Pull-E- The assistive shopping trolley

Reka Hajnovicsova

Master of Science Thesis

Delft University of Technology

Biomechanical Engineering

Pull-E- The assistive shopping trolley

Master of Science Thesis

For the degree of Master of Science in BioMedical Engineering at Delft University of Technology

Reka Hajnovicsova

August 29, 2017

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by Distributed Organisms B.V. Their cooperation is hereby gratefully acknowledged.

Delft University of Technology
Department of
Biomechanical Engineering

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Pull-E- The assistive shopping trolley

by

Reka Hajnovicsova

in partial fulfillment of the requirements for the degree of

Master of Science BioMedical Engineering

Dated: August 29, 2017

Chair: ______________________
Dr.ir. D. Abbink

Supervisor at university: ______________________
MSc Sarvesh Kolekar

Supervisor at company: ______________________
Ir. A.C. van Rossum

External expert: ______________________
Dr. ir. Winfred Mugge

Abstract

The aging society puts an increasingly heavy demand on the health care system. Elderly often have difficulties in mobility, reducing opportunities for exercise and social interaction that activities like running errands or shopping normally offer.

This thesis proposes a novel concept for an assistive motorised shopping trolley, called Pull-E. Based on interview results with elderly (n=40) the Pull-E should be able to carry heavy loads (~20 kg) and still move easily and lightly while pulled over flat ground, curbs, and even stairs; and should ideally also be able to move autonomously, without being pulled. It should therefore have capabilities to balance itself, climb the stairs, carry items, and resist perturbations.

First, a small proof-of-concept of the Pull-E was built, using tri-wheels which are separately powered by motors, that allow assistive stair-climbing when the Pull-E is pulled horizontally. In order to balance itself the tilt angle of the trolley needs to be measured. Therefore, the Pull-E was equipped with a gyroscope and an accelerometer. From this prototype one of the most challenging aspects emerged: stabilising the trolley.

The next step was to choose a suitable control system. Since humans can adapt to the movement of other humans, climb the stairs, and resist perturbations of other humans, the human inspired control theories have been investigated.

Based on the literature search the risk-aware control was the most promising one. In case of risk-aware control a value function is provided to the system, by which it can avoid the undesirable states. By using a value function it does not require the calculation of exact trajectories and therefore, it becomes computationally more efficient, than the other human inspired control methods. In addition it learns and adjusts its control parameters by which it can adapt to different situations.

In order to test its efficacy, it was implemented and tested in a simulation of the Pull-E, where a planar projection of the tri-wheeled actuated trolley was dynamically simulated. Force inputs could act on the trolley by using the mouse, as well as pre-determined sinusoids acting on its centre of mass. The simulation showed the risk-aware control method already resulted in instabilities when no perturbations affected the system. So far, the risk-aware

control method was tested only for zero order systems. Therefore an adaptation of the risk-aware control method was made to allow the 2nd order system, by including angular velocities in the state and modifying the value function. This control method is able to balance the assistive trolley for forces that cause lower angular perturbations than 15 degrees.

In the future this control method needs to be improved to become more robust against perturbation. The prototype needs to be rebuilt with stronger motors, and the control system needs to be tested on the prototype. Finally, this prototype will be evaluated when human is in the loop.

To conclude, Pull-E can facilitate elderly's longer independence, but still a lot of work needs to be done before it becomes a viable product.

Table of Contents

List of Figures

List of Tables

Preface

This master thesis is the final assignment for my degree of Master of Science in Biomedical Engineering at the faculty of Mechanical, Maritime and Materials Engineering at Delft University of Technology. To get a balance between theoretical research and a practical assignment, I made the decision to do this project at a research company: Distributed Organisms B.V. in Rotterdam. Although the main focus of this company is artificial intelligence and robotics with applications in CleanTech and GreenTech, I had the opportunity to work on a robotic project related to elderly care.

Acknowledgements

I would like to thank my daily supervisor Anne van Rossum for his supervision and guidance during the writing of this thesis. I have learned a lot from him for which I am very grateful. I would also like to thank my other colleagues, Bart van Vliet for his help with programming and building the prototype. Special thanks go to Marc Hulscher for providing me enormous help with the understanding of different control methods and for his valuable comments on this thesis. Besides Distributed Organisms B.V. I have had much help from people at Delft University of Technology, especially my supervisor David Abbink. His supervision and continuous encouragement was valuable for my research, as well as the advice of Sarvesh Kolekar, who are all part of my thesis committee. Of course I am also grateful for the university as a whole during the part of my study that led to this research. I would also like to thank all my friends and relatives who supported me during this research and my whole academic career, both in the Netherlands and abroad. Unfortunately I cannot mention all of them by name, but I love all of them and appreciate their support. Last but not least, I would like to thank my fiance, Gergo Berci for his love and unfailing support. Thank you.

Delft, University of Technology — Reka Hajnovicsova
August 29, 2017

"The best way to predict your future is to create it." — *Abraham Lincoln*

Chapter 1

Introduction

Elderly have to face several problems caused by (1) limited mobility, (2) memory loss, (3) deteriorating health, and (4) loneliness [7]. Depending on the severity of their problems after a given stage they have to move to a nursing home. Moving out of their own home to a nursing home is truly unpleasant. In addition it is expensive by which the aging society puts an increasingly heavy demand on the health care system. Therefore, the necessity of assisting elderly with new technological devices has increased. From the perspective of elderly, assistive robotic devices provide the opportunity to stay longer at home [7].

The first group of problems is due to (1) limited mobility. In this case the elderly person can choose from several mobility assistive devices. The second group of problems, which is caused by (2) memory loss, can be solved by safer homes and reminders. The third group of problems is due to the (3) deterioration of health, which could be slowed down by exercising. The easiest form of exercising is walking e.g. running errands. Lastly, the fourth group of problems which is one of the most unpleasant problems the elderly faces is (4) loneliness. The social contact with family members, the shop assistant or other customers can highly reduce the level of (4) loneliness. The problems of elderly with their possible solutions are listed below in table 1-1.

Table 1-1: Four groups of problems of elderly and solutions

Group	**Problem**	**Solution**
1	**Limited mobility**	Mobility aids
2	**Memory loss**	Safer homes, reminders
3	**Deteriorating health**	Exercising, walking
4	**Loneliness**	Socialization

The currently available devices mainly focus on the first two groups of problems and therefore, the aim was to provide a solution for the third and fourth problem.

As previously mentioned, running errands provides the opportunity of (3) exercising and the social contact with the shop assistants reduces the level of (4) loneliness. Therefore, the aim is to design a device, which can facilitate walking and socialization.

The first step in designing such a device was to understand what are the difficulties the elderly is facing, and what is the real reason for elderly people to run fewer errands as they are getting older. To find the reason for this phenomenon multiple senior people have been interviewed. (The interview questionnaire is provided in the appendix A.)

The results learned that the majority of the elderly people do their shopping twice a week, at the same time each week and that is the time when they meet their friends. Therefore, when they decrease the frequency of going to the stores they decrease the frequency of meeting their friends, and they will become more lonely. In addition, from the questionnaire it turned out that most of the elderly like to walk, although when carrying groceries they need to stop multiple times to take a rest before getting home. If they would not need to pull the trolley they would not need to stop to have a rest. Above all, pulling the trolley up the stairs is highly demanding for them. In some cases, they are not able to pull it up the stairs and they have to wait hours until someone helps them, which can be truly frustrating. This is a big problem which needs to be solved.

To conclude, there is a high need for an assistive shopping trolley which can move without being pulled and is able to climb the stairs.

1-1 Currently available mobility assistive devices

In the following section, several assistive mobility devices available for the elderly are going to be analyzed. They are going to be analyzed according to the criteria, whether they are able to climb the stairs while carrying packages. These devices are the canes, crutches, walkers, wheelchairs, and mobility scooters also shown in the figure 1-2.

1. **Canes**
   Canes, or walking sticks are used to increase stability. The user can choose between a simple cane or a multi-feet cane. Multi-feet walking sticks provide even greater support. There are robotic walking sticks as well, such as SmartCane [8], which supports the user while avoiding obstacles or the PAMM (Personal Aid for Mobility and Monitoring) which assists the user and monitors the user's vital signs [9]. Walking sticks can help in climbing stairs, although carrying bags is complicated.

2. **Crutches**
   The crutches, in contrast to the walking sticks, provide direct body support. Unfortunately, they are not so popular because they cause unnatural gait. Carrying bags while climbing stairs with crutches is really complicated, just as with canes.

3. **Walkers**
   The walkers are pushed by the user and give a support during walk using the patient's own remaining locomotion capability. Walkers can increase the confidence and the sense of safety, by which the user's level of activity increases [10]. Robotic walkers are available as well e.g. LEA, which in addition to helping with walking, reminds people

to eat, drink and take their medicine. To sum up, walkers are able to carry bags, but they are not able to climb the stairs. In addition, the user has to lean a bit forward to push the walker, which is not so healthy and makes them feel old.

4. **Wheelchairs**
In the case of mobility problems, the most widely spread solution is the manual wheelchair. Unluckily the continuous use of a wheelchair can cause serious health problems because of the sitting position. Therefore, the use of augmentative devices is highly encouraged if possible. To sum up, the wheelchairs have some storage, but they are not able to climb the stairs.

5. **Mobility scooters**
The popularity of mobility scooters has increased in the last decade. Although the continuous use of the mobility scooter can cause serious health problems just like the wheelchair. The mobility scooters have some room for shopping bags, but they are not able to climb the stairs.

Table 1-2: Mobility assistive devices

				
1. Canes	2. Crutches	3. Walkers	4. Wheelchairs	5. Mobility scooters

To conclude, to the best of my knowledge the current mobility assistive devices are not able to climb the stairs while carrying packages.

1-2 Pull-E- A conceptual prototype for an assistive shopping trolley

This thesis proposes a novel concept for an assistive motorised shopping trolley, called Pull-E. The name Pull-E ("Puli") comes from the name of a Hungarian dog called Puli. The Puli is a guard dog valued for its energy and determination, which is a result of its sheepdog history.

Based on interview results with elderly (n=40) the Pull-E should be able to carry heavy loads (~20 kg) and still move easily and lightly while pulled over flat ground, curbs, and even stairs; and should ideally also be able to move autonomously, without being pulled. It should therefore have capabilities to balance itself, climb the stairs, carry items, and resist perturbations. In addition, western societies have difficulties to accept robots and therefore Pull-E will look like a regular shopping trolley.

First, a small proof-of-concept of the Pull-E was built, using tri-wheels which are separately powered by motors, that allow assistive stair-climbing when the Pull-E is pulled horizontally.

1-2-1 Wheels

The aim was to choose a wheel type which has excellent obstacle-crossing ability. The list of the wheel types, which have been compared is provided in the Appendix B. After the comparison, the exterior planetary wheel (also called tri-wheels) has been chosen, since this type of wheel has great obstacle-crossing ability. It is able to negotiate obstacles by turning around the wheel frame. The steering of the robot can be realised by differential motion of the wheels [1].

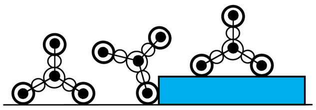

Figure 1-1: Exterior planetary wheel crossing an obstacle [1]

The first prototype consisted of a pair of exterior planetary wheels, two motors, and buttons to control the trolley. It was able to climb the stairs, although it was not able to balance or resist perturbations.

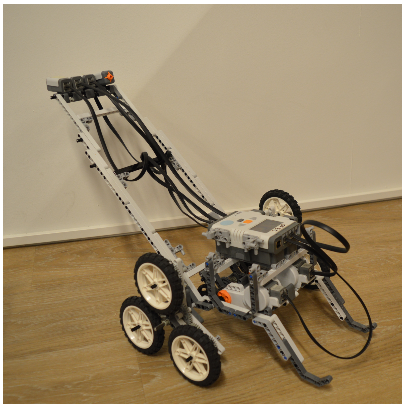

Figure 1-2: First prototype with exterior planetary wheels, motors, and buttons to control it.

Click on the link to watch the youtube video: First prototype
The exterior planetary wheels seemed to be a good choice, until the prototype was made from Lego. Unfortunately, power these wheels, several gears or drive belts are needed. The use of gears or drive belts requires precise alignment, what might complicate the building.

1-2-2 Force sensors

Since, most of the elderly is struggling with new technological devices the Pull-E has an intuitive control interface. Pull-E can be guided by turning and moving its handle just like in case of a bike handle, which is shown in the figure 1-3.

For this force sensors are used in the handle. The alignment of force sensors is similar to the alignment of the force sensors used in the robotic walker called CAIROW [11]. CAIROW has been designed for patients with Parkinson disease. The user does not need to pull the

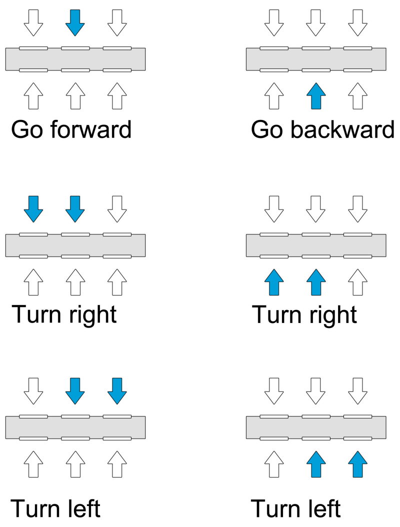

Figure 1-3: Alignment of the force sensors.

shopping trolley anymore, just place his hand on the handle and apply some force in the desired direction.

The force sensors measure an analog voltage, which needs to be converted to have the forces in Newton. The range of the analog voltage is from 0 to 1023, which is mapped to 0V to 5V. As a first step the incoming value has to be mapped to 0-5V. The voltage is calculated by using the following formula [12]:

$$V^* = \frac{V_{cc}R}{R + \frac{C}{80f}} \tag{1-1}$$

where V^* is the measured voltage, V_{cc} is 5V, R is the resistance, which in this case is $10K\Omega$, C is the conductance, and f is the force. The number 80 is a specific value of the force sensor, which depends on the measurable range. This value can be found in the data sheet of the sensor (Appendix C). Therefore the force can be calculated by

$$f = \frac{CV}{(Vcc - V^*)80R}. \tag{1-2}$$

To sum up that is how the values coming from the force sensors are converted to force in Newtons. These values are then sent to the software Processing . These values are stored in separate variables, which ones are responsible for moving forward, backward, and turning left, right. If the value responsible for moving forward (f_f) minus the value responsible for moving backward (f_b) is bigger than a threshold value (θ) then a force is applied on the virtual trolley in forward direction

$$f_f - f_b > \theta. \tag{1-3}$$

If the value responsible for moving backward (f_b) minus the value responsible for moving forward (f_f) is bigger than a threshold value (θ) a backward pulling force is applied on the virtual trolley

$$f_b - f_f > \theta. \tag{1-4}$$

To conclude, the force sensors were a good choice. They are flat and easy to use. In addition the bike handle type intuitive control interface could be made by using them. For future project the smaller size of the force sensors might be enough in order be able to make a smaller handle.

1-2-3 Inertial measurement unit (IMU)

In order to balance itself the tilt angle of the trolley needs to be measured. To measure the tilt angle the GY-521 IMU board containing Invensense's MPU-6050 has been used. This board contains an accelerometer and a gyroscope. The data received from the accelerometer is noisy on a short time scale. The data received from the gyroscope drifts on longer time scales. Therefore, the data coming from these two sensors need to be fused for greater accuracy. This can be done by using a Kalman filter or by using a simple complementary filter. Since this board contains a motion processing unit (MPU) from Invensense, this chip performs the data fusion on the IMU chip and there is no need for the implementation of a fusion algorithm. Although, the exact details of the data fusion algorithm are not published.

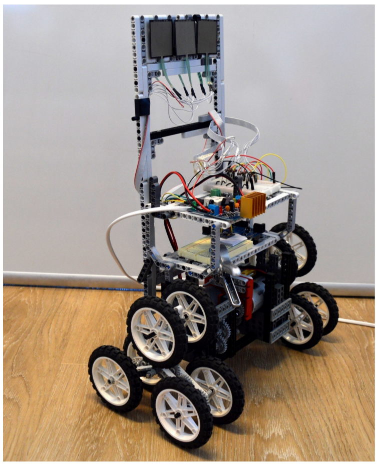

Figure 1-4: The second prototype equipped with force sensors and an IMU board.

Click on the link to watch the youtube video: Prototype with sensors
To summarize, the performance of the IMU board is satisfying within its price category. The precision is around 1 degree. Important to mention that the cables connecting the IMU board and the Arduino board should be as short as possible and preferably shielded. When the cables are long or not shielded noise can enter the system and the IMU board turns off to protect itself. To avoid it the easiest solution is to arrange all the cables to avoid the region of the IMU board and place the IMU board as close as possible to the Arduino board.

To sum up, the force sensors measure the input from the user and the angle is measured by the IMU board. Then the measured values are sent to the simulation. The simulation based on the received data, calculates the motor acceleration. This value is sent back to the Arduino board. Then the Arduino board powers the motors and the motors turn the wheels. Then new sensor values are read, which ones are sent to the simulation. The simulation calculates new motor accelerations, which ones are sent to the Arduino. Finally the Arduino sends a new value to the motors.

After testing all the hardware components a real size prototype has been built. This prototype has bigger wheels and a different structure. The real size prototype is not able to climb the

Figure 1-5: Real size prototype with sensors.

stairs yet, because more powerful motor would be needed. The power of the current motors is 16W. The motors were connected to a Lego gearbox to increase the output torque. The problem was that if the motors were spinning too fast the Lego parts melted at the connection to the metal parts. For future, whether stronger motors are needed to avoid the need of a gearbox or motors with a built-in metal gearbox.
Click on the link to watch the youtube video: Real size prototype with sensors

To conclude, all the features has been tested on the smaller prototypes, which means that the real size trolley could work as well with proper motors.

From the prototypes one of the most challenging aspects emerged: stabilising the trolley. Therefore the next step is to choose a suitable control system, which is the main focus of this thesis.

1-3 Outline of this thesis

This thesis is organised in the following way: in Chapter 2, an overview of traditional control principles and criteria needed for the control of the Pull-E are given. Since humans can adapt to the movement of other humans, climb the stairs, and resist perturbations of other humans, in Chapter 3 the human inspired control theories have been investigated. The most promising human inspired control method (risk-aware control) is described in more details in Chapter 4. So far this control method was tested only for zero order systems. Therefore in Chapter 5 an adaptation of this control method is made to allow the 2nd order system, by including angular velocities in the state and modifying the value function. The thesis is concluded in Chapter 6, which discusses the main contribution of this thesis and suggestions for future work.

Chapter 2

Overview of traditional control principles

Several demands need to be fulfilled by the controller of the assistive trolley. The first criterion is that the robotic shopping trolley should be (1) able to move. Secondly, the robotic trolley should be able to (2) balance, in order to be able to climb the stairs. Thirdly, it should (3) be resistant to perturbations. Therefore the controller has to 'notice' noise and uncertainties and react to them. In addition to these three criteria the robot needs to (4) adjust its control depending on the input of the user. The elderly person steers the trolley by pushing the handle in the right direction. In addition, when the user pulls harder the trolley should go faster. Therefore the trolley always needs to adjust its control based on the received input. In addition to receiving input from the user the trolley has to be able to (5) sense its environment. To sense the input from the user and the environment sensors are needed. The signal received from the sensors always contains some noise and uncertainties. Therefore the control system should be able to (6) cope with uncertainties. The robot has to (7) be able to make predictions to react faster to the environment e.g. to detect in forward when the user wants to stop. Finally, it would be desired to have a model with (8) low computational complexity, in order to reduce hardware requirements and the computation time.

In this chapter a short overview of different control principles is given in order to decide which ones are the most suitable to control the trolley. This will be done in pairwise comparisons. At first we compare the open loop and closed loop control methods, then the adaptive and robust control methods and finally, the stochastic and deterministic control principles.

2-1 Open-loop control

A straightforward control method is an open-loop control (figure 2-1). In this control method, the controller is independent of the result. Open-loop control in the case of the trolley would mean that it receives a command from the user, but it has no sensors. In the block diagram below the command from the user is the setpoint. The setpoint is the input of the

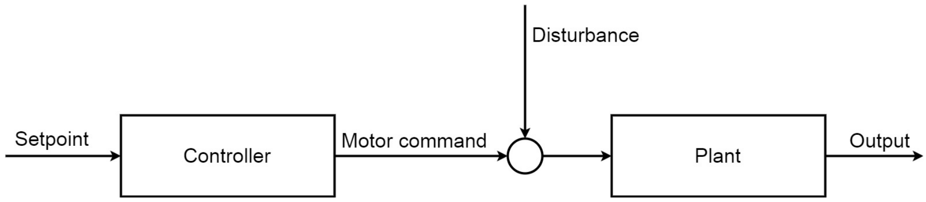

Figure 2-1: Block diagram of open loop control. The setpoint is the command from the user, which is the input of the controller. The controller in case of the trolley is an Arduino board. Based on the received input the controller generates a motor command. The motor command is sent to the plant which in this case is the trolley. This results in change in the state. The motor command might be disturbed before reaching the plant.

controller. The controller is the Arduino board which calculates the motor command. The motor command is sent to the plant, which in this case is the trolley.

When the user's command, for example, is to go forward, the trolley will try to move forward even if there is a wall. Therefore the open-loop system requires supervisory attention from the user. In addition, the trolley has to receive a feedback about its current position, to keep its balance. In the case of the open-loop control no feedback will be provided and therefore the trolley will not maintain its balance.

To conclude, the open-loop control principle is not suitable for the trolley because the position of the trolley and the environment is changing all the time and the trolley has to react to it.

2-2 Closed-loop control (PID)

The next control principle is closed-loop control. Closed loop control contrasts open loop control by using feedback to control the system. The block diagram of closed-loop control is shown in figure 2-2. The controller calculates an error signal based on the input signal and the received feedback. The biggest advantage of the closed loop system is that it enables the trolley to adjust its motor command, based on the received feedback. In contrast to open-loop control, the trolley by using its sensors can sense if there is a wall, and adjust its control. In addition, it can sense the position and balance itself.

The most widely used control mechanism using feedback loops is the proportional-integral-derivative controller (PID controller). The basic mechanism is that the desired state and the measured state is given to the controller. The controller calculates the error signal, which is the difference between the desired state and the measured state. Then the error is corrected by using proportional (P), integral (I) and derivative (D) terms. The reason for its high popularity is that most of the PID controllers can be adjusted by using some tuning rules. Unfortunately, PID is effective only within limited conditions [13]. When there are too big differences in the state of the plant or in the operating conditions, the PID cannot adapt since its proportional (P), integral (I) and derivative (D) parameters are fixed.

To conclude, the trolley needs to adjust the motor commands based on the feedback received from the sensors, and therefore closed-loop control is needed.

