Design of a setup for experimental research on stability of a bicycle-rider system subject to large perturbations

S. van de Velde

Design of a setup for experimental research on stability of a bicycle-rider system subject to large perturbations

by

S. van de Velde

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday March 29, 2022 at 11:00 AM.

Student number:	4577639	
Project duration:	February, 2021 – March, 2022	
Thesis committee:	Dr. ir. A. L. Schwab,	TU Delft, BmechE (chair)
	Dr. ir. R. Happee,	TU Delft, COR
	Dr. ir. J. P. Meijaard,	TU Delft, PME
	Ir. M. M. Reijne,	TU Delft, BmechE

TU Delft

Preface

Right at this moment, I am sitting at home on the couch, one day before I am going to hand in this report. My new home is a 180 kilometer drive away from Delft, where my setup has already been removed. The last two months, I have spent writing down the adventure that was my master thesis project. And what an adventure it has been...

This writing phase has not been my favourite stage, but it makes me even more grateful for the more exciting parts. Like the first time a real person, it was Marco I believe, was going to try out the setup, which was built to pull over whoever was cycling on it. Or of course the times I went for a ride myself.

One year ago, I wouldn't have been able to envision what it was all going to look like. I don't blame this former version of myself, considering all the different components that the setup consists of. By now, I have learned from a great man that even such big goals can be reached if we only divide and conquer.

I would like to thank everyone who participated in the pilot experiments. Especially Luuk made my day when he was last-minute willing to participate in my final pilot, after a part of the setup broke down during the intended final pilot. I would like to thank Heike for all her help with the motors and Jason for his thoughts on the data processing. We also couldn't have done without Judith, who explained us everything about the gait lab and made it possible for us to use it for so long. Another special thanks goes out to Thijs, who supported me with anything I needed when I was at the end of my rope. Last but not least, I would like to thank Arend and Marco for supervising me during this project. You have been an absolute pleasure to work with and you really showed me how fun research can be.

S. van de Velde
Leuven, March 2022

Abstract

Up to now, not much is known about how humans control bicycles, especially when subject to large perturbations. In order to learn more about the extent to which these perturbations can be handled, a new experimental setup is required, which can deliver large perturbations to cyclists and bring them to fall. This thesis describes the detailed design and evaluation of such a setup.

Several requirements are formulated regarding different experiment aspects that the setup must adhere to.

The proposed design consists of a treadmill on which a subject rides a bicycle. Ropes are guided from the ends of the bicycle's handlebar toward the front and back of the setup, where they are attached to four motor units of a robotic rope-pulling system. Based on a feed-forward conversion added with PI force feedback control, a shared controller commands the motors to maintain a tracking force or perturb the cyclist by applying a net torque on the handlebar for a short time. An active safety harness is the main feature that prevents the subject from harm. Meanwhile, motion capture recordings of strategically placed passive markers and data from inertial measurement units on the bicycle are collected. With appropriate processing, the angles and angular velocities that describe the dynamics and control of the bicycle-rider system can be obtained from these measurements. The force data is also collected, by the controller.

Interesting results that can be obtained with this setup include the probabilities to fall after perturbations of variable forces and the data required to evaluate the equations of motions of the bicycle-rider system during and shortly after a perturbation. This can be used to provide a baseline against which future bicycles can be compared and it is useful for validating rider models.

The experiment shows consistent performance and generates high quality measurements. Neither the 12 pilot participants nor the 26 subjects who participated in follow-up experiments encountered any safety issues. The experiment could even be improved by resolving issues regarding the motor behaviour and CPU overloads. The workload of the experiment operators could be decreased if intervention of the safety harness was trained on cycling and coupled to deactivation of the treadmill. A next step would be to automate the data processing by developing a classifier that distinguishes recoveries from falls.

Acronyms

DoFs	Degrees of freedom
EoMs	Equations of motion
GCP	Ground contact point
IMU	Inertial measurement unit
IQR	Interquartile range
MIMO	Multiple-input multiple-output
MoCap	Motion capture
SISO	Single-input single-output
STD	Standard deviation

Contents

1 Introduction

There are many reasons why cycling should be supported. For example, swapping the car for a bike can reduce the ever rising volume of traffic, lowering both emission and congestion issues. Another advantage of taking a bicycle trip can be the relief of stress and provision of exercise, which is a welcome change for people sitting behind a desk for most of the day. With the advent of the e-bike, efforts to cover long distances can be reduced and it is possible to compensate for lowered physical capabilities, being favourable to commuters and elderly people. Furthermore, many discovered cycling as a replacement for public transport and as a leisure activity during the COVID-19 pandemic, as it is a responsible way to spend time outside.

A drawback of cycling is that the rider is vulnerable to severe injury in traffic accidents. This shows in the road safety numbers of the Netherlands (SWOV, 2020a; SWOV, 2020b). While the number of fatal traffic accidents has decreased since 1973 for the most common modes of transport, that decrease has stagnated for cyclists since 2000. The number of serious injuries has even increased during the last decades, while it has gone down for other modes of transport. In 2019, 66% of all serious road injuries and 31% of all fatal road accidents concerned cyclists. Combined with the uptake in cycling, these numbers are expected to increase even further.

The causes for cycling accidents are often a combination of multiple factors. Some of the accidents are caused by collisions, but, as stated by Schepers and Klein Wolt (2012), “most cyclists admitted to hospitals in the Netherlands are single-bicycle crash victims.” In such crashes, the cyclist is the only road user involved. At the root of these accidents is the nature of the vehicle. The bicycle has only two contact points with the ground, lying in longitudinal direction. Therefore, it is unstable in lateral direction, implying the possibility to fall over. Although the vehicle is self-stable within a particular speed range (Meijaard et al., 2007), big enough external perturbations can still disrupt balance.

It is possible to enhance stability by ‘steering into the fall’ (Kooijman et al., 2009; Schwab and Meijaard, 2013). When the bicycle tends to lean towards one side, steering in that direction places the ground contact points back under the center of mass. This mechanism removes net moments that can lead to a fall. However, in order to ensure a safe situation on the road, the heading direction should also be considered.

Thus far, it is unknown what strategy cyclists use to control a bicycle in case of large perturbations and what are the limits of operation of the combined bicycle-rider system. A realistic model that captures this control behaviour could be of use to design safer bicycles, for example with an active steer assist that could help the rider to take the appropriate steering actions. Furthermore, once it is known what perturbations the system can and cannot cope with, it is clear what kind of perturbations should be eliminated from the cycling environment to maintain a safe situation.

Schwab and Meijaard (2013) report various theoretical rider control models that have been proposed over the years. The suggested control methods include *classical control*, based on the pilot models developed by McRuer, *optimal control*, which minimizes a cost function, *intermittent control*, that activates when a certain value is exceeded and *intuitive control*, which depends on the speed and is proportional with the feedback.

In order to confirm which model is most suited to mimic a real cyclist, empirical validation is of paramount importance. However, experimental data that can be used for this purpose is barely available. One of the very few existing datasets is gathered by Moore (2012), but only small perturbations were applied in this experiment. Large-perturbation studies with cyclists, leading to critical situations, are non-existent.

This asks for a new experimental setup, with the following requirements. The setup should allow for realistic cycling, be able to apply controlled perturbations of variable magnitudes to a subject who is cycling straight ahead at various forward speeds, with the capability to safely cause loss of stability, and collect data that captures the rider control behaviour.

Having such a setup makes it possible to validate which cyclist control model is most suited to mimic a real cyclist. As such, models can be used to predict which interventions can improve cycling safety and prevent falls. Additionally, it gives the opportunities to empirically compare performance using different bicycles, environments or cycling strategies. For example, it can be tested whether a bicycle designed to be safer actually leads to less falls in comparison with a baseline bicycle.

This thesis describes the detailed design of such a new experimental setup and evaluates the end product. With this, it follows up on an earlier literature study (included in the digital material) in which inspiration is drawn from gait perturbation experiments and the possibility to apply similar systems to cycling has been discussed.

The proposed setup consists of a treadmill on which a cyclist cycles, while a robotic rope pulling system can temporarily disturb the bicycle's handlebar by imposing a torque on it. Such point contact force perturbations by means of a pulling rope are effective to cause falls because they can generate large forces in appropriate directions considering the degrees of freedom (DoFs) and inertia of the different components. Meanwhile, they do not obstruct ordinary control behaviour. The relative motion environment in the form of a treadmill is chosen because this setting allows for more realistic motion with respect to a simulated environment and the restriction of space is convenient for safety reasons. Hence, the possibility for the cyclist to wear an active safety harness that freezes when a fall is detected. Additionally, the bicycle and surroundings are equipped with padding and the cyclist wears protective gear. Recordings of passive markers on the bicycle and cyclist by motion capture (MoCap) cameras and inertial measurement units (IMUs) are used to capture the dynamics and control of the bicycle-rider system during the experiment.

This thesis is structured as follows. First, the requirements that the experimental setup must adhere to are drawn up in Chapter 2. Chapter 3 then describes the different components of the experiment, after which the data processing is treated in Chapter 4. Subsequently, Chapter 5 reports the results. The discussion following in Chapter 6 evaluates the setup and Chapter 7 concludes with some final remarks and recommendations.

2 Requirements

As stated in the previous chapter, the objective of this thesis is to design an experimental setup that can safely apply controlled perturbations, with the capability to cause falls, to a subject who realistically cycles straight ahead at one of various possible prescribed speeds, while data is collected that captures the bicycle-rider system and the perturbations. This chapter draws up the specific requirements that can be validated to judge whether or not the eventual design is successful. The requirements are treated according to the part of the experiment concerned: *cycling*, *perturbations* and *data collection*.

2.1. Cycling

The first set of requirements considers everything that has to do with *cycling* during the experiment, in the broadest sense. The according requirements are listed below.

Realisticness
Cycling during the experiment should be realistic. This means that the bicycle dynamics should correspond to cycling in real life and that the rider should be able to give the bicycle all inputs that could be given during normal cycling. Normally, a cyclist can perform steering actions, upper body and knee movements and change the pedalling frequency. Furthermore, the available space for manoeuvring should correspond to that of a real traffic environment and the rider should not be prepared for a coming perturbation.

Effective steering
During cycling, a rider can perform stabilizing control actions by the mechanism of 'steering into the fall'. When the bicycle leans towards one side, the rider can place the support points back under the center of mass, by steering the front wheel in the direction of undesired lean (Schwab et al., 2013). During the experiment, this mechanism should remain executable. Reijne et al. (2021) mentions some fall mechanisms in which the rider can not steer into the fall, e.g. locking up of the steering assembly and loss of friction between the front tire and the ground surface. These should thus be avoided.

Speed
It should be possible to select the cycling speed such that experiments can be held below, in and above the bicycle's self-stable region. For the benchmark bicycle from Meijaard et al. (2007), the transition points of these regions lie around $4.3\,\mathrm{m\,s^{-1}}$ and $6.0\,\mathrm{m\,s^{-1}}$.

Safety
During the experiment, it is of paramount importance that no harm is brought to the participants. This also holds in the case of falls, whether or not as a result of perturbations.

2.2. Perturbations

Following on the cycling requirements, this set elaborates on requirements that the *perturbations* must adhere to.

Magnitude

The setup should be able to give perturbations of varying magnitude, but at least perturbations of the highest magnitude should cause the cyclist to fall. A fall is here defined as any case where the subject could not remain cycling without steering outside the designated area or putting a foot on the ground.

Responsiveness

For the subject to be unprepared for the perturbation, it is important that he does not notice the perturbation force building up to the eventual level. Therefore, the rise time of the force should be less then the time that a human needs to perceive it. The first coordinated action in response to a stimulus is the long latency reflex, which takes around 50-100 ms (Kurtzer, 2015). This is thus the rise time's order of magnitude that is aimed for.

Randomness

Again in order to prevent the subject from being prepared, more randomness in the perturbations is better. Therefore, the perturbation direction, magnitude and activation instant should be randomly determined.

Controllability

Having to do with the safety requirement from Section 2.1, the operator of the perturbation system should be in full control of the perturbations. This means that he should in the end decide whether or not to perturb at a certain moment and be in charge of the maximum forces that are delivered.

2.3. Data collection

The last set of requirements focuses on what *data* should be *collected* during the experiment.

Bicycle dynamics

During the experiment, data that describes the bicycle dynamics should be collected. Meijaard et al. (2007) describe benchmark equations of motion (EoMs) for the Whipple bicycle model (Whipple, 1899). They explain that the model's configuration space can be described by seven DoFs, in the paper parametrized as follows with accompanying symbols: the rear wheel ground contact point (GCP) location (x_P,y_P), the rear frame's yaw and lean rotations ψ and ϕ, the steering angle δ and the rotations of the rear and front wheels θ_R and θ_F. Of these coordinates, all but ϕ and δ are ignorable, meaning that they do not come forward in the EoMs. In the velocity space, only three velocity DoFs remain after taking into account the longitudinal and lateral no-slip conditions for each wheel. They are chosen as the derivates of the remaining DoFs in configuration space $\dot{\phi}$ and $\dot{\delta}$ and the rear wheel rotation rate $\dot{\theta}_R$, that dictates the forward speed. Furthermore, the EoMs require 25 parameters that characterize the bicycle design.

Extending the model with a rigid rider requires additional parameters that describe the human body (Moore et al., 2009). These could however be estimated from the participant's length and weight.

All in all, to capture the bicycle dynamics, it is required to identify the bicycle design parameters and the participants length and weight, and to measure the rear frame lean angle and rate, the steering angle and rate and the rear wheel rotation over time.

Rider control behaviour

As described earlier, the rider can apply steering, move his upper body and knees and change the pedalling frequency. Of these, the lean and steering actions correspond directly to the bicycle model's DoFs. Various rider control models only consider steer torque as output (Schwab et al., 2013), because it is shown that steering is the main contributor to lateral control and upper-body lean occurs very little (Moore et al., 2011, Kooijman et al., 2009). Measuring the steering torque would thus be desirable, but practically, this is hard to do. Instead, it can be deducted from the bicycle model's DoFs described above.

That lean actions appear not to be important, is convenient for using the Whipple bicycle model (Whipple, 1899), which is often used as a base for bicycle-rider models. This model assumes a rigid rider, which could thus be a valid assumption. To verify this rigid-rider assumption, upper body motion should be measured during the experiment.

Additionally, Kooijman et al. (2009) noted that the pedalling frequency has a substantial impact on the frequency of the steering actions. The cadence is thus measured as well.

Perturbation information

In order to discover the limits of operation of the bicycle-rider system, it is necessary to know at what perturbation magnitudes the cyclist is able to stay upright or falls. For this purpose, the desired and measured forces need to be collected. This data can then also be used to assess how well the measured forces correspond to the desired forces.

3

Experiment design

This chapter describes the design choices that were made for the experiment to satisfy the requirements established in the previous chapter. Each section describes an aspect of the experiment. Section 3.1 covers the experimental setup, starting with a general overview and followed by a more in-depth explanation of the different components. Section 3.2 then proceeds with the control architecture. After that, the data collection and experiment protocol are covered in Sections 3.3 and 3.4.

3.1. Experimental setup

Figure 3.1 shows a schematic illustration of the experimental setup. During the experiment, a subject rides a bicycle (A) on a treadmill (B), while wearing protective gear and being secured by a safety harness (I) attached to the ceiling. Around the treadmill, a profile construction provides support to gymnastic mats for additional safety, and to a perturbation system. The perturbation system makes use of four motors units (C1-C4), located around the treadmill (two on the front and two on the back). These motors units, which are operated by a single controller (F), can wind up ropes (D) that are connected to the ends of the bicycle's handlebar. When the perturbation system operator determines that a perturbation can be given, either a clockwise or counter-clockwise torque is induced on the handlebar, by simultaneous activation of motors 2 and 4, or motors 1 and 3, respectively. The four motor units are powered by two power supplies (G). Between the power supplies and the motor units, an emergency button (H) is installed to disconnect all motors from power. The forces in the ropes are measured by force sensors (E) and fed back to the controller. More details on the setup's main components are given below.

