

BACHELOR'S THESIS IN ARTIFICIAL INTELLIGENCE



RADBOUD UNIVERSITY NIJMEGEN

Human-Robot Interaction in the Coordinated Manipulation of an Object

Author:

Loes Habermehl - s4429451
Artificial Intelligence
Radboud University Nijmegen

Supervisor:

dr. ing. L.P.J. Selen
Donders Institute

January 2017

Abstract

Robotic arms become more common in daily life. However, the physical human-robot interaction still needs to be improved. Therefore, I examined which algorithms can improve the interaction in a joint task of a robotic agent and a human, in which they moved an object together. Three algorithms were examined on a robotic manipulandum, namely the first where the robotic agent produced forces towards the human, the second towards the goal and the third towards both the human and the goal. The results show that the algorithm that produced forces towards the goal scores the best in terms of both the trial time and the subjective performance measures (the questionnaire). The algorithm that produced forces towards both the goal and the human scores slightly better in terms of the stability, though in terms of the trial time this algorithm scores the worst. The algorithm that produced forces towards the human scores the worst in terms of both the stability and subjective performance measures. The ranking of the algorithms does not depend on the object size, although a large object (radius 1.5 cm) is easier to manipulate than a small object (radius 0.5 cm), indicated both by the trial time and the subjective performance measures. The ranking of the algorithms does depend on the goal location, which is indicated both by the objective (trial time and stability) and the subjective performance measures. To conclude, the physical interaction between a human and a robotic agent is most effective if the agent produces forces only towards the goal, not towards the human.

Acknowledgements

Firstly, I would like to thank my supervisor Luc Selen for mentoring me during the process of this thesis and for always helping me out. Secondly, I would like to thank Eireen Westland, who I closely worked with during the first stages of this project.

Contents

Abstract	i
Acknowledgements	ii
1 Introduction	1
2 Methods	2
2.1 Experimental setup	2
2.2 Experimental task	2
2.3 Agent algorithms	3
Algorithm 1: Forces towards the human	4
Algorithm 2: Forces towards the goal	4
Algorithm 3: Forces towards both the human and the goal	5
2.4 Task performance	6
Objective performance measures	6
Subjective performance measures	7
3 Results	8
3.1 Trial time	8
Algorithm	8
Object radius	8
Goal location	9
Interaction	10
3.2 Stability	11
Algorithm	12
Object radius	12
Goal location	13
3.3 Subjective performance measures	13
4 Discussion and Conclusions	16
Agent	16
Object radius	18
Goal location	18
Further research	19

A	Code Algorithms	21
A.1	Code algorithm 1: Forces towards the human	21
A.2	Code algorithm 2: Forces towards the goal	22
A.3	Code algorithm 3: Forces towards both the human and the goal	22
B	Questionnaire	25

Chapter 1

Introduction

Robotic arms become increasingly important for aiding humans in daily life. For example, prosthetics become more affordable and versatile and thus more popular and also robotic home assistants can have a manipulator arm which, among other aspects, allows elderly and handicapped people to live longer independently [2].

In order to aid humans, robots need to physically work together with the human. In the ideal situation, the robotic arm would work exactly like a human arm, including the ability to adapt its forces and movements to each specific situation it is in. In that situation, humans could interact intuitively, like interacting with other humans. However, the human-robot interaction can still be improved. Therefore, I researched which algorithms lead to improved human-robot interaction. I focused on the interaction in the coordinated manipulation of an object. More specifically, the joint task of the robotic agent and the subject was to move an object together into a goal, where they both had only one contact point onto the object and only their joint action could result in task completion.

I implemented three different algorithms on a robotic manipulandum, the vBOT [5], to examine which algorithm works best to complete the task. I quantified the performance in human-robot interaction for all three algorithms with seven subjects.

I focused on the robot part, so the idea was that the robotic agent reacted to the movements and forces of the subject in order to move the object in an effective and cooperative way. They had to make sure that the object was as stable as possible.

For my research I adapted a task introduced by Selen et al. (2009) [6], in which subjects had to push against a curved object with a predefined force while at the same time maintaining stability, in other words they needed to avoid slipping of the object.

Chapter 2

Methods

2.1 Experimental setup

Seven healthy, right-handed subjects (ages 18-21 years) participated in the experiment (five male, two female). The subjects were seated in front of a robotic manipulandum, the vBOT [5], which was controlled at 1000 Hz. During the trials, they were holding the handle of the vBOT with their right hand. There was a mirror located horizontally above the handle, in which the subjects could see the task presented via a monitor suspended above. The mirror prevented the subjects from seeing the handle and their own hand. Figure 2.1 shows this mirror from above with the arm of the subject underneath it.



FIGURE 2.1: A subject performing the experiment on the vBOT [5].

2.2 Experimental task

The joint task of the robotic agent and the subject was to move a simulated object (a yellow filled circle with a varying radius) together into a goal (a white filled circle with a radius of 3 cm), where they both had only one contact point. The robotic agent was represented by a red filled circle with a radius of 0.5 cm. The hand position was continuously represented by a blue filled circle with a radius of 0.5 cm as well.

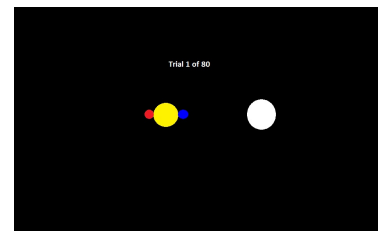


FIGURE 2.2: An example of the task.

The position of the hand representation was equal to the actual position of the subject's hand. A graphical representation of the task is shown in figure 2.2.

There were three different agents implemented, which are explained in more detail in the next section (2.3). Each subject performed 80 trials per agent, thus 240 trials in total. The order in which the three algorithms were presented was randomised, but blocked, and unknown to the subjects. Additionally, the subjects did not know what the differences were between the algorithms.

The subject initiated each trial by moving the representation of his/her hand into the start circle (a white filled circle with a radius of 0.8 cm). Then the goal, the object and the robotic agent became visible. The start position of the object was in the center of the screen (position (0,0)). In each trial, the radius of the object and/or the location of the goal differed. There were two different object radii, namely 0.5 and 1.5 cm. There were eight different goal locations divided equally in a circle around the center of the screen at a distance of 15 centimeters. For the convenience in the following chapters, these goal locations will be referred to by their angle in degrees conform to figure 2.3.