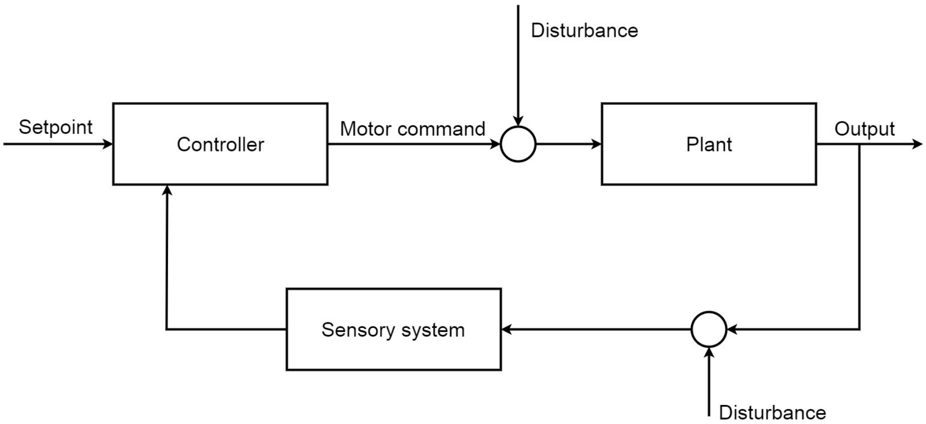

Figure 2-2: Block diagram closed-loop control. The controller based on the input and the received feedback signal generates a motor command. The motor command is sent to the plant, which results in the change of the state. This change is sensed by the sensory system. In case of the trolley the sensory system consists of several force sensors and an inertial measurement unit (IMU) board. The output of the sensory system is fed back to the controller.

Table 2-1: List of the criteria needed for the control of the trolley.

	Criteria	**Open-loop control**	**Closed-loop control**
1.	move	✓	✓
2.	balance	—	✓
3.	resistant to perturbation	—	—
4.	reacts on the input	✓	✓
5.	sense the environment	—	✓
6.	copes with uncertainties	—	—
7.	makes predictions	—	—
8.	low comp. complexity	✓	✓

2-3 Adaptive control

The adaptive controller modifies its controller gains and parameters based on the received feedback. The block diagram of adaptive control is shown in figure 2-3.

The setpoint is sent to the controller. The controller sends a motor command to the plant. The output of the plant is fed back into the controller. In case of adaptive control, in addition to the closed loop a separate block , called meta controller is added to the system. The meta controller adjusts the parameters based on the desired state, the internal copy of the motor command, and the output of the plant. Finally, the adjusted parameters are sent to the controller.

In case of the trolley, when the weight of the carried mass changes the controller needs to adapt to these changes.

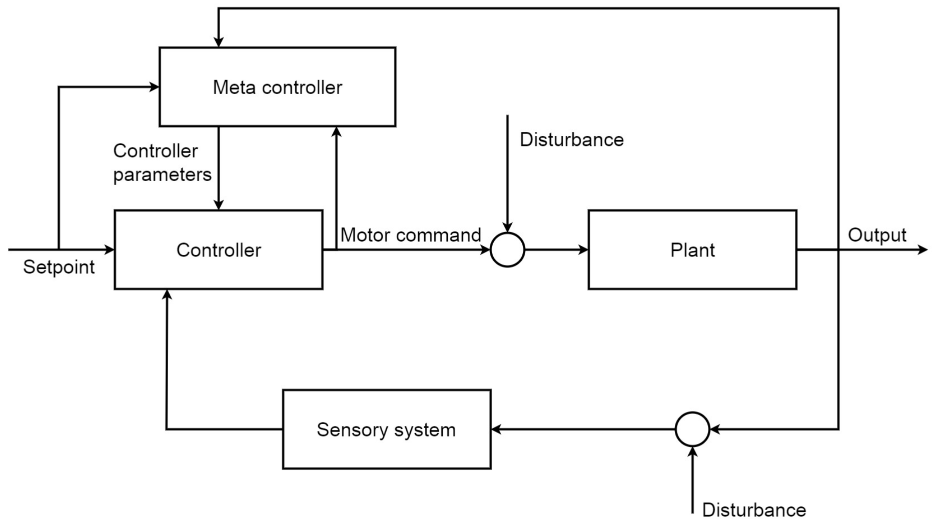

Figure 2-3: The block diagram of adaptive control. The controller generates a motor command based on the input, the received feedback signal, and the received parameters. The motor command is sent to the plant. The change in the state of the plant is fed back to the controller and to the meta controller. The meta controller adjusts its parameters based on the desired state, the internal copy of the motor command, and the output of the plant. Then the adjusted parameters are sent to the controller [2].

2-4 Robust control

The next control principle is the robust control. The aim of the robust control is to function properly even when the parameters of the system are uncertain or disturbances enter the system [14]. In addition to the noise entering the system, there are uncertainties in the model of the plant.

When a controller is robust it means that there are different demands on the controller, rather than a different control structure. Since the robust control can be open-loop control or closed-loop control as well no block-diagram is included at this section.

The robust controller assumes that certain variables are unknown, but bounded [14]. Given a bound on the uncertainty, the controller can produce motor commands, which meet the control system requirements in each case. Unfortunately, some performance may have to be sacrificed in order to guarantee that the system meets the requirements in each case [15]. An everyday example of this type of control could be the safety system of a nuclear power plant because it needs to work properly even in the worst case scenario.

To conclude, in the case of the trolley, it has to be able to adjust its parameters, but it also has to be robust to perturbations. Therefore the adjustment of parameters and robustness needs to be balanced in the control system.

Table 2-2: List of the criteria needed for the control of the trolley.

	Criteria	Adaptive control	Robust control
1.	move	✓	✓
2.	balance	✓	✓
3.	resistant to perturbation	—	✓
4.	reacts on the input	✓✓	✓
5.	sense the environment	✓	✓
6.	copes with uncertainties	—	—
7.	makes predictions	—	—
8.	low comp. complexity	✓	✓

2-5 Deterministic control

The deterministic system has no stochastic element, which means that there is no randomness involved in generations of the motor command. Therefore, the system will always produce the same output for the same initial state. This means that, the input and output relation of the system is fully determined.

The deterministic control describes the way the controller generates the motor command, rather than a different control structure. This means that, the deterministic controller can be used on different control systems. Therefore, no block diagram is included.

Unfortunately, the resulting state variables in the deterministic control can be drastically different from the state variables of the stochastic control e.g. when a lot of noise is entering the system [4].

2-6 Stochastic control

Stochastic control has stochastic elements which deal with uncertainties of the state. These uncertainties are modelled as probability distributions. The probability distributions are combined in the controller. The stochastic controller calculates the time path of the state variables. The control method can perform the task with minimum cost while taking into account the noise [15]. As already mentioned the resulting stochastic state variables can be drastically different from the deterministic control state variables [4]. The stochastic control, just like the deteriministic control describes the way the motor command is generated and not the control structure. This means that, the stochastic control can be used on different control systems. Therefore, no block diagram is included here neither.

The trolley has to be able to cope with uncertainties, and therefore stochastic control is needed. On the other hand, since stochastic control uses probability distributions not only deterministic values, it increases the computational complexity.

Table 2-3: List of the criteria needed for the control of the trolley.

	Criteria	Deterministic control	Stochastic control
1.	move	✓	✓
2.	balance	✓	✓
3.	resistant to perturbation	✓	✓
4.	reacts on the input	✓	✓
5.	sense the environment	✓	✓
6.	copes with uncertainties	—	✓
7.	makes predictions	—	—
8.	low comp. complexity	✓	—

2-7 Summary

In the previous section, a short overview of different control principles has been given in order to decide which ones are the most suitable for the control of the trolley. This has been done in pairwise comparisons. At first we compared the open-loop and closed-loop control methods, then the adaptive and robust control methods, and finally the stochastic and deterministic control principles.

The first criteria is that the robotic shopping trolley should (1) be able to move. By using only an open-loop control the trolley would be able to move. On the other hand, using open-loop control would not result in keeping the trolley in (2) balance, which is needed to climb the stairs. Therefore a closed-loop control is needed for the control of the trolley. Additionally, the fifth criterion is that the trolley has to (5) sense its environment for which a closed-loop control is needed.

The next two control principles which have been compared were the adaptive and robust control. The third criterion for the control of the trolley is to (3) be resistant to perturbations, which can be achieved by a robust control. on the other hand, the fourth criterion is that the trolley has to (4) adjust its control according to the input given by the user, which can be achieved by the adaptive control. Therefore the controller should find the right balance between adjusting its parameters and being robust to perturbations.

Finally, the controller should be stochastic, due to (6) uncertainties entering the system, which should be taken into account. On the other hand, since stochastic control uses probability distributions not only deterministic values, it increases the (8) computational complexity.

The human brain can merge all the described principles. Furthermore, humans (7) can make predictions and react faster to the environment, which was the seventh criterion for the control mechanism of the trolley. Unfortunately, as already mentioned in the introduction the human movement control is still not fully understood by scientists. Since humans are outperforming robots in all of these criteria, in the next chapter several theories about the human movement control will be analysed.

Table 2-4: List of the criteria needed for the control of the trolley.

	Criteria	**O-l. c.**	**C-l. c.**	**Adp. c.**	**Rob. c.**	**Det. c.**	**Sto. c.**
1.	move	✓	✓	✓	✓	✓	✓
2.	balance	—	✓	✓	✓	✓	✓
3.	resistant to perturbation	—	—	—	✓	✓	✓
4.	reacts on the input	✓	✓	✓✓	✓	✓	✓
5.	sense the environment	—	✓	✓	✓	✓	✓
6.	copes with uncertainties	—	—	—	—	—	✓
7.	makes predictions	—	—	—	—	—	—
8.	low comp. complexity	✓	✓	✓	✓	✓	—

Chapter 3

Human-inspired control methods for balancing the trolley

In the previous chapter, several control principles have been described to decide which ones are the most suitable for the control of the trolley. To decide the most suitable control principles, at first, the criteria for the control of the trolley have been defined.

Since humans can adapt to the movement of other humans, climb the stairs, and resist perturbations of other humans, the human inspired control theories are going to be investigated in this chapter. At first, the optimal sensorimotor control is going to be explained. At second, the equilibrium point hypothesis is going to be described. Then the active inference theory is going to be analyzed. Finally, the risk-aware control will be introduced.

3-1 Optimal sensorimotor control

The aim of the optimal motor control is to reach the desired state at minimal cost. Therefore the optimal motor control includes a cost function, which is the function of the state and the control variables. This means that the movement is determined by the cost function. The main advantage of optimal motor control is that it requires only a performance criterion (the goal) to calculate all the movement details [16].

3-1-1 Feed-forward control

The open-loop control, or feed-forward control, is the simplest control structure. In the case of the open-loop control, the brain sends a motor command to the musculoskeletal system, which results in movement, as depicted in figure 3-1.

In the following the feed-forward control is going to be described in more detail. The block diagram of the more detailed forward control can be seen in figure 3-2.

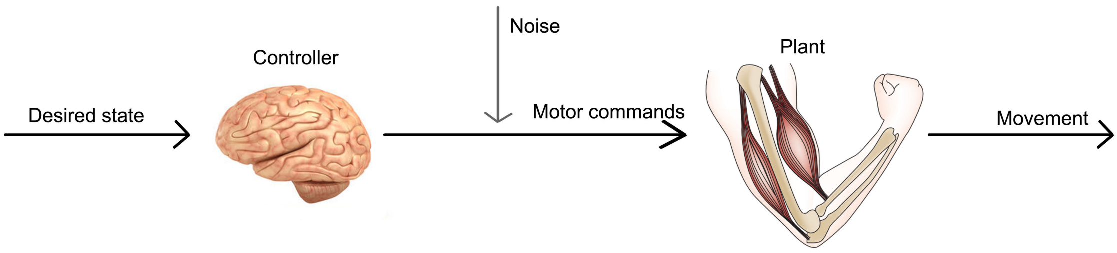

Figure 3-1: Scheme showing feed-forward motor control. The brain based on the received task generates motor commands. The motor commands are sent to the plant, which results in movement. Before the motor commands reach the plant noise might disturb the system. Compare the scheme of feed-forward motor control with the scheme of open-loop control in Chapter 2, section 2.1. observe that this block-diagram has the same structure. The controller is the brain, the plant is the human body and the change in the plant is the movement of the human body.

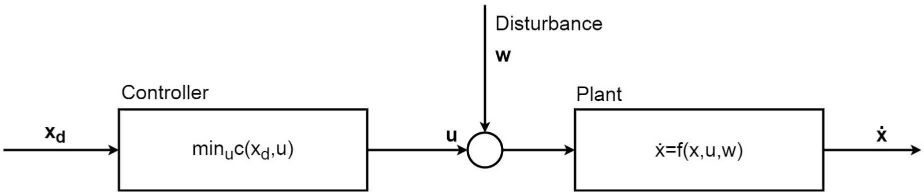

Figure 3-2: The block diagram of feed-forward motor control. The desired state x_d is the input of the controller. The controller calculates the cost function, which has been described in equation 3-1. The aim of the controller is to minimize the cost function. Then the controller sends a motor command u to the plant. The motor command before reaching the plant might be disturbed by noise w. The motor command is sent to the plant. This results in a change of the state $\dot{x}$, which can be described by the equation 3-2.

The input is the desired state denoted by x_d. The controller calculates the cost function $c(x_d, u)$

$$min_u c(x_d, u) \tag{3-1}$$

where x_d is the desired state of the system, and min_u is the minimization of the motor command u. The aim of the controller is to minimize the cost function. The controller sends the motor command u to the plant, which results in a change of the position $\dot{x}$

$$\dot{x} = f(x, u, w) \tag{3-2}$$

where x is the state of the system and w is the disturbance affecting the plant. Since there is no feedback about the resulting state, when the system is perturbed it cannot compensate the effect of disturbance. In optimal feed-forward control the behaviour is averaged over multiple repetitions of the task while minimizing the cost[16].

To conclude, the feed-forward model can optimize the cost functions, although when the system is disturbed it cannot correct the resulting position [16].

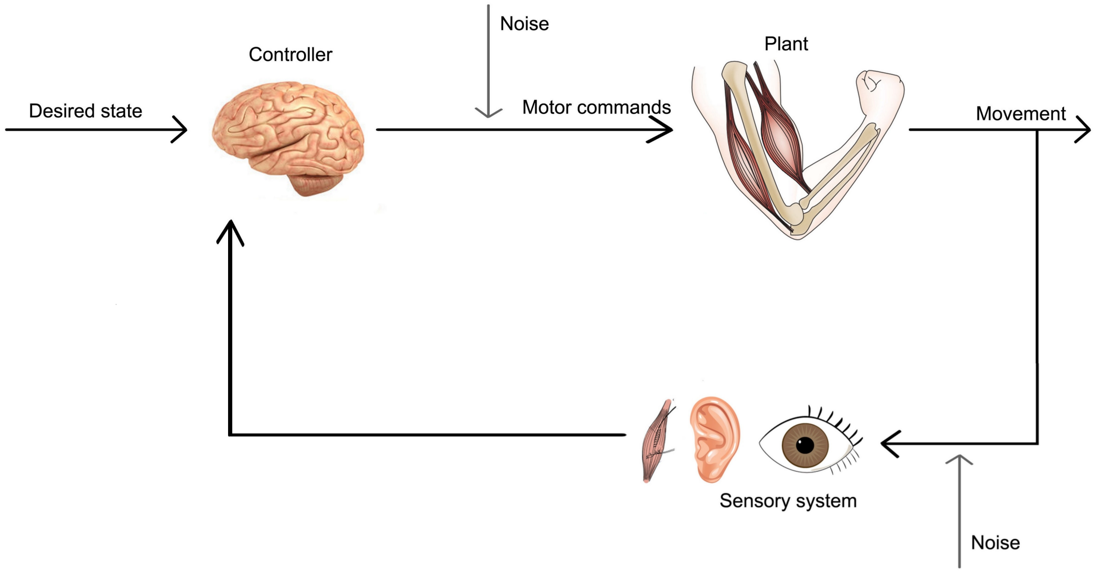

Figure 3-3: Scheme showing feedback motor control. The brain based on the received task generates motor commands. Before the motor commands reach the plant noise might disturb the system. The motor commands are sent to the plant, which results in movement. Before the state is measured noise might enter the system.The state of the system is measured by the sensory system. Finally, the measured state is fed back to the brain.

3-1-2 Feedback control

In the case of the feedback control, the brain sends motor commands to the muscles in order to reach the desired state. The states of the muscles are fed back to the brain by the sensory system and the motor command is adjusted.

Therefore, in the case of feedback control, the brain can improve its solution based on its feedback in contrast to the feed-forward control when the same behaviour is repeated. The basic structure of the feedback control is shown in figure 3-3.

The more detailed control scheme of the feedback control can be seen in the figure 3-4. The input is defined by the desired state x_d. The aim of the controller is to minimize the cost function over u

$$min_u c(x_d, x^*, u) \tag{3-3}$$

where x_d is the desired state of the system, x^* is the measured state, and u is the motor command.

After the motor command u is sent to the musculoskeletal system the system can be perturbed w. The plant kinetics depend on the motor command, and the perturbation

$$\dot{x} = f(x, u, w). \tag{3-4}$$

The plant kinetics are the same as in the open-loop control, although the received motor command u is different, which is based on the received feedback. The state of the plant is measured by the sensory system

$$x^* = g(x, v) \tag{3-5}$$

where v is the possible perturbation affecting the sensory system. Finally, the measured position x^* is fed back to the controller, which in the case of a human is the brain.

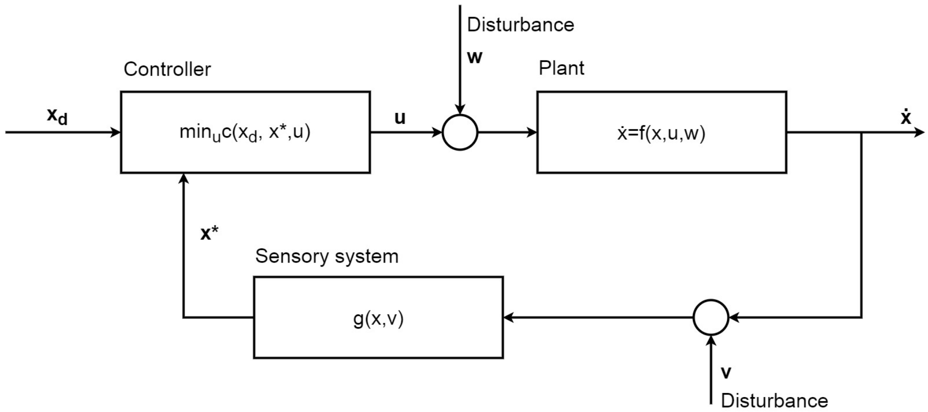

Figure 3-4: The block diagram of feedback motor control. The controller based on the received desired state x_d generates a motor command u. The motor command is sent to the plant, which results in movement $\dot{x}$. Before the motor command reaches the plant noise w can disturb the system. The state of the system is measured by the sensory system. Before the state is measured noise v might enter the system. Finally the measured state x^* is fed back to the brain.

Unfortunately, the optimal feedback controller can be really complex and computationally expensive [16].

In the case of the trolley, the input is given by the user. The user by pulling the handle gives an input to the controller. The controller calculates the motor command. The motor command is sent to the motors and the trolley moves. Finally, the sensors give a feedback about the state to the controller.

3-1-3 Feedback control with a state estimator

The sensory feedback received from our sensors is imperfect, therefore we can only estimate it [17]. Unluckily during motion, the estimation needs to change all the time. Furthermore, the system contains a lot of noise and the control signals are fed back with a considerable delay. To overcome these limitations the inner model of the human control is needed, which can make predictions to compensate for the time delay. Let's note that the forward model is not the same as the feed-forward control. In order to avoid any confusion in the following the forward model is called a predictor. The combination of the predictor and the sensory feedback is done by the estimator, which contains the Kalman filter (figure 3-5).

There is always a delay and an error between the predicted state and the measured feedback which can be corrected by using a Kalman gain. The Kalman filter is a recursive filter, which means that it updates its estimates based on the previous estimates, the current motor command, and the sensory feedback. The simplified scheme of optimal feedback control is shown in figure 3-5.

The controller sends a motor command to the plant and an efference copy of the motor command is sent to the predictor. The predictor makes a prediction $\tilde{x}$ about the state based on the received motor command

$$h(\tilde{x}, u). \tag{3-6}$$

The feedback signal of the sensory system x^* and the output of the predictor $\tilde{x}$ is the input of the state estimator. The state estimator sends the estimate of the state $\hat{x}$ to the controller

$$\hat{x} = j(\tilde{x}, x^*). \tag{3-7}$$

The controller calculates the costs and rewards and generates a motor command based on the received feedback and the desired state x_d. Figure 3-6 shows the detailed scheme of optimal feedback control.

In a sterile environment e.g. in a simulation, the robots predictions are the same as the measured feedback. In the real world, there is always some noise that cannot be predicted.

To conclude, a Kalman filter is needed as well to compensate for this error. To sum up until now a feedback, a predictor and an estimator with a Kalman filter are needed to control our system.

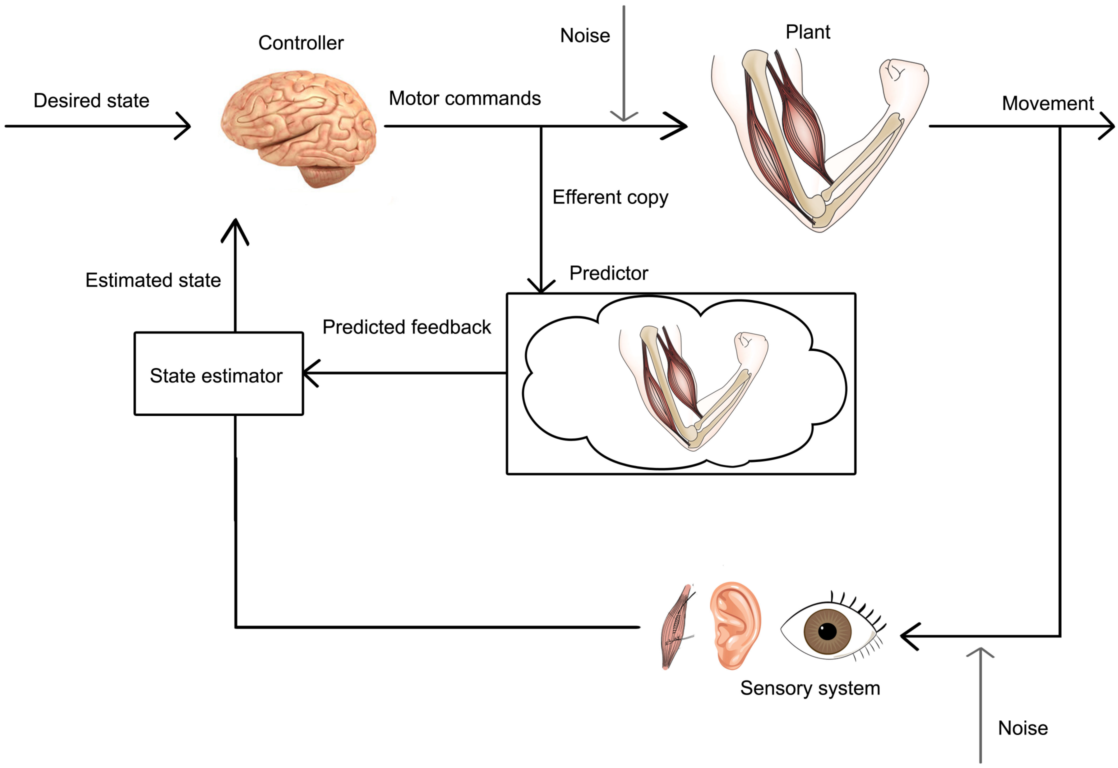

Figure 3-5: Scheme showing optimal feedback motor control. The brain based on the received task generates motor commands. The motor commands are sent to the plant, which results in movement. Before the motor command reaches the plant noise might disturb the system. The efferent copy of the motor commands is sent to the predictor. The predictor makes a prediction about the resulting state. The state of the system is measured by the sensory system. Before the state is measured noise might enter the system. Then the measured state and the predicted state are merged in the state estimator. Finally, the state estimator based on the received inputs generates a state estimate, which is fed back to the brain.

