HUMAN-LIKE COORDINATION OF BODY-ASSISTED ARM MOVEMENTS FOR OBJECT MANIPULATION

by
Kadir Cumali
B.S., Electronics Engineering, Uludağ University, 2013

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Systems and Control Engineering
Boğaziçi University
2019

HUMAN-LIKE COORDINATION OF BODY-ASSISTED ARM MOVEMENTS FOR OBJECT MANIPULATION

APPROVED BY:

Prof. H. Işıl Bozma . . . .
(Thesis Supervisor)

Prof. Yağmur Denizhan . . . .

Prof. Erhan Öztop . . . .

DATE OF APPROVAL:

ACKNOWLEDGEMENTS

I would first like to thank my thesis advisor Prof. H. Işıl Bozma, for her guiding and support during my thesis. The door to her office was always open whenever I ran into a trouble spot or had a question about my research or writing. She consistently allowed this study to be my work but steered me in the right direction whenever she thought I needed it.

Second, I also want to thank Prof. Yağmur Denizhan and Prof. Erhan Öztop as my thesis committee. I am gratefully indebted to them for their valuable comments on this thesis.

I must express my very profound gratitude to Dr. Selim Yannier for providing me with unfailing support and continuous encouragement throughout my years of study.

I want to thank to Kadir Türksoy, Berkan Höke, Çağatay Odabaşı, Mahmut Demir, Deniz Şenel, Halil Samed Çıldır, Aybars Safi, Enes Berekeci for being incredible companions in our laboratory.

I would also like to thank my friends for their support. Especially, I want to express my special thanks to Nesrin Yenihayat, Onur Kardaş, Smart Vehicles Team, for their valuable support. I do not think that I would come to this point without them.

Finally, this thesis is dedicated to my future children, my mother, my father, my sister, her family, and all the engineers who like science. Without their dedication, I would not be here.

ABSTRACT

HUMAN-LIKE COORDINATION OF BODY-ASSISTED ARM MOVEMENTS FOR OBJECT MANIPULATION

Manipulation is an integral capability for service robots. The goal of this thesis is to design and develop an approach that enables a mobile robot to mimic human manipulation abilities. We consider a differential type of mobile robot that is endowed with an arm and gripper. The robot is assumed to have visual sensing so that it can determine the relative position of the object of interest. First, it is observed that humans exhibit various basic modes of interaction with an object of interest, including extension, flexion, gripping, release and translation. As such, the robot can be programmed to have similar capabilities through establishing the correspondence between the robot and a human with respect to the underlying manipulation mechanisms. More complex behaviors such as putting, pulling, pushing, and shaking are defined using a sequential composition of basic operations. Second, humans are observed to achieve these tasks through the coordination of their body and arm movements. For this, a control approach in which the movements of the robot body and manipulator are coupled temporally and spatially is proposed. As such, if the object of interest is within the robot's reach, then only arm movements are made. If this is not the case, the robot starts moving its body. Depending on the vicinity of the object, this may be accompanied by arm motion or not. The control algorithm results in the robot's body and arm movements to be done in a coupled manner. The proposed approach is evaluated through an extensive set of experiments involving various manipulation tasks.

ÖZET

İNSANSI NESNE MANİPÜLASYONU İÇİN VÜCUT VE KOL KOORDİNASYONU

İnsansı robotlar son yıllarındaki kullanım alanları itibariyle bir nesnenin alınması, bir yere götürülmesi veya kaldırılması gibi görevleri üstlenmektedir. Bu tezde, insan ile yapılan obje manipülasyonu çalışmalarıdan esinlenerek gezgin bir robotun insansı şekilde gündelik nesneyi manipule etmesi amaçlanmaktadır. Ancak, robotun, manipülasyon öncesi vücut ve kol hareketinin koordinasyolu ve eş zamanlı çalışma durumunun planlanması gerekmektedir. Bu çerçevede, robot kol ve vücut hareketinin zamansal ve mekansal olarak birleşmesi için özgün bir yaklaşım benimsenmiştir. Bu yaklaşımda, robot ve nesne arasında uzaklık üzerinden, robot nesneye yaklaşırken belirli bir uzaklıkta kol hareketinin, vücut hareketine eşlik etmesi modellenmiştir. Bu kontrol algoritması, robot hareketinin ve kol hareketinin ayrışmış planmasını gereksiz kılmakta ve bütünleşik bir kontrol yapısı sunmaktadır. Ayrıca, robotun yapabileceği hareketler ile ilgili olarak alt kademe görev kontrol mekanizması oluşturulmuş ve mekanizmadan çıkan aksiyonlar birleştirilerek robota al-bırak gibi görev komutları verilmektedir. Yapılan kapsamlı deneyler ile önerilen yöntem gezgin bir robotun değişik kısımları arasındaki eş zamanlı koordinasyon problemine çözüm getirmektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$c \in R^2$	Robot position
$g \in R^2$	Goal position
J	Jacobian
q	Joints
S^2	2D Space
$x = [c \;\; \alpha]^T$	Robot's position vector
$\alpha \in S^1$	Heading
τ_1	Distance threshold for termination of robot body motion
τ_2	Distance threshold for reaching
γ_1	scale factor for robot arm movement
γ_2	scale factor for robot body motion
$\varphi(q)$	Artificial Potential Function
∇	Gradient

LIST OF ACRONYMS/ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
DoF	Degree of Freedom
APF	Artificial Potential Function
ISL	Intelligent Systems Laboratory
PRR	Prismatic Revolute Revolute
RGB-D	Red Green Blue-Depth

1. INTRODUCTION

Manipulation is a particularly important skill for service robots, applications including logistics, personal assistance, agriculture, and even in hospitals. Manipulation tasks include various modes of interaction with an object of interest, including extension, flexion, gripping, release and translation. Humans are observed to achieve these tasks through coordinating the motion of their bodies and their arms based on visual feedback. They will re-locate their trunks so that the object is reachable. Hence, the multi-segmental coordination between trunk and arm is established so that the hand moves to a coherent endpoint position while following a smooth path [1]. This coordination pattern presents a pivotal and instructive paradigm regarding how robots can mimic these movements. As such, as manipulation movements of the robot will become more human-like, interaction with robots may become more natural. This thesis considers the manipulation problem from this perspective and presents an approach based on mimicking human manipulation features.

1.1. General Approach

In the thesis, first, the correspondence between the robot and a human is established with respect to their respective topology of joints. This enables the identification of basic manipulation operations. As such, behavior associated with complex manipulation tasks are defined through the sequential composition of basic operations. The basic operations include extension, flexion, gripping, release and translation as defined by joint movements. Complex manipulation tasks such as take-and-put and take-and-shake are defined by sequentially composing primitive operations while taking the constraints of the environment into account. Next, a control approach in which the movements of the robot body and manipulator are coupled temporally and spatially is proposed. As such, if the object of interest is within the robot's reach, then only arm movements are made. If this is not the case, the robot starts moving its body. This is based on encoding the basic movements using an artificial potential function (APF). The gradient of the function defines the control inputs to the robot body, arm, and gripper as deemed necessary based on the visual feedback from an RGB-D sensor.

1.2. Contributions

The contributions of this thesis can be summarized as follows:

- Behavior associated with complex manipulation tasks are defined by the sequential composition of basic manipulation modes.
- A new controller in which robot's body and arm movements occur in a coupled manner in basic manipulation tasks is presented. This enables the robot to reach the object-of-interest in a human-like manner.

1.3. Outline

The thesis is organized as follows: In Chapter 3, the robot' motion mechanism is analyzed in detail including its comparison with its counterpart in humans. In Chapter 4, the proposed approach is presented. Extensive simulations and robot experiments are presented in Chapter 5. The thesis concludes with a brief summary and a discussion of future directions.

2. RELATED LITERATURE

Humans execute natural movements for the manipulation tasks that require the coordination between trunk and arm. Imitating from humans, the generation of human-like motions for the robot provides a vital and instructive paradigm to develop a point of view regarding these movements. This section provides an overview of the state-of-art studies related to human reaching strategy.

Many studies have been conducted to understand the human features in reaching [2–6]. The principal paradigms and models regarding the control of the human arm have been reviewed in [7]. However, it is inadequate to examine the effect of the human arm in reaching movement since the torso shifts when the object is not in the arm's workspace. The human body will approach the object and position itself accordingly. By taking this into account, several studies addressed the contribution of the trunk to the arm in manipulation tasks [8], [9]. Some studies proposed that the human trunk and arm have spatial and temporal coordination during the reaching movements. The trunk motion begins before the arm movement and continues until the arm motion ends [10]. Mark *et al.* acquired an approximate distance at which the trunk involves into the reaching movement [11].

While the above research discussed the problem of attaining trunk-assisted motions, a key issue is how the robots could incorporate body sections into the tasks. Reinhart *et al.* presented a controller framework based on a recurrent neural network for the generation of reaching movements [12]. In [13], it is used a physically inspired optimization method to imitate human reaching movements in an environment of everyday-life. In [14], it is proposed a reinforcement learning approach to generate human-like reaching movements in constrained environments. Arimoto *et al.* implemented the bell-shaped velocity profile for the point-to-point arm movements to confirm the human-likeness of the robot arm [15]. Other authors tried to characterize arm movements based on optimization of torque [16] and jerk [17]. Many robotic studies above regarding human-like reaching movements do not take account of the effect of involvement of trunk. Bhattacharjee *et al.* proposed a control method to exhibit human-motion characteristics in redundant robot arm-trunk systems for reaching tasks [18]. This model may be insufficient to embed the spatial relation of arm and robot body motion. Brandao *et al.* presented a control architecture for the integration of visually guided walking, and whole-body reaching in a humanoid robot [19]. They only modeled the movements of the arm and robot waist in reaching not the synchronous movements of trunk displacement and arm. In [20], it is proposed a framework that combines the complex full-body movements such as reaching a target while talking.

The present work is designed to coordinate all robot segments for manipulation tasks. It can be implemented to mobile robots that there is spatial and temporal relationship between arm and trunk during reaching. The primary contribution is to propose a mathematical model mimicking the spatial relation of the human body segments concerning the object location. The second is to present a coordinated control based on the coupling motion of parts in reaching tasks for manipulation. Besides, the presented approach is adaptable to any mobile robot. That creates a suitable environment for developing new algorithms on the robot.

3. ROBOT MODEL

The robot consists of a differential wheel type motion mechanism with a manipulator, as shown in Figure 3.1. The robot arm is mounted on the robot platform. The manipulator consists of a robotic arm built consisting of one prismatic (P) and two revolute (RR) joints and one degree of freedom (DOF) end effector. The robot has a pan-tilt head with an RGB-D camera and is able to determine the position of an object of interest.

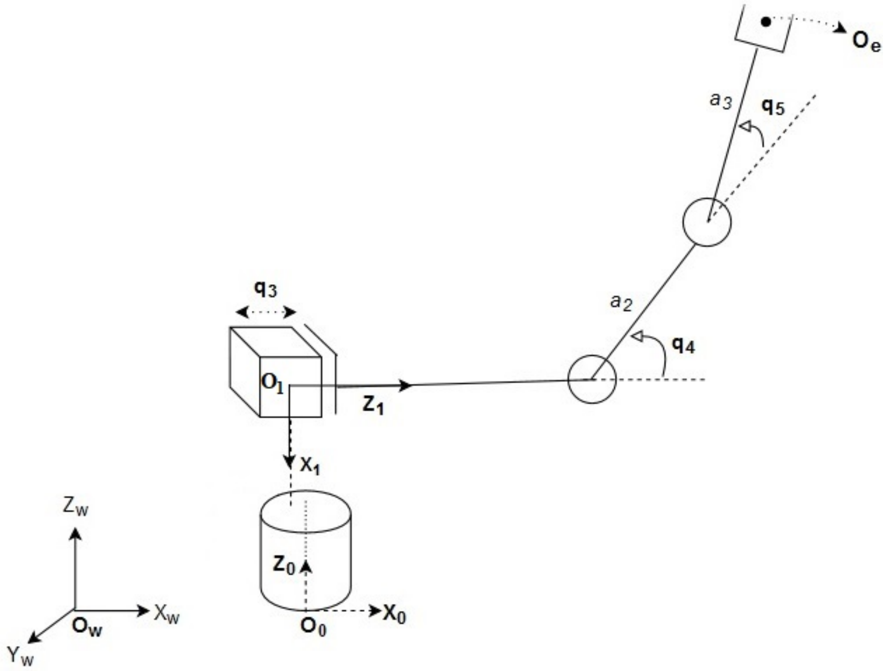


Figure 3.1: The robot is differential wheeled with a PRR arm and a 1 DOF gripper.

- The robot is assumed to move on a planar horizontal surface. Let $X = (O_w, \overrightarrow{X_w}, \overrightarrow{Y_w}, \overrightarrow{Z_w})$ be any fixed frame with $\overrightarrow{Z_w}$ vertical.
- O_1 is linked to the robot base and O_e is the center of the end-effector. The whole configuration related to robot body q^B and robot arm q^A is given by:

$$q = \begin{bmatrix} q^B & q^A \end{bmatrix}^T \tag{3.1}$$

- Consider a manipulator shown in Figure 3.1, where the four principal coordinate frames are shown: world frame O_w, robot frame O_0, robot arm frame O_1 and end effector frame O_e. Then, the manipulator's end effector position and orientation with respect to O_w.

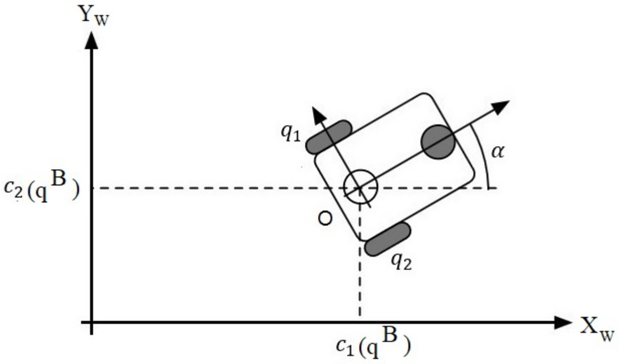


Figure 3.2: Differential drive mobile robot parameters

3.1. Robot Body Model

The robots' body is supported by two independently driven wheels with a common platform-fixed axis and one passive, self-aligning wheel as shown in Figure 3.2. To maneuver the robot in the plane, the robot requires two inputs: linear velocity υ and a heading angle α.

- The differential robot has two degrees of freedom that correspond the angular velocity of the left (q_1) and right (q_2) driving wheel. The robot variables are shown by q^B in the space. It is represented by $q_j \in S^1$, where $j = 1, 2$.

$$q^B = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^T \in S^2 \tag{3.2}$$

- The current position of the robot is $(c_1, c_2)^T \in \mathbb{R}^2$. The orientation of the robot is $\alpha \in S^1$. Thus, the robot's base point is defined as follows:

$$x(q^B) = \begin{bmatrix} c_1(q^B) & c_2(q^B) & \alpha(q^B) \end{bmatrix}^T \tag{3.3}$$

- The wheels have radius r. Given a point O centered between the two drive wheels, each wheel has a distance l from center.

- Assuming a constant rotational velocity for the wheels and the position c_1 and c_2 and orientation α of the robot pose can be estimated using Eq.3.4,

$$\dot{x} = \begin{bmatrix} \dot{c_1} \\ \dot{c_2} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} r(\frac{q_1 + q_2}{2})cos(\alpha) \\ r(\frac{q_1 + q_2}{2})sin(\alpha) \\ \frac{r}{2l}(q_1 - q_2) \end{bmatrix} \tag{3.4}$$

- While modeling the robot, the following assumptions made are as follows:
  a) The robot moves with a constant velocity.
  b) The wheels of the robot do not slip and the surface for robot motion is flat.

3.2. Arm Model

The robot arm has three degrees of freedom with one prismatic joint, moving on a horizontal plane, and two revolute joints. The prismatic joint permits a linear motion along a single axis (*i.e.*, an extension or a retraction), and the revolute joints allow a relative rotation about a single axis. The Figure 3.4 shows robot configuration with the coordinates presentation. The input to the system is two-dimensional vector E of e_x, e_z. Cartesian forces managing on the end-effector, with components expressed in a robot arm frame $O_1X_1Y_1Z_1$.

