Towards improving inter-robot knowledge transfer for developmental robots using non-stationary model.

Andani Alex Madula (alex@aims.ac.za)
African Institute for Mathematical Sciences (AIMS)

Supervised by: Dr Ndivhuwo Makondo
IBM Research Africa, South Africa

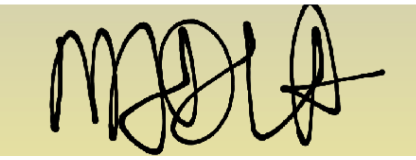
24 October 2019
Submitted in partial fulfillment of a structured masters degree at AIMS South Africa

Abstract

In this project the aim is to investigate the extension of the WNN regression model to be robust to corrupt transferred points. A robust WNN model will improve the effectiveness of knowledge transfer for accelerating the learning process of developmental robots. In particular, Non-Stationary Weighted Nearest Neighbors (NSWNN) could be a better inverse model learning and beneficial for knowledge transfer between different developmental robots. Virtual Robot Experiment Platform (V-REP) was used as a simulating environment. A series of self-exploration experiments was conducted motivated by research in developmental psychology, and the goal was to learn learning inverse models. The MATLAB robotics tool was used in all experiments for exploration and model learning. Short prior information about robot behavior and kinematics learning was provided to exploration algorithm. Two inverse models were learnt and compared. Average reaching error of reaching all test points and success rate as the ratio of the points reached within a threshold of 0.01 m was used as an empirical measure of the quality of the models learnt. The results show that NSWNN produces smaller reaching error and higher success rate in reaching the target points. Hence, it can be more beneficial for knowledge transfer. Future works are discussed in the closing chapter.

Declaration

I, the undersigned, hereby declare that the work contained in this research project is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.



Andani Alex Madula, 24 October 2019

Contents

1. Introduction

The most modern robots comprise flexible joints of different types, for example, spherical, circular, linear and others. Within technical documentation or guides, measurements of robot connections and joint limits are given by the supplier, and inverse and forward kinematic functions of the robot are available. There are several reason, however, for robots to undertake self-exploration and autonomously construct their kinematics models.

Although all the robots of the same design and manufacture are expected to have completely identical measurements and hence kinematic formulas, in reality that is not the case. At any point during robot development and installation, imprecisions and errors may be introduced. The measurements of the flexible linkages of the robot can also alter during operation, for example due to thermal expansion, structural contractions and mechanical wear and tear. Furthermore, before and regularly during service, a robot is used, and must be adjusted. Machine learning is one method to model robots to adapt to these changes. The body of the robot may suffer unwanted environmental damage. In cases where a robot can remain functional regardless of damage, the robot should be able to measure and adjust the internal model of its body by using self-exploration.

Autonomous robots are complex to model, and for these robots to explore the environment successfully requires a lot of time. Knowledge transfer between two or more robots within the environment can be beneficial. Such sharing of the kinematics data can accelerate the mapping of a sensorimotor of the target robot. However for a single robot, exploration should be done individually. The rest of the chapter is organized as follows. Section 1.1 provides an overview to robot kinematics learning, and briefly reviews model learning techniques, and exploration techniques from developmental learning.

1.1 Overview

In robot behavior modeling and control, kinematic modeling plays an important role. A robot kinematics model describes the motion of the robot with respect to some fixed reference frame, usually at the base of the robot neglecting any dynamical effects. This enables a robot to make quick predictions about whether or not it can reach target objects within the surrounding environment using its end-effector. In addition to that, for motion described in the task space of the robot, a robot kinematics model is required to map them into the joint space, where control mostly take place.

Kinematics also describes the relative positions and orientations (pose) between all the robot parts as well as with respect to the global reference frame. For instance, knowledge of poses of sensors located on the end-effector, together with knowledge of the location of the end-effector on the robot kinematics model, enables the transformation of the sensor data into a single reference frame for processing while the robot is in motion.

Traditionally, robot kinematics are designed analytically by engineers. This is done using parameters of the robot released by the manufacturer, such as link dimension, number of degree of freedom (DoFs), the configuration of joints and connections of the links. In cases where these parameters are not provided or are not correct, engineers must estimate them through analysis of the robot's structure, and this can result in modeling errors, requiring further calibration.

As stated, analytical modeling techniques fail when the exact values of such parameters are not provided, for instance in the case of complex robots with non-rigid links, flexible joints and when uncalibrated sensors provides noisy measurements. These techniques are not suitable for robots whose bodies change over time, possibly as a result of damage; as the static analytical model should be updated every time when there are changes in robot characteristics.

To resolve such issues, machine learning techniques are employed as an alternative where robot kinematics are learned from data generated by the robot through its sensor and actuators. However to learn kinematic models with success requires a large amount of data, and data collection from the robot can be time intensive process or computationally expensive for manipulators with high DoFs and large state spaces. In multi-robot systems data collection and model learning can take even longer, where each robot must learn self model from scratch.

Makondo and Rosman (2019), proposed to use knowledge transfer in the multi-robot case, where more experienced robots share data with new robots in order to accelerate their learning process. However, the authors noticed that for redundant robots (robots with more DoFs than required for a task) negative transfer occurs that is the final performance of the model with knowledge transfer is inferior to the model learned from scratch. Makondo and Rosman (2019), identified the use of a stationary weighted nearest neighbor (WNN) regression model for learning as one of the causes. In particular, since the transfer models used could be erroneous, some of the transferred points could be corrupt. The standard WNN regression model assumes all points are corrupt-free, and therefore suffers from corrupt transferred knowledge.

Project goal: In this project the aim is to investigate the extension of the WNN regression model to be robust to corrupt transferred points. A robust WNN model will improve the effectiveness of knowledge transfer for accelerating the learning process of developmental robots, as proposed by (Makondo and Rosman, 2019). In particular, we use a non-stationary weighted nearest neighbor regression model, in which in addition to weighting points according to their spatial distance from a query point, points are also weighted according to their 'age' – points added earlier in the robot's experience weigh less than recent points.

The intuition behind our extension is that since knowledge is transferred in the beginning of learning for the inexperienced robot Makondo and Rosman (2019), Makondo et al. (2018), Makondo et al. (2015) as the robot explores further and collects new data it will learn to rely on its own data rather the old transferred data. This has the benefit that before the new robot becomes experienced on its own, it relies on transferred knowledge, but eventually becomes experienced enough to use its own experience.

Instead of manual programming and fixed configuration, data and machine learning are typically employed as alternatives for robot model leaning, since learning from data could be more beneficial in case where robot physical parameters are inaccurate or not available at all. Unknown non-linearities can be handled while estimating the model from data. This could be important considering that modern robots are complex which also makes their model hard to model manually. In recent years learning robot models from data has attracted much interest and success.

In robot kinematics learning, models that are mostly learned are forward kinematic (FK) models, which predict the pose of the end-effector in the sensory space, the task space perceived by sensors tracking the end-effector given the robot joints angles in the motor space; and inverse kinematic (IK) models, which predict motor command (joint angles). Learning these models involves learning complex sensorimotor mapping, often in high-dimensional spaces.

Kinematics learning is mainly approached from two angles that supplement each other. Most robots generate large data in high-dimensions, hence machine learning algorithms have to be developed to learn regression models of such complex data. Online and incremental learning algorithms are highly preferable as they allow the robot to adapt environment adjustment or robot itself. Advanced statistical models have been developed, which typically make the assumption that good quality data is available in abundance and corresponding algorithms perform well depending on the quality of the data sets.

Developmental robotics attempts the study of robotics from the angle of building up capabilities progressively through embodied interaction with the physical world. With the idea of designing exploration mechanism that equip robots with capabilities of autonomously exploring their surrounding environments in order to generated data as the robot explores and sensorimotor mappings can be modelled from data. This work has led to several exploration strategies, for which some of them are motor babbling, which is exploration of motor space and goal babbling, which is exploration in sensor space. Robots can either explore those spaces randomly or actively making use of interest models in a given interest space resulting in motor babbling strategies when it corresponds to the motor space and in goal babbling strategies when it corresponds to the sensory space. Interest models are designed in such a manner that exploration maximizes some measure of progress(Makondo and Rosman, 2019).