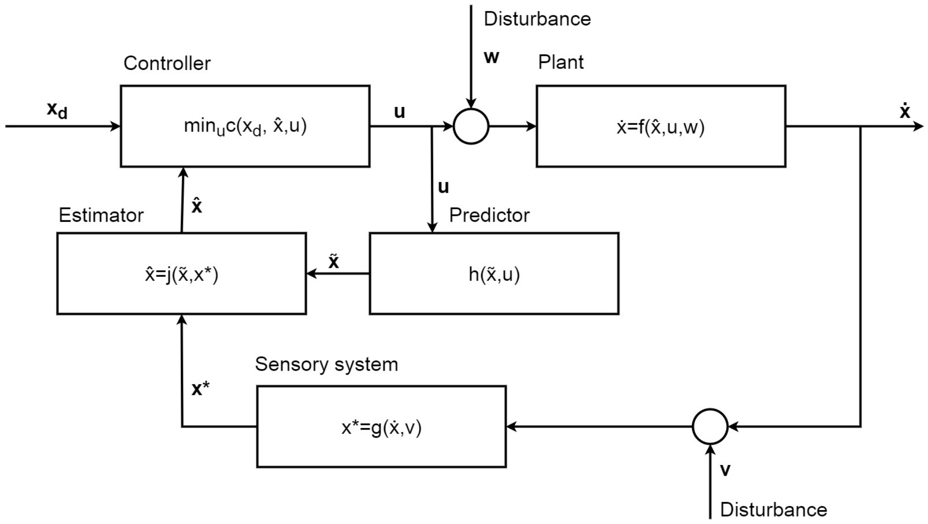

Figure 3-6: The block diagram of optimal feedback motor control. The controller based on the received desired state x_d generates motor command u. The motor command is sent to the plant, which results in movement $\dot{x}$. Before the motor command reaches the plant noise w might disturb the system. The efferent copy of the motor command u is sent to the predictor. The predictor makes a prediction $\tilde{x}$ about the resulting state. The state of the system is measured by the sensory system. Before the state is measured noise v might enter the system. Then the measured state x^* and the predicted state $\tilde{x}$ is merged in the state estimator. Finally, the state estimator based on the received inputs generates a state estimate $\hat{x}$, which is fed back to the brain.

3-1-4 Learning

In this subsection, the possible mechanism of learning is explained within the framework of optimal motor control. Since the physical environment is changing all the time and it is too complex to be modelled beforehand, it is necessary for a robot to learn and adjust its behaviour. The robot needs to learn its own model, also called internal model.

Learning the internal model is essential in order to predict the best action to achieve a goal. Making the robot to learn about itself and the environment is similar to the learning of infants [18]. Infants can learn the associations between the motor commands and the received feedback.

In the case of the robot, it can be implemented in a way that the robot sends out random motor commands u_r and receives information from the vision system x^*. The motor commands and the state of the robot are random variables. The connection between them is shown with arrows in figure 3-7. This type of learning, when a random motor command is generated to provoke some new events is also called motor babbling.

The feedback is used to learn the underlying network. In short, the system learns the associations between the robot's motor commands u_r, the change in the robot's state $\dot{x}$, and the measured state x^* that is received from the vision system.

At first, a random motor command is chosen u_r. This motor command is sent to the robot. The vision system returns the measured state x^*. Then the whole process is repeated with a different motor command to train the robot [18].

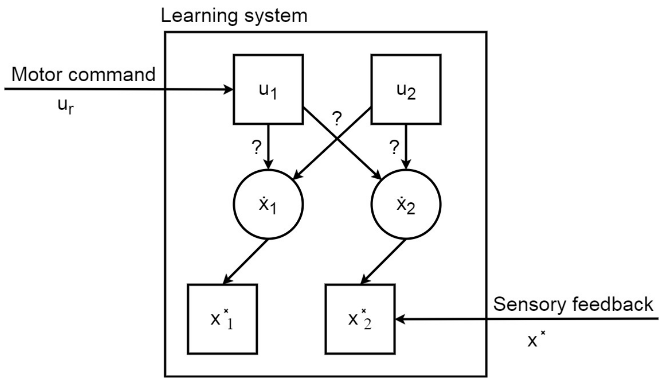

Figure 3-7: Scheme showing how internal models are learned. The random motor command u_r is sent to the plant. This results in a change of the state $\dot{x}$. The resulting state is measured by the sensory system x^*, which is fed back to the controller. The controller learns the connection between the generated motor command and the measured state. The connections are shown with arrows.

Using this structure the robot can build up its own internal model in some cases also named forward model or predictor because the system' s prediction about future states is based on the received feedback. Then this network can be used as a predictor to make predictions about the effect of an action.

This model can also be used as an inverse model. Given an observation, the most likely motor command can be calculated, which caused this state. This means that by using an inverse model it will be possible that a robot sees a state and from that, the most likely motor command can be calculated, which would lead to the same action.

Switching between inverse and forward models also called predictors, leads to the theory of multiple pairs of forward-inverse models. In order to deal with complex situations the humans need, whether a single controller, which would need to be highly complex or multiple controllers, with each controller suitable for one set of situations. The problem with using a single highly complex controller is that in the case of a new scenario, it would need to adapt before it could produce an appropriate motion. This would result in a large performance error. Therefore, in the next section the theory of multiple controllers is going to be described.

Learning Multiple Paired Forward-Inverse Models

The theory that the brain contains multiple pairs of forward-inverse models for motor learning and control (MPFIM) is a modular approach. Each controller is suitable for a small set of tasks and it can become an expert in it. They form a network which combines the output of separate controllers, each based on the received input [3]. The forward and inverse models are tightly coupled in which the forward model determine the contribution of each inverse model to the final motor command. This system is able to learn simultaneously multiple inverse models and the selection of the appropriate inverse models [3] as shown in figure 3-8.

At first, it is necessary to divide up the experience using the forward models. Let's denote x the state of the system and u the motor command as the input. When n forward models are available, the prediction of the ith forward model can be calculated as

$$\tilde{x}^i = \phi(L^i, x, u) \tag{3-8}$$

where L^i is a weight variable of the ith forward model.

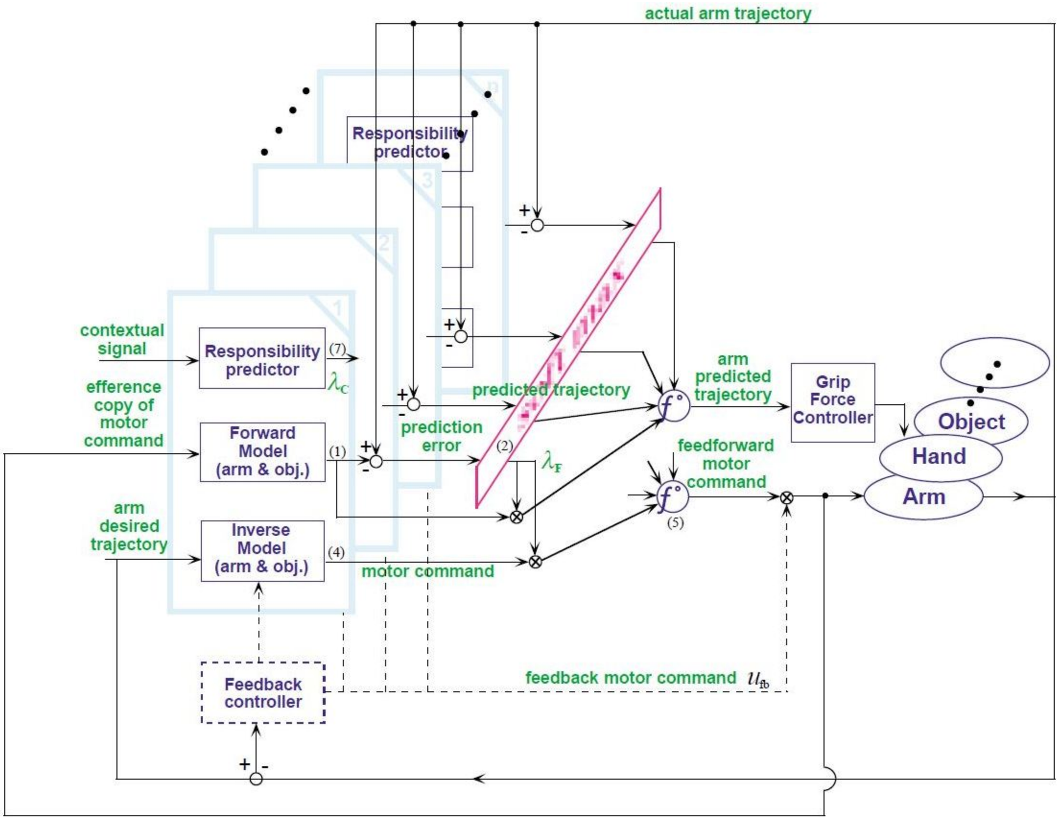

Figure 3-8: Block diagram showing how Multiple Paired Forward-Inverse Model is used to control arm movement while manipulating different objects [3].

The predicted states will then be compared with the actual next state, to calculate a so-called responsibility signal λ^i. The responsibility signal determines the contribution of each forward model to the resulting state. Depending on the prediction error the responsibility signal, denoted by λ^i is calculated for the forward-inverse pair using the soft-max function

$$\lambda^i = \frac{e^{-|x-\tilde{x}^i|^2/(2\sigma^2)}}{\sum_{j=1}^{n} e^{-|x-\tilde{x}^j|^2/(2\sigma^2)}} \tag{3-9}$$

where x is the state of the system, $\tilde{x}$ is the predicted state and σ is a scaling constant. The error between the ith predicted and real state is calculated. This error is divided by the sum of error of all the states. Therefore the soft-max function normalizes the responsibility to be between 0 and 1. This means that the forward model which has small error will have a high responsibility signal. The model which has a large error will have a small responsibility signal. Then the responsibilities are used for training the forward functions.

Each forward model has an inverse model pair. The input of the inverse model is the responsibility signal λ^i, the desired state x_d, and the current state x, from which the motor command is calculated as

$$u^i = \psi(\lambda^i, x_d, x). \tag{3-10}$$

The total motor command is the summation of the motor commands. To sum up, the responsibility function is used to make the correct forward-inverse pairs. Besides, it determines the contribution of inverse models to the final motor command.

Finally, the system has to be able to switch between the models before a motor command has been generated. Therefore a responsibility predictor needs to be added to the model. Each responsibility predictor makes a prediction about its own contribution. These predictions are compared with the actual contribution and the error is used to train the responsibility predictor [3].

To summarize, according to this theory that is how the brain is able to switch between multiple paired forward-inverse models.

3-1-5 Risk-sensitive optimal feedback control

In the following section the risk-sensitive optimal feedback control will be introduced.

In the previous subsection, the risk-neutral optimal feedback control has been described in detail. The goal of the risk-neutral optimal feedback control is to minimize cost.

In contrast, the risk-sensitive control not only minimizes the cost, it also minimizes the variability of the cost. Humans are making choices in a risk-sensitive manner, which can be easily seen from the following example. When a human has a choice of getting 45 euro for sure or winning 100 euro with a 50-50 chance, the majority of people will choose the sure bet. If the same choice would be offered to a robot with risk-neutral control it would choose to try to win 100 euro, because its mean is 50 euro, which is still higher than 45 euro [19].

The risk-neutral controller can be easily modified into a risk-sensitive controller. The risk-sensitive controller takes into account not only the cost, but also the variability of the cost. Therefore the motor command will be generated by the minimization of both the cost and the variability of cost. In the case of the trolley, if it uses risk-sensitive control, e.g. on the stairs if it experiences that on the edge of the stairs it is falling down, it will try to move away from the edge of stairs to the safer regions.

Therefore, the risk-sensitive control sounds promising for the control of the trolley.

3-1-6 Conclusion

The biggest advantage of optimal motor control is that by providing a performance criterion of the goal, it can calculate all the movement details.

By using optimal feedback motor control the robotic shopping trolley (1) is able to move. In addition it is a closed loop control method and therefore it is able to (2) balance, (4) reacts to the input, and (5) is able to sense the environment. The optimal motor control includes a predictor and an estimator by which it can (3) resist perturbations, and (6) cope with uncertainties. The disadvantage of optimal feedback control is that at first, an event needs to occur to react on it in contrast to active inference, where the prior beliefs about the state determine the control action. In addition, as already stated, the optimal motor control is (8) computationally expensive.

Therefore, the optimal feedback control fulfils six criteria from the eight criteria needed for the control of the trolley. The table with all the criteria is included below (table 3-1).

Table 3-1: List of criteria needed for the control of the trolley. Optimal control method meets six criteria from eight.

	Criteria	**Optimal feedback motor control**
1.	move	✓
2.	balance	✓
3.	resistant to perturbation	✓
4.	reacts on the input	✓
5.	sense the environment	✓
6.	copes with uncertainties	✓
7.	makes predictions	—
8.	low comp. complexity	—

3-2 Equilibrium point hypothesis

The next theory which is going to be described in more detail is the equilibrium point hypothesis shown in figure 3-9. The basic idea of equilibrium point hypothesis is the following: if someone would replace his arm with a robot arm in a way that all the muscles would be replaced by a pair of rubber bands, then the robot arm would end up in the same position as the human arm. When the properties of the rubber band or the force-length relation of the muscles is changed the equilibrium point would change as well. The equilibrium point means that the resulting force is zero and the state variables are zero.

The motor neurons or the brain is able to change the force-length relation of each muscle, resulting in a change of the equilibrium positions [20]. According to the equilibrium point theory, the central nervous system plans the movement path by the transition of equilibrium points through the trajectory. This theory is also called threshold control because the signals of the central nervous system changes the threshold length of the activated muscles.

The big advantage of this theory is that the brain does not estimate the dynamics of the limb, rather the spinal reflexes provide the necessary information. The equilibrium point theory can be implemented on robots if a good internal model is available [21].

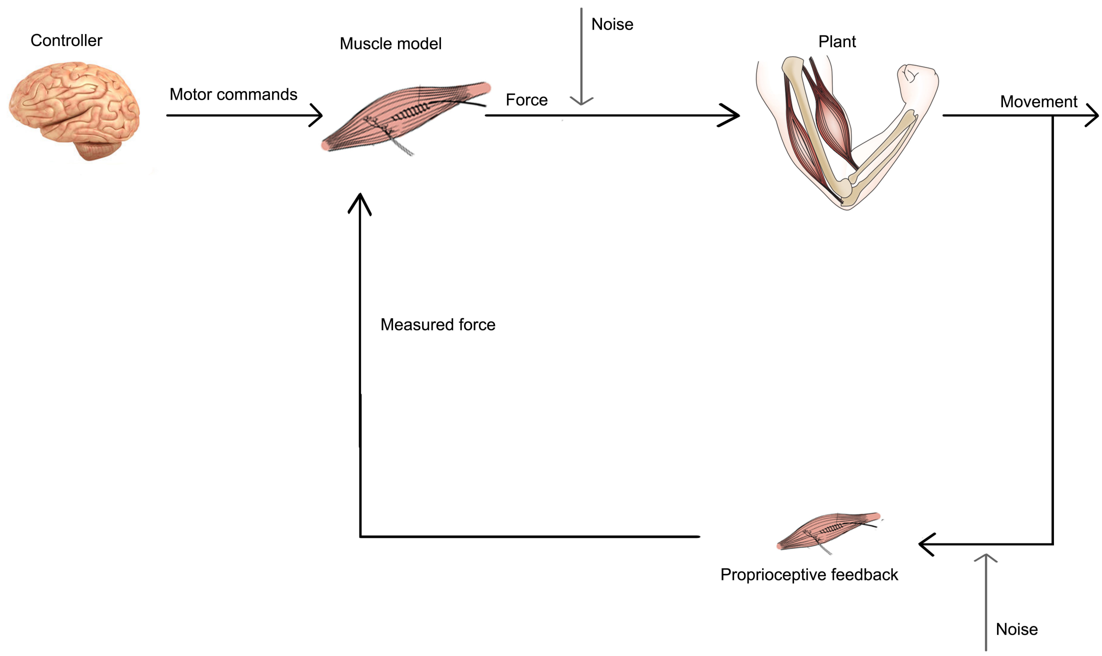

Figure 3-9: Scheme showing the equilibrium point hypothesis. The brain generates a motor command, which is sent to the muscle model. The muscle model based on the motor command and the received feedback it generates a force command. Before the command reaches the plant noise can disturb the system. The force command is sent to the plant, which results in movement. Before the state is measured noise might enter the system. The state of the system is measured by the sensory system. Finally, the measured force is fed back to the muscle model.

The brain sends what should be the threshold length of the activated muscles to the muscle model. The muscle model exerts forces f_c based on the proprioceptive force feedback f_m. The aim of the muscle model is to exert a force f_c by which the resulting force will be zero.

In the following part, the mathematical description of the theory will be provided. The more detailed block diagram can be seen in the figure 3-10. Let's denote the dynamics of the system by

$$\dot{x} = h(f_c, w) \tag{3-11}$$

where f_c is the exerted force. The state of the mechanical system is denoted by x. A change occurs when the forces acting on the system f_c, and the forces produced by motion f_m are not equal

$$I(x)^{-1}(f_c(x) \neq f_m(x) \tag{3-12}$$

where I is the system's inertia. The given state is an equilibrium point when the external force f_c minus the force produced by motion f_m is zero, and the state variable x is zero. When the system is in an equilibrium position it will stay there. Each state where a field has zero force is a null point field. The equilibrium points are a subset of these null points because the equilibrium point exists only where the velocity is zero.

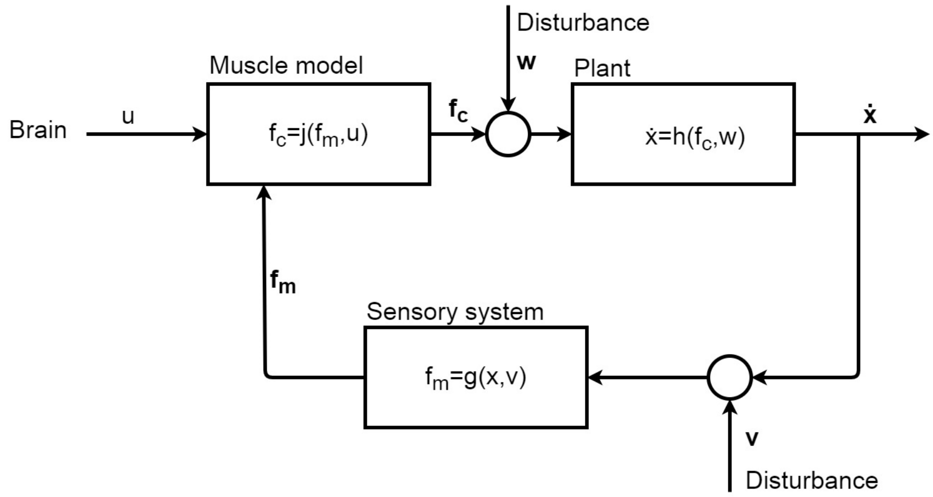

Figure 3-10: The block diagram of equilibrium point hypothesis. The brain generates a motor command u, which is sent to the muscle model. The muscle model based on the motor command u and the received feedback f_m, generates a force command f_c. Before the command reaches the plant noise w might disturb the system. This command is sent to the plant, which results in movement $\dot{x}$. Before the state is measured noise v might enter the system. The state of the system is measured by the sensory system. Finally, the measured force f_m is fed back to the muscle model.

In the next part, the control of the system will be described. The input u is selected in a way that the system reaches the desired state x_d. For this the muscle model needs to produce the force as follows

$$f_c = \hat{f}_m + \hat{I}\dot{x}_d \tag{3-13}$$

where $\hat{f}_m$ is the estimated force produced by motion, $\hat{I}$ is the estimated system's inertia, and $\dot{x}_d$ is the desired change in the state. In order to react to uncertainties in the environment, the muscle model should push to the position where it should be, if any disturbance affects the motion. If the estimations would be correct, the tracking error would be zero. The error equals zero when the measured state and the desired state is equal. The tracking error should decrease by time and the system becomes stable on the desired trajectory. The estimates used for the control are internal models that the brain has already learned.

3-2-1 Learning

In case of the equilibrium point theory, the brain needs to learn an internal model. This process is similar to the process of learning forward models in case of the optimal motor control. This system also learns the associations between the motor command and the received proprioceptive feedback by using motor babbling.

In contrast to optimal motor control in this case, the motor commands are used to change the force-length relationship of muscles while staying in equilibrium.

3-2-2 Conclusion

The biggest advantage of equilibrium point theory is that is (8) computationally efficient. In addition, since the equilibrium point theory is closed loop control, it meets 4 criteria needed for the control of the assistive trolley. These are the following: it can (1) move, (2) balance, (4) react on the input, and it is able to (5) sense the environment. In addition, the equilibrium point control is (3) resistant to perturbations.

On the other hand, the equilibrium point hypothesis has several disadvantages e.g. it is not (6) able to cope with uncertainties. In the case of the trolley in order to sense the input of the user and the environment sensors are needed. The signal received from the sensors always contains some noise and uncertainties. Therefore the control system should be able to cope with uncertainties. That is the main reason why the equilibrium point hypothesis is not suitable to control the trolley.

Table 3-2 shows the criteria needed for the control of the robotic trolley.

Table 3-2: List of criteria needed for the control of the trolley. The equilibrium point hypothesis meets four criteria from eight.

	Criteria	**Equilibrium hypothesis**
1.	move	✓
2.	balance	✓
3.	resistant to perturbation	✓
4.	reacts on the input	✓
5.	sense the environment	✓
6.	copes with uncertainties	—
7.	makes predictions	—
8.	low comp. complexity	✓

3-3 Active inference

In the section 3-1 the optimal control theory has been described. In the case of optimal control, the disadvantage is that, at first it needs to experience a state to react to it. In some cases, this might be too slow, which leads us to the active inference theory.