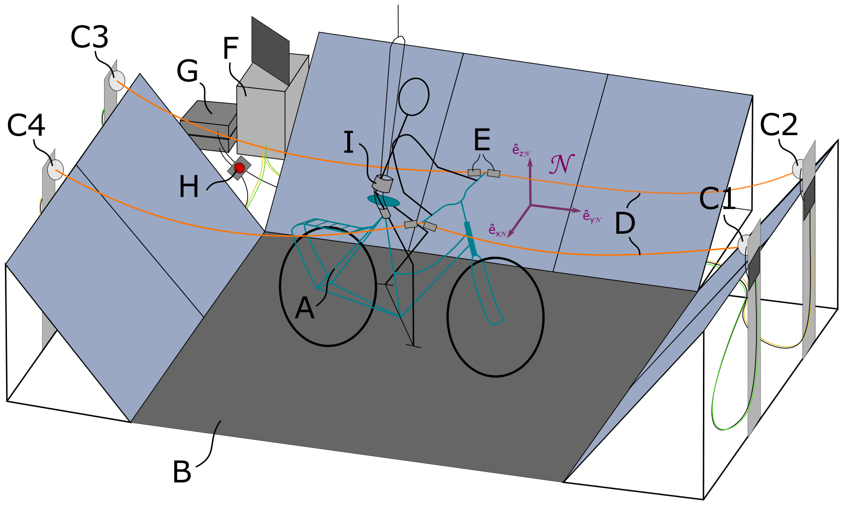


Figure 3.1: Experiment setup, consisting of (A) a bicycle, (B) a treadmill, (C1-C4) a perturbation system with four motor units, (D) ropes, (E) force sensors, (F) a controller, (G) power supplies, (H) an emergency button and (I) a safety harness.

(A) Bicycle

Figure 3.2 shows a picture of the bicycle used in the experiment. Because the goal is to find out more about an ordinary bicycle-rider system, an ordinary bicycle is the starting point of this setup. The bicycle is chosen such that its geometry matches that of the TU Delft steer assist bicycle (Dialynas et al., 2018), to compare performances in the future.

The perturbations are delivered by means of pulling a rope, that is connected to an extension of the bicycle's handlebar. The motivation for this choice of perturbations is discussed in part (C).

For the sake of safety, additional precautions are taken. The bicycle is for instance equipped with padding and because breaking on a treadmill can be dangerous, one of the two breaks is deinstalled and the other one is replaced, such that it is not easy to reach. A safety belt attached to the saddle fixes the bicycle to the rider, such that the bicycle does not sweep backwards under the rider when he hangs in the safety harness.

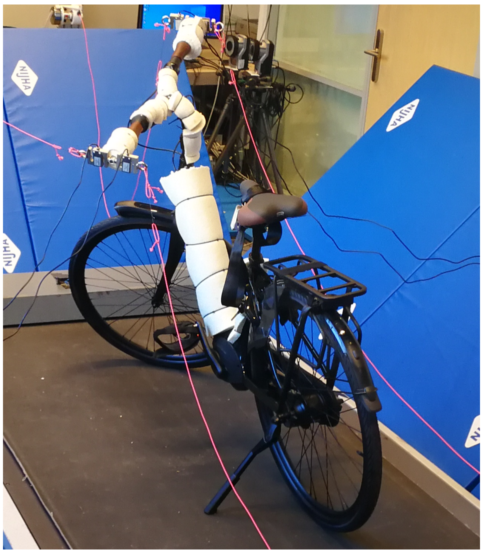


Figure 3.2: Bicycle with safety precautions.

(B) Treadmill

Cycling on a treadmill is chosen as a compromise between realistic cycling and a safe environment. It is supposed to be representative of cycling in the exterior because the relative motion between the wheels and the ground surface is realistic. Since both wheels are in contact with the same surface, the coupling between their speeds remains unaffected and the belt width allows some room for manoeuvre. This space is limited, but so is the space one has on a cycling lane. Regarding safety, the small absolute displacement makes it possible to secure the subject to the ceiling.

The treadmill that is used has a belt size of 1.2 m wide by 2.62 m long and can move with speeds from 0 to 18 km h^{-1}, with steps of 0.1 km h^{-1}.

(C) Perturbation system

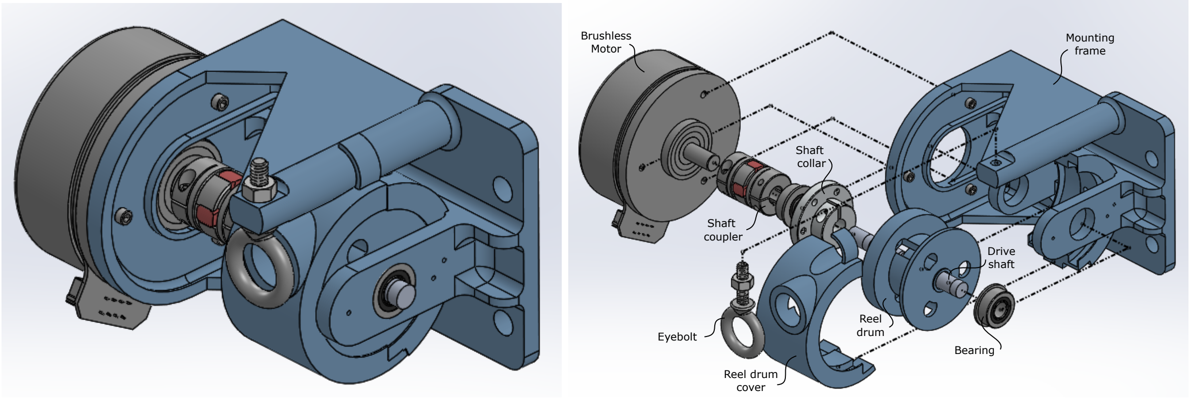
It is chosen to perturb the bicycle-rider system by inducing a torque on the steering assembly by means of a motor-driven rope-pulling system. Such perturbations are effective to cause falls, because they are applied in direction of steering. Together with the bicycle lean, this is one of the two DoFs directly related to balance that remain in velocity space and the lower inertia of the front frame with respect to the rear frame makes it easier to apply perturbations that suffice the magnitude requirement. The perturbation system itself does not impose a lot of additional weight on the handlebars, which is considered beneficial to keep the bicycle's handling qualities realistic.

Using motors to wind up the rope allows controllability in terms of magnitude and duration of the pull. The perturbations are only delivered for a short time, to represent and impulsive force. For the system to exert clockwise and counter-clockwise torques on the handlebar without leading to net forces, four ropes and motors are required.

The perturbation system is based on the open source Bump'em system developed by Tan et al. (2020). Their paper describes multiple modular configurations. Appendix A explains which of those was used in this experiment and what changes were applied in this setting with respect to the original. The authors report a maximum pull force of 200 N and rise times below 45 ms for a step response with this configuration. Based on a previously conducted pilot study, such pull forces are estimated sufficient to comply with the magnitude requirement and the rise time suffices the responsiveness requirement, as drawn up in Chapter 2.

The perturbation system consists of four motor units which are each controlled by a control unit. Figure 3.3 depicts one of the motor units, both in assembled condition (a) and as exploded view with all parts labelled (b). The blue parts are 3D-printed and the grey parts are essentially off-the-shelf components. A maxon brushless motor (EC-90 Flat, maxon Group, Switzerland), attached to a mounting frame, is connected to a drive shaft by a shaft coupler. The drive shaft is in turn connected by a shaft collar to a reel drum and held in place by ball bearings. On the reel drum, a rope can be wound, which is prevented from coming off by a reel drum cover. The loose end of the rope is guided through a hole in the reel drum cover and through an eyebolt towards the bicycle's handlebar. The mounting frame can be bolted on to the profile structure around the treadmill.

The control units each consist of a motor driver (ESCON 70/10, maxon Group, Switzerland), a shunt regulator (DRS 70/30, maxon Group, Switzerland) and are connected to a power supply. A shared controller prescribes the desired motor current for each unit, upon which the motor driver draws the required power and regulates the motor actions. The shunt regulator, that is connected in series with the power supply, protects the system from counter-electromotive force.



(a) Assembled

(b) Exploded view

Figure 3.3: Motor unit (a) in assembled condition and (b) as exploded view (figure inspired by Figure 1B from Tan et al. (2020)).

(D) Rope

The perturbations are delivered by a rope rather than by a rod or similar structure, because a rope imposes less restrictions on the freedom of movement. The rope that is used is high-strength, high-stiffness, ultra high molecular weight polyethylene rope, type Dyneema SK75 with a 2 mm diameter, similar to the rope shown in Figure 3.4.

The ropes are not directly tied to the handlebar's extensions, but in series with a breakaway cable, that is designed to break in case of too high forces as a safety precaution. For this breakaway cable, fishing line with a specified break strength is used.

Figure 3.4: Dyneema SK75 2mm rope (from Amazon (n.d.)).

(E) Force sensors

Scaime S-type load cells (ZFA, scaime, France), shown in Figure 3.5, measure the forces in the ropes. Due to their weight, the load cells can not hang halfway the rope without affecting the tension. Therefore, they are bolted on to the handlebar's extensions. A drawback of this construction is that the load cells are not always purely loaded in axial direction. It would be favourable to have light-weight force sensors, as described in Tan et al. (2020), but their manufacture did not succeed in the available time.

The force sensor signals each pass through an amplifier (CPJ, scaime, France) before being fed as input to the controller.

Figure 3.5: S-type load cell from scaime datasheet.

(F) Controller

A shared controller processes inputs from the perturbation system operator and directs the motors. It also reads out the measurements from the force sensors and uses this data to adjust the motor signals via PI-control. Furthermore, it controls a screen on which the force measurements, pull time recommendations and activation information are shown in real time to the operator and sends out a synchronization signal to a Qualisys data collection system. Section 3.2 touches more on the control architecture.

In order to process the force sensor measurements fast enough and thus maintain the short rise-time, the control loop must run at a high enough frequency. The controller that Tan et al. (2020) use runs at 1000 Hz, which is thus also taken as desired rate here. This requires a controller with sufficient CPU. Because the complete application is quite complicated, it is considered safest to opt for a real-time machine. A Speedgoat real-time machine (Performance real-time target machine, item ID 109000, Speedgoat, Switzerland) is thus used during the experiment, along with a compatible input/output module (IO133, item ID 2A133, Speedgoat, Switzerland).

The IO133 is an analog digital input/output module, containing 16 channels for differential analog inputs, 8 channels for analog outputs and 14 channels that can be used as either digital input or digital output. The voltage range of the analog input channels can be selected via software to be either ±5 V or ±10 V and it can reach sampling rates of up to 2000 Hz. The analog output channels voltage range can be selected to be 0-5 V, 0-10 V, 0-10.8 V, ±5 V, ±10 V or ±10.8 V, with a settling time of 10 μs. The current setup uses four analog inputs, ranged ±10 V, 8 digital outputs and 5 analog outputs, with ranges of ±10 V.

(G) Power supplies & (H) emergency button

Two power supplies provide the motors with power for operation. The motors on the front of the treadmill share a single power supply and so do the motors on the back. This division is chosen because the two pairs of motors will never pull simultaneously, and thus not draw maximum power at the same moment.

In case of an emergency, the motors can be disconnected from power by an emergency button. This button is installed between the power supplies and the motors. At this location, it is possible to shut down all motors with one button, because all wires are here together and it omits delays in the power supplies.

(I) Safety harness

During the experiment, the subject wears a safety harness connected to a RYSEN (RYSEN, Motek Medical, the Netherlands). The RYSEN is a low-power 3D overground bodyweight support system (Plooij et al., 2018). It allows the user freedom of movement in both longitudinal, lateral and vertical direction, while providing a support of 10-60% of the total bodyweight, according to the settings of the operator. An automatic fall detection algorithm is supposed to freeze the cables when a fall is detected, such that the user is caught and will not reach the ground. In addition to this automatic response, the system can also be frozen by the Rysen operator. Figure 3.6 shows a picture of a participant wearing the Rysen while sitting on the bicycle.

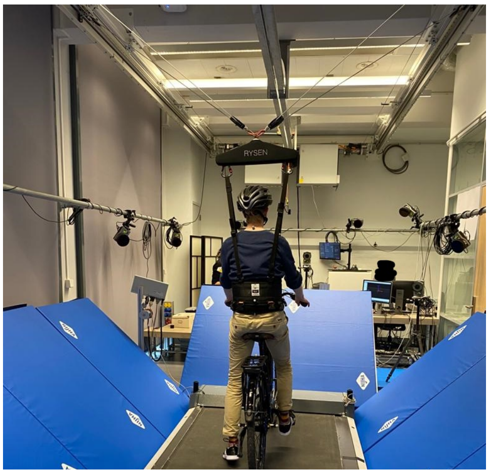


Figure 3.6: Cyclist wearing Rysen safety harness.

3.2. Control

The real-time target machine runs an application built in Simulink Real-Time. The perturbation system operator controls the application via a simple user interface with three buttons: *start*, *click to pull* and *stop*. This interface is shown in Figure 3.7. The stop button is made significantly larger than the other buttons to allow for quick responses when necessary, following Fitts' law. The programming behind this interface can be found in Appendix B.

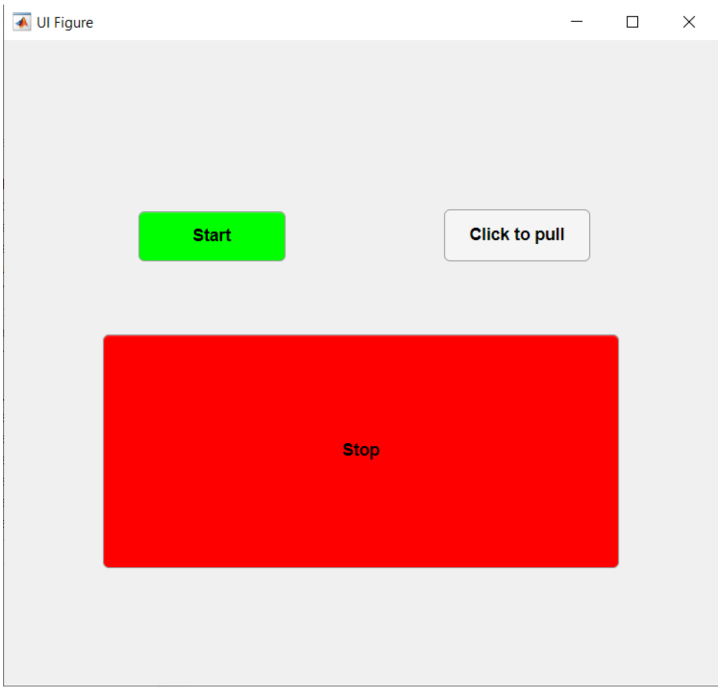


Figure 3.7: User interface through which the operator can start and stop the Simulink application and deliver pulls.

Figure 3.8 shows a simplified representation of the control architecture, explaining the general working principle. For a better overview, only two out of four motor units are visualised in this scheme. Motor units 3 and 4 are after all similar to units 1 and 2 respectively.

