



MSc Thesis

Haptic Assistance is a Promising
Method to Support Operators of a
Hunting Simulator for Training Lions

J. Kremer

MSc Thesis

Haptic Assistance is a Promising Method to Support Operators of a Hunting Simulator for Training Lions

by

J. Kremer

Student number: 1311581
Supervisors: Dr. ir. D. A. Abbink, TU Delft
Dr. ir. J. Schuurmans, DotX Control Solutions

Preface

This MSc thesis is the final project of my master in Mechanical Engineering with the track Biomechanical Engineering and specialization Biorobotics, at the Technical University of Delft. During this research I had the incredibly interesting task to try to improve a hunting simulator that is used to train lions and tigers. My goal was to use haptic feedback to support the operator of the hunting simulator, in order to improve their performance. To do so, I developed a virtual version of the hunting simulator, controlled through an old force feedback gaming joystick that I got from Marktplaats. As a result, my neighbors probably think that I'm playing computer games with a joystick all day, week in week out.

I would like to thank my supervisors during this project, Jan Schuurmans and David Abbink, their support was really important for me, especially their enthusiasm, which helped me to stay positive even in times when I was a bit fed up with my research. They helped me to keep up my confidence and moving forwards. Also, I want to thank Bastiaan Petermeijer for his guidance, even though he was technically not one of my supervisors, and Jelle Kalsbeek for his help to get the joystick working.

Furthermore, I would like to thank my parents for their patience and continuous support throughout my study. Above all, I want to thank my girlfriend, Sanne, who helped me get through the most difficult moments during my master and was absolutely crucial in getting me to where I am now.

*J. Kremer
Scheveningen, December 2017*

Contents

1 Paper	6
A The Hunting Simulator	16
A.1 The Lion Foundation	16
A.2 The Hunting Simulator	16
A.3 DotX Control Solutions	17
B The Virtual Hunting Simulator	18
B.1 Physical model	19
B.2 Controller	20
B.3 Joystick	22
B.4 Haptic Assistance	23
B.5 Visualization	29
B.6 GUI	29
C Demo version	33
D Experiment	34
D.1 Routes	34
D.2 Task Instruction	35
E Results	39
E.1 Across participants	39
E.2 Within participants	39
F Informed Consent Form	45
Bibliography	48

1

Paper

Haptic Assistance is a Promising Method to Support Operators of a Hunting Simulator for Training Lions

Jeroen Kremer, David A. Abbink, Jan Schuurmans

Abstract—At the Lion Foundation a hunting simulator is used to stimulate the hunting instincts and skills of lions and tigers, in order to improve their health and increase chances of relocation into the wild. The hunting simulator is an animal enclosure wherein a piece of meat, used as prey, is moved by means of a joystick-controlled cable-driven system. The animal caretakers find it difficult to operate the hunting simulator effectively: the position of the prey relative to the hunting cat and enclosure is hard to judge. Evading the fast unpredictable cats requires high prey velocities without crashing into obstacles or enclosure walls. In this study it is hypothesized that providing the operators with haptic assistance through the joystick will help them to improve their performance, i.e. it will allow them to navigate the prey through the enclosure at higher velocities while having fewer collisions. Haptic assistance was designed as a force-stiffness feedback algorithm based on the inverse Time-to-Collision. It was implemented on a purposefully designed virtual hunting simulator operated through a force-feedback gaming joystick. Naive subjects ($n=10$) participated in an experiment in which they were asked to navigate the virtual prey through a series of waypoints, without hitting the boundaries of the virtual enclosure or the obstacles within it, both with and without the designed haptic assistance. Results show that haptic assistance leads to significant benefits for the operators, as they moved significantly faster while having significantly fewer collisions. In conclusion, the designed haptic assistance is beneficial when used in the virtual hunting simulator, and therefore constitutes a promising method that warrants implementation and evaluation in the real-world hunting simulator.

I. INTRODUCTION

The Lion Foundation shelters large feline predators like lions and tigers that have lived in captivity for most of their lives. Big cats living in captivity are often in bad health, due lack of ways to exercise and release energy. For instance, often observed behaviors like pacing and circling are signs of bad mental health and unreleased energy. These are not natural, but stereotypical for big cats in captivity. Big cats have evolved to hunt and it is an important factor for their physical and mental health. Therefore, the Lion Foundation has developed a hunting simulator to simulate a situation in which the cats have to hunt for their food. With this hunting simulator, they aim to improve the quality of life of the animals by improving their health and increasing their chances of relocation into the wild.

The hunting simulator is an animal enclosure (shown in Figure 1) in which a piece of meat acting as prey, is moved around by a joystick-controlled cable-driven system. The hunting simulator enables an animal caretaker to control prey movement while mimicking the unpredictability of a real-world prey. The cable-driven system uses five cables to move the bait device, to which the meat is attached, through the rectangular enclosure, as shown in Figure 2. Horizontal motion



Fig. 1. The training enclosure at the Lion Foundation, in which the hunting simulator is installed.

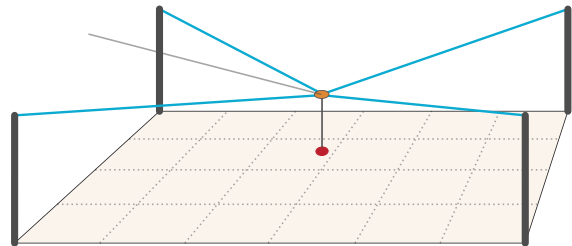


Fig. 2. Schematic representation of the cable-driven system used to control the simulated prey. The four cables (in blue), used to control horizontal motion, run from the corner winches to the bait device (in orange), to which a piece of meat is attached to act as prey (in red). The fifth cable (in grey), governs the height of the prey. During this research, the prey device and bait were combined and modeled as a single point mass. The fifth cable was omitted from the model. A joystick was used to control horizontal movement, while the height reference for the controller was kept constant.

is controlled by four cables, each running from the bait device to one of the winches in each corner of the enclosure. Vertical motion is controlled by the fifth cable, which maintains a constant prey height during simulation.

In its current state, the caretakers find it difficult to operate the hunting simulator effectively. They are not professional operators and they tend to have little affinity with joystick-controlled machinery. Additionally, lions and tigers are fierce hunters and a challenging hunt asks for a fast and agile prey. Consequently, the caretakers have to steer the prey through the enclosure at high velocity, while evading the unpredictable cat requires sudden changes in direction, all without crashing into the enclosure walls or the obstacles within it. Nevertheless, the simulated prey currently moves a lot slower than its real-world counterparts. To really challenge the cats, it is desired to

increase prey velocities even further. Additionally, the system is operated from the side of the enclosure, which makes it more difficult to accurately judge the movements of the prey with respect to the enclosure and the hunting cat. The poor situation awareness, combined with the high velocities required, make the hunting simulator difficult to operate.

This study investigates the application of haptic assistance to support operators of the hunting simulator. The added support should make the hunting simulator easier to operate, thereby improving operator performance. Haptic assistance supports a human operator by means of an automation system, which applies forces to the control interface. This enables the operator to feel the automation its intentions and choose to either accept or resist the provided guidance.

The use of haptic assistance has been investigated in a range of applications including telemanipulation [1], [2], surgery [3], [4], rehabilitation [5], and vehicle control (e.g. cars [6], [7], unmanned aerial vehicles (UAVs) [8]–[10], mobile robots [11], wheelchairs [12], etc). Performance benefits include reduced task completion times (e.g. [1]–[4], [12]–[14]) and increased safety margins (e.g. [11], [15]). Results from several studies show that the application of haptic assistance to collision avoidance led to a reduction in the number of collisions [8]–[12].

Haptic assistance seems like a promising approach to improve the performance of the animal caretakers operating the hunting simulator. The known geometry of the enclosure can be used to implement a haptic collision avoidance system that communicates with the operator by applying forces to the joystick. As a result, the operator is provided with information regarding the prey in relation to its environment, thereby raising the situation awareness of the operator. By helping operators to avoid collisions, the number of collisions is expected to go down. Combined with the increased situation awareness, that should increase the ability and confidence of the operators to move the prey at higher velocities, leading to a more challenging hunt.

The purpose of this research was to show that providing operators of the hunting simulator with haptic assistance is a promising method to improve their performance, i.e. it will allow them to navigate the prey through the enclosure at higher velocities while having fewer collisions. In order to test this, a virtual version of the hunting simulator was developed, operated by means of a force-feedback gaming joystick. Haptic assistance was designed as a force-stiffness feedback algorithm, which provides the operator with feedback based on the inverse Time-to-Collision (iTTC). The designed haptic assistance was implemented on the virtual hunting simulator and an experiment was conducted to evaluate its performance in comparison to manual control. It was hypothesized that, compared to manual control, providing operators of the virtual hunting simulator with haptic assistance would benefit their performance during two different versions of a navigation task, i.e. navigating through a series of waypoints that is either fixed, or contains a sudden route change.

II. METHOD

A. Participants

Ten participants, nine males and one female, aged between 18 and 26 years old ($M = 23$, $SD = 2.7487$), performed the experiment. All participants were right handed and indicated that they had little to no experience with joystick controlled systems. Prior to the experiment, all participants gave their informed consent. Participation was voluntary and participants received no compensation.

B. Apparatus

The experiment was conducted using a Microsoft Sidewinder Force Feedback 2 Joystick, which was connected to a laptop running MATLAB/Simulink to simulate the virtual hunting simulator. Utilizing the haptic feedback capabilities of the joystick requires the use of DirectInput, a legacy Microsoft API that is part of DirectX 9. DirectInput is not natively supported by MATLAB and therefore a MEX-file was written in C++ to handle the communication with the joystick. The virtual hunting simulator was animated using V-REP, which was controlled by MATLAB through an API, an external monitor was used to show the resulting animation to the participant.

The virtual hunting simulator consisted of a 42.8m by 21m enclosure (equal to the real-world equivalent), containing three objects. The cable-driven system was simulated using a three dimensional physical model. The prey device was modeled as a point mass, acted upon by forces representing the four cables running to the corner winches. Prey movement was governed by a controller with the joystick output as velocity reference. A closed-form force distribution algorithm [16] was used to calculate a set of cable tensions that provided the desired movement, while preventing cables from becoming slack (negative cable tension). The joystick was used to control horizontal movement of the prey, while the controller kept the height constant.

C. Haptic Assistance Design

The haptic assistance for the hunting simulator was designed to support the operator by generating two dimensional repulsive forces, guiding the controlled prey away from potential collisions.

One method to do this is to generate these repulsive forces based on the distance from the prey in relation to obstacles and walls. However, these repulsive forces will be generated continually, even when the prey is not moving towards the obstacle. Consequently, it is not a suitable method for the hunting simulator as it limits the operator. For instance, moving the prey parallel to a wall would require the operator to also push towards that wall to counteract the repulsive force generated by the haptic assistance. It is important that operators are able to use as much of the workspace as possible, as a bigger hunting area is more challenging and realistic. To prevent unwanted forces, haptic assistance should only be provided when the prey is moving towards an obstacle. Therefore, the prey velocity has to be taken into account, as well as its position.

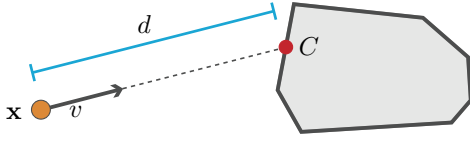


Fig. 3. To calculate the Time-to-Contact (TTC), the distance (d) between prey position (X) and the point of collision (C) is calculated and divided by the absolute velocity of the prey (v)

That could be accomplished by basing the haptic assistance on the Time-to-Collision (TTC), which is the estimated amount of time until a collision occurs, given the current movement. Research considering driver behavior during car following, has shown that drivers use TTC information to avoid collisions [17], [18]. As a result, it is likely that haptic assistance based on TTC would feel quite natural to the operator. Since the operator bases their actions on similar information, reducing the chances of conflict. Applications of TTC based support systems include an alert system [18] and a haptic gas pedal [7] used for car following, as well as haptic collision avoidance in UAVs [8].

For the hunting simulator, the TTC is calculated by taking the distance between the position of the prey x and the point of collision with the obstacle C based on the current movement of the prey (as shown in Figure 3), and dividing that by the absolute velocity of the prey:

$$TTC = \frac{d}{v} \quad (1)$$

where v is the absolute velocity of the prey and d denotes distance between the prey and the predicted point of collision.

The force-feedback applied by the haptic assistance is calculated using the inverse Time-to-Collision (iTTC).

$$\mathbf{F} = \frac{a}{TTC} \hat{\mathbf{n}} \quad (2)$$

with tunable constant a that scales the repulsive force \mathbf{F} , while unit vector $\hat{\mathbf{n}}$ determines its direction.

As a result, the repulsive force becomes larger when the criticality of the situation increases and smaller as the criticality decreases, i.e. an imminent collision will result in a large force.

The direction of the repulsive force is based on the normal w.r.t. the obstacle at the point of collision as shown in Figure 4. With this method, the prey is pushed away from an obstacle without limiting movement parallel to the obstacle.

To prevent discontinuities in the direction of the haptic assistance due to corners in the walls and obstacles, the normals at the corners are adjusted continuously. Near the

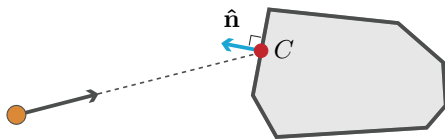


Fig. 4. The direction of the repulsive force \mathbf{F} is determined by the normal to the wall of the obstacle $\hat{\mathbf{n}}$ at the point of collision C .

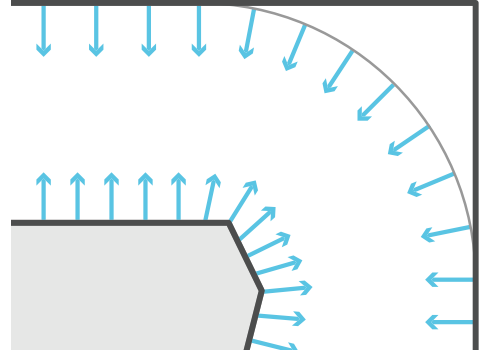


Fig. 5. The direction of the repulsive force near corners of the obstacles and enclosure is adjusted to prevent discontinuities. Near corners of obstacles, normals of both sides adjacent to the corner are linearly combined based on the distance from the corner. In the corners of the enclosure the direction of the repulsive force is based on the normal w.r.t. a circle arc.

corners of obstacles, the force direction is determined by linearly combining the normals of both sides adjacent to the corner, based on the distance between the corner and the predicted point of collision. In the corners of the enclosure, normals to a point of collision with a circle arc were used. Figure 5 shows an example of some these adjusted force directions w.r.t. an obstacle and a corner of the enclosure.

The force-feedback is applied to the joystick by shifting its neutral point. The joystick has a virtual spring, with its default neutral point at the center. By shifting this neutral point, the spring pulls the joystick away from its center in the direction of the shift, thereby applying a force to the joystick.

In addition to the force-feedback, stiffness feedback can be applied by varying the stiffness of the virtual spring. The combination is referred to as force-stiffness feedback. In previous work, haptic assistance systems with force-stiffness feedback have been implemented to support car drivers through the steering wheel [6] and to support operators during remote control of UAVs [10]. Similar to the force-feedback, stiffness feedback is applied to the hunting simulator based on the iTTC, by using it to vary the stiffness of the virtual joystick spring:

$$k = \frac{b}{TTC} \quad (3)$$

where k is the stiffness of the virtual joystick spring, which is scaled by tunable constant b .

Varying the spring stiffness based on the iTTC results in a stiffer spring as the TTC decreases and vice versa. This enables the haptic assistance to apply high forces when needed, while being relaxed during non-critical situations when the operator is not in need of support.

The haptic assistance was tuned on a trial-by-trial basis using pilot experiments.

D. Task Description

Participants were provided with a side-view of the virtual hunting simulator, displayed full screen on an external monitor. They were tasked with navigating the prey device through the virtual enclosure, along a predetermined route that was

indicated by waypoints. Figure 6 shows the provided side view, with two waypoints as an example. During the task, the two upcoming waypoints were shown as bright cylinders within the virtual enclosure, with the first waypoint being more prominent to indicate the order. Upon reaching the first waypoint it would jump to the location of the second waypoint, with the second jumping to that of the next waypoint.

Subjects were instructed to move the virtual prey device through the waypoints without colliding with the walls and obstacles. In addition, they were asked to take the most obvious route between waypoints, i.e. without taking unnecessary detours.

E. Experimental Design

A within subjects repeated measures was used. Each subject operated the virtual hunting simulator under two conditions: manual control (MC) and haptic assisted control (HA). To account for potential learning effects, one half of the participants started with manual control and the other half with haptic assistance.

The routes navigated during the experiment were composed of several different predefined sections, that were combined into a continuous series of waypoints. In order to test both hypotheses simultaneously, two types of section were used:

- 1) sections with an fixed series of waypoints
- 2) sections with a series of waypoints containing a sudden change

Each changing section was based on one of the unchanging ones, i.e. both used the same series of waypoints, while one contained a sudden change. The route change was triggered by the prey coming within a predefined distance from a certain waypoint, causing it and its successor to jump to a new location. An example of an fixed route section and its corresponding suddenly changing variant is shown in Figure 7. Of each section type, three versions were created, making a total of six different sections. By combining these section in varying orders, using two of each, five distinct predefined routes were created. When a collision occurred, a buzzer sounded and the simulator would pause for 5 seconds before

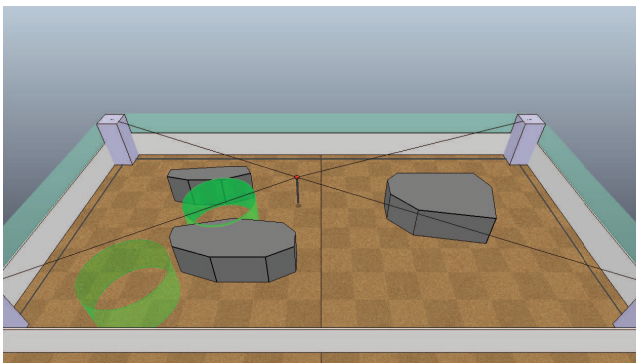


Fig. 6. Side view of the virtual hunting simulator. This view was used during the experiment. The green cylinders indicate the next two upcoming waypoints, of which the most prominent is the first in order.