In the next section, the active inference theory is going to be described in more details. The active inference theory explains the motion in terms of beliefs and inference. It eliminates the need for cost functions and the need for efferent copy. The scheme of active inference is shown in figure 3-11.

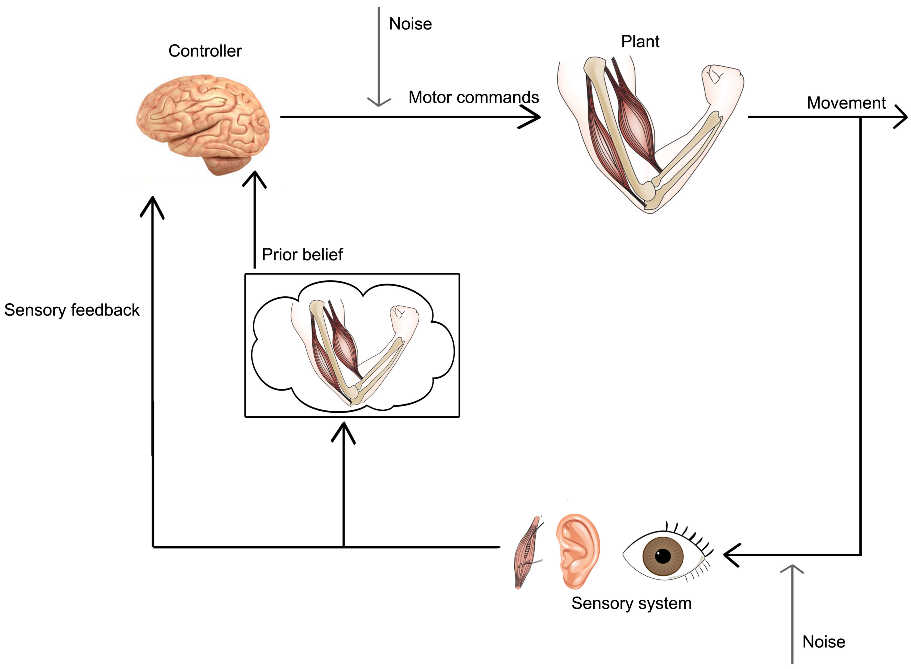

Figure 3-11: Scheme showing active inference control. The brain based on the received prior belief generates a motor command. The motor command is sent to the plant, which results in movement. Before the motor command reaches the plant noise can disturb the system. The state of the system is measured by the sensory system. Before the state is measured noise might enter the system. Then the measured state is sent to the brain and to the block, which stores the prior belief. The prior belief if needed it can be updated based on the received feedback. Finally, the brain generates a new motor command based on the received feedback and the updated belief.

The more detailed scheme of active inference is shown in figure 3-12. In active inference control the controller receives a prior belief about the state of the system μ and a sensory feedback x^*. The controller generates a motor command based on the prior belief and the sensory feedback

$$u = argmin_u F(x^*, \mu, u). \tag{3-14}$$

The motor command u is sent to the plant, which results in movement

$$\dot{x} = f(x, u, w). \tag{3-15}$$

The movement is measured by the sensory system. This sends the measured state of the system x^* to the controller and to the prior belief block.

The belief is updated based on the received feedback

$$\mu^* = argmin_{\mu} F(x^*, \mu). \tag{3-16}$$

The prior belief and the controller uses the same F function, which is the free energy. The active inference theory is also called as free-energy principle. The aim of the control system is to minimize free-energy, which can be done by either changing the prior belief or by changing the control action. Then the controller generates a new motor command based on the updated belief and the received sensory feedback.

In the following part, the optimal control theory and the theory of active inference are going to be compared to highlight the advantages and disadvantages of the active inference theory. In order to simplify the comparison of optimal control and active inference, the scheme of optimal control is included below (figure 3-13).

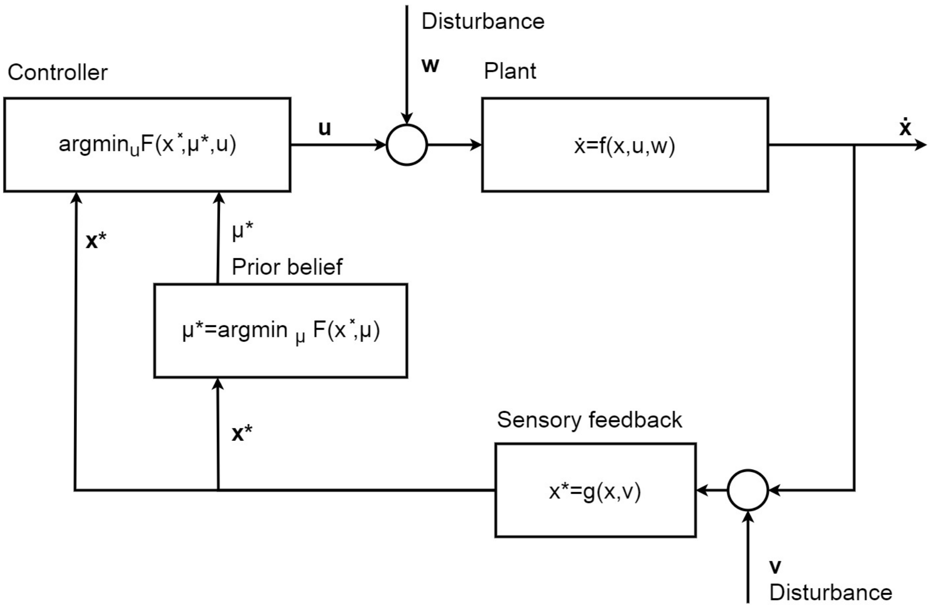

Figure 3-12: Block diagram of active inference control. The brain based on the received prior belief μ generates a motor command u. The motor command is sent to the plant, which results in movement $\dot{x}$. Before the motor command reaches the plant a noise w can disturb the system. The state of the system is measured by the sensory system. Before the state is measured a noise v might enter the system. Then the measured state x^* is sent to the brain and to the block which stores the prior belief. The prior belief μ if needed it can be updated based on the received feedback x^*. The prior belief and the controller uses the same F function, which is the free energy. The active inference theory is also called as free-energy principle. The aim of the control system is to minimize free-energy, which can be done by changing the prior belief or the control action. Finally, the brain generates a new motor command u based on the received feedback x^* and the updated belief μ.

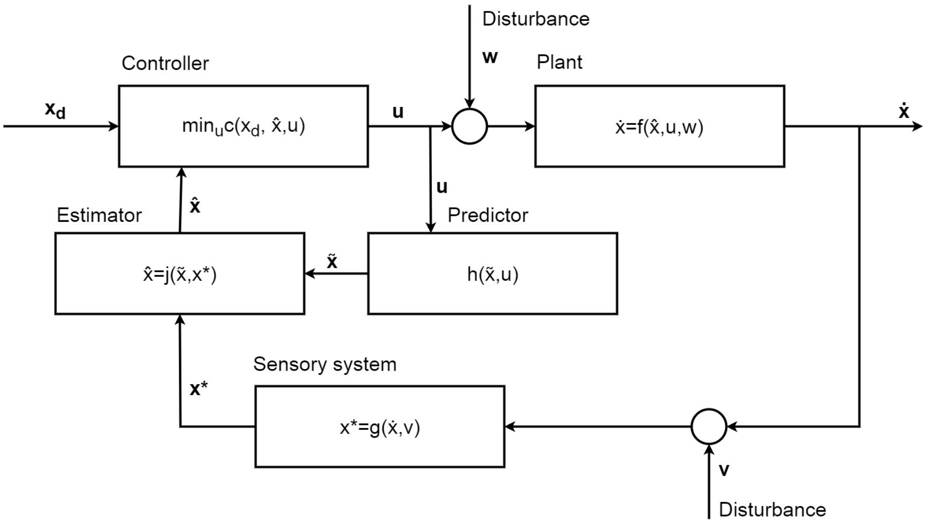

Figure 3-13: The block diagram of optimal feedback motor control. The controller based on the received desired state x_d generates a motor command u. The motor command is sent to the plant, which results in movement $\dot{x}$. Before the motor command reaches the plant noise w might disturb the system. The efferent copy of the motor command u is sent to the predictor. The predictor makes a prediction $\tilde{x}$ about the resulting state. The state of the system is measured by the sensory system. Before the state is measured noise v might enter the system. Then the measured state x^* and the predicted state $\tilde{x}$ is merged in the state estimator. Finally, the state estimator based on the received inputs generates a state estimate $\hat{x}$, which is fed back to the brain.

The optimal feedback control has been explained in more details in the previous sections, therefore, now only a quick overview of the system structure is given. The optimal motor control has a controller, which sends the motor command to the plant, and the efferent copy of the motor command to the predictor. The state of the plant is sensed by the sensory system, which output is connected to the state estimator. The predictor makes a prediction based on the received motor command copy and sends it to the state estimator. The state estimator integrates the estimated state calculated by the predictor and the measured state send by the sensory system. The output of the state estimator is sent to the optimal controller, which calculates a motor command depending on the received feedback and the cost function.

In the model of active inference, the predictor and the estimator is merged and it is modified into a block which stores the prior beliefs. The cost function $c(x, u)$ is replaced by prior beliefs μ about the limb trajectories.

This change is significant because this means that in comparison to the optimal control theory, the active inference theory is always at least one step ahead. In case of the optimal control at first something needs to happen, so the controller can react to it. In contrast, in the case of active inference, the system has a prior belief about events even which has never occurred.

Unfortunately, the replacement of cost functions with prior beliefs does not reduce the computational complexity [22].

To conclude the active inference provides an interesting theory about motor control.

3-3-1 Learning

In the following section, the process of learning is going to be described for active inference control. In the case of the active inference, the brain has to collect statistics and compute path integrals [4]. At first, the brain needs to collect data by inference sampling. Then the controller is adjusted based on the received data, which is called learning as it can be seen in the figure 3-14.

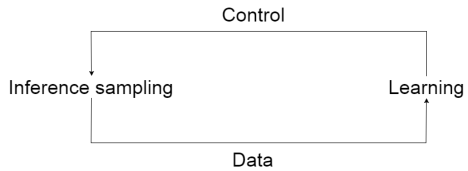

Figure 3-14: Scheme showing the learning process in case of active inference. The brain collects data by inference sampling. Then the controller is adjusted based on the received data, which is called learning. The controller generates a new control command and it collects new data by inference sampling [4].

In the case of the active inference control, the extrinsic frame and the intrinsic frame can be distinguished. In the case of the extrinsic frame, the brain is just a slave. It observes the physical world, but as soon as any planning starts it takes place in the inner world, which is the intrinsic frame. The inner world computation serves three purposes. The first one is the spontaneous activity which is a type of Monte Carlo inference sampling to collect data, denoted by MC in the figure below [4]. The second part computes actions using Bayesian estimations for the situation from the samples, denoted by ΔB. The third part is learning, which improves the sampler. Therefore, learning means improving the sampler [4].

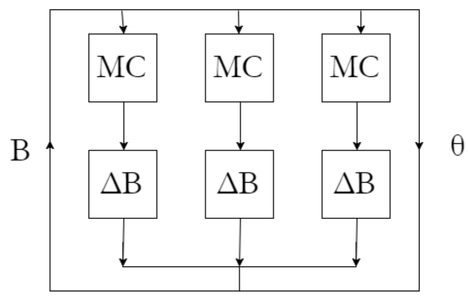

Figure 3-15: Scheme showing the purposes of inner world computations. The first purpose is to collect data by Monte Carlo inference sampling (MC). The second purpose is to compute actions using Bayesian estimations (B), from the data ΔB. The third purpose is to improve the sampler, which is called as learning [4].

3-3-2 Conclusion

The active inference control is a quite new theory of human motor control. Since it is a closed loop control method it fulfils 4 criteria needed for the control of the trolley. By using active inference control the trolley can (1) move, (2) balance, (4) reacts to the input, (5) sense the environment. In addition, by using its prior beliefs it is (3) resistant to perturbations, and it can (6) cope with uncertainties. Finally, the biggest advantage is that (7) it uses prior beliefs about the state to determine the control action and therefore it is always one step ahead in comparison to the optimal motor control. On the other hand, the replacement of cost functions with prior beliefs introduces new complex calculations, by which (8) the computational complexity is the same as in case of optimal motor control.

To conclude, the active inference control meets seven criteria needed for the control of the trolley. The criteria needed to control the trolley are listed in table 3-3.

Table 3-3: List of criteria needed for the control of the trolley. Active inference meets seven criteria from eight.

	Criteria	**Active inference**
1.	move	✓
2.	balance	✓
3.	resistant to perturbation	✓
4.	reacts on the input	✓
5.	sense the environment	✓
6.	copes with uncertainties	✓
7.	makes predictions	✓
8.	low comp. complexity	—

3-4 Risk-aware control

The last theory which is going to be described in more details is the risk aware control [5]. In the case of the risk- aware control, the human movement is governed by estimates of risk based on uncertainty about the current state and knowledge of the cost of errors. The humans move in a risky environment, therefore, prediction and avoidance of risk are necessary. In the risk-aware control theory the term risk ρ is defined as the product of the cost of an error $c(\epsilon)$ and the probability of an error $p(\epsilon)$

$$\rho = c(\epsilon)p(\epsilon). \tag{3-17}$$

An example of risk-aware control is when someone drives on a road and on one side of the road is deep ravine and on the other side of the road is a nice meadow. The driver will drive nearer to the meadow side of the road, even if the probability of falling into the ravine is really low, but the cost if it happens is too high. It is impossible to collect multiple samples of fatal events and correct according to the results. Therefore the human motor control has to take into account risk awareness [5].

The scheme of risk-aware control is shown in figure 3-16.

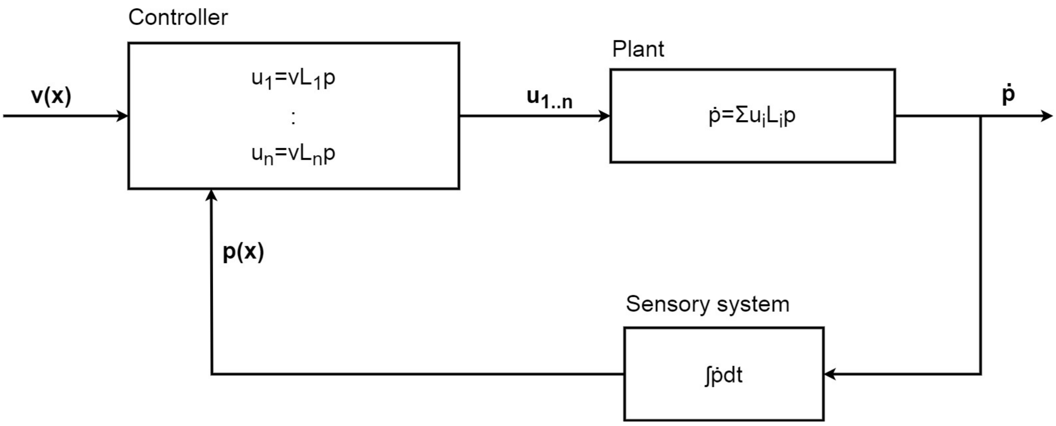

Figure 3-16: Block diagram of risk-aware control [5]

The input of the model is the value function $v(x)$ and probability of the state $p(x)$. The state is represented by a scalar or vector x and the probability distribution is $p(x)$ [5].

The belief in the cost of errors is denoted by the value function $v(x)$. The more negative value means an undesirable state. The estimated expected cost is

$$E[v] = \int v(x)p(x)dx. \tag{3-18}$$

When the cost of error is negative, the state is undesirable, then the probability of the error should be decreased. When the cost of error is positive or zero, it is desirable and the probability of that state should be increased.

The aim of the control is to reduce the cost of error. The best way to reduce cost is to choose a control action, which leads to the highest increase in the value

$$u = argmax_u vLp \tag{3-19}$$

where u is the motor command, v is the value function, L is the weight for each state, and p is the probability of being at a certain angles. The optimal control is to activate each control output proportional to the value by which it would improve the value of the state

$$u_i = vL_i p. \tag{3-20}$$

Each neuron can generate a +1,-1, or 0 motor command. After calculating the motor command for each neuron, the motor commands are summed and sent to the plant.

This results in an unusual feedback controller as it is shown in the figure 3-16. In case of optimal motor control a desired trajectory x_d is used. In contrast to the risk-aware controller, where a value function $v(x)$ is used. In addition, instead of the state value a probability density of the state $p(x)$ is used. The value function gives the relative value of each state instead of one state. This difference is really important, because it provides much more flexibility than in case of the optimal controller. The optimal controller defines a certain path, while in the risk aware controller if the goal is to remain on the road it does not matter where exactly on the road. Therefore, there is no path defined and there is no waste of energy to correct for perturbations which do not push off the road. If perturbation occurs the controller detects the deviation from the optimal cost and rapidly corrects it. This means that u will react proportional to the value of risk. If the perturbation is pushing to a safer state the u will not resist. On the other hand if the perturbation is pushing to a state with higher risk the u will be activated and it will try to resist the perturbation. The risk-aware control takes into account state uncertainty, control uncertainty, and a general cost function [5].

The risk-aware control is related to the concepts developed within the framework of active inference [5]. However, it does not update its value function based on the received feedback. This means that this system avoids the hard inference problem, by which it reduces the computational complexity.

3-4-1 Learning

By learning the efficiency of the system representation is improved, by which the storage size is reduced and the system becomes more efficient. In addition, the control has to learn the effect of individual neurons on the dynamics. The dynamics are described by L operator, which connects v and p, where v is value function and p is the probability of a particular angle. In the beginning our prior assumption does not know anything about the effect of u, therefore the weight L will be zero. This means that the prior assumption about the effect of the motor command is that it does nothing.

The change in the probability of an angle is calculated by

$$\dot{p} = Lp \tag{3-21}$$

and therefore the easiest way to learn L is to observe p and $\dot{p}$, and update the weight L. The change in the weights can be calculated by

$$\Delta L = \gamma(\dot{p} - \hat{\dot{p}})p \tag{3-22}$$

where $\hat{\dot{p}}$ is the predicted value of $\dot{p}$, $(\dot{p} - \hat{\dot{p}})$ is the prediction error and γ is the learning rate. At first the change in the probability of a given angle needs to be learned, only a prediction is available

$$\hat{\dot{p}} = Lp. \tag{3-23}$$

Then it adjusts its weight L based on the received feedback. When the prediction is smaller than the received feedback the weight is increased. When the prediction is bigger than the received feedback the weight is decreased.

After a time the error between the feedback and the prediction will become zero.

3-4-2 Conclusion

Risk-aware control is related to the concepts developed within the framework of active inference [5]. Therefore, it meets seven criteria needed for the control of the assistive trolley. In addition the risk-aware control reduces (8) the computational complexity, by which the risk-aware control fulfills all the criteria needed for the control of the trolley. The table with all the criteria is included below (table 3-4). In addition to fulfilling all the criteria, the risk-aware control takes into account the cost of error while executing a motor task. Above all it is flexible, since it does not require the calculation of exact trajectories and therefore it becomes even more computationally efficient.

To conclude, the risk-aware control seems to be the most suitable for the control of the trolley.

Table 3-4: List of the criteria needed for the control of the trolley. In case of the risk-aware control each criteria is met.

	Criteria	**Risk-aware control**
1.	move	✓
2.	balance	✓
3.	resistant to perturbation	✓
4.	reacts on the input	✓
5.	sense the environment	✓
6.	copes with uncertainties	✓
7.	makes predictions	✓
8.	low comp. complexity	✓

3-5 Summary

In this chapter, four control theories have been described and analysed. These are the optimal control theory, the equilibrium point hypothesis, the active inference theory and the risk-aware control theory. The table with all the criteria is included below (table 3-5).

The first theory which has been described is the optimal motor control theory. This theory can explain the control of many motor tasks, although the experiments are mainly reaching or pointing tasks. The biggest advantage of optimal motor control is that by providing a performance criterion of the goal, it can calculate all the movement details.

By using optimal feedback motor control the robotic shopping trolley (1) is able to move. In addition it is a closed loop control method and therefore it is able to (2) balance, (4) reacts to the input, and (5) is able to sense the environment. The optimal motor control includes a predictor and an estimator by which it can (3) resist perturbations,and (6) can cope with uncertainties. The disadvantage of optimal feedback control is that at first, an event needs to occur to react on it in contrast to active inference, where the prior beliefs about the state determine the control action. In addition, as already stated, the optimal motor control is (8) computationally expensive.

Therefore, the optimal feedback control fulfils six criteria from the eight criteria needed for the control of the trolley. The table with all the criteria is included below (table 3-1).

The second theory which has been introduced was the equilibrium point hypothesis. The biggest advantage of equilibrium point theory is that is (8) computationally efficient. In addition, since the equilibrium point theory is closed loop control, it meets 4 criteria needed for the control of the robotic shopping trolley. These are the following: it can (1) move, (2) balance, (4) reacts on the input, and it is able to (5) sense the environment. In addition, the equilibrium point control is (3) resistant to perturbations.

On the other hand, the equilibrium point hypothesis has several disadvantages e.g. it is not (6) able to cope with uncertainties. In the case of the trolley in order to sense the input of the user and the environment sensors are needed. The signal received from the sensors always contains some noise and uncertainties. Therefore the control system should be able to cope with uncertainties. That is the main reason why the equilibrium point hypothesis is not suitable to control the trolley.

The third theory which has been described in this chapter is the active inference control, which is a quite new theory of human motor control. Since it is a closed loop control method it fulfils 4 criteria needed for the control of the trolley. By using active inference control the trolley can (1) move, (2) balance, (4) reacts to the input, (5) sense the environment. In addition, by using its prior beliefs it is (3) resistant to perturbations, and it can (6) cope with uncertainties. Finally, the biggest advantage is that (7) it uses prior beliefs about the state to determine the control action and therefore it is always one step ahead in comparison to the optimal motor control. On the other hand, the replacement of cost functions with prior beliefs introduces new complex calculations, by which (8) the computational complexity is the same as in case of optimal motor control.

To conclude, the active inference control meets seven criteria needed for the control of the trolley.

Finally, the risk-aware control has been discussed. Risk-aware control is related to the concepts developed within the framework of active inference [5]. Therefore, it meets seven criteria needed for the control of the assistive trolley. In addition the risk-aware control reduces (8) the computational complexity, by which the risk-aware control fulfills all the criteria needed for the control of the trolley. The table with all the criteria is included below (table 3-4). In addition to fulfilling all the criteria, the risk-aware control takes into account the cost of error while executing a motor task. Above all it is flexible, since it does not require the calculation of exact trajectories and therefore it becomes even more computationally efficient.

As it has been already concluded in the section of risk-aware control, this theory seems to be the most suitable for the control of the trolley. In addition to fulfilling all the criteria Dr. T. Sanger concludes his article that he hopes that "this control method will provide a new type of flexible and scalable compliant adaptive controller for use in human-computer interactions, surgery, exoskeletons, or wherever rapid and appropriate response to a compliant yet dangerous environment with unpredictable perturbation is needed" [23].

