SSI-DT79115501

Doctoral Thesis

Imitation Learning Framework using Principal Component Analysis for Humanoid Robot Motion Generation

ヒューマノイドロボット動作生成のための主成分分析を用いた模倣学習フレームワーク

Garam Park

February, 2015

Division of System Science and Informatics
Graduate School of Information Science and Technology
Hokkaido University

Doctoral Thesis
submitted to Graduate School of Information Science and Technology,
Hokkaido University
in partial fulfillment of the requirements for the degree of
Doctor of Engineering.

Garam Park

Thesis Committee: Atsushi Konno
Satoshi Ogasawara
Hajime Igarashi

Imitation Learning Framework using Principal Component Analysis for Humanoid Robot Motion Generation*

Garam Park

Abstract

This dissertation investigates the efficient imitation learning framework based on principal component analysis to generate desired and abundant motions using few human demonstrations in the workspace. This framework is inspired from human motor learning in upper body. In real life, a robot is expected to autonomously act with varied and abundant motions. Imitation learning might be an efficient approach to enable a robot to generate natural and abundant motions.

The framework comprises off-line and on-line processes. In the off-line process, human demonstrations are used to develop a motion database. The database covers the workspace and includes robot properties. The evolved database has a clustered structure for efficiency. In the on-line process, a robot can generate desired motions using a real-time motion reconstruction method based on PCA in real time. However, during the motion reconstruction by few, slight movement, or noisy, it occurs the over-response problem. To prevent that, we uses Regularization method, and curve fitting method to reduce the tolerance. The performance of this method is verified through two case studies. The proposed framework is applied to the generation of reaching motions to an object on a table and a shelf.

Keywords: Machine Learning, Imitation Learning, Principal Component Analysis, Evolutionary Process

*Doctoral Thesis, Division of System Science and Informatics, Graduate School of Information Science and Technology, Hokkaido University, SSI-DT79115501, February 16, 2015.

Contents

List of Figures

Chapter 1. Introduction

1.1 General Overview

This thesis presents the imitation learning framework that the humanoid robot can learn human arm motions to generate various arm motions in real world. The proposed framework is inspired by biological studies of motor learning of human and robotics studies and focuses on how human learns new skills or movements and what is an important factor on robot's arm motion generation.

Generally, for generating robot arm motion, two approaches have been studied thus far. One is human-aided motion generation that uses techniques such as pre-programming or interpolation of waypoints taught via a teach pendant. However, these techniques are not realistic solutions for producing varied and abundant motions, because they are very time consuming and not possible to generate desired motions by itself in real-life. The second is robot learning from human demonstrations.

Programming by demonstrations of a task, which is also referred to as *imitation learning*, is a noteworthy and powerful solution to robot learning [2, 3]. In imitation learning, three key points need to be considered: (1) *what to imitate*, (2) *how to imitate*, and (3) *when to imitate* [4]. In [5], I proposed an imitation learning framework for teaching a robot to imitate human motions that are repetitive and similar [5]. The framework was designed such that movement primitives maintain human as well as robot properties. For example, a trajectory that is produced based on human demonstrations is not always optimal for robots. It is clear that the minimum energy consumption motion of a human is not always the same as that of a robot. Therefore, in the framework, while movement primitives were constructed basically from human demonstration data, a cost function for robots, including factors such as minimizing energy consumption and jerk, was considered. The framework generates the desired motions by using a combination method based on principal component analysis (PCA). Using this framework, the motion of catching a ball was generated in real time for a humanoid robot [6]. However, the imitation learning framework proposed in [5, 6] requires a large number of human demonstrations and sometimes produces a trajectory that deviates radically from that demonstrated by the human because of a rank deficiency in a principal component matrix. In order to reduce the number of human demonstrations, a clustered database structure was proposed [7], in which a workspace was divided into clusters and principal components were defined for a cluster. Although the revised framework [7] could reduce the number of the required demonstrations, it still suffered from the over-response problem and human demonstrations had to be previously specified, such as *reaching to points A, B, and C*. If less specified demonstrations, such as *reaching to any points*, are available as teaching data, the teaching phase is significantly easier.

PCA was applied for imitation learning in a few studies. Fod et al. used PCA to represent human movement and reconstruct the movement [8, 9]. Lim et al. took a similar approach, but a cost function was introduced to optimize the robot's movement [10]. However, they applied PCA to the joint trajectory to avoid the redundancy problem [8, 10]. Therefore, it is possible that even if the human demonstration is closely reproduced in joint space, the resultant endpoint trajectory will significantly deviate from that in the demonstration. Furthermore, since movement reconstruction in joint space was addressed in [8, 10], the authors' approach could not deal with arbitrarily given start and target points that differed from the demonstrated start and target points. In this study, PCA is applied to represent the endpoint trajectories in human demonstrations and to reconstruct them in Cartesian space. In order to solve the redundancy problem, elbow elevation angles [11, 12] are considered. Since the movement is reconstructed in Cartesian space, a movement is easily and adequately generated for arbitrarily given start and target points that differ from the demonstrated points. This is the main advantage of our method over the those proposed in previous studies such as [8, 10, 9].

The design of the proposed imitation learning framework refers to the human motion learning process. When one human learns new motions or postures from another through repetitive training, he/she does not try to strictly imitate the motion of the demonstrator, because differences in kinematics parameters, such as arm length, make this impossible. Instead, he/she will try to optimize the similarity of the motion to the motion of the demonstrator. In [journal], the improved imitation learning framework was proposed to complements the framework proposed in previous works [5, 6, 7] and overcomes their drawbacks.

The proposed improved imitation learning framework generates an efficient and abundant motions by learning and optimizing from human demonstrations. This framework dose not require strictly specified demonstrations, such as reaching to points A, B, and C, but accepts moderately specified demonstrations, such as reaching to any points. This tolerance to less specified demonstration contributes to reducing the number of demonstrations. Furthermore, in order to solve the over-response problem (radical deviation from human demonstrations), L_2 regularization

1.2 Research Background

1.2.1 Human Motor Learning and Motion Generation Researches

Humanoid robot have increasingly resembled human beings in such control ability as walking control, running control, etc., as well as in appearance. However, the robot cannot move like human. In order to imitate human movement, firstly, we need to understand how human assimilates new skills and movement. The Central Nervous System (CNS) conducts motor learning, motion generation, etc. Human behavior is generally separated momently actions by a sensory organ and actions by CNS. This thesis focuses on the the actions by CNS. In spite of the great differences between a vertebrate system and robotic system such as DOFs, joint limit, and arm lengths, the fields of biology and neuroscience have tried to derive benefits from the theories and procedures that may help the control of an artificial multi-joint system: motor learning [13, 14] and motor primitives [15] have been enthusiastically studied in vertebrate animals with a view to applying the knowledge thus gained in the control of an artificial multi-joint robotic system. Further, it is possible that the brain solves inverse kinematics and dynamics for voluntary movement [16]. Models of imitation based on instrumental learning, associative learning, and more complex cognitive processes have been proposed [17, 18]. Hollerbach and Flash [19] proved experimentally that brain may carry out inverse kinematics and dynamic computations when a human moves in a purposeful. In addition, inverse dynamics can be represented as the operation of a memory that associates a vector of joint torques, angles, and angular velocities and accelerations with a computational device like a look-up table [20]. Mussa-Ivalde et al. stated that the central nervous system can create, update, and exploit internal representations of limb dynamics like a memory and motor pattern generator to deal with the computation complexity of inverse dynamics [15, 21]. They also regarded the vectorial combination of the force field exerted by limbs as the representation. The representation was applied to a robotic system in the form of posture primitives [22].

1.2.2 Robot Programming by Demonstrations (PbD) Researches

Robot Programming by demonstration has been studied for several decades. as called imitation learning. In these thesis, the problem of how to get human movements is not dealt with, I focus on how the human movements should be converted to a robot and the information should be applied to imitation learning.

Representations of human movements are roughly classified into symbolic and trajectory level encoding [3]. Symbolic level encodings generally segment and encode tasks according to sequences of predefined actions by using classical machine learning techniques [23, 24, 25] , such as the Hidden Markov Model (HMM) [3, 26, 27, 28]. Alissandrakis et al. took a symbolic approach to encoding human motions as sets of pre-defined postures, positions, or configurations and considered different levels of granularity for the symbolic representation of the motion [29]. Symbolic approaches have a great advantage in representing high-level skills, such as sequences of several motions, by symbolic cues (for instance, ‘walk’→‘run’ or 'walk'→'run'→'kick'). However symbolic approaches need prior knowledge to define the important cues that resolve and determine a desired motion.

In trajectory level representation, human movements are encoded in joint, torque or task space [30, 31, 32]. The trajectory level representation is used mainly to generate cyclic motion or discrete motions [33, 34]. Statistical modeling is a noteworthy approach for encoding skills and movements that deals with noisy demonstration data. Ijspeert et al. presented a control policy that was defined in form of a differential equation to represent robot joint angles and end-effector's positions [35]. HMM is a popular approach for encoding, recognizing and retrieving several types of movement [36]. Lee et al. used the learning of gesture through a mimesis model based on HMM to encode trajectories and retrieve desired motions using a stochastic process [37].

Dynamical systems (DSs) have been employed as a powerful method for modeling robot motions [38, 39, 40] that uses statistical estimation based on locally weighted regression (LWR) [41, 42, 43, 44], Gaussian mixture regression (GMR) [40], etc.

Dynamic movement primitives (DMP) is a method by which a nonlinear DS can be estimated while ensuring global stability at an attractor point [45, 46]. DMP basically requires a reference trajectory generated by using GMR, weighted least-square (WLS), etc. Khansari-Zadeh and Billard presented the stable estimator of dynamical systems (SEDS) to learn arbitrary discrete motions by modeling them as a nonlinear autonomous DS [47]. DS, DMP, and SEDS are designed to handle the temporal or spatial perturbations on end-point trajectories. However, when a target or start point departs from the demonstrated point, these methods cannot guarantee the generated trajectory will be similar to that of the human demonstration. Furthermore, the performance results, such as those for similarity, strongly depend on the parameters.

In most programming by demonstration (PbD) systems proposed thus far only the end-point movements were addressed [48, 49, 50]. However, in general, humanoid robot control systems suffer from redundancy, a problem that must be solved to allow a robot motion to be generated. Therefore, we explore a method in which not only the endpoints of the limbs but also the entire posture are imitated.

1.3 Purpose of the thesis

This thesis is to design the efficient imitation learning framework for humanoid or manipulator robot. In real life, to execute various missions by user, it is hard to generate the desired motions by programmer. Imitation learning is great approach to carry out various motions by user.

In this thesis, to imitate human demonstrations, a robot uses few human demonstrations. Then, it is evolved by evolutionary process to extract the movement primitives database to cover the workspace, and to involve the robot property into that. Also, by using evolved movement primitives database, the robot generate the desired motions in real time. For generating the desired motions in real time, the framework adopts the motion reconstruction based on principal component analysis. Nevertheless, occasionally the motion reconstruction may generate extravagant or inadequate motions, as called the over-response, because of using few movement primitives database, slight movement or noisy movement primitives. Thus, the method to prevent the over-response discusses in the section which describes the motion reconstruction.

1.4 Summary of the thesis

Chapter 2

This chapter is discussed for motion reconstruction based on principal component analysis, which generate the desired motions using movement primitives, and the solution of over-response problem during generating the motion reconstruction is described by L_2 regularization. Also, the case of sudden change in the target position, the motion reconstruction shows that the generated trajectory is modified to track the changed target position.

Chapter 3

This chapter describes the evolution process of the movement primitives database. The framework is divided by two processes: 1) In Off-line process, capturing human demonstrations, transformation of movement primitives database from human to humanoid robot, and evolutionary process are discussed. finally, robot acquires an evolved movement primitives database. 2) In on-line process, by using the evolved movement primitives database, motion reconstruction generates a desired motions in real time.

Chapter 4

The performance of proposed imitation learning framework is shown through two cases study. One is a reaching motions to an object on the table, another is a reaching motion to an object on a shelf. By using few human demonstrations, My method shows the performance of human-like motion generation.

Chapter 5

The dissertation concludes with a general discussion of the results and benefits of this thesis.

1.4.1 Overview of the imitation learning framework

Figure 1.1 is shown the overview of proposed imitation framework. First, this is divided by two process: 1) Off-Line imitation learning Process for evolving and optimizing of movement primitives database from human demonstrations, 2) On-Line motion generation process for human-like and optimal robot motion in real-time.

In the off-line process, movement primitives database, which is transformed human demonstrations to robot specification, is evolved by using evolutionary process based on PCA. Through this process, one involves the extracted movement primitives database to cover the workspace and optimizing that to reflect the robot property. Movement primitives are defined according to task as trajectories in either joint or Cartesian space. Consequently, the robot acquires the evolved movement primitives database.

In the on-line process, the robot can generate the human-like and abundant motions in the workspace by using evolved movement primitives database in real

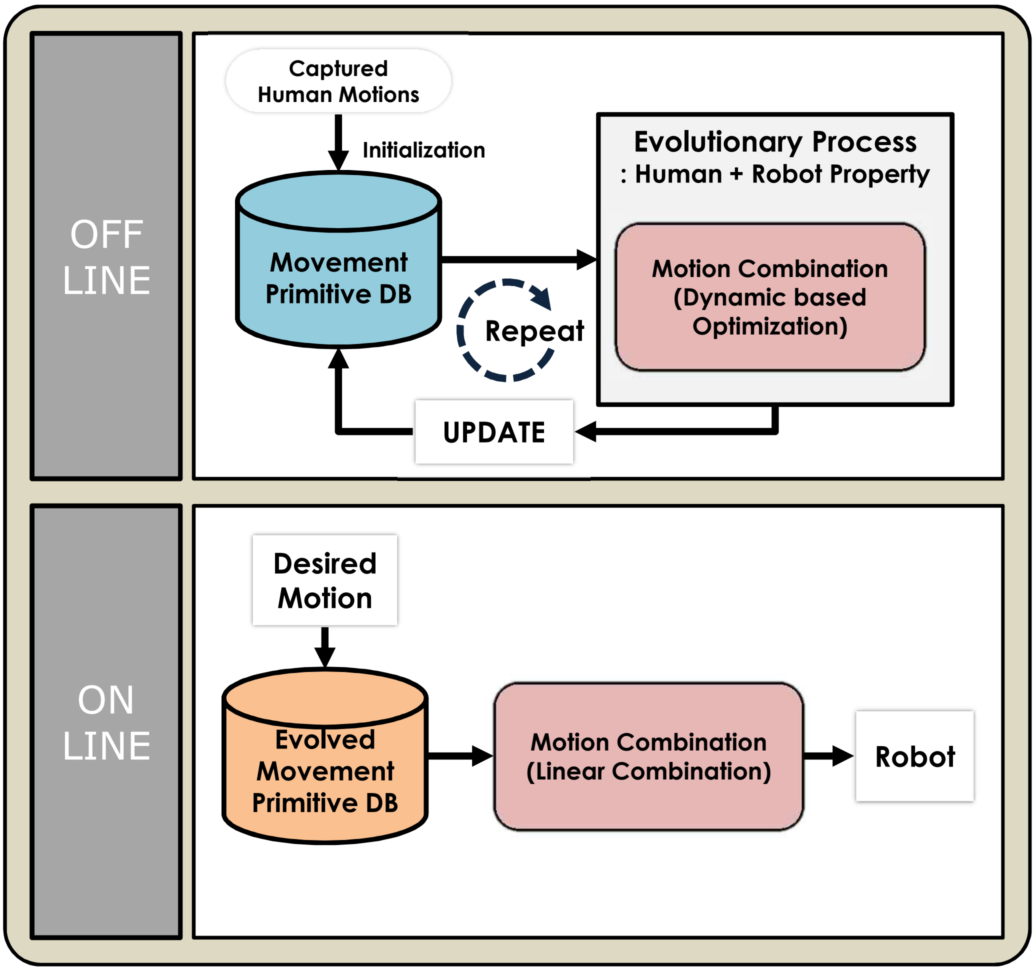


Figure 1.1: Proposed Imitation Learning Framework

time.

Chapter 2. Motion Reconstruction based on PCA

PCA is used in both the off-line and on-line processes to generate a new motion from the movement primitives database. In the on-line process, when the optimized and abundant movement primitives database has been constructed, a robot can use it to generate the desired motions in the workspace in real time. Motions are generated so that they resemble the human demonstrations as closely as possible, include redundant DOFs.

This section describes a motion generation method that uses the movement primitives database. The computational cost of data processing in high-dimensional space exponentially increases according to the dimension. PCA is a solution for avoiding *the curse of dimensionality* in data processing.

Park et al. generated a desired humanoid robot motion using three principal components in joint space [5, 6, 7]:

$$q_j(t) = q_{j,mean}(t) + \sum_{i=1}^{3} w_{j,i} q_{j,pc_i}(t) + w_{j,4} \quad (j = 1, \ldots, N), \tag{2.1}$$

where N is the number of joints and $q_j(t)$, $q_{j,mean}(t)$, and $q_{j,pc_i}(t)$ are the trajectory to be generated, mean values, and principal components of the movement primitives of the jth joint, respectively. $w_{j,i}(i = 1, 2, 3)$ are the scalar weighting coefficients and w_4 denotes a remaining offset. In general, the first four principal components dominate 90% of the data [10]. In [5, 6, 7], the first three principal components were used to uniquely determine the coefficients $w_{j,i}$ $(i = 1 - 4)$ from the four boundary conditions, the positions and velocities at the beginning and end of a motion.

When a motion is defined in Cartesian space, the motion reconstruction equation (Eg. (2.1)) is extended:

$$p_k(t) = p_{k,mean}(t) + \sum_{i=1}^{3} w_{k,i} p_{k,pc_i}(t) + w_{k,4} \quad (k = 1, \ldots, M), \tag{2.2}$$

where $p_k(t)$, $p_{k,mean}(t)$, and $p_{k,pc_i}(t)$ are the trajectory to be generated, mean values, and principal components of the movement primitives of the kth component in Cartesian space, respectively. M is the dimension of the augmented position vector defined in Cartesian space. If a 7-DOF arm is used, the augmented position vector is defined as

$$[p_1 \ldots p_7]^{\mathrm{T}} = [x \;\; y \;\; z \;\; \alpha \;\; \beta \;\; \gamma \;\; q_E]^{\mathrm{T}}, \tag{2.3}$$

where x, y, and z are the positions of the hand or wrist, α, β, and γ are three variables that define the posture, such as Euler angles, and q_E is a variable that defines elements of the elbow position, such as the elbow elevation angle [11, 12]. If the posture of the hand does not need to be considered, the augmented position vector may be defined as

$$[p_1 \ldots p_4]^{\mathrm{T}} = [x \;\; y \;\; z \;\; q_E]^{\mathrm{T}}. \tag{2.4}$$

Boundary conditions are specified by the initial and final positions ($p_{k,0}$, $p_{k,f}$) and velocities ($\dot{p}_{k0}$, $\dot{p}_{kf}$) as

$$p_k(t_0) = p_{k,0}, \;\; p_k(t_f) = p_{k,f}, \;\; \dot{p}_k(t_0) = \dot{p}_{k,0}, \;\; \dot{p}_k(t_f) = \dot{p}_{k,f}, \tag{2.5}$$

where t_0 and t_f are the beginning and final times of the motion, respectively. Substituting Eq. (2.5) into (2.2), the scalar constants are determined as

$$\begin{bmatrix} w_{k,1} \\ w_{k,2} \\ w_{k,3} \\ w_{k,4} \end{bmatrix} = \begin{bmatrix} p_{k,pc_1}(t_0) & p_{k,pc_2}(t_0) & p_{k,pc_3}(t_0) & 1 \\ p_{k,pc_1}(t_f) & p_{k,pc_2}(t_f) & p_{k,pc_3}(t_f) & 1 \\ \dot{p}_{k,pc_1}(t_0) & \dot{p}_{k,pc_2}(t_0) & \dot{p}_{k,pc_3}(t_0) & 0 \\ \dot{p}_{k,pc_1}(t_f) & \dot{p}_{k,pc_2}(t_f) & \dot{p}_{k,pc_3}(t_f) & 0 \end{bmatrix}^{-1} \left(\begin{bmatrix} p_k(t_0) \\ p_k(t_f) \\ \dot{p}_k(t_0) \\ \dot{p}_k(t_f) \end{bmatrix} - \begin{bmatrix} p_{k,mean}(t_0) \\ p_{k,mean}(t_f) \\ \dot{p}_{k,mean}(t_0) \\ \dot{p}_{k,mean}(t_f) \end{bmatrix} \right) \tag{2.6}$$

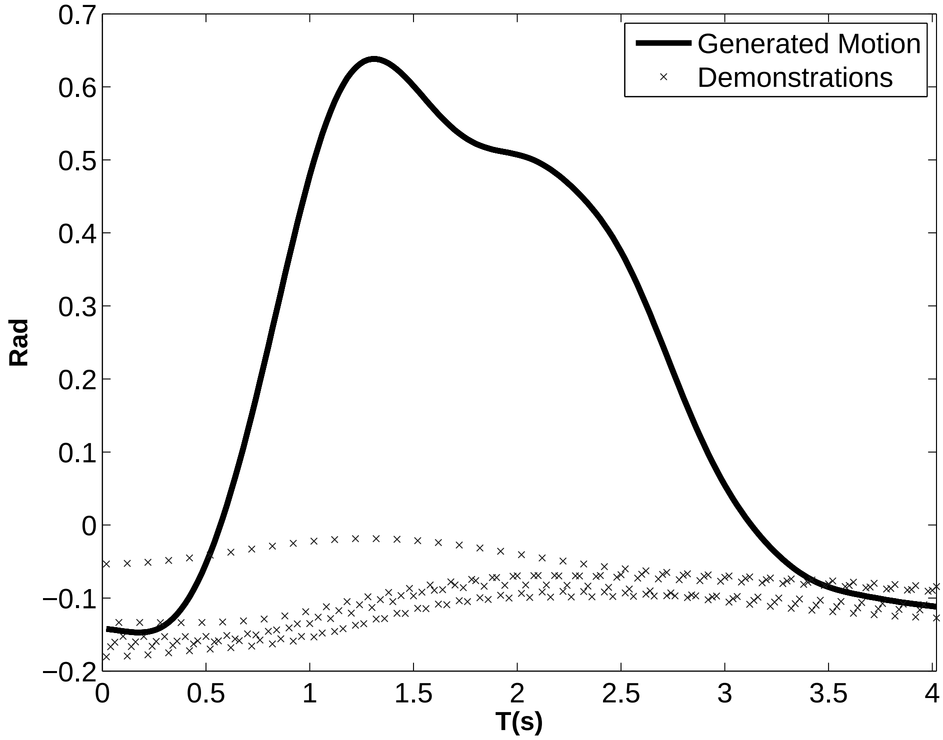


Figure 2.1: Over-response problem. The generated motion significantly deviates from the demonstrated human motion when the principal component matrix ${}^{bc}\mathbf{P}_{k,pc}$ approaches its singularity.

which can be rewritten as

$$\boldsymbol{w}_k = {}^{bc}\mathbf{P}_{k,pc}^{-1}(\boldsymbol{b}_k - \boldsymbol{b}_{k,mean}). \tag{2.7}$$

However, when the number of movement primitives is small, a slight movement or noisy movement primitives may cause extravagant and inadequate motion, as presented in Figure 2.1. The main reason for the inadequate motion is that w_k calculated by Eq. (2.7) is not adequate because of over-response.

In order to prevent over-response, L_2 regularization [51] is used. L_2 regularization reduces over-response by adding a complexity penalty to the loss function. Instead of the coefficients $\boldsymbol{w}_k$ given by Eq. (2.7), the coefficients $\hat{\boldsymbol{w}}_k$ that minimize $\|\boldsymbol{w}_k\|$ and $\|\boldsymbol{b}_k - \boldsymbol{b}_{k,mean} - {}^{bc}\mathbf{P}_{k,pc}\boldsymbol{w}_k\|$ are considered here. The coefficient vector $\hat{\boldsymbol{w}}_k$ is given as

$$\hat{\boldsymbol{w}}_k = \underset{\boldsymbol{w}}{\operatorname{argmin}}\{\|\boldsymbol{b}_k - \boldsymbol{b}_{k,mean} - {}^{bc}\mathbf{P}_{k,pc}\boldsymbol{w}_k\|^2 + \lambda\|\boldsymbol{w}_k\|_2^2\} \tag{2.8}$$

where $\lambda > 0$. The Euclidean norm, $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}}$, is used as the norm of a vector. The derivative of the right hand side of Eq. (2.8) is given as

$$\begin{aligned}\frac{d}{d\boldsymbol{w}_k}&\Big\{(\boldsymbol{b}_k - \boldsymbol{b}_{k,mean} - {}^{bc}\mathbf{P}_{k,pc}\boldsymbol{w}_k)^T(\boldsymbol{b}_k - \boldsymbol{b}_{k,mean} - {}^{bc}\mathbf{P}_{k,pc}\boldsymbol{w}_k) + \lambda\boldsymbol{w}_k^T\boldsymbol{w}_k\Big\}\\ &= -2{}^{bc}\mathbf{P}_{k,pc}^T(\boldsymbol{b}_k - \boldsymbol{b}_{k,mean}) + 2{}^{bc}\mathbf{P}_{k,pc}^T\,{}^{bc}\mathbf{P}_{k,pc}\boldsymbol{w}_k + 2\lambda\boldsymbol{w}_k\end{aligned} \tag{2.9}$$

By letting Eq. (2.9) be zero, $\hat{\boldsymbol{w}}_k$ is given as

$$\hat{\boldsymbol{w}}_k = ({}^{bc}\mathbf{P}_{k,pc}^{\mathrm{T}}\,{}^{bc}\mathbf{P}_{k,pc} + \lambda\mathbf{I})^{-1}\,{}^{bc}\mathbf{P}_{k,pc}^{\mathrm{T}}(\boldsymbol{b}_k - \boldsymbol{b}_{k,mean}), \tag{2.10}$$

where $\mathbf{I}$ is a 4×4 identity matrix. If λ is very large, the result penalizes all the parameters such that they are close to zero. λ should carefully be chosen. It should be noted that $\hat{\boldsymbol{w}}_k$ given by Eq. (2.10) is the approximation of $\boldsymbol{w}_k$. Hence, the boundary condition given by Eq. (2.5) may not be satisfied. The approximate trajectories ($\hat{p}_k(t)$) are given as

$$\hat{p}_k(t) = p_{k,mean}(t) + \sum_{i=1}^{3}\hat{w}_{k,i}p_{k,pc_i}(t) + \hat{w}_{k,4} \;\; (k = 1, \ldots, M). \tag{2.11}$$

In order to strictly satisfy the boundary condition, curve fitting is employed in Figure 2.2:

$$\begin{aligned} p_k(t) &= \hat{p}_k(t) + \alpha + \frac{t - t_0}{t_f - t_0}\beta, \\ \alpha = p_k(t_0) &- \hat{p}_k(t_0), \;\; \beta = p_k(t_f) - \hat{p}_k(t_f) - \alpha. \end{aligned} \tag{2.12}$$

By using Eq. (2.12), a motion is generated in real time based on principal components.