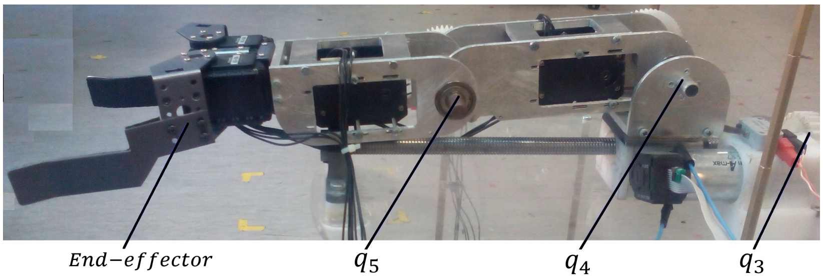


Figure 3.3: Joints and end-effector in the PRR robot manipulator

- The robot arm has three degrees of freedom and prismatic, revolute and revolute (PRR) configuration, respectively. a_2 and a_3 are link lengths. The joint parame-

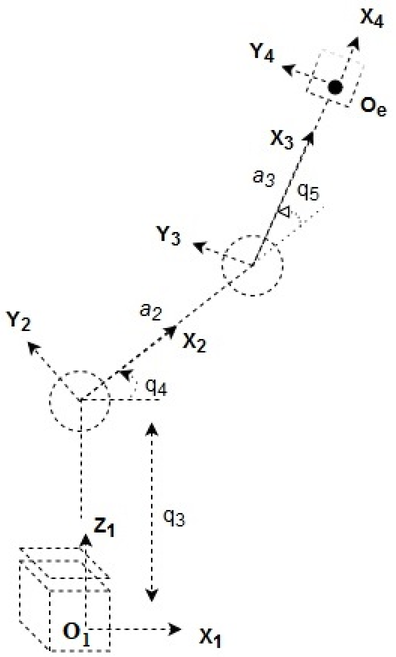


Figure 3.4: Three-link PRR manipulator and its coordinate frame

ters of the robot arm q^A are shown in Equation 3.5. It is represented by $q_j \in S^1$, where $j = 3, 4, 5$.

$$q^A = \begin{bmatrix} q_3 & q_4 & q_5 \end{bmatrix}^T \in S^3 \tag{3.5}$$

- Since there are more joint variables than there are DoFs for the end-effector, the manipulator is said to be redundant. Whereas there are 3 joint variables for the manipulator, there are 2 DoFs for the gripper. Therefore, the degree of redundancy for the end-effector is 1.
- The manipulator is linked with the robot platform from the center O_1.
- The center of the end-effector is O_e, which moves in a plane. The end-effector pose can be defined by two position components and one orientation angle.
- The forward kinematics of PRR manipulator with reference to O_1 is given below.

$$T_e^{O_1} = \begin{bmatrix} cos(q_4+q_5) & -sin(q_4+q_5) & 0 & q_3 + a_2cos(q_4) + a_3cos(q_4+q_5) \\ 0 & 0 & -1 & 0 \\ sin(q_4+q_5) & cos(q_4+q_5) & 0 & a_2sin(q_4) + a_3sin(q_4+q_5) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

- Notice that the first three entries of the last column of $T_e^{O_1}$ are the position vector $e(q^A) = (e_X(q^A), e_Y(q^A), e_Z(q^A))^T$ components of the origin of end-effector O_e relative to world frame O_W; that is,

$$\begin{aligned} e_X &= q_3 + a_2cos(q_4) + a_3cos(q_4 + q_5) \\ e_Z &= a_2sin(q_4) + a_3sin(q_4 + q_5) \end{aligned} \tag{3.7}$$

are the coordinates of the end-effector in the world frame. The rotational part of $T_e^{O_a}$ gives the orientation of the frame $O_eX_4Y_4Z_4$ relative to the robot frame.

- The forward kinematics offers insight into operational space - set of all settings that an end-effector can accomplish, joint space - possible joints can take, and precision - deviation between assigned position and end-effector current position.
- The end-effector Jacobian of interest is

$$\dot{e} = J_e\dot{q}^A \tag{3.8}$$

where

$$J_e(q) = \begin{pmatrix} 1 & -(a_2sin(q_4) + a_3sin(q_4 + q_5)) & -a_3sin(q_4 + q_5) \\ 0 & a_2cos(q_4) + a_3cos(q_4 + q_5) & a_3cos(q_4 + q_5) \end{pmatrix} \tag{3.9}$$

- The robot arm has singularity when $cos(q_4 + q_5) = cos(q_4) = 0$. It means that when the second and third links lie both (folded or stretched) along a line orthogonal to the prismatic joint axis, the manipulator loses one degree of freedom in Cartesian space, and changes in joint variables do not result in change in end-effector pose.

Given e_X, e_Z and α, the joint values q_3, q_4 and q_5, necessary to achieve the given position and orientation of the end-effector need to be calculated. In this chain, a prismatic joint p_3 is fixed at the base creating a fixed angle with the robot platform. The joints connecting p_3 to p_4 and p_5 to the platform are revolute. By parallel projection on the X and Z axes, the coordinates of point O_0 are as follows:

$$e_X = q_3 + a_2cos(q_4) + a_3cos(q_4 + q_5) \tag{3.10}$$

$$e_Z = a_2sin(q_4) + a_3sin(q_4 + q_5) \tag{3.11}$$

The orientation angle is as below.

$$\psi = q_4 + q_5 \tag{3.12}$$

When the joint displacement are known, e_X and e_Z generate a unique position for the gripper. A reverse analysis is performed by substituting orientation angle into Equation 3.11, which gives

$$q_4 = sin^{-1}(\frac{e_z - a_3sin(\psi)}{a_2}) \tag{3.13}$$

The corresponding value q_3 can be computed from Equation 3.10,

$$q_3 = e_x - a_3cos(\psi) - a_2cos(q_4) \tag{3.14}$$

After the determination of two joint variables, the left one is obtained using orientation angle.

$$q_5 = \psi - q_4 \tag{3.15}$$

3.3. Human Manipulation

The human arm mechanism is composed of 7 DoFs, 3 DoFs in shoulder joint, 2 DoFs in the elbow joint, and 2 DoFs in the wrist joint, as seen in Figure 3.5. The basic movements of human arm can be classified into eight individual motions which are shoulder vertical flexion/extension, shoulder horizontal flexion/extension, shoulder adduction/abduction, shoulder internal/external rotation, elbow flexion/extension, forearm supination/pronation, wrist flexion/extension, and wrist ulnar/radial deviation.

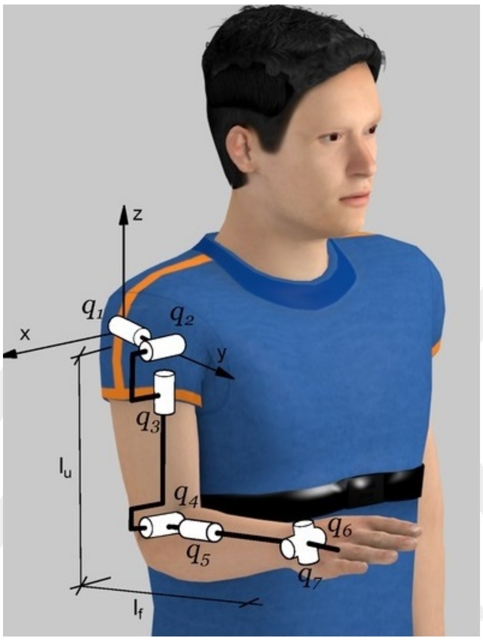


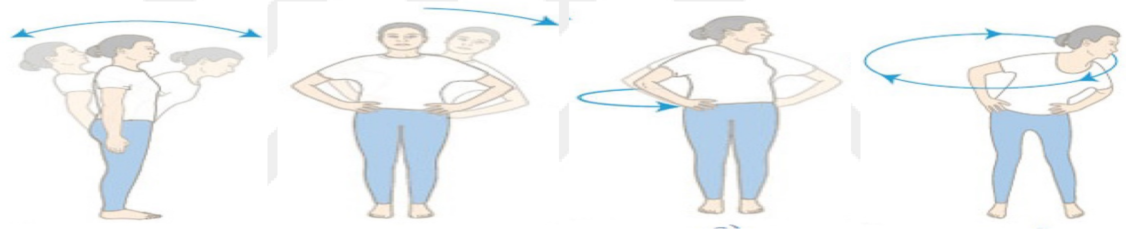
Figure 3.5: Human arm joints with 7 DoFs [21]

The correspondence between human arm joints and those of our robot is shown in Table 3.1. As such, it is evident that the manipulation capabilities of our robot is much simpler as it does not have counterparts to shoulder vertical flexion/extension and elbow flexion/extension.

In addition, humans have trunk muscles that move the torso freely in all three planes of movements, as shown in Figure 3.6. The trunk movements are flexion, extension, lateral flexion, and circumduction. The flexion takes place in forward bending or sitting lying. The extension expands the trunk that can bend backward. The trunk twists to the side laterally. The trunk rotates to the right or the left within circumduction. Since the robot is a differential robot that has non-holonomic constraints, the robot body can not move laterally. Due to the mechanical limitations, the robot body can not execute circumduction and flexion/extension. Our robot can achieve only rotation around its axis. Additionally, the robot arm has a prismatic joint that

Table 3.1: Correspondence between human arm and PRR arm joints

Human arm joints	Human movement	PRR Arm Joint
q_1	Shoulder adduction/abduction	–
q_2	Shoulder extension/flexion	q_4
q_3	Shoulder internal/external rotation	–
q_4	Elbow extension/flexion	q_5
q_5	Elbow pronation/supination	–
q_6	Wrist palmar/dorsal flexion	–
q_7	Wrist radial/ulnar rotation	–



(a) Flexion/Extension (b) Lateral flexion (c) Rotation (d) Circumduction

Figure 3.6: Human trunk motion modes

will correspond to perform extension and flexion. Furthermore, the robot has a gripper that acts as a finger.

Table 3.2 summarizes the correspondence between human motion and our robot. Since the robot can not perform trunk extension, the robot arm can execute extension / flexion movements with its prismatic joint. The robot can move with the joints q_1 and q_2. The shoulder extension / flexion in humans corresponds to the revolute joint q_4.

Complex manipulation behaviors include taking, pushing, pulling, putting, and shaking. These can be achieved by the sequential composition of the basic modes of interaction while taking visual feedback into consideration.

Table 3.2: Correspondence between human motion and our robot

Human motion	Related Robot Part	Robot Joint
Trunk Extension/Flexion	Robot Arm	q_3
Shoulder Extension/Flexion	Robot Arm	q_4
Elbow Extension/Flexion	Robot Arm	q_5

The automata for “taking” task is given in Figure 3.7. The task includes extension/flexion and translation motions in terms of the human movements. In Figure 3.7, the input of the task is object position, which is determined by the vision system. When the object to be grasped is behind the arm workspace, the robot requires the body motion. When the robot enters the arm workspace, then the manipulator reaches the object by the primitive actions such as translation and extension. After reaching, the manipulator opens the gripper and closes it.

Figure 3.8 represents the automata for “putting” task. The task includes extension/flexion and translation motions in terms of the human movements. In Figure 3.8, the input of the task is object position, which is determined by the vision system and also the position where to put. The automata assumes that the manipulator grasps an object before starting the putting task. When the object is behind the arm workspace, the robot requires the body motion. When the robot enters the arm workspace, then the manipulator reaches the object by the primitive actions such as translation and extension. After reaching, the manipulator opens the gripper, and putting process ends.

Figure 3.9 shows the automata for “pulling” task. The task includes extension/flexion and translation motions in terms of the human movements. In Figure 3.9, the input of the task is object position, which is determined by the vision system and also the position where to pull. When the object is behind the arm workspace, the robot requires the body motion. When the robot enters the arm workspace, then the manipulator reaches the object by the primitive actions such as translation and

