OXFORD
BROOKES
UNIVERSITY

Jamie Pierce, BSc

# Investigating how the morphology of a robot arm affects the learnability of aimed throwing

**Master Thesis**

submitted in partial fulfilment of the requirements of the award of

Master of Science by Research

submitted to

School of Engineering, Computing and Mathematics
**Oxford Brookes University**

Supervisors

Dr Matthias Rolf, Dr Tjeerd olde Scheper

Oxford, March 2020

# Abstract

This paper investigates the relationship between the hand design on a robot arm and the learnability of aimed throwing. Aimed throwing is selected as the task because of its importance as a milestone in human evolution, as well as being a task with an unambiguous outcome. The process of designing and building the arm, in addition to the full experimental setup, is described. By varying two different hand designs, the error rate and repeatability are compared to determine how to measure learnability. This metric is then used to compare the two morphologies. The findings of this are that the link between learnability and morphology is not as direct a link as previously believed, which leads to a separate experiment investigating how repeatability affects the learning algorithm within a test environment. This study finds that noise aids with exploration of the action space, however a more repeatable throw allows the algorithm to better learn the task.

# Contents

# 1 Introduction

Learning, in all disciplines related to it, is a very popular topic. This is mostly due to it providing a different perspective to a problem. By studying how well a task is learned, rather than studying how well it is performed after being learned for example, will help to better understand the task and why certain sub tasks are being carried out and why in that order.

The implementation of Machine Learning into systems permits the system to work independently as it no longer needs people working on it, using their own knowledge to adjust and correct any deviations from the desired function. As robots do not have pre-defined controllers, they are usually created and adjusted by the developers. To have the robots learn these themselves rather than need to have them hand crafted by the developer, would reduce the cost of the systems, allow constant improvement of performance and could succeed where traditional engineering methods often fail.

There have been many discoveries surrounding all aspects of learning algorithms and optimizing these algorithms, including in situations that involve physical parts of the system to be included in the learning loop. This includes studies such as (Farchy et al., 2013) investigating a robot learning to walk, and (Aboaf, Atkeson, and Reinkensmeyer, 1987) investigating a robot learning to throw a ball.

However, there is a lack of research into how these physical sides of the system affect the ease of learning of the model and how varying the physical morphology of the system might result in a difference in learning speed. Investigating not just the accuracy of the final model on completion of the learning process, but studying it throughout this process should be key to fully understanding the model as well as the task. Additionally, by being able to optimize the physical parts of a learning system, to make it learn faster or achieve a better final accuracy for example, to tailor it to

its application, could see an enormous increase in efficiency of learning implementations.

This project aims to investigate the effect of this varying physical morphology, in this particular case the varying hand design, in an aimed throwing scenario. Aimed throwing was a milestone in human evolution as it was the first representation of killing from a distance (Von Hippel and Nussbaum, 2019), and lead on to huge growth in brain capabilities (Calvin, 1983). Investigating whether the human body has evolved in such a way that made this, clearly crucial, talent easier to learn will add a different perspective to the current body of knowledge in this area.

Machine learning is a method of applying an algorithm to computer programs to allow them to optimize or sometimes write themselves (Brownlee, 2015). Where traditional programming is described as passing data and a program to an algorithm to achieve an output, from a high-level perspective, a generalised machine learning process is described as an algorithm learning from data and an outcome in order to produce a program.

Within this, are different types of learning algorithms: unsupervised, semi-supervised and supervised learning as well as reinforcement learning. Supervised learning is learning that knows the desired outcomes while it is learning from the training data, whereas unsupervised does not. Intuitively, semi-supervised is the area in between, where the learning model knows some of the desired outcomes of the training data. Reinforcement learning, however, is different from the other three in that it will carry out an action and be given some reward dependent on its performance. As will be detailed in Chapter 2, the most appropriate form of learning for this study, as the target distances will be known by the algorithm, is supervised learning due to the nature of the experiment, with the self-generated data from the throws being used to train the model.

## 1.1 Motivation

The motivation for this study comes from a combination of a fetus, seemingly impossible, ability to grasp objects within 6 months of conception and the theorized importance of the early humans' ability to throw accurately. The human requires a high degree of coordination to move in a purposeful manner. The many degrees of freedom that the body has, should require years of learning to be able to achieve goals such as grasping. However, preterm infants have been repeatedly observed grasping the umbilical cord as early as 25 weeks of post-conceptional age (Futagi, Toribe, and Suzuki, 2012). This ability, known as the Palmar Reflex, allows infants to bypass this learning process to be able to carry out a task that is a crucial building block for other complex tasks. This discovery begged the question, if this happened for grasping, is it possible that it happened for other crucial human skills?

Due to the training data of the learning model being self-generated, the exploration of possible actions becomes critical to the effectiveness of the learned model. Goal babbling is a method of exploration that mimics the goal-directed movements seen in infants rudimentary reaching skills that can be seen four months after birth (Thelen, Corbetta, and Spencer, 1996). By imitating this efficient bootstrapping, it is possible to yield useful results even without fully exploring the space of possible motor commands.

As well as being a crucial human skill, aimed throwing is a task that is very well suited for machine learning investigations due to its straight-forward and clear outcome. It is very prototypical for other problems of sensorimotor coordination in that given a certain behavioral goal, how is it achieved by some action(s). In the case of this study, this becomes a question of how a desired target is hit by means of certain movements and motor commands. This task allows for a simple outcome and therefore an uncomplicated comparison between different configurations is possible.

Learnability is a characteristic that describes how easily and accurately something can be learned. Learning is mostly assessed by the success rate of the finished model, irrelevant of the number of attempts or data learned from. A learning model should also be assessed on its path or journey to this desired success rate. By having this extra assessor, allows

for better comparison between models adding a different perspective, such as how efficiently a model gets to the goal success rate. How to assess this learnability characteristic will be investigated throughout this study, determining different methods of evaluation of the models.

This paper aims to investigate and evaluate how varying the morphology of the arm of a throwing robot will affect the learnability of aimed throwing. By comparing the varying combinations of morphology and exploration method, the final error rate, the variability and the speed of learning will be assessed in order to determine the learnability. The hypothesis for this study is that, if a morphology is able to increase the consistency of the throws, it will increase the learnability of aimed throwing.

## 1.2 Scope and Outline

To carry out this study, it was decided to use a physical robot for the system rather than use a simulation. This was due to multiple reasons. As it is linked to the investigation of human evolution, and the role aimed throwing played in it, it was important for this study to keep the scenarios similar. By using a physical robot, it exposes the system to unexpected variables and unknowns that might not have been present in a simulation. This process is detailed in Chapter 3.

By implementing machine learning into this study, as well as allowing the mimicry of the brain in human evolution, allows the system to be setup much easier, as every eventuality does not have to be explored and evaluated manually. Likewise, using existing framework aids with this, which resulted in the use of an existing learning library as well as exploration method. The goal-babbling exploration, explained in Chapter 2, allows for more goal-oriented exploration of the action space, compared to random exploration. This speeds up learning in high-dimensional situations. The implementation of this is also covered in Chapter 3

Using this implementation of the machine learning, the physical structure of the robot arm was varied to investigate the effect this had on the learnability of the model. The process of how this investigation is carried out is detailed

in Chapter 4. One of the key objectives of this study was to explore methods of measuring this learnability. Consequently, this had to be evaluated first using the results of the experiment, before the different morphologies could be compared. This is discussed in Chapter 5. Finally, the findings of the study are summarised in 7.

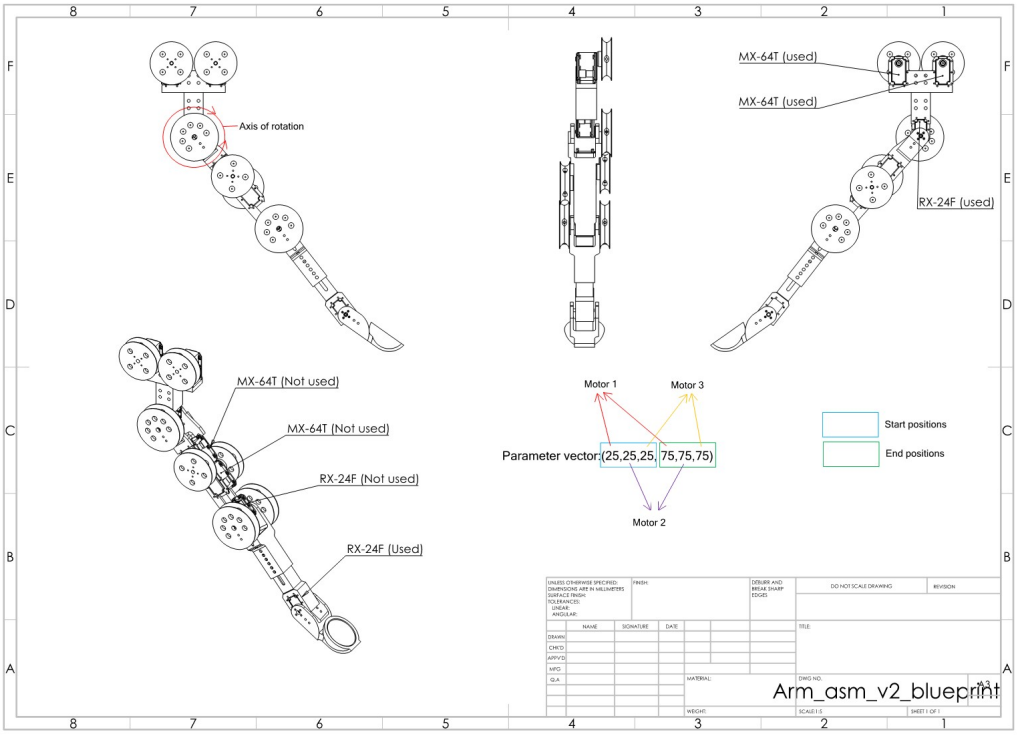


Figure 1.1: An image showing the different angles of the Solidworks model of the arm.

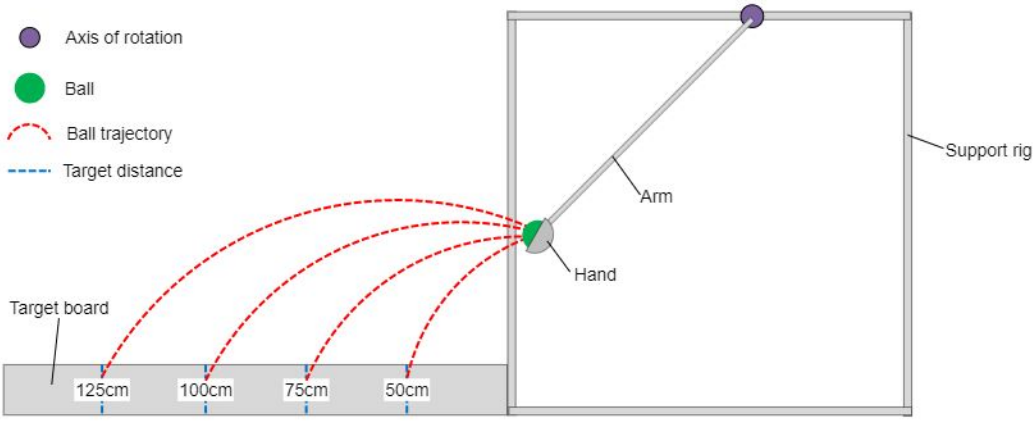


Figure 1.2: An image showing the experimental setup.

# 2 Literature Review

## 2.1 Human Throwing

This section aims to introduce the importance of aimed throwing in human evolution. It will cover why the ability to throw was so important in our survival at that time, but also the knock-on effects that it had on our evolution and put us on this path that has lead us to the intelligence of modern human life.

From a very young age, humans have the ability to grasp things in their hands. This is regularly seen in newborn babies, when their palm is stroked, they will instinctively close their hand. This palmar reflex has also been seen as early as 25 weeks of postconceptional age (Futagi, Toribe, and Suzuki, 2012). It was thought that this complex movement should not be possible so early in human development due to the number of muscles that would have to be used to individually close each finger. The baby has not had the time to explore the outcome of activating each of these muscles, so this should not be possible. However, this ability is possible due to a tendon in our forearm that runs through our carpal tunnel and to the tips of each of our fingers, apart from our thumbs. This tendon is known as the flexor digitorum profundus (Wheeless, Nunley, and Urbaniak, 2016), and it's insertion past the last joint of our fingers allows it to bend all three finger joints to close the hand. The development of this evolutionary trait allows humans to learn how to grasp much quicker. This ease of learnability of what is widely accepted as a hugely advantageous ability, allowing us to use our fingers and opposable thumbs to grasp tools, could have been a key component to humans' success in survival.

Another ability that humans evolved to have in their arsenal was the ability to throw, something that is thought to have had an immense effect on the

growth of the Homo erectus brain (Calvin, 1982). The structure of the human shoulder and arm has been extensively studied and is widely accepted to have evolved to allow our ancestors to be able to hunt and protect themselves and as Neil T. Roach, PhD, a fellow of human evolutionary biology at Harvard University believes, "turning our species into the most dominant predators on earth" (Potter, 2019). While throwing may seem trivial, especially the idea of attempting to fend off the attack of a predator by throwing rocks at them, William von Hippel, PhD, believes that it is "probably the single most important military invention in history" (Von Hippel and Nussbaum, 2019). In the same interview, he goes on to explain that this is because being able to throw objects represents the capacity to kill at a distance, allowing a larger group of weaker individuals to much more successfully take on a smaller group of stronger individuals. However, this is not just an advantage against stronger animals, but any animals. This kill-from-a-distance capacity likely also helped to be able to throw at the stationery animal from a distance at which the animal would tolerate the presence of a hominid (Calvin, 1983). The energy that before was going to muscles all over the body to provide the strength to be able to wrestle the animal being hunted, could now be redirected to uses in other places, like the brain, which is argued to be one of the enabling factors in the Homo erectus brain growth.

(Calvin, 1983) argues that this brain growth was also as a result of the unusual timing needed in accurate spear or rock throwing. This timing accuracy was beyond the known accuracy of single neurons and so Calvin posits that the only known solution to the problem, following the Law of Large Numbers, requires a larger number of neurons applied to the task. The two means of doing this would be to have an incredibly large increase in brain size or to temporarily borrow these neurons from elsewhere in the brain once a throwing movement is started. The latter method, as Calvin argues, has huge implications on not only brain size but also brain reorganisation. It has been hypothesised that this motor sequencing neural machinery has additional secondary uses; in particular, to provide a possible foundation for language specialisations (Ojemann, 1982).

