

Reinforcement Learning-Based Assistive Framework for Disabled Persons

Muhammad Talha

Quaid-e-Awam University of Engineering Science and Technology

Muhammad Aamir Panhwar (✉ muhamadtalha1@hotmail.com)

Mehran University of Engineering & Technology

Kamran Ali Memon

Beijing University of Posts and Telecommunications

Abbasi A

Mehran University of Engineering & Technology

Ghazala Panhwar

Mehran University of Engineering & Technology

Research Article

Keywords: Robot, reinforcement learning, liquid pouring, human, sensors

Posted Date: October 1st, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-939770/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Abstract

Robot programming by demonstration (PbD) enables human users to teach the robot and increases its capabilities by interactive teaching without having to manually program the robot. PbD combined with intelligent machine learning algorithms can help us to develop autonomous robots for various industrial and domestic tasks. One such task is the pouring of liquids from bottle into the cup/glass. In this paper the first step is to teach the robot liquid pouring task by the human user in order to facilitate physically disabled people in making various types of drinks and then dataset is created from the user taken demonstrations. In training stage the dataset obtained is feed to the decision-making algorithm based on reinforcement learning. The algorithm enables the robot to learn to pour different liquids under different pouring conditions with the help of minimum number of human user demonstrations needed for the learning of task. The acquired results show that the robot can learn to pour different liquids and is able to accurately adapt to different pouring conditions by using reward-based system and online feedback. Furthermore the results show that the proposed framework can work with different types of liquid and container sizes without any further need for reprogramming or demonstrations.

1 Introduction

Autonomous assistive robots aim to help humans in those tasks which are not possible by humans or to help people in their daily life activities for better living standards. Recent developments in the field of robotics and availability of computationally fast algorithms enable the robots to interact with humans and learn skills from human examples. Assistive robots can provide the most efficient way of assisting humans, especially physically disabled people to make their life independent [1] [2].

There are different ways to execute the pouring process such as tilting type pouring [3], pressure type [4], etc. The main task in this research is to control the falling position of the pouring container in such a way that the required amount liquid is poured without spilling liquid from the container. Numerous control schemes such as [5][6] and machine learning techniques combined with PbD [7] [8] have been developed to control the liquid level and flow rate of the liquid in order to avoid liquid spilling and sloshing of liquids. In [9] teleoperation teaching is used for the demonstrations and force-based perception is used for pouring operation. Kinesthetic teaching is used in [10] for learning of motion primitives using motion refinement tube. In [11] task is demonstrated by using motion-based sensors and markers attached to the robot in order to design humanoid behaviors. In our research the precise control of robot gripper rotation is achieved by using robot operating system (ROS) for the collection of demonstration. Furthermore, the liquid flow rate and gripper rotation are controlled by measuring the liquid weight poured in the cup. This eliminates the use of multiple sensors.

Tilting type automatic pouring system is developed in [5] by using a mathematical model that is based on Navier-Stokes and two-dimensional continuity equations. The proportional control schemes are used in [5] for the controlling purpose. Container's falling pouring position and also the height between the cup and container are controlled in [6] by feedforward control system which consists of a motor model, the

pouring model, and the free fall process. Tilting type automatic liquid pouring system is proposed in [12], where the modeling of flow rate is designed by using mass equilibrium and Bernoulli's theorem. In our research, as the robot learns the pouring task with help of human taken demonstration, so it doesn't require any complex mathematical modeling in order to establish a relationship between liquid flow rate and robot gripper rotation. Visual feedback is used for feedback purposes in [8] in order to pour specified amount of liquid. In this work weight machine is used to acquire the information of poured water in the cup for controlling the gripper rotation that reduces the computational complexity due to less processing. The current work is based on programming by demonstration for the convenience of user, it uses reinforcement learning to encode the relation between liquid flow rate and robot gripper rotation. The developed system learns new tasks with the help of online feedback.

2. Literature Review

In the recent few decades, industrial robots have become a significant commercial success. Due to the growing technology in robotics and the development of intelligent and fast algorithms, robots are not only limited to industrial applications but their use is also extending to domestic and assistive applications. An assistive robot is one that provides aid or support to a human user. Assistive robots can be used to improve the independence and quality of life for people with disabilities. Assistive robotics covers a wide range of research areas such as adaptable robots that act as aides or companions [13] or robots that are designed to provide physical assistance for mobility and support [14].

In the domestic domain, several kinds of assistive robots have been developed to provide assistance to elderly people in homes and hospitals. The robot pearl was designed and developed to autonomously guide elderly people with cognitive and physical disabilities [15]. Different robots have been designed to emulate domestic animals such as a cat or a seal in order to provide companionship to elderly people. The purpose of the research was to reduce the stress level in elderly people and nursing staff [16]. Various researches have been carried out to help people with physical impairments. Assistive robots in this domain are usually designed as wheelchairs or robot manipulators. FRIEND robot [17] [18] is an example of such an intelligent wheelchair. The robotic wheelchair was mounted by the robotic manipulator and different sensors for the perception of its surrounding environment like stereo camera vision, force-torque and weight measuring sensors. The robot wasn't only to provide mobility to the disabled person but also help in daily life activities such as serving food, drinks, etc. and to support at work [16]. Four different generations of FRIEND robot is shown in Fig. 1.

Research has shown that assistive robots can help children with cognitive order through therapeutical interaction [19]. In another research [20] assistive robot was employed to help people with cognitive impairments. The robot in such a case acted as a motivator and assisted people through cognitive stimulating challenges.

The approaches proposed in this research for a liquid pouring task are based on machine learning techniques. As described in [5] [6] and [12], the mathematical model of liquid flow rate and its environment is needed for controlling the pouring of a liquid. However, by using machine learning techniques such as [7] and [21] only human demonstrations are needed for learning and robot performs quite well even in the presence of disturbance and in unknown scenarios. The first approach proposed here is somehow similar to [7] in which the hidden Markov model, Gaussian mixture model, and Gaussian process regression are used to encode the relation between the robot's gripper trajectory and weight of the liquid in the cup. And Gaussian mixture regression is used to retrieve the learned trajectory. The second proposed approach is based on reinforcement learning as in [21], but instead of learning DMPs, a decision-based control system is developed by learning required actions of gripper arm for the given pouring conditions. The learning algorithm is reward-based and the robot tries to find the optimal path by selecting actions that give maximum reward values. In this way, a generalized algorithm is developed which can work with any type of pouring container and conditions because the algorithm can be updated online thus it can learn new pouring actions that are not given at the timing of training.

3 Experimental Setup

In our experiment, a square table is used on which the robotic arm is placed at a fixed. It is assumed that the robot has already gripped the bottle and it is already at the pouring location. The cup is placed on the weight machine at the target location. With the help of the experimental result, the initial tilt angle of gripper which is the angle after which the pouring process starts is calculated. Different types of cup sizes and bottle sizes are used in different experiments. The experimental setup is given in Fig. 2.

The robot is controlled by the user through ROS. The gripper rotation is discretized and only predefined gripper rotations (or actions) are allowed. The gripper rotation steps that can be taken by the robot are of 2° , 3° , 4° , 5° , 6° , 7° , and 8° . The user has to take demonstrations with all the gripper actions one by one by completely filling the empty cup and the pouring bottle is initially full. In this way, the overall demonstration needed for learning is seven. However, the second approach for learning used in this research needed fewer human demonstrations and is explained in Sect. 5.3.2. The demonstrated gripper data and corresponding weight machine data are synchronized with the help of timestamps.

The change in the quantity of liquid filled in the cup for all demonstrations using different gripper rotation step sizes or actions is shown in the figure. Where the x-axis shows the amount of liquid for a given sample in the y-axis. Figure 3 shows that the bigger gripper rotations pour more amount of liquid but as the level of cup rises sufficiently than smaller actions are needed to pour less water to avoid overspilling. The proposed method uses this feature to fill the cup with the desired amount of liquid without spilling.

Hardware used in this work consists of Jaco 2, 7 DOF spherical robotic arm for the pouring of liquid and an electronic weight balance to calculate the weight of the poured liquid in the cup for every robot gripper rotation. The robot gripper consists of three fingers and can manipulate different physical objects. The movement of the robot can be controlled by the controller or computer. The weight machine used is PCE-

PM 1.5 T. The weight machine data with timestamps is transferred online from the weight machine to the computer via serial communication. ROS is used for manipulation of robotic arm and data logging. Many functions can be performed in ROS such as control and planning, sensor processing and evaluation, etc.

There are three ways to control the Kinova robotics arm.

1. **Joystick control:** Joystick controls the arm in Cartesian velocity or angular velocity. The Kinova standard controller is a 3 axis joystick which includes 5 independent push buttons and 4 external auxiliary inputs. Details of Kinova standard joystick are shown in figure 4.
2. **Kinova software control:** There are two different software control panels called the development center and the torque console. They allow the control of robot arm movement in position, velocity, trajectories as well as to activate the direct torque control/force control.
3. **Application programming interface (API) control:** Kinova provides a library of functions to control its robotic arms. This library is called API. It is shown in Fig. 5.

4 Software

Software platforms that are used for designing algorithms for teaching pouring task to the robot are MATLAB and Jupyter notebook (python). For open-loop control of the robot, MATLAB is used and closed-loop decision-based robot control is implemented in Jupyter notebook using python. Jupyter notebook is an open-source online application and basically, a server client-based application used to create and share documents that can contain equations, visualization, and live coding, etc. Robot manipulation and data logging are carried out in ROS in Ubuntu. ROS is a modular framework and consists of many functions such as planning and control, evaluation, sensory processing, etc. It is mainly used for robotics but it can be used for other technical systems.

5 Demonstrations Via Kinova Joystick

The first method includes the teaching of a robot through Kinova joystick. In the method, the user has to operate the robot through a joystick and show the robot how to pour the liquid. In this experiment, the user controls the continuous movement of the gripper tilt angle until all the required liquid is successfully transferred from the bottle into the cup. Several demonstrations are performed in the same manner and at the end of each demonstration, robot and weight machine data are recorded.

6 Demonstration Via Kinova Software Control

In this method, the user controls the robot through Kinova software control. It allows precise movement of the Kinova robotic arm. The gripper joint angles are discretized and only specified gripper actions are allowed. The user can rotate gripper joint only in the step of 2°, 3°, 4°, 5°, 6°, 7°, and 8°. Demonstrations are taken by pouring liquid with the help of any combination of allowed gripper actions. Robot gripper

data and corresponding weight machine data are recorded with the timestamps. Demonstrated discrete robot gripper data and weight machine data are shown in Figs. 10 and 11.

Figure 10 shows the demonstration with different gripper angle steps. It can be seen that less a number of samples are required to fill the cup with bigger gripper angle steps. But it is not possible to use gripper actions with bigger steps only as in some cases or especially at the end of the demonstration when a cup is about to be filled completely, small gripper angle steps are required. Figure 11 shows the change in the amount of water filled in the cup during respective demonstrations.