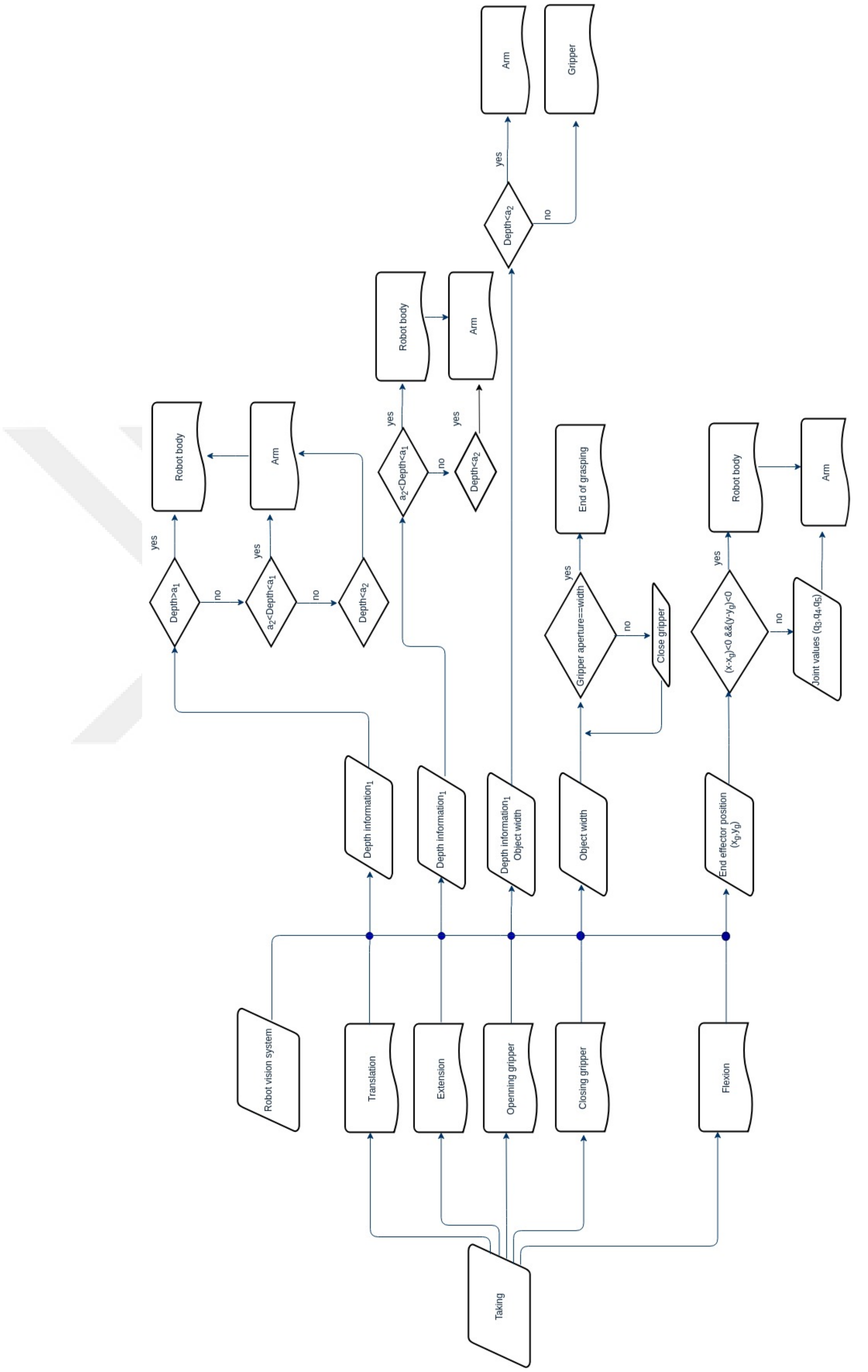


Figure 3.7: Taking task automata

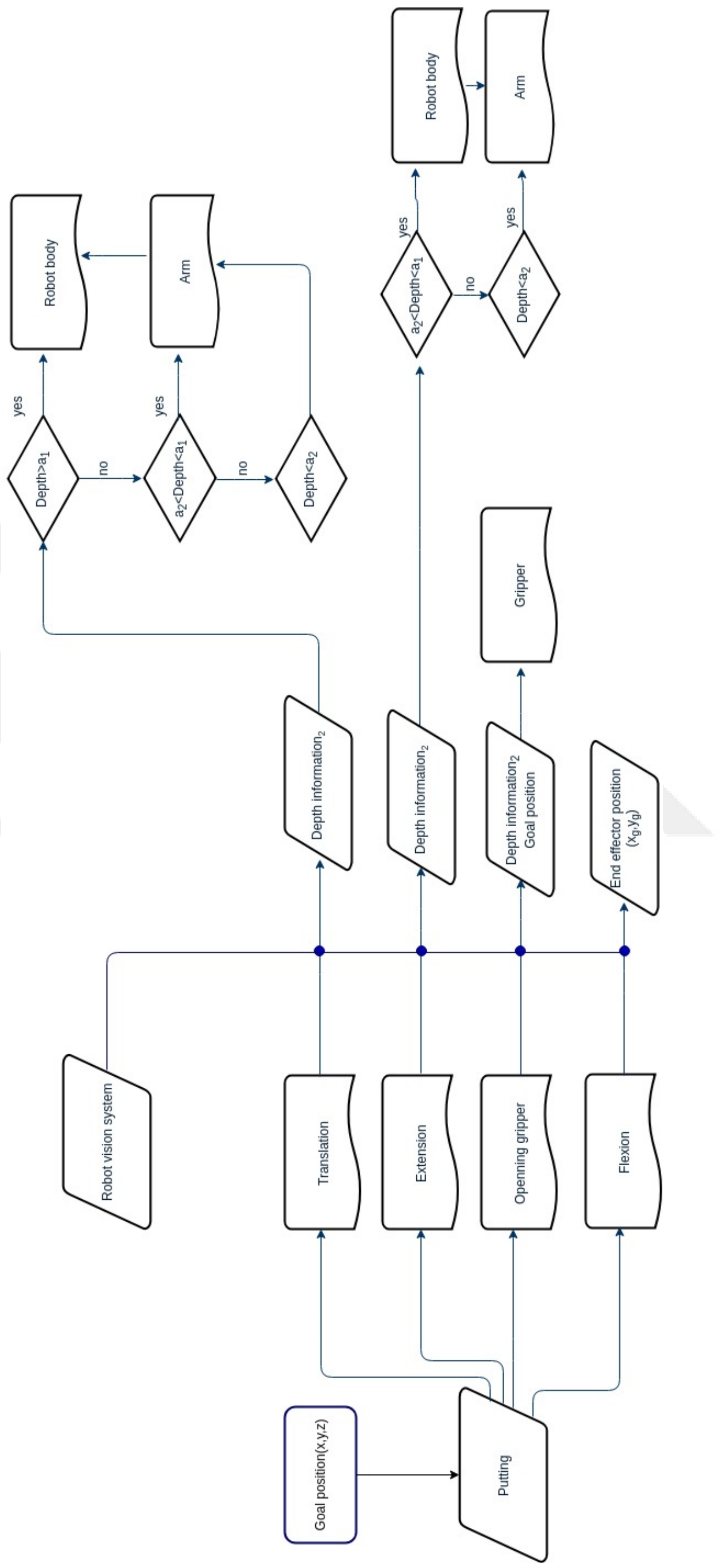


Figure 3.8: Putting task automata

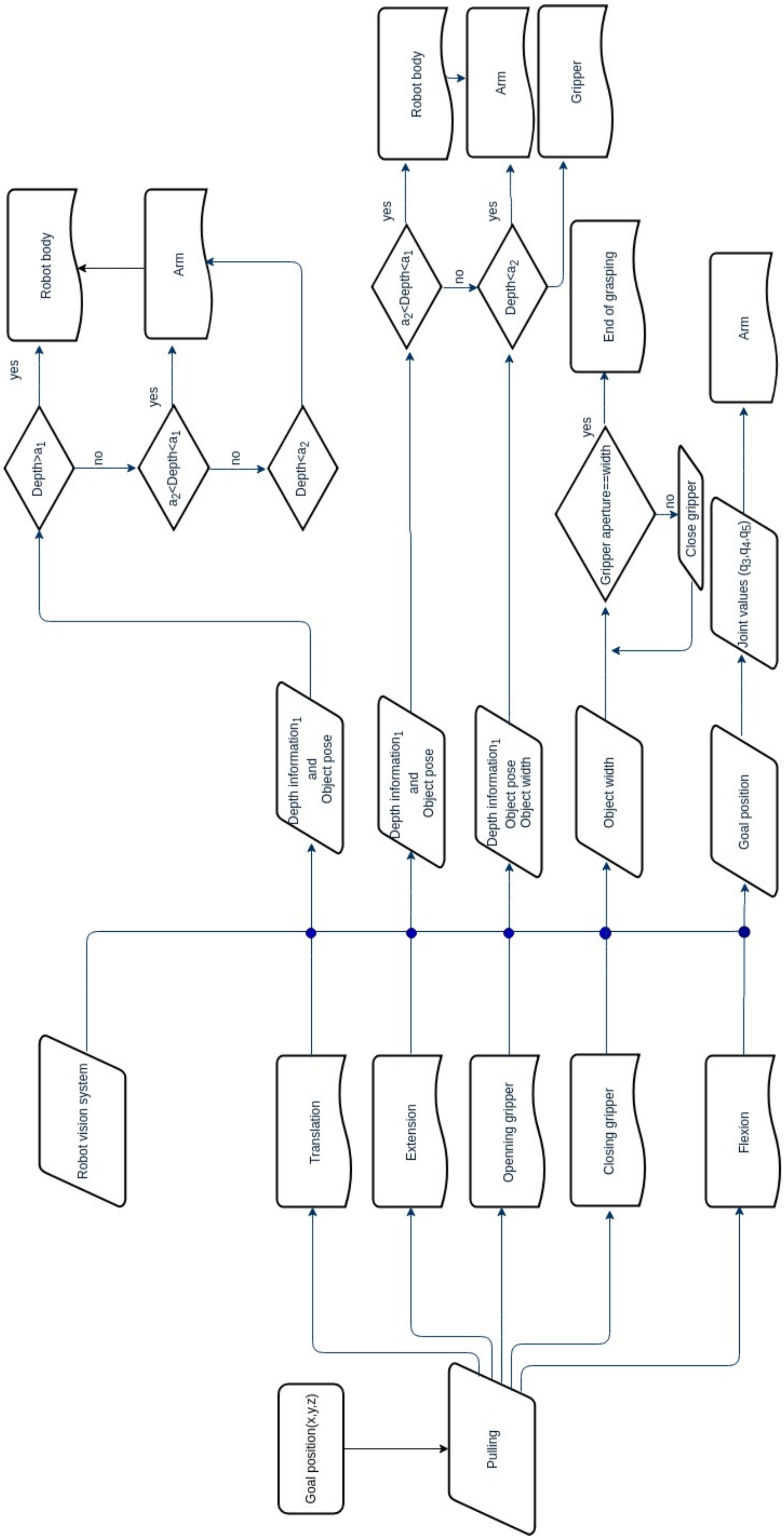


Figure 3.9: Pulling task automata

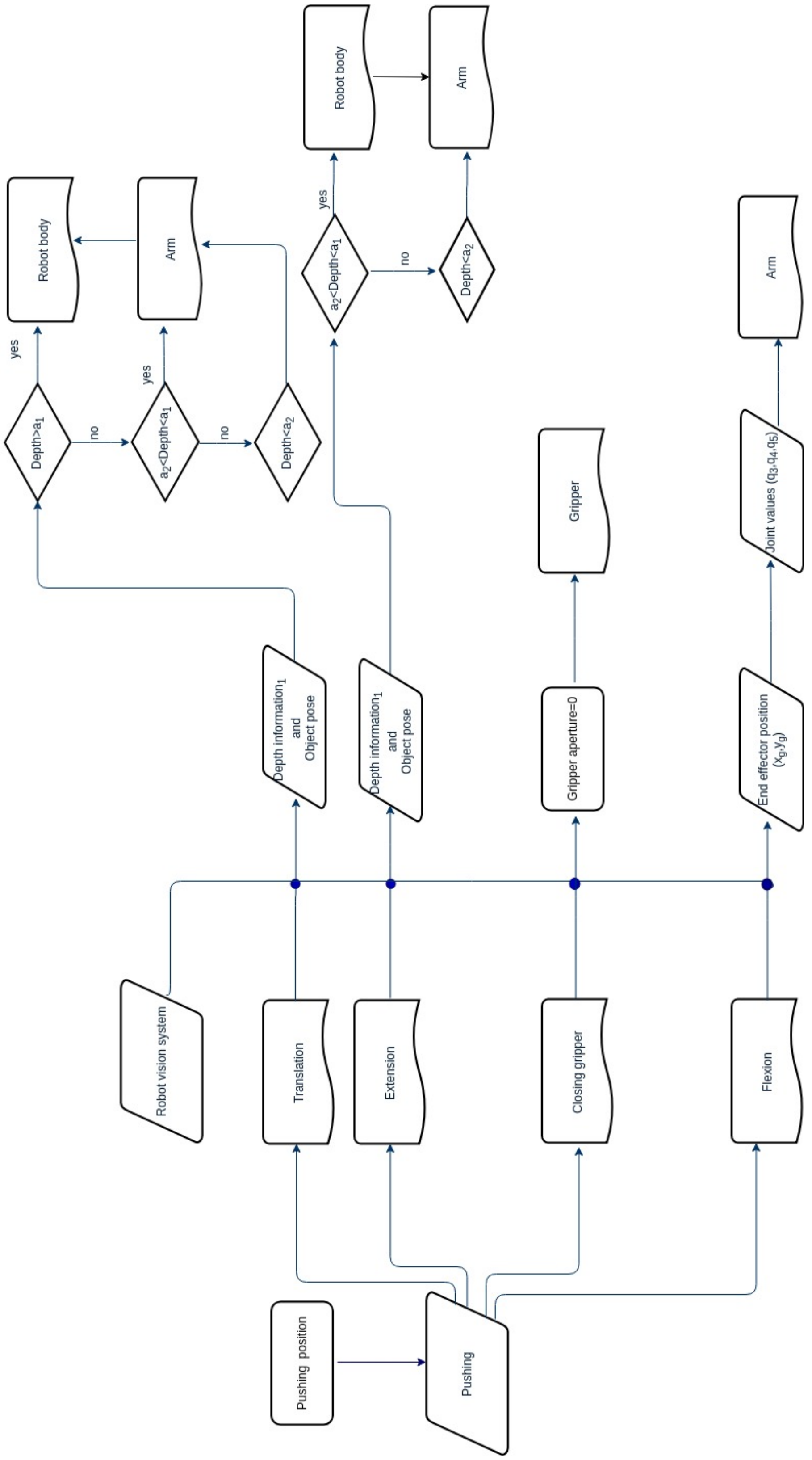


Figure 3.10: Pushing task automata

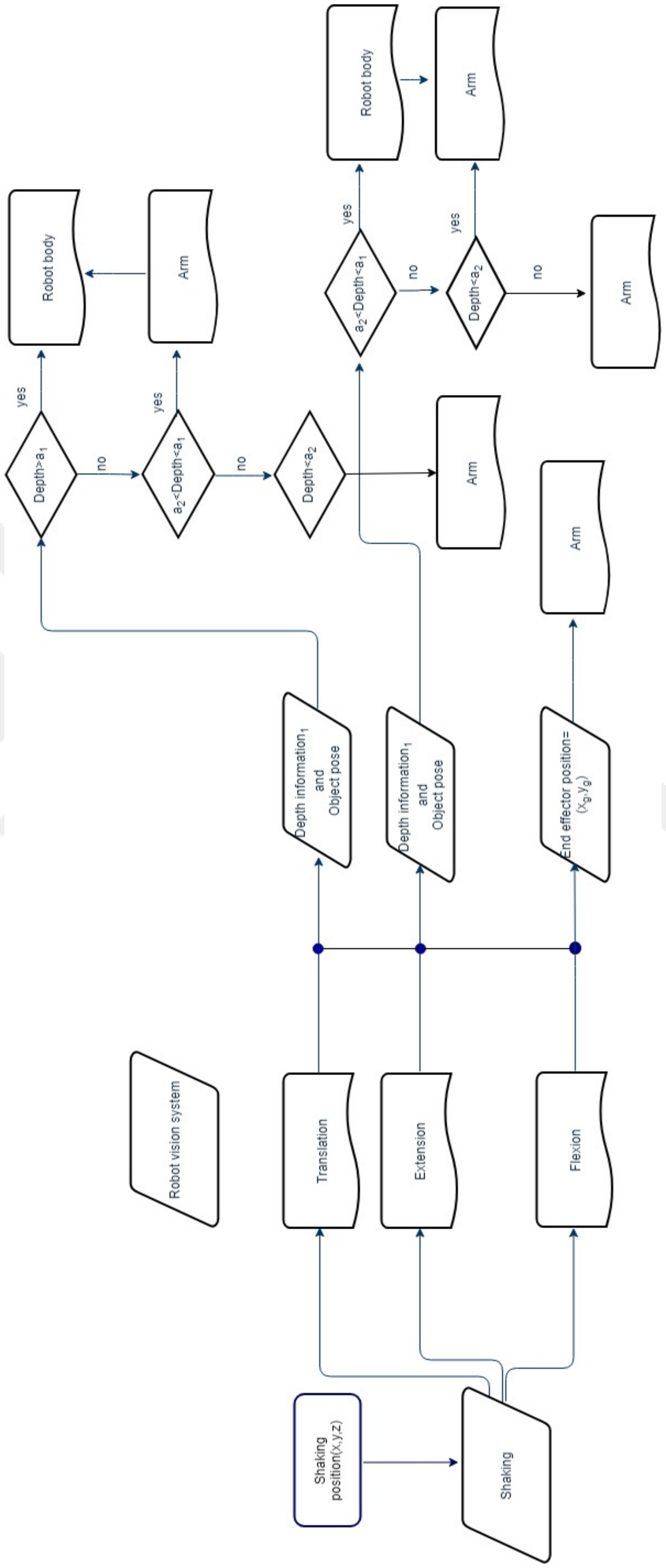


Figure 3.11: Shaking task automata

extension. After reaching, the manipulator opens the gripper and grasps the object. After the prehension, the manipulator pulls the object the predefined goal position, and pulling task process ends.

Figure 3.10 represents the automata for "pushing" task. The task includes extension/flexion and translation motions in terms of the human movements. In Figure 3.10, the input of the task is object position, which is determined by the vision system and also the position where to push. When the object is behind the arm workspace, the robot requires the body motion. When the robot enters the arm workspace, then the manipulator reaches the object by the primitive actions such as translation and extension. After reaching, the manipulator closes the gripper and pushes the object to the predefined goal position. Thereby, the process of pushing task ends.

Figure 3.11 describes the automata for "shaking" task. The task includes extension/flexion and translation motions in terms of the human movements. In Figure 3.11, the input of the task is object position, which is determined by the vision system and also the positions to shake. The automata assumes that the manipulator grasps an object before starting the shaking task. When the object is behind the arm workspace, the robot requires the body motion. When the robot enters the arm workspace, then the manipulator reaches the object by the primitive actions such as translation and extension. After reaching, the manipulator shakes the object between two points defined by the user.

3.4. Visual Processing

Fast and highly precise object detection is integral to manipulation. This is our system accomplished by using the OpenNI drivers [22] and a Kinect camera. Since these tasks require the object detection, the robot vision system obtains the position of the object relative to the manipulator. Thus, the visual processing aims to detect and track the color objects in real-time. This has been done as part of an EE492 Senior Project [23].

Object detection is done using RGB data. The approach is based on the color based detection that includes a range of color for the targeted object. An object which is in range of the color will be marked. The method relies on morphological operations and color segmentation. After the determination of the object, the algorithm calculates the distance between the robot and the object. Here, the depth data corresponding image area of the detected object is used to determine the mean distance. Together they are used to estimate the objects relative Cartesian coordinates.

The robot's visual system is also required to track the. Due to the robot motion, in some cases, the object may get out of the camera view. In that situation, the vision system uses the last information related to object position. Besides, the depth is calculated by the robot position and the last object position acquired from the camera.

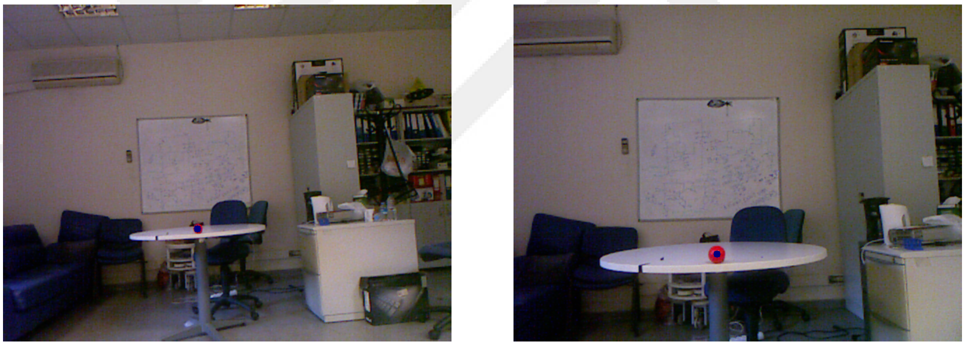


Figure 3.12: The detected and marked object with the vision system

4. BODY AND ARM COORDINATION IN OBJECT MANIPULATION

The proposed approach aims to generate human-like reaching movements for manipulation tasks. This requires the coordination of the robot's body and arm. The methodology consists of three related parts. We assume that the robot is able to determine the relative Euclidean coordinates of the object of interest. Additionally, objects must be graspable for the robot's two-fingered gripper. Due to configurational constraints related to robot arm, the approach assumes that no obstacle prevents the smooth manipulation.

Firstly, the robot uses a color-based object sensing algorithm for object detection and tracking. The vision system finds the position of the object relative to the manipulator. After that, during the robot motion for approaching, it tracks the object by marking. When the object is far away, the robot may not obtain the exact position of the object. In some cases, due to the robot motion, the object may get out of the camera view. If the vision system determines the position of the object once, the robot assumes that the object does not change any position and uses the previous object information.

Secondly, the proposed algorithm requires an input movement and makes it more natural by considering a spatio-temporal relationship, which imitates the coordination of the muscularly linked human bodies. The control model based on the distance between the robot and target enables the robot to perform concurrent movements of body segments during the reaching for the manipulation. The depth data is used for constructing the model considering the object and the robot location so that the object will be reachable with the motion of the robot body. Namely, if the object is beyond the reach, the robot requires body motion to make it enable to move towards object. In the distance between robot and target, the robot arm begins to accompany the robot body. The robot then ends its motion, though the arm continues its movement to reach the object.

