INVESTIGATION OF ROBOTIC ARM TRACKING OF INFANT REACHING ACTIONS

By

Pamodya N. Peiris

A capstone project submitted for Graduation with University Honors

May 06, 2022

University Honors
University of California Riverside

APPROVED

Dr. Konstantinos Karydis
Department of Electrical and Computer Engineering

Dr. Elena Kokkoni
Department of Bioengineering

Dr. Richard Cardullo, Howard H Hays Jr. Chair
University Honors

ABSTRACT

In this emerging research project, we investigate the performance of robotic arm tracking of infant reaching trajectories. Infant reaching trajectories have been developed by the overall project's team past work, based on infants of age between 5-12 months performing reaching tasks while in a sitting posture. In this work we consider a subset of the original dataset that meets specific inclusion criteria based on constraints on total reaching path length, camera angle, and ability to extract trustworthy metric distance information from the image plane via calibration. We use two 5-DoF (Degree of Freedom) robotic arms of different size and evaluate their efficacy in tracking infant reaching trajectories within a high-fidelity robotics simulation environment. We find that the ability of the robotic arms to track infant reaching trajectories is determined by 1) the start position of the reaching action, 2) the rate of sending position commands to the robot affecting tracking speed, 3) the maximum path length of individual infant reaching trajectories, and 4) the overall range of motion for all trajectories included in the dataset. Qualitative and quantitative analyses confirm the preliminary feasibility to track infant reaching motion using static robotic arms and reveal key position (related to robot workspace limits) and velocity (related to command rate of the robot) constraints that affect quality and accuracy of tracking.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude towards Dr. Konstantinos Karydis and Dr. Elena Kokkoni for their constant support and guidance during the last two years. I joined the project without any prior research experience. In the past two years, I have learned about robotics research, starting with the most basic fundamental tools and moving toward cutting-edge research directions such as assistive robotics. This capstone project is closely related to what I want to study in the future as a Ph.D. student. Through my mentors' advice and the project, I learned many tools needed in the robotics field, which overall broadened my knowledge. I am incredibly grateful for the experience that I received while writing the capstone, as I realized many skills that I need to sharpen for the future. I hope to make this a stepping stone in achieving my future goals in robotics.

I am deeply grateful to the University Honors Program for giving me to the opportunity to work on this capstone project and for providing guidance and inspiration toward graduate school. I am also thankful for the UC LEADS program, where I got the opportunity to work under my faculty mentors.

I would like to thank Ipsita Sahin for providing massive support to complete this capstone successfully. I would also like to thank Vikarn Bhakri, Arjun Modi, Julya Mestas, and Dannya Enriquez Barrundia, who have been part of this research during these years. I completed this capstone thanks to their hard work in the early stages of research. I would like to extend my gratitude to Georgia Kouvoutsakis, Amel Dechemi, Dimitris Chatziparaschis, and Merrick Campbell, who supports me both personally and professionally. Many thanks to all the Autonomous Robots and Control Systems (ARCS) Lab members and Pediatric Rehabilitation Technology (PRT) Lab for always being encouraging and supportive.

Lastly, I am deeply grateful to my parents, Sanjeewa and Subhani, sister Charithma, grandmother Soma, and friends for always trusting and keeping faith in me. I would not be here without their unconditional support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human-robot interaction (HRI) broadly concerns the communication and physical interaction between human(s) and robot(s) in ways that are beneficial to the human(s) and efficient for the robot(s) [2–6]. Enabling effective social and physical HRI is crucial for the successful application of robots across various contexts, such as assisted living [7–13] and rehabilitation over the lifespan [14–23]. The overall project that this work is part of studies the use of robotic manipulators in pediatric rehabilitation. A key component is to investigate the extent that robotic arm motion can imitate developing infants' arm motion trajectories, and eventually create new mechanisms to enable high-fidelity robotic imitation/trajectory tracking of infant reaching motions. The specific goal of this work is to investigate the preliminary feasibility to track/imitate arm trajectories of infants performing reaching using static robotic arms. This is a first step, trying to build up an understanding on how certain constraints of the robotic hardware (most importantly workspace and velocity constraints) can affect the quality and accuracy of tracking infant reaching trajectories. Once this information is available, it can enable similar directions for other static manipulators, wearable robotics, or mobile humanoids regarding physical assistance or social interaction (e.g., hand an object to an infant during play).

Previous work with humanoids has shown imitation of adult arm movements while performing general tasks such as reaching and grasping for objects [24–26], handshaking [27], passing objects to other(s) [26], as well as across different contexts, such as when doing kitchen work [28] and sport activities (e.g., boxing [29], etc.). However, all of the aforementioned works are based on adult arm movement data. This work, in contrast, focuses on the use of data from our target population which is very young children, and more specifically, infants (less than 1 year of age).

It is important to consider movements of infants for our application as humans during early development present greater variability on their control of movement as compared to adults [30]. This leads to a rich set of different reaching trajectories. In this work, we use a dataset of infant reaching actions that have been collected through open-source videos. We test the performance of two robotic arms with different motion spans to imitate these trajectories in a simulation environment. The main goals of this thesis are to 1) investigate whether these robotic arms can follow the trace of infant reaching action trajectories (i.e. exclude the temporal component of the reaching action), 2) investigate if, in addition, the robotic arms can follow the trajectories at the same rate of the data collection, 3) determine how smooth robotic arm trajectories can be, and 4) establish what is the effect of the configuration space and velocity constraints of the robotic arm on tracking infant reaching trajectories.

Chapter 2

Methods

2.1 Dataset for Infant Reaching Actions

This work considers a subset of a recent dataset of infant reaching in unconstrained environments [1]. The original dataset includes a total of 437 reaching actions from 32 neurotypical subjects and 7 neurodivergent subjects collected through 35 videos open-sourced at the YouTube video sharing platform. The age of the infants varies from 3 - 12 months. Reaching actions are considered only when the infants are in a 'sitting' position. The reaching actions are further classified based on the infant's position as viewed from the camera as 'front,' 'side,' and 'back.' The dataset includes reaching actions that are both calibrated (coordinates of the position of the arm are in cm) and/or uncalibrated (coordinates are in pixels).

The subset from the original data [1] was created by downselecting reaches that met three main inclusion factors:

1. The trajectory of the reaching action is calibrated and can be expressed in a metric (in cm) coordinate system.

2. The camera view and the infant are parallel to each other, i.e. the camera captures the side view of the infant.

3. The reaching trajectory is mostly planar, that is, it has minimal variation in depth with respect to the camera image plane.

After this downselection, the dataset used in this work for the investigation of how a robotic arm can track infant reaching trajectories includes 60 reaching actions from infants of age of 5 - 12 months. The subjects include 16 neurotypical subjects and 2 neurodivergent subjects with Congenital Anomaly and Autism Level 3. Table 2.1 contains information on the dataset used herein, whereby videos with neurotypical and neurodivergent infants are indicated as T-XX (e.g., T01) and D-XX (e.g., D02), respectively.

Subject ID	Age (months)	# reaches
T01	6-8	1
T02	8-10	5
T03	11-12 (estimated)	6
T04	6-12	1
T08	6	2
T09	6	1
T11	8	6
T19	7	2
T21	9	2
T23	9	1
T24	6	1
T25	6	8
T27	8	10
T29	7	2
T33	5-7	6
T34	9-10 (estimated)	2
D02	10	2
D07	unknown	2

Table 2.1: The dataset used in this work, which is a subset of the original infant reaching dataset [1].