As mentioned, humans' survival was now more dependent on their ability to work in groups than ever due to the success of this newfound power of accurate throwing. For these groups to be as effective as possible, communi-

cation became crucial as information would need to be passed to each other during group hunts of large game. This, coupled with the increased brain size and neural reorganisation, is why it is theorised that the discovery of throwing was crucial to the evolution of human language.

This all begs the question that if the ability to throw was so important to our survival, is it possible that we also evolved to be able to learn this crucial skill easier and therefore earlier in our lives? Is it plausible that there was, similar to how an algorithm is designed for a particular platform in robotics, a co-evolution of the brain and arm, optimising one for the other and vice-versa?

## 2.2 Hardware Components

This section discusses the decisions made for each of the hardware components for the physical arm. This is done by exploring what platforms previous studies have used, and also what other robots are available and appropriate for this study.

(Kober, Glisson, and Mistry, 2012) investigates playing catch and juggling with a Disney A100 Audio-Animatronics hydraulic humanoid robot. The throwing component of both juggling and playing catch, make it interesting to investigate for this project. Due to the targeted use of the robot being in an amusement park, they used a humanoid robot that would already be present at the park. It uses a camera to detect the balls and position the hand in the correct place for the predicted trajectory of the ball. The ball is thrown back two and a half metres for the user to catch. This robot uses hydraulics to move the joints. Controlling these hydraulics would have added unnecessary complexities to the system that were out of the scope. Consequently, using this robot as a platform would not be viable for this study.

(Kim and Doncieux, 2017) uses Rethink Robotics' Baxter robot to explore "Learning Highly Diverse Robot Throwing Movements through Quality Diversity Search". The throwing motions in this study are high-dimensional due to the 7 DOF of the robot, increasing the cost of sample generation

but allowing a wider dimension of exploration. Additionally, the physical configuration of this robot cannot be easily modified or prototyped, ruling out it's use due to the varying of morphology being the key concept of this study. Although this robot would be readily accessible for the study, and it having existing framework and control to build from, the lack of ability to modify its morphology means that it is not viable for this investigation.

A project from Plymouth University, (ROCO504, 2017), investigated the elastic energy storage of different materials in a ball throwing scenario. It's structure attempted to replicate the elastic energy storage of tendons in a human arm, presenting a number of appropriate elastics to use. The CAD designs for this arm are open-source so can be used and modified to tailor them to the needs of this study. This project also contained proof that the arm was able to throw a ball, meaning that it was a definite platform that could be built from because of this proof of concept. The findings of (ROCO504, 2017) could provide a good base for this study to build from.

## 2.3 Robotic learning of throwing

This section will cover previous studies that have investigated machine learning in robots, specifically involved with aimed throwing. These studies will be discussed in order to find the most suitable method of implementing machine learning into this project.

There has been extensive research into learning algorithms and methods, especially into a task such as throwing a ball. Throwing a ball, commonly aimed, is a task often selected for machine learning investigations due to its non-trivial nature and its obvious method of error evaluation. By carrying out an aimed throwing task, comparing the target and resulting throw allows for a clear assessment of the performance of the system.

Machine learning is an application of algorithms and processes that provides the system with the ability to learn and optimise a performance from past experience and/or example data without explicit programming. One popular approach is to calibrate the models that describe a robot's lower level systems – dynamics, kinematics and perception, as described in (Aboaf,

Atkeson, and Reinkensmeyer, 1987). Aboaf et al. investigate a different approach, allowing the robot system as a whole to practice the desired task and learn from those experiences. After each attempt, a task-level learning procedure monitors the result of the practiced task and evaluates the error in the performance. Applying learning at a task level mitigates the need to perfect the structural modelling of the robot's lower level systems, compensating for any errors. This allows the robot to learn the result of performing a command and monitoring the outcome of its performance and improving it without focusing on each lower level system.

(Rolf, 2012) explains that traditional approaches to the computational learning of coordination skills rely on an exhaustive exploration of possible actions, which is not feasible in high dimensions. As previously mentioned, the ability to grasp is one that is too high-dimensional to be fully explored from scratch within months of life. The ability of reaching is similarly in this fashion. However, they display rudimentary reaching skills already three months after birth (Thelen, Corbetta, and Spencer, 1996), showing enormous efficiency when bootstrapping their range of sensorimotor skills. (Rolf, 2012) sets out this view and introduces a concept called goal babbling, a goal-directed method of exploration for high dimensional domains that mimics the efficient exploratory movements performed by infants. It is shown that this approach allows for a bootstrapping that scales almost constantly with respect to the dimension of the action space, which is opposed to the exponential cost of exhaustive exploration. Goal babbling removes a majority of exploration, that is regularly carried out by an exhaustive method, of areas that are not necessarily important to the task, focusing the bulk of it on the ranges around the pre-determined targets. This newly introduced exploration method will be compared with an exhaustive random exploration to investigate how the changing of the morphology of the throwing arm affects the learnability of both methods. This will not only investigate the impact of the physical side to learnability but also allow for a further comparison of goal babbling against random exploration.

Reinforcement learning has also been explored in relation to aimed throwing tasks, including the use of paramaterised movement primitives. (Kober, Wilhelm, et al., 2012) investigates this type of algorithm within similar task of throwing movements in darts, of hitting movements in table tennis, and of throwing balls. This approach proposed a method that learns to

generalise parameterised motor programmes by adaping a small set of global parameters. This is done by employing reinforcement learning to learn the required parameters to deal with the desired situation. Due to the time costs of experience in this scenario, reinforcement learning was implemented to try to speed up the learning process by using a generalised teacher's demonstration. This is not necessary for this project as, although expensive, the sample generation for this scenario is viable.

(Kober and Peters, 2011) details the ball throwing side of (Kober, Wilhelm, et al., 2012). In this, Kober investigates the effect of implementing hierarchical learning, a type of learning where different landing locations result in different points, like a dart board. This is a novel problem in learning as improving the individual throws corresponds to a lower level of learning, however the game strategy to a higher level. The approach in this paper is to use a hierarchy of two levels: a strategy level and a behaviour level. The strategy level decides the next behaviour to be executed, and the behaviour level executes it. Due to the scope of this project, investigating the strategy level is not necessary. However, how the robot learns the behavioural level is. For this, (Kober and Peters, 2011) uses a Cost-regularized Kernel Regression, detailed in (Kober, Oztop, and Peters, 2011). This is a viable method that could be implemented into this project as the learning method.

To reduce the need to implement new frameworks, existing open-source frameworks were investigated. (Moulin-Frier, Rouanet, and Oudeyer, 2014) is a probabilistic framework unifying two exploration mechanisms: active learning driven by the maximization of empirically measured learning progress and (Rolf, 2012)'s notion of goal babbling. This paper explains how these exploration mechanisms are efficient in learning complex non-linear redundant sensorimotor mappings. For example, learning how arm movements make physical objects move. These methods, compared to random motor babbling, are much more efficient exploration strategies especially at high dimensional motor spaces. As this open-source library has already implemented the goal babbling exploration strategy, it is the most appropriate method to implement for this study.

As previously mentioned, there has been many studies into learning, comparing different learning algorithms and methods. However, there is a lack of research investigating how the physical components of a system affect

these algorithms. In particular, how changing these physical components change the learnability of the model. Could the performance of a task by learning systems be improved by not only modifying the command input to the model but also modifying the physical system components? This study aims to investigate this notion, using an aimed throwing task and a task-level learning approach.

## 2.4 Summary

| Literature reviewed | How it informed/motivated the work |
|---|---|
| (Futagi, Toribe, and Suzuki, 2012) | Used for motivation. Informing the study of the palmar reflex and it's early observation in newborn babies, |
| (Wheeless, Nunley, and Urbaniak, 2016) | Used for motivation. Informed the study of the tendons in the arm/hand, particularly the digitorium profundus, the tendon responsible for easy grasping |
| (Calvin, 1982)/(Potter, 2019)/(Von Hippel, 2018)/(Calvin, 1983)/(Ojemann, 1982) | Used for motivation. Informed the study of importance of aimed throwing in human evolution. |
| (Kober, Glisson, and Mistry, 2012) | Informs the study of a previous throwing study, "Playing catch and juggling with a humanoid robot". Used to investigate previous hardware used in studies. |
| (Kim and Doncieux, 2017) | Informs the study of a previous throwing study, "Learning Highly Diverse Robot Throwing Movements through Quality Diversity Search". Used to investigate previous hardware used in studies. |
| (ROCO504, 2017) | Informs the study of a previous throwing study, "Throwing arm with elastic energy storage". Used the designs in this as a base for the design of the arm used for this project. |
| (Paraschos et al., 2018) | Informs the study of the basics of movement primitives |
| (Nemec and Ude, 2012) | Informs the study of the issues with traditional movement primitives, if they're not made dynamic and continuous. This paper contributes new methodologies to try to solve this. |
| (Aboaf, Atkeson, and Reinkensmeyer, 1987) | Informs the study of a popular learning approach, called "Task-level learning" |
| (Rolf, 2012) | Informs the study of a goal-oriented exploration strategy called goal-babbling |
| (Kober, Wilhelm, et al., 2012) | A previous aimed throwing study, using reinforcement learning with parameterised movement primitives. |
| (Kober, Oztop, and Peters, 2011)/(Kober and Peters, 2011) | Informs the study of the possible use of Cost-regularised Kernel Regression for this study |
| (Moulin-Frier, Rouanet, and Oudeyer, 2014) | Provides the framework for the machine learning components of this study. |

Table 2.1: A summary of the literature reviewed and how it informed or motivated the work

# 3 Robotic System

This chapter aims to cover the different components that make up the system, the different iterations that they went through, and how they were implemented into the system. This will start with the physical setup of the experiment including the designs of the arm and its components as well as the chosen target and ball being used. It will then cover the software implementations of the vision component and motor controllers and how it was all integrated into the system.

## 3.1 Robot Arm Overview

Within this section, the design of the arm will be outlined. This will include the decisions and iterations involved within the design of the arm, and the rationale behind each of these decisions.

The first decision to be made was the material to make the arm out of. An aluminium skeleton would allow for a stronger structure able to withstand more torque and be much more robust while also being very lightweight. This is due to it having the highest strength-to-weight ratio of any metal (Noble, Harris, and Dinsdale, 1982). This is crucial due to it being a prototype design as it would be exposed to this during experimentation and testing of the design. The lightweight characteristic of aluminium would also reduce the stress on joints and the actuators, while also allowing a faster movement. However, to order these bespoke aluminum pieces would be expensive, economically and temporally, at the start, as well as any time the arm needs to be redesigned or parts to be prototyped.

3D printing the parts for the arm would allow this ability to prototype and compare different designs relatively inexpensively. Although strength and

robustness would be slightly sacrificed for this capability, it would allow for a much wider array of morphology comparisons due to the less expensive method. The arm used in (ROCO504, 2017) had been designed using CAD and had been 3D printed. The structure of the arm was very similar to what was needed for this study, so with minimal number of alterations, it could be made suitable for the experiment. Due to the similarities, and the designs being open-source and modifiable, this allowed for the ongoing alterations needed throughout the study to prototype parts and investigate the outcome of changes to these parts.

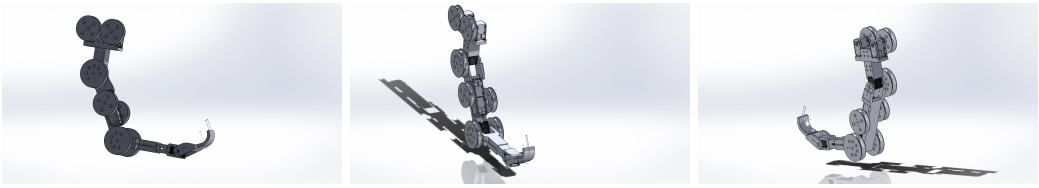

Figure 3.1: The initial design of the arm.

The initial design at the start of the study can be seen in Fig. 3.1. The pulleys on the side allow for artificial tendons to be attached. From the findings in (ROCO504, 2017), the material used for this elastic was a 3D print material called NinjaFlex as it was found to have the best effect of the materials that had been tested. These tendons aid the motors inside the joints, in addition to providing elastic tension that can be released during the throwing motion.

The majority of this design did not change throughout the study. The parts that did change were the hand and the shoulder section. These changes will be covered in detail later in this Chapter.

## 3.2 Throwing Setup

When measuring the distance that the ball reached, it was clear that decisions had to be made in regard to keeping the experiment fair and consistent. Obviously, due to their shape, balls will roll when thrown. This added an inconsistency to the measurement of results, while also making the

area needed for the experiment impractical. This resulted in iterations of apparatus that would reduce this effect, thereby making it much easier to measure whilst also obtaining fair results. The first iteration was to use sand. This would ensure that when the ball landed, the energy was transferred to the sand making it stop almost immediately. However, this method was quickly side-lined as it was not very viable. The cost and lack of mobility of having an area of sand attached to the system was unsuitable. The use of objects filled with different materials was the next iteration. These would have the same effect as the sand, of taking the energy out of the ball, without the need for a impractical target area. The two different types of objects were bean bags and standard juggling balls. The mean bag was unfortunately ruled out due to its shape and therefore was inconsistent and gave imbalanced results depending on its positioning in the hand. This bag also had a slight slide when landing. The juggling ball was also not viable. Although it was spherical and therefore would be more consistent when leaving the hand, it rolled when landing slightly. The chosen iteration for the system was to use hook and loop material, or more commonly known as Velcro. The loops of the fabric on the ball would be hooked by the material adhered to a target board. This set-up ensured more fair and consistent results as the thrown object was spherical and the ball stopped almost immediately on landing. This system was cheap to implement and could also be made very mobile. By attaching the hook material to a 2m target board with a hinge in the centre, the board would have more than enough length for the experiment. The hinge in the board allowed it to be folded in half for storage and movement. A drawback of this system is that each time the ball is pulled off of the target board, the hooks rip the loops on the ball. This causes the balls to lose some of their grip to the target. This effect can be seen in Fig. 3.2. This effect can be managed by switching out the old ball with a new one at predetermined milestones of the project.

The result of this was the creation of a system that allowed a ball to land and consistently stop at the point of contact. This reduced the inconsistency involved with measure aimed throwing of the ball bouncing or rolling different amounts each time. This also reduced the space needed to carry out the experiment, while keeping the cost of the setup down.

Figure 3.2: A figure showing the ball's felt after being used.

## 3.3 Shoulder Design

The shoulder joint is the most influential joint on the power transferred in a throwing motion, as has been mentioned in Fig. 2. Consequently, this part of the arm had to go through significant design alterations. These will be covered in this section.