7 Feature Selection

After demonstrating the task to the robot, the next step is to create a database for learning algorithms. Only the gripper joint is considered for the task and the rest of the joint angles are discarded. The weight data and gripper data are synchronized with each other with the help of timestamps. Each demonstration contains an equal number of weight and gripper angle samples. Several databases having a different number of demonstrations are created to compare the efficiency of different machine learning algorithms. First, databases are created by using only a single demonstration in order to check the response of the system for each demo and other databases having 2 and 4 number of demonstrations are also considered for comparing and improving system performances. Figure 12 shows the amount of liquid change in the cup during the demonstration concerning gripper rotation.

8 Liquid Pouring Framework

The framework of open-loop control of the pouring process through machine learning algorithms is described in Fig. 13.

In open-loop robot control, the robot only learns the pouring actions from given demonstrations and it transfers the same amount of liquid each time. There is no feedback from the output of the system. The overall system performs depend on the parameters of the learning algorithm. The open-loop robot control is implemented in MATLAB. The closed-loop decision-based pouring system is shown in Fig. 14.

In closed-loop control, the user has more options and flexibility while teaching the robot. The algorithm needs information about the size of the bottle, the required amount of water and online feedback from the user. Due to user-provided feedback, the algorithm has become more robust and accurate and at the same time provides pouring of different fluids from different bottles with little information provided by the user.

9 Implementation Of Pouring Task By Using Reinforcement Learning

Reinforcement learning provides learning from interaction and it is based on how the environment responds to the performed action. Bigger rewards are assigned to those actions that take the system closer to its target goal. The relationship between gripper actions and liquid flow out from the bottle is

developed by using reinforcement learning. The overall system is based on reward-based learning. The task of the system is to achieve the goal (i-e to fill the cup with a defined amount of liquid) by taking actions that can give maximum rewards and leads the system towards its final goal. The flowchart in Fig. 17 shows the used approach for the learning of robot using reward-based reinforcement learning.

The reinforcement learning components are defined as

- I. A set of K numbers of state $[S_1, S_2, \dots, S_k]$.
- II. A set of N number of possible actions $[a_1, a_2, \dots, a_N]$ And each action must affect the state of the system.
- III. A set of rewards for every possible action based on the current state of the system $[r_1, r_2, \dots, r_M]$.
- IV. A set of transition probabilities $[S_{11}, S_{12}, \dots, S_{1K}, S_{21}, S_{22}, \dots, S_{2K}, \dots, S_{KK}]$ that defines the possibility of moving one state to another state.

In our approach, three parameters play an important role in the learning of the task. These parameters are the states of the system, the actions that can be performed by the robot and the reward which is based on the amount of liquid poured in response to any robot action.

10 Reward Matrix

Each possible state of the system is represented by the reward matrix and it also shows the actions that can be taken from any given state in order to carry out a transition from one state to another state. Two approaches are used in this research for constructing the reward matrix.

10.1 Approach I

The first demonstration is taken by using only 2-degree rotation by completely filling the empty cup. The amount of liquid poured during each action is considered a reward value. The total amount of liquid present in the cup after each action is regarded as the state of the system. The state sequence sample after the 2-degree demonstration is shown in Table 1.

Table 1
State sequence for reward matrix

Gripper angle from an initial position	2°	4°	6°	8°	10°	12°	14°
Weight in grams	8	49.5	101.5	184.5	265	312	350

Then the user will perform the same experiment by using all remaining gripper actions. With the help of these demonstrations information about each action from any given state is stored in the reward matrix. The elements of the reward matrix represent the amount that will be poured in response to any action

from the given state. Table 2 shows the reward matrix for pouring water. Where - 100 represents that the respective action is not allowed from the current state.

Table 2
Reward matrix for pouring water using seven demonstrations of the task

Actions	2°	3°	4°	5°	6°	7°	8°
States							
1	8	1	56.8	37	45	111.5	153.6
2	41.5	55	70.5	6.5	180	204.5	84.9
3	52	77.5	10	3.5	99	158	119.25
4	83	106	176	268	108	158	119.25
5	80.5	107.5	27.5	78.5	-100	-100	-100
6	47	-100	-100	-100	-100	-100	-100
7	38	-100	-100	-100	-100	-100	-100

10.2 Approach II

In the first approach, seven demonstrations are needed for learning, which is quite time-consuming and difficult for the user. In this approach, the user has to take only two demonstrations with a 2-degree gripper action and 3-degree gripper rotation. Data for the rest of the actions are estimated by the algorithm based on the given demonstrations. The reward matrix with estimated reward values is given in Table 3.

Table 3
Reward matrix based on estimated values

Actions	2°	3°	4°	5°	6°	7°	8°
States							
1	8	1	49.5	75.5	101.5	143	184.5
2	41.5	55	135	172.75	210.5	71.5	184.5
3	52	77.5	127.5	124.15	156	107.25	184.5
4	83	106	104	124.15	156	107.25	-100
5	80.5	107.5	-100	-100	-100	-100	-100
6	47	-100	-100	-100	-100	-100	-100
7	38	-100	-100	-100	-100	-100	-100

The calculation of estimated reward values from the given demonstrations are shown in Table 4.

Table 4
Calculations criteria for the prediction of reward values

Step size	Formula to predict
4	$2[i] + 2[i + 1]$
5	Average($4[i]$, $6[i]$)
6	$2[i] + 2[i + 1] + 2[i + 2]$
7	Average($6[i]$, $8[i]$)
8	$2[i] + 2[i + 1] + 2[i + 2] + 2[i + 3]$

In Table 4, $i = 1, 2, 3, \dots, 7$ represents the state of the system.

11 Working Process

Once the reward matrix is created with the help of a demonstration, the robot is now able to pour the liquid and its online training is beginning which depends on the data of the reward matrix. In online training, Q-matrix is initialized with the help of a reward matrix. An initial amount of cup, the liquid present in the bottle and user requirements of liquid are also taken into consideration for suitable actions. Depending on the initial amount of liquid in a bottle the algorithm determines the initial tilt angle of the bottle after which the pouring process begins. According to the required amount of liquid and current state of the system (initial amount of liquid in cup) the algorithm chooses the most appropriate action from the given state. After each action the algorithm gets the feedback of the actual amount of liquid

poured in response to a certain action. This value is then compared with the predicted reward value, if there is a significant difference in predicted and actual reward values then the reward value for that action is updated in the Q-matrix. The algorithm completely learns the pouring task when there reward values in Q-matrix stop updating and algorithm converges.

12 Results

In this research, various experiments are done with different liquids and container sizes and also with different initial conditions. In all experiments, there are some user given requirements required by the algorithm to predict the current environment such as maximum cup capacity, required amount of water, the initial amount of water in cup and bottle and the maximum capacity of the bottle.

1. The first experiment is done by pouring water and reward matrix used herein constructed from seven user demonstrations and specifications of the experiment are maximum cup capacity: 350g required amount of water: 350 g, initial amount of water in the cup: 0 g, weight of liquid in a bottle: 434 g, maximum capacity of the bottle: 500 g. The Q-matrix used for prediction of reward used is given in Table 5.

Table 5
Initial Q-matrix based on reward matrix for training

Actions	2°	3°	4°	5°	6°	7°	8°
States							
1	8	1	56.8	37	45	111.5	153.6
2	41.5	55	70.5	6.5	180	70	84.9
3	52	77.5	10	3.5	99	158	119.25
4	83	106	176	-100	108	158	119.25
5	80.5	107.5	27.5	78.75	-100	-100	-100
6	47	-100	-100	-100	-100	-100	-100
7	38	-100	-100	-100	-100	-100	-100

In the 1st trial, the actions performed by the robot to fill the required amount in a cup is given in Table 6.

Table 6
1st experiment's 1st trail

Iteration	Action	Predicted liquid level	Actual liquid level	Error
1	8°	153.6 g	110g	43.6 g
2	4°	286 g	221 g	65 g
3	3°	328.5 g	300 g	28.5 g
4	2°	347 g	344 g	3 g

In the 1st trial, the robot fills the cup with 344g of liquid and the difference between the actual amount and predicted is only 6g. However, three reward values are updated in the Q-matrix so for the next trial the robot should not do a similar mistake. The updated matrix is given in Table 7.

Table 7
Updated Q-Matrix after pouring water

Actions	2°	3°	4°	5°	6°	7°	8°
States							
1	8	1	56.8	37	45	111.5	110
2	41.5	55	70.5	6.5	180	70	84.9
3	52	77.5	10	3.5	99	158	119.25
4	83	106	111	-100	108	158	119.25
5	80.5	79	27.5	78.75	-100	-100	-100
6	47	-100	-100	-100	-100	-100	-100
7	38	-100	-100	-100	-100	-100	-100

The value highlighted with red color represents the value that is updated in the Q-matrix.

II. In the second experiment, the water is poured in the cup but the reward matrix used here is constructed by only two demonstrations and the rest of the values are predicted. The user requirement is the same here as in experiment 1. The Q-matrix from predicted data is given in Table 8.

Table 8
Q-matrix for predicted data

Actions	2°	3°	4°	5°	6°	7°	8°
States							
1	8	1	49.5	75.5	101.5	143	184.5
2	41.5	55	135	172.75	50	71.5	184.5
3	52	77.5	127.5	124.125	156	107.25	184.5
4	83	106	104	124.125	156	107.25	-100
5	80.5	107.5	-100	-100	-100	-100	-100
6	47	-100	-100	-100	-100	-100	-100
7	38	-100	-100	-100	-100	-100	-100

In the 1st trial, the actions performed by the robot to achieve the tasks are given in Table 9.

Table 9
1st trial iterations using predicted data

Iteration	Action	Predicted liquid level	Actual liquid level	Error
1	8°	184.5 g	208 g	23.5 g
2	3°	315.5 g	288 g	27.5 g
3	2°	335 g	330 g	5 g

The final amount of water in the cup is 330g which represents the error of 20g because the required liquid was 350g. However, as the robot performs the task it learns about the mistake it made during tasks by online learning. In the 2nd trial, the robot action sequence is given in Table 10.

Table 10
2nd trial iterations using predicted data

Iteration	Action	Predicted liquid level	Actual liquid level	Error
1	7°	143 g	132 g	11 g
2	6°	288 g	290 g	2 g
3	2°	337 g	335 g	2 g

In the second trial, the robot fills the 335g of water which in comparing the first trial is more close to the required amount. In this way, the level of perfection increases in proportion to the number of iterations it performs.

III. Experimental results for pouring soft drinks In this experiment, the robot is first offline trained with soft drinks by taking two demonstrations and predicted data and the resultant reward matrix represent the amount for every action from the given state. The reward matrix used in the 1st experiment can also be used here but it will take a long time to converge as it was trained for water. The user requirements in this experiment are maximum cup capacity : 280g required amount of water: 280 g, Initial amount of water in a cup: 0 g, Weight of liquid in a bottle: 396g, maximum capacity of the bottle: 424 g, In this scenario the system states are given in Table 11.

Table 11
System states for pouring the soft drink

System states/ Cup states						
0	17	67	129.5	188.5	247.5	280

The reward matrix from the predicted soft drink data is given in Table 12.