Advanced model learning techniques with exploration give stand-alone robots abilities and capabilities for self exploration and learning sensorimotor mappings. Therefore, for a robot case, exploration in the environment have to be done alone and the robot explores its own capabilities. However, for the multi-robot case, it may be advantageous for the robots to have abilities to share knowledge they have acquired through their individual exploration. Such knowledge transfer can influence in speeding up learning and exploration of other robots without much experience about the environment.

The remainder of this project is organized as follows. Chapter 2 provides some background studies done in preparation of the project. Chapter 3 contains the method where we explain the online goal babbling method and sensorimotor models. To understand the content of the project the reader is recommended to go through Chapter 2 and Chapter 3. In Chapter 4 we explain the experiment conducted and give some short discussion. The ending chapter is Chapter 5 where we give conclusion and suggestions for future work.

2. Related work

The chapter provides a review of some related work including Section 2.1 example of forward and inverse models of simple 2D manipulator. Section 2.2 gives some insight in general robot kinematics. Section 2.3 involves ideas in learning kinematic models. Section 2.4 gives a short description of accelerating kinematics learning, including techniques based on sensorimotor mapping. Section 2.5 describes the general problem of learning kinematics from real robot data.

2.1 Robot Arm Configuration

The inverse kinematics problem is to find a vector of joint factors that produce an ideal end-effector position. If there exist a unique vector of joint angles for a target end-effector position, then there is a well-defined inverse to the forward kinematics, hence the inverse function is said to be well-posed. However, if the solution of the forward function is not unique, then the inverse problem can be ill-posed. In reality, solving the inverse kinematics problem is always a hard task and can often lead to an infinite number of solutions.

The manipulator's kinematics criteria shown in Figure 2.1 are based on the potential formula for forward kinematics. In forward kinematics, the idea is to map out joint angle coordinates say (θ_1, θ_2) to (x, y) for the observation location (Hasan et al., 2006). The mapping of the forward model is expressed as follows,

$$x = a_1 \cos\theta_1 + a_2 \cos(\theta_1 + \theta_2) \tag{2.1.1}$$

$$y = a_1 \sin\theta_1 + a_2 \sin(\theta_1 + \theta_2) \tag{2.1.2}$$

where θ_1 and θ_2 represent joint angles of arm segments, and $l_1 = a_2$ and $l_2 = a_2$ are lengths of respective arm segments. The above relation of x and y shows the problem of inverse kinematics with unique mapping for joint space to task space.

The computation of θ_1 and θ_2 from end-effector position (x, y) defines the inverse kinematics problem which is described as follows.

$$\theta_2 = \arccos\left[\frac{x^2 + y^2 - (a_1^2 + a_2^2)}{2a_1a_2}\right] \tag{2.1.3}$$

$$\theta_1 = \arctan\left(\frac{x}{y}\right) - \arctan\left[\frac{a_2 \sin\theta_2}{a_1 + a_2 \cos\theta_2}\right] \tag{2.1.4}$$

Over the range of angles $\arccos$ is not unique, there might be more that one possible orientation result from θ_1 and θ_2. The pose on the arm can be positioned upward or downward, but pointing at the same end-effector point (x, y). Hence, the inverse kinematics θ_1, θ_2 is not unique, that is where machine learning is involved.

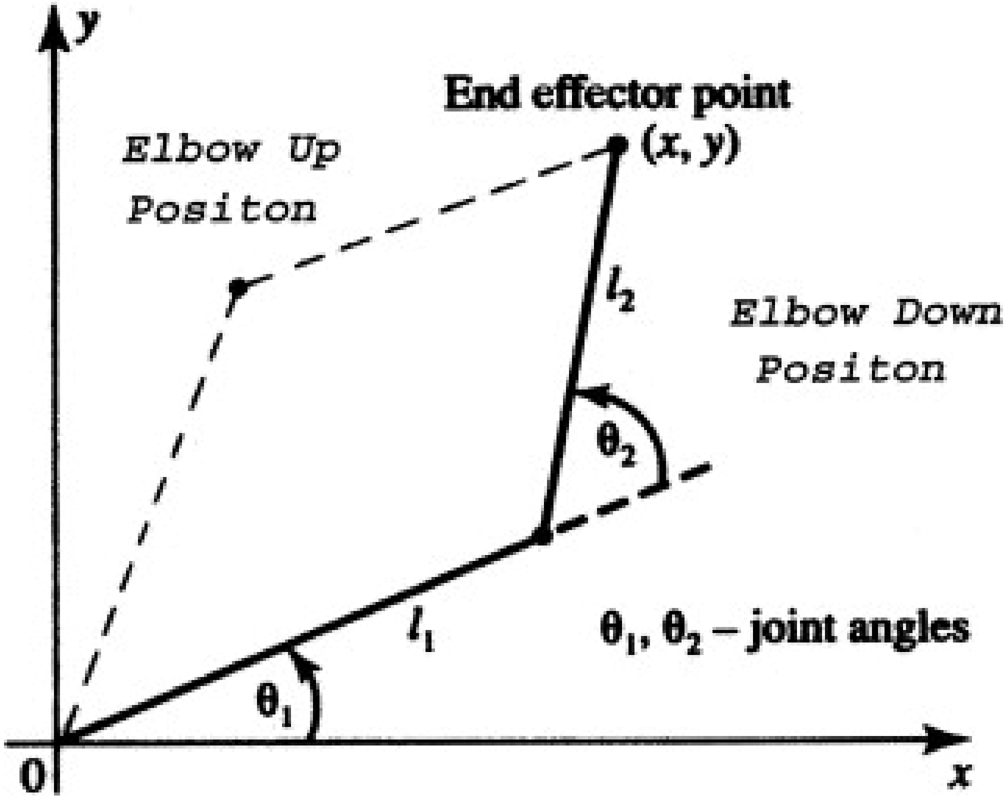


Figure 2.1: 2D structure of the robot arm, adopted from (Hasan et al., 2006)

2.2 Robot Kinematics

In general, kinematics is the analytical study of the geometry of motion of a mechanism focusing on two mains aspects, a fixed reference co-ordinate system and disregard to the forces that cause the motion. Knowledge of both robot spatial arrangement and a means of reference to the environment is needed for robot control. Denavit–Hartenberg and the screw-based methods are usually used to model kinematics models (Rocha et al. (2011)). Although analytical kinematic models are reliable, however they can not manage sudden morphological changes of the robot. As an example let's consider that the robot might loose one limb or change joint configuration settings during an accident, but should still be able to re-calibrate its kinematics and be functional. The developmental perspective proposes a solution that helps the robot to learn this sensory-motor mapping from self-exploration. There are different approaches which widely investigated the developmental approach to learning sensory-motor systems through self exploration, particularly learning from visuo-motor coordination. We single out two key approaches: motor babbling on one side, and target babbling on the other. We look at each of these strategies in this work.

In motor babbling Kuniyoshi et al. (2004), Schillaci (2014) random DOF motion are generated. Kuniyoshi et al. (2004) notice self-exploration of the embodiment on a simulated baby robot (Wieser and Cheng (2016)). The limbs are randomly driven into simulated water to acquire a body map using spatio-temporal correlation of the sampled tactile and motor signals. Schillaci (2014) employs motor babbling to study of the generation and sampling of rich proprioceptive and visual data, followed by a K nearest neighbor calculation (kNN) that implements both the forward and the inverse model, with the objective of learning a body map. Takahashi et al. (2014) uses babbling to supply enough data to train neural dynamic networks to obtain the body image.

Baranes and Oudeyer (2013a) reveals that exploring the goals in the task space is faster than exploring the motor space for the learning of inverse models. They use a method of exploration metric to direct training by selecting appropriate goals. Rolf et al. (2011) worked on a goal babbling exploration strategy, on which they showed that a goal-based strategy can be more effective than action based on motor babbling. They also explain the importance of adding exploratory noise to avoid singular configurations. They tested their thoughts in a bionic elephant and a humanoid robot. In their work, the observation space is legitimately related to the situation of a robot's end-effector in Euclidean space. In this work we have applied a goal-based strategy to self exploration of the manipulator.

