Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

6CCE3EEP/7CCEMEEP

Individual Project Submission 2022/23

Name: Saitarun Nadipineni

Student Number: 20059781

Degree Programme: Electronic Engineering (BEng)

Project Title: Machine Teaching for Robots

Supervisor: Dr. Mathew Howard

Word count:

RELEASE OF PROJECT

Following the submission of your project, the Department would like to make it publicly available via the library electronic resources. You will retain copyright of the project.

☒ I **agree** to the release of my project

☐ I **do not** agree to the release of my project

RELEASE OF VIDEO

Following the submission of your project demonstration, the Department would like to make it publicly available via youtube. You will retain copyright of the project.

☒ I **agree** to the release of my project video

☐ I **do not** agree to the release of my project video

Signature: (An e-signature is acceptable, i.e., attach your signature as a picture, sign using the draw function)

Date:

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

Originality Avowal

I verify, Saitarun Nadipineni, am the sole author of this report, except where explicitly stated to the contrary.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

Contents

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

Abstract

This project report will explore the optimal teaching that can be given to robots to enable them to perform to learn and perform useful dynamic motor skills. This project teaches two motor skills to two-link arm asses its learning performance. The teaching was done in the form of training data given to the learner modelled in MATLAB for multiple trials to obtaining a large set of results to increase the reliability of the data. The results obtained through the experiments support the defined optimality condition that optimal training data given to a learner can optimise their learning performance in learning the desired motor skill.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

1 Introduction

This section of the project report will explain the project motivation, aims and objectives of this project as well as the report structure.

1.1 Project Motivation

Robots have been integrated into our lives for many decades to perform tasks that can help speed up basic tasks by automating them. Novice users who wish to use robots with motor skills to perform useful tasks for them are increasing. This has made Learning from demonstration (LfD) an intuitive and promising approach to teach motor skills to robots by novice users [1].

1.2 Aims and Objectives

The aim of this project is to improve the quality of training data that is given by humans to robots, this is to increase their learning performance and learn the motor skill with less error. This will make robots cable enough to perform the tasks with a similar accuracy to humans. In past research conducted on this problem simple skills which aren't very beneficial to humans in the real world were taught to a rudimentary robot [1]. In this project two different motor skills which are useful will be taught a two-link robot arm and it must reproduce the demonstrated action. The error in learning the demonstrated skill will be compared against the quality of the training data asses the effectiveness of the training data.

1.3 Report Structure

- Chapter 2 of this project report will cover the basic background information from past research that will be beneficial to this project and how optimal training data can be produced.
- Chapter 3 will cover the experiments from past research to gain a deeper understanding of the problem and how a two-link arm can be modelled as well as how optimal training data can be produced for it.
- Chapter 4 discusses about the first experiment that was conducted and it will discuss about the results obtained from it.
- Chapter 5 discusses about the second experiment that was conducted and it will discuss about the results obtained form it as well as how they differ from the first experiment.
- Chapter 6 will discuss about the conclusions that can be drawn from both the experiments. In addition, it will discuss about the improvements that can be made for the future work.
- Chapter 7 will discuss about the professionalism and responsibility of this project.
- Chapter 8 contains the citation that was used for the research on this project.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

2 Background

This section of the report will give the required background information from past research done, which can aid this project. In addition, it will discuss how the teacher, learner, and the condition for obtaining optimal training data can be defined.

2.1 Learning from demonstration (LfD)

LfD is a simple and intuitive way for novice users who have no technical knowledge on robots to teach them useful motor skills. The major problem with this method is that it is heavily dependent on the teacher to giving the learner (i.e., the robot) good quality teaching. Otherwise, this method can fail by the learner reproducing the skill inaccurately. This condition is also true in the real world, for example a student can learn skills through demonstrations much better when the teacher breaks down the steps and explains it clearly. This is due to the teacher having a thorough understanding of the skill and has a clear process in mind in which they can use to teach the learner. However, if a novice teacher tries to teach the skill, then the demonstration is of low quality which isn't ideal for the learner to grasp the skill effectively [1,12].

Though many algorithms have been developed throughout the years, these algorithms are still heavily dependent on the quality of the teaching received form the teacher. Therefore, the demonstrations given by the teacher must be optimised to reduce teaching time, cost and the effort put into teaching the skill [1]. This problem can be solved by using methods that can give data that is tailored for optimal performance for the robot [1]. The machine teaching approach can be used as training data can be derived by solving bilevel optimisation problems, this is given that the teacher has knowledge of the learning algorithm [1]. The learning algorithm is already known through the research conducted; therefore, it is suitable to use this method.

2.2 Model of the teacher

The high-quality demonstrations given to a torque-controlled robot can be modelled in the form of $\boldsymbol{\mathcal{D}}$. The skills learnt by the robot can be represented as a closed loop controller. A closed loop controller will automatically regulate the system and maintain the optimal desired state without interaction from the user [2]. The closed loop controller that can be used is the following,

$$u = \pi(x, \theta) \quad (1)$$

where $\pi(x)$ maps the state of the system to the desired action that the robot must perform which is $u \in \mathbb{R}$ and $\theta \in \mathbb{R}^{\delta}$ is the skill parameter [1].

The data provided by the teacher in the demonstration set $\boldsymbol{\mathcal{D}}$ can contain the sates x and the list of desired actions u. The demonstration set $\boldsymbol{\mathcal{D}}$ contains the list of states $[x_1, \dots, x_{\mathcal{N}}] = \boldsymbol{X} \in \mathbb{R}^{\mathcal{P} \times \mathcal{N}}$ and the list of desired

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

actions $(u_1, \ldots, u_{\mathcal{N}})^\top = u \, \epsilon \, \mathbb{R}^{\mathcal{N}}$. In this case the desired action is the torque applied to the robot's joints. Given the data in $\boldsymbol{\mathcal{D}}$ the learner can create the model for the skill that must be learnt. This can be represented using,

$$\tilde{u} = \pi(x, \tilde{\theta}) \text{ (2)}$$

where $\tilde{u}$ is the action performed by the learner and $\tilde{\theta}$ is the skill learnt by the learner [1].

Using this model, the learner can reproduce the action taught by estimating the parameters through a learning algorithm. This can be modelled using,

$$\tilde{\theta} = \boldsymbol{\mathcal{A}}(\boldsymbol{\mathcal{D}}) \text{ (3) [1].}$$

The main goal of the teacher in this project is to produce a data set $\boldsymbol{\mathcal{D}}$ that helps the learner to create an estimated model of the target skill. The teacher's skill can be modelled as

$$u = \theta^{*\top} \phi(x) \text{ (4)}$$

where $\phi(x)$ is a vector of features or basis functions and θ^* is the target skill parameter being taught [1].

The teaching risk function $\rho(\tilde{\theta}, \theta^*)$ which limits the total teaching effort put in by the teacher must be minimised by the teacher by finding the training set $\boldsymbol{\mathcal{D}}$ from the space of possible data sets $\mathfrak{D}$ as training robots in the real world usually require a lot of resources such as time, equipment, and computational power. This relationship can be given as,

$$D = \underset{\boldsymbol{\mathcal{D}} \, \epsilon \, \mathfrak{D}}{\arg\min} \, \rho(\hat{\theta}, \theta^*) \text{ (5) [1].}$$

The teaching risk function can then be written as,

$$\rho(\tilde{\theta}, \theta^*) = \mathbb{E}\left[(\tilde{\theta} - \theta^*)^\top (\tilde{\theta} - \theta^*)\right] \text{ (6)}$$

where $\mathbb{E}$ is the expectation over the error in the skill parameter $\tilde{\theta}$ when there is Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma)$ on u [1]. This noise is assumed to be present from the sensor readings or the errors from the demonstrations.

The teacher must try and teach the target skill with a fixed budget of teaching effort so that the teaching method is efficient. The teaching effort in this case can be defined as,

$$\varepsilon(D) = \mathcal{N}_{TD} \text{ (7)}$$

where $\varepsilon(D)$ is the teaching effort function and $\mathcal{N}_{TD}$ is the teaching dimension. The teaching dimension $\mathcal{N}_{TD}$ is defined as the minimum number of training items needed to teach the target skill to the learner [1].