The hypothesis of this article is: **The risk-aware control method is suitable for the control of an assistive trolley.**

This leads us to the next chapter, where the risk-aware control method is going to be described in more details. The implementation of this control method might be challenging, hence it has been implemented only on one robot.

In table 3-5 all the previously described theories are compared with the list of criteria needed for the control of the trolley.

Table 3-5: Control methods and criteria needed for the control of the trolley

	Criteria	**Opt. c.**	**Eq. h.**	**Act. inf.**	**Risk-a. c.**
1.	move	✓	✓	✓	✓
2.	balance	✓	✓	✓	✓
3.	resistant to perturbation	✓	✓	✓	✓✓
4.	reacts on the input	✓	✓	✓	✓
5.	sense the environment	✓	✓	✓	✓
6.	copes with uncertainties	✓	—	✓	✓
7.	makes predictions	—	—	✓	✓
8.	low comp. complexity	—	✓	—	✓

Chapter 4

Risk-aware model structure

Based on the literature search the risk-aware control is the most promising one. In case of risk-aware control a value function is provided to the system, by which it can avoid the undesirable states. By using a value function it does not require the calculation of exact trajectories and therefore, it becomes computationally more efficient, than the other human inspired control methods. In addition it learns and adjusts its control parameters by which it can adapt to different situations.

Therefore, in this chapter the risk-aware model is going to be described in more details.

In the figure below the block diagram of risk-aware control is shown.

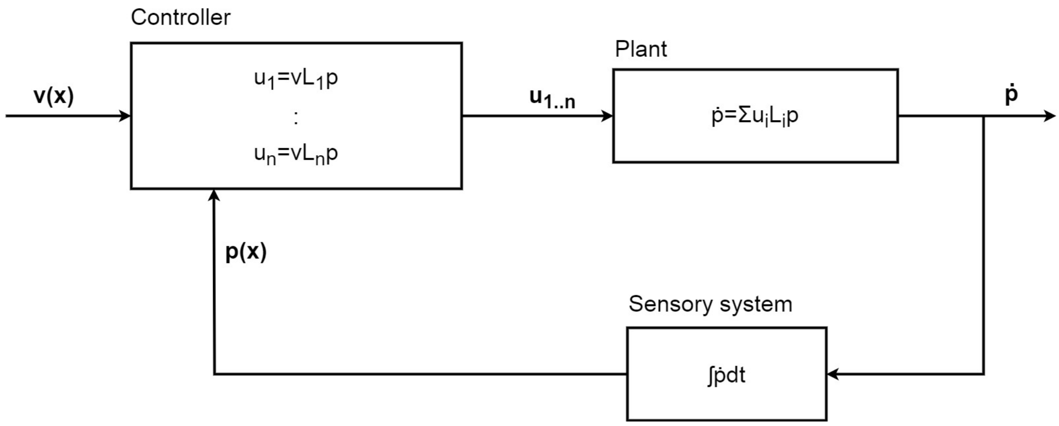

Figure 4-1: Block diagram of risk-aware control. The value function $v(x)$, which describes the desired states is the input of the controller. The controller receives the probability function $p(x)$ as well. Based on the input the motor command is generated. The motor command is sent to the plant, which results in the change of the state. The new probability function based on the new state is fed back to the controller [5].

A neural network has been used to implement the risk-aware control. For the trolley 1000 neurons were generated. Each neuron has a corresponding angle from 0 to 180 degrees denoted by x. This value is mapped between zero and one. Therefore x is an 1000 length array whose each element has a corresponding angle mapped between 0 and 1. In addition to the angle each neuron stores its weight denoted by w.

4-1 State

The controller receives the measured state of the trolley x^*, which is its current angle. From the measured angle x^*, the state probability distribution is calculated $p(x)$. At first, a Gaussian distribution is calculated, of which the mean is the current angle

$$p(x) = e^{-(x-x^*)^2/(2sd^2)}; \tag{4-1}$$

where sd is the variance of the Gaussian distribution. The function $p(x)$ is compared to an array of random numbers uniformly distributed between 0 and 1. By using an array of random numbers the state uncertainty can be represented. When the value of the random number α is lower than the value of the Gaussian function, the probability of that angle will be one. Otherwise the probability of that angle will be zero

$$\alpha\tilde{}U(0,1), \quad p(x) = (\alpha < p(x)). \tag{4-2}$$

This way the state will consist of only ones and zeroes, by which the computational complexity is reduced. Finally, $p(x)$ is convolved with the vector of [1,1,1] for smoothing.

4-2 Value function

In addition to calculating the probability of the angles the trolley receives a value function $v(x)$. The higher absolute value of the value function means that a state has a high cost, it is undesirable (unsafe). When a state is desirable (safe) the value of $v(x)$ is zero.

In case of the trolley the value function has been chosen in a way that when the deviation from the 90 degrees is more than a threshold value it is undesirable. Therefore, states which ones are undesirable have high absolute value. As the deviation from the desired angle increases the absolute value increases as well. The states where the deviation from the desired angle is less than the threshold value have zero value. Within the safe zone each state has the same value, because it does not matter what is the precise tilt angle within this zone. Therefore, if the trolley is within the safe zone the controller does not need to act on it.

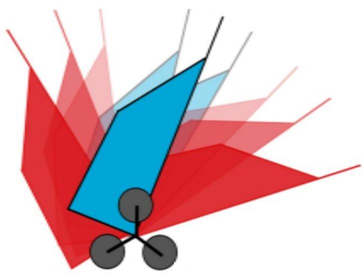

Figure 4-2: Different tilt angles of the trolley. As the tilt angle of the trolley increases, the absolute value of the value function increases (marked with red) . The higher absolute value of the value function means that a state has a high cost, it is undesirable (unsafe).

The value function is constructed in a similar way as the probability function. At first a Gaussian distribution is calculated of which the mean is the desired angle. Then the Gaussian distribution is convolved with the vector of [1,0,-1]. Therefore, the angles below the target angle will have a negative value and the angles above the target angle will have a positive value. Finally, this function is compared to an array of random numbers in the following way

$$\begin{aligned} \alpha\tilde{~}U(0,1), &\quad v(x) = (\alpha < v(x)) \\ \alpha\tilde{~}U(0,1), &\quad v(x) = -(\alpha < -v(x)). \end{aligned} \tag{4-3}$$

By using an array of random numbers the uncertainty in the value function is represented. This due to the fact that we do not know the exact cost of an error or the exact value function, it is only a belief. To summarize, at this point the value function is an array of ones, minus ones, and zeroes. Finally, the value is convolved with the vector of [1,1,1] for smoothing.

4-3 Motor command

The next step is to calculate the motor command u based on the received input $p(x)$ and $v(x)$. The trolley can move forward, backward or stay at the current position. The simulation is in 2D and the trolley is viewed from the side. Therefore now the turning left or right is not taken into account. The trolley can choose from different control actions: move forward or backward with different speed, or stay at the current position. The aim of the controller is to push the trolley back to the safe zone and stay there. Therefore, when the trolley is tilted backwards it needs to move backward. When the trolley is tilted forward it needs to move forward. When the trolley is within the safe zone, then it should stay there.

The aim of the controller is to reduce the cost of error. As already described in the section 3-4, the best way to reduce cost is to choose a control action, which leads to the highest increase in the value

$$u = argmax_u vLp \tag{4-4}$$

where u is the motor command, v is the value function, L is the weight for each state, and p is the probability of being at a certain angle.

At first u is calculated for each neuron, then these motor commands are summed, and the sum is sent to the plant.

For each neuron two motor commands are calculated: a motor command pushing forward u_f and a motor command pushing backward u_b. At first the motor command u_f is calculated

$$u_f = vLp \tag{4-5}$$

where v is the value of being at the given angle, L is the weight of the particular angle, and p is the probability of being at a certain angle. Then a random number is generated. When the random number is smaller than the value of u_f, then the value of u_f will be one for that neuron. Otherwise the value of u_f will be zero for that neuron

$$\alpha\tilde{~}U(0,1), \quad u_f(x) = (\alpha < u_f(x)). \tag{4-6}$$

Then the motor command pushing backward needs to be calculated. This is done in a similar way as the calculation of the motor command pushing forward. The only difference is that before comparing it with the random number the motor command is multiplied with -1

$$u_b = -vLp. \tag{4-7}$$

$$\alpha\tilde{}U(0,1), \quad u_b(x) = (\alpha < u_b(x)) \tag{4-8}$$

Finally, the two motor commands are subtracted from each other

$$u_d(x) = u_f - u_b. \tag{4-9}$$

Therefore, each neuron can generate a +1,-1, or 0 motor command. The comparison of the motor commands with an array of random numbers is necessary to represent the uncertainty in the effect of action. Even if the state is exactly known before the motor command is generated, several new states are possible.

After calculating the motor command for each neuron, the motor commands are summed and sent to the motors to turn the wheels

$$u = \sum u_d. \tag{4-10}$$

In this way the population of neurons generates the motor command.

Then a new tilt angle x^* is received from the IMU board and a new motor command u is calculated.

4-4 Learning own dynamics

The controller has to learn the effect of individual neurons on the dynamics. The dynamics are described by L operator, which connects v and p, where v is the value function and p is the probability of a particular angle. In the beginning our prior assumption does not know anything about the effect of u, therefore the weight L will be zero. This means that the prior assumption about the effect of the motor command is that it does nothing.

The controller generates a prediction about the change in the probability of the state $\hat{\dot{p}}$

$$\hat{\dot{p}} = Lp. \tag{4-11}$$

Therefore, the easiest way to learn L is to observe $\dot{p}$, which is the actual change in the state, and $\hat{\dot{p}}$, which is the predicted change in the state, and update the weight L. The change in the weights can be calculated by

$$\Delta L = \gamma(\dot{p} - \hat{\dot{p}})p \tag{4-12}$$

where $\hat{\dot{p}}$ is the predicted change of p, $(\dot{p} - \hat{\dot{p}})$ is the prediction error and γ is the learning rate.

When the prediction is smaller than the received feedback the weight is increased. When the prediction is bigger than the received feedback the weight is decreased. After a time the error between the feedback and the prediction will become zero. The weight of the particular angle is stored in the neuron which state is nearest to the particular angle.

4-5 Implementation in a simulation

In order to test the efficacy of the risk-aware control, it was implemented and tested in a simulation of the Pull-E, where a planar projection of the tri-wheeled actuated trolley was dynamically simulated.

At first the physical environment had to be built for the simulation of the Pull-E.

Then a graphical interface has been designed to be able to apply different perturbations and forces (figure 4-3). The most important values are visualised on graphs at the top of the screen.

The simulation has been written in the open source software called Processing, in Java language. The jbox2D physics library has been used for the simulation of the physical environment. The jbox2D library is based on the box2D library, which is a widespread physics library for c++. All data underlying the findings described in this thesis can be downloaded from: https://github.com/RekaH/RiskawareControl

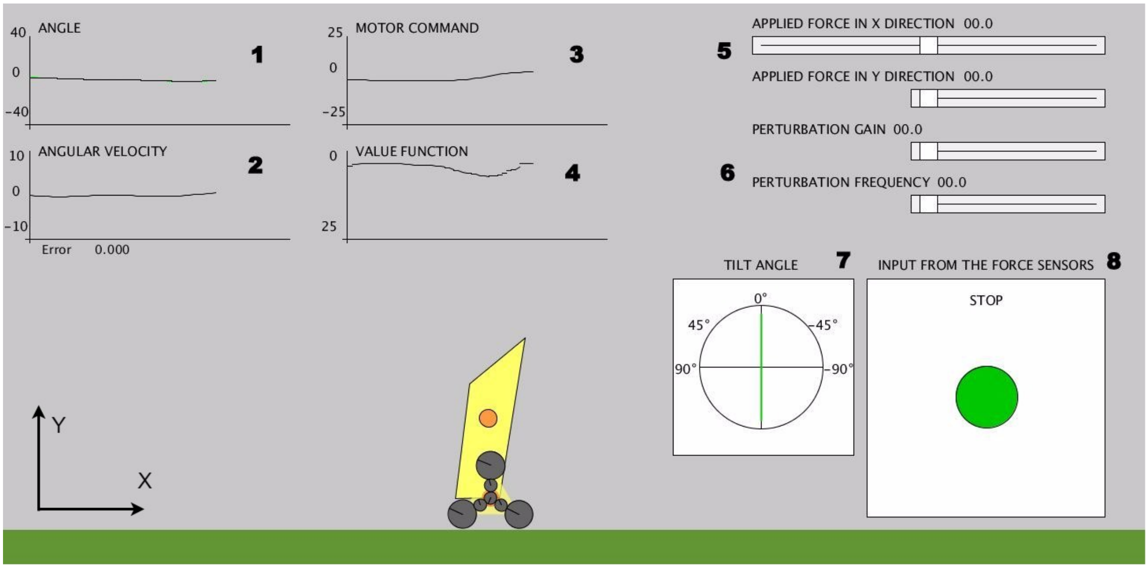

Figure 4-3: The graphical interface of the simulation. The (1) change of the angle, (2) the angular velocity, (3) the motor command, and (4) the value function are visualised at the top left corner of the screen. A (5) horizontal and vertical force can be applied on the trolley at the top right corner of the screen. In addition a (6) sinusoidal perturbation can be applied on the trolley. The (7) data received from the IMU board and (8) the data from the force sensors it is visualised at the bottom right corner of the screen.

The top left graph shows (1) the change of the angle. The bottom left graph shows (2) the angular velocity. The top right graph shows (3) the motor command. Finally, the bottom right graph shows (4) the value function.

In addition, the user can set (5) horizontal and vertical forces acting on the trolley. In addition the system can be perturbed by a (6) sinusoidal signal for which the amplitude nd the frequency can be set as well.

When the trolley receives external input from the Arduino board several new visualisation appear. Therefore, when the trolley is connected to the Arduino (7) the input from the IMU is visualised as well. The measured tilt angle is received from the Arduino board. Then the tilt angle is shown in the simulation.

In addition to the visualisation of the IMU input the (8) input coming from the force sensors is visualised as well. The user can see in the right bottom corner a green circle. When the user presses the force sensor the circle stretches in the corresponding direction. For example, when the user presses the force sensor to move forward the circle stretches upward.

Above all the trolley can be dragged by the mouse as well. The user needs to click on the body of the trolley and then it can be dragged. As soon as the button of the mouse is released the trolley is released as well.

In the following section the risk-aware control will be analysed for different conditions. There are almost limitless conditions for which the risk-aware control could be tested, but the risk-aware control should be able to learn the plants own dynamics and adapt to different conditions and therefore, in the following the controller will be tested only for couple of conditions. At first the controller can receive two different value functions. In the first case, the value function is designed in a way that the desired state is a sinusoidal signal, in the second case, the desired state is constant zero. At second, the plant can be zero-order system (0-ord.) or second-order non-linear system (2-ord.). The block diagram 4-4 shows all the different parameters and conditions which ones are going to be changed to analyse the behaviour of the system.

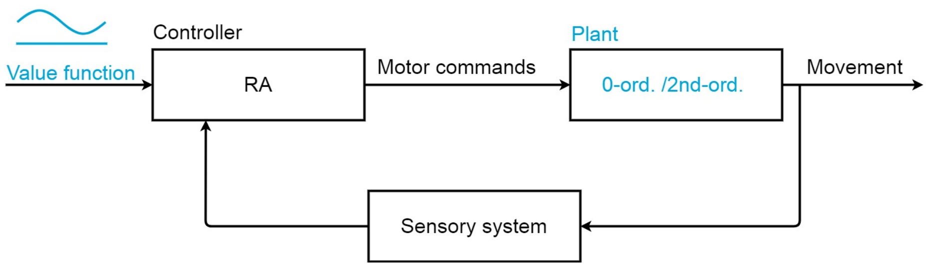

Figure 4-4: The block diagram of all different parameters and conditions which ones are going to be changed to analyse the behaviour of the system. At first the controller can receive two different signals as the desired state. The desired state can be constantly zero or it can be a sinusoidal signal. At second the plant can be a zero-order system or a second-order non-linear system.

4-5-1 Sinusoidal desired state on a zero-order system

In the next section the performance of the risk-aware control will be shown for a sinusoidal desired state on a zero-order system. These conditions are the same as in the article [23]. The conditions are visualised in the block diagram 4-5.

In the figure 4-8 the risk-aware control is shown for a sinusoidal desired state on a zero-order system. The plot A shows the sinusoidal desired state. The plot B shows the measured angle of the zero-order system in blue. The desired angle is shown in red with a dashed line. It can be seen that the system is able to follow the desired path even in case of higher frequencies. The plot C shows the error between the real state and the desired state. For higher frequencies the error slowly increases. The plot D shows the sum of weights. The weights are increasing

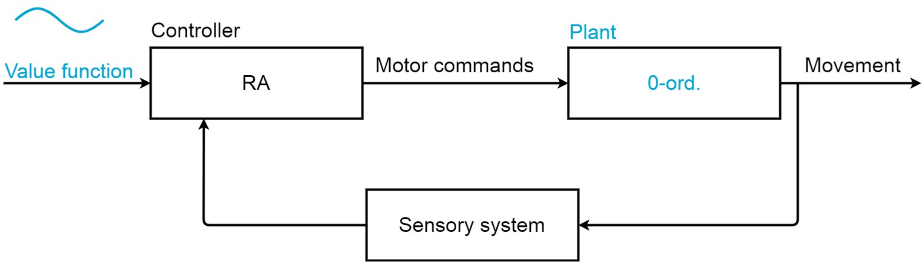

Figure 4-5: The block diagram of the conditions which ones are going to be used to analyse the behaviour of the risk-aware control. The controller receives a sinusoidal signal as the desired state and the plant is a zero-order system.

as the controller is learning. Finally, plot E shows the motor command. As the frequency of the desired state increases the motor command increases as well.

To conclude, the risk-aware control has been implemented on zero-order system for a sinusoidal desired state. The system is able to follow the desired path, even for higher frequencies. The results are the same as in the article [23]. In the following section, this control method will be used on a second-order non-linear system.

4-5-2 Sinusoidal desired state on a 2nd-order system

In the next section the same conditions are going to be used on second-order non-linear system. The used conditions are visualised in the block diagram 4-6.

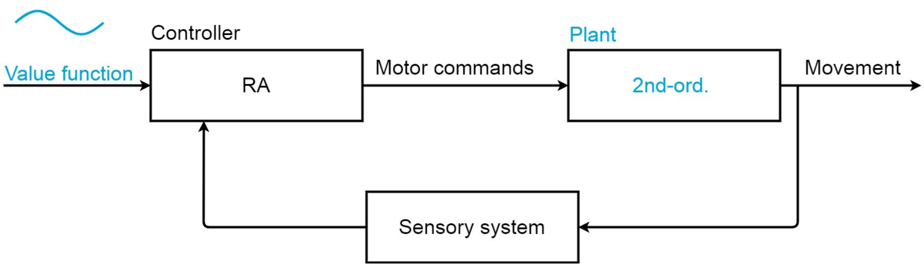

Figure 4-6: The block diagram of the conditions which ones are going to be used to analyse the behaviour of the risk-aware control. The controller receives a sinusoidal signal as the desired state and the plant is a second-order non-linear system.

In the figure 4-9 the risk-aware control is shown for a sinusoidal desired state on a second-order non-linear system. The plot A shows the sinusoidal desired state. The plot B shows the angle of the second-order system in blue. The desired state is shown in red with a dashed line. It can be seen that the system is not able to follow the desired path. It constantly overshoots and falls. When the trolley falls the learning part is stopped, the trolley is lifted to the starting position and the simulation starts again (figure 4-7). As it can be seen from the graph the trolley in spite of the learning it falls repeatedly.

The plot C shows the error between the real state and the desired state. Since the system is not able to follow the desired path the error between the real state and the desired state is huge. The plot D shows the sum of weights. The weights are continuously increasing, as

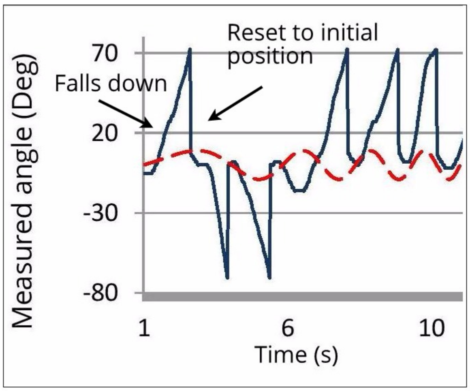

Figure 4-7: Part of the figure 4-9. The trolley is not able to follow the desired path marked with red dashed line and it falls. When the trolley falls it is lifted to the initial position.

the controller is learning. Finally, plot E shows the motor commands. It is interesting to notice, that in spite of the increasing weights the motor commands are not increasing for higher frequencies. The reason for this is that the weights are always compared to an array of random numbers and then they are converted to ones and zeroes. By increasing the weight only the probability increases that, the final weight will be one.

To conclude, the risk-aware control is not able to balance the second-order non-linear system in case of a sinusoidal desired state.

In the following section the desired state of the system will be constant zero. At first, it will be tested on zero-order system, then on second-order non-linear system.

4-5-3 Constant zero desired state on zero-order system

Since, the aim is to keep the trolley upright, the desired angle should be zero. Therefore, in the next section the desired angle will be set to zero. At first, these conditions are going to be tested on zero-order system. The applied conditions are shown in the block diagram 4-10.

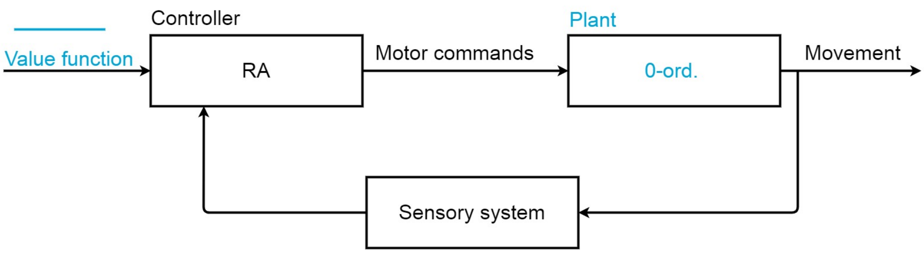

Figure 4-10: The block diagram of the conditions which ones are going to be used to analyse the behaviour of the risk-aware control. The controller receives constant zero signal as the desired state and the plant is a zero-order system.