Schmerling et al. (2015) address the kinematics learning problem, by suggesting a two-level model system, adding two distinct inverse models: one arm and one for the head kinematics. In this learning system, they explore the roles of motor babbling and curiosity-driven goal babbling. Makondo and Rosman (2019) use a goal babbling techniques to learn arm kinematics models of developmental robots that autonomously explore the environment, to explore knowledge transfer in multi-robot systems. In their work they showed that an inexperience robot that is autonomously exploring could benefit from the experienced robot by transfer of its kinematic data. They used two-links and three-links robot to study their transfer kinematics data between two robots. Transfer of data to a target robot makes the sensorimotor mapping converge faster and achieves better performance than robot which continues with individual exploration. Knowledge transfer analysis was performed in simulation, ranging from simple robots to redundant systems.

2.3 Kinematics Learning

With robot kinematics data available, learning forward kinematics is a simple task, as the mapping from the motor space to sensory space is uniquely determined and standard regression techniques can be used to learn such models. However, learning inverse kinematics on the other hand can be a complex problem to solve, as the problem is ill-posed for redundant systems, where the dimension of sensory space m is less than the dimension of motor space, d. In non-linear problems, and in such similar cases there may be multiple solutions, which means there are infinitely many joints configurations leading to the same end-effector pose, leading to the multi-valued regression problem. Therefore, standard regression techniques do not fit well for such cases. Several model learning algorithms have been proposed in the literature, some are based on neural networks and advanced statistical models. There are several techniques proposed as to conquer non-uniqueness of the inverse kinematics, including learning modular neural networks, learning on the velocity level local optimization and nearest neighbor. We use a nearest neighbor regression model, which easily handles the redundancies in the systems (Hasan et al., 2006).

2.4 Accelerating Kinematics Learning

Very little research has been done to attempt knowledge transfer across robots for accelerating kinematics models learning. Schmerling et al. (2015) employ Procrustes analysis to bootstrap the new robot with experience a robot kinematics data, to speed up learning of kinematics model. Bócsi et al. (2013) employed Procrustes Analysis (PA) to attempt to transfer kinematics data of experienced robots for learning forward kinematics, where analytical models of the robots were used to generate data. Chen et al. (2016) proposed an approach for transferring skills from human demonstrations for learning inverse kinematics of a soft-tendon driven manipulator. Reinforcement learning techniques were used to improve the transferred skills. Their targeted application was minimally invasive surgical tasks. Recently, Makondo and Rosman (2019) employed Local Procrustes Analysis Makondo et al. (2015) to transfer knowledge across developmental learning robots for kinematics models. In this project we aim to improve work proposed by Makondo and Rosman. In developmental robotics, training has recently been fascinating, and robots have shown success in allowing their sensorimotor mapping to be established increasingly by interacting with the environment. Nonetheless, a significant part of the work for kinematic model training has centered on the single-robot case, and this typically involves long training times.

Makondo and Rosman, proposed using knowledge transfer to accelerate learning of kinematic models in the multi-robot case. However, their use of a static weighted nearest neighbor regression model for representing the sensorimotor mappings suffers from negative transfer, caused by corrupt transferred data points. In this work we propose using a non-stationary weighted nearest neighbor regression model, that learns to rely on more recent data points and forgets old corrupt data points that were transferred.

2.5 Problem Statement

In this project, we consider the relationship between the motor commands $\mathbf{q} \quad Q \qquad ^{d}$ (joint position) and their respective points in the sensory space $\boldsymbol{x} \quad X \qquad ^{m}$ (hand position), with d being the number of DoFs and m the dimension of the sensory space. From the motor space to the sensory space, we use FW function $L(\mathbf{q}) = \boldsymbol{x}$, which basically describes the unique mapping between the two spaces. FK is generally used to predict $\boldsymbol{x}$ which is the position of a hand given the some joints configuration $\mathbf{q}$, while IK function $L^{-1}(\boldsymbol{x}) = \mathbf{q}$, is used to predict the settings of some joints configuration $\mathbf{q}$ given the hand pose $\boldsymbol{x}$. The two functions, FK and IK are used to model the sensorimotor mappings of the agent and are referred to as sensorimotor models (Makondo and Rosman, 2019).

The inverse model function is complex and is not unique in cases where $m < d$, due to infinitely many joint configurations for just one hand pose, hence there are infinitely many solutions as previously discussed. The robotic agent learns its sensorimotor model by collecting $(\mathbf{q}^i, \boldsymbol{x}^i)$ through the interaction with the environment.

Learning both forward and inverse kinematics for autonomous robot exploration, exploration can be a long task. It has been shown that the goal babbling technique, which is an exploration in sensory space, can learn inverse kinematics models for redundant systems more efficiently, compared to motor babbling which is an exploration in the motor space, the reason being dimensionality of sensory space which is usually lower than of motor space for manipulations. In addition to that, the use of active learning techniques has improved efficiency in exploration, where robots explore the environment in a way that maximizes the measure of learning progress (Baranes and Oudeyer, 2013b).

Here we work on the improvement of exploration and learning kinematics models, by employing time-weighted nearest neighbor or non-stationary nearest neighbor as to train kinematics models to forget old points or old neighbors to weigh less in the estimation of a query point. We employ a simple random goal exploration mechanism, a simple model learning algorithm, using non-stationary nearest neighbors.

2.6 Definitions

2.6.1 Definition. Manipulator is an autonomous robot that is used to manipulate an object without direct contact (hand robots)

2.6.2 Definition. Robot kinematics involves movement of a robot without consideration of external forces, influencing movements.

2.6.3 Definition. Developmental robotics is the study of developmental mechanisms in robotics.

2.6.4 Definition. Boostrapping is a model learning of robot sensorimotor mapping, and its environment, from zero prior information other than basic semantic.

2.6.5 Definition. Goal Babbling is a boostrapping of a coordination capacity by constantly attempting to achieve many goals associated with that skill.

2.6.6 Definition. Online Goal Babbling is an algorithm which allows Goal Babbling to be performed online.

2.6.7 Definition. End-effector is a device at the end of a robotic arm which interacts with the environment.

2.6.8 Definition. V-REP is a robotic simulator environment.

2.6.9 Definition. LPA is statistical shape analysis used to map set points between two or more robots during knowledge transfer.

3. Method

In this chapter we discuss the model used and supported by Section 3.1 which presents the technique used for continuous model learning. Section 3.2 presents the sensorimotor mapping, involving nearest neighbor search.

3.1 Online Goal Babbling

The main concept that ties together the goal-oriented initiatives and initial feedforward control is to take redundancy as an opportunity to reduce the demand for exploration, rather than challenge that needs to be addressed. If there are several ways to accomplish a common goal, it is not essential to know all of them. However, to adhere to this without comprehensive exploration, it will require an exploration strategy that will generate relevant training data. The motivation of goal babbling is that early goal-oriented initiatives of infants do not only show early exploitation of data, but that they account for every mechanism to generate that knowledge by exploration and for that reason enable efficient learning of a valid solution for the coordination problem. We employ the idea for kinematics learning.

Machine learning works better when dealing with a bundle of static data, however in cases where the data is changing with time online learning algorithm are preferred. Online learning is generally considered to be a stochastic approximation for batch gradients. When online learning happens through goal-oriented discovery, the situation is different because the distribution of the example changes over time based on the learning dynamics. Types of online learning from goal-oriented exploration are investigated for tuning inverse models in different flavors such as differential kinematics and operational space control. A key challenge is that a positive feedback loop on the instances is extended to online learning. Perturbations are either self-generated exploratory noise or from external sources reinforced by learning that can trigger the inverse model to become erratic or drift (Rolf et al., 2011). In the next section we discuss an online goal babbling algorithm.