Finally, we propose a behavior-based control by decomposing the tasks on several sub-tasks to execute. The behaviors are specified in the form of a target region to be reached by the robot arm. Assuming that the object is graspable and the robot position is within the reachable space with the robot arm, the tasks such as take-and-put, take-and-shake, pull, and knock defined by the user are performed.

4.1. Robot Body-Assisted Reaching Movements

This study deals with the problem of proposing a reaching planning for a mobile robot to manipulate graspable objects through its arm. Though, that planning requires the coupling of body segments including, robot body and manipulator. By modeling the control law on the human reaching characteristics, we demonstrated that the autonomous robot itself could achieve a manipulative task pre-defined before in a human-like way.

Body-assisted reaching might be the task in which different segments move in parallel. Depending on the location position of the object, reaching movements often need to be coordinated with trunk motion. In that coordination, relative to the object position, the arm does not attain the object, the trunk may involve the reaching movements to provide the arm displacement. The trunk motion begins before the arm movement by around ten milliseconds (ms) and continues after the arm reaches the object [24]. Additionally, the target distance at which the trunk involves the reaching movement corresponds to a distance equal to nearly 90 % the length of the arm [11]. These findings point out that there is an inherent coupling between human trunk and arm during the reaching for manipulation.

Based on the human's reaching properties, the robot's body motion is combined with the robot arm to move. That motion can contribute to the manipulator by enhancing the boundaries of the workspace and transporting the end-effector. The robot requires sensation pose of the object relative to an external coordinate system;

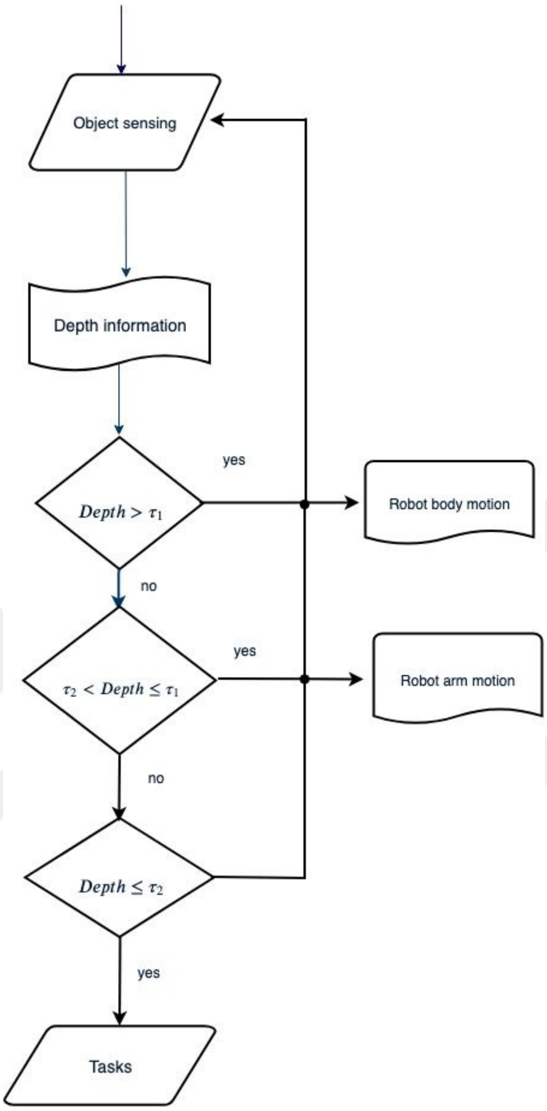


Figure 4.1: Flowchart of coordination of body and arm movements

it positions itself so that the object is reachable and then moves the associated joints leading to the desired movement. The robot's body motion starts at the beginning of the reaching and endures until the target is reachable for the manipulator. The control begins with the object sensing that covers to obtain the object and determine the position of it.

Let us consider a task that the robot seeks to manipulate an object, which is not in the reach space of the arm. It means that the robot can not grab the object without moving its body. The control begins with the object sensing that covers to obtain the object and determine the position of it. Relative to object position, the vision system calculates the depth information between the robot and target. While the robot body is approaching for the target, in the distance τ_2, where the robot body involves reaching, the robot arm begins a synchronous action with the body of the robot. When the target becomes reachable for the manipulator, the robot adjusts its orientation relative to the object and terminates its body motion in the distance τ_1. After that, the robot arm moves towards the target and performs its task. Figure 4.1 shows the whole process of the coordination of body and arm movements during the reaching for manipulation.

4.2. Robot Motion

The robot requires body motion when the target is beyond the fully extended manipulator. When the robot approaches the target, it shall bear human features in segmental control including, manipulator and robot body. It implies that the robot motion shall appear natural and smooth as well. However, robot motion is not straight forward since the mobile robot has the non-holonomic constraints. Besides, it is stated that a non-holonomic robot can not be stabilized by a state feedback control law [25]. Therefore, based on the previous work by [26], it is planned to use a kinematic position control law for comfortable motion of the mobile robot so that the robot can move the desired pose in space. The position of the differential drive mobile robot shown in Figure 3.2 is c_1 and c_2 and orientation α of the robot can be calculated using Eq. 4.1,

$$\begin{aligned} \dot{c_1} &= vcos(\alpha) \\ \dot{c_2} &= vsin(\alpha) \\ \dot{\alpha} &= \omega \end{aligned} \tag{4.1}$$

The robot R can be considered as a unicycle model which is driven by two independently parallel wheels such that linear velocity v and angular velocity ω can be controlled separately. Consider a sensor, which is mounted on the robot and an object O fixed at a distance r away from the robot. Let $\theta \in S^1$ defined in $(-\pi, \pi]$ be the orientation of O with respect to the line of sight from the sensor to the object. Moreover, let δ defined in $\in (-\pi, \pi]$ be the orientation of the robot heading relative to the line of

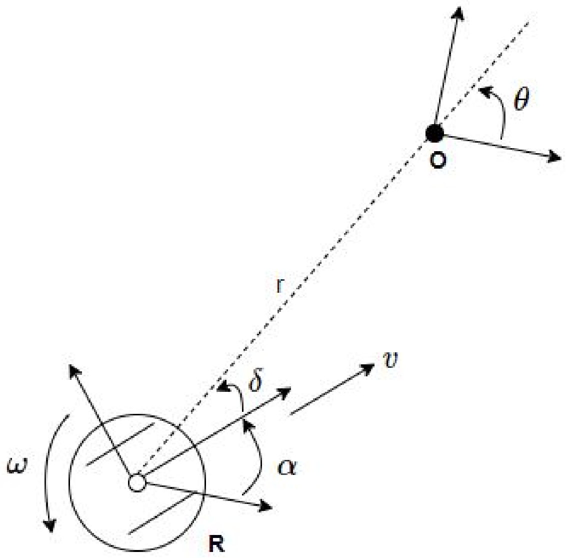


Figure 4.2: Polar coordinate system relative to the sensor

sight, as shown in Figure 4.2. By representing the robot kinematic in polar coordinate system, the distance, r, and robot orientation relative to the object O, and the defining α as the angle between robot principal axis and the distance. The angular velocity ω and linear velocity v are considered as the control variables.

Consequently, the kinematic controller proposed enables the mobile robot to follow a trajectory described by its velocity profiles as a function of time.

4.3. Control Law for Coupled Motion of Robot Body and Arm

The target reaching movements take on an essential role in mobile robots to gain general utility. In a primary task such as take-and-place requires reaching movement planning to drive the manipulator to a given target position. When it is considered that robots will be assigned to perform tasks such as serving and assisting, robots should have human-compatible and adaptable movement abilities so that they might work in a human environment without any need and changing the environment. Thus, the generation of human-like reaching movements is essential in robotics.

Humans inherently perform the reaching tasks by sensing the pose of an object relative to an external coordinate system; they then need to locate their trunks so that the object is reachable and then move the associated limbs leading to the desired movement of the arm. This natural process demonstrates that the coordination necessary to execute an everyday task such as picking a glass requires the integration of a number of body segments, including trunk and arm. Humans have multisegmental control strategies associated with target reachability. When a spatial target is selected for manipulation, body segments are evaluated for the contributions they can make to the task.

The fact that the reaching movements require the collocation of the considerable number of arm muscles and coupling body segments makes it highly redundant relative to the task. Though, biological systems generally solve the redundancy issue by applying the principle of biological inspiration-synergy [4, 5]. It has been noted that certain trajectories are preferable from the infinite amount of alternatives when executing arm movements [27, 28]. These behaviors can only be explained through inherent optimization that manages the acquisition of motor skills in humans. In terms of robotics, the synergy concept imposes particular constraints such as velocity on the control variables of joints related to the tasks consisting of the primary movements. Hence, our work is based on the resolved kinematic redundancy and redundant actuation of a given robot system, applying a biologically inspired synergy approach together with an optimization procedure.

The proposed control approach has three phases in which the robot moves, approaches, and reaches. To model that approach, we propose three connected regions based on the distance where the robot body and arm are coupled for the reaching movement and the stagnation distance where the robot body terminates its motion. In the far zone, the robot body plays the primary role in positioning the manipulator at the target. Since the robot does not participate in reaching movements, the robot's body motion is required for approaching the object. In the intermediate zone, the robot body involves the reaching. In the near zone, within the robot arm length, only arm movements take part in reaching towards the object for manipulation. Figure 4.3 demonstrates the zones in the approach.

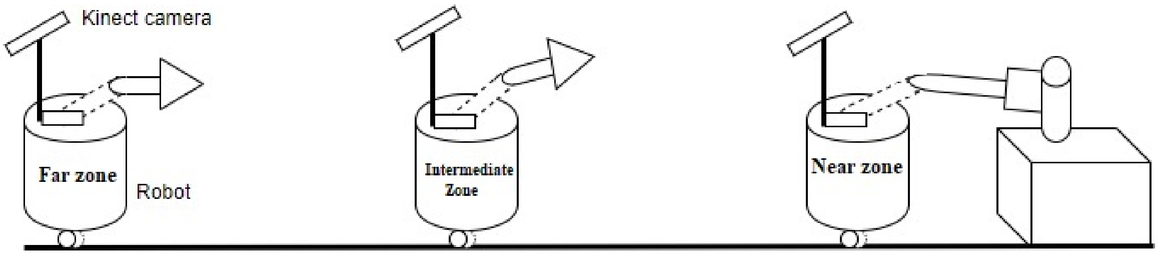


Figure 4.3: Robot arm and body coupling vs distance

- Far zone: When the robot is too far away from the target, it can not grab the object without moving its body. Therefore, the robot should act for it.
- Intermediate zone: While the robot moves for the target, in a pre-determined distance, where the robot body motion involves reaching, the robot arm begins the simultaneous and coordinated movement with the body of the robot.
- Near zone: When the target becomes reachable for the robot manipulator, the robot adjusts its orientation and position relative to the object and terminates its body motion. After that, the robot arm moves towards the target.

$$\varphi(q) = \gamma_2(d(c, g) - \tau_1))^2 + \gamma_1||e(q) - g||^2 \tag{4.2}$$

The approach to coupled motion of the robot body and arm involves constructing potential artificial fields (APF) which are designed to attract the mobile robot to the desired goal. The coupled motion can then be guided by considering the gradient of this potential function. APF represents the energy of the system and generating appropriate joint velocities on the robot so that the energy of the system is minimized and reaches its minimum value at the goal position. The information of free space and goal is encoded in the form of potential function, called a coupled function that connects the robot body and manipulator motion during approaching and reaching for the target. $e(q) = [e_X(q), e_Y(q), e_Z(q)]^T$ represents the current position of end-effector.

What we have is a potential function, $\varphi(q)$, which takes a parameter q for each joint of the robot. That parameter represents the current angles of the joints. Given a specific configuration of the joints, q, the function $\varphi(q)$ returns a value that indicates how far the end-effector of the robot manipulator is from the target object g. The objective is to find the values for q that minimize the potential function.

Equation 4.2 shows the robot body motion that enables the robot to approach and go into the coupling motion workspace. The coupling motion workspace can be defined as workspace in which the robot body and manipulator operate concurrently. Equation 4.2 points out the coupling motion of the robot arm and body. Whereas the robot terminates motion in the distance τ_1, the manipulator proceeds the motion. $d(c, g)$ is the distance between the robot position $[c_1, c_2]^T$ and the goal position $[g_x, g_y]^T$.

$$d(c, g) = \sqrt{(c_1 - g_x)^2 + (c_2 - g_y)^2} \tag{4.3}$$

γ_1 and γ_2 variables scale the velocity of the robot and manipulator joints when the robot is in the coupling motion workspace.

$$\gamma_1 = \begin{cases} 0 & d(c, g) \geq \tau_2 \\ \gamma_1^* & \tau_1 \leq d(c, g) < \tau_2 \\ 1 & d(c, g) < \tau_1 \end{cases} \tag{4.4}$$

$$\gamma_2 = \begin{cases} 1 & d(c, g) \geq \tau_2 \\ \gamma_2^* & \tau_1 \leq d(c, g) < \tau_2 \\ 0 & d(c, g) < \tau_1 \end{cases} \tag{4.5}$$

The first part of the Eq. 4.2 is related to robot locomotion. Until a constant distance, τ_1, the robot performs body motion to reach the object. After a constant distance, τ_2, robot manipulator accompanies the robot locomotion. It implies that between distances τ_1 and τ_2, the robot arm and body motion act together in a coordinated and simultaneous way. The distances denoted by τ_1 and τ_2 are determined by considering operating measuring range of camera and the feature of body-assisted reaching movement, respectively. To obtain joint velocities during coupled motion based on distance between robot and goal, the gradient of the potential function is used.

To find a minimized solution for the potential function, it is required to take the gradient of the function $\varphi(q)$ with respect to the robot joints. Since the robot has five joints, two body joints and three arm joints, we will have the function that takes five parameters: q_1, q_2, q_3, q_4 and q_5. Then, the gradient $\nabla\varphi(q)$ is given by

$$\nabla\varphi(q_1, q_2, q_3, q_4, q_5) = [\nabla\varphi_{q_1}(q), \nabla\varphi_{q_2}(q), \nabla\varphi_{q_3}(q), \nabla\varphi_{q_4}(q), \nabla\varphi_{q_5}(q)] \tag{4.6}$$

where

$$\nabla\varphi_{q_1}(q) = \frac{\varphi(q_1 + \Delta q_1, q_2, q_3, q_4, q_5) - \varphi(q_1, q_2, q_3, q_4, q_5)}{\Delta q_1} \tag{4.7}$$

$$\nabla\varphi_{q_2}(q) = \frac{\varphi(q_1, q_2 + \Delta q_2, q_3, q_4, q_5) - \varphi(q_1, q_2, q_3, q_4, q_5)}{\Delta q_2} \tag{4.8}$$

$$\nabla\varphi_{q_3}(q) = \frac{\varphi(q_1, q_2, q_3 + \Delta q_3, q_4, q_5) - \varphi(q_1, q_2, q_3, q_4, q_5)}{\Delta q_3} \tag{4.9}$$

$$\nabla\varphi_{q_4}(q) = \frac{\varphi(q_1, q_2, q_3, q_4 + \Delta q_4, q_5) - \varphi(q_1, q_2, q_3, q_4, q_5)}{\Delta q_4} \tag{4.10}$$

$$\nabla\varphi_{q_5}(q) = \frac{\varphi(q_1, q_2, q_3, q_4, q_5 + \Delta q_5) - \varphi(q_1, q_2, q_3, q_4, q_5)}{\Delta q_5} \tag{4.11}$$

and Δq_i where $i = 1, .., 5$ are the sufficiently small values. To minimize potential function, it is required to move in the opposite direction of the gradients.

- $\varphi(q)$ is a multivariable defined function and differentiable in a neighborhood of a point q. For small enough κ, then $\varphi(q_{i_k}) > \varphi(q_{i_{k+1}})$. The term $\nabla\varphi(q_{i_k})$ is subtracted from q since it is moved against the gradient toward the minimum. q_{i_k} represents angle of i-th joint at k-th sample.

$$q_{i_{k+1}} = q_{i_k} - \kappa \nabla \varphi(q_{i_k}) \tag{4.12}$$

where κ is a non-negative scalar minimizing. That parameter controls how fast the function paramater moves away from the ascending gradient.

- $\dot{q}$ is the vector directed toward the goal with magnitude linearly related to distance from current position to goal.

$$\dot{q} = -\nabla \varphi(q_{i_k}) \tag{4.13}$$

- $\dot{q}$ converges to zero as q approaches goal.