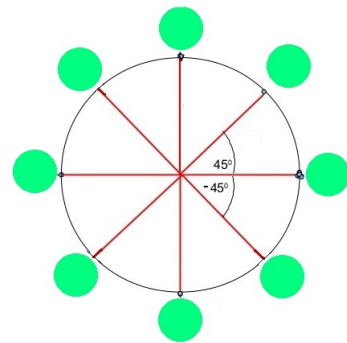


FIGURE 2.3: The eight different goal locations.

The setting of the experiment was an adapted version of the experiment described in Selen et al. (2009) [6]. One of the most significant differences in the task itself was that in this experiment, the object was not rigidly fixed to the world, but it was a simulated mass that moved in reaction to the robotic agent and the human.

2.3 Agent algorithms

The basis of the code for this experiment was the code used for the experiment of Selen et al. (2009) [6]. One of the major differences in the algorithm was that Selen et al. measured the stiffness of the human arm in relation to changes in the stability, while this experiment was focused on the behaviour of the robotic agent.

In this research, there were three different algorithms used to calculate the forces that the robotic agent produced. In the first algorithm, the robotic agent generated forces towards the human. In the second algorithm, the robotic agent generated forces towards the goal. The third algorithm was essentially a combination of the first two algorithms,

that is the robotic agent generated forces towards both the human and the goal. More details about each algorithm are explained next. The complete algorithms can be found in appendix B. In all three algorithms, the position of the robotic agent was the mirrored position of the human relative to the object position, as illustrated in figure 2.2.

Algorithm 1: Forces towards the human

In the first algorithm, the robotic agent produced forces in the direction of the human. This had the consequence that the human only needed to steer the object towards the goal with little effort and the robotic agent delivered the forces for this. These forces were determined as follows:

$$\text{AgentForces}(1,1) = (\text{TotalForce}/(\text{DistanceHuman})) * \text{DirectionXHuman} \quad (1)$$

$$\text{AgentForces}(2,1) = (\text{TotalForce}/(\text{DistanceHuman})) * \text{DirectionYHuman} \quad (2)$$

The "TotalForce" was the total amount of forces that the agent produced, which was set to 15 Newton. This was more than the subject could produce, so that the subject needed less effort, only steering, while the robotic agent delivered all forces needed. The "DistanceHuman" was the distance between the human and the object. This variable determined the direction of the forces, namely towards the human. Since the position of the robotic agent was the mirrored position of the human relative to the object, the distance between the robotic agent and the human would have had the same effect. To distribute the total force between the X and the Y direction, the calculation was multiplied by "DirectionXHuman" and "DirectionYHuman", which represented the distance between the positions of the human and the object, in the direction of respectively the X coordinates and the Y coordinates.

Algorithm 2: Forces towards the goal

The idea of the second algorithm was that the robotic agent produced forces in the direction of the goal, in order to move the object towards it. These forces were calculated in a similar way as the forces towards the human from the previous algorithm. However, the total amount of forces that the agent produced ("TotalForce") was set to 6.5 Newton, instead of the 15 Newton used in the previous algorithm. Moreover, since in this algorithm the direction of the forces was towards the goal instead of towards the human as in algorithm 1, the distance between the object and the goal ("DistanceGoal") was used instead of the distance between the object and the human ("DistanceHuman"). This resulted in the following formulas for the forces of the robotic agent:

$$\text{AgentForces}(1,1) = (\text{TotalForce}/(\text{DistanceGoal})) * \text{DirectionXGoal} \quad (3)$$

$$\text{AgentForces}(2,1) = (\text{TotalForce}/(\text{DistanceGoal})) * \text{DirectionYGoal} \quad (4)$$

To distribute the total force between the X and the Y direction, the calculation was multiplied by "DirectionXGoal" and "DirectionYGoal", which represented the distance between the positions of the goal and the object, in the direction of respectively the X coordinates and the Y coordinates.

Algorithm 3: Forces towards both the human and the goal

The third algorithm was a combination of the previous two algorithms. That is, the forces that the robotic agent produced consisted of the summation of a force in the direction of the human and a force in the direction of the goal. Consequently, the robotic agent both tried to collaborate with the human and to move the object towards the goal.

Force towards the human

The force towards the human was calculated with the same formulas ((1) and (2)) as used in algorithm 1. The only difference in this part of the algorithm was that the force in this algorithm was not set to 15 Newton as it was in algorithm 1. In algorithm 3, the force depended on the object radius:

$$\text{TotalForceHuman} = \text{pickUpThreshold} + \text{ForceDifferenceRadius} * 0.5 \quad (5)$$

The variable "pickUpThreshold" represented a threshold of 2 Newton, which was the minimum force that the subject should produce in order to move the object. The "ForceDifferenceRadius" was equal to the object radius. Therefore, the smaller the object, the less force the robotic agent produced, such that the human needed less effort to move the object. This was necessary, because a smaller object had a higher curvature. If a human then produced much force, chances were bigger that he/she would lose contact with the object.

Force towards the goal

The force towards the goal was calculated in a similar way as in algorithm 2, namely with the following formulas:

$$\text{GoalForces}(1,1) = (\text{GoalForce}/\text{DistanceGoal})*\text{DirectionXGoal} \quad (6)$$

$$\text{GoalForces}(2,1) = (\text{GoalForce}/\text{DistanceGoal})*\text{DirectionYGoal} \quad (7)$$

The "GoalForce" was equal to the "TotalForce" of algorithm 2 (6.5 Newton), which was the total amount of force that the agent produced towards the goal. Also, the variables "DirectionXGoal" and "DirectionYGoal" corresponded to the variables "DirectionXGoal" and "DirectionYGoal" of algorithm 2, which were the distances between the positions of the object and the goal, in the direction of respectively the X- and the Y-coordinates. These were used to distribute the total force between the X and the Y direction. The "DistanceGoal" in this algorithm is the distance between the robotic agent and the goal, while in algorithm 2, the "DistanceGoal" is the distance between the object and the goal. However, both these terms determined the direction of the force, namely towards the goal.