The prototypical nature of the arm meant that it was inevitable that it would go through many different iterations. The starting configuration can be seen in Fig. 3.1. To do this, it was separated from the rest of the system and the throwing motion was hard-coded, being altered slightly manually to try to find an estimate for the furthest possible throw. Initially, this was done by comparing the ideal release angle to that of a canon, this angle being 45 degrees. However, this is incorrect as it is a swinging pendulum and not a stationary canon. The correct equation can be seen in Fig. 3.3, that calculates the distance travelled, using the pull-back angle,$\theta_1$, the release angle $\theta_2$ and the length of the pendulum $L$.

$$x = L\sin(2\theta_2)(\cos\theta_2 - \cos\theta_1)\left[1 + \sqrt{1 + \frac{1-\cos\theta_2}{\sin^2\theta_2(\cos\theta_2 - \cos\theta_1)}}\right]$$

Figure 3.3: The equation to find the distance travelled using the pull back angle and release angle of the ball. (Anson, 2017)

Rearranging this equation, the optimal angle of release for this arm was approximately 30 degrees. Even with this knowledge, the maximum distance achieved was approximately 50cm. This distance was not suitable for the experiment as it would not allow enough variation in throws. Through

further investigation, it was found that the pulleys at the top were not running at a sufficient speed, slowing the movement of the arm down significantly. With the arm disconnected from power, moving purely under the force of gravity, the ball was being thrown further. This highlighted the issue. (*MX-64AR, MX-64AT (Protocol 2.0)* n.d.) shows that the centre motor in the pivot of the shoulder can run at a maximum speed of 12-13 rad/sec. This speed was being achieved under freefall. The motors at the top, with no torque, can run at only half of this at 6.7 rad/sec. This is shown in Fig. 3.4, where 64 RPM is equivalent to 6.7 rad/sec. Compounding this issue, due to a design issue that had been overlooked, the pulleys at the top had a smaller diameter than the pulley at the pivot. This was effectively gearing down the pulleys that were already running too slow. It was clear that this was the main contributor to the lack of distance achievable.

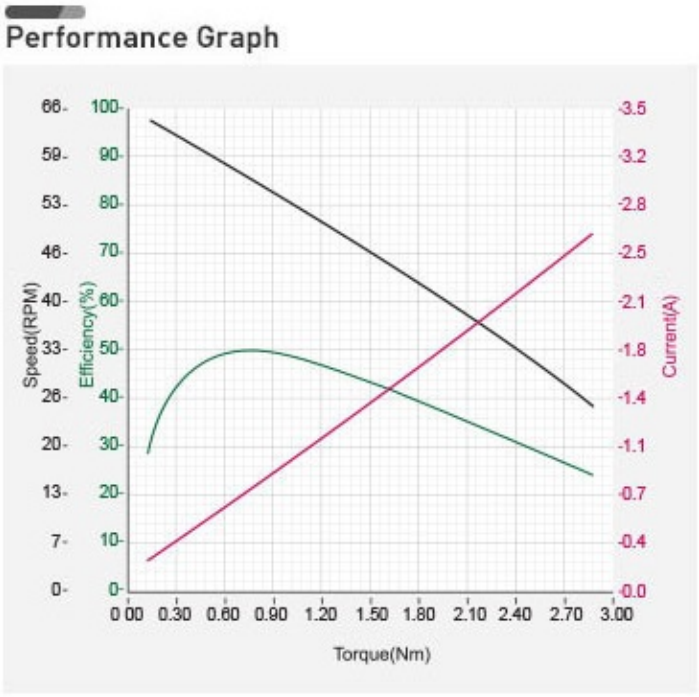


Figure 3.4: A graph showing the performance of the MX-64 motor used in this study. (*MX-64AR, MX-64AT (Protocol 2.0)* n.d.)

To rectify this, eccentric pulleys as well as increasing the diameters of the

two top pulleys were experimented with. As increasing the gear ratio was the less complicated method, it would be much easier to implement without the concern of further complications. This was achieved by increasing the diameter of the front pulley and decreasing the diameter of the centre pulley. This achieved a gear ratio of approximately 2.25:1. The back pulley was kept at the same diameter and was implemented as a supporting pulley, being connected to the front motor via a pulley of the same size. This can be seen in Fig. 3.5.

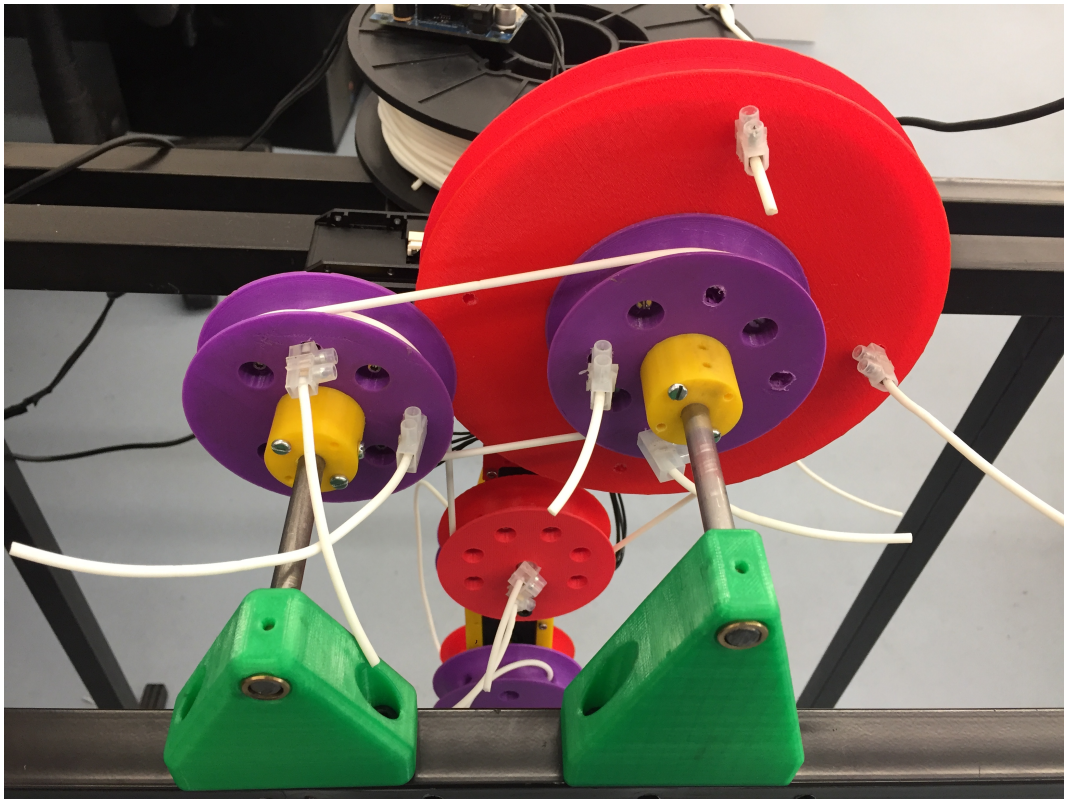

Figure 3.5: A centre view of the shoulder configuration.

This figure also shows the supports that were added at this point as well. These connect the motor to the frame, reducing the force on the bearings of the motors. These supports are connected in such a way that allows the metal shaft to spin with the pulleys, but the supports connected to the frame stop any angular shifts. The combination of these adjustments allowed the

ball to reach approximately 160cm using manually entered hard coded throws.

## 3.4 Hand Design

This section covers the process that resulted in the creation of the two designs of the hand that were to be compared.

Once the shoulder design was altered to allow for a much faster movement, the throws from this new setup were investigated to see if there were other issues hindering the throw. The result of this was the discovery that there was slight movement of the ball in the hand during the throwing motion. To combat this, another hand was designed that has a much higher support at the back, with a slight increase in the height at the front. The comparison of the two hands can be seen in Fig. 3.6. The aim of this was to investigate if this increased the repeatability of the throws, and if so, whether this helped improve the learnability or not.

By improving the repeatability of the throws, it is predicted that this will improve the learnability as throws with more consistent results should make it easier to learn from.

Figure 3.6: A front view of the two hands side by side.

## 3.5 Vision

To reduce the need for human input in the system, it was important to have an automated measuring method. The most appropriate method for this was to use computer vision to detect the ball and find the distance it had travelled.

Initially, the distance thrown was entered into the system via a keyboard input. However, this would prove to be too time consuming during the experiment. As a result, the process was automated using machine vision to detect the ball and determine the distance it had been thrown.

Figure 3.7: The tennis ball being used in this study.

The ball detection went through different iterations finding a method that was most effective for the scenario. To begin with a simple colour detection algorithm was implemented to distinguish the ball from its environment. This could be done due to the vibrant colour of the ball, as can be seen in 3.7. This meant that much of the noise of the background would be removed due to it being likely dissimilar. Although, once calibrated, it was effective, it needed calibrating quite often due to changing light levels having a large effect on the results.

To see if a different technique would be more effective, a background subtraction method was implemented. This method uses an image supplied to it to use as a "background". The algorithm then compares the current

Figure 3.8: The vision detecting of the ball, left, and the board, right.

camera view with the background and highlights any differences between these images. The implementation of this method was slightly less effective than the previous due to any small changes in the image giving false results. Subsequently, the two methods were combined to in an attempt to make the method much more robust. Although this was successful, a logic 'AND' was also implemented between the image of the board and the ball, as can be seen in Fig. 3.9. This would further remove noise by essentially removing any objects detected that were not within the target board, while also having a distinct difference in colour between the blackness of the board and the brightness of the ball.

The next essential feature of the vision node was determining the distance that the ball had been thrown. Similarly, to the detection, this also had different iterations. Initially, this was done using the size of the ball. With the real world size of the ball known, it could be compared to the measured size of the ball from the camera and calculated to find out how far the ball was from the camera. Yet, this was a rudimentary and primitive method. The inconsistency came mainly due to the ball changing appeared size through use as the fabric expanded, however this was not surprising as it was not a robust method.

To find a more consistent method, a calculation using Pythagoras' theorem was used, as can be seen in 3.10, showing the experimental setup. With the known height of the camera and its lens' angle of view the horizontal distance travelled could be calculated. This calculation can be seen in Eq. 3.1 where $\theta_{view}$ is the angle of view of the lens in radians, $\theta_{pixel}$ is the angle per pixel, $w$ is the width of the image brought into OpenCV, $x$ is the $x$

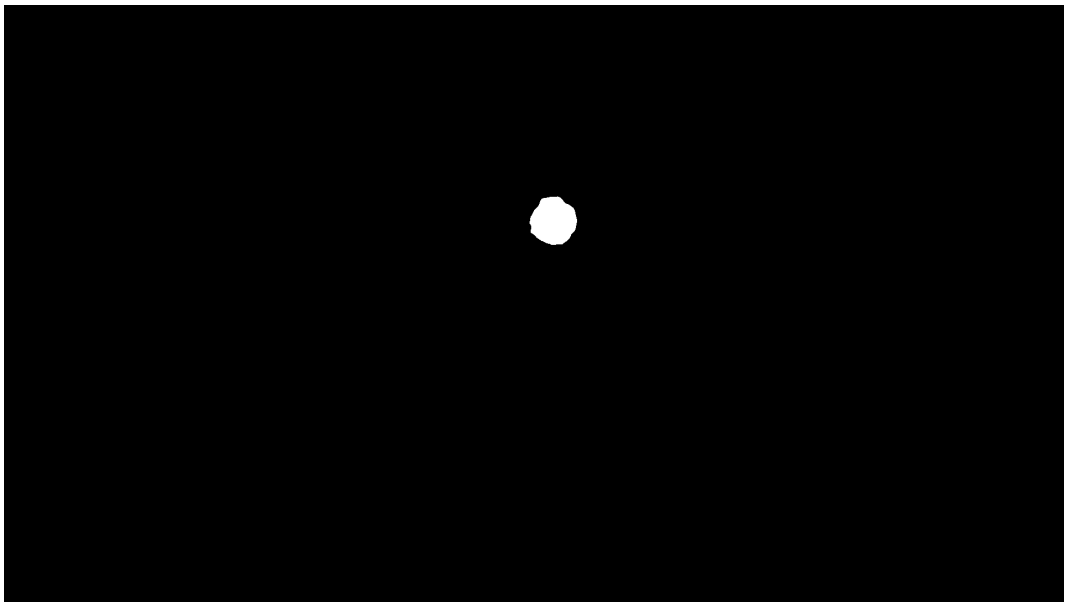

Figure 3.9: The outcome of the logic AND.

coordinate of centre of the ball detected in the image, $\delta$ is the distance the ball has travelled, $h$ is the vertical height that the camera is from the board and $\theta_{camera}$ is the angle of the camera to the vertical.

$$\theta_{pixel} = \frac{\theta_{view}}{w} \tag{3.1}$$

$$\theta_{offset} = \theta_{pixel}(x - \frac{w}{2}) \tag{3.2}$$

$$\delta = (h * \tan(\theta_{camera} + \theta_{offset})) - \delta_{offset} \tag{3.3}$$

By calculating the difference from the centre of the screen, it reduces any compounding error that might occur if measured from one side of the screen.

Through this method, the camera was able to be mounted to the supporting frame, as can be seen in 3.11. This allowed the setup to be much more compact if it needed to be moved, while also keeping a consistent positioning. This removed any issues with having to replace the camera in exactly the

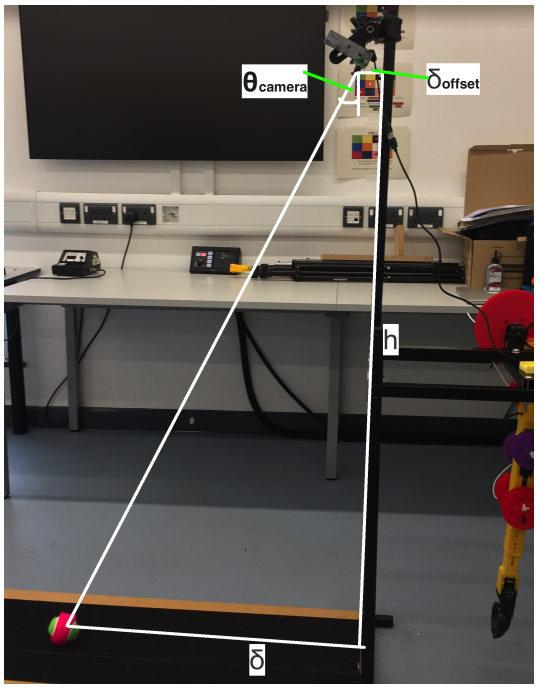


Figure 3.10: A diagram of the values used to calculate the horizontal distance the ball had travelled

right place if it was moved, to keep the experimental results reliable. The accuracy of this technique was then tested by comparing the measured values with the real values measured in the real world. These values were plotted, as can be seen in Fig. 3.12, and show that the system was accurate but with a slight offset.