A reaching action is defined by determining two specific instances at the video, the onset and offset frames. The onset corresponds to the very first frame the infant moves their hand toward the object whereas the offset is the very first frame that the infant touches the object [1]. Then, manual annotation frame-by-frame[1] of the midpoint of the infant's wrist in the image plane (Figure 2.1) between the onset and offset frames is used to create an infant reaching trajectory. Snapshots of a sample reaching action is illustrated in Figure 2.2, with the tracked path (in yellow) indicated as well.

[1] We use the Kinovea video editing software for the annotations.

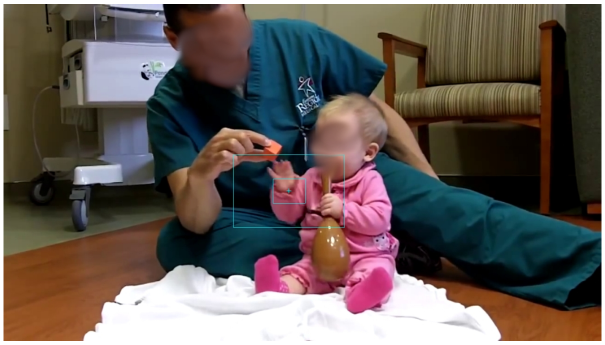
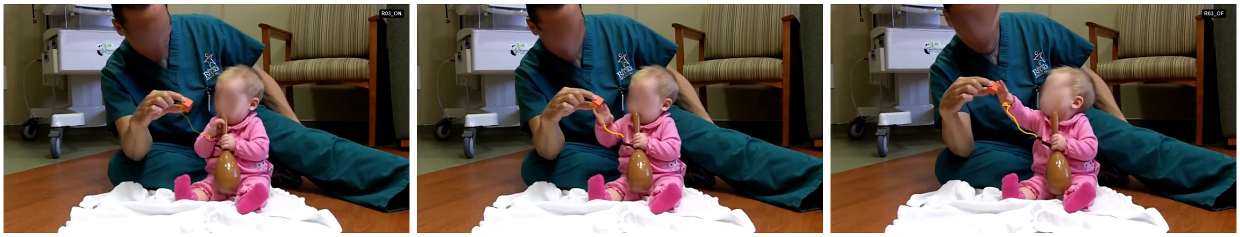


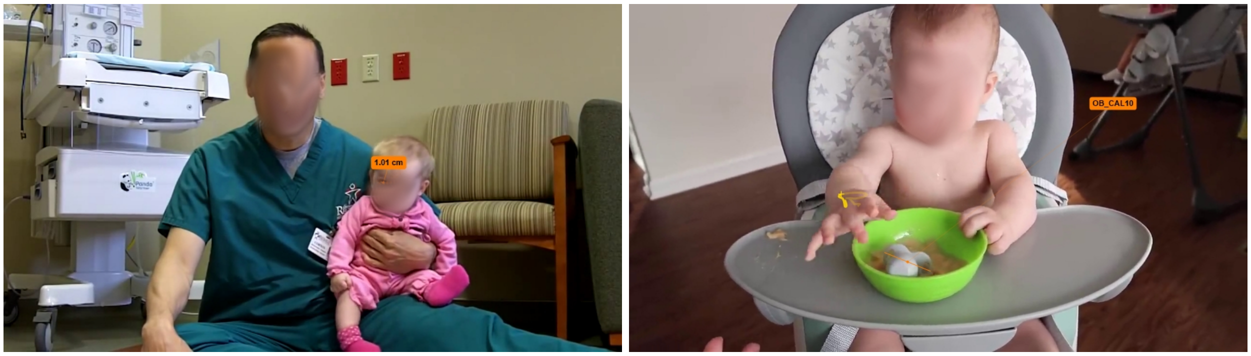
Figure 2.1: Example of the point (midpoint of the wrist) tracked in frame-by-frame manual annotation to obtain an infant reaching trajectory.



(a) (b) (c)

Figure 2.2: Snapshots of a sample reaching action at its (a) onset, (b) some interim point, and (c) offset. The tracked path is highlighted in yellow. (Figure best viewed in color.)

The output of this tracking procedure, by default, is a trajectory expressed in units of pixels. However, to enable tracking of this trajectory with a robotic arm, we need to know the corresponding coordinates in the metric system (in cm). The three inclusion criteria mentioned above help ensure that we can attain metric trajectories for all reaching actions considered in this work. Attaining the correspondence between cm and pixels relies on a calibration phase. Calibrations is performed in one of two ways. The primary way is by using the infant's cornea in all videos the cornea is clearly visible. If this cannot be viewed clearly, then we calibrate using some known object in the scene. To remedy for lack of depth estimation in the 2D image plane, which can lead to significant distortion in the pixel-to-cm correspondences, we seek to select objects that are located as close to the infant as possible. The calibration process is done rigorously for every reaching action individually. Representative examples from both calibration cases are shown in Figure 2.3.



(a) (b)

Figure 2.3: Instances of the calibration process to acquire correspondences between pixels in the image plane and their respective metric (in cm) representation using (a) the infant's cornea, or (b) a known object located close to the infant in cases that the cornea is not clearly visible.

2.2 Software Infrastructure and Setup

We employ two different robotic arms, and evaluate their efficacy in tracking the infant reaching trajectories in a faithful and physically-relevant simulation environment. We use the ReactorX 150 [rx150] and ReactorX 200 [rx200] robotic arms from the X-series Trossen Robotics. Robot motion is programmed in Python 2 within a Linux (Ubuntu 16.04) environment using the Robot Operating System (ROS; Kinetic version). Simulations are run in the RViz environment provided by ROS.

ROS [31] is an open-source middleware that connects hardware with sensors and actuators with software that simulates physical components in the 3D space using environments such as RViz or physics simulation engines like Gazebo. ROS has nodes, topics, services, and parameter servers. ROS nodes are set up to establish communication between ROS topics. The nodes can publish data to the robot or subscribe data from the robot using topics. ROS services are used to advertise these topics. The way the different components of the rx150 robot communicate via ROS shown in Figure 2.4. The links of the robot arm are connected to each other all the way to root node of the world frame, and this allows to easily publish and subscribe data from the nodes using topics.

Both rx150 and rx200 robots have the same structure and mostly same link lengths except for the lengths of links 3 and 4, as seen in Figure 2.5. For the rx150 arm these are both 150 mm long, while for the rx200 they are 200 mm long. These have rostopics /rx150/joint_states and /tf

