \quad (4.14)$$

The values  $x_{robot\ previous}$  and  $y_{robot\ previous}$  are the previous position of the robot arm and for the initial state it is the initial position. The values  $x_{robo\ t_{new}}$  and  $y_{robo\ t_{new}}$  are updated the robot arm position. Inverse kinematics is used because we have Cartesian coordinate and we would like to get joint space to move robot arm by specific angles. These values go out from the computer to a robot arm controller to move the arm to a new position after that the am will take a new snapshot and correct predicted values with a measured values, the procedure is repeated by finding new  $\Delta x$  and  $\Delta y$  and so on. The height of the robot arm still constant until grasp the object. The robot arm track the object until center of object is equal to the center of robot arm position that mean  $\Delta x$  and  $\Delta y$  is equal to zero, at that time the arm swoop to the object and grasp it. Another issue is included, it is the using of windowing technique that mean after detecting the object at the first iteration, the scanning of the object will not be to whole scene that seen by camera but the scanning be to window around the predicted position for the object. By this way the size of image is reduced and leads to the reduction of time cost by image processing [2]. The following flow chart explain the main steps of our algorithm (Fig. 17)



**Figure 17:** Object tracking flow chart

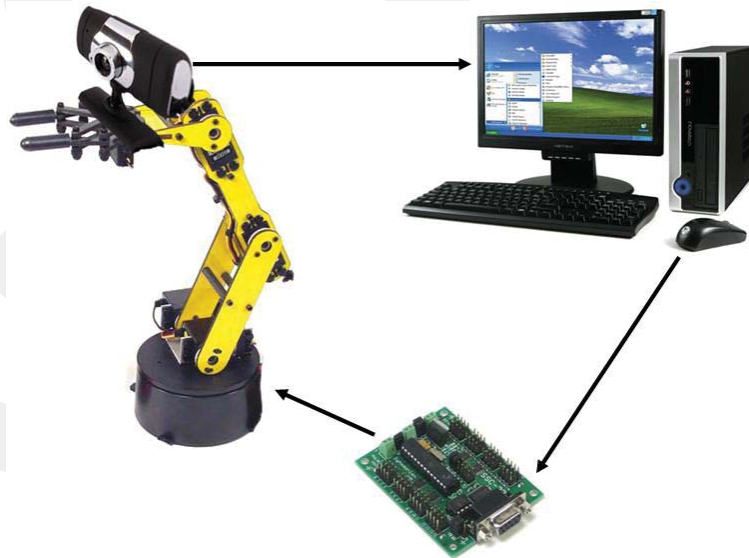
## CHAPTER 5

### EXPERIMENTAL RESULT

In this chapter, the implementation of the proposed tracking algorithm is introduced and tested for various tasks in the real world application. The description of experiment setup that includes the using of five degrees of freedom robot arm, vision system and software part.

#### 5.1. Experiment Framework

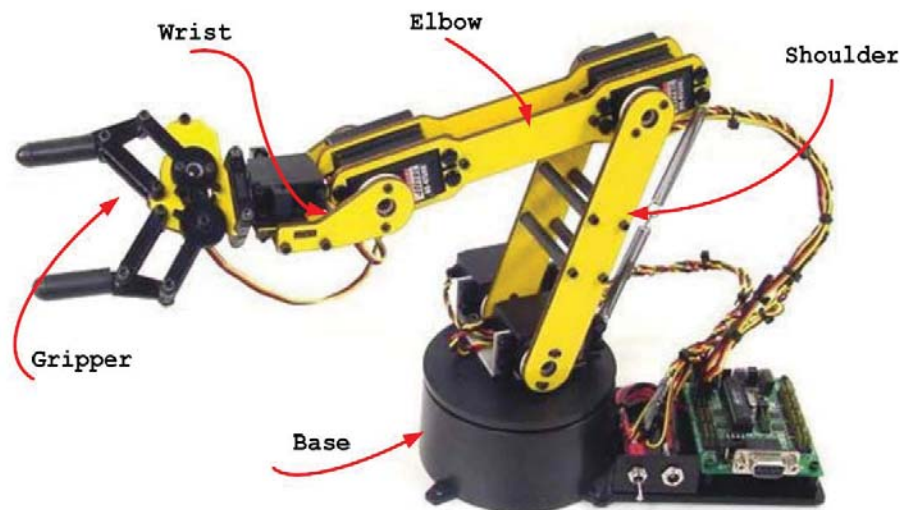
The main parts of our system are Lynx6 robot arm, SSC-32 robot arm controller, personal computer (PC), and webcam, as shown in Fig. 18.



**Figure 18:** Block diagram of experimental setup

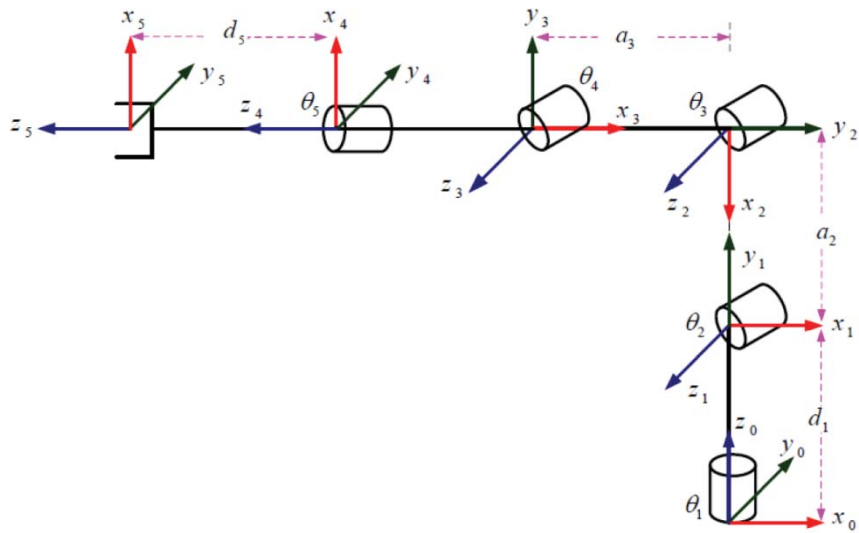
## 5.2. Description of Robot Arm

The Lynx6 robot arm is selected as a case of study due to its small size, lightweight, and price. Lynx6 robot arm consists of five DOF in addition to gripper motion, that is more likely behaves as human arm in the number of joints and the movement [2]. These joints provide shoulder rotation, shoulder back and forth motion, elbow motion, wrist up and down motion, wrist rotation, and wrist's roll motion as shown in Fig. 19.



**Figure 19:** Lynx6 robot arm

The joint angles of Lynx6 are  $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$ . Joint<sub>1</sub> represents the Base and its axis of rotation is  $z_1$ . This joint provides a rotational  $\theta_1$  angular motion around  $z_0$  axis in  $x_0y_0$  plane. Joint<sub>2</sub> is identified as the shoulder and its axis is vertical to Joint<sub>1</sub>. It provides an angular motion  $\theta_2$  in  $x_1z_1$  plane. Z-axes of Joint<sub>3</sub> (Elbow) and Joint<sub>4</sub> (Wrist) are parallel to Joint<sub>2</sub> z-axis. They support  $\theta_3$  and  $\theta_4$  angular motions in  $x_2y_2$  and  $x_3y_3$  planes respectively. Joint<sub>5</sub> is identified as the grip's rotation. The plane  $z_5$  axis is vertical to  $z_4$  and provides  $\theta_5$  angular motion in  $x_4y_4$  plane. A graphical view of all the joints is displayed in Fig. 20.



**Figure 20:** Frame assignment for the Lynx6 robot arm

As shown in the Fig. 20 Lynx6 has six servo motors, joint names and the combination of the actuators that control the motion of these joints are given in Table 2.

**Table 2:** Joint- Motors Robot Arm

Joint name	Rotation angle $\theta$	Actuators
Shoulder (base) rotation	$\theta_1$	HS-422
Shoulder	$\theta_2$	(HS-422) x 2
Elbow	$\theta_3$	HS-422
Wrist	$\theta_4$	HS-422
Wrist rotation	$\theta_5$	HS-81
Gripper	-	HS-81

The characteristic of the robot arm is as the following:

- The distance between axis is equal to 4.75 inches, Gripper opening = 2 inches.
- The style of servo motion control = local closed loop.
- Motion limitation range per-axis=180°.

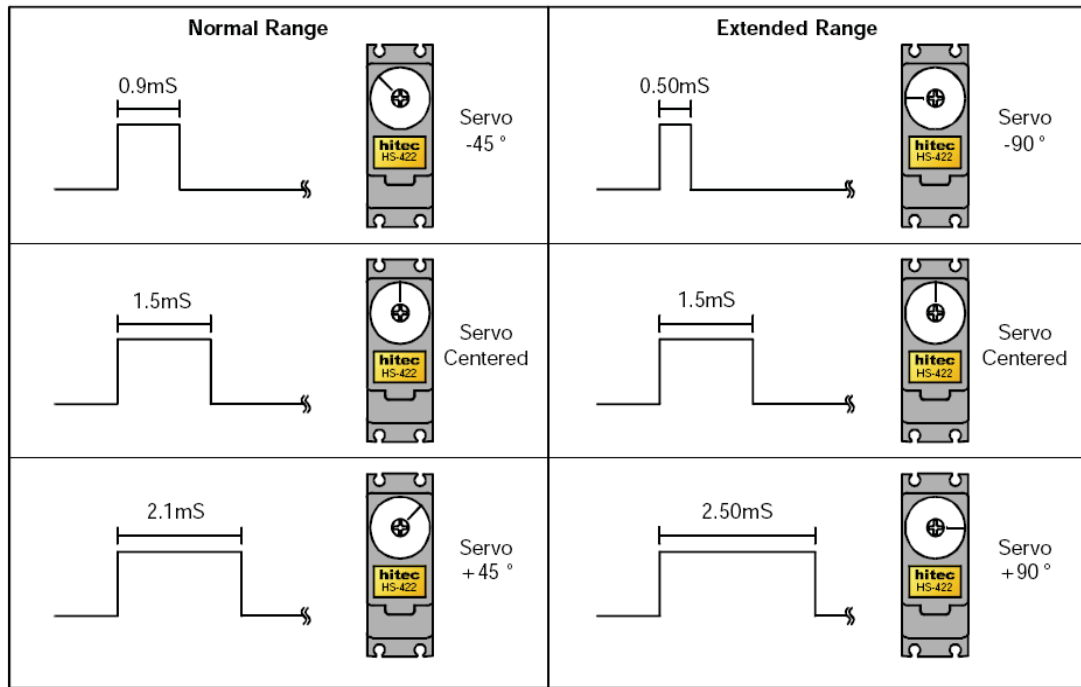
- The Accuracy of the motion per-axis = Servo controller dependant (SSC-32=0.09° ).
- Supported voltage=6 v, Weight of the robot: 2.13lb.

### 5.2.1. Servo Controller Card

Servomotors control the joints and the gripper of Lynx6 robot arm. SSC-32 has the task of controlling the servomotors; SSC-32 is considered as a link between the motors and the personal computer is connected serially. The SSC-32 servo controller has 32 channel servo controls which SSC-32 provide the hardware interfacing between the PC and the robot arm servo motors

The PC transmits commands via serial port with COM port (RS 232) with baud rate equal to 115200. The single command consists of three bytes; the first one has numerical value and holds from 0 to 255, this byte used to achieve the synchronization. The second byte gives the identification of the joints that have to be moved, in other word specifying the servomotors that have to be moved. The second byte provides the identification of the joint to be moved. The third byte value specify an angular position for the joint that is commanded to be moved move.

The controller SSC-32 digitized the signal that comes from PC, whereas the computer supports SSC-32 controller by a certain angle to move the servo. The signal that is generated from SSC-32 to the servo motor consists of positive going pulses ranged between 0.5-2.5 ms (milliseconds) long, and repeated 50 times per second. In Fig. 21 shows the position value of 500 represents 0.5ms pulse, and the position value of 2500 represents to a 2.5 ms pulse. The position value is acted accordingly such that a change in 1uS leads to a change in the pulse width. The positioning resolution is 0.09°/unit (180°/2000). Hereby, the term pulse width and the position are the same, as shown in Fig. 21.



**Figure 21:** Servo motor PWM control.

There are two modes to generate motion via SSC-32 card (Fig. 22) .the first mode is a speed control motion, and the second mode is a time control motion or combining between them. The monitoring of the actual joint position is done through the transducer interface program that runs on the micro controller. The specification of the controller in Table 3.



**Figure 22:** SSC-32 controller card

**Table 3:** SSC-32 Properties

Feature	Characteristic
Microcontroller kind	ATMEGA 168-20pu atmel
EEPROM	24LC32P
Speed	14.75 MHz
Internal sequencer	12 servo
Serial input	RS232
Output	32 servo
input	4 analog or digital
Current requirement	31 mA
Servo travel range	180 degree
Servo motor resolution	0.09 degree
Servo speed resolution	1 $\mu$ s
Board size	3.0" x 2.3"

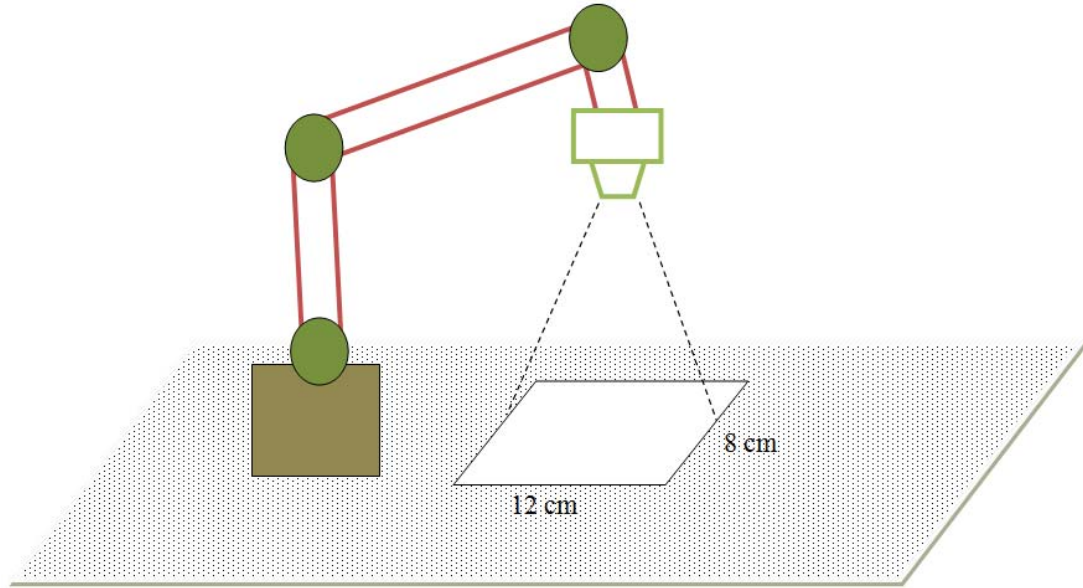
### 5.3. Camera and PC

A CMOS USB webcam is used to take a snapshot to the workspace, as shown in Fig. 23. The maximum frame rate reach to 30 farms per second.



**Figure 23:** Webcam

The camera is fixed at the center of the gripper, the resolution of image that had taken from the camera is 320x240 pixels, the height of camera from the robot base is equal to 18 centimeter, and the dimensions camera scope is 12 x 8 cm as shown in Figure. 24.



**Figure 24:** Camera scope

A personal computer with Dual-Core processor (2.6 GHZ -2 CPU) is used in the thesis, 2GB RAM and windows XP operation system. The software part is implemented using Matlab 2013.

#### **5.4. Experiments**

Many steps are applied to achieve object tracking. The initialization of the Kalman filter step is necessary to initialize the KF coefficients as it shown in the equations 4.8 and 4.9. Another step includes the estimation of object position and this done according to the KF estimation stage (equation 4.3 and 4.4). In order to move the robot arm to the predicted location of the object, the difference between the robot arm location and the object location is computed according to equations 4.10 and 4.11, and then the value of difference is multiplied by a gain, as it is clear in equation 4.13 and 4.14. In order to move the robot arm to any position, the values of joints angles should be available, therefore the inverse kinematics is applied to convert the

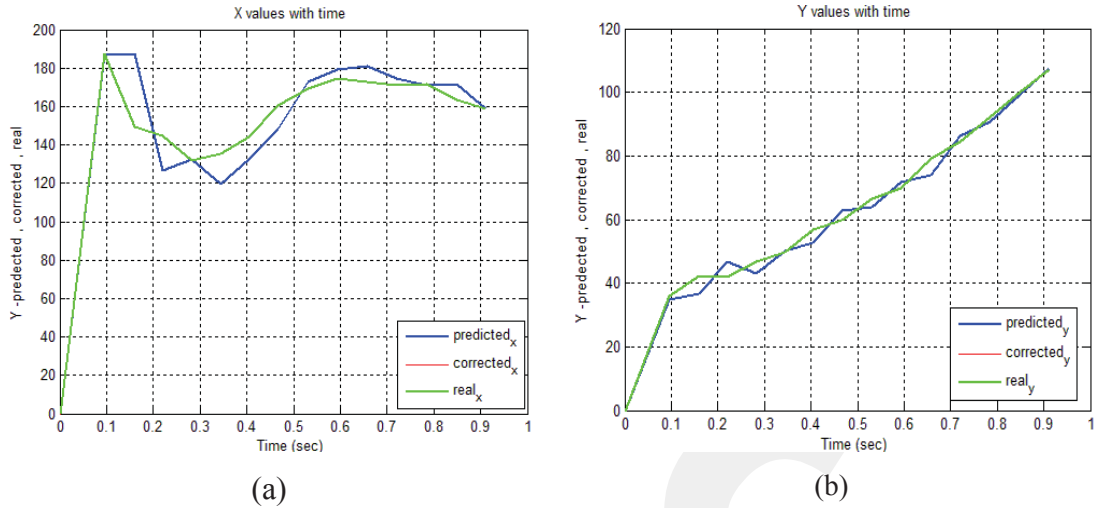
Cartesian space into joint space and specify each joint at which angle value should move. the angle value was digitized by SSC-32 card to generate a discrete motional steps. At the same time, the gripper updates its orientation (equation 4.12) and rotates the gripper according to the orientation of the object, after the robot arm, the camera is moved according to the robot arm end effector motion, moved to the estimated location, the camera gets a new snapshot and apply detection algorithm as mentioned in object detection algorithm. The information has gotten from the detection stage acts measurement values. The measured (real) values are used at the Kalman filter correction stage to update the Kalman filter coefficients with correction equations (equations 4.5, 4.6, and 4.7). From these corrected values, the Kalman predict the new position of the object and repeats the algorithm until the center of the object is equal to the center of the robot arm at that time the arm catch the object.

The initial location of the robot arm is  $x=14$ ,  $y=0$ ,  $z=15$  centimeters, and as soon as the object enters the camera scope, the arm goes towards the object and start tracking it , the correction and prediction stages of the Kalman filter is repeatedly applied and the arm keeps following object until grasp it.

This section presents various sets of experiments to evaluate the performance of the system. Three cases are applied in this chapter, the first case for stationary object, the second case for object moves linearly and in different velocities, and the third case for object moves randomly. The experimental results for tracking and grasping an object with three cases presented previously is applied and discussed by a 5-DOF Lynx 6 robot arm. MATLAB 2013 is used as a software part to get an image from camera, implement image processing, tracking, and commanding the robot arm using the interfacing card between the computer the robot arm SSC-32.

#### **5.4.1. Stationary Object**

Tracking stationary object is the simplest case among others, the object at this case is not moving but, the robot arm here is moving only and the camera that mounted on the arm. The position of the object not falling at the center of robot arm but it is at somewhere in the camera scope, according to our algorithm the arm doesn't moves toward the object position directly but it multiplied by gain therefore it take many steps until the object and the robot arm be at same center. Fig. 25 shows values of prediction, measurement, and correction.

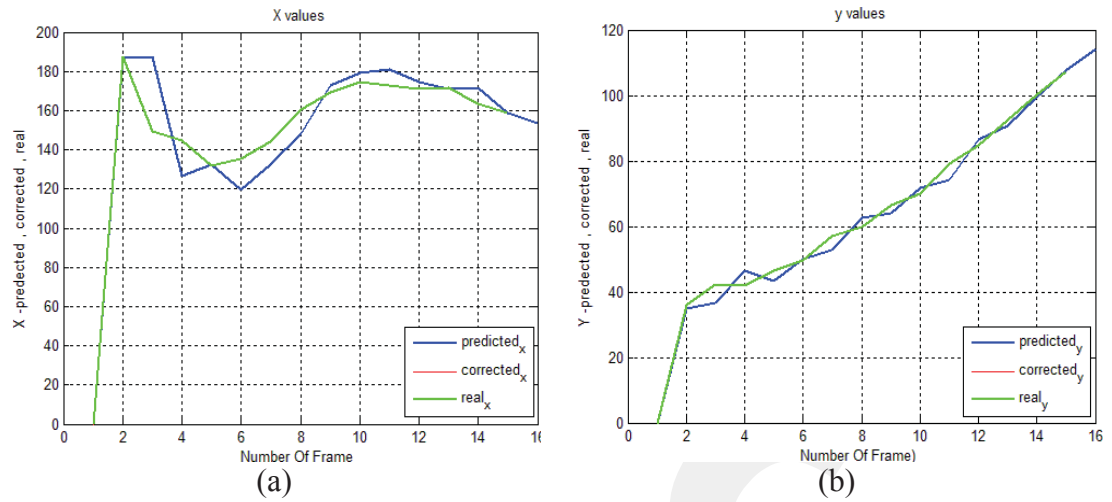


**Figure 25:** Tracking predicted corrected and stationary objects: (a) the x values (b) the y values.

As shown in the figure above, the x-axis is the time (in second) and the y-axis is the number of pixels. As mentioned previously the goal is putting the object at the center of robot arm end-effector (center of the camera). The image dimensions are 320 pixels for x-axis and 240 pixels for y-axis, and the center of image is 160 x 120 therefore the curves is trying to close from the values 120 of the x-axis and 160 of y-axis as shown in Fig. 25 (a, b).

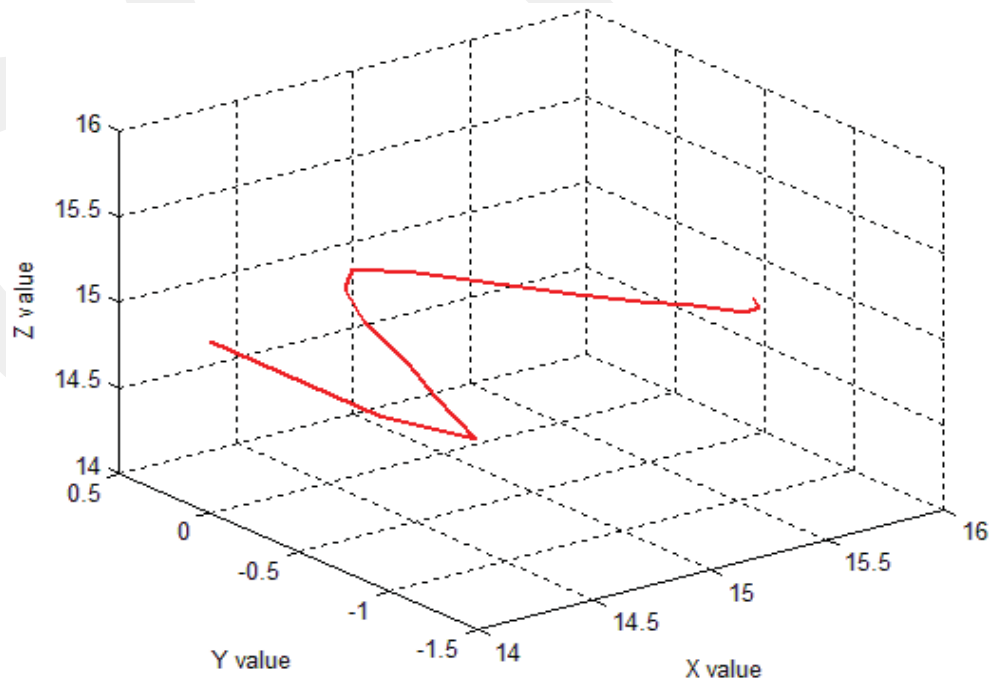
The predicted curve is going with real value and corrected value also the real and corrected are corresponded, whereas the red curve is under the green one that shows that the corrected and real time is matched, and there is a little difference between predicted and real curves, the aim of correcting is improving the predicted values. At 0.9 Sec, the center of robot and object center are matched with coordinating and orientation, therefore the arm grasps the object and grips it by the gripper.

As shown, 0.9 Sec is a very short time and this time costs 16 frames Fig. 26 is the same as Fig. 25 (but in frame coordinate not in time). At Fig. 26 there is some points shows the predictor and corrector is to close, but the robot arm did not catch the object because of the condition of being the object and the robot arm at same centers didn't achieve and that shown at frame two.



**Figure 26:** Tracking predicted, corrected and stationary objects tracking according to the number of frames: (a) the x values (b) the y values

The 3-D trajectory end effectors are shown in Fig. 27, the first position of the robot arm end-effector is  $(x=14, y=0, z=15)$  centimeter, and the object is not under the robot arm directly, figure 4.10 shows the trajectory has moved by the arm from the initial point until achieve the condition and grasp the object. Although the main idea of our thesis goes to track a moving object, but the stationary object also applied to our algorithm and achieves grasping. Whereas the object is stationary, the camera and robot arm is moving only, so it is easier than moving object.



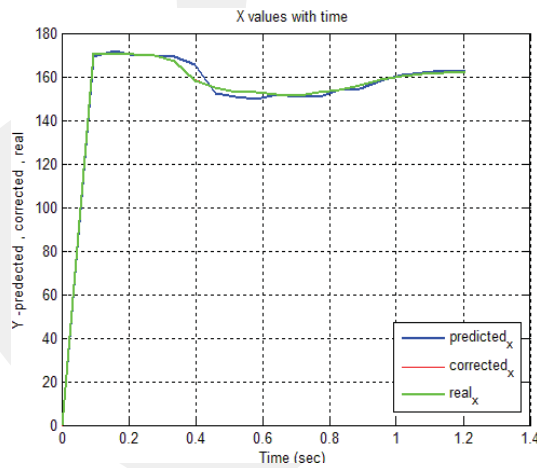
**Figure 27:** 3D robot arm end-effector positions

### 5.4.2. Moving Object with Linear Trajectory

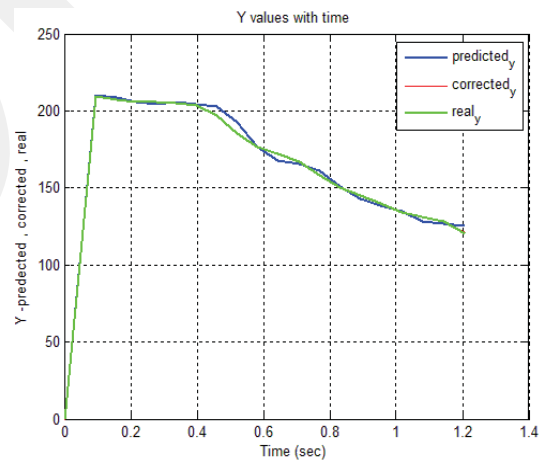
At the following cases, the object is moved linearly with different velocities, 1 cm/sec, 2 cm/sec, 3 cm/sec and 4 cm/sec is introduced to test the performance of the

As soon as the object is in the sight of the robot arm camera, the arm start follows the object and adjusts the rotation of gripper to be with same angle with the object. The arm keeps track the object until both of them move at the same speed, and as soon as the object be at the center of the robot arm the arm grasp the object.

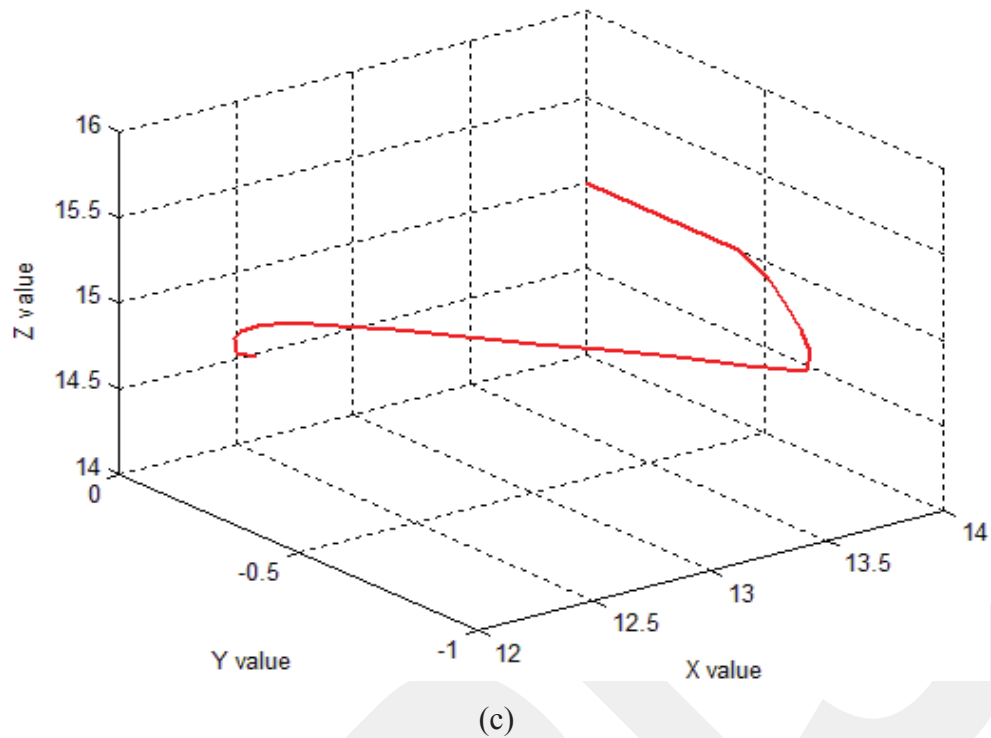
The first applied speed is 1 cm/sec which is the slowest speed introduced in this chapter whereas the robot arm follows the object until grasp it. In Fig.28 (a & b) shows three curves and it is clear how they are close. At 1.2 Sec object center and robot, arm cent is matched and the arm swoops the object. The changes of x and y predicted, measured, and corrected respectively. Fig.28 (c) shows the 3D end effector positions of the robot arm that start form the initial position and start track the object until grasp the object.



(a)

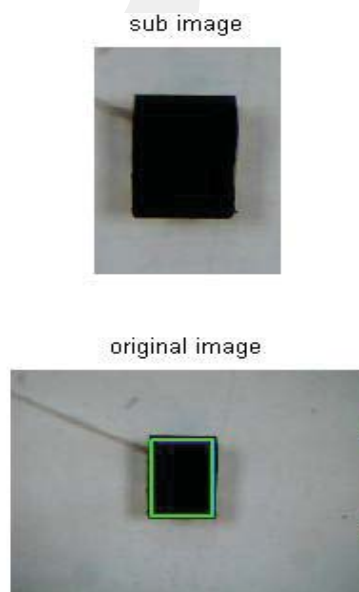


(b)



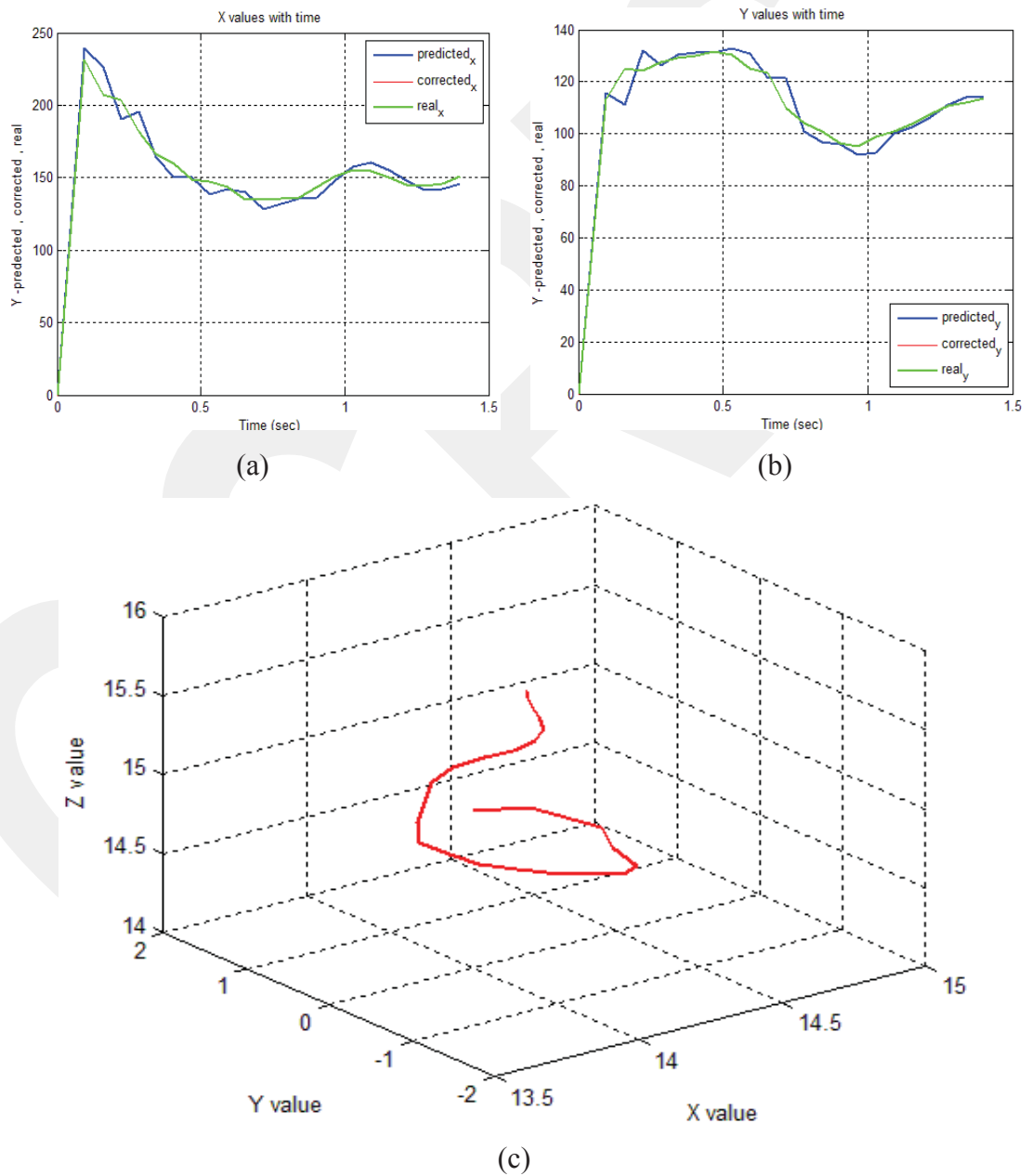
**Figure 28:** Tracking predicted, corrected, and measurement for 1 cm/sec moving object: (a) the x values (b) the y values. (c) 3D robot arm end-effector positions.

Fig.29 shows the whole image that titled with the original image and the partial windowing image, which titled with sub image. The detection operations are not applied to the whole image but it applied to the sub image and the difference between the two images in size and in dimension is clear.



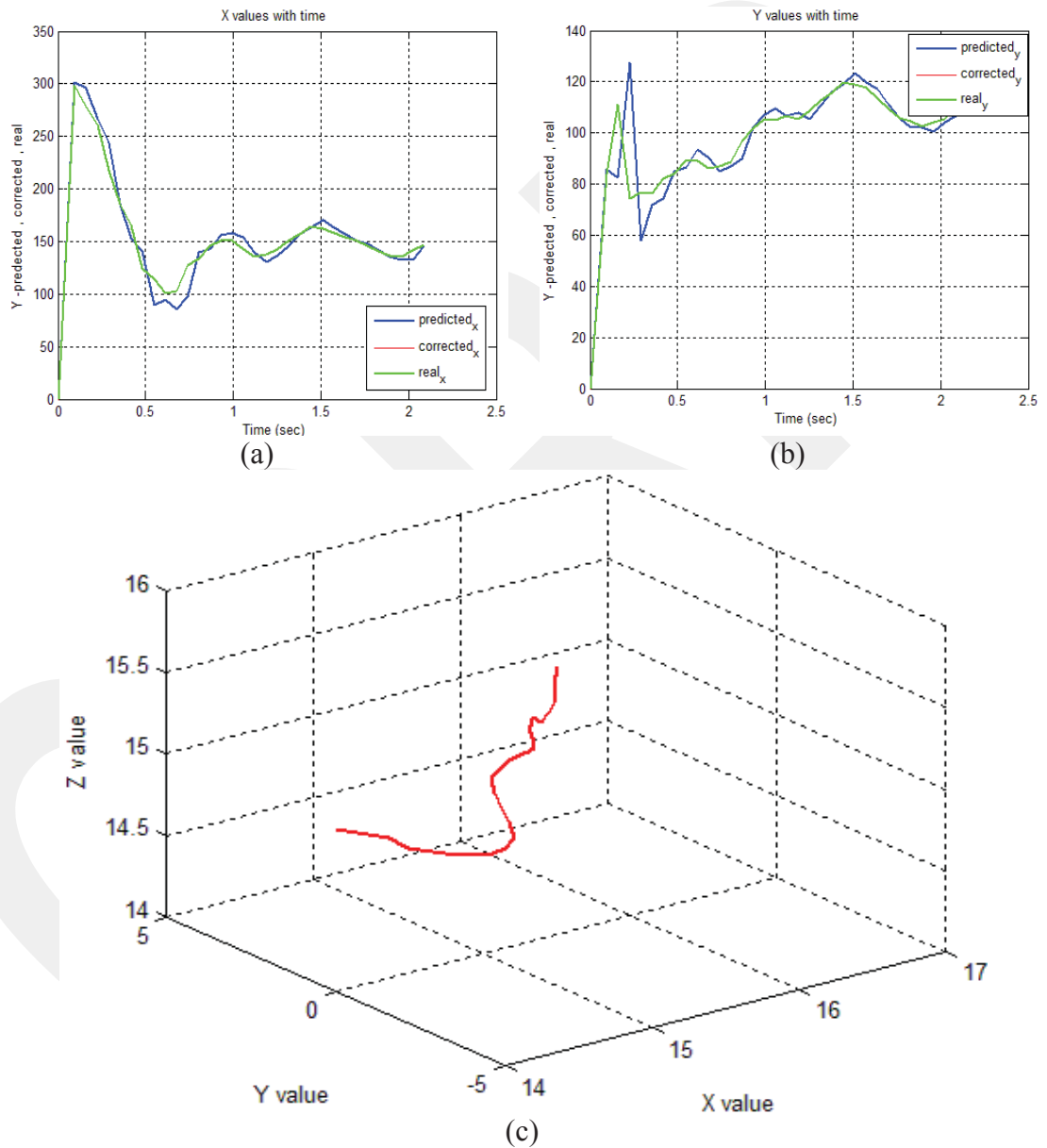
**Figure 29:** Image snapshot

The following figures show figures related to the prediction, correction, and measurement curve for tracking object also a three dimensional of the robot arm end-effector positions for 2 cm/sec, 3 cm/sec, and 4 cm/sec . Figure 30 (a) & (b) shows the x-coordinate tracking curves and y-coordinate for 2cm per sec, this figure shows the prediction and real curves are close together, but many values has some difference and that due to the robot arm oscillation and this a problem are faced in this thesis. Fig 30 (c) shows the three dimensional trajectory of the robot arm and that start to track the moving object form the initial position of the robot arm.



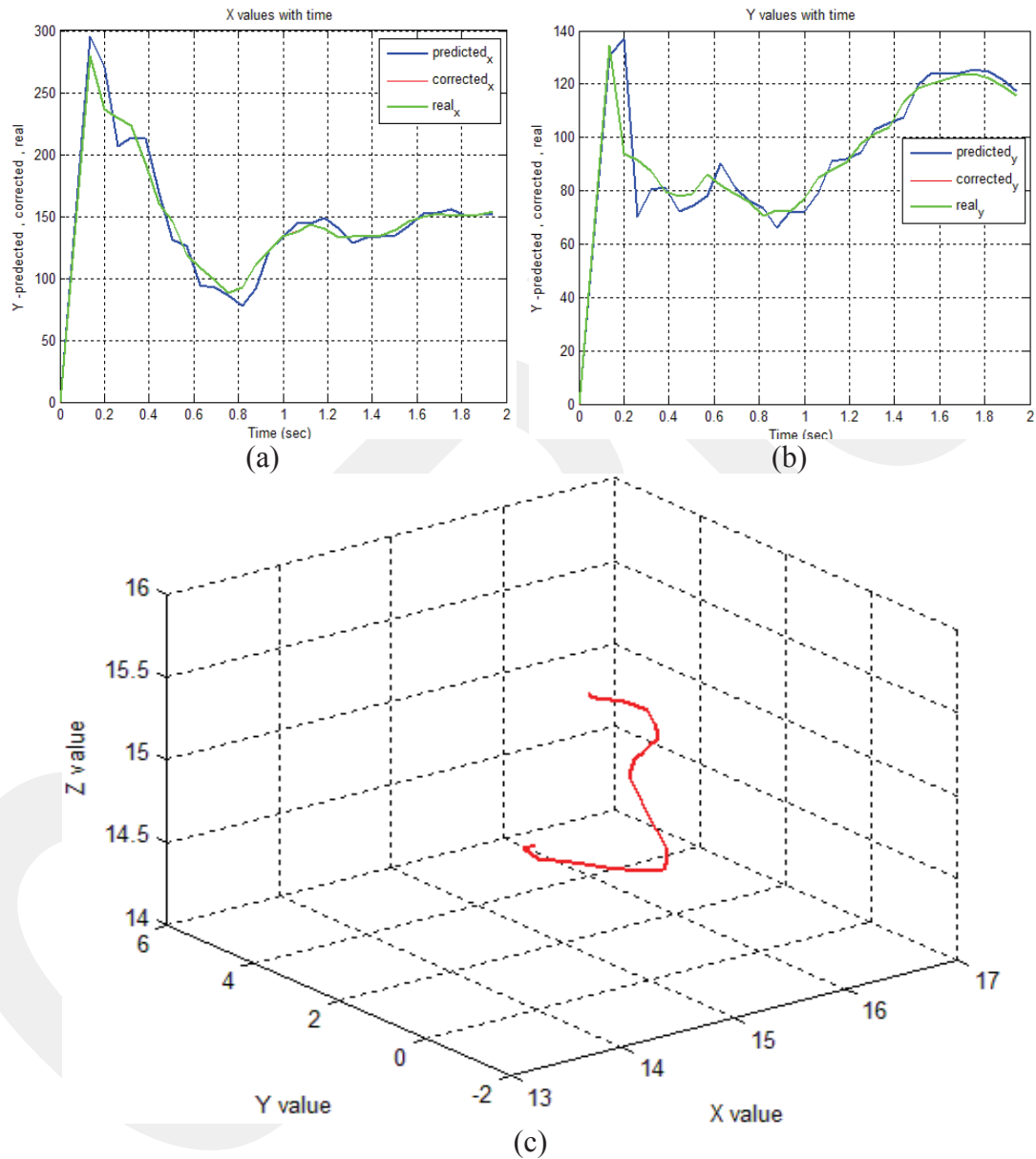
**Figure 30:** Tracking predicted, corrected, and measurement for 2 cm/sec moving object: (a) the x values (b) the y values. (c) 3D robot arm end-effector positions.

Fig. 31(a & b) shows the x-coordinate and y-coordinate in the case of speed equal to 3 cm/sec. The y-coordinate has more oscillation than x-coordinate, that due to the object moves in y-coordinate more than the motion in x-coordinate. Fig.31(c) shows the three dimensional trajectory of robot arm end-effector from the initial state of the robot arm until grasp the object.



**Figure 31:** Tracking predicted, corrected, and measurement for 3 cm/sec moving object: (a) the x values (b) the y values. (c) 3D robot arm end-effector positions.

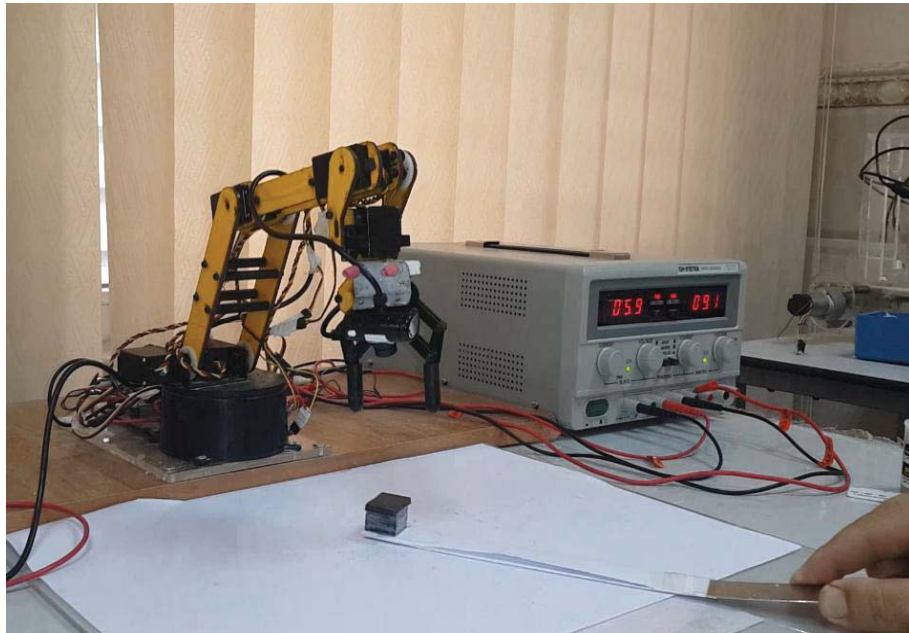
The fourth case of linearly moving object with velocity equal to 4 cm/Sec the x-coordinate and y-coordinate are shown in Fig. 32(a & b), notice that the operation take about 1.9 sec, the y-coordinate has more oscillation than other. Figure 32(c) shows the trajectory of robot arm end-effector from the initial state of the robot arm until grasp the object.



**Figure 32:** Tracking predicted, corrected, and measurement for 4 cm/sec moving object: (a) the x values (b) the y values. (c) 3D robot arm end-effector positions.

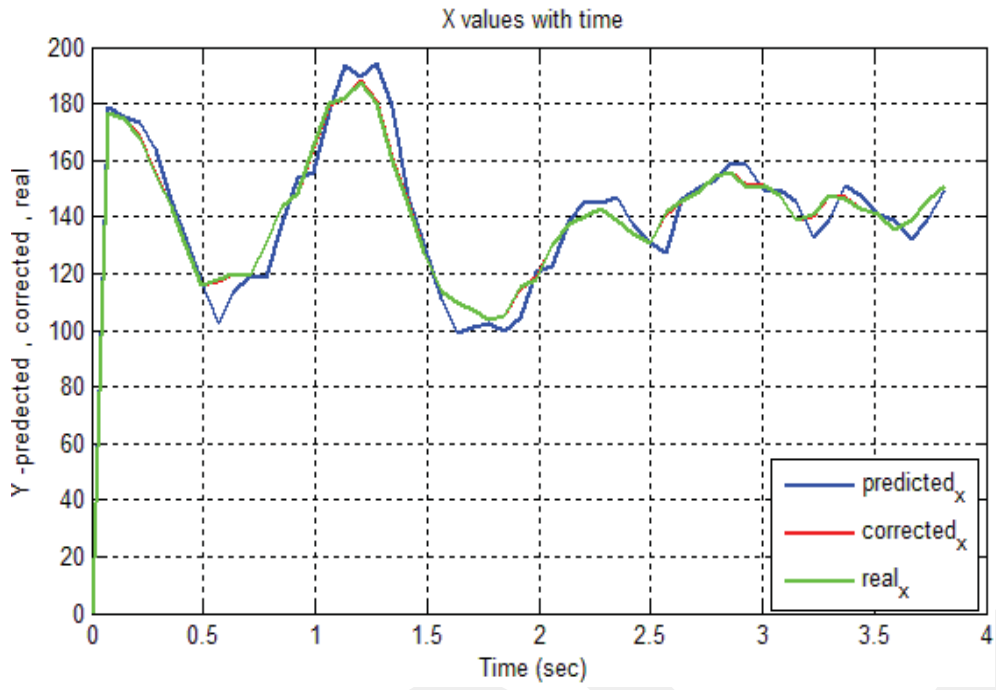
### 5.4.3. Tracking Moving Object with Random Trajectory

At this case, the object is moved randomly therefore this case is more complex than others. The motion is achieved by moving the object randomly by hand Fig 33. Shows how the object is moved in random manner.

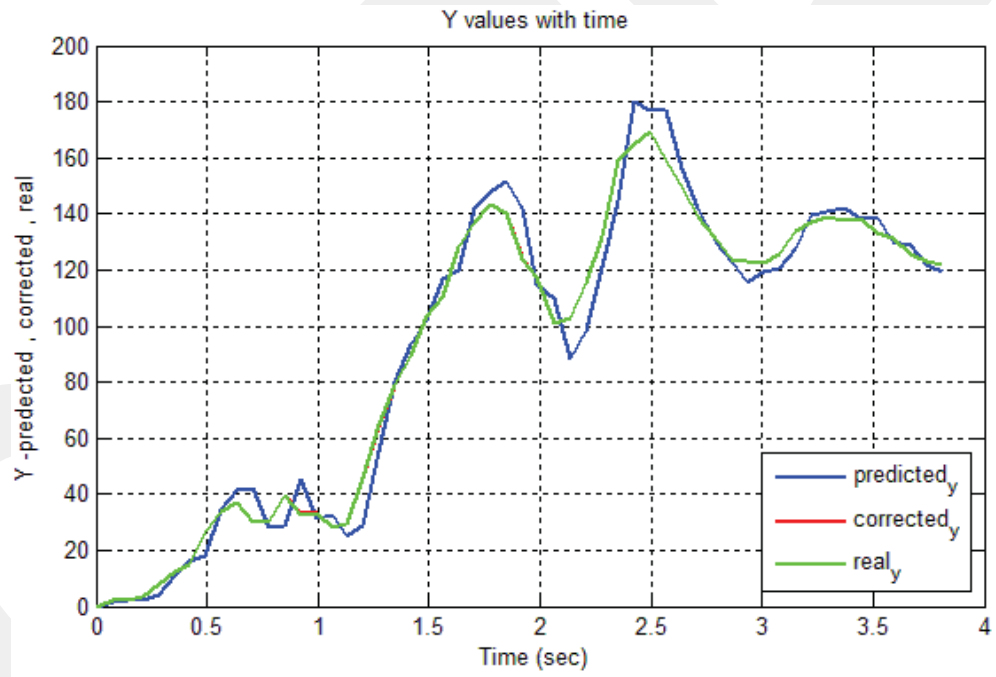


**Figure 33:** Moving object in random trajectory

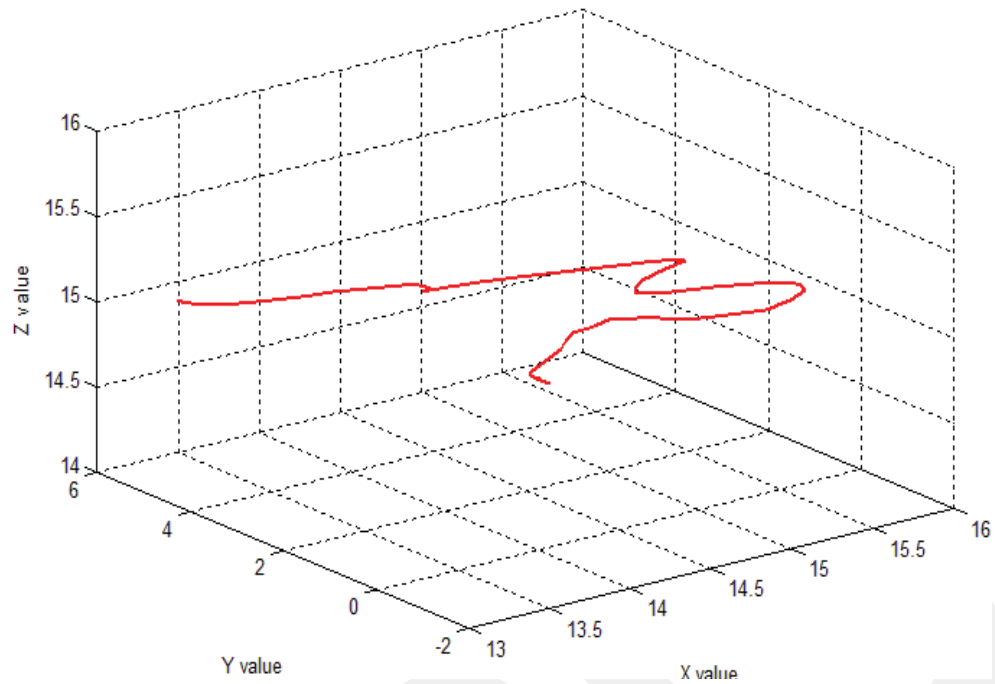
Figure. 34 (a) shows the x-coordinate curves of the tracking operation, as shown the oscillation is increased here because of the random motion. The y-coordinate of the track curves is shown in Fig. 34 (b). The three dimensional trajectory motion of the robot arm end effector is shown in Figure 34 (c).



(a)



(b)

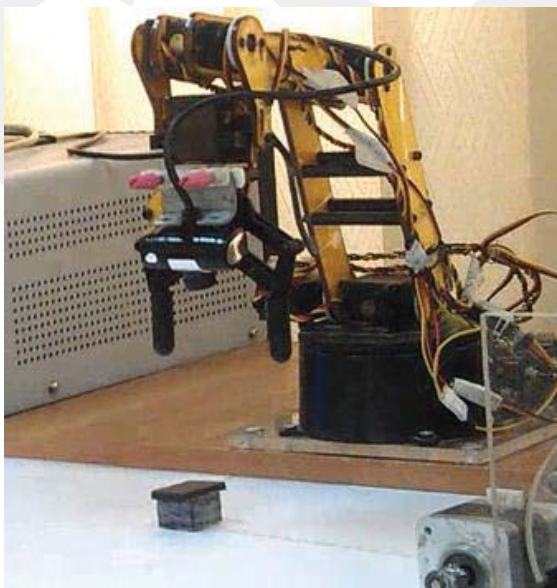


(c)

**Figure 34:** Tracking predicted, corrected, and measurement for 4 cm/sec moving object: (a) the x values (b) the y values. (c) 3D robot arm end-effector positions.

#### 5.4.4. Grasping Object Steps

The steps of tracking and grasping the object by the robot arm are shown in Fig. 35. In figure 35 (a), the object enters the scope of the robot arm camera, after that the arm starts tracking the object as shown in Figure 35 (b). In Figure 35 (c) the arm swoop the object, and in Figure 35 (d), the object has elevated by the arm.



(a)



(b)



(c)



(d)

**Figure 35:** Object tracking and grasping steps

### 5.5. Correction Error and Prediction Error

The correction error is computed by finding the difference between measured values and corrected values as the following equation:

$$\text{Correction Error} = \hat{x} - x \quad (5.1)$$

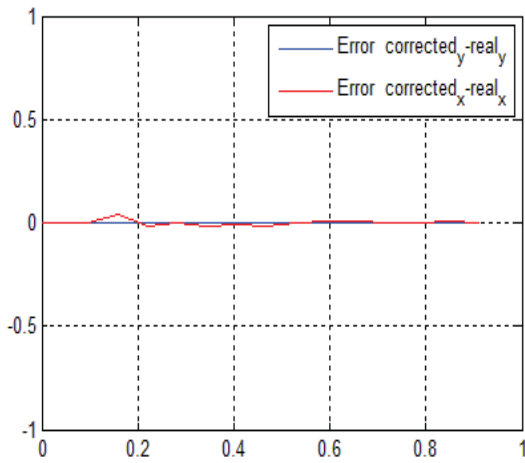
Also, the predicted is calculated by finding the difference between predicted values and real values as the following:

$$\text{Prediction Error} = x' - x \quad (5.2)$$

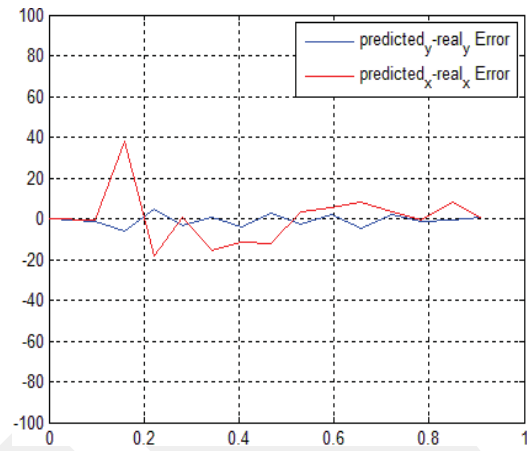
Where  $x$  is the real values,  $x'$  is the predicted values, and  $\hat{x}$  is the corrected values. Figure 36 shows the corrected and predicted error for all velocities and cases are applied. For all figures, the correction error is very small while the predicted error at first has oscillated but at the end, it is more stable.

The small correcting error in all cases refers to correcting error the corrected values is so close form real values. Also the prediction error has the max difference at first iterations but the oscillation is decreased in next iterations that refer to predicted values are enhanced and be better, x-axis represents the time in second and the y-axis

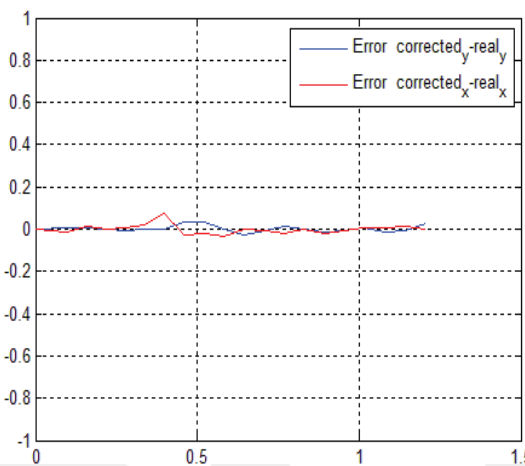
is the number of pixels at each curve which acts the difference (error) between the real value and the corrected or predicted value.



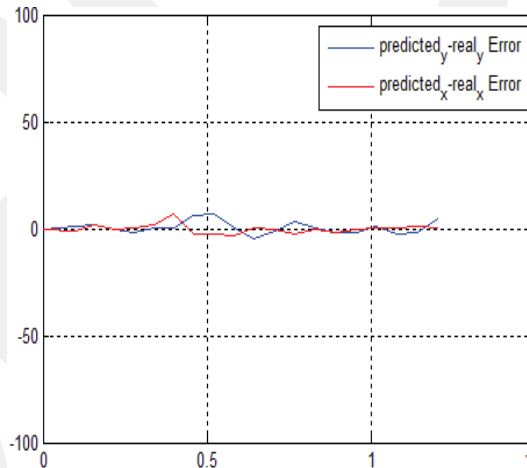
(a)



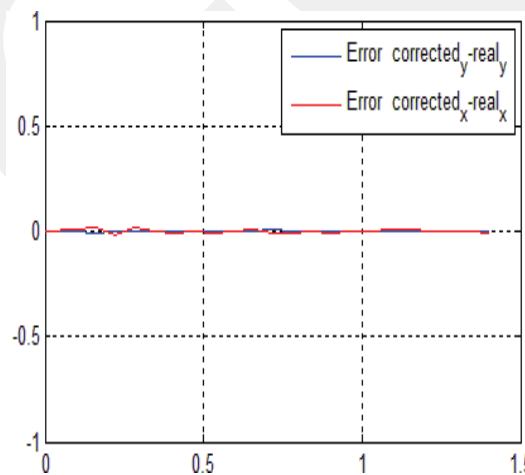
(b)



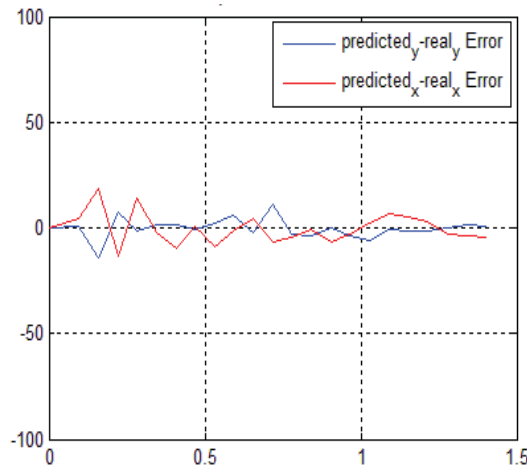
(c)



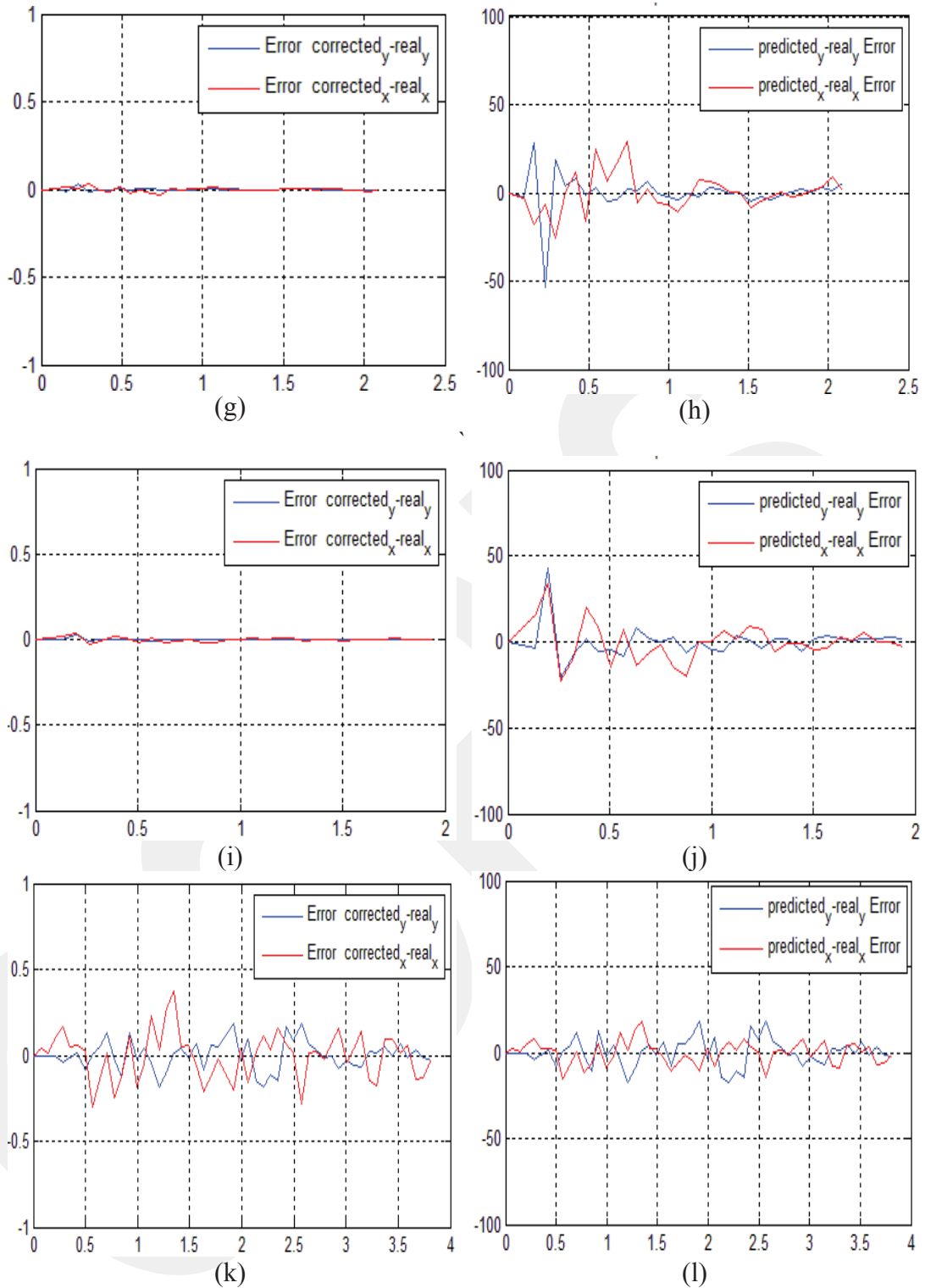
(d)



(e)



(f)



**Figure 36:** Correction/prediction error (a) correction error for stationary object (b) prediction error for stationary object (c) correction error for 1 cm/sec moving object (d) prediction error for 1 cm/sec moving object (e) correction error for 2 cm/sec moving object (f) prediction error for 2 cm/sec moving object (g) correction error for 3 cm/sec moving object (h) prediction error for 3 cm/sec moving object (i) correction error for 4 cm/sec moving object (j) prediction error for 4 cm/sec moving object (k) correction error for random moving object (l) prediction error for random moving object.

## 5.6. Comparative with another Work

Mohebbi A. introduced a method to achieve the tracking and the grasping of moving object through his work in master thesis [38], a six DOF DENSO 6242G robot arm is introduced (Fig. 37). The controller of the robot arm is an open architecture control model, two PCs are used, one of them is connected to the controller to control the robot and another PC is connected to the vision system to apply image processing.



**Figure 37:** DENSO robot arm.

The image processing PC is used to extract features of the image also to apply tracking algorithm, The Kalman filter is applied to track the object.

The image and tracking information transmitted to the PC that connected to the controller through the UDP network connection protocol.

Both monocular and stereo visual studio presented in this work, but we concern with single camera. A big difference between Lynx-6 robot arm and DENSO because the last one is larger, more stable, and more expensive. The object was moving in small speed and randomly. Our robot arm is cheaper, smaller, and less stable than DENSO. This work uses a Kalman filter to predict the trajectory of moving object that is moving randomly and in small speed Fig. 38 shows how the object is moved.



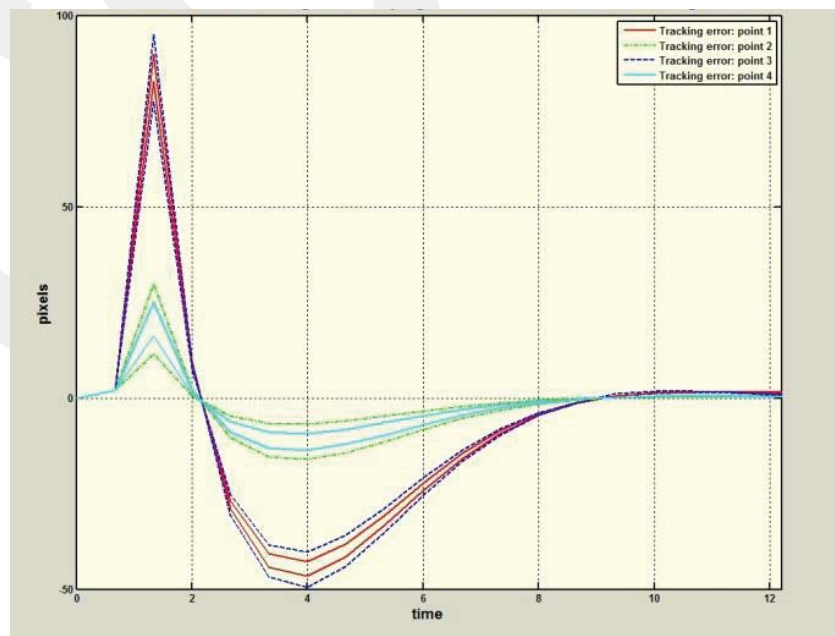
**Figure 38:** DENSO robot arm and moving object

The conditions of the two working space are similar, where the experiments are carried out in standard office environment, without any extra or special illuminations.

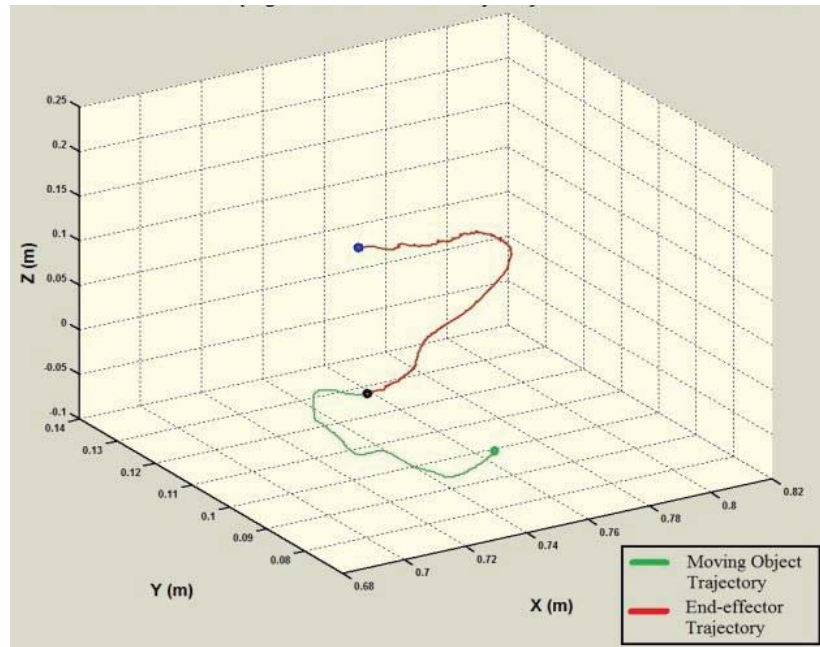
The experiments are designed and performed using Mathworks Matlab.

A comparison between this work and our work shows that our system is faster and has less overshoot, but the oscillation of ours is larger. Fig. 39 shows the tracking error of the tracking and grasping error the coverage time of this work spent 8.8 sec.

Fig. 40 shows the 3-D trajectory of the robot arm end effector.



**Figure 39:** Tracking error



**Figure 40:** 3D robot arm end-effector positions.

## CHAPTER 6

### CONCLUSION AND FUTURE WORK

#### 6.1. Conclusion

This thesis introduces an eye-in-hand robotic system for tracking and grasping the moving object in the real time.

The advantage of using an eye-in-hand camera configuration is that there is no need to calibrate the environment and measure the geometry of the framework. Since the cameras are mounted on five degree of freedom robot arm, the fields of the sight are limited and cannot cover the whole environment around the robot. Thus, the space for the target maneuvers is limited.

Kalman filter is used to estimate the trajectory of the moving object. Many moving object speeds applied and tested on the system. The tests show that the system works properly in both tracking and grasping the object.

The system is applied on a stationary object, 1 cm/sec, 2 cm/sec, 3 cm /sec, 4 cm /sec, for a linear moving object, also random direction is applied in order to show the difference between the moving ways.

The experiments lead us to reach a distinctive fact, which states that: The object speed is directly proportional with the time that will be taken for the grasping issues. That means, if the object speed is increased, the grasping time will be increased accordingly.

The using of partial windows is applied and the main advantage of this technique is to reduce the size of the image window that reduces the time cost of the image processing to detect the object, also to prevent the occlusion.

Finally and for more clarity, some limits of the used robot arm are taken into consideration such as the arm is not big enough. In addition, the robot arm has some oscillations, which affected directly on the accuracy of prediction.

The results show that the robot arm has the ability of tracking and grasping moving objects with a speed less than 5 cm/Sec. Due to the limitations of the speed of the robot joints 5-DOF, should not be the target at high speed. Thus, in some cases, such as flying object thrown, this method fails at this case.

## **6.2. Future Work**

The proposed method shows an efficient tracking and grasping moving object system. The limitation of the robot arm that has five DOF robot arm, the moving object should not have high speed, and fail at some cases for example flying thrown object. The using of eye in hand also has limited field of sight that cannot cover the whole environment around the robot.

As a future research works, the following advancements are suggested:

- To solve the drawback of eye-in-hand, both categories could use at the same time Eye-in-Hand and eye-to-hand that extends the vision system visibility constraint.
- Mobile robot arm gives the system more freedom in moving and the motion, extend the scope of the system.
- Using extended Kalman filter that is the nonlinear version of Kalman filter to improve the prediction of nonlinear moving object.

## REFERENCES

1. **Gans N. R., Hutchinson S. A., (2007)**, “*Stable Visual Servoing Through Hybrid Switched-System Control*”, IEEE Transactions on Robotics, vol. 23, no. 3, pp. 530-540.
2. **Lippiello V., Siciliano B., Villani L. , (2007)**, “*Position-Based Visual Servoing in Industrial Multirobot Cells Using A Hybrid Camera Configuration*”, IEEE Transactions on Robotics, vol. 23, no. 1, pp 73-86.
3. **Welch G., Bishop G., (2006)**, “*An Introduction to The Kalman Filter*”, University of North Carolina, pp. 1-16.
4. **Chen S., (2012)**, “*Kalman Filter for Robot Vision: A Survey*”, IEEE Transactions on Industrial Electronics, vol. 59, no. 11, pp. 4409-4420.
5. **Paul Z., Howard M., (2009)**, “*Fundamentals of Kalman Filtering: A Practical Approach*”, 3<sup>rd</sup> Edition, American Institute of Aeronautics and Astronautics, Arlington, Texas, pp. 131-142.
6. **Rafael C. G., (2007)**, “*Digital Image Processing*”, 3<sup>rd</sup> Edition, Prentice Hall, New Jersey, pp. 650-662.
7. **Rama B. M., (2010)**, “*A New Approach for Border Extraction Using Morphological Methods*”, International Journal of Engineering Science and Technology, vol. 2, no. 8, pp. 3832-3837.
8. **Qing L., Chengyu L., (2011)**, “*Edge Detection Based on Mathematical Morphology Theory*”, Image Analysis and Signal Processing (IASP), IEEE International Conference, pp. 151-154.
9. **Yuan H., (2011)**, “*Blind Forensics of Median Filtering in Digital Images*”, IEEE Transactions on Information Forensics and Security, vol. 6, no. 4, pp. 1335-1345.
10. **Siciliano B., Sciavicco L., Villani L., (2009)**, “*Robotics: Modeling, Planning and Control*”, IEEE Journals & Magazines, Robotics and Automation Magazine, vol. 16, no. 4. pp. 759-762.

11. **Baki K., Mehmet G., (2007)**, “*Software Development for The Kinematic Analysis of a Lynx 6 Robot Arm*”, World Academy of Science, Engineering and Technology, vol. 1, no. 6, pp.1578-1583.
12. **Craige J., (2005)**, “*Introduction to Robotics: Mechanics and Control*”, 3<sup>rd</sup> Edition, Prentice-Hall, New Jersey, pp. 68-85.
13. **Chun H. A., Khin T. L., (2008)**, “*Modeling Motion Control System for Motorized Robot Arm using Matlab*”, Proceedings of World Academy of Science, Engineering and Technology, vol. 32, pp. 351-354.
14. **Antonia Y., (2004)**, “*Heterogeneous Modeling & Design of Robot Arm Control System*”, University of Puerto Rico, Mayagüez, pp. 1-6.
15. **Peter K. A., Billibon Y., Aleksandar T., Paul M., (1993)** , “*Automated Tracking and Grasping of A Moving Object with A Robotic Hand-Eye System,*” IEEE Transactions on Robotics and Automation, vol. 9, no. 2 , pp. 152-165.
16. **Gonçalves P. J., Mendonça L. F., Sousa J. M., Caldas J. R., (2008)**, “*Uncalibrated Eye-to-Hand Visual Servoing Using Inverse Fuzzy Models*”, IEEE Transactions on Fuzzy Systems, vol. 16, no. 2, pp. 343-353.
17. **Pomares J., Perea I., Torres F., (2014)**, “*Dynamic Visual Servoing With Chaos Control for Redundant Robots*”, IEEE Transactions on Mechatronics, vol. 19, no. 2, pp. 423-431.
18. **Janabi S., Marey M., (2010)**, “*A Kalman-Filter-Based Method for Pose Estimation in Visual Servoing*”, IEEE Transactions on Robotics, vol. 26, no. 5, pp. 939-947.
19. **Kragic D., Miller A. T., Allen P. K., (2001)**, “*Real-Time Tracking Meets Online Grasp Planning*”, IEEE International Conference on Robotics and Automation, Seoul, Republic of Korea, vol. 3, pp. 2460-2465.
20. **Hidekazu S., Mamoru M., (2005)**, “*Visual Servoing to Catch Fish Using Global/Local GA Search,*”, Mechatronics, IEEE Transactions, vol. 10, no. 3, pp. 352-357.
21. **Vincenzo L., Fabio R., Bruno S., (2013)**, “*3D Monocular Robotic Ball*

- Catching*”, *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1615–1625.
22. **Comport A., Marchand E., Chaumette F., (2006)** “*Statistically Robust 2-D Visual Servoing*”, *IEEE Transactions on Robotics*, vol. 22, no. 2, pp 415-420.
  23. **Luca B., Paolo R., (2006)**, “*Two-Time Scale Visual Servoing of Eye-in-Hand Flexible Manipulators*”, *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 818-830.
  24. **Hesheng W., Yun L., Dongxiang Z., (2008)**, “*Adaptive Visual Servoing Using Point and Line Features with An Uncalibrated Eye-in-Hand Camera*”, *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 843–857.
  25. **Heng-Tze C., Zheng S., Pei Z., (2011)**, “*Imirok: Real-Time Imitative Robotic Arm Control for Home Robot Applications*”, *Pervasive Computing and Communications Workshops, IEEE International Conference*, pp. 360-363.
  26. **Graziano C., Koichi H., (2004)**, “*A Simple Technique for Improving Camera Displacement Estimation in Eye-in-Hand Visual Servoing*”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1239-1242.
  27. **Jorge P. and Fernando T., (2005)**, “*Movement-Flow-Based Visual Servoing and Force Control Fusion for Manipulation Tasks in Unstructured Environments*”, *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications And Reviews*, vol. 35, no. 1, pp. 4-15.
  28. **Changchun H., Yaoqing W., Xinping G., (2014)**, “*Visual Tracking Control for An Uncalibrated Robot System with Unknown Camera Parameters*”, *ELSEVIER, Robotics and Computer Integrated Manufacturing*, vol. 30, no. 1, pp. 19-24.
  29. **Gavin P., Stephen W., Dikai L., Gamini D., (2011)**, “*Autonomous Robot Manipulator-Based Exploration and Mapping System for Bridge Maintenance*”, *ELSEVIER, Robotics and Autonomous Systems*, vol. 59, no. 7, pp. 543–554.
  30. **Jacques A. G., Michel F., (2002)**, “*Visual Servoing of A 6-DOF Manipulator for Unknown 3-D Profile Following*”, *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 511-520.

31. **Janabi-Sharifi F., Lingfeng D., William J. W., (2011)**, “*Comparison of Basic Visual Servoing Methods*”, IEEE/ASME Transactions on Mechatronics, vol. 16, no. 5, pp. 967-983.
32. **Abdul M., Kouhei O., (2005)**, “*Eye-to-Hand Approach on Eye-in-Hand Configuration Within Real-Time Visual Servoing*”, Mechatronics, IEEE Transactions, vol. 10, no. 4, pp. 404-410.
33. **Shirai Y., Inoue H., (1973)**, “*Guiding A Robot By Visual Feedback in Assembling Tasks*”, IEEE Transactions in *Pattern Recognition*, vol. 5, no. 2, pp. 99-108.
34. **Center S., Agin G., (1979)**, “*Real Time Control of A Robot with A Mobile Camera*”, SRI International, 1979, vol.4 no. 2, pp.137-143.
35. **Indrazno S., Laxmidhar B., Martin T., Coleman S., (2014)**, “*Image-Based Visual Servoing of A 7-DOF Robot Manipulator Using An Adaptive Distributed Fuzzy PD Controller*”. IEEE Transactions on Mechatronics, vol. 19, no. 2, pp. 512-523.
36. **John C., (2011)**, “*The Image Processing Handbook*”, 6<sup>th</sup> Edition, CRC Press, North Carolina, pp. 24-30.
37. **Marsal R. C., (2011)**, “*Morphological And Statistical Analysis Of Biomaterials with Applications in Tissue Engineering by Means of Microscopy Image Processing*”, IEEE Latin America Transactions, vol. 9, no. 3, pp. 399-407.
38. **Abolfazl M., (2013)**, “*Real-Time Stereo Visual Servoing of A 6-DOF Robot for Tracking and Grasping Moving Objects*”, M.Sc. Thesis, Department of Mechanical and Industrial Engineering, Concordia University, Canada, pp.7-14.

## APPENDICES A

### A. CURRICULUM VITAE

#### PERSONAL INFORMATION

Surname, Name: YOSIF, Zead Mohammed

Nationality: Iraqi

Date and Place of Birth: 11 April 1984, Mosul, Iraq

Material Status: Single

Phone: +90 535 971 8046, + 964 770 386 9770

Fax:

Email: [zeyadmyousif@gmail.com](mailto:zeyadmyousif@gmail.com)

#### EDUCATION

Degree	Institution	Year of Graduation
B.Sc.	Mosul Univ. College of Engineering. Computer Engineering Dept.	2006
High School	Al-Sharkia for Boys	2002

#### WORK EXPERIENCE

Year	Place	Enrollment
2007-Present	Mosul Univ. College of Engineering. Mechatronics Eng. Dept.	Laboratories Instructors

## **FOREIGN LANGUAGES**

Arabic (mother language), English.

## **PUBLICATION**

**Elbasi E., Abdilatif M. A., YOSIF Z. M., (2013),** *"Data Mining Based Analysis of Breast Cancer Risks in Iraq"*, the 6<sup>th</sup> engineering conference, Cankaya University, Ankara, Turkey, pp. 227-231.

## **HOBBIES**

Reading, Travelling.