4.4. Generation of Human-Like Arm Movements

The latest trends in humanoid robotics works to make humanoids really look like people and even more so. There is, however, a large distinction between a robot's appearance and the capacity to behave like a person. With robots being brought more into culture today, new problems emerge, such as robot anthropomorphism. Several studies [29–32] proposed that humanoid robots should imitate human social abilities and be able to deliver consistent behaviors. This is partly due to the reality that human-like movements promote natural human-robot interaction by making it easier for the humans to interpret the robot's motions in terms of objectives. Besides, such movements need to look like human beings; otherwise, individuals may misunderstand the significance of behaviors. Human-like arm movements are the ultimate demands for humanoid robots to do as human beings do. Therefore, the human-inspired control of the robot manipulator should achieve human - motion characteristics for their acceptability by humans.

Robot manipulators are excessively nonlinear and composed systems in which their control is complex and challenging. As the robot manipulators have adequate DoFs to perform the tasks by providing final position and orientation for the target, the workspace will be restricted due to the mechanical constraints (, *i.e.*, singularities and joint limit avoidance) and control issues (, *i.e.*, stability). Some robot manipulators can be redundant since they have more joint variables than the end-effector has DoFs. These redundant manipulators have multiple solutions for the inverse problem. Thus, it is challenging to plan a comprehensible trajectory that considers both mechanical and control problems. Though, an effective controller for kinematic redundant robotic manipulators should, therefore, integrate robot dynamics and be consistent with multivariate uncertainties connected with dynamics and internal disturbances.

Velocity control the robot manipulator is essential to execute a specific task that requires the reach-to-grasp movements. Hence, in this work, point-to-point control of a three-link redundant robot arm is studied. A gradient-based optimization method is proposed for the human-like movement generation by minimizing the error between the desired position and initial position of the end-effector. Our strategy seeks to imitate the spatio-temporal coordination of human joints that allow the arm to move an area of interest for the task. Let us consider the pulling task. The robot manipulator requires prehension and drags the object along a distance. Let the object be in the workspace of the robot arm. Figure 4.4 shows the case in which the object is in the workspace of the arm. It implies that the robot can reach the object without moving its body. Then, the robot should perform only arm movements to pull the object and finish the task. In that scenario, the control law Eq. 4.2 for the coupled movements of the robot body and arm becomes as below.

$$\varphi(q_A) = ||x(q) - g||^2 \tag{4.14}$$

This solution described by Eq. 4.14 for the joint rates generates the desired motion at the end-effector. g represents the position of the object relative to the camera onto the robot. The robot end-effector must be at the object position in the final configuration.

Then, we can rewrite Eq. 4.14 as below

$$\varphi(q_A) = ||J_e q_A - g||^2 \tag{4.15}$$

where J_e is Jacobian of the end-effector. Then the cost function can be expanded in the following form

$$\begin{aligned}\varphi(q_A) &= (J_e q_A - g)^T (J_e q_A - g)\\ &= q_A^T J_e^T J_e q_A - 2q_A^T J_e^T g + g^T g\end{aligned} \tag{4.16}$$

The partial derivative of the cost function with respect to q_A^T vanishes for q_A that minimizes $\varphi(q_A)$. Solving the partial derivative of the cost function $\varphi(q_A)$ for the unknown q_A results in

$$q_A = 2(J_e^T J_e)^{-1} J_e^T g \tag{4.17}$$

The mobile robot performs reaching movements for prehension. After termination of the robot locomotion that manages approaching for the object as proposed in the coordinated control law, the robot manipulator continues its movement for the manipulation task. During this process, a path that lies in the manipulator's working space must be assumed. To solve the coordinated control problem for the robot manipulator, it is obtained $\dot{q}_A$ the joint velocities profile that corresponds to the assigned manipulator velocity profile $\dot{q}_e$.

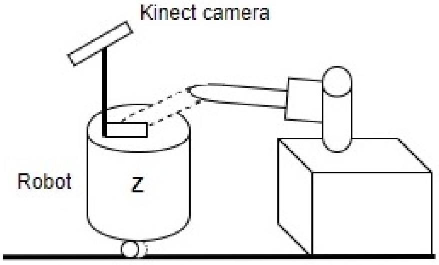


Figure 4.4: Robot manipulation task

To control the robot manipulator motion, the joint trajectories and velocity references for the robot arm are required. It implies that position control of robot arm joints is ensured by joint velocity profiles obtained from the gradient of the APF. To find the joint positions required to bring the gripper to the object position, the joint rates calculated by the gradient descent method at the velocity level must be integrated. The mathematical expression for the position of the end-effector q_e is described as finding q related to robot body and arm such that

$$q_e = f(q_A, q_B) \tag{4.18}$$

When the joint velocity profiles of the manipulator are completely determined by the control law, the corresponding joint position profile could be attained by time integral

$$q_A(t) = q_A(t_0) + \int_{t_0}^{t} \dot{q}_A(\tau) d\tau \tag{4.19}$$

Since the problem is solved by integrating the manipulator's joint velocities, initial conditions for each joints are needed. Physically, the initial conditions of joints show the initial position and orientation of the manipulator from which the motion toward the target position begins. Though, real-time implementation of robot manipulator control makes the Eq. 4.19 more likely that a discrete-time sequence of samples $\dot{q}_k$ of joint velocities at the time instants t_k will exist. Therefore, a discrete-time approximation method corresponding to the time integral is acquired. Using Euler's forward formula, the time integral becomes

$$q_{A_k} = q_{A_{k-1}} + \dot{q}_{A_{k-1}} \Delta t \tag{4.20}$$

where Δt is the step time. The large step time will result in time delay and also decrease the position accuracy of the robot manipulator. Thus, it must be chosen sufficiently small. However, shorter step time will complicate the computation.

4.5. Human Reaching Characteristic and Robot Control Law Comparison

The lack of dexterous human arm articulation problematizes to generate human-like motion in robots. Thus, to design a human behavior based controller for robots to perform arm movements, it is required to consider the human spatiotemporal characteristics such as bell-shaped velocity profile of the hand in reaching.

In daily life, there is an immense potential to help and unload people to carry out their duties through the provision of service or care. A challenging task in the development of adaptive and self-sufficient robots is to generate movements that fit naturally to a daily human environment. Such a system would require to generate motions based on the current task, the type of object, while considering reaching pattern of the human for manipulation.

In humans, reaching movements require commanding of various DoFs of motor components so that a desired trajectory and inter-joint coordination may be decided from the feasible strategies pointing to the end. The control of the arm movements is challenging due to the requirement of coordinating multiple muscles. According to one approach, by mapping the initial point and final point into the muscle activation through the synthesis of muscle synergies, the central nervous system might simplify the control of reaching [33]. According to another approach, a muscle synergy contains the coordinated activation of group muscles with precise time-varying profiles [5]. These studies demonstrate that reaching movements require joint coordination to reduce complexity.

In the study [34], it is experimented to examine velocity of the hand in an unconstrained environment. The pose of the object is the different shown as Figure 4.5. This experiment points out that the hand velocity has a belly-shaped profile regardless of the target distance. The target distance determines the time of the reaching and magnitude of the hand velocity. In conclusion, there is a coordination between the trunk and arm while reaching towards objects placed outside the arm workspace.

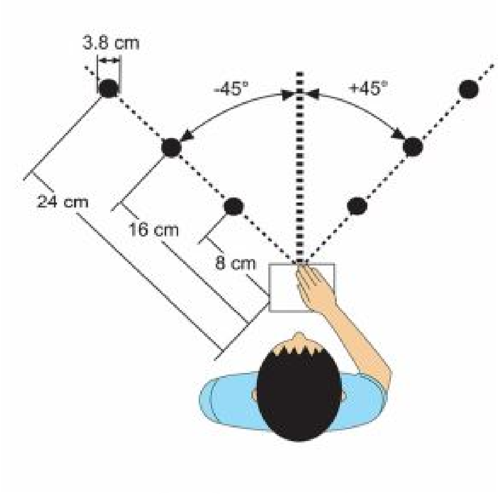


Figure 4.5: Experimental setup for human studies: Six targets in 2 directions and 3 distances were shown [34]

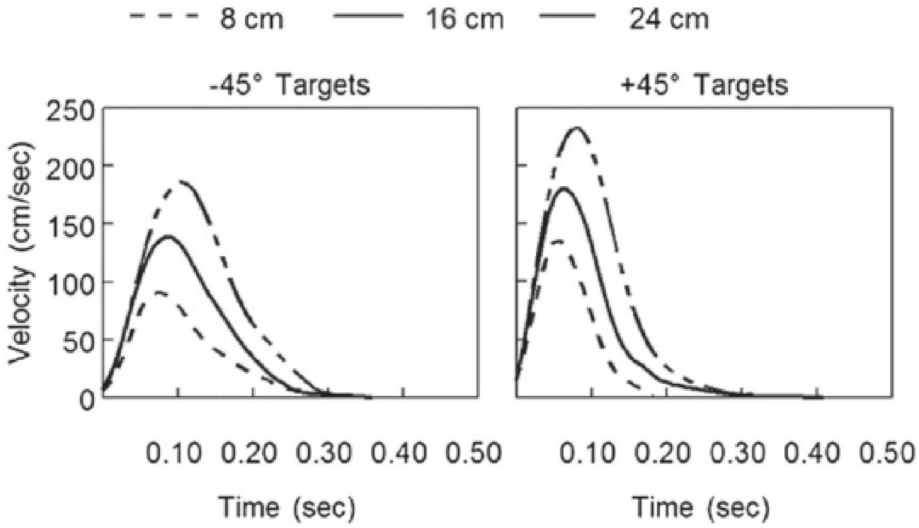


Figure 4.6: Mean velocity trajectories for one participant reaching in real-world environment [34]

In [35], the authors obtained that the arm and trunk move in parallel in reaching movements. The trunk and arm movements are generated sequentially. Besides, they found that the trunk began to contribute to the hand transport only at a later phase of movement, beginning at around the time when the hand reached peak velocity. Figure 4.7 shows the parallel hand and trunk movements during the reaching in two cases: when the trunk is free and when the trunk is blocked.

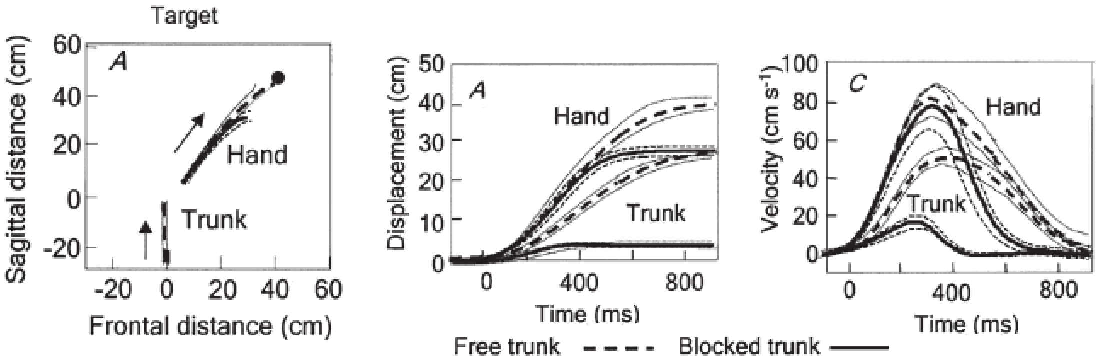


Figure 4.7: Typical kinematic effects of the trunk in movements to targets located beyond the reach of the arm [35]

4.6. Behavior-Based Tasks

Behavior-based control executes task decomposition to structure the overall control scheme as layers of behavior modules, which are defined as a task-performing controllers. The controllers complete a mechanism that gathers sensory information related to the unfinished task to compute actuator outputs. It is required to model the behavior modules firstly to construct a behavior-based system. Later, it proceeds to design an arbitrator to combine the individual results from different behavior modules into commands.

The controlling a robot manipulator in a human-like way is a complicated problem. For imitation, the process of learning new movement patterns and skills by observation should be present. In our approach, we assume that motion is structured for control by primitive modules i) action primitives, ii) behavior primitives. Action primitives refer to information of simple actions that can be captured from humans. Translation and extension might be illustrative examples. Action primitives can be combined and sequenced to form complex movements that will constitute behavior primitives such as dragging. Our robot can perform five basic low-level behaviors such as to take, put, shake, and pull. It is required to combine these behaviors into an upper level, including taking-putting, taking-shaking, pulling, and knocking.

We deal with the different types of primitives to generate full-arm movements and sequences, and parallel movement primitives to accomplish more complex motion tasks. According to our approach, each action primitive represents a set of joint trajectory basis functions; these basis functions are extracted by analysis of human motion capture data. Since our ultimate goal is to develop efficient methods for generating natural, or physically meaningful, motions for a wide variety of our robot. We have attempted to pattern human movements and apply them to robot movement environment so that we can cope with the complexity of the robotic arm control and learning. Furthermore, we will obtain a computationally, efficient, modular, and reusable design. The general features of our approach are i) to select and classify a set of movement primitives, ii) to extract basic motions considering observed human movements, iii) combine these motions sequentially to achieve more complicated movements for a task, iv) to perform collection of the motions to resemblance to the natural human movements.

The action primitives are necessary actions adopting human motions into the robot. A set of action primitives allows the robot manipulator to perform complex movements. While action behaviors provide a valuable structuring of control of the robot arm, an important question remains as to which movements should may the action primitives. Our robotic manipulator with the end-effector can perform four primitive behaviors that correspond to the human arm motion. These are extension, flexion, gripping, and releasing as shown in Table 4.1

Table 4.1: Primitive actions and behaviors in the manipulator

Primitive actions	Low-level Behaviors
Flexion	Flexion + Gripping
Gripping	Gripping + Extension
Extension	Extension + Gripping
Releasing	Releasing + Extension

We will study the sequence of movements that must be made to create a controlled movement between action primitives. We need to approach the problem of composing the action primitives so that we can construct behavior primitives to achieve more complex manipulation tasks. For example, extension and gripping may constitute a motion to hold an object. For the low-level behavior-based design, it is required to combine the primitive actions. The concept of primitive actions and low-level behaviors is listed below. For example, extension and gripping may constitute a motion to hold an object. Thus, the low-level behaviors may construct the movements such as pull, put, take, and shake. The parameters related to simulation and real-time experiments are the same with the method. These parameters are shown in Table 4.2.

Table 4.2: The parameters and descriptions

Parameter	Description
γ_1	Scale factor for the robot arm movement
γ_2	Scale factor for the robot body motion
τ_1	Distance threshold for robot body motion
τ_2	Distance threshold for reaching

5. EXPERIMENTS AND RESULTS

The experimental evaluation is conducted in two stages. First, simulations are performed to validate the proposed approach and compare the resulting behavior with that of humans. Real-time experiments conducted on a mobile robot follows this. The values of parameters used in the experiments and simulation are shown in Table 5.1.

Table 5.1: Parameter values

Parameters	Value	Unit
γ_1	0.73	-
γ_2	0.37	-
τ_1	1.1	meter
τ_2	0.5	meter

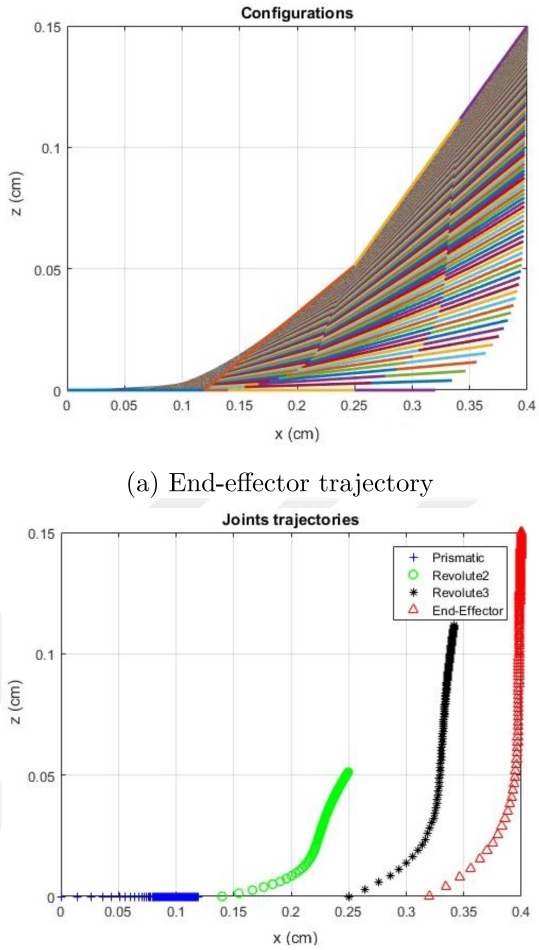
5.1. Simulation Results

The extensive simulations are conducted to validate the proposed approach. Robot's kinematics is simulated using Runge-Kutta. Velocity profiles for the robot's various joints are obtained and then compared in form with those of humans. As discussed, the velocity profiles of human joints exhibit bell-shaped tangential velocity profiles as a function of distance to the object-of-interest. The controller is designed to have a similar pattern. This is verified in the simulations.