Table 12
Reward matrix constructed from the predicted data for pouring the soft drink

Actions	2°	3°	4°	5°	6°	7°	8°
States							
1	0	49.1	17	42	67	98.25	129.5
2	17	3	112.5	146.5	180.5	49.125	129.5
3	50	65.5	118	94.25	123.75	73.6875	129.5
4	62.5	93.5	82.5	94.25	123.75	73.6875	129.5
5	59	83.5	82.5	-100	-100	73.6875	-100
6	59	-100	-100	-100	-100	-100	-100
7	32.5	-100	-100	-100	-100	-100	-100

In the 1st trail, the robot action sequence is given in Table 13.

Table 13
1st trail of the robot action sequence

Iteration	Offset angle	Action	Predicted liquid level	Actual liquid level	Error
1	4°	6°	180.5 g	144 g	36.5 g
2		3°	227.5 g	227 g	0.5 g

The total amount poured in a cup is 227g with an error of 53g. Where the offset angle defines the amount of rotation required to start the pouring process. The actions performed in the second trial is given in Table 14.

Table 14
2nd trial iterations for pouring a soft drink

Iteration	Offset angle	Action	Predicted liquid level	Actual liquid level	Error
1	4°	5°	146.5	219 g	72.5 g
2		2°	278 g	260 g	18 g

In the 2nd trial, the performance of the robot is improved. It poured overall 260g of water with the error of 20g only which is much less as compared to the error of 53g in the first trial. Due to online feedback, the algorithm continuously updates its reward values in Q-matrix until it achieves satisfactory results.

IV. In the last experiment, the water is poured into the glass and some water is already present in the cups. This represents a non-zero initial condition. The updated Q-reward matrix that was used in the 1st experiment is used here. The user requirements are given as Maximum cup capacity: 350g, Required amount of water: 350 g, Initial amount of water in the cup: 77 g, Weight of liquid in a bottle: 263 g, Maximum capacity of the bottle: 500 g. The actions performed by the robot to perform the task are given in Table 15.

Table 15
1st Trial for pouring water under non zero initial condition

Iteration	Offset angle	Action	Predicted liquid level	Actual liquid level	Error
1	8°	8°	299.5 g	180 g	119.5 g
2		4°	285.5 g	296 g	8.5 g
3		2°	341 g	348	7 g

The final value in the cup is 348g and the required amount is 350g. The error of the only 2g. This shows that as the robot performs the operation more and more, its performance becomes much better due to online adaptation.

13 Gaussian Process For Regression

Gaussian process regression is also used to encode and retrieve the required trajectories for the pouring process using different demonstrations. The single demonstration is given to GPR which gives a trajectory for pouring liquids after processing the input trajectory.

Figure 18 shows the input trajectory with a black dotted line and retrieved trajectory is shown by the red solid line. The retrieved trajectory is similar to the demonstrated one and is suitable for the pouring process. It successfully covers all three regions required to pour the liquid.

GPR using two demonstrations gives better results as compared to using a single demonstration except at the start of the pouring process where gripper will go back slightly before pouring begins. This is because of the large difference in the beginning points in both demonstrations. The rest of the regions are successfully reproduced according to the requirements. The results are shown in Fig. 19. The results of GPR with four demonstrations are not very encouraging. As can be seen in Fig. 20, an almost linear relationship between the gripper angle and the amount of liquid has been established, which is not the case here.

14 Comparison Between Predicted And Demonstrated Data

The performance of the decision-based pouring process gives satisfactory results for both demonstrated and predicted data. In both cases, the algorithm is able to perform suitable actions for the pouring process. However, for demonstrated data, the user has to perform seven demonstrations to teach the robot. In the pouring process, it is difficult to maintain the same initial conditions for all the demonstrations. For example, it is difficult for the user to keep the same grasping point of a bottle for the robot gripper at the start of each demonstration. Also, to refill the bottle with the same amount of liquid each time is not possible for a human user. Varying initial conditions for the demonstration may produce wrong reward values for the actions. Additionally, it is time-consuming for the user to demonstrate a pouring task for each gripper action. To avoid this situation, predicted data can be used so that the user has to take only two demonstrations. If the predicted reward values are not similar to actual reward values, the algorithm updates its reward value during online training.

Conclusion

Programming by demonstration and reinforcement learning-based approach is used here in order to teach pouring tasks to an assistive robot so that they prepare drinks, especially for disabled people. The advantage of using PbD is that the robot can be reprogrammed easily by any unskilled user. The proposed approach is based on reward-based learning where the learning agent always tries to improve its action sequence based on reward values it receives in consequence of particular actions. Due to online learning, the generalized pouring model is developed that can work with various sizes of glasses and bottles and with different types of liquids. The results show the satisfactory performance of the algorithm even in unknown environments and have the capability to correct actions sequence that would result in better performance. In future work, the amount of liquid will be measured by visual data instead of the weight of the liquid. The weight machine takes more space and the sampling frequency for a weight machine is not high enough for the pouring process. Furthermore, to measure the weight of the liquid, a separate weight machine has to be installed to give real-time feedback to the algorithm. However, visual devices take less space and are easy to install. With the help of visual data, the volume of the fluid can be estimated, which can be further used to control the pouring process. In this research only gripper rotation is controlled, however, the gripper movement can also be controlled in other dimensions in Cartesian space as well to eliminate fluid sloshing.

Declarations

Acknowledgement

Author 1: ¹Muhammad Talha

☒ Conceived and design the analysis

Author 2: ²Muhammad Aamir Panhwar

☒ Collecting the Data

Author 3: ³Kamran Ali Memon

☒ Contributed data and analysis stools

☒ Performed and analysis Author

Author 4: ⁴Abbasi

☒ Wrote the Paper

Author 5: ⁵Ghazala Panhwar

☒ Editing and Figure Design

Ethical approval: This article does not contain any studies with animals performed by any of the authors.

Funding: This research work is not supported by Government and Non- Governmental Organizations.

Conflict of Interest: The authors declare that they have no conflict of interest.

Informed consent: Informed consent was obtained from all individual participants included in the study

References

[1] G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsh, D. Magaritis, M. Montemerlo, J. Pineau, N. Roy, J. Schulte, and S. Thrun. Towards personal service robots for the elderly. In Proceedings of the Workshop on Interactive Robots and Entertainment, Pittsburgh, PA, April-May 2000.

[2] P. Aigner and B. McCarragher. Shared control framework applied to a robotic aid for the blind. Control Systems Magazine, IEEE, 19(2):40– 46, April 1999.

[3] J. Sato and K. Yoshida: Industrial heating 29 (1992) 19.

[4] V. I. Dubodelov, V. K. Pogorsky and M. S. Goryuk: Proc. 9th International Symposium on Science and Processing of Cast Iron, Egypt (TTP,2009) p.481.

[5] Modeling and Motion Control of Fluid Behavior in Automatic Pouring System Kazuhiko TERASHIMA, Akihiro KANESHIGE, Hiroshi KOYAMATSU, Kazuto YAMAURA, Welly Yoel RONDA and Tomochika INAGAKI.

[6] Outflow Liquid Falling Position Control by Considering Lower Ladle Position and Clash Avoidance with Mold Atsushi Ito, Yoshiyuki Noda, Kazuhiko Terashima.

- [7] Force-based Robot Learning of Pouring Skills using Parametric Hidden Markov Models Leonel Rozo and Pablo Jimenez and Carme Torras.
- [8] Visual Closed-Loop Control for Pouring Liquids Connor Schenck, Dieter Fox University of Washington
- [9] L. Rozo, P. Jim´enez, and C. Torras, “Robot learning from demonstration of force-based tasks with multiple solution trajectories,” in Intl. Conf. on Advanced Robotics, 2011, pp. 124–129.
- [10] D. Lee and C. Ott, “Incremental kinesthetic teaching of motion primitives using the motion refinement tube,” *Autonomous Robots*, vol. 31, pp. 115–131, 2011.
- [11] M. Riley, A. Ude, C. Atkeson, and G. Cheng, “Coaching: An approach to efficiently and intuitively create humanoid robot behaviors,” in Intl. Conf. on Humanoid Robots, 2006, pp. 567–574.
- [12] Modeling and Feedforward Flow Rate Control of Automatic Pouring System with Real Ladle Yoshiyuki Noda and Kazuhiko Terashima, Department of Production Systems Engineering, Toyohashi University of Technology, 1-1 Hibarigaoka, Tempaku-cho, Toyohashi, Aichi 441-8580, Japan.
- [13] G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsh, D. Magaritis, M. Montemerlo, J. Pineau, N. Roy, J. Schulte, and S. Thrun. Towards personal service robots for the elderly. In *Proceedings of the Workshop on Interactive Robots and Entertainment*, Pittsburgh, PA, April-May 2000.
- [14] P. Aigner and B. McCarragher. Shared control framework applied to a robotic aid for the blind. *Control Systems Magazine, IEEE*, 19(2):40– 46, April 1999.
- [15] M. Montemerlo, J. Prieau, S. Thrun, and V. Varma. Experiences with a mobile robotics guide for the elderly. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 587–592, Edmonton, Alberta, August 2002.
- [16] K. Wada, T. Shibata, T. Saito, and K. Tanie. Analysis of factors that bring mental effects to elderly people in robot assisted activity. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 2, pages 1152–1157, Lausanne, Switzerland, October 2002.
- [17] Volosyak, I., Ivlev, O. and Graser, A., 2005. Rehabilitation robot FRIEND II-the general concept and current implementation. In *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on* (pp. 540-544). IEEE.
- [18] Graser, A., Heyer, T., Fotoohi, L., Lange, U., Kampe, H., Enjarini, B., Heyer, S., Fragkopoulos, C. and Ristic-Durrant, D., 2013. A supportive friend at work: Robotic workplace assistance for the disabled. *IEEE Robotics & Automation Magazine*, 20(4), pp.148-159.
- [19] F. Michaud and C. Thberge-Turmel. Mobile robotic toys and autism. In K. Dautenhahn, A. Bond, L. Canamero, and B. Edmonds, editors, *Socially Intelligent Agents – Creating Relationships with Computers and Robots*, pages 125–132. Kluwer Academic Publishers, 2002.

[20] Designing intelligent socially assistive robots as effective tools in Cognitive interventions, Jeanie Chan, Goldie Nejat, and Jingcong Chen, *Int. J. Human. Robot.* 08, 103 (2011).

[21] Learning to pour with a robot arm combining goal and shape learning for dynamic movement primitives, Minija Tamosiunaite, Bojan Nemec, Ales Ude, Florentin Wörgötter.

[22] <http://www.kinovarobotics.com/wp-content/uploads/2017/04/JACO-7-Spherical-DOF-User-Guide.pdf>.