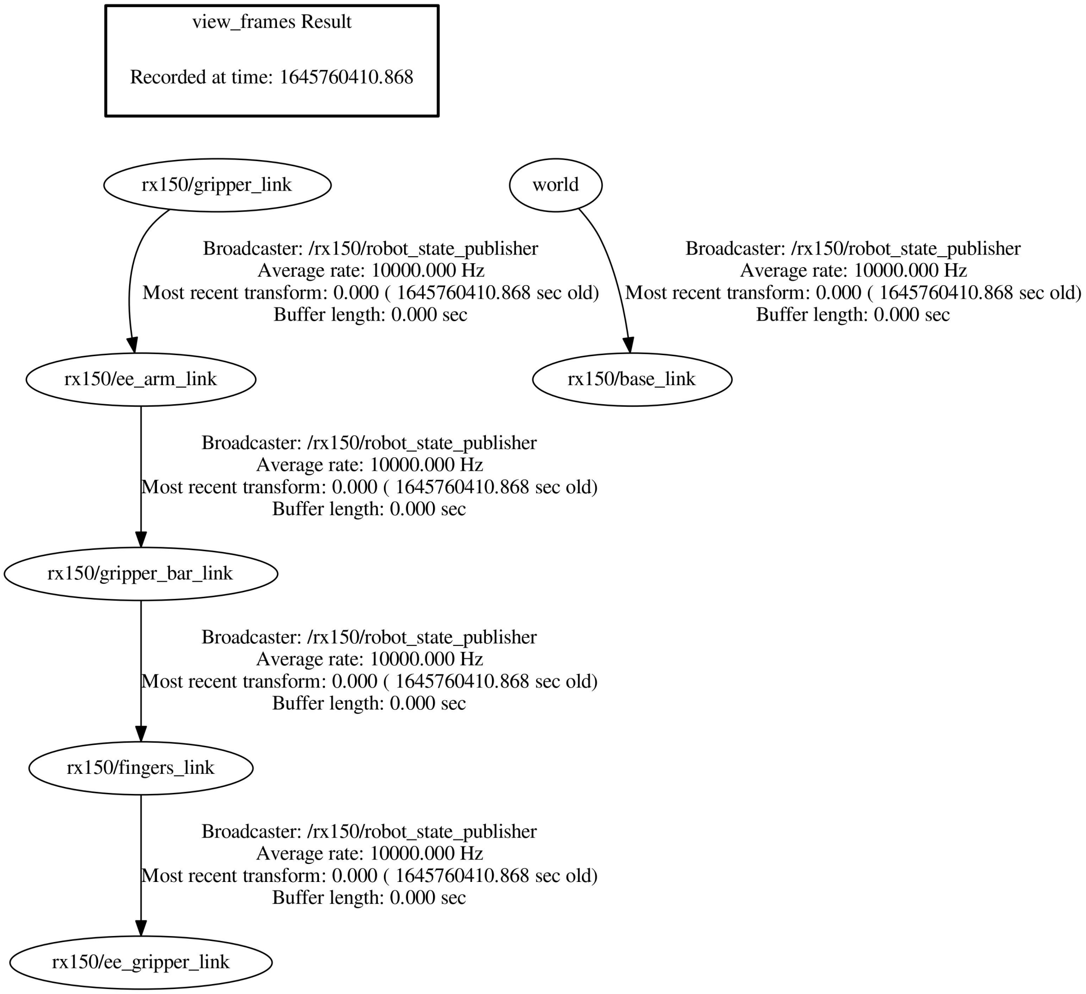

view_frames Result
Recorded at time: 1645760410.868
rx150/gripper_link
world
Broadcaster: /rx150/robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1645760410.868 sec old)
Buffer length: 0.000 sec
Broadcaster: /rx150/robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1645760410.868 sec old)
Buffer length: 0.000 sec
rx150/ee_arm_link
rx150/base_link
Broadcaster: /rx150/robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1645760410.868 sec old)
Buffer length: 0.000 sec
rx150/gripper_bar_link
Broadcaster: /rx150/robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1645760410.868 sec old)
Buffer length: 0.000 sec
rx150/fingers_link
Broadcaster: /rx150/robot_state_publisher
Average rate: 10000.000 Hz
Most recent transform: 0.000 (1645760410.868 sec old)
Buffer length: 0.000 sec
rx150/ee_gripper_link

Figure 2.4: ROS tf topic tree for the rx150 robot arm.

to publish angles to move the robot to the given position and get the end-effector position from the robot respectively. ROS topic /rx150/joint_states is used to publish the joint angles calculated to move the end-effector to the position according to the trajectory data. ROS topic /tf is used to subscribe to the 'world' frame and the 'wrist_link' to calculate the robot's position using ROS nodes. This allows to compare the trajectory data with the real-time data of the end-effector.

	rx150 (mm)	rx200 (mm)
reaching action	450	550
span	900	1100
wrist-finger	174.15	174.15
limited reach	275.85	375.85
limited span	551.7	751.7

Table 2.2: Key specification of the rx150 and rx200 robotic arms considered in this work.

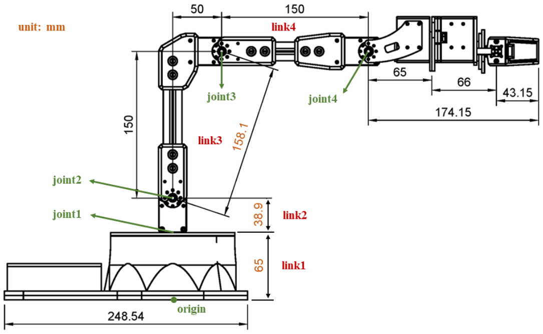


Figure 2.5: Schematic of the rx150 robotic arm.

2.3 Robot Arm Kinematics

2.3.1 Forward Kinematics

Forward kinematics calculates the robot's end-effector position and orientation when the joint angles are given. This can be calculated using Denavit-Hartenberg parameters [32].

i	α_{i-1}	a_{i-1}	d_i	ϕ_i
1	0	0	link1z	θ_1
2	-90	0	link2z	θ_2
3	180	$\sqrt{link3x^2 + link3z^2}$	0	θ_3
4	0	link4x	link4x	0

Table 2.3: Denavit-Hartenberg parameters for the forward kinematics.

The transformation from frame {i-1} to {i} can be calculated using (2.1). The expressions for each rotation (Rot) and Translation (Trans) are calculated using the matrices below.

$$T_{i-1,i} = Rot(\hat{x}, \alpha_{i-1}) Trans(\hat{x}, a_{i-1}) Trans(\hat{z}, d_i) Rot(\hat{z}, \phi_i) \tag{2.1}$$

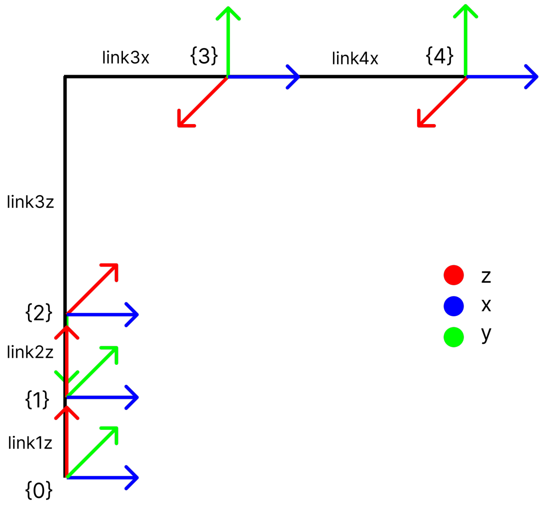


Figure 2.6: D-H parameters

$$Rot(\hat{x}, \alpha_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_{i-1} & -\sin\alpha_{i-1} & 0 \\ 0 & \sin\alpha_{i-1} & \cos\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

$$Trans(\hat{x}, a_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

$$Trans(\hat{z}, d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

$$Rot(\hat{z}, \phi_i) = \begin{bmatrix} \cos\phi_i & -\sin\phi_i & 0 & 0 \\ \sin\phi_i & \cos\phi_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