A sample scenario is as follows: Let P_i and P_f be the initial and final positions of robot arm, respectively:

$$P_i = \begin{bmatrix} x_i \\ z_i \end{bmatrix} = \begin{bmatrix} 0.32 \\ 0 \end{bmatrix}, \text{ and } P_f = \begin{bmatrix} x_f \\ z_f \end{bmatrix} = \begin{bmatrix} 0.40 \\ 0.15 \end{bmatrix}$$

The trajectory of the robot's end-effector and those of the joints are presented in

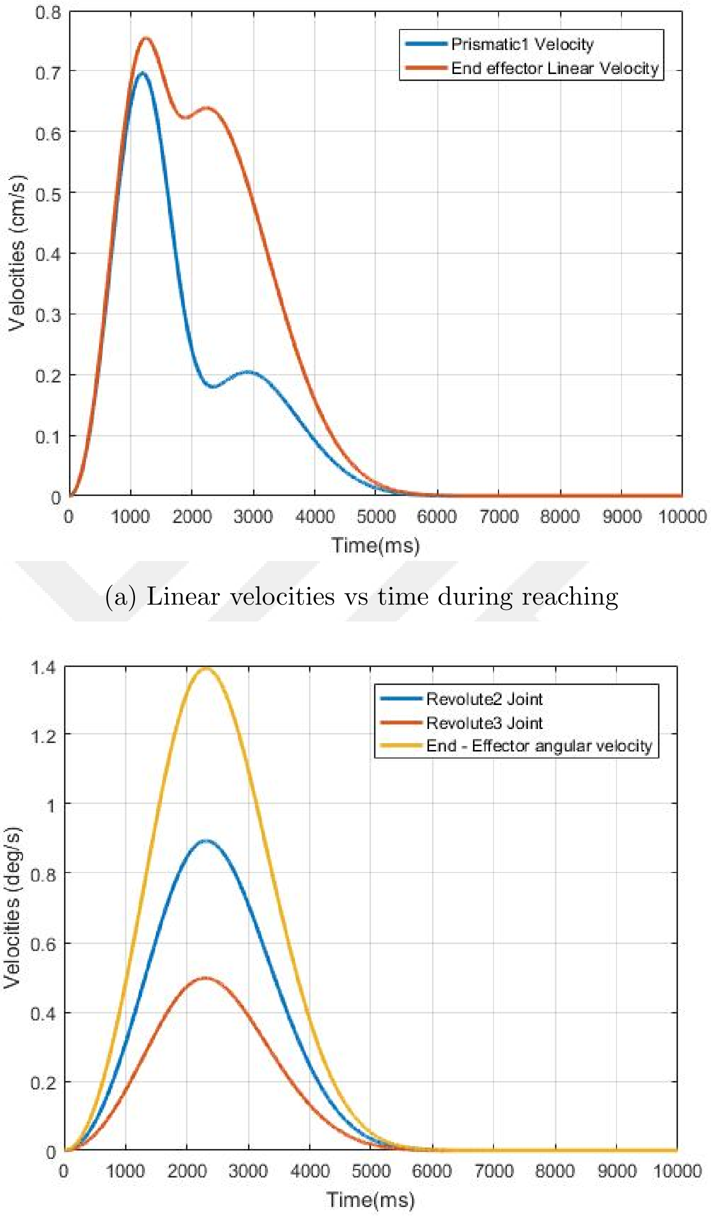


(a) End-effector trajectory

(b) Joint trajectories

Figure 5.1: Manipulator and joint trajectories during reaching

Figures 5.1a - 5.1b, respectively. While the robot arm performs human-like reaching movements, the robot and arm joints follow these trajectories. The distance between the robot arm and the target reduces when the end-effector approaches it. The velocity of the robot's end-effector and those of the joints versus time are presented in Figures 5.2a - 5.2b, respectively. Similar to the bell-shaped velocity profile of the human hand, the speeds increase to a point to the distance; after a peak value, they decrease. The velocity of the robot's end-effector and those of the joints versus distance are shown in Figures 5.3a - 5.3b, respectively. These results show that the proposed control approach ensures the human reaching features such as quasi-straight line trajectory and bell-shaped tangential velocity profile of hand.



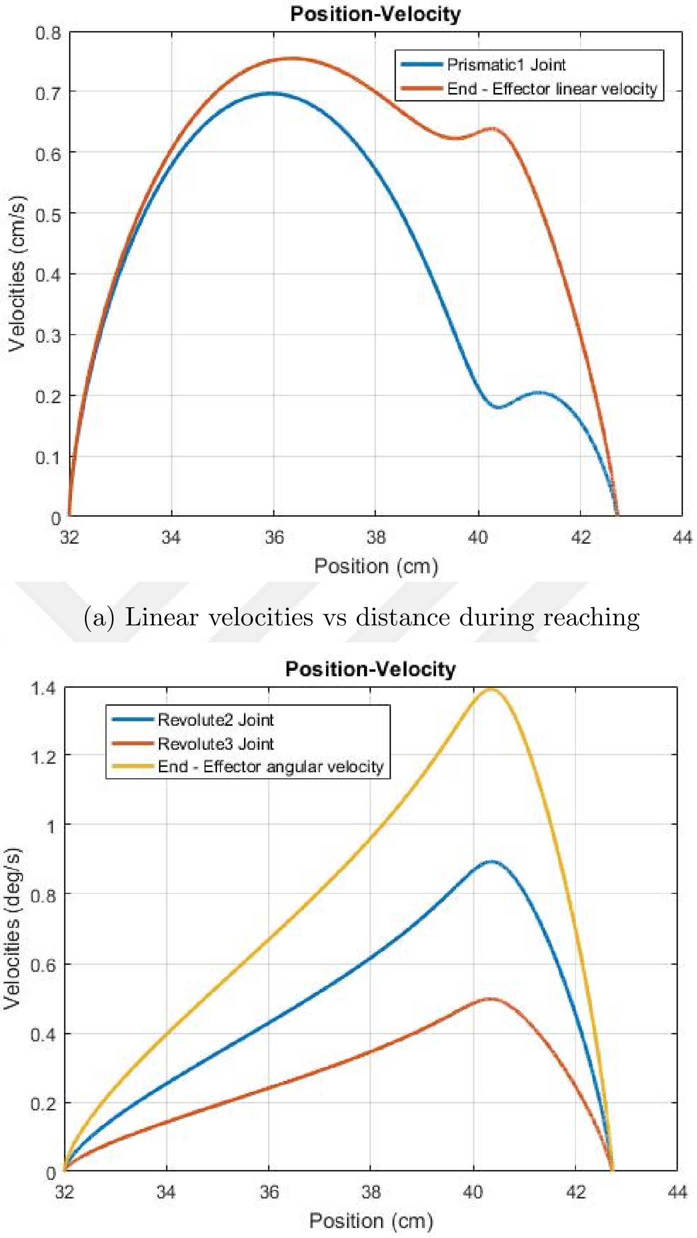
(a) Linear velocities vs time during reaching

(b) Angular velocities vs time in reaching

Figure 5.2: Manipulator and joint velocities vs time during reaching

5.2. Robot Experiments

In the robot experiments, the robot's task is to do one of five basic manipulation tasks (take, put, pull, push, and shake) with an object-of-interest. The experiments are conducted with a differential wheeled mobile robot with a PRR arm and 1 DOF, as explained in Chapter 3. Two scenarios are considered by considering the robot's initial position and object of interest's position.



(a) Linear velocities vs distance during reaching

(b) Angular velocities vs distance during reaching

Figure 5.3: Manipulator and joint velocities vs distance during reaching

5.2.1. Varying Object's Position

In the first scenario, the robot's initial position is fixed while the object of interest's location is varied, as shown in Figure 5.4. The latter is done considering three different zones: near zone in which only arm motion will suffice as represented by the red area. Intermediate zone in which both body and arm motion are required concurrently, as shown by the orange area. Finally, the far zone in which

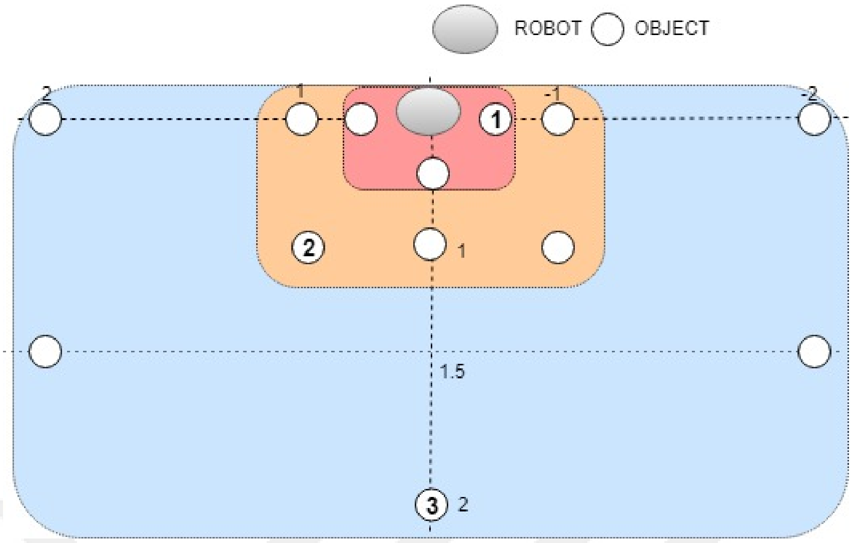


Figure 5.4: Varying object's position

the robot should exhibit only body motion. In each zone, we consider a set of different positions that vary in the respective orientation to the robot's heading, as shown in the figure. As the workspace of the robot's covers only x and y planes, all objects are placed at a reachable and the same height. In the near zone, three different positions are considered as: $(0.5, 0, 0.78)^T, (-0.5, 0, 0.78)^T$ and $(0, 0.5, 0.78)^T$. The units are meters. In the intermediate zone, five different positions are considered: $(0, 1, 0.78)^T, (1, 1, 0.78)^T, (1, 0, 0.78)^T, (1, -1, 0.78)^T$, and $(0, -1, 0.78)^T$. Finally, in the far zone, five positions are considered: $(0, 2, 0.78)^T, (1.5, 2, 0.78)^T, (2, 0, 0.78)^T$, $(1.5, -2, 0.78)^T$, and $(0, -2, 0.78)^T$. For each object position, experiments are repeated ten times. In the tests, the success of the task relies on whether the robot achieves the task or not. Samples of the visual feedback as a function of object distance are as shown in Figure 5.5. As expected, as the robot approaches the target, it becomes more apparent in the incoming images.

The results are presented in Table 5.2. The average success rate of the manipulator is computed to be 66.4%. Task completion failures are observed to be due to problems in the visual feedback - namely, if the object-of-interest cannot be detected, then the task terminates. Another reason is the limitation in the measuring range of the camera in the near zone since the camera can not compute depth data under approximately 50 centimeters. We also compute average task completion times and

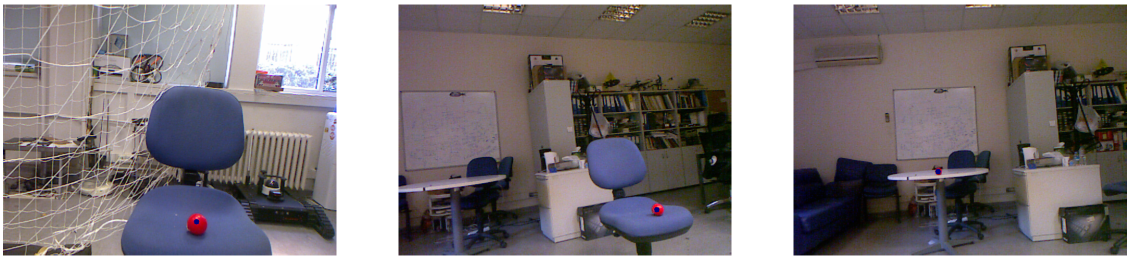


Figure 5.5: Appearances from robot's initial position with the object-of-interest (blue point) at positions (1), (2), and (3), respectively.

Table 5.2: Varying object positions: Experimental results

Zones	Task	Success Rate (%)	Average task time (sec)	
			Proposed Approach	Without coordination
Near zone	Take	76.7	5.2	5.2
	Put	73.3	4.8	4.8
	Pull	70	4.7	4.7
	Push	76.7	5.4	5.4
	Shake	63.3	6.4	6.4
Intermediate zone	Take	64	12.1	16.4
	Put	60	11.4	14.4
	Pull	70	10.1	14.2
	Push	62	11	14.6
	Shake	56	12.5	15.7
Far zone	Take	64	19.2	23.5
	Put	66	16.5	22.7
	Pull	64	17.9	21
	Push	68	18.1	22.7
	Shake	62	20.1	25.4
Average		66.4	14.4	11.7

compare them with those that are obtained if the tasks are done without body and arm coordination, as is proposed. As expected, task completion times turn out to be identical in the near zone. However, in the intermediate and far zones, the advantage of the proposed approach becomes apparent due to the coordination of body and arm motion. In the intermediate zone, the gain is around 24% while in the far zone, this is around 20%. This is also expected since the task is completed with a greater overlap of body and arm movements. Sample cases of this coordination are shown in Figure 5.6. In the left figure, for an object-of-interest in the intermediate zone, body and arm motion occur concurrently, but body motion terminates approximately 3.6 seconds later. The right figure is for an object-of-interest in the far zone. In this case, the coordination time increases to 4.7 seconds. This suggests that as the robot is further away from the object, its arm movements are slower.

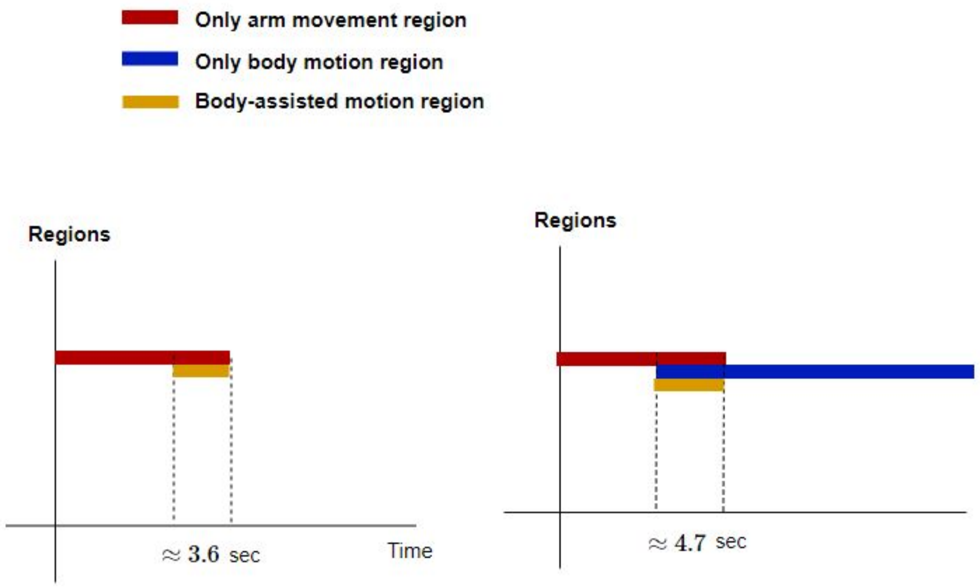


Figure 5.6: Body and arm motion vs time. Left: Object-of-interest in intermediate zone; Right: Object-of-interest in far zone.

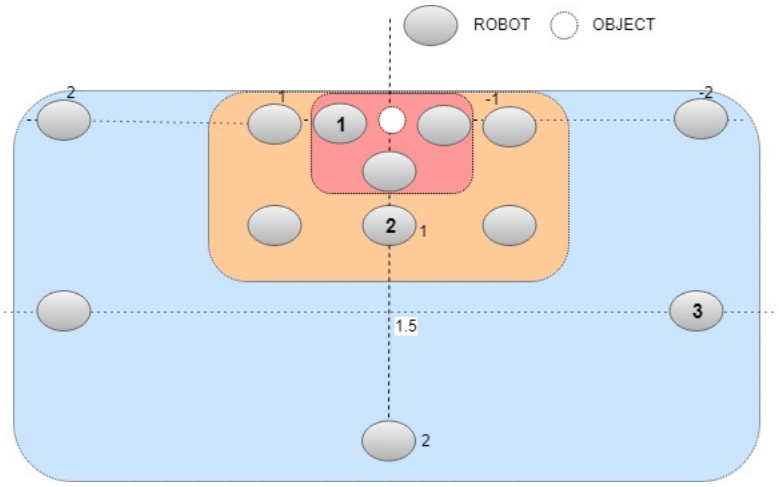


Figure 5.7: Varying robot's initial position

5.2.2. Varying Robot's Initial Position

In the second scenario, the object of interest's position remains the same while the robot's initial position is changed. This scenario intends to test whether the robot can perform the task when it is placed at different positions. Again we consider three different zones depending on whether only arm motion (near zone), only body motion (far zone), or both (intermediate zone) are required, as shown in Figure 5.7. In the near zone, robot's positions are varied as: $(0, 0.5)^T$, $(0.5, 0)^T$, and $(0, -0.5)^T$ meters. In the intermediate zone, its positions are varied as: $(0, 1)^T$, $(1, 1)^T$, $(1, 0)^T$, $(1, -1)^T$, and $(0, -1)^T$ meters. Finally, in the far zone, its positions are: $(0, 2)^T$, $(1.5, 2)^T$, $(2, 0)^T$, and $(1.5, -2)^T$, $(0, -2)^T$ meters. Again, all the tasks are repeated ten times for each different robot position. Appearances, as seen from various robot positions are as shown in Figure 5.8. Again, as the robot gets closer, the object-of-interest is seen larger.