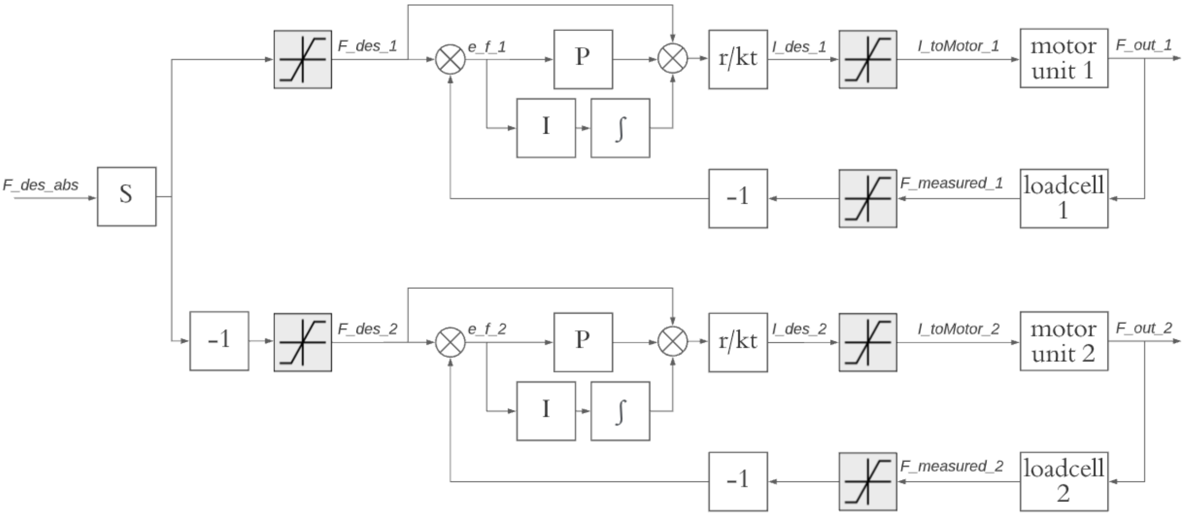


Figure 3.8: Main control architecture for two motors of opposite pulling direction.

The scheme is explained from left to right. The input *F_des_abs* determines the desired pull force. It has a value of zero, unless the *click to pull* button is clicked. From that moment onward, it changes to a random value between a pre-installed minimum and maximum for a predefined pull duration of 0.3 s. At this point, it is yet undetermined whether the perturbation shall be clockwise or counter-clockwise. This is taken care of through the multiplication with gain S, which contains a value of 1 or -1, randomly changing upon each click on the *click to pull* button. After this multiplication, the signal splits into two signals: one for each motor unit. The lower of the two is multiplied by -1, such that the split signals are of opposite sign. Both signals then pass a saturation that cuts off values below 5. This facilitates a tracking force of 5 N when the input is zero or negative and passes on higher values when a perturbation in the according direction is desired. Now the desired forces for each motor unit is determined and from here on, the control schemes are similar.

The desired force is added with a PI controller contribution. The resulting force is multiplied by the reel drum radius and divided by an experimental motor torque constant to convert the force to a motor current. The exact values of the feedback gains and the conversion factors are addressed in the subsection below. Again a saturation is applied, to guarantee that the current will not damage the motor and then the current is sent via the motor controller as input to the motor. This results in the attached rope pulling on the handlebar.

The loadcell placed in series with the rope measures the force that is applied. Another saturation cuts off measured forces below 0 N, as they can not physically be the result of a pulling rope. The signal is then multiplied with -1 such that it is subtracted from the desired force at the addition.

The complete Simulink application that employs this scheme is visualized in Appendix B. This also contains additional features such as

- a synchronization signal that enables communication with the data acquisition system. This indicates the activation and deactivation instants of the perturbation system and all pulls,
- a signal that counts down from a random value to advise the instant of the next perturbations

  and
- the monitoring and/or logging of data of interest (e.g. the force signals) to a screen/the Speedgoat memory.

3.2.1. Feedback gains & conversion factors

This subsection touches on the determination of the conversion factors and feedback gains used in the controller. The reel drum radius is based on the physical property and the motor torque constant was adjusted from the motor specifications such that the open loop response were adequate when performed on a static weight. The P and I gains were tuned by trial and error such that fast rise times were obtained and the tracking performance was increased. The eventual values can be found in Table 3.1. The motor constants are here reported as values divided by 3. This is because it was found out during the writing down of the results that the inputs to the motor controllers were supposed to be voltages instead of currents, which were then internally multiplied by 3 to convert the voltage back to current. This was in practice compensated by the too high values for the motor torque constants. The motor torque constants divided by 3 are thus the values that lead to the current input that was actually given to the motors.

		Motor Unit 1	Motor Unit 2	Motor Unit 3	Motor Unit 4
r	(m)	19.05×10^{-3}	19.05×10^{-3}	19.05×10^{-3}	19.05×10^{-3}
Kt	(Nm/A)	0.4620 / 3	0.5313 / 3	0.5313 / 3	0.5313 / 3
P	(-)	1.5	2	1.75	1.5
I	(-)	2	2	2	2

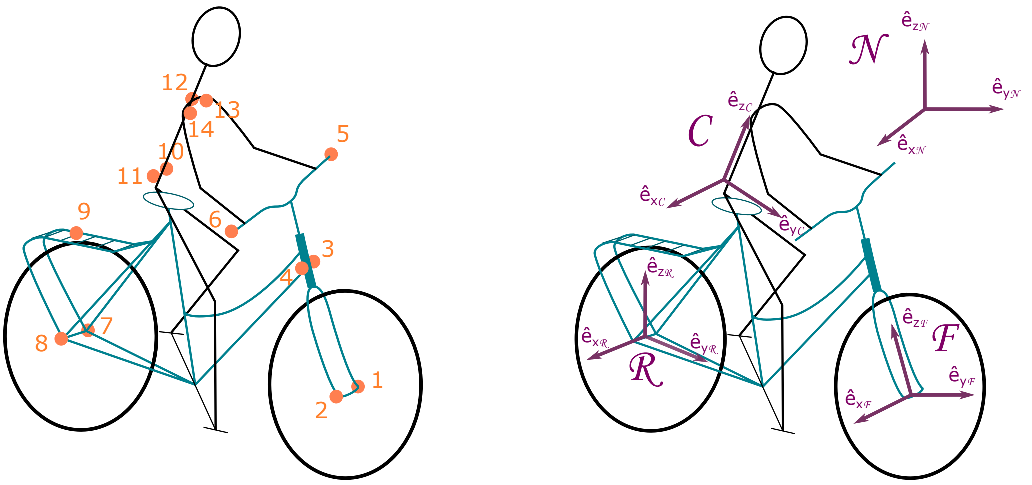
Table 3.1: Values for feedback gains and force-current conversion factors for all motor units.

3.3. Data collection

From the data collection requirements in the previous chapter, it became evident which angular positions and velocities need to be measured, together with the desired and measured forces. Each of these kinds of data requires its specific approach, which will be explained below.

3.3.1. Angular positions

Reviewing the requirements, the lean angle, steering angle and upper body angles should be collected. For this purpose, a Qualisys optoelectronic measurement system, containing 12 MoCap cameras, is used to record the positions of passive markers on the bicycle and cyclist at 100 Hz. Two additional cameras make ordinary video recordings. The markers are placed such that the local coordinate systems of the bicycle's front frame F and rear frame R and that of the cyclist C can be obtained with respect to an inertial reference frame N. This N frame is established by initial calibration of the Qualisys system before starting the actual recordings. Its axes follow from markers on an L-shaped frame, which is placed against the left side of the treadmill before calibration such that the y-axis points in forward and the x-axis in sideways direction. The location of the origin is not exactly known. The marker locations used in the experiment are shown and described in Figure 3.9 (a) and Table 3.2. Figure 3.9 (b) shows the coordinate system orientations.



(a) Markers placed on bicycle and rider.

(b) Coordinate systems $\mathcal{R}$, $\mathcal{F}$, $\mathcal{C}$ and $\mathcal{N}$.

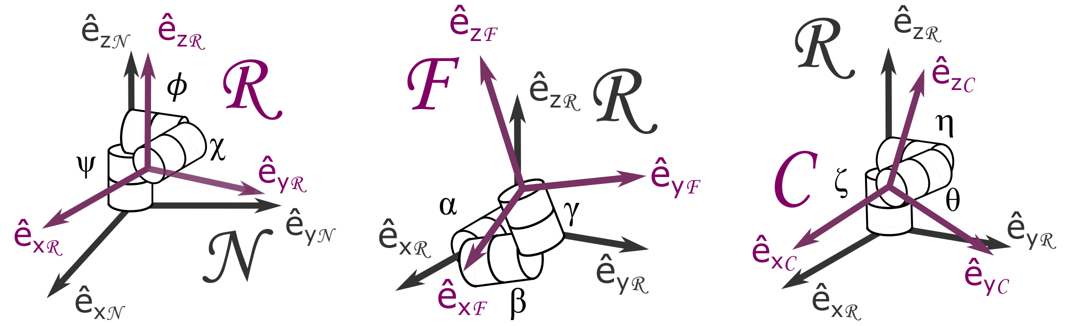
Figure 3.9: Visualization of bicycle and rider with (a) the marker positions and (b) the different frames.

Marker	Position
1	front wheel axis left
2	front wheel axis right
3	headtube left
4	headtube right
5	handlebar left
6	handlebar right
7	rear wheel axis left
8	rear wheel axis right
9	bike rack
10	lower back left (on top of safety harness)
11	lower back right (on top of safety harness)
12	seventh cervical (C7) vertebra
13	sternal end of left clavicle
14	sternal end of right clavicle

Table 3.2: Descriptions of the locations at which markers are placed during the experiment.

The local frame orientations can be described with respect to a reference frame by a succession of three rotations. This is visualized in Figure 3.10 by cans-in-series, which rotate with respect to each other around their primary axis. Figure 3.10 (a) shows how frame R is oriented with respect to frame N, by first a rotation ψ around the z-axis, followed by a rotation ϕ around the rotated y-axis and then a rotation χ around the rotated x-axis. This series of rotations will be referred to as *yaw-lean-pitch*. In (b) and (c), the rotations of respectively coordinate systems F and C are given relative to frame R. The latter succeed in the same order as the rotations seen in (a), here denoted as ζ, η and θ. For the rotations of frame F with respect to frame R, another succession order is chosen to comply with the structure of the bicycle. This frame orientation is described by first a rotation α around the x-axis, then a rotation β around the rotated y-axis and lastly a rotation γ around the rotated z-axis. These frame rotations are named here *tilt-bank-steer*. Positive rotation direction is determined by the right-hand rule.

Rotations ϕ, γ, ζ, η and θ respectively correspond to the desired lean angle, steering angle and upper body angles. Note that the symbols differ from those in Meijaard et al. (2007). With the processing explained in Section 4.1, these angles can be calculated from the marker positions. In addition to these angles, the rear frame yaw angle ψ is also of interest. In Section 2.3, this was designated as an ignorable coordinate for the bicycle configuration, but together with the GCPs of the wheels, it does reveal the bicycle's configuration on the treadmill. Other angles that are also obtained but not of direct interest can be used for sanity checks.



(a) R w.r.t. N (b) F w.r.t. R (c) C w.r.t. R

Figure 3.10: Frame rotations of body-fixed frames represented as cans-in-series.

3.3.2. Angular velocities

Besides the positions, measurements of the lean rate, steering rate, rear wheel rotation rate and the cadence are also required. To obtain those, Delsys Trigno avanti sensors are used. They contain an IMU, which measures angular rates in three axis directions. The data is sent out at 2000 Hz and automatically synchronized with the MoCap data.

Four such sensors are placed on the bicycle. One is on the bike rack, to measure the rear frame's velocity. It is placed such that the sensor's z-axis is aligned with the $\hat{e}_{zR}$ vector from Figure 3.10 (a) and its x- and y-axes are aligned with $-\hat{e}_{xR}$ and $-\hat{e}_{yR}$ respectively. This means that the gyroscope's y-data represent the negative lean rate $-\dot{\phi}$. Another sensor is attached to the rear wheel axis. Because the wheel rotates, it can't be constantly aligned with frame R, but the sensor's y-axis is aligned with $\hat{e}_{xR}$, to provide the rear wheel rotation rate. Similarly, another sensor, attached to the left crank arm, gives the cadence. These rotation rates will be denoted by $\dot{\chi}_R$ and $\dot{\chi}_C$ respectively. Finally, a last sensor is placed on the bicycle's front fender. Its z-axis is approximately aligned with $\hat{e}_{zF}$, and its x- and y-axes with $-\hat{e}_{xF}$ and $-\hat{e}_{yF}$. This gyroscope's z-data should thus give the steering rate $\dot{\gamma}$.

3.3.3. Force data

The desired and measured force signals are already available during the experiment in the controller. From here, they are saved on the target machine and copied to the host computer. As mentioned in Section 3.2, the controller sends out a synchronization signal to the Qualisys software for synchronization with the other data. This analog signal comes in to the data acquisition system at 100 Hz with a reach of ±10 V. The signal is designed such that activation, pulls and deactivation of the perturbation system are recognizable events.

3.4. Experimental protocol

The procedure of this experiment, with the purpose to validate the working of the experimental setup, consists of four phases: preparation, familiarization, exploration and randomization. Each of these phases are explained below.

Preparation

The first phase of the protocol is the preparation. Before arrival of the participant, the Rysen software is started and the Qualisys system is calibrated. The bicycle is provided with markers and sensors. When the participant arrives, their clothing is inspected on visibility of the bony landmarks around the neck and any reflecting elements are covered with tape. They are offered a helmet, gloves and shin protectors and helped into the safety harness. Their height and weight is given into the Rysen software. The remaining markers are placed on the participant and the bicycle is set to their size.

Familiarization phase

The next phase of the protocol aims at comforting the participant and building confidence in the system. It starts with mounting the bicycle without riding it. The participant is instructed to fall to one side, upon which they are caught by the safety harness. This is repeated until the participant feels comfortable with falling. Then, cycling on the treadmill without perturbations is practised. Although all participants know how to ride a bicycle, starting to cycle on a treadmill can be challenging. Once this goes well, the participants are asked to start a self-intended fall while cycling. They can choose to skip this if falling already happened during the learning phase.

Exploration phase

In the following phase, the critical pull force is explored. This is the desired perturbation force (in tens of Newtons) at which the participant is not able to recover, while he is instructed to ride as relaxed as possible. This exploration consists of a rough search and a finer search. The full procedure consists of the steps described below:

1. Set the desired force to 20 N and start the perturbation system.
2. Bring the treadmill belt to a speed of 12 km h^{-1}.
3. Induce a pull. If the participant falls, proceed to step 5. Otherwise, repeat this step until five pulls are given.
4. Increase the desired force by 30 N and go back to step 2.
5. Decrease the desired force by 20 N.
6. Bring the treadmill belt to a speed of 12 km h^{-1}.
7. Induce five pulls. If the participant recovers at least once, increase the desired force by 10 N and go back to step 6.
8. Denote the current desired force as Fmax.

Randomized phase

During the last phase of the experiment, the participant receives 10 pulls with a random magnitude. This desired force is chosen randomly between a minimum of 20 N and a maximum of Fmax+20 N.

4

Data processing

During the experiment, various kinds of data are collected to obtain the relevant angular positions and velocities of the bicycle and rider and the measured force profiles. This chapter covers their processing. The used notations correspond to Vallery and Schwab (2018), except that vectors are noted as $\vec{r}$.

4.1. Rotation matrices

The angular positions can be obtained from the rotation matrices that describe the rotations of the local coordinate systems with respect to a reference frame. These matrices, whose columns are the according coordinate system's unit vectors, can be found based on the marker positions recorded by the MoCap cameras. For this purpose, the set of markers described in Table 3.2 and illustrated in Figure 3.9 (a), is first extended with some virtual markers. The new marker set is described in Table 4.1. The next step is to find the initial rotation matrices for all three local coordinate systems. How this is done, is described in Section 4.1.1. After that, the rotation matrices for each timestep can be calculated using a singular value decomposition method (Challis, 1994). This will be explained further in Section 4.1.2. Once these rotation matrices are known, they can easily be converted to the according rotation angles by means of the `rotm2eul` MATLAB command. Another purpose that the rotation matrices can be used for, is to find the GCPs. This is elaborated on in Section 4.1.3.