Exceptions

In general, the total force that the robotic agent produced in this algorithm was the summation of the previous explained forces towards the human and the goal. However, the position of the robotic agent was the mirrored position of the human relative to the object position, which had two consequences. Firstly, when the robotic agent was positioned exactly between the object and the goal, i.e. between the human and the goal, it had to produce only forces towards the human, not towards the goal anymore. Otherwise, the sum of the forces towards the human and the goal were below the "pickUpThreshold", which made that the object could not move. Secondly, when the human was positioned between the object and the goal, the total amount of force that the robotic agent produced was increased. This made that the human needed less effort to move the object towards the goal.

2.4 Task performance

To determine the task performance, both objective and subjective performance were measured. The results were validated with a repeated measures anova with the variable "subject" as random factor.

Objective performance measures

Trial time:

The time it took a subject to complete a trial. The time was measured from the moment the subject first touched the object until the object was in the goal.

Stability:

The stability of the object. To measure this, I analysed the movements of the

subject's hand. The shorter this track, the straighter it was, which indicates more stability of the interaction.

Subjective performance measures

The subjective performance measures were determined by means of a questionnaire that the subjects filled in after each session of 80 trials. This questionnaire contained the following items:

Performance:

How do you rate your own performance? (from – to ++)

Agent:

How do you rate this agent? (from – to ++)

Intuition:

How intuitive was the task? (from – to ++)

Effort:

How much effort did you need for the task? (from – to ++)

Object size:

Which object size was the easiest to manipulate? (small / big)

Position:

Which position towards the goal did you prefer? (in front of the object / behind the object / on the side of the object)

Further comments

Chapter 3

Results

3.1 Trial time

Figures 3.1a, b and c show the average trial time of all subjects per condition, which was a combination of a goal (eight different locations, figure 2.3) and a object (radius 0.5 or 1.5 cm), in respectively algorithm 1, 2 and 3. The trial time was defined as the time it took subjects to complete a trial, measured from the moment that subjects first touched the object until the object was in the goal. Figure 3.1d shows the same average trial times of all three algorithms in one plot.

Algorithm

On average, subjects had the fastest trial time (4.0 s) in algorithm 2 (forces towards the goal), the slowest (4.8 s) in algorithm 3 (forces towards both human and goal), which makes the trial time (4.4 s) in algorithm 1 (forces towards the human) in between. The table in figure 3.2b shows these average trial times.

In the repeated measures anova, the p-value of the variable "algorithm" is smaller than 0.5, namely 0.0012 (figure 3.2a), which suggests that the three algorithms differ significantly in trial time.

Object radius

Figure 3.1 shows that in all algorithms, the average trial time is higher when the object radius is smaller. This makes sense, because when the object radius is smaller, the

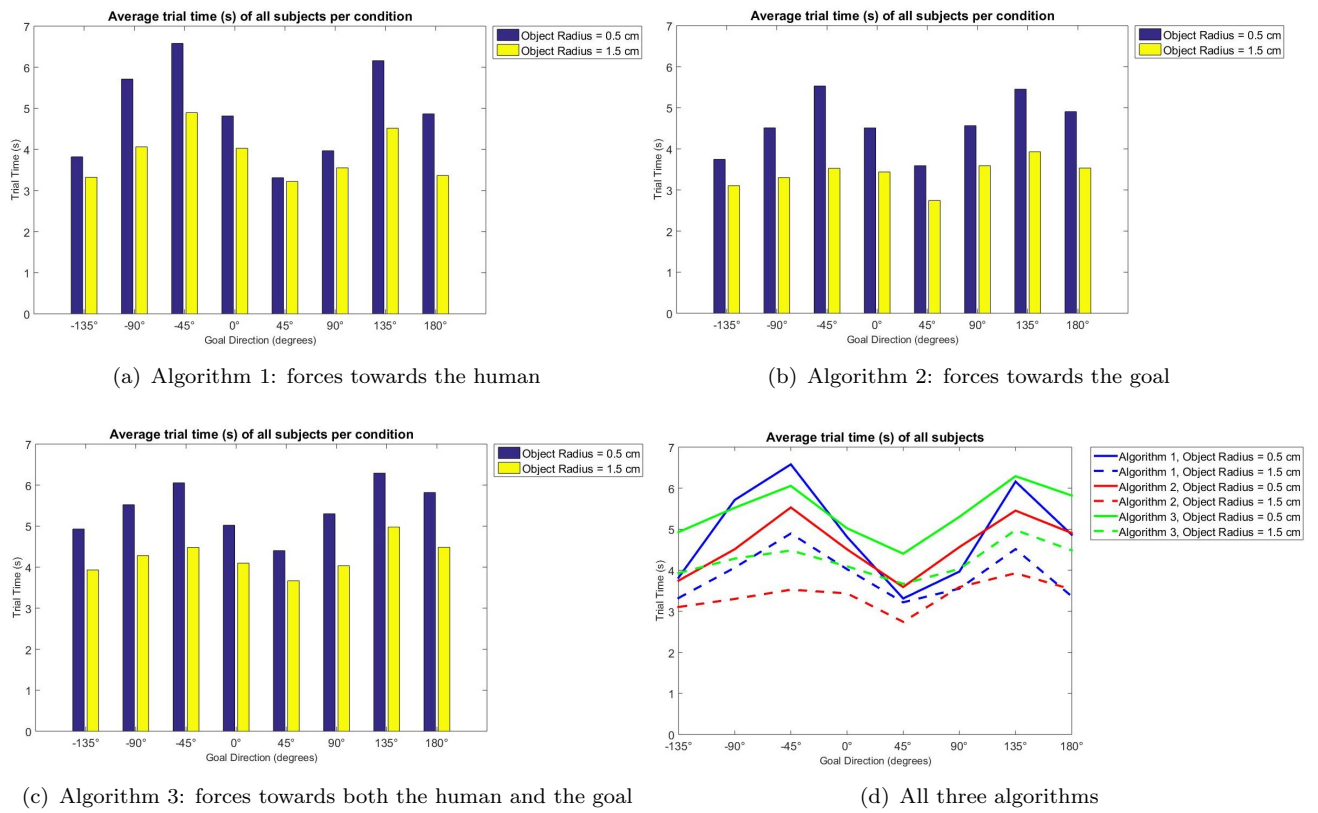


FIGURE 3.1: Average trial time (s) for all subjects per condition (combination of goal location and object radius). (a) Algorithm 1. (b) Algorithm 2. (c) Algorithm 3. (d) All algorithms in one plot. The blue lines represent algorithm 1, the red lines algorithm 2 and the green line algorithm 3. The solid lines represent trial times with object radius = 0.5 cm, the dashed lines trial times with object radius = 1.5 cm.

curvature of the object is higher, which makes it more difficult for subjects to use the right amount of force to push the object without losing contact with it.