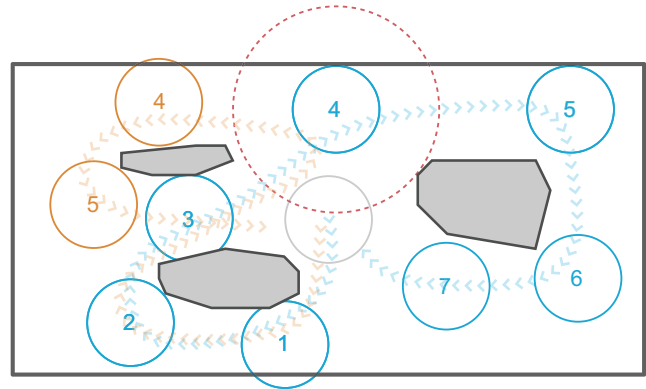


Fig. 7. A top view of the virtual enclosure, with an example of a fixed and a suddenly changing route. The fixed route is indicated by the blue waypoints, which have to be reached in order, starting from the grey waypoint in the center of the enclosure. The suddenly changing route starts off in the same way, but as soon as the red boundary around waypoint four is crossed, the waypoints jump to the new route indicated by the orange waypoints. The arrow lines are examples of possible trajectories for each route.

resetting the prey to the center of the enclosure. Following the timeout, the simulator would resume at the start of the section in which the collision occurred. With the exception of collisions happening after a route change, those would cause the simulator to resume from the subsequent section, otherwise the sudden change would not be very sudden. Each route took approximately 1.5-2.5 minutes to navigate.

F. Procedure

Before the experiment, participants read and signed an informed consent form, after which they filled in a demographics questionnaire. The participants were given some background information about the hunting simulator and were instructed on the navigation task. They were informed on the experiment procedure and in what order they would receive the conditions. Each condition started with a familiarization phase, enabling participants to get a feel for the system without performing the navigation task. After that, they performed two training routes, followed by five routes of which the data was recorded for the experiment. The duration of the experiment was approximately 1 hour per participant.

G. Measured Variables & Metrics

During the experiment, a large number of variables was logged from Simulink, all sampled at 100Hz. For each route segment, the data between the first and the last waypoint were analyzed. For each participant, the mean of the combined results from the segments within each type was used for statistical analysis. Operator performance was measured based on the following metrics:

- **Collision ratio:** The ratio of the mean number of collisions per segment w.r.t. the number of started segments.
- **Average velocity [m/s]:** The mean average velocity during the successfully completed route segments.

In addition to these metrics, the following metrics were also used to gain insight into the effects of the designed haptic assistance:

- **Completion time [s]:** The mean time duration across all successfully completed route segments.
- **Minimum Distance-to-Collision (DTC) [m]:** The mean of the lowest DTCs recorded across successfully completed route segments. The minimum DTC is an indication of the most critical moment during each segment, in terms of distance to a potential collision.
- **Minimum Time-to-Collision (TTC) [s]:** The mean of the lowest TTCs recorded across successfully completed route segments. The minimum TTC is an indication of the most critical moment during each segment, in terms of time to a potential collision.

H. Statistical Analysis

The experiment data was analyzed separately for the two types of route segment (i.e. route segments with or without sudden change). For each participant, the data was averaged across the segments of the same type.

The hypotheses were that operator performance would improve with the support of haptic assistance, which were made a priori and predicted the direction of the effect. Therefore, collision ratios and average velocities were analyzed as planned comparisons using one-tailed paired samples t-tests [19]. Before analysis, a rank transformation was applied to the collision rates, to account for non-normality of the distribution [20]. No hypotheses were made regarding the additional metrics. Those metrics were tested with two-tailed paired samples t-tests.

Each metric was tested twice (i.e. for both route segment types), and thus a Bonferroni correction was applied to control the family-wise error rate at level $\alpha = 0.05$. Therefore, results were regarded as significant for p-values below $\alpha/2$, i.e. for $p \leq 0.025$. For all comparisons, Pearsons r was calculated as an estimation of effect size.

III. RESULTS

The means and standard deviations of the experimental data across participants for both types of route segment are shown in Table I. These means were compared using paired samples t-tests, the results of which are presented in Table II and will be presented in the following sections.

A. Effects of haptic assistance on navigation of fixed routes

The results for operator performance during the navigation of fixed route segments, expressed in collision ratios and average velocities, are shown Figures 8 and 9 respectively. The planned comparisons show that, compared to manual control, the support of haptic assistance led to significant improvements in terms of both performance metrics. It significantly reduced the mean collision ratio by 80% ($t(9) = 3.55, p = 0.0031, r = 0.76$), while average velocities were significantly increased by 16% ($t(9) = -3.68, p = 0.0025, r = 0.78$). For both

TABLE I
EXPERIMENTAL RESULTS, MEAN (SD)

Metrics	Fixed Route		Changing Route	
	MC	HA	MC	HA
Collision Ratio [%]	10.33 (8.70)	2.14 (3.17)	13.23 (8.97)	5.40 (5.29)
Average Velocity [m/s]	8.30 (1.41)	9.64 (1.06)	7.64 (1.33)	9.01 (0.98)
Completion Time [s]	9.91 (1.78)	8.39 (0.95)	6.73 (1.06)	5.87 (0.61)
Minimum DTC [m]	3.12 (0.45)	3.45 (0.30)	2.71 (0.52)	2.92 (0.20)
Minimum TTC [s]	0.41 (0.08)	0.39 (0.07)	0.44 (0.11)	0.41 (0.06)

TABLE II
SUMMARY OF THE STATISTICAL ANALYSIS PERFORMED USING PAIRED SAMPLES T-TESTS

Metrics	Fixed Route	Changing Route
<i>one-tailed</i>		
Collision Ratio [%]	●●	●●
Average Velocity [m/s]	●●	●●
<i>two-tailed</i>		
Completion Time [s]	●	●
Minimum DTC [m]	●	-
Minimum TTC [s]	-	-

A Bonferroni correction was applied to control the family-wise error rate at level $\alpha = 0.5$. Therefore, (●●) and (●) denote significance levels of $p \leq 0.005$ and $p \leq 0.025$, respectively, while (-) indicates that no significant differences were found.

performance metrics Pearsons r is considered large ($r > 0.5$), indicating substantial effects.

The haptic assistance resulted in significantly lower mean completion times for the fixed route segments ($t(9) = 3.58, p = 0.0059, r = 0.77$). Furthermore, the minimum DTC showed a significant increase ($t(9) = -2.82, p = 0.020, r = 0.69$). Both conditions resulted in a similar minimum TTC, no significant differences were found ($t(9) = 0.82, p = 0.43, r = 0.26$).

B. Effects of haptic assistance on navigation of suddenly changing routes

Experimental results for operator performance during navigating the suddenly changing route segments are shown in Figure 10 (collision ratios) and Figure 11 (average velocities). Again, the planned comparisons showed that haptic assistance resulted in significant improvements in operator performance, with a significant 59% reduction in collision ratio ($t(9) = 3.28, p = 0.0048, r = 0.74$) and significantly higher average velocities ($t(9) = -3.80, p = 0.0021, r = 0.78$), representing a 16% increase. Estimated effect sizes for both performance metrics were large ($r > 0.5$) indicating substantial effects.

Meanwhile, significantly lower completion times were found with haptic assistance ($t(9) = 3.21, p = 0.011, r =$

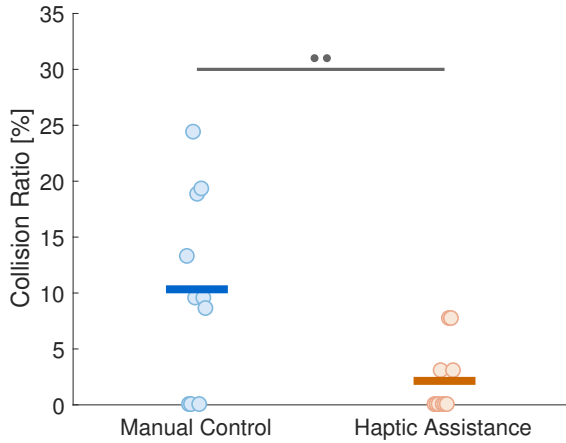


Fig. 8. Mean collision ratios across the three fixed route segments. Each circle represents an individual participant. The horizontal bars denote the means across all participants.

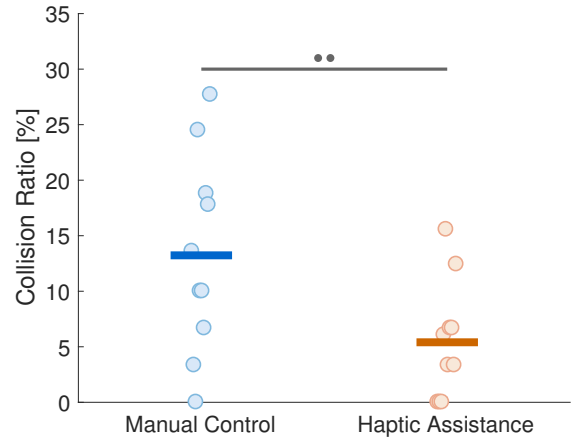


Fig. 10. Mean collision ratios across the three suddenly changing route segments. Each circle represents an individual participant. The horizontal bars denote the means across all participants.

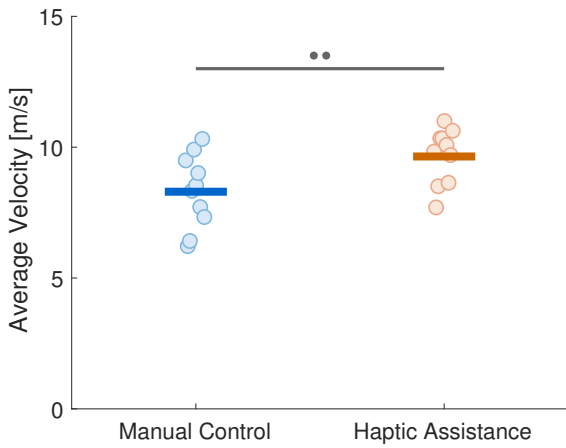


Fig. 9. Average velocities across the three fixed route segments. Each circle represents an individual participant. The horizontal bars denote the means across all participants.

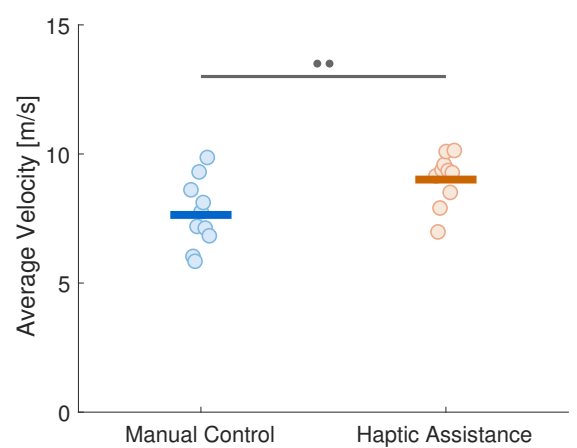


Fig. 11. Average velocities across the three suddenly changing route segments. Each circle represents an individual participant. The horizontal bars denote the means across all participants.

0.73). No significant differences were found between both conditions in terms of the minimum DTC ($t(9) = -1.51, p = 0.16, r = 0.45$) and the minimum TTC ($t(9) = 1.54, p = 0.16, r = 0.46$).

IV. DISCUSSION

The experimental results show that supplying the operators of the virtual hunting simulator with haptic assistance is beneficial to their performance in comparison to manual control.

With haptic assistance, participants were able to perform the navigation task while making far fewer collisions, during both the fixed and the suddenly changing route segments. As a result, collision ratios were significantly lower. This reduction in the amount of collisions due to haptic assistance is in line with other research regarding collision avoidance (e.g. [8]–[12]). In one of these studies [8], a Time-to-Impact (TTI) algorithm is used for 3d haptic collision avoidance on a simulated UAV, in which the force feedback is based on the inverse TTI. This TTI algorithm is similar to the iTTC algorithm implemented in this study, since the TTI and TTC

metrics are equivalent. However, the TTI algorithm does not provide stiffness feedback.

In addition to substantially reducing the number of collisions, haptic assistance also enabled participants to significantly increase their average velocity during both types of route segment. This confirmed the hypothesis that haptic assistance would be beneficial to operator performance. These increased average velocities compared to manual control, are in contrast with several other studies on haptic collision avoidance [8], [9], [11], including the TTI algorithm used in [8]. Although these studies also found a reduction in the number of collisions, their results showed significant decreases in the average velocities. Therefore, their reduced number of collisions might in part have been a result of the lower velocities, rather than the provided assistance. However, the haptic assistance designed in this study managed to successfully improve both the collision ratios and the average velocities, simultaneously. Hence, the reduced amount of collisions is not the result of lower velocities, but a consequence of improved collision avoidance. The haptic assistance did not improve

safety by encouraging operators to move slower, but rather by improving their abilities to avoid collisions, enabling them to go faster while having fewer collisions.

That said, instead of judging performance based on average velocities, many studies use the completion time as a performance metric. The completion time is closely related to the average velocity, but not equivalent, so the completion time was also reported to allow for a comparison. In this study, a significant reduction of completion times was found with haptic assistance, which is in line with a large body of research (e.g. [1]–[4], [12]–[14]), although only [12] focussed specifically on haptic collision avoidance. The lower average velocities resulting from the TTI algorithm studied in [8] did not result in significant differences in terms of completion time.

The results from the additional safety metrics, i.e. the minimum DTC and minimum TTC, indicate that the higher velocities did not lead to significantly reduced safety margins, on the contrary, the only significant change was an increased minimum DTC for the fixed route segments. In conclusion, the experimental results show that haptic assistance would be a promising method to improve the performance of the animal caretakers operating the real hunting simulator, as was hypothesized.

Two alternative methods to improve performance could be either through operator training or full automation of the system. However, to rely on training is not desirable. The operators are animal caretakers that tend to have little affinity with joystick-controlled machinery. They chose their job because they love to take care of animals, not to become professional operators. Besides, a well designed control interface should not rely on highly trained operators in order to be effective, it should be easy to operate. Meanwhile, automating the control of the hunting simulator is really complex, since it entails mimicking the intelligence and unpredictability of realistic prey. Furthermore, evading the hunting cat requires tracking of its movements requiring additional sensors, which is not desired. On the other hand, humans have build-in vision systems suited for lion tracking and they tend to think of themselves as quite intelligent. Therefore it seems a more promising solution to take the best of both worlds and support the operator through haptic assistance.

The iTTC algorithm was chosen because it is a relatively simple algorithm that was likely to feel natural to the operator. The results show that iTTC indeed performed really well. Nevertheless, other methods are available and might perform even better. A limitation of the current iTTC algorithm is, that it only helps to avoid collisions with obstacles that are directly on the predicted path based on its current direction of movement. Operators might be unaware of obstacles that are close to (but just off) the current path. As a result, a small direction change might lead to a sudden large repulsive force from the haptic assistance, possibly too late to avoid a collision.

Potential alternatives are haptic assistance based on a parametric risk field (PRF) [21], [22] or dynamic kinesthetic boundaries (DKB) [23], [24], which are both used for haptic collision avoidance in UAVs. The PRF method creates an

artificial force field around the controlled object, the shape of which changes based on velocity. It provides haptic assistance by generating repulsive forces based on any obstacle entering the force field. The DKB method creates virtual boundaries around obstacles and uses an admittance joystick to prevent operators from crossing them. In [24] the PRF and DKB algorithms are compared to each other, as well as the aforementioned TTI algorithm used in [8]. They concluded that both PRF and TTI performed very similar, although, the PRF algorithm was much more difficult to tune. Furthermore, they found that the DKB method performed better compared to both PRF and TTI, resulting in fewer collisions while increasing the average velocity. However, the DKB method requires a joystick capable of resisting large forces (an admittance joystick was used), and the gaming joystick used in this study is not able to generate the required forces.

The tested iTTC algorithm was tuned on a trial by trial basis and as a result, the tuning is probably not optimal. Although, optimal tuning is difficult to quantify, as it varies between operators. What might feel natural to one operator, might be a constant source of conflict to another. It would be desirable to have a more systematic way to tune the algorithm. Future research on the virtual hunting simulator could be used to study how the different tuning parameters affect performance.

That said, this study shows that the iTTC algorithm already provides a large improvement over manual control. Therefore, it makes sense to implement it on the real hunting simulator, before trying to improve the algorithm or its tuning on the virtual system. Implementation on the real system will show how well the experimental results translate to the real world. Subsequently, adjustments can be made to the virtual system before using it for further research. If it turns out, that it is difficult to find a satisfactory tuning for all animal caretakers on the real-world system, a solution might be to investigate personal tuning. Either the real or virtual hunting simulator could be used to identify which tuning parameters are best suited for personalized tuning.