Marker	Position
1	front wheel axis left
2	front wheel axis right
3	headtube left
4	headtube right
5	handlebar left
6	handlebar right
7	rear wheel axis left
8	rear wheel axis right
9	bike rack
10	lower back left (on top of safety harness)
11	lower back right (on top of safety harness)
12	seventh cervical (C7) vertebra
13	sternal end of left clavicle
14	sternal end of right clavicle
Virtual Marker	Position
21	average of markers 1 and 2
22	average of markers 3 and 4
24	average of markers 7 and 8

Table 4.1: Marker-position descriptions extended with virtual markers.

4.1.1. Initial rotation matrices

Each of the three coordinate systems require an individual approach to find their initial rotation matrix. These approaches will be explained below. For completeness, they are also given an origin.

Rear Frame

The rear frame coordinate system's origin is chosen to be located at marker 24, such that

$$ {}^{N}\vec{r}_{O_R} = {}^{N}\vec{r}_{M24}. \tag{4.1}$$

Vector ${}^{N}\hat{e}_{xR,t_1}$ is easily obtained from the positions of marker 7 and 8 at this instance:

$$ {}^{N}\hat{e}_{xR,t_1} = \frac{{}^{N}\vec{r}_{M8/M7,t_1}}{|\,{}^{N}\vec{r}_{M8/M7,t_1}\,|}. \tag{4.2}$$

For vector ${}^{N}\hat{e}_{zR,t_1}$, use is made of the unit vector pointing from marker 24 to 21:

$$ {}^{N}\hat{r}_{M21/M24,t_1} = \frac{{}^{N}\vec{r}_{M21/M24,t_1}}{|\,{}^{N}\vec{r}_{M21/M24,t_1}\,|}. \tag{4.3}$$

The participant is instructed to hold the steer straight at the beginning of the recording, such that marker 21, 7 and 8 span the local xy-plane. Then,

$$ {}^{N}\hat{e}_{zR,t_1} = {}^{N}\hat{e}_{xR,t_1} \times {}^{N}\hat{r}_{M21/M24,t_1}. \tag{4.4}$$

Another cross product between vectors ${}^{N}\hat{e}_{zR,t_1}$ and ${}^{N}\hat{e}_{xR,t_1}$ then gives ${}^{N}\hat{e}_{yR,t_1}$:

$$ {}^{N}\hat{e}_{yR,t_1} = {}^{N}\hat{e}_{zR,t_1} \times {}^{N}\hat{e}_{xR,t_1}. \tag{4.5}$$

Now that all unit vectors of the local coordinate are known, the initial rotation matrix can be constructed as follows

$$ {}^{N}R_{R,t_1} = \begin{bmatrix} {}^{N}\hat{e}_{xR,t_1} & {}^{N}\hat{e}_{yR,t_1} & {}^{N}\hat{e}_{zR,t_1} \end{bmatrix}. \tag{4.6}$$

Front Frame

Similarly to the rear frame, the front frame coordinate system's origin is located at marker 21

$$ {}^{N}\vec{r}_{O_F} = {}^{N}\vec{r}_{M21}. \tag{4.7}$$

Unlike in the previous case, it is difficult to construct all unit vectors of this coordinate system from the marker positions. The ${}^{N}\hat{e}_{xF,t_1}$ vector could be calculated from markers 1 and 2, or from markers 5 and 6, but there is no third marker that spans the local xy- or xz-plane, such that the z-axis is aligned with the steering axis. Instead, one can make use of the fact that

$$ {}^{N}R_{F} = {}^{N}R_{R}\,{}^{R}R_{F}. \tag{4.8}$$

As ${}^{N}R_{R,t_1}$ is already known, only ${}^{R}R_{F,t_1}$ needs to be found to obtain ${}^{N}R_{F,t_1}$. Following the rotations from the cans in series illustrated in Figure 3.10 (b),

$$ {}^{R}R_{F} = R_{\alpha}R_{\beta}R_{\gamma}. \tag{4.9}$$

Because the construction of the bicycle inhibits a bank rotation β, R_{β} is assumed to be equal to the identity matrix. That leaves

$$ {}^{R}R_{F} = R_{\alpha}R_{\gamma}, \tag{4.10}$$

in which

$$R_\alpha = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\ \alpha & -sin\ \alpha \\ 0 & sin\ \alpha & cos\ \alpha \end{bmatrix} \tag{4.11}$$

and

$$R_\gamma = \begin{bmatrix} cos\ \gamma & -sin\ \gamma & 0 \\ sin\ \gamma & cos\ \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.12}$$

The parameter α in Equation 4.11 is the steer axis tilt, a constant inherent to the bicycle design, and the steering angle γ in Equation 4.12 can be obtained from the marker positions. How this is done, is explained with help of Figure 4.1. Here, the handlebar makes a steering angle γ about the steering axis. The orientation of the front frame after rotation is depicted in orange, while the original position of the handlebar is dashed blue. This rotation is in a plane parallel to the local xy-plane.

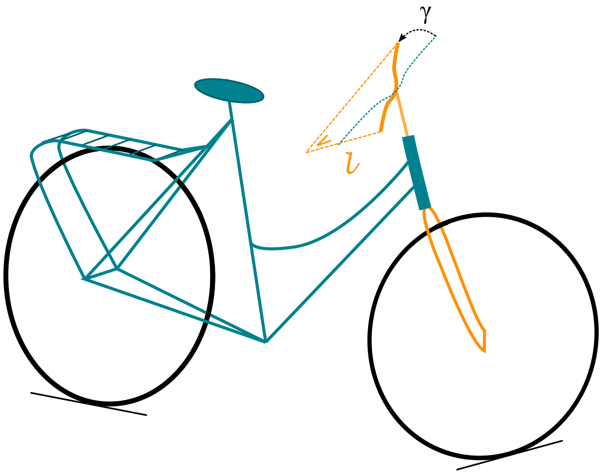


Figure 4.1: Visualization of the front frame making a steering angle γ.

To eventually find the value of γ_{t_1}, l_{t_1} is calculated first. This is done by projecting the vector from marker 5 to 6 on ${}^N\hat{e}_{yR,t_1}$ and correcting for the steer axis tilt α as follows:

$$l_{t_1} = \frac{{}^N\hat{e}_{yR,t_1} \cdot {}^N\vec{r}_{M6/M5,t_1}}{cos\ \alpha}. \tag{4.13}$$

Subsequently, γ_{t_1} can be calculated by

$$\gamma_{t_1} = arcsin(\frac{l, t_1}{|\ {}^N\vec{r}_{M6/M5,t_1}\ |}). \tag{4.14}$$

This sums up all the ingredients required to calculate ${}^R R_{F,t_1}$.

Cyclist

For the cyclist coordinate system, the origin is chosen to lie at marker 12

$${}^N\vec{r}_{O_C} = {}^N\vec{r}_{M12}. \tag{4.15}$$

Similar to the rear frame's ${}^N\hat{e}_{xR,t_1}$, ${}^N\hat{e}_{xC,t_1}$ can be calculated straightforwardly from markers 10 and 11, by

$${}^N\hat{e}_{xC,t_1} = \frac{{}^N\vec{r}_{M11/M10,t_1}}{|\ {}^N\vec{r}_{M10/M11,t_1}\ |}. \tag{4.16}$$

It should be noted that the safety harness, on which these markers are positioned, can move a little with respect to the human body. To guarantee an orthogonal coordinate system despite these small displacements, ${}^{N}\hat{e}_{zC,t_1}$ is calculated as the normalized distance from the projection of marker 12 on ${}^{N}\hat{e}_{xC,t_1}$ to marker 12. This is calculated as

$$ {}^{N}\vec{r}_{proj_{\hat{e}_{xC,t_1}}(M12,t_1)} = {}^{N}\hat{e}_{xC,t_1} \cdot \left({}^{N}\vec{r}_{M12,t_1} - {}^{N}\vec{r}_{M10,t_1}\right) {}^{N}\hat{e}_{xC,t_1} + {}^{N}\vec{r}_{M10,t_1}, \tag{4.17} $$

$$ {}^{N}\hat{e}_{zC,t_1} = \frac{{}^{N}\vec{r}_{M12,t_1} - {}^{N}\vec{r}_{proj_{\hat{e}_{xC,t_1}}(M12,t_1)}}{|\, {}^{N}\vec{r}_{M12,t_1} - {}^{N}\vec{r}_{proj_{\hat{e}_{xC,t_1}}(M12,t_1)} \,|}. \tag{4.18} $$

Again, the last unit vector can be found by means of the cross product between the previous two:

$$ {}^{N}\hat{e}_{yC,t_1} = {}^{N}\hat{e}_{zC,t_1} \times {}^{N}\hat{e}_{xC,t_1}. \tag{4.19} $$

Equivalently to Equation 4.6,

$$ {}^{N}R_{C,t_1} = \begin{bmatrix} {}^{N}\hat{e}_{xC,t_1} & {}^{N}\hat{e}_{yC,t_1} & {}^{N}\hat{e}_{zC,t_1} \end{bmatrix}. \tag{4.20} $$

4.1.2. Rotation matrices over time

To find how the rotation matrices evolve over time, a technique described in Challis (1994) is used. With this technique, one can find a rotation matrix ${}^{B}R_{A}$, such that

$$ {}^{B}\vec{r}_{i} = {}^{B}R_{A}\, {}^{A}\vec{r}_{i} + {}^{B}\vec{r}_{O_A}, \tag{4.21} $$

where

- ${}^{B}\vec{r}_{i}$ is the position vector of point i measured in frame B,
- ${}^{A}\vec{r}_{i}$ is the position vector of point i measured in frame A,

  and
- ${}^{B}\vec{r}_{O_A}$ is the position vector of the origin of frame A, expressed in frame B.

This requires a matrix C, calculated according to

$$ C = \frac{1}{n} \sum_{i=1}^{n} \left({}^{B}\vec{r}_{i} - {}^{B}\bar{r}\right)\left({}^{A}\vec{r}_{i} - {}^{A}\bar{r}\right), \tag{4.22} $$

with ${}^{B}\bar{r}$ the average of the position vectors of points 1 to n, expressed in the B frame (and ${}^{A}\bar{r}_{i}$ similar, but expressed in the A frame). Subsequently, the singular value decomposition of this matrix is computed,

$$ C = UWV^{T}. \tag{4.23} $$

${}^{B}R_{A}$ is then given by

$$ {}^{B}R_{A} = V^{T}U. \tag{4.24} $$

For the proof of this method, the reader is referred to the original paper (Challis, 1994). The benefit of this method over the method used for the initial rotation matrices, is that it can cope with some markers being temporarily occluded, whereas the other method relies strongly on the visibility of specific markers.

To apply this method to this situation, it is first necessary to define sets of markers that correspond to each frame. This is specified in the Table 4.2

Frame	Markerset
R	[3, 4, 7, 8, 9]
F	[1, 2, 22, 5, 6]
C	[10, 11, 12, 13, 14]

Table 4.2: Sets of markers associated with each coordinate system.

For each marker in a set, the local position vector can be found using the previously established initial rotation matrix.

$$^{X}\vec{r}_i = (^{N}R_{X,t_1})^T(^{N}\vec{r}_{i,t_1} - {^{N}\vec{r}_{O_X,t_1}}), \tag{4.25}$$

with X standing for R, F or C, depending on the frame that the markerset corresponds to. Assuming that the bodies associated with the frames do not deform, $^{X}\vec{r}_i$ remains the same over time, despite the change in the other terms of Equation 4.25. With these position vectors and with position vectors $^{N}\vec{r}_{i,t_j}$, as captured by the MoCap recordings for all visible markers in the set at a certain time t_j, $^{N}R_{X,t_j}$ can be computed following Equations 4.22 to 4.24.

4.1.3. Ground contact points

As mentioned in Section 3.3.1, the GCPs of the wheels can give information about the configuration on the treadmill. With the Qualisys measurements and rotation matrices available, their locations can be easily computed using the relative position vectors from the wheel centers to the GCPs:

$$^{N}\vec{r}_{P/M24,t_j} = {^{N}R_{R,t_j}} \begin{bmatrix} 0 \\ 0 \\ -r \end{bmatrix} \tag{4.26}$$

and

$$^{N}\vec{r}_{Q/M21,t_j} = {^{N}R_{F,t_j}} \begin{bmatrix} 0 \\ -rsin(\alpha) \\ -rcos(\alpha) \end{bmatrix}, \tag{4.27}$$

where P is the rear wheel GCP, Q the front wheel GCP, r the wheel radius and α the steer axis tilt. The absolute location of the GCP's is then given by

$$^{N}\vec{r}_{P,t_j} = {^{N}\vec{r}_{M24,t_j}} + {^{N}\vec{r}_{P/M24,t_j}} \tag{4.28}$$

and

$$^{N}\vec{r}_{Q,t_j} = {^{N}\vec{r}_{M21,t_j}} + {^{N}\vec{r}_{Q/M21,t_j}}. \tag{4.29}$$

It should be noted that this absolute position is relative to the origin of the inertial N frame, of which the location was not precisely known. Because the L-frame for the calibration was placed against a flange on the left side of the treadmill, the x-coordinate at the farmost left side of the belt shall approximately be zero, with an estimated uncertainty of 0.05 m at maximum.

4.2. Angular velocities

The angular velocities of interest can easily be obtained from the IMUs. The lean rate $\dot{\phi}$ is given by the rear frame IMU's y-axis gyroscope data, multiplied by -1, and the steering rate $\dot{\gamma}$ by the front frame IMU's z-axis gyroscope data. The rear wheel rotation rate $\dot{\chi}_R$ and the cadence $\dot{\chi}_C$ follow from the rear wheel IMU's y- and the crank arm IMU's z-axis gyroscope data.

For angles that have both position and velocity data available, additional processing can give further insight in the correspondence of the two signals. This is the case for ϕ and γ. Use can be made of the fact that

$$\phi = \int \dot{\phi}\, d\phi + c_1 \tag{4.30}$$

and

$$\gamma = \int \dot{\gamma}\, d\gamma + c_2. \tag{4.31}$$

Before this integration, the mean of each velocity signal is computed and subtracted to compensate for sensor bias. Then, the integration can be done by means of a cumulative trapezoidal numerical integration and the resulting signals are downsampled to the MoCap frame rate. Now the position data obtained from two different methods is available and computing the cross correlation as a function of the signal shift can provide insight in the similarity of the two signals and the lag between them. However, an angle of $0°$ here corresponds to the bicycle being in upright position for ϕ and steering straight ahead for γ. Slight inaccuracies in the measurements can thus result in one signal being positive while the other is negative, which gives poor results for the cross correlation. Therefore, $90°$ is added to both signals before computing the cross correlation, such that a value of $0°$ corresponds to the bicycle lying flat on the ground for ϕ and a $90°$ steering angle for γ.

4.3. Force data

The force data are acquired and saved by the controller. This contains both the desired and measured forces for all motors. Performance measures will be defined and used to evaluate the controller performance. In order to regard the effect of the forces on the other signals acquired by the Qualisys system, they need to be synchronized. Both these topics will be discussed further in the subsections below.