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

2.3 Model of the learner

The goal of the learner is to learn the target skill θ^* which is taught from the demonstrations $\boldsymbol{\mathcal{D}}$ using the learning algorithm $\boldsymbol{\mathcal{A}}$, this is the relationship that was modelled earlier. As mentioned before the learner can estimate the target skill θ^* but, in this project it is assumed to be done with a linear-in-the-parameters (LiP) model which can be represented as

$$\tilde{u} = \tilde{\theta}^{\top}\phi(x) \text{ (8) [1].}$$

To avoid overfitting when learning, it is assumed that the learner approximates through ridge regression. This minimises the loss function which is,

$$\min_{\tilde{\theta} \in \mathbb{R}^{\mathcal{S}}} \sum_{i=1}^{\mathcal{N}} \frac{1}{2}\left(\tilde{\theta}^{\top}\phi_i - u_i\right)^2 + \frac{\lambda}{2}\left\|\tilde{\theta}\right\|^2 \text{ (9)}$$

where λ represents the regularisation parameter and $\|\theta\|$ is the Mahalanobis norm of θ [1]. Overfitting is a problem when the modelled learner is learning as it could cover all the data points, or more data points than needed in the given dataset. This can cause unwanted noise and the model can produce data points that are inaccurate, this can reduce the learner's efficiency and accuracy to learn the target skill [3]. Ridge regression is a technique used to solve this problem as it analyses the data for the data points that have multicollinearity (when multiple data points have linearity between each other) [4]. The closed form solution of (9) can be given as,

$$\tilde{\theta} = (\boldsymbol{\Phi}\boldsymbol{\Phi}^{\top} + \lambda I)^{-1}\boldsymbol{\Phi}u \text{ (10)}$$

where $I \in \mathbb{R}^{\mathcal{S}\times\mathcal{S}}$ represents an identity matrix and $\boldsymbol{\Phi} \in \mathbb{R}^{\mathcal{S}\times\mathcal{N}}$ is a feature matrix that's mapped to $\boldsymbol{X}$ through the basis functions $\phi(x)$ [1].

2.4 Machine Teaching

For the modelled learner the teaching dimension is $\mathcal{S}$ for the target skill so the teaching effort can be written as,

$$\varepsilon(\boldsymbol{\mathcal{D}}) = \mathcal{S} \text{ (11).}$$

The optimal data set to teach the learner can be written as,

$$\boldsymbol{\mathcal{D}} = \arg\min_{\boldsymbol{\mathcal{D}} \in \mathfrak{D}} \mathbb{E}\left[\left(\tilde{\theta} - \theta^*\right)^{\top}\left(\tilde{\theta} - \theta^*\right)\right] \text{ (12)}$$

$$\text{s.t. } \tilde{\theta} = (\boldsymbol{\Phi}\boldsymbol{\Phi}^{\top} + \lambda I)^{-1}\boldsymbol{\Phi}u \text{ and } \varepsilon(D) = \mathcal{S} \text{ (13) [1].}$$

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

The solution to (12)-(13) is the optimal data set the teacher can teach the learner however, as stated in [1], this solution is non-unique, and it is difficult to find the closed form solution for cases where the teaching dimension $\mathcal{S} > 2$. Therefore, it is better to use the strategy in [1] where it is better to find the conditions for demonstration optimality under (12)-(13) instead of using a specific teaching method.

2.4.1 Optimal teaching for $\mathcal{S} > 2$

As mentioned in [1] the feature matrix for the optimal data set is defined to be square $\boldsymbol{\Phi} \epsilon \mathbb{R}^{\mathcal{S}\times\mathcal{S}}$.The feature vectors are assumed to be normalised $(i.e., 0 \leq \|\phi(x)\| \leq 1)$ and the optimal set regularisation λ is not needed so $\lambda = 0$, then (10) can be reduced to

$$\tilde{\theta} = \boldsymbol{\Phi}^{-1} u \text{ (14) [1].}$$

The condition given in [1] to solve (14) is that an inverse exists, or equivalently the determinant of $\boldsymbol{\Phi}$ is non-zero so,

$$\det \boldsymbol{\Phi} \neq 0 \text{ (15).}$$

The condition for optimal teaching can therefore be defined as,

$$\text{Optimal demonstrations maximise } |\det \boldsymbol{\Phi}|.$$

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

2.4.2 Optimal teaching for $\mathcal{S} = 2$

Analytically it possible to find the closed form solution of (12)-(13) for cases up to $\mathcal{S} = 2$. The general expression of the feature matrix $\boldsymbol{\Phi}\ \epsilon\ \mathbb{R}^{2\times 2}$ assuming that the feature vectors are normalised is

$$\boldsymbol{\Phi} = (\boldsymbol{\phi_1} \quad \boldsymbol{\phi_2}) = \begin{pmatrix} 1 & \cos\omega \\ 0 & \sin\omega \end{pmatrix} (16)$$

where ω is the angle between the feature vectors $\boldsymbol{\phi_1}$ and $\boldsymbol{\phi_2}$ [1].

It is stated in [1] the teaching risk function can be written as

$$\rho(\tilde{\theta}, \theta^*) = \sigma^2 \sum_{i=1}^{2} \frac{b_i}{(b_i+\lambda)^2} + \lambda^2 \theta^{*\mathsf{T}} (\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathsf{T}} + \lambda I)^{-2} \theta^* \ (17)$$

by combing the equations (6) and (10). In this equation b_i is the ith eigenvalue $\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathsf{T}}$ and σ^2 is the variance of the noise on the training data. Since λ is small, the second term can be ignored and (15) can be reduced to be given as

$$\rho(\tilde{\theta}, \theta^*) = \sigma^2 \sum_{i=1}^{2} \frac{b_i}{(b_i+\lambda)^2} \ (18)$$

where

$$b_1 = 1 + \sqrt{1 - \sin^2\omega} \text{ and } b_2 = 1 - \sqrt{1 - \sin^2\omega} \ (19) \text{ [1].}$$

By combing both (16) and (17) it is stated in [1] that the derivative of this problem can be given as

$$\frac{d\rho}{d\omega} = \frac{4\sigma^2 \cos\omega \sin\omega \left((2\lambda+1)\sin^2\omega - 2\lambda^3 - 3\lambda^2 - 2\lambda\right)}{\left(\sqrt{1-\sin^2\omega} - \lambda - 1\right)^3 \left(\sqrt{1-\sin^2\omega} + \lambda + 1\right)^3} \ (20).$$

It is shown in [1] that $\omega^* = \pi/2$ satisfies $\frac{d\rho}{d\omega} = 0$ and $\frac{d^2\rho}{d\omega^2} > 0$ so this means for the feature matrix (15) the optimality condition is,

$$|\det \boldsymbol{\Phi}| = |\sin\omega| \ (21)$$

which is maximised when $\omega = \omega^*$ which meets the optimality condition.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

3 Theory

In this section the simulation created by Marina Y. Aoyama and Matthew Howard will be conducted from the research previously done on this problem [1] to gain a deeper understanding of the problem and how it can be solved.

In addition, this section discusses the theory behind the dynamic model of a two-link arm and the calculations made for modelling the target skill taught to the learner for the experiment related to two-link arm.

It will also cover how optimal training data can be given to the two-link arm in specific by using the optimality condition discussed in the background section.

3.1 Teaching Motor Skills to a Simple Pendulum

The demonstrations in this simulation must meet the condition for optimal training data that is defined in section 2.4.1. In this simulation the learner is a simple pendulum, it must reproduce the same trajectory demonstrated by in the simulation. The pendulum is modelled with a mass m of 1 kg, a length l of 1 m and the gravitational field strength g is taken as 9.81 m/s^{-2}.