In general the idea of learning $\hat{L}^{-1}$, which is an inverse estimate of L, is to explore the consequence of motor commands $\mathbf{q}_n$ over iteration steps. Then to determine the corresponding end-effector position $\boldsymbol{x}^i$. Henceforth $(\mathbf{q}^i, \boldsymbol{x}^i)$ is then use for supervised adaptation of the $\hat{L}^{-1}$. In a goal babbling algorithm, the robot randomly explores its sensory space X, and that is achieved by choosing a random goal $\boldsymbol{x}_g$, and employs the current inverse estimate to predict the command $\hat{\mathbf{q}}_g$ required to reach $\boldsymbol{x}_g$ with its end-effector. Note that at the beginning of the learning process, prediction of the motor commands $\hat{\mathbf{q}}_g$ from the estimate L^{-1} will be poor resulting in $\boldsymbol{x}^i = \boldsymbol{x}_g$. A general exploration and learning strategy is illustrated Fig 3.1. Primarily, a model learning algorithm is embedded in the system that incrementally learns inverse model $\hat{L}^{-1}$ as the robot continuously explores the environment and generates data, and also exploration is guided by predicting the motor estimate input for the next chosen goal to be explored. In general, a forward kinematics model can also be learned, however mostly it is not needed by the goal babbling technique.

The goal blabbing strategy allows the robot to cover the sensory space more efficiently, avoiding a waste of time in high-dimensional redundant systems, as motor blabbing may execute motor commands that basically reach the same goal. Exploration is initiated near some $\mathbf{q}^{home}$, which results in an inverse estimate $\hat{L}^{-1}$ learning to predict commands locally around $\mathbf{q}^{home}$ and as time progresses, due to some exploration noise added to the motor commands, exploration gradually moves away from $\mathbf{q}^{home}$ to more varied movements. In this case, robot exploration focuses on learning only simple movements that reach

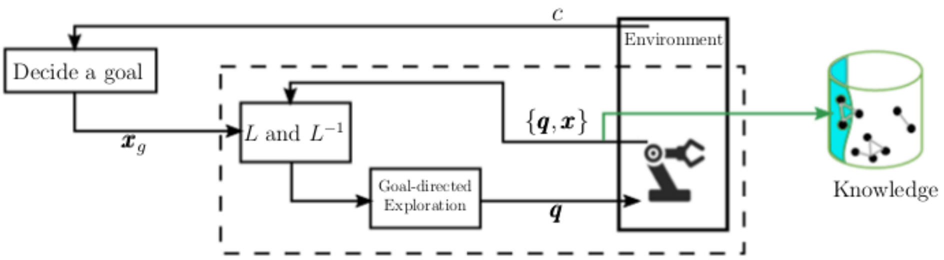


Figure 3.1: Illustrate a goal babbling structure. The next target goals are generated depending on environmental settings c, which the agent must attempt to randomly reach all goals. Without prior knowledge of the inverse model L^{-1} , the agent explores its environment by executing random motor commands $\boldsymbol{q}$ and observing the consequences end-effect position $\boldsymbol{x}$, and uses the generated data $\{\boldsymbol{q}, \boldsymbol{x}\}$ to update its models, L and L^{-1}. This process repeats until the agent is competent at reaching the goals and has gained knowledge about its internal models that it will use in the future when encountering the same goals. Adopted from Makondo and Rosman (2019).

all goals in the sensory space and refining the changes at a later stage; however, this leaves the strategy being biased.

In this essay, we employ a simple online goal babbling scheme proposed in Rolf et al. (2011), which is summarized in Algorithm 2. A data generated by the robot is used for online updating of parameterized inverse model L_θ^{-1}. As inputs, the algorithm must be provided with sensory space X_{bounds}, the number of points T on the path when reaching for selected random goal $\boldsymbol{x}_g$ and a home configuration $\mathbf{q}^{home}$. The inverse estimate function is initialized with the home configuration : $\hat{L}_{\theta_0}^{-1} \leftarrow train(\mathbf{q}^{home}, \boldsymbol{x}^{home})$, such that it always predicts the home configuration in the beginning .

In exploration of sensory space, random goals $\boldsymbol{x}^\star$ are iteratively chosen form X_{bounds}, and the robot attempts to reach a goal from the current configuration $\mathbf{q}^{home}$ in the beginning , via $T-1$ intermediate targets $\boldsymbol{x}_t$. A linear interpolation defines the targets between the current robot state $\boldsymbol{x}_0$ and the goal $\boldsymbol{x}_t : \boldsymbol{x}_t^\star = \frac{T-t}{T}.\boldsymbol{x}_g + \frac{t}{T}.\boldsymbol{x}_{t-1}$, where $t = 1, ..., T$ denotes the sub-steps within one movement. The robot reaches intermediate targets by use of current estimate of the inverse model to infer a motor command $\hat{\mathbf{q}}_t$. For biased exploration strategy into other areas of the sensory space , we add some exploratory noise $E_t(\boldsymbol{x}^\star)$ to the inferred motor command and the noisy command is sent to the robot. The parameters of inverse estimate or forward estimate (if needed) are updated using generated $(\mathbf{q}^t, \boldsymbol{x}^t)$.

We model exploration noise as a randomly chosen linear function,

$$E_t(\boldsymbol{x}^\star) = A_t \cdot \boldsymbol{x}^\star + \mathbf{b}_t, A_t \qquad {}^{m\times d}, \mathbf{b}_t \qquad {}^{d}. \tag{3.1.1}$$

At $t = 0$, entries A and $\boldsymbol{b}$ are chosen independent identically distributed from a normal distribution with zero mean and variance σ^2, and then perturbed slowly with a normalised Gaussian walk, to produce a smooth , slowly changing linear function $E_t(\boldsymbol{x})$ that adds smooth variations to the motor commands applied to the robot (Rolf et al., 2011).

Algorithm 2 Online Goal Babbling

Input: End-effector space bounds $X_{bounds} = \{\boldsymbol{x}_{min}, \boldsymbol{x}_{max}\}$
Input: Time for reaching selected goals
Input: Home configuration $\mathbf{q}^{home}$
Initialise learner: $L_{\theta}^{-1} \leftarrow L_{\theta_0}^{-1}$
for number of epochs **do**
  Select a target $\boldsymbol{x}^{\star}$ randomly with X_{bounds}
  Select a sequence of targets towards $\boldsymbol{x}^{\star}$: $\boldsymbol{x}_t, t = 1, ..., T$
  for t $[1, T]$ **do**
    Estimate motor command: $\hat{\mathbf{q}}_t = \hat{L}_{\theta}^{-1}(\boldsymbol{x}_t)$
    Generate exploration noise :$E_t(\boldsymbol{x}^{\star})$
    Execute perturbed motor command $\hat{\mathbf{q}}_t + E_t(\boldsymbol{x}^{\star})$ on robot
    Update model with generated data: $\hat{L}_{\theta}^{-1} \leftarrow train(\mathbf{q}^t, \boldsymbol{x}^t)$
Output: inverse estimate $\hat{L}_{\theta}^{-1}$

3.2 Sensorimotor Models

Suppose there are two objects of different sizes but equal weight. When people lift those objects, initially, they use too much force for the large object and too little force for the small object. Nonetheless, through repeated lifts of the two objects, they learn to eliminate the size-weight correlation used to measure the requirements of force and accurately scale their lifting forces to the objects real and equivalent weights. Hence, sensorimotor memory from previous lifts thus dominates knowledge about visual size in terms of force prediction (Flanagan et al. (2006)).

Here, we employ a simple memory-based sensorimotor model based on nearest-neighbor look-up to represent forward function L and inverse function L^{-1}. The implementation of the model is simple. Learning with this model consists of iteratively storing example pairs $(\mathbf{q}^i, \boldsymbol{x}^i)$ into a database.

