



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Thesis for the Degree of Master of Engineering

**Object Following Control of a Six-legged
Walking Robot Using Kinect Camera
Sensor**

by

Amruta Vinod Gulalkari

**Department of Interdisciplinary Program of
Mechatronics Engineering,
The Graduate School,
Pukyong National University**

August 2015

**Object Following Control of a Six-legged
Walking Robot Using Kinect Camera
Sensor**

**키넥트 카메라센서를 사용한
육족보행로봇의 물체 추종제어**

by

Amruta Vinod Gulalkari

Advisor: Professor Sang Bong Kim

**A thesis submitted in partial fulfillment of the requirements for
the degree of Master of Engineering**

**In the Department of Interdisciplinary Program of
Mechatronics Engineering,
The Graduate School,
Pukyong National University**

August 2015



Object Following Control of a Six-legged Walking Robot Using Kinect Camera Sensor

A thesis

by

Amruta Vinod Gulalkari

Approved as to styles and contents by:


(Chairman) **Gi Sik Byun**
(Member) **Yeon Wook Choe**
(Member) **Sang Bong Kim**

August 21, 2015

Acknowledgements

‘No Herculean task can be consummated without the support and contribution of a number of individuals and that is the very first essence of success of any colossal work.’ Here I take the opportunity to express my gratitude toward all those who helped me directly and indirectly to complete my studies in Korea.

First of all, I would like to give my highest praise to God ‘Jinendra Bhagwan’ for all helps and guidance that have been given such that I could finish all my research and thesis in Master program very well. I would like to express my deepest gratitude toward my *late* father Mr. Vinod S. Gulalkari who had always encouraged me and blessed me during his entire life. All achievement and success that have been granted to me, I would like to return to him.

Most importantly, I would like to express my special and sincere gratitude to my professor, Professor Sang Bong Kim, who has helped me significantly in completing my Master program. I really appreciate his excellent guidance, encouragement, support and kindness that he has shown to me during my study in his magnificent laboratory.

I would like to thank the members of my thesis committee, Professor Gi Sik Byun and Professor Yeon Wook Choe, for their helpful feedback and great suggestions for better contribution of my thesis.

I would like to express my profound gratitude to Professor Hak Kyeong Kim for his valuable great helps and advice during my studies to finish my research and master thesis in CIMEC Lab. very well.

I would like to thank all members of Computer Integrated Manufacturing Electronics Commerce (CIMEC) laboratory for their cooperation, kindness and friendship, Dr. Phuc Thanh Doan, Dr. Bui Thanh Luan, Dr. Hoang Giang, Mr. Pandu Sandi Pratama, Mr. Yuhanes Dedy Setiawan, Mr. Van Tu Duong, Mr. Nguyen Tron Hai, Mr. Quang Tuyen Tran, Mr. Min Jung Hu, Mr. Jin Wook Kim, Mr. Husam Hasan Khudhir El-dulaimi, Mr. Ding Xingkun, Mr. Jeong Jae Hoon, Mr. Sheng Dongbo and Mr. Hung Nguyen Huy. I would also like to express my genuine gratitude to CIMEC Laboratory secretary Mrs. Kim Hyeon for her love, care and kindness to me during my living in Korea.

Last but not least, I would like to express my deepest gratitude to my mother, Rita V. Gulalkari, my brother, Rishabh V. Gulalkari, my aunt Shubha S. Gulalkari, my grandmother Kamal S. Gulalkari, my brother-in-law Yogiraj Bakale, my friend and wellwisher Vishal Raina and all my relatives and friends for their endless love, prayers, encouragement and support not only in completing Master program but also in the whole of my life.

Pukyong National University, Busan, Korea

May 28, 2015



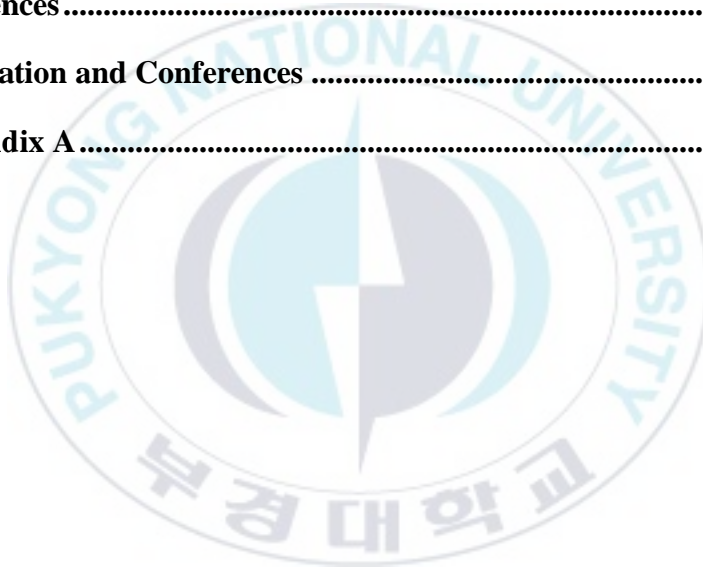
Amruta Vinod Gulalkari

Contents

Acknowledgement	i
Contents	iii
Abstract.....	vi
List of Figures.....	ix
List of Tables	xii
Nomenclature	xiii
Chapter 1: Introduction	1
1.1 Background and motivation.....	1
1.2 Objective and research method.....	5
1.3 Outline of thesis and summary of contributions	8
Chapter 2: System Description.....	12
2.1 Mechanical design.....	12
2.2 Electrical design	14
2.2.1 Kinect camera sensor	15
2.2.2 Microcontroller	21
2.2.3 Servomotors	22
2.2.4 Bluetooth.....	25
2.2.5 Power supply.....	26
2.3 Graphical Unit Interface (GUI) development	27
2.4 Basic terminologies of the 6LR.....	29
2.4.1 Typical definitions for legged robot	29

2.4.2 Walking gait.....	33
Chapter 3: System Modeling.....	36
3.1 Kinematic modeling of one leg with four joints	36
3.2 Kinematics of the six leg robot	40
3.3 Kinematic modeling to define control inputs	44
3.4 Kinematic modeling of the 6LR with a candidate object.....	46
Chapter 4: Image Processing for Object Detection and Kalman Filter for Object Tracking	49
4.1 Color-based object detection algorithm	49
4.2 Kalman filter algorithm for moving object tracking	54
Chapter 5: Controller Design based on Backstepping control method	59
5.1 Walking control	59
5.1.1 End effector position control of one leg	60
5.1.2 Translational and rotational body movement	65
5.2 Object following control	70
5.2.1 Heading angle error.....	70
5.2.2 Distance error.....	73
5.2.3 Object following movement of the 6LR and a candidate object movement	74
5.2.3.1 Rotational following of the 6LR (without straight moving)	74
5.2.3.2 Straight following of the 6LR (without rotation).77	

5.2.4 Angular and translational velocity controller design ...	78
Chapter 6: Simulation and Experimental Results	81
6.1 Rotational following of the 6LR (without straight moving) ..	81
6.2 Straight following of the 6LR (without rotation)	89
Chapter 7: Conclusions and Future Works	97
7.1 Conclusions	97
7.2 Future works.....	99
References	101
Publication and Conferences	108
Appendix A	110



Object Following Control for a Six-legged Walking Robot Using Kinect Camera Sensor

Amruta Vinod Gulalkari

**Department of Interdisciplinary Program of Mechatronics
Engineering, The Graduate School,
Pukyong National University**

Abstract

The development of object following systems using camera sensors has been a topic being researched and reported widely in the field of autonomous robotics. The problems in the development of such systems are choosing robot, selecting camera for object detection, maintaining stability of the robot and designing appropriate control law to control the robotic system.

To deal with these problems, this thesis proposes a new type of object following algorithm applied to the six-legged walking robot (6LR) using a Kinect camera sensor based on a Kalman filter and a backstepping control method. To do this task, the following steps are executed. Firstly, a real 6LR with four joint legs is developed with several interconnected devices such as servomotors, Micro Control Unit (MCU), Bluetooth, etc. The Kinect camera sensor is installed on the 6LR platform for image processing. In this thesis, a candidate blue colored object is used as an object for detection and following. For the 6LR, the tripod walking gait is chosen to control six legs in

this thesis. The walking cycle of tripod gait is presented with swing phase of 3 legs and retract phase of other 3 legs. A graphical user interface (GUI) is developed using Visual studio C# software to send control commands to the robot system and display its results. Secondly, a system modeling is described to understand the behavior of the 6LR system as follows. First, a forward kinematic modeling of one leg with four joints and four links is presented. The Denavit-Hartenberg (DH) convention is adopted to define the modeling parameters which allow the forward kinematics function by composition of the individual coordinate transformation. Second, a kinematics of 6LR is presented to derive the homogeneous transformation matrices from body coordinate frame to each leg coordinate frame. Third, kinematic modeling of the 6LR with a candidate object is described. Fourth, kinematic modeling of 6LR to define control inputs such as translational velocity and angular velocity for object following control is presented. Thirdly, a color-based object detection algorithm is applied to the RGB images obtained by the Kinect camera sensor to detect a blue colored candidate object and to find its center coordinate in the RGB image frames. Using simple trigonometry and Kinect depth data, the pixel position coordinate is converted to the real world coordinate in mm. Then, a Kalman filter algorithm based on an object motion model is used to track the real world position and velocity of the moving candidate object. Fourthly, based on the system modeling, controllers for the 6LR are designed using backstepping method and Lyapunov stability criterion. The walking control of 6LR is designed using tripod walking gait and differential kinematics algorithm instead of inverse kinematics, and then the object following controller is

designed to make the 6LR follow the candidate object. Finally, simulations and experiments are conducted to verify the effectiveness and performance of the proposed controller for following the moving candidate object. The results show that the proposed controller makes the six-legged robot follow the candidate object very well using color-based object detection algorithm and Kalman filter tracking algorithm.

Keywords: Six-legged robot, Kinect camera sensor, DH Convention, Kalman filter, Backstepping control, Lyapunov function, Object following.



List of Figures

Fig. 2.1 Configuration of Kinect camera based 6LR system.....	13
Fig. 2.2 Real Kinect camera based 6LR system	13
Fig. 2.3 One leg configuration	14
Fig. 2.4 Electrical configuration of Kinect camera based 6LR system	15
Fig. 2.5 Kinect Xbox camera sensor	16
Fig. 2.6 Stereo-triangulation of Kinect camera sensor to obtain depth data	17
Fig. 2.7 Illustration of depth calculation of a point R on an object...	18
Fig. 2.8 Working sequence of Kinect camera sensor to obtain depth data	20
Fig. 2.9 DSP TMS320F28335	21
Fig. 2.10 Dynamixel AX-12A	22
Fig. 2.11 Half duplex UART circuit	23
Fig. 2.12 Touch detection method	24
Fig. 2.13 Bluetooth Promi SD202	25
Fig. 2.14 12V battery ATLASBX ITX100	26
Fig. 2.15 GUI of the proposed system	28
Fig. 2.16 Leg cycle of one leg.....	29
Fig. 2.17 Support polygon of the 6LR	30
Fig. 2.18 Front end rear margins.....	31
Fig. 2.19 Statically unstable gait.....	32
Fig. 2.20 Workspace of one leg	33
Fig. 2.21 Walking cycle of the tripod gait	34
Fig. 2.22 Swing phase of step cycle of leg i	34
Fig. 2.23 Retract phase of step cycle of leg i	35

Fig. 3.1 Configuration of one leg.....	37
Fig. 3.2 Frame coordinates of the 6LR	40
Fig. 3.3 Kinematic configuration of the 6LR.....	45
Fig. 3.4 Kinematic configuration of the 6LR following a candidate object	46
Fig. 4.1 Kinect RGB image of the detected candidate object with its center coordinate within a desired object area.....	51
Fig. 4.2 Extended view of the current angle from the Kinect camera in the RGB image frame and the real world frame.....	52
Fig. 4.3 Desired (reference) depth for the moving candidate object	53
Fig. 4.4 Flow chart of the color-based object detection algorithm ...	54
Fig. 5.1 End effector position control of leg i	60
Fig. 5.2 Block diagram of walking controller.....	65
Fig. 5.3 Translational and rotational body movement of the 6LR....	65
Fig. 5.4 End effector trajectory of one leg of the 6LR.....	67
Fig. 5.5 Swing phase of leg i	68
Fig. 5.6 Retract phase of leg i	68
Fig. 5.7 Each state time in one cycle time	68
Fig. 5.8 Kinect RGB image showing detected candidate object outside the desired object area	71
Fig. 5.9 Heading angle error definition.....	72
Fig. 5.10 Distance error definition.....	74
Fig. 5.11 6LR and candidate object movement in rotational following.....	76
Fig. 5.12 6LR and candidate object movement in straight following.	78
Fig. 5.13 Block diagram of the proposed system.....	80

Fig. 6.1 Schematic of the initial positions of the 6LR and a candidate object in rotational following case.....	83
Fig. 6.2 Estimated angular velocity of the 6LR	85
Fig. 6.3 Estimated heading angle error of the 6LR.....	85
Fig. 6.4 Kalman filter results of a moving candidate object during rotational following case of the 6LR	87
Fig. 6.5 Images showing sequence of events occurred during rotational following case of the 6LR and a candidate object	89
Fig. 6.6 Schematic of the initial positions of the 6LR and a candidate object in straight following case	90
Fig. 6.7 Positions of the 6LR and a candidate object in the global coordinate during straight following case of the 6LR	92
Fig. 6.8 Estimated translational velocity of the 6LR	93
Fig. 6.9 Estimated distance error of the 6LR	93
Fig. 6.10 Kalman filter results of a moving candidate object during straight following case of the 6LR.....	95
Fig. 6.11 Images showing sequence of events occurred during straight following case of the 6LR and a candidate object ..	95
Fig. A.1 Denavit-Hartenberg kinematic parameters	110

List of Tables

Table 2.1 Specification of Kinect Xbox camera sensor.....	21
Table 2.2 Specification of Dynamixel AX-12A actuator	22
Table 2.3 Specification of Bluetooth Promi SD202	25
Table 2.4 Specification of 12V battery ATLASBX ITX100.....	26
Table 3.1 D-H parameters of 4-Link leg.....	38
Table 6.1 Parameters and initial values of the Kalman filter and color-based object detection algorithm	81
Table 6.2 Parameters and initial values of the rotational following simulation.....	82
Table 6.3 Parameters and initial values of the straight following simulation	89

Nomenclatures

j_kT_i	Transformation matrix from joint j coordinate system to joint k coordinate system of leg i
\mathbf{p}_{ei}^0	The end effector position vector of leg i with respect to joint 0 frame
$\mathbf{k}(\boldsymbol{\theta}_i)$	Forward kinematic function
$\boldsymbol{\theta}_i$	Angular position vector of each joint
$O_b x_b y_b z_b$	Body coordinate frame
$O_{ji} x_{ji} y_{ji} z_{ji}$	Leg coordinate frame
$O_G X_G Y_G$	Global coordinate frame
$O_k XY$	Local coordinate frame of the Kinect camera sensor
α_{0i}	Angle to transform from $O_b x_b y_b z_b$ coordinate frame to the $O_{ji} x_{ji} y_{ji} z_{ji}$ coordinate frame of joint 0 of leg i
$x_{0i}^{x_b}, y_{0i}^{y_b}$	Translational distances to transform from $O_b x_b y_b z_b$ coordinate frame to the $O_{ji} x_{ji} y_{ji} z_{ji}$ coordinate frame of joint 0 of leg i along x_b axis and y_b axis, respectively
W	Distance between Leg-L2 and Leg-R2
L	Length of the robot
E	Distance between Leg-L1 and Leg-R1
\mathbf{p}_{ei}^b	Position vector of the end effector with respect to body frame of leg i
\mathbf{p}_{di}^b	Desired end effector position vector with respect to body

	frame of leg i
v	Translational velocity of 6LR along y_b axis
ω	Angular velocity of 6LR
r	Step distance
T	Cycle time
θ_d	Desired angle required to achieve AEP
d	Distance between center of the 6LR and the end effector AEP
Δt	Sampling time
\mathbf{x}_k	State vector at time step k
\mathbf{F}	State transition matrix
k	Discrete time step
\mathbf{w}_k	Gaussian process noise at time step k
\mathbf{Q}	State covariance matrix
\mathbf{z}_k	Measurement vector at time step k
\mathbf{H}	Measurement matrix
\mathbf{v}_k	Gaussian measurement noise at time step k
\mathbf{R}	Measurement covariance matrix
X_{RGB}, Y_{RGB}	Pixel position coordinate of the object inside the RGB image frames (pixels)
X_R, Y_R	Real world object position coordinate in X and Y direction (mm), respectively
\dot{X}_R, \dot{Y}_R	Real world object velocity coordinate in X and Y direction (mm), respectively
\hat{X}_R, \hat{Y}_R	Estimated real world object position coordinate in X and Y

	direction (mm), respectively
ϕ_C	Current angle between the center of the detected object and the center of RGB image
α	Coefficient of transformation from pixel to length
f	Focal length of the Kinect camera
e_ϕ, e_y	Heading angle error and distance error, respectively
$\phi_D, \phi_C, \hat{\phi}$	Desired, current and estimated angular positions of the candidate object, respectively
Y_D, Y_R, \hat{Y}_R	Desired, current and estimated distance of the candidate object, respectively
$V_i(\mathbf{e}_i^b)$	Candidate Lyapunov function of leg i
$\mathbf{J}_{Ai}^b(\boldsymbol{\theta}_i)$	Analytical Jacobian of \mathbf{p}_{ei}^b with respect to $\boldsymbol{\theta}_i$
\mathbf{K}	Symmetric positive definite matrix
$(\mathbf{J}_{Ai}^b)^+$	Right pseudo-inverse of \mathbf{J}_{Ai}^b leg i
$\boldsymbol{\theta}_i^{t_k}$	Joint angle of leg i at time t_k
ω_D, v_D	Desired angular velocity and desired translational velocity of the 6LR
$\hat{\omega}, \hat{v}$	Estimated angular velocity and estimated translational velocity of the 6LR
V	Candidate Lyapunov function for angular and translational controller design
K_1, K_2	Positive constants
X_C, Y_C	Current position coordinate of the 6LR in global coordinate frame
X_o, Y_o	Current position coordinate of the candidate object in

	global coordinate frame
e_x, e_y	Error coordinate in global coordinate frame
X_R, Y_R	Current position coordinate of the candidate object in body coordinate frame
θ	Current angular position of the 6LR in global coordinate measured from X_G axis
D	Distance between the Kinect camera sensor and the center of the 6LR body



Chapter 1: Introduction

1.1 Background and motivation

The problem of object following has been studied by many researchers in robotic vision literature. This is a field being researched and developed extensively with multiple techniques and distinct robots. The camera based object following robots has numerous applications in the field of disaster relief operations, navigation, defense, surveillance, etc. Thus, this interesting topic needs more analysis.

Many developed camera-based object following systems used legged mobile robots such as bipedal (two legged) [1-6], and quadruped (four legged) [7-14] robotic platforms. However, maintaining stability with such legged configurations is difficult. The bipedal robots can only be statically stable within some limits. For the better stability of bipedal robots, each leg must have sufficient capacity to support the full weight of the robot. In case of the quadruped robots, the robot is passively stable when it is stationary. However, walking control remains challenging because the robot's center of gravity must be actively shifted to maintain its balance during walking. Therefore, legged mobile robots with more stable configuration are needed.

Camera selection is also an important problem for object detection and following systems. Many vision-based systems used stereo cameras [15-18], pan-tilt-zoom (PTZ) camera [19-21], omni-directional camera [22, 23], web camera [24] and wireless camera

[25]. The disadvantages of stereo cameras must use stereo vision technique which requires more hardware because stereo-vision uses at least two cameras and have its corresponding matching issues. Therefore, the stereo vision technique becomes quite expensive technique. In addition, due to poor quality images and high storage cost, PTZ, omni-directional and web cameras are not preferable. Moreover, web cameras can't work without internet connection availability. The other issue in object detection systems is the requirement of depth data (i.e., the distance between the object and the robot) to obtain an exact position of the candidate object. The above mentioned cameras can't provide depth data directly. Therefore, this requirement may lead to complex image processing techniques and thus increases the computational time of the system. Therefore, to deal with these problems, a new type of RGB-D camera sensor is needed to perform image processing for object detection.

In object following systems, one very significant perception task is to track the positions of a moving object using selected camera. The goal of object tracking is to find an object's location in consecutive video frames. Many different algorithms have been proposed for object tracking such as mean-shift tracking [26], optical flow [27, 28], feature matching [29], particle filter [30], camshaft [31], SIFT [32, 33] and SURF [34, 35], etc. Each algorithm has strengths in certain environments and weaknesses in others. The tracking algorithms used in [26-31] worked well with non-varying light conditions, whereas these algorithms lost control over tracking of the candidate object with non-rigid object motion and rapid appearance changes caused by image noise. The crucial drawback of algorithms used in [32-35] was the high computational time required

to create interest points and match feature vectors. Therefore, these algorithms are incapable of producing real-time results for object tracking. Thus, an efficient, reliable and robust tracking method is needed.

For a multi-leg robot to follow the moving object keeping robot's stability is a very challenging task. This is because the position of each leg has to be adjusted with particular formation such that the robot can follow the direction of the candidate object. Thus, an appropriate controller is crucially required for such systems.

There are many research works which have developed controllers for the legged robots to perform an object following task. I. A. Sulistijono [36] proposed a fuzzy logic control scheme for a humanoid soccer robot EFuRIO to search a ball using vision sensor, follow the detected ball and then kick that ball toward an opponent goal. This fuzzy logic controller was applied to find the appropriate joint angles from the vision system to generate the appropriate reaction of the robot body. The simulation and experimental results have showed the validity of the controller. This fuzzy logic based humanoid robot had searched the ball, walked toward the detected ball by achieving desired turning angle and speed smoothly, and finally kicked the ball to the goal. A. Sanz [37] developed a hierarchical fuzzy control algorithm for a six-legged walking robot called PILAR to follow an object. This controller has controlled the linear and angular velocities of each leg well. However, in fuzzy logic-based controllers, selection of membership functions is not an easy task. Membership function must be carefully chosen to achieve maximum performance, and its selection requires much efforts.

Instead, a non-fuzzy logic controller can be designed much more quickly with similar performance.

A. A. Saputra et al. [38] has proposed a combination control algorithm of PID and PD controller for a humanoid robot EROS to control its posture and angular velocity, respectively. In this ball following system of the biped robot, a posture control and an angular velocity control are the essential requirements of the system. In this robot, a PID controller was used to control robot's posture, whereas a PD controller was used to control robot's angular velocity to fulfill the desired requirement. However, PD and PID controllers are linear and can't perform well with the nonlinear multi-legged systems. Moreover, systems using PD and PID controller do not require system modeling, but can't satisfy every multi-legged system requirement.

M. Liem et al. [39] proposed a hybrid control algorithm for the quadruped robot named Sony AIBO robot dog to follow a person. This system used a hybrid algorithm that made the small robot possible to follow a person. This robot was able to perceive its environment using a small threshold camera mounted in its nose. The hybrid control algorithm mainly consisted of two stages: The first stage is tracking a person using a combination of number of tracking algorithms such as Gaussian mixture model (GMM) background subtraction method, Expectation Maximization (EM) color histogram and Kanade-Lucas-Tomasi (KLT) tracker. The second stage is controlling the heading angle and body of the robot depending on the position of the tracked person and then following the tracked person. However, the efficiency of this algorithm purely depended on the

result of the mixed tracking algorithm which can be poor with the increased shaky movements of the robot during its walking. Due to the mixture of many algorithms, the computational cost of this control algorithm became high. To solve the problems mentioned above, a new control system is needed to follow an object.

1.2 Objective and research method

To solve the problems mentioned in the above discussion, this thesis proposes a vision-based object following system for a six-legged walking robot (6LR) using a Kinect camera sensor based on a Kalman filter and a backstepping control method. To do this task, a system design and modeling, an image processing algorithm, a tracking algorithm, a control algorithm design, and simulation and experiment to evaluate the controller are presented as follows.

Firstly, a 6LR is developed for the experimental purpose. The 6LR has its body and six legs. The 6LR can walk statically by supporting at the most five legs. Thus, with the bigger support polygon, the 6LR becomes more stable than biped or quadruped legged robots during its walking. Therefore, a six-legged configuration has been chosen for this thesis. Each leg of the developed 6LR has four links, four rotational joints and an end effector which can make the robot achieve more complex postures. Each leg uses four servo motors to control the locomotion of leg. A Kinect camera sensor is installed on the top front of the robot for image processing. This low cost RGB-D camera is able to provide depth data directly and thus, reduces the complexity of the image processing algorithm. The robot is controlled by the computer and the

microcontroller. The microcontroller used to control the robot is DSP (Digital Signal Processor) TMS320F28335. The computer and microcontroller are connected by Bluetooth through RS232 interface. Microcontroller directly controls the servo motors, read encoders and receives feedback data from servo motors through half duplex community. The computer calculates the image processing algorithm and the control algorithm, and sends the joint angles signals to the microcontroller. A Graphical User Interface (GUI) is designed using Visual studio C# software to send control commands to the 6LR and to display the results to users.