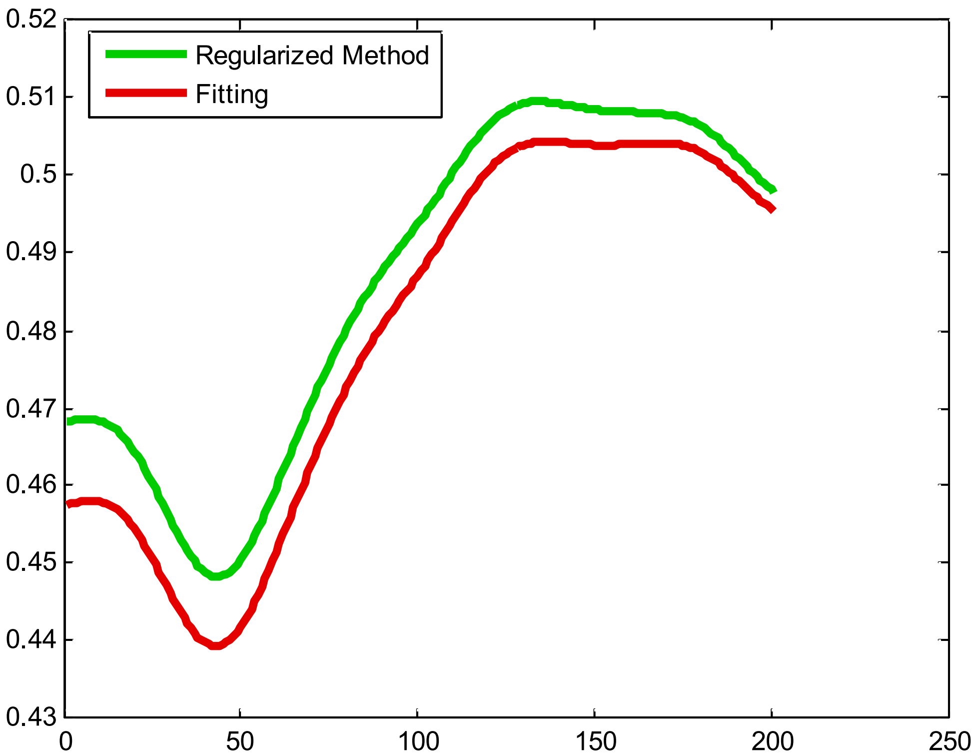
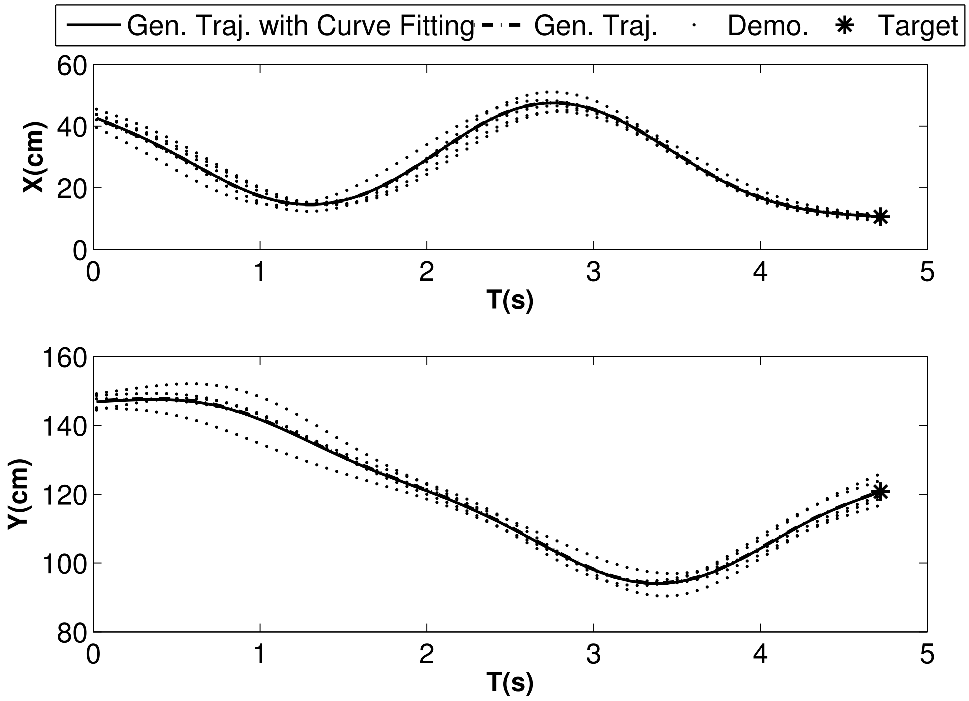
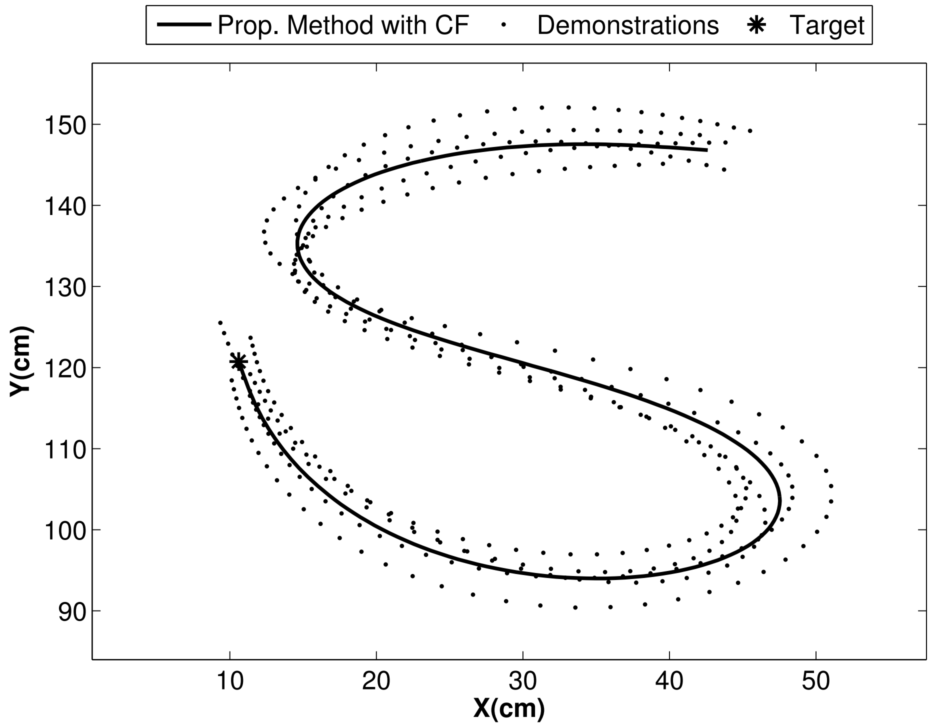


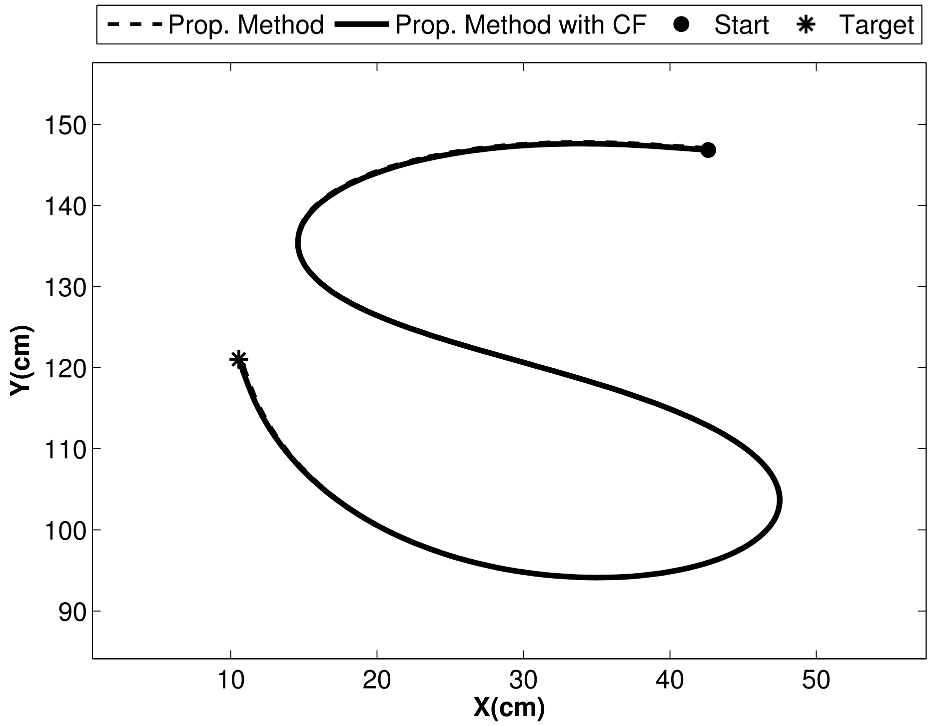
Figure 2.2: Example of curve fitting method


Gen. Traj. with Curve Fitting
Gen. Traj.
Demo.
Target
X(cm)
T(s)
Y(cm)
T(s)

(a)

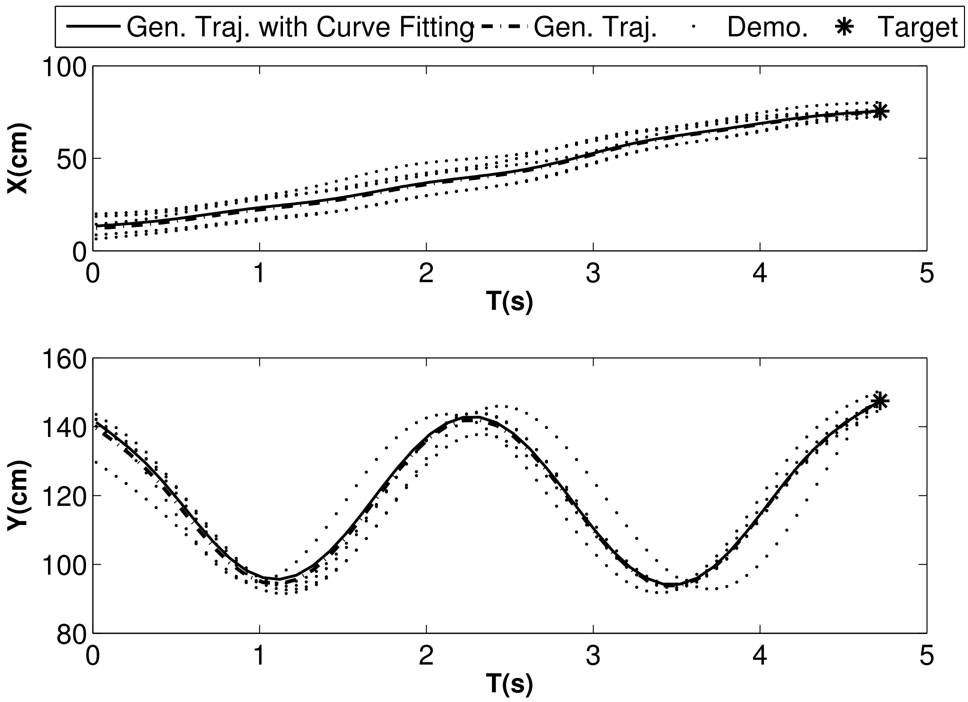

Prop. Method with CF
Demonstrations
Target
Y(cm)
X(cm)

(b)



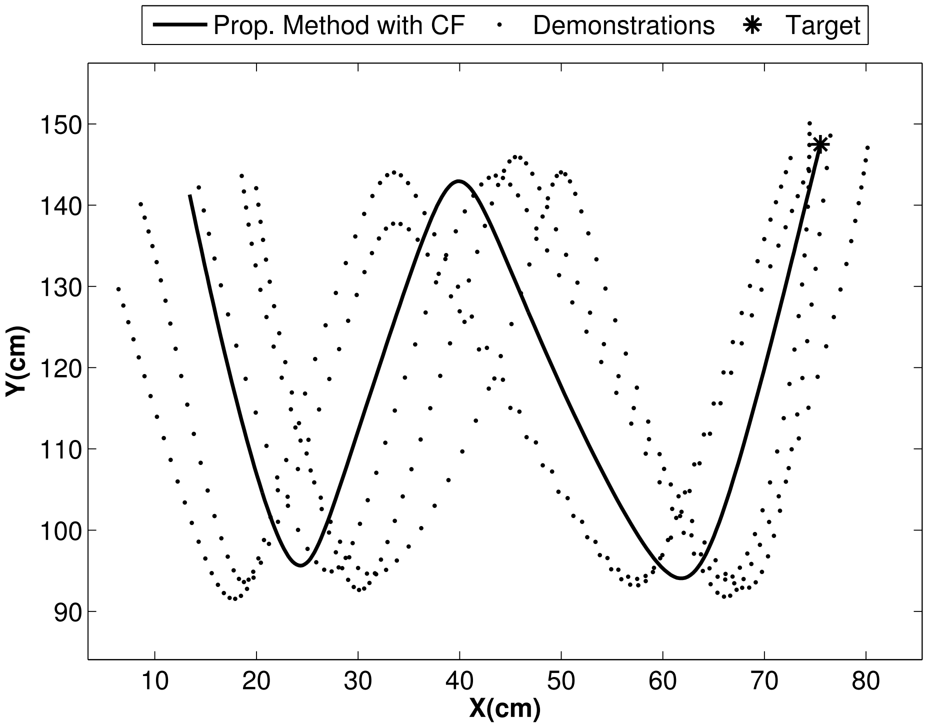
Prop. Method
Prop. Method with CF
Start
Target
150
140
130
120
110
100
90
Y(cm)
10
20
30
40
50
X(cm)

(c)

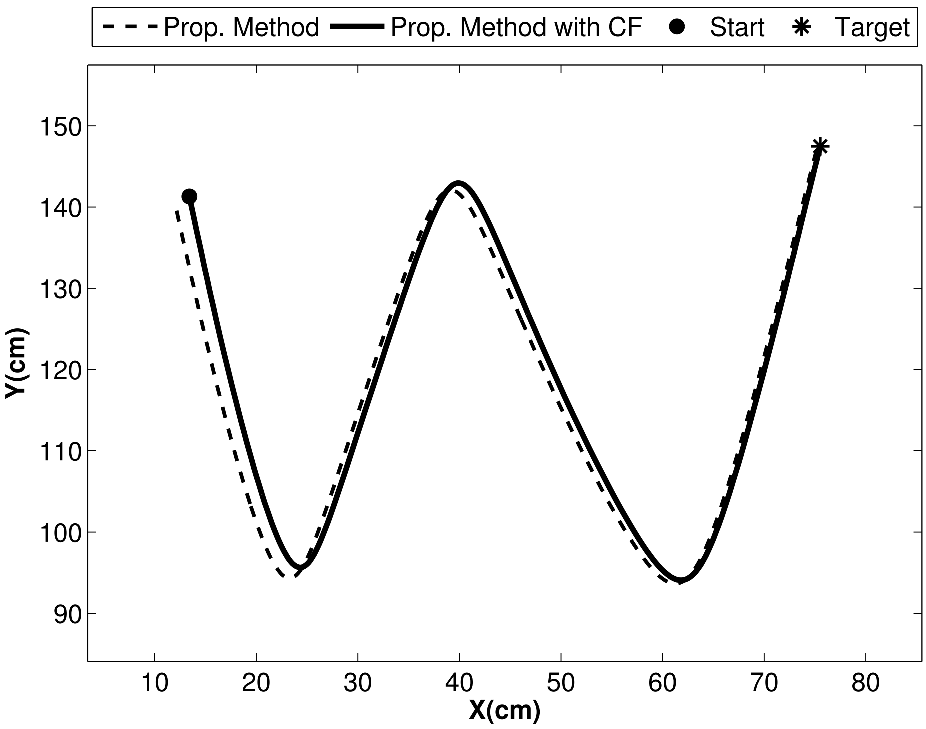


Gen. Traj. with Curve Fitting
Gen. Traj.
Demo.
Target
100
50
0
X(cm)
0
1
2
3
4
5
T(s)
160
140
120
100
80
Y(cm)
T(s)

(d)

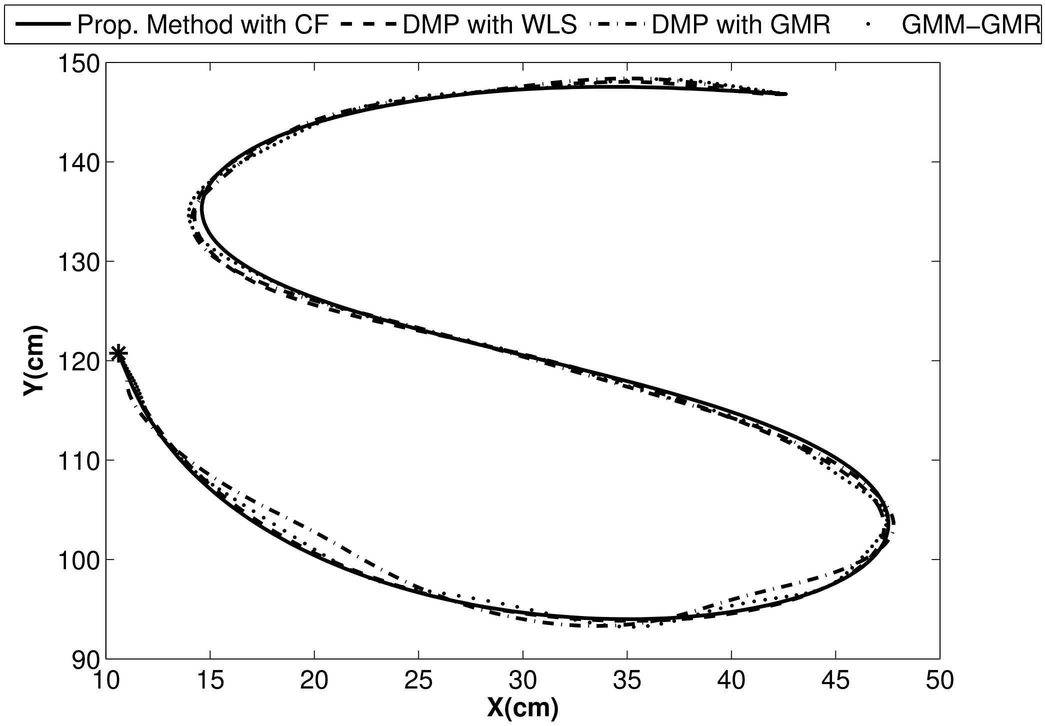


(e)

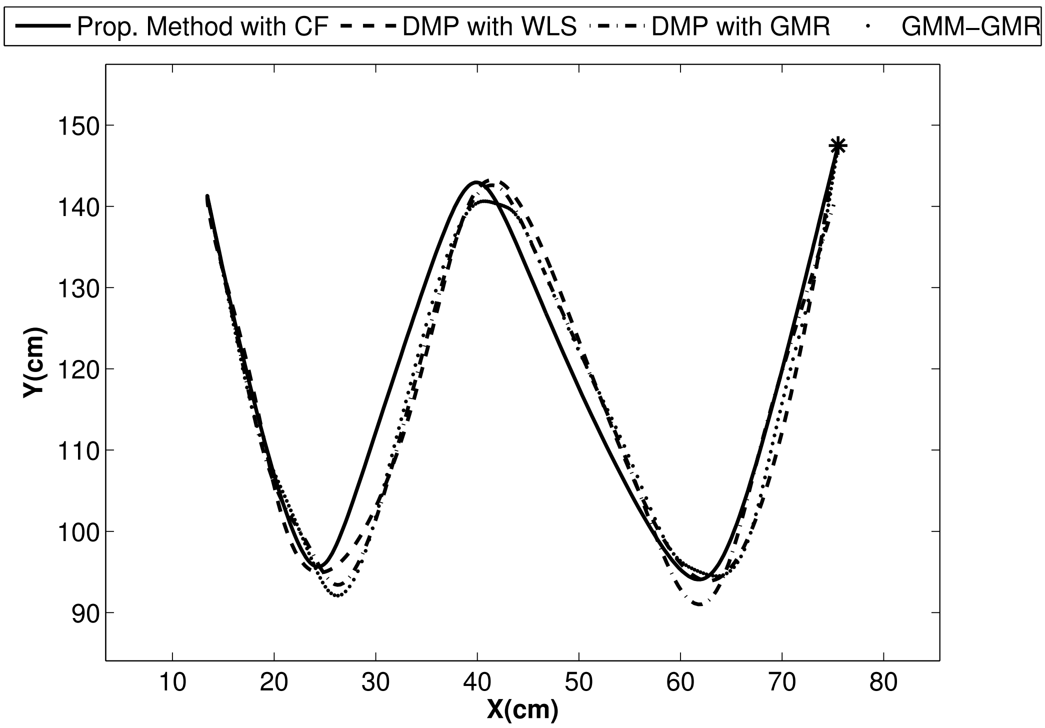


(f)

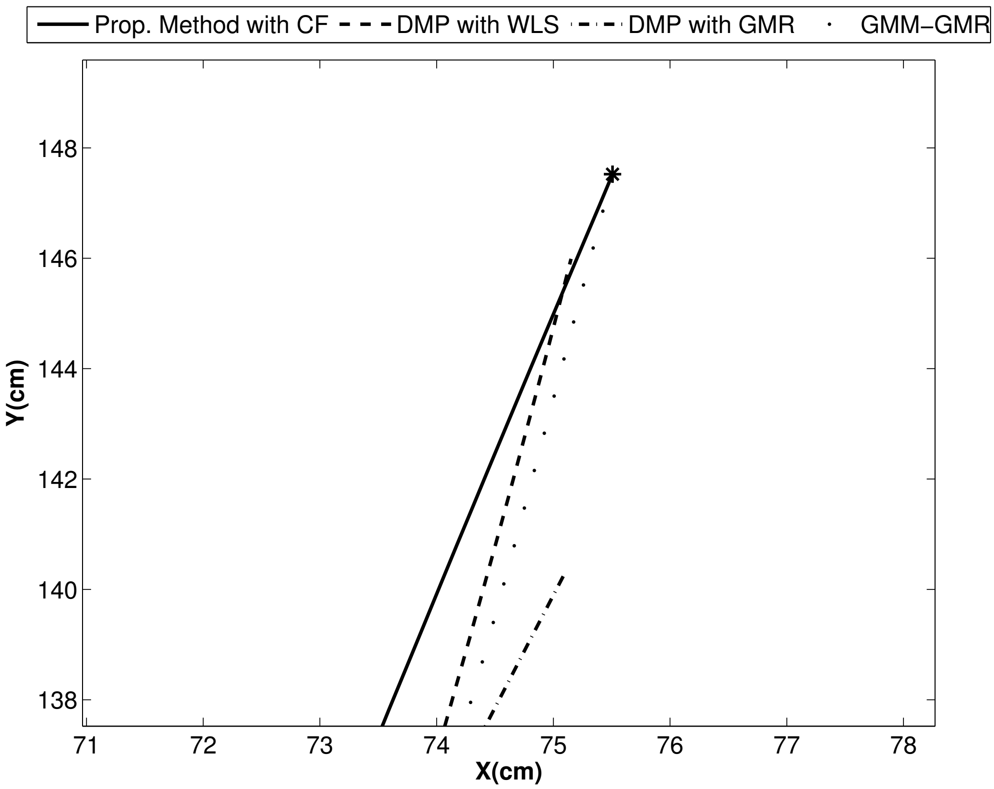
Figure 2.3: Example of 2D motions (S-motion, W-motion) using the robust motion reconstruction method based on PCA.


Prop. Method with CF
DMP with WLS
DMP with GMR
GMM−GMR
Y(cm)
X(cm)
150
140
130
120
110
100
90
10
15
20
25
30
35
40
45
50

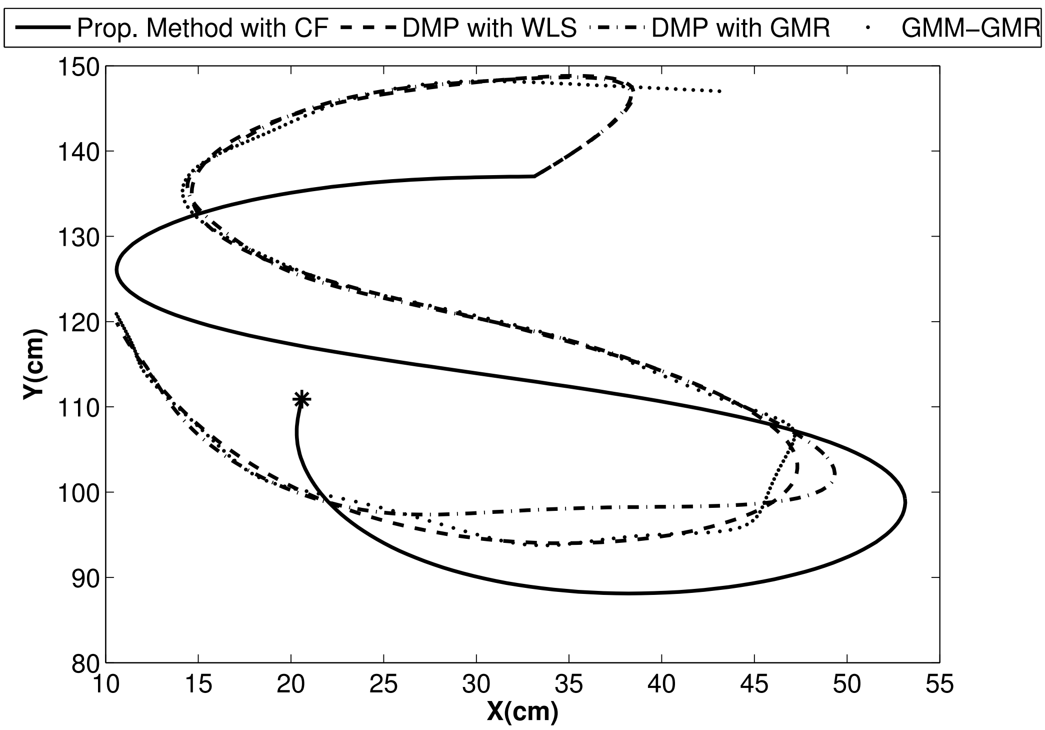
(a)


Prop. Method with CF
DMP with WLS
DMP with GMR
GMM−GMR
Y(cm)
X(cm)
150
140
130
120
110
100
90
10
20
30
40
50
60
70
80

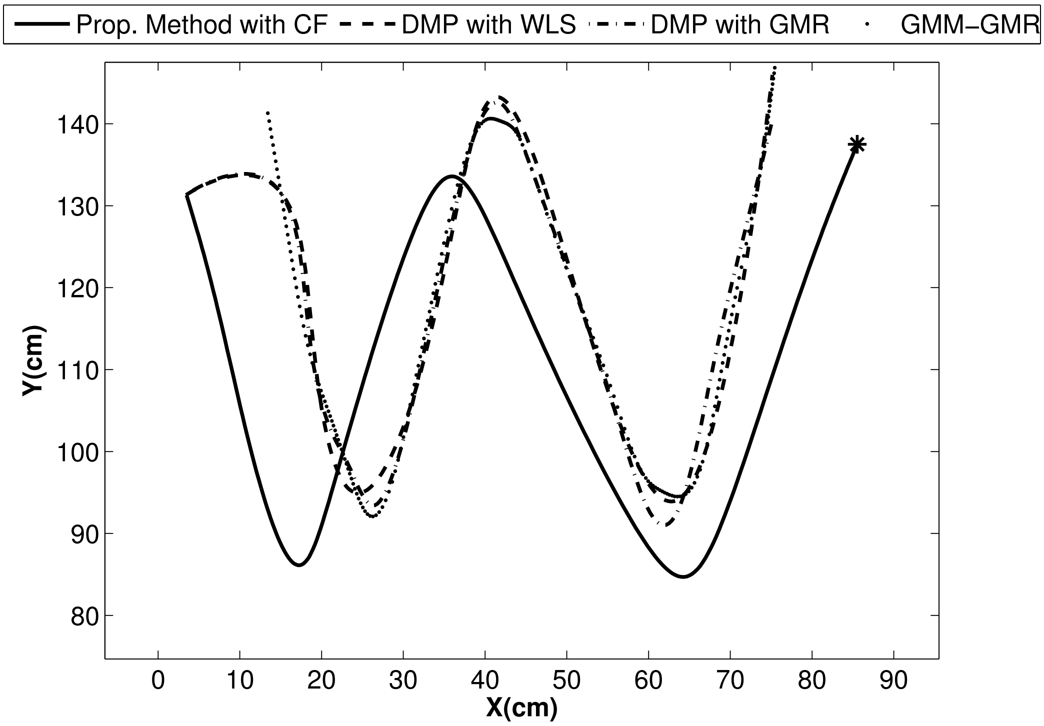
(b)