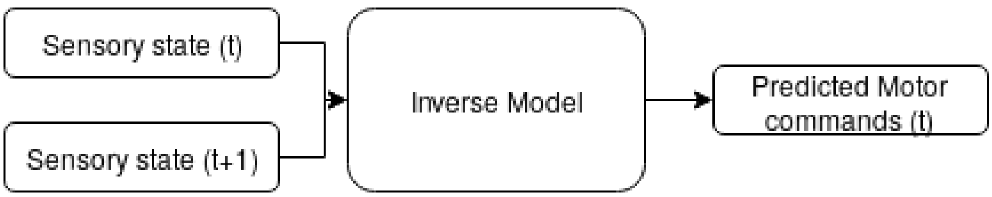


Figure 3.2: Illustration of Inverse Model

3.2.1 Illustration of inverse model. We learn sensorimotor models (inverse models in this case) using the NN search method. Some of the NN search methods are actively used are Encoding, Computer vision, artificial skin and closest search of a codebook. Thus, optimal NN search techniques rely on building structures to achieve exclusively. In our case NN return the closest neighbor to the query point. For example we assume given a end-effector position $\boldsymbol{x}$ X, the NN inverse searches through the data base and returns the motor command $\boldsymbol{q}$ Q that corresponds to the end-effector position $\boldsymbol{x}$.

3.2.2 K-nearest neighbors(K-NN). The K-NN algorithm is proposed to sample out K training samples that are nearest to the query point in the training set. In addition, evaluate the average of the K training samples; then assign the average to the query point, where K is the number of training samples. In K-NN, all samples listed in the same category in the feature space are assumed to have same characteristics, the category includes the closest adjacent K samples. In the determination of the regression decision, the procedure will determine the estimate of query point only according to the group of the nearest one or several samples. In addition, the K-NN algorithm is only applicable for a minimal number of adjacent samples in regression decision making (Fan et al., 2019) (see Figure 3.3).

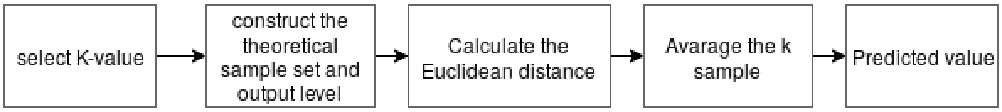


Figure 3.3: K-NN regression algorithm

The algorithms relies on the metric distance $d(f, y)$ between forecast data and unknowns,

$$d(f, y) = \sqrt{\sum_{j=1}^{k} (f_j - y_j)^2}, \tag{3.2.1}$$

where f_j represent neighbors of unknown data points and y_j represent the known data points and $y_1 < y_2 < ... < y_k$. Hence the shortest load forecasting for regression is described by,

$$s_i = \frac{1}{k} \sum_{j=1}^{k} f_j, \tag{3.2.2}$$

with s_i representing the i-th forecasted value, which represents the forecast value of the j-th known data points of y_i. In this case, we consider given **q** Q the forward K-NN template search the database and return K observer $\boldsymbol{x}$ X corresponding to the nearest $\boldsymbol{q}$-to motor control.

3.2.3 Weighted Nearest Neighbors (WNN). It implements K-NN, and it just adds weights to the k samples. The ideal plan being for points with large distances to weigh less than the closest ones. Weights are defined as the inverse of the Euclidean distance (see Figure 3.4).

$$w_i = \frac{1}{d(f_i, y_i)}. \tag{3.2.3}$$

Hence, the final forecasted value (s_i) of each data point was calculated as

$$s_i = \frac{\sum_{j=1}^{k} w_{i,y_j} \times y_j}{\sum_{j=1}^{k} w_{i,y_j}}. \tag{3.2.4}$$

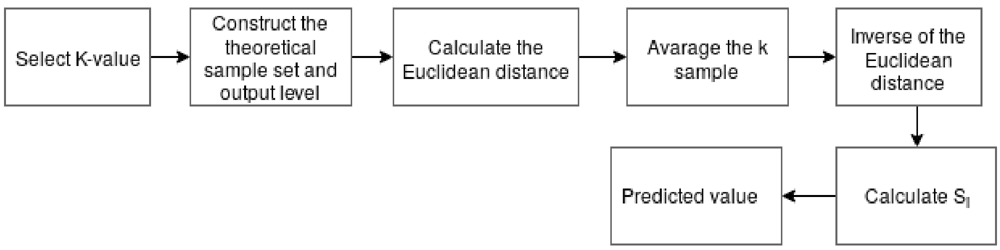


Figure 3.4: WNN regression algorithm

The model looks up the K nearest neighbors of $\mathbf{q}^\star$ in the database and returns the weighted average of the K corresponding end-effector points $\{\boldsymbol{x}^k\}_{k=1}^K$ to carry out a forward kinematics prediction for a given motor command $\mathbf{q}^\star$.

Similarly, to compute the inverse kinematics prediction for a given end-effect point $\boldsymbol{x}^\star$ the model looks up the K nearest neighbors of $\boldsymbol{x}^\star$ in the database and returns the corresponding $\mathbf{q}^{near}$. The inverse model computes the opposite of the forward model that is, it returns estimated motor command $\boldsymbol{q}_t$ to go from a current sensory position $\boldsymbol{x}_t$ to a state of interest $\boldsymbol{x}_{t+1}$. Then looks up the K nearest neighbors $\{\boldsymbol{x}^k\}_{k=1}^K$ of $\mathbf{q}^{near}$ and then returns the weighted average as the prediction $\mathbf{q}^\star$, for example,. This resolves the cases of redundancies that arise due to infinitely many solutions of L^{-1} for high-dimensional systems. This is in contrast to finding K neighbours in the end-effector space and averaging the corresponding motor commands. In both forward and inverse predictions, the Inverse Distance Weighting (IDW) and Non-Stationary Weighting methods are used to compute the weighted average, where the inverse of the distances and decaying time function of the points to the query point are used as weights.

3.2.4 Non-Stationary Weighted Nearest Neighbor (NSNN). This is an improvement of WNN, where the weighting depends on age. We weigh the points depending on how old they are in the data base. In this case we employ a Gaussian kernel, and the algorithm is similar to Figure 3.4, the only difference is the weighting.

$$K_{y_i} = \exp\left(c - c/(1 - t_{y_i}/l)^2\right), \quad t_{y_i} < l \tag{3.2.5}$$

where l and c are constants, and the new weight w_i is defined as,

$$w_i = \frac{1}{d(f_i, y_i)K_{y_i}}, \tag{3.2.6}$$

$$s_i = \frac{\sum_{j=1}^{k} w_{i,y_j} \times y_j}{\sum_{j=1}^{k} w_{i,y_j}}, \tag{3.2.7}$$

where t is the age of points in a database $t = 1, 2, ..., T$, classification being that $t = 1$ weighs more than $t > 1$. In the next chapter we present how the experiment was carried out and we also present some preliminary results.

4. Experiment and Discussion

In this section we report on some preliminary results demonstrating model learning using online goal babbling with non-stationary nearest neighbors and weighted nearest neighbors with transfer and only model learning. We report on the benefit of using non-stationary nearest neighbors as a kinematics model learner. All experiments were simulated using V-REP (see Figure 4.1).