So all things considered, should implementing haptic assistance be the first step in improving the hunting simulator? Well no, probably not. Currently, prey movement is controlled by a controller that is largely based on heuristics combined with some kinematic equations. Over time, various issues have been solved by adding numerous quick fixes and tricks, resulting in a complex controller that leaves a lot of room for improvement. A model-based controller with an algorithm to calculate the required cable tensions would likely be a large improvement in terms of for instance, speed, reliability, and transparency. Therefore the first step should probably be to implement a new controller. However, to enable the animal caretakers to really challenge the lions and tigers, supplying them with haptic assistance would be a promising next step.

V. CONCLUSION

This research investigated the application of haptic assistance to a hunting simulator, with the goal to improve operator performance, in terms of collision ratios and average velocities. It was hypothesized that the use of haptic shared

control would be beneficial for operator performance, both while navigating a fixed series of waypoints, as well as a series of waypoints containing a sudden route change.

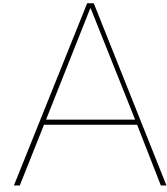
An experiment was performed, in which subjects operated a virtual hunting simulator. For the experimental conditions studied, it can be concluded that:

- As hypothesized, haptic assistance led to significant improvements in operator performance, in comparison with manual control. It resulted in a large and significant reduction in the amount of collisions during navigation of the virtual enclosure, while average velocities were significantly higher, enabling a significant reduction in task completion times.
- Haptic assistance simultaneously improved collision ratios and average velocities, indicating that it helped operators to improve their ability to avoid collisions and that the improvements were not a consequence of encouraging operators to trade speed for safety.
- Safety margins, in terms the minimum TTC and minimum DTC, were quite similar for both conditions. The increased velocities enabled by the haptic assistance did not lead to significantly reduced safety margins, on the other hand, it did yield a significant increase for the minimum DTC during fixed route segments.

It can be concluded that the designed haptic shared control is beneficial to operator performance when applied to the virtual hunting simulator, and therefore constitutes a promising method that warrants implementation and evaluation in the real-world hunting simulator.

REFERENCES

- [1] H. Boessenkool, D. Abbink, C. Heemskerk, and F. Van Der Helm, "Haptic shared control improves tele-operated task performance towards performance in direct control," *2011 IEEE World Haptics Conference, WHC 2011*, pp. 433–438, 2011.
- [2] J. G. W. Wildenbeest, D. A. Abbink, C. J. M. Heemskerk, F. C. T. Van Der Helm, and H. Boessenkool, "The impact of haptic feedback quality on the performance of teleoperated assembly tasks," *IEEE Transactions on Haptics*, vol. 6, no. 2, pp. 242–252, 2013.
- [3] J. J. Abbott, P. Marayong, and A. M. Okamura, "Haptic Virtual Fixtures for Robot-Assisted Manipulation," in *Robotics Research*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 49–64. [Online]. Available: http://link.springer.com/10.1007/978-3-540-48113-3_5
- [4] S. Park, R. D. Howe, and D. F. Torchiana, "Virtual Fixtures for Robotic Cardiac Surgery," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2208, pp. 1419–1420, 2001.
- [5] D. J. Reinkensmeyer, C. T. Pang, J. A. Nessler, and C. C. Painter, "Web-based telerehabilitation for the upper extremity after stroke," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, no. 2, pp. 102–108, 2002.
- [6] D. A. Abbink and M. Mulder, "Exploring the Dimensions of Haptic Feedback Support in Manual Control," *Journal of Computing and Information Science in Engineering*, vol. 9, no. March 2009, p. 011006, 2009.
- [7] M. Mulder, D. A. Abbink, M. M. Van Paassen, and M. Mulder, "Design of a haptic gas pedal for active car-following support," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 268–279, 2011.
- [8] A. M. Brandt and M. B. Colton, "Haptic collision avoidance for a remotely operated quadrotor UAV in indoor environments," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, pp. 2724–2731, 2010.
- [9] T. M. Lam, M. Mulder, and M. M. Van Paassen, "Haptic feedback for UAV tele-operation - Force offset and spring load modification," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1618–1623, 2007.
- [10] T. Lam, M. Mulder, and M. van Paassen, "Haptic interface in UAV tele-operation using force-stiffness feedback," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, no. October. IEEE, oct 2009, pp. 835–840. [Online]. Available: <http://ieeexplore.ieee.org/document/5346913/>
- [11] S. Lee, G. Sukhatme, G. J. Kim, and C.-M. Park, "Haptic Teleoperation of a Mobile Robot: A User Study," *Presence: Teleoperators and Virtual Environments*, vol. 14, no. 3, pp. 345–365, jun 2005. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S147466701733447X> <http://www.mitpressjournals.org/doi/10.1162/105474605323384681>
- [12] A. Hadj-Abdelkader, B. Cherki, and G. Bourhis, "Powered wheelchair driving using a 3D haptic device," *International Conference on Virtual Rehabilitation, ICVR*, pp. 106–114, 2015.
- [13] A. W. De Jonge, J. G. W. Wildenbeest, H. Boessenkool, and D. A. Abbink, "The Effect of Trial-by-Trial Adaptation on Conflicts in Haptic Shared Control for Free-Air Teleoperation Tasks," *IEEE Transactions on Haptics*, vol. 9, no. 1, pp. 111–120, 2016.
- [14] A. Bettini, P. Marayong, S. Lang, A. M. Okamura, and G. D. Hager, "Vision-assisted control for manipulation using virtual fixtures," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 953–966, 2004.
- [15] M. Mulder, D. A. Abbink, and E. R. Boer, "Sharing Control With Haptics," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 54, no. 5, pp. 786–798, 2012. [Online]. Available: <http://journals.sagepub.com/doi/10.1177/0018720812443984>
- [16] A. Pott, T. Bruckmann, and L. Mikelsons, "Closed-form Force Distribution for Parallel Wire Robots," pp. 25–34, 2009. [Online]. Available: [http://www.springerlink.com/content/14m272820122432j/%5CnClosed-form Force Distribution for Parallel Wire Robots.pdf](http://www.springerlink.com/content/14m272820122432j/%5CnClosed-form%20Force%20Distribution%20for%20Parallel%20Wire%20Robots.pdf)
- [17] W. Van Winsum and A. Heino, "Choice of time-headway in car-following and the role of time-to-collision information in braking," *Ergonomics*, vol. 39, no. 4, pp. 579–592, 1996.
- [18] R. J. Kiefer, D. J. Leblanc, and C. A. Flannagan, "Developing an inverse time-to-collision crash alert timing approach based on drivers' last-second braking and steering judgments," *Accident Analysis and Prevention*, vol. 37, no. 2, pp. 295–303, 2005.
- [19] M. B. Castañeda, J. R. Levin, and R. B. Dunham, "Using planned comparisons in management research: A case for the Bonferroni procedure," *Journal of Management*, vol. 19, no. 3, pp. 707–724, 1993.
- [20] W. J. Conover and R. L. Iman, "Rank Transformations as a Bridge Between Parametric and Nonparametric Statistics," *The American Statistician*, vol. 35, no. 3, p. 124, aug 1981. [Online]. Available: <http://www.jstor.org/stable/2683975?origin=crossref>
- [21] H. Boschloo, T. Lam, M. Mulder, and M. van Paassen, "Collision avoidance for a remotely-operated helicopter using haptic feedback," *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, vol. 1, pp. 229–235, 2004. [Online]. Available: <http://ieeexplore.ieee.org/document/1398302/>
- [22] T. M. Lam, H. W. Boschloo, M. Mulder, and M. M. Van Paassen, "Artificial force field for haptic feedback in UAV teleoperation," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 39, no. 6, pp. 1316–1330, 2009.
- [23] Xiaolei Hou and R. Mahony, "Dynamic kinesthetic boundary for haptic teleoperation of aerial robotic vehicles," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, nov 2013, pp. 4549–4950. [Online]. Available: <http://ieeexplore.ieee.org/document/6697070/>
- [24] X. Hou and R. Mahony, "Dynamic kinesthetic boundary for haptic teleoperation of VTOL aerial robots in complex environments," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 5, pp. 694–705, 2016.



The Hunting Simulator

The hunting simulator is a system that is used to train Lions and Tigers. It was developed at the Lion Foundation, in part with help from DotX Control Solutions.

A.1. The Lion Foundation

The Lion Foundation (Stichting Leeuw), in the dutch town of Anna Palowna, is a large shelter for big cats such as lions and tigers. These cats are often rescued from private owners and circuses. Since, as a result of legislation, circuses are no longer allowed to longer allowed to own wild animals acts. Often these cats have lived in captivity for (most of) their whole life. The Lion Foundation seeks to shelter feline predators and, if possible, relocate them into their natural habitat.

Large predators, living in captivity generally get very little exercise, there is simply not much to do in their enclosure except for the occasional stroll along the fence. This is detrimental to the physical and mental health of these animals, as well as being quite boring to watch for visitors. Due to bad health and underdeveloped hunting instincts, rehabilitating big cats in the wild is a challenging task.



Figure A.1: A lion hunting in the hunting simulator

A.2. The Hunting Simulator

Other exercise systems for big cats derived from dog races have been used before. However, feline predators proved to be smart enough to understand the predefined trajectory which the bait follows, they simply lie and wait until the bait passes by and grab it. Therefore the Lion Foundation uses a different approach: a cable-driven parallel manipulator (CDPM) is used to move bait through the training enclosure, operated by an animal carer through a joystick. Controlling the hunting simulator through a joystick prevents the animals from predicting bait movements, making for an interesting chase.

The hunting simulator consists of a 43x21m training enclosure in which the CDPM is installed. The CDPM is used to move a prey device, to which a piece of meat acting as prey is attached, through the enclosure. The enclosure and prey device are shown in Figure C.1. The CDPM consists of four corner winches and one top winch. The four cables connected to the corner winches are used to control the horizontal position of a device to which the bait is attached, the fifth cable is used to control the height of the prey. A schematic representation of the CDPM is shown in Figure A.3.



Figure A.2: The training enclosure where the CDPM is installed (left) and the prey device to which the bait is attached.

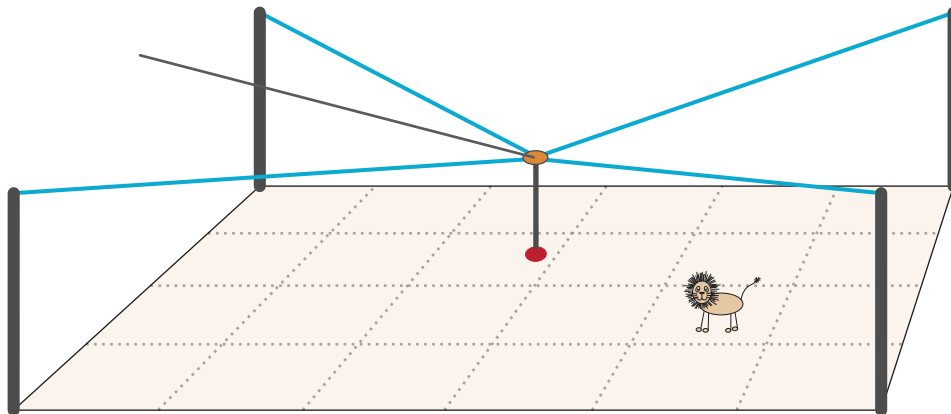


Figure A.3: Schematic representation of the hunting simulator

Currently, the hunting simulator is already used to train animals and it is quite successful, see Figure A.1. However, the current system still leaves some room for improvement. It is difficult to operate and the system is limited in terms of speed and acceleration. As a result, the hunting simulator is not very challenging for good hunters. Therefore, improvements in both the controller and the operation of the system are desired.

A.3. DotX Control Solutions

DotX Control Solutions is a small company that focusses on solving challenging control engineering problems for their costumers. The company was founded by Jan Schuurmans and is based in Haarlem. Besides Jan, the team consists of two engineers one or two interns. Its costumers come from varying fields including wind turbines, process engineering, and mechatronics. Most projects focus on improving and tuning of control algorithms in order to improve their efficiency, generally resulting in lower process costs. DotX became involved with the hunting simulator when they were asked to help to improve the controller of the system.

B

The Virtual Hunting Simulator

The experiments in this study were performed using a virtual hunting simulator, which was designed specifically for the purpose of this research. The virtual hunting simulator is simulated on a computer running Microsoft Windows using MATLAB and Simulink, and it is visualized using V-REP. The virtual hunting simulator consists of a 42.8m by 21m enclosure (equal to the real-world equivalent), containing three objects, a top view of which is shown in Figure B.1. The cable-driven system is simulator using a three dimensional physical model, in which the prey device is modeled as a point mass, acted upon by forces representing the four cables running to the corner winches. Prey movement is governed by a controller with the joystick output as velocity reference.

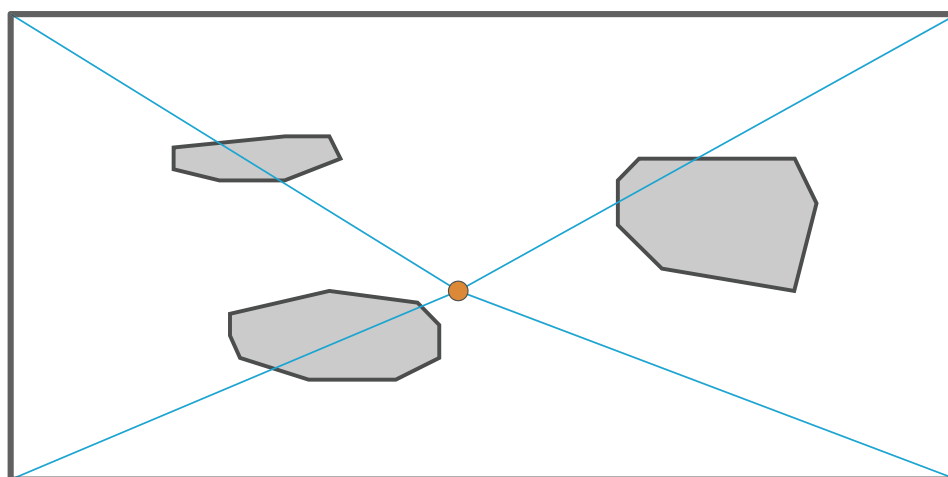


Figure B.1: Top view of the virtual hunting simulator.

The Simulink model of the virtual hunting simulator is shown in Figure B.2, which consists of five main parts:

- **Physical Model:** A simple 3-dimensional physical model of the hunting simulator, used to simulate its behavior
- **Controller:** A two part controller, consisting of a PD feedback controller with an algorithm used to determine the require cable tensions in order obtain the desired movement
- **Joystick:** Communication with the Microsoft Sidewinder Force Feedback 2 joystick, which is used to operate the virtual hunting simulator
- **Haptic Assistance:** The iTTC algorithm which calculates the repulsive forces used to assist the operator, based on the current position and velocity of the prey
- **Visualization:** Communication with V-REP, which is used to visualize the virtual hunting simulator

In addition to these, the 'Real-Time Sync' block is used in order to simulate the system in real-time. The 'experiment controller' supervises the experiment, i.e. it governs the position of the waypoints

and tracks the progress within the route that is being navigated. The aforementioned five main parts of the model will be discussed in the following sections.

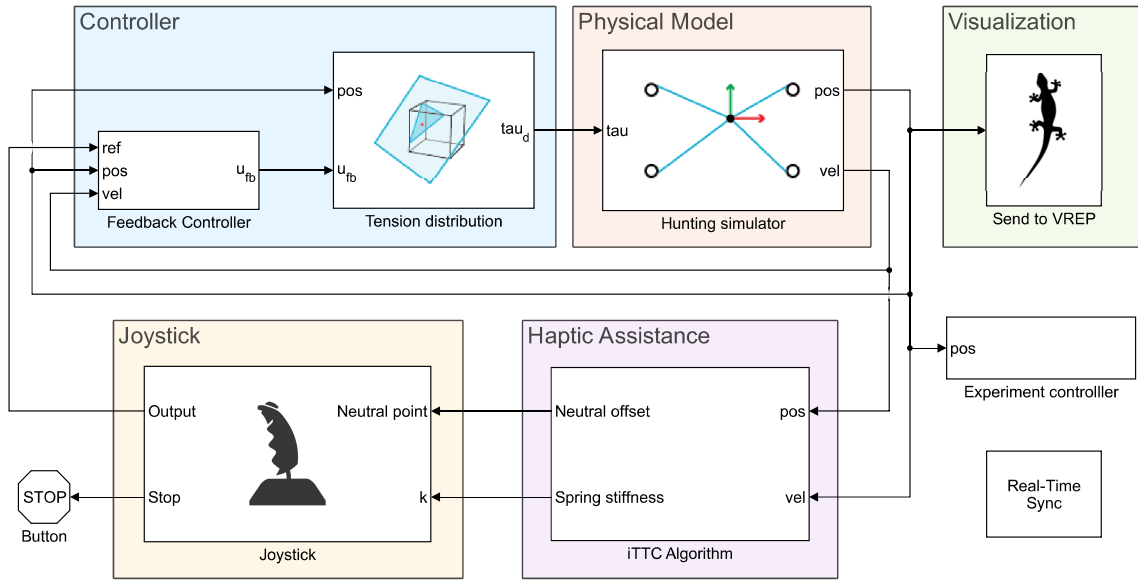


Figure B.2: Simulink model of the virtual hunting simulator

B.1. Physical model

The virtual hunting simulator uses a cable-driven parallel manipulator (CDPM) to control the movement of the simulated prey. The model is based on modeling of similar CDPMs in literature (e.g. [4, 8, 12]). Only translations of the prey are of interest, so rotations are not taken into account. Therefore, the prey device is modeled as a point mass with three degrees of freedom (DOF), to which all the cables are connected. The cables are assumed to be straight and perfectly stiff, while the winches are considered as points. Consequently, the cables can be modeled as vectors pointing from the prey device towards their respective winches. A schematic representation of the model is shown in Figure B.3.