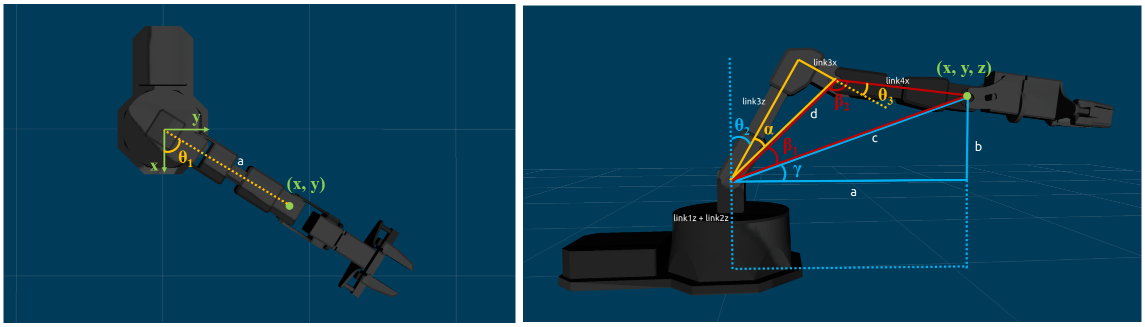
2.3.2 Inverse Kinematics

To move the robotic arm according to the tracking data, the joint angles are needed. The joint angles are obtained through the geometric method [32] of inverse kinematic by using the geometry shown in Figure 2.7. Length ‘a’ shown in Figure 2.7a in the xy plane corresponds to the length ‘a’ shown in Figure 2.7b in xz plane.

The link lengths can be found in Figure 2.5 and the end-effector is placed on the wrist of the arm which is labeled ‘joint4’ in the figure. The joint angles are solved using Pythagorean theorem as per (2.6) and the law of cosines (2.7). These equations were written in reference to the blue triangle in Figure 2.7b. Joint1, Joint2, and Joint3 in Figure 2.5 refer to θ_1, θ_2, and θ_3 respectively.

$$\begin{array}{lll}
a = \sqrt{x^2 + y^2} & \alpha = \arctan(\frac{link3x}{link3z}) & \theta_1 = \arctan(\frac{y}{x}) \\
b = z - link1z - link2z & \beta_1 = \arccos(\frac{c^2+d^2-link4x^2}{2cd}) & \theta_2 = \frac{\pi}{2} - \alpha - \beta_1 - \gamma \\
c = \sqrt{a^2 + b^2} & \beta_2 = \arccos(\frac{d^2+link4x^2-c^2}{2dlink4x}) & \theta_3 = \beta_2 - \frac{\pi}{2} - \alpha \\
d = \sqrt{link3x^2 + link3z^2} & \gamma = \arctan(\frac{b}{a}) &
\end{array}$$

$$c = \sqrt{a^2 + b^2} \qquad (2.6) \qquad\qquad b = \sqrt{a^2 + c^2 - 2ac \cdot (\gamma)} \qquad (2.7)$$



(a)

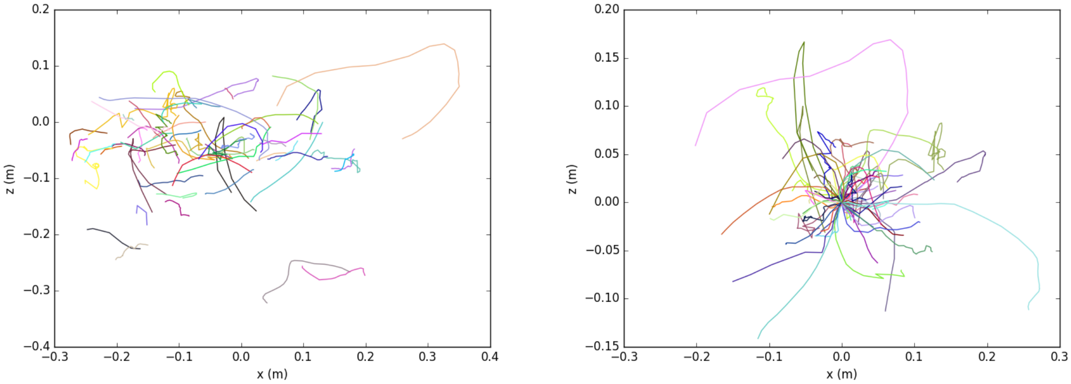
(b)

Figure 2.7: Inverse Kinematics for the ReactorX robotic arm. To solve this, employ trigonometry by identifying appropriate triangles containing the robot’s joint angles in both (a) top-down and (b) side views of the robot, respectively.

2.4 Algorithmic implementation

2.4.1 Change of Coordinate Systems

As the videos are 2D, the data collected are in the xz plane. Therefore, y is always equal to zero. As seen in Figure 2.8a, raw data take values in the range of [-300, 400] mm for the x-coordinate and in the range of [-400, 200] mm for the z-coordinate. The rx150's wrist has a span of 550 mm, but the finger portion of the arm must also be taken into consideration to avoid any self collisions. To reduce the range, the first point of every reaching action is converted into (0, 0, 0) as seen in Figure 2.8b. This way we reduce the x range to [-300, 300] mm and the z range to [-150, 200] mm. The rest of the points are shifted accordingly when the initial point is set at (0, 0, 0). Although this change of coordinate systems allows the arm to perform more reaching actions, having the initial position at (0, 0, 0) can make the end-effector collide with the base of the robot as seen in Figure 2.9, and hence this needs to be taken into consideration too.



(a)

(b)

Figure 2.8: (a) Original data to be tracked by the robotic arms and (b) data after change of the coordinate system.

A range of coordinates from 0 - 200 mm has been used on all (x, y, z) for the initial position. During these changes, the best initial position that does not go out of bound or collide with the robot is (0.00, 0.15, 0.20) m. All the trajectories change their coordinate system by making this point as the initial. This point is taken to be the best position through trial and error. The value of the y-axis is always equal to 0.15 m as the trajectories are 2D and only give x-z coordinates.

2.4.2 Simulation of the Trajectories

The complete algorithm to perform simulation of the robotic arms tracking infant reaching trajectories is shown in Figure 2.10. First, the initial position taken via the change of coordinate

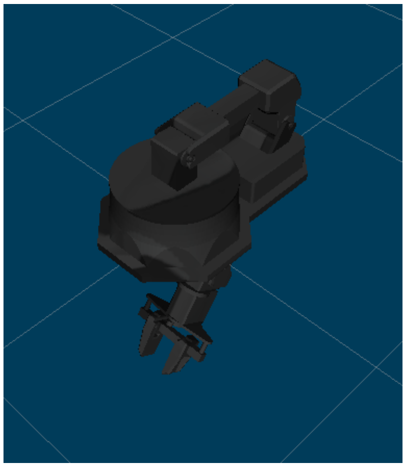


Figure 2.9: Initial position (0, 0, 0) leading to a self collision.

systems is used to shift every point in the reaching action dataset. The shifted coordinates are used in the calculation of Inverse Kinematics to compute the joint angles for joints 1, 2 and 3 (Figure 2.5) that are needed to make the robotic arm move to track the infant trajectories. These joint angles are published to the robot arm using ROS topic /rx150/joint_state for the rx150 robot (similarly for the rx200). After the robotic arm moves to the published point, the subscriber /tf is used to get the end-effector position of the robot. Then, the process repeats for all points on an infant reaching trajectory, and for all available trajectories. Finally, the trajectory data and the end-effector data are plotted overlaid to each other to qualitatively evaluate the algorithm performance. These trajectories are plotted using both rx150 and rx200 robots with different time rates to determine performance. Quantitative analysis is then used to explicitly determine the quality of tracking.

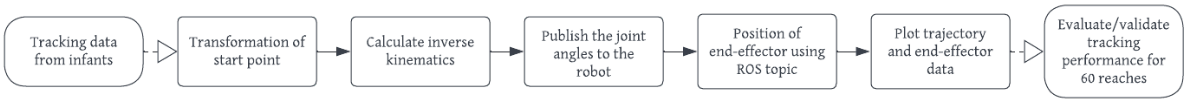

Tracking data from infants
Transformation of start point
Calculate inverse kinematics
Publish the joint angles to the robot
Position of end-effector using ROS topic
Plot trajectory and end-effector data
Evaluate/validate tracking performance for 60 reaches

Figure 2.10: Flowchart of the developed algorithm used for simulations.

Trajectories are simulated in the RViz environment for different time rates. The time rate essentially determines how fast the robotic arm is tasked to follow the infant reaching trajectories. By default, the infant reaching actions determined via the video software Kinovea are at an average frame rate of 30 fps (frames per second). This is because the frames from Kinovea are obtained in a sequence of 0.3 s, 0.3 s, and 0.4 s; that is, every second it gives three frames from the video using this sequence. Note that (2.8) converts Kinovea frame rate to Hz for simulation.

$$(\frac{1}{0.3}+\frac{1}{0.3}+\frac{1}{0.4})/3 = 30.5Hz \tag{2.8}$$

To investigate the effect of the rate on the robots' tracking performance, the robotic arms are simulated at 1 Hz, 5 Hz, 10 Hz, and 30.5 Hz.

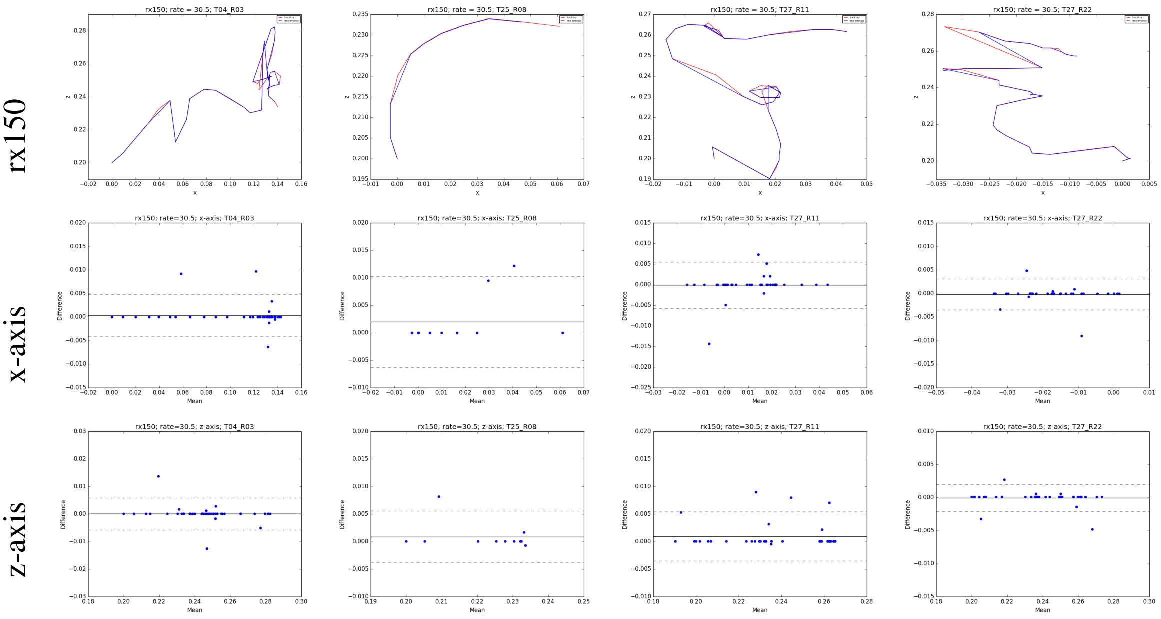
Chapter 3

Results and Discussion

We use Bland-Altman plots to determine how the infant reaching trajectories correlate to the output (end-effector) trajectories of the two robotic arms. The Bland and Altman plots use the mean and standard deviation of the two signals [33]. If the infant reaching data and the end-effector data along the x- and z-axes separately correlate with each other, then the points in the scatter plot will be closer to the horizontal line. Instead, low correlation would give rise to a more scattered plot.

Representative graphical results for the rx150 and rx200 robotic arms tracking infant reaching trajectories at 30.5 Hz are depicted in Figures 3.1 and 3.2, respectively. The top row of panels overlays the two signals, whereas the middle and bottom row panels show the Bland-Altman plots in the x-axis and z-axis, respectively. As done commonly in the literature, we also consider the 95% limits of agreement (dashed horizontal lines) around the mean (solid horizontal line). The narrower the 95% band (± 1.96 standard deviations from the mean) the higher the correlation, thus indicating that the subset of reaches we have selected is sufficiently diverse to test the efficacy of the robotic arms to follow these trajectories. From the graphs it can be observed that in most cases the two signals have high correlation (small bands), suggesting we can effectively use these robotic arms to track infant reaching trajectories. Representative samples of high correlation are shown in columns (a), (c) and (d), whereas a reaching action that leads to lower correlation is included in column (b).

However, it is important to note that a high correlation alone does not always imply good



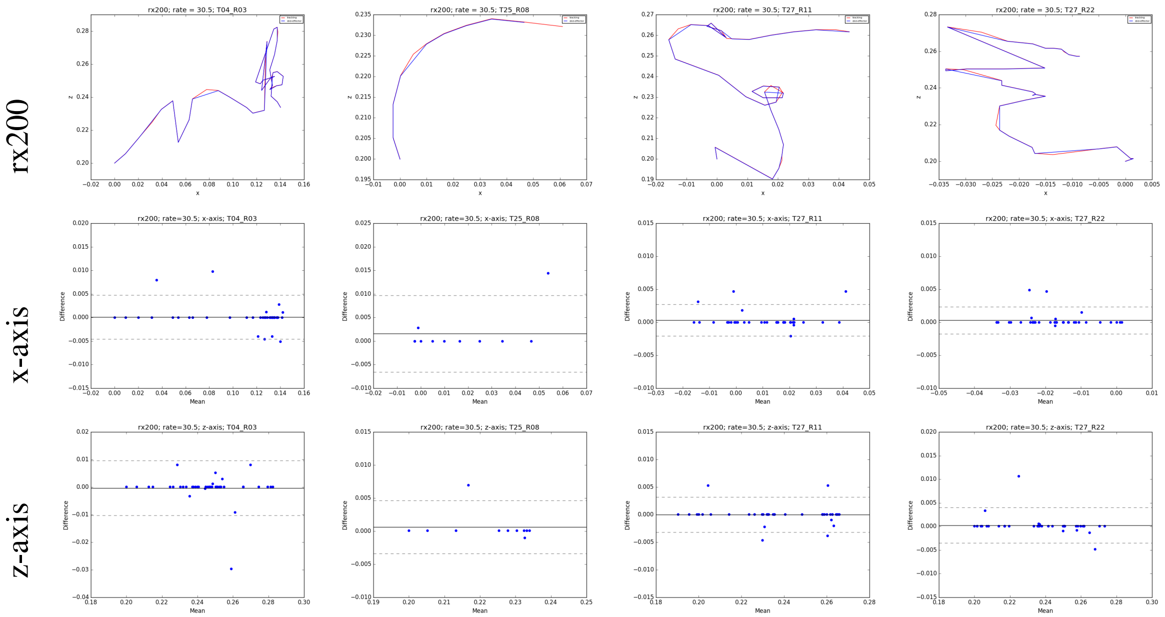
(a) (b) (c) (d)

Figure 3.1: Selected Bland-Altman plots for the rx150 robot tracking infant reaching trajectories at a rate of 30.5 Hz.

agreement between the two signals. For instance, the actual trajectories shown in the graphs in the top panels of columns (a) and (b) differ close to the end of the trajectory, but the case in column (a) has a high correlation whereas the case in column (b) has a lower correlation. To be able to capture such differences, we need to quantify the actual tracking error as well. For that, we use the mean squared error (MSE), as discussed next.

The MSEs for all tracked trajectories at all three rates for the rx150 and rx200 robots are depicted graphically in Figures 3.3 and 3.4, respectively. We observe that tracking errors are overall very low, with some notable exceptions.

First, despite the shift in the coordinate system, the robot may still be tasked to reach a point that is outside its reachable workspace. This is the case mostly for the 'shorter' rx150 robot, happening specifically when tasked to track trajectories T09_R01 and T11_R43 (notice the spikes Figure 3.3). When using the rx200 robot that has longer link lengths, this issues is resolved. As shown in Figure 3.5, the rx150 robot cannot reach to every point of the given trajectory, but in the same trajectories the rx200 can perform well (Figure 3.6). In the rx200 robot case, tracking errors



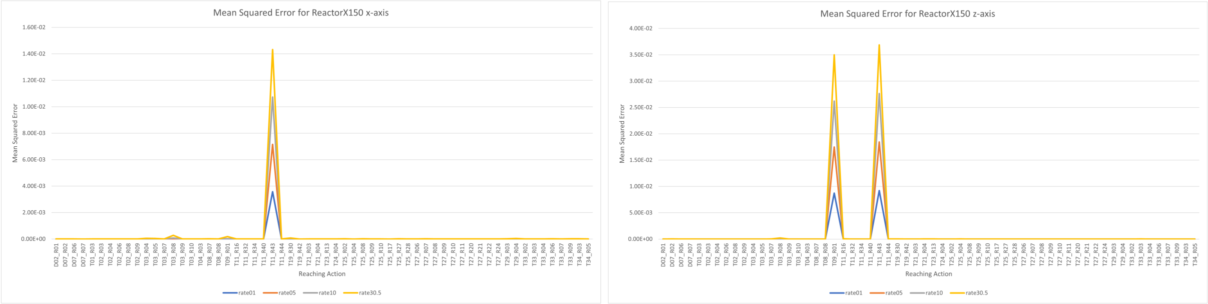
(a) (b) (c) (d)

Figure 3.2: Selected Bland-Altman plots for the rx200 robot tracking infant reaching trajectories at a rate of 30.5 Hz.

in T03_R08 are associated with the delay caused by starting the simulation (Figure 3.4b).

The second source of errors concerns the desired rate to follow the infant reaching trajectories. Although there are no visibly major error spikes for both rx150 and rx200 robots, it is the case that when in higher rates (30.5 Hz), the arm tends to avoid some points when there is a sharp change in the direction of movement. Figure 3.7 depicts some instances where the arm fails to follow the exact trajectory. Rapidly-variable parts of infant reaching trajectories (such as the last part of the one shown in Figure 3.7c), cannot be tracked by the robot arm, unless it is tasked to operate at a lower rate (i.e. follow the infant reaching action at a slower speed than the actual infant motion). We expect that this limitation is mostly related to the specific joint torque limitations imposed by the robots' design. However, in practice, an additional possible constraint would be the rate at which the robot can accurately estimate its position. In simulation we did not face this challenge as position feedback was readily available, but future testing with a physical robot arm will need to take this into consideration as well.

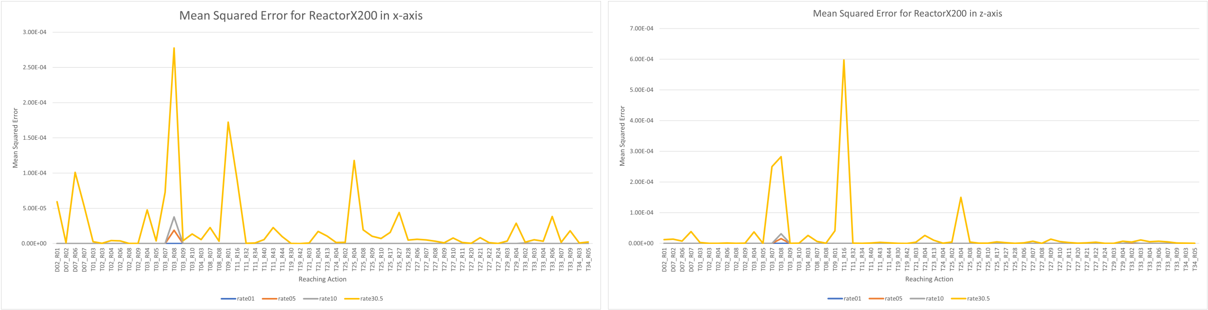
Additionally, the comparison between different rates (10 Hz and 30.5 Hz) is highlighted in



(a) x-axis

(b) z-axis

Figure 3.3: MSE for rx150

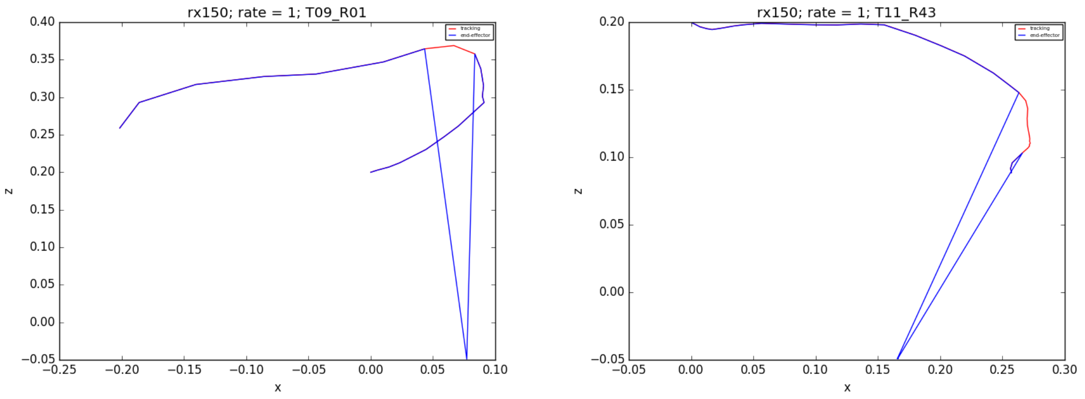


(a) x-axis

(b) z-axis

Figure 3.4: MSE for rx200

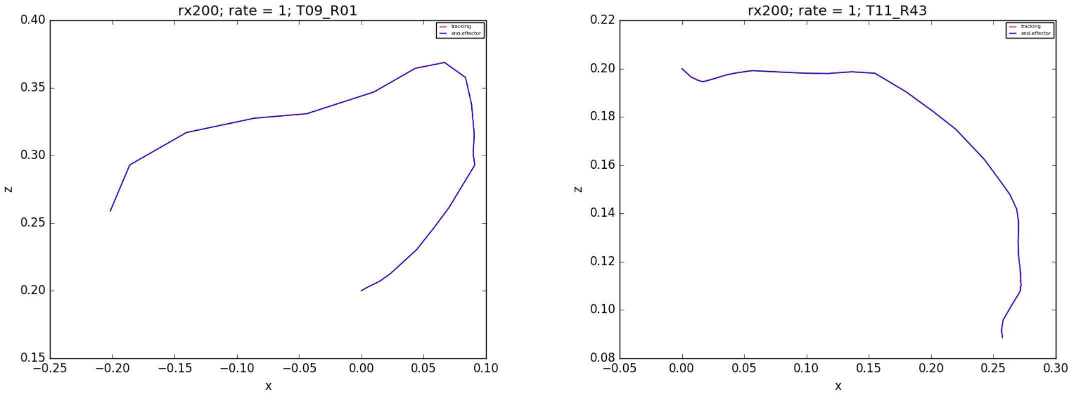
Figures 3.8 and 3.9. It can be readily verified that when the rate is 10 Hz, the end-effector follows the reaching action trajectory without skipping any points. However, when the rate is set to 30.5 Hz, the reaching action does not follow the path accurately. This is mainly because at higher rates the robot is given less time to move from one point to the next, which is, as discussed above, linked to maximum feasible joint torques. It is also noted that a robot with a larger configuration space may require more effort to move hence impacting tracking performance as the rate increases.



(a)

(b)

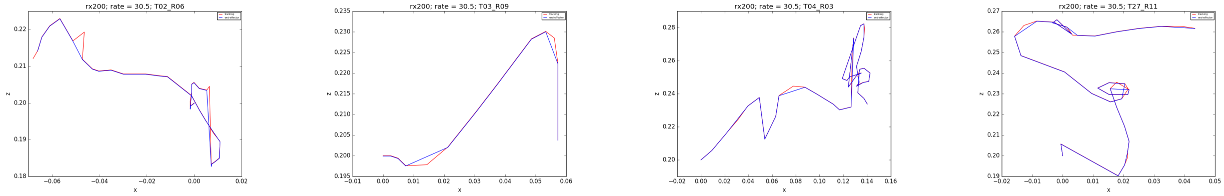
Figure 3.5: Trajectory tracking performance of the rx150 robot in samples (a) T09_R01 and (b) rx150 T11_R43, showing that the trajectory moves outside the reachable workspace of the robot and thus leading to tracking errors.



(a)

(b)

Figure 3.6: Trajectory tracking performance of the rx200 robot in samples (a) T09_R01 and (b) rx150 T11_R43. In contrast to the rx150 robot, there is no workspace limitation here hence tracking accuracy is high.



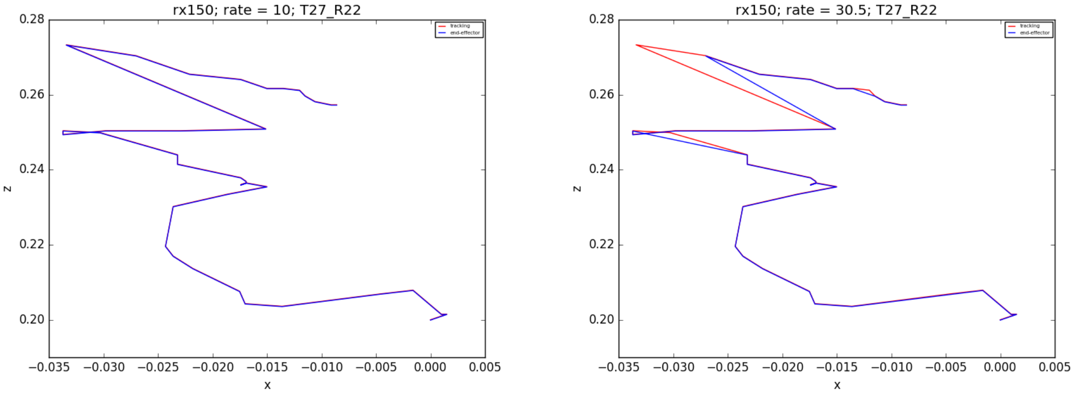
(a) T02_R06

(b) T03_R09

(c) T04_R03

(d) T27_R11

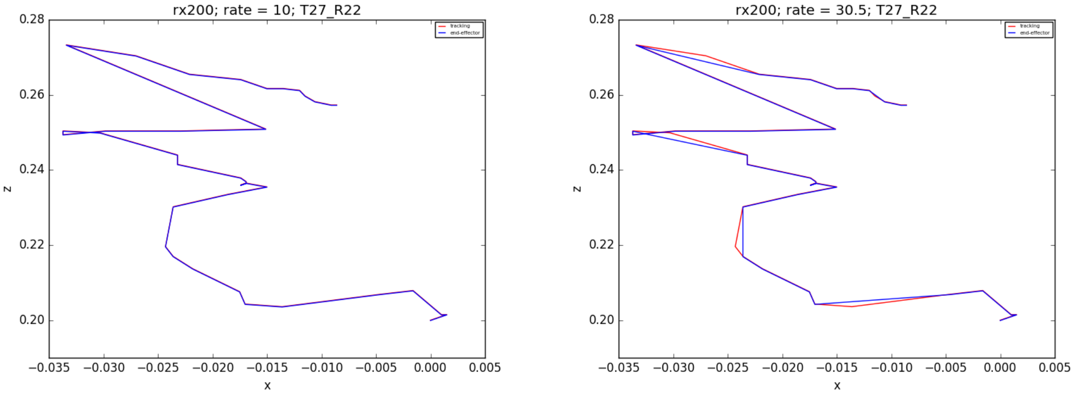
Figure 3.7: Some representative trajectory tracking errors at the higher rate of 30.5 Hz.



(a) rx150 rate 10 Hz

(b) rx150 rate 30.5 Hz

Figure 3.8: Effect of the rate on trajectory tracking for the rx150 robot.



(a) rx200 rate 10 Hz

(b) rx200 rate 30.5 Hz

Figure 3.9: Effect of the rate on trajectory tracking for the rx200 robot.

Chapter 4

Conclusion

This work investigated the preliminary feasibility of tracking infant reaching trajectories using small, servo-based robotic arms, in high-fidelity simulation. Infant reaching trajectories were obtained by online videos meeting specific inclusion criteria, mostly related to viewing angle and ability to faithfully transform pixel coordinates from the image plane to metric coordinates that needed for tracking with the robot arm. Frame-by-frame manual annotations helped us create the infant reaching trajectories. The latter were then given as input to two robotic arms to follow. Qualitative and quantitative analyses confirmed the preliminary feasibility to track infant reaching motion using static robotic arms.

Testing and evaluation revealed certain limitations to be taken into consideration to further improve tracking performance. We noticed that infant reaching may at times exceed the reachable workspace of the robot. While this issue cannot always be addressed without restricting the area an infant is presented an object to reach for or changing the robot arm, the logic employed to the robot regarding what to do in such situations can be controlled. One way would be for the robot to hold position until the infant arm's position enters the reachable workspace of the robot and then resume tracking. Another way would be for the robot to try to predict the missing information and pick a direction to move. A second interesting observation was the limitations on how fast a infant reaching trajectory could be in fact followed by the robot. The slower, the more accurate (as anticipated) but extending the operational speed could be made possible by employing a more advanced trajectory tracking controller.

The aforementioned two key observations also guide future work enabled by this present work. Specifically, we are interested in tracking infant reaching trajectories at real-time using a soft wearable robotic device [34]. The intent is to, first, ensure the robot can capture the temporal component of infant reaching, and, second, use that ability to close a real-time control loop when the device is worn so to offer assistive feedback. Based on the findings of this work, we now have a better understanding about the impact of tracking speed, which will be a critical component for future assistive control design.

Bibliography

[1] A. Dechemi, V. Bhakri, I. Sahin, A. Modi, J. Mestas, P. Peiris, D. E. Barrundia, E. Kokkoni, and K. Karydis, “Babynet: A lightweight network for infant reaching action recognition in unconstrained environments to support future pediatric rehabilitation applications,” in *30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, 2021, pp. 461–467.

[2] D. Feil-Seifer and M. J. Mataric, “Human-robot interaction,” *Encyclopedia of complexity and systems science*, vol. 80, 2009.

[3] P. A. Hancock, D. R. Billings, K. E. Schaefer, J. Y. Chen, E. J. De Visser, and R. Parasuraman, “A meta-analysis of factors affecting trust in human-robot interaction,” *Human factors*, vol. 53, no. 5, pp. 517–527, 2011.

[4] T. B. Sheridan, “Human–robot interaction: status and challenges,” *Human factors*, vol. 58, no. 4, pp. 525–532, 2016.

[5] M. A. Goodrich and A. C. Schultz, *Human-robot interaction: a survey*. Now Publishers Inc, 2008.

[6] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, “An atlas of physical human–robot interaction,” *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, 2008.

[7] R. H. Wang, A. Sudhama, M. Begum, R. Huq, and A. Mihailidis, “Robots to assist daily activities: views of older adults with alzheimer’s disease and their caregivers,” *International psychogeriatrics*, vol. 29, no. 1, pp. 67–79, 2017.

[8] P. Dario, E. Guglielmelli, C. Laschi, and G. Teti, "Movaid: a personal robot in everyday life of disabled and elderly people," *Technology and Disability*, vol. 10, no. 2, pp. 77–93, 1999.

[9] H. Robinson, B. MacDonald, and E. Broadbent, "The role of healthcare robots for older people at home: A review," *International Journal of Social Robotics*, vol. 6, no. 4, pp. 575–591, 2014.

[10] D. Bassily, C. Georgoulas, J. Guettler, T. Linner, and T. Bock, "Intuitive and adaptive robotic arm manipulation using the leap motion controller," in *ISR/Robotik 2014; 41st International Symposium on Robotics*. VDE, 2014, pp. 1–7.

[11] T. L. Mitzner, T. L. Chen, C. C. Kemp, and W. A. Rogers, "Identifying the potential for robotics to assist older adults in different living environments," *International journal of social robotics*, vol. 6, no. 2, pp. 213–227, 2014.

[12] G. Wilson, C. Pereyda, N. Raghunath, G. de la Cruz, S. Goel, S. Nesaei, B. Minor, M. Schmitter-Edgecombe, M. E. Taylor, and D. J. Cook, "Robot-enabled support of daily activities in smart home environments," *Cognitive Systems Research*, vol. 54, pp. 258–272, 2019.

[13] J. Mehrholz, A. Hädrich, T. Platz, J. Kugler, and M. Pohl, "Electromechanical and robot-assisted arm training for improving generic activities of daily living, arm function, and arm muscle strength after stroke," *Cochrane database of systematic reviews*, no. 6, 2012.

[14] W. H. Chang and Y.-H. Kim, "Robot-assisted therapy in stroke rehabilitation," *Journal of stroke*, vol. 15, no. 3, p. 174, 2013.

[15] H. I. Krebs, L. Dipietro, S. Levy-Tzedek, S. E. Fasoli, A. Rykman-Berland, J. Zipse, J. A. Fawcett, J. Stein, H. Poizner, A. C. Lo *et al.*, "A paradigm shift for rehabilitation robotics," *IEEE engineering in medicine and biology magazine*, vol. 27, no. 4, pp. 61–70, 2008.

[16] J. L. Pons, “Rehabilitation exoskeletal robotics,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 29, no. 3, pp. 57–63, 2010.

[17] H. S. Lo and S. Q. Xie, “Exoskeleton robots for upper-limb rehabilitation: State of the art and future prospects,” *Medical engineering & physics*, vol. 34, no. 3, pp. 261–268, 2012.

[18] M. A. Gull, S. Bai, and T. Bak, “A review on design of upper limb exoskeletons,” *Robotics*, vol. 9, no. 1, p. 16, 2020.

[19] E. Kokkoni, E. Mavroudi, A. Zehfroosh, J. C. Galloway, R. Vidal, J. Heinz, and H. G. Tanner, “Gearing smart environments for pediatric motor rehabilitation,” *Journal of neuroengineering and rehabilitation*, vol. 17, no. 1, pp. 1–15, 2020.

[20] I. Babik, E. Kokkoni, A. B. Cunha, J. C. Galloway, T. Rahman, and M. A. Lobo, “Feasibility and effectiveness of a novel exoskeleton for an infant with arm movement impairments,” *Pediatric physical therapy: the official publication of the Section on Pediatrics of the American Physical Therapy Association*, vol. 28, no. 3, p. 338, 2016.

[21] G. R. Kouvoutsakis, K. Baxevani, H. G. Tanner, and E. Kokkoni, “Feasibility of using the robot sphero to promote perceptual-motor exploration in infants,” in *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, 2022, pp. 850–854.

[22] J. R. Vora, A. Helmi, C. Zhan, E. Olivares, T. Vu, M. Wilkey, S. Noregaard, N. T. Fitter, and S. W. Logan, “Influence of a socially assistive robot on physical activity, social play behavior, and toy-use behaviors of children in a free play environment: A within-subjects study,” *Frontiers in Robotics and AI*, vol. 8, 2021.

[23] J. Fasola and M. J. Matarić, “A socially assistive robot exercise coach for the elderly,” *Journal of Human-Robot Interaction*, vol. 2, no. 2, pp. 3–32, 2013.

[24] J. Zhao, B. Xie, and C. Song, “Generating human-like movements for robotic arms,” *Mechanism and Machine Theory*, vol. 81, pp. 107–128, 2014.

[25] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. Van Der Smagt *et al.*, "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm," *Nature*, vol. 485, no. 7398, pp. 372–375, 2012.

[26] K. Yamane, M. Revfi, and T. Asfour, "Synthesizing object receiving motions of humanoid robots with human motion database," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 1629–1636.

[27] V. Prasad, R. Stock-Homburg, and J. Peters, "Learning human-like hand reaching for human-robot handshaking," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3612–3618.

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

[29] W. Suleiman, E. Yoshida, F. Kanehiro, J.-P. Laumond, and A. Monin, "On human motion imitation by humanoid robot," in *IEEE International conference on robotics and automation*, 2008, pp. 2697–2704.

[30] J. Konczak and J. Dichgans, "The development toward stereotypic arm kinematics during reaching in the first 3 years of life," *Experimental brain research*, vol. 117, no. 2, pp. 346–354, 1997.

[31] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[32] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.

[33] D. Giavarina, “Understanding bland altman analysis,” *Biochemia medica*, vol. 25, no. 2, pp. 141–151, 2015.

[34] E. Kokkoni, Z. Liu, and K. Karydis, “Development of a soft robotic wearable device to assist infant reaching,” *Journal of Engineering and Science in Medical Diagnostics and Therapy*, vol. 3, no. 2, 2020.