In the figure 4-12 the risk-aware control is shown for a constant zero desired state on a zero-order system. The plot A shows the zero desired state. The plot B shows the angle of the

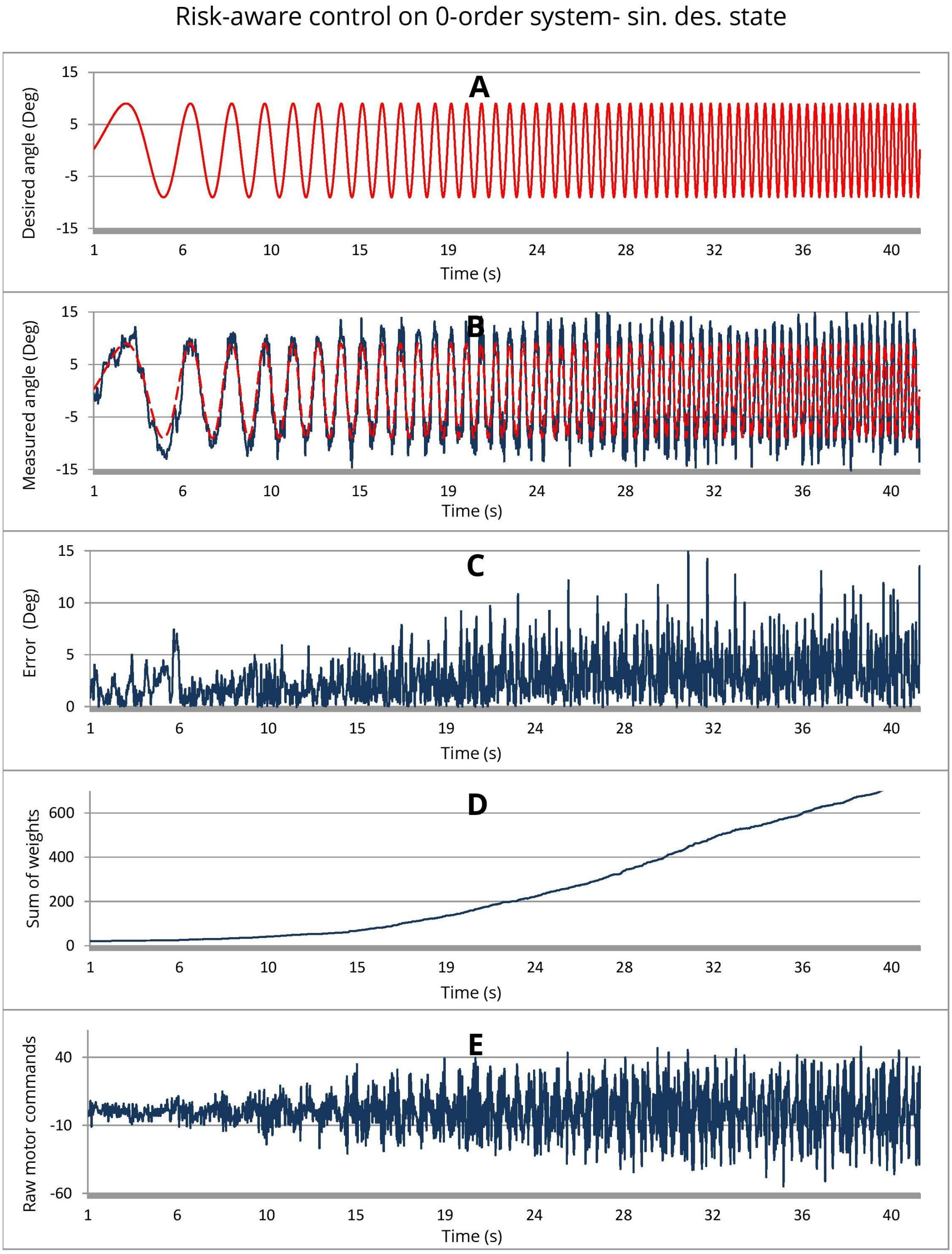

Figure 4-8: Risk-aware control for a sinusoidal desired state on a zero-order system. The plot A shows the sinusoidal desired state. The y axis shows the desired angle. The x axis shows the frequency of the sinusoidal signal. The plot B shows the angle of the zero-order system in blue. The desired state is shown as well in red with dashed line. The plot C shows the error between the real state and the desired state. The plot D shows the sum of weights. Finally, plot E shows the motor command.

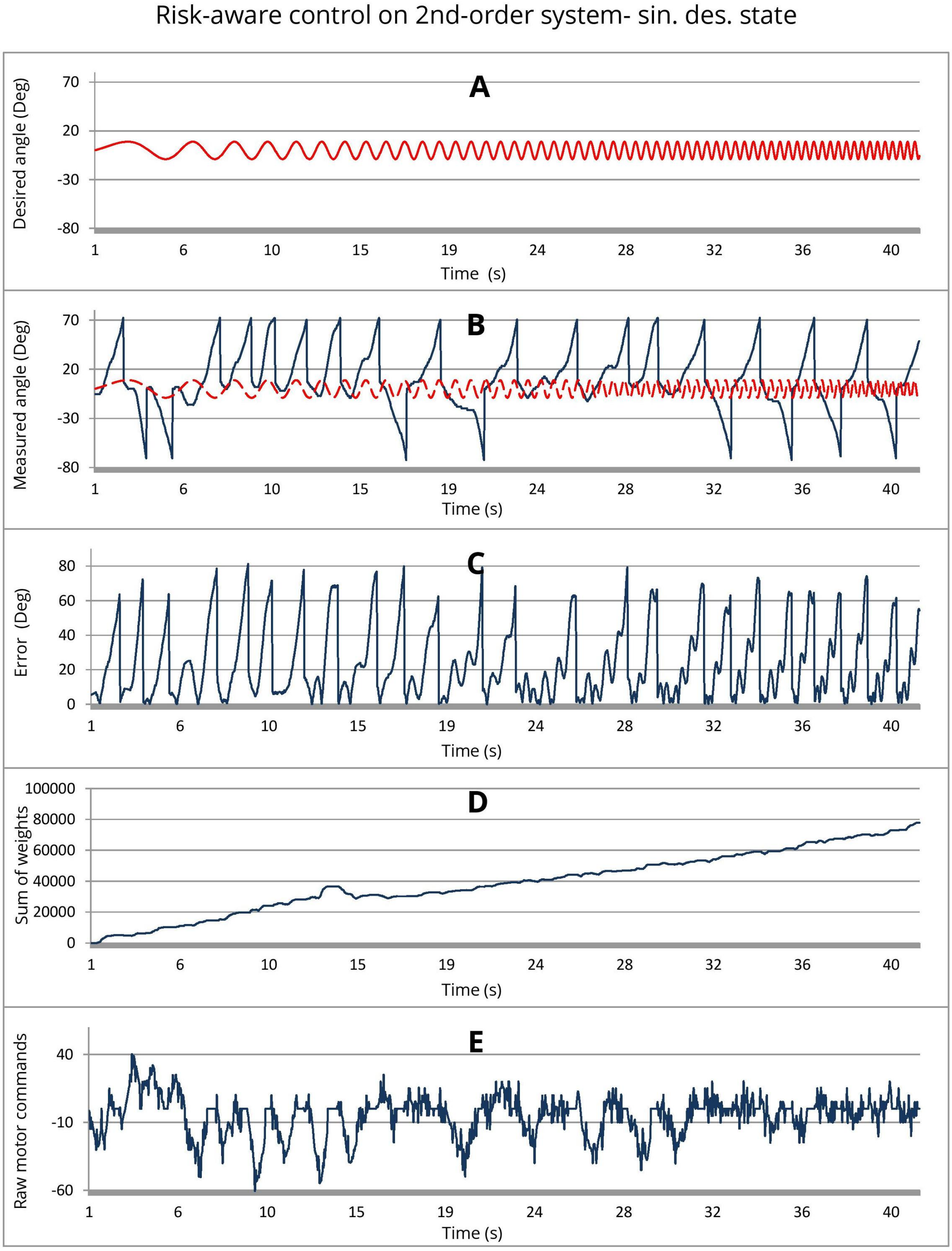

Figure 4-9: Risk-aware control for a sinusoidal desired state on a second-order non-linear system. The plot A shows the sinusoidal desired state. The y axis shows the amplitude of the sinusoidal signal. The x axis shows the frequency of the sinusoidal signal. The plot B shows the angle of the second-order system in blue. The desired state is shown in red. The plot C shows the error between the real state and the desired state. Since the system is not able to follow the desired path the error between the real state and the desired state is huge. The plot D shows the sum of weights. Finally, plot E shows the motor command.

zero-order system in blue. The desired state is shown in red with dashed line. It can be seen that the system is not able to follow the desired state. In the beginning the system crosses the desired state, but then it stays below the desired state. The value function has non zero values only within a small range around the desired state. Therefore, when the angle of the system is around 5 degrees it is out of the range and the state is multiplied by zero. Therefore, the system is not able to move back to the desired position. The plot C shows the error between the real state and the desired state. The plot D shows the sum of weights. The weights are increasing as the system learns and tries to find its equilibrium. Finally, plot E shows the motor command.

To conclude, the risk-aware control is suitable to control a system, which desired state is changing.

4-5-4 Constant zero desired state on 2nd-order system

As already described in the section 4-5-3, the aim is to keep the trolley upright. Therefore, in the next section the desired angle will be set to zero for the second-order non-linear system. These conditions are shown in the block diagram 4-11 for second-order non-linear system.

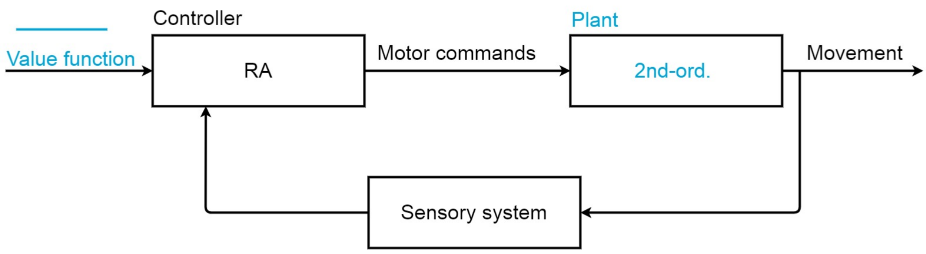

Figure 4-11: The block diagram of the conditions which ones are going to be used to analyse the behaviour of the risk-aware control. The controller receives constant zero signal as the desired state and the plant is a second-order non-linear system.

In the figure 4-13 the risk-aware control is shown for a constant zero desired state on a second-order non-linear system. The plot A shows the zero desired state. The plot B shows the angle of the second-order non-linear system in blue. The desired state is shown in red with dashed line. It can be seen that the system is not able to follow the desired state. When the trolley falls it is reset to its initial position. In spite of the fact that it always starts at the desired state it is not able to improve its performance. The plot C shows the error between the real state and the desired state. The plot D shows the sum of weights. The weights are increasing as the system is learning. Finally, plot E shows the motor command. As already described at the section 4-5-2 the motor commands are not increasing even tough the weights are increasing. This can be explained by the fact, that by increasing the weight only the probability increases that, the final weight will be one.

4-6 Summary

So far, the risk-aware control method was tested only for zero order systems and it turned out that the implementation of the risk-aware control does not include dynamics.

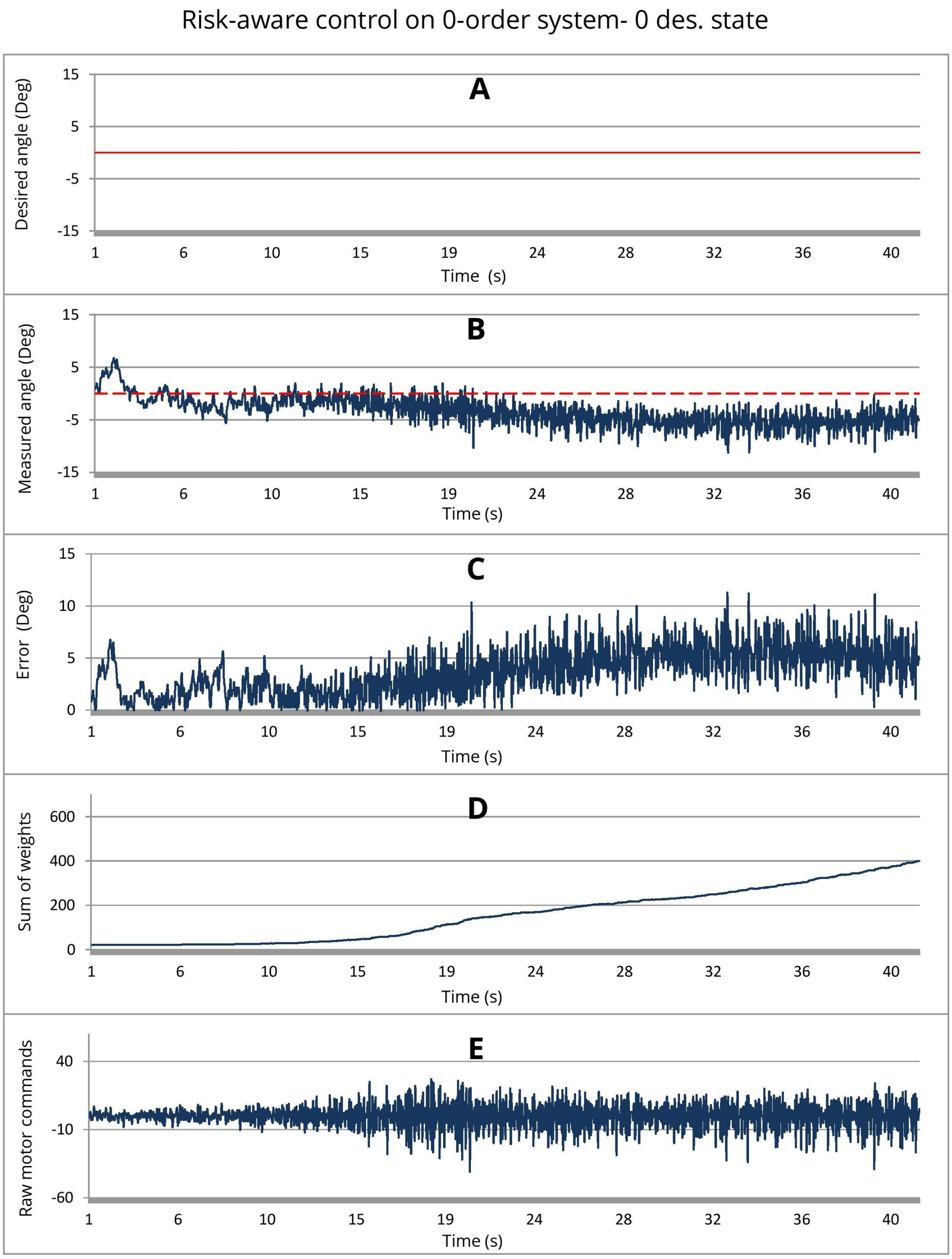

Figure 4-12: The risk-aware control is shown for a constant zero desired state on a zero-order system. The plot A shows the zero desired state. The plot B shows the angle of the zero-order system in blue. The desired state is shown in red with dashed line. The plot C shows the error between the real state and the desired state. The plot D shows the sum of weights. Finally, plot E shows the motor command.

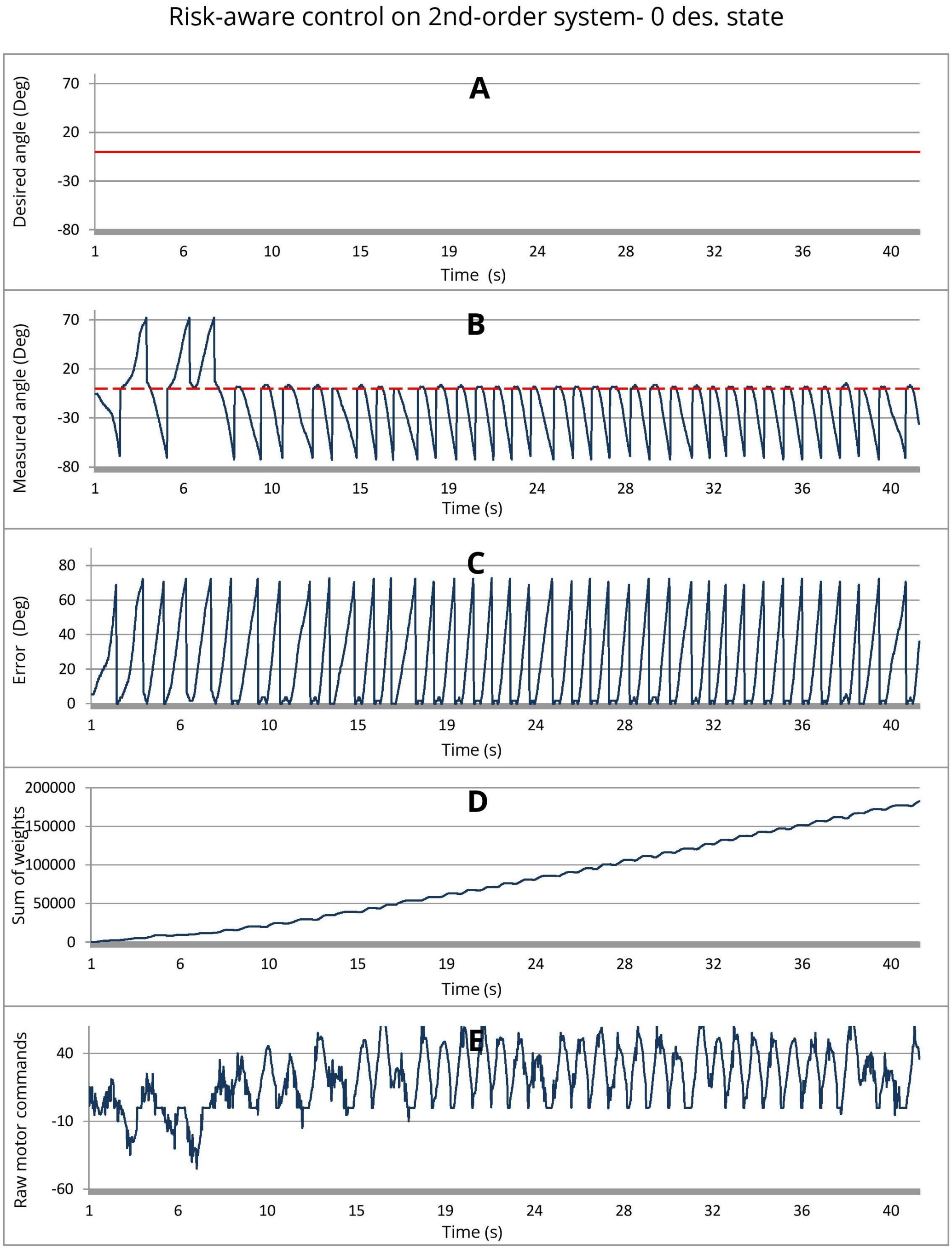

Figure 4-13: The risk-aware control is shown for a constant zero desired state on a second-order non-linear system. The plot A shows the zero desired state. The plot B shows the angle of the second-order non-linear system in blue. The desired state is shown in red with dashed line. The plot C shows the error between the real state and the desired state. The plot D shows the sum of weights. Finally, plot E shows the motor command.

Since, the trolley is a second- order system, an adaptation of the risk-aware control method has to be made to be able to control the 2nd order system. This risk-aware inspired control method is going to be described in more details in the following chapter.

Chapter 5

Risk-aware inspired control method

In the previous chapter it has been concluded that, the risk-aware control needs to be changed to be suitable for stabilising a 2nd order system. These changes are going to be described in this chapter.

The risk-aware inspired control is implemented on a neural network. For the trolley 180 neurons were generated. Each neuron has a corresponding angle from -90 to 90 denoted by x. The precision of the sensor, which is used to measure the angle is around 1 degree, therefore generating more neurons would not improve the performance of the prototype. Although, in the simulation it might improve the performance, but that would be misleading. In addition to the angle each neuron stores its weight denoted by L.

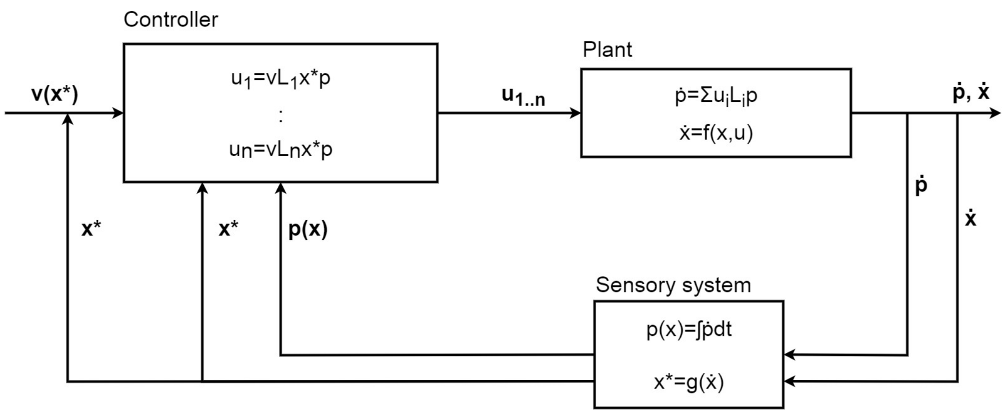

Figure 5-1: Block diagram of risk-aware inspired control. The value function $v(x)$, which describes the desired states is the input of the controller. The controller receives the probability function $p(x)$, and the measured angle x^* as well. Based on the input the motor command is generated. The motor command is sent to the plant, which results in the change of the state. The new probability function based on the new state is fed back to the controller. In addition the measured state is fed back to the controller and it is included in the value function as well.

5-1 State

The controller receives the state of the trolley. The state of the trolley is the measured angle x^* , the probability distribution of the angles $p(x)$, and the angular velocity x_v. From the angle the probability distribution of the angles $p(x)$ is calculated in the same way as it has been described in the risk-aware control chapter, section 4-1. In addition, the angular velocity x_v is calculated from the current angle, the previous angle, and the elapsed time. In this case the state of the trolley includes the angle as well not only its probability distribution. In addition, the angular velocity is part of the state as well.

5-2 Value function

In addition to the state, the trolley receives a value function $v(x)$. The higher absolute value of the value function means that a state has a high cost, it is undesirable (unsafe). When a state is desirable (safe) the value of $v(x)$ is zero. In this case the value function is more complex than the value function used in the risk-aware control. In case of the risk-aware control only the probability distribution of the angles has been taken into account. In case of the trolley it has to be able to balance and therefore the angular velocities have to be taken into account. The value function in this case is

$$v = x^* A + x_v B \tag{5-1}$$

where x^* is the measured angle, x_v is the angular velocity, A is the gain of the state, and B is the gain of the angular velocity.

5-3 Motor command

The next step is to calculate the motor command of each neuron based on the received input. The aim of the control is to choose a control action, which reduces the cost of error. As already described in the risk-aware control chapter, the best way to reduce cost is to choose a control action, which leads to the highest increase in the value

$$u_i = argmax_{u_i} v_i L_i x^* \tag{5-2}$$

where u_i is the motor command of the ith neuron, v_i is the value function of the ith neuron, L_i is the weight of the ith neuron, and x^* is the measured angle. For each neuron a motor command is calculated. Since each neuron has a corresponding angle each motor command is multiplied by the probability of the corresponding angle

$$u = \sum u_i p(x). \tag{5-3}$$

Hence, each motor command will be proportional to the probability of the corresponding angle. Therefore, the angles which ones has zero probability will have no effect on the resulting motor command. Finally, the motor commands are summed and sent to the motors. This way each neuron is activated proportional to its probability. Then the trolley moves and a new tilt angle x^* is received from the IMU board. Based on the new angle a new motor command u is calculated. Finally, from any angle the trolley is moved back to the safe zone.