The system state of this pendulum is defined as x $= (q, \dot{q})^{\mathsf{T}}$ where q and $\dot{q}$ are the angular displacement and angular velocity of the pendulum. The desired action u the learner must learn is the torque τ applied at the pivot by a motor, so $u = \tau$. There is no torque limit that is set in the simulation on MATLAB and the sampling rate is set to 10 kHz. The aim to teach skills S1 undamped oscillation and S2 rapid movement without overshoot which have the form of LiP controller (1). The skill parameters are $\theta = (k, d)^{\mathsf{T}}$ and the feature vector is $\phi(x) = (\sin q\,, \dot{q})^{\mathsf{T}}$ which will determine the state of the pendulum.

The simulation code is made by Marina Aoyama and Mathew Howard who are the authors of the 'Training Humans to Train Robots Dynamic Motor Skills' research paper [1] which is the core material that this project is based on.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

3.1.1 S1 Undamped oscillation

In this simulation the learner is trying to learn undamped oscillation which is the target skill S1 that can be represented using the LiP controller of the form $\theta = (k, d)^{\mathsf{T}}$ which results in the target skill parameters being $\theta = (g/l, 0)^{\mathsf{T}}$. The simulation was conducted for 500 trials and collected the samples $S = 30{,}000$ with each trial. The noise level used is $\sigma = 0.1$.

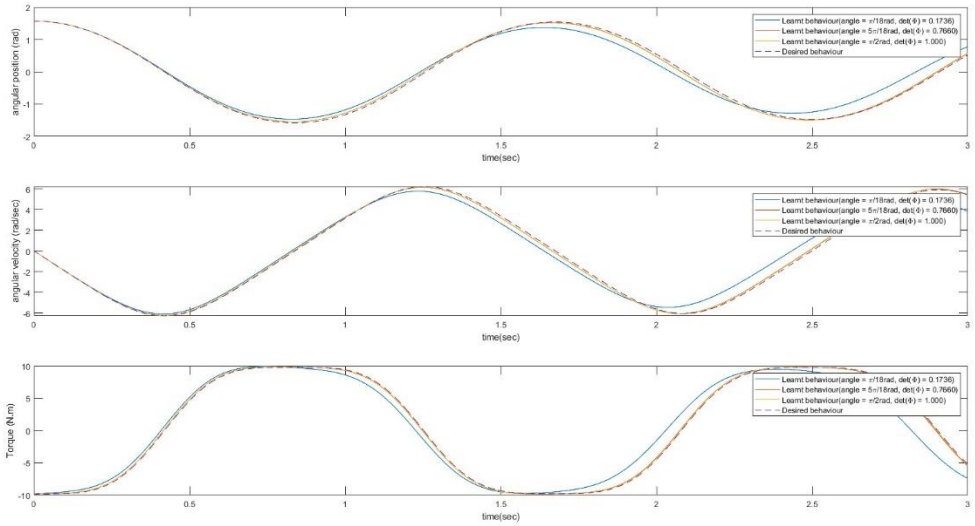


Figure 3.1: S1 Results

Figure 3.1 shows the results obtained from Marina Aoyama's MATLAB simulation show when ω approaches $\frac{\pi}{2}\ rad$ which is when the determinant of $\mathbf{\Phi}$ is maximised, the learnt behaviour almost matches the desired behaviour. Whereas when less optimal choices of ω were used the learnt trajectory had a greater difference from the desired behaviour. This supports the optimality condition defined in section 2.4.1. This shows that choosing high quality training is very important to maximising the learner's performance.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

3.1.2 S2 Rapid movement without overshoot

In this simulation the learner is trying to learn rapid movement without overshoot (akin to critical damping) which is the target skill S2 that can be represented using the LiP controller of the form $\theta = (k, d)^{\mathsf{T}}$ which results in the target skill parameters being $\theta = \left(g/l, 2\sqrt{g/l}\right)^{\mathsf{T}}$. The simulation was conducted for 500 trials and collected the samples $S = 30{,}000$ with each trial. The noise level used is $\sigma = 0.1$.

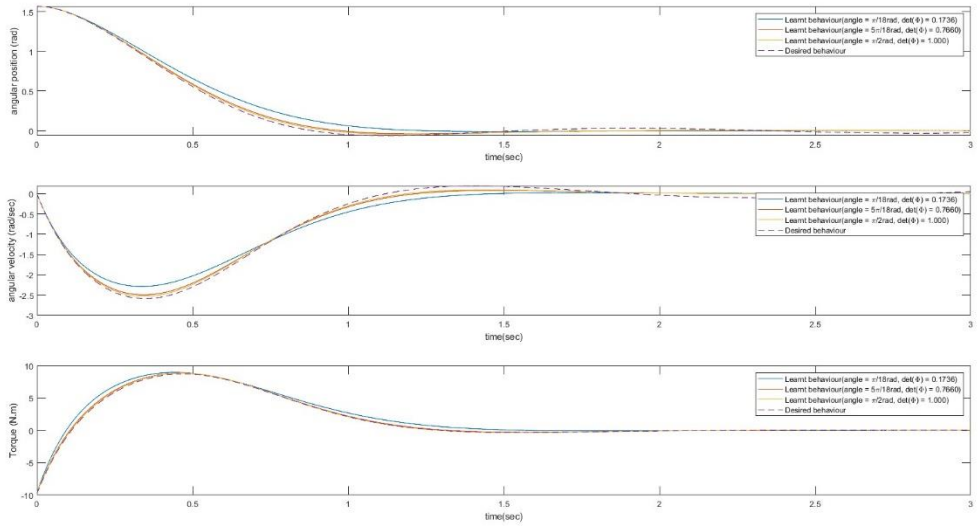


Figure 3.2: S2 Results

Figure 3.2 shows the desired behaviour of the pendulum to be critically damped instead of the undamped oscillations in the simulation for S1. S2 results show when ω approaches $\frac{\pi}{2}\ rad$ which is when the determinant of $\mathbf{\Phi}$ is maximised, the learnt behaviour almost matches the desired behaviour. Whereas when less optimal choices of ω were used the learnt trajectory had a greater difference from the desired behaviour. The results from this experiment also support the optimality condition defined in section 2.4.1. Though the skills taught in both simulations were different they lead to the same conclusion through the results where choosing high quality training is very important to maximising the learner's performance.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

3.2 Physical model of the two-link arm

The following diagram shows the dynamic model of a two-link arm where m_1 and m_2 represent the mass of the links. l_1 and l_2 represent the length of the individual links and y_1, x_1, y_2, x_2 represent the forces in the x-y directions. q_1 and q_2 represent angular displacements of the corresponding joints. Both links l_1 and l_2 are modelled to be rods that have a length of 1 m and their mass m_1 and m_2 is 1 kg.

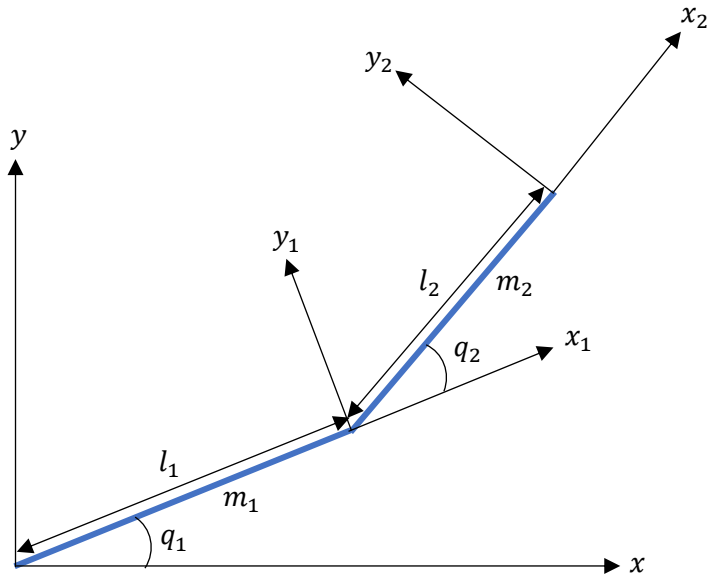


Figure 3.3: Coordinate model of 2-link system

The two-link arm's dynamic torque from the origin can be represented using,

$$\tau = M(q)\ddot{q} + N(q,\dot{q})\dot{q} + G(q) \quad (22)$$

where $q, \dot{q}, \ddot{q} \in \mathbb{R}$,are the joint angular displacement, joint angular velocity, and joint angular acceleration vectors. $M(q)$ is the inertia matrix, $N(q,\dot{q})$ is the Coriolis matrix and $G(q)$ is the gravity moment component. Equation (22) models the two-link system's physical properties in the environment that it is present in. In this project it is assumed that the arm is present in an environment where a significant gravitational force is present, hence the gravity moment component is included [5].