4.3.1. Performance measures

How well the actual force signals track the reference will be evaluated for each signal in terms of rise time and overshoot. Usually, these measures are expressed in terms of the output signal's steady-state value for step responses. However, because of the short duration of the perturbations, the output signal does not settle to a steady state value in this experiment. Therefore, these measures are expressed here in terms of the target value F_{des}. The custom definitions of the measures are shown below:

- The rise time is defined as the time after which the output signals first cross 80% of the target value F_{des} since the instant at which F_{des} rises.
- The overshoot is defined as the extent that the output signals first peaks exceed the target value F_{des}, expressed as a percentage. In the case that the output signal's first peak does not reach the target value, the amount that the peak value is short of the target is reported as a negative percentage.

Additionally, the resulting moment on the handlebar will be computed and integrated to obtain the angular momentum. Assuming that the forces are applied in straight forward and backward directions, the moment vector can be calculated as

$$^{N}M_{t_j} = (r\,^{N}\hat{e}_{xF,t_j}) \times ((F_{1,t_j} + F_{3,t_j} - F_{2,t_j} - F_{4,t_j})\,^{N}\hat{e}_{yN}), \tag{4.32}$$

in which

- r is half the distance between the forces' points of attachment on the handlebar,

- ${}^{N}\hat{e}_{xF,t_j}$ is the first column of ${}^{N}R_{F,t_j}$,
- F_{1,t_j} to F_{4,t_j} are the force measurements of module 1 to 4,

  and
- ${}^{N}\hat{e}_{yN}$ is the y-directional unit vector in the inertial frame.

By computing the dot product of ${}^{N}M_{t_j}$ and ${}^{N}\hat{e}_{zF}$ the moment about the steering axis can be obtained. Cumulative trapezoidal numerical integration over the time of the perturbation leads then to the angular momentum H. This result can then be compared to the angular momentum resulting from the desired force.

4.3.2. Synchronization

To synchronize the force signals saved by the controller with the data signals acquired on the Qualisys, the synchronization signal can be used, through which the Qualisys receives information about the perturbation system. All signal indices at which the difference between the according value and the previous one is greater than 1 V, are marked as occasions. Based on the pull duration and the signal value before the pull, occasions that mark the start of a pull can be extracted. Removing those plus the subsequent indices leaves the occasions that mark the activation and deactivation of the perturbation system.

Having identified all the relevant occasions, the according times can be obtained. Adding the start times of the perturbation system to the Qualisys recording datetime approximately gives the force files datetime, by which they can be found.

4.4. Participant performance

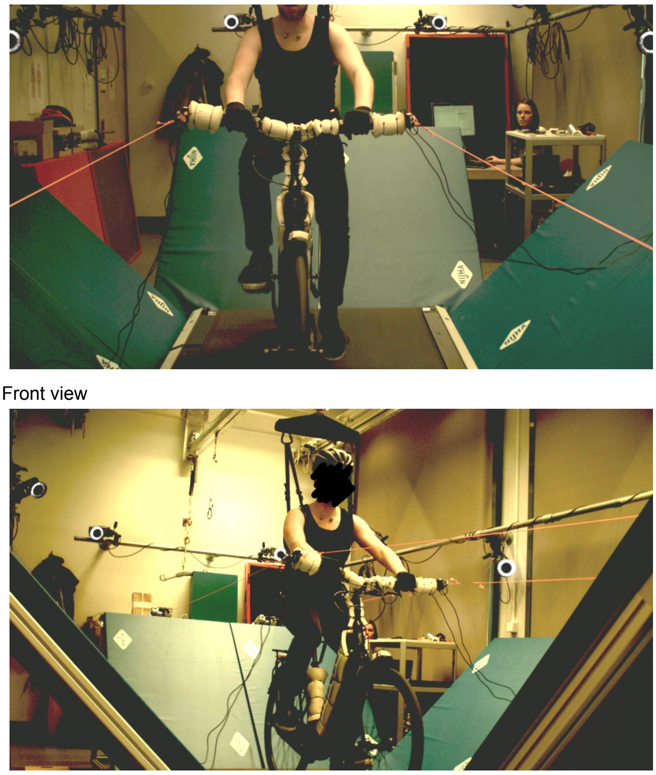
For each of the perturbations, the reaction of the participant is reviewed and the outcome is labelled as a recovery or a fall. This results thus in a binomial distribution in which the perturbation force can be regarded as a predictor variable. Subsequently performing a linear regression on this data can give insight in the probabilities to fall for a continuous perturbation force interval. This is done by fitting a line of the form

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{4.33}$$

through the data, with β_0 and β_1 chosen such that the likelihood function is maximized.

5 Results

In the process of developing this experiment, 12 people have participated in various pilot tests. The first part of these pilots was used to test individual aspects of the experiment and ask participants their opinions, which have been taken into account in the eventual design choices. In later pilots, more aspects came together to the eventual full experiment. All of the participants indicated that they felt safe during these tests and were not injured in any way. The results presented in this chapter originate from the final pilot experiment. Figure 5.1 shows pictures of the experimental setup, while the participant was cycling on it. In the next sections an overview of all perturbations given during this pilot study is presented, after which some examples of them are shown in more depth, regarding the perturbation forces and the bicycle-rider system data. The final section discusses the correspondence between the MoCap data and the IMU data.



(a) Front view

(b) Oblique view

Figure 5.1: Pictures from final pilot study.

5.1. Participant performance

This section gives an overview of all perturbations that were delivered during the experiment and how the cyclist handled them, which is shown in Figure 5.2 in the form of a bubble chart. On the x-axis, one finds the magnitudes of the perturbations expressed in the desired pull force and the left y-axis indicates whether the participant could recover or falls. The size of the bubbles indicates the number of pulls with that magnitude-outcome combination. It can be seen that for perturbations smaller than 50 N, the participant was always able to recover and that perturbations of 70 N and larger consistently resulted in a fall. Between these values is a transient region, where the results are more variable. Larger bubbles, indicating more pulls with the same magnitude, typically originate from the exploration phase of the experiment whereas smaller bubbles generally come from the randomized phase. Additionally, the dashed line, associated with the right y-axis, gives the fall probability based on the logistic regression as explained in Section 4.4. The low probabilities to fall below forces of 50 N and high probabilities above 70 N support the previous observations.

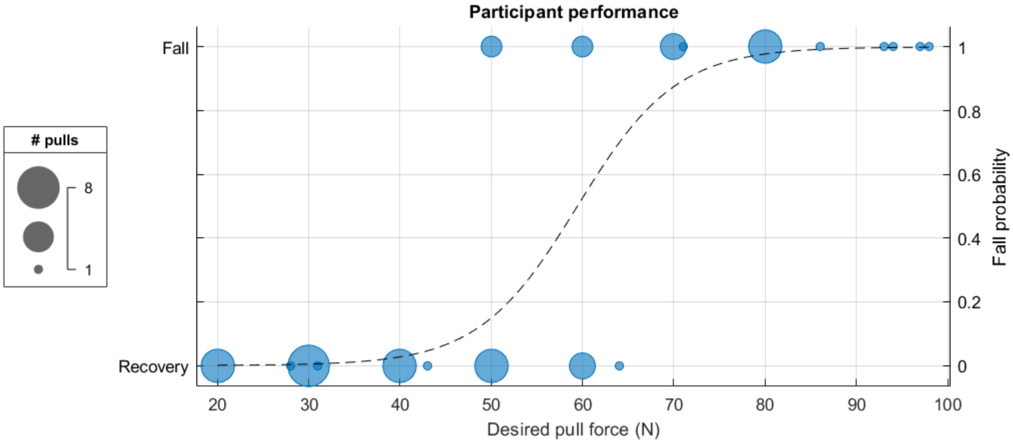


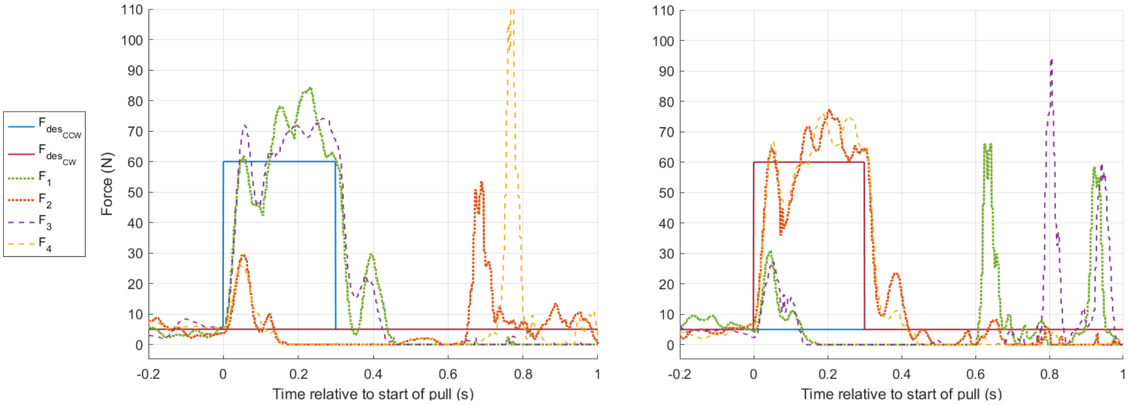
Figure 5.2: Participant performance as a function of the desired pull force. The size of the bubbles indicates the number of pulls with a certain force and outcome. In total, this overview contains 48 perturbations. The dashed line shows the fall probability based on the logistic regression. The 50% probability to fall is crossed at a desired pull force of 59.43 N.

5.2. Controller performance

Whereas the overview in the previous section only shows the desired perturbation forces, this section dives deeper into the forces that are actually delivered. By means of analysed examples, it is illustrated how well the forces measured in the ropes follow the reference forces. Figure 5.3 presents two such examples for perturbations with desired pulling forces of 60 N, in which (a) is a counter-clockwise perturbation and (b) is a clockwise perturbation. Motors 1 and 3 need to pull to perform a counter-clockwise perturbation and motors 2 and 4 for a clockwise perturbation. $F_{des,CCW}$ is thus the reference for F_1 and F_3 and $F_{des,CW}$ for F_2 and F_4. From here on, the motors associated with to the rising reference shall be called 'the perturbation motors' and the motors that correspond to the other, low reference are named 'the tracking motors'. The time scales are adjusted such that $t = 0$ s corresponds to the instant at which the reference is starting a perturbation. All measurements below 0 N are cut off, as they can not physically be the result of a pulling rope. The rise times and overshoots as defined in Section 4.3.1 are given below each plot.

Both plots show that soon after $t = 0$ s, the forces effectuated by the perturbation motors start to rise, as intended. After this initial increase, they fluctuate around the desired force, without settling to a steady state before the perturbation ends. Shortly after the reference drops, the perturbation forces follow this signal downward with some bumps and undershoot.

Interestingly, the forces associated with the tracking motors also start to rise shortly after the perturbation forces at the beginning of the perturbation, although to a lesser extent. After these peaks, they fall down to values cut off at zero, only to rise to high peaks after at least 0.6 s. Looking at the according video fragments included in the additional digital material, reveals that this effect is the result of the following course of events: when the perturbation motors start to pull on the rope, the handlebar rotates in the direction of their pulls, resulting in higher forces in the opposing ropes. The controller corrects for the extreme forces in the tracking signals and the peaks decrease, but then the participant starts to correct for the pull on the handlebar by steering to the other side and the ropes attached to the tracking motors are hanging slack. The controller tries to adjust for this low force and the tracking motors wind up the ropes. However, because of the slack in the rope, feedback lacks and new peaks arise when the ropes suddenly come to tension.

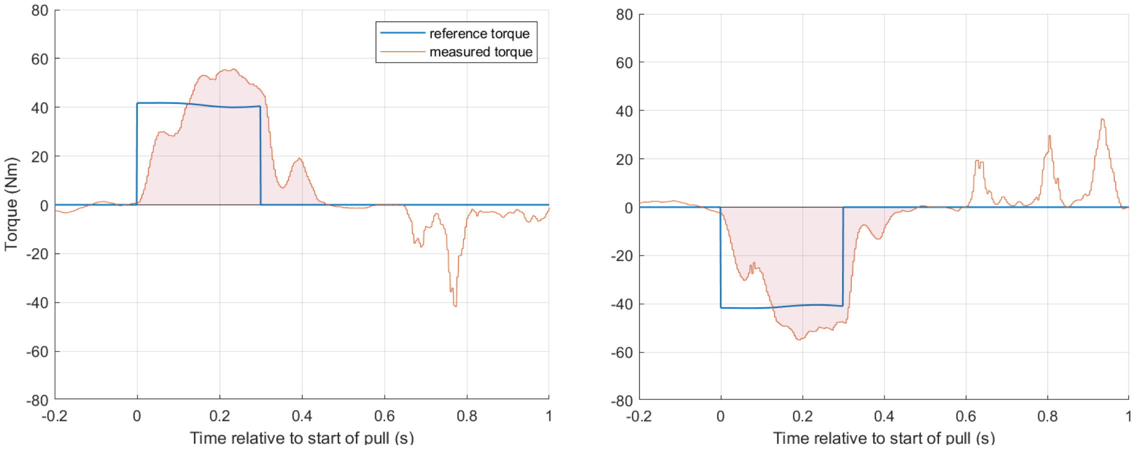


(a) in counter-clockwise direction
F_1: rise time = 0.036 s; overshoot = 2.97%
F_3: rise time = 0.036 s; overshoot = 19.77%

(b) in clockwise direction
F_1: rise time = 0.032 s; overshoot = 7.64%
F_3: rise time = 0.032 s; overshoot = 11.49%

Figure 5.3: Measured forces around two perturbations of opposite directions, with desired forces of 60 N.

All these forces contribute to the net torque on the handlebar. For the perturbations from Figure 5.3, this torque is approximated as described in Section 4.3.2 and visualized in Figure 5.4. The red line shows the net torque from all measured forces and the blue line represents the torque when all forces would be equal to their reference, but with equal steer rotation (since that data is otherwise unavailable). The area under the graph represents the angular momentum H. That of the reference is almost rectangular-shaped, except for the small decline at the end of the perturbation, which is caused by the rotating steer. The actual torque is less consistently shaped. At first, the torque does not directly reach the value of the reference torque, because the opposing ropes deliver an opposite contribution. After a dip, the value from the reference is exceeded. From the moment that the perturbation is supposed to end, the actual torque keeps lingering for a while before it reaches a value of 0 N m. This greatly extends the angular momentum of the pull, which is visualized by a substantial amount of red area after $t = 0.3$ s. The opposite force peaks that were visible in Figure 5.3, result here in opposite torque peaks as well. Their areas are considerable, yet not close to the area of the actual pull.