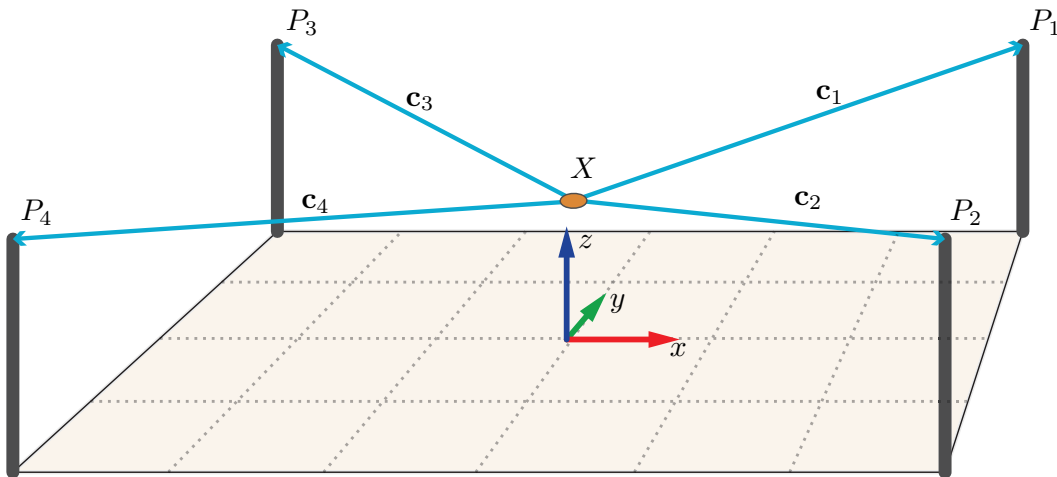


Figure B.3: Schematic representation of the hunting simulator

The vectors representing the cables are calculated using the coordinates of the prey device and the winches:

$$\mathbf{c}_i = P_i - \mathbf{x}, \quad i = 1, 2, 3, 4. \quad (\text{B.1})$$

where \mathbf{c}_i denote the cable vectors, while P_i and X represent the coordinates of the winches and prey device, respectively. These cable vectors are used to determine the force that each cable applies to the prey device, as shown in the free body diagram in Figure B.4.

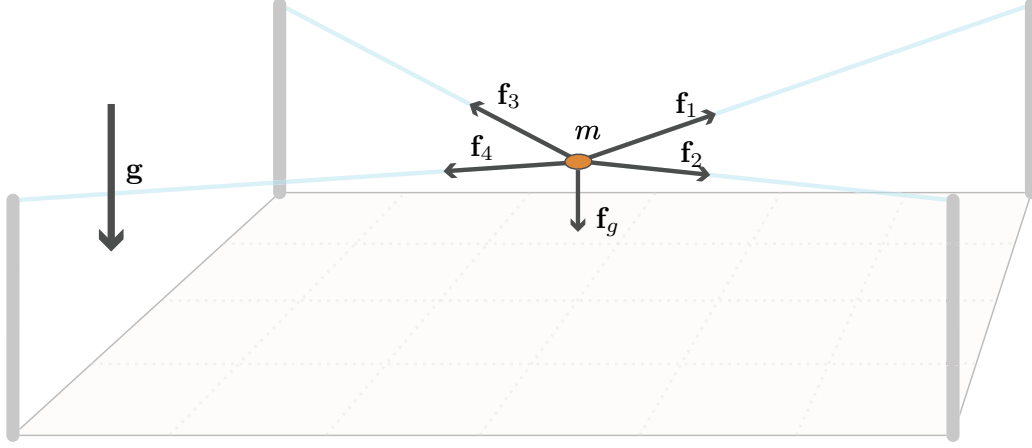


Figure B.4: Schematic representation of the end-effector and the i th cable.

The force \mathbf{f}_i that each of the four cables applies is calculated as follows:

$$\mathbf{f}_i = \tau_i \frac{\mathbf{c}_i}{\|\mathbf{c}_i\|_2} = \tau_i \hat{\mathbf{c}}_i, \quad i = 1, 2, 3, 4 \quad (\text{B.2})$$

where t_i represents the tension in the i th cable and $\hat{\mathbf{c}}_i$ denotes the unit vector along that cable. The cable tension must always be positive, since cables can only pull and not push:

$$\tau_i \geq 0. \quad (\text{B.3})$$

The Newton-Euler equations for the point mass end-effector yield the following dynamics

$$\sum_{i=1}^4 \mathbf{f}_i + \mathbf{f}_g = m\ddot{\mathbf{x}} \quad (\text{B.4})$$

which can be rewritten to

$$\begin{bmatrix} \hat{\mathbf{c}}_1 & \hat{\mathbf{c}}_2 & \hat{\mathbf{c}}_3 & \hat{\mathbf{c}}_4 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} + \mathbf{f}_g = m\ddot{\mathbf{x}} \quad (\text{B.5})$$

and in a more compact form

$$\mathbf{A}\boldsymbol{\tau} + \mathbf{f}_g = m\ddot{\mathbf{x}} \quad (\text{B.6})$$

where the 3×4 matrix \mathbf{A} is called the structure matrix and the 4-dimensional vector $\boldsymbol{\tau}$ contains the four cable tensions. Equation (B.6) is used to simulate the hunting simulator, its implementation into Simulink is shown in Figure B.5. The motion of the virtual prey is control by means of the tension distribution $\boldsymbol{\tau}$, which is determined by the controller.

B.2. Controller

A PD feedback controller is used to control the virtual hunting simulator, of which the output, i.e. the desired resultant force on the prey device, is fed into an algorithm that calculates a suitable tension distribution, which in term is applied to the virtual hunting simulator.

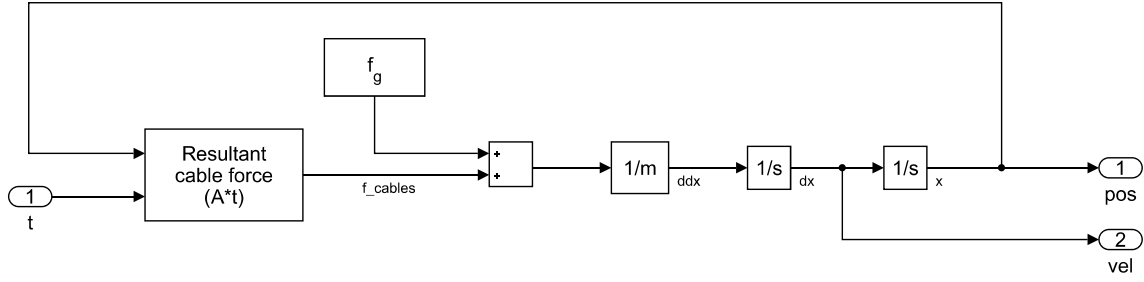


Figure B.5: Implementation of the physical model of the virtual hunting simulator into Simulink

Feedback controller

The motion of the virtual hunting simulator is controlled by a PD feedback controller in Cartesian/workspace coordinates, i.e. it acts on position and velocity errors of the prey device in relation to the enclosure. Workspace coordinates are independent of each other, therefore control in workspace coordinates allows to tune the controller for horizontal and vertical movement separately. As opposed to jointspace control, in which the system would be controlled based on cable lengths, and therefore only allows separate tuning per cable.

The control law for a workspace PD controller is given by:

$$\mathbf{u} = \mathbf{K}_d(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}) + \mathbf{K}_p(\mathbf{x}_d - \mathbf{x}) \quad (\text{B.7})$$

where \mathbf{u} denotes the control output, \mathbf{x}_d is the desired position based on the reference, while \mathbf{K}_p and \mathbf{K}_d represent the proportional and derivative gain matrices, respectively.

In the controller for the hunting simulator, movement in the horizontal plane is based on the joystick output, which is used as a velocity reference. The horizontal position is not of interest (from a control perspective), as there is no reference for the horizontal position within the enclosure. Therefore the proportional gains in the x and y directions are set zero, i.e. the controller does not act on the horizontal position. Meanwhile, the controller does act on errors in vertical position, with a constant height as reference. Preferably there would be no vertical movement at all, so the velocity reference in the z direction is set to zero. Adjusting the control law (B.7) for the virtual hunting simulator yields

$$\mathbf{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} K_{d,x}(\dot{x}_d - \dot{x}) \\ K_{d,y}(\dot{y}_d - \dot{y}) \\ -K_{d,z}\dot{z} + K_{p,z}(z_d - z) \end{bmatrix} \quad (\text{B.8})$$

The resulting control output \mathbf{u} is fed into the aforementioned tension distribution algorithm, to translate control action into cable tensions which can be applied to the virtual hunting simulator.

Tension distribution calculation

The hunting simulator is a system with three degrees of freedom (DOFs), while it is actuated by four cables, i.e. it is a redundantly actuated cable-driven parallel manipulator (CDPM). As a result, finding a cable tension distribution that will achieve a desired movement, is an underdetermined problem, as multiple sets of cable tensions will result in the same resultant force. The number of possible cable distributions is in fact infinite, assuming the end-effector is not in a singular position and cable limits are not exceeded [1, 4, 12]. Furthermore, the tension in each cable should always remain positive to prevent cable from becoming slack, resulting in a loss of control.

In other words, the objective is to find an acceptable solution for the cable tension distribution, which will result in the required motion. That is, solve the dynamics equation (B.6) for $\boldsymbol{\tau}$, while keeping it within bounds. Equation (B.6) can be rewritten as follows:

$$\mathbf{A}\boldsymbol{\tau} + \mathbf{w} = \mathbf{0} \quad (\text{B.9})$$

where the 4-dimensional vector \mathbf{w} contains inertial forces, externally applied forces, and gravity. Determining a suitable tension distribution involves solving equation (B.9) for $\boldsymbol{\tau}$. However, it can not be

solved for $\boldsymbol{\tau}$ uniquely since it is an underdetermined system of linear equations. The structure matrix \mathbf{A} is a non-square (3×4) matrix and thus not invertible. However, if any solution exists, then a particular solution to the problem can be found as follows [5, 7]:

$$\boldsymbol{\tau} = -\mathbf{A}^+\mathbf{w} \quad (\text{B.10})$$

where \mathbf{A}^+ is the Moore-Penrose pseudo-inverse of structure matrix \mathbf{A} , which has the following property:

$$\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A} \quad (\text{B.11})$$

Using a Moore-Penrose pseudo-inverse results in a least-squares solution for the tension distribution [3, 9]. Nevertheless, this is just one of the infinite possible solutions and it is not necessarily an acceptable one, as it might contain negative cable tensions.

A more general solution can be found that is also admitted by this underdetermined linear system [5–7, 11]:

$$\boldsymbol{\tau} = -\mathbf{A}^+\mathbf{w} + (\mathbf{I} - \mathbf{A}^+\mathbf{A})\mathbf{k}, \quad (\text{B.12})$$

where \mathbf{I} is the 3×3 identity matrix and \mathbf{k} is an arbitrary 4-vector.

Equation (B.12) is an adapted version of the resolved motion rate control for kinematically redundant manipulators [12], which consists of a particular and a homogenous part. The first term of (B.12) is the particular solution corresponding to the desired force, and the second term is the homogeneous solution which projects \mathbf{k} into the null space of \mathbf{A} . Any linear combination of the columns of the orthogonal projector $(\mathbf{I} - \mathbf{A}^+\mathbf{A})$ leads to an internal tension without affecting the desired end-effector force and can thus be used to control the pretension in the cables. The vector \mathbf{k} can be used to modulate this pretension such that the resulting tension distribution $\boldsymbol{\tau}$ lies within bounds [2]. In other words, if the particular solution ($\boldsymbol{\tau} = -\mathbf{A}^+\mathbf{w}$) contains cable tensions that are either too high or too low, the homogenous part ($(\mathbf{I} - \mathbf{A}^+\mathbf{A})\mathbf{k}$) can be used to 'shift' the solution to one that is acceptable.

In the virtual hunting simulator \mathbf{k} is determined based on the closed-form method proposed in [10], where \mathbf{k} is chosen as the mean feasible tension distribution $\boldsymbol{\tau}_m$, i.e. for each cable:

$$\tau_{m,i} = \frac{\tau_{min} + \tau_{max}}{2}, \quad i = 1, 2, 3, 4 \quad (\text{B.13})$$

where τ_{min} and τ_{max} denote the minimum and maximum allowed cable tension, respectively.

Inserting $\boldsymbol{\tau}_m$ into \mathbf{k} in equation (B.12) and rewriting yields the closed-form solution that was applied in the virtual hunting simulator:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_m - \mathbf{A}^+(\mathbf{w} + \mathbf{A}\boldsymbol{\tau}_m) \quad (\text{B.14})$$

Which can be used to transform the output from the feedback controller into a suitable cable tension distribution, by inserting the control output into \mathbf{w} :

$$\mathbf{w} = \mathbf{f}_g + \mathbf{u}_{fb} \quad (\text{B.15})$$

where \mathbf{u}_{fb} denotes the output from the feedback controller

It is possible that the tension distribution required to achieve the desired movement does not exist, resulting in a tension distribution that contains cable tensions that are either too high or too low. Therefore tension distribution is checked for tensions that are out of bounds. If that is the case, the cable tension in question is set equal to the bound that it crossed (i.e. minimum or maximum tension).

B.3. Joystick

The joystick used to operate the virtual hunting simulator is a Microsoft Force Feedback 2, which is a haptic joystick designed for computer gaming. This is a joystick that was discontinued in 2003, so a second hand joystick was acquired for this study. Although force-feedback joysticks were fairly popular and widely supported around that time, their popularity have declined since and so has their support. MATLAB and Simulink have limited support for force-feedback joysticks, they can read the outputs and have the ability to control the actuators directly. However, due to slow communication, the joystick can only be actuated at a frequency of about 30-50Hz. As a result, the joystick jitters noticeably, rendering it quite useless for haptic assistance. To provide high quality feedback despite the slow communication, the joystick has an onboard 25MHz processor that can control the actuator at a high frequencies.

When the joystick is controlled via that processor, a command can be repeated until the next command is received, resulting in much smoother behavior. Furthermore, the onboard controller has several in-built haptic feedback effects, e.g. spring, damper, inertia, constant force, etc. To use these effects, the joystick must be controlled through DirectInput, a legacy Microsoft API that was part of DirectX. MATLAB does not offer support for DirectInput, which is why its abilities to control the joystick are limited.

For this research it was important to have the ability to use the high quality haptic feedback and the preprogrammed effects were also very useful. Therefore, a MEX-file was programmed in C++, which allowed MATLAB to communicate with the joystick via the DirectInput API. Using Microsoft Visual Studio the necessary functions and effects were programmed into the MEX-file. MATLAB can use the MEX-file as if it a native function and send it commands, these commands are subsequently used by the MEX-file to call the necessary API functions to control the joystick. The spring and damper effects were build into the MEX-file and used for the virtual hunting simulator.

B.4. Haptic Assistance

The haptic assistance for the hunting simulator was designed to support the operator by generating two dimensional repulsive forces, guiding the controlled prey away from potential collisions. These repulsive forces are based on the inverse Time-to-Collision (iTTC). To calculate the iTTC, first the point of collision (POC) and the distance to collision (DTC) are determined.

Determining the distance to collision

The walls of the enclosure and the obstacles within it are defined through the coordinates of their vertices. Using these coordinates, the unit vectors along the walls of the obstacles and their (unit) normal can be determined, as shown in Figure B.6. The obstacles are convex, with the unit direction of the unit vectors such that they point towards the next vertex in clockwise direction. These unit vectors and normals, as well as the lengths of the sides, are calculated offline and passed to the DTC algorithm.

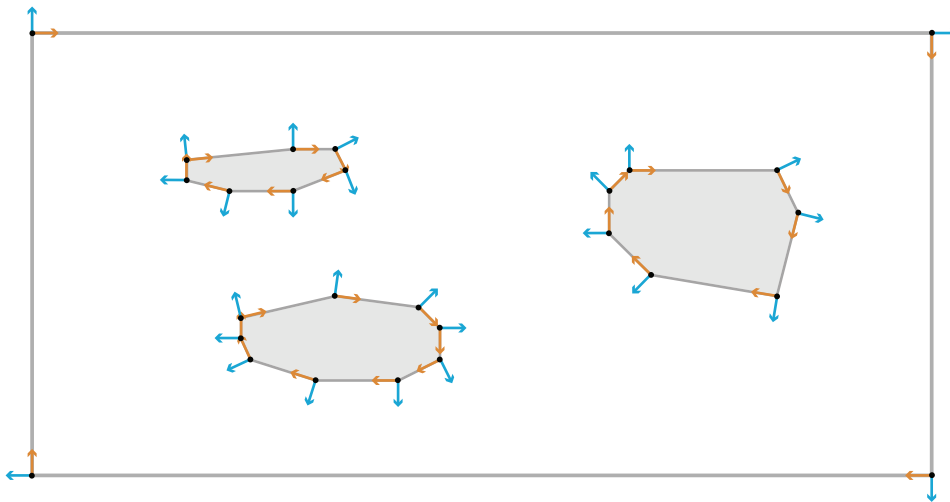


Figure B.6: The walls of the enclosure and the obstacles are defined based on the coordinates of their vertices, which are used to determine the unit vectors along, and the normals to the sides.