The matrices $M(q), N(q,\dot{q})$ and $G(q)$ can be defined as

$$M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = \begin{bmatrix} m_1(l_1/2)^2 + m_2 l_1^2 + I_1 & (1/2)m_2 l_1 l_2 \cos(q_2 - q_1) \\ (1/2)m_2 l_1 l_2 \sin(q_2 - q_1) & m_2(l_2/2)^2 + I_2 \end{bmatrix},$$

$$N(q,\dot{q}) = \begin{bmatrix} N_{11} \\ N_{21} \end{bmatrix} = \begin{bmatrix} -(1/2)m_2 l_1 l_2 \sin(q_2 - q_1) \\ (1/2)m_2 l_1 l_2 \sin(q_2 - q_1) \end{bmatrix}$$

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

and

$$G(q) = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} = \begin{bmatrix} m_1 g(l_1/2)\cos(q_1) + m_2 g l_1 \cos(q_1) \\ m_2 g(l_2/2)\cos(q_1) \end{bmatrix}$$

where I_1 and I_2 are the moment of inertia of the respective links. The moment of inertia for the first link is $I_1 = \frac{1}{12} m_1 (l_1)^2$ and for the second link it is $I_2 = \frac{1}{12} m_2 (l_2)^2$.

The joint torques for the respective joints can then be computed using (22) and can be defined as

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} m_1(l_1/2)^2 + m_2 l_1^2 + I_1 & (1/2) m_2 l_1 l_2 \cos(q_2 - q_1) \\ (1/2) m_2 l_1 l_2 \sin(q_2 - q_1) & m_2(l_2/2)^2 + I_2 \end{bmatrix} \begin{bmatrix} \ddot{q_1} \\ \ddot{q_2} \end{bmatrix} +$$

$$\begin{bmatrix} -(1/2) m_2 l_1 l_2 \sin(q_2 - q_1) \\ (1/2) m_2 l_1 l_2 \sin(q_2 - q_1) \end{bmatrix} \begin{bmatrix} \dot{q_2}^2 \\ \dot{q_1}^2 \end{bmatrix} + \begin{bmatrix} m_1 g(l_1/2)\cos(q_1) + m_2 g l_1 \cos(q_1) \\ m_2 g(l_2/2)\cos(q_1) \end{bmatrix} \text{(23) [7].}$$

3.3 PD Controller

A Proportional Derivative (PD) Controller is necessary implementation for the robot arm, if the robot arm needs to follow a fixed trajectory it needs to operate on both the current state and predicted state. This is where the PD controller can be implemented as it is a combination of both feedforward and feedback control. The Proportional control or the P term is a form of continuous feedback control that can be used in a closed loop system. The Derivative control or the D term can predict the error which can increase the stability of the closed loop system. The PD controller correlates the controller output to the error and the derivative of the error, therefore the controller can be given as

$$c(t) = K_c \left(e(t) + T_d \frac{de}{dt} \right) + C \text{ (24)}$$

where $c(t)$ is the output of the controller,K_c is the proportional gain, $e(t)$ is the error and C is the initial value of the controller [6].

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

3.4 Joint torque

3.4.1 Joint Torque of Single-link arm

The joint torque of a single link can be modelled using the LiP controller defined in (4).

The joint torque is the desired action so $u = \tau$, joint torque can be written as,

$$\tau = [k \quad d] * \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$$

$$= kq + d\dot{q} \quad (25)$$

where $\theta = (k, d)^\top$ and the feature vector is $\phi(x) = (q, \dot{q})^\top$ which is a function of the system state $x = (q, \dot{q})^\top$.

3.4.2 Joint Torque of Two-link arm

The joint torque of the two-link arm can be computed in the same way as (25). The system state is $x = (q_1, q_2, \dot{q_1}, \dot{q_2})^\top$ which contains the angular displacement and angular velocity of the respective joints. The joint torque is the desired action so $u = \tau$, therefore the joint torque can be computed using (4). The target skill is given as $\theta^* = L$ where L is the gain matrix which contains the proportionality constants for the angular displacement and the angular velocity of the two joints respectively. The feature vector can therefore be given as $\phi(x) = (q_1, q_2, \dot{q_1}, \dot{q_2})^\top$. The torque for the two-link arm can then be computed in the following way:

$$u = \theta^{*\top}\phi(x)$$

$$= L * x$$

$$= \begin{bmatrix} L_{11} & L_{12} & L_{13} & L_{14} \\ L_{21} & L_{22} & L_{23} & L_{24} \end{bmatrix} * \begin{bmatrix} q_1 \\ q_2 \\ \dot{q_1} \\ \dot{q_2} \end{bmatrix}$$

$$= \begin{bmatrix} (L_{11})(q_1) & + & (L_{12})(q_2) & + & (L_{13})(\dot{q_1}) & + & (L_{14})(\dot{q_2}) \\ (L_{21})(q_1) & + & (L_{22})(q_2) & + & (L_{23})(\dot{q_1}) & + & (L_{24})(\dot{q_2}) \end{bmatrix}$$

the torque for the respective joint can be given as,

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} (L_{11})(q_1) & + & (L_{12})(q_2) & + & (L_{13})(\dot{q_1}) & + & (L_{14})(\dot{q_2}) \\ (L_{21})(q_1) & + & (L_{22})(q_2) & + & (L_{23})(\dot{q_1}) & + & (L_{24})(\dot{q_2}) \end{bmatrix} \quad (26).$$

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

However, for the robot arm to follow a fixed trajectory a start state and target state must defined. This is where the PD controller (24) can be implemented to calculate the joint torque for each time step in the MATLAB simulation by computing the error between the target state and the current state. The full trajectory can then be plotted using a 'for loop' in MATLAB. This creates a continuous feedforward and feedback loop as it operates on both the current process conditions and predicted process conditions.

The desired joint torques computed using the LiP controller u_1 and u_2 can be applied by motors at the joints of both links. Torques for the two-link arm model τ_1 and τ_2 in (23) can therefore, be given as

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

where the desired torque is applied to the two-link arm model.

3.5 Optimal Demonstrations

The optimal teaching condition defined in 2.4.1 is that optimal demonstrations maximise $|\det \boldsymbol{\Phi}|$. This is only possible when the feature matrix for the optimal data set is square $\boldsymbol{\Phi} \epsilon \mathbb{R}^{\mathcal{S}\times\mathcal{S}}$ and when the feature vectors are normalised ($i.e., 0 \leq \|\phi(x)\| \leq 1$). The teaching dimension $\mathcal{S}$ for the two-link arm model should be $\mathcal{S} = 4$ as that is dimensionality of target model [8]. As mentioned in 2.4.1 a closed form solution is difficult to obtain for cases where $\mathcal{S} > 2$, therefore the feature matrix $\boldsymbol{\Phi}$ used must satisfy the optimality condition by following the assumptions made in 2.4.1.

Since $\mathcal{S} = 4$ the general expression for the feature matrix $\boldsymbol{\Phi} \epsilon \mathbb{R}^{4\times4}$,assuming that the feature vectors are normalised ($i.e., 0 \leq \|\phi(x)\| \leq 1$), is

$$\boldsymbol{\Phi} = (\phi_1 \quad \phi_2 \quad \phi_3 \quad \phi_4) = \begin{pmatrix} 1 & \cos\omega & 0 & 0 \\ 0 & \sin\omega & 0 & 0 \\ 0 & 0 & 1 & \cos\alpha \\ 0 & 0 & 0 & \sin\alpha \end{pmatrix} (27)$$

where ω is the angle between the feature vectors ϕ_1 and ϕ_2 and α is the angle between the feature vectors ϕ_3 and ϕ_4.