5-4 Learning own dynamics

In the following, the risk-aware inspired learning method will be described in more details. As already described in the chapter 4, the controller has to learn the effect of the motor commands on the dynamics.

In order to avoid the hard inference problem at first the weights, then the gains of the value function are learned. When the measured state is bigger than the previous measured state and the difference between the two states is bigger than a threshold value, then the weight needs to be increased by ΔL

$$\Delta L = \gamma(|x_t^* - x_{t-1}^*|)p(x) \tag{5-4}$$

where x_t^* is the current measured state, x_{t-1}^* is the previous measured state, γ is the learning rate, and $p(x)$ is the probability of the angle x^*. When the angle of the trolley is increasing it is undesired, therefore, in this case, the motor command needs to be increased. On the other hand, when the motor command is too big the trolley overshoots the safe zone. In this case the weight needs to be decreased by ΔL. Finally, if the weights are the same for a half minute the tuning of the weights is done. Since this means that the weights could not be further tuned. Then the tuning of the value function gains starts, which is described in the next section.

5-5 Learning the gains of the value function

As already described in the section 5-2, the value function used in the case of the trolley is much more complex than the value function used in case of the risk-aware control. The value function

$$v = x^*A + x_vB \tag{5-5}$$

where x^* is the measured angle, x_v is the angular velocity, A is the gain of the angle, and B is the gain of the angular velocity. As already mentioned in the learning section, in order to avoid the hard inference problem at first the weights, then the gains of the value function are learned. Therefore, when the learning of the weights is done, the adjusting of gain A starts. At first the gain will be decreased until the change in the state decreases. When the change in the state does not decrease anymore the gain A will be increased. Hence, the gain which results in the minimum change in the state can be found. In the previous section it has been descried that when the motor command is too big the trolley overshoots the safe zone. Therefore when the tuning of the gain A starts at first it is decreased, when the change in the state does not decrease anymore the gain A is increased. In the next paragraph this process will be described in more details.

At first the gain A is decreased by ΔL. When the resulting change in the state is smaller than the previous one the gain is decreased even more. Otherwise, the gain A is increased by ΔL. When the performance cannot be further improved by decreasing the gain A the gain is increased and the experimenting with increasing the gain starts. When the performance of the system cannot be further improved by increasing the gain A the tuning of gain A is done. After the tuning of the gain A is done, the tuning of the gain B starts. The tuning of gain A and gain B at the same time would result in the hard inference problem. Therefore, before the tuning of gain B starts, the tuning of gain A is finished. The tuning of gain B is the same as the tuning of gain A.

After the gains are learned the weights can be tuned again. This is necessary in case the system is perturbed and it needs to accommodate to it.

5-6 Implementation in simulation

In the next section the performance of the risk-aware inspired control method will be analysed. As already described in the risk-aware control chapter, there are almost limitless conditions for which the controller could be tested. The risk-aware inspired control should also be able to learn the plants own dynamics and adapt to different conditions and therefore, in the following the controller will be tested only for couple of conditions.

The block diagram 5-2 shows all the different parameters which ones are going to be changed for the risk-aware inspired control. For the trolley the desired state is to stay at zero angle. Therefore the desired state for the risk-aware inspired controller will be zero. The trolley is a second-order non-linear system, therefore the risk-aware inspired control will be applied only on the second-order system. The performance of the risk-aware inspired control will be analysed for three different cases. At first, there is no perturbation affecting the system. In this case the performance of the risk-aware inspired control method can be compared to the performance of the risk-aware control. The simulation showed the risk-aware control method already resulted in instabilities when no perturbations affected the system, therefore, the risk-aware control was not tested for cases when perturbation is added to the system. At second, a sinusoidal signal with different frequencies perturbs the system. At third, a step function with different amplitudes perturbs the system.

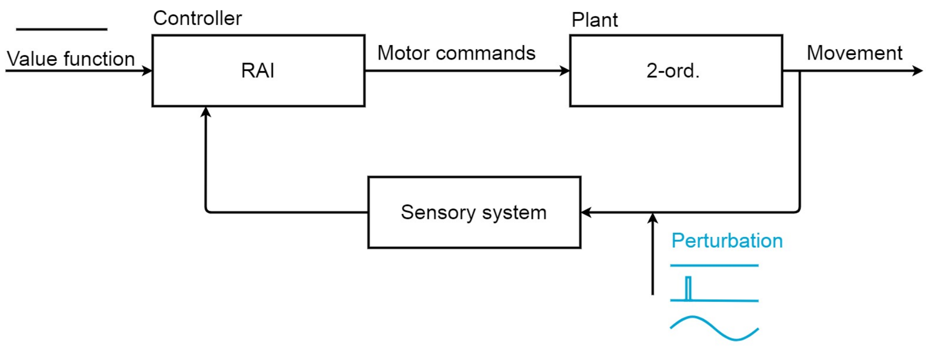

Figure 5-2: The block diagram of all different parameters and conditions which ones are going to be changed to analyse the behaviour of the system. The desired state in case of the risk-aware inspired control is zero. The plant is a second-order non-linear system. Since the aim is to control the trolley, which is a second-order non-linear system. Finally, the plant might be perturbed. When the system is not perturbed, the perturbation is constant zero. When the system is perturbed two different type of perturbations are used: a sinusoidal or an impulse signal.

5-6-1 Risk-aware inspired control in case of no perturbation

At first, no perturbation will be applied on the system. The conditions of the system are shown in the block diagram 5-3.

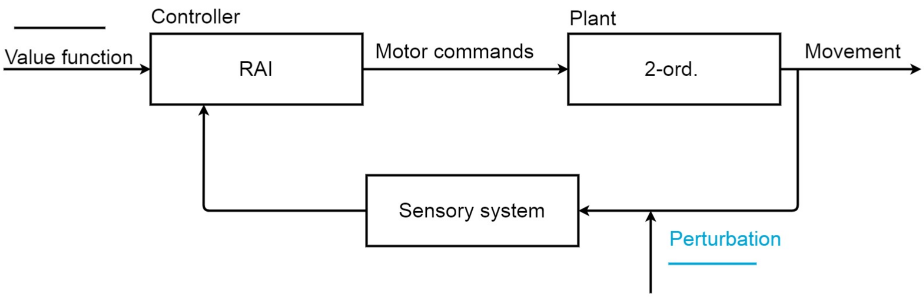

Figure 5-3: The block diagram of risk-aware inspired control, for the case when the system is not perturbed. The desired state in case of the risk-aware inspired control is zero. The plant is a second-order non-linear system. Since the aim is to control the trolley, which is a second-order non-linear system. Finally, the perturbation in this case is zero.

In the figure 5-6 the risk-aware inspired control is shown when no perturbation affects the system. The plot A shows the perturbation affecting the system. The plot B shows the angle of the second-order non-linear system in blue. The desired state is shown in red with dashed line. In case of the risk-aware inspired control, the system is able to balance and it does not fall, which is a great difference in comparison to the risk-aware control! The plot C shows the error between the real state and the desired state. The error does not decrease, but does not increase either. The plot D shows the sum of weights. The weights are increasing as the error is not decreasing. Although after a while as the error cannot be further decreased the weights settle. Finally, plot E shows the motor command. The motor command is not increasing, since the weights are only slightly increased.

To conclude, the risk-aware inspired control is able to balance the trolley, when there is no perturbation. However, the system does not stay at the desired angle, it oscillates between -8 and 8 degrees.

5-6-2 Risk-aware inspired control in case of sinusoidal perturbation

In contrast to the risk-aware control, the risk-aware inspired control was able to balance the trolley. Therefore, in the next section the risk-aware inspired system will be tested for different perturbations. At first, the system is perturbed by a sinusoidal signal. The block diagram 5-4 shows the conditions used in this case.

In the figure 5-7 the risk-aware inspired control system is shown during sinusoidal perturbation. The plot A shows the sinusoidal perturbation affecting the system. The plot B shows the angle of the second-order non-linear system in blue. The desired state is shown in red with dashed line. In case of low frequency perturbations the system is able to balance. For higher frequency perturbations the system falls. The plot C shows the error between the real state and the desired state. The error is constantly increasing. The plot D shows the sum of weights. The weights are increasing as the change in the state is increasing. Although after a

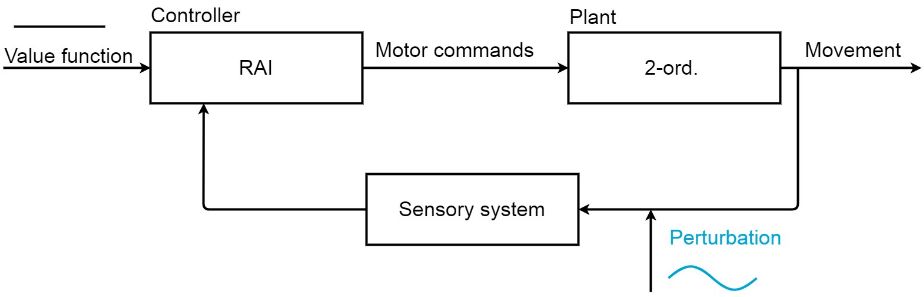

Figure 5-4: The block diagram of risk-aware inspired control, for the case when the system is perturbed by a sinusoidal signal. The desired state in case of the risk-aware inspired control is zero. The plant is a second-order non-linear system. Since the aim is to control the trolley, which is a second-order non-linear system. Finally, the perturbation in this case is a sinusoidal signal.

while as the change in the state is still increasing the controller tries to decrease the weights. Finally, plot E shows the motor command. The motor command increases as the angle of the plant increases.

To conclude, the risk-aware inspired control is able to balance the trolley for lower frequencies. When the frequency of the perturbation is the same as the trolley's eigenfrequency the oscillation of the trolley is amplified, and the trolley falls.

5-6-3 Risk-aware inspired control in case of impulse function perturbation

In the next section the system is perturbed by several impulse functions. The block diagram 5-5 shows the conditions used in this case.

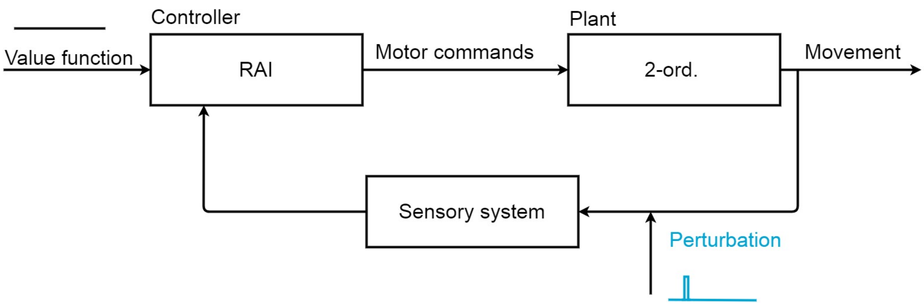

Figure 5-5: The block diagram of risk-aware inspired control, for the case when the system is perturbed by a sinusoidal signal. The desired state in case of the risk-aware inspired control is zero. The plant is a second-order non-linear system. Since the aim is to control the trolley, which is a second-order non-linear system. Finally, the perturbation is an impulse signal with increasing amplitude.

In the figure 5-8 the risk-aware inspired control system is perturbed by several impulse functions with increasing amplitude. The plot A shows the impulse signal applied on the system. The plot B shows the angle of the second-order non-linear system in blue. The desired state is shown in red. When the amplitude of the impulse function is lower than 1.4 Newton the trolley is able to balance. When the amplitude of the impulse function is higher than 1.4 Newton the trolley falls. The plot C shows the error between the real state and the desired state. The error stays the same. The plot D shows the sum of weights. The weights are increasing as the change in the state is increasing. When the amplitude of the impulse function is higher than 1.4 Newton, the change in the state is increased, even tough that the weights are increased. Therefore, the controller tries to decrease the weights. Finally, plot E shows the motor command. To conclude, the trolley is able to resist perturbations lower than 1.4 Newton.

5-7 Summary

To summarize, an adaptation of the risk-aware control method was made to allow the 2nd order system, by including angular velocities in the state and modifying the value function. This control method is able to balance the assistive trolley for forces that cause lower angular perturbations than 15 degrees.

In the future this control method needs to be improved to become more robust against perturbation.

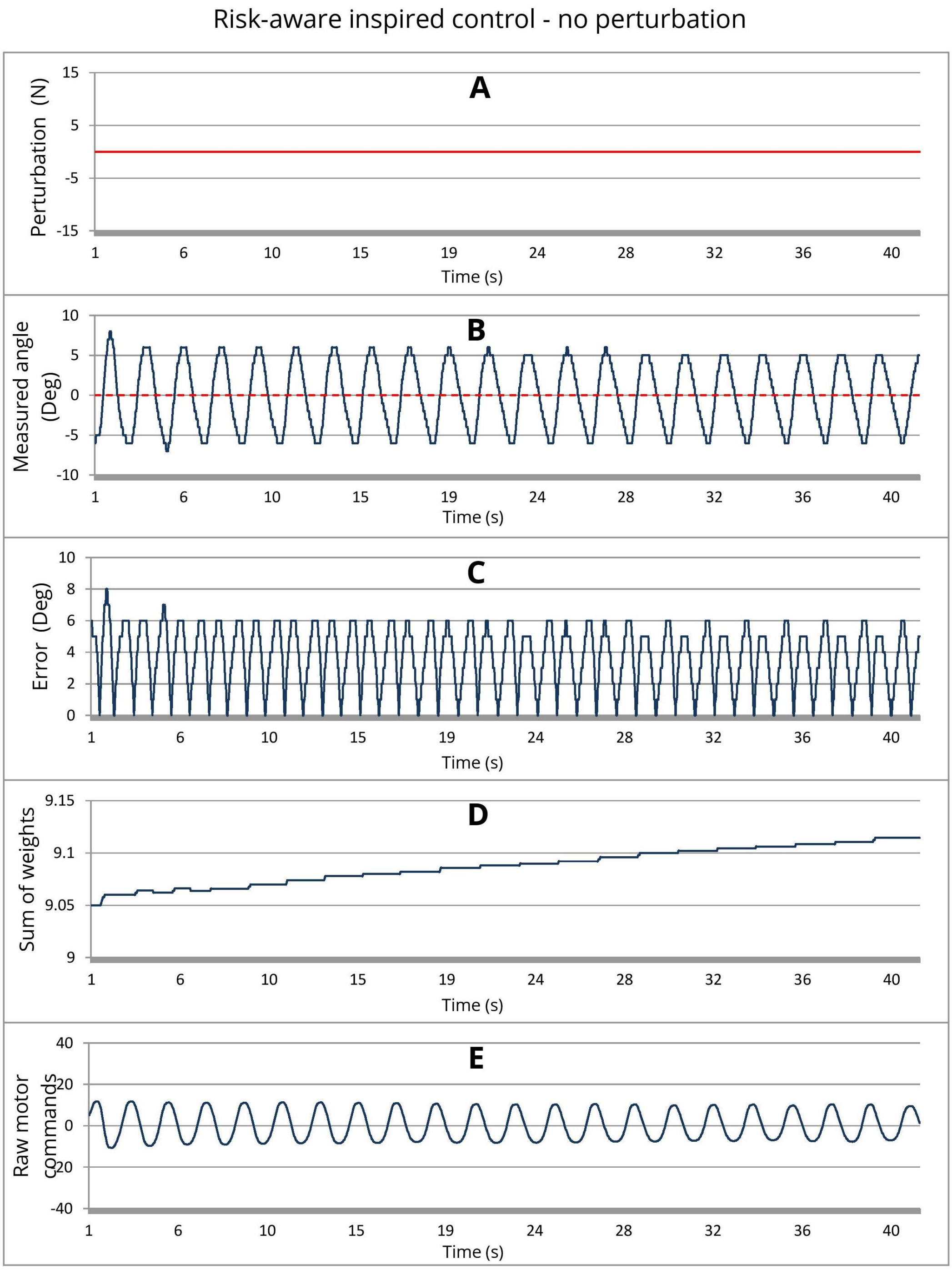

Figure 5-6: The risk-aware inspired control is shown when no perturbation affects the system. The plot A shows the perturbation. The plot B shows the angle of the second-order non-linear system in blue. The desired state is shown in red. The plot C shows the error between the real state and the desired state. The plot D shows the sum of weights. Finally, plot E shows the motor command.

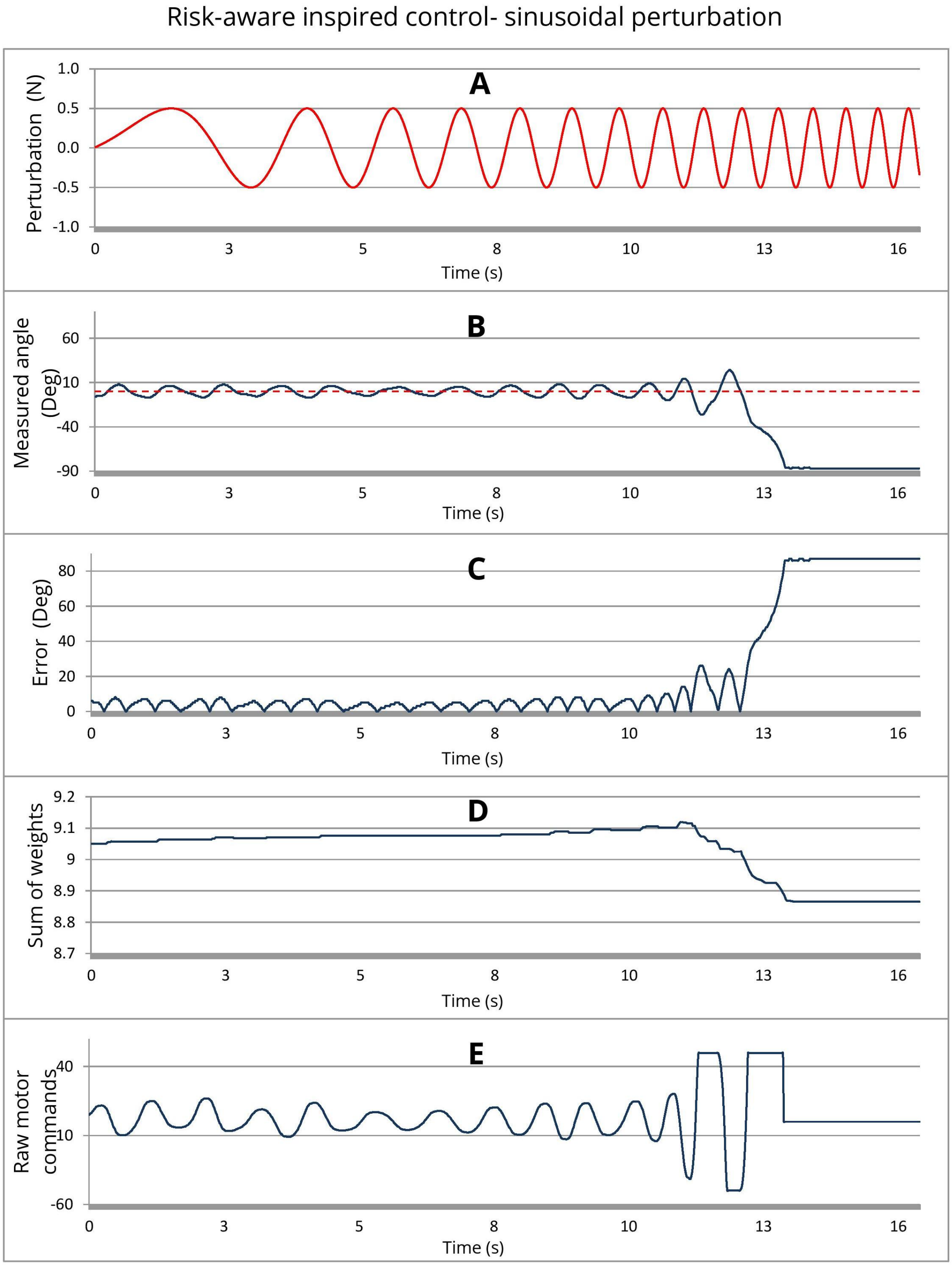

Figure 5-7: The risk-aware inspired control system is shown during sinusoidal perturbation. The plot A shows the perturbation. The plot B shows the angle of the second-order non-linear system in blue. The desired state is shown in red. The plot C shows the error between the real state and the desired state. The plot D shows the sum of weights. Finally, plot E shows the motor command.

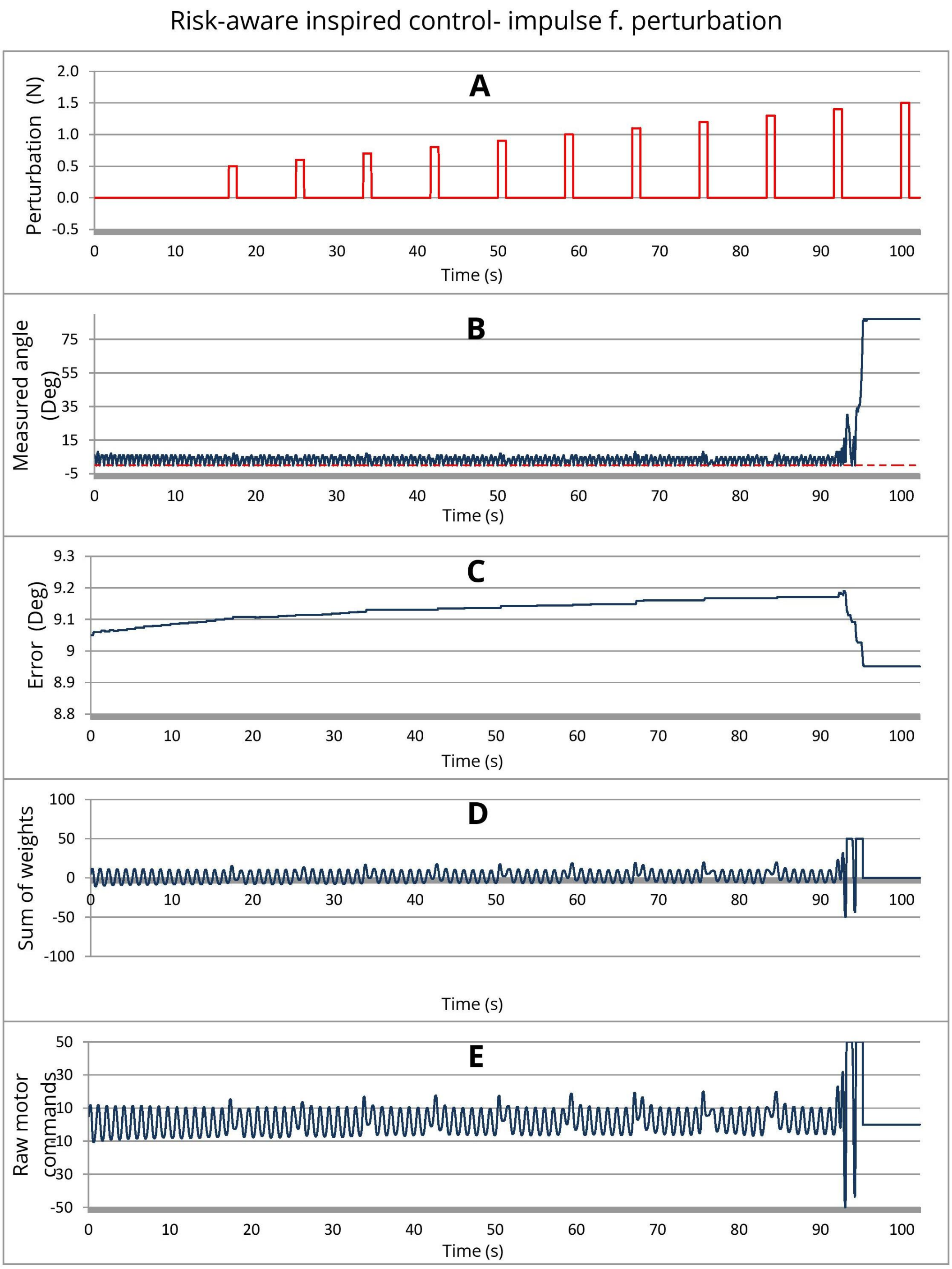

Figure 5-8: The risk-aware inspired control system is perturbed by several impulse functions with increasing amplitude. The plot A shows the perturbation. The plot B shows the angle of the second-order non-linear system in blue. The desired state is shown in red. The plot C shows the error between the real state and the desired state. The plot D shows the sum of weights. Finally, plot E shows the motor command.