The equation of the best fit line was taken and implemented into the distance calculation as a calibration function, similar to many measuring devices have gain or calibration ability. This equation can be seen in Eq. 3.4, where $\delta_{new}$ is the calibrated distance and $\delta_{old}$ is the measured distance.

$$\delta_{new} = \frac{\delta_{old} - 6.1787}{0.9304} \tag{3.4}$$

To allow the ball time to land after it had left the hand, and not have the vision node transfer a value for the ball while it is in the air, a slight delay

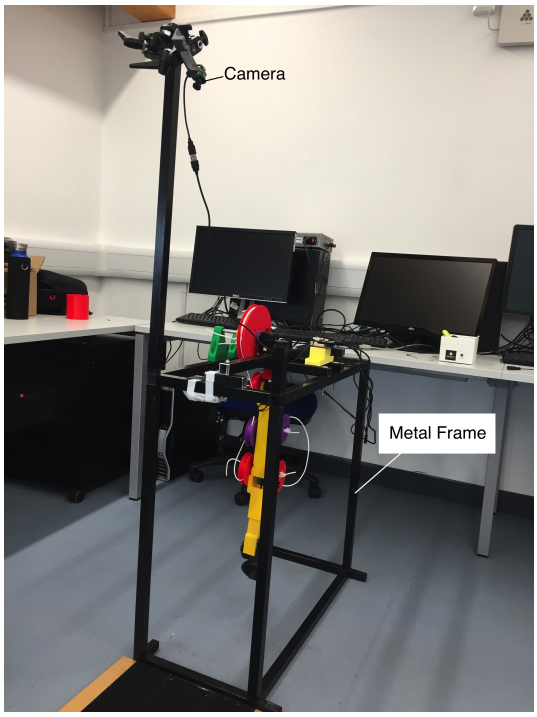


Figure 3.11: An image showing how the camera is attached to the frame.

was added. The delay also had a second responsibility of waiting for the distance value to stop fluctuating before it published the value. This was done by ensuring that the value was equivalent over 10 frames, before issuing the outcome value to the learning node to close the loop.

## 3.6 Motor control

The framework to control the motors was necessary to allow the learning algorithm to control the arm. This integration would give the model access to dictate the motor control, while then being passed the sensory effect from the vision node, as mentioned previsously.

For this, an OpenCM9.04 microcontroller was used to control the motors. This was due to the board being made by the same company as the actuators.

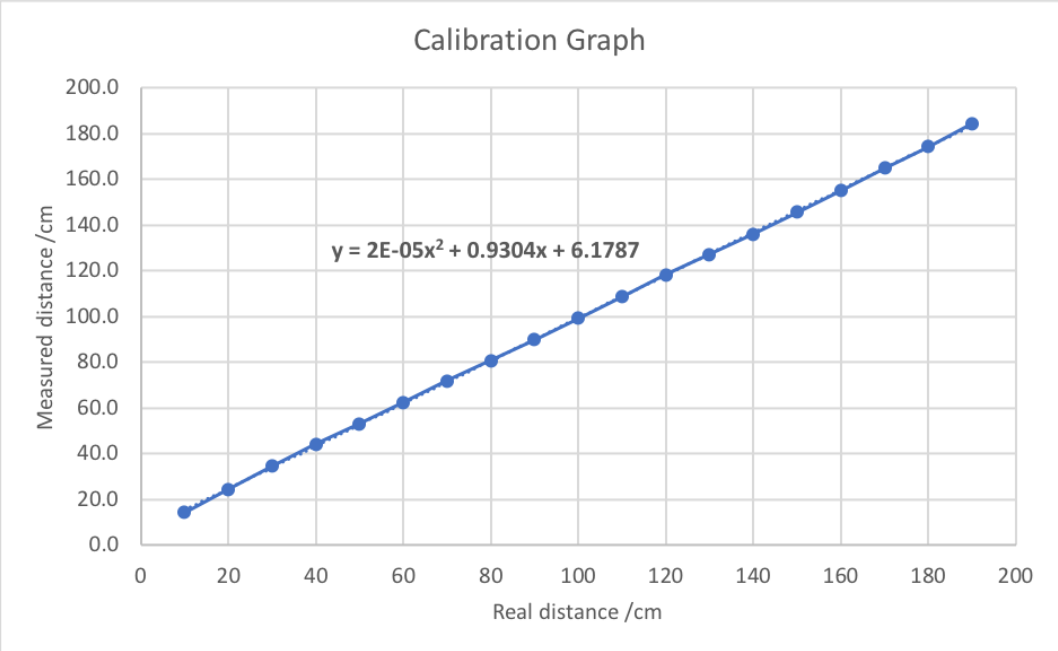


Figure 3.12: The graph used to calculate the offset of measured distance.

This would mitigate concerns of incompatibility. The microcontroller was also connected to an motor shield to control the motors from, as well as draw in an external power supply to power the motors. This board was programmed using the open-source Arduino software that allows programming and uploading to the microcontroller, as well as ROS compatibility. Arduino also has serial communication ability which makes a generally difficult task of debugging a programmed board, slightly less complicated.

Using Arduino, the OpenCM board was programmed to control the motors. In this program, an array of values would be accepted, these would be the motor commands, which consisted of a set of start values and a set of end values. In terms of a throwing motion, the start values represented the position to wind back to and the end values represented the position for the motors to get to during the throwing motion. Before these values were carried out by the motors, limits of the positions that the motors could be sent to needed to be set to avoid any unnecessary breakages. These limits were set conservatively as a result of this being a prototypical system and

breaks would slow progress. As stated in Chapter 2, the decision to 3D print the parts of the arm would allow more freedom in fixing and redesigning. However, a level of strength was sacrificed when opting not to use metal due to this being a prototype and the need to redesign throughout the project outweighed the need for enormous robustness.

## 3.7 Middleware Integration

This section will detail the structure of the system and data transfer, and the process involved in integrating the different components of the system.

As there needed to be data transfer between many different nodes, most of the time of different programming languages, it was appropriate to use ROS. ROS (Robot Operating System) is a robotics middle-ware, that contains many very useful software libraries (Robotics, 2017). ROS allows straight forward development of robotic applications through its powerful developer tools. ROS was chosen for this study due to its facilitation of data transfer through nodes setup as publishers and/or subscribers. The advantage of using publishers and subscribers is that many different subscribers can access the data sent by the publisher simply by subscribing to the topic name. The structure of these nodes can be seen in Fig. 3.13. The nodes are shown in ellipses while the data transfer arrows are labelled with their topic name.

The motor control node was integrated into the system using the rosserial_arduino package. This allowed the ROS packages and framework to be used directly within Arduino IDE. Rosserial provides a ROS communication protocol that works over Arduino's UART, that transmits through the serial port on the microcontroller. It allows microcontrollers to be run as a ROS node, enabling it to carry out all the functions that a normal ROS node would, such as publish and subscribe to ROS messages (*Wiki* n.d.).

As is shown in Fig. 3.13, the learning algorithm passes the actions, or motor commands in this case, to the throwing node which then converts these into values that the Arduino node will accept. The need for this middle node is to remove the need to edit the entire framework. Instead, this node could

be edited easily and therefore did not need to edit the code on the OpenCM board, changing the structure. The result of the actions from the learning node were then measured by the vision node and transferred to the learning algorithm to complete the learning loop.

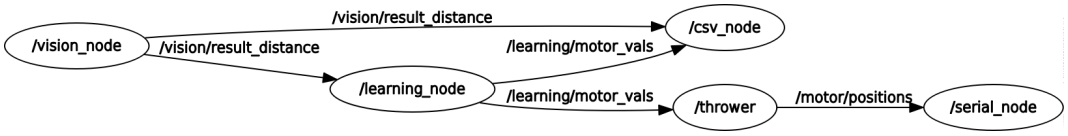


Figure 3.13: A diagram showing the architecture of the ROS nodes.

In parallel to this process, a node named "csv_node" was running. This node was listening to the data transfer on the topics "/learning/motor_vals" and "/vision/result_distance". These topics held the data motor commands created by the learning algorithm, and the measured distance from the vision node respectively. The csv node then stores these values in a csv. This is crucial due to how expensive the data is because of the time it takes to gather. This also allowed for the ability to load up previous models at different points of the experiment.

# 4 Methodology

## 4.1 Throwing Model

This section explains the measures taken, on the physical side, to set-up the throwing model, being the full experimental apparatus. These measures are taken to limit the inconsistencies or variability that come from the setup changing.

The variables that are independent in the experiment are the starting and ending positions of each of the motors. This is represented in the learning by a percentage of its total range, and by its raw positional value in the motor controller. These are varied by the learning algorithm. The dependent variable, therefore, is the resulting distance the ball achieves using these values. This is measured using the vision node as mentioned earlier. It is measured to 1 decimal place.

As the experiment takes place over a number of days, the other variables have to be controlled in a way such that the set-up can be constructed each day in exactly the same configuration. This is crucial to limiting the effect that they have on the reliability of the results. As mentioned earlier in the chapter, one of these variables is the ball's ability to bounce and roll. By implementing the Velcro into the experiment, this inconsistency is greatly reduced. To reduce the effect of the ball losing its hoops, a new ball is used for each new model.

The board is also positioned in the same place each time. By placing the end of the board in line with the bottom of the frame certifies that this is in the same place every time. This mitigates any variation in the board placement, reducing any difference in readings from the vision node.

The frame that was built to hold the arm also controls any changing variables related to the positioning of the arm. The arm is bolted to the frame to inhibit its movement, obviously during the throwing movements, but also when moving the frame between experiments.

The positioning of the camera is similarly set up to allow the configuration to be constructed the same way each time. Through attaching the camera to the frame holding the arm, there are less parts that could be moved and therefore that need calibration each time the setup is moved. Previous iterations were to attach the camera to the ceiling to give a wider view of the experiment. However, due to inconsistencies in setting up the experiment this would cause, coupled with the range of the arm being less than two metres meant that this was not practical.

A feature to load up previous models from the CSV files is also implemented in case an experiment has to be stopped before completion. This enables the experiment to be stopped, any issues dealt with, and then load the data that the model has learned from and continue the experiment.

## 4.2 Learning Model

This section details the process for the learning algorithm to be integrated into the system. Due to the comparison of goal babbling and random motor babbling, a supervised learning algorithm is chosen for this study. This algorithm uses a Nearest Neighbour regression.

The learning node is comprised of an algorithm programmed using the Explauto library (Moulin-Frier, Rouanet, and Oudeyer, 2014). This library supplies the framework for simple model creation, by dictating the number of values desired, the range of these numbers and then programming the function to compute the sensorimotor effect a basic learning model could be created. The first iteration of the algorithm initialised a set of actions as 14 values in the range of 0-100. These would represent the percentage values for the seven motors' start and end position. This was done as a percentage, as the maximum position values for some of the motors were different due to them being different models. As a percentage, it also meant that the values

are easier to compare by eye without having to check what the maximum position value is. In the development period, the number of dimensions was decreased to 6 values instead of 14. This signifies that instead of learning for the entire arm, the lower arm is kept rigid while the control of the shoulder motors would still be learned. This is done with the aim of determining how the model learned with this number of dimensions, and then build up to 14 depending on the outcome of this test. The results from this showed that 6 dimensions are a suitable number of values for the time frame due to the temporal cost of learning. For the random exploration, the random motor commands generated are published to the board for the throw to be carried out. The vision node then determines the distance reached and returns the value to finish the learning loop. However, for goal babbling, 50% of the actions are generated by the inverse model when given one of the targets at random. The actions that are predicted have a Gaussian noise, of maximum amplitude 0.3, applied to the actions. This allows the model to focus its exploration around the actions that result in desired outcomes.

Explauto also includes pre-programmed classifiers that can be used. For this study, the K-nearest neighbour classifier was used due to its robustness to noisy training data as well as its strength when dealing with large quantities of data. For this study K was set to 1, meaning that the new data point would just be classified based on it's nearest neighbour, as the Explauto library did not allow this value to be changed. However, this was not pivotal as the classifier was not a key area being researched.

The data points are classified by the distance achieved by the throw. When predicting, this distance achieved is the target, while the motor values are used as the predictors to select the single nearest neighbour and predict the motor values that will achieve the desired distance.

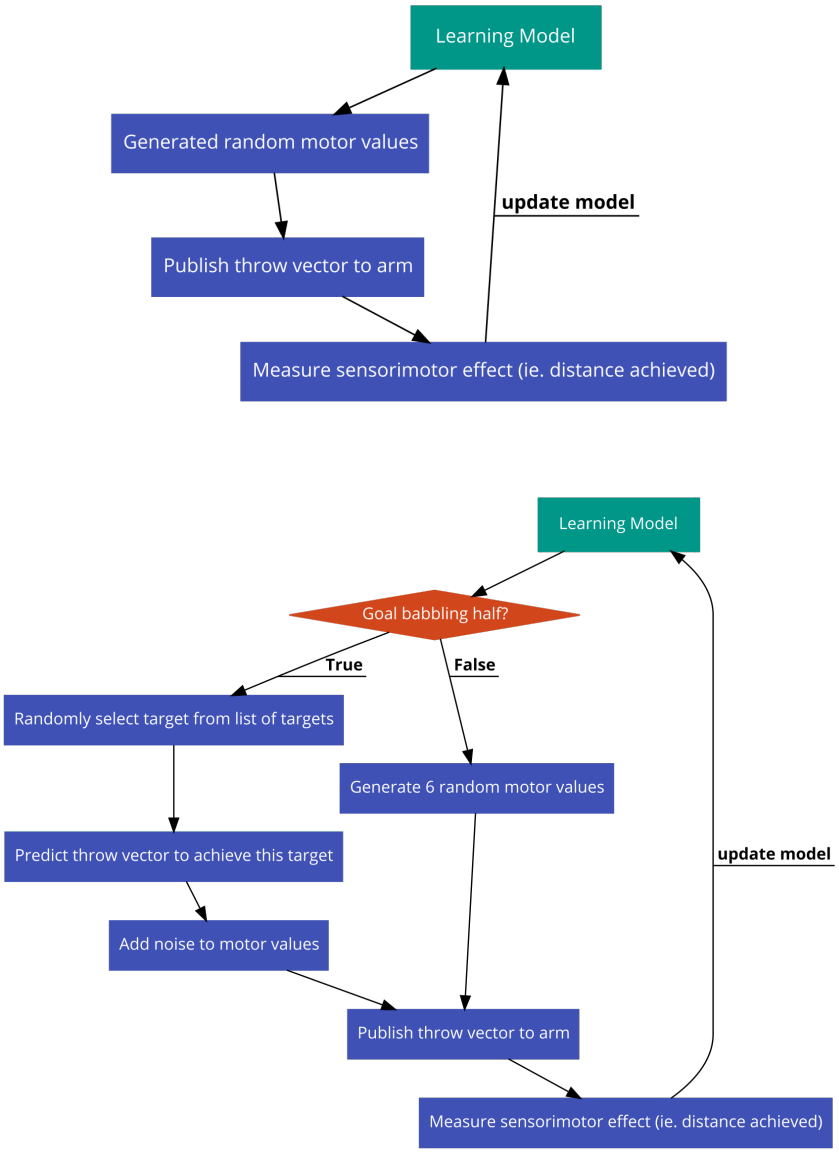


Figure 4.1: Flowcharts showing the random (left) and goal babbling (right) exploration algorithms

## 4.3 Data Handling

Throughout the experiment, it is important to record the data that was being created. During the learning process, the data that needs to be collected is the motor commands and the resulting distance achieved. When testing the model, the target distance, the predicted motor commands that would achieve this and the actual distance achieved are recorded for later analysis. These values stored are the significant details, and it is not necessary to store the rest of the data transfer in the system.