The repeated measures anova confirmed these results, because the p-value for the variable "object" is small enough to conclude that the trial times differ significantly for the two different object radii (0.5 and 1.5 cm) ($p = 0.0005$, figure 3.2a).

The results from the questionnaire also confirm this, namely all subjects preferred the "big" object over the "small" object, in all three algorithms (figure 3.4d).

Goal location

Figure 3.1 suggests that the different goal locations have a different influence on the trial time, because the bars in figure 3.1a to c, that represent the average trial times per goal location, are not equally high. Also, the corresponding lines in figure 3.1d would have been horizontal if there was no effect of the goal location.

		p-value	
		Trial Time	Stability
Variables	Subject	0.0044	0.0057
	Algorithm	0.0012	0.0332
	Object	0.0005	0.4148
	Goal	0	0
Interactions	Algorithm*Object	0.3011	0.1031
	Algorithm*Goal	0	0
	Object*Goal	0	0.0589

(a)

	Object radius = 0.5	Object radius = 1.5	Average
Model 1	4,9	3,9	4,4
Model 2	4,6	3,4	4,0
Model 3	5,4	4,2	4,8

(b)

FIGURE 3.2: (a) Results repeated measures anova. (b) Average trial time (s) per algorithm and object radius and average of both object radii per algorithm.

These results are supported by the repeated measures anova, since the p-value of the variable "goal" is 0 (figure 3.2a), which indicates that there is enough evidence to believe that the different goal locations have a significantly different effect on the trial time.

Interaction

Algorithm and Object size

The repeated measures anova gives a non-significant p-value of 0.3011 for the interaction between the variables "algorithm" and "object" (figure 3.2a), which suggests that the differences in trial time between the two object sizes do not depend on which algorithm was used.

Figure 3.1 confirms this result, because in all algorithms, the average trial times are higher for the small object (radius 0.5 cm) than for the big object (radius 1.5 cm). In figures 3.1a to c, this is indicated by the fact that the blue bars are all higher than the yellow bars with the same goal location and in figure 3.1d, this is indicated by the fact that all solid lines are located higher than the dashed lines of the corresponding colour.

Furthermore, the table in figure 3.2b shows that for both object sizes the average trial time of algorithm 2 is the lowest, that of algorithm 3 is the highest and that of algorithm 1 is in between, which corresponds to the previous results.

Algorithm and Goal location

The repeated measures anova calculated a p-value of 0 for the interaction term "algorithm*goal" (figure 3.2a), which indicates that there is significant evidence that the effect of the goal location depends on which algorithm was used. You can see this effect in figure 3.1. For example, in algorithm 1 (figure 3.1a) the goal

location with the highest average trial time is the goal with an angle of -45° (for both object radii), while in algorithm 3 (figure 3.1c) this is the goal with an angle of 135° .

In figure 3.1d both the solid lines cross each other and the dashed lines cross each other, which illustrates that the different goal locations have indeed a different effect on the algorithms. If the lines were parallel, the effect of the goal location would not depend on the algorithm used.

Object size and Goal location

The repeated measures anova implies that there is a significant interaction effect of the object size and the goal location, since the p-value for this interaction term is 0 (figure 3.2a). This means that the effect of the goal location on the trial time depends on the object size.

You can see this effect in figure 3.1, because if there was no interaction between the object size and the goal location, then the difference between the blue bar (object radius 0.5 cm) and the yellow bar (object radius 1.5 cm) in figures a to c would be equal for each goal location. Consequently, in figure d the solid and dashed lines of the same colour would have run in parallel.

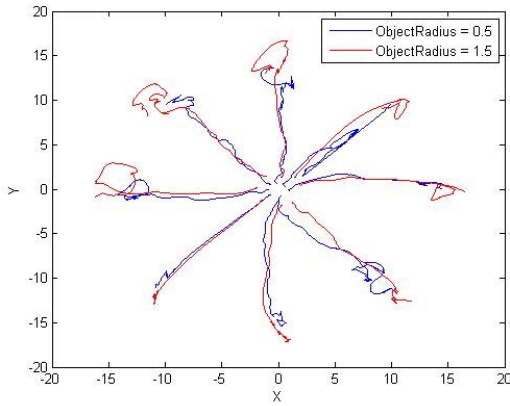
3.2 Stability

The stability of an algorithm was defined as the average length of the track that the hand of the subjects went. The shorter the track was, the straighter the object went from its start position to the goal position. When the object was moved straight to the goal, there was more stability than when the object was moved with a lot of curves. That is because an unstable object requires more compensating of, in this case, both the human and the robotic agent. This effect is illustrated in figure 3.3b and c, which represent two examples of the average hand tracks of subject 1 for two different algorithms to each goal location with each object radius. For this subject, the tracks in algorithm 1 (figure 3.3b) are less straight than those in algorithm 3 (figure 3.3c). This illustrates the longer average length of the hand tracks in algorithm 1 (17.8 cm, figure 3.3a) against those in algorithm 3 (17.3 cm, figure 3.3a).

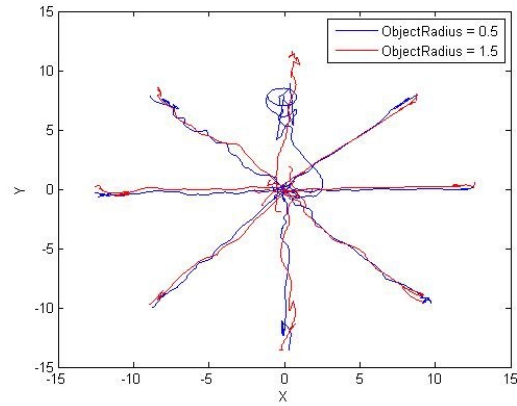
The distance of the start position of the object to the goal was 15 cm for all goal locations, so that would be the length of the hand track if the object would go completely straight to the goal. Figure 3.3a shows a table of the average length of the hand tracks of all subjects and the overall average per algorithm.