Secondly, a system kinematic modeling is described to understand the behavior of the 6LR system as follows. In the first step, a kinematic modeling of one leg with four joints and four links is presented. The Denavit-Hartenberg (DH) convention is adopted to show the relation between joint angles and the end effector position to control the walking of each leg. In the second step, a kinematics of the 6LR is presented to derive the homogeneous transformation matrices from body coordinate frame to each leg coordinate frame. In the third step, a kinematic modeling of the 6LR with a candidate object is described. Finally, a kinematic modeling of the 6LR is presented to define its control inputs such as translational velocity and angular velocity for the object following control. In this thesis, a tripod gait is chosen for the walking of 6LR. It means that one tripod of legs (the foreleg and hind leg from one side and the middle leg from the other side) is in the swing phase (3 legs: lift up) and the other tripod of legs is in the retract phase (other 3 legs: contact with the ground) when the robot walks. The tripods are alternated every cycle.

Thirdly, a color-based object detection algorithm is developed using AForge.NET C# framework to detect the position coordinate of a blue colored candidate object. This image processing algorithm is applied to RGB image frames obtained from the Kinect camera sensor. Primarily, the pixel coordinate of the center of the candidate object inside the RGB image frames is obtained. Then, using simple trigonometry and Kinect depth information, a real world coordinate of the candidate object is obtained in mm. In this thesis, a Kalman filter algorithm based on the object motion model is adopted to estimate the position and velocity of the moving candidate object. The Kalman filter is a recursive estimator that estimates the state from the previous time step and the current measurement efficiently.

Fourthly, controllers for the 6LR are designed based on the backstepping method using Lyapunov function. The controller using backstepping method is simple for the non-linear system as the control law can be obtained directly during stability checking of the system. In the controller design, firstly, a walking controller design of the 6LR is introduced using tripod walking gait and differential kinematic algorithm. Secondly, the object following movement of the 6LR and a candidate object movement is described in two separate cases: rotational following of the 6LR and straight following of the 6LR. Finally, an object following controller of the 6LR is designed to make the 6LR follow the candidate object in these two cases using the color-based object detection algorithm and the Kalman filter tracking algorithm.

Finally, simulations and experiments are conducted to verify the effectiveness and performance of the proposed control algorithm

for following the moving candidate object in both cases: rotational following and straight following of the 6LR. The results show that the proposed controller makes the 6LR follow the candidate object very well in both rotational and straight following cases of the 6LR.

1.3 Outline of thesis and summary of contributions

This section describes the contents of the thesis and their contributions briefly. The contents consist of seven sections, i.e. introduction, system description, system modeling, image processing for object detection and tracking, controller design for object following control, simulation and experimental results and conclusions as follows:

Chapter 1: Introduction

In this chapter, background and motivation of this thesis are presented. The objective and research method of this thesis are then described. Finally, the outline and summary of contribution of this thesis is given.

Chapter 2: System description

This chapter describes a prototype of the experimental Kinect camera sensor based six-legged robot (6LR) system and basic terminologies of the 6LR. The robot developed in this thesis is explained in three parts such as mechanical design, electrical design and Graphical User Interface (GUI) development. Firstly, the mechanical design of the 6LR which consists of a body and leg

configuration is explained. Secondly, the electrical design of the system is explained. In this section, all the figures and specifications of the developed robot such as Kinect camera sensor, motors, microcontroller, host computer and power supply are described in detail. Thirdly, development of a Graphical User Interface (GUI) which is built to control the Kinect based 6LR is explained. Finally, basic terminologies of the 6LR are summarized in last section of this chapter.

Chapter 3: System modeling

This chapter presents the mathematical modeling of the proposed 6LR system. Firstly, forward kinematic modeling of one leg with four joints and four links is presented. Secondly, kinematics of the six leg robot is explained. Thirdly, kinematic modeling of the 6LR with a candidate object is described. Finally, kinematic modeling of the 6LR is described to define its control inputs such as translational velocity and angular velocity for object following control of the robot.

Chapter 4: Image processing for object detection and Kalman filter for object tracking

This chapter describes an image processing algorithm for a candidate object detection and then Kalman filter algorithm for tracking a moving candidate object. Firstly, a color-based object detection algorithm based on Aforge.NET C# framework is developed to detect the

blue colored candidate object obtained by a Kinect camera sensor. Using this image processing algorithm, the center position coordinate of the candidate object (in pixel) inside RGB image frames is obtained. Secondly, the real world coordinate of the candidate object is obtained in mm. Finally, based on the object motion model, a Kalman filter algorithm is applied to estimate the position and velocity coordinate of the moving candidate object.

Chapter 5: Controller design based on Backstepping control method

This chapter describes a walking control and an object following control of the 6LR using backstepping control method based on Lyapunov stability theory. The walking control section explains the end effector position control of the 6LR using tripod walking gait and differential kinematic algorithm, whereas the object following control section explains the error definition and the controller design for the 6LR to perform an object following task. Finally, the object following movement of the 6LR and a candidate object movement is explained with two cases: rotational following and straight following of the 6LR.

Chapter 6: Simulation and experimental results

This chapter presents simulation and experimental results of the proposed system. These results are shown to prove the effectiveness, performance and applicability of the

designed controller for the 6LR to follow the moving candidate object. The simulation and experimental results show that the proposed controller makes the 6LR able to follow the candidate object well with an acceptable error.

Chapter 7: Conclusions and future works

Conclusions from this research and some ideas for future works are presented.

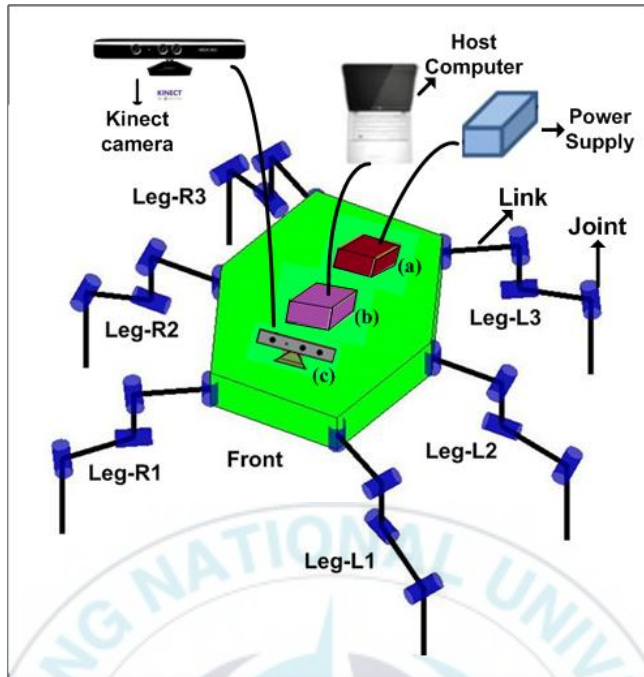


Chapter 2: System Description

This chapter describes the prototype of the experimental Kinect camera sensor based six-legged robot (6LR) system and basic terminologies of the 6LR. The robot developed in this thesis consists of mechanical design, electrical design and Graphical User Interface (GUI) software development. The mechanical design is comprised of a body and six legs. The electrical design consists of a Kinect camera sensor, motors, microcontroller, host computer and power supply. The GUI development explains the software used to build the interface for the Kinect based 6LR.

2.1 Mechanical design

The configuration of the Kinect camera based 6LR system used for this thesis is shown in Fig. 2.1. The 6LR developed in this thesis consists of one body and six legs such that each leg has four rotational joints and four links. For the simplicity of symbols, six legs is numbered as 1, 2, 3 for Leg-L1, Leg-L2, Leg-L3 and 4, 5, 6 for Leg-R3, Leg-R2, Leg-R1. The Kinect camera sensor is mounted on the top front of the body frame of 6LR to perform image processing task. The microcontroller and Bluetooth are attached on the inner side of the body frame of 6LR to control the servomotors and to communicate with the host computer, respectively.



(a)	Microcontroller and Power supply
(b)	Host computer and Bluetooth communication
(c)	Kinect camera sensor

Fig. 2.1 Configuration of Kinect camera based 6LR system

The real Kinect camera based 6LR system developed for this thesis is shown in Fig. 2.2.

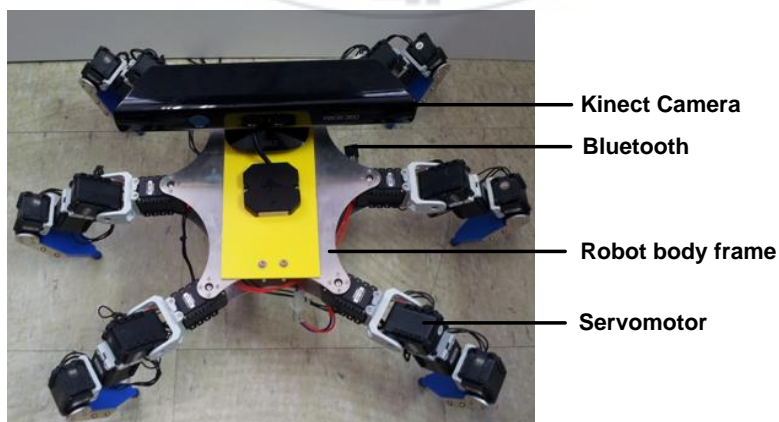


Fig. 2.2 Real Kinect camera based 6LR system

For the leg configuration, there are 24 servo motors used to control the locomotion of 6LR by operating 6 legs (i.e., 24 rotational joints). Each leg has four rotational joints, four links and one end effector to touch the ground. This design of legs can make the robot get more complex postures and better stability. Fig. 2.3 shows one leg configuration of 6LR with four rotational joints, four links and end effector.

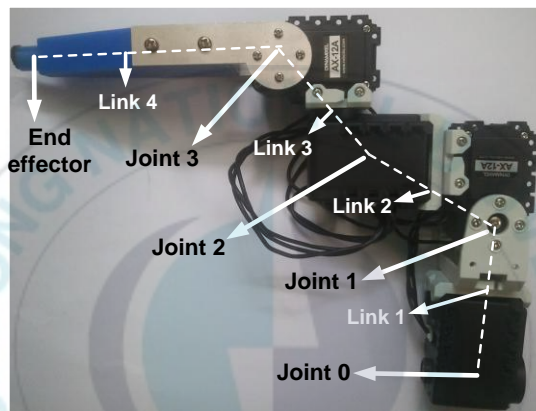


Fig. 2.3 One leg configuration

2.2 Electrical design

The electrical configuration of the Kinect camera based 6LR system is shown in Fig. 2.4. The robot is controlled by host computer and microcontroller. A Kinect camera sensor which is directly connected to the host computer is attached to the top front of 6LR to capture RGB and depth images. User gives commands to the 6LR through the GUI in the host computer. The host computer calculates both image processing and control algorithm and then sends

commands to the microcontroller via Bluetooth through RS232 interface. The microcontroller controls all servomotors and receives feedback data from servomotors through half-duplex community. The following sections describe these components with detailed specifications and figures.

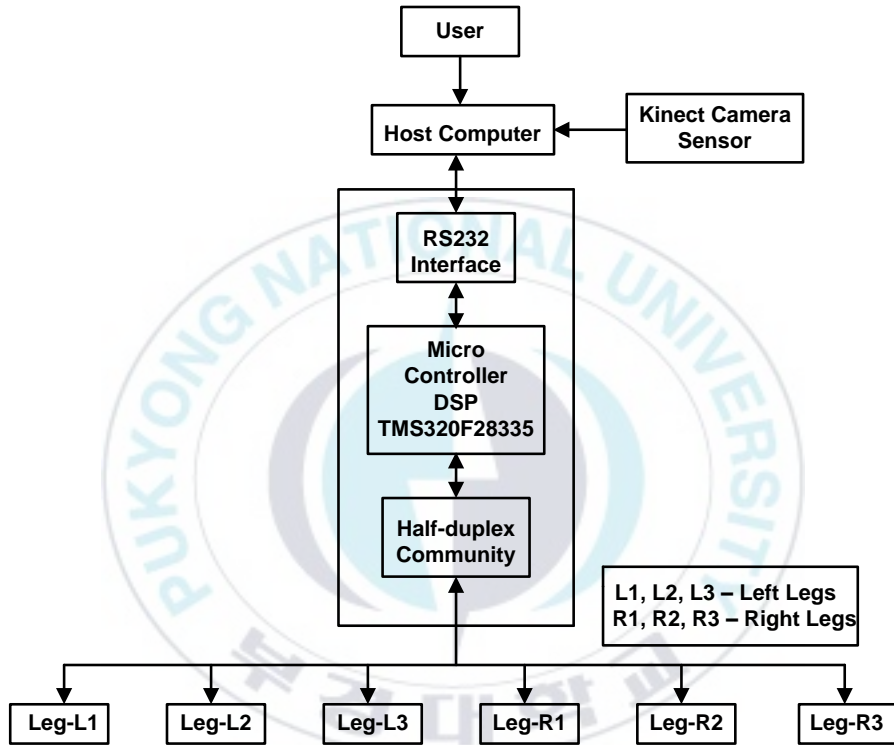


Fig. 2.4 Electrical configuration of Kinect camera based 6LR system

2.2.1 Kinect camera sensor

The Microsoft Kinect camera sensor is a revolutionary low-cost RGB-D camera sensor that is primarily built as an input device for Xbox gaming console [40]. Due to its capability of producing better quality images and depth information, this low cost device becomes

popular in the field of scientific study especially in the field of computer vision and robotics. A software development kit (SDK) provided by Kinect for Microsoft windows offers easy development of Kinect applications [41]. This SDK allows developers to write Kinect applications in C#, C++ or Visual Basic.NET programming languages. Fig. 2.5 shows the Kinect Xbox camera sensor consisting of an IR (infrared) projector, an IR camera, a RGB (color) camera, four microphones array, and a tilting system.

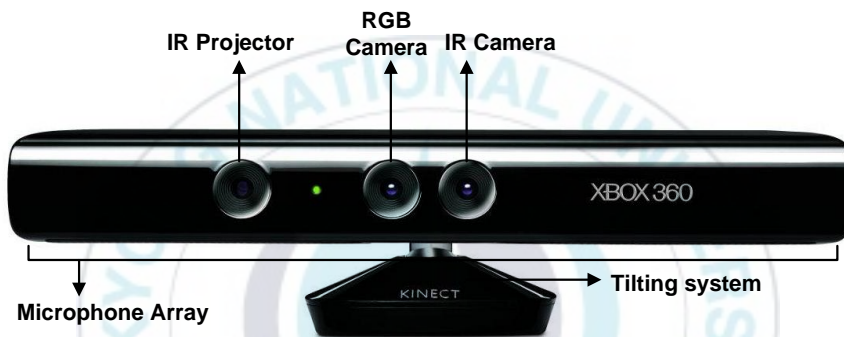


Fig. 2.5 Kinect Xbox camera sensor

The Microsoft Kinect also has its own image processor known as the Primesense's PS1080-A2 system on chip (SoC) processor that processes the images captured by the monochrome and infrared camera. Apart from all these components, Kinect camera sensor also has a power adapter for external power supply and a USB adapter to be connected with a computer.

The depth camera is comprised of an IR laser projector and a monochrome CMOS sensor called as IR camera, which captures video data in 3D under any ambient light conditions. The Kinect camera sensor works in entirely different way through a process

called stereo-triangulation. Stereo-triangulation requires two images in order to get the depth of each point. The Kinect camera produces only one image and the other image is actually invisible. The invisible image contains a “pseudo-random dot” pattern of codes embedded into a chip logic which is projected by an infrared emitter. The above two images are not equivalent since there is an offset between the IR projector and IR camera. Therefore, these images allow the use of stereo-triangulation technique to calculate depth with the help of the infrared pattern produced. This process is shown in Fig. 2.6.

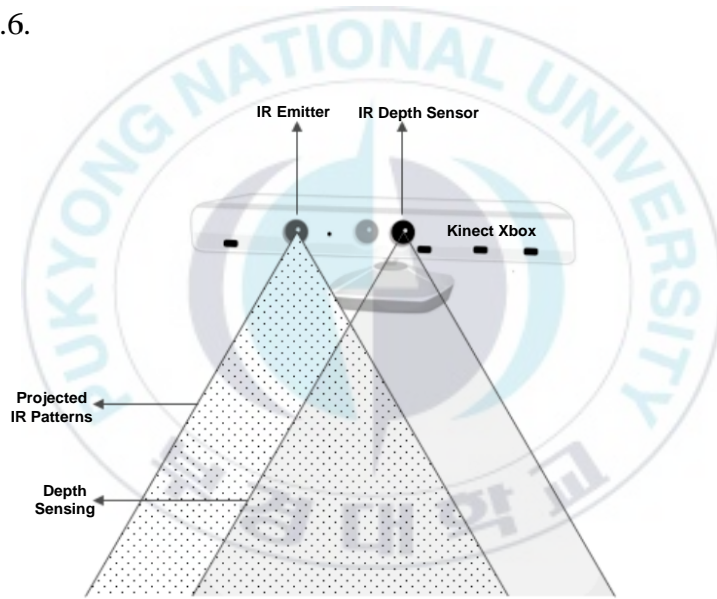


Fig. 2.6 Stereo-triangulation of Kinect camera sensor to obtain depth data

A mathematical modeling of the depth measurement is explained in [42]. Fig. 2.7 shows an illustration of depth calculation of a point R on an object. To express 3D coordinates of the object points, we consider a depth coordinate system with its origin at the perspective center of the IR camera. The Y axis is orthogonal to the

Fig. 2.7 shows that an object is on the reference plane at a distance Y_o from the sensor, and a point on the object is captured on the image plane of an IR camera. If the object is shifted closer to (or further away from) the Kinect camera location, the position of a point on the image plane is displaced in X direction. This is measured in an image as disparity d corresponding to a point R in the object space. From the similarity triangles, it can be written as

$$\frac{D}{b} = \frac{Y_o - Y_R}{Y_o} \quad (2.1)$$

$$\frac{d}{f} = \frac{D}{Y_R} \quad (2.2)$$

where

Y_R : Depth (distance) of the point R in an object space

b : Base length

f : Focal length of the IR camera

D : Distance of the point R in an object space

d : Observed disparity in an image space

Substituting D from Eq. (2.2) into Eq. (2.1), Y_R can be obtained as follows:

$$Y_R = \frac{f}{d} D = \frac{Y_o}{1 + \frac{Y_o}{fb} d} \quad (2.3)$$

where Y_o is the constant determined by calibration.

The RGB camera is responsible for capturing and streaming the color video data at the speed of 30 frames per second. The function of RGB camera is to detect the red, blue and green colors from the captured view. The Kinect camera sensor also exhibits microphone array to support audio data stream. The microphone array consists of

four separate microphones which are placed at the bottom of the body of the Kinect camera sensor. The base and body part of the Kinect camera sensor are connected by a tiny motor called tilting system. Using this, the Kinect camera sensor can change its field of view by moving its body vertically upwards or downwards upto 27° .

Fig. 2.8 shows the working sequence of the Kinect camera sensor to obtain depth data of the objects in front of it. When there is a need to capture depth data, the PrimeSense chip sends a signal to the infrared emitter to turn on the infrared light, and sends another signal to the IR depth sensor to initiate depth data capture from the current viewable range of sensor. The IR emitter starts sending an infrared light invisible to human eyes to the objects in front of the device. The IR depth sensor starts reading the infrared data from the object based on the distance of the individual light point of reflection and passes it to the PrimeSense chip. The chip then analyzes the captured data, and creates a per-frame depth image and passes it to the output depth stream as a depth image.

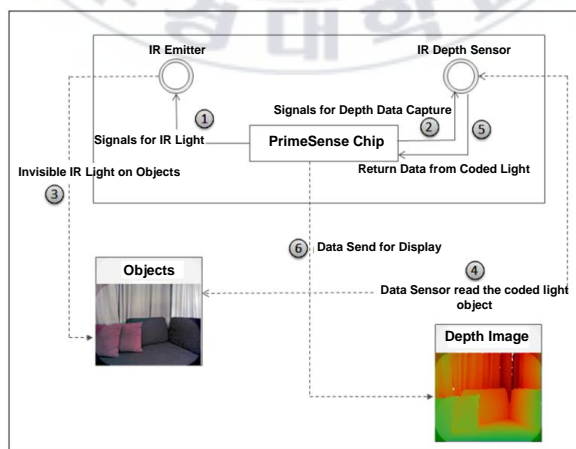


Fig. 2.8 Working sequence of Kinect camera to obtain depth data

The specifications of Kinect Xbox camera sensor are shown in Table 2.1.

Table 2.1 Specifications of the Kinect Xbox camera sensor

No.	Parameters	Values
1	Depth sensor range (mm)	800 to 4000
2	Horizontal field of view (degree)	57
3	Vertical field of view (degree)	43
4	Physical tilt range (degree)	± 27
5	Color data stream	640x480 32-bit color @ 30 FPS
6	Depth data stream	320x240 16-bit depth @ 30 FPS

2.2.2 Microcontroller

In this thesis, a microcontroller used to control the 6LR is DSP (Digital Signal Processor) TMS320F28334 shown in Fig. 2.9. This DSP controller sends joint angles calculated by the host computer to the servomotors of each leg, and receives feedback data from the servomotors.

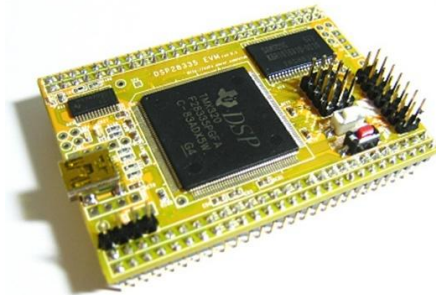


Fig. 2.9 DSP TMS320F28335

2.2.3 Servomotors

Servomotors used in this thesis are Dynamixel AX-12A which is shown in Fig. 2.10. The dynamixel series servomotor is a smart, modular actuator that incorporates a gear reducer, a precision DC motor and a circuitry with network functionality. Despite its compact size, it can produce high torque and is made with high quality material to provide the necessary strength and structural resilience to withstand large external forces. The specifications of Dynamixel AX-12A servomotor is listed in Table 2.2.



Fig. 2.10 Dynamixel AX-12A

Table 2.2 Specifications of Dynamixel AX-12A actuator

No.	Parameters	Values
1	Weight (g)	54.6
2	Dimension (W x L x H) (mm)	32 x 50 x 40
3	Resolution (degree)	0.29
4	Gear reduction ratio	254:1
5	Torque (N.m)	1.5
6	No load speed (rpm)	59
7	Running Degree (degree)	0~300

8	Operating temperature (°C)	-5~70
9	Voltage range (V)	9~12
10	Communication speed (bps)	7343~10 ⁶
11	Material	Engineering plastic

6LR have six legs and each leg has four joints. There are 24 servomotors to control six legs. To control the dynamixel actuators, the main controller needs to convert its UART signals to the half duplex type. Its recommended circuit diagram is shown in Fig. 2.11. The protocol used by Dynamixel AX-12A is Half Duplex Asynchronous Serial Communication (8bit, 1stop, No Parity).