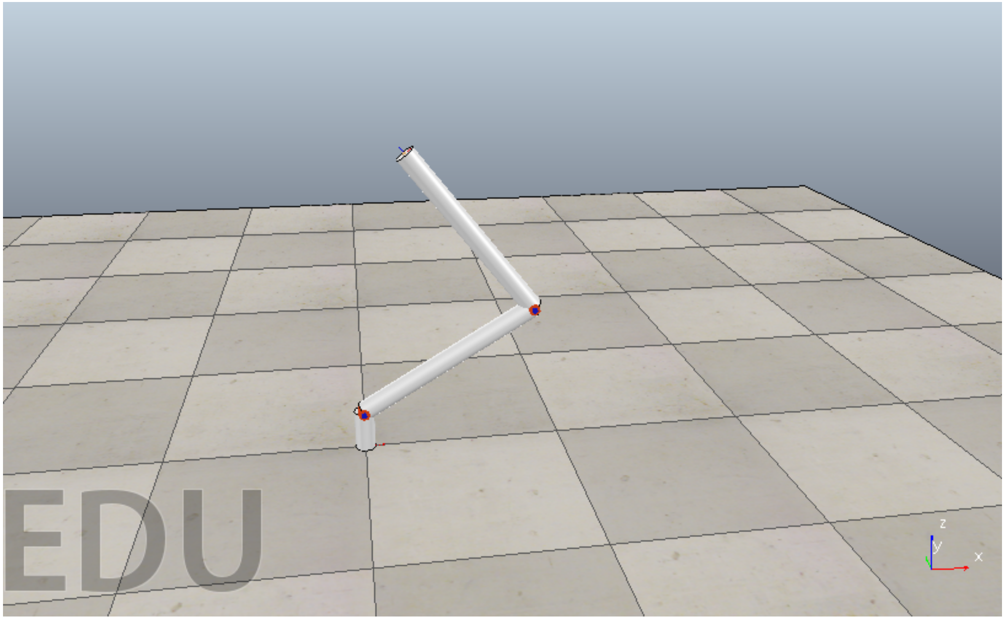


Figure 4.1: Example of V-REP two-link robot

4.1 Two-link Models Learning

First, we present results for kinematics learning for two-link planar robots with $2D$ task spaces, to illustrate the comparison of the inverse models.

A series of experiments was conducted to test the quality of kinematic models learnt, with Petercorke MATLAB robotics toolbox. In each experiment conducted, the simulation was executed for 1000 iterations, in each episode of 10 per experiment. In each iteration, a motor command $\boldsymbol{q}$ Q was obtained using goal-based exploration strategy. Then the motor command obtained from the goal-based method was executed in the simulation environment, and a corresponding end-effector position $\boldsymbol{x}$ X was obtained. The two vectors were used to model the database with $(\boldsymbol{q}, \boldsymbol{x})$.

As suggested Rolf et al. (2011), we added exploratory noise to motor commands and executed during learning process. The goal driven exploration may generate degenerated data sets and get stuck in intermediate local minima if the exploratory noise is not added. In this experiment we have used Gaussian sigma $\sigma = 0.05$. During testing of the quality of model learnt there was no exploratory noise added.

We evaluated how far the models have learnt using some test points or goals generated, for which after every 100 iterations of 1000 we evaluated the learning by testing if the robot is able to reach goals after a bit of training. We use two measures of progress which is reaching error and reaching rate. If the model attempted to reach target goals within the specified threshold of 0.01 m, the reaching error for each point was calculated and recorded. After-all the mean reaching error and reaching rate calculated for all target goals was used as an observed measure of model quality. For good quality models, the observed measurement should approach zero. To ensure that the model learnt to reach all points in space without any biases, the target goals or test points were generated in such a way that the distribution of the points was even in the work space.

In the process of evaluating the models learning, we basically compare how the reaching error function behaves from when the robot started collecting data to the end of simulation time. In this case, we only conducted the experiment for learning inverse kinematics model since learning forward kinematics is easier, and the same procedure for learning should be applicable. In the experiment, we use parameter k=3 for learning sensorimotor models using memory-based sensorimotor models.

4.1.1 Two-link Inverse Models Comparison. The experiment showed that non-stationary nearest neighbor inverse model performs better overall, however initially the two models are equivalent and differ around 200s (see Figure 4.2). The parameter used are link1=0.7, link2=0.5 and motor1=motor3=$[-\pi/2, \pi/2]$.

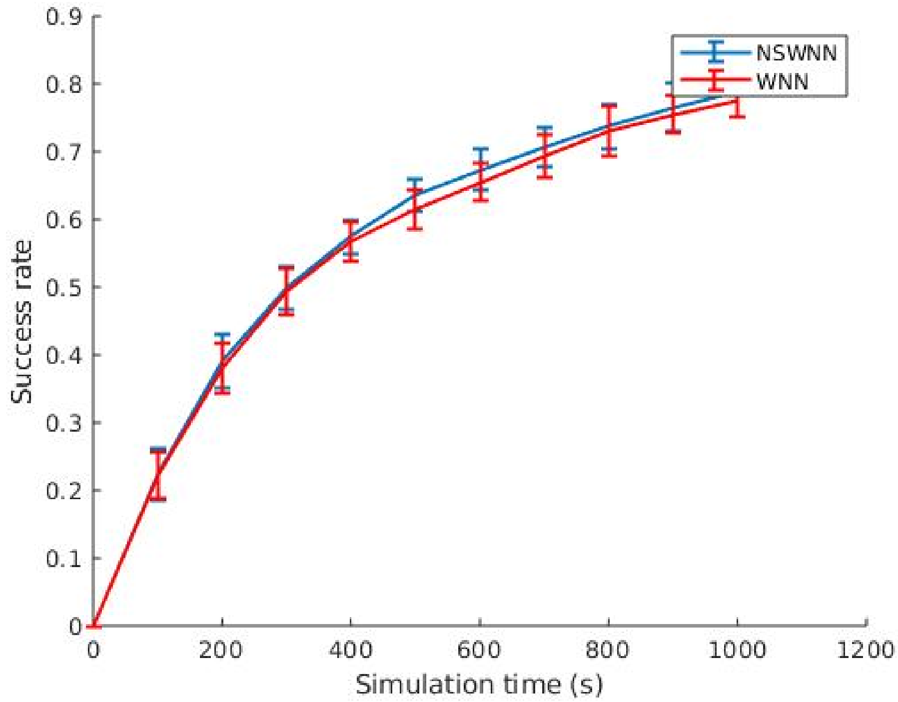


Figure 4.2: reaching rate of NSWNN and WNN inverse models

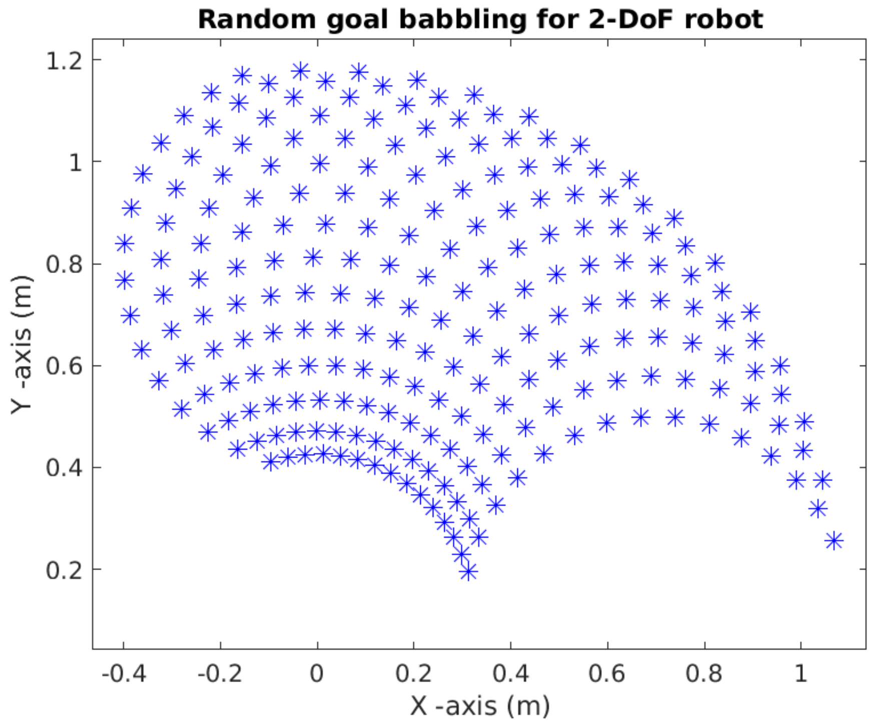


Figure 4.3: 2D task space structure of the robot and also showing the test points which were generated fairly, to avoid robot performing poorly in some part of the work space.

4.2 Three-link Model Learning and Transfer

We repeated the experiment, but this time using a three-link robot. Due to redundancy we increased the simulation time to 2000s, and the number of runs was not changed. We plotted work-space for robots with parameters in Table 4.1 and test points generated (see Figure 4.4).