	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	Subject 7	Average (cm)
Model 1	17,8	27,5	19,9	21,7	20,7	19,0	20,8	21,0
Model 2	17,1	22,5	20,8	20,0	17,2	18,7	18,6	19,3
Model 3	17,3	21,7	18,6	19,0	16,9	17,6	21,7	19,0

(a) Average length of the hand tracks in cm



(b) Average hand tracks of subject 1 in algorithm 1



(c) Average hand tracks of subject 1 in algorithm 3

FIGURE 3.3: (a) Table of the length of the hand tracks of the subjects in cm. (b) Average track of the hand of subject 1 to all of the goal locations in algorithm 1. (c) Average track of the hand of subject 1 to all of the goal locations in algorithm 3.

Algorithm

Figure 3.3a shows that on average algorithm 3 was most stable, defined as the average length of the hand tracks of all subjects, which was 19.0 cm against 19.3 cm (algorithm 2) and 21.0 cm (algorithm 1). The repeated measures anova indicated that there was indeed a significant difference between the three algorithms, because the p-value was 0.0332 (figure 3.2a).

However, not every subject made the shortest average hand track in algorithm 3. Subjects 1 and 7 made the shortest average hand track in algorithm 2. Though, the differences in track length between algorithm 2 and algorithm 3 were minimal (figure 3.3a).

Object radius

Figures 3.3b and c suggest that trials with a big object (radius 1.5 cm) were more stable than trials with a small object (radius 0.5 cm), because the red lines, indicating the average hand tracks of the subject with an object radius of 1.5 cm, seem slightly straighter than the blue lines, indicating the average hand tracks of the subject with an object radius of 0.5 cm. However, the repeated measures anova indicates that there is no significant evidence that the object radius would have an effect on the stability ($p =$

0.4148, figure 3.2a). There is also no interaction between the algorithm and the object ($p = 0.1031$, figure 3.2a), which indicates that the ranking of the algorithms in terms of stability is not dependent on the object size.

Goal location

The goal location has an effect on the stability, because the average hand tracks are not exactly the same for every goal location (figure 3.3b and c). For all subjects, the average hand tracks seem to indicate that trials with the two goals that were located at the upper right (45°) and the lower left (-135°) corner were most stable, because those hand tracks are the most straight. Multiple subjects confirmed this for algorithm 1 in the questionnaire, including the subject who created the hand tracks of figure 3.3b and c, namely with the following comment:

”It was when I had to move the yellow ball to the upper right corner, because then I had to only turn my elbow, and that made it easier to control the ball and the agent, because instead of turning my shoulder and my elbow together, I only had to use my elbow which made it easier to give subtle feedback.”

The repeated measures anova confirmed that there is a significant effect of the goal location on the stability, because the p-value was 0 (figure 3.2a). The effect of the goal location is dependent on which algorithm was used, which is indicated by the significant interaction between the algorithm and the goal ($p = 0$, figure 3.2a). However, the effect of the goal location is independent on the object radius, because the interaction between the object and the goal was not significant ($p = 0.0589$, figure 3.2a).

3.3 Subjective performance measures

To measure subjectively the performance of the algorithms, each subject filled in the questionnaire that can be found in appendix B. The questionnaire consisted of the same six questions for all three algorithms. The subjects answered one part after each session of 80 trials. The results are shown in figure 3.4. The first four questions were rating questions for the three algorithms. The scores of these questions (figures 3.3a and b) were determined by the value for each possible answer times the frequency of that answer. The values were: ”-” = 1, ”-” = 2, ”-/ +” = 3, ”+” = 4 and ”++” = 5. For the question about the effort subjects needed for the task, the values were set in the opposite direction, thus ”-” = 5 up to and including ”++” = 1. Therefore, the higher the score for each question, the better the rate.

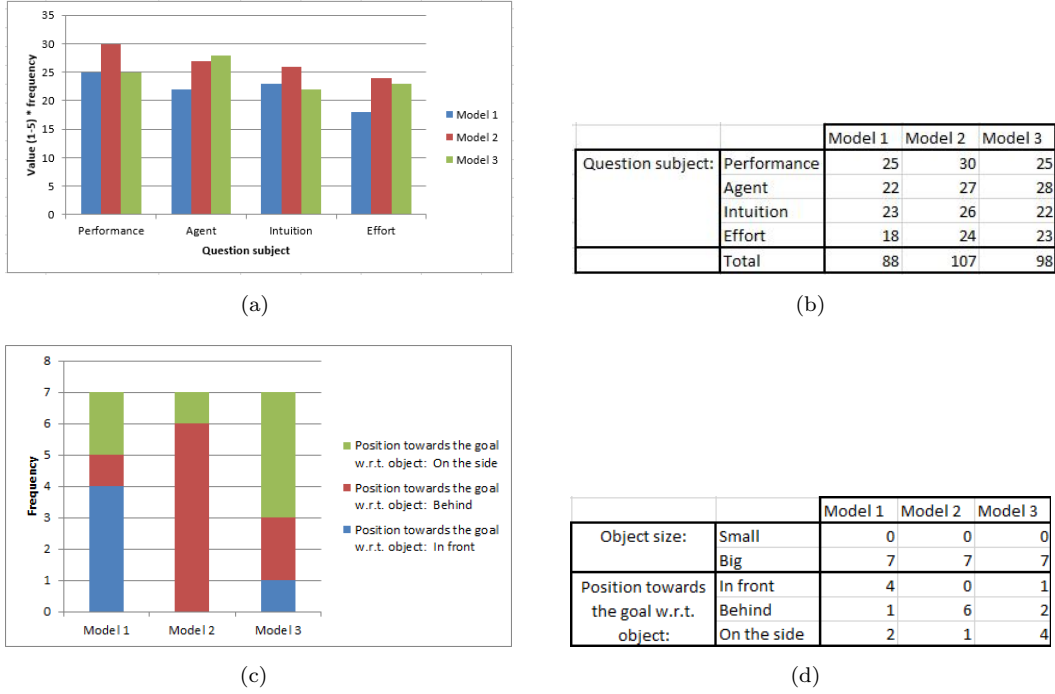


FIGURE 3.4: Results of the subjective performance measures (questionnaire). (a) Frequency of answers on the first 4 questions. The score on the y-axis represents the value for the answers ”-” = 1 up to and including ”++” = 5 times the frequency of those answers. (b) Table of the score (value(1-5) times frequency) that represents the answers on the first 4 questions. (c) Frequency of answers for the question about which position towards the goal with respect to the object subjects preferred. (d) Table of the frequency of answers for the questions about which object size preferred and which position towards the goal with respect to the object subjects preferred.