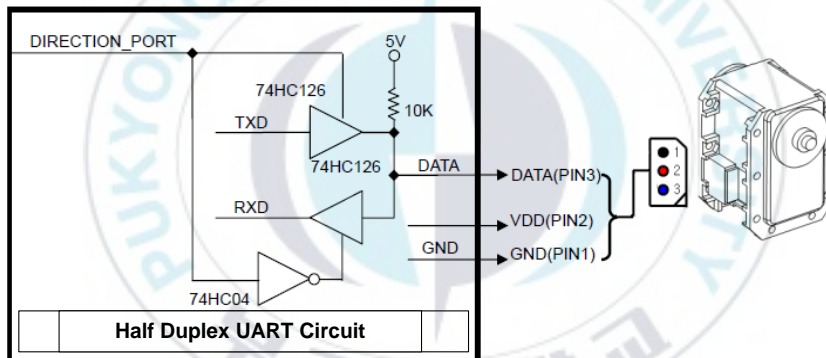
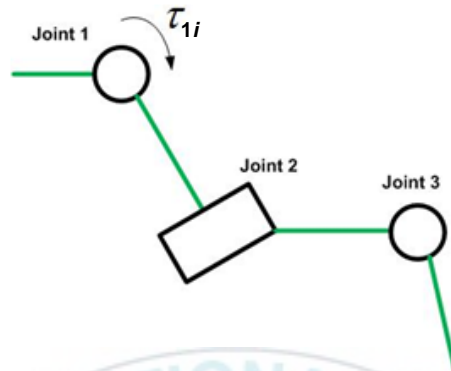


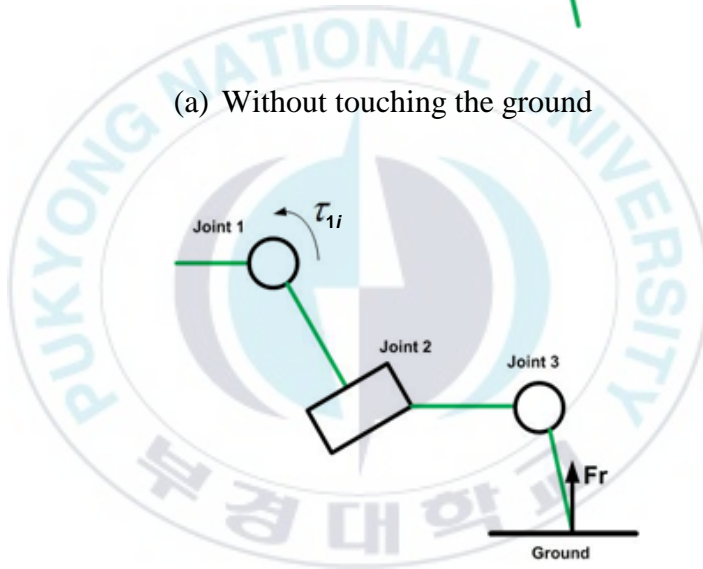
Fig. 2.11 Half duplex UART circuit

The servomotor also has the ability to detect the direction of torque on the motor which can be used as touch detector on joint 1 of each leg. Touch detector is installed at Dynamixel AX-12 servomotor. This kind of servomotor provides data of torque on the actuator. The data are not accurate. They are used only to detect the direction of the torque τ_{li} . Touch detecting method is applied on joint 1 of each leg

as shown in Fig. 2.12. τ_{1i} is the torque of joint 1 of leg i ($i = 1, 2, 3, 4, 5, 6$).



(a) Without touching the ground



(b) With touching the ground

Fig. 2.12 Touch detection method

When the leg doesn't touch the ground, the masses of joints 2 and 3 will make the load on joint 1 negative. When the leg touch the ground, the reaction force $\mathbf{F_r}$ exert on end effector and make the load on joint 1 positive.

2.2.4 Bluetooth

In this thesis, Bluetooth Promi SD202 is used as a wireless communication between 6LR and the host computer. The host computer sends control commands to the DSP microcontroller through this wireless device shown in Fig. 2.13. Table 2.3 shows the specifications of Bluetooth Promi SD202 device.



Fig. 2.13 Bluetooth Promi SD202

Table 2.3 Specifications of Bluetooth Promi SD202

No.	Parameters	Values
1	Range (m)	~100
2	Range of baud rate (bps)	1200-230400
3	Supply voltage (V)	4~12
4	Supply current (mA)	150
5	Dimension (mm)	60 x 26 x 16
6	Operating temperature (°C)	-20~70
7	Level(dBm)	18

8	Bluetooth protocols	RFCOMM, L2CAP, SDP
9	Output power (mW)	63

2.2.5 Power supply

In the Kinect camera based 6LR system, voltage requirement for all motors is 9V~12V. Therefore, ATLASBX ITX100 12V battery is used as shown in Fig. 2.14. The specifications of this battery are listed in Table 2.4.



Fig. 2.14 12V battery ATLASBX ITX100

Table 2.4 Specifications of 12V battery ATLASBX ITX100

No.	Parameters	Values
1	Nominal Voltage (V)	12 V 100Ah
2	Weight	24.2 kg
3	Terminal type	Bolt terminal
4	Dimension (mm)	330 x 171 x 217

2.3 Graphical User Interface (GUI) development

In this thesis, a software program to control an entire Kinect based 6LR system is developed using C# programming language of Microsoft Visual Studio 2010 Express platform. For the ease of application to the user, a Graphical User Interface (GUI) is developed as shown in Fig. 2.15. To build this GUI, Windows Presentation Foundation (WPF) graphical system is used, which employs XAML-language to define and link various interface elements such as Toolbox, Device window, etc.

This program primarily requests the Kinect camera sensor to display the RGB video stream. Then, depending on the color selection, a candidate object is chosen as the object for detection and following. Based on the color detection algorithm, the position of the candidate object inside the RGB video stream is calculated. If the object is moving, the position of the moving object is estimated using Kalman filter algorithm. Finally, this program computes the control algorithm which is based on the backstepping method and sends translational and angular velocities to the 6LR to make it follow the candidate object.

On the GUI shown in Fig. 2.14, there are five partitions. For the ease of understanding, these partitions are numbered as 1, 2, 3, 4 and 5. The partition '1' has two combo boxes: one is for displaying COM port list and the other is for color selection of the candidate object. The candidate object can also be chosen by selecting 'ClickImage' option of the color selection combo box. The color of the candidate object can be customized by changing the values inside the text boxes named as 'Red', 'Green' and 'Blue'. The partition '2' is for object

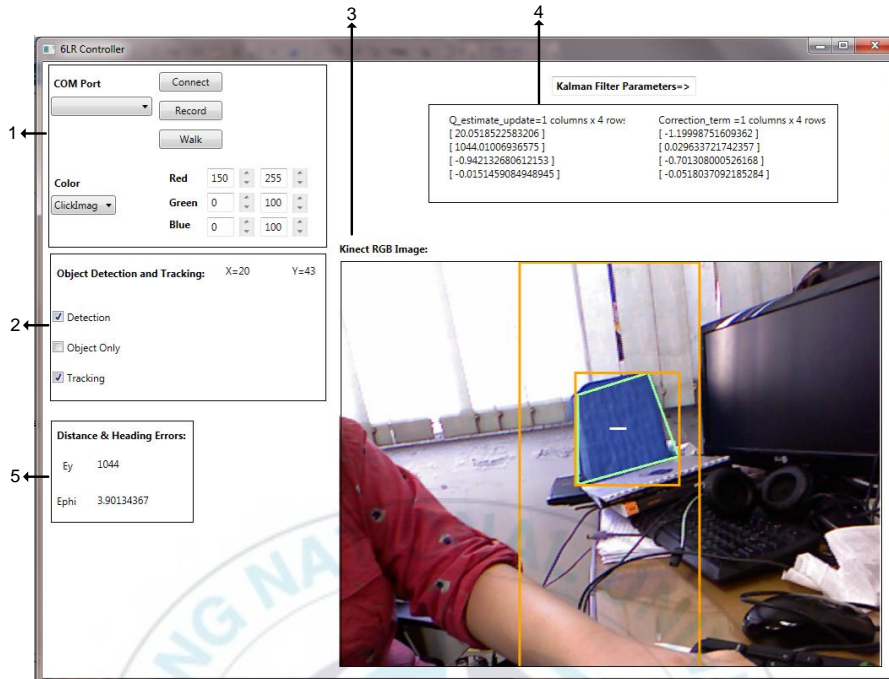


Fig. 2.15 GUI of the proposed system

detection and tracking control. It includes three check boxes namely 'Detection', 'Object only' and 'Tracking'. When the 'Detection' check box is ticked, Kinect camera sensor is requested to display RGB video stream inside the image screen referred as partition '3'. The size of the image screen is chosen as 640×480 pixels. The X and Y position of the candidate object is then calculated and displayed inside the partition '2'. The 'Object only' check box displays only the candidate detected object, if ticked. When the 'Tracking' check box is ticked, firstly, Kalman filter algorithm starts computing the estimated position of the moving candidate object and then the errors and control parameters are calculated by the program using backstepping control algorithm. The estimated parameters obtained from the Kalman filter are displayed in partition '4', whereas the distance error and heading angle error obtained from the control

method are shown in partition ‘5’. Finally, the ‘Connect’ and ‘Walk’ buttons in the partition 1 connects this interface to the 6LR. The ‘Connect’ button in the partition 1 connects the 6LR to the COM port (i.e., to this interface). The ‘Walk’ button in the partition 1 sends the computed control input parameters to the microcontroller installed on the 6LR and thus, the 6LR starts walking. And, the ‘Record’ button in the partition 1 records all the essential computational data for further use.

2.4 Basic terminologies of the 6LR

This section explains typical definitions and the concept of walking gait for the legged robot.

2.4.1 Typical definitions for legged robot

A *leg cycle* is composed of two phases as shown in Fig. 2.16.

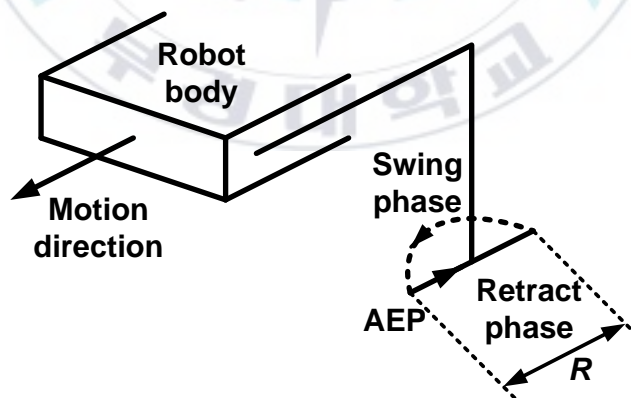


Fig. 2.16 Leg cycle of one leg

where R is the translational distance that the robot body moves after one cycle and called *stroke length*.

- The *swing phase* is the period while the leg moves from stand position to the *Anterior Extreme Position* (AEP) which is the position at the end of the swing phase.
- The *retract phase* is the period while the leg is in contact with the ground and propels the body.

The *cycle time* T is the sum of swing phase time and retract phase time.

$$T = T_{sw} + T_{rt} \quad (2.4)$$

where T_{sw} is the swing phase time and T_{rt} is the retract phase time.

The *support polygon* is the polygon determined by the leg end effectors touching the ground. It is a dotted triangle in the tripod walking gait as shown in Fig. 2.17. There must be at least three touching points on the ground to support the 6LR.

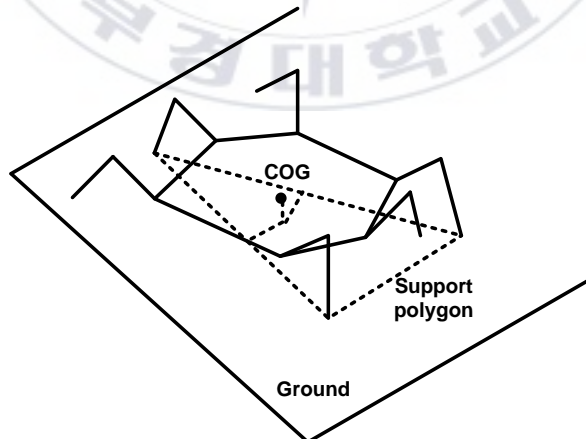


Fig. 2.17 Support polygon of the 6LR

For any support polygon, the distance between ‘the projection of the center of gravity (COG) on the support polygon’ and ‘the intersection of the front edge (in the walking direction) of the polygon with the longitudinal centerline of the body’ is called the *front stability margin*. The distance between ‘the projection of the center of gravity on the support polygon’ and ‘the intersection of the rear edge (in the opposite direction of walking) of the polygon with the longitudinal centerline of the body’ is called the *rear stability margin*. The definitions of the front and rear stability margins are shown in Fig. 2.18.

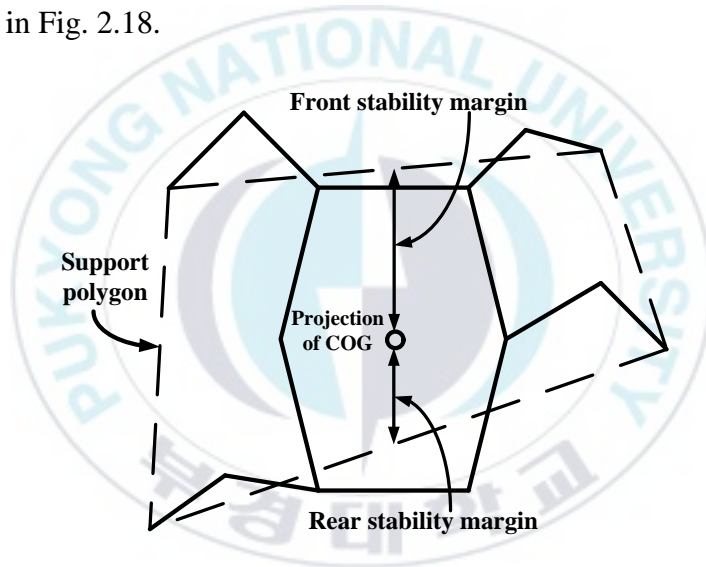


Fig. 2.18 Front and rear margins

Longitudinal Stability Margin: The minimum of the front and rear stability margins is called the longitudinal stability margin of the support polygon. (Longitudinal Gait) Stability Margin: The minimum of the longitudinal stability margins of all possible support polygons during walking according to a gait is called the stability margin of the gait. If the stability margin of a gait is larger than that of another gait, the former gait is called to be more stable than the latter.

Statically Stable Gait: A gait is said to be statically stable if its stability margin is greater than zero as in Fig. 2.18. Otherwise, the gait is said to be statically unstable as shown in Fig. 2.19. If a gait is stable, this means that the projection of center of gravity on the horizontal plane always remains inside the support polygon; otherwise, the center of gravity remains outside the support polygon in some instant times of the walk and the body falls down.

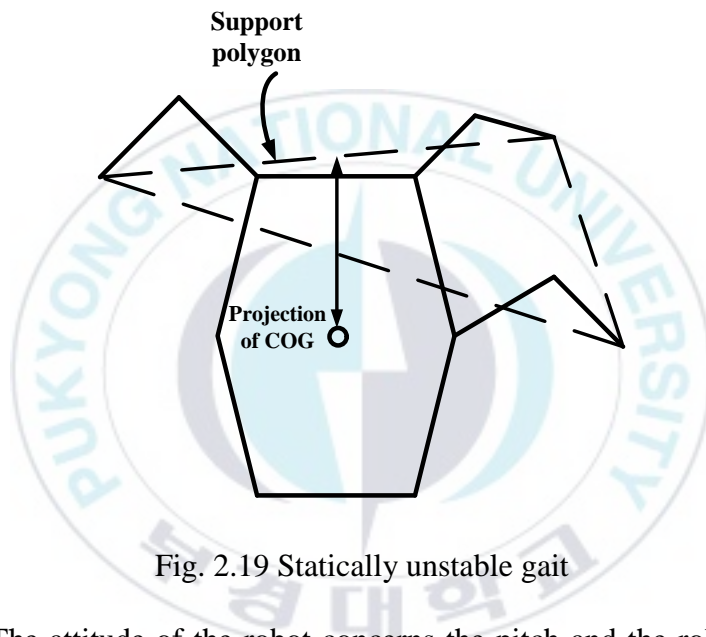


Fig. 2.19 Statically unstable gait

The attitude of the robot concerns the pitch and the roll of the robot with respect to the vertical global axis. The altitude of the robot is defined as the height of its center of mass with respect to the ground reference.

The posture of the robot is defined as the position of the six legs with respect to the main body.

The workspace of a leg is a reachable space by the end effector of the leg. This space has been restricted to a simple cylinder as shown in Fig. 2.20.

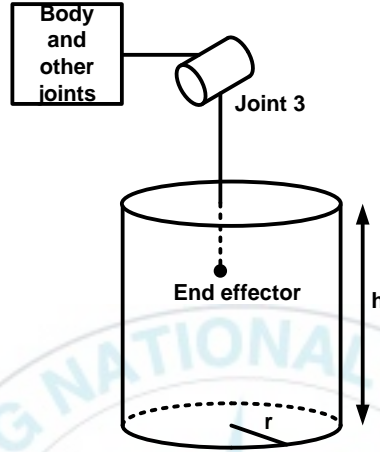


Fig. 2.20 Workspace of one leg

where r is the workspace radius and h is the workspace height.

2.4.2 Walking gait

In this thesis, periodic gait is chosen for the six legs. Periodic gait means that the step cycle is same for the six legs. There exist several periodic gaits depending on the duty factor and the phase shift between legs such as tripod gait, regular gait, symmetric gait and wave gait.

Among all these gaits, tripod gait is chosen as the walking gait for 6LR in this thesis. The walking cycle imitating this gait is shown in Fig. 2.21. When the robot walks, three legs support the body and the other three legs lift up and move for next walking motion. The moving leg's sequence in Fig. 2.21 can be described as follows: At

first, Leg-L1, Leg -L3 and Leg-R2 are at swing phase, Leg-L2, Leg-R1 and Leg-R3 are at retract phase. After a half of cycle, Leg-L1, Leg -L3 and Leg-R2 are at retract phase, and Leg-L2, Leg-R1 and Leg-R3 are at swing phase. The sequence is repeated every cycle.

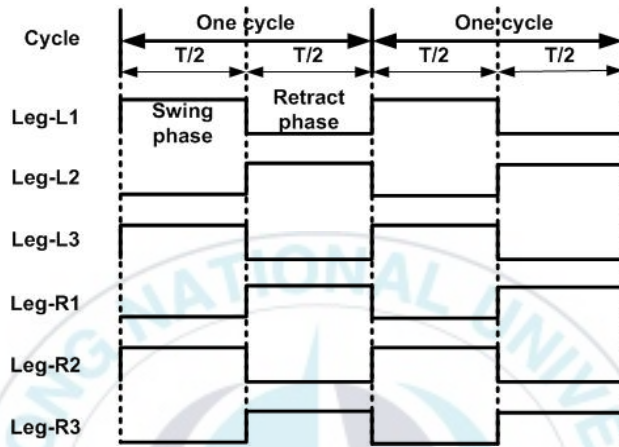


Fig. 2.21 Walking cycle of the tripod gait

In the swing phase, the end effector of each leg lifts up diagonally and moves to the AEP. In the retract phase, the end effector contacts the ground and moves back to the default standing position. The movement of the end effector in retract phase makes the body of the robot be propelled. The swing phase is designed as shown in Fig. 2.22. The swing phase is composed of two states: Diagonal lift up and Put down.

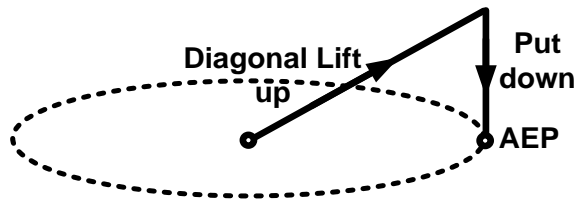


Fig. 2.22 Swing phase of step cycle of leg i

The swing phase and the retract phase have the same duration in one step cycle. A leg in the retract phase only moves when the other leg is in the swing phase as shown in Fig. 2.23.

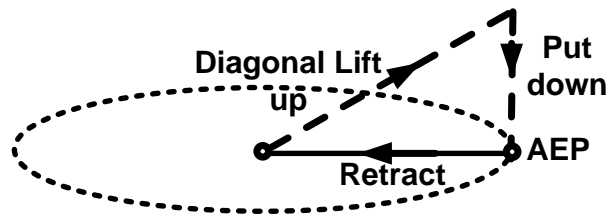
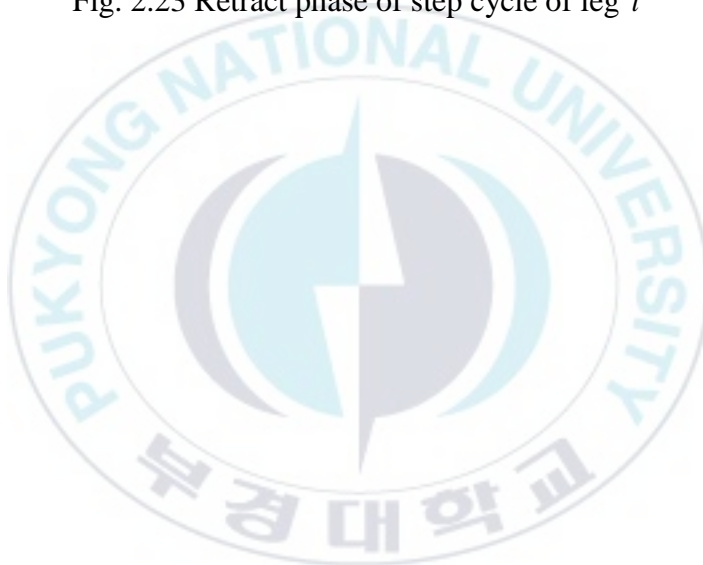


Fig. 2.23 Retract phase of step cycle of leg i



Chapter 3: System Modeling

This chapter presents the mathematical modeling of the proposed 6LR system. Firstly, a forward kinematics of one leg with four joints and four links is presented. Secondly, kinematics of the 6LR is explained. Thirdly, kinematic modeling of the 6LR with a candidate object is described. Finally, kinematic modeling of the 6LR to define its control inputs such as translational velocity and angular velocity is described in this chapter.

3.1 Kinematic modeling of one leg with four joints

Fig. 3.1 shows the configuration of one leg with four joints and four links of the developed 6LR. The Denavit-Hartenberg (DH) convention is adopted to define the link frame as follows [43]:

- Choose axis z_{ji} along the shaft of joint j of leg i .
- Locate the origin O_{ji} at the intersection of axis z_{ji} with the common normal to axes $z_{(j-1)i}$ and z_{ji} . Also, locate O'_{ji} at the intersection of the common normal line with axis $z_{(j-1)i}$ from O_{ji} .
- Choose axis x_{ji} along the common normal line to axes $z_{(j-1)i}$ and z_{ji} with the direction from joint j to joint $j+1$ of leg i .
- Choose axis y_{ji} so as to complete a right-handed frame.
- a_i is the distance between O_{ji} and O'_{ji} .
- d_i is coordinate of O'_{ji} along $z_{(j-1)i}$.

- α_{ji} is angle between $z_{(i-1)i}$ and z_{ji} about axis x_{ji} to be taken positive when rotation is in counter-clockwise.
- θ_{ji} is angle between $x_{(j-1)i}$ and x_{ji} about axis $z_{(j-1)i}$ to be taken positive when rotation is in counter-clockwise.

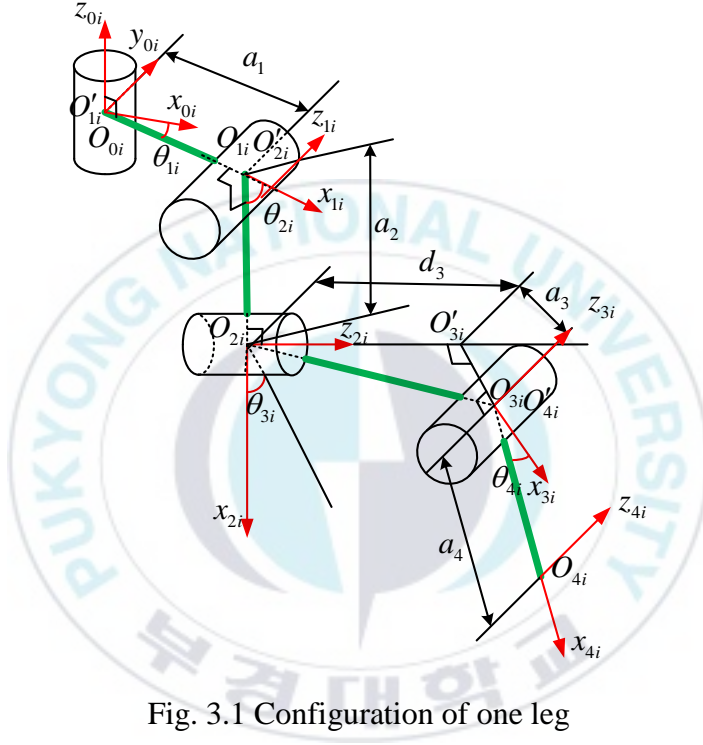


Fig. 3.1 Configuration of one leg

where

O_{ji} : Origin of the joint j ($j = 0, 1, 2, 3$) of leg i ($i = 1, 2, 3, 4, 5, 6$)

θ_{1i} : Rotational angle around z_{0i} axis of leg i

θ_{2i} : Rotational angle around z_{1i} axis of leg i

θ_{3i} : Rotational angle around z_{2i} axis of leg i

θ_{4i} : Rotational angle around z_{3i} axis of leg i

The above Denavit-Hartenberg (DH) convention is described clearly in Appendix A.

A coordinate of $(x_{ji}y_{ji}z_{ji})$ is located on joint j of leg i . DH parameters are obtained by these coordinates for forward kinematic.

Table 3.1 D-H Parameters of 4-Link leg

Joint	θ_{ji}	α_i	a_i	d_i
0	θ_{1i}	$-\pi/2$	a_1	0
1	θ_{2i}	$\pi/2$	a_2	0
2	θ_{3i}	$-\pi/2$	a_3	d_3
3	θ_{4i}	0	a_4	0

The homogeneous transformation matrix can be obtained by D-H parameters as the following matrices.

$${}^0_1\mathbf{T}_i = \begin{bmatrix} c_{1i} & 0 & -s_{1i} & a_1c_{1i} \\ s_{1i} & 0 & c_{1i} & a_1s_{1i} \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

$${}^1_2\mathbf{T}_i = \begin{bmatrix} c_{2i} & 0 & s_{2i} & a_2c_{2i} \\ s_{2i} & 0 & -c_{2i} & a_2s_{2i} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$${}^2_3\mathbf{T}_i = \begin{bmatrix} c_{3i} & 0 & -s_{3i} & a_3c_{3i} \\ s_{3i} & 0 & c_{3i} & a_3s_{3i} \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$${}^3_4\mathbf{T}_i = \begin{bmatrix} c_{4i} & -s_{4i} & 0 & a_4c_{4i} \\ s_{4i} & c_{4i} & 0 & a_4s_{4i} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

where $c_{ji} = \cos \theta_{ji}$, $s_{ji} = \sin \theta_{ji}$ and ${}^j_k\mathbf{T}_i$ is the transformation matrix from joint j coordinate system to joint k coordinate system of leg i .

The transformation matrix from joint 0 to end effector of leg i can be calculated as follows:

$${}^0_4\mathbf{T}_i = {}^0_1\mathbf{T}_i {}^1_2\mathbf{T}_i {}^2_3\mathbf{T}_i {}^3_4\mathbf{T}_i \quad (3.5)$$

The end effector position vector with respect to joint 0 frame of leg i $\mathbf{p}_{ei}^0 = [x_{ei}^0 \ y_{ei}^0 \ z_{ei}^0]^T$ can be calculated by the first three elements of the last column of ${}^0_4\mathbf{T}_i$ also called forward kinematic equation $\mathbf{k}(\boldsymbol{\theta}_i)$. The forward kinematic equation is shown as follows:

$$\mathbf{k}(\boldsymbol{\theta}_i) = \mathbf{p}_{ei}^0 = \begin{bmatrix} x_{ei}^0 & y_{ei}^0 & z_{ei}^0 \end{bmatrix}^T \quad (3.6)$$

$$x_{ei}^0 = a_1c_{1i} - a_4c_{4i}(s_{1i}s_{3i} - c_{1i}c_{2i}c_{3i}) + a_2c_{1i}c_{2i} + d_3c_{1i}s_{2i} - a_3s_{1i}s_{3i} + a_3c_{1i}c_{2i}c_{3i} - a_4c_{1i}s_{2i}s_{4i} \quad (3.7)$$

$$y_{ei}^0 = a_1s_{1i} + a_4c_{4i}(c_{1i}s_{3i} - s_{1i}c_{2i}c_{3i}) + a_2s_{1i}c_{2i} + a_3c_{1i}s_{3i} - d_3s_{1i}s_{2i} + a_3s_{1i}c_{2i}c_{3i} - a_4s_{1i}s_{2i}s_{4i} \quad (3.8)$$

$$z_{ei}^0 = d_3c_{2i} - a_2s_{2i} - a_3c_{3i}s_{2i} - a_4c_{2i}s_{4i} - a_4s_{2i}c_{3i}c_{4i} \quad (3.9)$$

where $\boldsymbol{\theta}_i = [\theta_{1i}, \theta_{2i}, \theta_{3i}, \theta_{4i}]^T$ is the angular position vector of each joint and is the solution of Eqs. (3.6) ~ (3.9) using the inverse kinematics.

3.2 Kinematics of the six leg robot

The legs are attached to the platform of the robot. Such a solution implies the constant values in transformation of particular legs. The body frame assignment on the platform depends only on the geometrical dimensions of the robot as shown in Fig. 3.2 in which W is the distance between Leg-L2 and Leg-R2, L is the length of the robot and E is the distance between Leg-L1 and Leg-R1. It is assumed that the center of gravity is also the center of body of the 6LR.

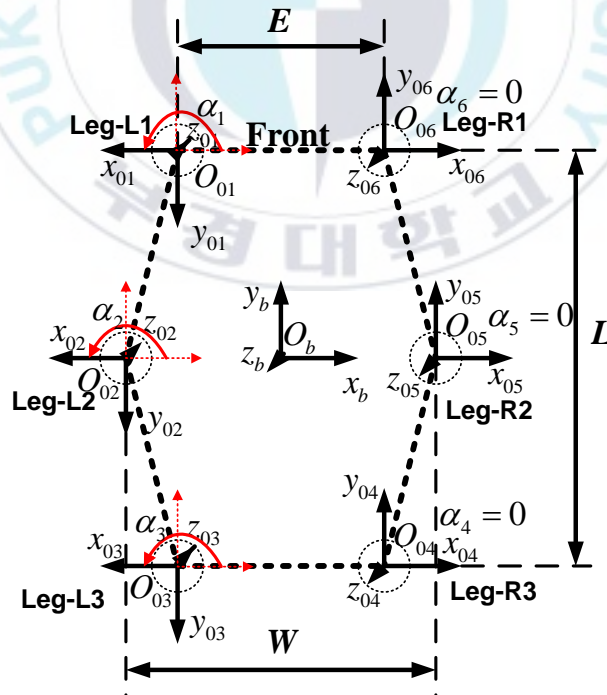


Fig. 3.2 Frame coordinates of the 6LR

$O_b x_b y_b z_b$ is a body coordinate frame located on the center of body of the 6LR, and $O_{ji} x_{ji} y_{ji} z_{ji}$ is a leg coordinate frame located on joint j of leg i .

At swing phase, the shoulder of the leg is fixed and the end effector of the leg is moving. The end effector of the leg moves to the front. At the retract phase, the end effector moves to the rear. But in this phase, the leg is fixed to the ground so that the shoulder moves to the front. The overall movement of the platform can be computed as a function of foot movement by using kinematics of the whole robot.

Using Fig. 3.2, the homogeneous transformation matrix ${}^b_0\mathbf{T}_i$ from body frame to the joint 0 of leg i can be derived as follows:

$${}^b_0\mathbf{T}_i = \begin{bmatrix} \cos \alpha_{0i} & -\sin \alpha_{0i} & 0 & x_{0i}^{x_b} \\ \sin \alpha_{0i} & \cos \alpha_{0i} & 0 & y_{0i}^{y_b} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

where α_{0i} is the angle to transform from $O_b x_b y_b z_b$ coordinate frame to the $O_{ji} x_{ji} y_{ji} z_{ji}$ coordinate frame of joint 0 of leg i and $x_{0i}^{x_b}, y_{0i}^{y_b}$ are the translational displacements to transform from $O_b x_b y_b z_b$ coordinate frame to the $O_{ji} x_{ji} y_{ji} z_{ji}$ coordinate frame of joint 0 of leg i along x_b axis and y_b axis, respectively.

Using Eq. (3.10), the homogeneous transformation matrices from body frame to the first joint of each leg can be written as follows:

- The body frame transformation matrix to Leg-R1 frame by translating a distance $(E/2)$ along x_b axis and a distance $(L/2)$ along y_b axis is expressed by

$${}^b_0\mathbf{T}_1 = \begin{bmatrix} 1 & 0 & 0 & E/2 \\ 0 & 1 & 0 & L/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

- The body frame transformation matrix to Leg-R2 frame by translating a distance $(W/2)$ along x_b axis is expressed by

$${}^b_0\mathbf{T}_2 = \begin{bmatrix} 1 & 0 & 0 & W/2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

- The body frame transformation matrix to Leg-R3 frame by translating a distance $(E/2)$ along x_b axis and a distance $(L/2)$ along inverse direction of y_b axis is expressed by

$${}^b_0\mathbf{T}_3 = \begin{bmatrix} 1 & 0 & 0 & E/2 \\ 0 & 1 & 0 & -L/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

- The body frame transformation matrix to Leg-L3 frame by translating a distance $(E/2)$ along inverse direction of x_b axis and a distance $(L/2)$ along inverse direction of y_b axis and rotating an angle π about z_b axis is expressed by

$${}^b_0\mathbf{T}_4 = \begin{bmatrix} -1 & 0 & 0 & -E/2 \\ 0 & -1 & 0 & -L/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

- The body frame transformation matrix to Leg-L2 frame by translating a distance $(W/2)$ along inverse direction of x_b axis and rotating an angle π about z_b axis is expressed by

$${}^b_0\mathbf{T}_5 = \begin{bmatrix} -1 & 0 & 0 & -W/2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$

- The body frame transformation matrix to Leg-L1 frame by translating a distance $(E/2)$ along inverse direction of x_b axis and a distance $(L/2)$ along y_b axis and rotating an angle π about z_b axis is expressed by

$${}^b_0\mathbf{T}_6 = \begin{bmatrix} -1 & 0 & 0 & -E/2 \\ 0 & -1 & 0 & L/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

The end effector position vector of each leg with respect to body frame can be derived from the transformation matrix as follows:

$$\tilde{\mathbf{p}}_{ei}^b = {}^b_0\mathbf{T}_i \tilde{\mathbf{p}}_{ei}^0 \quad (3.17)$$

where $\tilde{\mathbf{p}}_{ei}^b = \begin{bmatrix} \mathbf{p}_{ei}^b \\ 1 \end{bmatrix}$, $\tilde{\mathbf{p}}_{ei}^0 = \begin{bmatrix} \mathbf{p}_{ei}^0 \\ 1 \end{bmatrix}$, and \mathbf{p}_{ei}^b is the position vector of the end effector with respect to body frame of leg i .

3.3 Kinematic modeling to define control inputs

Fig. 3.3 shows the kinematic configuration of the 6LR. In this thesis, two control parameters are chosen to control the 6LR for object following based on its kinematic equations. These control inputs are angular velocity and translational velocity of the 6LR which can be expressed as follows:

$$\omega = \frac{\theta_d}{T} \quad (3.18)$$

$$\theta_d = \frac{r}{2\pi d_d} \times 360^\circ \quad (3.19)$$

$$v = \frac{r}{T} \quad (3.20)$$

where

$O_b x_b y_b$: Body coordinate frame of the 6LR

ω : Angular velocity of the 6LR

θ_d : Desired angle to achieve anterior extreme position (AEP) for each leg measured in counter-clockwise direction (rad)

T : Cycle time

d_d : Distance between the center of the 6LR and the end effector anterior extreme position (AEP)

v : Translational velocity of the 6LR along y_b axis

r : Step distance of each leg

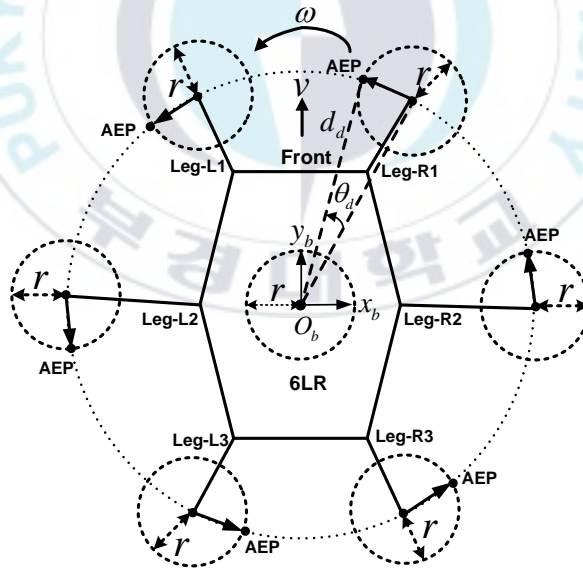


Fig. 3.3 Kinematic configuration of the 6LR

3.4 Kinematic modeling of the 6LR with a candidate object

Fig. 3.4 shows the kinematic configuration of the 6LR with a blue colored candidate object.

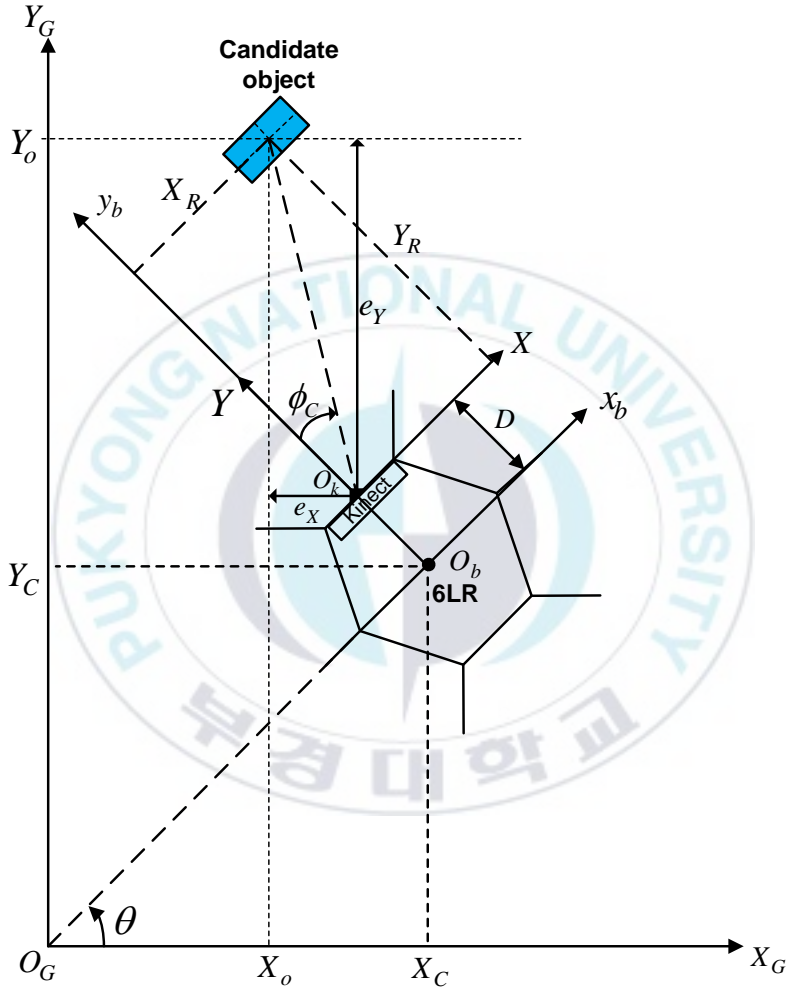


Fig. 3.4 Kinematic configuration of the 6LR following a candidate object

where

$O_G X_G Y_G$: Global coordinate frame

- $O_b x_b y_b$: Body coordinate frame of the 6LR
- $O_k XY$: Local coordinate frame of the Kinect camera sensor
- X_C, Y_C : Current position coordinate of the 6LR in global coordinate frame
- X_o, Y_o : Current position coordinate of the candidate object in global coordinate frame
- e_x, e_y : Error coordinate between the center of the Kinect camera sensor and center of a candidate object in X and Y direction, respectively
- X_R, Y_R : Current position coordinate of the candidate object in body coordinate frame
- θ : Current angular position of the 6LR in global coordinate measured from X_G axis
- ϕ_C : Current angular position of the candidate object measured from the Kinect camera

The kinematic equation for this 6LR configuration in global coordinate frame $O_G X_G Y_G$ is expressed as follows:

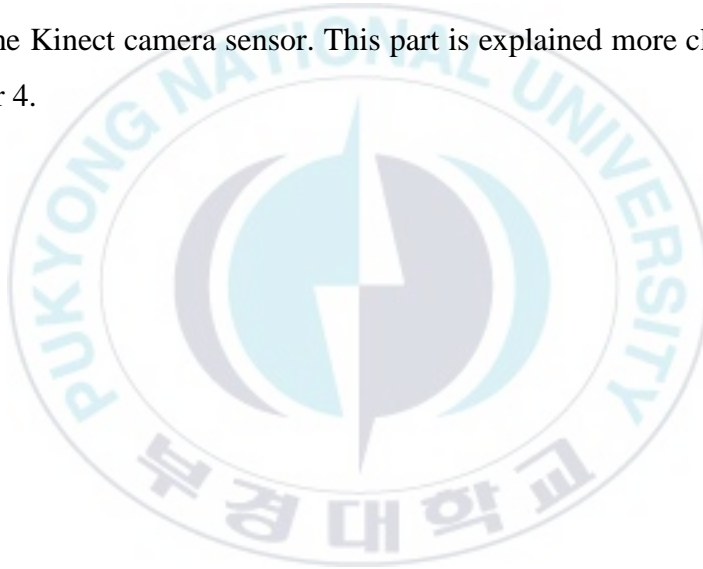
$$\begin{bmatrix} \dot{X}_C \\ \dot{Y}_C \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.21)$$

Using the coordinate transformation, the global position coordinate of the candidate object in coordinate frame $O_G X_G Y_G$ is obtained as follows:

$$\begin{bmatrix} X_o \\ Y_o \end{bmatrix} = \begin{bmatrix} X_c \\ Y_c \end{bmatrix} + \begin{bmatrix} \cos \phi_c & \sin \phi_c \\ -\sin \phi_c & \cos \phi_c \end{bmatrix} \begin{bmatrix} X_R \\ Y_R \end{bmatrix} - D \begin{bmatrix} \sin \theta \\ \cos \theta \end{bmatrix} \quad (3.22)$$

where D is the distance between the Kinect camera sensor and the center of the 6LR body.

Practically, values of the position coordinate of a candidate object (X_R, Y_R) in the local coordinate frame O_kXY are obtained from the Kinect camera sensor. This part is explained more clearly in chapter 4.



Chapter 4: Image Processing for Object Detection and Kalman Filter for Object Tracking

This chapter introduces an image processing algorithm for the Kinect camera sensor to detect a candidate object and then a Kalman filter algorithm is described to track the position and velocity of a moving candidate object. Firstly, a color-based object detection algorithm based on Aforge.NET C# framework is developed to obtain the center coordinate of the candidate object in pixel inside the Kinect RGB image frames. Secondly, using trigonometric laws and depth data from the Kinect camera sensor, the pixel coordinate of the candidate object are converted into the real world coordinate in mm. Finally, based on the object motion model, the Kalman filter algorithm is described to estimate the real positions and velocities of the moving candidate object.

4.1 Color-based object detection algorithm

In this thesis, a color-based object detection algorithm is presented for the Kinect camera sensor to detect the position of a blue colored candidate object. This algorithm is developed using AForge.NET C# framework [44] which provides useful set of image processing filters and tools designed to develop image processing algorithms. In this algorithm, the first step is to detect a blue colored candidate object in the Kinect RGB image frame. The second step is to get the center coordinate of the detected candidate object in the

Kinect RGB frames (in pixel). The last step is to convert the pixel coordinate of the candidate object to the real world coordinate in mm. This algorithm is executed as follows:

Step 1: Using Kinect camera sensor, both RGB and depth data are collected at the speed of 30 frames per second. The simplest object detection is achieved by performing color filtering on the Kinect RGB images. The color filtering process filters pixels inside or outside of specified blue color range and fills the rest with black color. In this process, only the object with blue color is kept and all the rest are removed out.

Step 2: The second step is to find out the coordinate of the blue colored candidate object. This is done by using ‘Blob counter’ tool which counts and extracts stand-alone objects in images using ‘connected component labeling’ algorithm [45]. The connected component labeling algorithm is used to detect connected regions in binary digital images. This algorithm treats all pixels with values less than or equal to ‘background threshold’ as background, but pixels with higher values are treated as object’s pixels. However, ‘Blob counter’ tool works with grayscale images, therefore, gray scaling is applied on the images before using this tool. Now, the final step is to locate the detected candidate object in RGB image frames. Once the candidate object is located, the center coordinate of the object in pixels are obtained. Fig. 4.1 shows the Kinect RGB image of the detected candidate object with its center coordinate (X_{RGB}, Y_{RGB}) within a desired object area.

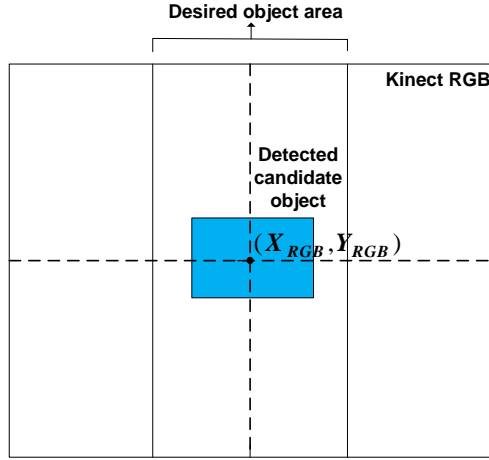


Fig. 4.1 Kinect RGB image of the detected candidate object with its center coordinate within a desired object area

Step 3: This step explains the conversion of pixel coordinate (X_{RGB}, Y_{RGB}) of the detected candidate object to the real world coordinate (X_R, Y_R) in mm.

Fig. 4.2 shows an extended view of the current angle from the Kinect camera with detected candidate object in the RGB image frame and the real candidate object in the real world frame. According to the image principle explained in [46], the current angle between the center of the detected candidate object and the center of RGB image is given by,

$$\phi_c = \arctan\left(\frac{n\alpha}{f}\right) \quad (4.1)$$

where α is the coefficient of transformation from pixel to length, f is the focal length of the Kinect camera and n is the pixel difference

between the center of the detected candidate object and the center of the RGB image.

In this thesis, only heading angle is considered for the measurement purpose. Therefore, $n = X_{RGB}$. Thus, the current angle in Eq. (4.1) is rewritten as follows:

$$\phi_c = \arctan\left(\frac{X_{RGB}\alpha}{f}\right) \quad (4.2)$$

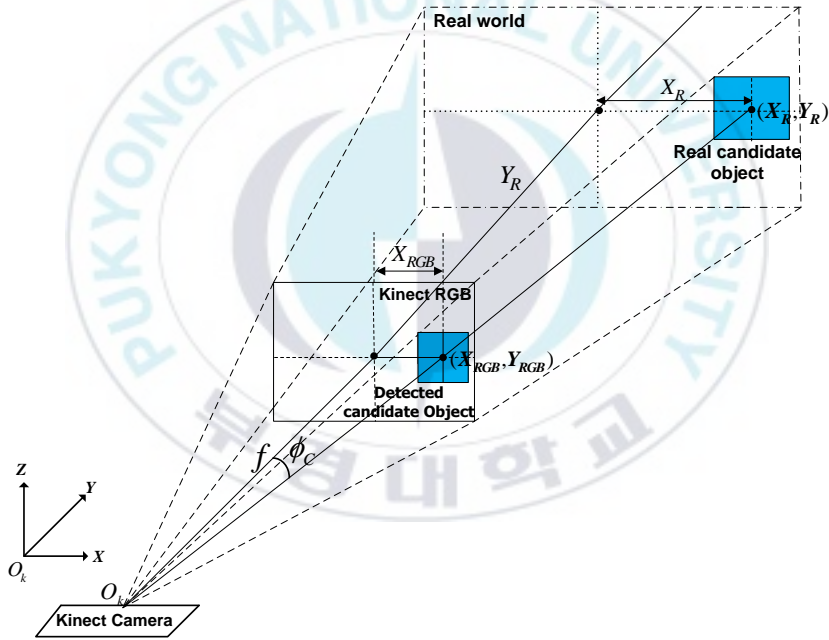


Fig. 4.2 Extended view of the current angle from the Kinect camera in the RGB image frame and the real world frame

By calculating the current angle ϕ_c and using simple trigonometry, from Fig. 4.2, the position of the real candidate object in X direction can be obtained as

$$X_R = Y_R (\tan \phi_C) \quad (4.3)$$

where Y_R is the position of the real candidate object in Y direction which represents depth in mm and is obtained directly from the Kinect depth data as explained in section 2.2.1.

In this thesis, a desired (reference) depth between the 6LR and the candidate object is 1100 mm as shown in Fig. 4.3. If the moving candidate object is at a distance more than 1100 mm, the 6LR goes forward. If the desired depth between the 6LR and the candidate object reaches 1100 mm, the 6LR stops.

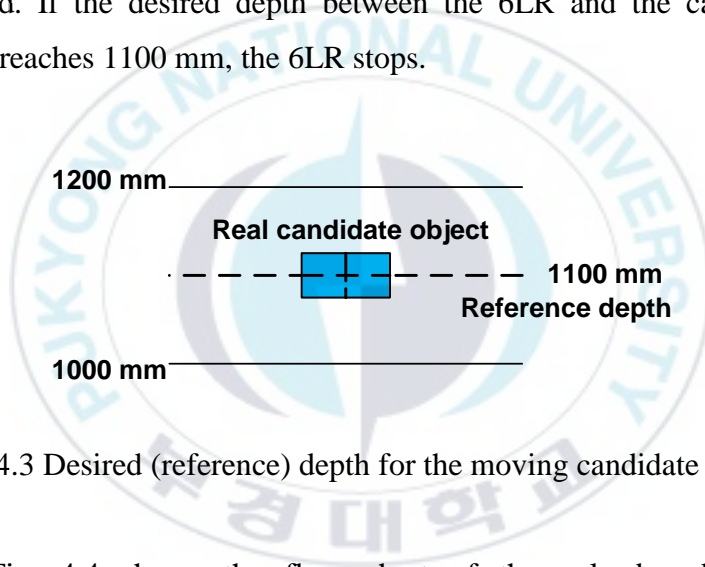


Fig. 4.3 Desired (reference) depth for the moving candidate object

Fig. 4.4 shows the flow chart of the color-based object detection algorithm.

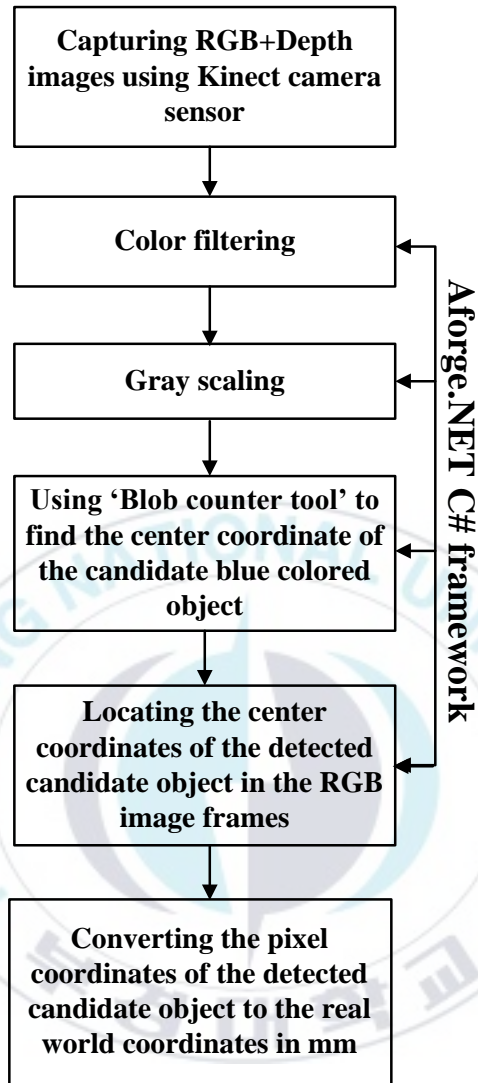


Fig. 4.4 Flow chart of the color-based object detection algorithm

4.2 Kalman filter algorithm for moving object tracking

To follow a moving object well, it is essential to estimate and track its position and velocity. However, due to the variations in ambient light and noise of the Kinect camera sensor, tracking an

object well is not an easy task. Kinect camera sensor may detect multiple unwanted objects due to different real-time conditions. Hence, it is necessary to keep track of the detected candidate object. To overcome these problems, a Kalman filter algorithm [47, 48] is adopted in this thesis. The Kalman filter is a recursive estimator that estimates the current state of the candidate object by using estimated state from the previous time step and the current measurement.

To implement the Kalman filter algorithm, the first step is to develop an object motion model. To do this, a candidate object is assumed to be moved on a perfectly frictionless, straight path. Initially, the candidate object is stationary, but then it is pushed by random acceleration.

The linear state space \mathbf{x}_k at k time step for object motion is expressed as follows:

$$\mathbf{x}_k = \begin{bmatrix} X_R & Y_R & \dot{X}_R & \dot{Y}_R \end{bmatrix}^T \quad (4.4)$$

where \dot{X}_R is the object velocity in X direction i.e., the derivative of object position X_R with respect to time and \dot{Y}_R is the object velocity in Y direction i.e., derivative of object position Y_R with respect to time.

Between $(k-1)$ time step and k time step, the object undergoes a constant acceleration of a_k that is normally distributed (white Gaussian noise) with mean 0 and standard deviation σ_a . From

the Newton's laws of motion of the object, the linear state space of the object can be expressed into

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}a_k \quad (4.5)$$

where \mathbf{F} is the state transition matrix given as

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{and } \mathbf{G} = \begin{bmatrix} \frac{\Delta t^2}{2} & \frac{\Delta t^2}{2} & \Delta t & \Delta t \end{bmatrix}^T$$

where Δt is the sampling time.

Eq. (4.5) can be expressed as

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k \quad (4.6)$$

where \mathbf{w}_k is the Gaussian process noise at k time step which is normally distributed (i.e., mean 0 and standard deviation σ_a) with covariance matrix \mathbf{Q} such that $\mathbf{w}_k \sim N(0, \mathbf{Q})$. The state covariance matrix \mathbf{Q} is given as,

$$\mathbf{Q} = \begin{bmatrix} \Delta t^4 / 4 & 0 & 0 & 0 \\ 0 & \Delta t^4 / 4 & 0 & 0 \\ 0 & 0 & \Delta t^2 & 0 \\ 0 & 0 & 0 & \Delta t^2 \end{bmatrix} \sigma_a^2.$$

At each time step, a measurement of the true position of the object with noise obtained from the Kinect camera sensor is given by

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (4.7)$$

where \mathbf{H} is the measurement matrix given as

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and \mathbf{v}_k is the Gaussian measurement noise vector at k time step which is normally distributed with covariance matrix \mathbf{R} such that $\mathbf{v}_k \sim N(0, \mathbf{R})$. The measurement covariance matrix \mathbf{R} is obtained as follows:

$$\mathbf{R} = \mathbf{E}[\mathbf{v}_k \mathbf{v}_k^T] = \begin{bmatrix} \sigma_p^2 & 0 & 0 & 0 \\ 0 & \sigma_p^2 & 0 & 0 \\ 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & \sigma_v^2 \end{bmatrix}$$

where σ_p and σ_v are the standard deviation in object position and standard deviation in object velocity, respectively.

In this thesis, the objective of the Kalman filter is to obtain the estimated position and velocity coordinate of the moving candidate object i.e., (\hat{X}_R, \hat{Y}_R) and $(\dot{\hat{X}}_R, \dot{\hat{Y}}_R)$ using the measurement data from the Kinect camera sensor. The Kalman filter algorithm is implemented in following two steps:

Step1: Prediction step

$$\begin{cases} \hat{\mathbf{x}}_k^- = \mathbf{F}\hat{\mathbf{x}}_{k-1} + \mathbf{w}_k \\ \mathbf{P}_k^- = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q} \end{cases} \quad (4.8)$$

where $\hat{\mathbf{x}}_k^-$ is the predicted state estimation at k time step, $\hat{\mathbf{x}}_{k-1}$ is the predicted state estimate at previous time step $k-1$, \mathbf{P}_k^- is the predicted covariance error estimation at k time step and \mathbf{P}_{k-1} is the predicted covariance error at previous time step $k-1$.

Step2: Update step

$$\begin{cases} \mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \\ \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_k^-) \\ \mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^- \end{cases} \quad (4.9)$$

where \mathbf{K}_k is the Kalman gain at k time step, $\hat{\mathbf{x}}_k$ is the updated state estimation at k time step, \mathbf{P}_k is the updated covariance matrix estimation at k time step.

Chapter 5: Controller Design Based on Backstepping Control Method

This chapter describes the walking control and the object following control of the 6LR using backstepping control method based on Lyapunov stability theory. The walking control section explains the end effector position control of the 6LR, whereas the object following control section explains the error definition, object following movement of the 6LR and a candidate object movement, and the controller design for the 6LR to perform object following task.

5.1 Walking control

In legged robot, the end effectors of each leg are the parts that are contacted to the ground. Therefore, the end effector positions of legs are crucial to the stability and movement of the legged robot. The end effector position vector of leg i with respect to body frame, \mathbf{p}_{ei}^b , can be calculated by forward kinematics when the angular position vector of each joint $\boldsymbol{\theta}_i$ is known. Usually, the inverse kinematics is used to get $\boldsymbol{\theta}_i$. But it is difficult to solve $\boldsymbol{\theta}_i$ by inverse kinematics because of its complexity as in Eqs. (3.6)~(3.9). Instead of inverse kinematics, the differential kinematics algorithm is used to solve this problem for the 6LR legs in this thesis as follows [43]:

5.1.1 End effector position control of one leg

For one leg of the 6LR, the end effector position error vector between the desired end effector position vector and the end effector position vector of leg i with respect to body frame is defined as follows:

$$\mathbf{e}_i^b = \mathbf{p}_{di}^b - \mathbf{p}_{ei}^b \quad (5.1)$$

where $\mathbf{p}_{di}^b = [x_{di}^b \ y_{di}^b \ z_{di}^b]^T$ is the desired end effector position vector with respect to body frame of leg i as shown in Fig.5.1 .

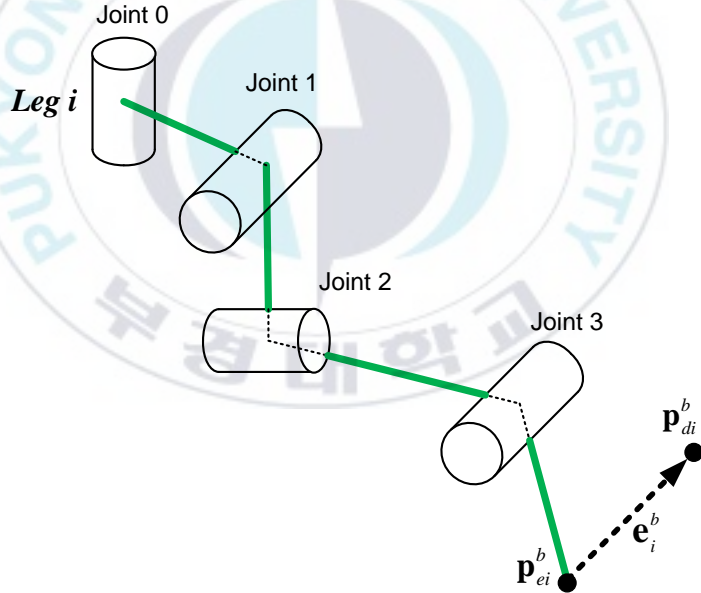


Fig. 5.1 End effector position control of leg i

In this thesis, to keep the end effector in the desired position, we only consider the Jacobian relating the translational velocity of the end effector and the joint angular velocity. The translational

velocity of the end effector can be expressed as the time derivative of the end effector position vector of leg i with respect to body frame \mathbf{p}_{ei}^b .

$$\dot{\mathbf{p}}_{ei}^b = \frac{\partial \mathbf{p}_{ei}^b}{\partial \boldsymbol{\theta}_i} \dot{\boldsymbol{\theta}}_i = \mathbf{J}_{Ai}^b(\boldsymbol{\theta}_i) \dot{\boldsymbol{\theta}}_i \quad (5.2)$$

where $\mathbf{J}_{Ai}^b(\boldsymbol{\theta}_i)$ is the analytical Jacobian of \mathbf{p}_{ei}^b with respect to $\boldsymbol{\theta}_i$ as:

$$\dot{\mathbf{p}}_{ei}^b = \frac{\partial \mathbf{p}_{ei}^b}{\partial \boldsymbol{\theta}_i} \dot{\boldsymbol{\theta}}_i = \mathbf{J}_{Ai}^b(\boldsymbol{\theta}_i) \dot{\boldsymbol{\theta}}_i \quad (5.3)$$

The time derivative of Eq. (5.1) is written as

$$\dot{\mathbf{e}}_i^b = \dot{\mathbf{p}}_{di}^b - \dot{\mathbf{p}}_{ei}^b \quad (5.4)$$

From Eq. (5.2), Eq. (5.4) can be written as follows:

$$\dot{\mathbf{e}}_i^b = \dot{\mathbf{p}}_{di}^b - \mathbf{J}_{Ai}^b(\boldsymbol{\theta}_i) \dot{\boldsymbol{\theta}}_i \quad (5.5)$$

The candidate Lyapunov function (clf) can be chosen in the following positive definite quadratic form:

$$V_i(\mathbf{e}_i^b) = \frac{1}{2} (\mathbf{e}_i^b)^T \mathbf{K} \mathbf{e}_i^b \quad (5.6)$$

where \mathbf{K} is a symmetric positive definite matrix.

This function is so that:

$$V_i(\mathbf{e}_i^b) > 0 \text{ for } \forall \mathbf{e}_i^b \neq 0, V_i(0) = 0 \quad (5.7)$$

Differentiating Eq. (5.6) with respect to time and substituting Eq. (5.5) into it, the derivative of the clf can be derived as follows:

$$\dot{V}_i(\mathbf{e}_i^b) = (\mathbf{e}_i^b)^T \mathbf{K} \dot{\mathbf{e}}_i^b = (\mathbf{e}_i^b)^T \mathbf{K} \dot{\mathbf{p}}_{di}^b - (\mathbf{e}_i^b)^T \mathbf{K} \mathbf{J}_{Ai}^b(\boldsymbol{\theta}_i) \dot{\boldsymbol{\theta}}_i \quad (5.8)$$

The leg of 6LR has 3 degrees of freedom but it has four joints. Therefore, this is a redundant manipulator case. The redundancy can be used for other additional constraint of joint configuration. An objective function of the joint variables $\omega(\boldsymbol{\theta}_i)$ can be chosen as follows [46]:

$$\omega(\boldsymbol{\theta}_i) = -\frac{1}{2} \theta_{3i}^2 \quad (5.9)$$

The additional vector $\dot{\mathbf{q}}_{0i}$ for constraint of joint configuration is defined as

$$\dot{\mathbf{q}}_{0i} = k_0 \left(\frac{\partial \omega(\boldsymbol{\theta}_i)}{\partial \boldsymbol{\theta}_i} \right)^T \quad (5.10)$$

where $k_0 > 0$.

From Eqs. (5.9) and (5.10), $\dot{\mathbf{q}}_{0i}$ can be rewritten as

$$\dot{\mathbf{q}}_{0i} = k_0 [0 \quad 0 \quad -\theta_{3i} \quad 0]^T \quad (5.11)$$

A control input vector is chosen as an angular joint velocity vector of leg i as follows:

$$\dot{\boldsymbol{\theta}}_i = (\mathbf{J}_{Ai}^b)^T (\boldsymbol{\theta}_i) \mathbf{K} \mathbf{e}_i^b + \left(\mathbf{I}_n - (\mathbf{J}_{Ai}^b)^+ \mathbf{J}_{Ai}^b \right) \dot{\mathbf{q}}_{0i} \quad (5.12)$$

where $(\mathbf{J}_{Ai}^b)^+$ is right pseudo-inverse of \mathbf{J}_{Ai}^b as

$$(\mathbf{J}_{Ai}^b)^+ = (\mathbf{J}_{Ai}^b)^T \left(\mathbf{J}_{Ai}^b (\mathbf{J}_{Ai}^b)^T \right)^{-1} \quad (5.13)$$

Substituting Eq. (5.12) into (5.8) leads to

$$\begin{aligned} \dot{V}_i(\mathbf{e}_i^b) &= (\mathbf{e}_i^b)^T \mathbf{K} \dot{\mathbf{p}}_{di}^b - (\mathbf{e}_i^b)^T \mathbf{K} \mathbf{J}_{Ai}^b (\mathbf{J}_{Ai}^b)^T \mathbf{K} \mathbf{e}_i^b \\ &\quad - (\mathbf{e}_i^b)^T \mathbf{K} \mathbf{J}_{Ai}^b \left(\mathbf{I}_n - (\mathbf{J}_{Ai}^b)^+ \mathbf{J}_{Ai}^b \right) \dot{\mathbf{q}}_0 \\ &= (\mathbf{e}_i^b)^T \mathbf{K} \dot{\mathbf{p}}_{di}^b - (\mathbf{e}_i^b)^T \mathbf{K} \mathbf{J}_{Ai}^b (\mathbf{J}_{Ai}^b)^T \mathbf{K} \mathbf{e}_i^b \\ &\quad - (\mathbf{e}_i^b)^T \mathbf{K} \left((\mathbf{J}_{Ai}^b) - (\mathbf{J}_{Ai}^b) (\mathbf{J}_{Ai}^b)^+ \mathbf{J}_{Ai}^b \right) \dot{\mathbf{q}}_0 \\ &= (\mathbf{e}_i^b)^T \mathbf{K} \dot{\mathbf{p}}_{di}^b - (\mathbf{e}_i^b)^T \mathbf{K} \mathbf{J}_{Ai}^b (\mathbf{J}_{Ai}^b)^T \mathbf{K} \mathbf{e}_i^b \\ &\quad - (\mathbf{e}_i^b)^T \mathbf{K} (\mathbf{J}_{Ai}^b - \mathbf{J}_{Ai}^b) \dot{\mathbf{q}}_0 \end{aligned} \quad (5.14)$$

If $\dot{\mathbf{p}}_{di}^b$ is constant, because the third term of Eq. (5.14) is zero, Eq. (5.8) is negative definite as follows:

$$\begin{aligned}
\dot{V}_i(\mathbf{e}_i^b) &= -(\mathbf{e}_i^b)^T \mathbf{K} \mathbf{J}_{Ai}^b (\mathbf{J}_{Ai}^b)^T \mathbf{K} \mathbf{e}_i^b \\
&= -\left[(\mathbf{J}_{Ai}^b)^T \mathbf{K} \mathbf{e}_i^b \right]^T (\mathbf{J}_{Ai}^b)^T \mathbf{K} \mathbf{e}_i^b < 0
\end{aligned} \tag{5.15}$$

The condition $\dot{V}_i < 0$ and $V_i > 0$ implies that $\mathbf{e}_i^b = 0$ when time $t \rightarrow \infty$, i.e., the system is asymptotically stable.

The matrix $\mathbf{I}_n - (\mathbf{J}_{Ai}^b)^+ \mathbf{J}_{Ai}^b$ in Eq. (5.12) allows the projection of the vector $\dot{\mathbf{q}}_0$ in the null space of \mathbf{J}_{Ai}^b , so that the condition (5.15) is not violated. It means internal motions described by $(\mathbf{I}_n - (\mathbf{J}_{Ai}^b)^+ \mathbf{J}_{Ai}^b) \dot{\mathbf{q}}_0$ can reconfigure the leg structure without changing the end effector position and orientation. The chosen controller of Eq. (5.12) makes the end effector to be in a desired position.

The joint angles are used to control the servomotors. The joint angles can be calculated in discrete time by the Euler integration method. With sampling time Δt , if the joint angles and angular velocities at time t_k are known, the joint angles at time $t_{k+1} = t_k + \Delta t$ can be computed as follows:

$$\boldsymbol{\theta}_i^{t_{k+1}} = \boldsymbol{\theta}_i^{t_k} + \dot{\boldsymbol{\theta}}_i^{t_k} \Delta t \tag{5.16}$$

Fig. 5.2 shows the block diagram of walking controller as follows:

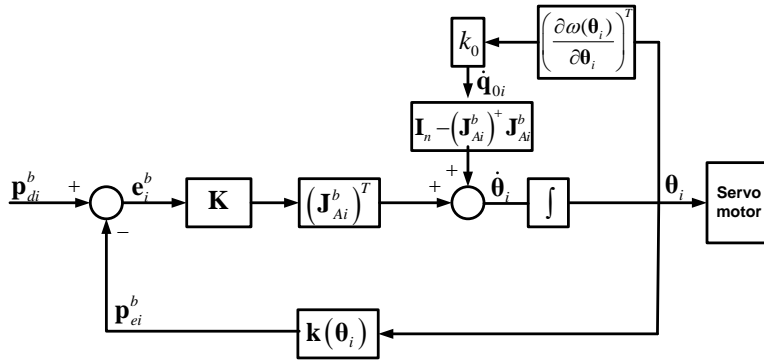


Fig. 5.2 Block diagram of walking controller

5.1.2 Translational and rotational body movement

Fig. 5.3 shows the translational and rotational body movement of the 6LR.

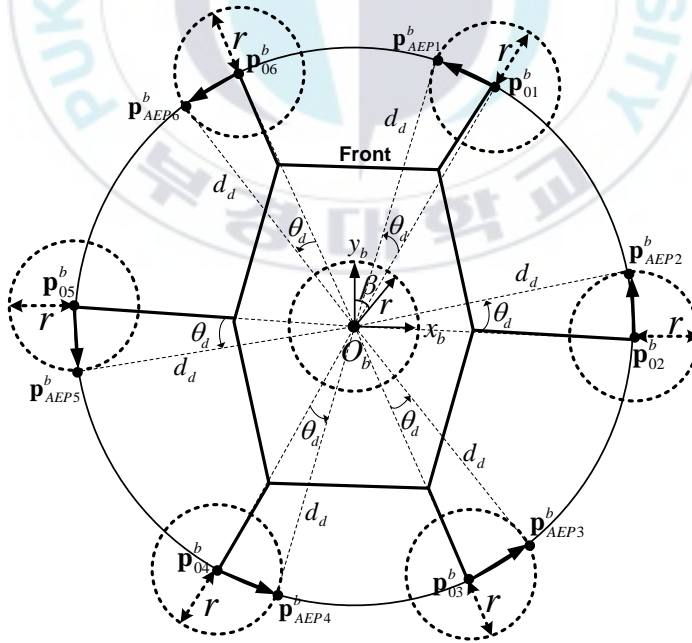


Fig. 5.3 Translational and rotational body movement of the 6LR

where

$O_b x_b y_b$: Body coordinate frame of the 6LR

$\mathbf{p}_{0i}^b = [x_{0i}^b \ y_{0i}^b \ z_{0i}^b]^T$: Default standing position vector of leg i with respect to body frame

$\mathbf{p}_{AEPi}^b = [x_{AEPi}^b \ y_{AEPi}^b \ z_{AEPi}^b]^T$: Anterior Extreme Position vector of leg i

β : Orientation angle of the 6LR

θ_d : Desired angle to achieve anterior extreme position (AEP) for each leg measured in counter-clockwise direction

r : Step distance

For each leg, to achieve AEP from the default standing position, the anterior extreme position (AEP) vector of leg i , \mathbf{p}_{AEPi}^b is calculated as follows:

$$\mathbf{p}_{AEPi}^b = \mathbf{R}_m \mathbf{p}_{0i}^b + \mathbf{T}_m \quad (5.17)$$

$$\mathbf{R}_m = \begin{bmatrix} \cos \theta_d & -\sin \theta_d & 0 \\ \sin \theta_d & \cos \theta_d & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{T}_m = \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix}$$

where

\mathbf{R}_m : Rotational movement vector with desired angle θ_d

\mathbf{T}_m : Translational movement vector with orientation angle β and step distance r

s_x : Translational movement of the end effector in x direction

s_y : Translational movement of the end effector in y direction

s_z : Translational movement of the end effector in z direction

The movement path for the end effector of each leg is designed as a set of desired points. The designed controller in the previous section can be used to control the end effector of each leg to follow the moving path. After β and θ_d are set, the AEP point is computed and the end effector trajectory is generated as shown in Fig. 5.4.

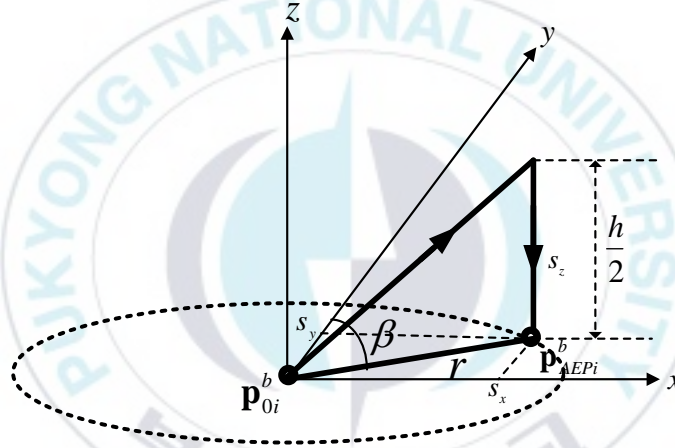


Fig. 5.4 End effector trajectory of one leg of the 6LR

From the Fig. 5.4, the components of the translational movement vector \mathbf{T}_m can be written as

$$\begin{cases} s_x = r \sin \beta \\ s_y = r \cos \beta \\ s_z = h/2 \end{cases} .$$

In this thesis, the orientation angle $\beta = 0$. The path of the end effector of each leg is designed as follows:

The swing phase is composed of two states: Diagonally lift up and Put down as shown in Fig. 5.5 with $\beta = 0$.

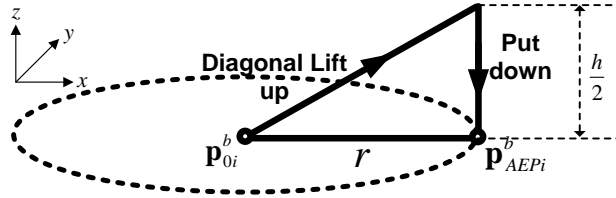


Fig. 5.5 Swing phase of leg i

The retract phase is composed of one state: Move horizontal as shown in Fig. 5.6 with $\beta = 0$.

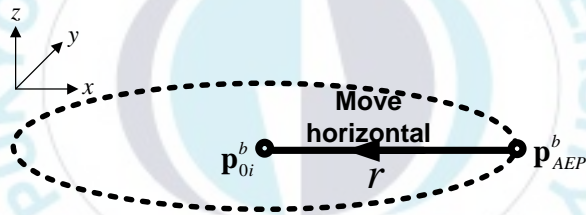


Fig. 5.6 Retract phase of leg i

Each state time in one step cycle time T is designed as shown in Fig. 5.7.

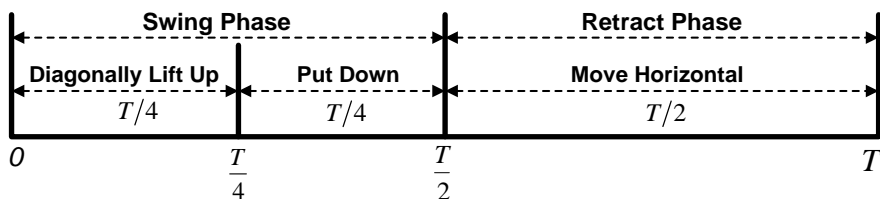


Fig. 5.7 Each state time in one cycle time

The path of the end effector is designed as a sequence of real position vector $\mathbf{p}_{di}^b(t_k)$ at time t_k for the end effector to reach AEP, where $\mathbf{p}_{di}^b(t_k) = \mathbf{R}_m(t_k)\mathbf{p}_{0i}^b(t_k) + \mathbf{T}_m(t_k)$.

In Diagonally lift up state, the end effector only moves diagonally vertical.

$$\left\{ \begin{array}{l} \theta_d(t_k) = \theta_d \left[\frac{t_k}{T/4} \right] \\ s_x(t_k) = (r \sin \beta) \left[\frac{t_k}{T/4} \right] \\ s_y(t_k) = (r \cos \beta) \left[\frac{t_k}{T/4} \right] \\ s_z(t_k) = \left(\frac{h}{2} \right) \left[\frac{t_k}{T/4} \right] \\ 0 \leq t_k \leq T/4 \end{array} \right. \quad (5.18)$$

In Put down state, the end effector only moves vertically.

$$\left\{ \begin{array}{l} \theta_d(t_k) = \theta_d \\ s_x(t_k) = r \sin \beta \\ s_y(t_k) = r \cos \beta \\ s_z(t_k) = \left(\frac{h}{2} \right) - \left(\frac{h}{2} \right) \left[\frac{t_k - (T/4)}{(T/4)} \right] \\ T/4 \leq t_k \leq T/2 \end{array} \right. \quad (5.19)$$

In Move horizontal state, the end effector only moves horizontally.

$$\left\{ \begin{array}{l} \theta_d(t_k) = \theta_d \left[\frac{(T-t_k)}{(T/2)} \right] \\ s_x(t_k) = (r \sin \beta) \left[\frac{(T-t_k)}{(T/2)} \right] \\ s_y(t_k) = (r \cos \beta) \left[\frac{(T-t_k)}{(T/2)} \right] \\ s_z(t_k) = 0 \\ T/2 \leq t_k \leq T \end{array} \right. \quad (5.20)$$

5.2 Object following control

In this section, firstly, the definition and measurement methods of the heading angle error and the distance error are explained. Secondly, an object following movement of the 6LR and a candidate object is explained. Finally, the object following controller design for the 6LR is described.

5.2.1 Heading angle error

Consider the candidate object is detected in the right side of the Kinect RGB image frame outside the desired object area as shown in Fig. 5.8.

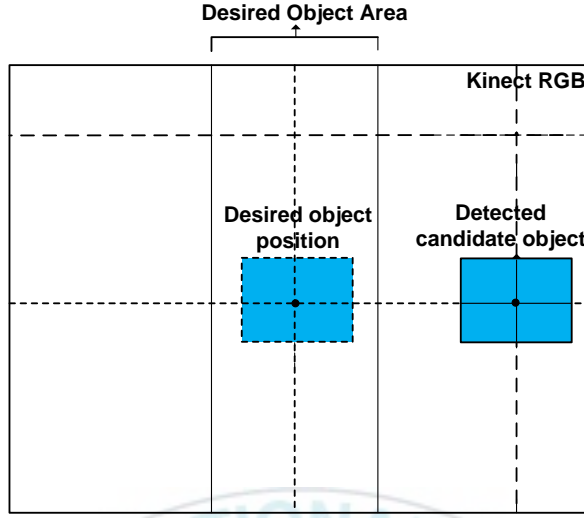


Fig. 5.8 Kinect RGB image showing detected candidate object outside the desired object area

The objective of the proposed system is to bring the candidate detected object inside the desired object area. To make the 6LR turn toward the candidate object, it should change its current angular position to the desired angular position to achieve object following. Therefore, the heading angle error e_ϕ is defined as the difference between the current angular position and the desired angular position of the candidate object as follows:

$$e_\phi = \phi_C - \phi_D \quad (5.21)$$

where ϕ_C is the current angular position of the candidate object measured from the y_b axis and ϕ_D is the desired angular position of the candidate object measured from the y_b axis as shown in Fig. 5.9.

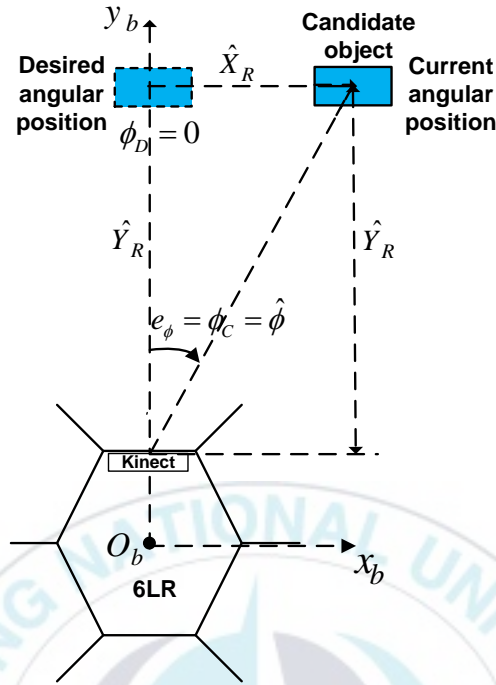


Fig. 5.9 Heading angle error definition

However, after Kalman filter is applied, the current angular position ϕ_C is the estimated angular position $\hat{\phi}$ of the moving candidate object. Therefore, from Eq. (5.21), the estimated heading angle error can be written as,

$$e_\phi = \hat{\phi} - \phi_D \quad (5.22)$$

where $\hat{\phi}$ can be calculated as follows:

$$\hat{\phi} = \arctan\left(\frac{\hat{X}_R}{\hat{Y}_R}\right) \quad (5.23)$$

where \hat{X}_R and \hat{Y}_R are the estimated real world positions of the candidate object obtained from the Kalman filter in X and Y direction, respectively.

In this thesis, $\phi_D = 0$. Therefore, Eq. 5.22 can be expressed as

$$e_\phi = \hat{\phi} \quad (5.24)$$

5.2.2 Distance error

Fig. 5.10 shows the definition of the distance error for the Kinect camera based 6LR system. The distance error e_y is defined as the difference between the current distance and the desired (reference) distance of the candidate object as follows:

$$e_y = Y_R - Y_D \quad (5.25)$$

where Y_R is the current distance of the candidate object from the 6LR and Y_D is the desired distance of the candidate object from the 6LR as shown in Fig. 5.10.

However, after Kalman filter is applied, the current distance Y_R is the estimated distance \hat{Y}_R between the candidate moving object and the 6LR. Therefore, from Eq. (5.25), the estimated distance error can be expressed as follows:

$$e_y = \hat{Y}_R - Y_D \quad (5.26)$$

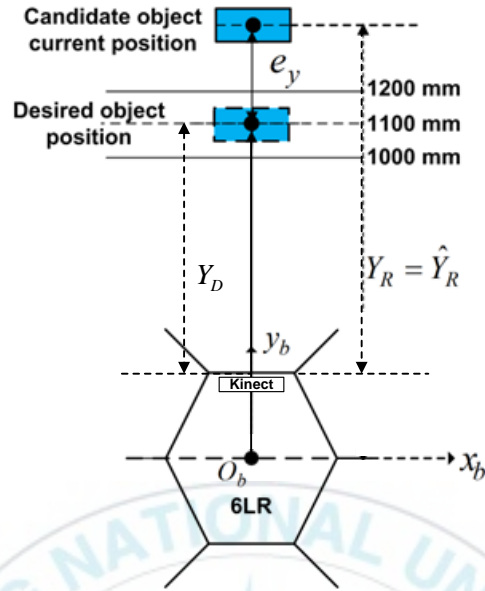


Fig. 5.10 Distance error definition

5.2.3 Object following movement of the 6LR and a candidate object movement

In this thesis, for the sake of simplicity, the movement of the 6LR and a candidate object is explained in two separate cases as follows:

5.2.3.1 Rotational following of the 6LR (without straight moving)

Fig. 5.11 shows the movements of the 6LR and a candidate object in rotational following case. In this case, a candidate object moves horizontally straight, whereas the 6LR follows the object by rotating its body based on the heading angle error angle e_ϕ without straight movement. The direction of rotation of the 6LR depends on

the direction of movement of the candidate object, ultimately depends on the value of heading angle error. For example, if the candidate object moves from left to right horizontally, the heading angle error have positive values (according to heading angle error definition) and therefore, the 6LR rotates in the clockwise direction as shown in Fig. 5.11 and vice versa. The rotational following case of the 6LR can be explained in detail as follows. Consider that the initial positions of the 6LR and the candidate object are as shown in Fig. 5.11(a). Initially, the candidate object is placed in front of the 6LR. At this instant, both the 6LR and the candidate object are at stationary positions. Fig. 5.11(b) shows the rotating position of the 6LR. In this case, the candidate object starts moving horizontally straight from left to right and thus, the value of heading angle error e_ϕ changes to positive. Therefore, the 6LR starts rotating in the clockwise direction more toward the moving candidate object trying to reduce e_ϕ . Fig. 5.11(c) shows the stop positions of the 6LR and the candidate object. In this case, the candidate object stops at the right side, however, the 6LR continues rotating until the heading angle error equal to zero i.e., $e_\phi = \hat{\phi} = 0$. As soon as the 6LR reaches this condition, further rotation of the 6LR is stopped.

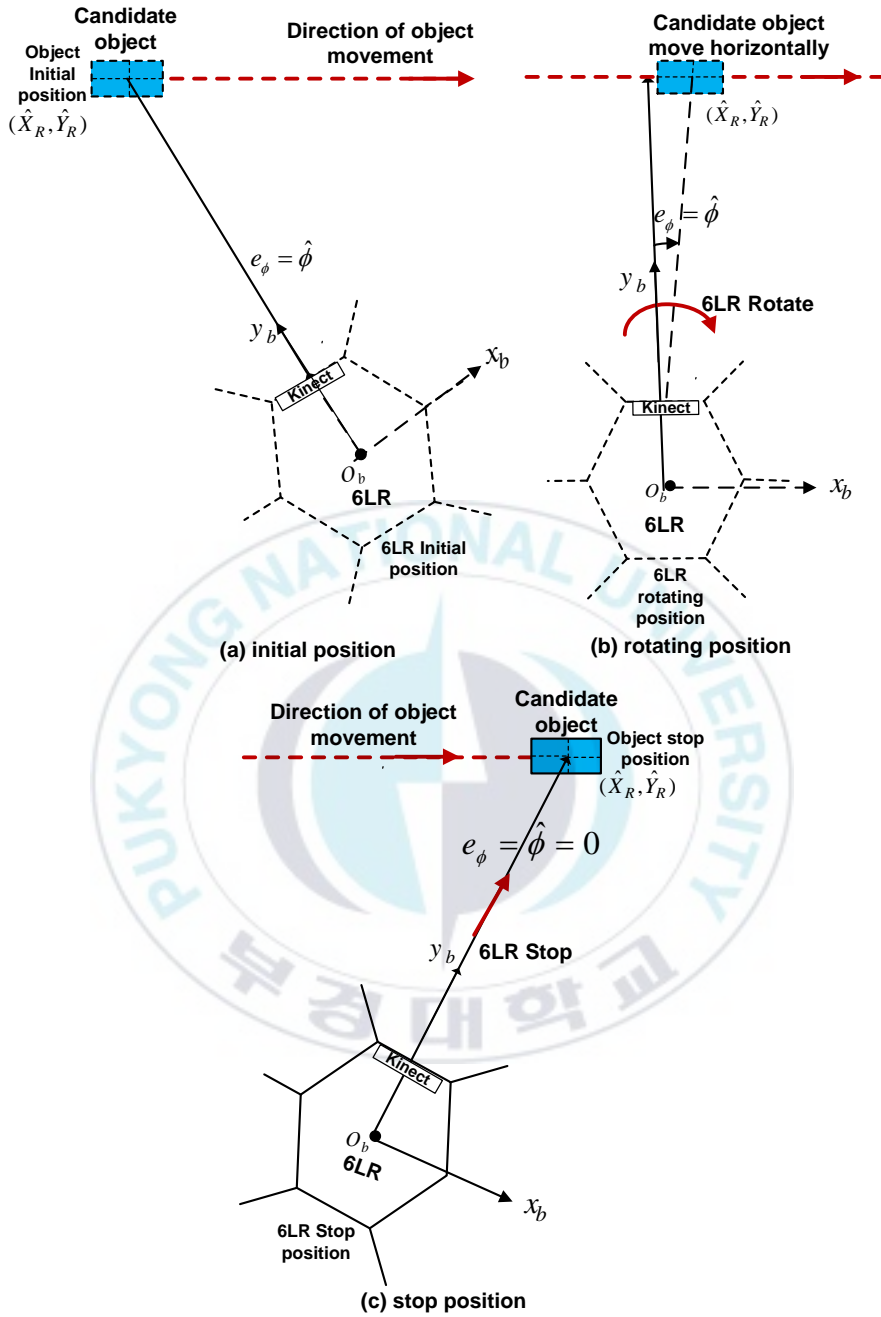


Fig. 5.11 6LR and candidate object movement in rotational following

5.2.3.2 Straight following of the 6LR (without rotation)

Fig. 5.12 shows the movements of the 6LR and a candidate object in straight following case. In this case, both the 6LR and a candidate object moves in straight forward direction along the y_b axis without rotational movement. For the straight following of the 6LR, consider a candidate object is placed at a distance more than the desired distance ($Y_D = 1100 \text{ mm}$) from the 6LR. The 6LR follows the moving candidate object until the distance between the robot and the object equal 1100 mm . In other words, the 6LR stops when the distance error becomes zero.

The straight following of the 6LR can be explained in detail as follows. Fig. 5.12(a) shows the initial positions of the 6LR and a candidate object. At this event, both the robot and the candidate object are at stationary positions. The candidate object is placed at a distance more than 1100 mm ($\hat{Y}_R > 1100 \text{ mm}$). Therefore, the distance error e_y is big. Fig. 5.12(b) shows the forward moving position of the 6LR. In this case, the candidate object starts moving in the forward direction. Therefore, the 6LR starts following the object by moving in forward direction. Due to this reason, the distance error e_y is reduced than the previous event. Fig. 5.12(c) shows the stop positions of the 6LR and the candidate object. In this case, the candidate object stops moving, however, the 6LR continues to move forward until $\hat{Y}_R = Y_D = 1100 \text{ mm}$. In other words, the 6LR stops moving when the distance error equals to zero i.e., $e_y = 0$.

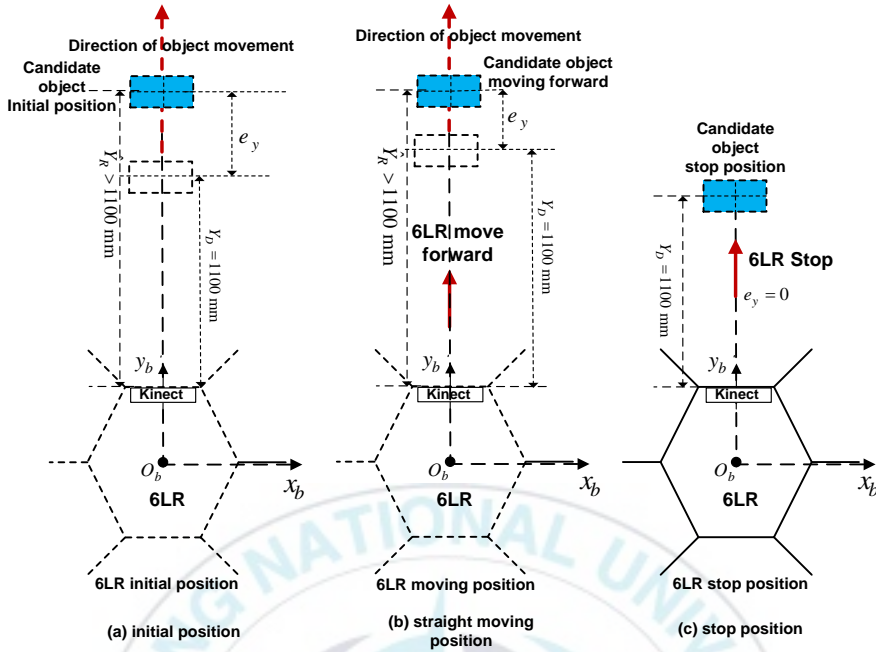


Fig. 5.12 6LR and candidate object movement in straight following

5.2.4 Angular and translational velocity controller design

This section explains the controller design for the Kinect based 6LR system to minimize the heading angle error and the distance error calculated in the previous section. In this thesis, a backstepping method is adopted to design a controller for the 6LR.

Using Eq. (5.22) and Eq. (5.26), the error vector \mathbf{e} can be written as follows:

$$\mathbf{e} = \begin{bmatrix} e_\phi \\ e_y \end{bmatrix} = \begin{bmatrix} \hat{\phi} - \phi_D \\ \hat{Y}_R - Y_D \end{bmatrix} \quad (5.27)$$

From Eq. (5.27), the time derivative of the error vector $\dot{\mathbf{e}}$ is obtained as follows:

$$\dot{\mathbf{e}} = \begin{bmatrix} \dot{e}_\phi \\ \dot{e}_y \end{bmatrix} = \begin{bmatrix} \dot{\hat{\phi}} - \dot{\phi}_D \\ \dot{\hat{Y}}_R - \dot{Y}_D \end{bmatrix} = \begin{bmatrix} \hat{\omega} - \omega_D \\ \hat{v} - v_D \end{bmatrix} \quad (5.28)$$

where $\dot{\hat{\phi}} = \hat{\omega}$ and $\dot{\hat{Y}}_R = \hat{v}$ are the estimated angular velocity and estimated translational velocity of the 6LR obtained from the Kalman filter, and $\dot{\phi}_D = \omega_D$ and $\dot{Y}_D = v_D$ are the desired angular velocity and desired translational velocity of the 6LR.

The stability of the system is checked by Lyapunov stability theory. First, a candidate Lyapunov function (clf) is chosen as follows:

$$V = \frac{1}{2} \mathbf{e}^T \mathbf{e} = \frac{1}{2} e_\phi^2 + \frac{1}{2} e_y^2 \quad (5.29)$$

Stability checking is used to find a control law U that can make the system stable by converging all errors to zero. The time derivative of the clf is obtained as follows:

$$\begin{aligned} \dot{V} &= \dot{e}_\phi e_\phi + \dot{e}_y e_y \\ &= (\hat{\omega} - \omega_D) e_\phi + (\hat{v} - v_D) e_y \end{aligned} \quad (5.30)$$

According to Lyapunov stability criterion, the system is stable if $\dot{V} \leq 0$. Therefore, to meet this condition, a control law U can be chosen as follows:

$$U = \begin{bmatrix} \hat{\omega} \\ \hat{v} \end{bmatrix} = \begin{bmatrix} \omega_D - K_1 e_\phi \\ v_D - K_2 e_y \end{bmatrix} \quad (5.31)$$

where K_1 and K_2 are positive constants.

Substituting Eq. (5.31) to Eq. (5.30) yields

$$\dot{V} = -K_1 e_\phi^2 - K_2 e_y^2 \leq 0 \quad (5.32)$$

This means that the value \dot{V} is always negative by using the chosen control law U . Furthermore, $\dot{V} = 0$ from Barbalat's lemma, therefore, $e_\phi, e_y \rightarrow 0$ as $t \rightarrow \infty$ which indicates that the system is asymptotically stable.

The block diagram of the proposed system is shown in Fig. 5.13.

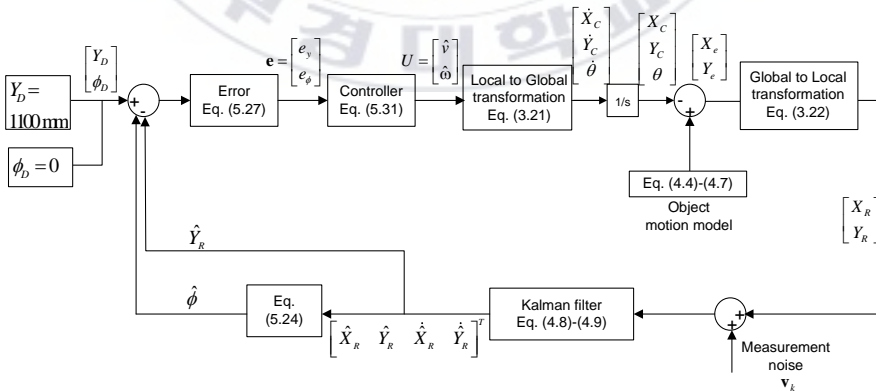


Fig. 5.13 Block diagram of the proposed system

Chapter 6: Simulation and Experimental Results

To verify the effectiveness and performance of the proposed object following control algorithm for the 6LR, simulations and experiments are conducted. In this thesis, for the sake of simplicity, two cases of the object following 6LR are explained separately as follows:

- Rotational following of the 6LR (without straight moving)
- Straight following of the 6LR (without rotational moving)

The parameters and initial values for the simulation and experiment of the Kalman filter and color-based object detection algorithm are listed in Table 6.1.

Table 6.1 Parameters and initial values of the Kalman filter and color-based object detection algorithm

Parameter	Values	Units
Δt	50	ms
σ_a	10	mm/s ²
σ_v	1	mm/s
σ_p	10	mm
α	5.2	μm
f	6.1	mm

6.1 Rotational following of the 6LR (without straight moving)

In the rotational following case of the 6LR, a blue colored candidate object moves horizontally from left to right and the 6LR follows the moving object by rotating its body without straight

movement. In this case, a moving candidate object stops after few seconds of movement, however, the 6LR keeps following the candidate object by rotating its body toward the object until the heading angle error ($e_\phi = 0$) becomes zero. The parameters and initial values used for the simulation of rotational following case are given in Table 6.2.

Table 6.2 Parameter and initial values of the rotational following simulation

Description	Symbols	Values
Initial object position*	$X_o(0)$	-1000 mm
Initial object position [#]	$Y_o(0)$	3000 mm
Initial object velocity*	$\dot{X}_o(0)$	20 mm/s
Initial object velocity [#]	$\dot{Y}_o(0)$	0 mm/s
Initial 6LR position*	$X_c(0)$	0 mm
Initial 6LR position [#]	$Y_c(0)$	0 mm
Initial 6LR angular position	$\theta(0)$	1.92 rad
Controller gain 1	K_1	1 s ⁻¹

* represents in X direction, [#] represents in Y direction

Fig. 6.1 shows the schematic of the initial positions of the 6LR and a candidate object in the rotational following case.