The DTC algorithm uses these vectors to find point of collision (POC) of the current prey trajectory. The POC is the closest intersection of the current prey trajectory with the side of an obstacle or a wall. Figure B.7 shows an example of a prey trajectory and its intersections with the lines of which the obstacles are made. In the example the POC is clearly located at intersection 7. Through a process of elimination the POC can be determined algorithmically, which involves the properties of two intersection lines.

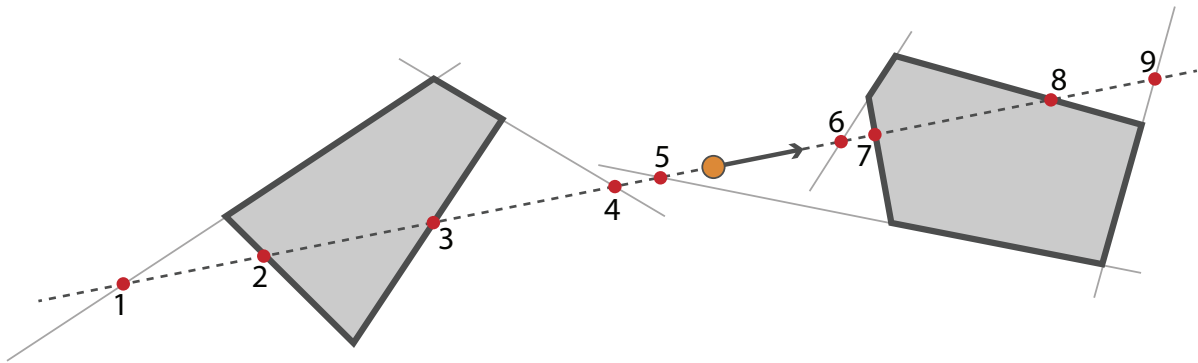


Figure B.7: Example of a prey trajectory intersecting with several obstacle sides.

Intersection between two lines

The intersection of two lines, in this case the prey trajectory with one obstacle side or wall, is illustrated in Figure B.8. To find the POC the algorithm uses the properties determined by the following problem:

Given:

- the current prey position (X)
- the unit vector describing the current prey trajectory (\hat{x})
- the corner of a wall or obstacle (P)
- the unit vector along the wall or side of the obstacle (\hat{p})
- the normal to the wall or side of the obstacle (\hat{n})

Find:

- the point of intersection (C)
- the distance from the prey to intersection (b)
- the distance from the corner to intersection (a)
- the shortest distance from the prey to the wall or side of the enclosure (h)

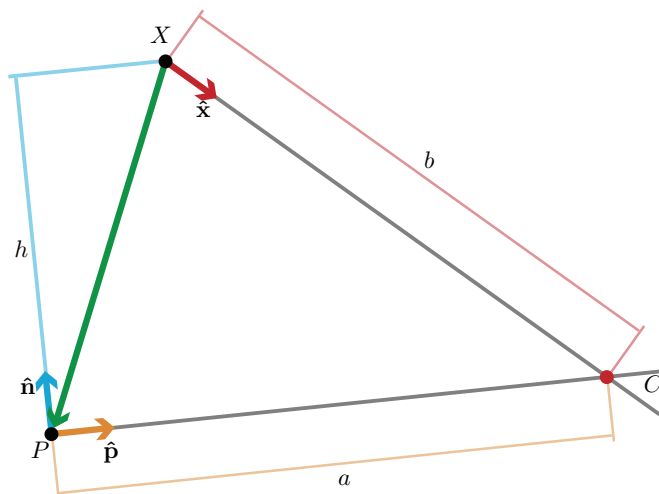


Figure B.8: Intersection of the current prey trajectory with an obstacle/wall vector.

The point of intersection C is given by:

$$C = X + b\hat{\mathbf{x}} = P + a\hat{\mathbf{p}} \quad (\text{B.16})$$

which can be rewritten to

$$b\hat{\mathbf{x}} = (P - X) + a\hat{\mathbf{p}}. \quad (\text{B.17})$$

By multiplying both sides with $\hat{\mathbf{n}}$, the second term on the right side is dropped, since $\hat{\mathbf{n}}$ and $\hat{\mathbf{p}}$ are perpendicular to each other:

$$b\hat{\mathbf{x}} \cdot \hat{\mathbf{n}} = (P - X)\hat{\mathbf{n}} + \underbrace{a\hat{\mathbf{p}} \cdot \hat{\mathbf{n}}}_{=0} \quad (\text{B.18})$$

which can be rewritten to obtain b :

$$b = \frac{(P - X)\hat{\mathbf{n}}}{\hat{\mathbf{x}} \cdot \hat{\mathbf{n}}} \quad (\text{B.19})$$

Negative values for b indicate that the vectors point away from each other and do not intersect in the 'future', i.e. no collision is predicted.

The lengths a and h are determined using scalar projection, which is illustrated in Figure B.9. Given two vectors, \mathbf{j} and \mathbf{k} , where $\hat{\mathbf{k}}$ and $\hat{\mathbf{n}}$ are the unit vector and normal of \mathbf{k} , respectively. The scalar projection of \mathbf{j} onto \mathbf{k} is given by:

$$j_k = \mathbf{j} \cdot \hat{\mathbf{k}} \quad (\text{B.20})$$

while the scalar projection of \mathbf{j} onto $\hat{\mathbf{n}}$ is

$$j_n = \mathbf{j} \cdot \hat{\mathbf{n}} \quad (\text{B.21})$$

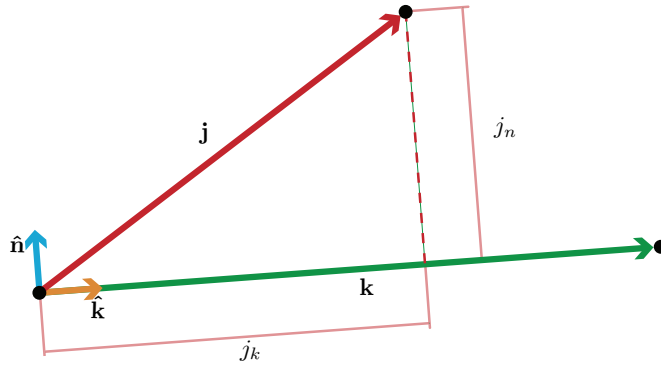


Figure B.9: Schematic representation of the end-effector and the i th cable.

The scalar projection is used because it preserves direction, in contrast to taking the norm. That is, the scalar projection becomes negative depending on the location of the point of intersection in relation to vertex P .

By projecting the vector $C - P$ onto $\hat{\mathbf{p}}$, the length of $C - P$, which is a , can be found, since both vector lie on the same line:

$$a = (C - P) \cdot \hat{\mathbf{p}} \quad (\text{B.22})$$

The values for a are used to determine whether C lies on the side of an obstacle. Given the length of the current side l , the point of intersection C lies on that side if and only if:

$$0 \leq a \leq l \quad (\text{B.23})$$

The smallest distance to intersection b to a point of intersection C located on an obstacle side is Distance-to-Collision (DTC). For that side h is obtained by projecting the vector $X - P$ onto $\hat{\mathbf{n}}$:

$$h = (X - P) \cdot \hat{\mathbf{n}} \quad (\text{B.24})$$

The sign of h indicates whether X lies on the left or right side of $\hat{\mathbf{p}}$. This is used to determine whether the X is located on the inside or outside of the obstacle. That is, a negative value for h indicates that X is located to the right of the obstacle side, which means that the X lies inside of the obstacle. This is a result of the clockwise direction in which the unit vectors of the sides are described.

Finding the point of collision

With the properties of two intersecting lines the POC can be found through a process of elimination. Starting with all lines describing the obstacles, the following steps are used to find the POC and DTC:

1. Check if X is located within the walls of the enclosure, if its on or outside of the walls, a collision has occurred.
2. Calculate b for all lines. If b is negative, that means the POC lies behind X , therefore these lines can be discarded. In the example of Figure B.7 this eliminated intersections 1 to 5
3. For the remaining candidates, calculate the intersections C and use those to calculate a . Discard all lines for which a is either negative or larger than the length l of the corresponding side. In those cases the intersection C does not lie on the obstacle side itself. In the example intersections 6 and 9 are now eliminated.
4. The remaining intersections are guaranteed to lie on one of the sides of an obstacle. The closest of those is the POC, that is the intersection corresponding to the remaining line with the lowest value for b , which in turn is the distance to collision (DTC). For the example this indeed results in intersection 7.
5. For the side on which the POC is located, calculate h . If h is negative X is located to the 'right' of the obstacle side. Due to the direction in which the obstacles are defined, this means that X is 'inside' the obstacle, so a collision has occurred. This situation is illustrated in Figure B.10.

If during any of these steps all candidates were eliminated, then the POC does not lie on an obstacle, so the POC must lie on a wall. Calculate b for the enclosure walls, the smallest positive value corresponds to the DTC, use that to calculate the POC (C).

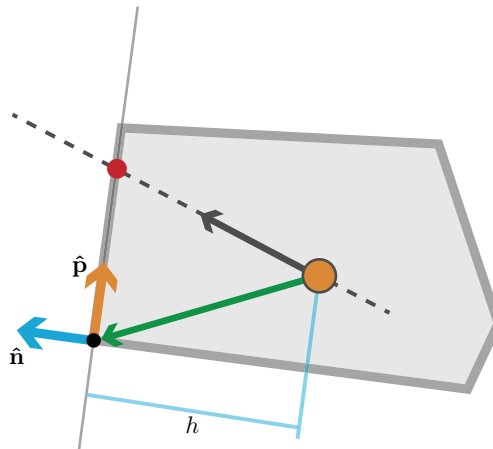


Figure B.10: Example of a situation in which X is located inside an obstacle

Haptic assistance based on the inverse Time-to-Collision

The Time-to-Collision (TTC) is calculated by dividing the Distance-to-Collision (DTC) by the absolute velocity of the prey:

$$TTC = \frac{DTC}{\|\dot{\mathbf{x}}\|} \quad (\text{B.25})$$

The force-feedback applied by the haptic assistance is calculated using the inverse Time-to-Collision (iTTC):

$$\mathbf{F} = \frac{a}{TTC} \hat{\mathbf{n}} \quad (\text{B.26})$$

with tunable constant a that scales the repulsive force \mathbf{F} , while unit vector $\hat{\mathbf{n}}$ determines its direction. As a result, the repulsive force becomes larger when the criticality of the situation increases and smaller as the criticality decreases, i.e. an imminent collision will result in a large force. The direction of the repulsive force is based on the normal w.r.t. the obstacle at the point of collision as shown in Figure B.11. With this method, the prey is pushed away from an obstacle without limiting movement parallel to the obstacle.

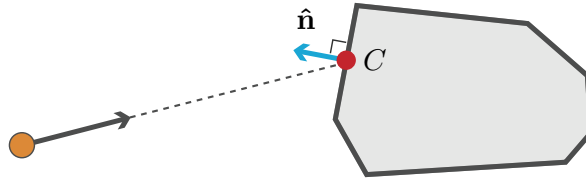


Figure B.11: The direction of the repulsive force is determined by the normal to the wall of the obstacle $\hat{\mathbf{n}}$ at the intersection C .

Preventing discontinuities in corners

Near the corners of obstacles and walls, discontinuities will occur in the direction of the normal used to determine the direction of the feedback. When a corner is passed, the normal suddenly jumps from one side/wall to the next. This is especially problematic in the corners of the enclosure, since the two normals on each side of the corner will push the prey towards the other side of the corner. This could lead to an oscillation ending up in the corner itself. To prevent discontinuities in the direction haptic assistance due to corners in the walls and obstacles, the normals at the corners are adjusted continuously, as shown in Figure B.12. Near the corners of obstacles, the force direction is determined by linearly combining the normals of both sides adjacent to the corner. The combination is based on the distance between the corner and the point of collision, if that distance is below a certain threshold the vectors are combined. This is done using the following equation:

$$\mathbf{n}_{nc} = \frac{1+d}{s} \hat{\mathbf{n}}_1 + \frac{1-d}{s} \hat{\mathbf{n}}_2 \quad (\text{B.27})$$

where d denotes the distance to the corner and s is the threshold distance. The normal w.r.t the POC is $\hat{\mathbf{n}}_1$, while the normal of the side around the corner is denoted by $\hat{\mathbf{n}}_2$.

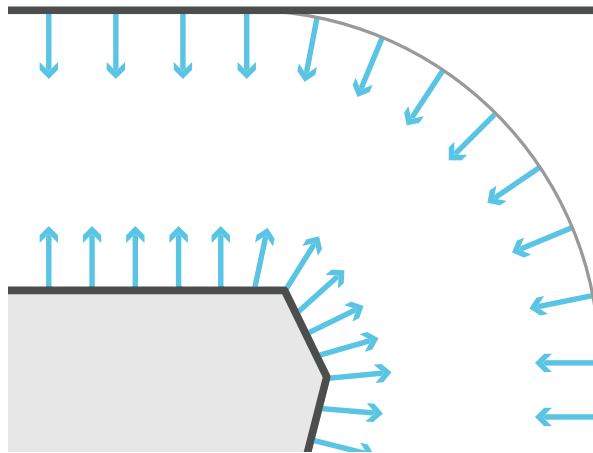


Figure B.12: The direction of the repulsive force near corners of the obstacles and enclosure is adjusted to prevent discontinuities. Near corners of obstacles, normals of both sides adjacent to the corner are linearly combined based on the distance from the corner. In the corners of the enclosure the direction of the repulsive force is based on the normal w.r.t. a circle arc.

In the corners of the enclosure, normals to a point of collision with a circle arc were used. The normals w.r.t. a circle arc were chosen because the felt better compared to the linear combination when applied to the corner of the enclosure. An example of how these normals are determined is illustrated in Figure B.13, it involves solving the following problem:

Given:

- the current prey position (X)
- the unit vector describing the current prey trajectory ($\hat{\mathbf{x}}$)
- the center of the circle arc (M)
- the radius of the circle arc (c)
- the unit vector pointing from the prey towards the center of the arc ($\hat{\mathbf{m}}$)
- the distance from the prey to the center of the circle (a)

Find:

- the point of intersection on the circle (C)
- the distance the prey to intersection with the circle (b)

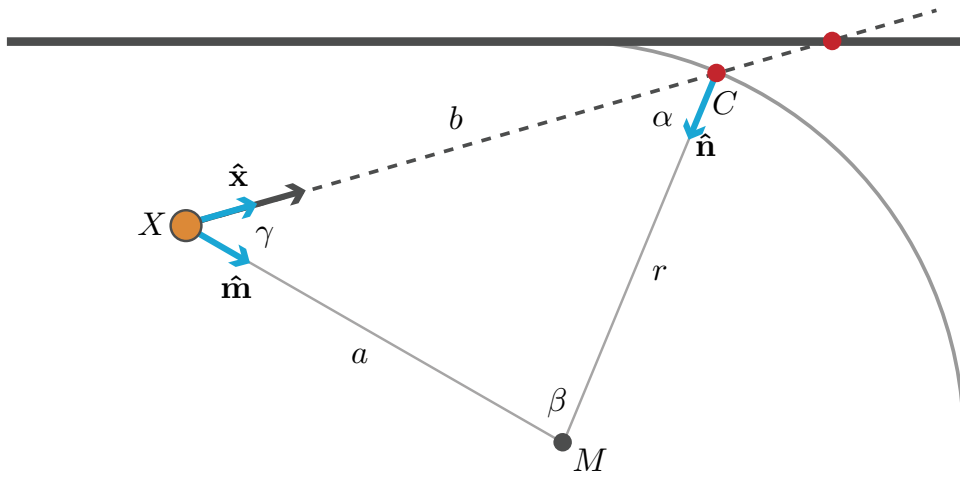


Figure B.13: Determination the normal w.r.t. a circle arc in a corner of the enclosure.

The first step is to calculate the angles, starting with γ , the cosine of which is given by

$$\cos \gamma = \hat{\mathbf{m}} \cdot \hat{\mathbf{x}} \quad (\text{B.28})$$

and therefore,

$$\gamma = \cos^{-1}(\hat{\mathbf{m}} \cdot \hat{\mathbf{x}}). \quad (\text{B.29})$$

The other angles are calculated using the law of sines:

$$\frac{a}{\sin \alpha} = \frac{b}{\sin \beta} = \frac{r}{\sin \gamma} \quad (\text{B.30})$$

that yields the following equation for α

$$\alpha = \sin^{-1} \left(\frac{a - \sin \gamma}{r} \right) \quad (\text{B.31})$$

which is used to find β

$$\beta = \pi - (\alpha + \gamma). \quad (\text{B.32})$$

With these angles the distance to the intersection b is determined by

$$b = \frac{r \sin \beta}{\sin \gamma} \quad (\text{B.33})$$

resulting in the point of intersection C itself:

$$C = X + b\hat{\mathbf{x}}. \quad (\text{B.34})$$

The unit normal to the point of intersection is now calculated by:

$$\hat{\mathbf{n}} = \frac{M - C}{\|M - C\|}. \quad (\text{B.35})$$

Distance-to-Collision algorithm

The calculations in the sections above are used in an algorithm to find the Distance-to-Collision (DTC), direction of the repulsive force $\hat{\mathbf{n}}$ and to check if a collision has already happened. These are used to determine the TTC (Equation B.25) and the repulsive force \mathbf{F} (Equation B.26). The pseudocode of the simplified algorithm is shown in Algorithm 1.