The last two questions were specific questions about the object size and the position of the human in the task. The scores on those questions (figure 3.4c and d) are equal to the frequencies of the answers. There were seven subjects, so the answers to each question add up to seven for each algorithm.

On average, the subjects rated algorithm 2 the best, in terms of the highest total score of 107, compared to 88 (algorithm 1) and 98 (algorithm 3) (figure 3.4b). However, in the second question in which the subjects rated the agent itself, algorithm 3 scored slightly better than algorithm 2, but that difference is only of value 1. In three out of four questions, algorithm 1 is rated the worst (figure 3.4a and b). The repeated measures anova confirmed that there is a significant difference between the scores of the algorithms, because the p-value was 0.0227.

A result that is unambiguous is that subjects preferred a big object (radius 1.5 cm) over a small one (radius 0.5 cm), as they answered unanimously for each algorithm (figure 3.4d). This is in line with the objective performance measures. Furthermore, the majority of the subjects agreed on which position towards the goal they preferred. Namely, in algorithm 1 most subjects preferred to be in front of the object, in algorithm 2 behind the object

and in algorithm 3 on the side of the object (figure 3.4c and d). An example of the tracks of the human, the agent and the object in algorithm 3 is illustrated in figure 3.5. The object (green line) starts at position (0.0), the human (blue) starts at the start circle, as described in section 2.2, the position of the robotic agent is the mirrored position of the human and for this trial the goal is located at the upper left corner. The spacing between both the agent and the object track and the human and the object track indicate that the human, thus the agent as well, was positioned on the side of the object, except for the last centimeters, where the human moved to the position behind the object. The thickness of the gaps between the tracks show the size of the object, in this case the object was big (radius 1.5 cm).

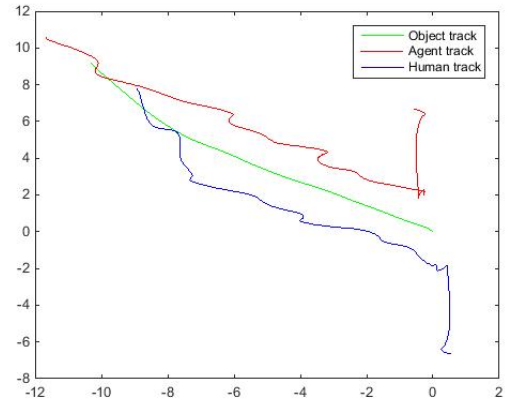


FIGURE 3.5: An example of the tracks of the object, the agent and the human.

Chapter 4

Discussion and Conclusions

In this research, I investigated which algorithms can improve the human-robot interaction in a joint task of a robotic agent and a human, in which they moved an object together. This was examined by implementing three different algorithms on a robotic manipulandum, the vBOT [5]. In the first algorithm, the robotic agent produced forces towards the human. In the second, the robotic agent produced forces towards the goal. And in the third, the robotic agent produced forces towards both the human and the goal. The performance of the algorithms was measured with both objective (trial time and stability) and subjective (questionnaire) performance measures. The second algorithm (forces towards the goal) turns out to be the best algorithm in terms of both the trial time and the questionnaire. The third algorithm (forces towards both the human and the goal) scores slightly better in terms of the stability, which was defined as the shortest average length of the track that the hand of the subjects went. The latter algorithm scores the worst, though, in terms of the trial time. The first algorithm (forces towards the human) appears to be the worst algorithm in terms of both the stability and the subjective performance measures.

Agent

On basis of the reasoning above, for the robotic agent the fastest way to complete the task is to move the object towards the goal, without pushing towards the human (algorithm 2). The subjects experienced this as the best agent to work with, in terms of the highest total score on the questionnaire. The slowest way for the robotic agent is to push towards both the human and the goal (algorithm 3). However, the object is the most stable when the robotic agent does push towards both the human and the goal (algorithm 3). Though, the difference in score between the stability in algorithm 2 and algorithm 3 is minimal, namely a difference in the average hand track of 0.3 cm.

The object is the least stable when the robotic agent pushes only towards the human (algorithm 1) and the subjects experienced this as the worst algorithm as well.

One would expect that a faster trial time corresponds to a shorter hand track, thus more stability, because it seems logical that if the distance between your start and goal position is smaller, your travel time is shorter. However, the ranking of the three algorithms is not the same for the trial time and the stability. As mentioned above, the difference in stability between algorithm 3 (most stable) and algorithm 2 (less stable) is minimal, so maybe algorithm 2 would be the most stable in further research with more subjects, which would then correspond to the fastest trial time.

Still, there would be a difference in the ranking of the worst two algorithms. A possible explanation for this difference between the trial time on the one hand and the stability and questionnaire on the other hand could be that in algorithm 1, subjects were indeed faster than in algorithm 3, however they missed the goal more often by passing it closely. This was possible, because the robotic agent produced only forces towards the human, therefore it made no effort to move the object towards the goal. Moreover, those forces were higher than in the other algorithms. Missing the goal has the consequence that the average length of the hand tracks increases, in other words the magnitude of the stability measures decreases. Additionally, it makes the subjects feel like the goal is harder to reach, which translates into a lower scoring on the questionnaire for this algorithm.