The feature matrix is maximised when $\omega = \pi/2$ and when $\alpha = \pi/2$,this is shown below:

$$\boldsymbol{\Phi} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \Rightarrow |\det \boldsymbol{\Phi}| = 1.$$

This shows that $\boldsymbol{\Phi}$ supports the optimality condition set for the demonstrations as the $|\det \boldsymbol{\Phi}|$ is maximised.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

4 Experiment 1

The skill taught to the learner in this experiment is to follow a fixed trajectory from the starting point and must come rest when it reaches the target point. The main code snippets used in the simulation are shown and explained below in detail. The results obtained from this experiment are also shown and their analysed in detail.

4.1 Simulation Code

4.1.1 Code for the Parameters of the two-link arm:

The physical model of the two-link arm described in section 3.2 was already created by Stefan Klanke, Matthew Howard, Djordje Mitrovic and Sethu Vijayakumar under the MATLAB file name 'fn_dynamics_2link0.m' at King's College London. This function will be used to compute analytical dynamics of a two-link arm.

```
g=9.81;
m1=1;
m2=1;

L1=1;
L2=1;
I1=(1/12)*m1*(L1^2);
I2=(1/12)*m2*(L2^2);

M = [m1*((L1/2)^2)+m2*(L1^2)+I1      (1/2)*m2*L1*L2*cos(q(2)-q(1));
       (1/2)*m2*L1*L2*cos(q(2)-q(1))   m2*((L2/2)^2)+I2];


N = qd' * [0         -(1/2)*m2*L1*L2*sin(q(2)-q(1));
          (1/2)*m2*L1*L2*sin(q(2)-q(1))          0] * qd;


G = [ m1*g*(L1/2)*cos(q(1)) + m2*g*L1*cos(q(1));
      m2*g*(L2/2)*cos(q(2))];
```

Figure 4.1: Two-link arm model in MATLAB

Figure 4.1 shows the code snippet from 'fn_dynamics_2link0.m' that creates the parameters of the two-link arm model described in section 3.2. The 'fn_dynamics_2link0' function takes the current state and the desired action as the input it will return the state change of the two-link arm.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

4.1.2 Code for the trajectory setup:

```
% Machine Teaching for Robots project by Saitarun Nadipineni.
clear;close all;clc

robot   = 0;
fnDyn   = @(x, u)  fn_dynamics_2link0(x,u,robot);

x0 = [ pi;  %start state
       pi;
        0;
        0];

xt = [ pi/2; %target state
       pi/2;
          0;
          0];

L= [20 0 6 0;   %Gain Matrix L
    0 20 0 6];

dt= 0.0002; % sampling rate (unit: dt = s)
T = 2; % time (unit: T = s)
N = T/dt;
t=(0:N-1)*dt;
```

Figure 4.2: Trajectory setup code

Figure 4.2 shows the code for the all the parameters that have been setup prior to their usage. A function handle is created which accesses the 'fn_dynamics_2link0' function mentioned in section 4.1.1, this will make it easier to call the function locally. The start state $x_0 = (\pi, \pi, 0,0)^{\top}$ and the target state $x_t = (\pi/2, \pi/2,0,0)^{\top}$ are chosen to follow to desired trajectory. The target skill $\theta^* = L = \begin{bmatrix} 20 & 0 & 6 & 0 \\ 0 & 20 & 0 & 6 \end{bmatrix}$ has been setup with the chosen values which produce the most stable trajectory through testing. The sampling rate is 20 kHz and the total number of time steps or samples S over the trajectory time of 2 seconds is 10,000.

4.1.3 Code for defining the feature matrix:

```
%Defining the feature matrix
function Phi = createPhi(degree)
    omega1 = 0; omega2 = degree; alpha1 = 0; alpha2 = degree;

    Phi = [cosd(omega1) cosd(omega2) 0           0           ;
           sind(omega1) sind(omega2) sind(alpha1) 0          ;
           0            0            cosd(alpha1) cosd(alpha2);
           0            0            0           sind(alpha2)];
end
```

Figure 4.3: Feature Matrix code

Figure 4.3 shows the code for the function that contains the feature matrix (27) which has been verified to satisfy the condition for optimal demonstrations. This function takes the angle between the feature vectors as the input then assigns it to ω and α compute the matrix of form (27), the function then returns the computed matrix.

Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

4.1.4 Code for the learner model:

```
%Learner model
function theta = learn(theta_l,Phi,std_noise)
Y_noise = std_noise*randn(2,4);
Y = theta_l*Phi;
Y = Y + Y_noise;
lambda = 10^(-6);    % regularisation term (10^(-8) to 10^(-4))
I = eye(size(Phi,1));
theta = pinv(Phi*transpose(Phi)+lambda*I)*Phi*transpose(Y); % Learning algorithm
end
```

Figure 4.4: Learner model code

Figure 4.4 shows the code for the function that computes the model of the skill learnt by the learner. This function takes the target skill parameters θ^*, the feature matrix $\mathbf{\Phi}$ represented and the noise σ as the inputs. A matrix of random numbers is created and then it is multiplied by the noise σ which is then added to the torque. The desired action is the joint torque so $u = \tau$ and it is computed using (14). The regularisation term is $\lambda = 10^{-6}$ and learnt model $\tilde{\theta}$ is computed using (10) and function then returns it.

4.1.5 Code for the PD controller:

```
x_l(:,1)=x0;

for k=1:N-1
   xk_l=x_l(:,k);
   err_l = xt - xk_l;
   uk_l = theta_l*err_l;
   xdot_l=fnDyn(xk_l,uk_l);
   x_l(:,k+1)=xk_l + dt*xdot_l;
   u_l(:,k+1)=uk_l;
end
```

Figure 4.5: PD Controller code

Figure 4.5 shows the code snippet where the PD controller is implemented to produce the trajectory of the robot arm. The start state is assigned as the current state before entering the 'for loop'. The 'for loop' then runs for 10,000 time steps and computes the error between the target state and the current state. The state error is then multiplied with gain matrix L to compute the joint torque of both joints (4). Then the joint torque and current state are given as inputs to the 'fn_dynamics_2link0' function which returns the state change of the two-link arm. The next state of robot arm computed is now the current state and the computed joint torques are stored in an array.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

4.1.6 Computing error:

The absolute error which is the difference between the target skill θ^* and the learnt skill $\tilde{\theta}$. It is computed using the equation,

$$E_{ABS} = \frac{(\theta^* - \tilde{\theta})^\top (\theta^* - \tilde{\theta})}{\|\theta^* - \tilde{\theta}\|} \text{ (28).}$$

The root mean squared error is the error in the learnt trajectory. The root mean squared error is calculated used the equation,

$$E_{RMSE} = \frac{1}{S}\sum_{s=0}^{S}\sqrt{(u_s - \widehat{u_s})^\top (u_s - \widehat{u_s})} \text{ (29)}$$

where time steps S is the sample size, u_s and $\widehat{u_s}$ are the desired and learnt trajectories. Since the simulation runs for 2 seconds ($i.e. S = 10000\ steps$). Root mean squared error is a useful measure for evaluating the quality of predictions [9].