(a) in counter-clockwise direction

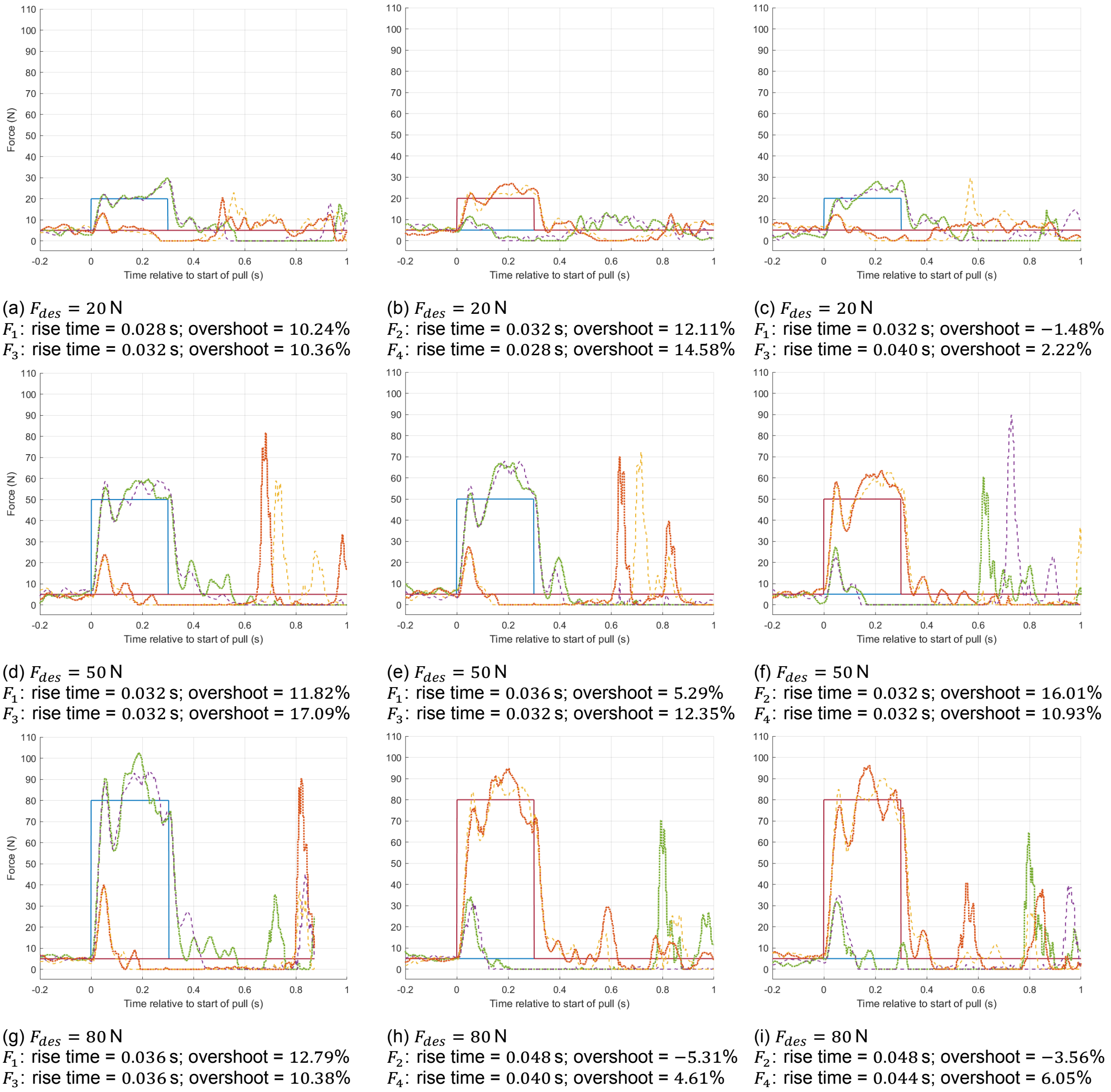
(b) in clockwise direction

Figure 5.4: Net torque on handlebar based on the measured and reference forces for the perturbations from Figure 5.3.

Figure 5.5 and Figure 5.6 show more plots like these, for pulls with references of 20 N, 50 N and 80 N. There are three examples for each magnitude, chosen such that they represent a variety of pull directions and outcomes, which will be revealed further in this chapter. From figures 5.3 and 5.5 it can be seen that short rise times (<50 ms) are achieved for all perturbations, even though they become longer with higher perturbation magnitudes. The occurrence of an initial peak with an overshoot of -10 to 20%, followed by a dip and then some fluctuation above the reference force seems to be a general pattern for all perturbations. The behaviour of the opposite forces described earlier also appears typical.

The torque plots from Figure 5.4 and Figure 5.6 all show the relatively low values at the beginning and higher values at the end of the perturbation. It also becomes clear that the effect of the rotating steer on the torque is more substantial for higher perturbations.

An overview of the angular momentums from all given examples is shown in Figure 5.7. This figure shows a continuous *static* estimate, which is based on constant forces acting on a perpendicular arm for 0.3 s, and discrete points resulting from the given examples. Those based on the reference forces are called the *dynamic* estimates and those based on the measured forces the *dynamic nonlinear* estimates. It appears from this figure that the dynamic nonlinear estimates are consistently higher than the dynamic estimates. What also stands out, is that multiple pulls with the same reference magnitude lead to very similar angular momentums.



(a) F_{des} = 20 N
F_1: rise time = 0.028 s; overshoot = 10.24%
F_3: rise time = 0.032 s; overshoot = 10.36%

(b) F_{des} = 20 N
F_2: rise time = 0.032 s; overshoot = 12.11%
F_4: rise time = 0.028 s; overshoot = 14.58%

(c) F_{des} = 20 N
F_1: rise time = 0.032 s; overshoot = -1.48%
F_3: rise time = 0.040 s; overshoot = 2.22%

(d) F_{des} = 50 N
F_1: rise time = 0.032 s; overshoot = 11.82%
F_3: rise time = 0.032 s; overshoot = 17.09%

(e) F_{des} = 50 N
F_1: rise time = 0.036 s; overshoot = 5.29%
F_3: rise time = 0.032 s; overshoot = 12.35%

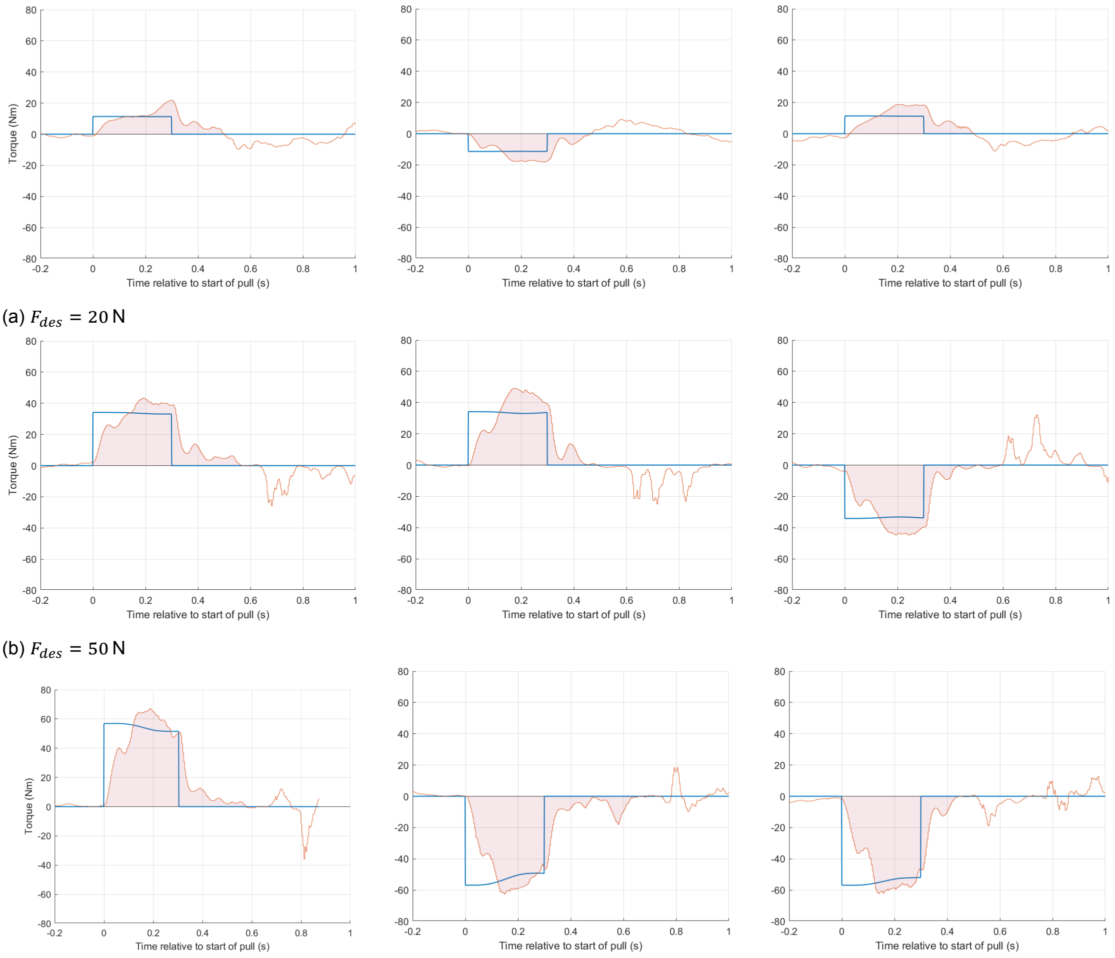
(f) F_{des} = 50 N
F_2: rise time = 0.032 s; overshoot = 16.01%
F_4: rise time = 0.032 s; overshoot = 10.93%

(g) F_{des} = 80 N
F_1: rise time = 0.036 s; overshoot = 12.79%
F_3: rise time = 0.036 s; overshoot = 10.38%

(h) F_{des} = 80 N
F_2: rise time = 0.048 s; overshoot = -5.31%
F_4: rise time = 0.040 s; overshoot = 4.61%

(i) F_{des} = 80 N
F_2: rise time = 0.048 s; overshoot = -3.56%
F_4: rise time = 0.044 s; overshoot = 6.05%

Figure 5.5: Extension to Figure 5.3 with desired forces of (a-c) 20 N, (d-f) 50 N and (g-i) 80 N. Three examples of varying directions are shown for each case, with rise times and overshoots reported below each plot. For the legend, the reader is referred to Figure 5.3.



(a) F_{des} = 20 N

(b) F_{des} = 50 N

(c) F_{des} = 80 N

Figure 5.6: Extension to Figure 5.4 for the perturbations from Figure 5.5. For the legend, the reader is referred to Figure 5.4.

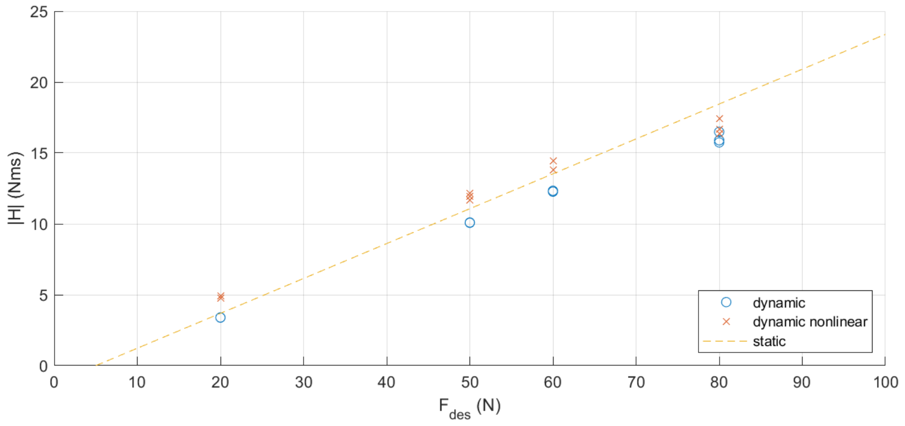


Figure 5.7: Angular momentum as obtained from static, dynamic and dynamic nonlinear estimates.

At this point it might be a logical next step to recreate Figure 5.2 with angular momentum on the x-axis, because this gives more insight in the actual delivered inputs to the system. However, one set of force data was unfortunately not saved properly, so not all data to make this overview is available.

5.3. Bicycle-rider data

In addition to the force data, the data capturing the bicycle-rider dynamics was also gathered. This section presents that data during the time around the same perturbations from the previous section.

Figure 5.8 shows this data during 2 s before and 5 s after the starts of the perturbations from Figure 5.3. This data comprises the force references $F_{des,CCW}$ and $F_{des,CW}$, the bicycle yaw and lean angles ψ and ϕ, the steering angle γ, the upper body angles relative to the bicycle ζ, η and θ, the derivatives $\dot{\phi}$ and $\dot{\gamma}$, the rear wheel rotation rate $\dot{\chi}_R$ and the cadence $\dot{\chi}_C$. The rear and front wheel GCP positions are also shown, along with horizontal lines that represent the sides of the treadmill. These lines are however approximations, because it is not exactly known where the $x = 0$ line is located. Section 4.1.3 mentions an uncertainty up to 0.05 m. In Figure 5.8 (a), the participant recovers from the perturbation and in (b), the participant falls.

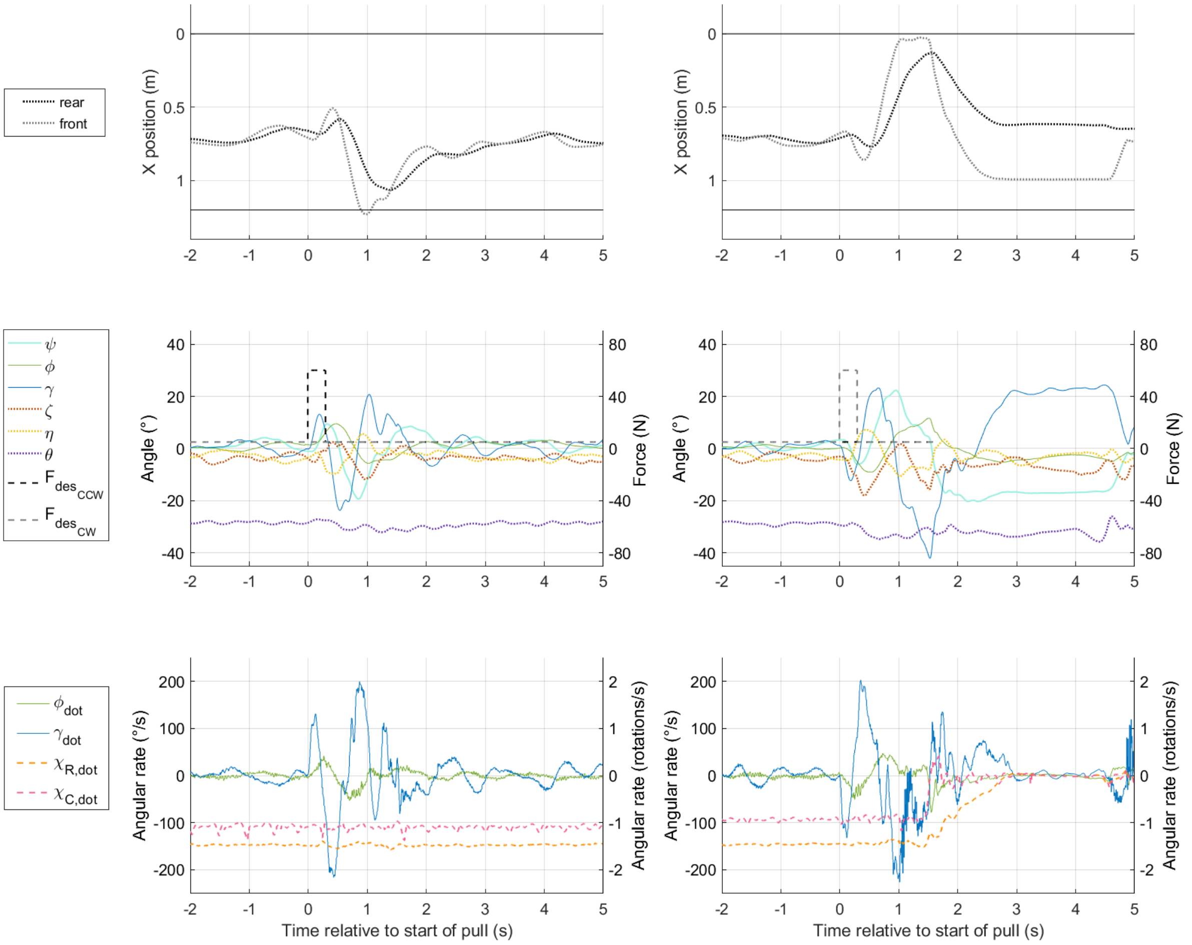
In the middle plot of Figure 5.8 (a), it can be seen that immediately after the start of the counter-clockwise perturbation, the steering angle γ rises beyond the small oscillations that occur during steady cycling to a positive value of approximately $13°$. Halfway during the perturbation, the steering angle already comes back down and at the end of the perturbation, it shoots to a negative angle of even higher magnitude, which is in turn corrected again. The bicycle's yaw angle ψ is strongly influenced by the steering angle, showing similar behaviour with a small delay. The lean angle ϕ shows smaller peaks of opposite sign in comparison with the steering angle, but with a similar frequency. Note that a steer angle to the left is defined positive, whereas a lean angle to the left is defined negative. The cyclist's yaw angle ζ and lean angle η show most severe deviations within 1.5 s starting halfway the perturbation. The latter moves in opposite direction of the bicycle lean. The cyclist's pitch angle θ is little affected by the perturbation. Velocity signals $\dot{\phi}$ and $\dot{\gamma}$ in the lower plot look like plausible derivatives of the corresponding position data, satisfying the prerequisite to have zero-crossings where the position data peaks. The cadence and rear wheel rates $\dot{\chi}_R$ and $\dot{\chi}_R$ remain steady over time. Their negative sign is just a consequence of the axes definitions. After a few seconds, all signal oscillations look similar to those from before the perturbation.