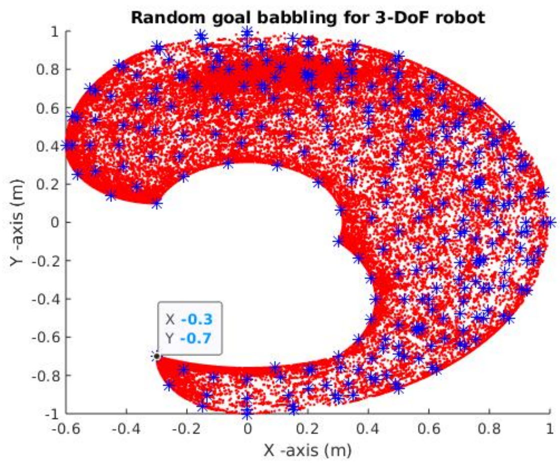


Figure 4.4: Illustration of task space of target and source robot in 2D.

4.2.1 Three-link Inverse model comparison. The experiment of inverse model learning shows that non-stationary (NSWNN) weighted nearest neighbor algorithm learns more accurately compared to weighted nearest neighbors (WNN). This implies NSWNN always produces better motor commands for any given end-effect position. However, within 600s both models are likely to give the same motor commands, and the variation from about 600s until the end of simulation time (see Figure 4.5).

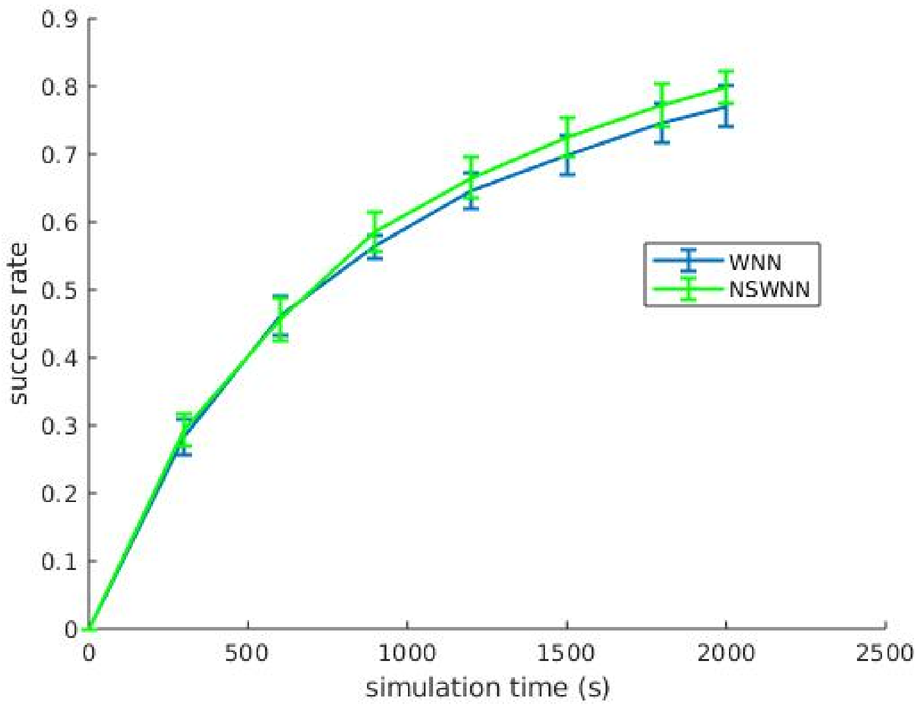


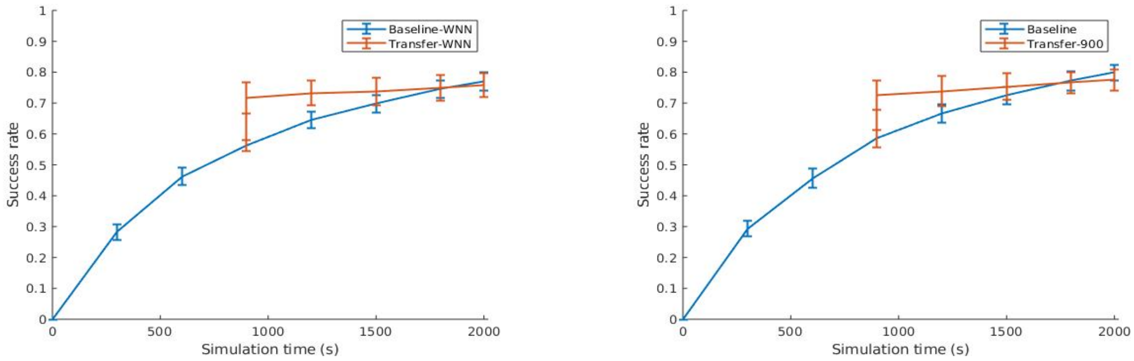
Figure 4.5: reaching rate of NSWNN and WNN inverse models

4.2.2 Three-link Inverse Models comparison with Transfer . In this part of the experiment, we analyse WNN and NSWNN sensorimotor models for the two-link planar robots. The goal of the research is to illustrate how the use of non-stationary nearest neighbour can be more beneficial for knowledge transfer and model learning. The kinematic parameters of the robot are shown in Table 4.1. The dimensionality of the motor is 3, that is $\boldsymbol{q}_s, \boldsymbol{q}_t, \boldsymbol{x}_s, \boldsymbol{x}_t$ [2] and sensory spaces that is $\boldsymbol{X}_s, \boldsymbol{X}_t, \varepsilon_s, \varepsilon_t$ [4]. More details of knowledge transfer in explained in (Makondo and Rosman, 2019).

Three-link Planar Robot		
Parameter	Source Robot	Target robot
Link 1	0.4	0.5
Link 2	0.3	0.25
Link 3	0.3	0.5
Motor 1	$[-\pi/2, \pi/2]$	$[-\pi/2, \pi/2]$
Motor 2	$[0, \pi/2]$	$[0, \pi/2]$
Motor 3	$[-\pi/2, \pi/2]$	$[-\pi/2, \pi/2]$

Table 4.1: Parameters {lengths(m) and joint limits(rad)} of three-link robot.

The results showed that there is not much difference between NSWNN and WNN transfer models. The transfer was done in 900s, within that 900s the target robot collected the sample which allows the LPA to map source robot data to the target robot. Here, we assume that we have some transfer data.



(a) Reaching rate WNN

(b) Reaching rate for NSWNN

Figure 4.6: Knowledge transfer with Local Procrustes Analysis.

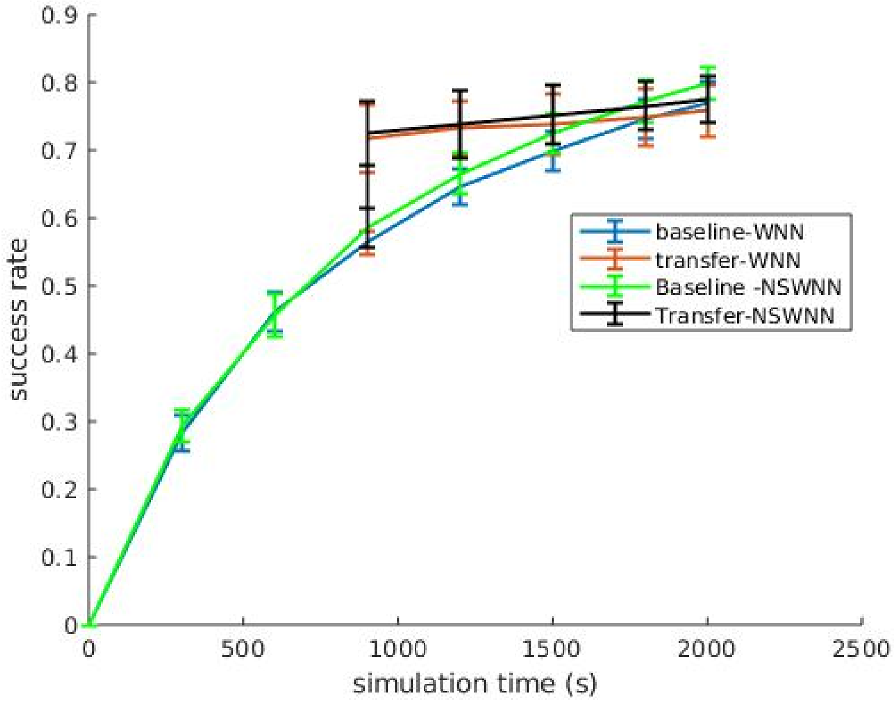


Figure 4.7: over plot of reaching rate of NSWNN and WNN

4.3 Discussion

Firstly, the results presented in the previous section shows that NSWNN is a better inverse model based on experiment report from the two-link robot and three-link robot which is more complicated. However, there was slight improvement in the transfer model, which implies that there is still some bad transfer in the model — our inverse learning kinematics only improved in the case of self-exploration. The NSWNN model depends on the choice of the weighting function, which we believe can be improved by investigating a suitable function that does the weighting properly. In this work the choice of age weighting was an exponential decay function, which decays smoothly. The idea was that old points in the database should weigh less. From Equation 3.2.5 the parameters of choice $c = 0.5$ and $l = 0.5$.