Figures

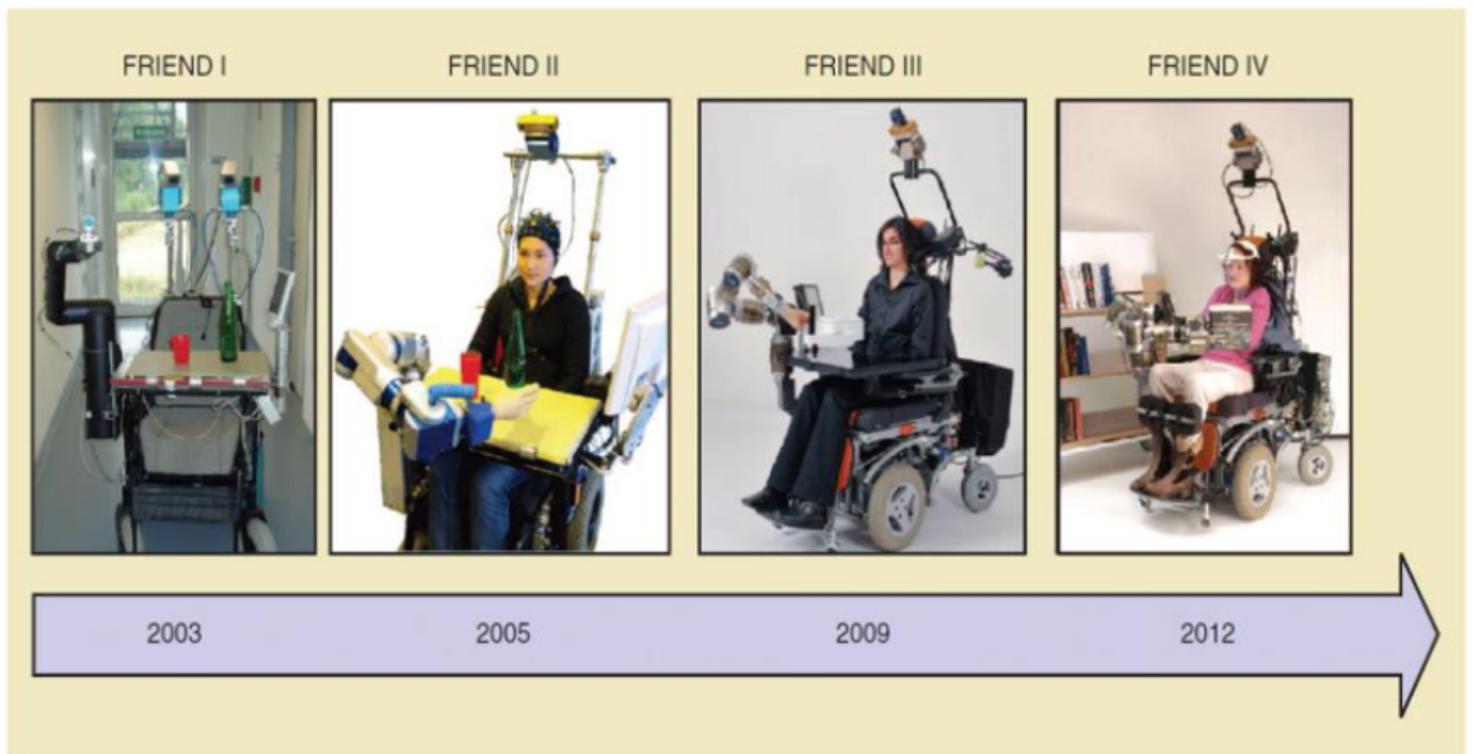


Figure 1

Four different generations of FRIEND robot

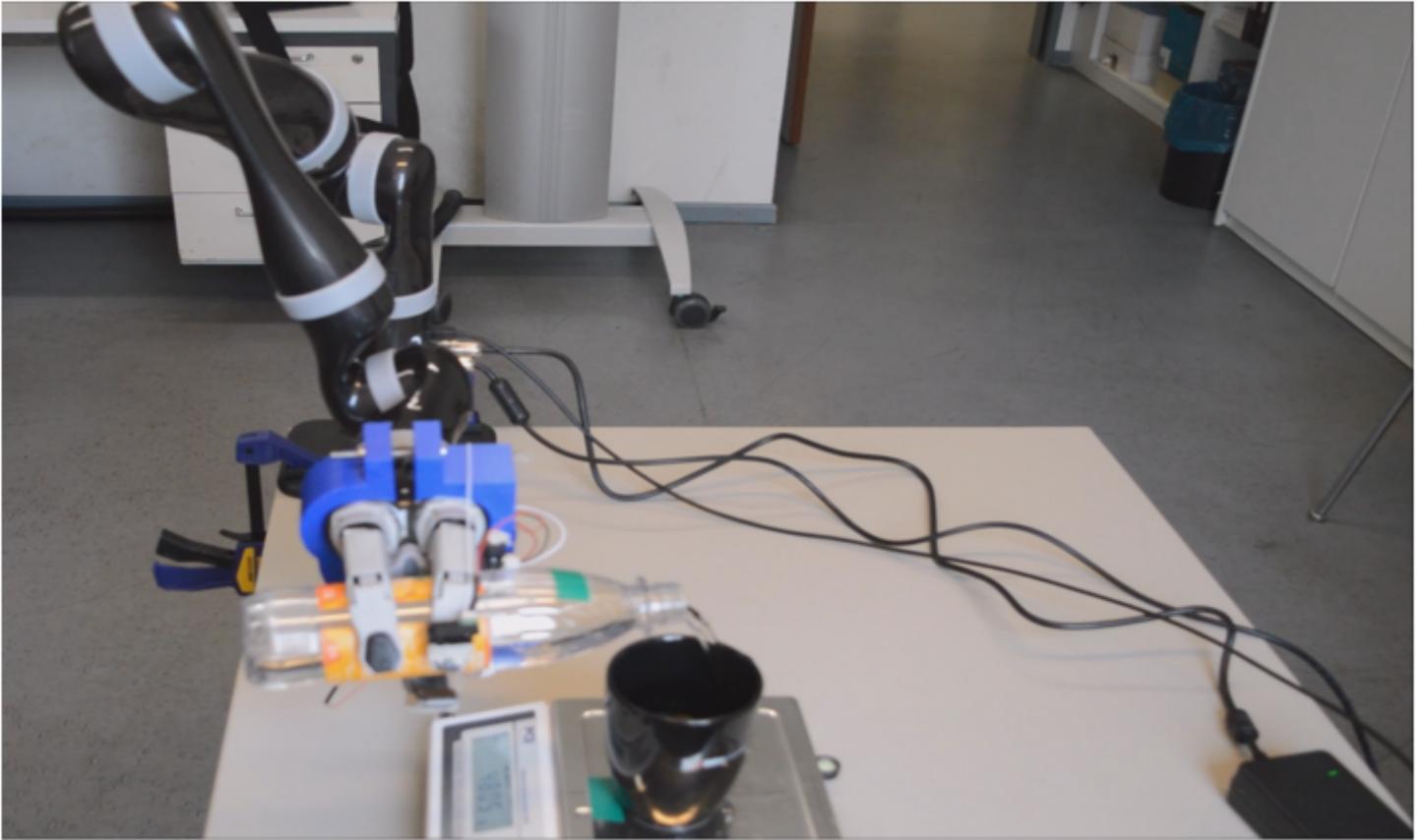
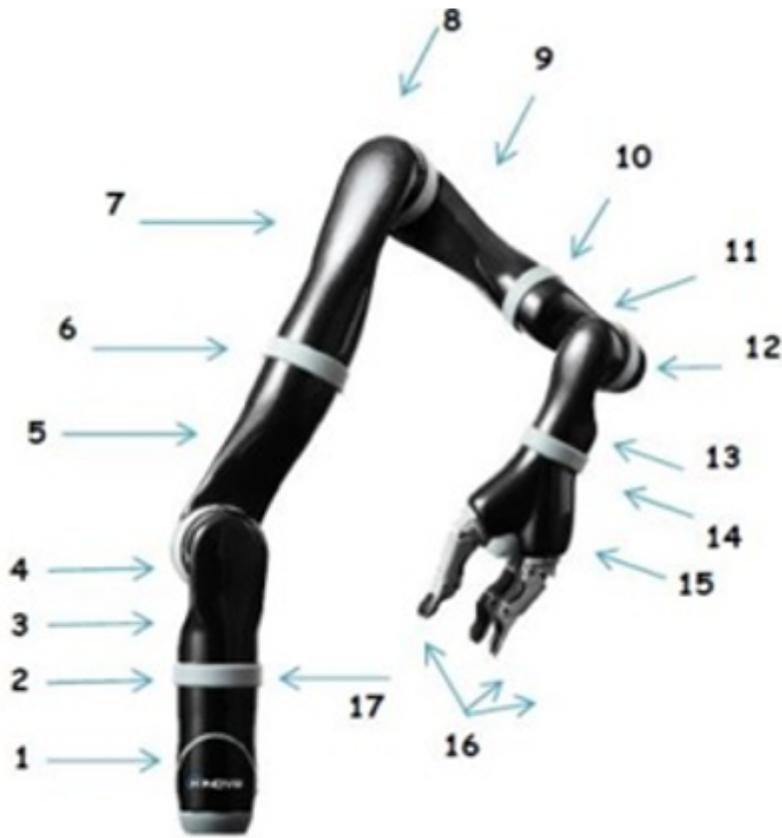


Figure 2

Experimental setup for Robot Pouring System



PART ID	NAME
1	Controller
2	Actuator #1
3	Shoulder
4	Actuator #2
5	Half-arm #1
6	Actuator #3
7	Half-arm #2
8	Actuator #4
9	Forearm
10	Actuator #5
11	Wrist #1
12	Actuator #6
13	Wrist #2
14	Actuator #7
15	Gripper
16	Fingers
17	Plastic Ring

Figure 3

JACO-7 part identification [22]