So maybe, the higher velocity of the robotic agent in algorithm 1 outweighs the longer distance of the track. For further research it could be interesting or necessary to rule out the possibility that the total amount of forces of the robotic agent influences the general score of that agent by setting the amount of forces equal for each algorithm. Harris & Wolpert (1998) [4] showed that the velocity of the arm movement slows down if the curvature of the track increases. In other words, the trial time should increase, thus become worse, if the curvature of the track increases, which means that the length of the track increases and therefore the stability decreases. This suggests that the trial time and the stability should indeed be related.

Moreover, I expected that algorithm 3 would score the best, because it is a combination of the previous two algorithms and therefore a combination of the advantages of those two. However, apparently the forces towards the human are not experienced as helpful to move the object together, like those are helpful if you are moving for example a heavy table with two humans. In this task, the forces were experienced as counterproductive, as if the robotic agent was working against the human instead of together with the human. Perhaps, the reason for this is that the task was in a 2D world instead of the 3D real world. A 2D world has the consequence that the object cannot drop. In order to have a good human-robot interaction in real life, this is an aspect that should be taken

into account in further research. Maybe then, the forces towards the human appear to be necessary.

Object radius

The ranking of the three algorithms in terms of the trial time does not depend on the object size, because with both object radii (0.5 and 1.5 cm) algorithm 2 has the lowest trial time, algorithm 3 the highest and algorithm 2 is in between. However, the object size does matter for the trial times itself, because a bigger object (radius 1.5 cm) leads to lower trial times than a smaller object (radius 0.5 cm) does. This is true in all three algorithms. There is not enough evidence though, to state that the object size influences the stability in the algorithms as well. But as expected, the subjects unanimously denoted their preference for a big object in the questionnaire.

Goal location

From the results can be concluded that the ranking of the three algorithms in terms of the trial time depends on the goal location. As mentioned before, averaged over all goal locations, the fastest algorithm is algorithm 2. However, in trials with for example the goal with an angle of 90° , the fastest algorithm is algorithm 1. The goal location has also an effect on the stability, because the average hand tracks of the subjects are not equal for every goal location. And again, this effect of the goal location is dependent on which algorithm is used.

Moreover, the effect of the goal location on the trial time depends on the object size. For example, for the goal with an angle of 45° , algorithm 1 is the fastest if the object radius is 0.5 cm, while algorithm 2 is the fastest if the object radius is 1.5 cm. However, all three algorithms have the fastest trial time to the goal with an angle of 45° , independent of the object size. This concurs with the results of the stability, because the average hand tracks of all subjects indicate that the trials in which the goals had the angle -135° or 45° were most stable.

Several subjects confirmed in the questionnaire that the goal with an angle of 45° , which was for them the upper right corner, was the easiest to reach. This was because for that goal location, they only needed to use their elbow instead of both the elbow and the shoulder. This made the movement more fluent and faster. Especially in algorithm 1, because in that algorithm the robotic agent produced only forces towards the human and those forces had relatively more power than the forces in the other two algorithms. This resulted in a task in which the subject did not need much force to move the object, he/she only needed to steer the object towards the goal. The latter is why the effect of

the goal location was more experienced in algorithm 1 than in the other algorithms. The effect of the goal location could be influenced by noise. Harris & Wolpert (1998) [4] showed that the track that the arm makes, thus also the hand, in a task where a goal needs to be reached with an arm movement, is based on minimizing the variance of the final arm position. This requires a trade-off between the speed and the accuracy of the movement, which is described by Fitt's law [1]. This is forced by signal-dependent noise, which makes that a high level of speed could be counterproductive. That is, if it increases the variance in the end position of the arm, because there are large motor control signals necessary, it could result in missing the goal. However, if the movement to a particular goal location requires less accuracy than the movement to an other goal location, the speed can be increased, which leads to a faster trial time.

Further research

This research can serve as a basis for future research in which a larger sample is used to be able to draw more solid conclusions. The small number of subjects of this experiment is probably the reason that the repeated measures anova suggested the significant result of the variable "subject" ($p = 0.0044$ for the trial time and $p = 0.0057$ for the stability). One would not expect that different subjects would have a different effect on the results. This could also have had an effect on the results from the questionnaire, because the differences between the scores of the algorithms are small and the opinion of one subject has more effect on the results in a small experiment than in a large one. Thus for further research I suggest to do the experiment with more subjects.

Moreover, to further improve the human-robot interaction, specifically in changing task circumstances, it could also be interesting to look at the level of co-contraction, which is the activation of antagonist muscles at the same time, where those muscles are positioned around a joint, like the shoulder or the elbow. Gribble et al. [3] showed that the accuracy of multi-joint arm movements of humans might be smoothed by co-contraction. Maybe it is interesting to examine if this could be used for robotic arms as well.

Bibliography

- [1] Cross, S. H. and Bird, A. P. (1995). Cpg islands and genes. *Current Opinion in Genetics Development*, 5(3)(6):309–314.
- [2] Graf, B., Hans, M., and Schraft, R. D. (2004). Care-o-bot ii development of a next generation robotic home assistant. *Autonomous Robots*, 16(2)(3):193–205.
- [3] Gribble, P. L., Mullin, L. I., Cothros, N., and Mattar, A. (2003). Role of cocontraction in arm movement accuracy. *Journal of Neurophysiology*, 89(5)(5):2396–2405.
- [4] Harris, C. M. and Wolpert, D. M. (1998). Signal-dependent noise determines motor planning. *Nature*, 394(8):780–784.
- [5] Howard, I. S., Ingram, J. N., and Wolpert, D. M. (2009). A modular planar robotic manipulandum with end-point torque control. *J Neurosci Methods*, 181(2)(7):199–211.
- [6] Selen, L. P. J., Franklin, D. W., and Wolpert, D. M. (2009). Impedance control reduces instability that arises from motor noise. *The Journal of Neuroscience : the Official Journal of the Society for Neuroscience*, 29(40)(10):1260612616.