Secondly, all models compared in this project depend on nearest neighbor search, which is well known as lazy learning. Basically for all the data in the database, has no mathematical function that describes the model. In the case of predicting the queries, the inverse or forward model queries the maintained database.

Memory and computational time complexity are some of the drawbacks of this method, as the number of the sample in the database increases linearly with time. After every query, the model must perform a new search and possible regression.

5. Conclusion and Future work

5.1 Conclusion

We proposed non-stationary weighted nearest neighbor regression model, in which in addition to weighting points according to their spatial distance from a query point, points are also weighted according to their 'age' – points added earlier in the robot's experience weigh less than recent points. The model is an improvement to weighted nearest neighbor. However, the transfer model seems to have improved and could be erroneous, some of the transferred points could be corrupt. Learning from scratch NSWNN performs better.

5.2 Future Work

For future work, we plan to use parameterized sensorimotor models that can quickly forget inaccurately transferred knowledge and employ active exploration strategies that will take more significant advantage of the transferred knowledge to guide exploration towards unexplored regions. The concept of knowledge transfer, that is, Local Procrustes Analysis and Procrustes Analysis, was not explored in detail; this could be helpful in the improvement of the experiment. One of the important works could be knowledge transfer from the same baseline and compare different sensorimotor mappings. Lastly, it could be important to learn the inverse model of more complex redundant systems, that is n-links robots.

Acknowledgements

I would like to acknowledge AIMS and funders for supporting this work, as well as my supervisor, Dr Ndivhuwo Makondo from IBM(South Africa). I would also like to thank friends, Alice, Fhumulani, Blessing, AIMS-soccer and AIMS-family who supported me throughout the essay.

References

Baranes, A. and Oudeyer, P. Active learning of inverse models with intrinsically motivated goal exploration in robots. *CoRR*, abs/1301.4862, 2013a. URL http://arxiv.org/abs/1301.4862.

Baranes, A. and Oudeyer, P.-Y. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robot. Auton. Syst.*, 61(1):49–73, Jan. 2013b. ISSN 0921-8890. doi: 10.1016/j.robot.2012.05.008. URL http://dx.doi.org/10.1016/j.robot.2012.05.008.

Bursztyn, L. L. and Flanagan, J. R. Sensorimotor memory of weight asymmetry in object manipulation. *Experimental brain research*, 184(1):127–133, 2008.

Bócsi, B., Csató, L., and Peters, J. Alignment-based transfer learning for robot models. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, Aug 2013. doi: 10.1109/IJCNN.2013.6706721.

Chen, J., Lau, H. Y. K., Xu, W., and Ren, H. Towards transferring skills to flexible surgical robots with programming by demonstration and reinforcement learning. In *2016 Eighth International Conference on Advanced Computational Intelligence (ICACI)*, pages 378–384, Feb 2016. doi: 10.1109/ICACI.2016.7449855.

Corke, P. Matlab toolboxes: robotics and vision for students and teachers. *IEEE Robotics Automation Magazine*, 14(4):16–17, Dec 2007. doi: 10.1109/M-RA.2007.912004.

Delhaisse, B., Esteban, D., Rozo, L., and Caldwell, D. Transfer learning of shared latent spaces between robots with similar kinematic structure. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4142–4149. IEEE, 2017.

Fan, G.-f., Guo, Y.-H., Zheng, J.-M., and Hong, W.-C. Application of the weighted k-nearest neighbor algorithm for short-term load forecasting. *Energies*, 12:916, 03 2019. doi: 10.3390/en12050916.

Flanagan, J. R., Bowman, M. C., and Johansson, R. S. Control strategies in object manipulation tasks. *Current opinion in neurobiology*, 16(6):650–659, 2006.

Hasan, A. T., Hamouda, A., Ismail, N., and Al-Assadi, H. An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 d.o.f serial robot manipulator. *Advances in Engineering Software*, 37(7):432 – 438, 2006. ISSN 0965-9978. doi: https://doi.org/10.1016/j.advengsoft.2005.09.010. URL http://www.sciencedirect.com/science/article/pii/S0965997805001493.

Kuniyoshi, Y., Yorozu, Y., Ohmura, Y., Terada, K., Otani, T., Nagakubo, A., and Yamamoto, T. From humanoid embodiment to theory of mind. In *Embodied artificial intelligence*, pages 202–218. Springer, 2004.

Li, T., Li, Q., and Payendeh, S. NN-based solution of forward kinematics of 3DOF parallel spherical manipulator. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2005. ISBN 0780389123. doi: 10.1109/IROS.2005.1545083.

Makondo, N. and Rosman, B. Towards improving incremental learning of manipulator kinematics with inter-robot knowledge transfer. In *2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)*, pages 68–73, Jan 2019. doi: 10.1109/RoboMech.2019.8704838.

Makondo, N., Rosman, B., and Hasegawa, O. Knowledge transfer for learning robot models via local procrustes analysis. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 1075–1082. IEEE, 2015.

Makondo, N., Rosman, B., and Hasegawa, O. Accelerating model learning with inter-robot knowledge transfer. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2417–2424. IEEE, 2018.

Rocha, C., Tonetto, C., and Dias, A. A comparison between the denavit–hartenberg and the screw-based methods used in kinematic modeling of robot manipulators. *Robotics and Computer-Integrated Manufacturing*, 27(4):723 – 728, 2011. ISSN 0736-5845. doi: https://doi.org/10.1016/j.rcim.2010.12.009. URL http://www.sciencedirect.com/science/article/pii/S073658451100010X. Conference papers of Flexible Automation and Intelligent Manufacturing.

Rolf, M., Steil, J. J., and Gienger, M. Online goal babbling for rapid bootstrapping of inverse models in high dimensions. *2011 IEEE International Conference on Development and Learning (ICDL)*, 2:1–8, 2011.

Schillaci, G. *Sensorimotor learning and simulation of experience as a basis for the development of cognition in robotics*. PhD thesis, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, 2014.

Schmerling, M., Schillaci, G., and Hafner, V. V. Goal-directed learning of hand-eye coordination in a humanoid robot. In *2015 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 168–175, Aug 2015. doi: 10.1109/DEVLRN.2015.7346136.

Takahashi, K., Ogata, T., Tjandra, H., Murata, S., Arie, H., and Sugano, S. Tool-body assimilation model based on body babbling and a neuro-dynamical system for motion generation. In Wermter, S., Weber, C., Duch, W., Honkela, T., Koprinkova-Hristova, P., Magg, S., Palm, G., and Villa, A. E. P., editors, *Artificial Neural Networks and Machine Learning – ICANN 2014*, pages 363–370, Cham, 2014. Springer International Publishing. ISBN 978-3-319-11179-7.

Wieser, E. and Cheng, G. Progressive learning of sensory-motor maps through spatiotemporal predictors. In *2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 43–48. IEEE, 2016.