Prop. Method with CF
DMP with WLS
DMP with GMR
GMM−GMR
148
146
144
142
140
138
Y(cm)
71
72
73
74
75
76
77
78
X(cm)

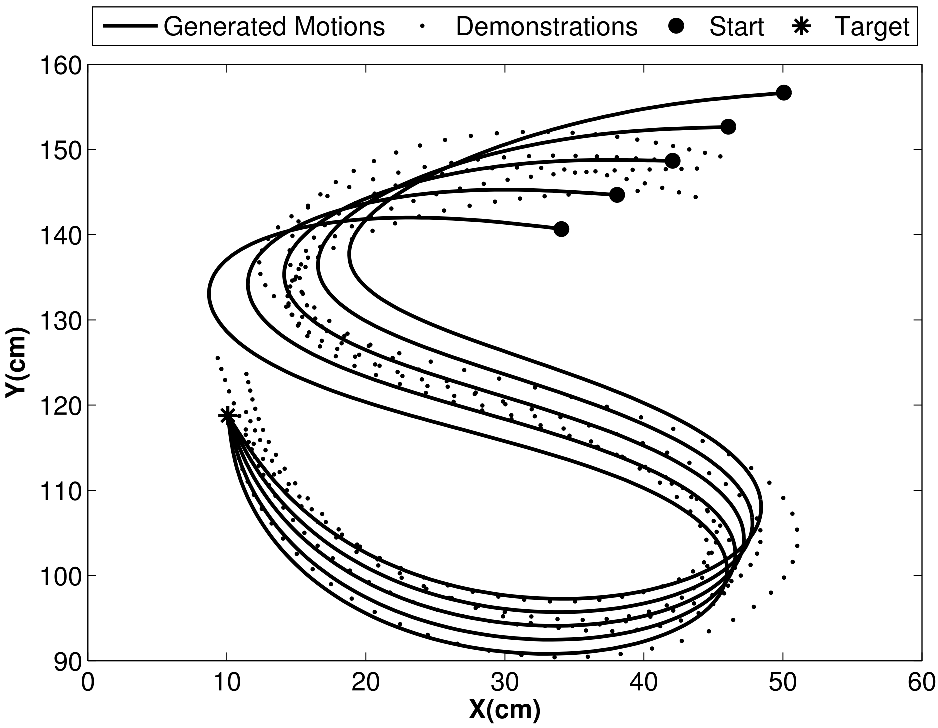
(c)


Prop. Method with CF
DMP with WLS
DMP with GMR
GMM−GMR
150
140
130
120
110
100
90
80
Y(cm)
10
15
20
25
30
35
40
45
50
55
X(cm)

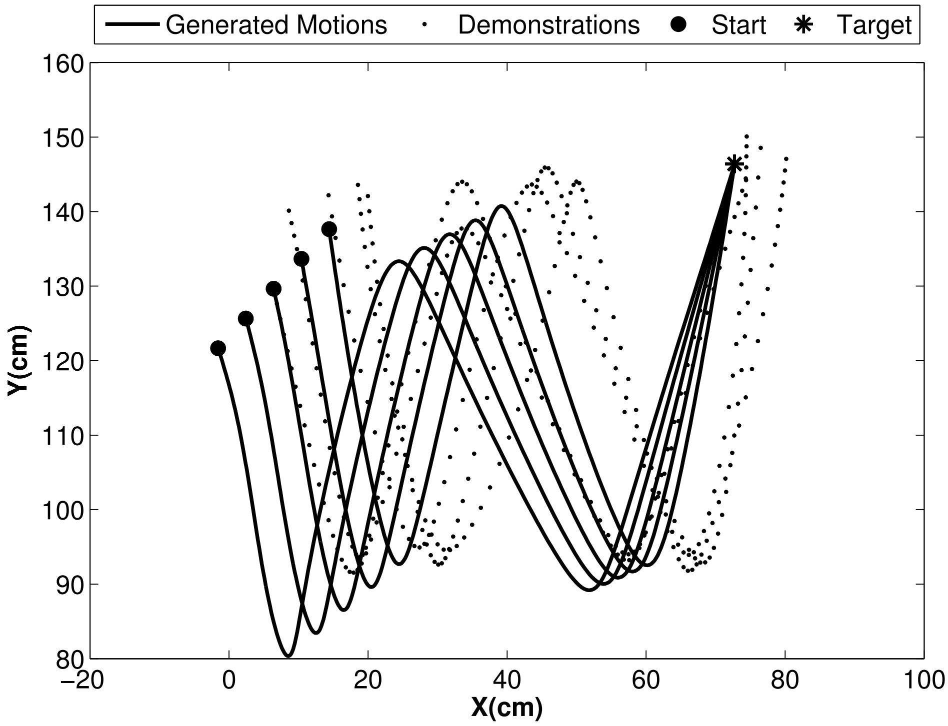
(d)


Prop. Method with CF
DMP with WLS
DMP with GMR
GMM−GMR
Y(cm)
X(cm)
140
130
120
110
100
90
80
0
10
20
30
40
50
60
70
80
90

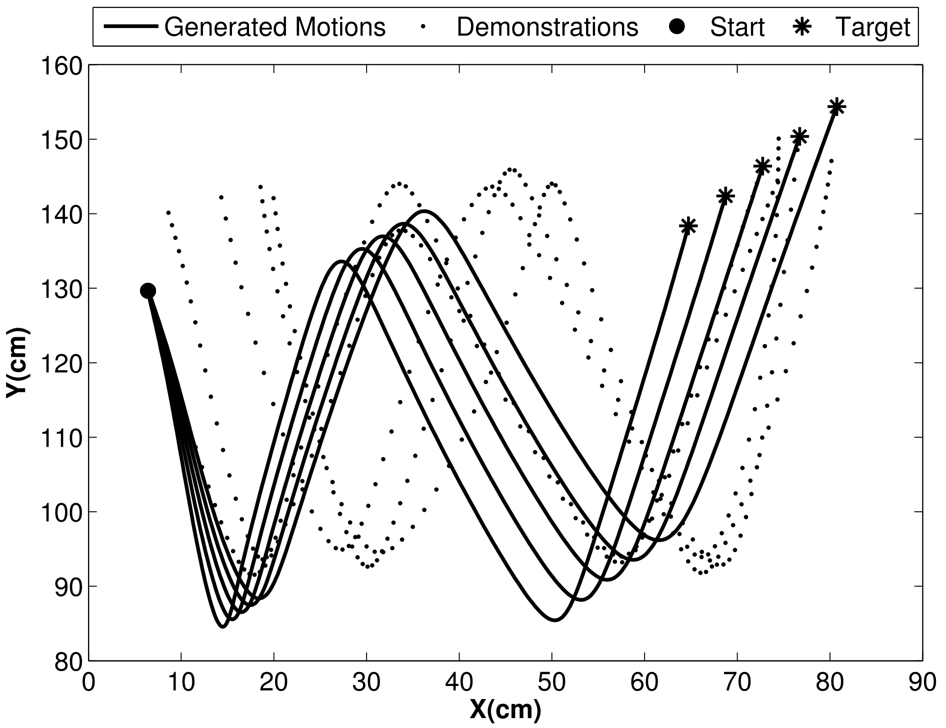
(e)


Generated Motions
Demonstrations
Start
Target
Y(cm)
X(cm)
160
150
140
130
120
110
100
90
0
10
20
30
40
50
60

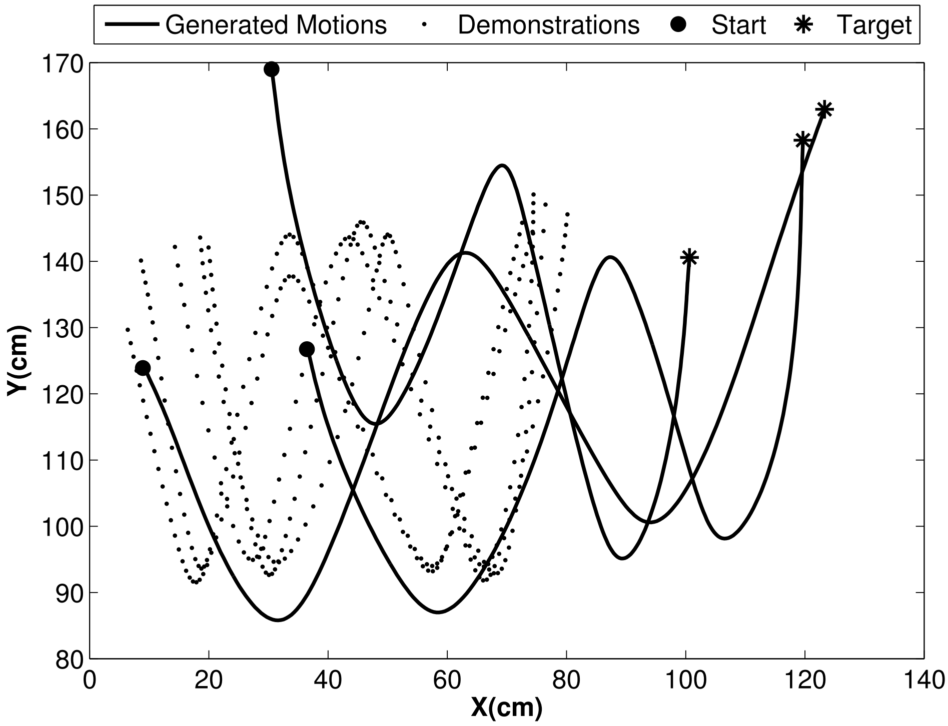
(f)


Generated Motions
Demonstrations
Start
Target
Y(cm)
X(cm)
160
150
140
130
120
110
100
90
80
−20
0
20
40
60
80
100

(g)


Generated Motions
Demonstrations
Start
Target
Y(cm)
X(cm)
160
150
140
130
120
110
100
90
80
0
10
20
30
40
50
60
70
80
90

(h)



(i)

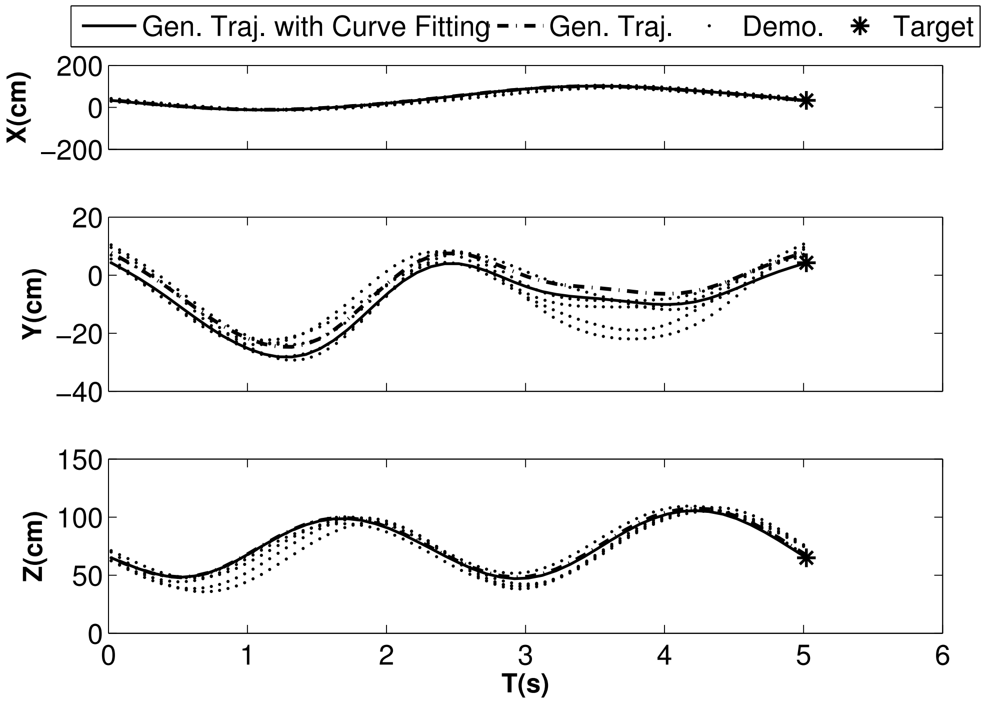
Figure 2.4: Example of 2D motions generated by a proposed robust motion reconstruction based on PCA (multi-target(f) with different methods: DMP with WLS, DMP with GMR and GMM-GMR.

To verify the performance of the proposed motion reconstruction method based on PCA, S-shaped and W-shaped 2D motions, and a ∞-shaped 3D motion are demonstrated and learned.

Figures 2.3 (a), (b), (d), and (e) show the human demonstrations (dotted lines) and trajectories generated by the proposed motion reconstruction method (solid lines) for S-shaped and W-shaped motions, respectively. Figures 2.3 (c) and (f) show the differences between the trajectories generated by Eq. (2.11) (dashed line) and the trajectories modified with curve fitting (CF) given by Eq. (2.12) (solid line). The figures also show that CF satisfactorily modifies the generated trajectories to satisfy the boundary conditions.

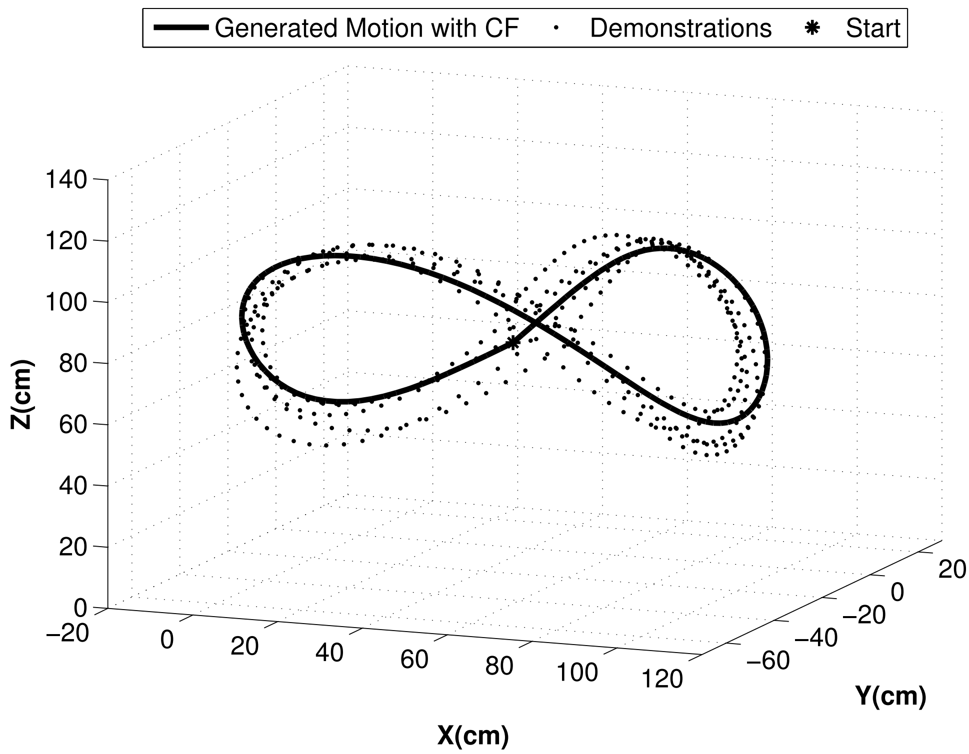
Figure 2.4 shows the advantages of the proposed method as compared with DMP generated by GMR [46] and by WLS [45]. Figures 2.4 (a) and (b) show the trajectories reconstructed by using the proposed method (solid line), DMP with WLS (dashed lines), DMP with GMR (dashed-dotted lines), and a Gaussian mixture model (GMM)-GMR [52] (dotted lines) generated from the human demonstrations (see Figures 2.3 (b) and (e)). As shown in Figure 2.4 (c), which is a closeup of the area of the endpoint of the W-shaped trajectories, the trajectories generated by using DMP with WLS and DMP with GMR reach the given target point with some small error. Since DMP is based on DSs, its performance is controlled by appropriate stiffness and damping gains. By using DMP, the robot can reach the target point, but it is basically difficult to avoid slight errors at the endpoint of the generated trajectory (see Figure 2.4 (c)).

Figures 2.4 (d) and (e) clearly show the advantage of the proposed method over the other methods. When the start and target points significantly change, only the proposed method generates the appropriate trajectories. The other methods try to recover to the demonstrated trajectories. Pastor et al. proposed a modified DMP to deal with significant changes in the start and target points [46]; however, the accuracy of positioning at the start and target points strongly depends on parameters such as the stiffness and damping coefficient. As shown in Figures 2.4 (f) – (i), the proposed method can adaptively generate trajectories according to significant changes in the start and target points.



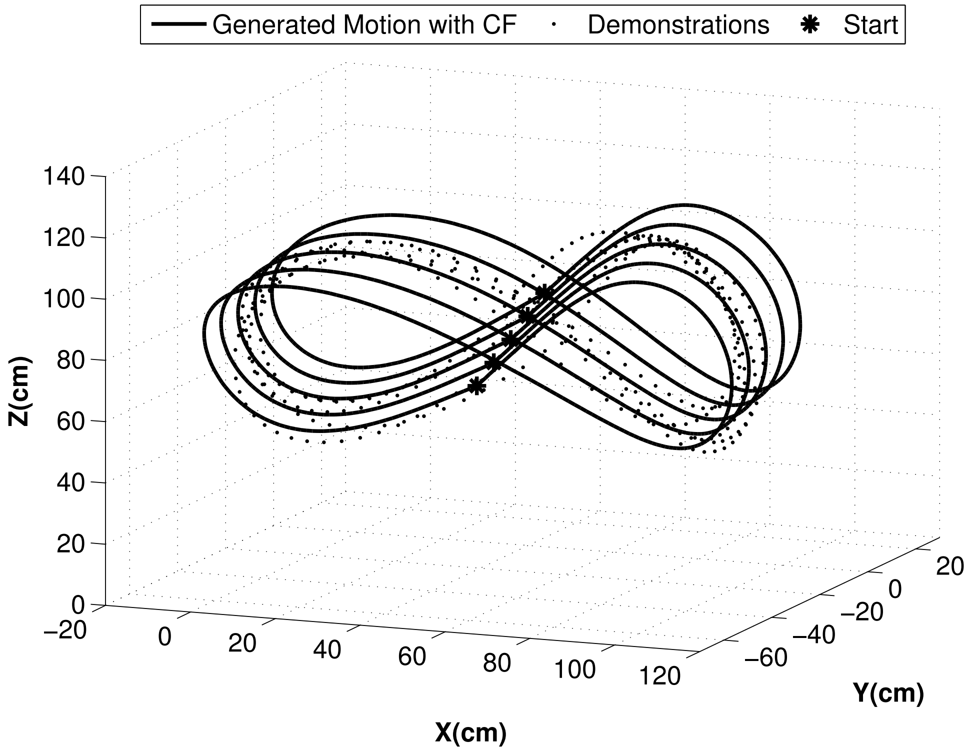
Gen. Traj. with Curve Fitting
Gen. Traj.
Demo.
Target
200
0
−200
X(cm)
20
0
−20
−40
Y(cm)
150
100
50
0
Z(cm)
0
1
2
3
4
5
6
T(s)

(a)

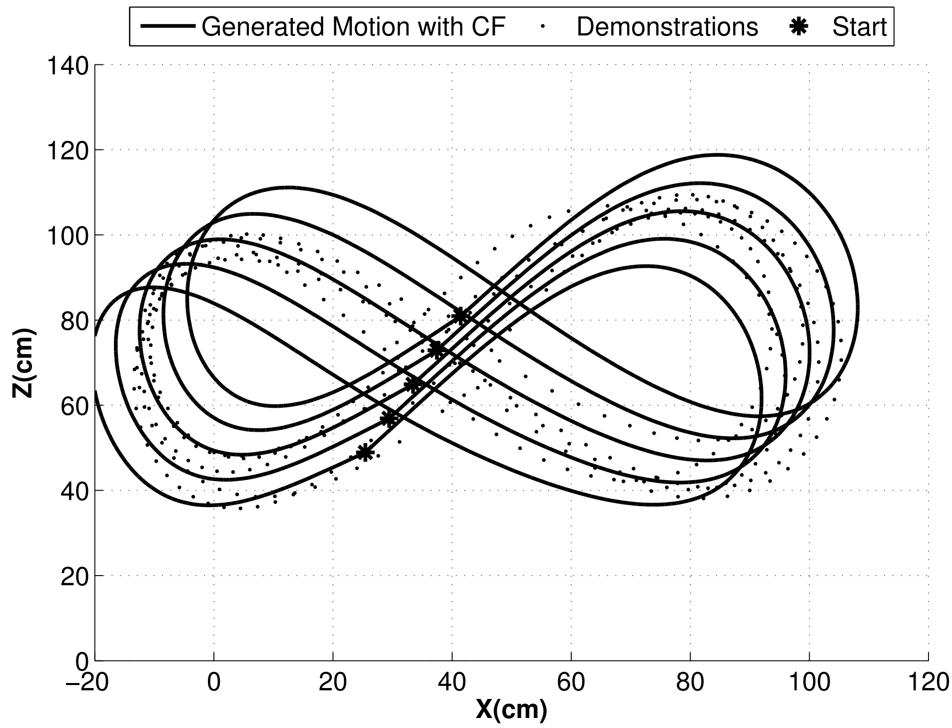


Generated Motion with CF
Demonstrations
Start
140
120
100
80
60
40
20
0
Z(cm)
−20
0
20
40
60
80
100
120
X(cm)
20
0
−20
−40
−60
Y(cm)

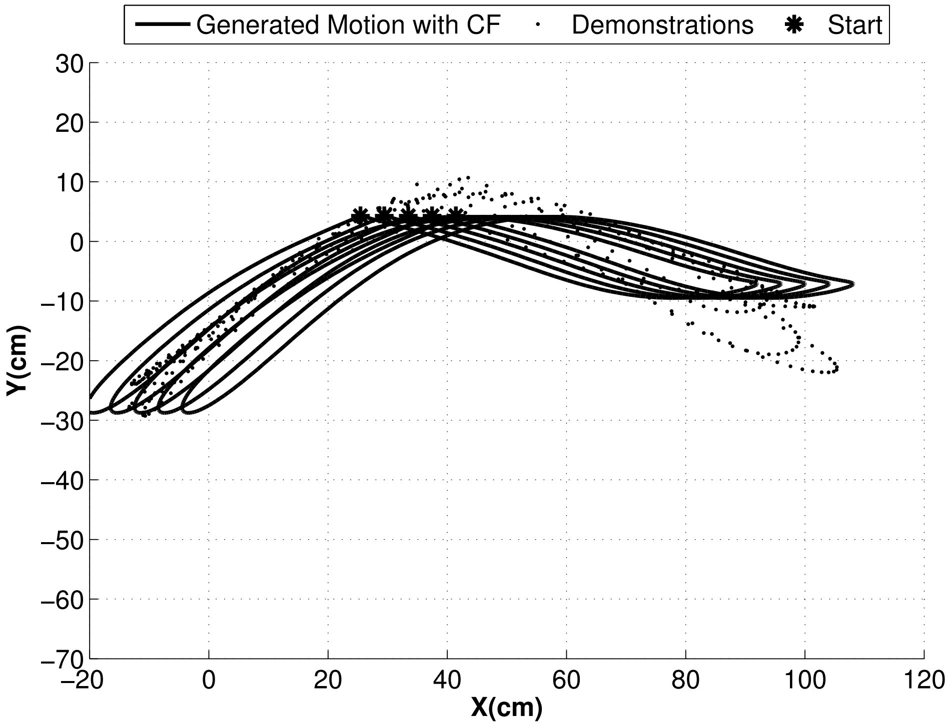
(b)


Generated Motion with CF
Demonstrations
Start
Z(cm)
140
120
100
80
60
40
20
0
−20
0
20
40
60
80
100
120
X(cm)
−60
−40
−20
0
20
Y(cm)

(c)


Generated Motion with CF
Demonstrations
Start
Z(cm)
140
120
100
80
60
40
20
0
−20
0
20
40
60
80
100
120
X(cm)

(d)



(e)

Figure 2.5: Example of continuous 3D motions generated by a proposed robust motion reconstruction based on PCA.

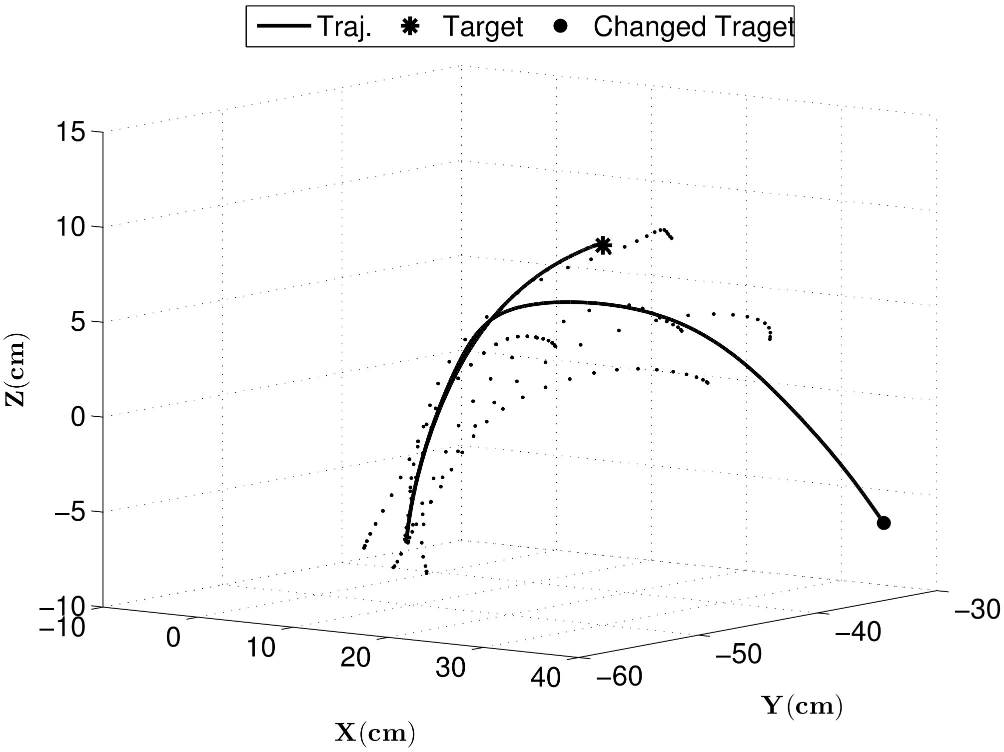
Figure 2.5 shows the result of motion generation for the ∞–shaped 3D motion. In Figure 2.5 (b), the five dotted lines are the trajectories of the human demonstrations, while the solid line is the generated motion. The proposed method adapts well to changes in the start or target points even for 3D motions, as shown in Figures 2.5 (c) – (e).

Figure 2.6 shows the adaptability of the proposed method to a sudden change in the target position. The original trajectory is designed so that a robot hand reaches the target point ('$*$' in Figures 2.6 (a) and (b)) in 2.0 s. In order to verify the adaptability of the proposed method, the target position is suddenly changed at $t_c = 1.0$ s to '$\bullet$' in Figures 2.6 (a) and (b). According to the change in the target position, the boundary condition is modified as

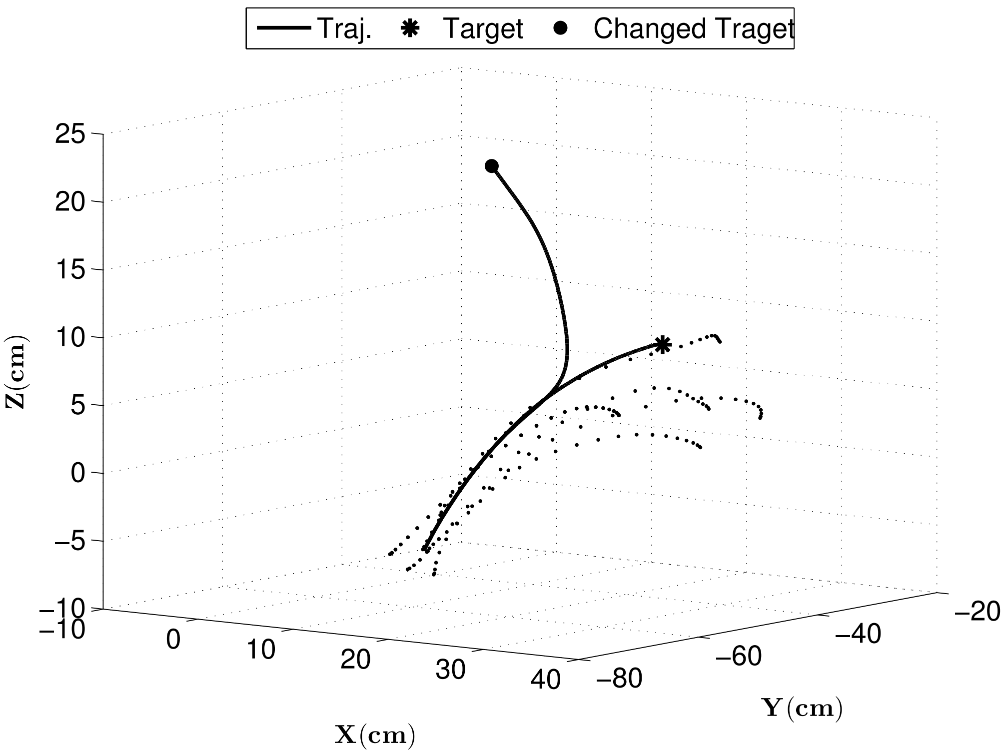
$$\boldsymbol{b}_k = [\, p_k(t_c) \;\; p_k^{\mathrm{new}}(t_f) \;\; \dot{p}_k(t_c) \;\; \dot{p}_k^{\mathrm{new}}(t_f) \,]^{\mathrm{T}}, \tag{2.13}$$

where $p_k(t_c)$ and $\dot{p}_k(t_c)$ are the position and velocity at $t = t_c$, while $p_k^{new}(t_f)$ and $\dot{p}_k^{new}(t_f)$ are the position and velocity of the changed target. Substituting the new boundary condition $\boldsymbol{b}_k$ into Eqs. (2.10) and (2.12), the coefficients $\hat{\boldsymbol{w}}_k$, α, and β are updated in real time. Then, a new trajectory is generated from $t = t_c$ to $t = t_f$ using Eqs. (2.11) and (2.12) implementing the updated coefficients.

The proposed motion reconstruction method based on PCA requires only the boundary conditions, i.e., the positions and velocities at the beginning and end of a motion. Even if the start or target positions or the motion size are significantly different from those in the human demonstrations, the proposed method has good adaptability to the changes, maintaining the characteristics of the demonstrations.



(a) Example of tracking to changed target 1



(b) Example of tracking to changed target 2

Figure 2.6: Tracking adaptability of proposed method to changes in the target.

Chapter 3. Evolution of Movement Primitives Database

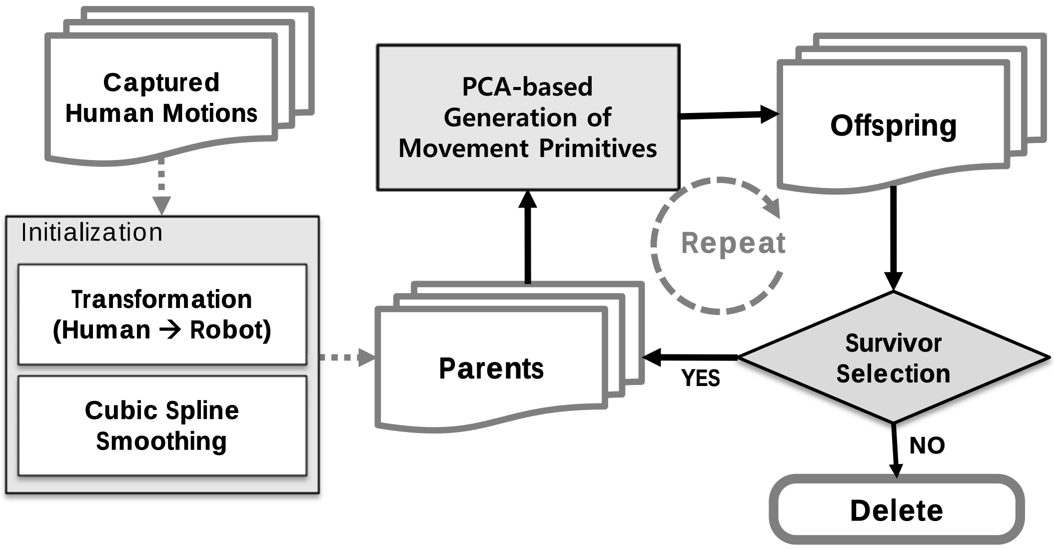


Figure 3.1: Proposed Off-line Imitation Learning Framework

The generation of a movement primitives database from human demonstrations is discussed in this section.

The proposed imitation learning framework ensures that the shape of the human demonstration is retained. In this section, we first discuss how a humanoid robot can imitate human motions, retaining their shape, while a cost function for robots, such as minimizing the sum of torques, is considered.

Figure 3.1 shows the proposed off-line imitation learning process. The off-line process is composed of four subprocesses: (1) transformation from human motion to humanoid motion, (2) evolutionary process of movement primitives, (3) construction of a clustered database of principal components, and (4) PCA-based generation of movement primitives.

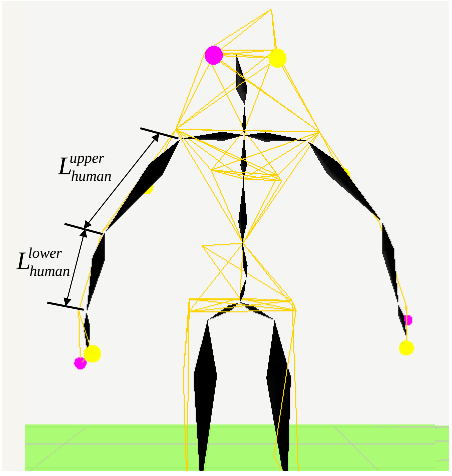
3.1 Transformation from Human to Humanoid Robot Motion

First, human demonstrations are captured by using a motion capture system. The captured data are usually composed of a sequence of joint angles, end-point's position, and orientation. In the transformation of the captured human motions into optimized robot motion data for a humanoid robot, the differences between a human and the humanoid robot, such as the lengths of limbs and limitation of joint angles, must be considered. Therefore, first a scaled model that has the same dimensions as the humanoid robot is designed, as shown in Figure 3.2. Then, the transformation from human motion to the motion of the scaled model is performed so that errors in arm and hand postures are minimized [1] (Appendix A).

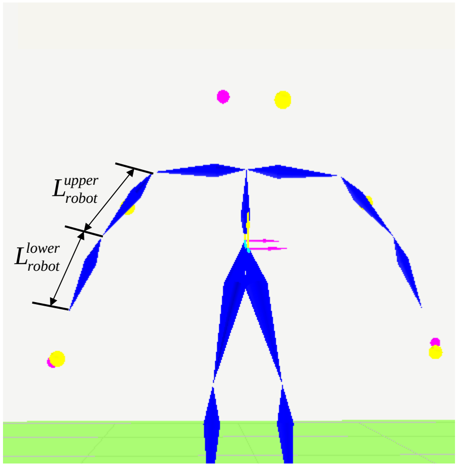
Each human demonstration may have a different completion time (i.e., different data length). Therefore, the difference is adjusted as

$$\boldsymbol{p}_{\mathrm{mod},l}\left(\frac{\bar{T}}{T_l}t\right) = \boldsymbol{p}_{\mathrm{demo},l}(t), \quad (l = 1, \ldots, N_{\mathrm{demo}},\ 0 \le t \le T_l), \tag{3.1}$$

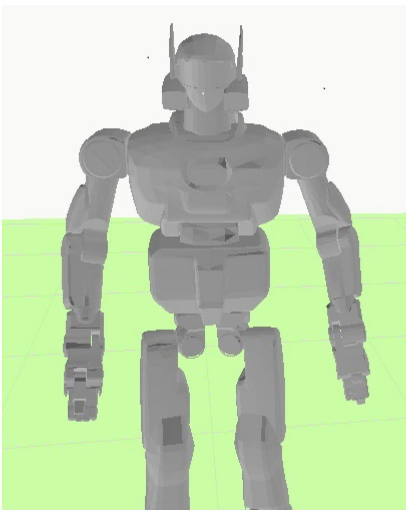
where $\boldsymbol{p}_{\mathrm{mod},l}$ and $\boldsymbol{p}_{\mathrm{demo},l}$ are the modified and original augmented position trajectories of the l-th demonstration, respectively, N_{demo} is the number of the demonstration, T_l is the completion time of the l-th demonstration, and $\bar{T}$ is the average of the completion time of the N_{demo} demonstrations. Figure 3.3 shows an example of the adjustment when $N_{\mathrm{demo}} = 3$. A smoothing spline is applied to the human demonstrations to remove the noise caused by the optical motion capture system.



(a) Upper body of human in capturing



(b) Scaled human model



(c) Humanoid model

Figure 3.2: Captured human model, scaled model, and humanoid model

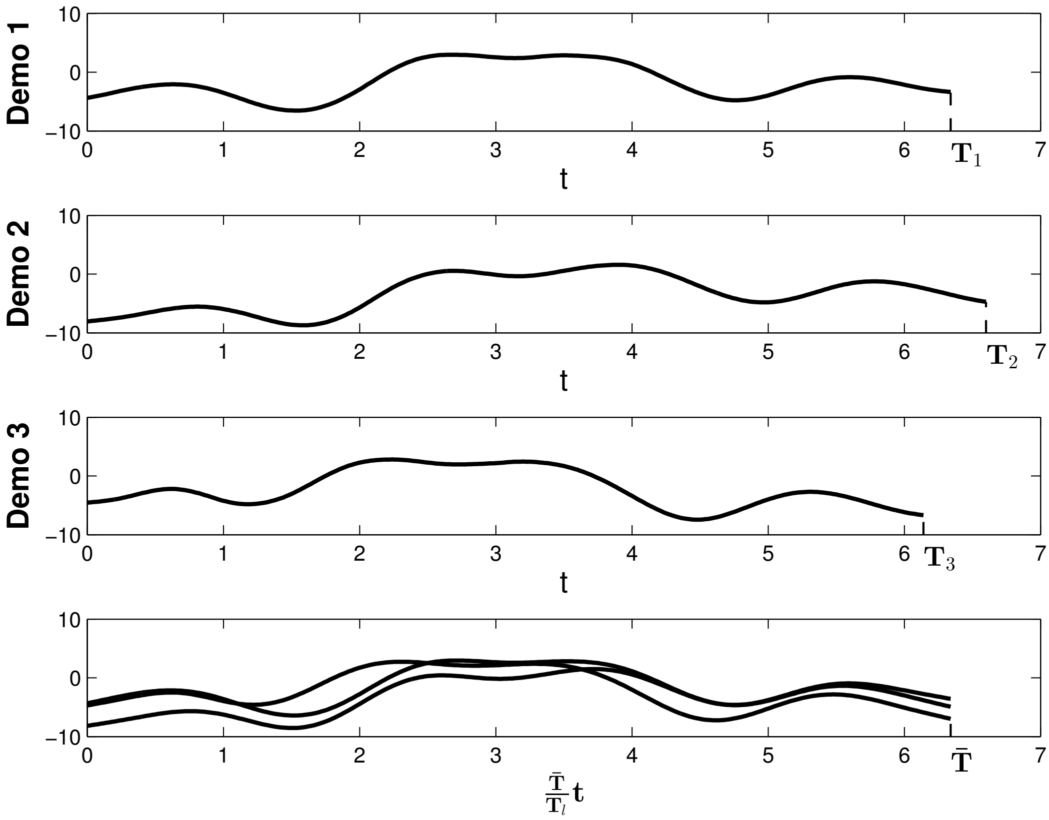


Figure 3.3: Synchronization of data's length by the mean time

3.2 Concept of Evolutionary Process of Movement Primitives

The terminology of evolutionary algorithms is used here to explain the development of movement primitives. In Figure 3.1, each iteration of the evolutionary process is called a *generation*. The movement primitive corresponds to an *individual*, a set of individuals is called a *parent*. From the parents, a new individual is created. The newly generated individual is iteratively modified so that a fitness function is minimized (or maximized). The optimized individual is called an *offspring*.

3.2.1 Parent

Each parent consists of a collection of individuals, which are defined as the movement primitives. The initial individuals are transformed human demonstrations. Therefore, the initial parent contains only the characteristics of the human demonstrator.

3.2.2 Condition Set and Database Structure

A trajectory can be represented in Cartesian or joint space. Although both representations are possible in the proposed framework, if a trajectory is given in Cartesian space, the redundancy problem must be solved. In order to solve the redundancy problem, the elbow elevation angle of an arm [11, 12] is considered. Before starting the evolutionary process of movement primitives, the workspace size and *condition set* must be determined. Usually a motion is made up of the trajectories from the start to the target point. A condition set is composed of the target position and elbow elevation angle.

3.2.3 Generation of New Individuals

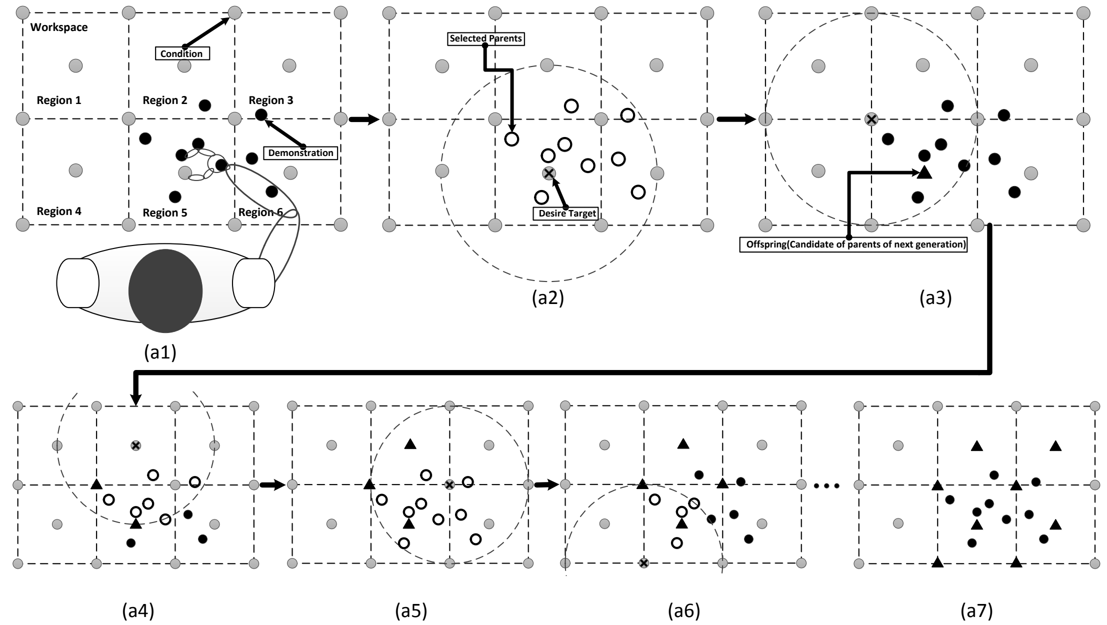
Movement primitives (individuals) the targets of which neighbor each other become parents to generate a new individual. First, the principal components of parents are computed. Then, PCA-based motion generation is applied to generate a new individual. In the on-line PCA-based motion generation discussed in Section 2, the first three principal components are used (see Eq. (2.11)). However, in the generation of a new individual, the first four principal components are used to incorporate a robot property, such as energy consumption. In Section 3.4, the PCA-based generation of new individuals is described in detail.

3.2.4 Survivor Selection

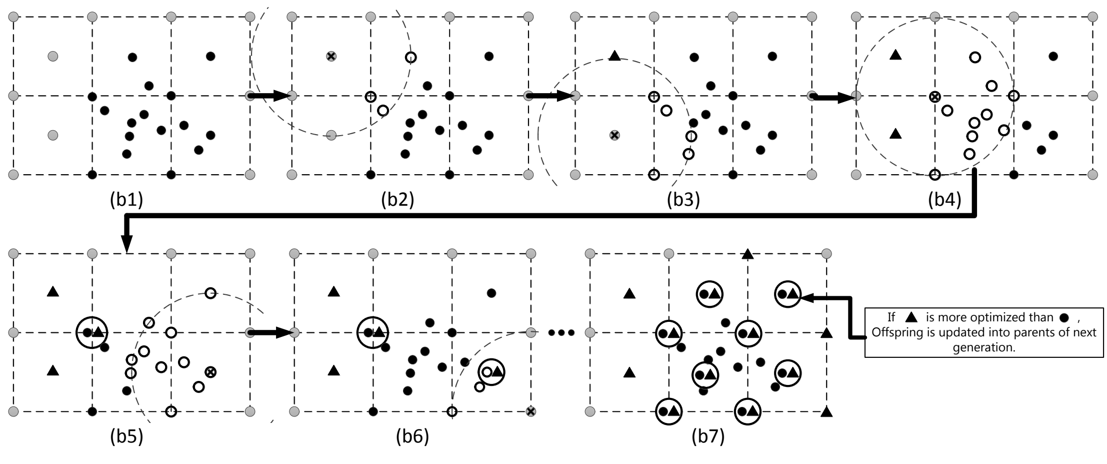
The generated individual has to be evaluated in terms of a fitness function specified by the user. The generated individual is iteratively modified to minimize (or maximize) the fitness function. The optimized individual, i.e., offspring, is compared with the previous offspring (generated in the previous generation) and if it survives the comparison, the offspring is overwritten in the movement primitive database.

3.2.5 Termination of the Evolutionary Process

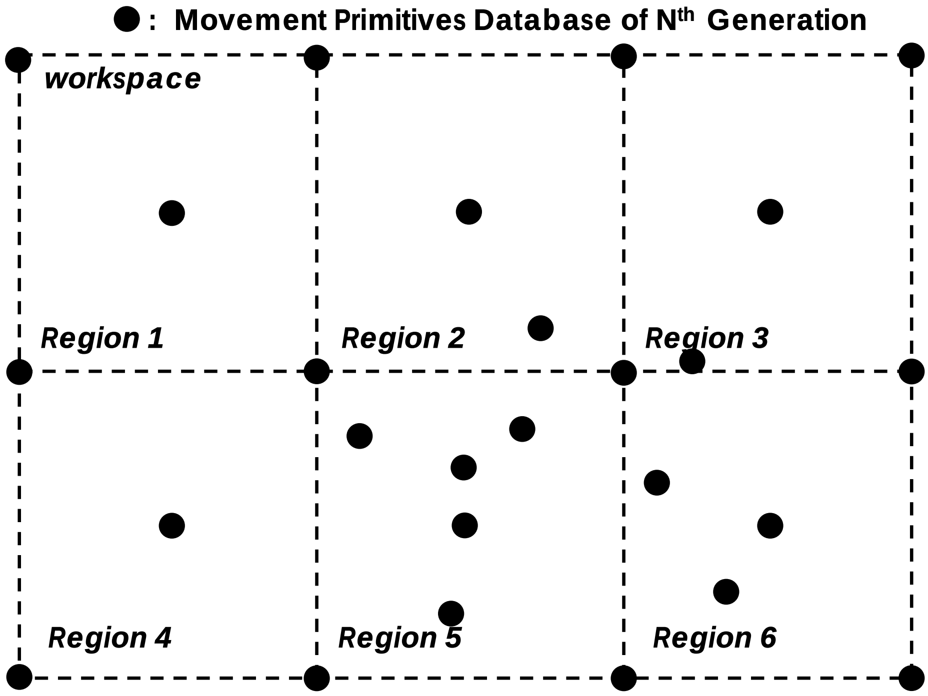
In each generation, the average evaluation value of the fitness function is computed. If the average of the current generation is worse than that of the previous generation, it may be impossible to improve it in the next generation. Accordingly, the evolutionary learning process is terminated.



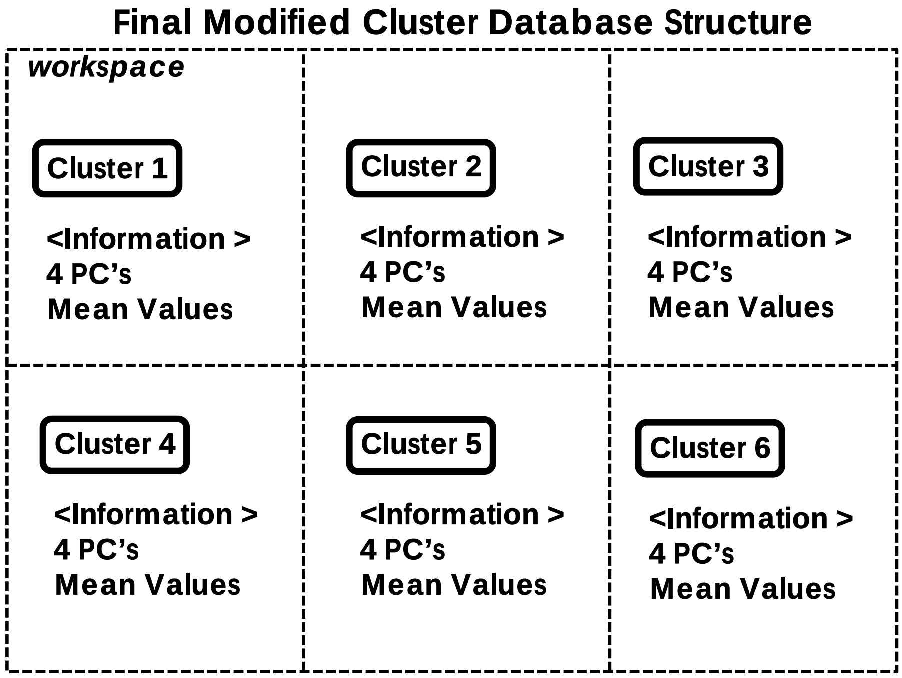
(a) Learning process of 1^{st} Generation



(b) Learning process of 2^{nd} Generation



(c) Fully filled up for the condition set and well-trained final Movement primitives of N^{th} Generation



(d) Cluster Database Structure

Figure 3.4: Simple model explaining the modified cluster database structure

3.3 Details of Evolutionary Process of Movement Primitives with Improved Cluster Database Structure

Figure 3.4 shows the concept of the evolutionary process of movement primitives and the improved cluster database structure. In order to explain the details of the evolutionary process, imitation learning of a human reaching motion (Figure 3.4 (a)) is taken as an example. Suppose that the human demonstrates the reaching motions nine times. The demonstration data are transformed into robot motions that meet the robot properties, as described in Section 3.1. The black dots in Figure 3.4 (a) represent the augmented target position vectors (position and elbow elevation angle) of the robot motions transformed from those in human reaching demonstrations.

The workspace is divided into several regions, as shown in Figure 3.4 (a1). A database associated with a region is called a *cluster* in this paper. At the corners and the center of each region, condition sets (target position and elbow elevation angle) are previously specified (Figure 3.4 (a1)). A gray dot in Figure 3.4 indicates that a condition set is specified; however, the movement primitive (motion trajectory) has not yet been generated for the point.

Initially, the transformed demonstrations in Section 3.1 contribute the initial movement primitives database, which is called the parents of the first generation.

In the first generation, a condition set (target point) is selected to generate a movement primitive for the point. The neighboring parents within a previously specified radius are chosen as the parents, as illustrated in Figure 3.4 (a2). If the number of the neighboring parents is less than three, another condition set is searched. A new movement primitive that minimizes a given fitness function is evolved from the parents for the chosen condition set point. The generation of the new movement primitive is discussed in detail in Section 3.4. The evolved movement primitive is regarded as the offspring and as candidate parents of the next generation (black triangle plotted in Figure 3.4 (a3)). This process is iterated for the other condition sets, as illustrated in Figure 3.4 (a3)–(a7). Then, the offsprings are added to the parents of the current generation. The updated parents are to be new parents of second generation (Figure 3.4 (b1)).

In the second generation (Figure 3.4 (b)), the learning process is completed again for all feasible condition sets. A black dot in Figure 3.4 (b) indicates that the point is one of the demonstration points or the condition set for which a movement primitive was generated in the past generation. Even if a parent has already been generated for a condition set, an offspring is generated for the condition set in the current generation. Hence, a condition set may have two movement primitives, one generated in the past generation (black dot, i.e., parent) and one generated in the current generation (black triangle, i.e., offspring). The evaluation results of the two movement primitives are compared using a cost function (in this study, minimizing the sum of joint torque), and if the evaluation value of the offspring is lower than that of the parent, the offspring becomes a parent in the next generation.

Through repetition of the evolutionary process, the movement primitives database covers most of the condition sets in the workspace. Even if condition sets remain without generating movement primitives, the evolutionary process may be terminated by evaluating the average of the optimization values (Section 3.2.5).

It should be noted that only the demonstrated points have movement primitives at the beginning of the evolutionary process (Figure 3.4 (a1)). However, after the evolutionary process, movement primitives are generated also on the specified point that cover almost the entire workspace, as illustrated in Figure 3.4 (c). Then, the robot can generate human-like desired motions in the workspace that reflect both robot and human properties.

The movement primitives the target points of which are involved in *region i* are classified into *cluster i*, as presented in Figure 3.4 (d). As described in Section 2, principal components and mean values are used to produce a new motion. Hence, only the principal components and mean values are stored in the database. They are calculated on each cluster (Figure 3.4 (d)).

Consequently, the improved cluster database structure comprises the principal components and mean values on each cluster. This database structure has significant advantages as compared with that presented in [7]. First, the requirements for the demonstrations are relaxed, and hence, the number of demonstrations can be reduced. Second, the size of the demonstrations and the database is dramatically reduced. Third, it autonomously extends the database to cover the workspace.

Finally, the calculation time for generating the desired motion is reduced because of the composition of database.

3.4 Optimization-based Generation of Movement Primitives in the Evolutionary Process

As mentioned in Section 3.3, when a condition set (target point) is selected to generate an offspring for the point, the neighboring parents within a specified radius are chosen, as illustrated in Figure 3.4 (a2). By applying PCA to the parents, the principal components are calculated. It should be noted that the principal components are time-dependent. In the on-line process described in Section 2, the first three principal components are used to produce a new motion (Eq. (2.11)), since the four coefficients are sufficient to generate a motion satisfying the given boundary conditions. However, the first four principal components are used to produce a new motion (movement primitive) here to incorporate a robot property, such as energy consumption or joint limitation. A new motion is generated as

$$p_k(t) = p_{k,mean}(t) + \sum_{i=1}^{4} \gamma_{k,i}\, p_{k,pc_i}(t) + \gamma_{k,5} \quad (k = 1, \ldots, M), \tag{3.2}$$

where M is the dimension of the augmented position vector. $\gamma_{k,1}, \ldots, \gamma_{k,5}$ are unknown scalar weighing coefficients. These five coefficients can be determined by four boundary conditions (Eq. (2.5)) and a fitness function.

As an example of the fitness function, torque summation is considered. The equation of motion for a robot system is represented as

$$\mathbf{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \mathbf{g}(\boldsymbol{q}) = \boldsymbol{\tau}, \tag{3.3}$$

where $\mathbf{M}(\boldsymbol{q}) \in \Re^{n\times n}$ is the mass matrix, $\mathbf{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \Re^{n}$ is the vector of Coriolis and centrifugal force, and $\mathbf{g}(\boldsymbol{q})$ is the vector of gravitational force. The fitness function is defined as

$$f(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \frac{1}{2}\int_{t_o}^{t_f} \|\boldsymbol{\tau}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\|^2 dt. \tag{3.4}$$

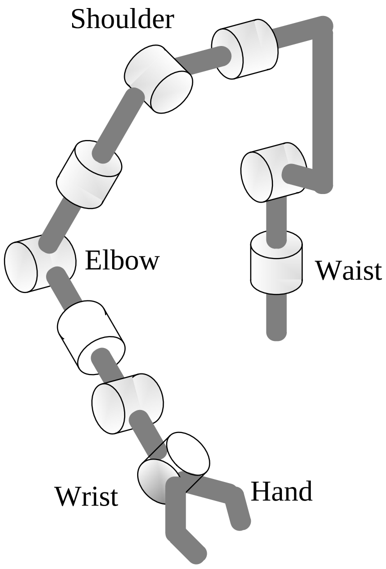
The following optimization problem is considered here :

$$\min_{\boldsymbol{\gamma}} \frac{1}{2} \int_{t_0}^{t_f} \|\boldsymbol{\tau}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\|^2 dt \tag{3.5}$$

$$\text{subject to} \quad \gamma_{k,i_0} - \sigma_{k,i} \leq \gamma_{k,i} \leq \gamma_{k,i_0} + \sigma_{k,i} \quad i = 1, 2, ..., 5 \tag{3.6}$$

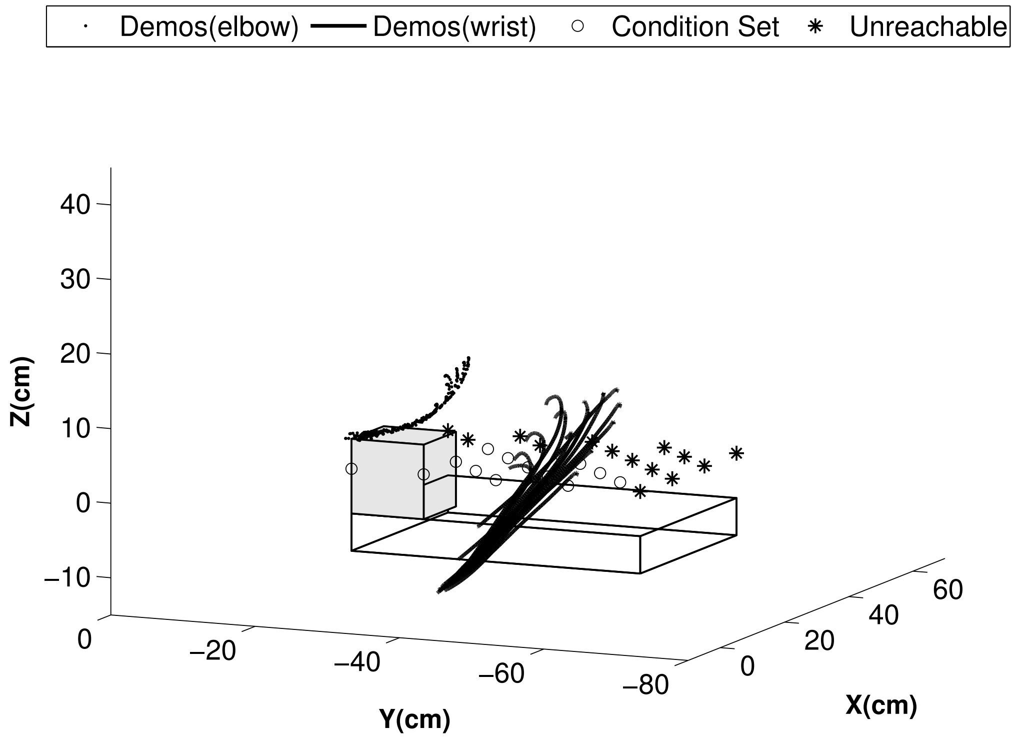
where γ_{k,i_0} are initial values of the coefficients $\boldsymbol{\gamma}_k$ and $\sigma_{k,i}$ are user-defined values, which are appropriate values for the optimization. The coefficients $\hat{\boldsymbol{w}}_k$ calculated by Eq. (2.10) are used as the initial values of $\gamma_{k,i}$ $(i = 1, \ldots, 4)$. $\gamma_{k,5_0}$ are set to zero. Since the first and second principal components are dominant, $\sigma_{k,1}$ and $\sigma_{k,2}$ must be carefully chosen and the others are usually set to small values. By solving the optimization problem using the analytic gradient and Hessian information [53, 54], the coefficients are determined. When the coefficients have been determined, a new individual is generated by Eq. (3.2).

(a) HRP-2

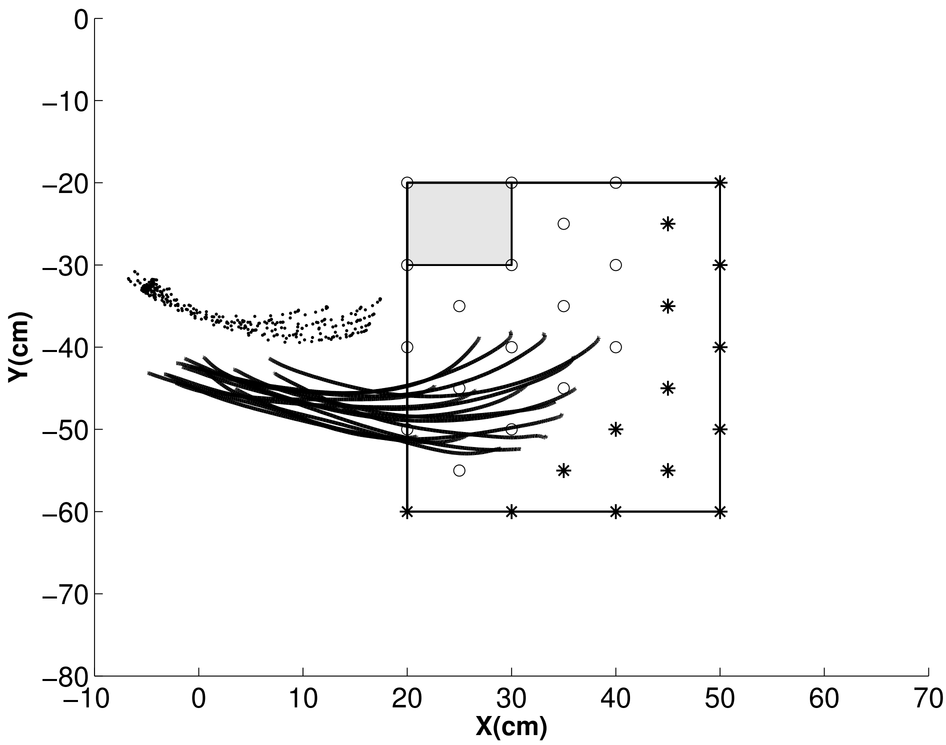


(b) Arm's joint structure

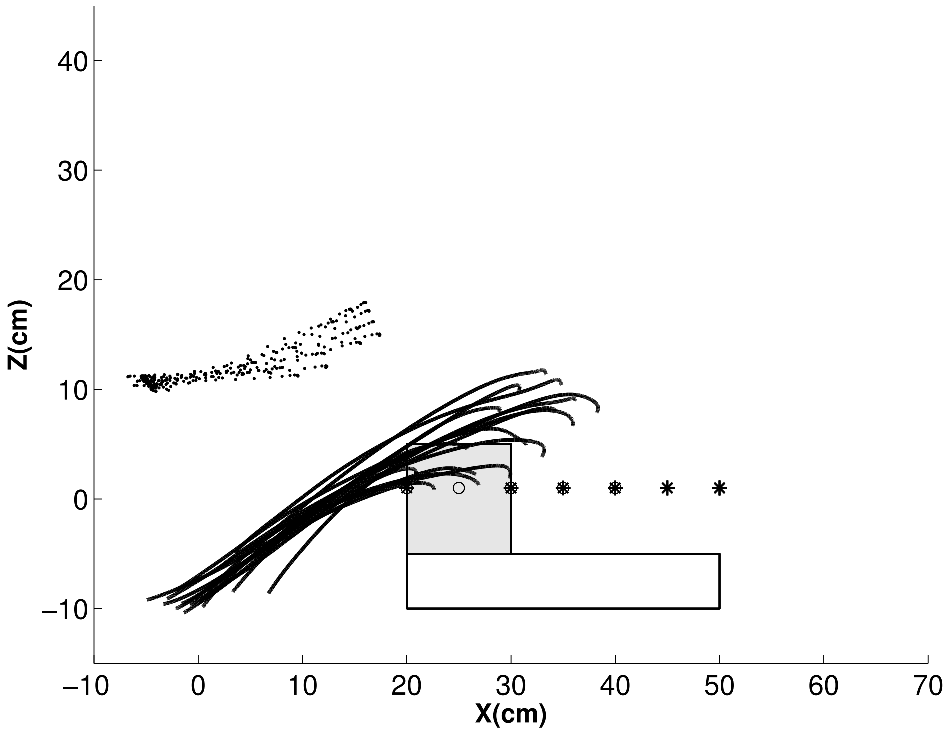
Figure 3.5: Humanoid robot HPR2 and 7-DOF arm structure



(a) XYZ plot

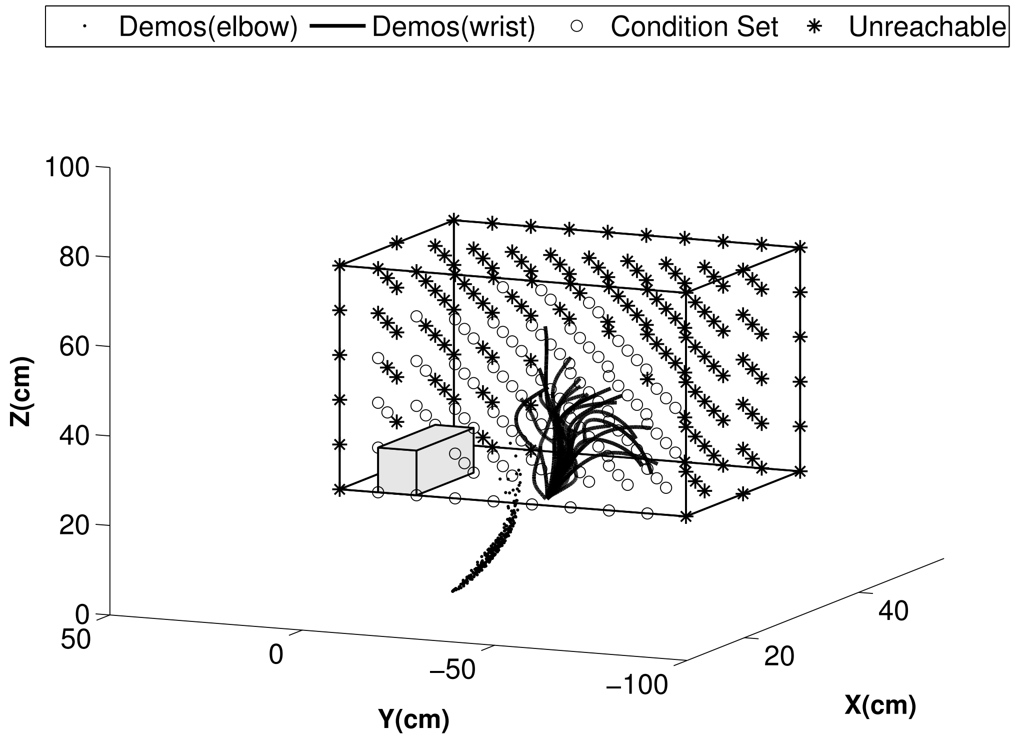


(b) XY plot

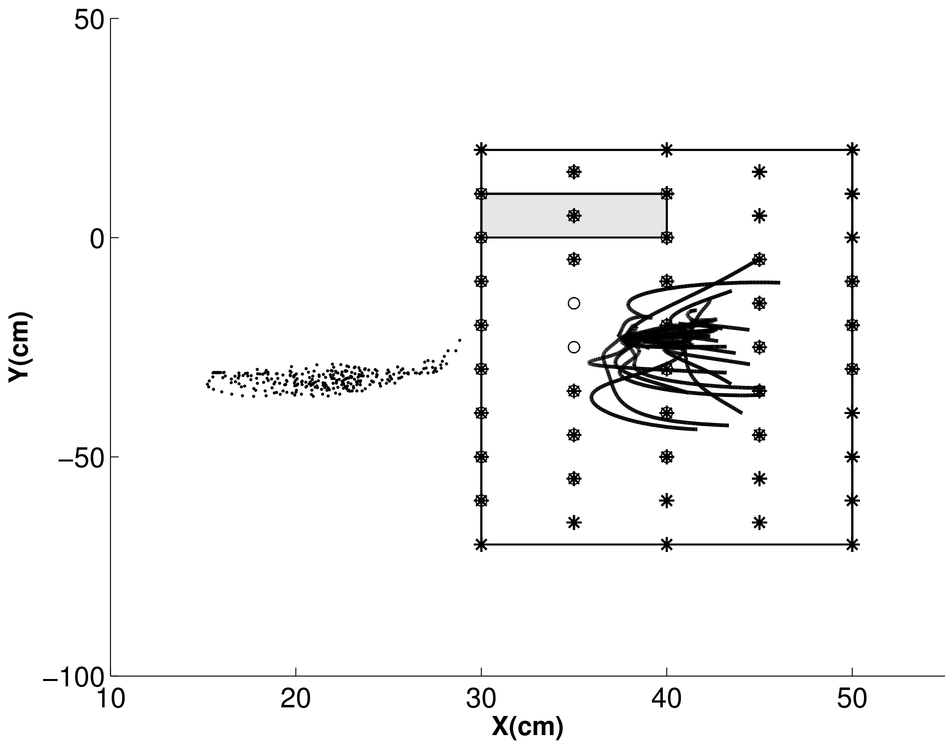


(c) XZ plot

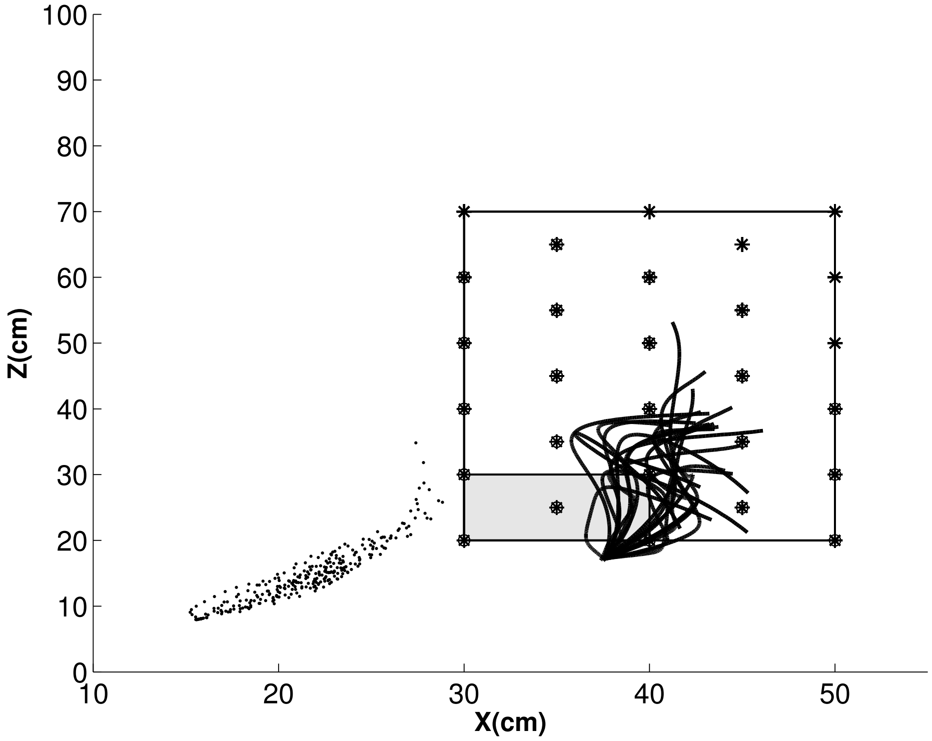
Figure 3.6: Case 1: Reaching an object on the table. Target end-effector positions of demonstrations and condition sets in workspace (gray box shows the size of a region)



(a) XYZ plot



(b) XY plot



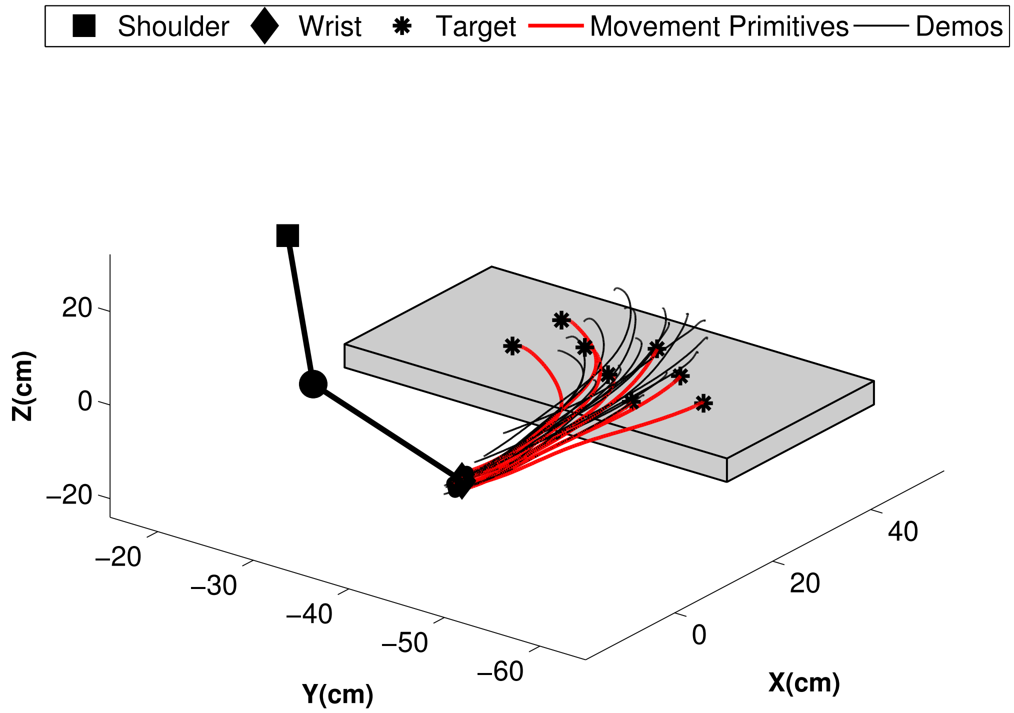
(c) XZ plot

Figure 3.7: Case 2: Reaching an object on the shelf. Target end-effector positions of demonstrations and condition sets in workspace (gray box shows the size of a region)

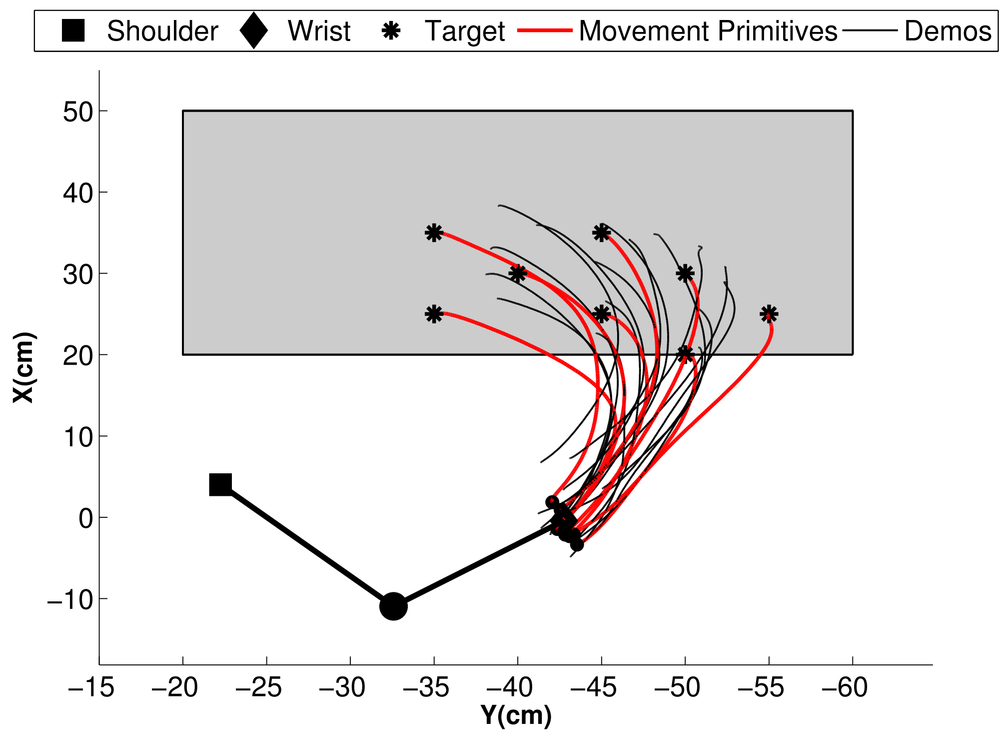
Chapter 4. Case Study

4.1 Given Task

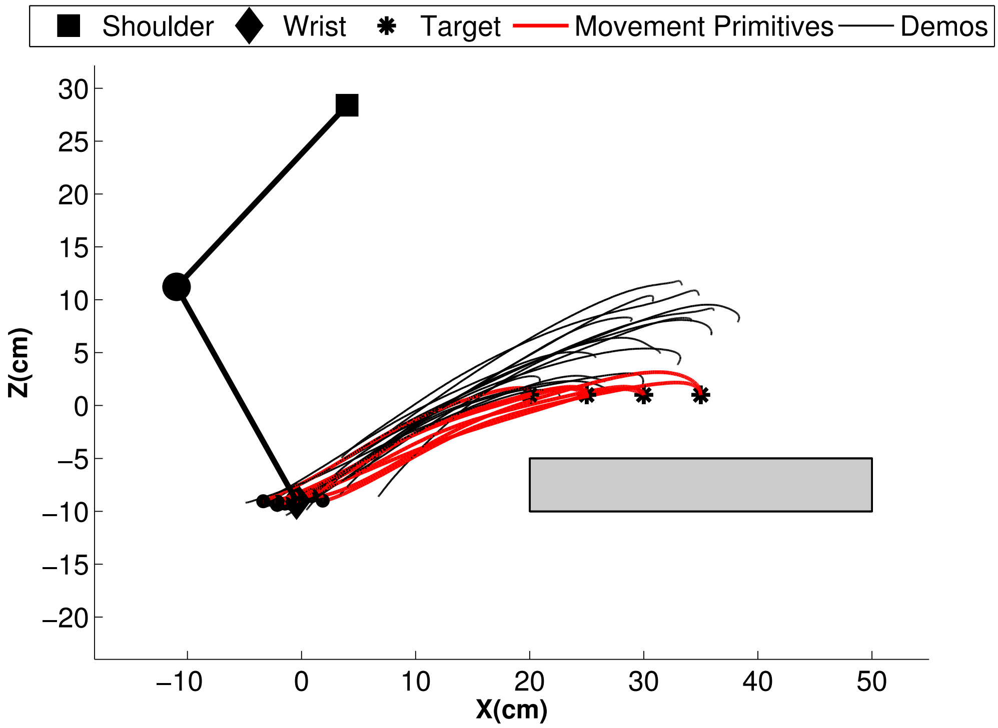
In order to verify the performance of the proposed imitation learning framework, two cases were studied. In Case 1, a human demonstrated a reaching motion to an object on a table 16 times (completion times are $1.9 - 2.1\,\mathrm{s}$), while in Case 2, the human demonstrated a reaching motion to an object on a shelf 21 times (completion times are $1.5 - 2.0\,\mathrm{s}$). The human demonstrations were captured by a motion capture system. Sampling frequency was 50 Hz. The shelf was positioned 60 cm above the table. A VRML model of the 7-DOF right arm of the humanoid robot HRP-2 (Figure 3.5) was used to imitate the human demonstration. The captured human motions were transformed into the motions of the HRP-2 so that they were consistent with the robot kinematics, as presented in Section 3.1. The trajectories of the demonstrations were rearranged into trajectories of 4 s (201 Frames at 50 Hz) to avoid overloading the humanoid robot. Prismatic workspaces (Case 1: $X \times Y : 30\,\mathrm{cm} \times 40\,\mathrm{cm}$, Case 2: $X \times Y \times Z : 20\,\mathrm{cm} \times 90\,\mathrm{cm} \times 50\,\mathrm{cm}$), as illustrated in Figures 3.6 and 3.7, were considered. The workspaces were divided into 7 regions in Case 1 and 42 regions in Case 2. The dimension of a region was $10\,\mathrm{cm} \times 10\,\mathrm{cm} \times 10\,\mathrm{cm}$. Regions in the workspace correspond to clusters in the database. The trajectories of the wrist position (x,y,z) and elbow elevation angle q_E were considered in these case studies. Condition sets (target wrist position and elbow elevation angle) were specified at the corners and median point of each region, as illustrated in Figures 3.6 and 3.7. The number of conditions sets initially specified in Case 1 and Case 2, was 18 and 116, respectively, except for points ('$*$' in Figures 3.6 and 3.7) that were unreachable because of the joint limits or arm length.



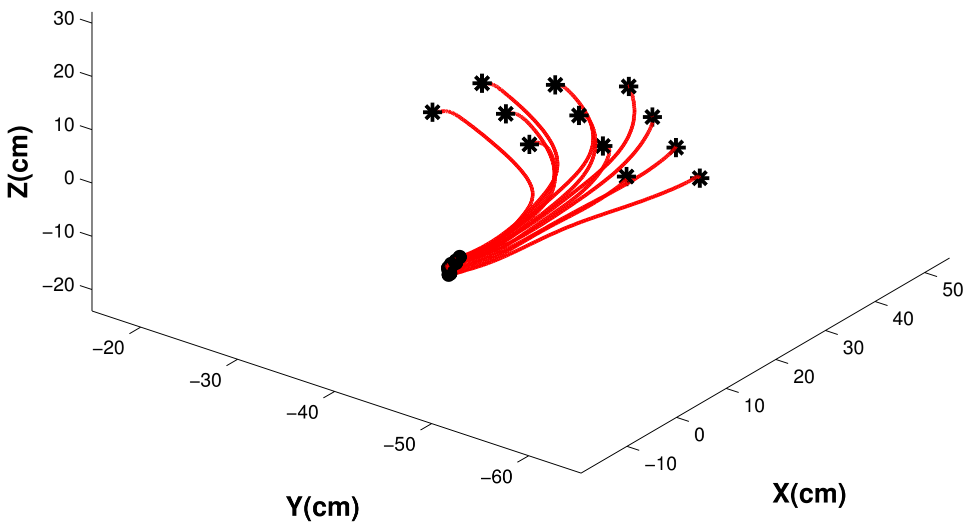
(a) 1^{st} Generation



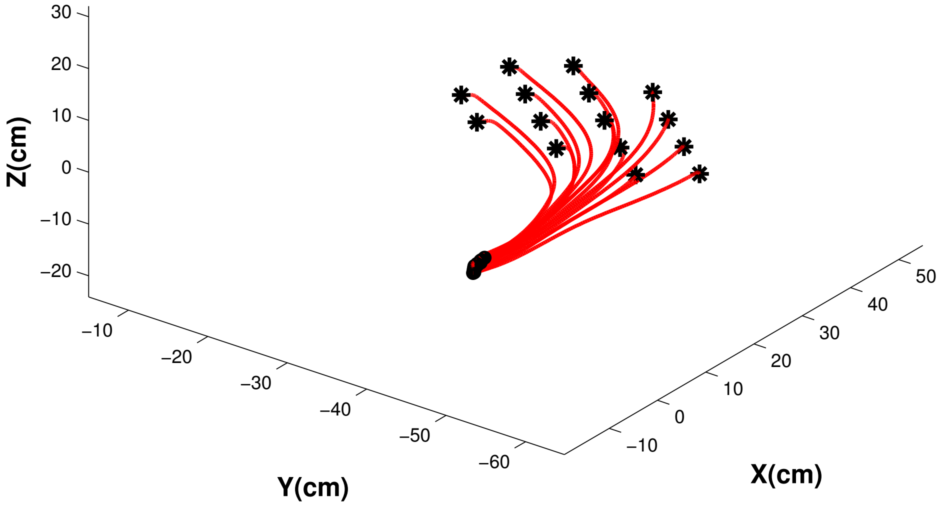
(b) Top view



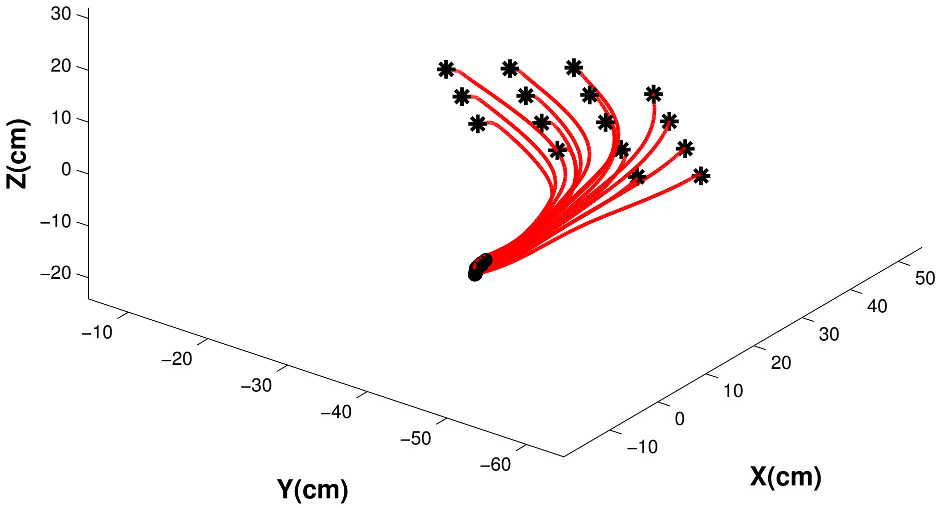
(c) Side view



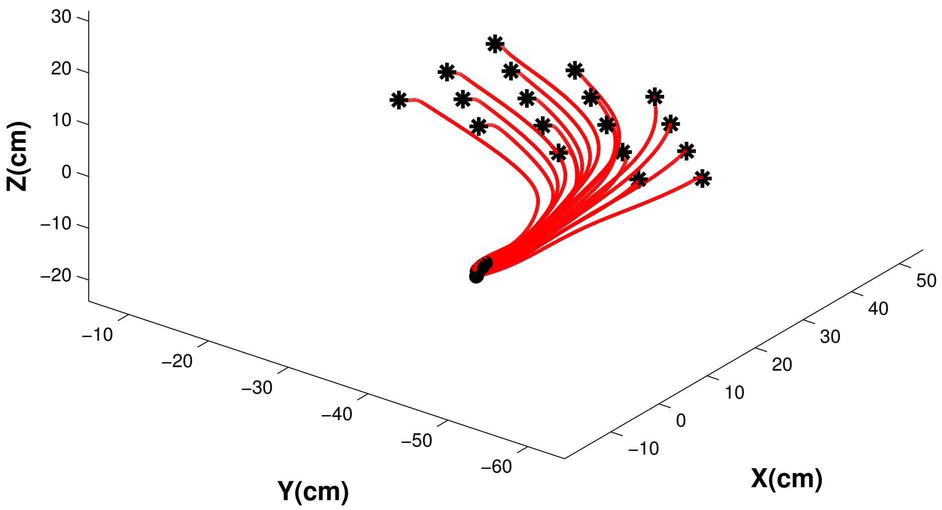
(d) 2^{nd} Generation



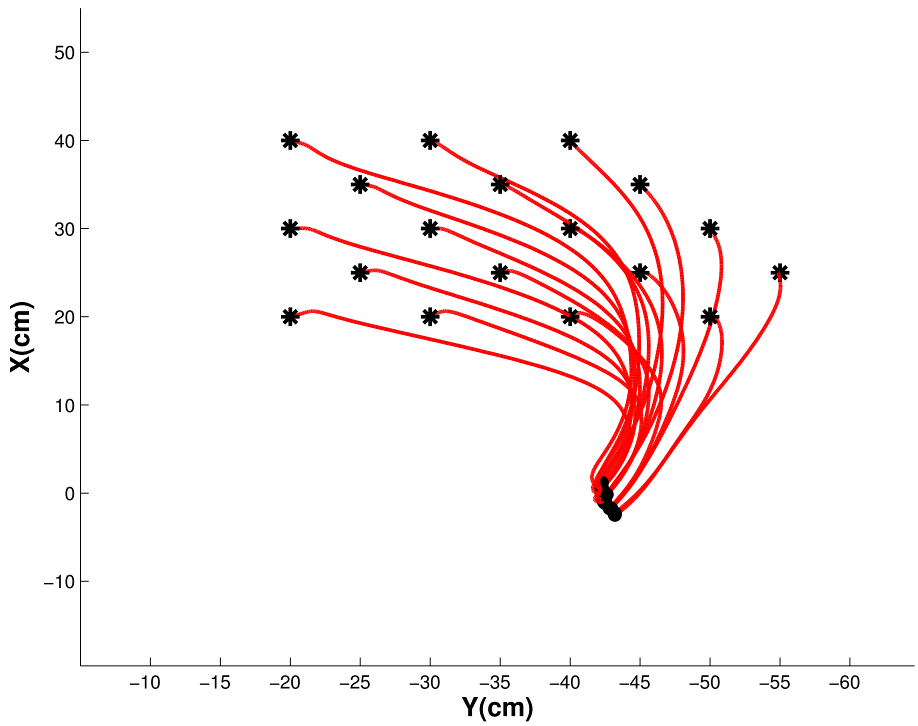
(e) 4^{th} Generation



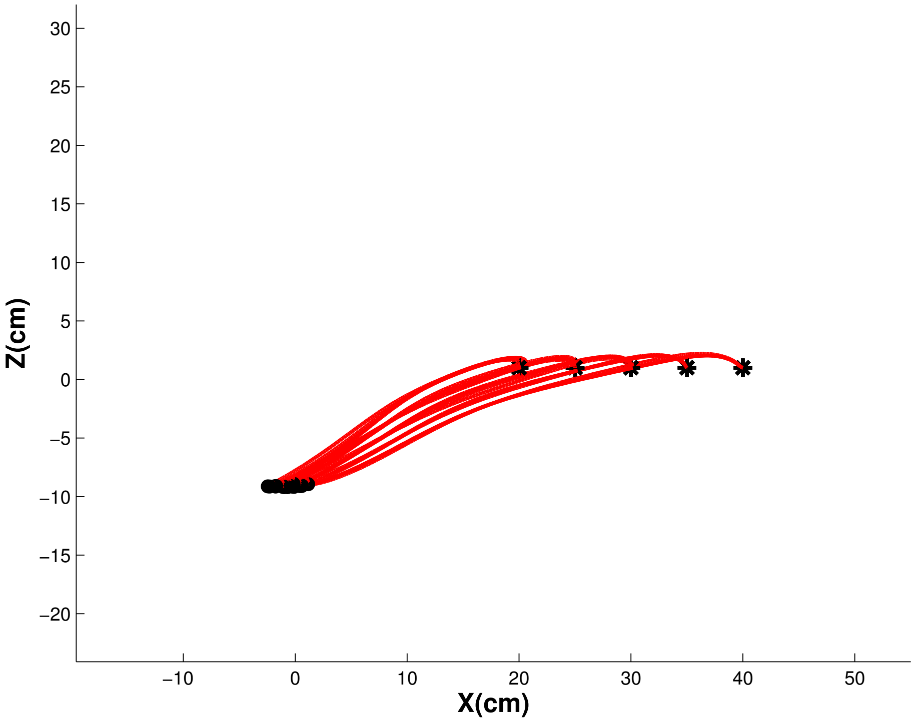
(f) 6^{th} Generation



(g) 8^{th} Generation

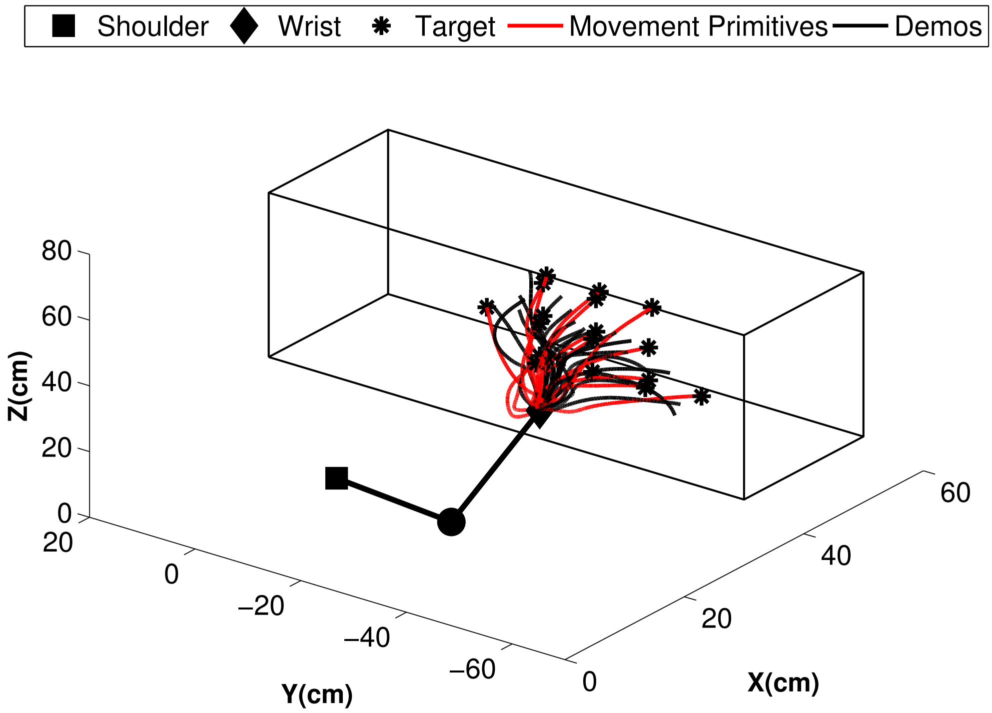


(h) Top view

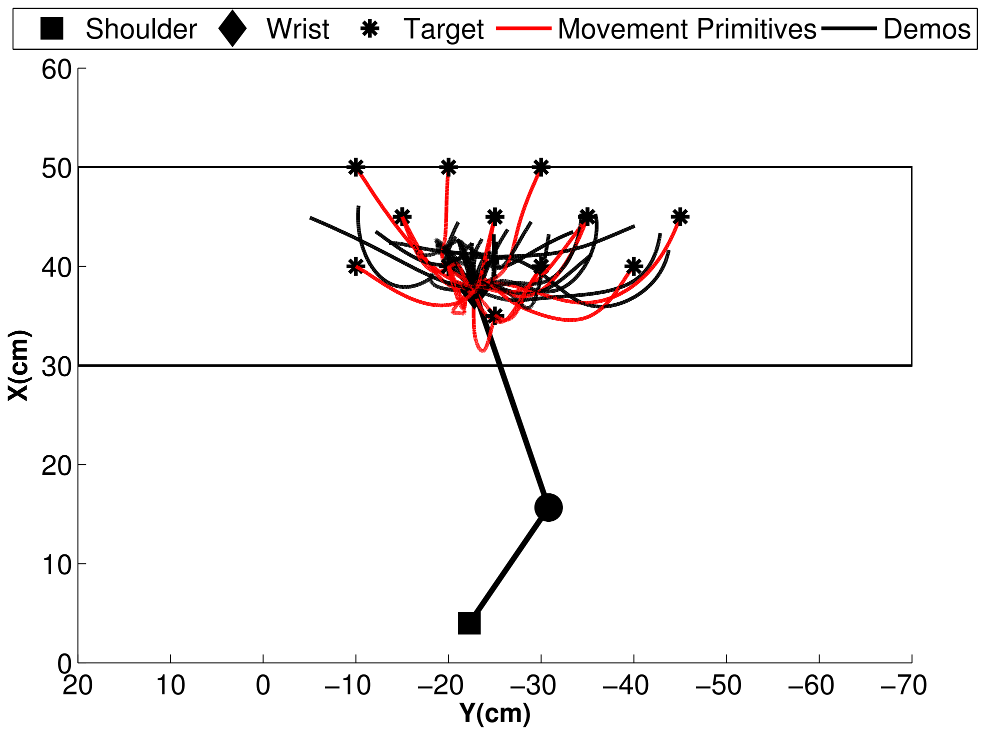


(i) Side view

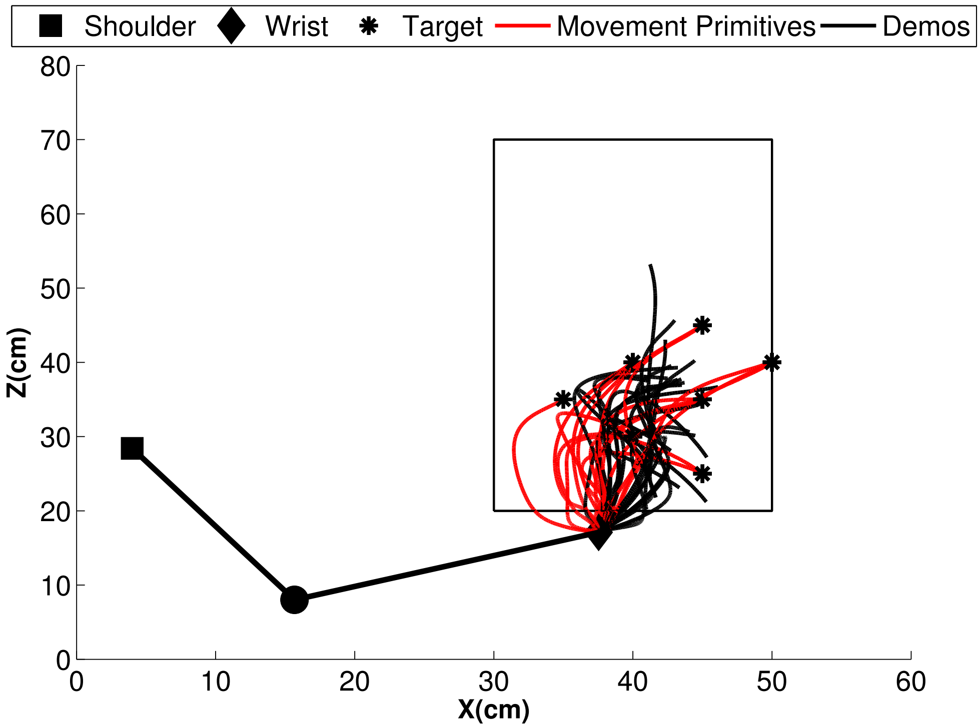
Figure 4.1: The movement primitives generated at each generation in Case 1. The gray box is a table.



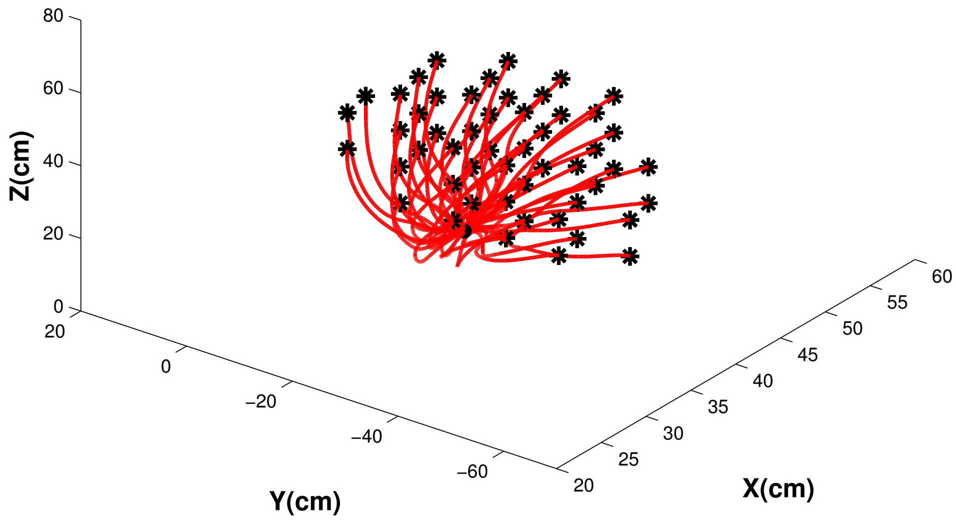
(a) 2^{nd} Generation



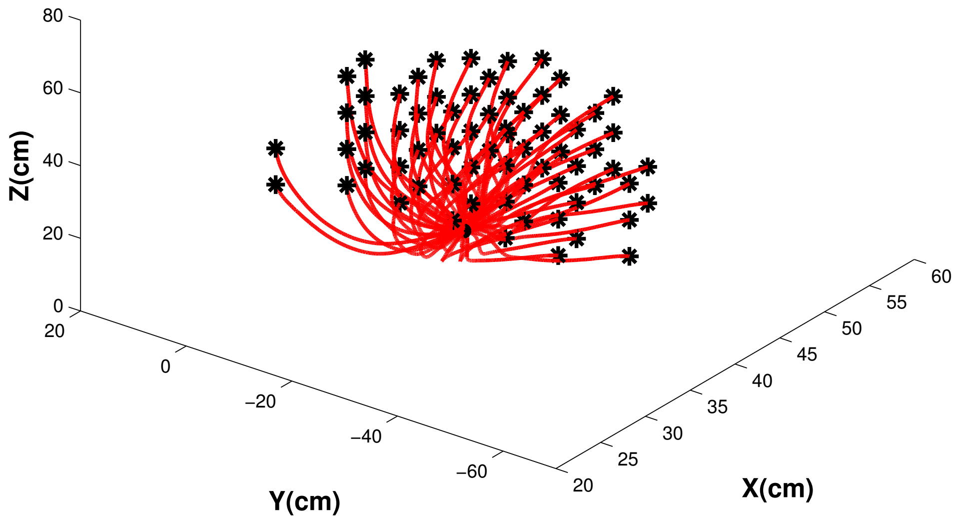
(b) Top view



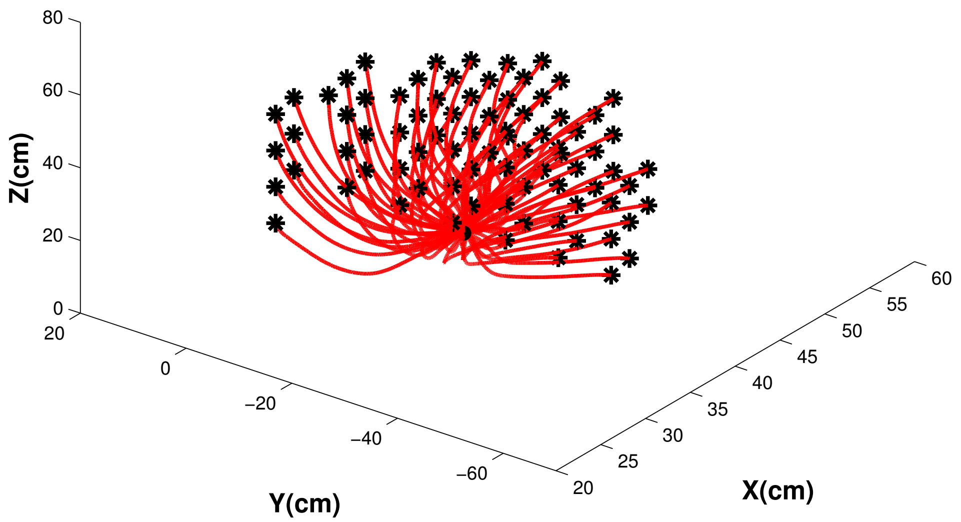
(c) Side view



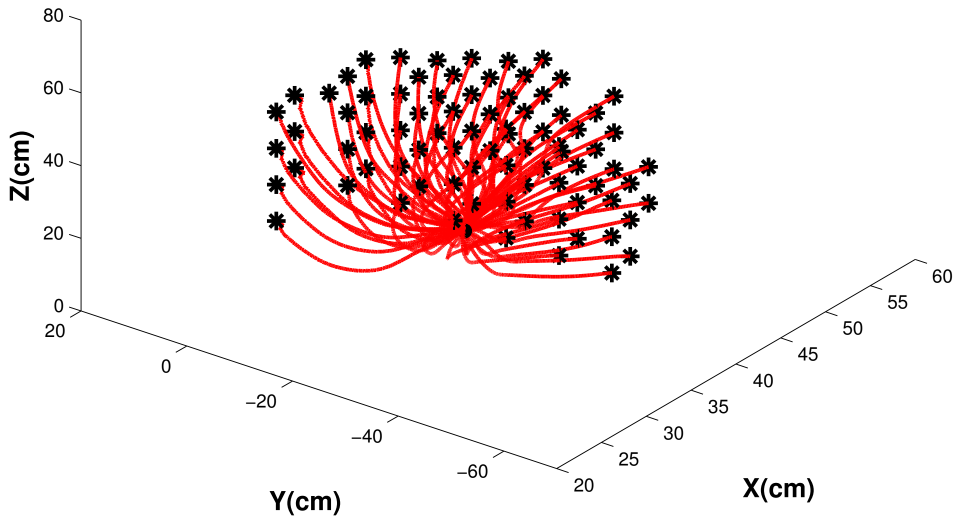
(d) 4^{th} Generation



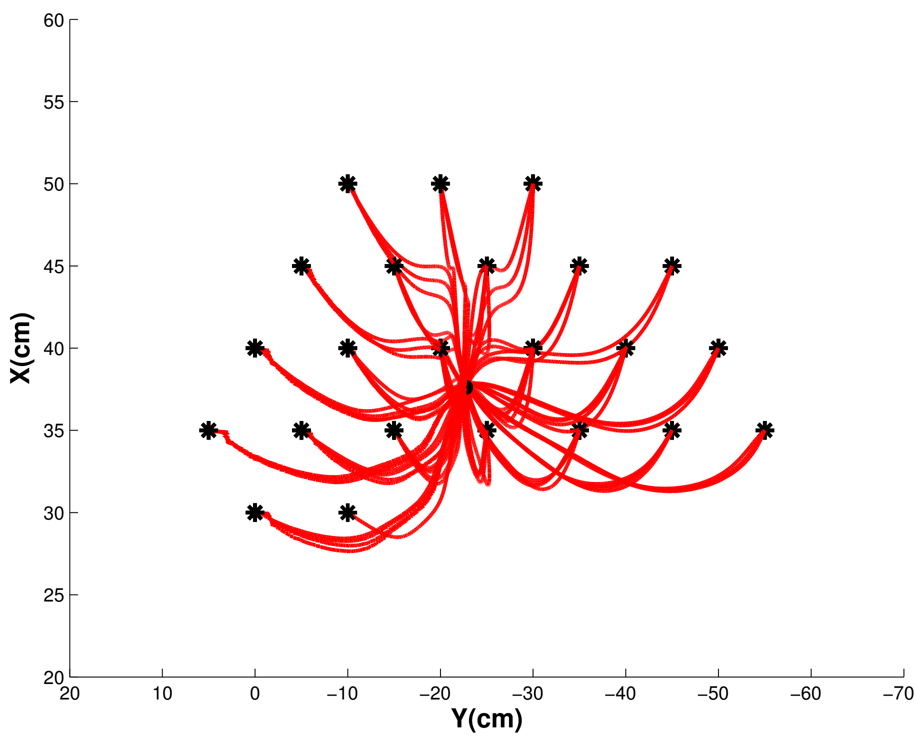
(e) 6^{th} Generation



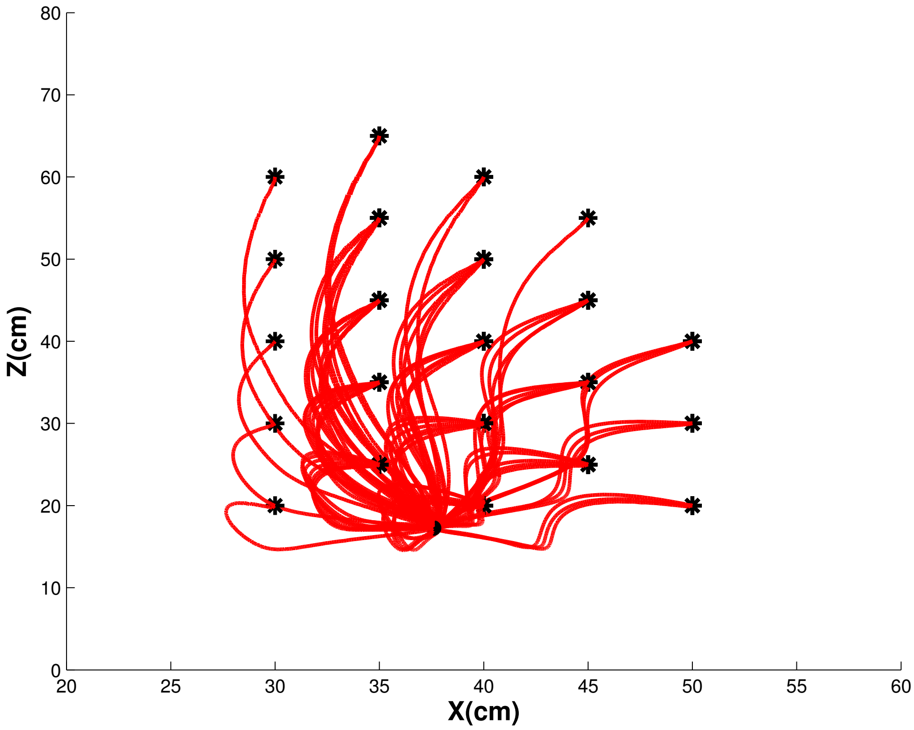
(f) 8^{th} Generation



(g) 11^{th} Generation

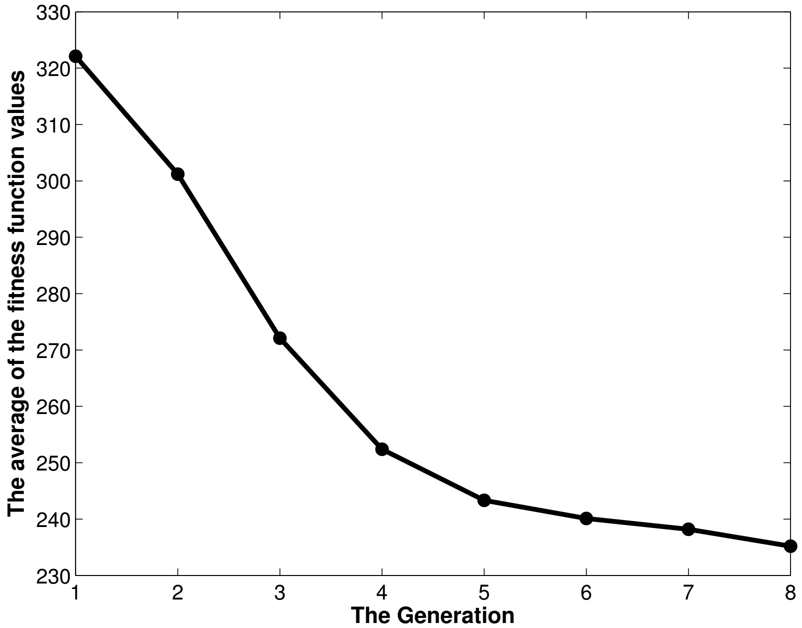


(h) Top view

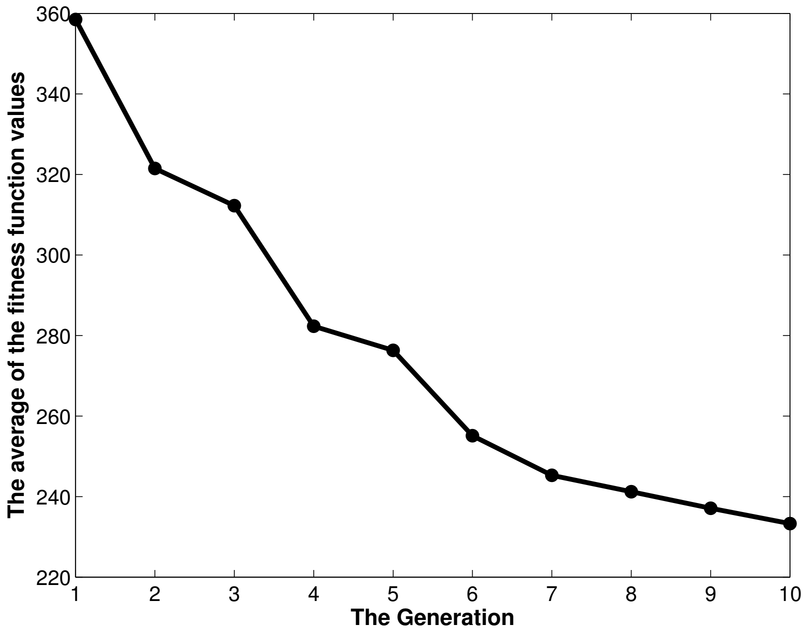


(i) Side view

Figure 4.2: Movement primitives generated at each generation in Case 2. The transparent box is a workspace.