JOYSTICK PART IDENTIFICATION

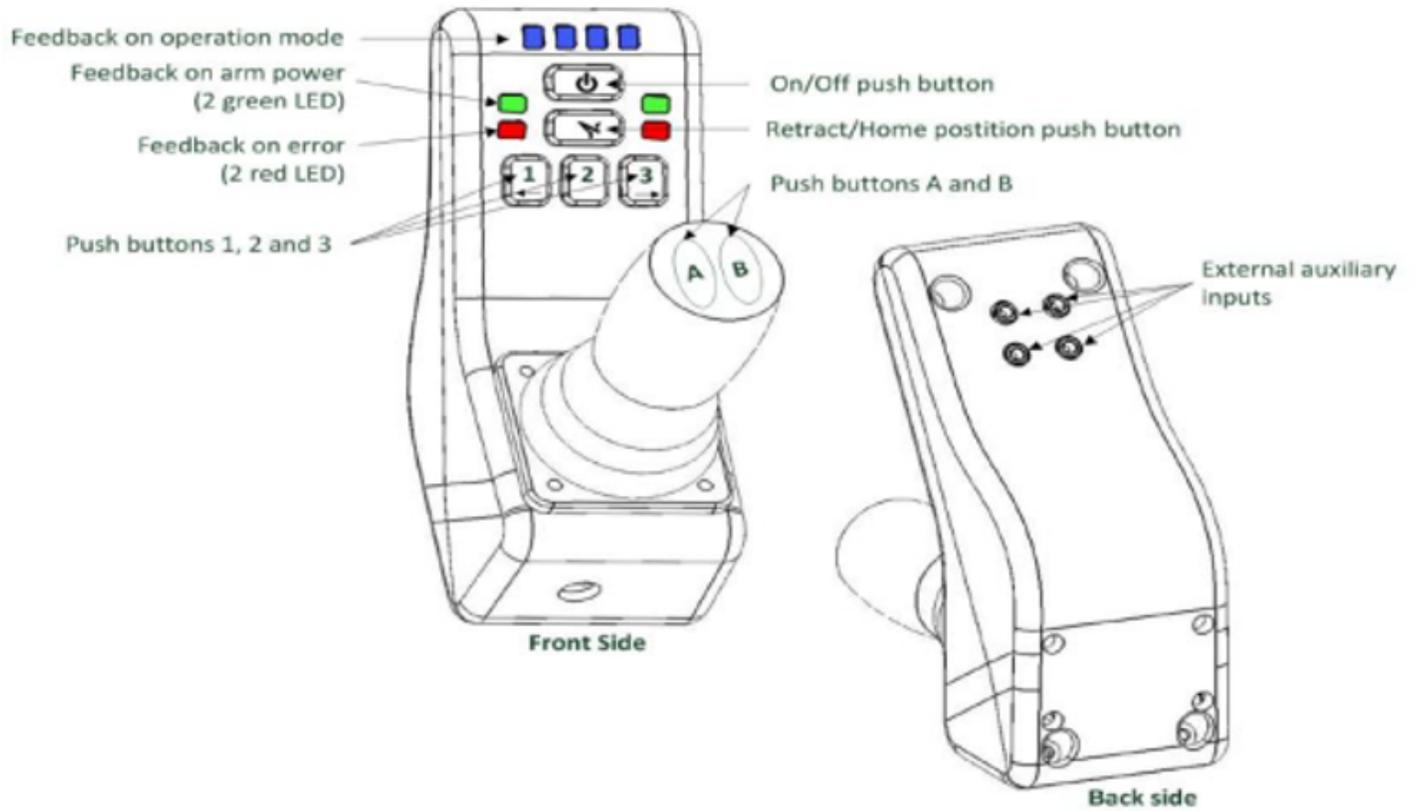


Figure 4

Kenova standard robotic controller [22]

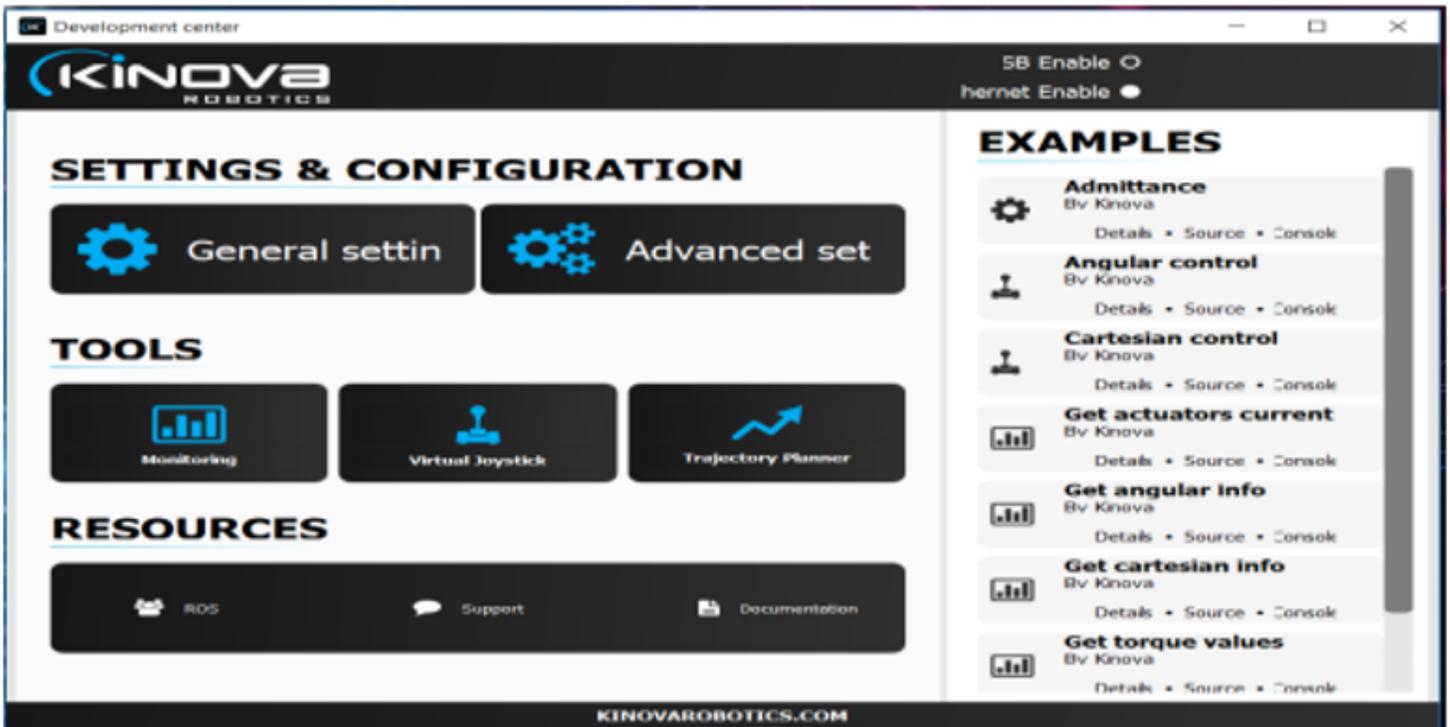


Figure 5

Kinova API for robot control [22]

Image not available with this version

Figure 6

Caption not included with this version.

Image not available with this version

Figure 7

Caption not included with this version.

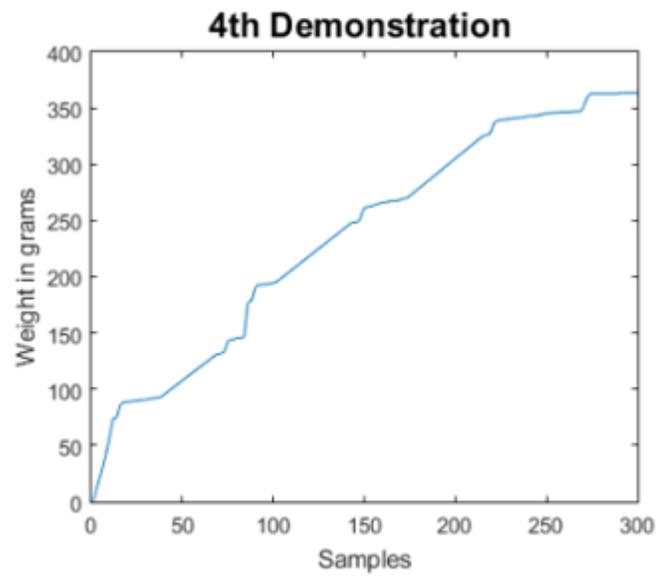
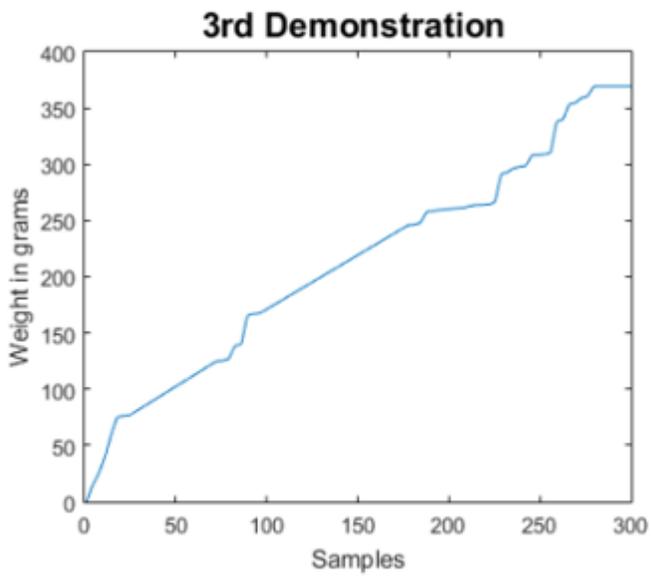
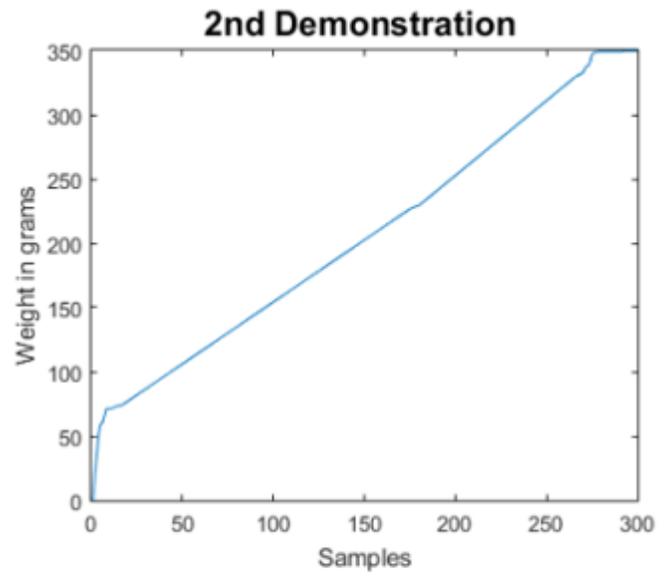
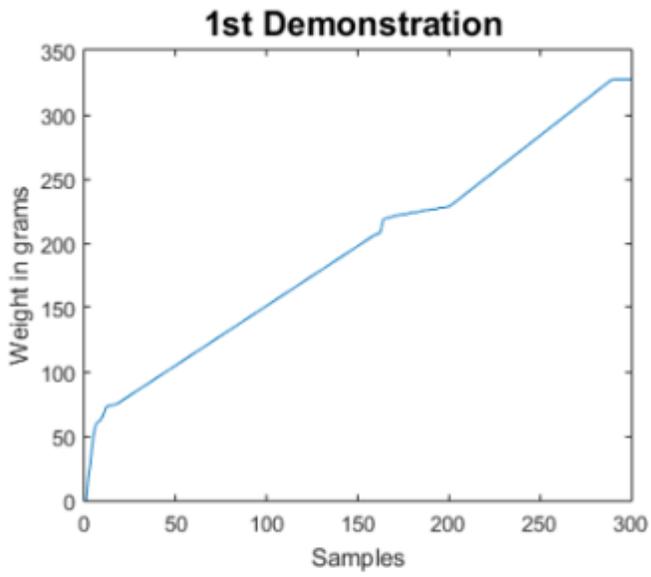


Figure 8

Continuous robot gripper data

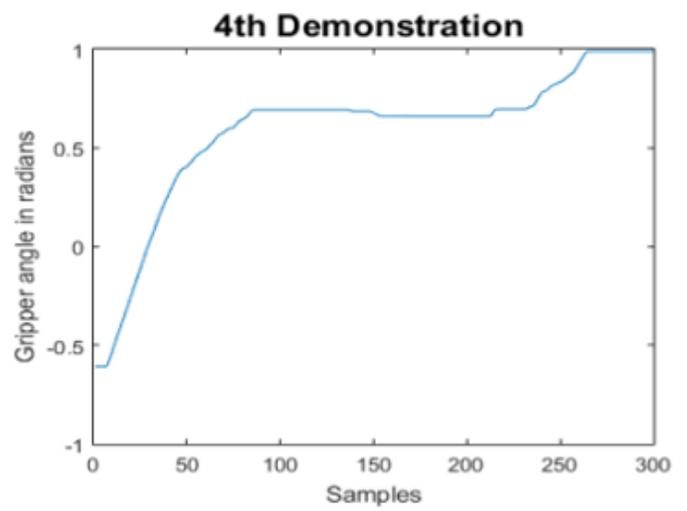
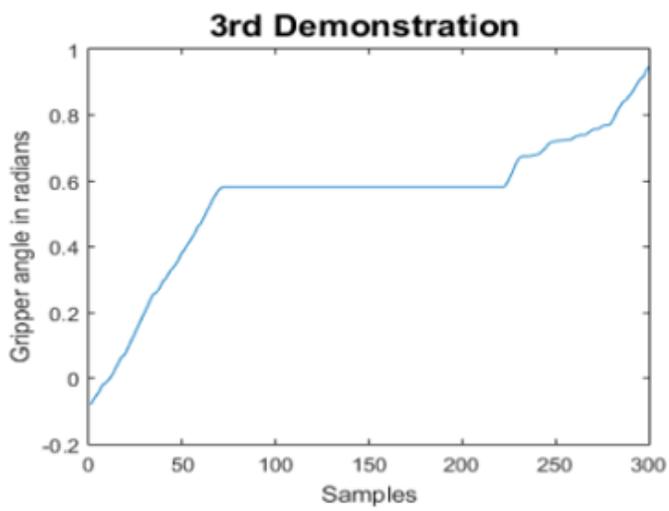
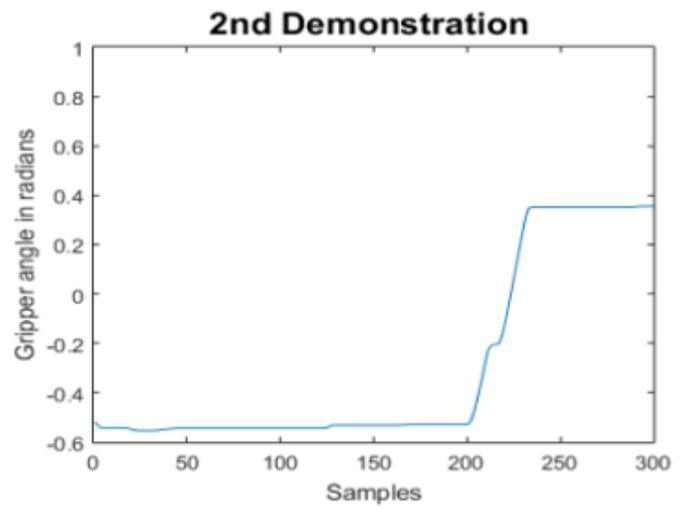
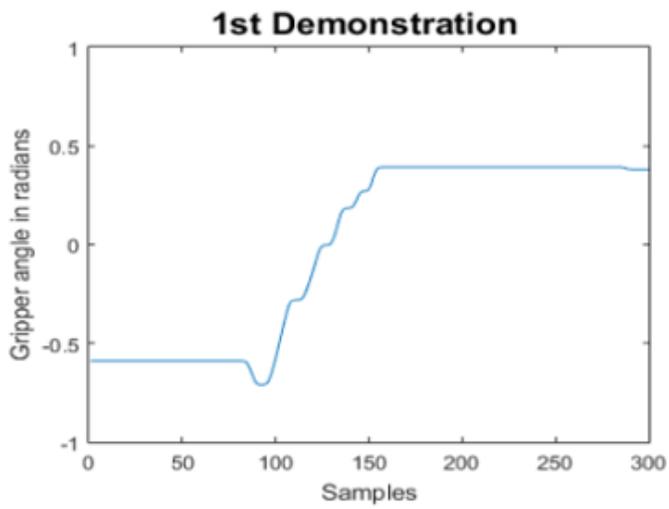


Figure 9

Weight machine data

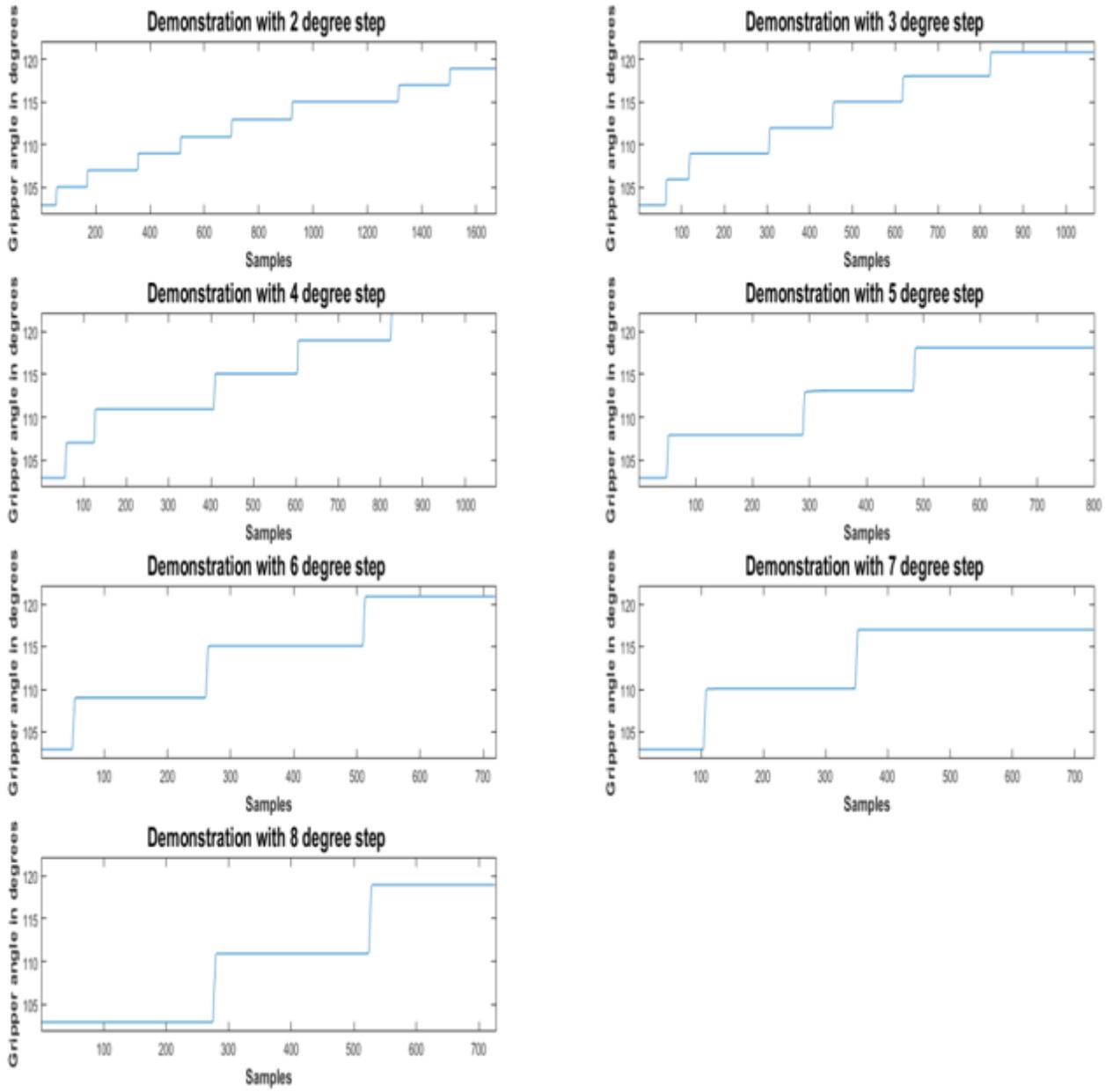


Figure 10

Discrete gripper data

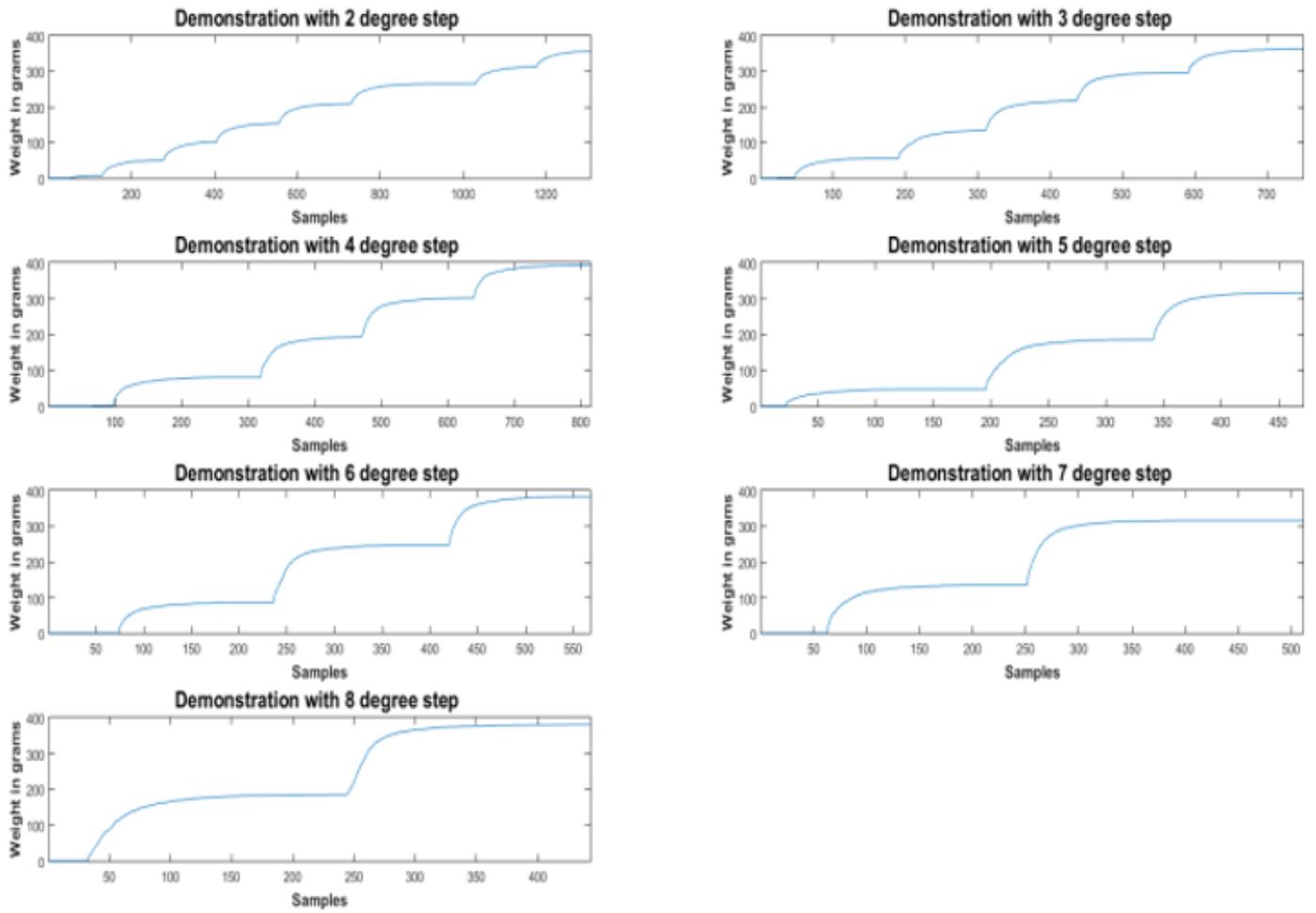


Figure 11

Seven demonstrations of liquid pouring using different gripper rotation steps

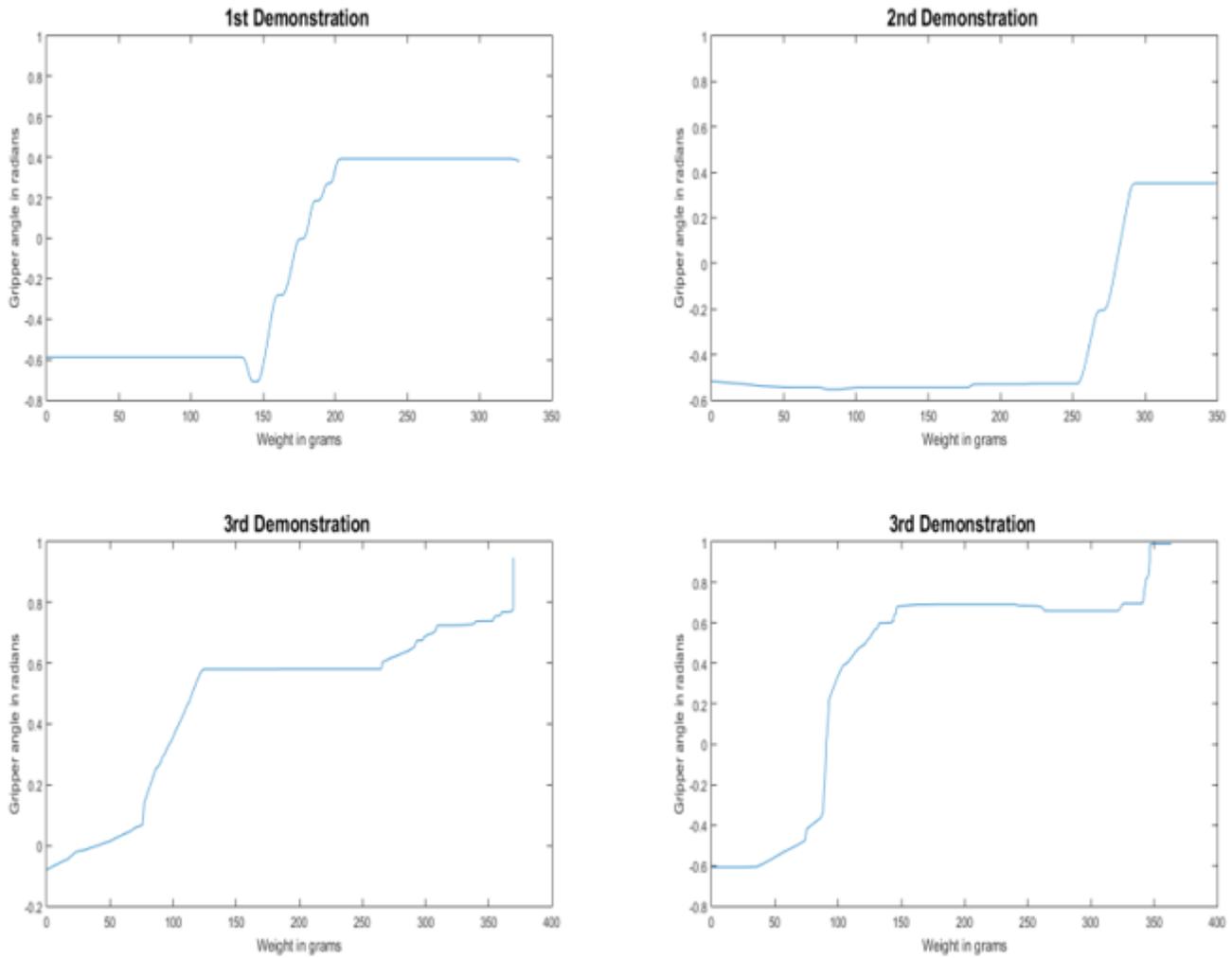


Figure 12

Gripper rotation with respect to change in weight

Probabilistic modeling approach for Robot Control

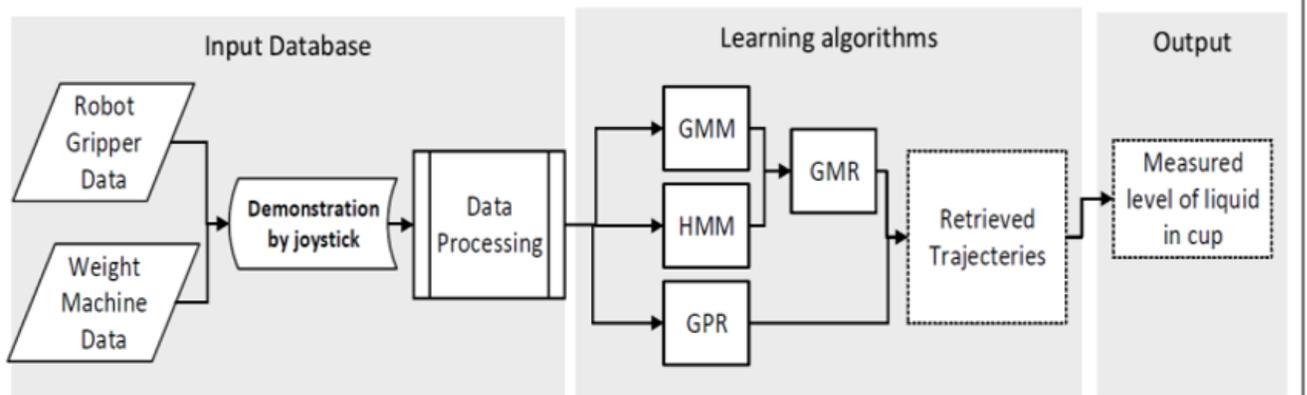


Figure 13

Robot control for pouring liquids using probabilistic modeling approaches

Decision based Robot Control

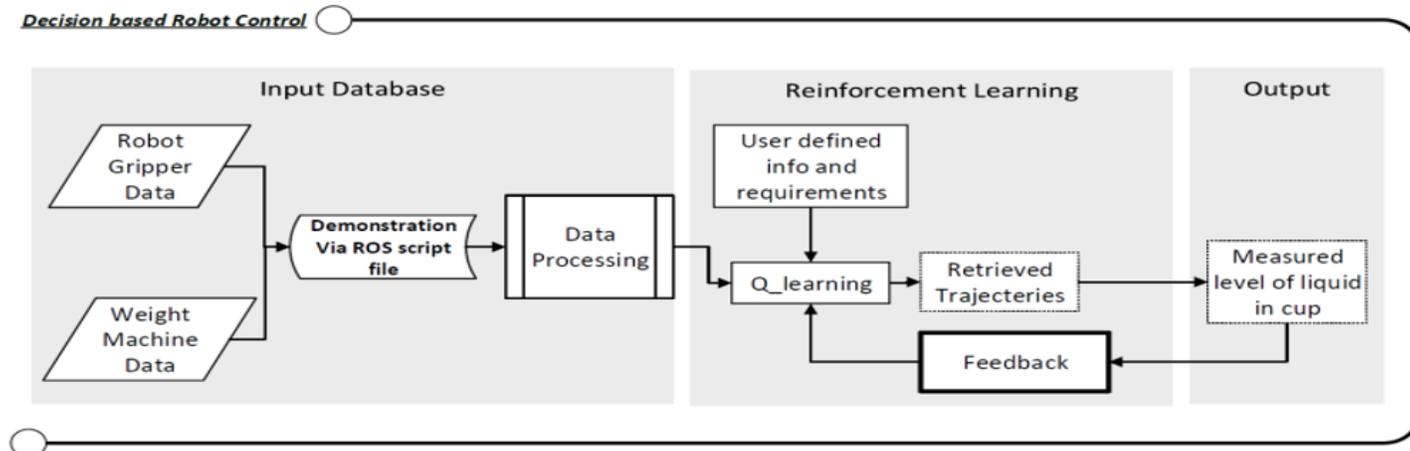


Figure 14

Decision-based robot control for pouring liquids

Image not available with this version

Figure 15

Caption not included with this version.

Image not available with this version

Figure 16

Caption not included with this version.

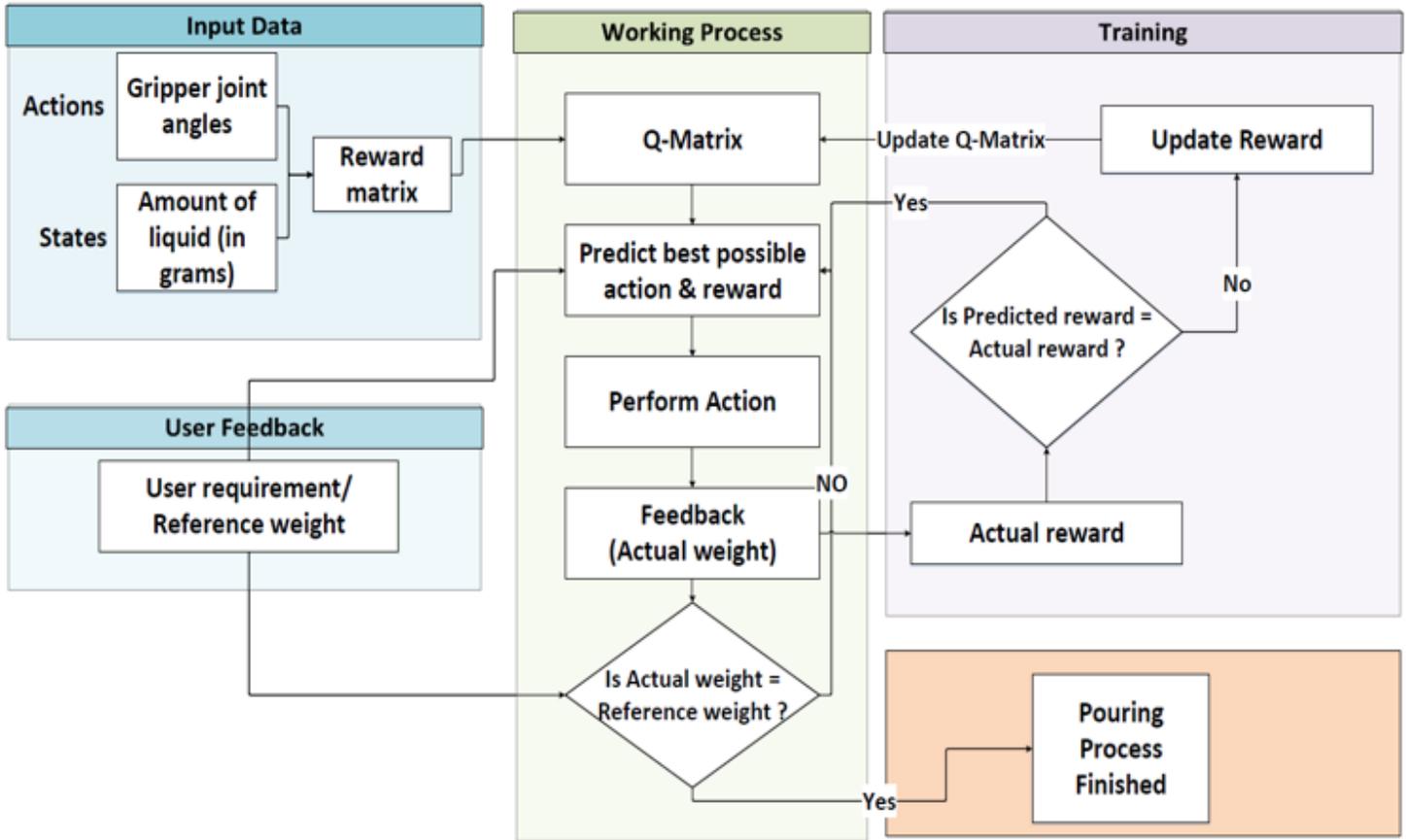


Figure 17

Reward-based reinforcement learning of the liquid pouring task The reinforcement learning components are defined as