The upper plot, which shows the rear and front wheel GCP positions, indicates a large deviation of the front wheel to the right starting at $t \approx 0.4$ s. Around $t \approx 1$ s, it crosses the right border of the treadmill belt, but inspecting the according video material shows that this doesn't actually happen. This thus indicates that the horizontal lines should have been placed at slightly higher values for x. The rear wheel GCP shows a similar movement, but in a reduced form. Soon after the extreme deviation, both GCPs move back to the middle of the belt.

When comparing the data from Figure 5.8 (a) to that shown in (b), which considers a clockwise perturbation, one sees that all recorded angles move in opposite direction. Although opposite, the bicycle's yaw, lean and steering angles initially show peaks of comparable magnitudes, but after 1 s, it becomes clear that these angles become bigger than in the previous case. Simultaneously, the upper plot visualizes that the front wheel GCP comes very close to the indicated left edge of the treadmill belt. The peak is not smooth, but seems cut off at the top. Shortly hereafter, the cadence and rear wheel speed in the bottom plot come to a halt. These are indications of a fall. When inspecting the video, it becomes clear that the cyclist steered off the treadmill in this case, after which the Rysen operator stopped the treadmill. This again shows that the horizontal lines representing the belt edges are placed at too low values for x.

More such data visualisations, corresponding to the previously shown perturbations of 20 N, 50 N and 80 N are contained in Figure 5.9. The amount of falls and recoveries is representative of the usual outcomes for each perturbation force as discussed in Section 5.1.

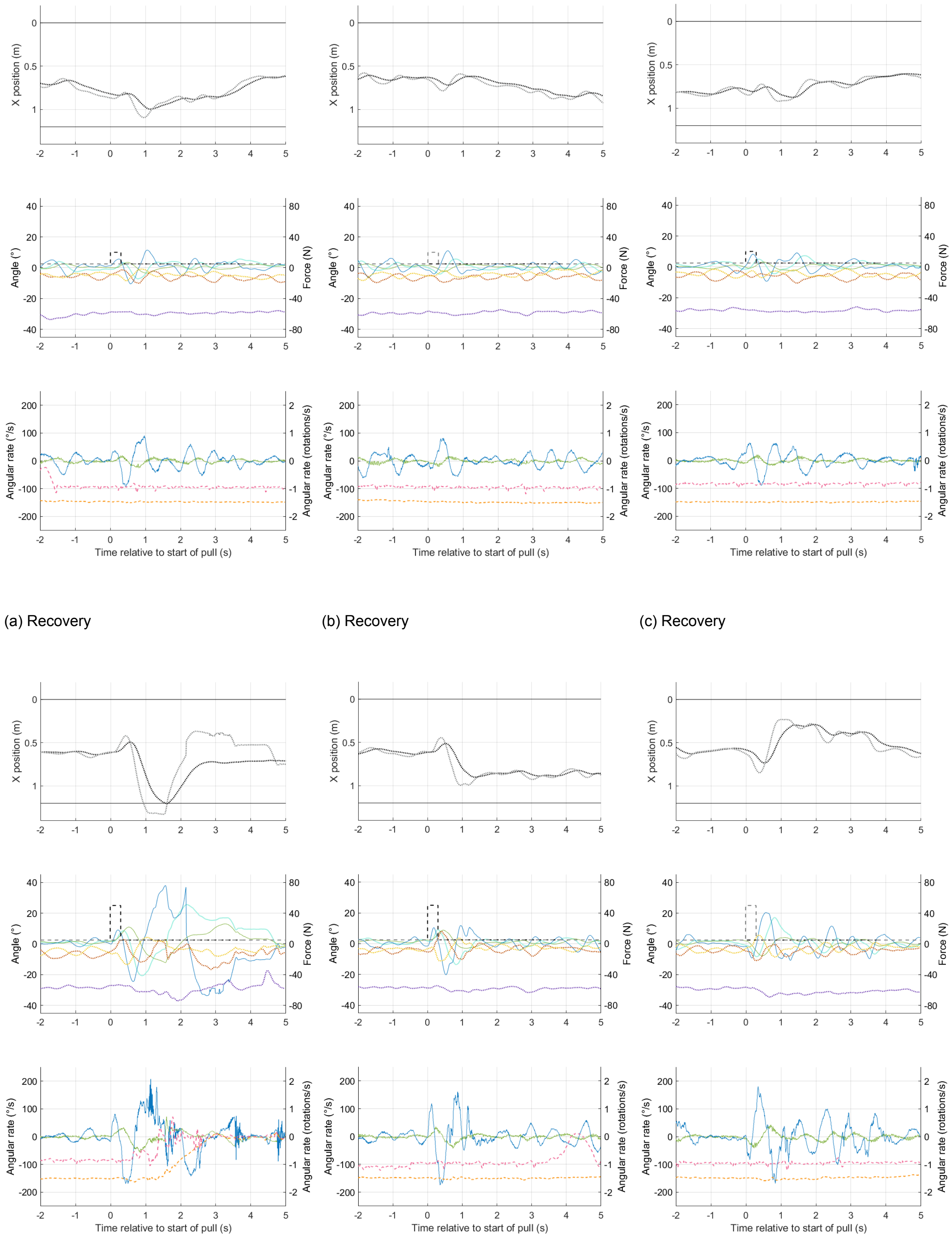
Overall, higher perturbations lead to higher initial steering angles and thus higher chances to fall. For perturbations with relatively low forces, the width of the treadmill is often the limiting factor that keeps the cyclist from recovery. For high-force perturbations, the cyclist often falls before the bounds of the treadmill are reached. A halt in the cadence shortly after the start of the perturbation clearly indicates such an event. Clear, early signs of the cyclist riding off the treadmill are given away by the front wheel GCP. In all cases, it seems to be determined within approximately 1 s whether the person falls. For high-perturbation falls, the cadence signal often indicates the fall significantly earlier than this.



(a) Recovery after counter-clockwise perturbation

(b) Fall after clockwise perturbation

Figure 5.8: Combined datasets for the perturbations from Figure 5.3. The desired pull forces were 60 N, with in (a) a counter-clockwise pull which the rider could recover from and in (b) a clockwise pull which led to a fall. The upper plot shows the x-coordinates of the front and rear wheel GCPs together with the approximate locations of the belt edges, the middle plot shows the bicycle's yaw (ψ), lean (ϕ) and steering (γ) angles and the cyclist's yaw (ζ), lean (η) and pitch (θ) angles and the bottom plot shows the angular rates of the bicycle's lean (ϕ_{dot}) and steering (γ_{dot}) and the rear wheel rotation speed ($\chi_{R,dot}$) and cadence ($\chi_{C,dot}$). Data shown as solid or dotted lines are associated with the left y-axes and data shown as dashed lines are associated with the right y-axes



(a) Recovery

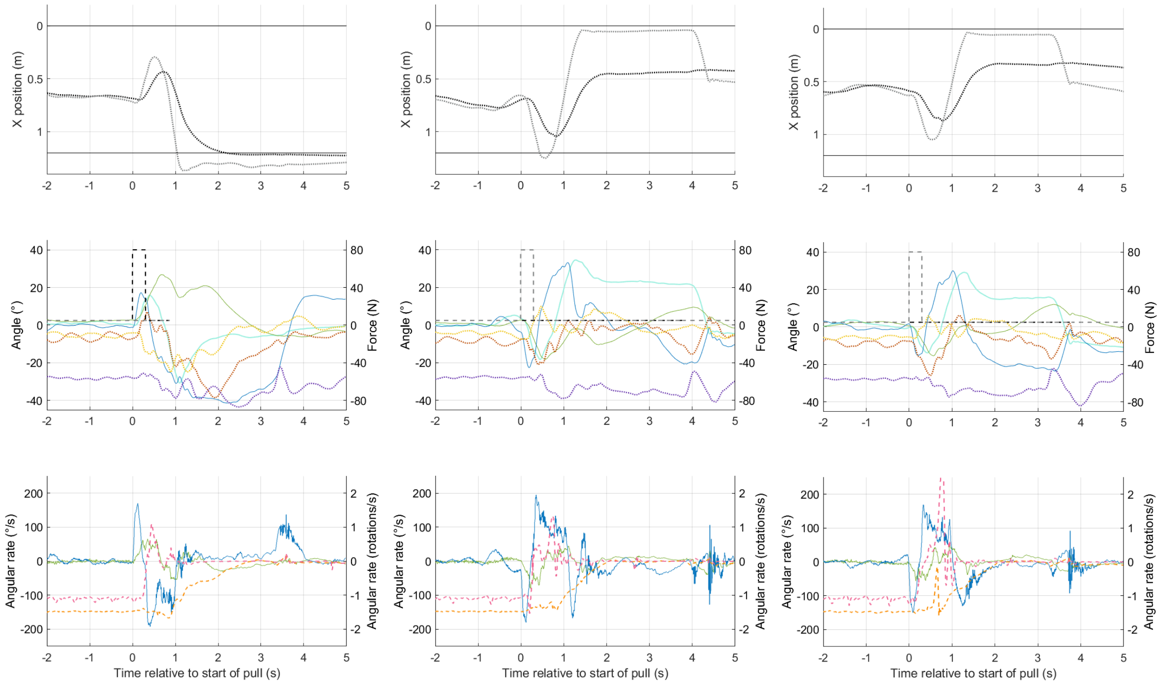
(b) Recovery

(c) Recovery

(d) Fall

(e) Recovery

(f) Recovery



(g) Fall

(h) Fall

(i) Fall

Figure 5.9: Extension to Figure 5.8, for the perturbations from Figure 5.5. It is indicated for each case whether the disturbance resulted in a fall or recovery. For the legend, the reader is referred to Figure 5.8.

5.4. Qualisys data correspondence

Having shown all relevant captured data, this section discusses the quality of the MoCap and IMU data based on their similarity. Figure 5.10 shows the bicycle lean angle ϕ and steering angle γ as obtained from the MoCap and from the integrated IMU signals for the first full Qualisys recording. From a first view, both signals seem to have a comparable shape. It also appears that the integrated IMU signals contain some random walk, which is a known problem intrinsic to IMU signals.

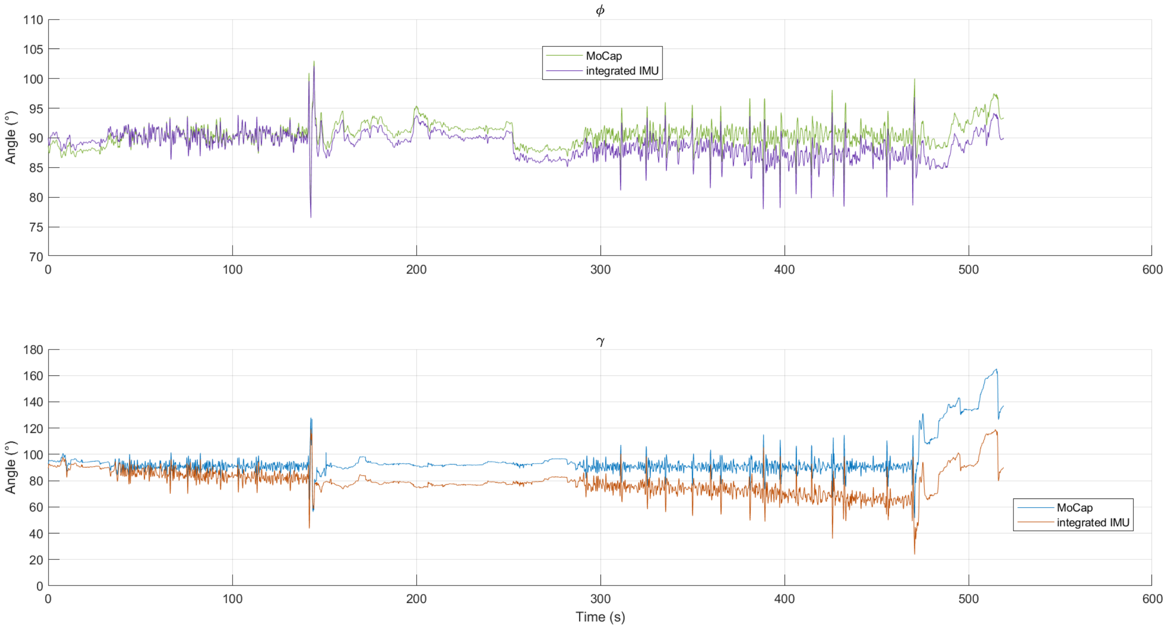


Figure 5.10: Recording 1 of the lean angle ϕ and steering angle γ from the MoCap and from the integrated IMU signals.