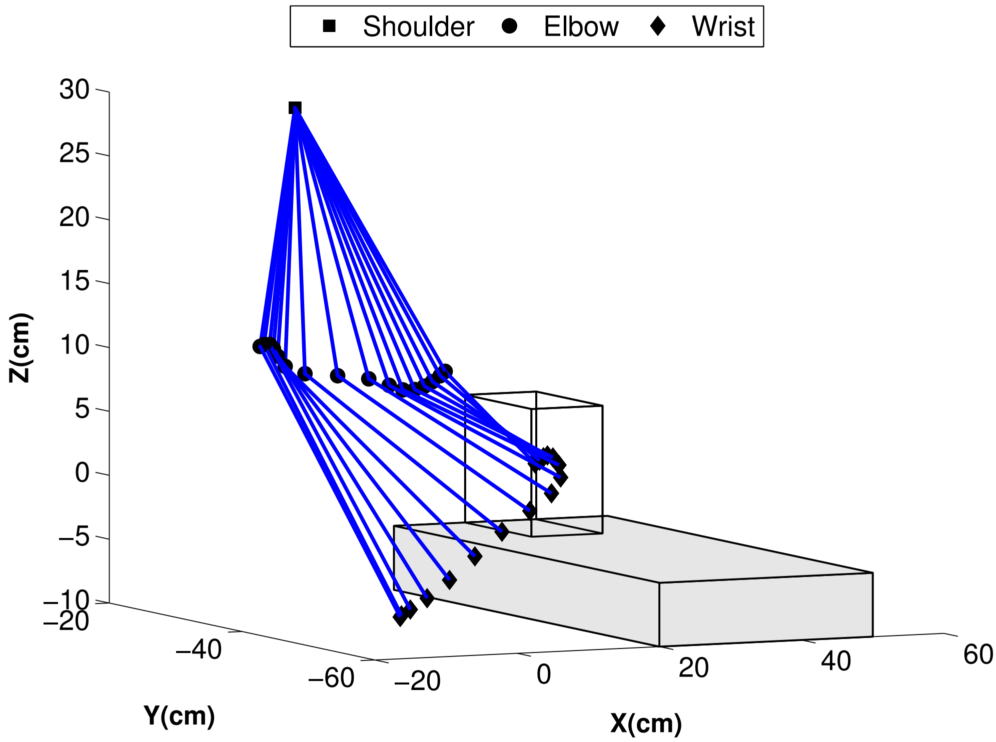


(a) case 1

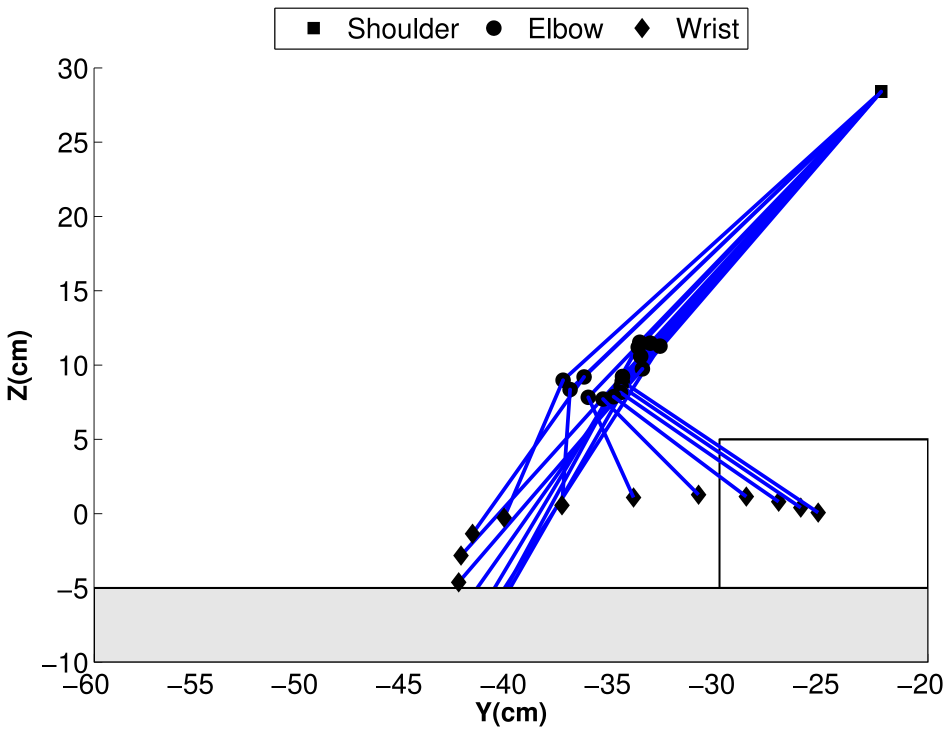


(b) case 2

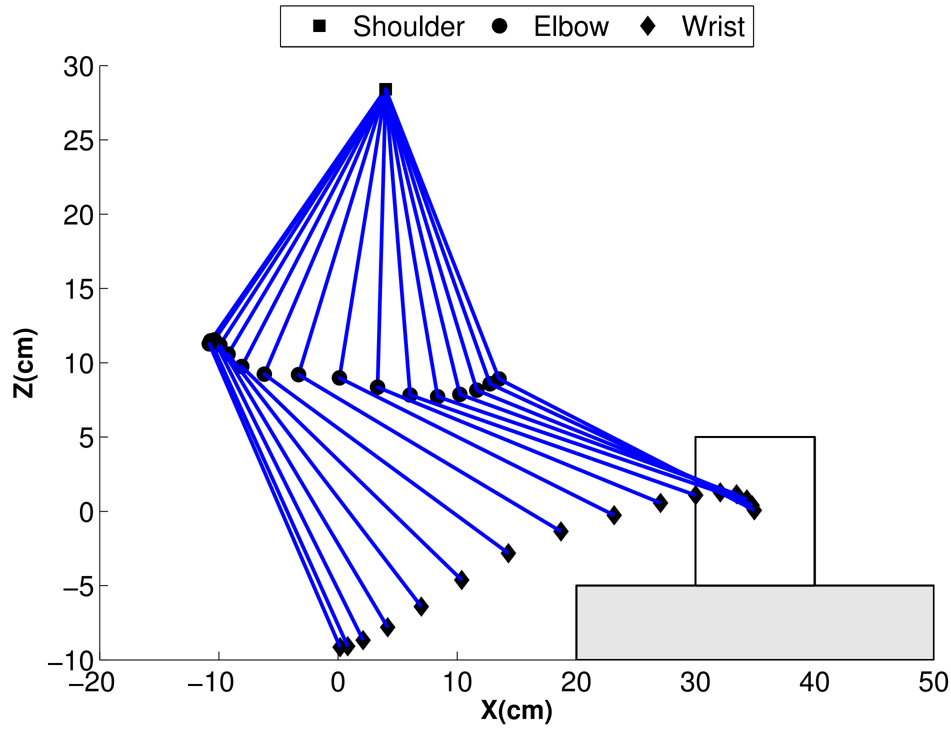
Figure 4.3: Average of the total torque summation at each generation



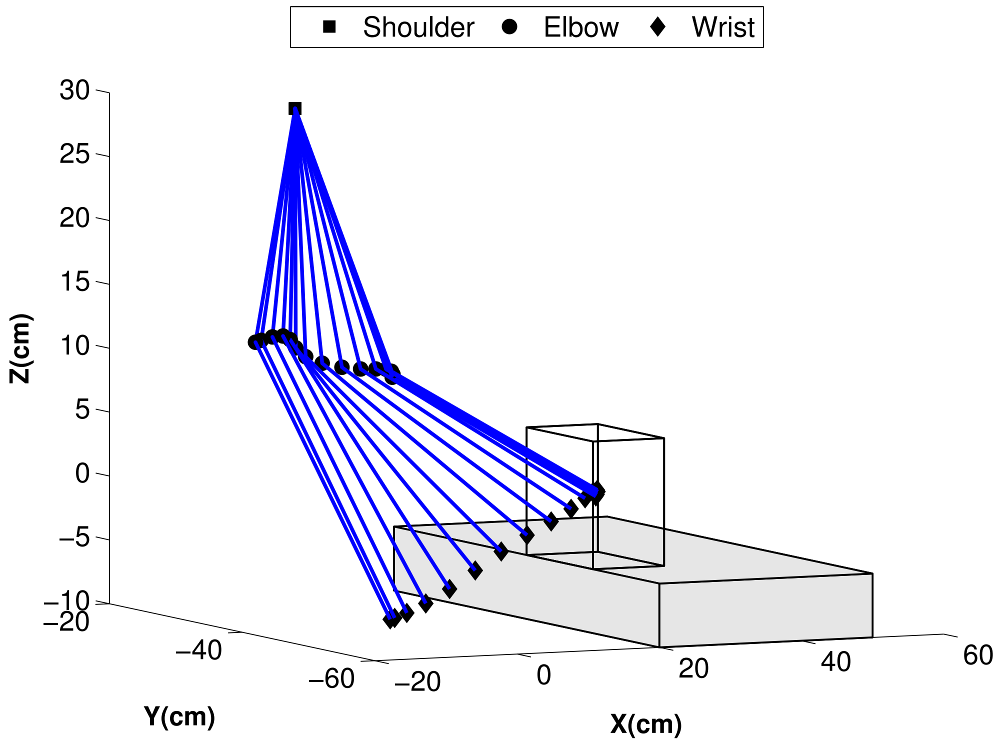
(a) Case 1: reaching motion to an arbitrary given target (example 1)



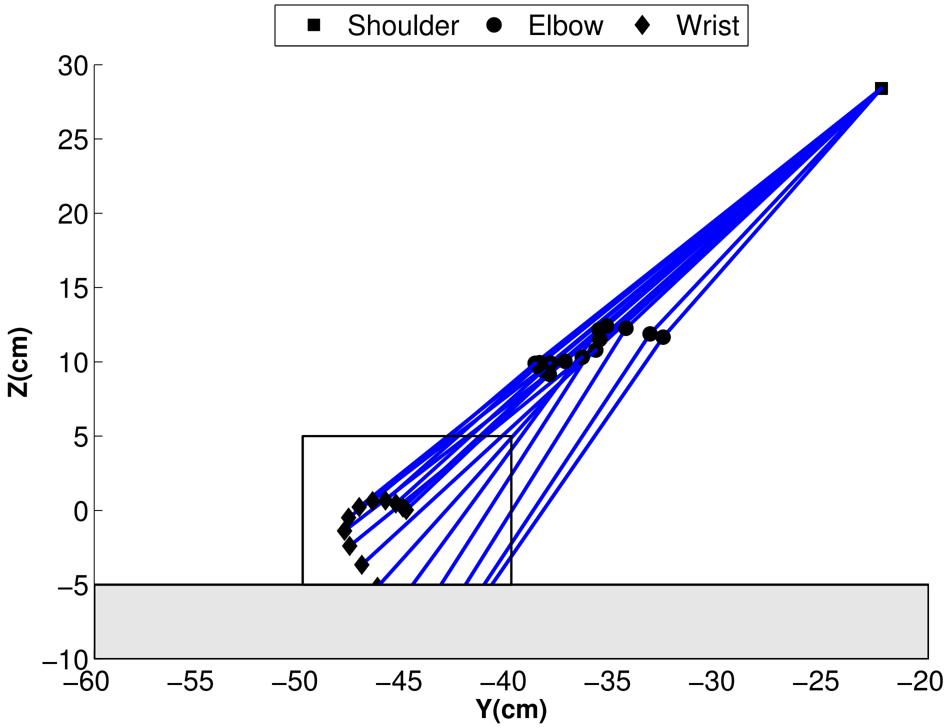
(b) Case 1: Front view (example 1)



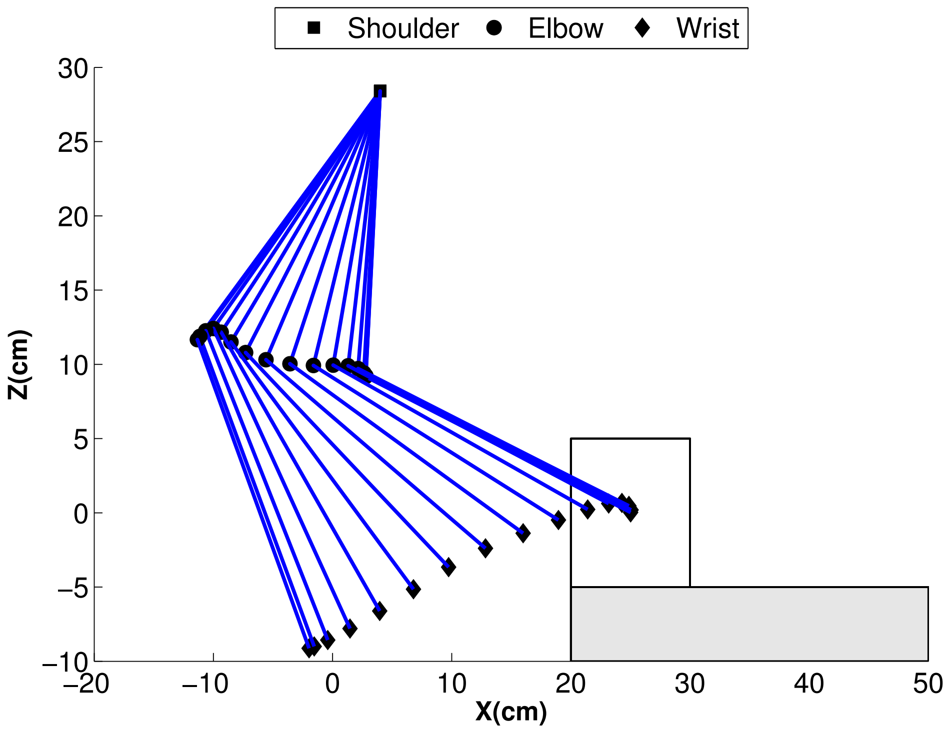
(c) Case 1: Side view (example 1)



(d) Case 1: reaching motion to an arbitrary given target (example 2)

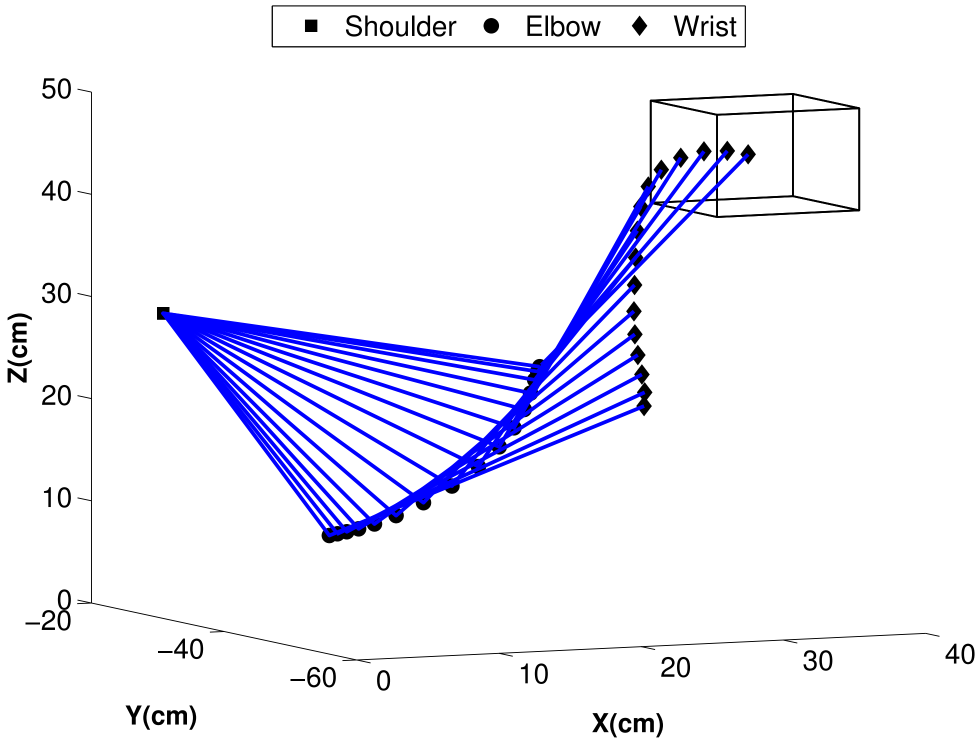


(e) Case 1: Front view (example 2)

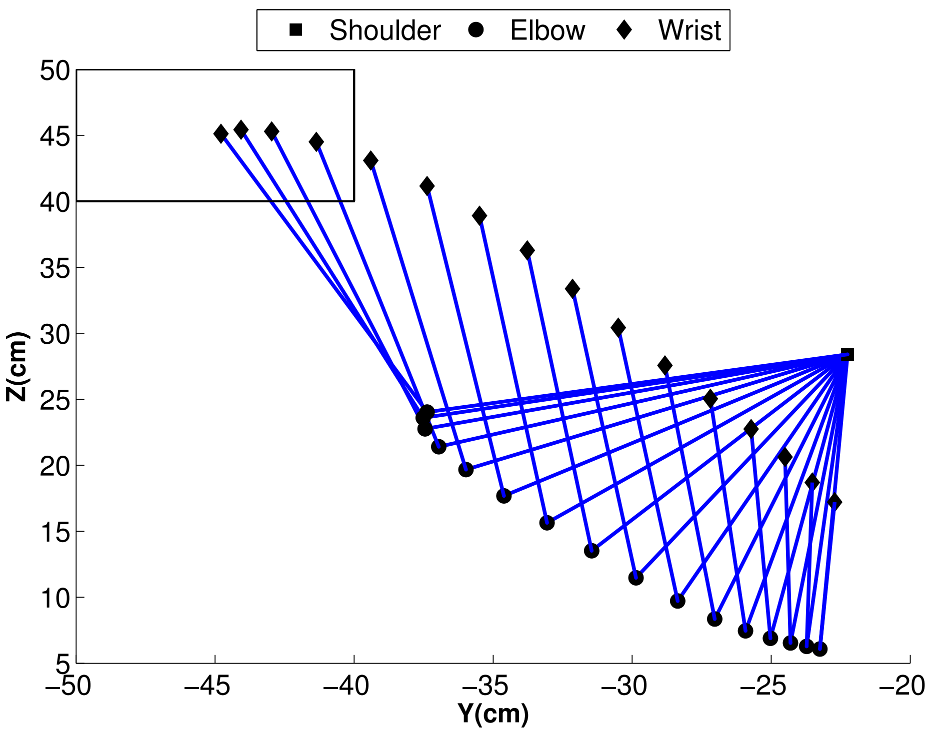


(f) Case 1: Side view (example 2)

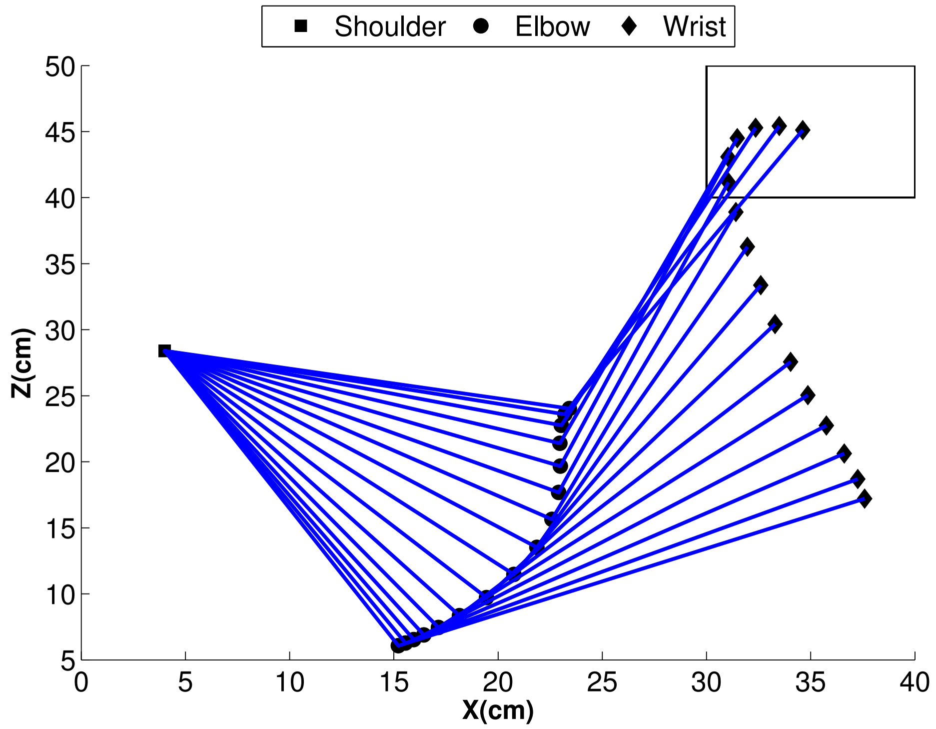
Figure 4.4: Generated reaching motions to arbitrarily given targets. The transparent box is the region corresponding to the cluster and the gray box is a table.



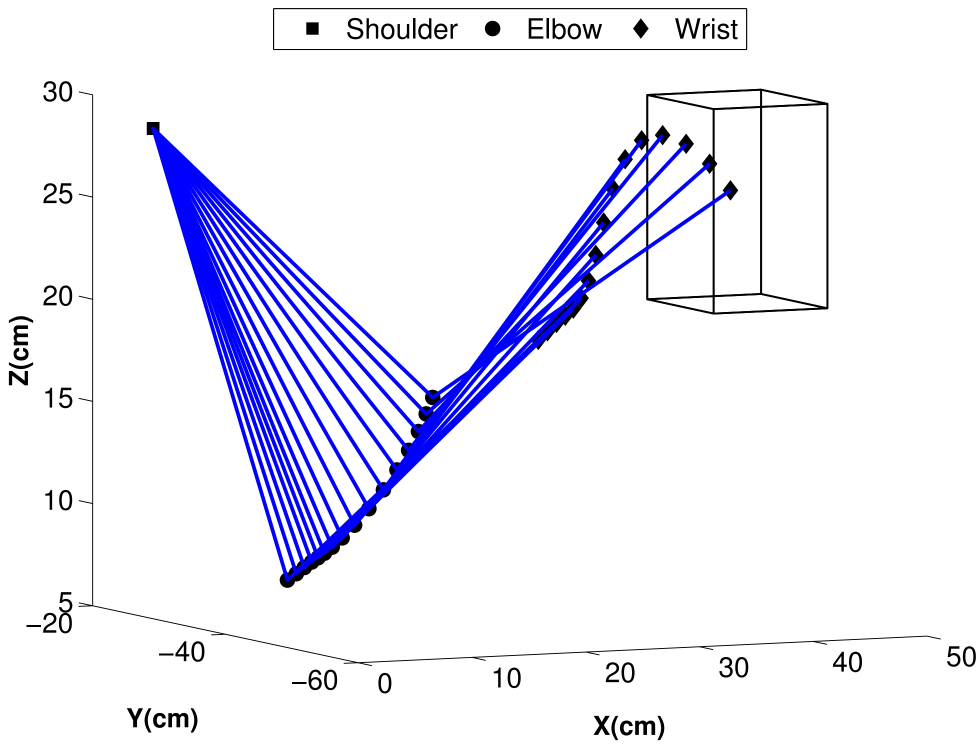
(a) Case 2: reaching motion to an arbitrary given target (example 3)



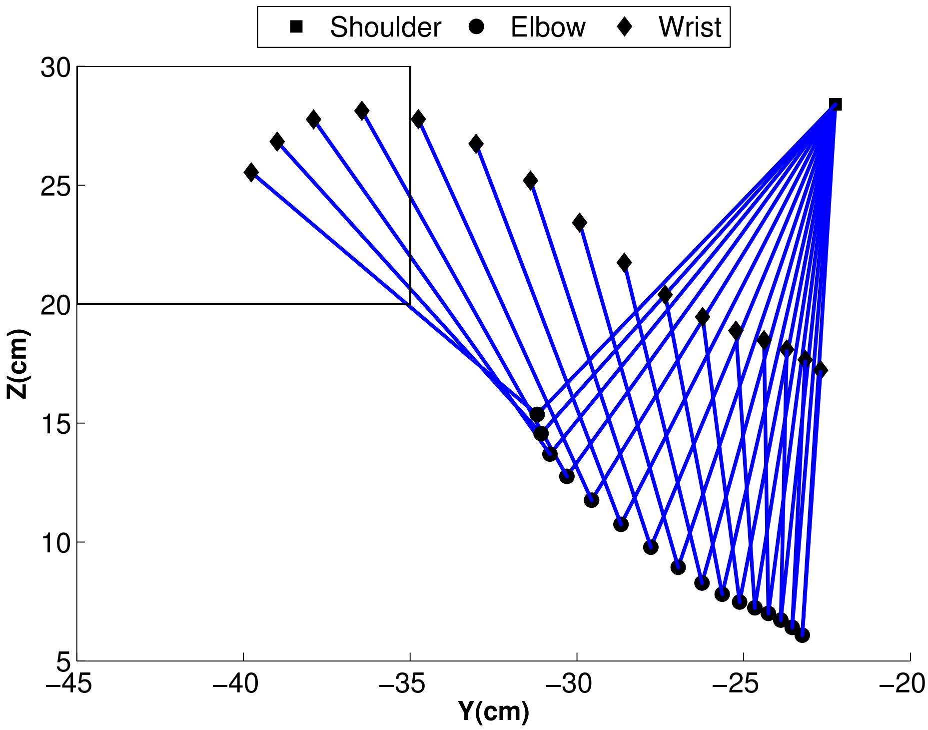
(b) Case 2: Front view (example 3)



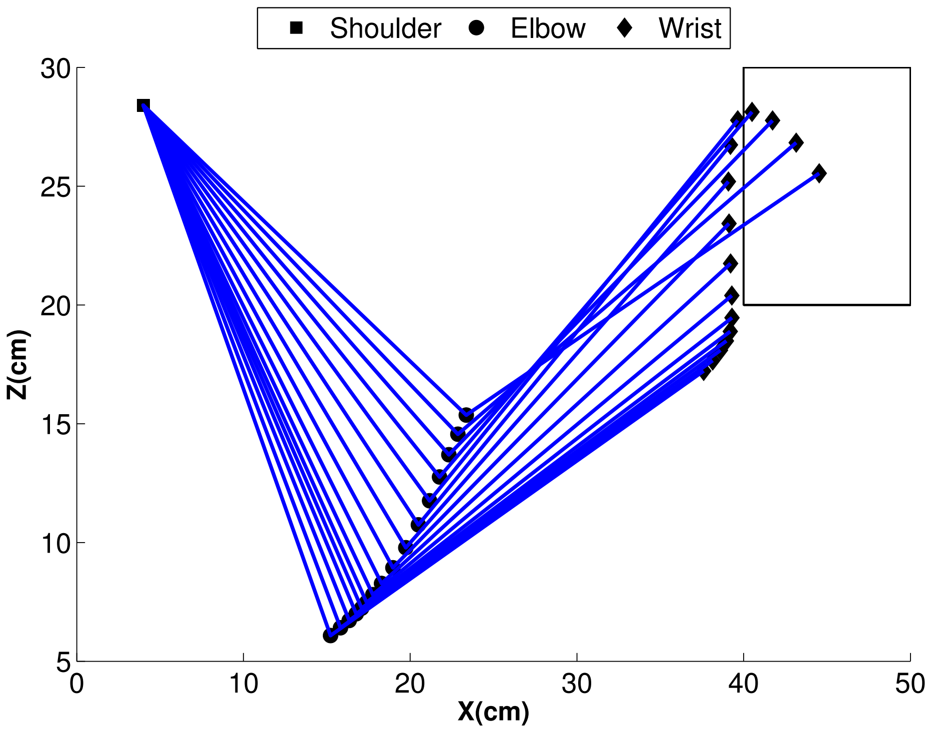
(c) Case 2: Side view (example 3)



(d) Case 2: reaching motion to an arbitrary given target (example 4)



(e) Case 2: Front view (example 4)



(f) Case 2: Side view (example 4)

Figure 4.5: Generated reaching motions to arbitrarily given targets. The transparent box is the region corresponding to the cluster.