B.5. Visualization

The virtual hunting simulator is animated using VREP, which is a robot simulator developed by Coppelia Robotics. Although VREP has an built-in physics engine and a quite extensive set of features, during this study it was only used for the purpose of animation. VREP has an API which allows it to communicate with MATLAB. This API was used to move the prey through the virtual enclosure according to the simulation performed in Simulink. Figures B.14 and B.15 show, respectively, the top and side view of the animated enclosure. The virtual enclosure was built in VREP using the simple 3d modeling tools available in the program. The obstacles were defined in MATLAB to provide the flexibility to change their shape and location without redefining the 3d model. VREP has the ability to write scripts and functions in LUA, which was used to create additional functionality to add to the existing API. A function was added that allowed MATLAB to draw the obstacles by building them out of triangles. In addition, a function was added that allowed MATLAB to provide participants with on screen information, such as count-downs. Furthermore, the 'experiment controller' block communicated with the API to move the waypoints when necessary.

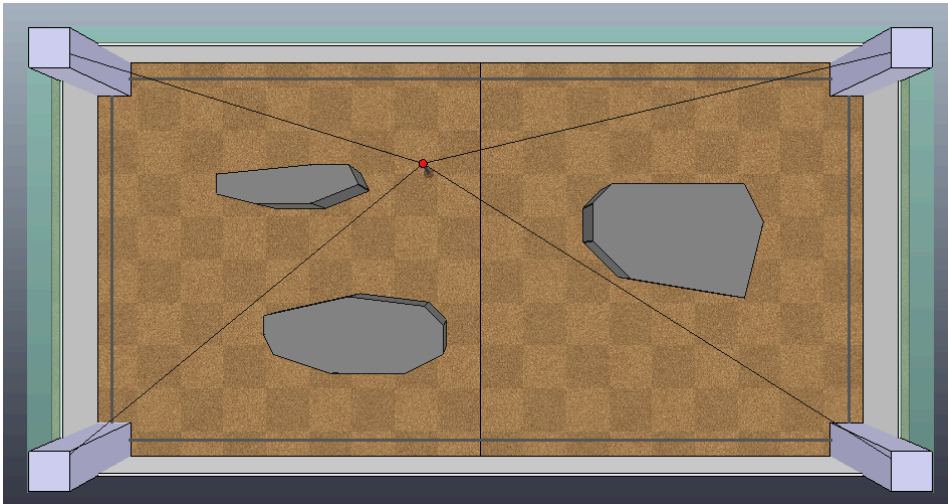


Figure B.14: A top view of the virtual hunting simulator animated using VREP

B.6. GUI

To control the experiment several GUIs were used, two of which will be described briefly, one that was used to control the experiment itself and another for the setup of the experiment.

Algorithm 1 DTC calculation

```

function distance2contact
  DTC ← 0                                ▷ distance to collision
  direction ← 0                          ▷ direction of repulsive force
  collision ← 0                          ▷ has a collision occurred?

  if X is inside enclosure then
    for i ← 1 do
      Q ← [1 : n]                       ▷ all potential obstacle intersections
      calculate b(Q) with Eq B.19
      Q ← Q(b(Q) > 0)                  ▷ lines with future intersection
      if Q is empty then
        break                             ▷ no future collision with obstacle
      end if
      calculate C(Q) with Eq B.16
      calculate a(Q) with Eq B.22
      Q ← Q(0 ≤ a(Q) < l(Q))        ▷ vectors with intersection on obstacle side
      if Q is empty then
        break                             ▷ no future collision with obstacle
      end if
      ID ← Q(min(b(Q)))                 ▷ ID of side with closest collision
      DTC ← b(ID)                       ▷ distance to closest collision
      calculate h(ID) with Eq B.24
      if h(POC) ≤ 0 then
        collision ← 1                    ▷ a collision has occurred
      end if
      if d(ID) < s(ID) then
        calculate  $\hat{\mathbf{n}}_{nc}$ (ID) with Eq B.27
        direction ←  $\hat{\mathbf{n}}_{nc}$ (ID)
      else
        direction ←  $\hat{\mathbf{n}}$ (ID)           ▷ POC is not near corner
      end if
    end for

    if ID is empty then
      Q ← [1 : 4]                         ▷ collision must be with wall
      calculate b(Q) with Eq B.19
      Q ← Q(b(Q) > 0)
      ID ← Q(min(b(Q)))
      DTC ← b(ID)
      if d(ID) < r then
        calculate b(ID) with Eq B.28-B.33
        calculate C(ID) with Eq B.34
        calculate  $\hat{\mathbf{r}}$ (ID) with Eq B.35
        direction ←  $\hat{\mathbf{r}}$ (ID)
      else
        direction ←  $\hat{\mathbf{n}}$ (ID)           ▷ POC is not near corner
      end if
    end if

    else
      collision ← 1                       ▷ a collision has occurred
    end if

  return DTC, direction, collision
end function

```

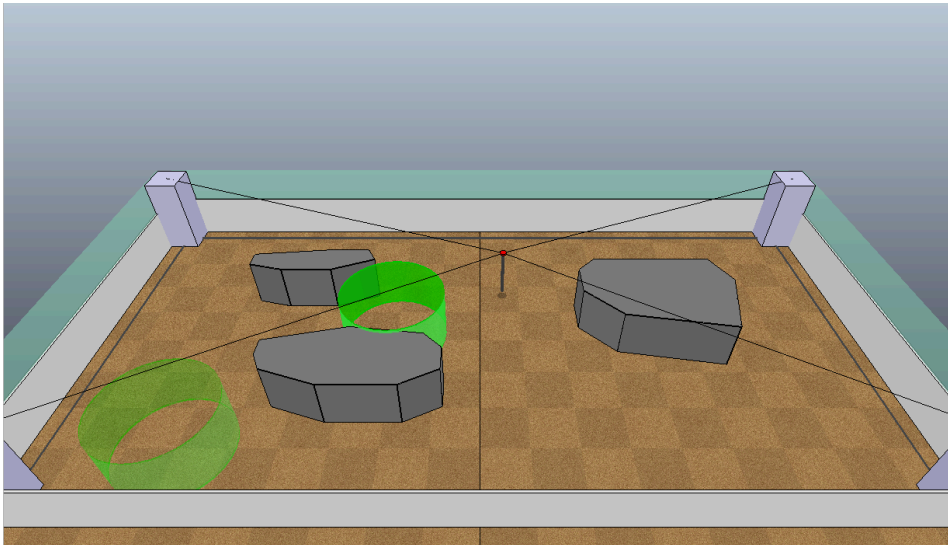


Figure B.15: A side view of the virtual hunting simulator animated using VREP, the green cylinders indicate the waypoints.

Experiment GUI

The experiment GUI was used to control the simulator during the experiment itself, it is shown in Figure B.16. The main purpose of this GUI was to prevent mistakes during the experiment, mainly through taking care of things that are easy to forget, like changing some variable in a setup file. Therefore it kept track of progress during the experiment and only allowed the experiment to start when a subject number was filled in and confirmed, and while VREP was running. Furthermore, it could reload the last used datafile, which enabled quick resets after a crash or restart, without any mistakes.

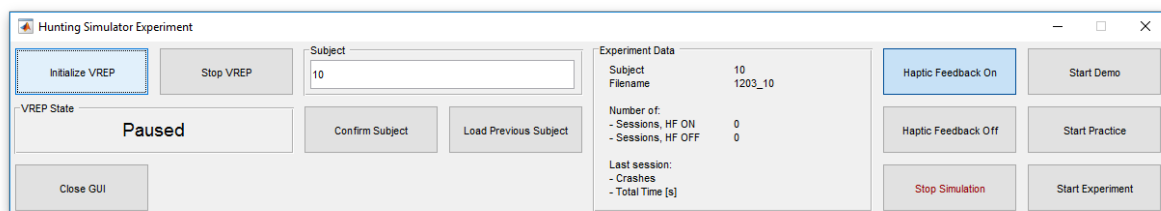


Figure B.16: The GUI used to run the experiment

Setup GUI

The setup GUI was used for the setup of the experiment and is shown in Figure B.17. Its purpose was to define the obstacles in the enclosure, as well as designing the route segments used during practice and the experiment itself. The obstacles and route segments could be defined visually, by clicking on the desired coordinates in a preview of the enclosure. This made it easy to set things up and make adjustments, rather than manually entering individual coordinates in a matrix. Small adjustments could be made to individual waypoints using the arrow buttons, as well as creating entire route segments at once.

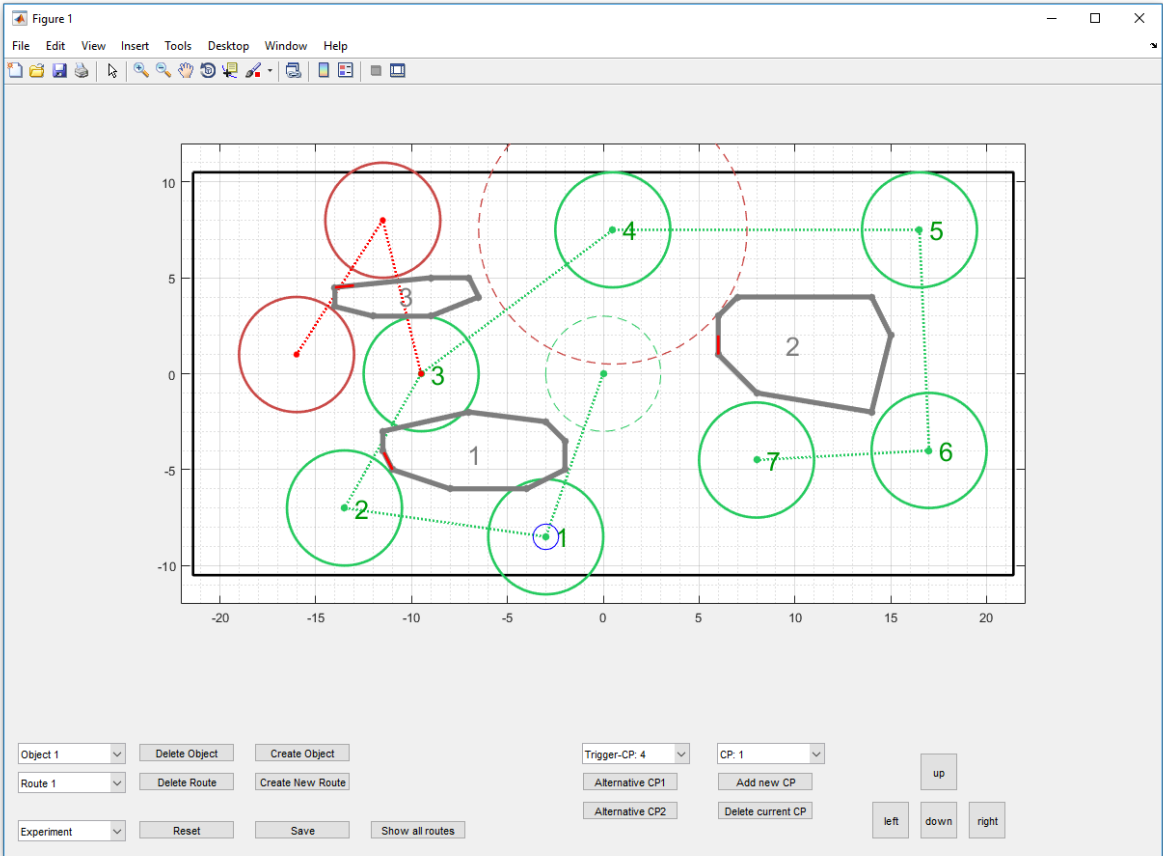
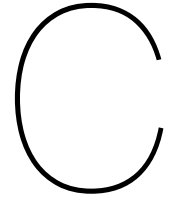


Figure B.17: The GUI used for the setup of the experiment, i.e. defining obstacles and creating route segments



Demo version

An alternative 'gamified' version of the virtual hunting simulator was created as a demo. In this alternate version, 'players' received points, based on their performance. The points were awarded as follows: at the moment a waypoint was reached, players had a certain amount of time to reach the next waypoint, the faster the next waypoint was reached, the more points they received. If it took too long to reach the next waypoint, no points would be awarded upon reaching it. Collisions caused players to lose quite a lot of points, so to reach a high score it was important to balance speed and safety.

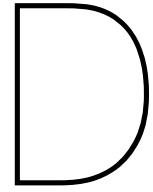
The demo version was used during MAAK Open, an open day at MAAK in Haarlem. DotX Control Solutions has its office at MAAK and therefore participated in the open day with a couple of demos. The hunting simulator game was really popular, especially amongst children. During the whole day people were waiting in line to play.

All in all, it was a very successful day, with a lot of people interested in trying the demo and to learn about the research (and lions of course). It was very interesting to see that most players really benefited from the haptic assistance. Compared to manual control, most players managed to reach much higher scores with haptic assistance. Especially interesting was that a couple of very young children had a lot of difficulties to play the game manually, with them being barely able to reach any waypoint without crashing. However, the haptic assistance helped them to perform much better, enabling them to play with a relatively small amount of collisions.

Additionally, it was really nice and satisfying to receive a lot of positive comments and to see that people were genuinely interested in the research itself and not just in the lions.



Figure C.1: Impression of the demo at MAAK Open



Experiment

D.1. Routes

The routes that participants navigated during the experiment were built by combining six different route segments in varying orders. Of these segments, three consisted of a fixed series of waypoints, while the other three contained a sudden change. The route segments that were used during the experiment are illustrated in Figure D.1. Each combined route contained each route segment twice, so twelve segments in total. Between the route segment a waypoint in the center of the enclosure was added as a transition. This was done to reduce variations in the start of each time a segment was navigated, making for a better comparison.

During the two training route that were navigated as part of the experiment, different route segments were used to prevent participants from becoming too familiar with the route segments.

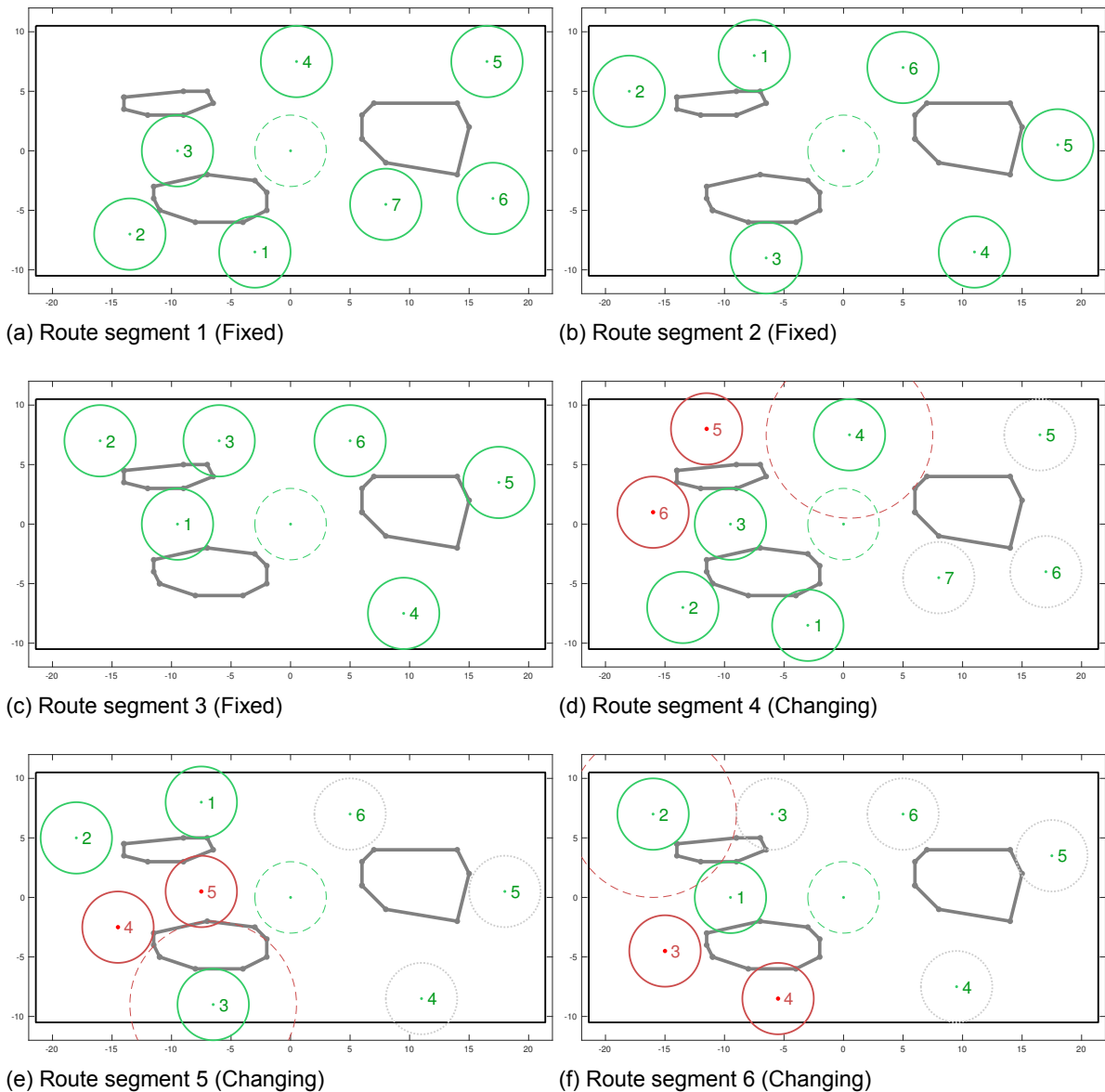


Figure D.1: The route segments which were used to build the routes that were navigated during the experiment.

D.2. Task Instruction

At the start of the experiment, each participant was shown a short slide show and were instructed on their task. Participants were asked to ask any questions they might have had and asked if they understood their task before starting the experiment. The slides are shown in Figures D.2 to D.6. The slides were accompanied with the following information:

1. Subjects were given some background information about the Lion Foundation and the hunting simulator. They were told the objective of the research
2. An explanation about the virtual hunting simulator. Participants were told how the joystick would be used to operate the system. Furthermore they were given some information about the virtual hunting simulator.
3. Subjects were briefed on the navigation task they would perform during the experiment. The waypoints and route they indicate were explained. They were asked to follow the most obvious routes without taking unnecessary detours. Furthermore they were told what would be regarded as a collision and the consequences (a short time-out before resuming)
4. Participants were instructed to follow the route indicated by the waypoints while minimizing the number of collisions.

5. An explanation of the procedure that would be followed during the experiment. Furthermore, subjects were told that they could take a break whenever necessary. They were asked to focus on their task and not to talk while navigating a route. If they had any questions, they were urged to ask those in between routes.



Figure D.2: Slide 1, Introduction and background about the Lion Foundation and the hunting simulator

Virtual Hunting Simulator

- Virtual enclosure containing 3 obstacles
- The joystick controls the horizontal position of the red disk
- The grey rectangle indicates the boundary of the allowed area
- The side view will be used during the experiment

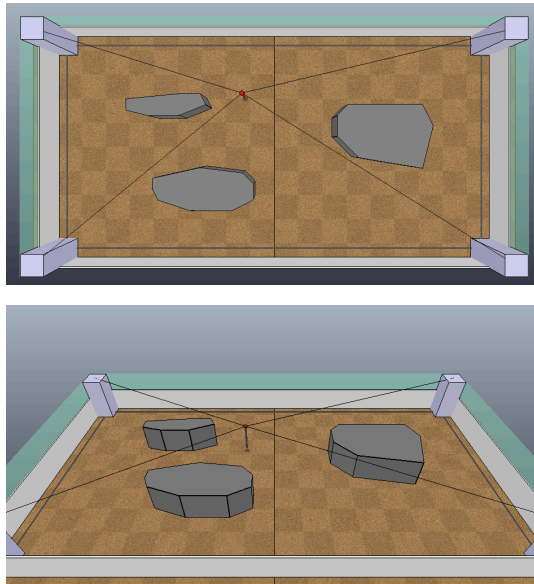


Figure D.3: Slide 2, Explanation of the virtual hunting simulator

Navigation

- The green cylinders are checkpoints that indicate the desired route
- At some random times the route suddenly changes
- Please do not deviate from the most obvious route
- Collisions will temporarily pause the simulation before resuming from the center

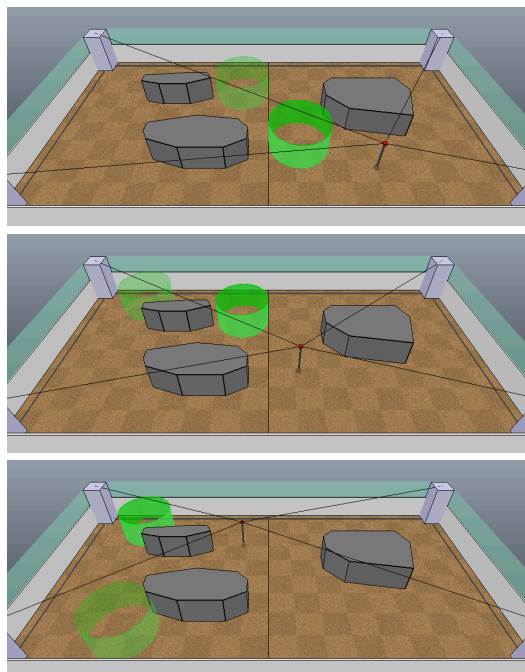


Figure D.4: Slide 3, Explanation of the navigation task that participants had to perform

Task Description

- Follow the route indicated by the checkpoints
- Avoid colliding with obstacles and walls (grey boundary)

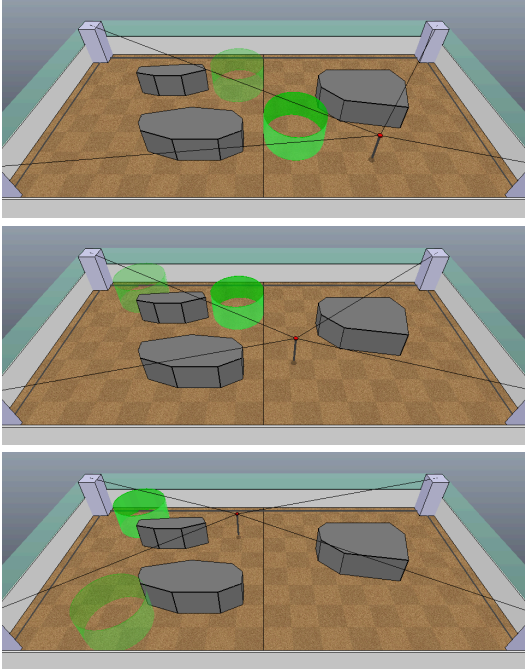


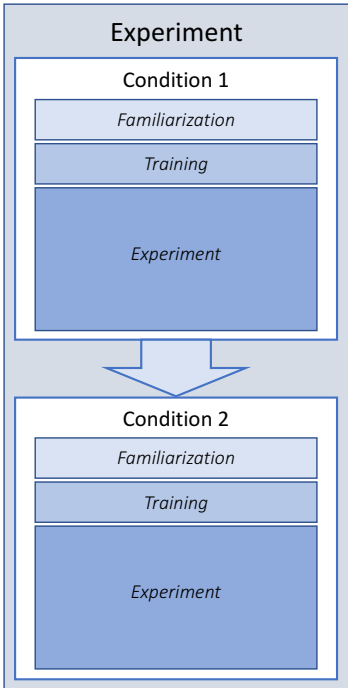
Figure D.5: Slide 4, Task instruction

Experiment procedure

- 2 conditions (with/without haptic support)

Per condition:

- Familiarization with the environment
- 2 Training runs
- 5 Experiment runs



```

graph TD
    subgraph Experiment
        subgraph Condition 1
            C1_Fam[Familiarization]
            C1_Train[Training]
            C1_Exp[Experiment]
        end
        C1_Exp --> C2_Exp
        subgraph Condition 2
            C2_Fam[Familiarization]
            C2_Train[Training]
            C2_Exp[Experiment]
        end
    end
  
```

Figure D.6: Slide 5, Information regarding the experiment procedure

Results

E.1. Across participants

The results of the mean collision ratios and average velocities are shown in Figure E.1, these figures were also presented in the paper. Not presented in the paper were figures for the mean of the completion times, minimum DTC, and minimum TTC, which are shown in Figure E.2

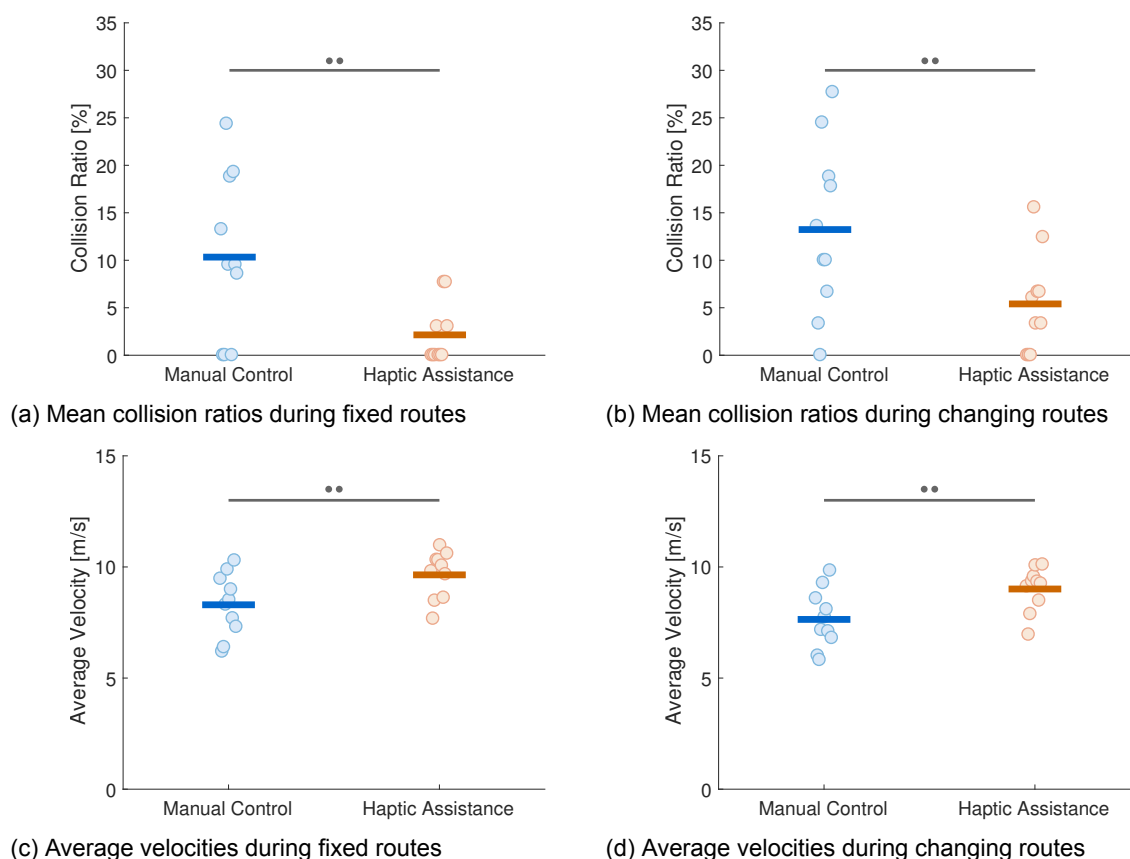
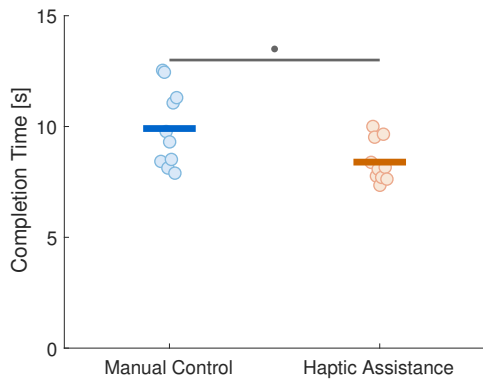


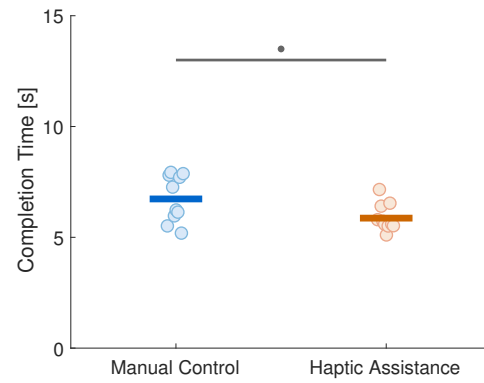
Figure E.1: Results across participants, mean of the collision ratios and average velocities

E.2. Within participants

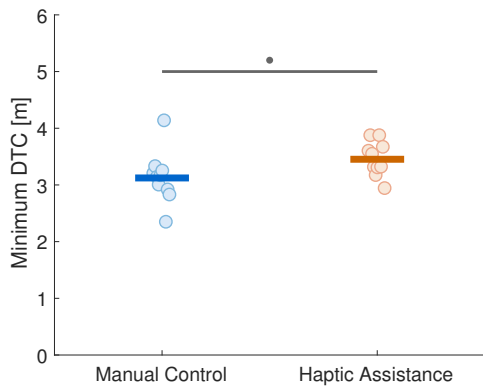
Results for four individual participants one, four, seven, and ten are shown in Figures E.3 to E.6. These include the average velocities, number of collisions and trajectories for each route segment.



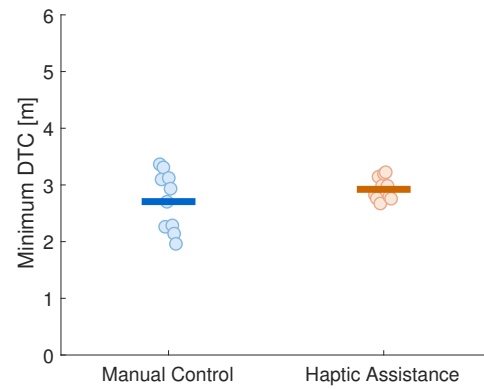
(a) Mean completion times during fixed routes



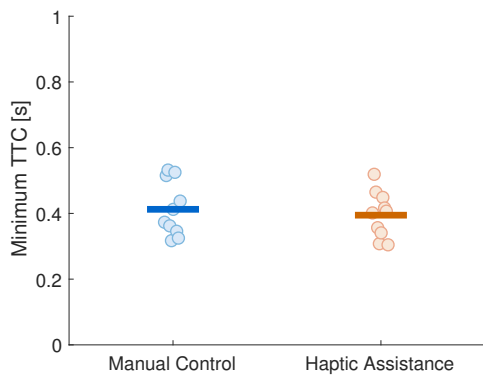
(b) Mean completion times during changing routes



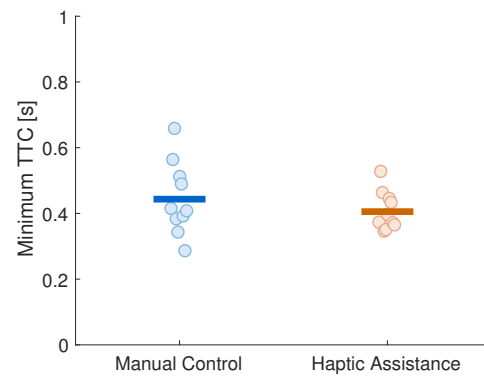
(c) Mean minimum DTC during fixed routes



(d) Mean minimum DTC during changing routes



(e) Mean minimum TTC during fixed routes



(f) Mean minimum TTC during changing routes

Figure E.2: Results across participants, mean of the completion times, minimum DTC, and minimum TTC

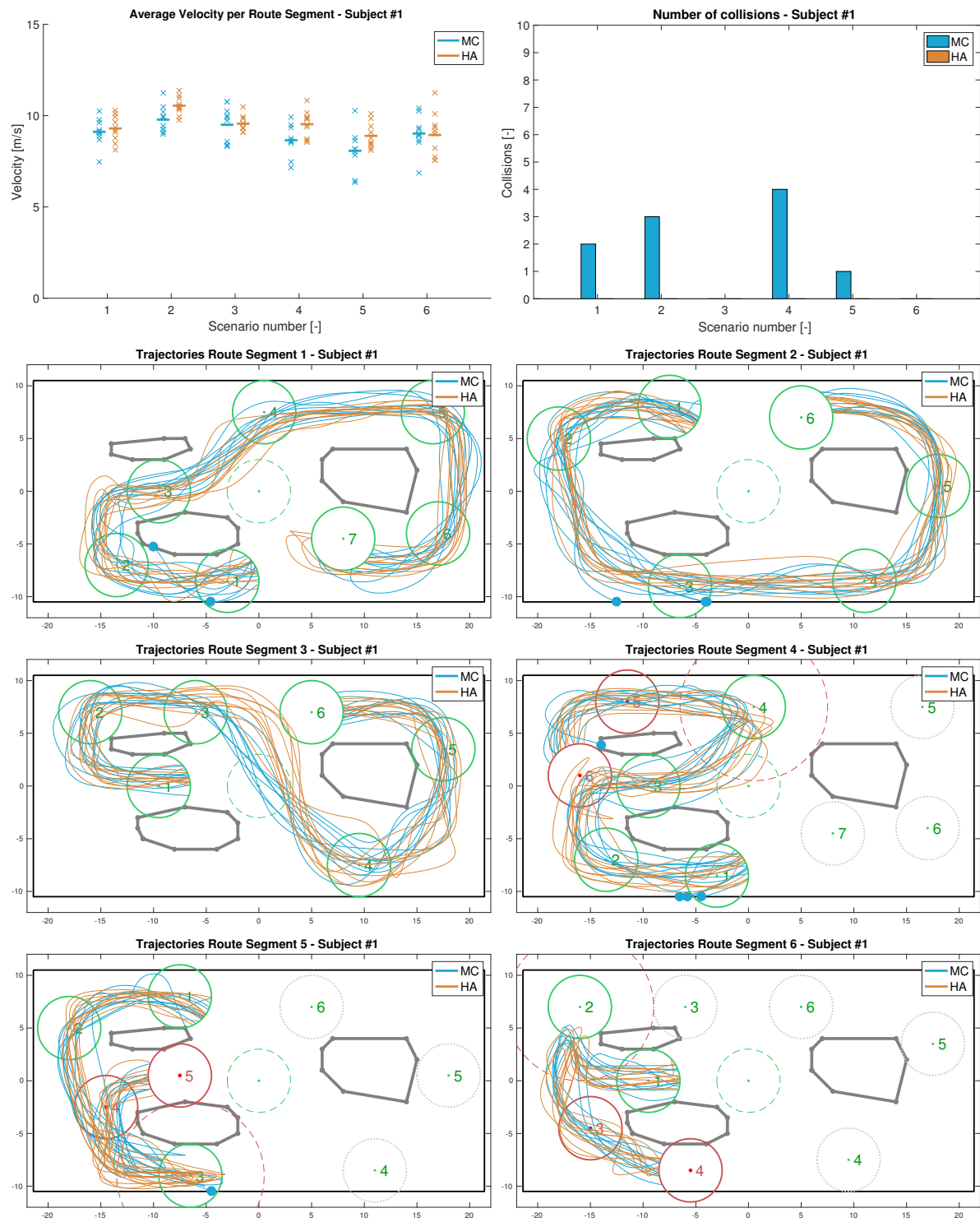


Figure E.3: Results per route segment for subject 1, average velocities, number of collisions, and trajectories

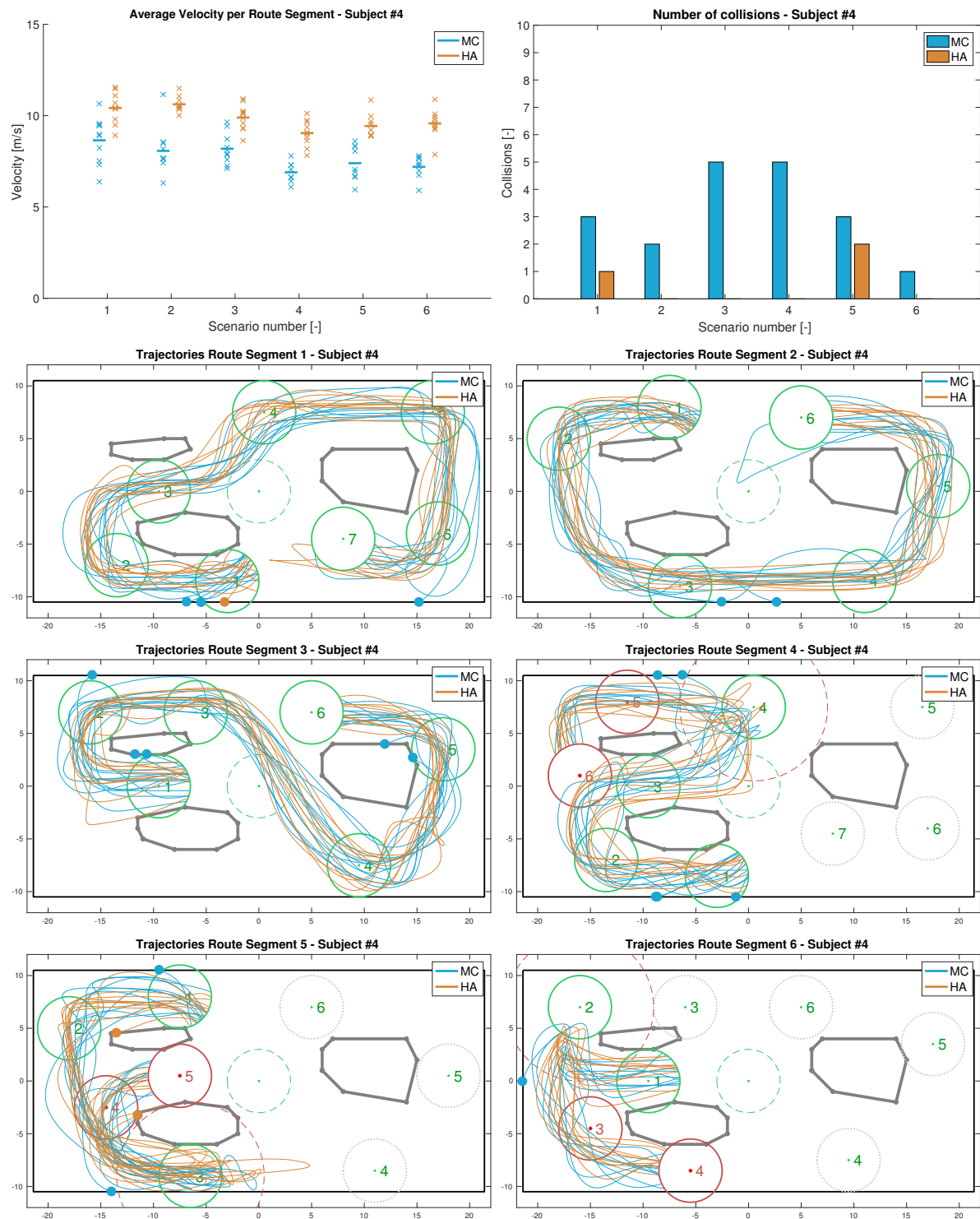


Figure E.4: Results per route segment for subject 4, average velocities, number of collisions, and trajectories

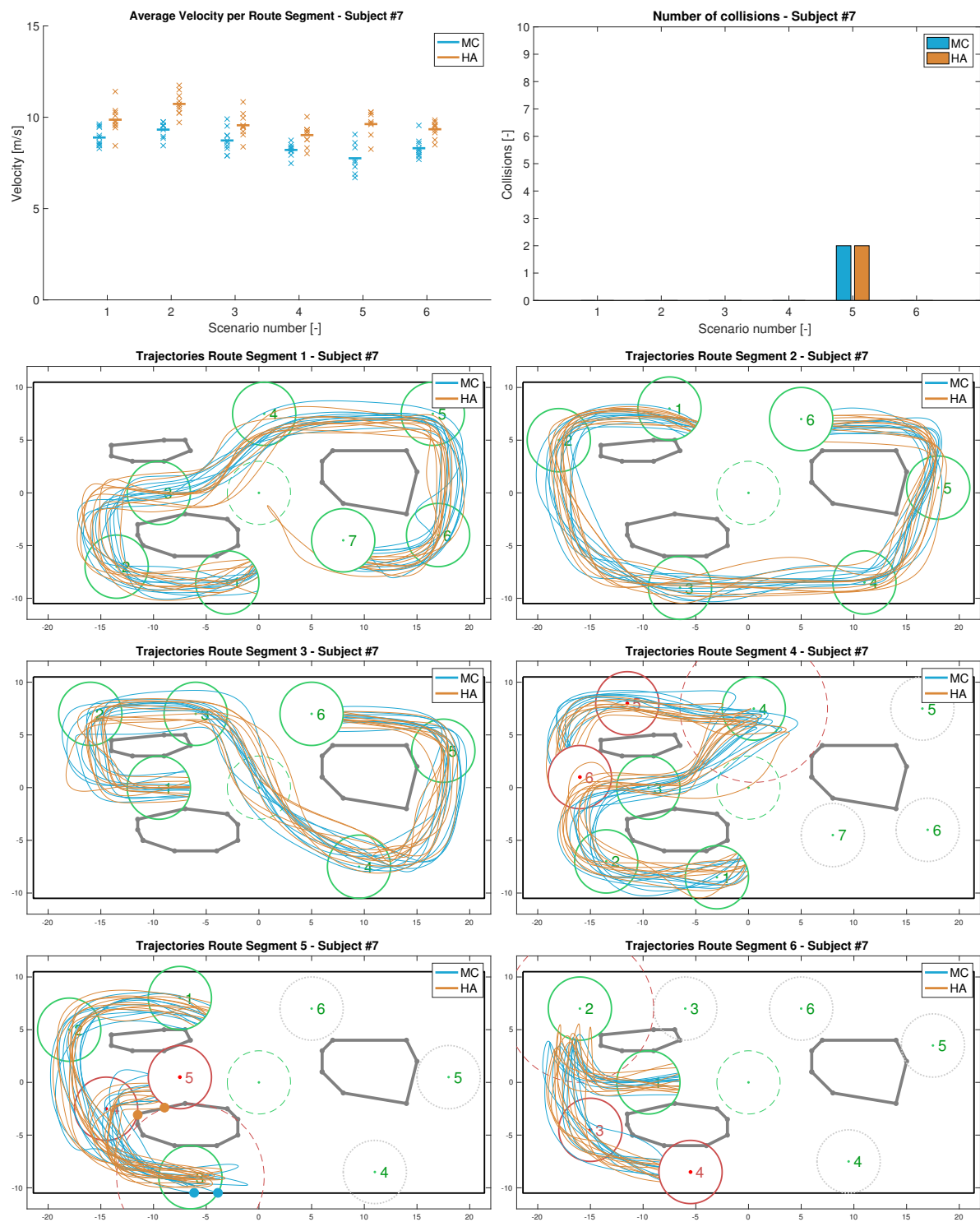


Figure E.5: Results per route segment for subject 7, average velocities, number of collisions, and trajectories

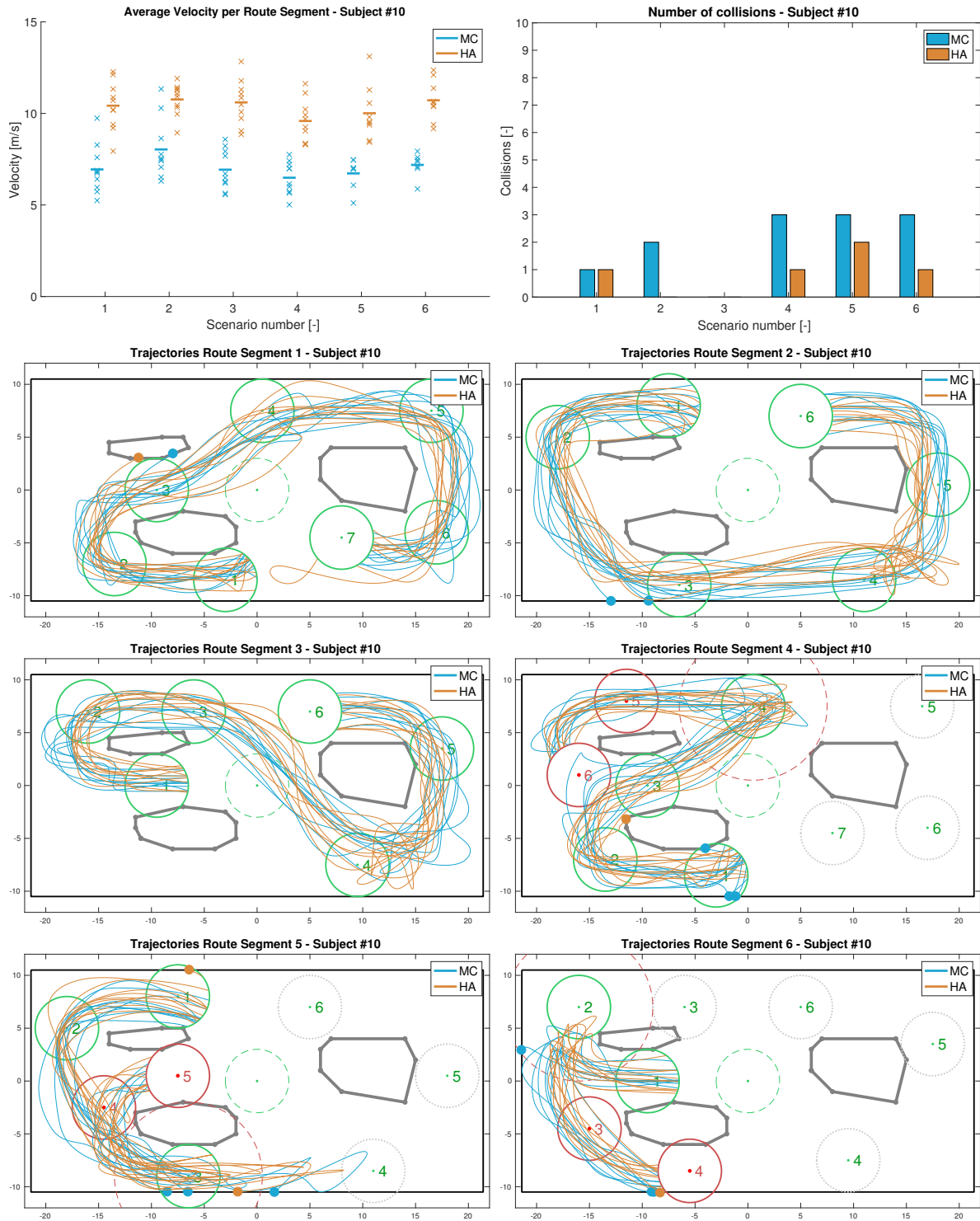


Figure E.6: Results per route segment for subject 10, average velocities, number of collisions, and trajectories

F

Informed Consent Form

Consent to participate in a research study

Application of haptic feedback to a joystick used to navigate a virtual environment.

Researchers:

J. Kremer - Master student

Email: j.kremer@student.tudelft.nl

Tel: +31 645402434

dr.ir. D.A. Abbink - Supervisor

Email: d.a.abbink@tudelft.nl

dr.ir. J. Schuurmans - Supervisor

Email: j.schuurmans@dotxcontrol.com

Introduction

You are asked to participate in a research study conducted by J. Kremer as part of his master's thesis in Mechanical Engineering, at the The Delft Haptics Lab, Faculty of 3mE, Delft University of Technology. Your participation in this study is entirely voluntary and you have the right to withdraw from the study at any point. Please read the information contained in this document carefully and ask any questions that may arise, before agreeing to participate.

Purpose of the study

The purpose of this study is to investigate the effectiveness and user experience of a haptic support system applied to a joystick. The haptic support system will support the operator (you), during a 2d navigation task through a 3d virtual environment. In other words, you will use a joystick to move an object horizontally through an environment containing obstacles, while the joystick will provide forces to help you to avoid these obstacles. Ultimately the results of this research will be used as part of a master's thesis and possibly a scientific publication.

Procedure

Before the experiment starts you will be asked to answer a short questionnaire containing some general questions about you and your experience with computer games. Subsequently, you will be briefed on some background regarding the research study and your task during the experiment.

The experiment consists of testing the navigation task under two different conditions, one with and one without haptic support. Each condition will take up approximately 15 to 20 minutes and starts with some training runs to familiarise yourself with the system, followed by 5 trials testing that condition. After both conditions are finished you are asked to fill out a questionnaire regarding your experience with the haptic support system. The total duration of your participation in this study will be approximately 50 minutes.

Risks and discomforts

In some persons, simulators and virtual environments may cause different types of sickness: visuo-motor dysfunctions (such as eyestrain, blurred vision, difficulty focussing), nausea, drowsiness, fatigue, or headache. These symptoms are similar to motion sickness. If you feel uncomfortable in

any way, you are advised to stop the experiment.

Using the joystick for an extended period of time can result in muscle fatigue, please ask for a break if your muscles need some rest. You can stop participating at any point during the experiment without any negative consequences, if you do not feel well, then please take sufficient rest before leaving the laboratory.

Confidentiality

All the data collected in this study will be kept confidential and will be used for research purposes only. Throughout the study you will only be identified by a subject number only.

Right to refuse or withdraw

Your participation in this research study is entirely voluntary and you may refuse to participate. If you accept to participate in this study you may withdraw at any time, without any negative consequences.

Consent

I understand the information provided in this document and my questions have been answered to my satisfaction. I give permission to process the data for the described purposes and I agree to participate in this study.

Name of subject:

Signature of subject:

Date:

Bibliography

- [1] Per Henrik Borgstrom, Brett L. Jordan, Gaurav S. Sukhatme, Maxim a. Batalin, and William J. Kaiser. Rapid computation of optimally safe tension distributions for parallel cable-driven robots. *IEEE Transactions on Robotics*, 25(6):1271–1281, 2009. ISSN 15523098. doi: 10.1109/TRO.2009.2032957.
- [2] Tobias Bruckmann, Andreas Pott, and Manfred Hiller. Calculating force distributions for redundantly actuated tendon-based Stewart platforms. In *Advances in Robot Kinematics*, pages 403–412. Springer Netherlands, 2006. doi: 10.1007/978-1-4020-4941-5_44. URL http://link.springer.com/10.1007/978-1-4020-4941-5_44.
- [3] Tobias Bruckmann, Lars Mikelsons, and Thorsten Brandt. Wire Robots Part I: Kinematics, Analysis & Design. *Parallel Manipulators, New Developments*, 1(April):109–132, 2008. doi: 10.5772/5365. URL <http://scholar.google.com/scholar?hl=en{%&}btnG=Search{%&}q=intitle:Wire+Robots+Part+I+Kinematics+,+Analysis+{%&}+Design{#}0>.
- [4] Clément Gosselin and Martin Grenier. On the determination of the force distribution in overconstrained cable-driven parallel mechanisms. *Meccanica*, 46(1):3–15, 2011. ISSN 00256455. doi: 10.1007/s11012-010-9369-x.
- [5] M. James. The Generalised Inverse. *The Mathematical Gazette*, 62(420):109, jun 1978. ISSN 00255572. doi: 10.2307/3617665. URL <http://www.jstor.org/stable/3617665?origin=crossref>.
- [6] S Kawamura and K Ito. A new type of master robot for teleoperation using a radial wire drive system. *Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on*, 1(C):55–60 vol.1, 1993. doi: 10.1109/IROS.1993.583079.
- [7] Alain Liegeois. Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(12):868–871, 1977. ISSN 0018-9472. doi: 10.1109/TSMC.1977.4309644. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4309644>.
- [8] Lars Mikelsons, Tobias Bruckmann, Manfred Hiller, and Dieter Schramm. A real-time capable force calculation algorithm for redundant tendon-based parallel manipulators. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3869–3874, 2008. ISSN 10504729. doi: 10.1109/ROBOT.2008.4543805.
- [9] R. Penrose and J. A. Todd. On best approximate solutions of linear matrix equations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 52(01):17, 1956. ISSN 0305-0041. doi: 10.1017/S0305004100030929.
- [10] Andreas Pott, Tobias Bruckmann, and Lars Mikelsons. Closed-form Force Distribution for Parallel Wire Robots, 2009. URL <http://www.springerlink.com/content/14m272820122432j/{%}5CnClosed-formForceDistributionforParallelWireRobots.pdf>.
- [11] Daniel Whitney. Resolved Motion Rate Control of Manipulators and Human Prostheses. *IEEE Transactions on Man Machine Systems*, 10(2):47–53, 1969. ISSN 0536-1540. doi: 10.1109/TMMS.1969.299896.
- [12] Robert L. Williams and Paolo Gallina. Translational planar cable-direct-driven robots. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 37(1):69–96, 2003. ISSN 09210296. doi: 10.1023/A:1023975507009.