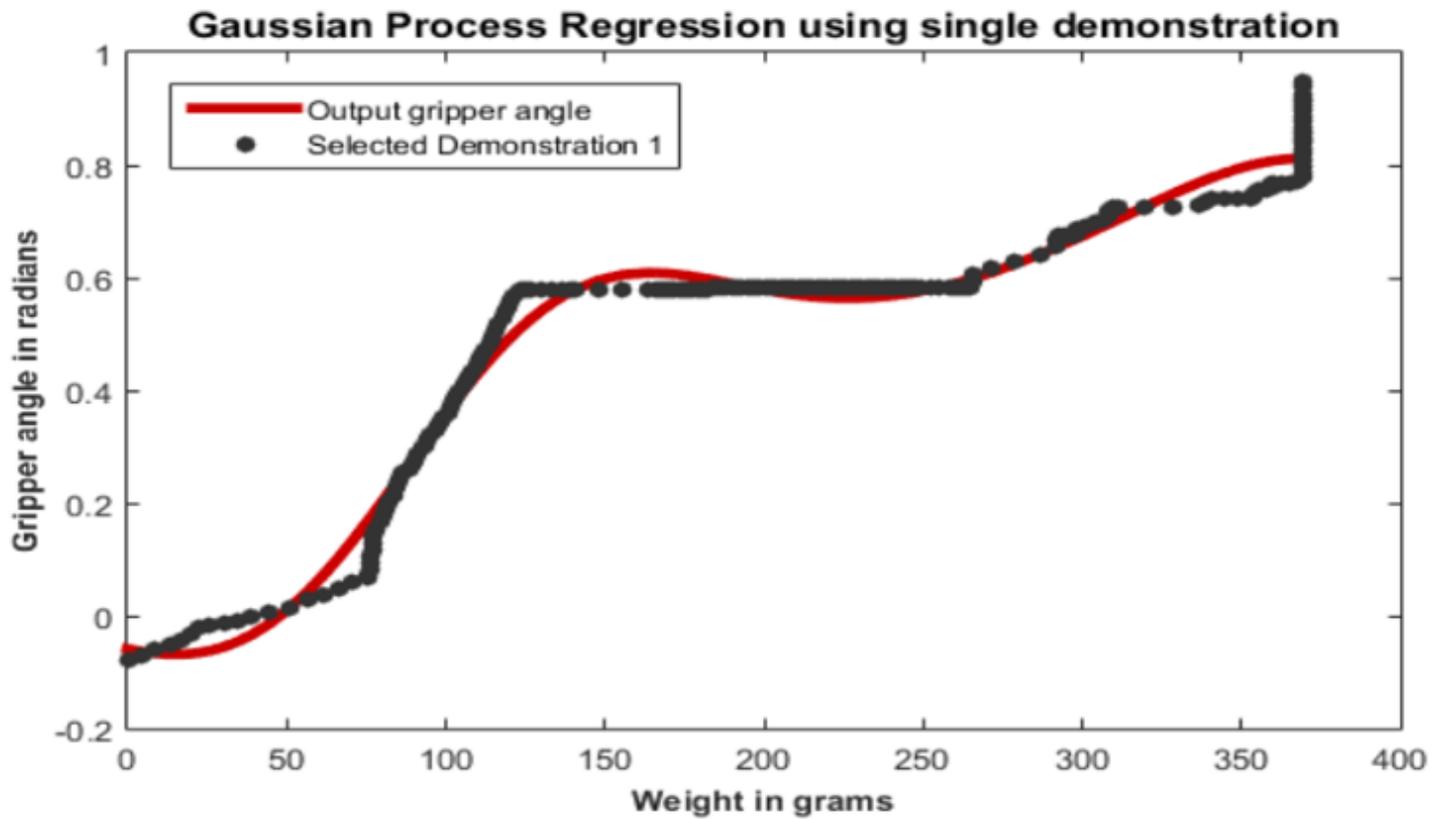


Figure 18

Demonstrated and retrieved trajectories using GPR

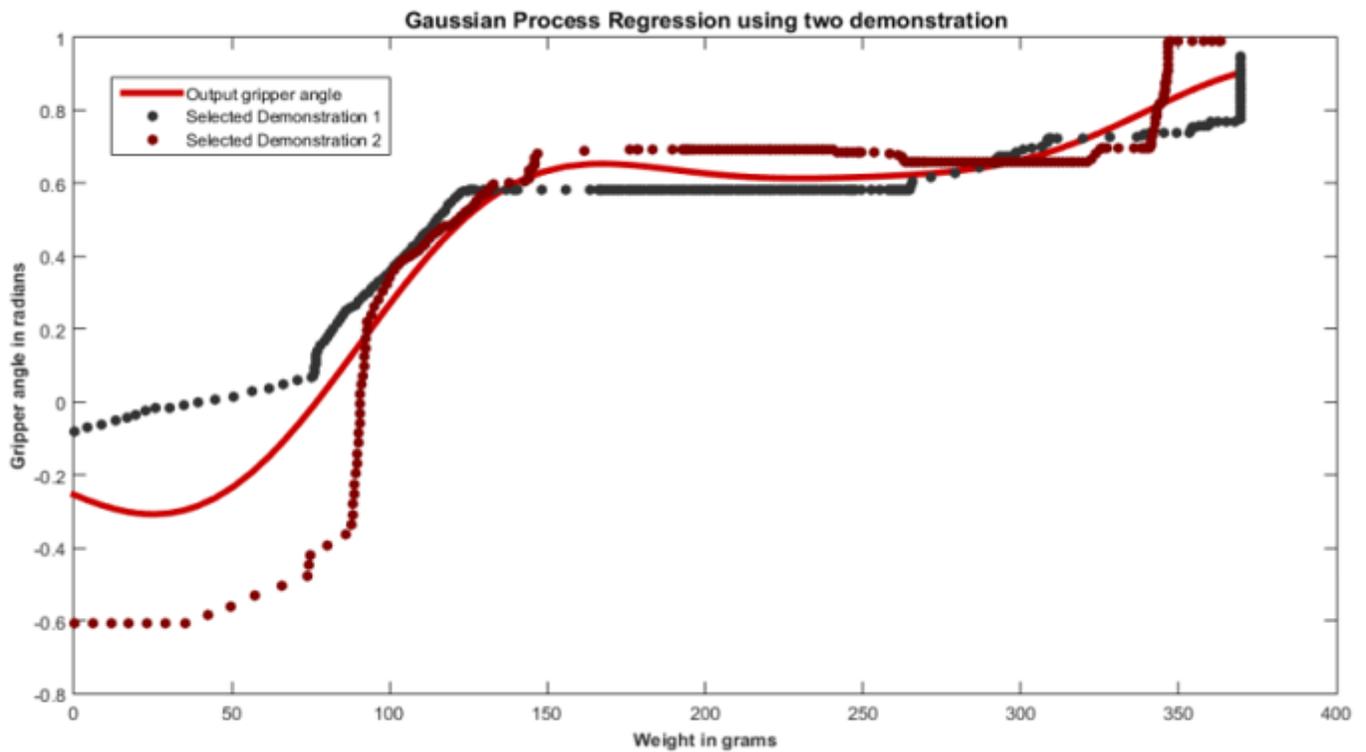


Figure 19

GPR results using two demonstrations

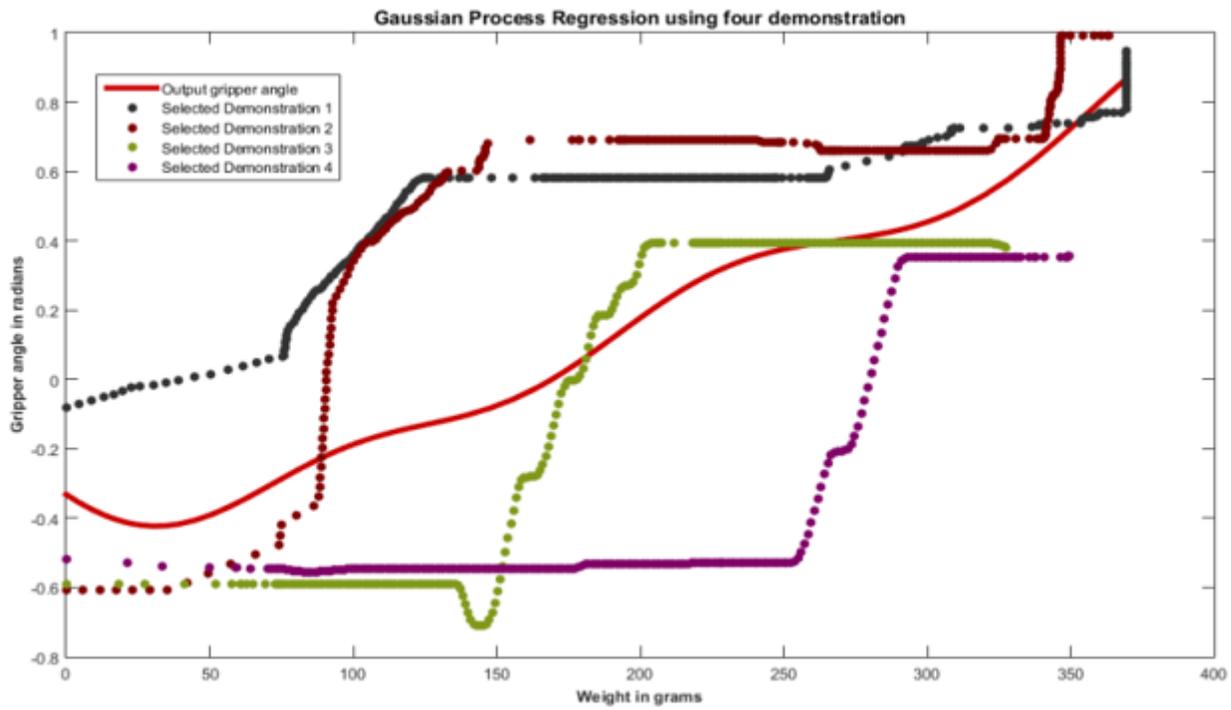


Figure 20

GPR results using four demonstrations