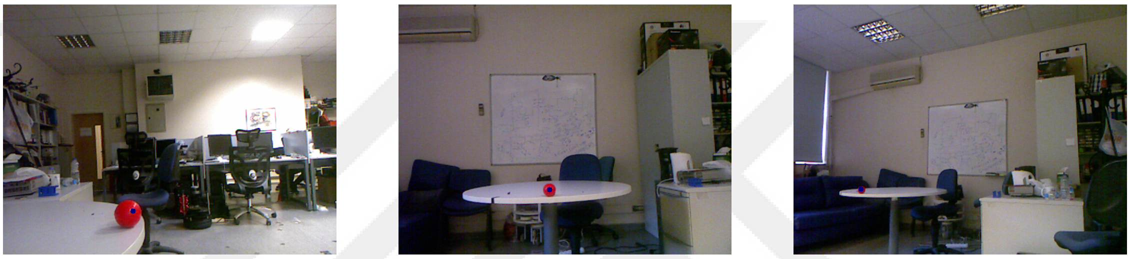


Figure 5.8: Appearances as seen from robot positions (1), (2), and (3) respectively with the object-of-interest (blue point)

Table 5.3: Varying robot's initial position: Experimental results.

Zones	Task	Success Rate (%)	Average task time (sec)	
			Proposed Approach	Without coordination
Near zone	Take	73.3	5.2	5.2
	Put	73.3	4.8	4.8
	Pull	70	4.7	4.7
	Push	73.3	5.4	5.4
	Shake	63.3	6.4	6.4
Intermediate zone	Take	64	11.2	15.1
	Put	62	10.9	14.9
	Pull	72	10.3	14.5
	Push	60	11.1	14.6
	Shake	56	12.1	15.4
Far zone	Take	62	17.5	22.5
	Put	62	17.4	21.2
	Pull	64	17.3	21.9
	Push	64	17.4	22.1
	Shake	58	19.1	24.9
Average		65.1	11.6	14.2

The results are presented in Table 5.3. The average task completion rate is computed to be 65.1%. As such, it is close to the previous scenario. Similarly, failures occur primarily to object not being detected, robot's limited grasping capability, and operating range of the camera in near zone. In some cases, the robot arm performed reaching movements; but it could not complete the task due to the mentioned issues. We also compare average task completion times and compare them with those that are obtained if the tasks are done without body and arm coordination, as is proposed. Again, task completion times turn out to be identical in the near zone. In the intermediate zone, the gain is around 25% while in the far zone, this is around 21%. Sample cases of this coordination are shown in Figure 5.9. In the left figure, for an object-of-interest in the intermediate zone, body and arm motion occur concurrently, but body motion terminates approximately 3.8 seconds later. The right figure is for an object-of-interest in the far zone. In this case, the coordination time increases to 4.8 seconds. This again validates the observation that with the proposed controller, as the distance between the robot and the object-of-interest increases, arm motion is executed more slowly.

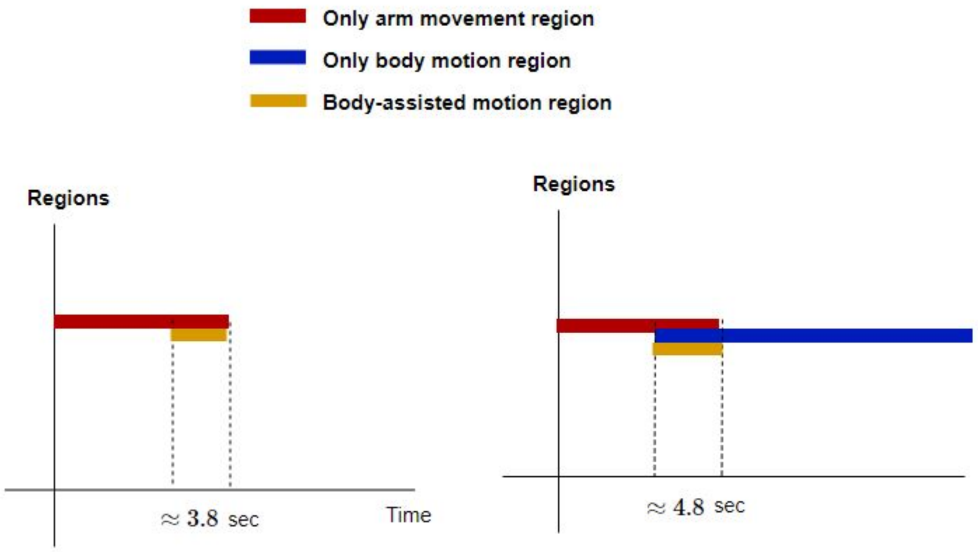


Figure 5.9: Body and arm motion vs time. Left: Robot's initial position in intermediate zone; Right: Robot's initial position in far zone.

6. CONCLUSION

This thesis is focused on object manipulation. We consider a differential type of mobile robot that is endowed with an arm and gripper. The robot has visual sensing so that it can determine the relative position of the object of interest. Results obtained from human manipulation studies inspire our work. First, it is observed that manipulation tasks include various basic modes of interaction with an object of interest. These include extension, flexion, gripping, release and translation. As such, complex manipulation tasks such as putting, pulling, pushing, and shaking are defined using the sequential composition of primitive operations. It is shown that through establishing a correspondence between the robot's and human's underlying manipulation mechanisms, the robot can be programmed to achieve these tasks. Second, humans are observed to achieve these tasks through the coordination of their body and arm movements. For this, a control approach in which the movements of the robot body and manipulator are coupled temporally and spatially is proposed. As such, if the object of interest is within the robot's reach, then only arm movements are made. If this is not the case, the robot starts moving its body. Depending on the vicinity of the object, this may be accompanied by arm motion or not. The control algorithm results in the robot's body and arm movements to be done in a coupled manner. The proposed approach is evaluated through an extensive set of experiments involving various manipulation tasks. The experiments point out that our integrated controller generates a solution to reaching a problem for the body-assisted manipulation that satisfies all the desired constraints, e.g., coordinated motion of the robot body and arm to pull a bottle or to pick up an object. As we demonstrated in the implementation of a real mobile robot, the synchronous control of robot motion and arm movements can be reliably executed.

In future work, we will include human-like movements with a highly articulated robot arm to accomplish more complex tasks. Additionally, we will extend the current framework to enable the robot to perform these same tasks in environments containing obstacles.

REFERENCES

1. Kaminski, T., C. Bock and A. Gentile, “The coordination between trunk and arm motion during pointing movements”, *Experimental brain research*, Vol. 106, No. 3, pp. 457–466, 1995.

2. Karniel, A. and G. F. Inbar, “A model for learning human reaching movements”, *Biological cybernetics*, Vol. 77, No. 3, pp. 173–183, 1997.

3. Desmurget, M. and S. Grafton, “Forward modeling allows feedback control for fast reaching movements”, *Trends in cognitive sciences*, Vol. 4, No. 11, pp. 423–431, 2000.

4. d’Avella, A., A. Portone, L. Fernandez and F. Lacquaniti, “Control of fast-reaching movements by muscle synergy combinations”, *Journal of Neuroscience*, Vol. 26, No. 30, pp. 7791–7810, 2006.

5. d’Avella, A. and F. Lacquaniti, “Control of reaching movements by muscle synergy combinations”, *Frontiers in computational neuroscience*, Vol. 7, p. 42, 2013.

6. Flash, T., Y. Meirovitch and A. Barliya, “Models of human movement: Trajectory planning and inverse kinematics studies”, *Robotics and Autonomous Systems*, Vol. 61, No. 4, pp. 330–339, 2013.

7. Campos, F. and J. Calado, “Approaches to human arm movement control—A review”, *Annual reviews in control*, Vol. 33, No. 1, pp. 69–77, 2009.

8. Seidler, R. D. and G. E. Stelmach, “Trunk-assisted prehension: specification of body segments with imposed temporal constraints”, *Journal of motor behavior*, Vol. 32, No. 4, pp. 379–389, 2000.

9. Wang, J. and G. E. Stelmach, “Spatial and temporal control of trunk-assisted

prehensile actions", *Experimental brain research*, Vol. 136, No. 2, pp. 231–240, 2001.

10. Adamovich, S. V., P. S. Archambault, M. Ghafouri, M. F. Levin, H. Poizner and A. G. Feldman, "Hand trajectory invariance in reaching movements involving the trunk", *Experimental Brain Research*, Vol. 138, No. 3, pp. 288–303, 2001.

11. Mark, L. S., K. Nemeth, D. Gardner, M. J. Dainoff, J. Paasche, M. Duffy and K. Grandt, "Postural dynamics and the preferred critical boundary for visually guided reaching.", *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 23, No. 5, p. 1365, 1997.

12. Reinhart, R. F. and J. J. Steil, "Reaching movement generation with a recurrent neural network based on learning inverse kinematics for the humanoid robot iCub", *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pp. 323–330, IEEE, 2009.

13. Albrecht, S., K. Ramirez-Amaro, F. Ruiz-Ugalde, D. Weikersdorfer, M. Leibold, M. Ulbrich and M. Beetz, "Imitating human reaching motions using physically inspired optimization principles", *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pp. 602–607, IEEE, 2011.

14. Guenter, F., M. Hersch, S. Calinon and A. Billard, "Reinforcement learning for imitating constrained reaching movements", *Advanced Robotics*, Vol. 21, No. 13, pp. 1521–1544, 2007.

15. Arimoto, S. and M. Sekimoto, "Human-like movements of robotic arms with redundant DOFs: virtual spring-damper hypothesis to tackle the Bernstein problem", *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 1860–1866, IEEE, 2006.

16. Matsui, T., M. Honda and N. Nakazawa, "A new optimal control model for reproducing human arm's two-point reaching movements: A modified minimum torque

change model", *2006 IEEE International Conference on Robotics and Biomimetics*, pp. 1541–1546, IEEE, 2006.

17. Zhao, J., B. Xie and C. Song, "Generating human-like movements for robotic arms", *Mechanism and Machine Theory*, Vol. 81, pp. 107–128, 2014.

18. Bhattacharjee, T., Y. Oh and S.-R. Oh, "Validation of a Control Algorithm for Human-like Reaching Motion using 7-DOF Arm and 19-DOF Hand-Arm Systems", *arXiv preprint arXiv:1311.0959*, 2013.

19. Brandão, M., L. Jamone, P. Kryczka, N. Endo, K. Hashimoto and A. Takanishi, "Reaching for the unreachable: integration of locomotion and whole-body movements for extended visually guided reaching", *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 28–33, IEEE, 2013.

20. Mukovskiy, A., C. Vassallo, M. Naveau, O. Stasse, P. Souères and M. A. Giese, "Adaptive synthesis of dynamically feasible full-body movements for the humanoid robot HRP-2 by flexible combination of learned dynamic movement primitives", *Robotics and Autonomous Systems*, Vol. 91, pp. 270–283, 2017.

21. Bertomeu-Motos, A., L. Lledó, J. Díez, J. Catalan, S. Ezquerro, F. Badesa and N. Garcia-Aracil, "Estimation of human arm joints using two wireless sensors in robotic rehabilitation tasks", *Sensors*, Vol. 15, No. 12, pp. 30571–30583, 2015.

22. Occipital, I., *Openni*, `https://structure.io/openni`, 2013, accessed in October 2019.

23. Berekeci, E., *RGB-D Based Object Recognition*, Tech. rep., Boğaziçi University, Turkey, 2019, submitted for senior design project.

24. Adamovich, S. V., P. S. Archambault, M. Ghafouri, M. F. Levin, H. Poizner and A. G. Feldman, "Hand trajectory invariance in reaching movements involving the trunk", *Experimental Brain Research*, Vol. 138, No. 3, pp. 288–303, 2001.

25. Brockett, R. W. *et al.*, "Asymptotic stability and feedback stabilization", *Differential geometric control theory*, Vol. 27, No. 1, pp. 181–191, 1983.

26. Park, J. J. and B. Kuipers, "A smooth control law for graceful motion of differential wheeled mobile robots in 2d environment", *2011 IEEE International Conference on Robotics and Automation*, pp. 4896–4902, IEEE, 2011.

27. Latash, M. L., *Synergy*, Oxford University Press, 2008.

28. Todorov, E., "Optimality principles in sensorimotor control", *Nature neuroscience*, Vol. 7, No. 9, p. 907, 2004.

29. Bartneck, C. and J. Forlizzi, "A design-centred framework for social human-robot interaction", *RO-MAN 2004. 13th IEEE International Workshop on Robot and Human Interactive Communication (IEEE Catalog No. 04TH8759)*, pp. 591–594, IEEE, 2004.

30. Oestreicher, L. and K. S. Eklundh, "User expectations on human-robot cooperation", *ROMAN 2006-The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 91–96, IEEE, 2006.

31. Dautenhahn, K., "Methodology & themes of human-robot interaction: A growing research field", *International Journal of Advanced Robotic Systems*, Vol. 4, No. 1, p. 15, 2007.

32. Kwon, M., M. F. Jung and R. A. Knepper, "Human expectations of social robots", *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 463–464, IEEE, 2016.

33. Gopura, R., K. Kiguchi and E. Horikawa, "A study on human upper-limb muscles activities during daily upper-limb motions", *International Journal of Bioelectromagnetism*, Vol. 12, No. 2, pp. 54–61, 2010.

34. Stewart, J. C., J. Gordon and C. J. Winstein, “Planning and adjustments for the control of reach extent in a virtual environment”, *Journal of neuroengineering and rehabilitation*, Vol. 10, No. 1, p. 27, 2013.

35. Rossi, E., A. Mitnitski and A. G. Feldman, “Sequential control signals determine arm and trunk contributions to hand transport during reaching in humans”, *The journal of physiology*, Vol. 538, No. 2, pp. 659–671, 2002.

36. Bradski, G., “The OpenCV Library”, *Dr. Dobb’s Journal of Software Tools*, 2000.

37. Quigley, M., K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler and A. Y. Ng, “ROS: an open-source Robot Operating System”, *ICRA Workshop on Open Source Software*, 2009.

38. Qt Company, *Qt Creator*, `https://www.qt.io`, 2012, accessed in October 2019.

APPENDIX A: KINEMATIC AND DYNAMIC ANALYSIS OF ROBOT ARM

A.1. Kinematics of PRR Arm

- Denavit-Hartenberg parameters of PRR robot arm shown in Fig. 3.4 are shown in Table A.1, where p_i^* is the joint variable.

Table A.1: Link parameters for 3-link serial PRR manipulator

Link	a_i	α_i	d_i	θ_i
1	0	$\pi/2$	p_3^*	0
2	a_2	0	0	p_4^*
3	a_3	0	0	p_5^*

- The corresponding matrices A_i that define homogeneous transformation for each joint are defined as below.

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.1}$$

$$A_2 = \begin{bmatrix} cos(q_4) & -sin(q_4) & 0 & a_2 cos(q_4) \\ sin(q_4) & cos(q_4) & 0 & a_2 sin(q_4) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.2}$$

$$A_3 = \begin{bmatrix} cos(q_5) & -sin(q_5) & 0 & a_3cos(q_5) \\ sin(q_5) & cos(q_5) & 0 & a_3sin(q_5) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.3}$$

A.2. Dynamics of PRR Arm

The dynamics of a robot arm involves the relationship between the actuator torques acting on the joints and the motion. To analyze the dynamics of the manipulator, it can be used the Lagrangian form, which relies on the kinetic and potential energy of the related system.

$$M(q^A)\ddot{q}^A + C(\dot{q}^A, q^A)\dot{q}^A + G(q^A) = \tau \tag{A.4}$$

where $q^A \in \mathbb{R}^3$ shows the joint variables of robot arm; $M(q^A) \in \mathbb{R}^{3x3}$ is the symmetric, bounded, positive definite generalized inertia matrix, defined in arm frame; $C(\dot{q}^A, q^A) \in \mathbb{R}^{3x3}$ denotes the Coriolis and Centrifugal forces matrix obtained from $M(q^A)$; $G(q^A) \in \mathbb{R}^{3x1}$ is the gravitational force, and $\tau \in \mathbb{R}^{3x1}$ is the vector of non-conservative generalized forces including external and friction forces and torques. The term $M(q^A)\ddot{q}^A + C(\dot{q}^A, q^A)$ represents kinetic energy of the manipulator, and the potential energy is described in the gravity term $G(q^A)$.