To record these values the easiest and simplest method is to use CSV files. As it is not important how the data is stored or the format it is stored in, as there is not many fields, the CSV method is used for this. Due to the structure of the data transfer within the system, this is easily done by ear-wigging the ROS topics carrying this information and creating a script that stores this data in a CSV file.

By storing the training data during the learning process, this meant that the experiment could be paused and loaded back up whenever it was necessary. This is achieved by simply retraining the learning model from the data that is stored. This is a much simpler process than trying to find a method that saves the model in its current state. This feature allows the experiment to be paused and resumed, reducing any issues that could occur from the experiments taking place over separate days for example.

## 4.4 Evaluation Targets and Metrics

One of the key objectives of this study is to determine a method to measure and quantify the learnability of a particular morphology or in wider uses, a full system. For this scenario however, it is first important to decide on the evaluation targets and the metric to compare the two models.

During preliminary testing, the range of the bowl hand was between 5cm and approximately 160cm. The throws that achieved the maximum distance of approximately 160cm were hard coded without much exploration, so therefore it was deemed that it was worth having a target that was slightly

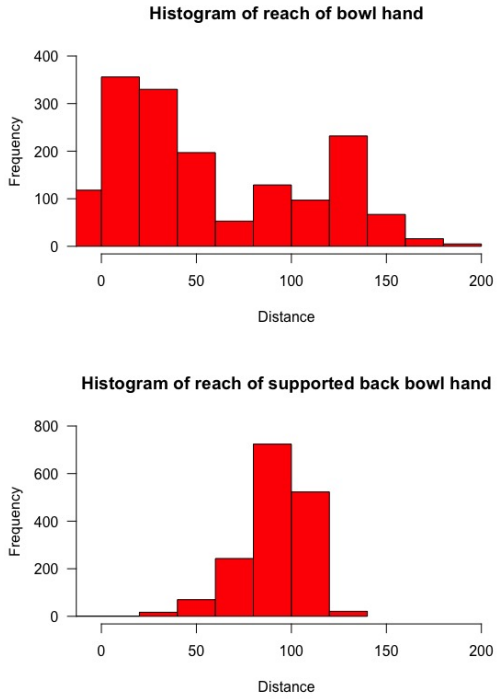


Figure 4.2: The distance achieved by all training data throws for both hands. Left - Bowl hand. Right - Back support hand.

larger than this value to see if the system could be stretched to reach it. The minimum and maximum values of this range, of 25cm and 175cm respectively, would allow full exploration of the capabilities of the arm. The 25cm increments allow for an appropriate number of targets in the range, while still allowing enough of a difference between that keep the targets separate. With all of this in mind, the evaluation targets that are used are in the range of 25cm and 175cm at 25cm increments.

To compare the data gathered from the experiment, the data needs to be analysed in a suitable way. As each configuration will have two repeats, these also must be combined in a reasonable fashion. For this experiment, it is suitable to define the error rate as the difference between each of the targets and the average distance achieved when aiming for this target. This is calculated for each repeat separately and then the values are then joined,

and the mean and spread are then calculated from this list.

The two repeats should be kept separate when finding the average errors for each target as they are discrete models. Therefore, these value could be skewed if a model had not explored that target yet, but the other model had. After the average deviations are found from the targets, the spread of all these values will then be calculated.

## 4.5 Experimental Methodology

To carry out the experiment, the apparatus is configured as described earlier and shown in 4.3. The nodes are then initialised. Once these are confirmed to be running successfully, the control node is given an input authorising the system to carry out the first throw. The ball is placed in the arms hand and then thrown. The ball should then land on the Velcro of the target board. The vision node takes a moment to wait for the value to stop fluctuating and then the value is published. The ball can then be retrieved, being careful not to move the board when pulling the ball from the Velcro. The process can then be repeated.

This experiment consists of 4 different configurations that are being compared: random motor babbling with either hand and goal babbling with either hand. These configurations are chosen to investigate not just how the physical morphology affect the learnability of one type of exploration method, but to be able to compare two separate exploratory approaches and the effect that it has on either of them. Each of these configurations have two repeats of the experiment to be able to compare these to spot any outliers or false readings better. Due to the cost of sample generation in this experiment, two repeats were deemed sufficient but not ideal. Within one of these models, there are 400 learned throws in total. This value was chosen, as during pre-testing it was deemed that this was enough data for the model to sufficiently learn, as the model's accuracy plateaued. For these 400 throws, at each 100 throw milestone the model's accuracy is tested to measure its progress. This allows its learning progress to later be investigated to determine either configurations learnability at consistent and appropriate intervals.

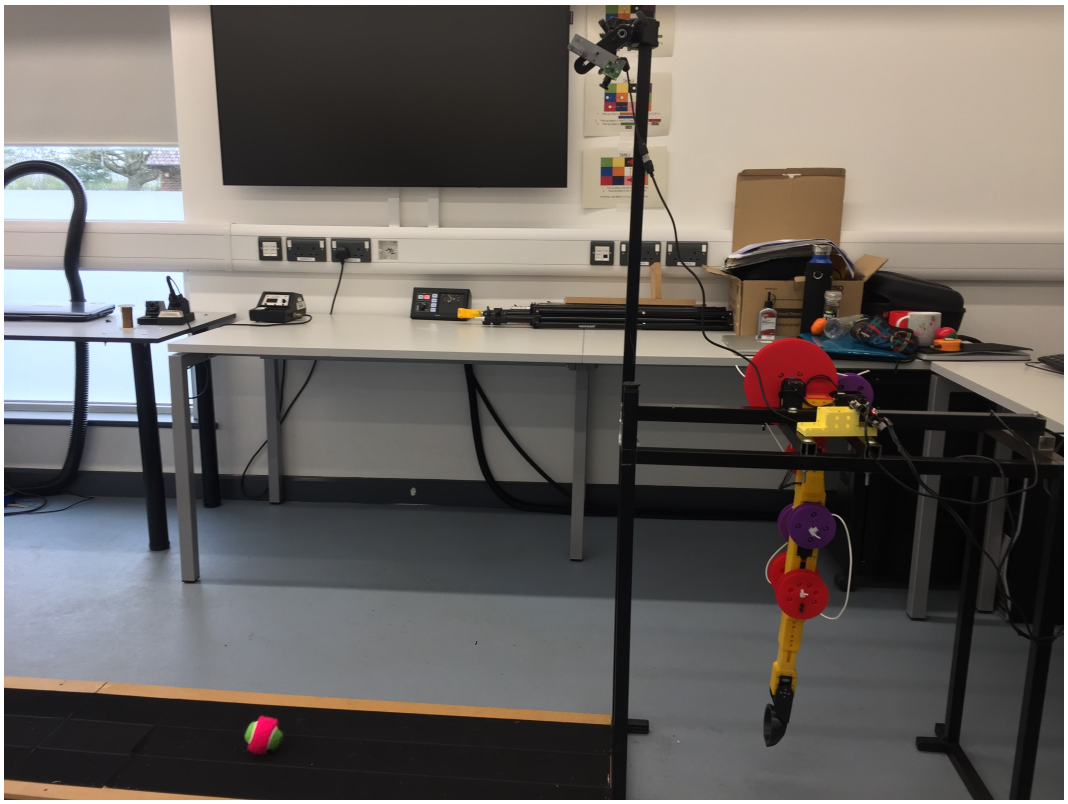

Figure 4.3: An image showing the setup of the apparatus used in the experiment.

If a false throw takes place, such as the ball does not land on the board, or the ball bounces and then lands on the board, these scenarios are recorded as false throws and the distance is recorded as a negative value. This will deter the learning algorithm from selecting this action when predicting.

In the case of a false reading from the vision node, the experiment is stopped and the line of data from the csv file is removed. The experiment is then loaded back to it previous position and the throw is repeated to get the correct value.

# 5 Results and Discussion

This section will cover the experiments carried out to investigate and explore the notion of learnability, and the affect that the varying morphologies have on it. The first experiment, looking into comparing how far from the target each attempted throw was, presented more questions about what was happening during the experiment and what was causing it. This led to the exploring of the variability in the investigation, what was causing it and how this was affecting the predictions from the model. Finally, this led to the investigation with the learning model simulation, exploring how this variability was affecting which values the model predicted.

## 5.1 Experiments

### 5.1.1 Error Comparison

The spread of the data in Fig. 4.2 shows that there were quite distinct limits of achievable distance by both hands, particularly the back-supported hand. This could be due to the height of the lip of the hand being too high up the ball, limiting the release of the ball. Having the front of the hand so high up the ball may cause a number of possible exit angles to be blocked due to the ball having to go up and out of the hand, rather than forward. On the other hand, it is possible that by allowing the rock of the ball during the movement, when the ball leaves the hand it is already rolling forwards which gives the ball added energy allowing it to travel further. This coupled with the lower front lip allowing the ball to roll forward out of the hand with less aggressive style throws, allows the ball to reach closer and further distances than the back-supported hand. The rock of the ball in the shallow

bowl hand could mimic the way humans sometimes throw, by allowing the object the roll down the fingers at release. This gives the ball added velocity on release (Senoo, Namiki, and Ishikawa, 2008), slightly decreasing the accuracy.

Due to the limited reach of the hands, the range of targets tested against were reduced, removing values at the extremities. The new targets were 50, 75, 100 and 125, having removed 25, 150 and 175. The figures show the error for each configuration, calculated by getting the average distance achieved for each target and computing the deviation from the target. Fig. 5.1 shows the error calculated with all of the targets included on the left, while the right shows the limited target range. Comparing these supports the claim that the extreme values were skewing the results, causing them to be erratic due to the arm struggling to throw that distance. As can be seen, the orange and blue lines on the left in Fig. 5.1, representing the back-support hand, are much higher relative to the green and red line, representing the shallow bowl hand, than they are in the graph of the limited targets on the right. This highlights how the back-support hand's limited reach affects the error calculations, much more than the bowl hand. The bowl hand seems to be only very slightly affected.

To calculate the error for 0 learned throws, all of the random exploration throws were compared against each of the targets. The mean and median were then calculated similarly to the other models and plotted. The goal babbling and random exploration models have the same error at time 0 as the goal babbling data could not be used to calculate this error due to the goal babbling using already learned throws.

When the targets are limited, the final accuracy of both morphologies are very similar, with little evidence of a difference between the exploration methods. However, at the start of the learning there is a very distinct separation between the different morphologies. This is likely due, again, to the possible distances that the bowl hand can reach compared to the back-support hand. As the bowl hand has a further reach, this will mean that when testing an unlearned model, the difference between the actual distance from the desired target can be much higher. As the models learn from more throws, they all start to converge. Both variations of the back-support hand configuration seem to plateau slightly after 100 throws, as

opposed to the bowl hand which has a slightly slower learning curve. This is likely attributable to the difference in starting error rates and the bowl hand having a much wider range of achievable results.

The plateau towards the end of both of the configurations is presumably caused by the variability in each of the throws. Due to the lack of consistency in results for the same throw, the models are likely unable to pinpoint a throw that will be able to achieve the target distance. If the same throw sometimes hits the target but other times is 30cm off, it will be hard to make a prediction from that. This is further explored in the next section.