4.2 Off-line Imitation Learning Process of Given Task

In the first generation, only 16 and 21 movement primitives were transformed from human demonstrations in Case 1 and 2, respectively. The condition sets that had more than two neighboring movement primitives within a 10 cm radius were searched. In the first generation, 8 condition sets in Case 1 and 17 condition sets in Case 2 were found. By using Eqs. (3.2) - (3.5), 8 offsprings in Case 1 (Figures 4.1 (a) - (c)) and 17 offsprings in Case 2 (Figures 4.2 (a) - (c)) were generated as parent candidates of the next generation. As shown in Figures 4.1 (d) - (g) and 4.2 (d) - (g), the movement primitives database was updated by the offsprings if they survived the evaluation using the fitness function (torque summation). As a result, the fully updated and evolved movement primitives database (Figures 4.1 (g) - (i) and 4.2 (g) - (i)) was obtained. The database had 34 movement primitives (18 condition sets + 16 demonstrations) in Case 1 and 137 movement primitives (116 condition sets + 21 demonstrations) in Case 2 for which movement primitives were evolved. From the evolved movement primitives, the principal components and mean trajectory were calculated on each cluster (7 clusters in Case 1 and 42 clusters in Case 2). Only the principal components and mean trajectory of the clusters were stored in the database.

Figures 4.3 (a) and (b) show that the averages of the torque summation during the motion decreased as the generations were completed. Hence, we can say that the motions were well evolved.

4.3 On-line Motion Generation by Robust Motion Reconstruction Method in Real Time

Since movement primitives were evolved in 7 clusters in Case 1 and 42 clusters in Case 2, the movement primitives database contained 7 and 42 sets of principal components and mean trajectory for Case 1 and 2, respectively.

Therefore, when an arbitrary target position was given for the robot to reach, the cluster to which the target position belonged was searched, and then, a reaching motion to the target position was generated in real time using the principal components and mean trajectory of the cluster (Eqs. (2.10), (2.11) and (2.12)). Figures 4.4 and 4.5 show the generated motions when the target positions were arbitrarily given.

The calculation time to generate a motion using MATLAB was 0.01 s.

Chapter 5. Conclusion

In this dissertation, an imitation learning framework based on PCA was proposed. The proposed framework solves the over-response (radical deviation from human demonstration) problem, which constituted a serious problem in the previous version [5, 6, 7] of the framework, by introducing L2 regularization [51]. Furthermore, the framework relaxes the demonstration requirements, which contributes to reducing the number of demonstrations and hence to improving efficiency in teaching.

The main feature of the proposed imitation learning framework is that an arbitrary motion is generated by a linear combination of principal components and mean trajectory defined in a cluster that includes the over-response solution. In the first iteration of an evolutionary process, motions are generated for specified points (condition sets) that are close to the demonstrated points. Then, the generated motions are also used to generate new motions in the next iteration. A generated motion for a target is iteratively updated if the newly generated motion shows better fitness than does the old motion. In the evolutionary process, the motion generation spreads toward the peripheral region of the workspace from the neighborhood of the demonstrated region. From the generated motions, principal components are computed on each cluster. When a target is arbitrarily given, a motion is generated on-line by a linear combination of the principal components and mean trajectory of the cluster to which the given target belongs. As aforementioned, the off-line imitation learning process has two main purposes. The first is to generate a motion for a peripheral point that is far from the demonstrated points. The second purpose is to consider a user-defined fitness function, such as energy consumption, joint limitations, or torque summation, as well as to imitate the human demonstrations.

In the case study, the performance of the proposed imitation learning framework was validated.

Appendix A Minimization of errors of arm and hand postures [1]

In order to transform the captured human motion to humanoid robot motion, an adaptation process [1] is adopted so that the differences in the arm and hand postures between human motion and robot motion are minimized. The method overcomes the difference in DOF between a human and a humanoid robot. First, a scaled model is designed using the captured human model. A general optimization scheme is applied to the model with limits on the capacities of the joint motors such that it can provide a solution procedure for inverse kinematics problems. The error function to be minimized is given as

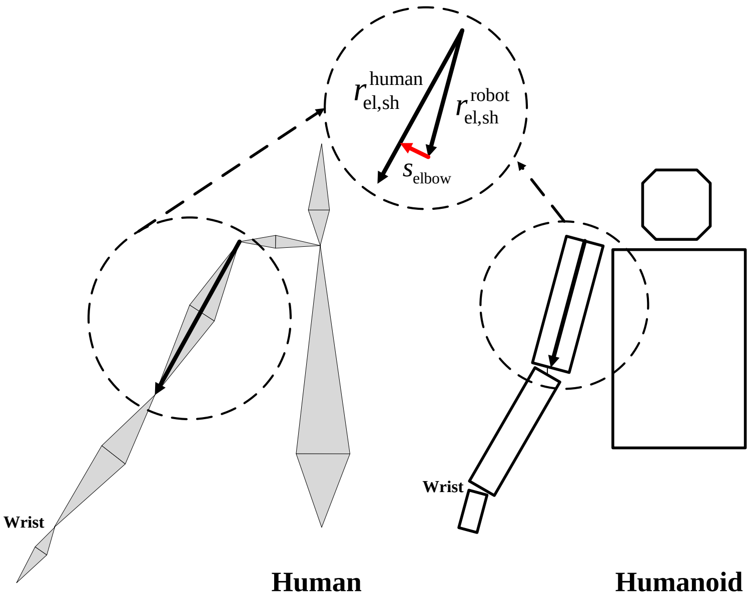


Figure A.1: Definition of $\boldsymbol{s}_{\text{elbow}}$

$$\min_{\Delta \boldsymbol{q}(t)} f(\Delta q(t)) = \mathbf{E}_{\text{hand}}(\Delta \boldsymbol{q}(t))^T \mathbf{W} \mathbf{E}_{\text{hand}}(\Delta \boldsymbol{q}(t)) + W_{10} \|\boldsymbol{s}_{\text{elbow}}(\Delta \boldsymbol{q}(t))\|^2 \tag{A.1}$$

$$\boldsymbol{E}_{\text{hand}}(\Delta \boldsymbol{q}(t)) = \begin{pmatrix} \boldsymbol{r}^{\text{human}}_{\text{wrist}}(t) - \boldsymbol{r}^{\text{robot}}_{\text{wrist}}(\Delta \boldsymbol{q}(t)) \\ \boldsymbol{r}^{\text{human}}_{\text{thumb}}(t) - \boldsymbol{r}^{\text{robot}}_{\text{thumb}}(\Delta \boldsymbol{q}(t)) \\ \boldsymbol{r}^{\text{human}}_{\text{pinky}}(t) - \boldsymbol{r}^{\text{robot}}_{\text{pinky}}(\Delta \boldsymbol{q}(t)) \end{pmatrix}$$
$$\mathbf{W} = \text{diag}[W_1 \ W_2 \ ... \ W_9] \tag{A.2}$$
$$\Delta \boldsymbol{q} = [\Delta q_1 \ ... \ \Delta q_7]^T$$

where W_{10} is a weight for the second term in Eq. (A.1). $\boldsymbol{r}^{\text{human}}_{\text{wrist}}(t)$, $\boldsymbol{r}^{\text{human}}_{\text{thumb}}(t)$, and $\boldsymbol{r}^{\text{human}}_{\text{pinky}}(t)$ are the captured trajectories of the 3D position vectors of three markers located on the wrist, thumb, and pinky of the hand, respectively. $\boldsymbol{r}^{\text{robot}}_{\text{wrist}}(\Delta \boldsymbol{q}(t))$, $\boldsymbol{r}^{\text{robot}}_{\text{thumb}}(\Delta \boldsymbol{q}(t))$, and $\boldsymbol{r}^{\text{robot}}_{\text{pinky}}(\Delta \boldsymbol{q}(t))$ are the position vectors on the humanoid robot corresponding to the three points of a human hand.

$\|\boldsymbol{s}_{\text{elbow}}(\Delta \mathbf{q}(t))\|$ is the magnitude of the vector representing the orientation difference between the upper arm of the human and a humanoid robot (Figure A.1):

$$\|\boldsymbol{s}_{\text{elbow}}(\Delta \boldsymbol{q}(t))\|^2 = \left(\frac{\boldsymbol{r}^{\text{human}}_{\text{el,sh}}(t)^T}{\|\boldsymbol{r}^{\text{human}}_{\text{el,sh}}(t)\|} \boldsymbol{r}^{\text{robot}}_{\text{el,sh}}(\Delta \boldsymbol{q}(t)) \right)^2 - \boldsymbol{r}^{\text{robot}}_{\text{el,sh}}(\Delta \boldsymbol{q}(t))^T \boldsymbol{r}^{\text{robot}}_{\text{el,sh}}(\Delta \boldsymbol{q}(t)) \tag{A.3}$$

where

$$\boldsymbol{r}^{\text{human}}_{\text{el,sh}}(t) = \boldsymbol{r}^{\text{human}}_{\text{el}}(t) - \boldsymbol{r}^{\text{human}}_{\text{sh}}(t), \ \ \boldsymbol{r}^{\text{robot}}_{\text{el,sh}}(\Delta \boldsymbol{q}(t)) = \boldsymbol{r}^{\text{robot}}_{\text{el}}(\Delta \boldsymbol{q}(t)) - \boldsymbol{r}^{\text{robot}}_{\text{sh}}(\Delta \boldsymbol{q}(t)) \tag{A.4}$$

$\boldsymbol{r}^{\text{human}}_{\text{el}}(t)$ and $\boldsymbol{r}^{\text{human}}_{\text{sh}}(t)$ are the position vectors of the elbow and shoulder of the human, respectively, at time t, according to captured marker data. $\boldsymbol{r}^{\text{robot}}_{\text{el}}(\Delta \boldsymbol{q}(t))$ and $\boldsymbol{r}^{\text{robot}}_{\text{sh}}(\Delta \boldsymbol{q}(t))$ are the position vectors of the elbow and shoulder of a humanoid robot, respectively.

Appendix B General Solution of Geometric Inverse Kinematics Problem

In Figure B.1, where L_u and L_l are a length of upper and lower, R is a length of between the wrist and shoulder for a humanoid robot. Let $q_i (i = 0, 1, 2, \ldots, 6)$ represent the joint angles of arm(shoulder pitch, roll, yaw, elbow pitch, yaw, wrist pitch, roll, yaw) for a humanoid robot and a elbow elevator angle (EEA), a hand and wrist position are known for a target. The q_0, q_1 and q_3 are obtained as :

$$q_3 = \pi - \cos^{-1}\left(\frac{L_u^2 + L_l^2 - R^2}{2L_uL_l}\right) \tag{B.1}$$

$$q_0 = \begin{cases} \cos^{-1}\left(\frac{-E_z}{\sqrt{E_y^2+E_z^2}}\right) & \text{if } E_z \leq 0, E_x \geq 0 \\ -\cos^{-1}\left(\frac{-E_z}{\sqrt{E_y^2+E_z^2}}\right) & \text{else if } E_z \leq 0, E_x \leq 0 \\ \cos^{-1}\left(\frac{E_x}{\sqrt{E_y^2+E_z^2}}\right) + \frac{\pi}{2} & \text{else} \end{cases} \tag{B.2}$$

$$q_1 = \sin^{-1}\left(\frac{E_y}{L_u}\right) \tag{B.3}$$

q_2 is easly obtained by the foward kinematics($\mathbf{q} = [q_0, q_1, 0, q_3, 0, 0, 0]$) of arm. q_4, q_5 and q_6 are determined by the inverse kinematics solution (q_0, q_1, q_2, q_3, hand position and orientation) according to an object .

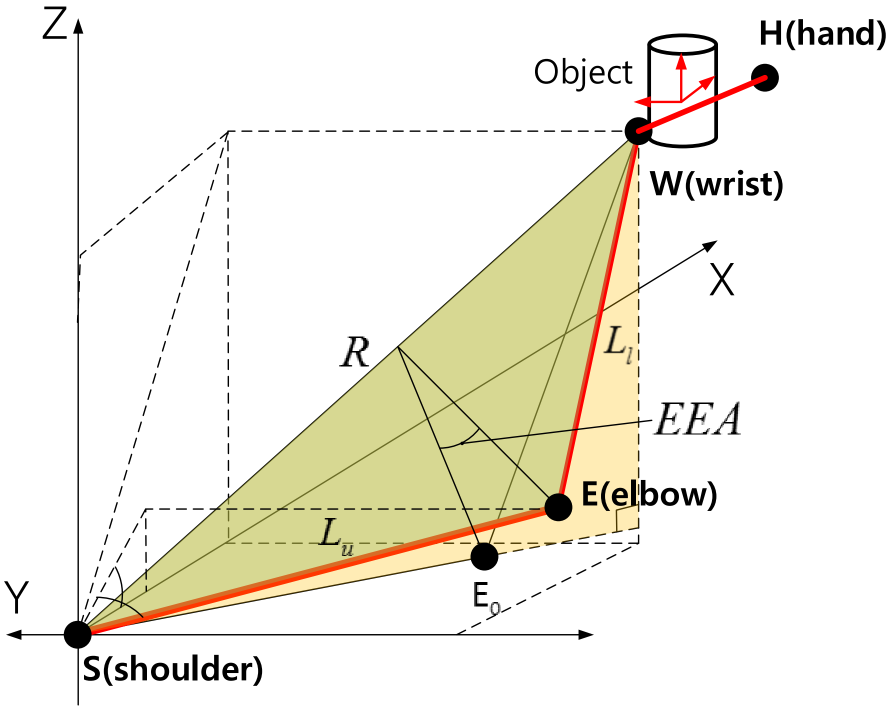


Figure B.1: The definition of elbow elevation angle and Inverse kinematics problem

References

[1] Y. O. ChangHwan Kim, Doik Kim, "Adaptation of human motion capture data to humanoid robots for motion imitation using optimization," *Integrated Computer-Aided Engineering*, vol. 13 issues 4, pp. 377–389, 2006.

[2] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233 – 242, 1999.

[3] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg, 2008, pp. 1371–1394.

[4] C. L. Nehaniv and K. Dautenhahn, "Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications," in *Interdisciplinary Approaches to Robot Learning*. World Scientific Press, 1999, pp. 136–161.

[5] G.-R. Park, S. Ra, C. Kim, and J. Song, "Imitation learning of robot movement using evolutionary algorithm," in *Proceedings of the 17th World Congress The International Federation of Automatic Control*, 2008, pp. 730–735.

[6] G.-R. Park, K. Kim, C. H. Kim, M.-H. Jeong, B.-J. You, and S. Ra, "Human-like catching motion of humanoid using evolutionary algorithm(ea)-based imitation learning," in *The 18th IEEE International Symposium on Robot and Human Interactive Communication*, 2009, pp. 809–815.

[7] G.-R. Park and C. H. Kim, "Constructing of optimal database structure by imitation learning based on evolutionary algorithm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 2698–2703.

[8] A. Fod, M. Mataric, and O. Jenkins, "Automated derivation of primitives for movement classification," *Autonomous Robots*, vol. 12, no. 1, pp. 39–54, 2002.

[9] T. D. Sanger, "Human arm movements described by a low-dimensional superposition of principal components," *The Journal of Neuroscience*, vol. 20, no. 3, pp. 1066–1072, 2000.

[10] B. Lim, S. Ra, and F. Park, "Movement primitives, principal component analysis, and the efficient generation of natural motions," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, April 2005, pp. 4630–4635.

[11] T. Kang, Y. Shimansky, and J. He, "Angle of elbow elevation depends on the reach target coordinates," in *Proceedings of the Second Joint Engineering in Medicine and Biology, 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference*, vol. 3, Oct 2002, pp. 2571–2572.

[12] S. Kim, C. H. Kim, and J. H. Park, "Human-like arm motion generation for humanoid robots using motion capture database," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006, pp. 3486–3491.

[13] R. Schmidt and T. Lee, *Motor Control and Learning: A Behavioral Emphasis.* Human Kinetics, 1999.

[14] ——, *Motor Control and Learning, 5E.* Human Kinetics, 2011.

[15] F. A. Mussa-Ivaldi, S. F. Giszter, and E. Bizzi, "Linear combinations of primitives in vertebrate motor control," *Proceedings of the National Academy of Sciences*, vol. 91, no. 16, pp. 7534–7538, 1994.

[16] M. Kawato, K. Furukawa, and R. Suzuki, "A hierarchical neural-network model for control and learning of voluntary movement," *Biological Cybernetics*, vol. 57, no. 3, pp. 169–185, 1987.

[17] J. Piaget, *Play, Dreams and imitation in Childhood*, Norton, New York, 1962.

[18] D. A. Goslin, *Handbook of socialization theory and research.* Chicago, Rand McNally, 1969.

[19] J. Hollerbach and T. Flash, "Dynamic interactions between limb segments during planar arm movement," *Biological Cybernetics*, vol. 44, no. 1, pp. 67–77, 1982.

[20] M. Raibert, "A model for sensorimotor control and learning," *Biological Cybernetics*, vol. 29, no. 1, pp. 29–36, 1978.

[21] F. A. Mussa-Ivaldi and E. Bizzi, "Motor learning through the combination of primitive," *Philosophical Trans. of the Royal Society of London*, vol. Series B, no. 355, pp. 1755–1769, 2000.

[22] M. M. Williamson, "Postural primitives: Interactive behavior for a humanoid robot arm," in *In Fourth International Conference on Simulation of Adaptive Behavior.* MIT Press, 1996, pp. 124–131.

[23] B. Dufay and J.-C. Latombe, "An approach to automatic robot programming based on inductive learning," *The International Journal of Robotics Research*, vol. 3, no. 4, pp. 3–20, 1984.

[24] A. Segre and G. DeJong, "Explanation-based manipulator learning: Acquisition of planning ability through observation," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, Mar 1985, pp. 555–560.

[25] A. Levas and M. Selfridge, "A user-friendly high-level robot teaching system," in *Robotics and Automation. Proceedings. 1984 IEEE International Conference on*, vol. 1, Mar 1984, pp. 413–416.

[26] G. Hovland, P. Sikka, and B. McCarragher, "Skill acquisition from human demonstration using a hidden markov model," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, Apr 1996, pp. 2706–2711.

[27] J. Yang, Y. Xu, and C. S. Chen, "Hidden markov model approach to skill learning and its application to telerobotics," *Robotics and Automation, IEEE Transactions on*, vol. 10, no. 5, pp. 621–631, Oct 1994.

[28] S. Tso and K. Liu, "Hidden markov model for intelligent extraction of robot trajectory command from demonstrated trajectories," in *Industrial Technology, 1996. (ICIT '96), Proceedings of The IEEE International Conference on*, Dec 1996, pp. 294–298.

[29] A. Alissandrakis, C. L. Nehaniv, and K. Dautenhahn, "Correspondence mapping induced state and action metrics for robotic imitation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B : Cybernetics, Special*, vol. 37, no. 2, pp. 299–307, 2007.

[30] A. Ude, "Trajectory generation from noisy positions of object features for teaching robot paths," *Robotics and Autonomous Systems*, vol. 11, no. 2, pp. 113 – 127, 1993.

[31] J. Yang, Y. Xu, and C. S. Chen, "Human action learning via hidden markov model," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 27, no. 1, pp. 34–44, Jan 1997.

[32] K. Yamane and Y. Nakamura, "Dynamics filter - concept and implementation of online motion generator for human figures," *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 3, pp. 421–432, June 2003.

[33] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *Systems, Man, and Cybernetics, Part B: IEEE Transactions on Cybernetics*, vol. 37, no. 2, pp. 286–298, 2007.

[34] A. Ude, "Trajectory generation from noisy positions of object features for teaching robot paths," *Robotics and Autonomous Systems*, vol. 11, no. 2, pp. 113–127, 1993.

[35] A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings of the 2002 IEEE*

International Conference on Robotics and Automation (ICRA), vol. 2, 2002, pp. 1398–1403.

[36] S. Calinon, F. Guenter, and A. Billard, "Goal-directed imitation in a humanoid robot," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 299–304.

[37] D. Lee and Y. Nakamura, "Mimesis scheme using a monocular vision system on a humanoid robot," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 2162–2168.

[38] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning control policies for movement imitation and movement recognition," *In Neural Information Processing System (NIPS)*, vol. 15, pp. 1547–1554, 2003.

[39] S. Khansari-Zadeh and A. Billard, "Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 2676–2683.

[40] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Dynamical system modulation for robot learning via kinesthetic demonstrations," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1463–1467, Dec 2008.

[41] S. Schaal, C. Atkeson, and S. Vijayakumar, "Scalable techniques from nonparametric statistics for real time robot learning," *Applied Intelligence*, vol. 17, no. 1, pp. 49–60, 2002.

[42] C. G. Atkeson, "Advances in neural information processing systems 2," D. S. Touretzky, Ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990, ch. Using Local Models to Control Movement, pp. 316–323.

[43] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," *Artificial Intelligence Review*, vol. 11, no. 1-5, pp. 11–73, Feb. 1997.

[44] A. Moore, "Fast, robust adaptive control by learning only forward models," in *Advances in Neural Information Processing Systems*, J. E. Moody, S. J.

Hanson, and R. P. L, Eds. 340 Pine Street, 6th Fl., San Francisco, CA 94104: Morgan Kaufmann, April 1992.

[45] S. Calinon, Z. Li, T. Alizadeh, N. Tsagarakis, and D. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," in *12th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2012, pp. 323–329.

[46] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2009, pp. 763–768.

[47] S. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, Oct 2011.

[48] S. Calinon, F. D'halluin, E. Sauser, D. Caldwell, and A. Billard, "Learning and reproduction of gestures by imitation: An approach based on Hidden Markov Model and Gaussian Mixture Regression," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 44–54, 2010.

[49] S. Calinon and A. Billard, "Statistical Learning by Imitation of Competing Constraints in Joint Space and Task Space," *Advanced Robotics*, vol. 23, no. 15, pp. 2059–2076, 2009.

[50] F. Guenter, M. Hersch, S. Calinon, and A. Billard, "Reinforcement learning for imitating constrained reaching movements," *RSJ Advanced Robotics, Special Issue on Imitative Robots*, vol. 21, no. 13, pp. 1521–1544, 2007.

[51] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[52] S. Calinon, *Robot Programming by Demonstration: A Probabilistic Approach.* EPFL/CRC Press, 2009, EPFL Press ISBN 978-2-940222-31-5, CRC Press ISBN 978-1-4398-0867-2.

[53] S. hee Lee, J. Kim, F. C. Park, M. Kim, and J. E. Bobrow, "Newtontype algorithms for dynamics-based robot motion optimization," *IEEE Transactions on Robotics*, vol. 21, pp. 657–667, 2005.

[54] J. E. Bobrow, B. Martin, G. Sohl, E. C. Wang, F. C. Park, and J. Kim, "Optimal robot motions for physical criteria," *Journal of Robotic Systems*, vol. 18, no. 12, pp. 785–795, 2001.

Acknowledgements

I would like to express the deepest appreciation to my dissertation advisor, Professor Atsushi Konno, for their guidance and encouragements all along this research. He continually and convincingly conveyed a spirit of adventure in regard to research, and Without his guidance and persistent help this dissertation would not have been possible.

I would like to thank my committee members, Professor Satoshi Ogasawara and Professor Hajime Igarashi. Many thanks to the members of the Intelligent Robots and System Lab for making the lab a lively place. Special thanks to Assistant Professor Shunsuke Komizunai, whose helped me live in japan and lab.

Specially a great thanks to my lovely wife. She always believe and encourage me in everything. I wouldn't have been as successful without her, I can't imagine life without her.

Finally a great thanks to my parents and sisters. I am greatly indebted to and enormously grateful for their constant support and multiple encouragements.

List of Publications

Refereed Journal Papers

[1] G. L. Park, S. K. Ra, C. H. Kim, and J. B. Song, "Motion Imitation Learning and Real-time Movement Generation of Humanoid Using Evolutionary Algorithm," *The J. of Institute of Control, Robotics and Systems*, Vol. 14, No. 10, pp. 1038-1046, 2008 (In korean).

[2] Garam Park and Atsushi Konno, "Imitation Learning Framework based on Principal Component Analysis," *Advanced Robotics*, Vol. 29, No. 5, 2015 (to appear).

Refereed Conference Papers

[1] Garam Park, Syungkwon Ra, Changhwan Kim and Jaebok Song, "Imitation Learning of Robot Movement Using Evolutionary Algorithm," Proceedings of the 17th World Congress International Federation of Automatic Control, pp. 730-735, 2008.

[2] Syungkwon Ra, Garam Park, Changhwan Kim and Bun-Jae You, "PCA-based Genetic Operator for Evolving Movements of Humanoid Robot," Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1219-1225, 2008.

[3] Garam Park, KangGeon Kim, ChangHwan Kim, Mun-Ho Jeong, Bum-Jae You and Syungkwon Ra, "Human-like Catching Motion of Humanoid Using Evolutionary Algorithm(EA)-based Imitation Learning," Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication, pp. 809-815, 2009.

[4] Garam Park, Changhwan Kim, "Constructing of Optimal Database Structure by Imitation Learning based on Evolutionary Algorithm," Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2698-2703, 2010.