4.1.7 Main code:

```
std_noise=0.05;
for s=1:1:3
for trial=1:1:trial_size
trial;
rng("shuffle");

for d=5:5:90 %Obtains data
Phi = createPhi(d);

theta_l=L;
theta = learn(theta_l,Phi,std_noise);
```

Figure 4.7: Main code

Figure 4.7 shows a code snippet of the main code which computes the feature matrix for the given angle in the first 'for loop' then it calls the function which computes the learnt skill model $\tilde{\theta}$. Then the absolute difference is computed between the target model and the learnt model. The root mean squared error in the learnt trajectory is also computed. This loop is repeated 250 times to obtain data for 250 trails and the data is stored in arrays which are then used compute the mean of the total data collected for a certain angle. New noise added for each trial to assess how the learning performance of the learner can change. This done for $\frac{\pi}{36} \leq \omega \leq \frac{\pi}{2}$ and $\frac{\pi}{36} \leq \alpha \leq \frac{\pi}{2}$ in increments of $\frac{\pi}{36}$ for the noise levels $\sigma\ \epsilon\ \{0.05, 0.1, 0.15\}$.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

4.2 Results

4.2.1 Target Trajectory

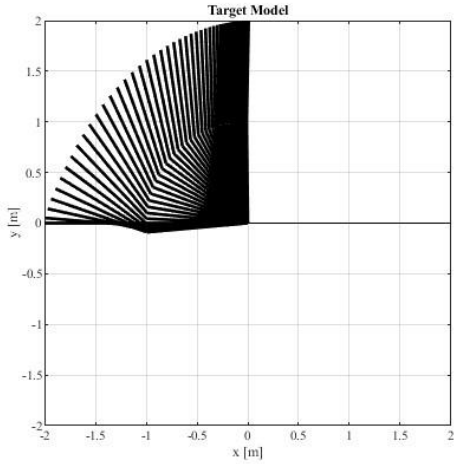


Figure 4.8: Target Trajectory

Figure 4.8 shows the target trajectory followed by the two-link arm using the PD controller. The arm starts from the intial position and comes to rest when it reaches the target postion, the trajectory is plotted using the 'stroboscopic' function which takes angular displacements of both links at each time step as the input.

4.2.2 Sample Trajectories

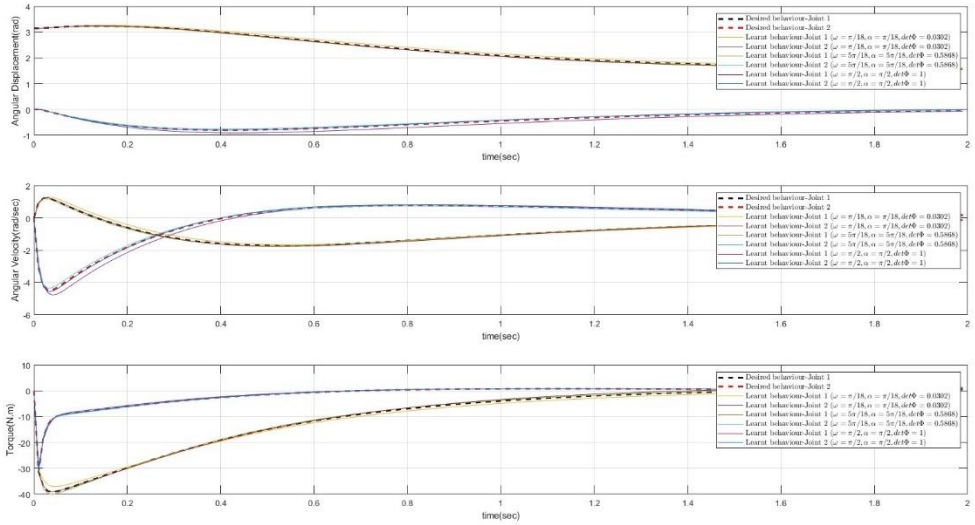


Figure 4.10: Sample Trajectories

Figure 4.10 shows the sample trajectories reproduced by using the learnt model $\tilde{\theta}$ when from the data generated using $\omega = \alpha = \frac{\pi}{36}$, $\omega = \alpha = \frac{\pi}{18}$ and $\omega = \alpha = \frac{\pi}{2}$, the noise level used is $\sigma = 0.1$.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

4.2.3 Error Analysis

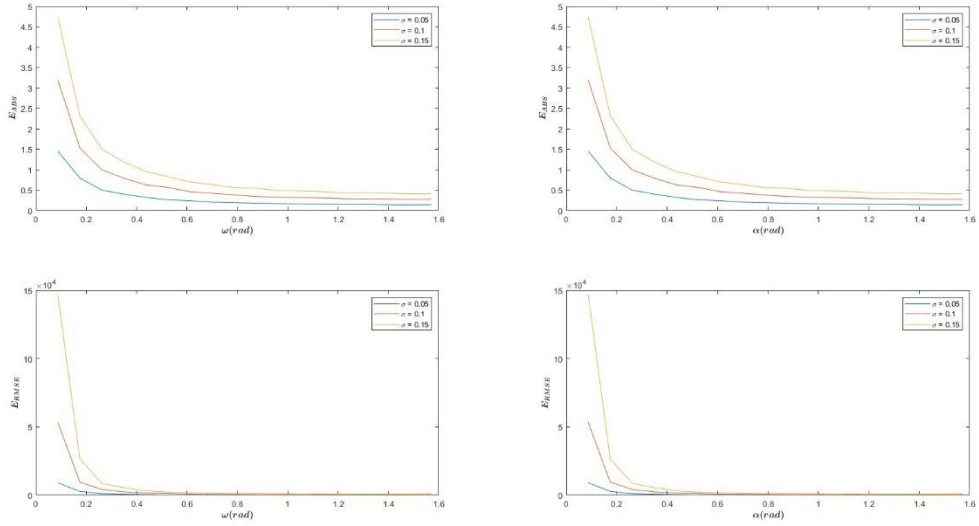


Figure 4.11: Error Results

Figure 4.11 shows the error in behaviour reproduced by the learner and difference between the target model and the learn model. The results recorded in this experiment was for 250 trials at noise levels $\sigma \epsilon \{0.05,0.1,0.15\}$. The root mean squared error E_{RMSE} and the absolute error E_{ABS} are plotted against the angles between the feature vectors ω and α. It can be seen in the figure, for all the noise levels as ω and α approach $\frac{\pi}{2} rad$ (i.e. when the determinant of $\mathbf{\Phi}$ is maximised) the error in the reproduced trajectories and the learnt model rapidly drop. Figure 4.10 shows the sample trajectories reproduced by using the learnt model $\tilde{\theta}$ when from the data generated using $\omega = \alpha = \frac{\pi}{36}$, $\omega = \alpha = \frac{\pi}{18}$ and $\omega = \alpha = \frac{\pi}{2}$, the reproduced trajectory almost perfectly overlaps with the desired trajectory when $\omega = \alpha = \frac{\pi}{2}$. The suboptimal choices for ω and α produce trajectories that have lower accuracy. As expected, the error is much greater when the noise added is higher. The results obtained in this experiment suggest that providing the learner with high quality training data (i.e. meeting the optimality condition defined in section 2.4.1) can significantly optimise the learner's performance and that it is crucial for effective learning.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

5 Experiment 2

The skill taught to the learner in this experiment is to reach out from a folded position to the and must come to rest when it reaches the target point. The main code snippets used in the simulation are shown and explained below in detail. The results obtained from this experiment are also shown and their analysed in detail.

5.1 Simulation code

```
% Machine Teaching for Robots project by Saitarun Nadipineni.
clear;close all;clc

robot   = 0;
fnDyn   = @(x, u)  fn_dynamics_2link0(x,u,robot);

x0 = [  pi/2;  %start state
       -pi/2;
           0;
           0];

xt = [ -pi/2; %target state
          pi;
           0;
           0];

L= [10 0 9 0;   %Gain Matrix L
    0  10 0 11];
```

Figure 5.1: Trajectory setup code

Figure 5.1 shows the code for the all the parameters that have been setup prior to their usage later. A function handle is created which accesses the 'fn_dynamics_2link0' function mentioned section in 4.1.1, this will make it easier to call the function locally. The start state $x_0 = (\pi/2, -\pi/2, 0, 0)^{\top}$ and the target state $x_t = (-\pi/2, \pi, 0, 0)^{\top}$ are chosen to follow to desired trajectory. The target skill $\theta^* = L = \begin{bmatrix} 10 & 0 & 9 & 0 \\ 0 & 10 & 0 & 11 \end{bmatrix}$ has been setup with the chosen values produce the most stable trajectory through testing. The sampling rate $500kHz$ and the total number of time steps S over the trajectory time 0.5 seconds is 10,000, which also the number of samples.

The remaining code and two-link arm model remain the same from experiment 1. 250 trials are conducted in this experiment where new noise added after each trial. The noise levels used are $\sigma \in \{0.05, 0.1, 0.15\}$.This experiment aims to assess how teaching a different skill can affect the results.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

5.2 Results

5.2.1 Target Trajectory

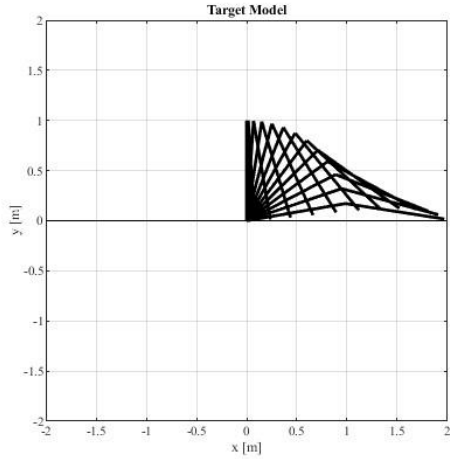


Figure 5.2: Target Trajectory

Figure 5.2 shows the target trajectory followed by the two-link arm using the PD controller with the target skill $\theta^* = L = \begin{bmatrix} 10 & 0 & 9 & 0 \\ 0 & 10 & 0 & 11 \end{bmatrix}$. The arm starts from the intial position and comes to rest when it reaches the target postion, the trajectory is plotted using the 'stroboscopic' function which takes angular displacements of both links at each time step as the input.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

5.2.2 Sample Trajectory

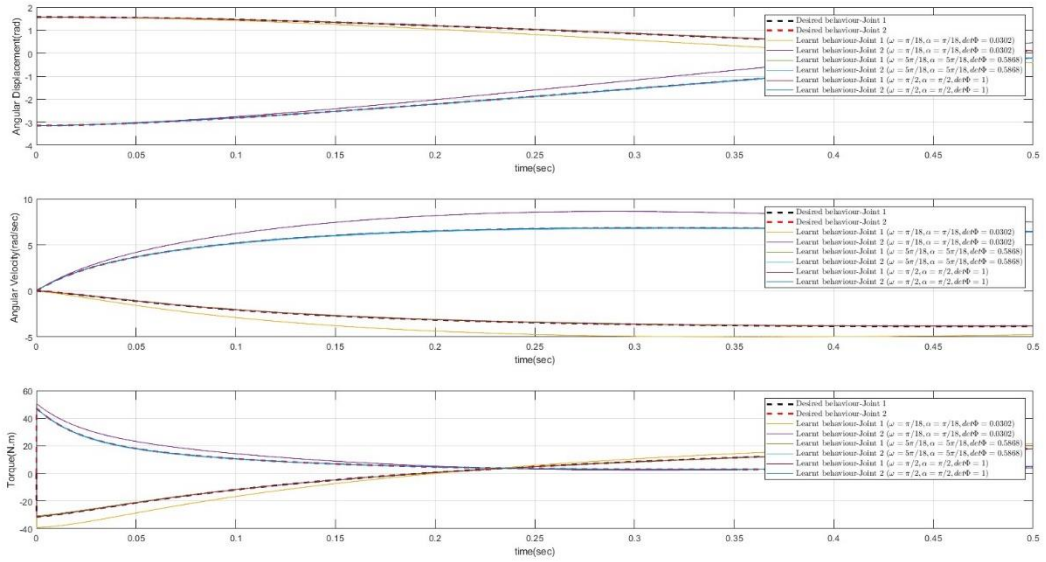


Figure 5.4: Sample Trajectories

Figure 5.4 shows the sample trajectories reproduced for experiment 2 by using the learnt model $\tilde{\theta}$ when from the data generated using $\omega = \alpha = \frac{\pi}{36}$, $\omega = \alpha = \frac{\pi}{18}$ and $\omega = \alpha = \frac{\pi}{2}$, the noise level used is $\sigma = 0.15$.

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

5.2.3 Error Analysis

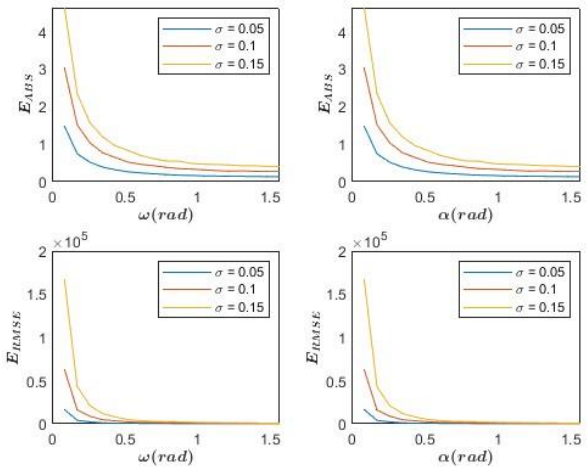


Figure 5.5: Error Results

Figure 5.5 shows the error in behaviour reproduced by the learner and difference between the target model and the learn model. The results recorded in this experiment was for 250 trials at three different noise levels $\sigma \epsilon \{0.05,0.1,0.15\}$. The root mean squared error E_{RMSE} and the absolute error E_{ABS} are plotted against the angles between the feature vectors ω and α. It can be seen in the figure, for all the noise levels as ω and α approach $\frac{\pi}{2} rad$ (i.e. when the determinant of $\mathbf{\Phi}$ is maximised) the error in the reproduced trajectories and the learnt model rapidly drop. Figure 5.4 shows the sample trajectories reproduced by using the learnt model $\tilde{\theta}$ when from the data generated using $\omega = \alpha = \frac{\pi}{36}$, $\omega = \alpha = \frac{\pi}{18}$ and $\omega = \alpha = \frac{\pi}{2}$, the reproduced trajectory almost perfectly overlaps with the desired trajectory when $\omega = \alpha = \frac{\pi}{2}$. The suboptimal choices for ω and α produce trajectories that have lower accuracy. As expected, the error is much greater when the noise added is higher. The results obtained in this experiment also support the optimality condition defined in section 2.4.1. However, the error in the learnt trajectory and learnt model is greater when learning the second skill.

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

6 Conclusion

The aim of this project is to improve the quality of training that is given to robots so that they can learn to perform useful motor skills with high accuracy and less resources. This report works on teaching useful motor skills to a two-link robot arm and assessing how its learning performance varies between suboptimal training data and optimal training data.

The work in this report is based on the work done by [1], however the motor skills taught to the pendulum in that research were two types of oscillations, many humans teachers may not benefit from teaching a robot these skills. The pendulum used in that research is a rudimentary robot model and has very little practical use in the real world.

For the experiments that were conducted in this paper the selected robot is a two-link arm and the skills that were taught to it are following a fixed trajectory and to reach out to certain point from a folded position. Both the motor skills taught are useful skills that humans can benefit from teaching a robot. The two-link robot arm is a much more practical and versatile robot for performing complex tasks compared the simple pendulum used in [1]. This was the aim set for this work in the future from the research conclusion of [1].

The aim of experiment 1 was to teach the learner (i.e., two-link arm) a fixed trajectory from the starting point and must come rest when it reaches the target point. In this experiment the results show that giving the learner optimal training data (i.e., data that meets the optimality condition defined in section 2.4.1) rapidly reduced the error in learning the target trajectory and in the learning the target model rapidly.

The aim of experiment 2 was to teach the learner how to reach out from a folded position to the and come to rest when it reaches the target point. In this experiment the results also showed that giving the learner optimal training data the error in learning the target trajectory and the learnt model rapidly decreases. However, in this experiment showed that teaching this skill resulted in greater error in the learnt trajectory and learnt model. This is due to this skill being much more complex as the arm's motion is less progressive in this experiment.

The limitation faced in both experiments is the number of trials and the number of samples that could be collected for each angle between the feature vectors. This is due to the implementation of the 'for loop' used for the PD controller in MATLAB. The PD controller needs to update the current state for the next iteration, this meant the loop couldn't be optimised further by using other MATLAB toolboxes [10] and the number of iterations needed to be higher if more samples of the learnt trajectory needed to be collected. This automatically limited the number of trials conducted in total as the execution time for the code will be much longer for three different noise levels. Therefore, the samples collected, and the trials conducted in both experiments are much lower compared to the 500 trials and 30,000 samples in the experiments conducted in

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

[1]. This is the reason the results obtained in this paper had more anomalous data points. This is also the reason that the experiments in this paper had approximately 50% more learning error compared to [1] as only half of the trials were conducted.

Future work can be done to implement the PD controller in MATLAB so that it takes less time to execute the code, this way it can be used for storing larger samples. This way the error in the learnt trajectories and models can be analysed much more accurately.

Future work can be done to implement an interface like the one in [1] where humans can demonstrate the skill through the interface and the robot must reproduce the skill using the learning algorithm. The teaching given by novice users can further be analysed and optimal teaching methods can be further explored.

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

7 Professionalism and Responsibility

7.1 Benefits of this project

This project is beneficial for novice users who wish to teach useful motor skills to robots which have functional and practical uses in the real world. This will benefit them for of tasks that require a lot of effort or are time consuming can be easily automated. The motor skills can be taught through demonstrations which will save a significant amount of time and resources. Teaching through demonstrations is also a very intuitive technique for novice users to use as very little technical knowledge is required. This project proves that this is a promising approach to take for teaching as the accuracy that the robot learns the skill is very high.

7.2 Positive Impacts

A positive impact that this project can have been that robots can be much more versatile at performing certain tasks and they don't require specific programming for performing a certain task. This has been proven in this project as two useful motor skills were taught to one two-link arm model and only the parameters of the skill being taught needed to be changed. In both experiments the learner reproduced the target skill with very high accuracy as better training data was given to it.

7.3 Possible negative Impacts

Though robots in general can be much more versatile as they can perform more tasks, they do face limitations in terms of their hardware properties. For example, a two-degree of freedom robot arm cannot perform a task that requires four degrees of freedom to complete the same task. Another, negative impact this project can have is choosing the interface which the humans can use to demonstrate the skills as the software and hardware required to provide the demonstrations can be expensive to create [12]. Implementing the software for different types of robots can require a lot of time and resources. Though this project in specific may not have any potential injury risks to humans directly, further practical testing may lead to it. Ultimately direct human and robot interaction is unavoidable whilst testing or teaching. This can put the safety of the human interacting with robot at risk if the robot malfunctions and causes an injury [13]. Humans may face risks of losing their jobs and face potential problems where their own intuition may become dull due to the fact simple tasks become automated [11].

Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

8 Bibliography

[1] Marina Y. Aoyama and Mathew Howard (2021). Training Humans to Train Robots Dynamic Motor Skills. Available at: https://arxiv.org/pdf/2104.08631.pdf

[2] Electronics Tutorials. Closed-loop Systems. Available at: https://www.electronics-tutorials.ws/systems/closed-loop-system.html

[3] Java T Point. Overfitting and Underfitting in Machine Learning. Available at: https://www.javatpoint.com/overfitting-and-underfitting-in-machine-learning

[4] Dataaspirant. Ridge Regression. Available at: https://dataaspirant.com/ridge-regression/

[5] Sheng Dong, Zhaohui Yuan and Fuli Zhang (2019). A Simplified Method for Dynamic Equation of Robot in Generalized Coordinate System. Available at: https://iopscience.iop.org/article/10.1088/1742-6596/1345/4/042077/pdf

[6] LibreTexts Engineering. 9.2.4.3: Proportional-Derivative (PD) Control. Available at: https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Chemical_Process_Dynamics_and_Controls_(Woolf)/09%3A_Proportional-Integral-Derivative_(PID)_Control/9.02%3A_P%2C_I%2C_D%2C_PI%2C_PD%2C_and_PID_control

[7] Dynamics and Modelling. Available at: https://slideplayer.com/slide/6997490/

[8] Ji Liu and Xiaojin Zhu (2016). The teaching Dimension of Linear learners. Available at: http://proceedings.mlr.press/v48/liua16.pdf

[9] C3.ai. Root Mean Squared Error (RMSE). Available at: https://c3.ai/glossary/data-science/root-mean-square-error-rmse/#:~:text=What%20is%20Root%20Mean%20Square,true%20values%20using%20Euclidean%20distance.

[10] MathWorks. parfor-Loops - Ensure that parfor-Loop Iterations are Independent. Available at: https://uk.mathworks.com/help/parallel-computing/ensure-that-parfor-loop-iterations-are-independent.html

[11] Jim Torresen (2018). A Review of Future and Ethical Perspectives of Robotics and AI. Available at: https://www.frontiersin.org/articles/10.3389/frobt.2017.00075/full

[12] Aude Billard and Daniel Grollam (2013), Scholarpedia, 8(12):3824. Robot Learning by demonstration. Available at: http://dx.doi.org/10.4249/scholarpedia.3824

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

Bush House, King's College
London, Strand Campus, 30
Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

[13] Askarpour, M., Mandrioli, D., Rossi, M., Vicentini, F. (2016). SAFER-HRC: Safety Analysis Through Formal vERification in Human-Robot Collaboration. In: Skavhaug, A., Guiochet, J., Bitsch, F. (eds) Computer Safety, Reliability, and Security. SAFECOMP 2016. Lecture Notes in Computer Science(), vol 9922. Springer, Cham. https://doi.org/10.1007/978-3-319-45477-1_2

Faculty of Natural, Mathematical and Engineering Sciences
Department of Engineering

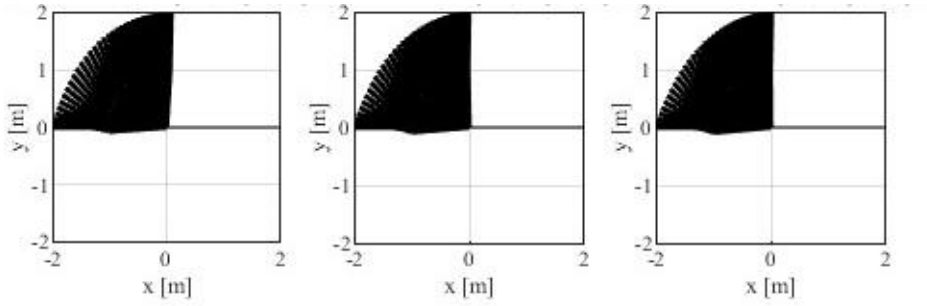
Bush House, King's College London, Strand Campus, 30 Aldwych, London WC2B 4BG

Telephone 020 7848 2145
Fax 020 7848 2851

KING'S College LONDON

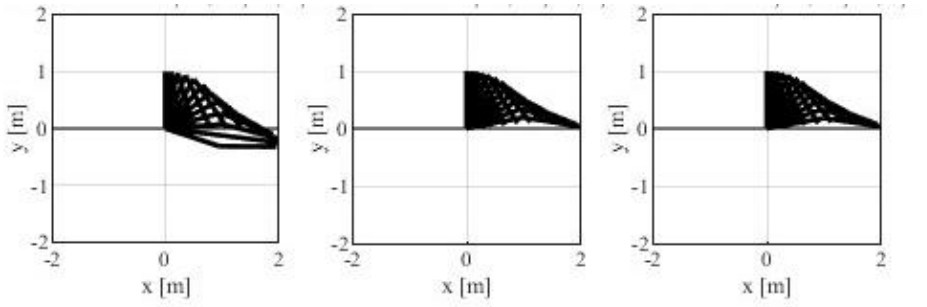
9 Appendix

9.1 Experiment 1 random trajectories reproduced:



$\omega = \frac{\pi}{36}$ and $\alpha = \frac{\pi}{36}$ $\omega = \frac{\pi}{18}$ and $\alpha = \frac{\pi}{18}$ $\omega = \frac{\pi}{2}$ and $\alpha = \frac{\pi}{2}$

9.2 Experiment 2 random trajectories reproduced:



$\omega = \frac{\pi}{36}$ and $\alpha = \frac{\pi}{36}$ $\omega = \frac{\pi}{18}$ and $\alpha = \frac{\pi}{18}$ $\omega = \frac{\pi}{2}$ and $\alpha = \frac{\pi}{2}$