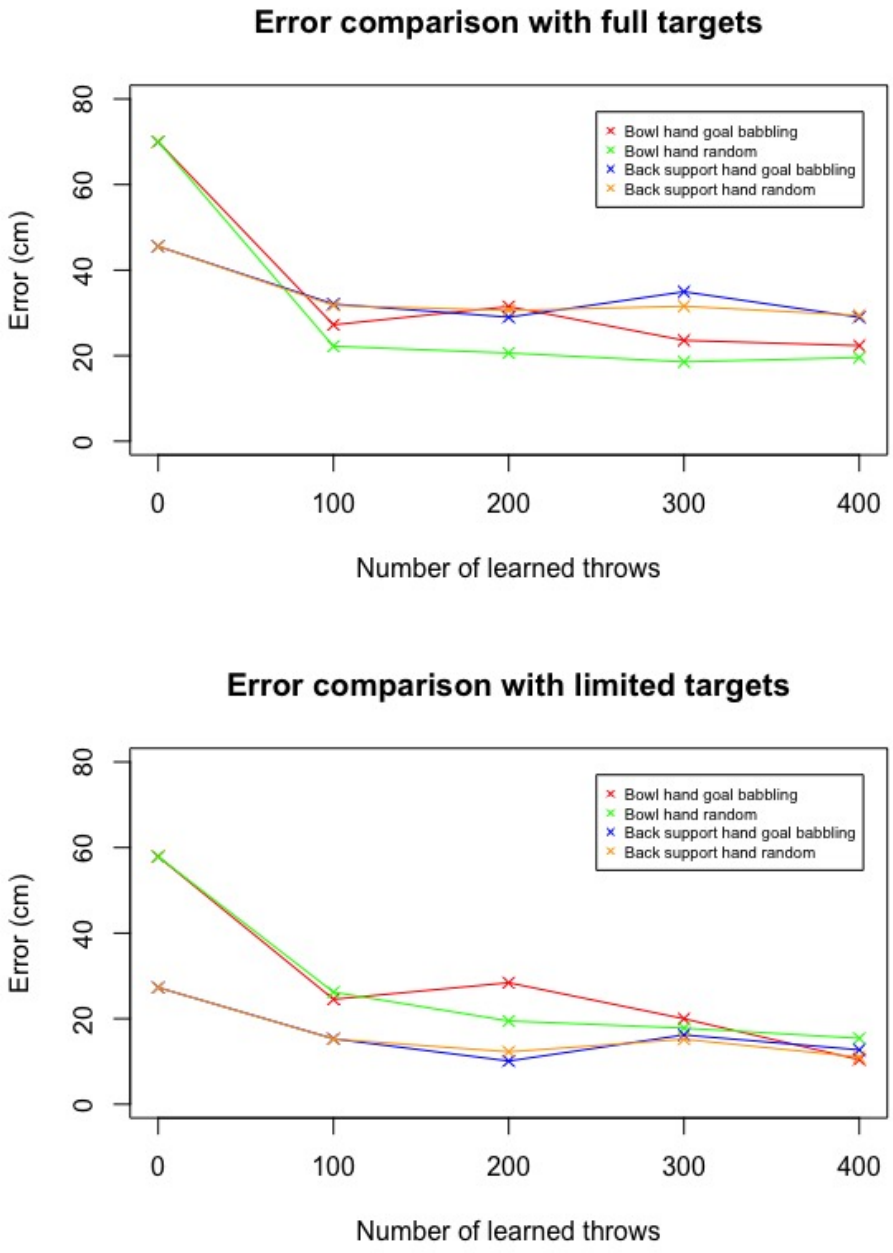


Figure 5.1: The error of each setup as the number of learned throws increased. Left - full targets, right - limited targets

### 5.1.2 Repeatability

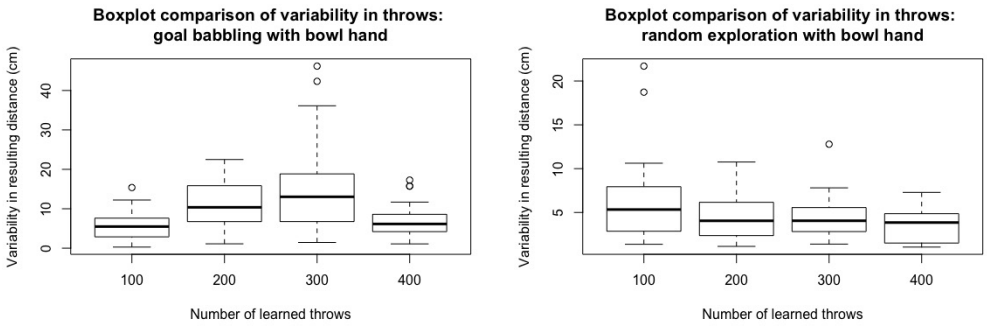


Figure 5.2: The variability of throws with the bowl hand as the number of learned throws increased.

| Learned throws | 100 | 200 | 300 | 400 |
|---|---|---|---|---|
| Lower whisker | 0.28 | 1.11 | 1.46 | 1.09 |
| Lower hinge | 2.875 | 6.75 | 6.76 | 4.195 |
| Median | 5.485 | 10.365 | 13.03 | 6.145 |
| Upper hinge | 7.62 | 15.845 | 18.845 | 8.595 |
| Upper whisker | 12.22 | 22.49 | 36.12 | 11.69 |

Table 5.1: Data spread of bowl hand, goal babbling

| Learned throws | 100 | 200 | 300 | 400 |
|---|---|---|---|---|
| Lower whisker | 1.37 | 1.13 | 1.39 | 1.05 |
| Lower hinge | 2.86 | 2.365 | 2.82 | 1.51 |
| Median | 5.335 | 4.05 | 4.055 | 3.85 |
| Upper hinge | 7.94 | 6.16 | 5.56 | 4.86 |
| Upper whisker | 10.62 | 10.76 | 7.81 | 7.29 |

Table 5.2: Data spread of bowl hand, random exploration

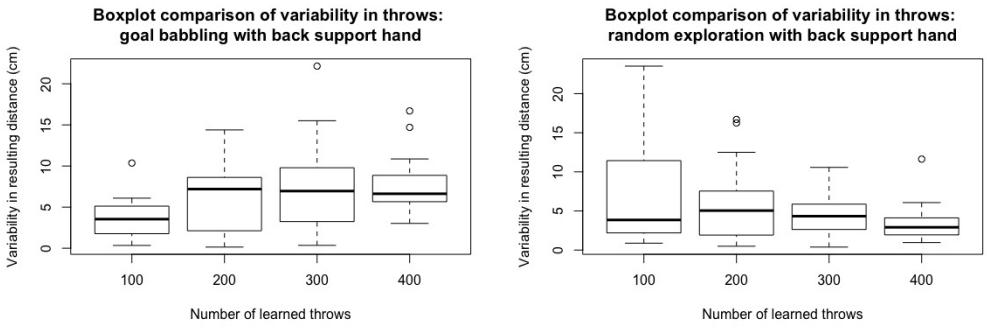


Figure 5.3: The variability of throws with the back support hand as the number of learned throws increased.

| Learned throws | 100 | 200 | 300 | 400 |
|---|---|---|---|---|
| Lower whisker | 0.34 | 0.14 | 0.35 | 3.02 |
| Lower hinge | 1.77 | 2.125 | 3.245 | 5.66 |
| Median | 3.54 | 7.195 | 6.955 | 6.625 |
| Upper hinge | 5.12 | 8.615 | 9.79 | 8.865 |
| Upper whisker | 6.09 | 14.39 | 15.51 | 10.86 |

Table 5.3: Data spread of back support hand, goal babbling

| Learned throws | 100 | 200 | 300 | 400 |
|---|---|---|---|---|
| Lower whisker | 0.88 | 0.49 | 0.38 | 0.96 |
| Lower hinge | 2.195 | 1.915 | 2.635 | 1.94 |
| Median | 3.85 | 5.045 | 4.33 | 2.91 |
| Upper hinge | 11.435 | 7.545 | 5.87 | 4.115 |
| Upper whisker | 23.52 | 12.5 | 10.57 | 6.07 |

Table 5.4: Data spread of back support hand, random exploration

Fig. 5.2 and Fig. 5.3 show the variability in results achieved using the same predicted throwing movement. This was calculated by finding the mean of the 5 data points for each target and finding the difference between this average and each point. This was done for both models of each configuration, and these were averaged out. This allowed for 20 data points for each milestone of 100 throws, which were box plotted as can be seen.

The variability for the hand without the back support, shown in Fig. 5.2, likely came from the ball rocking back and forth in the hand during the throwing movement. Due to the hand being quite shallow, the lip of the hand did not reach very far up the ball and so movement back and forth was not restricted during the throw. This likely caused the ball to have different amounts of momentum in different throws accounting for some of the variability.

The left graphs of Fig. 5.2 and Fig. 5.3 also show that the goal babbling exploration method, in the first 300 throws, has a trend of exploring less consistent throws, possibly in an attempt to explore more of the action space, but then consolidating this information in the final 100 throws to bring some final, more repeatable actions that result in outcomes with less variability. This is shown by the decrease in the spread of the box plots. These more inconsistent actions could be throws having a much later release time and so the ball travels more vertically in its trajectory. This results in smaller changes to the release timing having greater outcomes on the resulting distance thrown. This is likely also the case for throws with have an earlier release as the ball will be much closer to the board in its trajectory so a slight deviation in timing can also produce large deviations in outcome.

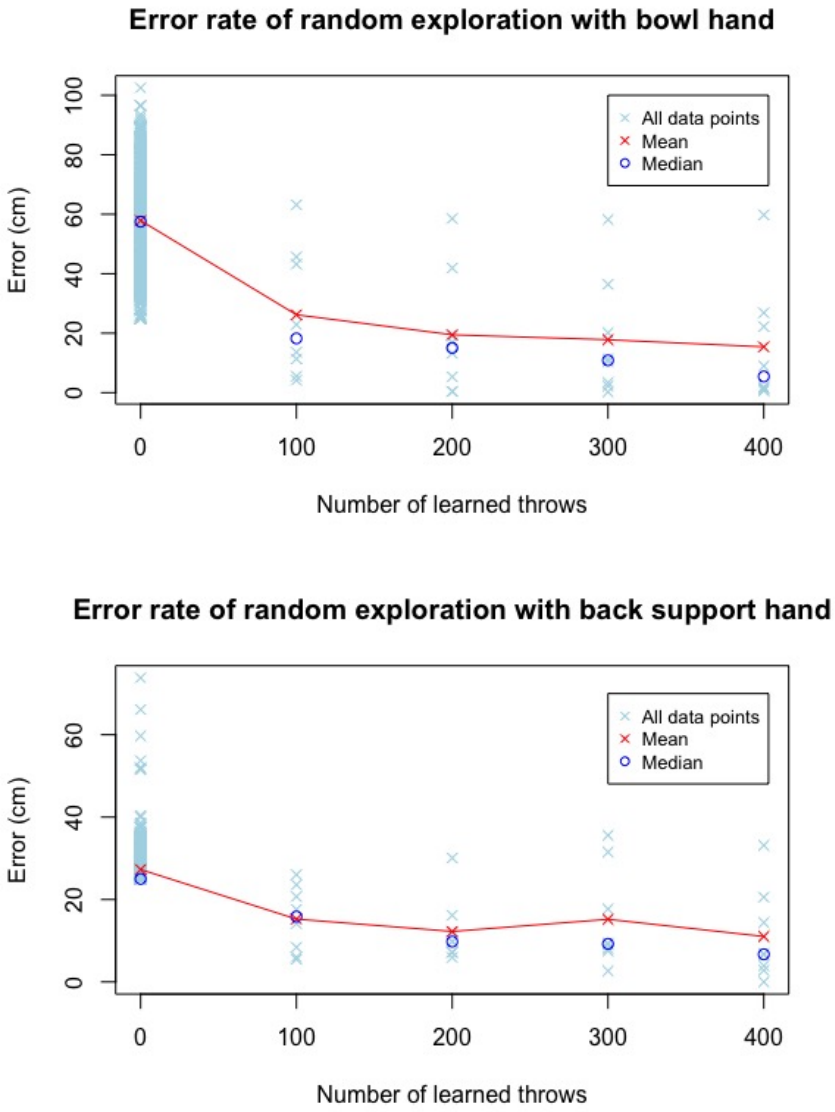


Figure 5.4: The error of random exploration for both hands as the number of learned throws increased. Left - Bowl hand, right - back support hand

| Learned throws | 0 | 100 | 200 | 300 | 400 |
|---|---|---|---|---|---|
| Lower whisker | 25 | 4.16 | 0.40 | 0.26 | 0.66 |
| Lower hinge | 25 | 8.43 | 2.89 | 2.81 | 1.41 |
| Median | 25 | 18.24 | 15.05 | 10.87 | 5.45 |
| Upper hinge | 28.475 | 44.43 | 30.43 | 28.27 | 24.55 |
| Upper whisker | 33.65 | 63.18 | 58.54 | 58.2 | 26.86 |

Table 5.5: Data spread of Fig. 5.4, bowl hand

| Learned throws | 0 | 100 | 200 | 300 | 400 |
|---|---|---|---|---|---|
| Lower whisker | 25 | 5.48 | 5.94 | 2.66 | 0 |
| Lower hinge | 25 | 7.2 | 7.16 | 7.82 | 3.36 |
| Median | 25 | 15.86 | 9.78 | 9.23 | 6.7 |
| Upper hinge | 28.475 | 22.14 | 14.03 | 24.6 | 17.48 |
| Upper whisker | 33.65 | 26.04 | 16.16 | 35.58 | 33.14 |

Table 5.6: Data spread of Fig. 5.4, back support hand

This graphs in Fig. 5.4 and 5.5 show a trend that perhaps the noise the throws with the bowl hand help with learning. With the inconsistency in the throw it is possible that this allows the algorithm to explore more of the action space, compared to the back-support hand's more repeatable throws. This is significant as it appears to support findings in (Rolf, 2012), which suggest that before repeatability in an action is important, first that action has to be discovered by the model.

This notion of the noise adding to the ability to explore is supported also by the speed of learning of the random exploration with the bowl hand, Fig. 5.4, compared to the back-supported hand. By exploring more of the space, possible actions are discovered that might not otherwise be discovered. This raises a case that before repeatability becomes a useful feature to the system, an action has to first be discovered that can reach the particular target.