Appendix A

Code Algorithms

A.1 Code algorithm 1: Forces towards the human

```
//Robotic agent produces forces towards the human
void ComputeAgentForcesModel1 ( void)
{
    //RobotBallPosition is the position of the robotic agent
    RobotBallPosition(1,1) = ObjectPosition(1,1)-(HumanPosition(1,1)
        -ObjectPosition(1,1));
    RobotBallPosition(2,1) = ObjectPosition(2,1)-(HumanPosition(2,1)
        -ObjectPosition(2,1));

    double DirectionXHuman = HumanPosition(1,1) - ObjectPosition(1,1);
    double DirectionYHuman = HumanPosition(2,1) - ObjectPosition(2,1);

    //Distance between the human and the object
    double DistanceHuman = sqrt(DirectionXHuman*DirectionXHuman
        + DirectionYHuman*DirectionYHuman);
    double TotalForce = 15;

    AgentForces(1,1) = (TotalForce/(DistanceHuman)) * DirectionXHuman;
    AgentForces(2,1) = (TotalForce/(DistanceHuman)) * DirectionYHuman;
}
```

A.2 Code algorithm 2: Forces towards the goal

```
//Robotic agent produces forces towards the goal
void ComputeAgentForcesModel2 ( void)
{
    //RobotBallPosition is the position of the robotic agent
    RobotBallPosition(1,1) = ObjectPosition(1,1)-(HumanPosition(1,1)
        -ObjectPosition(1,1));
    RobotBallPosition(2,1) = ObjectPosition(2,1)-(HumanPosition(2,1)
        -ObjectPosition(2,1));

    double DirectionXGoal = GoalPosition(1,1) - ObjectPosition(1,1);
    double DirectionYGoal = GoalPosition(2,1) - ObjectPosition(2,1);

    //Distance between the goal and the object
    double DistanceGoal = sqrt(DirectionXGoal*DirectionXGoal
        + DirectionYGoal*DirectionYGoal);
    double TotalForce = 6.5;

    AgentForces(1,1) = (TotalForce/(DistanceGoal)) * DirectionXGoal;
    AgentForces(2,1) = (TotalForce/(DistanceGoal)) * DirectionYGoal;
}
```

A.3 Code algorithm 3: Forces towards both the human and the goal

```
//Robotic agent produces forces towards both the human and the goal
void ComputeAgentForcesModel3 ( void)
{
    //RobotBallPosition is the position of the robotic agent
    RobotBallPosition(1,1) = ObjectPosition(1,1)-(HumanPosition(1,1)
        -ObjectPosition(1,1));
    RobotBallPosition(2,1) = ObjectPosition(2,1)-(HumanPosition(2,1)
        -ObjectPosition(2,1));

    double DirectionXHuman = HumanPosition(1,1) - ObjectPosition(1,1);
    double DirectionYHuman = HumanPosition(2,1) - ObjectPosition(2,1);
```

```

double DirectionXGoal = GoalPosition(1,1) - ObjectPosition(1,1);
double DirectionYGoal = GoalPosition(2,1) - ObjectPosition(2,1);

matrix DistanceAgentGoal(3,1);
DistanceAgentGoal.zeros();
DistanceAgentGoal(1,1) = RobotBallPosition(1,1) -GoalPosition(1,1);
DistanceAgentGoal(2,1) = RobotBallPosition(2,1) -GoalPosition(2,1);

matrix DistanceHumanGoal(3,1);
DistanceHumanGoal.zeros();
DistanceHumanGoal(1,1) = HumanPosition(1,1) -GoalPosition(1,1);
DistanceHumanGoal(2,1) = HumanPosition(2,1) -GoalPosition(2,1);

// To make the force dependent on the objectsize
ForceDifferenceRadius = ObjectRadius;

double DistanceDifference = norm(DistanceAgentGoal)
                          - norm(DistanceHumanGoal);
double DistanceHuman = sqrt(DirectionXHuman*DirectionXHuman
                          + DirectionYHuman*DirectionYHuman);
double TotalForceHuman = pickUpThreshold + ForceDifferenceRadius*0.5;
double GoalForce = 6.5 ;
double DistanceGoal = norm(DistanceAgentGoal);

//Force towards goal
matrix GoalForces(3,1);
GoalForces.zeros();
GoalForces(1,1) = (GoalForce/DistanceGoal)*DirectionXGoal;
GoalForces(2,1) = (GoalForce/DistanceGoal)*DirectionYGoal;

//Forces towards both the human and the goal
GoalForces(1,1) = (TotalForceHuman/(DistanceHuman)) * DirectionXHuman
                + GoalForces(1,1);
GoalForces(2,1) = (TotalForceHuman/(DistanceHuman)) * DirectionYHuman
                + GoalForces(2,1);

// If the total of forces is lower than the pickUpThreshold, namely
// when the agent is between the object and the goal, then the agent

```

```
// should only produce forces towards the human, not towards the
// goal anymore.
if(norm(GoalForces) < pickUpThreshold) //
{
    GoalForces(1,1) = (TotalForceHuman/(DistanceHuman))*DirectionXHuman;
    GoalForces(2,1) = (TotalForceHuman/(DistanceHuman))*DirectionYHuman;
}

// If the human is between the object and the goal, the forces that
// the agent produces are increased.
else if(DistanceDifference > 0)
{
    GoalForces(1,1) += 0.1 * pickUpThreshold * DirectionXHuman;
    GoalForces(2,1) += 0.1 * pickUpThreshold * DirectionYHuman;
}

AgentForces(1,1) = GoalForces(1,1);
AgentForces(2,1) = GoalForces(2,1);
}
```

Appendix B

Questionnaire

Trial session 1

	--	-	-/+	+	++
How do you rate your own performance?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
How do you rate this agent?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
How intuitive was the task?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
How much effort did you need for the task?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Which object size was the easiest to manipulate? small / big

Which position towards the goal did you prefer? in front of the object /
behind the object /
on the side of the object

Further comments:

Trial session 2

	--	-	-/+	+	++
How do you rate your own performance?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
How do you rate this agent?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
How intuitive was the task?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
How much effort did you need for the task?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Which object size was the easiest to manipulate? small / big

Which position towards the goal did you prefer? in front of the object /
behind the object /
on the side of the object

Further comments:

Trial session 3

	--	-	-/+	+	++
How do you rate your own performance?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
How do you rate this agent?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
How intuitive was the task?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
How much effort did you need for the task?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Which object size was the easiest to manipulate? small / big

Which position towards the goal did you prefer? in front of the object /
behind the object /
on the side of the object

Further comments: