RADBOUD UNIVERSITY

BACHELOR THESIS

Sharing Control with a Robotic Ball

About continuous sharing of control between two agents

Author:
J. Pieter MARSMAN
J.P.Marsman@student.ru.nl
s3055078

Supervisor:
Dr. Louis G. VUURPIJL
l.vuurpijl@donders.ru.nl
Artificial Intelligence
Faculty of Social Science Nijmegen

August 17, 2013

Contents

Abstract

This research investigates the possibilities of a system for shared control suggested by Poncela and others in 2009 [22]. They propose to continuously share control between two agents by weighting their motion commands with the locally computed efficiencies of those motion commands. From earlier literature, four important properties for shared control are extracted. The system for shared control presented here supports three of those properties: it supports the different capabilities of the agents, adjusts to the performance of the agents and creates a seamless integration of the motion commands.

To explore these properties an experiment is conducted with a robotic ball called Sphero. This proves to be a simple and low-cost environment for research on shared control in path navigation. In the conducted experiment the system decreases the length of a path, driven with the robotic ball Sphero, with 29.6% ($p < 0.001$), the duration of a path with 20.0% ($p < 0.05$) and increases the efficiencies of the actual given command with 11.4% ($p < 0.001$). These results indicate that this system is indeed capable of increasing the performance of a human agent by sharing control with an intelligent system.

Keywords: shared control, shared autonomy, Sphero, autonomous robot, human operator.

Chapter 1

Introduction

In the last decades the field of Artificial Intelligence has made a lot of progress in decision making under uncertainty. This resulted among others in an improvement of the performance of autonomous robotic systems. These autonomous robots have to find their own way in unstructured, unpredictable and sometimes hostile environments. Autonomous robots are getting better in making our life easier by autonomously performing dangerous, dull or dirty tasks. But, as explained in the next section, they are still not reliable enough.

1.1 Disadvantages of computers and humans in controlling mobile robots

The variability of a mobile robot environment makes the control of a mobile robot difficult and complicated [2]. Unlike static robots, *mobile robots* can change their own position in the environment. As explained in the following paragraphs, both computers and humans have disadvantages in controlling such mobile robots.

Intelligent computer systems (IS) that control autonomous robots have, in general, too little general knowledge about the environment. Also, the reasoning of an IS is hard to specify for unpredictable situations. Although, the mobile robot should not harm the environment [1], autonomous robots cannot guarantee this level of safety [4]. Due to this deficiencies, an IS is not sufficient enough for reliable control of mobile robots.

Humans can also operate mobile robots from a distance, a process that is called teleoperation. Human operators are intelligent decision makers, but they also have their downsides. Most of the times, human operators cannot be presented with contextual or complete information about the environment of the mobile robot. Mobile robots frequently show the human operator a limited view of the environment. Consequently, for complex robots or environments, human operators cannot oversee all the possibilities and dangers, and in such situations they are less capable of making the right decisions. A possible high workload also decreases the opportunity fo a human operator to make the best decision [2].

The ultimate risk for human operators and IS agents is that the mobile robot gets damaged, and maybe unable to continue its mission, due to their erroneous command. In the case of expensive robots there is too much at stake to let only one agent control the robot. Consequently, a single agent is not reliable enough to control a mobile robot.

1.2 Sharing control

As illustrated above, both a IS and humans have disadvantages in controlling mobile robots. The IS cannot perform as well as humans on a lot of complicated cognitive tasks. On the other hand, human operators cannot be as precise and consistent as autonomous robots when the workload increases. Perhaps there could be a way wherein the human could assist the IS and vice versa. It could be used to [2, 22, 28, 29]:

- Reduce the disadvantages of agents.
- Boost the performance of a mobile robot.
- Make mobile robots more reliable.
- Relieve the human from the burden of direct control.

The research fields of Shared Control and Human-Robot Collaboration (HRC) are concerned with the possibilities of cooperation between IS and human agents. These fields offers a compelling opportunity to combine human intelligence and computer efficiency [6]. Shared control between agents is defined as follows:

Definition 1. Shared control *is the process where human agents and intelligent systems cooperate to achieve a common goal [29].*

But the achievement of shared control is difficult. Typically, human operators who are responsible for the mobile robot always want to be in control over a mobile robot, even when most tasks can be delegated to autonomous agents [19]. Notwithstanding that sharing of control can reduce the workload for human operators [26]. Hence, there is a trade-off between minimal human input and maximum feeling of control. This consideration is important for small domestic robots but also for expensive military- or company-robots and makes the acquisition of shared control hard.

1.3 Applications of Shared Control

Since robots are not exclusively used in experimental or manufacturing environments, but also in more complex environments with humans such as homes, offices and hospitals the range of applications is wide. Already humans work together with robots in elderly care and urban search & rescue [20]. Other possible applications for HRC are in health care, construction, tour guiding, home service and entertainment [4]. In the future, collaborating with robots will be even more part of our every-day life.

A specific application that shows the need for shared control is concerned with automatic wheelchair control. Automatic wheelchairs are used by users with a physical or cognitive disability. Most of the time, they are not capable of safely navigating themselves through corridors and along obstacles. These people could benefit if their wheelchairs were safer due to the sharing of control with an IS.

A prominent example of shared control in automatic wheelchair control is the use of Brain Computer Interfacing (BCI) [10]. The automated comprehension of intentions from brain signals, to control a automatic wheelchair, is hard and extracted intentions are not always what the user wants [15]. For this purpose a framework is needed that allows the user to navigate freely while preventing the automatic wheelchair for safety hazards like bumping into obstacles. Shared control could increase the performance of automatic wheelchair drivers while guarding the safety of the vehicle and driver [29].

Another application is the control of multiple robots at the same time. In the American military mobile robots are developed to perform reconnaissance, surveillance and target acquisition. These tasks typically require human resources, however not all the time [9]. A shared control system can relieve the human operator from the burden of continuously controlling or supervising a single robot and thereby enabling him to control multiple robots at the same time.

1.4 Approach

There already have been research to specific implementations of shared control. A major trend until now is concerned with adjustable autonomy that allows agents to adjust their level of autonomy[1]. However, these approaches have a discrete nature, the number of levels is fixed. Recently, research focussed on a more *continuous sharing of control* wherein humans and computers are both fully autonomous and contribute to the robots behaviour at the same time. This delivered some promising results for intelligent wheelchairs [21, 28, 29], other mobility aids for elderly [30], joystick driven four-wheel robots [22] and simulated tele-operation [17].

[1] See Section 2.3.1 for more information about adjusting levels of autonomy.

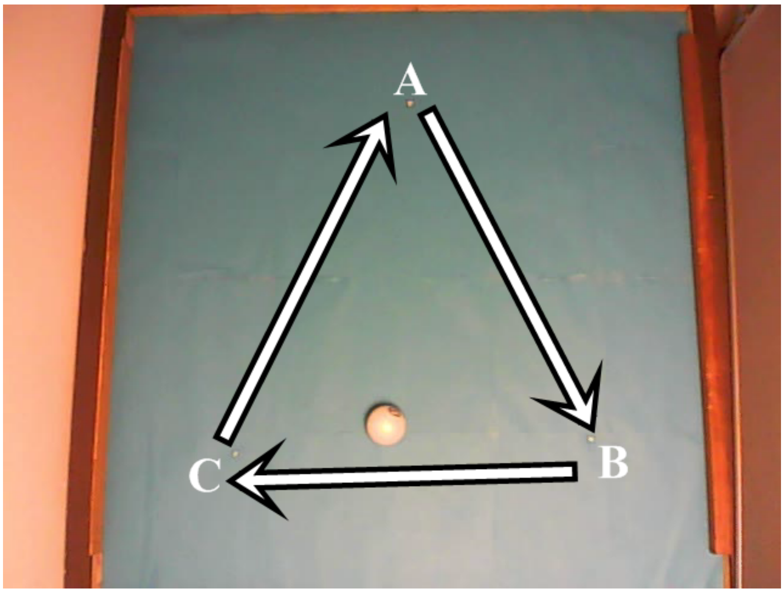


Figure 1.1: *The path that Sphero should travel in the research environment.*

The research pursued in this thesis aims to learn more about continuous shared control for robots by applying a specific system on a robotic ball called Sphero (see Appendix C.1 for an image of Sphero). The specific system is proposed by Poncela and others in 2009 [22]. They propose to combine two commands of different agents continuously with respect to the efficiency of the those commands. The first research question will be:

> RQ1: How well does the system of Poncela and others increase the performance of agents controlling Sphero?

Sphero is used because it is a cheap (±130$) toy that can easily be controlled with a mobile phone or computer via Bluetooth. It is used to create a research environment for shared control. Sphero should drive a circular path along three markers, as can be seen in Figure 1.1. The robotic ball is controlled by a human, who gives command using a tablet, and a IS which commands are based on a video stream of the environment. The system of Poncela and others is applied to generate continuous shared commands.

In contrast to traditional research environment, such as automatic wheelchair control, this environment is low-cost, because no expensive equipment is needed. Also, participants can learn fast how to control Sphero, something that is not possible with more complicated robots. The second research question is concerned with this new promising research environment:

RQ2: To which extend is the control of Sphero a suitable research environment for shared control?

The research questions are explained in more detail in Section 2.6.

1.5 Outline of thesis

The thesis will have the following outline. In Chapter 2 will be explained which desired properties are emphasized in earlier work on shared control. These desired properties are used at the end of Chapter 2 to evaluate the system for continuous shared control as described by Poncela and others. In Chapter 3 the experiment and used methods for this research will be explained and in Chapter 4 the results of that will be shown. In Chapter 5 the results will be discussed and from this conclusions will be drawn.

Chapter 2

Background

Since robots are becoming more complex, research is focussing on how humans can better collaborate with robots. In the beginning, the human was the supervisory agent that gave commands to the robot. But since the robots are becoming more complex simple commands are unsatisfactory because the possibilities of a robot are overwhelming.

A first approach to solve this problem was to let a IS control the robot and deal with the details of execution, while the human makes more global decisions. An example of such a system is proposed by Connell and Viola in 1990 [8]. A series of behaviours such as Approach, Trek, Retreat and Align could be used instead of commands. The user does not have to give direct motor commands but rather selects a behaviour that the IS executes. For safety reasons there is a hierarchy in the behaviours which enables more important behaviours, such as collision prevention, to override other behaviours. Also the human agent could still override the commands of the IS with its own commands. This system allows the agents to think on a tasks level while also being able to prevent unsafe situation with direct commands. Though this system has some advantages, it is far from ideal for a robust, fast and intense collaboration between multiple agents.

In the last decades there have been several other attempts to share control between a human and computer agents. Several authors emphasize different desirable properties for shared control, with associated different names. Some of the attempts are: mutual initiative [7], cooperative control [8], traded control [18], adjustable autonomy [13, 14, 24, 25, 31], collaborative control [6, 9], adaptive shared control [30] and lately just shared control [21, 22, 28].

In this chapter, four desired properties will be summarized: the different capabilities of agents, mutual understand for each others state and environment, adjusting to the current needs of other agents and the continuity of shared control. Eventually the approach of Poncela and others will be evaluated by these desired properties. After that the research questions will be further elaborated.

2.1 Different capabilities and equality of agents

Different agents have different capabilities. By cooperating with other agents the downsides of an individual agent can be decreased while the advantages of other agents can be amplified. In this section it is argued that human and IS agents have different capabilities and by enabling them to take initiative and intervene decisions of other agents the mobile robot is made more resilient.

2.1.1 Capabilities of agents

On the one hand, computers are increasingly fast and accurate, this allows them to make fast and precise decisions. In addition, if operation is at a distance the IS that controls an autonomous robot often makes a better judgement of the environment than humans do [7]. But, the current technology does not allow the creation of efficient fully autonomous agents [13, 31].

On the other hand, humans are not that fast, especially not when their decisions need to be transferred through the mobile robot with a joystick or BCI device, and their commands are not precise either. But their decision making process is more reliable, in the way that they can justify their actions, and they have better capabilities in a wide range of unpredictable possible situations [31].

Three levels of capabilities of agents can be distinguished: [31]:

- The capability of an agent to accomplish its task.
- The capability of an agent to decide how to accomplish its task.
- The capability of an agent to define a plan of actions.

The capability of an agent to do something is defined as the skill, capacity and prescription of an agent to do that particular thing. Skill means that the agent has the necessary technologies that it can use. This includes the efficiency and reliability of an agent, because sometimes a technology only allows the agent to give a near-optimal solution or a solution with a certain probability. Capacity refers to the environment the agent is currently in and whether or not this prevents the agent to use the skill. The prescription of an agent determines if the agent is allowed to use a skill. As an example, an agent may have the technology to dig a hole with a radius of 2 metres, it is also in a position where this is possible, but its prescription prevents it from digging a hole because it is not allowed to dig a hole of two metres in its parents backyard. On another level, an agent may have the technology to define a plan of actions, but is not capable of doing so because there is not enough computational power.

2.1.2 The use of capabilities

In shared control both human and IS agents should be enabled to use their capabilities to prevent the mobile robot from major mishaps. To do this agents should take initiative in a task when their capabilities are best suited for the job. The agents should also use their capabilities to intervene other agents at all times when they foresee a problem [7]. To prepare both agents for such initiatives and interventions, they must:

- Be equal [18]. Being equal means that all agents can intervene the current actions that are taken. Not only can humans intervene decisions that are made by computers but also computers can intervene human decisions. Nowadays, agents are not equal. Rather a human supervisory control schema is used to control mobile robots. In this schema the robot supplies information to augment the human cognition [7].

- Understand and predict each others performance [4, 6]. This is discussed in Section 2.2.

- Remain aware of the current situation and the other agents state [16, 18]. Along with the understanding and prediction of each others performance, this is discussed in Section 2.2.

The goal for a mobile robot is to be resilient, that is defined as "the intrinsic ability (...) to adjust its functioning prior to, during, or following changes and disturbances, so that it can sustain required operations under both expected and unexpected conditions." [16]. Teaming between humans and computers can increase the level of resilience by using the unique capabilities of each agent.

2.2 Mutual understanding

This section argues that mutual understanding for the goals, plans and actions of other agents, is important for efficient collaboration (Section 2.2.1). To collaborate agents should be able to predict the intentions of other agents [4, 6] and assist other agents if necessary or ask for assistance of others (Section 2.2.2). Also the understanding for the current environment of other agents is important (Section 2.2.3).

2.2.1 Shared goal, plan and actions

A team is defined as a small number of agents with supplementary capabilities that are committed to a shared goal. Effective teams share their intentions and inform each other about the current situation [7]. The first step of collaboration is to agree on the shared goal, without a shared goal collaboration is useless. In most collaborative control systems the goal is determined by the human agent, however this can change in the future.

To work efficiently together, a shared plan between all agents to reach the goal is required. This is important because different agents can make different plans to reach the same goal. In human-robot teams it is often the task of a IS to estimate the intended plan of a human and to assist the human agent properly [7]. However, an IS can also make plans, an early example of a system for robot control that separates the general intelligence for planning into several levels of cognition is developed by Kortenkamp and others in 1997 [18]. The system consists of three interacting layers - the skill manager, a sequencer and the deliberative planner - to arrange a plan for a goal, divide it into subtask and subsequently execute these subtasks. This approach creates a flexible, robust and tractable way that could increase the involvement of the IS in the shared planning process.

With the shared plan actions can and executed that satisfy all agents in the team. Also, the shared goal and plan allows the agents of a team to anticipate which information is needed by other agents [4]. This is discussed in the following section.

2.2.2 Predict intentions

To understand and predict each others actions it is important that the human has a theory of behaviour for the IS [7]. This allows the human to understand what actions the IS will take, in order to follow the plan and in response to the current environment of the mobile robot. If this happens the human feels like (s)he is physically near the mobile robot and doing the task directly, something that is called high situational awareness or sometimes transparency [17].

Often in human-robot teams, there is a lack of understanding the decisions of the IS from the human perspective, which results in a bad theory of behaviour of the IS. For example in the case of teleoperation there is often a lack of understanding of the state of the IS. This is because it is difficult for humans to build a mental model of a remote location [9] because the human receives a narrow view on the environment of the mobile robot. The estimation of a distance or angle based on a camera feed is hard and the humans theory of behaviour of an the IS suffers from this lack of information. But despite this high operator workload, communication constraints and poor visibility, camera feeds remain the primary mean of providing situational awareness.

Humans need a theory of behaviour for the other agents, but IS agents also should also have a theory of human behaviour. This does not mean that an IS should fathom the human mental model but it is rather a learned expectation developed through practical experience [7]. This could be as simple as an average of the past period but also as complex as a machine cognition of the human needs while encountering for workload, arousal and degraded performance. The creation and use of the human mental model is very difficult and it may require the human to be augmented with instrumentation to provide information for the IS.

The theories of behaviour give the agents the unique capability to assist each other but also ask for assistance in situation that requires intervention from another agent [18]. This gives rise to a dynamic switching of influence of agents. This subject is treated in Section 2.3.1.

2.2.3 Situational awareness

The lack of situational awareness is a known cause of frustration in humans while operating a mobile robot. Because the human cannot understand the current state of the robot, (s)he is also not able to understand why the robot is not following the given commands. This frustration can lead to the generation of even worse commands. In the case of, automatic wheelchairs frustration could lead to a decreased sense of mobility while it is precisely the point of automatic wheelchairs to increase the sense of mobility [28].

Besides frustration, a lack of situational awareness can prohibit the agents to understand the actions of other agents. To prevent this, continuously monitoring actions of other agents is necessary because then an agent knowns the current state of other agents and the mobile robot. This is important for each agent in order to be capable of resuming or intervening the current operations of the other agents [4, 18].

2.3 Adjust to needs

In the next section four stages of sharing control are explained and discussed. In section 2.3.2 a comparison between human-robot teams and strictly human teams is made to illustrate adjustable autonomy.

2.3.1 Adjusting the level of autonomy

Based on what the agents know about each other they can assist or ask for help. Several frameworks in which the need for assistance can be controlled are proposed and most of them let either the human agent [13] or both agents modify the level of autonomy. Often a distinction between four levels of autonomy is made [7, 13, 18, 19]:

- In *teleoperation* mode the human agent must manually control all mobile robot movement. In this mode there is no influence of the IS on any level. This level can tolerate no neglect of the human agent.

- In *safe mode* the human agent also controls all the robot movement manually but the IS prevents the human operator from colliding with obstacles. The IS functions as a safeguard layer around the actions of the human agent. This level can tolerate few neglect of the human agent; nothing dangerous will happen but neither any progression is made when the human operator neglects the robot.

- In *shared mode* the IS takes over the direct control of the human agent. The IS takes initiative to use its own navigation algorithm to find a path based on the directions of the human agent. Depending on the implementation the IS accepts the interventions of the human operator or the human agent and IS work on independently on separate tasks while reporting to each other. This level can compensate for a lot of human agent neglect.

- In *autonomous mode* the IS is responsible for the all the actions of the mobile robot. Often it consists of several high-level tasks for the IS such as wall-following, area-search or building a map. Depending on the implementation, the human agent can intervene the IS at the task level [18] or not at all [13]. This level can tolerate a high level of neglect from the human agent.

These modes describe the influence and responsibilities of agents in a team. The key to these levels is that agents can dynamically adjust the level of autonomy based on the current situation, their capabilities [18, 24] and the capabilities of other agents [31].

Figure 2.1 shows the trade-off between the performance of the system and the neglect by the human agent for the different levels of autonomy [13]. The amount of neglect is, among others, related to the human workload, time delays between robot and human operator and tiredness. For a dull task the IS can take over the control in the autonomous mode to prevent fatigue of the human agent. When the current workload of the human agent is high the level can be set to shared control to unburden the human agent from simple tasks. A high workload of the human agent means lower capacity to give input to the system and as a consequence the system should compensate for this neglect. In other situations where the human agent has its own plan, but it does not know the environment well, the safe mode helps the human agent to navigate safely. If a task is too complicated or special for a IS, the teleoperation mode can be used to enable the human agent to control the mobile robot entirely.

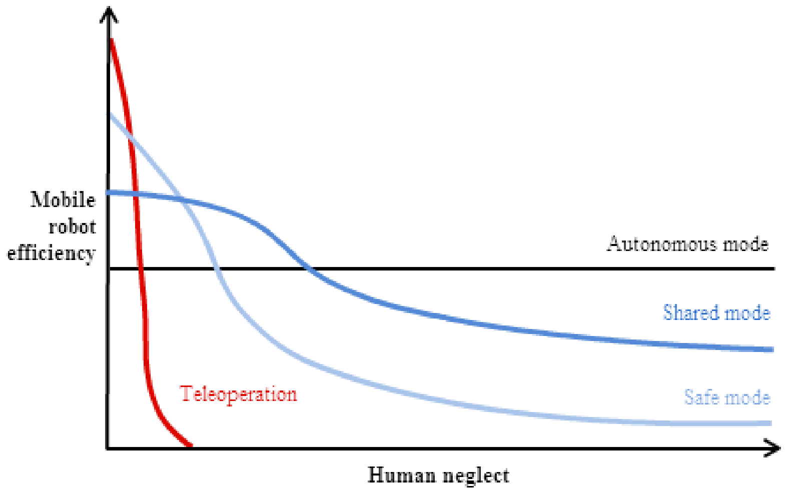


Figure 2.1: *Effect of human neglect in various levels of autonomy [13].*

2.3.2 Comparison with human teams

Adjusting the level of autonomy can be compared with what happens in human teams. Members of human teams do not have static roles but their roles change dynamically to optimally use the capabilities of members to meet unforeseen challenges. Many members of human teams end up playing roles that are different from their first role in a team that gained them admission into the group [6]. This switching of roles to cope with new problems makes a team flexible and in case of human-robot teams it makes them resilient.

However, nowadays the division of labour in robot-human teams is static. The robot is used as a tool by the human agents and there are few adjustments in responsibilities. This switching in the level of autonomy can compensate for the limitations of both agents. A system that can adjust to the current capabilities of agents and the demanding tasks can improve robot performance [6].

But switching between a number of predefined states has some disadvantages. When the level of control changes from one to another agent, sudden changes in the mobile robots actions can occur. The next section will discuss why it is important that the level of control changes continuously.

2.4 Continuous sharing

Most approaches to shared control use some kind of discrete intervention to swap the control from one agent to the other [22]. These interventions can be initiated by a human request, hazard detection by the IS or by some more complex algorithm in the shared control system. But, this discrete swapping of control has three disadvantages:

1. When control is switched from one agent to another, there also can be a abrupt change of direction in the motion command that is sent to the robot. This happens when the agents that share control have different perspectives on the direction to take.
2. Both agents can suddenly lose their influence on the robot when other agents decide to take over the command. Unfinished plans of the injured agent can cause frustration and may disrupt the well going collaboration.
3. The amount of options in the user interface increases as the human operator should be able to select the level of autonomy. But, to relieve the human operator from a high workload, fewer options in the user interface are desired.

A more continuous sharing of control could avert the first two disadvantages and should be used when sharing control [1, 18, 21, 22, 28, 29].

2.4.1 System for shared control by Aigner and McCarragher

In 2000, Aigner and McCarragher [2] have created a system that enables humans to interact, simultaneously, continuously and discretely with autonomous robots. The system allows the human agents to input continuous signals, such as a steering direction, and discrete signals, such as a stop order. Especially interesting is that it can combine continuous velocity commands of a human agent and an IS, with the so-called Continuous Time System. The combination of the two commands is just a simple addition but it shows how multiple agents can work together in real-time.

2.4.2 System for shared control by Poncela and others

From 2009 on, Poncela and others [22, 28, 29] propose a similar system for the sharing of control. This system will be the main topic of the research done in this thesis. The system focusses on continuous collaboration between two agents to achieve a better performance. In earlier research the responsibilities of control were discretely divided between agents. But in this approach two agents simultaneously provide a motion command for each time frame. The system also incorporates the performance of the human agent and IS in the combination of the two commands.

Shared control, in the system of Poncela and others, is achieved by deriving a shared motion command from the separate motion commands of the agents. Motion commands can be gathered in multiple ways (for example joysticks, pads, voice, BCI interfaces or from algorithms) but in any case they should be converted to a vector representation. A standard representation of 2D vectors is (r,ϕ) where r is the length of the vector, and ϕ the angle with the positive x-axis.

The outputted shared motion command v_S is a weighted average of the human agent motion command v_H and IS motion command v_{IS}:

$$v_S = (1-k_H)*\eta_{IS}*v_{IS}+k_H*\eta_H*v_H \tag{2.1}$$

As can be seen in Equation 2.1 the motion commands are weighted by two constants:

- The constant k_H allows a global increase or decrease of the human contribution to the shared command. The human motion command is multiplied by k_H while the IS motion command is multiplied by opposite $(1-k_H)$.

- The efficiency η_x ensures that efficient commands have a bigger influence on the outputted shared motion command. To measure the efficiency of a command, global measures of efficiency, such as the length of a path or the duration, are not sufficient because they measure the performance of an agent at the end of each task. Instead, the efficiency should be specific to the performance of a single command and available at the same time as the commands is received. For this reason a local measure of efficiency is needed.

To locally measure the efficiency of the motion commands, three specific measures are used: smoothness, directness and safety. The efficiency of a command is the average of the those three specific efficiencies.

- Smoothness expresses the size of the deviation between the current direction of the robot and the direction of the motion command (see figure 2.2). In the Equation (2.2), α_{dif} is the difference between the current direction and the direction of the motion command. This is multiplied by a constant C_{sm}. Note that the result of the equation e^{-x} will be from zero to one for $x \in [0,\infty)$, as can be seen in the graph of this equation in Appendix A.1.

$$\eta_{sm} = e^{(-C_{sm}*\alpha_{dif})} \tag{2.2}$$

- Directness expresses how well the motion command is directed to the current goal (see figure 2.3). In equation (2.3), α_{dest} is the difference between the heading of the command and the current goal. After its is multiplied by C_{dir} it is scaled between minus one and one.

$$\eta_{dir} = e^{-C_{dir}*(\alpha_{dest})} \tag{2.3}$$

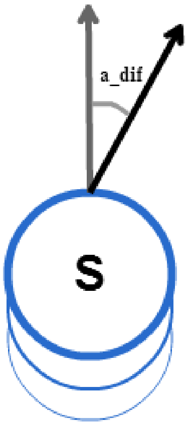


Figure 2.2: *Smoothness is based on the deviation α_{dif} between the current direction and the heading of the command.*

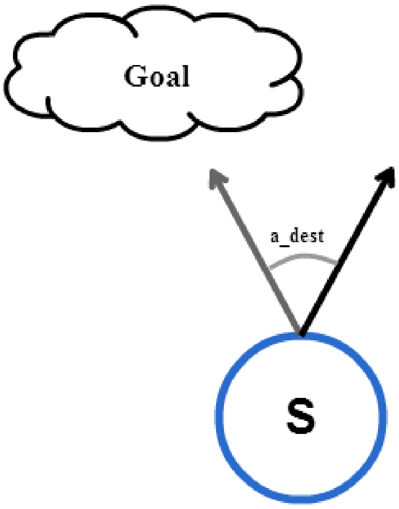


Figure 2.3: *Directness is based on the deviation α_{dest} between the direction to the goal and the heading of the command.*

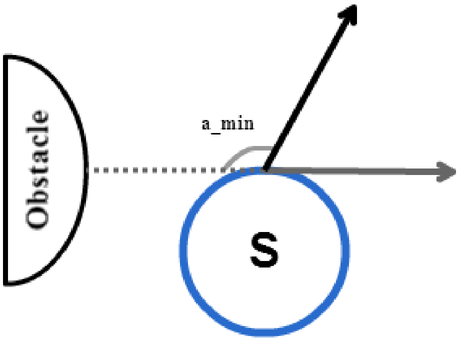


Figure 2.4: *Safety is based on the deviation α_{min} between the direction to the closest obstacle and the heading of the command.*

- Safety expresses how well the motion command is aimed away from the closest obstacle (see figure 2.4). Consequently, when the motion command is not aimed at the closest obstacle the safety measure is high. In equation (2.4) the difference between the heading of the command and the direction to the closest obstacle is expressed by α_{min}. Once it is multiplied by a constant C_{sf} and scaled between zero and one, it expressed how well the motion command is aimed at the closest obstacle. However, to get the safety efficiency, this number is subtracted from one, and it expresses how well the motion command is aimed away from the closest obstacle.

$$\eta_{sf} = 1 - e^{-C_{sf}*(\alpha_{min})} \tag{2.4}$$

To summarize, the three specific efficiencies are all a difference between the direction of the motion command given by an agent and a preferred direction. This allows the system to continuously monitor the efficiencies of the commands given by the agents.

2.4.3 Results of earlier experiments with a continuous shared control system

The proposed system of Poncela and others is tested with thirteen participants in [22]. The participants have driven a Pioneer AT robot along obstacles and through corridors. The performance of the participants, on several driving tasks, are compared to the performance of the shared control system. In the latter condition the human and IS have equal influence because the influence of humans k_H is set to 0.5. It is reported that the system improved the performance of the human agent and the performance of the IS.

Another experiment is conducted with thirty participants with changing physical and mental disabilities [29]. The participants are automatic wheelchair drivers and their motion commands received with a joystick that is mounted on an automatic wheelchair. The motion command of the IS is the output of a reactive system that used several range sensors on the automatic wheelchair as input. The reactive system includes three components: A safeguard layer to stop the wheelchair in case of imminent danger. A reactive layer to drive directly to the next subgoal and a deliberative layer that can plan a path to the final goal and thereby finding subgoals.

The reactive system is very simple but, in all cases the performance of the participants increases when they shared control with an IS. The efficiencies (see Section 2.4.2) of the given commands increased with ± 19%. In almost every case the performance of the IS also increases. It is documented that the human agents assist the reactive IS in avoiding local traps, a well documented drawback of reactive systems. The shared control system also assists human agents in driving a smooth path along a wall, where normally oscillations in the path occur.

The final experiment with eighteen participants in automatic wheelchairs [28] has also good results. The system provides continuously shared control which resulted in an increased smoothness, directness and safety of the followed path. But more amazingly, it compensates the problems of the differently disabled users. Overall, the results show that the proposed system for shared control improved the performance on several tasks and is very promising. The average efficiency of the commands improved with 6%.

2.5 Qualitative evaluation of the system for shared control by Poncela and others

To evaluate the system of Poncela and others, the properties of the system are compared to the desired properties that were derived from the literature in Sections 2.1 - 2.4.

The proposed system for continuous shared control makes use of the *different capabilities* the agents have. Especially, the capability to achieve a task is according to the system. This is because the motion command, which is the uttering of the capability of an agent to achieve a task, is constantly weighted by its efficiency. This causes the most efficient agent, and thereby the agent that is most capable of achieving the task, to have the most influence. The proposed system has no bias to a specific type of agent, they are considered as *equal agents*. The capability of an agent to decide how to achieve a task and plan actions are implicitly taken into account because they influence the agents motion command. However, these capabilities should be used more explicitly to decide on the efficiency of an agent.

There are no additional measures to increase the *mutual understanding* of both agents in the proposed system. There is no shared creation of a goal because it is assumed that both agents have the same goal. A shared plan to work efficiently together is neither presented in the system. A shared action is created but for the two agents there is no tendency to have similar commands. However, it is suggested that the same idea, of continuously sharing control weighted by efficiencies, could also be applicable to the creation of a shared plan but there is no implementation given [22]. Another deficiency is the lack of support for either of the agents to understand the actions of the other.

On the positive side, the agents are forced to update their model of the situation continuously, this improves the situational awareness of the agents. But the system does not provide any additional clues, such as haptic feedback [17], for both agents to understand the environment. Overall there is a poor support for mutual understanding in the proposed system, this could lead to frustration of the human agent.

The proposed system provide a means for *adjusting to the needs* of agents. The level of autonomy can be differentiated between teleoperation, shared mode, autonomous mode and all shades in between on the level of direct control of the mobile robot. On this level of direct control, the system adjusts the control mode based on the current performance of each agent and environment. Depending on the IS implementation the system can handle a lot of human neglect. The system adjusts to needs of agents in the way that bad performing agents are supported by the better performing agents which get more influence. But there is no explicit exchange of needs; agents cannot ask for assistance or offer help to other agents. Also, there is neither a shared creation of plans, nor there is a divided responsibility for planning and execution of tasks. Summarizing, the system adjusts to the needs of agents, but this should happen more explicitly.

The system is created with the aim to have a *continuous sharing* of control. Indeed, as explained in Section 2.4.2, the system provides a seamless integration of two motion commands. There are no control swaps since k_H, which determines the human influence, is constant and the efficiencies are changing gradually.

2.6 Formalization of the problem

As can be seen in Section 2.5 the proposed system of Poncela and others makes use of the different capabilities of agents, adjusts implicitly the level of influence of agents to their current needs and it does this continuously. However, it does not help the agents to understand each other more. These properties and previous results (see Section 2.4.3 for previous results) from earlier experiments with the proposed system, are promising.

However, the previous research is only done in two specific domains. In the experiments with automatic wheelchairs all the participants are trained users of automatic wheelchairs. The question arises whether the proposed system is generalizable to other domains of mobile robots. As a first attempt, this research focuses on the implementation of a system with Sphero, which is different from all earlier implementations because Sphero is a robotic ball. The associated research question is:

> RQ1: How well does the system of Poncela and others increase the performance of agents controlling Sphero?

The increase of performance will be measured with two global measures of performance: the duration of a lap driven with Sphero and the length of the lap.

Furthermore, the average of the specific efficiencies will be used to evaluate the performance of the system. More on this topic can be read in Section 3.2.

Because Sphero is never been used before in shared control experiments this is a new way of performing research on shared control. Most research in the domain of shared control is done with expensive and complex robots. Instead, Sphero offers an cheap and simple environment, something that could be useful for further research. This raises the second research question:

> RQ2: To which extend is the control of Sphero a suitable research environment for shared control?

Chapter 3

Methods

In this research an experiment was conducted to answer the research questions. The answer to the first question should been found in the data that comes from the experiment. The answer to the second question appeared while performing the experiment.

The robotic ball Sphero (see Appendix C.1) was used in this experiment. Sphero is made of an opaque polycarbonate shell, has a size of about 7 centimetres, weights about 170 grams, can be controlled remotely through Bluetooth with a range of approximately 20 metres and reaches speeds of around 1 meter per second. The users are able to roll the ball in any direction they want. The mobile robot is able to light the shell into several colours with its internal multi-colered LED's. Just as the LED's all the motors and sensors are positioned inside the polycarbonate shell.

In this chapter the setup of the conducted experiment will be described in Section 3.1. In Section 3.2 the measures that were used to evaluate the performance of a participant are explained. The design and analysis of the results from the experiment are in Section 3.3. Finally, the software used to conduct the experiment is described in Section 3.4.

3.1 Experiment

The experiment was used to determine the effect of the theory of Poncela et al. [22] on the driving performance. This experiment answered the question if the proposed way of shared control could improve the performance of participants in controlling Sphero.

	Duration	**Content**
1	5 min	Sphero test drive and explanation of task
2	7.5 min	Experimental run 1
3	1 min	Environment change
4	7.5 min	Experimental run 2
5	5 min	Post-test questionnaire

Table 3.1: *Experiment design*

3.1.1 Phases in the experiment

The experiment consisted of five phases as can be seen in Table 3.1. In the *first phase*, the participants were introduced to Sphero and the task was explained. They received a tablet that could be used to control Sphero. Participants were told to drive Sphero in a circle that was marked in the environment with three white markers. The completion of a single circle was called a lap. As can be seen in Figure 3.1, a map of the markers in the environment was also presented to the participants in the user interface of the tablet. It was explained that the Sphero could be controlled by tilting the tablet, and that the direction of the tilt was the same as the direction that Sphero would move in. When the participant was ready, (s)he was allowed to drive Sphero through the artificial environment in order to get a feeling about how the robot should be controlled. During this test drive it was explained that the tablet would give a short haptic burst to notice the participant that a goal, one of the three white markers, was reached. Also, it was explained that the participants should rate their ‘feeling of control’, which is explained in the second part of Section 3.1.2, when a lap was completed. It was also explained that Sphero needs more force to start rolling than to keep rolling and that it would be difficult sometimes to start rolling.

The *second phase* is the first experimental run. In this run the participant had to control Sphero. After each lap the shared control condition [1] was randomly changed without the participants knowing after the participant rated their ‘feeling of control’. While rating the ‘feeling of control’ Sphero stopped rolling.

The *third phase* was after the first experimental run. In this phase the experiment leader changed the obstacles in the environment (see the obstacle condition later in this section and also see Figure 3.2 and 3.3). This also allowed participants to pause for a second.

The *fourth* phase was the second experimental run which did not differ from the first experimental run except the obstacle condition. More about the obstacle conditions can be read in Section 3.1.4.

[1] The conditions of the experiment are explained in Section 3.1.4

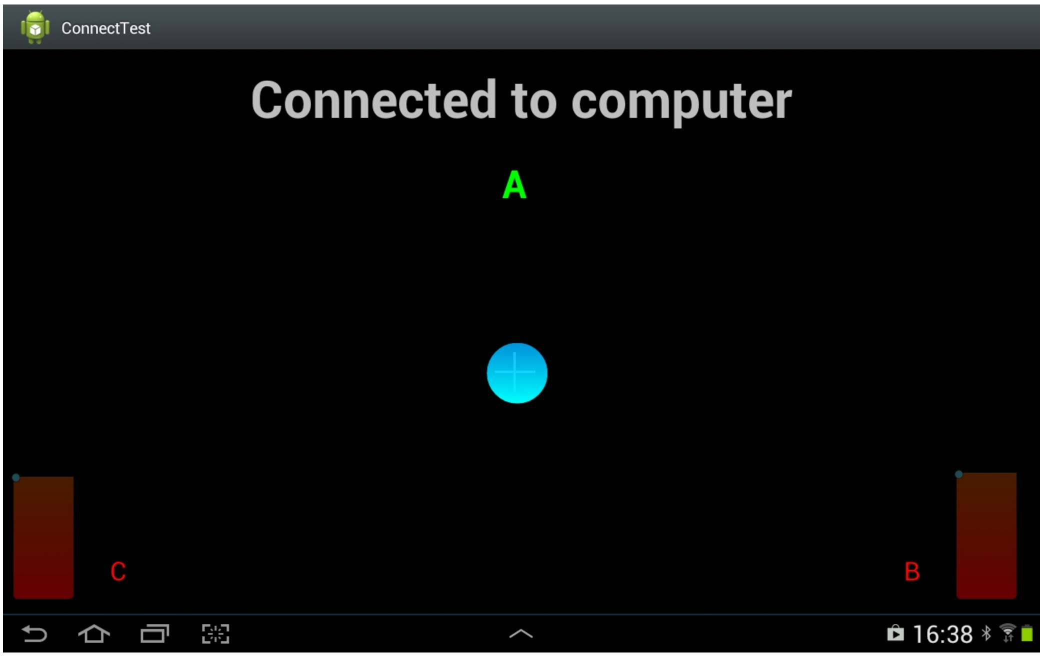


Figure 3.1: *The user interface of the tablet presented information about the markers, the current goal, the current command and enabled the participant to give a rating.*

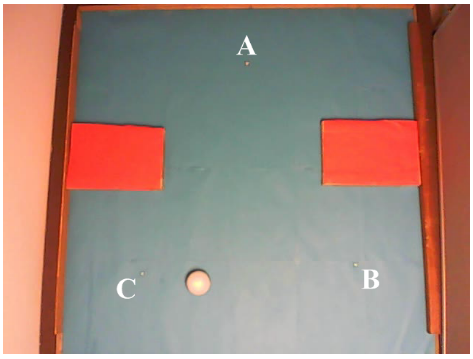


Figure 3.2: *The experimental environment from above with obstacles*

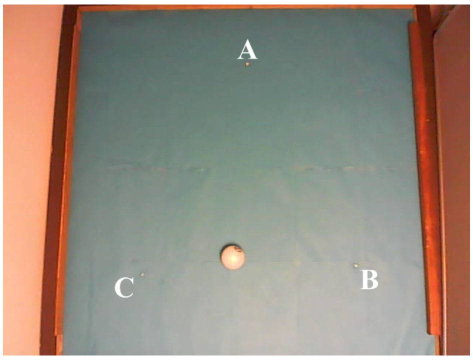


Figure 3.3: *The experimental environment without obstacles*

After the second experimental run the last and *fifth phase* was a short post-test questionnaire that can be seen in Appendix B. This was used to get some general knowledge about the participants and some feedback on how the experienced the experiment. The total experiment took less than 30 minutes.

3.1.2 Tasks for the participants

A single experimental run consisted of two tasks. The *first task* was navigating through the environment while avoiding the wall and obstacles, this was called the driving task. The three fixed markers in the environment were small, easily detectable white points. These markers were waypoints for navigation. The paths between these markers were all about the same length. The participants were instructed to continuously drive from one marker to another in a fixed order. The order was clockwise. On the tablet a map of the three markers was shown, this also indicated which of the markers was the current goal (see Figure 3.1). The start of a new lap would always start with the next marker, after the marker that finished the previous lap. Consequently, the start point changed each time a new lap was started. This meant that there were three different start point for each lap that the participants travelled, probably multiple times, during each experimental run. Each lap was treated as an item in the analyses.

The *second task* for the participants was to rate the 'feeling of control' on a scale from 0 to 7. The participants were clearly instructed to rate how much they felt that they were influencing Sphero's direction. It was explicitly told that the participants should not rate their own performance in driving Sphero around the environment.

3.1.3 Environment and participants

The environment of the experiment was a 1.5 meter square container with a small ridge on the side. Sphero could not escape from this environment. The participants were standing before this environment and could oversee all the details. An image of the environment can be found in figure 3.2.

In total, 23 participants participated in the experiment. The participants were 19 students from the Radboud University Nijmegen or the Hogeschool van Arnhem en Nijmegen and four workers. The average age of the participants was 26. Seven of the participants were male and four of the participants had some earlier experience with controlling robots. The participants were asked if they would like to participate this experiment and did not receive any reward.

3.1.4 Conditions of the experiment

Two conditions were changed during the experiment. The *first condition* was the shared control condition. During the experimental runs the influence of the participants on the actual given commands was changed between four states. In one of the four states there was no shared control between the participants and the computer. The control of Sphero was strictly in hands of the participants, this was called the strictly human condition. Commands that were given with the tablet would immediately be send to the Sphero. This condition functions as the control condition. In the other three conditions the participants shared their control with the computer with Equation (2.1). The human influence parameter in the formula was set to either 0.25, 0.5 or 0.75. The state of the shared control condition was randomly changed after each lap that the participants drove.

The *second condition* differed the presence of obstacles in the environment. Participants would either have no obstacles in the first experimental run and obstacles in the second or vice versa. In the obstacle condition two rectangles of about 30 by 20 centimetres were placed on two of the three lines that the participants had to drive by. The obstacles had a clearly detectable red colour from above. To see an view from above the environment with obstacles placed in it look at figure 3.2. Sphero was only allowed to drive on the floor and was not able to come on the obstacles. When one of the obstacles was moved by Sphero the experiment leader would place the obstacle back in its position. The order of the obstacle condition was counterbalanced between the participants.

3.1.5 Additional experiment for strictly human condition

However, after 18 participants it was noted that there was something wrong in the strictly human condition. In this condition the human command had accidentally,

	First experiment	Additional experiment
Strictly human condition	Invalid	Valid
Shared control with HI = 0.25	Valid	Not measured
Shared control with HI = 0.5	Valid	Not measured
Shared control with HI = 0.75	Valid	Not measured

Table 3.2: *The analysis uses only the valid results from both experiments. HI is an abbreviation of human influence.*

due to a human error, a double velocity compared to the other conditions. The change in velocity was so large that the participants could notice this at the start of each lap and while driving Sphero around. Due to this higher velocity it was harder to precisely reach a goal and to navigate around the environment. Consequently, the data of the conditions where control was shared with a computer could not be compared to the condition where only the participants controlled the robot. For this reason, another small experiment was conducted with the same design but with one exception. After each lap the shared condition was not randomly changed but stayed the same, namely the condition where the participants were fully in control. This condition replaced the strictly human condition from the earlier experiment, as can also be seen in Table 3.2.

There were five participants in this separate experiment and together they drove 108 laps, approximately the same as gathered in the other conditions. Because these laps were gathered by only five participants, in contrast to the laps in the shared control conditions, there are some validity issues concerning the generalizability of this research. These issues will be addressed in chapter 5.

3.2 Measures

From the experimental runs, ‘feeling of control’ ratings and post-test questionnaires several measures could be derived. They were categorized as performance measures from the experimental run and user measures from the ‘feeling of control’ ratings and questionnaires. First, the performance measures will be explained and then the user measures.

3.2.1 Performance measures

The performance measures indicate how well the driving task was executed. In total there were four performance measures.

- The obvious measure for a driving task was the time to complete one lap. The time at which the start point was reached was subtracted from the time that the start point was reached for the second time. The duration was measured in milliseconds.

- A second measure was the length of the path. Notwithstanding the change in start point, the length of the lap stayed the same for all conditions. However, because Sphero was a robotic ball it probably did not take the most efficient path. To measure the deviation from the most efficient path, the length of the trajectory that was made was measured. The length of the lap was the sum over all Euclidean distances between two consecutive observed locations (x_i, y_i) and (x_{i+1}, y_{i+1}) of Sphero:

$$\text{length} = \sum_i \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{3.1}$$

- The third kind of measure were the averages of the local efficiencies. These efficiencies constantly monitor the safety, direction and smoothness of the commands. The efficiencies for the command that was given to Sphero was used for evaluating the performance. For each of the conditions this gave three measures of how well the actual commands were. The equations for the efficiencies can be found in Section 2.4.

- To analyse the sharing of control between the two agents, disagreement was used. This measure was the average of all the differences between the angles of the commands of the two agents. This was useful to estimate how much the agents agree in their commands. The command of agent a was represented as a vector (r_a, ϕ_a), consequently the disagreement between two agents was:

$$\text{difference} = |\phi_H - \phi_{IS}| \tag{3.2}$$

$$\text{disagreement} = \begin{cases} \text{difference} > 180 & 360 - \text{difference} \\ \textit{else} & \text{difference} \end{cases} \tag{3.3}$$

3.2.2 User measures

The user measures from the questionnaires and ratings indicate how the participants experienced the experiment. The *questionnaires* only measured what participants thought of the whole experiment. Multiple choice questions 1, 3, 4, 6 measured if the participants found it easy to drive Sphero around. Question 5 and 9 measured if the participants became frustrated while driving Sphero. Questions 7 and 8 measured how well the participants thought they were driving. Question 2 was just an evaluation of the user interface on the tablet and how helpful it was. Besides the quantitative measures there were some qualitative questions that could be used for evaluation of this experiment as a whole.

The *'feeling of control'* was measured after each lap by the participants as a rating between 0 and 7. This measured how much the participants believed the behaviour of the robot was a result of their own actions. This was an important measure because this sense of agency influences the level of frustration [15, 19], which should be as low as possible to guarantee a good experience with controlling robots. When the participants should rate their feeling of control a progress bar appeared on both sides of the screen. While sliding the progress bar the users got instant feedback on the rating they were currently selecting. When they took their finger of the screen the progress of the progress bar was selected as the rating for the particular lab. One participant complained in the post-test questionnaire that she accidentally four times gave a wrong rating.

3.3 Design and analysis

The experiment should prove if there was a difference between the human condition and the shared control conditions in terms of performance.

3.3.1 Independent and dependent variables

Two qualitative *independent variables*, related to the changes made during the experiment, were used:

- The degree of shared control.
- The presence of obstacles.

The *dependent variables* are the performance measures earlier described in Section 3.2.1. The following seven dependent variables were used:

- The average time to complete a lap. It was expected that this variable was minimal when the human influence on the final command was 0.5, because this setting gave the best results in earlier research [22, 28, 29].

- The length of the trajectory path taken by Sphero to complete a lap. It was expected that this variable would be minimal when the human influence on the actual command was 0.5.

- The third, fourth and fifth dependent variables were the averages of the specific local efficiencies: directness, smoothness and safety. The average was computed over all specific efficiency values of the commands in one lap. It was expected that the variables were maximal when the human influence on the actual command is 0.5.

- The ‘feeling of control’ that is measured during the experiment. It was expected that this variable decreases when human influence decreases.
- The disagreement between the IS and the human agent. It was expected that this variable did not change between the conditions.

3.3.2 Analysis

The laps that were driven by the 23 participants were assumed to be gathered as random samples, from the whole population of driven laps. For this reason, all the laps that were driven were considered as items in the analysis. This resulted in an N of 394, the number of driven laps.

A possible validity issue was that participants that performed better had driven more laps than participants that performed worse. This results in a higher weight for the performance of high performing participants in the final average. Although this increased the average performance, it did not matter because the better performing participants increased the average performance in all conditions.

For each driven lap, the length, duration, efficiency and the rating for ‘feeling of control’ was known. It was also known in what shared condition, obstacle condition and start position the lap was driven. On the 394 valid laps a General Linear Model (GLM) multivariate analysis was done, with a Turkey’s post-hoc test, because there were multiple quantitative dependent variables and multiple qualitative independent variables.

3.4 Software implementation

The software to conduct the experiment consisted of three parts: a vision module, command system and a output module. In this section the details of these implementations will be explained. The pseudocode can be found in Algorithm 1. A flow diagram, with corresponding numbers to the steps described in this section, can be found in Figure 3.4. All the code was written in Java.

Before an experimental run started, the experiment leader explicitly extracted the background of the environment using the software implementation. Ten camera images from above the environment were taken and used to initialize a background image. Also a Bluetooth connection to Sphero and the tablet was set up. In the final initialization step of each experiment, the experiment leader located the goal location by hand with the graphical user interface (see Figure 3.5).

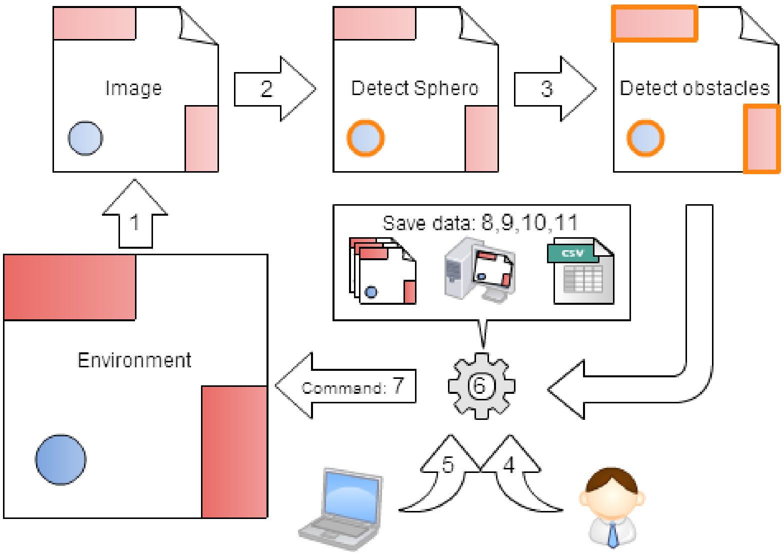


Figure 3.4: *The flow of the software used in the experiment. The numbers correspond to the described steps in Section 3.4.*

3.4.1 Vision

The vision module detected the location of Sphero and the obstacles and borders of the environment. This information was necessary to compute the efficiencies used in equation (2.1) and to implement a layered control system. The vision module used the OpenCV library [11] which was written in C++. To achieve this, the implementation made use of JavaCV [3] which is a JNI interface from Java to other native programming languages. OpenCV had a lot of standard algorithms for image processing. The resulting code for image recognition consisted of the following steps:

1. A new image was retrieved from the webcam and converted to the Hue Saturation Brightness (HSB) colour system and a grey image. The HSB colour system provided a better representation of the colours than the traditional Red Green Blue (RGB) representation because it had a single variable, the hue, that expressed the perceived colour.

   In the implementation, it was also possible to extract the background real-time. This was not used because occasionally Sphero would also be considered as background and this influenced the detection of borders in a later stage. The use of a static background was satisfactory because the light conditions and the orientation of the webcam did not change.

2. If the previous location of Sphero was not known the location of Sphero was interpreted from a grey image. If the previous location of Sphero, and

Algorithm 1 Shared control for Sphero

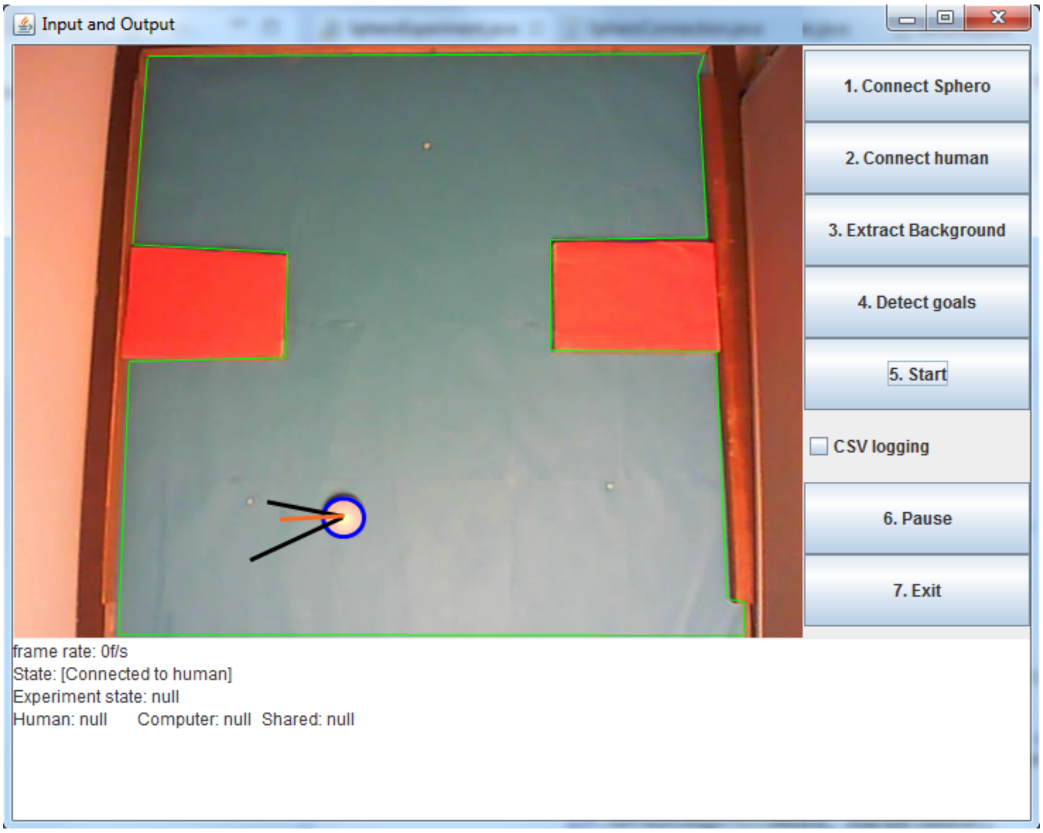
```
Require: Extracted background
Require: Computer connected to Sphero
Require: Computer connected to tablet
Require: goals = {Goal}
  while Experiment is running do
    $i = 0$
    while $i <$ size(goals) do
      Retrieve image
      Detect location
      if location == goal[$i$] then
        Current goal $\leftarrow$ goal[$i++$]
      end if
      Detect obstacles
      Get agent commands
      Create shared command
      if In strictly human condition then
        Send human command
      else
        Send shared command
      end if
      Log data
    end while
    Get rating
  end while
```

thereby its colour, was known, a HSB image was thresholded within a range from the previous colour of Sphero. The hue search range was 30 and the saturation and brightness search range was 75. The resulting black and white image was used to interpret the location of Sphero. A Canny edge detection algorithm [27] was used to detect the edges in the image and a Hough transform was used to detect where possible circles where located. In the case where the previous location was not known the observed circle with the highest probability was used. In the case where the previous location was known the observed circle most similar to the previous observation of Sphero was used.

3. With the new location of Sphero known the obstacles could be detected. Because the whole environment had the same colour except for the obstacles or borders of the environment, the edges of the obstacles were equal to the

edges of the passable environment around Sphero. Five samples of the HSB colours around the location of Sphero were taken. The average of those samples was filtered with a low pass filter to obtain a good approximation of the colour of the surrounding environment. A range with the colour as the centre was used to threshold the background image.

In the resulting black and white image the contours were analysed with the Canny edge detection algorithm [27]. The edge detection algorithm found the borders in the black and white image and represented these as a list of list of points. Each list of points was a polygon that encircles a region and there were no overlaps between the polygons. The appropriate polygon, that comprise the location of Sphero, was selected. With the Ramer-Douglas-Peucker algorithm [23] the points were reduced to 100 to made further computations easier. The polygon represented the border of the environment and since the real size of the environment was known, the scale of the image



```
frame rate: 0f/s
State: [Connected to human]
Experiment state: null
Human: null      Computer: null  Shared: null
```

Figure 3.5: *GUI for the experiment leader. The retrieved image of the webcam is shown with some annotations on it. Also there are several buttons that the experiment leader can use to control the experiment.*

could be calculated. This was necessary for the translation of pixel distances to real world distances.

3.4.2 Obtaining commands

After all the necessary information was extracted from the image the commands from both agents were obtained. The following steps were taken:

4. The human command was generated with a tablet that was running on Android (see Figure 3.1 for the user interface of the tablet). An newly developed application was used that, among other things, sensed the orientation of the tablet and sent this via Bluetooth to the computer. The tablet could sense its orientation with an internal hardware gravity sensor. The sensed orientation (see Appedix C.2) consisted of gravity values along three axis. A scaled gravity along the x and y-axis of the tablet was used to sent to the computer. The x and y-axis gravities were scaled by the total gravity, that also includes the z-axis. The total gravity was approximately $9.81m/s^2$. As a result, the tablet would send two doubles from -1.0 to 1.0, these represented a vector that could be used as a command. The computer would translate the received values to a suitable command for Sphero.

5. For a computer command an implementation of the layered control system proposed by Brooks [5] with three distinct layers was used. The first layer was only active if Sphero had been close to the border, while not moving, although the command velocity was high. In other words, Sphero was stuck behind an obstacle. If this happened a command with a relative high velocity away from the border would have been given. The second layer was an avoid layer that was only active when the Sphero was 0.3 seconds away from crashing into an obstacle. When this happened the command would be zero if the current speed was above 20 cm/s and else a command that was directly avoiding the wall was generated. The third layer always gave a command that was directed at the current goal. The velocity of this command depended on three factors: the distance between Sphero and the goal, the current speed and the angle between the current heading of Sphero and the angle to the goal location.

Note that the layered control system was not a state-of-the-art control system for mobile robots. However, this research investigated shared control and not control systems for robots. For that reason, the improvement of performance was important and not just the performance of the control system. Furthermore, if a simple control system could improve performance, a more complex state-of-the-art control system certainly could.

6. After receiving the commands from the human agent and IS, a shared command was created by an implementation of equation (2.1). This implementation used the information that was gathered by the vision module. The amount of human influence was set by the program depending on the shared control condition the experiment was in. Effectively, both command vectors were multiplied by the human influence constant (or one minus the human influence constant, in case of the IS) and their efficiencies. The newly created vectors were added together to get the shared command.

3.4.3 Sending command and logging data

After the receipt and creation of the commands, motion vectors were sent to Sphero. Also, several kinds of data were logged to the hard drive. A real-time output was shown to the experiment leader.

7. Depending on the state of the shared control condition either the human command or the shared command would be sent to Sphero. To this end, an open source unofficial application programming interface (API) [12] in Java for Sphero was used. Each command was sent as a direction and velocity to Sphero which would execute these commands immediately.

8. A message logging system logged the errors, warnings and information messages that were generated during the experiment to a single logging file.

9. A videologger logged the image from the webcam to a video file. It was possible to also include annotations of the current knowledge to this file.

10. A datalogger logged the timestamp, the current location and radius of Sphero, the three commands with all efficiencies, the current goal, if a goal was reached, in what condition the experiment was and the current travelled distance of Sphero. With the information if a goal was reached or not, the time and duration of the path were calculated from the data that was been logged by the datalogger.

11. A image with annotation was already created while certain parts of the image were recognized. This image was shown to the experiment leader (see Figure 3.5). Also the raw data about the location and colour of Sphero, the colour of the ground was shown to the experiment leader in a text box.

12. Finally, all the images would be released to prevent a memory leak. This was very important because JavaCV is a wrapper around the C++ library OpenCV. Although Java does not use pointers, C++ does, and the data behind the pointer could either be removed too early or too late. In the first case an

exception for unauthorized memory access is given and in the latter case a memory leak occurs. Both are not wanted in a software program and for that reason the pointers to the images should be released at the proper time.

Chapter 4

Results

In this section the results from the experiment will be shown. First, the durations of the laps are treated. Secondly, the differences in length will be shown. After that, the efficiencies of both commands and shared conditions will be presented. Fourth, the ratings of ‘feeling of control’ are compared for all the conditions. Fifth, the disagreement between the two commands will be shown. Lastly, the user measures from the questionnaire will be displayed.

4.1 Duration

The duration of the lap varied between the conditions. The shared control condition had a small effect ($p < 0.01$, $\text{eta}^2 = 0.037$) on the duration of the lap as can be seen in Appendix A.2. On average, the duration of a lap was longer in the strictly human condition than in the shared control condition with HI = 0.5 ($p < 0.05$) and HI = 0.75 ($p < 0.01$). It took participants 20.0% less time in the shared control condition with HI = 0.5 to complete a lap, compared to the strictly human condition.

4.2 Length

Just as the duration, the shared control condition had an effect ($p < 0.001$, $\text{eta}^2 = 0.071$) on the length of the lap. The length of the strictly human condition is longer ($p < 0.001$) than the shared control conditions with a human influence (HI) of 0.5 and a HI of 0.75. There is a reduction of the length with 29.6% from the condition where the human is in full control to the shared control condition with HI = 0.5. Also the shared control condition with HI = 0.5 is shorter ($p < 0.05$) than HI = 0.25. There is no difference between the condition where the human is strictly in control and the shared control condition with HI = 0.25. A graph of the length of the laps can be found in Appendix A.3.

Efficiency	**Human**	**Intelligent System**
Smoothness	0.51 (SD=0.06)	0.53 (SD=0.05)
Directness	0.82 (SD=0.05)	0.98 (SD=0.03)
Safety	0.56 (SD=0.04)	0.62 (SD=0.03)
Average	0.63 (SD=0.03)	0.71 (SD=0.03)

Table 4.1: *Human efficiency compared to Intelligent System efficiency*

Efficiency	**Human condition**	**Shared condition (HI=0.5)**
Smoothness	0.46 (SD=0.05)	0.55 (SD=0.05)
Directness	0.84 (SD=0.04)	0.91 (SD=0.03)
Safety	0.57 (SD=0.03)	0.61 (SD=0.03)
Average	0.63 (SD=0.03)	0.69 (SD=0.03)

Table 4.2: *Efficiencies during the human condition and the shared condition with HI = 0.5*

4.3 Efficiencies

Both human and IS gave continuous commands during the experiment. Overall the efficiency of the IS commands were smoother, more directed to the goal and safer (all $p < 0.001$) than the commands of the human agents (see Table 4.1).

Between the conditions there were big differences in smoothness ($p < 0.001$, $\text{eta}^2 = 0.389$), directness ($p < 0.001$, $\text{eta}^2 = 0.582$) and the safety ($p < 0.001$, $\text{eta}^2 = 0.214$) of the commands. A graph of the efficiencies can be seen in Appendix A.1. As can be seen in Table 4.2, all the specific efficiencies and the average efficiency were higher ($p < 0.001$) in the shared control condition with HI = 0.5 compared to the strictly human condition. The average efficiency of the given commands improved with 11.4% from the human condition to the shared condition with HI = 0.5. All the shared control conditions had higher ($p < 0.001$) specific efficiencies than the human condition.

There were also some significant differences in efficiency within the shared control conditions. For smoothness the shared condition with HI = 0.75 was higher ($p < 0.05$) than HI = 0.25. For directness all the differences between all the conditions were significant ($p < 0.001$). There was a stable trend that indicates that a decrease of human influence increases the directness of the command. The shared control condition with HI = 0.25 and HI = 0.5 were safer ($p < 0.05$) than HI = 0.75. Also the average efficiency of the shared control condition with HI = 0.75 was lower ($p < 0.01$) than the other shared control conditions with HI = 0.25 and H = 0.5.

The presence of obstacles had a small effect on the directness efficiency (p

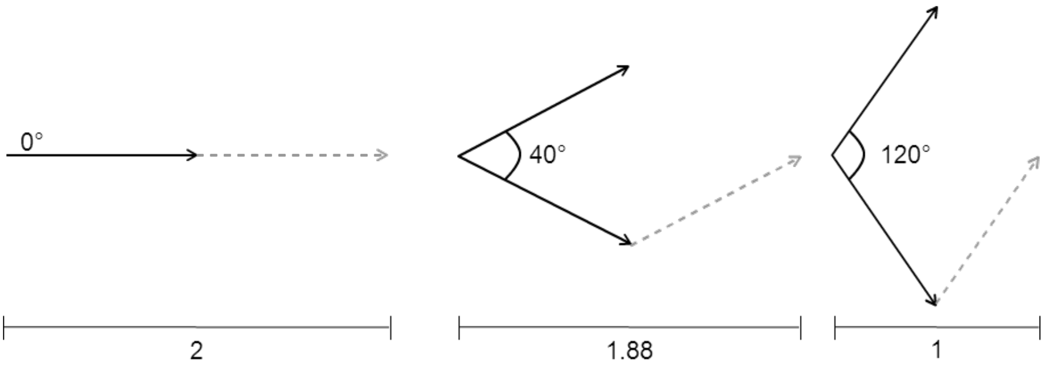


Figure 4.1: *The velocity of the resulting command gets lower when the disagreement between two added commands get higher.*

< 0.01, $\text{eta}^2 = 0.028$) and the safety efficiency (p = 0.001, $\text{eta}^2 = 0.040$). The directness and safety were lower in the condition with obstacles.

4.4 Rating of 'feeling of control'

There was an medium effect ($p < 0.001$, $\text{eta}^2 = 0.066$) of the shared control condition on the ratings that were given for the participants 'feeling of control'. The graph of this effect can be seen in Appendix A.4. The ratings for the shared control conditions with HI = 0.5 and HI = 0.75 were significantly higher than the condition with HI = 0.25 (for both $p < 0.01$) and the human full condition ($p < 0.01$ and $p < 0.05$ respectively).

4.5 Disagreement

On average, the angle of the command of a human differed 40 degrees (SD = 11) with the angle of the command of the IS. The disagreement between the human agent and IS was in the condition with HI = 0.25 higher ($p < 0.001$) than in the other conditions. No further differences were found.

With figure 4.1 can be seen that when the disagreement gets higher, the result of adding two factors together gets lower. With a disagreement of 40 degrees the result of adding two commands together is 6% smaller compared two the addition of two commands with no disagreement.

Question	**Rating**
1. It was easy to control Sphero	4.42 (SD = 1.213)
2. The on-screen instructions were helpful	3.75 (SD = 2.069)
3. Learning to control Sphero was easy	5.46 (SD = 1.587)
4. I noticed that the robot sometimes not listening to me	5.71 (SD = 1.367)
5. The experiment stressed me	2.77 (SD = 1.751)
6. New users will find this controlling system easy to use	4.92 (SD = 1.816)
7. I had problems with avoiding the wall	4.29 (SD = 1.429)
8. I had problems with avoiding the obstacles	3.79 (SD = 1.382)
9. It was fun to use Sphero	6.21 (SD = 0.779)

Table 4.3: *Efficiencies during the human condition and the shared condition with HI = 0.5*

4.6 Post-test questionnaire

An average rating of 4.42 out of 7 was given on the statement "It was easy to control Sphero". This correlated with the statement "Learning to control Sphero was easy" with an average rating of 5.46 ($p < 0.001$) and the statement "New users will find this controlling system easy to use" with an average rating of 4.92 ($p < 0.001$). This correlation was expected as explained in the methods section. The statement "I had problems with avoiding the wall" with average 4.29 and "I had problems with avoiding the obstacles" with average 3.79 are also connected ($p < 0.01$). The other questions did not correlate but in Table 4.3 the averages can be seen.

In the post-test questionnaire eleven participants noted that the most difficult was to control the speed of Sphero, either when they started rolling or when they had to slow down to reach a goal. Five participants noted that something changed in the control during the experiment. But actually two of them were in the human condition with five participants were actually nothing changed.

Chapter 5

Discussion and conclusion

This thesis presents the results of an experiment in shared control with use of the robotic ball Sphero. This section will (1) discuss the results, (2) draw conclusions from this and (3) state some recommendations for the future.

5.1 Discussion

The first thing to note is that the results from the human condition were gathered with only five participants instead of 18 participants for the other conditions. The number of driven laps were the same, but it could be that those five participants are not representative for the whole population of Sphero users. The results should be judged with this information in mind.

The performance measures showed clearly that there is an improvement of performance between the human condition and the shared conditions. All the efficiencies, the length and duration of the lap improve. This indicates that users benefit from using the shared control algorithm. The shared condition with HI = 0.5 is preferable because the performance is always better or equal to the other conditions. Between the shared control conditions there are less variations than expected, the influence of humans does not rapidly decrease or increase the performance. The improvement on the average efficiency (11.4%) is comparable to improvements the earlier described implementations of the system in Section 2.4.3. Along with the improved duration and length of the lap it can be said that this shared control theory improves the performance in driving Sphero.

It is remarkable that the smoothness of the shared control condition is higher than the separate smoothness's of the human motion command and the IS motion command. This indicates that the average of both commands is better, and that perhaps the motion commands of both agents are approximations of the best command.

The ratings that are given showed a counter-intuitive result. Participants rate their influence on the final command higher when they are in a shared condition then in the condition where humans where in full control. Perhaps the message that they should rate their influence on Sphero's behaviour was not clear enough, although it was said multiple times. Another cause could be that five participants in the human conditions have no reference to other conditions, and for this reason those results cannot be compared to the other conditions. There was a negative correlation of -.433 ($p < 0.001$) between duration and rating, and -.442 ($p < 0.001$) between length and rating. These correlations support the idea that participants were rating the performance of Sphero and not their influence on Sphero.

There is no measure of frustration to compare the conditions with each other, but the post-test questionnaire shows that participants have fun (6.21 out of 7.0) and are not stressed (2.77 out of 7.0). In earlier literature there is been enough research to what causes frustration and the feeling of control, that is measured in this experiment, is one of the main causes [19]. However, during the experiment participants noted that Sphero is sometimes not moving when they desired it to move, this causes visible stress but, as can be seen in the results of the post-test questionnaire, the effects are limited.

There was a great difference in velocity between the human condition and the shared conditions. The velocity was lowered in the shared control conditions by the efficiencies, but also by the disagreement between the agents. The efficiency lowered the command velocity of the command with 31% (average efficiency = 0.69) and the disagreement lowered it with 6%. Something peculiar in the theory is that the shared control commands only can have a lower velocity than the separate commands from both agents. There is no reinforcement of the shared control command when both agents give the same efficient command, with a high velocity.

5.2 Conclusion

In this thesis the possibilities of the system proposed by Poncela et al. [22] for sharing control between two agents are investigated. It is found that it meets three of the four requirements on shared control that were extracted from earlier papers on shared control:

- The system provides a good way to use the capabilities of multiple agents while sharing control.
- It is good at adjusting continuously to the current performance of the agents.
- The system is very helpful when there is a need for continuously sharing control without abrupt switches in control.
- But it does *not* increase the mutual understanding for either of the agents for each other and for the environment.

The research done with Sphero indicates that the proposed system of Poncela and others effectively improves the human performance on controlling Sphero (RQ1). Together with earlier research [22, 28, 29] it suggests that the system could be a robust and flexible way to share control between agents. The research also shows that Sphero is a suitable environment for shared control (RQ2).

5.3 Recommendations

Based on the results presented in this thesis there are some recommendations for similar experiments with Sphero on shared control in the following section. In Section 5.3.2 questions that arise from this research are posited.

5.3.1 Recommendations based on the research in this thesis

There are some comments on this research. First, a more accurate measure of the 'feeling of control' of the participants is necessary for conclusions about how participants were experiencing the unknown sharing of control. A possible solution for this problem could be to inform the participants anyway about the sharing of control so they are more focussed on influences of other agents on the final behaviour of the mobile robot. Secondly, there are no comparisons made between the performance of the IS and the performance of both agents together in the shared control condition. This should be done to be sure about the probable improvement when sharing control.

In general, the Sphero robotic ball is a simple and cheap way of investigating the possibilities of shared control. Other research into shared control is done for practical reasons, for example to improve automatic wheelchair performance, and these are often in domains wherein participants need foreknowledge and have assumptions. The control of Sphero is based on simple vector commands and the control method is easily explained to participants. This research used a camera to track the position of Sphero and this is not scalable to larger environments. The camera will not be able to capture the whole environment from above. However, in the new API of Sphero there are ways to track the position of the ball based on its internal sensors but the question is how precise this tracking is. Thereupon, it is recommended to use Sphero to create a research environment for shared control.

5.3.2 Further research

Based on this research there are four recommendations for the field of shared control and specific for the shared control theory developed by Poncela et al in 2009.

First, this theory scales the velocity of motion commands by efficiencies that are based on the angles of the motion commands. To compute the efficiencies the velocity of the commands are ignored but they are also important. For example, the safety measure can benefit from the use of velocities. Consequently, the performance of the system could be increased even further if the velocity of the motion commands are also considered.

Secondly, until now the theory is always applied to two agents. Perhaps, the use of efficiencies to scale the motion commands can also be a good way to improve the performance of a single agent. Furthermore, this theory is perfectly scalable to more than two agents, this also could give interesting results for shared control.

Third, this research uses the same three specific efficiencies that are also used by Poncela et al. [22] and Urdiales et al. [28, 29]. For different domains these efficiencies are maybe not applicable, and maybe other efficiencies are also applicable to the domain of controlling of Sphero. More research should be done to find out what specific efficiencies could be used and what their influence is on the resulting behaviour.

Lastly, this theory only focusses on the sharing of control of direct motion commands. Higher cognitive tasks such as planning or selecting goals are not considered. Perhaps the same idea of sharing with efficiencies could be used on these higher cognitive tasks, which would be very promising, but this should be investigated.

The system for shared control by Poncela and others is already promising for many applications, but since the system is relative new it can improve even more.

Chapter 6

Bibliography

[1] David a. Abbink, Mark Mulder, and Erwin R. Boer. Haptic shared control: smoothly shifting control authority? *Cognition, Technology & Work*, 14(1):19–28, November 2011.

[2] P Aigner and B J McCarragher. Modeling and constraining human interactions in shared control utilizing a discrete event framework. *IEEE Transactions on Systems Man and Cybernetics Part A Systems and Humans*, 30(3):369–379, 2000.

[3] Samuel Audet. https://code.google.com/p/javacv/.

[4] Andrea Bauer, Dirk Wollherr, and Martin Buss. HumanâĂŞRobot Collaboration: a Survey. *International Journal of Humanoid Robotics*, 05(01):47, 2008.

[5] Rodney A. Brooks. A Robust Layered Control System. *ieee journal of robotics and automation*, 2(I):14–23, 1986.

[6] D J Bruemmer, D a Few, R L Boring, J L Marble, M C Walton, and C W Nielsen. Shared understanding for collaborative control. *IEEE Transactions on Systems Man and Cybernetics Part A Systems and Humans*, 35(4):494–504, 2005.

[7] D J Bruemmer, J L Marble, and D D Dudenhoeffer. Mutual initiative in human-machine teams. *Proceedings of the IEEE 7th Conference on Human Factors and Power Plants*, pages 7–22–7–30, 2002.

[8] J Connell and P Viola. Cooperative control of a semi-autonomous mobile robot. *Proceedings IEEE International Conference on Robotics and Automation*, 2:1118–1121, 1990.

[9] T Fong, S Grange, C Thorpe, and C Baur. Multi-robot remote driving with collaborative control. *Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication*, 50(4):699–704, 2003.

[10] F Galán, M Nuttin, E Lew, P W Ferrez, G Vanacker, J Philips, and J Del R Millán. A brain-actuated wheelchair: asynchronous and non-invasive Brain-computer interfaces for continuous control of robots. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, 119(9):2159–69, September 2008.

[11] Willow Garage and Itseez. http://opencv.org/.

[12] Nicklas Gavelin. https://github.com/nicklasgav/Sphero-Desktop-API, 2012.

[13] MA Goodrich. Experiments in adjustable autonomy. *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, 3:1624 – 1629, 2001.

[14] Michael A Goodrich, Timothy W Mclain, Jeffrey D Anderson, and Jacob W Crandall. Managing Autonomy in Robot Teams : Observations from Four Experiments. pages 25–32.

[15] Pim Haselager. Did I Do That? BrainâĂŞComputer Interfacing and the Sense of Agency. *Minds and Machines*, December 2012.

[16] E Hollnagel, J Pariès, David Woods, and J Wreathall. *Resilience Engineering in Practice*. 2004.

[17] Behzad Khademian and Keyvan Hashtrudi-Zaad. Dual-User Teleoperation Systems: New Multilateral Shared Control Architecture and Kinesthetic Performance Measures. *IEEE/ASME Transactions on Mechatronics*, 17(5):895–906, October 2012.

[18] Dan Kortenkamp, R. Peter Bonasso, Dan Ryan, and Debbie Schreckenghost. Traded Control with Autonomous Robots as Mixed Initiative Interaction. *AAAI Technical Report*, 04:89–94, 1997.

[19] J.L. Marble, D.J. Bruemmer, and D.a. Few. Lessons learned from usability tests with a collaborative cognitive workspace for human-robot teams. *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*, 1:448–453, 2003.

[20] R R Murphy R R Murphy. Human-robot interaction in rescue robotics. *IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews*, 34(2):138–153, 2004.

[21] Marcelo R. Petry, Antonio Paulo Moreira, Rodrigo a. M. Braga, and Luis Paulo Reis. Shared control for obstacle avoidance in intelligent wheelchairs. *2010 IEEE Conference on Robotics, Automation and Mechatronics*, pages 182–187, June 2010.

[22] a Poncela, C Urdiales, E J Perez, and F Sandoval. A New Efficiency-Weighted Strategy for Continuous Human/Robot Cooperation in Navigation. *IEEE Transactions on Systems Man and Cybernetics Part A Systems and Humans*, 39(3):486–500, 2009.

[23] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256, November 1972.

[24] Paul Scerri, David Pynadath, Milind Tambe, Admiralty Way, and Marina Rey. Adjustable Autonomy in Real-world Multi-Agent Environments. 2001.

[25] Nathan Schurr and Time-critical Domains. Improving Adjustable Autonomy Strategies for Time-Critical Domains. 2009.

[26] TB Sheridan and WL Verplank. Human and computer control of undersea teleoperators. 1978.

[27] Satoshi Suzuki and KeiichiA Be. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, April 1985.

[28] C Urdiales, M Fernández-Carmona, J M Peula, U Cortés, R Annichiaricco, C Caltagirone, and F Sandoval. Wheelchair collaborative control for disabled users navigating indoors. *Artificial intelligence in medicine*, 52(3):177–191, July 2011.

[29] C. Urdiales, J. M. Peula, M. Fdez-Carmona, C. Barrué, E. J. Pérez, I. Sánchez-Tato, J. C. Toro, F. Galluppi, U. Cortés, R. Annichiaricco, C. Caltagirone, and F. Sandoval. A new multi-criteria optimization strategy for shared control in wheelchair assisted navigation. *Autonomous Robots*, 30(2):179–197, October 2010.

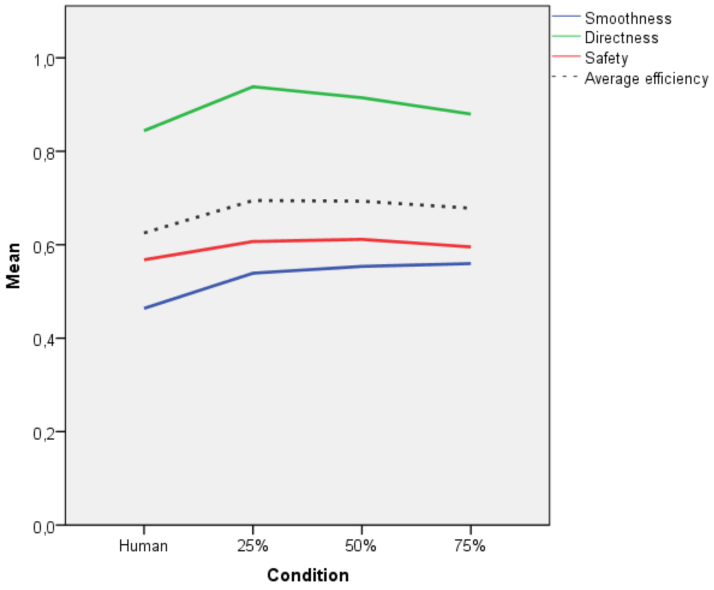
[30] Haoyong Yu, Matthew Spenko, and Steven Dubowsky. An Adaptive Shared Control System for an Intelligent Mobility Aid for the Elderly. pages 53–66, 2003.

[31] Stéphane Zieba, Philippe Polet, Frédéric Vanderhaegen, and Serge Debernard. Principles of adjustable autonomy: a framework for resilient humanâĂŞmachine cooperation. *Cognition, Technology & Work*, 12(3):193–203, October 2009.

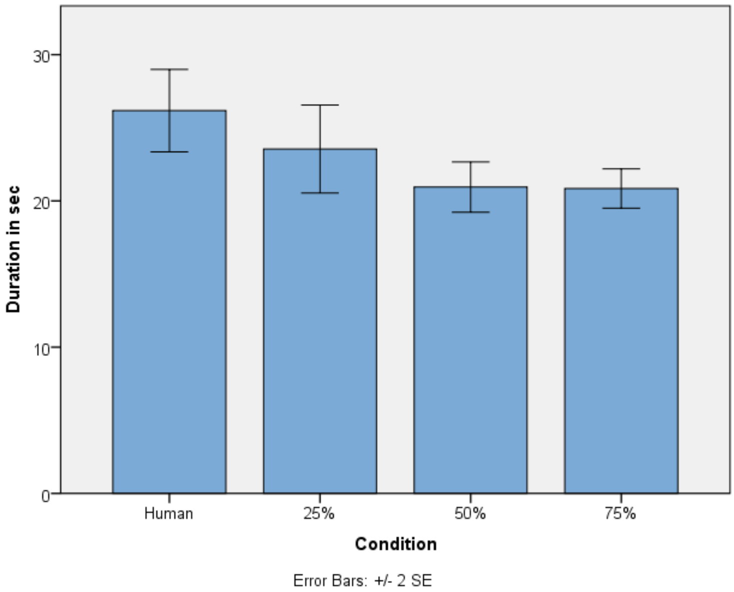
Appendix A

Graphs

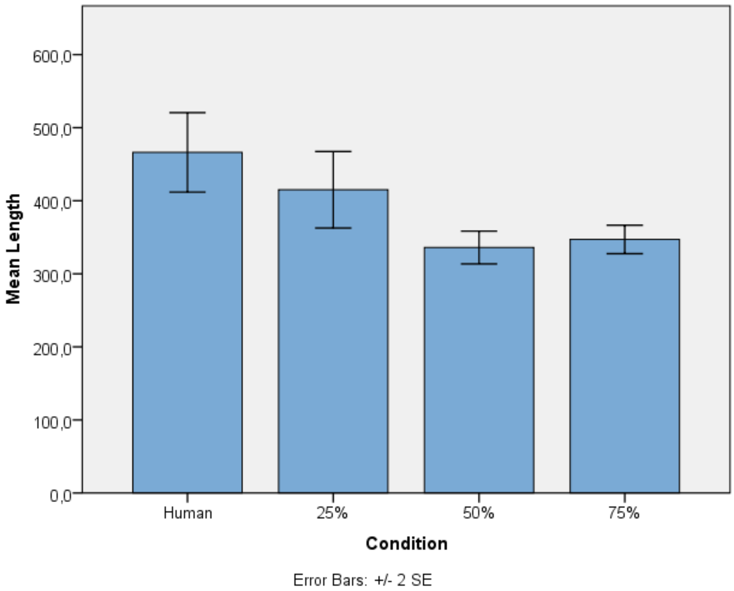
A.1 Efficiencies



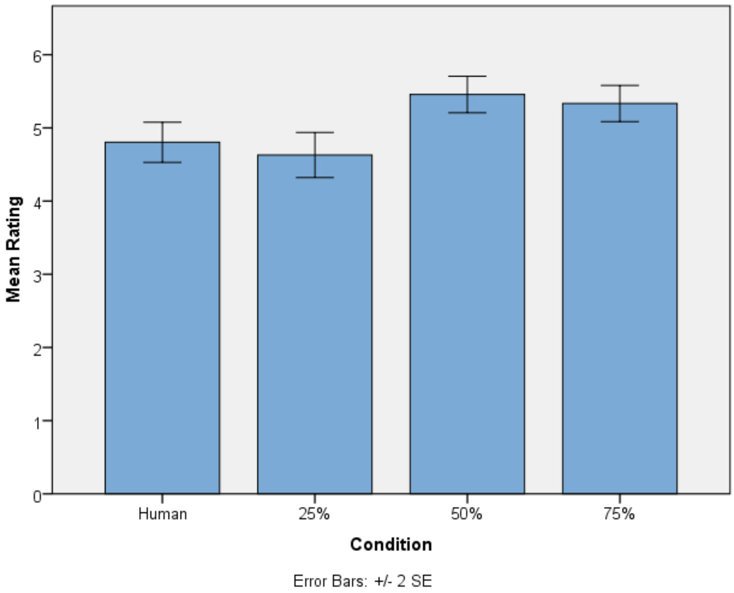
A.2 Duration



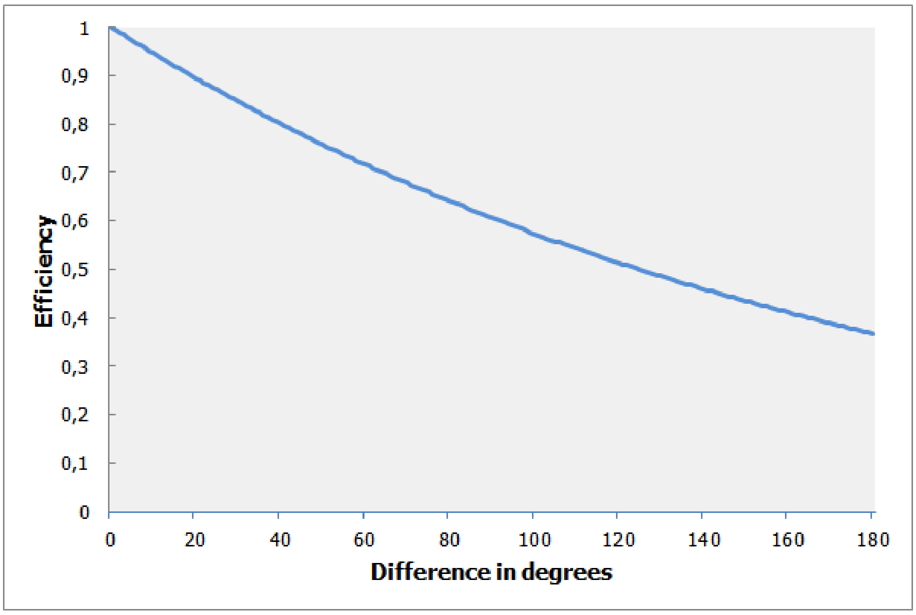
A.3 Length



A.4 Rating



A.5 Specific efficiency vs. angle



Appendix B

Post-test Questionnaire

POST-TEST QUESTIONNAIRE

Subject ID: Test date:

PERSONAL INFORMATION

1. What is your age? ______ year
2. What is your gender? Male / Female
3. What is your daily occupation?

ABOUT ROBOTS

4. Do you have earlier experience with robots?

THE EXPERIMENT

5. Please rate your agreement with the following statements from 1 (disagree) to 7 (agree).

1. It was easy to control Sphero	1	2	3	4	5	6	7
2. The on-screen instructions were helpful	1	2	3	4	5	6	7
3. Learning to control Sphero was easy	1	2	3	4	5	6	7
4. I noticed that the robot sometimes not listening to me	1	2	3	4	5	6	7
5. The experiment stressed me	1	2	3	4	5	6	7
6. New users will find this controlling system easy to use	1	2	3	4	5	6	7
7. I had problems with avoiding the wall	1	2	3	4	5	6	7
8. I had problems with avoiding the obstacles	1	2	3	4	5	6	7
9. It was fun to use Sphero	1	2	3	4	5	6	7

USER FEEDBACK

6. What was the MOST DIFFICULT about controlling Sphero?

7. What was the EASIEST about controlling Sphero?

8. Did you notice anything special about the experiment?

9. Any other remarks?

Appendix C

Images

C.1 Sphero

C.2 Orientation of Android devices