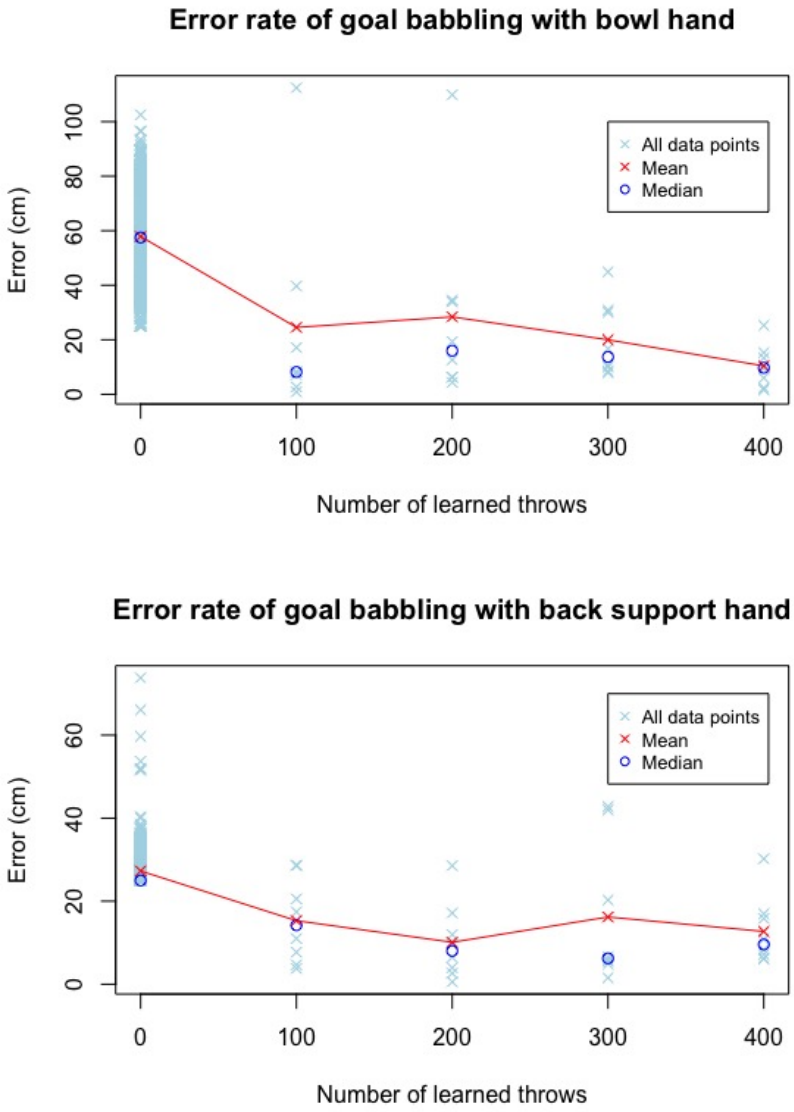


Figure 5.5: The error of goal babbling exploration for both hands as the number of learned throws increased. Left - Bowl hand, right - back support hand

To test the significance of the results from the robot investigations, it was aimed to use a T-test as this test is used to determine if there are significant differences between two groups of data which may be related in certain

| Learned throws | 0 | 100 | 200 | 300 | 400 |
|---|---|---|---|---|---|
| Lower whisker | 25 | 0.88 | 4.38 | 7.96 | 1.64 |
| Lower hinge | 25 | 4.92 | 6.27 | 9.27 | 4.17 |
| Median | 25 | 8.17 | 15.98 | 13.71 | 9.73 |
| Upper hinge | 28.475 | 28.4 | 34.22 | 30.5 | 14.29 |
| Upper whisker | 33.65 | 39.68 | 34.58 | 44.88 | 25.26 |

Table 5.7: Data spread of Fig. 5.5- bowl hand

| Learned throws | 0 | 100 | 200 | 300 | 400 |
|---|---|---|---|---|---|
| Lower whisker | 25 | 3.78 | 0.56 | 1.54 | 6.12 |
| Lower hinge | 25 | 6.19 | 3.3 | 5.24 | 6.69 |
| Median | 25 | 14.21 | 8.02 | 6.24 | 9.58 |
| Upper hinge | 28.475 | 24.55 | 14.54 | 31.14 | 16.4 |
| Upper whisker | 33.65 | 28.72 | 28.54 | 42.8 | 30.22 |

Table 5.8: Data spread of Fig. 5.5- back support hand

features. However, for this test to be carried out the data must be Gaussian distributed. A Shapiro-Wilks test is used to test this, with the null hypothesis assuming that the data is normally distributed. As can be seen from Fig. 5.6, the low p-values mean that this null hypothesis must be rejected, showing that this data is not Gaussian distributed.

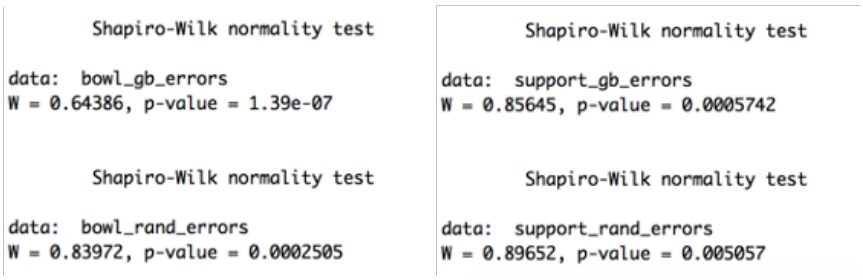


Figure 5.6: Shapiro-Wilks test of bowl and back support hand.

As the data is not normally distributed, the parametric T-test was not applicable. The Kolmogorov-Smirnov test is a non-parametric test used

to compare two samples, with the null hypothesis assuming that they are drawn from the same distribution. The results from this, as can be seen from Fig. 5.7, show that the data is not hugely significant. This is not surprising as there is a need for more data, or larger margins of differences. Due to the cost of creating this data, this was not possible.

```
        Two-sample Kolmogorov-Smirnov test

data:  bowl_gb_errors and support_gb_errors
D = 0.1875, p-value = 0.6272
alternative hypothesis: two-sided
```

```
        Two-sample Kolmogorov-Smirnov test

data:  bowl_rand_errors and support_rand_errors
D = 0.25, p-value = 0.273
alternative hypothesis: two-sided
```

Figure 5.7: Kolmogorov-Smirnov test of comparing the two bowl hand variations and the two back support hand variations.

### 5.1.3 Simulation of Learning Algorithm

To see if the decreasing variability, particularly shown in the random exploration of Fig. 5.2 and 5.3, was attributable to the learning model selecting actions that gave more consistent outcomes, a new example environment was created with a simple function used to calculate the outcome of a number of actions with half of the outcomes having a Gaussian noise added to them. This would distribute the outcomes, creating a simulated inconsistency for the same actions. The learned model would then be tested to discern whether the "more consistent" actions were being selected. One investigation using this algorithm, was to study the affect that a varying number of actions had on the spread of chosen sets of these actions. These actions would be randomly generated numbers in the range of -10 to 10. For this investigation, the function split the actions into pairs. In these pairs, the two numbers were added together. The pairs were then subtracted from each other. Once this calculation had been completed, a Gaussian noise of maximum amplitude 20 was applied to the outcome if the sum of the first half of values was larger than the sum of the second half. This was a simple and trivial method of splitting the values.

| % of values selected | 2 actions | 4 actions | 6 actions | 8 actions |
|---|---|---|---|---|
| Goal babbling 1 | 26.3 | 38.0 | 32.7 | 40.0 |
| Goal babbling 2 | 28.0 | 36.3 | 45.7 | 41.7 |
| Goal babbling 3 | 29.7 | 45.0 | 38.3 | 36.0 |
| Motor babbling 1 | 34.0 | 45.0 | 40.0 | 36.0 |
| Motor babbling 2 | 29.7 | 32.7 | 52.3 | 44.0 |
| Motor babbling 3 | 28.3 | 32.3 | 45.0 | 40.7 |

Table 5.9: A table showing the percentage of noisy values selected when the dimensions are varied.

As can be seen in Fig. 5.9, more actions do not seem to have an effect on the ratio of actions selected that did or did not have noise applied to the outcome. It can be seen that the affect is getting weaker with the more actions, however, this is probably due to the variability increase as the number of numbers is increased. Another possible reason for this could be that noise of an amplitude of 20 has much less of an effect with the sum of

8 numbers compared to the sum of 2 numbers. It was concluded that the evidence was not substantial to support a theory that a different number of actions had an impact on the spread of selected action sets.

| Number of actions | Successes | Trials | p-value |
|---|---|---|---|
| 2 | 79 | 300 | ¡2.20E-16 |
| 2 | 85 | 300 | 3.766E-14 |
| 2 | 89 | 300 | 1.394E-12 |
| 4 | 114 | 300 | 3.829E-05 |
| 4 | 109 | 300 | 2.546E-06 |
| 4 | 135 | 300 | 9.390E-02 |
| 6 | 98 | 300 | 1.901E-09 |
| 6 | 137 | 300 | 1.488E-01 |
| 6 | 115 | 300 | 6.321E-05 |
| 8 | 120 | 300 | 6.342E-04 |
| 8 | 125 | 300 | 4.589E-03 |
| 8 | 108 | 300 | 1.421E-06 |

Table 5.10: Binomial test of simulation model, varying the number of actions (goal babbling)

A sign test produces a probability value, or p-value, to help determine a level of significance of the data. The p-value is the probability of the data being under the null hypothesis. The null hypothesis of this test is that there is a 50/50 split between noisy or non-noisy actions selected, the alternative hypothesis being that there is not a 50/50 split. Low p-values, p¡0.05 or 5%, suggests that the Null-hypothesis is considered implausible.

As mentioned previously, the investigation of the model selecting more consistent actions, does not seem to have conclusive results when the number of actions are varied. Table 5.10 and 5.11 supports this theory, showing the p-value fluctuating throughout the variations.

As the number of actions did not seem to have an effect, the number of targets was varied. The number of actions were kept consistent at 2. This would allow the results to be plotted to visualise any trends better. The

| Number of actions | Successes | Trials | p-value |
|---|---|---|---|
| 2 | 102 | 300 | ¡3.187E-08 |
| 2 | 89 | 300 | 1.394E-12 |
| 2 | 85 | 300 | 3.766E-14 |
| 4 | 135 | 300 | 9.390E-02 |
| 4 | 98 | 300 | 1.901E-09 |
| 4 | 97 | 300 | 9.056E-10 |
| 6 | 120 | 300 | 6.342E-04 |
| 6 | 157 | 300 | 4.530E-01 |
| 6 | 135 | 300 | 9.390E-02 |
| 8 | 108 | 300 | 1.421E-06 |
| 8 | 132 | 300 | 4.313E-02 |
| 8 | 122 | 300 | 1.455E-03 |

Table 5.11: Binomial test of simulation model, varying the number of actions (random)

number of targets were varied from 1 to 10. These can be seen in 5.8. The blue line, y=x, shows the decision line of whether the outcomes would be made noisy, above the line being the ones with the noise applied. The green lines display the targets, with the red plots showing predicted actions to achieve that target.

An example of the goal babbling process can be seen in Fig. 5.9. As explained above, the blue line shows the split between noisy values, the red lines show the targets and the plots are the actions chosen to achieve the desired target. This graph shows the clustering around actions known to result in the desired outcome, representing the goal babbling's target oriented exploration.

The wider spread data points represent the random component of the exploration. This component allows the algorithm to first initially discover some of the action space, finding preliminary actions to explore further, but also allows the exploration to still discover actions that will improve the outcome of the learning. This is shown by the two clusters on the bottom

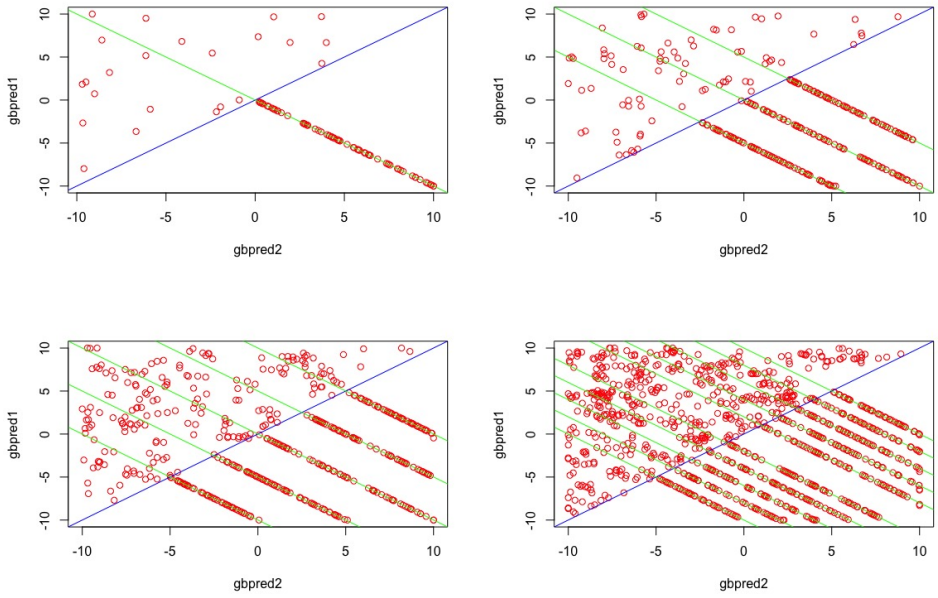


Figure 5.8: A table showing the percentage of noisy values selected when the number of targets are varied

red line, the target of -5. The likely explanation for this is that the goal babbling started exploring the cluster on the top half of the blue line, before the random part to the exploration discovered an action that resulted in a better outcome. The cluster under the blue line was then explored as the better action.

| % of values selected | 1 target | 3 targets | 5 targets | 10 targets |
|---|---|---|---|---|
| Goal babbling 1 | 23.0 | 26.3 | 39.6 | 50.9 |
| Goal babbling 2 | 10.0 | 28.0 | 45.2 | 50.9 |
| Goal babbling 3 | 10.0 | 29.7 | 40.6 | 51.1 |
| Motor babbling 1 | 24.0 | 34.0 | 32.8 | 34.0 |
| Motor babbling 2 | 26.0 | 29.7 | 37.6 | 34.8 |
| Motor babbling 3 | 22.0 | 28.3 | 31.6 | 34.5 |

Table 5.12: A table showing the percentage of noisy values selected when the number of targets are varied

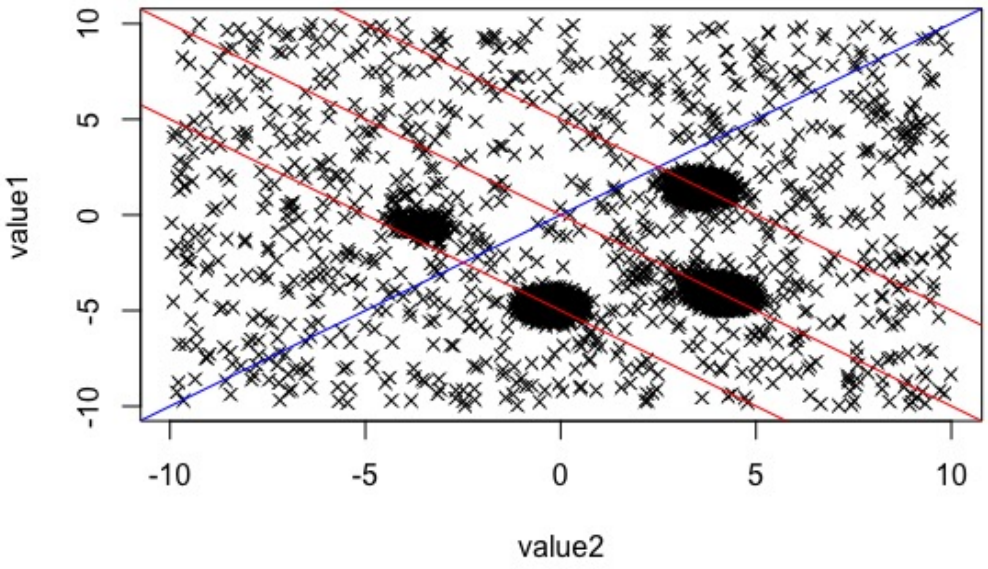


Figure 5.9: A plot showing goal babbling during the investigation into the affect of varying the number of targets. This is the graph of one of the repeats of 3 targets.

5.12 shows the percentage of noisy values selected by the learned model. This shows that, for goal babbling, there is a correlation between the number of targets and the number of noisy values selected. However, there does not seem to be as large of an effect on the random explored model, the percentage selected seems to stay consistent. This shows that the effect seems to be much stronger with the goal-babbling exploration, with the effect weakening as the number of targets is increased.

Table 5.13 and 5.14 show the binomial test for the varying number of targets, showing that there is an effect with the varying number of targets. This is present for both the random exploration and the goal babbling, with the very low p-values proving the argument previously presented that a significant effect has been discovered.

| Number of targets | Successes | Trials | p-value |
|---|---|---|---|
| 1 | 23 | 100 | 5.514E-08 |
| 1 | 10 | 100 | ¡2.20E-16 |
| 1 | 10 | 100 | ¡2.20E-16 |
| 3 | 79 | 300 | ¡2.20E-16 |
| 3 | 84 | 300 | 1.467E-14 |
| 3 | 89 | 300 | 1.394E-12 |
| 5 | 198 | 500 | 3.799E-06 |
| 5 | 226 | 500 | 3.546E-02 |
| 5 | 203 | 500 | 3.039E-05 |
| 10 | 509 | 1000 | 5.909E-01 |
| 10 | 509 | 1000 | 5.909E-01 |
| 10 | 511 | 1000 | 5.067E-01 |

Table 5.13: Binomial test of simulation model, varying the number of targets (goal babbling)

| Number of targets | Successes | Trials | p-value |
|---|---|---|---|
| 1 | 24 | 100 | 1.810E-07 |
| 1 | 26 | 100 | 1.667E-06 |
| 1 | 22 | 100 | 1.591E-08 |
| 3 | 102 | 300 | 3.187E-08 |
| 3 | 89 | 300 | 1.394E-12 |
| 3 | 85 | 300 | 3.766E-14 |
| 5 | 164 | 500 | 1.128E-14 |
| 5 | 188 | 500 | 3.236E-08 |
| 5 | 158 | 500 | ¡2.20E-16 |
| 10 | 340 | 1000 | ¡2.20E-16 |
| 10 | 348 | 1000 | ¡2.20E-16 |
| 10 | 345 | 1000 | ¡2.20E-16 |

Table 5.14: Binomial test of simulation model, varying the number of targets (random)

## 5.2 Research Findings

The main aims of this research were to explore the idea of learnability, determine a method to measure it and then use this to compare two separate hand designs to determine if the learnability was affected by the variations. The initial hypothesis was that by increasing the consistency of the throws, a morphology would have an increased learnability.

However, due to the differing ranges that these two hand designs had, comparing the two proved difficult. The bowl hand had a much greater range compared to the back-support hand, so was able to explore more sensory effects, clearly shown by Fig. 4.2. When the error rate was calculated with the full list of targets, Fig. 5.1 it showed that the bowl hands accuracy at throw 0 was far less than the support hand. However, after 100 throws it was able to overtake the back support hand. As mentioned previously, this was likely due to the back-support hand being evaluated on distance that it could not achieve.

With the reduced targets, possibly due to the larger range, the speed of learning of the bowl hand was slightly slower than the back-support hand. Due to the larger range, the bowl hand has explored outside this new range more than the back-support hand. Consequently, this makes comparing the two morphologies difficult due to the difference in ranges.

This supports the results from the Repeatability section, that shows that there is not as simple a link between the hand design and its learnability. Between these two characteristics, there are many more factors in effect. It could be argued that variability is part of the morphology. However, without knowing whether the variability is brought about by the hands or the system as a whole, there is insufficient evidence to determine this.

Nevertheless, with the reduced targets, the goal babbling graphs in Fig. 5.2 and 5.3, suggest that this variability can be learned. The spread of the boxplots decreasing suggests that the goal babbling starts with relatively consistent throws and then branches to less consistent throws as it progresses. Towards the end it then seems to consolidate this data and return back to consistent throws using the data it had collected. This evidence is not conclusive, however it does show a slight trend in both graphs.

Comparing these two figures, it can be seen that on average the bowl hand has more variability in its throws. The reasons for this have been postulated in previous sections, relating to the rock of the ball in the hand. This could possibly be what is causing the bowl hand to have a slower speed of learning than the more consistent back-support hand. However, as Rolf found in (Rolf, 2012), this noise possibly helped with the exploration of the action space. This can be seen by Fig. 4.2, which show the learning data of each hand design and how the space was explored. The back-supported hand had approximately 1500 throws that resulted in a distance between 60 and 120, opposed to the bowl hand which had a much wider spread of 0-160. The negative values here represent invalid throws, such as ones that didn't land on the board or the ball fell out of the hand due to the throwing motion.

This theory of the noise adding to the ability to explore is supported also by the speed of learning of the bowl hand, Fig. 5.4 and Fig. 5.5, compared to the back-supported hand. By exploring more of the space, possible actions are discovered that might not otherwise be discovered. This raises a case that before repeatability becomes a useful feature to the system, an action has to first be discovered that can reach the particular target.

The notion that the goal babbling seems to select more consistent throws is also supported by the results in section 4.1.3. The effect that the number of actions have on the percentage of noisy values selected are inconclusive as Fig. 5.9 shows. As the number of actions are increased, both the goal and motor babbling percentages approach the 50 percent mark. However, this is likely due to how this simulation was carried out by finding a sum of the "actions". Due to this, the effect seen is probably due to the variability increase as the size of the data set is increased. As mentioned previously, another possible reason for this could be that noise of an amplitude of 20 has much less of an effect with the sum of 8 numbers compared to the sum of 2 numbers. These reasons seemed to combine to create this effect.

On the other hand, Fig. 5.12 suggests a much more conclusive effect. With one target, the goal babbling only selects noisy values 14.3 percent of the time on average. The suggests that it is selecting consistent actions. This effect gets weaker with more targets. With the reduced target list having 4 targets, this means that this effect is definitely present in this study.

## 5.3 Limitations

There were a number of limitations present in this study. The first of which was the limiting of the number of actions able to be explored. It was planned to use all of the joints present in the robot arm for the learning to explore. However, these were limited to just the motors involved in the shoulder joint, going from 7 actions to 3. This was done initially with the plan to then increase back up to 7. However, during the testing of the rigid arm system, allowing just the shoulder to move, the sample generation was proving too costly to increase the actions back up to 7. Consequently, the experiment was limited to only moving the shoulder joint.

Similarly, due to time constraints, the number of throws and repeats carried out were also limited. From pre-testing, it was apparent that by limiting the number of throws for each model to 400 would not have a significant effect on the outcome of the investigation due to the error rate plateauing. This effect can be seen in Fig. 5.1, as both lines seem to level off. However, the number of repeats of each run had to be limited to 2 as more would not have been able to be completed in the time frame.

Finally, the different configurations investigated also had to be limited to two different morphologies and 2 different exploration methods. These two exploration methods used the same regression algorithm: Nearest neighbour.

# 6 Future Work

The findings of this paper have presented many more questions surrounding the link between morphology and learnability. Many of these come from the limitations that were present for this study.

Firstly, further studies could incorporate the full arm rather than limiting the movement to only the shoulder and assessing the effect that this could have on the learnability of the system. Investigating whether an increase of actions, although it seems to not have an effect on the percentage of noisy values selected by the learning, could identify a different outcome to this study. By altering the morphologies when they have many more possible actions could uncover different findings.

Similarly, comparing more morphologies could also provide new insights into the topic. As a result of the differing ranges of the morphologies having such an effect on the findings of the study, it shows that the morphologies being compared must have very similar ranges. Otherwise, the results are not fully conclusive.

The accuracy of the models created were only tested up to a total of 400 throws. While this was deemed suitable for this study from preliminary testing, it could become apparent that this is only the start of the learning process for these models. Continuing these experiments past 400 throws could show further results either supporting or opposing the findings of this paper.

This study only investigated two different exploration methods for the learning process, and found that these have a large effect on the findings. Future studies could investigate this more by including more explorations methods, and varying the regression algorithms. For this study only Nearest Neighbour was used to approximate the results. This could be expanded to include others such as logistic regression.

# 7 Conclusion

This aim of this study was to investigate how the hand design of a robot arm affects the learnability of a task, in this case aimed throwing. This included exploring the notion of learnability and attempting to conclude a method to measure it. This was done by designing a robot arm that was able to be altered, allowing the morphologies being tested to be varied. The effect this had on the 2 different learning algorithms, the goal and motor babbling exploration methods, was analysed at periodic intervals to assess the progress of learning of each configuration.

Due to the time costs of the learning process for this study, many different aspects of the study had to be limited such as the configurations tested and the number of repeats carried out. This opens up the possibilities of further work into the variations that could not be included in this study, such as including more learning algorithms or different morphologies, including comparing a rigid arm to one that can bend at more joints. By investigating these in future work could provide further findings to this topic.

In conclusion, study finds that there is not as direct a link between the hand design and learnability as initially thought. Instead, there are many factors that affect this, that could be argued to be contained under the umbrella of morphology. The main factors are the range and variability of the throws from each hand design. The study finds that by comparing two different hands with different ranges, it is very hard to compare the learnability of them. However, by limiting the range of the evaluation targets to within both hands' ranges it becomes clear that the added repeatability that the back-support hand provides to its throws adds to the speed of learning of the configuration. Due to this, it could be argued that this therefore shows that the back-support hand has an increased learnability. However, due to the difference in the range of achievable distances, it can not be considered a conclusive finding.

# Bibliography


Aboaf, Eric W, Christopher G Atkeson, and David J Reinkensmeyer (1987). "Task-level robot learning: Ball throwing". In: (cit. on pp. 1, 10, 15).

Anson, Erik (2017). *Given the angle the pendulum is released, what is the launch angle for maximum distance?* URL: https://www.quora.com/Given-the-angle-the-pendulum-is-released-what-is-the-launch-angle-for-maximum-distance (visited on 03/18/2020) (cit. on p. 19).

Brownlee, Jason (Dec. 2015). *Basic Concepts in Machine Learning*. URL: https://machinelearningmastery.com/basic-concepts-in-machine-learning/ (cit. on p. 2).

Calvin, William H (1982). "Did throwing stones shape hominid brain evolution?" In: *Ethology and Sociobiology* 3.3, pp. 115–124 (cit. on pp. 8, 15).

Calvin, William H (1983). "A stone's throw and its launch window: Timing precision and its implications for language and hominid brains". In: *Journal of theoretical Biology* 104.1, pp. 121–135 (cit. on pp. 2, 8, 15).

Farchy, Alon et al. (2013). "Humanoid robots learning to walk faster: From the real world to simulation and back". In: *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 39–46 (cit. on p. 1).

Futagi, Yasuyuki, Yasuhisa Toribe, and Yasuhiro Suzuki (2012). "The grasp reflex and moro reflex in infants: hierarchy of primitive reflex responses". In: *International journal of pediatrics* 2012 (cit. on pp. 3, 7, 15).

Kim, Seungsu and Stéphane Doncieux (2017). "Learning highly diverse robot throwing movements through quality diversity search". In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1177–1178 (cit. on pp. 9, 15).

Kober, Jens, Matthew Glisson, and Michael Mistry (2012). "Playing catch and juggling with a humanoid robot". In: *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE, pp. 875–881 (cit. on pp. 9, 15).

Kober, Jens, Erhan Oztop, and Jan Peters (2011). "Reinforcement learning to adjust robot movements to new situations". In: *Twenty-Second International Joint Conference on Artificial Intelligence* (cit. on pp. 12, 15).

Kober, Jens and Jan Peters (2011). "Learning elementary movements jointly with a higher level task". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 338–343 (cit. on pp. 12, 15).

Kober, Jens, Andreas Wilhelm, et al. (2012). "Reinforcement learning to adjust parametrized motor primitives to new situations". In: *Autonomous Robots* 33.4, pp. 361–379 (cit. on pp. 11, 12, 15).

Moulin-Frier, Clément, Pierre Rouanet, and Pierre-Yves Oudeyer (2014). "Explauto: an open-source Python library to study autonomous exploration in developmental robotics". In: *4th International Conference on Development and Learning and on Epigenetic Robotics*. IEEE, pp. 171–172 (cit. on pp. 12, 15, 33).

*MX-64AR, MX-64AT (Protocol 2.0)* (n.d.). Robotis (cit. on p. 20).

Nemec, Bojan and Aleš Ude (2012). "Action sequencing using dynamic movement primitives". In: *Robotica* 30.5, pp. 837–846 (cit. on p. 15).

Noble, B, SJ Harris, and K_ Dinsdale (1982). "Yield characteristics of aluminium–lithium alloys". In: *Metal science* 16.9, pp. 425–430 (cit. on p. 16).

Ojemann, George A (1982). "Interrelationships in the localization of language, memory, and motor mechanisms in human cortex and thalamus". In: *New perspectives in cerebral localization*, pp. 157–175 (cit. on pp. 8, 15).

Paraschos, Alexandros et al. (2018). "Using probabilistic movement primitives in robotics". In: *Autonomous Robots* 42.3, pp. 529–551 (cit. on p. 15).

Potter, Lisa M (Jan. 2019). *Shouldering the Burden of Evolution*. URL: `https://www.ucsf.edu/news/2015/09/131526/shouldering-burden-evolution` (cit. on pp. 8, 15).

Robotics, Open (2017). *About ROS*. URL: `https://www.ros.org/about-ros/` (visited on 03/18/2020) (cit. on p. 30).

ROCO504 (Jan. 2017). *Throwing arm with elastic energy storage*. URL: `https://github.com/ROCO504/2016-group4` (cit. on pp. 10, 15, 17).

Rolf, Matthias (2012). "Goal babbling for an efficient bootstrapping of inverse models in high dimensions". In: (cit. on pp. 11, 12, 15, 49, 60).

Senoo, Taku, Akio Namiki, and Masatoshi Ishikawa (2008). "High-speed throwing motion based on kinetic chain approach". In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3206–3211 (cit. on p. 41).

Thelen, Esther, Daniela Corbetta, and John P Spencer (1996). "Development of reaching during the first year: role of movement speed." In: *Journal of experimental psychology: human perception and performance* 22.5, p. 1059 (cit. on pp. 3, 11).

Von Hippel, William (2018). "The Social Leap". In: (cit. on p. 15).

Von Hippel, William and Dave Nussbaum (May 2019). *Making the Social Leap: A Conversation on How Our Psychology Evolved*. URL: https://behavioralscientist.org/making-the-social-leap-a-conversation-on-how-our-psychology-evolved/ (cit. on pp. 2, 8).

Wheeless, Clifford R, James A Nunley, and James R Urbaniak (2016). *Wheeless' Textbook of Orthopaedics*. Data Trace Internet Publishing, LLC (cit. on pp. 7, 15).

*Wiki* (n.d.). URL: http://wiki.ros.org/rosserial (cit. on p. 30).