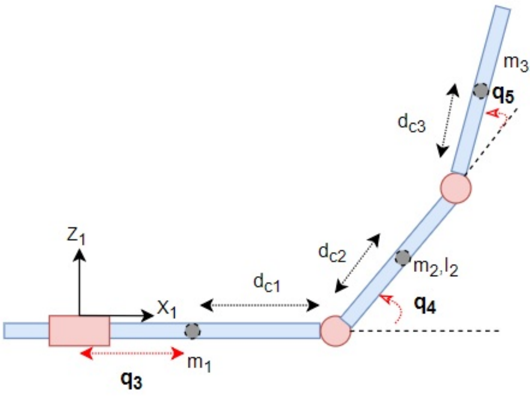


Figure A.1: The definition of generalized coordinates and dynamic parameters for the planar PRR robot (the link lengths a_2 and a_3)

We define the reference frame axes and the generalized coordinates $q^A \in \mathbb{R}^3$ for the planar PRR robot arm as in Figure A.1. Letting $\upsilon_{c_i} \in \mathbb{R}^3$ be translational velocity of the center of the mass for the link i and $\omega_i \in \mathbb{R}^3$ be angular velocity for each link, the kinetic energy of the robot manipulator,

$$K(q^A, \dot{q}^A) = \sum_{i=1}^{3} K_i = \sum_{i=1}^{3} \frac{1}{2}[m_i||\upsilon_{c_i}||^2 + \omega_i^T I_i \omega_i] = \frac{1}{2}\dot{q}^{A^T} M(q^A)\dot{q}^A \tag{A.5}$$

where $q^A, \dot{q}^A, \ddot{q}^A \in \mathbb{R}^{3x1}$ denote vector of joint position, joint velocity and acceleration, respectively. $I_i \in \mathbb{R}^{3x3}$ represents inertia tensor of link i. m_i is the mass of the link. d_{c_i} is the center of mass of link i.

- We define the reference frame axes and the generalized coordinates $q^A \in \mathbb{R}^3$ for the planar PRR robot arm as in Fig. A.1. Letting $\upsilon_{c_i} \in \mathbb{R}^3$ be translational velocity of the center of the mass for the link i and $\omega_i \in \mathbb{R}^3$ be angular velocity for each link, the kinetic energy of the robot manipulator,

$$K(q^A, \dot{q}^A) = \sum_{i=1}^{3} K_i = \sum_{i=1}^{3} \frac{1}{2}[m_i||\upsilon_{c_i}||^2 + \omega_i^T I_i \omega_i] = \frac{1}{2}\dot{q}^{A^T} M(q^A)\dot{q}^A \tag{A.6}$$

where $q^A, \dot{q}^A, \ddot{q}^A \in \mathbb{R}^{3x1}$ denote vector of joint position, joint velocity and acceleration, respectively. $I_i \in \mathbb{R}^{3x3}$ represents inertia tensor of link i. m_i is the mass of the link. d_{c_i} is the center of mass of link i. p_{c_2} and p_{c_3} are positions of links' centers for revolute joints.

$$\begin{aligned} p_{c_2} &= \begin{pmatrix} q_3 + d_{c_1} + d_{c_2}cos(q_4) \\ d_{c_2}sin(q_4) \end{pmatrix} \\ p_{c_3} &= \begin{pmatrix} q_3 + d_{c_1} + a_2cos(q_4) + d_{c_3}cos(q_4 + q_5) \\ a_2sin(q_4) + d_{c_3}sin(q_4 + q_5) \end{pmatrix} \end{aligned} \tag{A.7}$$

- The translational velocities related to q^A from Eq. A.7,

$$\begin{aligned} \upsilon_{c_2} &= \begin{pmatrix} \dot{q}_3 - d_{c_2} sin(q_4)\dot{q}_4 \\ d_{c_2} cos(q_4)\dot{q}_4 \end{pmatrix} \\ \upsilon_{c_3} &= \begin{pmatrix} \dot{q}_3 - a_2 sin(q_4)\dot{q}_4 - d_{c_3}(\dot{q}_4 + \dot{q}_5) sin(q_4 + q_5) \\ a_2 cos(q_4)\dot{q}_4 + d_{c_3}(\dot{q}_4 + \dot{q}_5) cos(q_4 + q_5) \end{pmatrix} \end{aligned} \tag{A.8}$$

The kinetic energy of for the link 1 related to prismatic joint is given by,

$$K_1 = \frac{1}{2} m_1 \dot{q}_3^2 \tag{A.9}$$

The kinetic energy of for the link 2 related to q_4 is given by,

$$K_2 = \frac{1}{2} m_2 \big((\dot{q}_3 - d_{c_2} sin(q_4)\dot{q}_4)^2 + (d_{c_2} cos(q_4)\dot{q}_4)^2 \big) + \frac{1}{2} \dot{q}_4^T I_2 \dot{q}_4 \tag{A.10}$$

The kinetic energy of for the link 3 related to q_5 is given by,

$$\begin{aligned} K_3 = \frac{1}{2} m_3 \big((\dot{q}_3 - a_2 sin(q_4)\dot{q}_4 - d_{c_3}(\dot{q}_4 + \dot{q}_5) sin(q_4 + q_5))^2 \\ + (a_2 cos(q_4)\dot{q}_4 + d_{c_3}(\dot{q}_4 + \dot{q}_5) cos(q_4 + q_5))^2 \big) \\ + \frac{1}{2} (\dot{q}_4 + \dot{q}_5)^T I_3 (\dot{q}_4 + \dot{q}_5) \end{aligned} \tag{A.11}$$

- The total kinetic energy for the PRR robot arm is as follows:

$$K = K_1 + K_2 + K_3 \tag{A.12}$$

Since the robot arm is mounted on a robot platform, which has a fixed distance d_f from the ground, the total potential energy for the manipulator is,

$$V(q^A) = m_1 g d_f + m_2 g (d_f + d_{c_2} sin(q_4)) + m_3 g (d_f + a_2 sin(q_4) + d_{c_3} sin(q_4 + q_5)) \tag{A.13}$$

- In order to obtain equations of the motion, it is defined Lagrangian, L, as the difference between the kinetic and potential energy of the system.

$$L(q^A, \dot{q}^A) = K(q^A, \dot{q}^A) - V(q^A) \tag{A.14}$$

The equations of motion with arm coordinated $q^A \in \mathbb{R}^3$ and Lagrangian L are given by,

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i^A} - \frac{\partial L}{\partial q_i^A} = \Upsilon_i \qquad i = 3, 4, 5 \tag{A.15}$$

where Υ_i denotes the external force acting on the link i generalized arm frame. The first part of the Eq. A.15 represents the time derivative of the momentum. Substituting L into Lagrange's equations satisfy overall dynamics, where the variables m_{ij} are inertia matrix M,

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_3 \\ \ddot{q}_4 \\ \ddot{q}_5 \end{bmatrix} + \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \end{bmatrix} + \begin{bmatrix} g_3 \\ g_4 \\ g_5 \end{bmatrix} = \begin{bmatrix} \Upsilon_1 \\ \Upsilon_2 \\ \Upsilon_3 \end{bmatrix} \tag{A.16}$$

$$\begin{aligned} m_{11} &= m_1 + m_2 + m_3 \\ m_{12} &= -(m_2 d_{c_2} + m_3 a_2) sin(q_4) - m_3 d_{c_3} sin(q_4 + q_5) \\ m_{13} &= -m_3 d_{c_3} sin(q_4 + q_5) \\ m_{21} &= -(m_2 d_{c_2} + m_3 a_2) sin(q_4) - m_3 d_{c_3} sin(q_4 + q_5) \\ m_{22} &= I_2 + m_2 d_{c_2}^2 + I_3 + m_3 d_{c_3}^2 + m_3 a_2^2 + 2 a_2 m_3 d_{c_3} cos(q_5) \\ m_{23} &= I_3 + m_3 d_{c_3}^2 + a_2 m_3 d_{c_3} cos(q_5) \\ m_{31} &= -m_3 d_{c_3} sin(q_4 + q_5) \\ m_{32} &= I_3 + m_3 d_{c_3}^2 + a_2 m_3 d_{c_3} cos(q_5) \\ m_{33} &= I_3 + m_3 d_{c_3}^2 \end{aligned} \tag{A.17}$$

- For the potential energy,

$$
\begin{aligned}
g_3 &= 0 \\
g_4 &= gsin(q_4)m_2d_{c_2} \\
g_5 &= g(d_{c_3}sin(q_5) + a_2sin(q_4))m_3
\end{aligned}
\tag{A.18}
$$

- The product $C(q^A, \dot{q}^A)\dot{q}^A$ is a 3x1 vector whose elements are quadratic functions of joint velocities $\dot{q}^A$. The elements c_{ij} are obtained in Eq. A.19.

$$
\begin{aligned}
c_{11} &= 0 \\
c_{12} &= \dot{q}_4\big(- d_{c_2}m_2 - a_2cos(q_4)m_3 - d_{c_3}m_3cos(q_4 + q_5)\big) + \dot{q}_5\big(- 2d_{c_3}m_3cos(q_4 + q_5)\big) \\
c_{13} &= \dot{q}_4\big(- 2d_{c_3}m_3cos(q_4 + q_5)\big) + \dot{q}_5\big(- d_{c_3}m_3cos(q_4 + q_5)\big) \\
c_{21} &= \dot{q}_4\big(- d_{c_2}m_2cos(q_4) - m_3cos(q_4) - d_{c_3}cos(q_4 + q_5)\big) + \dot{q}_5\big(- d_{c_3}m_3cos(q_4 + q_5)\big) \\
c_{22} &= \dot{q}_3\big(- d_{c_2}m_2cos(q_4) - m_3cos(q_4) - d_{c_3}m_3cos(q_4 + q_5)\big) \\
c_{23} &= \dot{q}_3\big(- d_{c_3}cos(q_4 + q_5)\big) + \dot{q}_4\big(- 2d_{c_3}a_2sin(q_4)\big) + \dot{q}_5\big(- d_{c_3}a_2sin(q_4)\big) \\
c_{31} &= \dot{q}_4\big(- d_{c_3}m_3cos(q_4 + q_5)\big) + \dot{q}_5\big(- d_{c_3}m_3cos(q_4 + q_5)\big) \\
c_{32} &= \dot{q}_3\big(- d_{c_3}m_3cos(q_4 + q_5)\big) + \dot{q}_5\big(- a_2m_3sin(q_5)\big) \\
c_{33} &= \dot{q}_3\big(- d_{c_3}m_3cos(q_4 + q_5)\big) + \dot{q}_4\big(- a_2m_3sin(q_5)\big)
\end{aligned}
\tag{A.19}
$$

APPENDIX B: HARDWARE & SOFTWARE

The robot is described with respect to its hardware and software components.

B.1. Robot System

ISL-Turtlebot is a mobile robot that is designed for indoor applications, as shown in Figure B.1. The robot can move around and observe the environment. The system also has a 3-DoFs manipulator attached to the robot platform. The arm has a linear joint and two rotational joints. It also has a gripper. With this robotic arm, the robot should be guided well enough to hold an object with its gripper. The expanse of the gripper is approximately 15 centimeters. The robot has an RGB-D sensor camera. It generates an RGB image and a depth image. Turtlebot is used to supply power to the camera and motors. It also carries the robot arm.

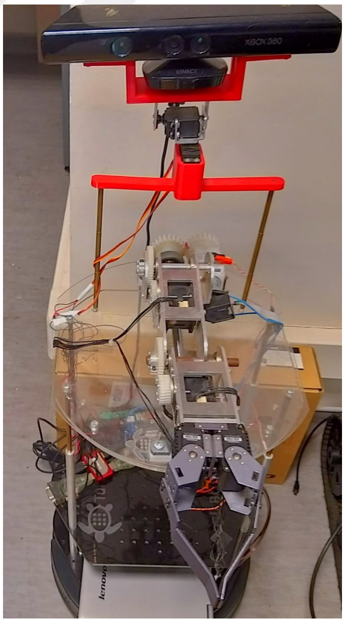


Figure B.1: Turtlebot with Kinect and manipulator

The sensing capabilities of the robot are as follows:

- Kinect (laser range scanner + camera),
- Cliff sensors,

- Wheel drop sensors,
- Bump sensors

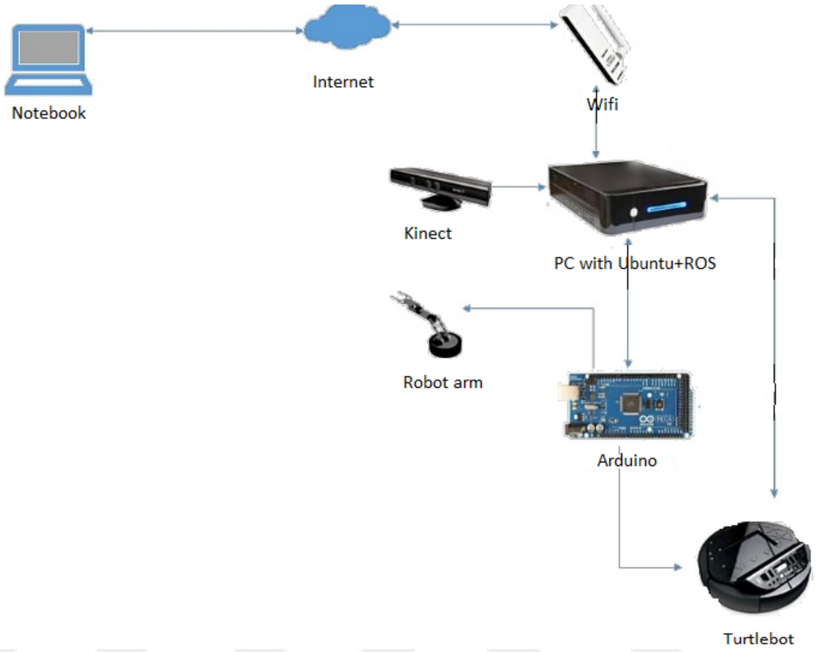


Figure B.2: The hardware specification of the whole system

The hardware specification of the whole system is shown in Figure B.2. Whereas the robot base and computing unit have their own integrated batteries, Kinect sensor requires 12V power sources. After powering on the robot base, the netbook, and the sensors, the robot is ready for the operation. The robot arm has five motors; two Dynamixel AX-18A for the end-effector, two Dynamixel RX-64 motors for revolute type joints, and one for the prismatic joint. The overall system consists of power supplies, slider card, USB ports, RX-64 and AX-18A motors, mechanical parts. The revolute, gripper, and slider motors require different voltage supply. The revolute motors work with 14.7V, gripper motors work with 11.2V, and slider motor works with 12V. If less voltage is applied, motors do not operate with full power.

B.2. Programming Languages

In this thesis, the core systems are all written in C++, since the mobile robots generally do not have adequate processing powers. Additionally, it is one of the fastest programming languages. MATLAB is undoubtedly one of the best fast-prototyping programming languages. For that reason, it is used as a simulation environment. The kinematic model of the robot and control system are defined and simulated.

B.3. Libraries

OpenCV [36] is the best library in computer and machine vision areas. It covers all the core algorithms efficiently. We use its C++ API to perform image processing tasks. It also makes it easier to work on the matrix in C++.

Nearly all of the works are written under ROS [37] due to its publisher-subscriber system, modularity, compilation tools, community support, hardware support. Besides, it supports different languages, including C++ and Python. Thus, one can run several programs written in various languages at the same time.

Qt Creator [38] presents a cross-platform, complete integrated development environment (IDE) for application developers to generate applications. It is accessible for Linux and Windows operating systems. ROS has been very helpful to Qt by providing a package, catkin_create_qt_pkg, to help building the Qt environment in ROS.

B.4. Software

The software covers the codes of object sensing and manipulator movement. The codes are under the ISL_ws in the ISL-computer number 7.

B.5. Object Sensing Software

It is required to run all functions to execute object sensing algorithm. The node of vision1 computes the depth information between the robot and the object. The node of pseudo finds the object in the environment by using the color-based approach.

(i) roscore (to activate ROS)
(ii) roslaunch freenect_launch freenect.launch (to activate Kinect)

(iii) rqt_image_view (to see the topic)

(iv) rosrun beginner_tutorials pseudo

(v) rosrun beginner_tutorials vision1

B.6. Manipulator Software

It is required to run all functions to execute the movement of manipulator.

(i) roscore (to activate ROS)

(ii) rosrun dynamixel_driver dynamixel_driver_node (to activate motors)

(iii) rosrun dynamixel_arduino dynamixel_arduino_node (to run motors)

(iv) rosrun dynamixel_all dynamixel_all_node (to run robot joints)