Chapter 6

Conclusion

6-1 Aim of the thesis

The aging society puts an increasingly heavy demand on the health care system. Elderly often have difficulties in mobility, reducing opportunities for exercise and social interaction that activities like running errands or shopping normally offer.

This thesis proposes a novel concept for an assistive motorised shopping trolley, called Pull-E. Based on interview results with elderly (n=40) the Pull-E should be able to carry heavy loads (~20 kg) and still move easily and lightly while pulled over flat ground, curbs, and even stairs; and should ideally also be able to move autonomously, without being pulled. It should therefore have capabilities to balance itself, climb the stairs, carry items, and resist perturbations.

6-2 Design conclusions

First, a small proof-of-concept of the Pull-E was built, using tri-wheels which are separately powered by motors, that allow assistive stair-climbing when the Pull-E is pulled horizontally. In order to balance itself the tilt angle of the trolley needs to be measured. Therefore, the Pull-E was equipped with a gyroscope and an accelerometer. The real size prototype is not able to climb the stairs yet, because more powerful motor would be needed. On the other hand all the features has been tested on the smaller prototypes, which means that the real size trolley could work as well with proper motors. From this prototype one of the most challenging aspects emerged: stabilising the trolley.

6-3 Control system conclusions

The next step was to choose a suitable control system. Since humans can adapt to the movement of other humans, climb the stairs, and resist perturbations of other humans, the human inspired control theories have been investigated.

Based on the literature search the risk-aware control was the most promising one. In case of risk-aware control a value function is provided to the system, by which it can avoid the undesirable states. By using a value function it does not require the calculation of exact trajectories and therefore, it becomes computationally more efficient, than the other human inspired control methods. In addition it learns and adjusts its control parameters by which it can adapt to different situations.

In order to test its efficacy, it was implemented and tested in a simulation of the Pull-E, where a planar projection of the tri-wheeled actuated trolley was dynamically simulated. Force inputs could act on the trolley by using the mouse, as well as pre-determined sinusoids acting on its centre of mass. The simulation showed the risk-aware control method already resulted in instabilities when no perturbations affected the system. So far, the risk-aware control method was tested only for zero order systems. Therefore an adaptation of the risk-aware control method was made to allow the 2nd order system, by including angular velocities in the state and modifying the value function. In addition, the mechanism of learning had to be change as well. This control method is able to balance the assistive trolley for forces that cause lower angular perturbations than 15 degrees. To conclude, the risk-aware inspired control method is able to balance the trolley, although it needs to be improved to become more robust against perturbations.

6-4 Future work

In the future this control method needs to be improved to become more robust against perturbation. The prototype needs to be rebuilt with stronger motors, and the control system needs to be tested on the prototype. In addition, the control system needs to be able to react on the input from the human received through the force sensors. Finally, this prototype should be evaluated when human is in the loop.

To conclude, Pull-E can facilitate elderly's longer independence, but still a lot of work needs to be done before it becomes a viable product.

6-5 Data availability

All data underlying the findings described in this thesis can be downloaded from:
https://github.com/RekaH/RiskawareControl

Appendix A

Questionnaire

The aim of this questionnaire is to gather more information about the habits of elderly in order to develop an assistive robotic device for them.

Tasks of daily living:

1. Do you go for shopping?
    - How often do you do it?
    - When did you do it for the last time?
    - Have you done it alone or with help?
    - Is there any problematic part?
    - Have you ever used the online shopping?
    - Do you use a shopping trolley?
    - Do you have pain or any difficulties when the trolley or bag is heavy?
2. Do you use transportation?
    - Do you have your own car?
    - Does anybody provide you transportation?
    - Do you use public transport?
    - When do you use transportation?
3. Do you go for walk?
    - Do you use a walker?
    - How often do you go for a walk?
    - Do you walk alone or with a partner?

- Do you have a shortness of breath during walk?
- Is there any problematic part?

4. Do you do any sports, gardening?
   - How often do you do it?
   - When did you do it for the last time?
   - Do you sport alone or with a partner?
   - Is there any problematic part?
5. Do you do housework?
   - How often do you do it?
   - When did you do it for the last time?
   - Do you do it alone or with help?
   - Is there any problematic part?
6. Do you have any hobbies?
   - How often do you do it?
   - When did you do it for the last time?
   - Do you do it alone or with a partner?
   - Is there any problematic part?
7. Do you participate in social activities?
   - Are you a member of social club, church club, hobby group, etc?
   - How often do you take part in these activities?
8. Do you meet your friends or family regularly?
   - How often do you meet them?
   - When did you meet them for the last time?
   - Is there any problematic part?
9. Is there any activity what you do on daily basis and I did not mention?

Technology related questions:

1. Can you use the telephone?
   - How often do you use it?
   - When did you use it for the last time?
   - Do you need help to use it?
   - Is there any problematic part?

2. Do you watch TV?
    - How often do you watch it?
    - When did you watch it for the last time?
    - Is there any problematic part?
3. Can you use the computer?
    - How often do you use it?
    - When did you use it for the last time?
    - Do you use it alone or with help?
    - Is there any problematic part?
4. What is your opinion on the following devices?
    - Which one do you like?
    - Which one would you use to do shopping?
    - What are the important for criteria for you?

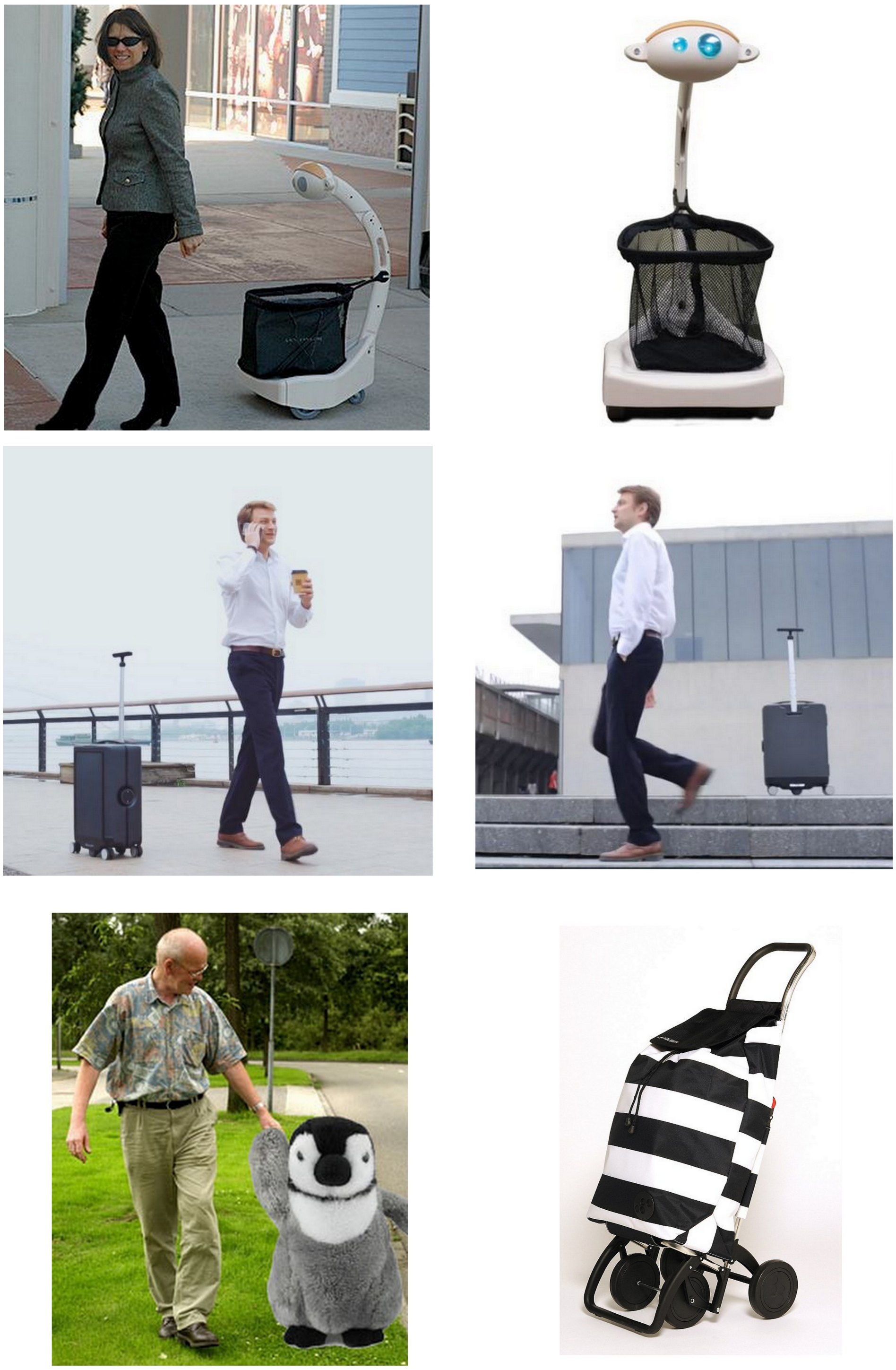

Figure A-1: Robotic devices assisting elderly

Appendix B

Wheel types

In the following section a list of several wheel types is included, which ones have been compared to choose the wheel type which has the best obstacle-crossing ability.

1. **Integration wheel**
   The integration wheel has a simple and reliable design, which integrates the driving and steering sensors in the wheel. The biggest disadvantage is that it can not cross obstacles higher than it's half radius [1].

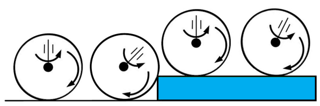

Figure B-1: Integration wheel crossing an obstacle [1]

2. **Exterior planetary wheel**
   It is able to cross obstacles by turning around the wheel frame, by which it has excellent obstacle-crossing ability. The steering of the robot can be realized by differential motion of the wheels [1].

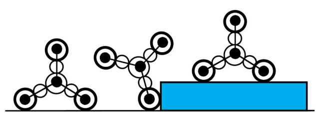

Figure B-2: Exterior planetary wheel crossing an obstacle [1]

3. **Planetary tracked wheel**
   Combines the advantages of the track and the exterior planetary module. When the obstacle is small it has a better ground adaptability than the simple exterior planetary wheel. In addition when the obstacle is high it turns around its axle and crosses the obstacle like the exterior planetary wheel [1].

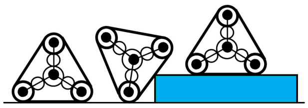

Figure B-3: Planetary tracked wheel crossing an obstacle [1]

4. **Omni wheel**
   The biggest advantage of omni wheel is that it is holonomic. In case of non-holonomic constraints more complex planning is needed. On the other hand the control of the omni-wheel is complex and it's ability to cross obstacles is poor [24].

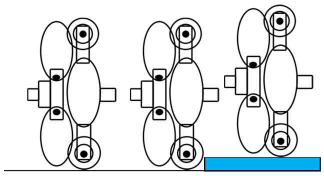

Figure B-4: Omni wheel crossing an obstacle [own figure]

5. **Star shaped wheel**
   The star shaped wheel has excellent obstacle-crossing ability just like the exterior planetary wheel. In addition while driving with high velocities it behaves like a wheel [25].

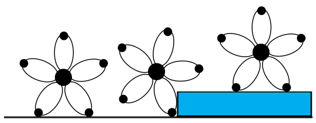

Figure B-5: Star shaped wheel crossing an obstacle [own figure]

6. **Regular tracked wheel**
   Tracked wheels can operate through diverse terrain. Its biggest advantage is that it provides greater surface in comparison to regular wheels, by which it is more stable [26].

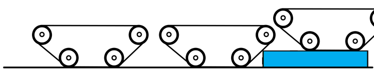

Figure B-6: Regular tracked wheel crossing an obstacle [own figure]

7. **Zebro type legs**
   The zebro robot can overcome rough terrain by switching gait. In addition the optimal gait switching can be easily achieved by switching Max-Plus linear system [27].

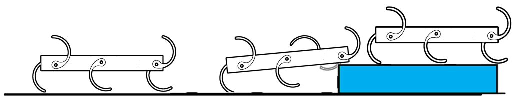

Figure B-7: Zebro type legs crossing an obstacle [own figure]

The exterior planetary wheel has been chosen for my device, because it has excellent obstacle-crossing ability. The tracked planetary wheel could be a good choice as well, but in everyday situations it would be a bit strange and elderly prefer the designs they are more used to.

Appendix C

Force sensing resistor details

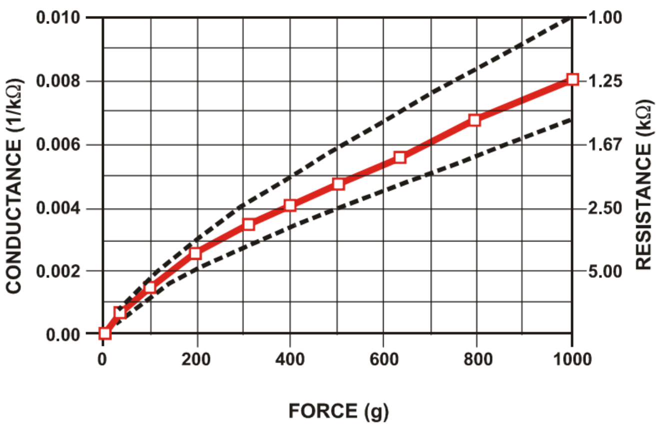

Figure C-1: Force conductance graph [6]

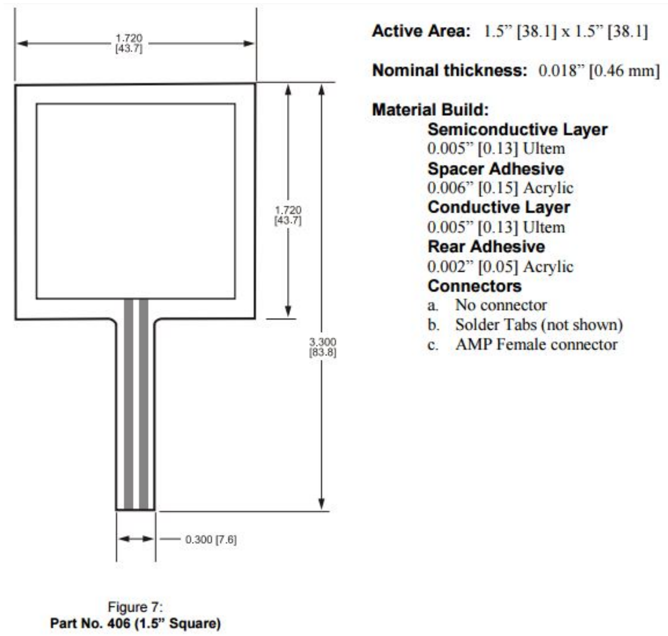

Figure C-2: Descriptions and dimensions of the force sensor [6]

Bibliography

[1] Z. Luo, J. Shang, and Z. Zhang, "Innovative design of six wheeled space exploration robot using module combination," in *Mechatronics and Machine Vision in Practice (M2VIP), 2012 19th International Conference*, pp. 460–465, IEEE, 2012.

[2] electrical4u, "Historical review and types of control engineering," 2011.

[3] M. Haruno, D. M. Wolpert, and M. Kawato, "Multiple paired forward-inverse models for human motor learning and control," *Advances in neural information processing systems*, pp. 31–37, 1999.

[4] B. Kappen, "Integrating control, inference and learning. is it what the brain does?," 2015.

[5] T. D. Sanger, "Risk-aware control," *Neural computation*, 2014.

[6] I. Electronics, "Force sensing resistor integration guide and evaluatin parts catalog," 2016.

[7] H. Lehmann, D. Syrdal, K. Dautenhahn, G. Gelderblom, S. Bedaf, and F. Amirabdollahian, "What should a robot do for you?-evaluating the needs of the elderly in the uk," in *The 6th International Conference on Advances in Computer–Human Interactions, Nice, France, February*, pp. 83–88, 2013.

[8] M. H. A. Wahab, A. A. Talib, H. A. Kadir, A. Johari, A. Noraziah, R. M. Sidek, and A. A. Mutalib, "Smart cane: assistive cane for visually-impaired people," *arXiv preprint arXiv:1110.5156*, 2011.

[9] S. Dubowsky, F. Genot, S. Godding, H. Kozono, A. Skwersky, H. Yu, and L. S. Yu, "Pamm-a robotic aid to the elderly for mobility assistance and monitoring: a âĂIJhelping-handâĂİ for the elderly," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1, pp. 570–576, IEEE, 2000.

[10] M. M. Martins, C. P. Santos, A. Frizera-Neto, and R. Ceres, "Assistive mobility devices focusing on smart walkers: Classification and review," *Robotics and Autonomous Systems*, vol. 60, no. 4, pp. 548–562, 2012.

[11] M.-F. Chang, W.-H. Mou, C.-K. Liao, and L.-C. Fu, "Design and implementation of an active robotic walker for parkinson's patients," in *SICE Annual Conference (SICE), 2012 Proceedings of*, pp. 2068–2073, IEEE, 2012.

[12] N. of Excellence Robotic & Mechatronics HomeLab Community, "How to convert force sensor values to newton," 2013.

[13] A. Ibrahim and R. R. Alexander, "Control systems in robotics: A review," *International Journal of Engineering Inventions*, vol. 5, pp. 29–38, 2016.

[14] L. Fridman, A. S. Poznyak, F. J. Bejarano, *et al.*, *Robust output LQ optimal control via integral sliding modes.* Springer, 2014.

[15] L. Rollins, "Robust control theory," 1999.

[16] E. Todorov, "Optimality principles in sensorimotor control," *Nature neuroscience*, vol. 7, no. 9, pp. 907–915, 2004.

[17] D. M. Wolpert, "Probabilistic models in human sensorimotor control," *Human movement science*, vol. 26, no. 4, pp. 511–524, 2007.

[18] A. Dearden and Y. Demiris, "Learning forward models for robots," in *IJCAI*, vol. 5, p. 1440, 2005.

[19] A. J. Nagengast, D. A. Braun, and D. M. Wolpert, "Risk-sensitive optimal feedback control accounts for sensorimotor behavior under uncertainty," *PLoS Comput Biol*, vol. 6, no. 7, p. e1000857, 2010.

[20] R. Shadmehr, "The equilibrium point hypothesis for control of movement," *Baltimore, MD: Department of Biomedical Engineering, Johns Hopkins University*, 1998.

[21] I. V. Blagouchine and E. Moreau, "Control of a speech robot via an optimum neural-network-based internal model with constraints," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 142–159, 2010.

[22] K. Friston, "What is optimal about motor control?," *Neuron*, vol. 72, no. 3, pp. 488–498, 2011.

[23] T. D. Sanger, "Learning visually guided risk-aware reaching on a robot controlled by a gpu spiking neural network," in *International Conference on Neural Information Processing*, pp. 282–289, Springer, 2016.

[24] H. Asama, M. Sato, L. Bogoni, H. Kaetsu, A. Mitsumoto, and I. Endo, "Development of an omni-directional mobile robot with 3 dof decoupling drive mechanism," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 2, pp. 1925–1930, IEEE, 1995.

[25] M. Eich, F. Grimminger, and F. Kirchner, "A versatile stair-climbing robot for search and rescue applications," in *2008 IEEE International Workshop on Safety, Security and Rescue Robotics*, pp. 35–40, IEEE, 2008.

[26] P. Hornback, "The wheel versus track dilemma," *Armor Magazine*, vol. 26, pp. 33–34, 1998.

[27] B. Kersbergen, G. A. Lopes, T. J. van den Boom, B. De Schutter, and R. Babuška, "Optimal gait switching for legged locomotion," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2729–2734, IEEE, 2011.

Glossary

List of Symbols

A	gain of the angle
B	gain of the angular velocity
c(x,u)	cost function based on the state (x) and the motor command (u)
E(v)	expected cost of error
f	force
f_b	force to move backward
f_c	exerted force
f_f	force to move forward
f_m	force produced by motion
$\hat{f}_m$	estimated force produced by motion
I	inertia of the system
$\hat{I}$	estimated inertia of the system
L	weight needed for the neural network
$p(x)$	probability distribution of state (x)
$\dot{p}(x)$	change in the probability distribution of state (x)
$\hat{\dot{p}}(x)$	prediction about the change in the probability distribution of state (x)
R	resistance
sd	variance of the Gaussian distribution
u	motor command
$\hat{u}$	estimation of the motor command (u)
u_b	motor command to move backward
u_d	difference between u_b and u_f
u_f	motor command to move forward
u_r	random motor command with uniform distribution

v	disturbance affecting the sensory system
V^*	measured voltage
V_{cc}	maximum voltage
v(x)	value function of the states (x)
w	disturbance affecting the plant
x	state of the system
x_d	desired state of the system
x_r	random state of the system
x_v	angular velocity of the system
x^*	measured state of the system
$\hat{x}$	estimated state of the system
$\hat{x}^*$	estimation of the measured state (x^*)
$\dot{x}$	change in the state of the system (x)
$\tilde{x}$	prediction about the state of the system
γ	learning rate
λ_t^i	responsibility signal
$\hat{\lambda}_t^i$	estimate of the responsibility signal
μ	prior belief
ψ	function approximator
ρ	risk
σ	scaling constant
τ	movement at each time instant