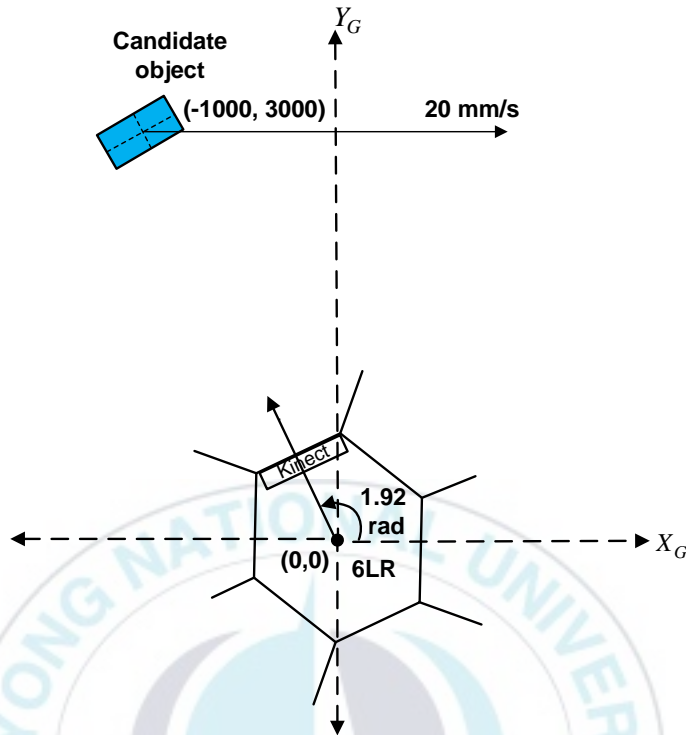


Fig. 6.1 Schematic of the initial positions of the 6LR and a candidate object in rotational following case

The simulation and experimental results of the rotational following case of 6LR are compared and are shown in Figs. 6.2~6.4. Fig. 6.2 and Fig. 6.3 show the estimated angular velocity and the heading angle error of the 6LR when the robot is turning toward the moving candidate object. Fig. 6.2 shows that in the simulation, the initial estimated angular velocity of the 6LR is -0.012 rad/s and then it is decreased to -0.0645 rad/s until 7 s. At 7 s, the candidate object stops moving. Therefore, the estimated angular velocity increases and then converges to zero at 17 s. In the experimental result, initially, the estimated angular velocity is -0.011 rad/s and then it is decreased to -0.0638 rad/s until 7 s. At 7 s, as the candidate object stops moving,

the estimated angular velocity is increased and finally bounded within a small value of ± 0.0005 rad/s along the simulation result. In this case, it should be noted that when a candidate object moves from left to right direction, the 6LR rotates in the clockwise direction for the robot to follow the moving candidate object. In this thesis, the negative values of the estimated angular velocity of the robot represent the clockwise movement of the robot. Fig. 6.3 shows the estimated heading angle error of the 6LR. In the simulation result, initially the estimated heading angle error is 0.027 rad and then it is increased to 0.13 rad until 7 s. At 7 s, the candidate object stops moving. Therefore, the heading angle error is decreased and finally converges to zero at 17 s. In the experimental result, the estimated heading angle error is increased to 0.142 rad until 7 s. At 7 s, as the candidate object stops moving, the heading angle error is decreased and finally bounded within a small value of ± 0.0015 rad along the simulation result. From both Figs. 6.3~6.4, it can be observed that, when the estimated heading angle error of the 6LR increases, the estimated angular velocity of the robot decreases so that the robot can follow the candidate object properly.

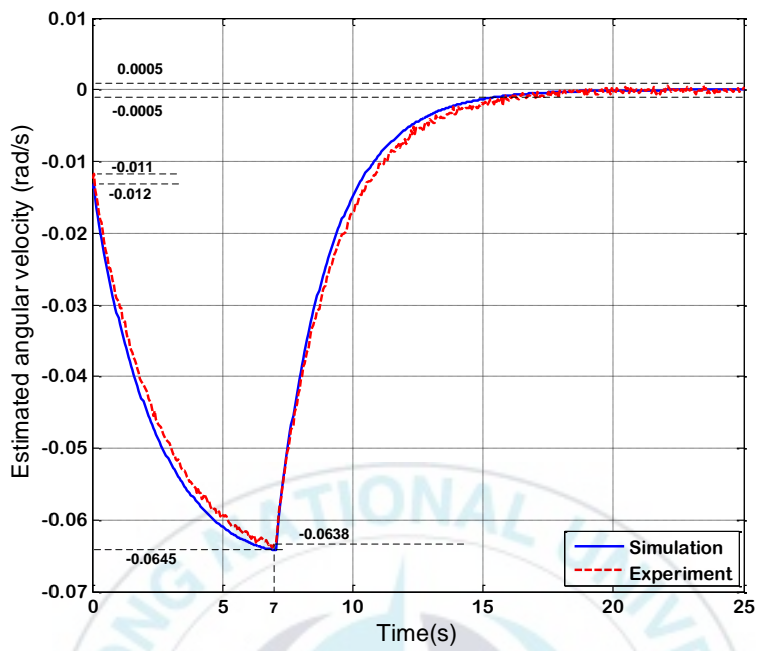


Fig. 6.2 Estimated angular velocity of the 6LR

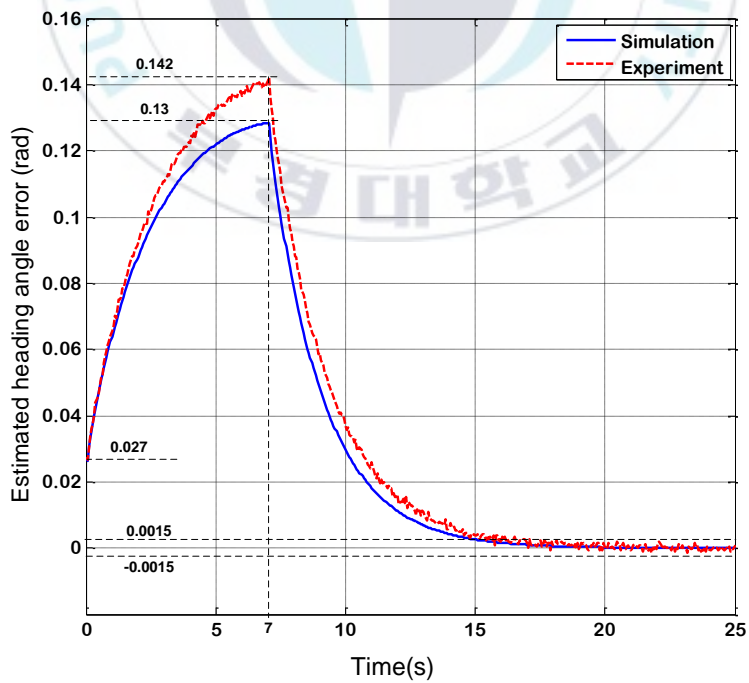


Fig. 6.3 Estimated heading angle error of the 6LR

The movement of the candidate object from left to right direction in the perspective of the 6LR during the rotational following case of the 6LR can be observed from the Kalman filter results. Fig. 6.4 shows the simulation and experimental results of the estimated values related to the object movement (from left to right) such as \hat{X}_R , \hat{Y}_R , $\dot{\hat{X}}_R$, $\dot{\hat{Y}}_R$ and $\hat{\phi}$ obtained from the Kalman filter during the rotational following case. Fig. 6.4(a) shows the estimated position \hat{X}_R of the candidate object in X direction in the simulation and experiment. It is increased to about 387 mm and 440 mm until 7 s in the simulation and experiment, respectively. At 7 s, the candidate object stops moving and hence \hat{X}_R is decreased and then reached to zero until 15 s. Fig. 6.4(b) shows the estimated position \hat{Y}_R of the candidate object in Y direction. The initial value of \hat{Y}_R is about 3175 mm and then it decreased to 2980 mm at 5 s and then increased to 3000 mm until 7 s. At 7 s, as the candidate object stops moving. Therefore, \hat{Y}_R remains constant at about 3040 mm in both simulation and experimental results. Fig. 6.4(c) shows the estimated velocity $\dot{\hat{X}}_R$ of the candidate object in X direction. It can be observed that, $\dot{\hat{X}}_R$ is increased to 151 mm/s and then decreased to zero until 7 s. At 7 s, as the object stops moving, $\dot{\hat{X}}_R$ appeared to be negative from the 6LR perspective with a value of -185 mm/s and finally reaches to zero in the simulation, whereas in the experiment it is bounded to a small value of ± 10 mm/s along the simulation result. Fig. 6.4(d) shows the estimated velocity $\dot{\hat{Y}}_R$ of the candidate object in Y direction. Initially, it is decreased to -56.75 mm/s and then increased

to about 38.35 mm/s until 7 s and finally converged to zero in the simulation result, whereas in the experiment, it is bounded to ± 10 mm/s along the simulation result. Fig. 6.4(e) shows the estimated angular position $\hat{\phi}$ of the candidate object. It is initially increased to about 0.13 rad and 0.142 rad until 7 s in the simulation and experiment, respectively. At 7 s, as the candidate object stops, $\hat{\phi}$ is decreased and then converged to zero in the simulation, whereas in the experiment it is bounded within a small value of ± 0.0015 rad along the simulation result.

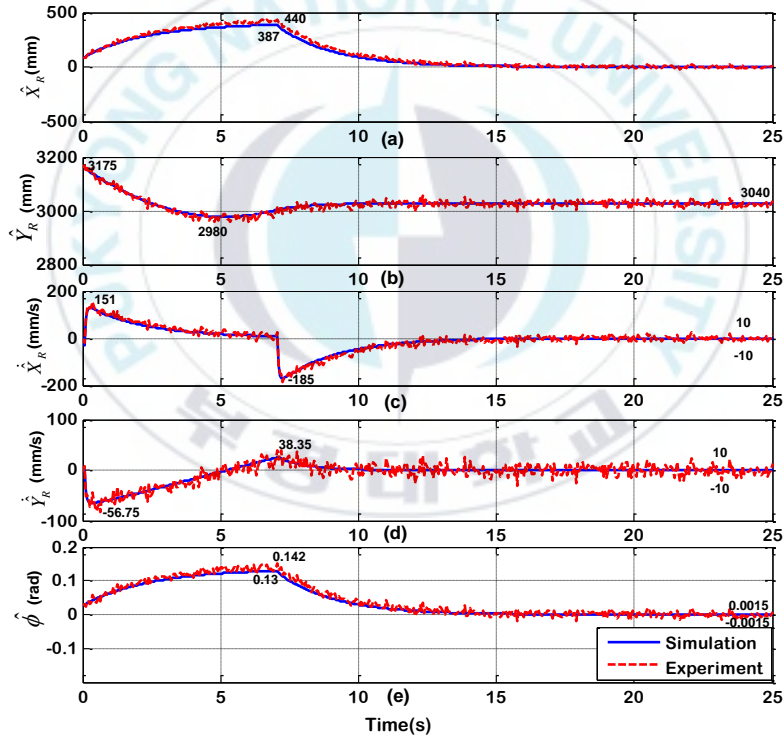


Fig. 6.4 Kalman filter results of a moving candidate object during rotational following case of the 6LR

Fig. 6.5 shows the images showing the sequence of events occurred during the experiment of the rotational following case of the 6LR. Fig. 6.5(a) shows the starting position of the 6LR in which it is heading toward the detected candidate object. The corresponding RGB image shows that the detected blue colored candidate object is inside the desired area and this is the initial position of the candidate object. Figs. 6.5(b)-6.5(e) show the movements of the candidate object toward the right hand side. In these events, the 6LR is changing its heading angle from its initial position more toward the right hand side to follow the moving candidate object. The corresponding RGB images show that the positions of the detected object are moving more toward the outside of the desired area since the object is moving more toward the right hand side. Fig. 6.5(f) shows the stop position of the object at the extreme right hand side. The fact that the candidate object stops at 7 s can be observed from the Fig. 6.4. When the object stops, the 6LR continues to rotate toward the object until the robot heads exactly toward the object. At this instant, the 6LR stops as shown in Fig. 6.5(f). The corresponding RGB shows that the object image is exactly in the center of the desired area. It can be concluded that the 6LR follows the object until the image of the detected object reaches the center of the desired area.

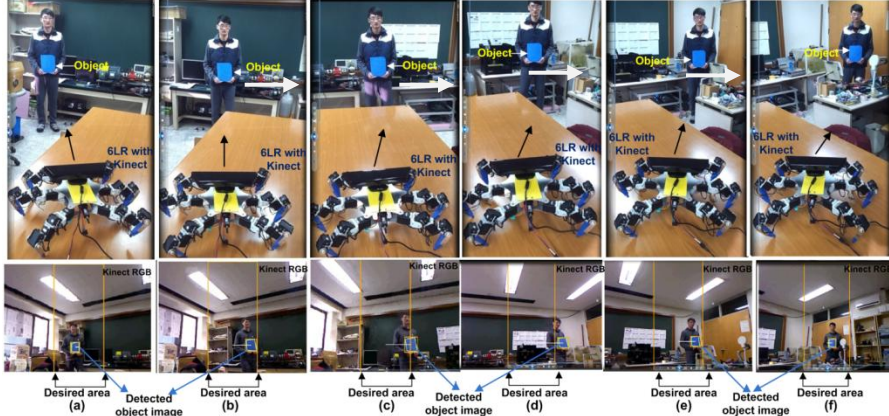


Fig. 6.5 Images showing sequence of events occurred during rotational following case of the 6LR and a candidate object

6.2 Straight following of the 6LR (without rotation)

In the straight following case of 6LR, a blue colored candidate object moves in the forward straight direction, and the 6LR follows the object by moving its body in the forward straight direction without rotational movement. In this case, a moving candidate object stops after few seconds of movement, however, the 6LR keeps following the candidate object until the distance error ($e_y = 0$) becomes zero. The parameters and initial values used for the simulation of the straight following case are given in Table 6.3.

Table 6.3 Parameters and initial values of the straight following simulation

Description	Symbols	Values
Initial object position*	$X_o(0)$	0 mm
Initial object position [#]	$Y_o(0)$	1500 mm
Initial object velocity*	$\dot{X}_o(0)$	0 mm/s

Initial object velocity [#]	$\dot{Y}_o(0)$	40 mm/s
Initial 6LR position [*]	$X_C(0)$	0 mm
Initial 6LR position [#]	$Y_C(0)$	0 mm
Initial 6LR angular position	$\theta(0)$	$\pi/2$ rad
Controller gain 2	K_2	0.05 s^{-1}

^{*} represents in X direction, [#] represents in Y direction

Fig. 6.6 shows the schematic of the initial positions of the 6LR and a candidate object in the straight following case.

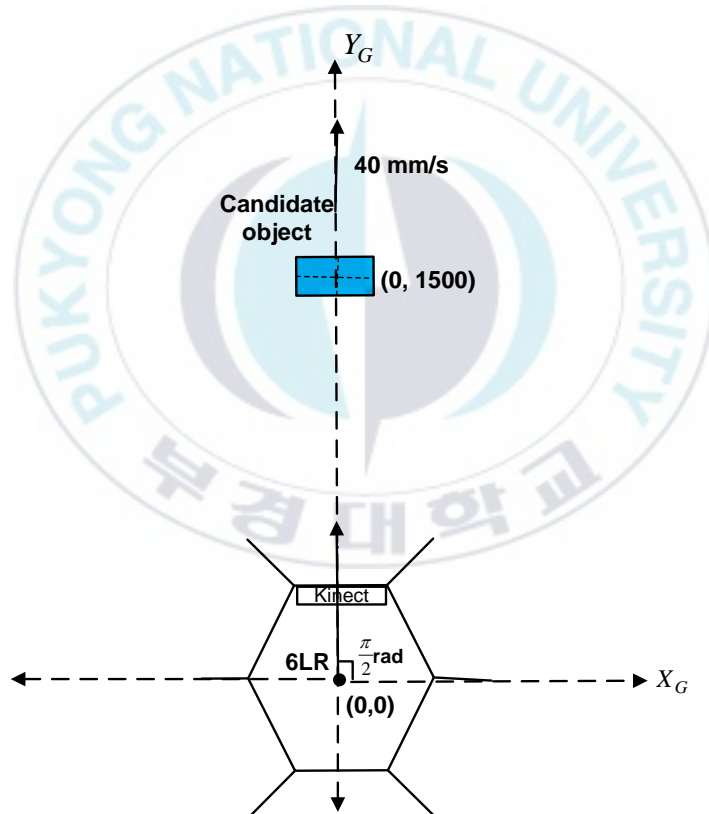


Fig. 6.6 Schematic of the initial positions of the 6LR and a candidate object in straight following case

The simulation and experimental results of the straight following case of the 6LR are compared and are shown in Figs. 6.7~6.9. Fig. 6.7 shows the simulation results of the positions of the 6LR and a candidate object in the global coordinate during the straight following case. In this case, initially, both the 6LR and the candidate object move until 35 s. However, at 35 s, the candidate object stops moving. The 6LR continues to follow the candidate object until the distance between them is equal to the reference distance i.e., 1100 mm. Figs. 6.7(a)~(b) show that the initial positions of the robot and the candidate object are 0 mm and 1500 mm, respectively. Fig. 6.7(c) shows the desired position of the 6LR where the 6LR maintains the constant reference distance of 1100 mm from the candidate object. It can be observed that at 35 s, when the candidate object stops moving at a distance of 2900 mm, the 6LR continues to move reducing the distance between the robot and the object and finally reaches to the desired position at about 95 s. Fig. 6.8 and Fig. 6.9 show the estimated translational velocity and estimated distance error of the 6LR when the robot moves forward straight toward the moving candidate object. Fig. 6.8 shows that in the simulation, the maximum estimated translational velocity of the 6LR is 60 mm/s and is decreased to 44.5 mm/s until 35 s. At 35 s, as the candidate object stops moving, the estimated velocity is decreased to 3.3 mm/s and then converged to zero until 95 s. In the experimental result, the maximum estimated translational velocity of the 6LR is 55.1 mm/s and is decreased to 43.5 mm/s at 35 s. After 35 s, as the object stopped, it is decreased to 3.91 mm/s and then finally converged to a small value of 0.5 mm/s at 95 s. Fig. 6.9 shows the estimated distance error of the 6LR. In both simulation and

experimental result, initially the estimated distance error is 400 mm. In the simulation result, the estimated distance error is decreased to 70 mm until 35 s as the object is stopped at 35 s and then converged to a very small value of 3.5 mm at 95 s. In the experimental result, the estimated distance error is decreased to 98 mm until 35 s and then finally bounded within a small value of 8 to 12 mm at 95 s along the simulation result.

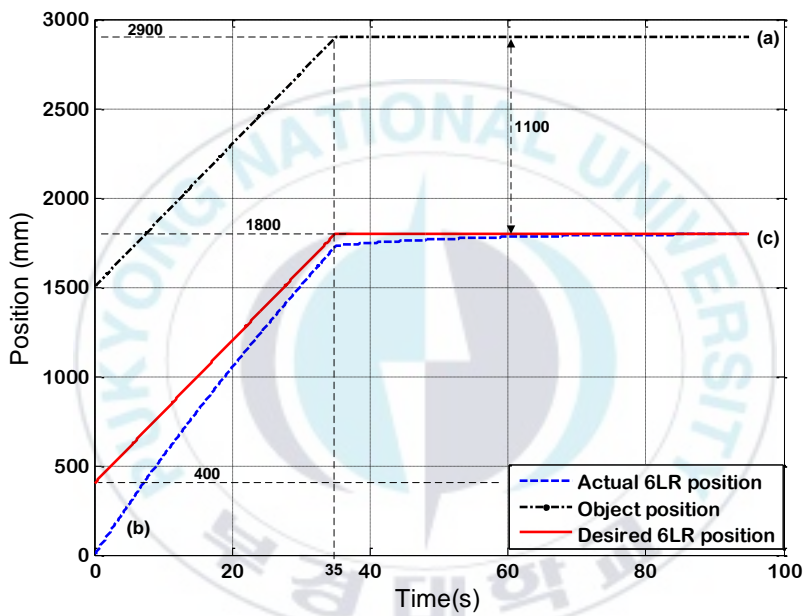


Fig. 6.7 Positions of the 6LR and a candidate object in the global coordinate during straight following case of the 6LR

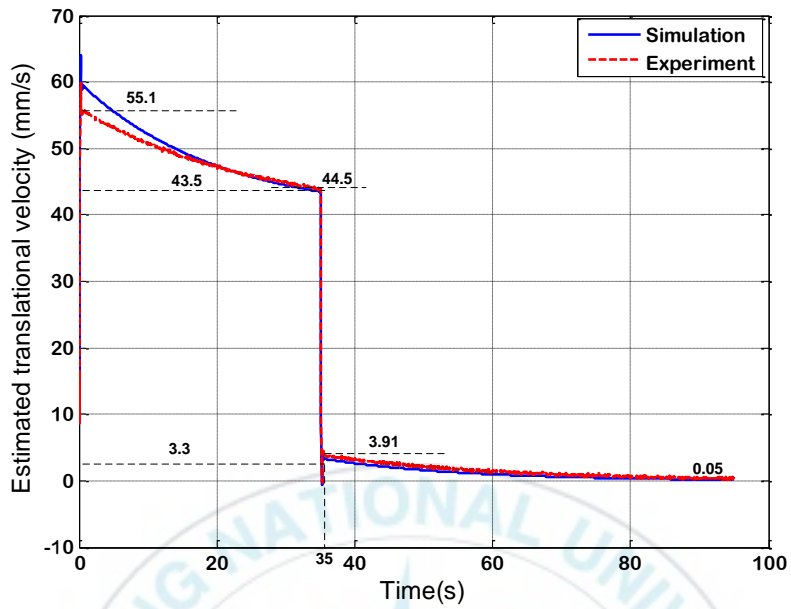


Fig. 6.8 Estimated translational velocity of the 6LR

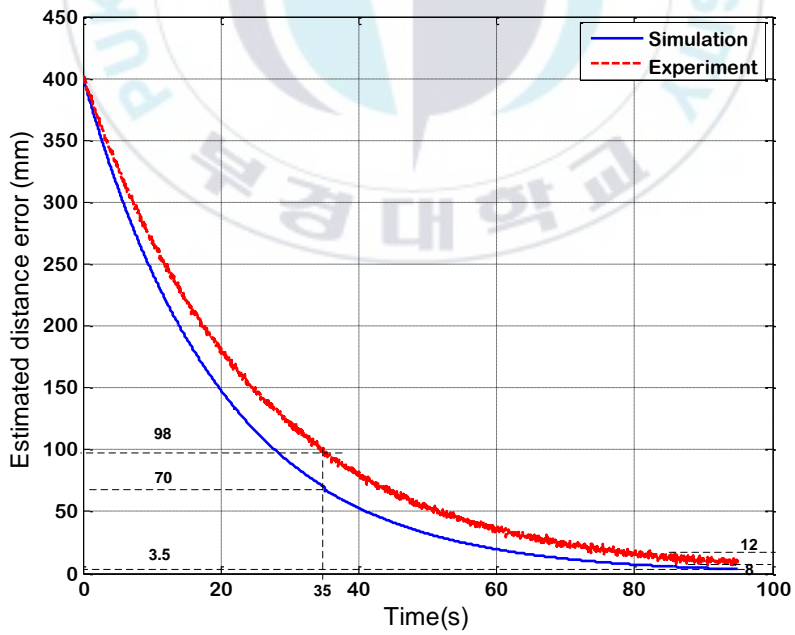


Fig. 6.9 Estimated distance error of the 6LR

The movement of the candidate object in the forward straight direction in the perspective of the 6LR during the straight following case of the 6LR can be observed from the Kalman filter results. Fig. 6.10 shows the simulation and experimental results of the estimated values of the candidate object movement such as \hat{X}_R , \hat{Y}_R , $\dot{\hat{X}}_R$, $\dot{\hat{Y}}_R$ and $\hat{\phi}$ obtained from the Kalman filter during the straight following case. Fig. 6.10(a) shows the estimated position \hat{X}_R of the candidate object in X direction. The value of \hat{X}_R is almost zero since the robot heading is exactly toward the candidate object. Fig. 6.10(b) shows the estimated position \hat{Y}_R of the candidate object in Y direction. The initial value of \hat{Y}_R is 1500 mm and then it is decreased to 1166 mm and 1196 mm until 35 s and finally converged to 1100 mm (i.e., the reference distance) and 1110 mm in the simulation and experimental result, respectively. Fig. 6.10(c) shows the estimated velocity $\dot{\hat{X}}_R$ of the candidate object in X direction which is almost zero throughout the simulation and the experiment. Fig. 6.10(d) shows the estimated velocity $\dot{\hat{Y}}_R$ of the candidate object in Y direction. In the simulation result, initially, $\dot{\hat{Y}}_R$ is -22 mm/s and then it is increased until 35 s and finally converged to zero at 95 s. In the experimental result, initially, $\dot{\hat{Y}}_R$ is -18 mm/s and then it is increased until 35 s and finally converged to very small value of -0.5 mm/s at 95 s. Fig. 6.10(e) shows the estimated angular position $\hat{\phi}$ of the candidate object. In the simulation, the value of $\hat{\phi}$ is 1.571 rad and is constant throughout the simulation whereas in the experiment, $\hat{\phi}$ is bounded within 1.5706~1.571 rad throughout the experimental result.

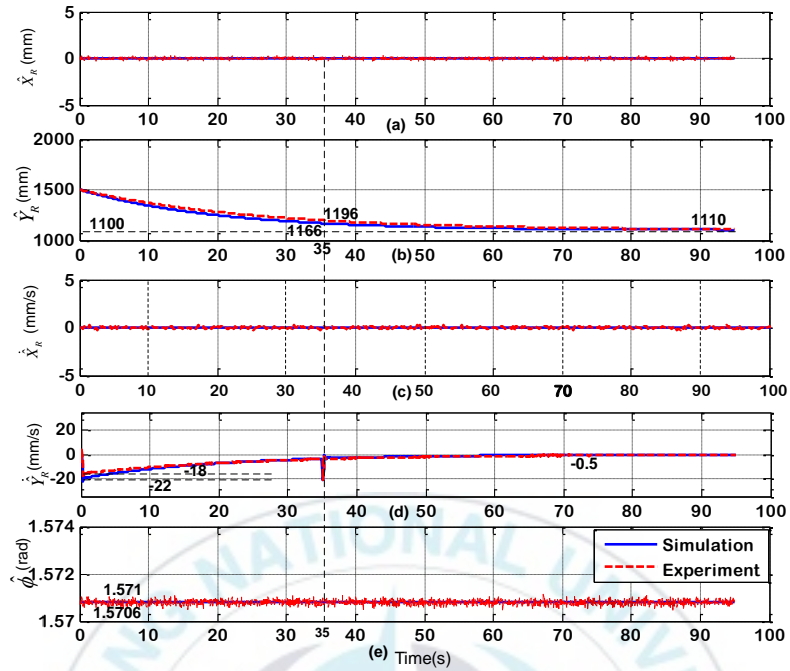


Fig. 6.10 Kalman filter results of a moving candidate object during straight following case of the 6LR

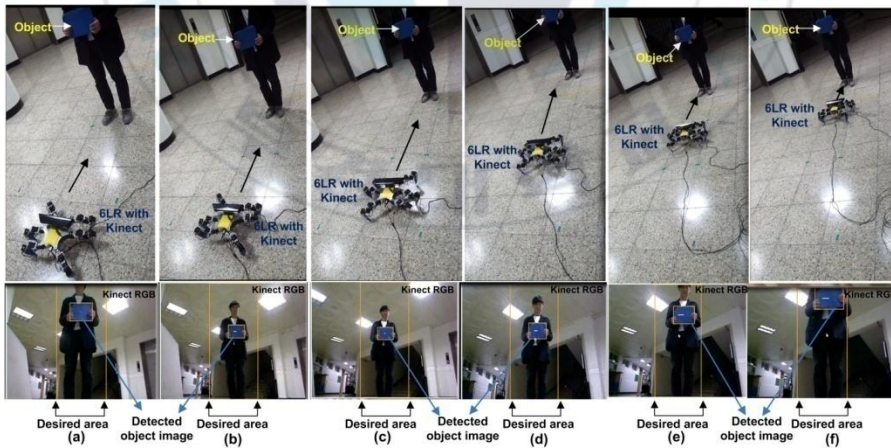


Fig. 6.11 Images showing sequence of events occurred during straight following case of the 6LR and a candidate object

Fig. 6.11 shows the images showing the sequence of events occurred during the experiment of the straight following case of the 6LR. Fig. 6.11(a) shows the initial position of the 6LR in which it is heading straight toward the detected candidate object at a distance of 1500 mm from the 6LR. The corresponding RGB image shows that the detected candidate object is inside the desired area and the detected object image is big. Figs. 6.11(b)-6.11(e) show the straight forward moving steps of the 6LR following the forward moving candidate object. The corresponding RGB images show the positions of the detected moving object inside the desired area. It can be observed that as the 6LR comes closer to the object, the size of the detected object image inside the desired area becomes bigger and vice versa. Fig. 6.11(f) shows the stop positions of the candidate object and the 6LR. In this event, the object stops at 35 s from its initial position, however, the 6LR continues to move forward until the distance between the 6LR and the object becomes equal to the reference distance (i.e., 1100 mm). The corresponding RGB image shows that the size of the detected object image inside the desired area is bigger than that of all the previous image sequences.

Chapter 7: Conclusions and Future Works

7.1 Conclusions

This thesis is about a development of a vision-based object following control of a six-legged walking robot (6LR) using Kinect camera sensor. The conclusions of this thesis are summarized as follows:

- Typical theories of 6LR were introduced. The walking gait and the stability of the 6LR were stated. The tripod walking gait was chosen for the 6LR. The walking gait cycle of the tripod gait was presented. Tripod gait was designed with swing phase and retract phase.
- A kinematic modeling of the 6LR was presented. The kinematic of one leg with four joints and four links was obtained using DH convention. The end effector positions were derived from the kinematic modeling. The control inputs of the 6LR for object following control were obtained using kinematic modeling.
- A color-based object detection algorithm was developed using Aforge.NET C# framework and applied to RGB images obtained from the Kinect camera sensor to detect the center coordinate of a blue colored candidate object. This pixel coordinate was then converted to real world coordinate in mm using Kinect depth data and simple trigonometry. A Kalman filter algorithm was used to estimate the real world position coordinate and velocity coordinate of the moving candidate object.
- Controllers for the 6LR were designed based on the backstepping method using Lyapunov stability criterion. A walking controller

for one leg was obtained to control the end effector position of one leg to track the desired end effector position. An object following controller was designed to make the 6LR follow the moving candidate object.

- The real 6LR was developed with six four joint legs to implement the proposed controllers. A Kinect camera sensor was installed on the top front of the 6LR to perform image processing. The robot was controlled by computer and microcontroller. The Kinect camera captured RGB images and depth data to detect a candidate object. The host computer computed both image processing algorithm and control algorithm, and sent commands to the microcontroller via Bluetooth. The microcontroller controlled the servomotors by providing computed joint angles to each motor through half duplex communication. The control commands were given to the 6LR through the Graphical User Interface (GUI) built in the host computer. The GUI was also displayed the image processing and tracking results to users.
- Simulations and experiments were conducted to verify the effectiveness and performances of the mathematical modeling and the designed controllers for the 6LR. For the object following control, two cases were considered: rotational following case and straight following case. In the rotational following case, the 6LR followed the horizontally moving candidate object well by rotating its body in the clock-wise direction. In this case, the estimated heading angle error was bounded to $e_\phi = \pm 0.0015$ rad.
- In the rotational following case, the 6LR followed the forward straight moving candidate object well by moving its body in the

forward straight direction along the candidate object movement. In this case, the estimated distance error was bounded to $e_y = 8$ to 12 mm.

- The simulation and experimental results showed that the proposed algorithm made the 6LR detect and follow the moving candidate object very well with an acceptable error.

7.2 Future works

This thesis has presented a development of an object following 6LR system using Kinect camera sensor. However, the modification and development for upgrading and extending the system can be done totally. There are some ideas that will be considered as future works:

- Because the current system uses a color-based object detection algorithm, it has some limitations if similar color objects appear in the field of view. Therefore, to deal with this problem, a shape detection algorithm based on Kinect depth data can be developed for the system to make the image processing algorithm comparatively stable for similar colored objects.
- The proposed controller considered only kinematic modeling. Since our robot uses servomotors, it is difficult to apply a dynamic modeling and a dynamic controller. The dynamic modeling of the 6LR should be stated to improve the control performance. With dynamic modeling, the mass of the robot, the reaction force on each leg and other parameters can be considered to design a better controller for the robot.

- The simulations and experiments were performed on the flat terrain and the rough terrain condition is not considered. However, the walking controller design of the 6LR on rough terrain can be an interesting topic for future work.



References

- [1] J. C. Wang, Y. T. Lin, H. T. Jheng, J. S. Wu and R. J. Li, “*Object Tracking for Autonomous Biped Robot*”, Proceedings of the 8th World Congress on Intelligent Control and Automation, pp. 1782-1789, 2010.
- [2] K. Baba, A. Kawamura, N. Motoi and Y. Asano, “*A Prediction Method Considering Object Motion for Humanoid Robot with Visual Sensor*”, Proceedings of the 12th International Workshop on Advanced Motion Control (AMC), pp. 320-325, 2014.
- [3] Y. Asano and A. Kawamura, “*Stability on Orientation Motion of Biped Walking Robot Aiming at a Target Object*”, Proceedings of the 9th IEEE Workshop on Advanced Motion Control (AMC), pp. 428-432, 2006.
- [4] A. Pretto, E. Menegatti and E. Pagello, “*Reliable Features Matching for Humanoid Robots*”, Proceedings of the 2007 7th IEEE/RAS International Conference on Humanoid Robots, pp. 532-538, 2007.
- [5] P. Michel, J. Chestnutt, S. Kagami, K. Nishiwaki, J. Kuffner and T. Kanade, “*GPU Accelerated Real-Time 3D Tracking for Humanoid Locomotion and Stair Climbing*”, Proceedings of the 2007 IEEE/RAS International Conference on Intelligent Robots and Systems, pp. 463–469, 2007.
- [6] H. B. Suay and S. Chemova, “*Humanoid Robot Control Using Depth Camera*”, Proceedings of the 6th International Conference on Human-Robot Interaction, pp. 401, 2011.
- [7] Q. Chen, H. Xie and P. Woo, “*Vision-Based Fast Object Recognition and Distances Calculation of Robots*”,

- Proceedings of the 31st Annual Conference of IEEE Industrial Electronics Society (IECON), pp. 363–368, 2005.
- [8] K. Cho, S. H. Baeg and S. D. Park, “*3D Pose and Target Position Estimation for a Quadruped Walking Robot*”, Proceedings of the 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp. 466-467, 2013.
- [9] Q. Chen, F. Tan and P. Y. Woo, “*An Improved Distance Measurement Method for Four-Legged Robots Based on the Color Region*”, Proceedings of the 7th World Congress on Intelligent Control and Automation, pp. 3040-3044, 2008.
- [10] M. Fujita and K. Kageyama, “*A Proposal of a Quadruped Robot Platform for Robocup*”, Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 3, pp. PS2-PS3, 1997.
- [11] D. Pack, “*Perception-Based Control for a Quadruped Walking Robot*”, Proceedings of the 1996 International Conference on Robotics and Automation, pp. 2994-3001, 1996.
- [12] M. Sridharan, G. Kuhlmann and P. Stone, “*Practical Vision-Based Monte Carlo Localization on a Legged Robot*”, Proceedings of the 2005 International Conference on Robotics and Automation, pp. 3366-3371, 2005.
- [13] F. L. Tong and M. Q. H. Meng, “*Localization of Legged Robot with Single Low-Resolution Camera Using Genetic Algorithm*”, Proceedings of the 2007 International Conference on Integration Technology, pp. 161-166, 2007.
- [14] O. Zoidi, N. Nikolaidis, A. Tefas and I. Pitas, “*Stereo Object Tracking with Fusion of Texture, Color and Disparity*

- Information*”, Journal of Image Processing: Image Communication, Vol. 29, No. 5, pp. 573-589, 2014.
- [15] A. Chillian and H. Hirschmuller, “*Stereo Camera Based Navigation of Mobile Robots on Rough Terrain*”, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4571-4576, 2009.
- [16] M. Muffert, D. Pfeiffer and U. Franke, “*A Stereo-Vision Based Object Tracking Approach at Roundabouts*”, IEEE Intelligent Transportation Systems Magazine, Vol. 2, No. 2, pp. 22-32, 2013.
- [17] R. Ginhoux and J. S. Gutmann, “*Model-based Object Tracking Using Stereo-vision*”, Proceedings of the International Conference on Robotics and Automation, pp. 1226-1232, 2001.
- [18] F. Suligoj, B. Sekoranja, M. Svaco and B. Jerbic, “*Object Tracking with a Multiagent Robot System and a Stereo Vision Camera*”, Procedia Engineering, Vol. 69, pp. 968-973, 2014.
- [19] C. S. Yang, R. H. Chen, C. Y. Lee and S. J. Lin, “*PTZ Camera Based Position Tracking in IP-Surveillance System*”, Proceedings of the 3rd International Conference on Sensing Technology, pp. 142-146, 2008.
- [20] X. Li, Q. Chen and H. Chen, “*Detection and Tracking of Moving Object Based on PTZ Camera*”, Proceedings of the 5th International Conference on Advanced Computational Intelligence, pp. 493-497, 2012.
- [21] D. D. Doyle, A. L. Jennings and J. T. Black, “*Optical Flow Background Estimation for Real-Time Pan/Tilt Camera Object Tracking*”, Measurement Journal, Vol. 48, pp. 195-207, 2014.

- [22] Y. Sun, Q. Cao and W. Chen, “An *Object Tracking and Global Localization Method Using Omnidirectional Vision System*”, Proceedings of the 5th World Congress on Intelligent Control and Automation, pp. 4730-4735, 2004.
- [23] A. J. Neves, A. J. Pinho, D. A. Martins and B. Cunha “An *Efficient Omnidirectional Vision System for Soccer Robots: from Calibration to Object Detection*”, Mechatronics, Vol. 21, No. 2, pp. 399-410, 2011.
- [24] K. M. Hasan, A. A. Nahid and A. A. Mamun, “*Implementation of Vision Based Object Tracking Robot*”, Proceedings of the International Conference on IEEE/OSA/IAPR Informatics, Electronics and Vision, pp. 859-864, 2012.
- [25] H. Medeiros, J. Park, and A. C. Kak, “*Distributed Object Tracking Using a Cluster-Based Kalman Filter in Wireless Camera Networks*”, IEEE Journal of Selected Topics in Signal Processing, Vol. 2, No. 4, pp. 448-463, 2008.
- [26] G. Xing, “*Real-time Follow Up Tracking Fast Moving Object with a Active Camera*”, Proceedings of the 2nd International Conference on Image and Signal Processing, pp. 1-4, 2009.
- [27] K. Kinoshita, M. Enokidani, M. Izumida and K. Murakami, “*Tracking of a Moving Object Using One-dimensional Optical Flow With a Rotating Observer*”, Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision, pp. 1-6, 2006.
- [28] S. Bota and S. Nedevschi, “*Tracking Multiple Objects in Urban Environment Using Dense Stereo and Optimal Flow*”, Proceedings of the 14th IEEE Conference on Intelligent Transportation Systems, pp. 791-796, 2011.

- [29] D. Hsu, J. Leu, S. Chen, W. Chang and W. Fang, “*Using Shape Feature Matching to Track Moving Objects in Image Sequence*”, Proceedings of the IEEE International Conference on Multimedia and Expo., Vol. 2, pp. 785-788, 2000.
- [30] A. Ahmad and P. Lima, “*Multi-Robot Cooperative Spherical-Object Tracking in 3D Space Based on Particle Filters*”, Proceedings of the 5th European Conference on Mobile Robotics, pp. 1084-1093, 2013.
- [31] X. Chen, X. Li, H. Wu and T. Qiu, “*Real-Time Object Tracking via CamShift-Based Robust Framework*”, Proceedings of the 2012 International Conference on Information Science and Technology, pp. 527-530, 2012.
- [32] Y. Yan, J. Wang, C. Li and Z. Wu, “*Object Tracking Using SIFT Features in a Particle Filter*”, Proceedings of the IEEE 3rd International Conference on Communication Software and Networks, pp. 384-388, 2011.
- [33] L. Ying, G. Chengjiao and T. Ikenaga, “*Estimation-Correction Scheme Based Articulated Object Tracking Using SIFT Features and Mean Shift Algorithm*”, Proceedings of the 4th International Conference on New Trends in Information Science and Service Science, pp. 275-280, 2010.
- [34] Y. S. Do and Y. J. Jeong, “*A New Area Efficient SURF Hardware Structure and its Application to Object Tracking*”, Proceedings of the 2013 IEEE Region 10 Conference, pp. 1-4, 2013.
- [35] Z. Zhou, X. Ou and J. Xu, “*SURF Feature Detection Method Used in Object Tracking*”, Proceedings of the 2013

- International Conference on Machine Learning and Cybernetics, pp. 1865-1868, 2013.
- [36] I. A. Sulistijono, S. Kuswadi, O. Setiaji, I. Salfikar and N. Kubota, “*A Study on Fuzzy Control of Humanoid Soccer Robot EFuRIO for Vision Control System and Walking Movement*”, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 16, No. 3, pp. 444-452, 2012.
 - [37] A. Sanz, “*A Six Legged Robot Controlled by ARS*”, Proceedings of the 6th IEEE International Conference on Fuzzy Systems, Vol. 2, pp. 1089-1093, 2014.
 - [38] A. A. Saputra, I. A. Sulistijono, A. S. Khalilullah, T. Takeda and N. Kubota, “*Combining Pose Control and Angular Velocity Control for Motion Balance of Humanoid Robot Soccer EROS*”, Proceedings of the 2014 IEEE Symposium on Robotic Intelligence in Informationally Structured Space (RiiSS), pp. 1-6, 2014.
 - [39] M. Lien, A. Visser and F. Groen, “*A Hybrid Algorithm for Tracking and Following People Using a Robotic Dog*”, Proceedings of the 2008 3rd ACM/IEEE International Conference on Human-Robot Interaction, pp. 185-192, 2008.
 - [40] R. A. El-laithy, J. Huang and M. Yeh, “*Study on the Use of Microsoft Kinect for Robotics Applications*”, Proceedings of the Position Location and Navigation Symposium, pp. 1280-1288, 2012.
 - [41] J. Han, L. Shao, D. Xu and J. Shatton, “*Enhanced Computer Vision with Microsoft Kinect Sensor: A Review*”, IEEE Transactions on Cybernetics, Vol. 43, No. 5, pp. 1318-1334, 2013.

- [42] K. Khoshelham and S. O. Elberink, “*Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications*”, Journal of Sensors, Vol. 12, No. 2, pp. 1437-1454, 2012.
- [43] G. Hoang, “*A Study on Control of a Six-Legged Walking Robot with Four Joint Legs*”, Doctoral Thesis, Pukyong National University, August, 2014.
- [44] K. Kungcharoen, P. Palangsantikul and W. Premchaiswadi, “*Development of Object Detection Software for a Mobile Robot Using an AForge.NET Framework*,” Proceedings of the 9th International Conference on ICT and Knowledge Engineering, pp. 201-206, 2012.
- [45] L. He, Y. Chao and K. Suzuki, “*Two Efficient Label-Equivalence Based Connected Component Labeling Algorithms for 3-D Binary Images*”, Proceedings of the IEEE Transactions on Image Processing, Vol. 20, No. 8, pp. 2122-2134, 2011.
- [46] G. Xing, S. Tian, H. Sun, W. Liu and H. Liu, “*People-Following System Design for Mobile Robot Using Kinect Sensor*”, Proceedings of the 25th Chinese Control and Decision Conference, pp.3190-3194, 2013.
- [47] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki and S. Thrun, Principles of Robot Motion: Theory, Algorithms and Implementations, MIT Press, 2005.
- [48] S. Y. Chen, “*Kalman Filter for Robot Vision: A Survey*”, IEEE Transactions on Industrial Electronics, Vol. 59, No. 11, pp. 4409-4420, 2012.

Publications and Conferences

A. Under Submission

1. **A. V. Gulalkari**, P. S. Pratama, H. Giang, H. K. Kim, S. B. Kim, and B. H. Jun, “*Object Tracking and Following Six-Legged Robot System Using Kinect Camera Based on Kalman Filter and Backstepping Controller*”, Submission to International Journal of Mechanical Science and Technology, Korea, 2014.
2. Y. D. Setiawan, P. S. Pratama, **A. V. Gulalkari**, H. K. Kyeong, and S. B. Kim, “*Path Tracking Controller Design Four Wheel Independent Steering Automatic Guided Vehicle*”, Submission to International Journal of Control, Automation and Systems, Korea, 2015.
3. **A. V. Gulalkari**, D. Sheng, P. S. Pratama, H. K. Kyeong, G. S. Byun and S. B. Kim, “*Kinect Camera Sensor-Based Object Tracking and Following of Four Wheel Independent Steering Automatic Guided Vehicle Using Kalman Filter*”, Submission to 2015 15th International Conference on Control, Automation and Systems (ICCAS 2015), Bexco, Korea, 2015.
4. P. S. Pratama, **A. V. Gulalkari**, J. W. Kim, C. H. Lee, T. K. Yeu, S. Hong, H. K. Kim and S. B. Kim, “*Obstacle Avoidance Control for Four Wheels Independent Steering Automatic Guided Vehicles*”, Submission to The International Conference

on Advanced Engineering – Theory and Applications (AETA 2015), Ho Chi Minh City, Vietnam, 2015.

B. Conferences

1. **A. V. Gulalkari**, H. Giang, P. S. Pratama, H. K. Kim, S. B. Kim, and B. H. Jun, “*Object Following Control of Six-Legged Robot Using Kinect Camera*”, Proceeding of IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI), Greater Noida, India, 2014 IEEE, pp. 758-764, 2014.
2. **A. V. Gulalkari**, P. S. Pratama, J. W. Kim, H. K. Kim, B. H. Jun, and S. B. Kim, “*Moving Object Following Control of Six-legged Robot with Four Joint Legs Based on Backstepping Method*,” Proceedings of 2015 Korean Unmanned Underwater Vehicle (KUUV) Conference, KRISO, Daejeon, Korea, pp. 47-53, 2015.
3. P. S. Pratama, **A. V. Gulalkari**, D. H. Kim, H. K. Kim, Y. S. Jung, J. H. Min, and S. B. Kim, “*Fault Detection and Identification of Induction Motor for a Crossbelt Sorter Shooting System*,” Proceedings of 2015 The Korean Society for Logistics Science and Technology (KLST) conference, KINTEX, Seoul, Korea, 2015.

Appendix A

Denavit-Hartenberg (DH) convention can be described as follows:

- Choose axis z_i along the axis of joint $i + 1$.
- Locate the origin O_i at the intersection of axis z_i with the common normal to axes z_{i-1} and z_i . Also, locate $O_{i'}$ at the intersection of the common normal with axis z_{i-1} .
- Choose axis x_i along the common normal to axes z_{i-1} and z_i with direction from joint i to joint $i + 1$.
- Choose axis y_i so as to complete a right-handed frame.

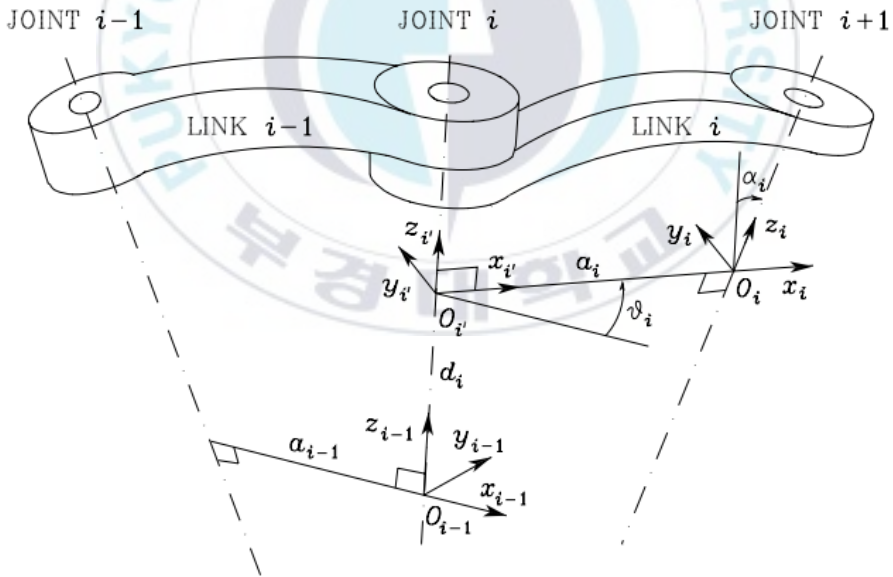


Fig. A.1 Denavit-Hartenberg kinematic parameters

Once the link frames have been established as shown in Fig. A.1, the position and orientation of frame i with respect to frame $i - 1$ are completely specified by the following parameters:

- a_i : Distance between O_i and O_{i-1}
- d_i : Coordinate of O_{i-1} along z_{i-1}
- α_i : Angle between axes z_{i-1} and z_i about axis x_i to be taken positive when rotation is counter-clockwise
- θ_i : Angle between axes x_{i-1} and x_i about axis z_{i-1} to be taken positive when rotation is counter-clockwise.



E.O.D.