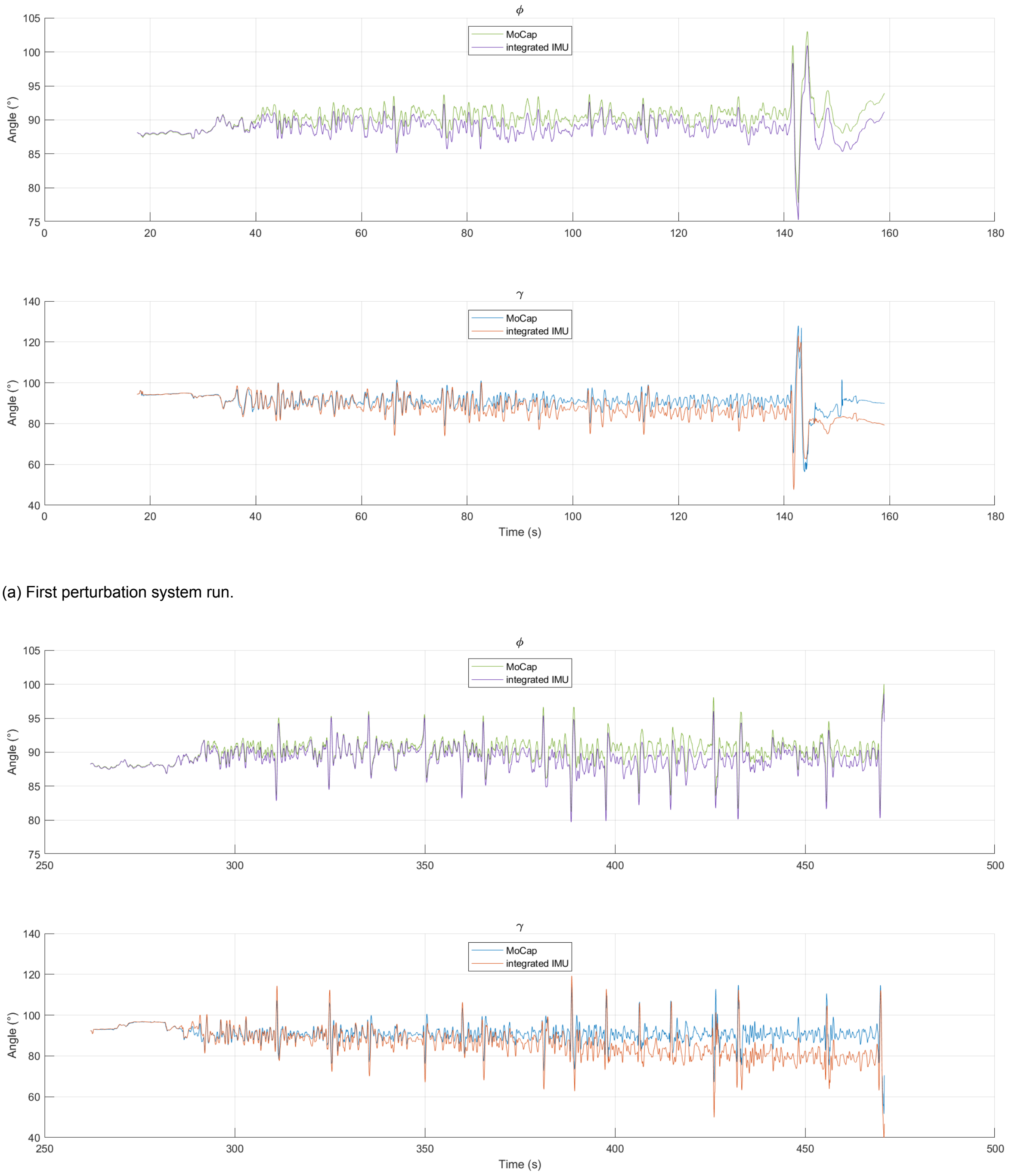
This Qualisys recording contains two runs of the perturbation system. Figure 5.11 zooms in on these, to see the impact the random walk has within this time frame. The first run shown in (a) takes approximately 140 s and the second run shown in (b) takes a little over 200 s. The first time instants are corrected such that the signal pairs start at the same angles. within these relatively short durations, the respective maximum differences are $3.28°$ and $2.35°$ for ϕ and $25.94°$ and $23.77°$ for γ.

Normalized outcomes of a cross correlation analysis are shown in Table 5.1, along with the IMU lag at which the maximum cross correlation is found, for all five Qualisys recordings. The latter is expressed in the number of Qualisys frames, which are captured at 100 Hz. The maximum cross correlations are all close to 1, indicating good correspondence of the two signals. Most of these maximum cross correlations were found at zero IMU lag, with exception of the first two steering angle recordings. Here values of -1 and -2 are reported, implying that the IMU recordings lead up to 0.02 s.

		Recording 1	Recording 2	Recording 3	Recording 4	Recording 5
ϕ	max x-cor	0.9999	1.0000	1.0000	0.9998	0.9995
	IMU lag	0	0	0	0	0
γ	max x-cor	0.9936	0.9990	0.9993	0.9739	0.9926
	IMU lag	-1	-2	0	0	0

Table 5.1: Correspondence of MoCap data with integrated IMU data expressed as maximum cross correlation and the lag of the IMU signals at which the maximum cross correlation occurs. The latter is expressed in the number of MoCap frames.

As another check of the quality of the MoCap data, the computed rear frame pitch χ and the steer axis tilt α and bank β are analysed in Appendix C.



(a) First perturbation system run.

(b) Second perturbation system run.

Figure 5.11: Zoomed in version of Figure 5.10, to inspect the recordings during the time frames that the perturbation system was active.

6 Discussion

Having shown the most important results, this chapter proceeds with the discussion. This discussion evaluates the eventual setup that was built during this project, with the goal to investigate cycling falls in order to learn more about how a cyclist controls a bicycle in presence of perturbations and especially to what extent they can be handled without falling. This review consists of two parts. The first part checks the setup against the requirements from Chapter 2, with the help of the results presented in Chapter 5. Subsequently, the second part discusses other findings that emerged during the pilot studies.

6.1. Adherence to requirements

This section discusses to what extent the requirements set in Chapter 2 are fulfilled. The requirements were ordered in three sets, each concerning different parts of the experiment. The first set of requirements concerned *cycling* during the experiment. The setup must allow *realistic* cycling and *effective steering*. Moreover, it must be possible to conduct the experiment at different cycling *speeds* and *safety* of the cyclist must be assured (also during falls).

Regarding the *realisticness*, the following observations are made. During the experiment, an ordinary bicycle is used without changing any of the usual mechanisms. This bicycle is placed freely on a treadmill, which means that the cyclist is in charge of its balance and direction. The relative motion between the wheels and the ground surface is similar as on a road and the rotation speeds of the front and rear wheels are coupled because they are in contact with the same surface. All these arguments are in favour of the hypothesis that cycling dynamics on a treadmill are realistic.

This theory is supported by the findings of the pilot studies: once the participants mastered the skill of getting started, the cycling itself offered no additional problems. This condition is formulated as such because getting started is for sure different on the treadmill than in the exterior. After all, on a treadmill the ramp-up of the cycling speed is determined by the acceleration of the belt, whereas normally the cyclist himself is in control of this speed. Additionally, the cyclist needs to get used to the contradiction of external inputs: one is used to see a moving environment when cycling, which doesn't happen on a treadmill. However, participants of the pilot studies got accustomed to these differences quite fast. Some people could even cycle successfully from the second try and the 'slowest' learners managed to do so within 10 minutes.

Also the possible rider inputs must correspond to those in a normal situation. During the experiment, the participants were free to lean, steer and change gears as they liked. Steering felt heavier than usual because of the pretension in the ropes, which could be compared to cycling with a loaded basket on the steering assembly. This effect could be reduced by attaching the ropes closer to the middle of the handlebar. The participant's feet were semi-constrained to the pedals by straps, from which they could escape if they tried. Knee movements could still be performed with feet in the straps. The participant was also constrained to the bicycle by means of a safety belt around the waist, attached to the saddle. This makes that he is not able to stand, but the measure is necessary for safety reasons.

Regarding the space the cyclist has to manoeuvre, the belt width of the treadmill is compared to the guidelines for road designs. The road design manual advices a minimal width of 1.25 m for bike lanes, in contrast to more spacious dimensions for separate bicycle paths (Fietsersbond, n.d.). The belt width of 1.20 m comes quite close to this and is thus considered *realistic*.

The last thing comprised in the *realisticness* requirement, was that the cyclist is not prepared for a coming perturbation. Such a naive mindset is hard to replicate in the experiment environment. For this to be possible, the participant should be deceived as to the intention of the experiment, which is ethically questionable, and even then, the element of surprise is lost after the first perturbation. For these reasons, the participant was informed about the perturbations, but instructed to ride as relaxed as possible.

The requirement of *effective steering* means that the mechanism of steering into the fall must be executable. This clearly is the case when no perturbations are given and the results presented in Section 5.3 show that even during perturbations, the cyclist is able to decrease the steering angle before a perturbation ends. This is for example visible in Figure 5.8. This argues that the cyclist still has the possibility to steer into the fall. The extent to which this is applicable of course depends on the available space left on the belt.

The *speed* requirement states that it must be possible to select the cycling speeds such that experiments can be conducted below, in and above the bicycle's self-stable region. The treadmill that was available for this experiment can facilitate speeds up to 18 km h^{-1}, which corresponds to 5 m s^{-1}. For the benchmark bicycle from Meijaard et al. (2007), boundaries of the self-stable region lie at approximately 4.3 m s^{-1} and 6.0 m s^{-1}. In Schwab et al. (2012), the stability regions for the Whipple bicycle model extended with a passive rider are discussed. It emerges that for the models that do show self-stable behaviour, the boundaries lie higher than for the benchmark model. According to the theory, the maximum treadmill speed is thus definitely too slow to represent speeds above the self-stable region, and representation of the self-stable region itself is doubtful in some cases. In practice however, participants reported that balancing felt easiest around speeds of 10-12 km h^{-1}. Cycling at slower speeds on the one hand, e.g. 6 km h^{-1}, required more control effort. On the other hand, when cycling at higher speeds of 18 km h^{-1} the cyclist ended up off the belt more quickly after perturbations.

The extent to which the setup satisfies the *safety* requirement is hard to express with absolute certainty. After all, accidents can never be ruled out entirely. However, during the development of the setup, close contact was maintained with the Human Research Ethics Committee, whose approval is required before experiments can take place. Under their supervision, risks have been evaluated and prevention measures have been taken. This resulted in their approval being granted. Additionally, none of the pilot study participants said to experience discomfort during the experiments and after the pilot studies, the experiment was safely conducted with another 26 participants. It once happened that one of the ropes hit backwards after a rupture of the breakaway cable. Although this is not dangerous for the participant, it can form a risk for bystanders of the experiment.

The second set of requirements is aimed at the perturbations. These requirements considered *magnitude*, *responsiveness*, *randomness* and *controllability*.

The *magnitude* requirement states that it should be possible to give perturbations of multiple magnitudes and that at least perturbation of the highest magnitude should cause a fall. During the final pilot study, falls started to occur from desired pull forces of 50 N onwards. This was comparable to performances of earlier participants. This pull magnitude is not even close to the maximum system capabilities, which is reported at 200 N.

The *responsiveness* requirement directly relates to the rise times of the perturbation force, which should be in the order of magnitude of $50-100$ ms. In Section 5.2, it was shown that they generally stay under 50 ms and thus comply with the requirement.

According to the *randomness* requirement, perturbation direction, magnitude and activation instant should be randomly determined. The Simulink application is supposed to directly take care of the former two. For the direction this works well, but less so for the magnitude. It appears that the same sequence of magnitudes repeats itself each time that the controller activates. For the activation instant of the perturbation, the controller is supposed to give an advice. This can be visualized on the screen and then the operator of the perturbation system can decide to give the perturbation. However, during the pilot studies, the task of operator was found intensive enough without keeping an eye on this screen. How well this works is thus not evaluated here. This could be done by applying a series of perturbation of the same force, which reduces the effort for the operator, but the contribution is questionable if the operator does not use it during an actual experiment.

The *controllability* requirement declares it essential that the perturbation system operator is in charge of when perturbations are delivered and what maximum forces they convey. The operator determines when perturbations are applied by clicking a button on the user interface. By default, the perturbation force is set on a low force, but the operator can change this through interaction with the MATLAB Command Window. No problems are encountered with the working of these mechanisms, but the actual forces can show some overshoot with regard to the desired force during the perturbations. The angular momentums that result from the perturbation force are however consistently close to the intended goal, as is shown in Figure 5.7. More severe is the occurrence of force peaks after correcting slack in the ropes. The sudden tension is accompanied with high forces that the operator is not in charge of. Figures 5.5 (d-f) for example show peaks after the intended perturbation that reach higher forces than the desired perturbation force. This slack in the rope seems to be related to sudden movements of the handlebar, that the controller can not cope with accordingly. In order to control this better, another control strategy together with additional sensors might be helpful. This suggestion will be elaborated upon in Section 7.1. Fortunately, the stop button on the user interface immediately stops the motors when pressed by the operator, for example when perturbations rise too high or other undesired events occur.

The last set of requirements concerned the data collection. In order to capture the *bicycle dynamics*, *rider control behaviour* and *perturbation information*, collection of relevant angles, angular velocities and force profiles was required. These were the angles that indicate the bicycle's lean and steering and the cyclist's yaw, lean and pitch, the angular velocities that represent the bicycle's lean and steering rates, the cadence and the rear wheel rotation rate and the desired and measured forces in the ropes.

Acquisition of the angles and angular velocities went very well. All this data from moments that the participant was cycling is available. The quality of the measurements was checked by evaluating the correspondence of the lean angle with the lean rate and the steering angle with the steering rate, as shown in Section 5.4. This gave good results. The IMU signals did show random walk after integration, which led to high deviations from the MoCap data over time. Even within single runs of the perturbation system, the differences rise high, which makes it undesirable to let the IMU signals replace the MoCap recordings. To evaluate the bicycle's position on the treadmill belt, the GCPs of the wheels and the bicycle's yaw angle were also obtained.

Collecting the force signals went well for most of the cases. Unfortunately, there was one Speedgoat run for which the data was not logged. This run was precisely the longest one, with the most perturbations. The length of this run might be related to the failure to save this run. As runs get longer, a point arrives at which the associated data does not fit into one file and a second logfile is created. This has not occurred during the logging test phase, and it is thus not known whether this imposes problems.

In addition to this data, parameters that describe the bicycle design are also required for the EoMs. These have not been measured for the pilots, but the bicycle's specifications sheet from the manufacturer is available and this can thus be used for follow-up experiments.

6.2. Other findings

Besides the findings directly related to the requirements, other results and observations emerged that should be discussed as well.

First of all, the participant performance as shown in Figure 5.2, which presents the perturbation outcomes and the estimated chances to fall for a continuous input interval, was not discussed in the previous section. Nonetheless, this image presents one of the results of main interest that can be obtained with this experiment. Acquiring such data for a broad group of participants allows us to say what perturbations are easy to handle, unmanageable or variable in outcome. For the latter sort it is of course interesting to dive deeper into the perturbations with different outcomes and learn what causes this distinction. This data can also serve as a baseline to compare future bicycles with, to evaluate their stability performance.

Despite the promising results, the setup does still have some teething troubles. These include occasional crashes of the controller due to CPU overloads. Such a CPU overload indicates that the processor didn't finish all calculations corresponding to a certain time step in the allocated time. Not only is it bothersome that this sometimes happens, but it also makes that no additional features can be added to the application. That would make the application even heavier and CPU overloads would occur more often. It might be worthwhile to investigate whether running the control loop at a lower frequency significantly affects the feedback behaviour. If not, this could be a solution for the CPU overloads. Otherwise, a more powerful controller is required.

Another problem is caused by temporary defects in the motors. At random moments, they stop working for a short time, leading to a slack rope. Inspecting the current sent out by the controller does show a higher value from the moment this slack occurs, which increases further over time as a result of the integral gain contribution. However, the motor does not seem to react on this input at first. When the motor does react, after approximately 0.5 s, this leads to similar high peaks seen earlier after slack.

The next point considers the outcome classification. Whether the cyclist was able to recover or fell was now determined by reviewing the recordings. Most of the times, the last perturbation before the treadmill and perturbation system were stopped, resulted in a fall and the others not, but exceptions sometimes apply. This naive classification would on the one hand miss a fall when the cyclist for example manages to regain control of the bicycle after steering off and back on the belt or after making a step on the ground with one foot. On the other side, a recovery could be faultily classified as a fall if the operators misjudged the situation and stopped the devices too soon. It is however desirable to be able to classify the perturbations without having to review all video material. In Section 5.3, it came forward that a sudden halt in the cadence and cut-off peaks in the motion of the front GCP seem to be good indicators of a fall. Hence, an outcome classification algorithm to automatically processes the data could be based on these parameters.

The last point of discussion gets back to something briefly mentioned in the previous section. It was noted that the advice on the activation instant was not used because the task of the perturbation system operator was already intensive enough. Here, I'd like to comment on the workload of the executive staff. The experiment requires at the moment three people for that. One person operates the perturbation system, another one the Rysen and treadmill and the last person is in charge of the Qualisys data acquisition.

The perturbation system operator activates and deactivates the application and delivers the perturbations. Meanwhile, he needs to adjust the perturbation forces, keep an eye on the data monitored on the screen to see if all force measurements are active and to check for CPU overloads and the location of the cyclist on the treadmill needs to be checked before delivering the pulls for safety reasons. During the final pilot study it was also tried to immediately register the outcome of the perturbation in a separate document, but the many responsibilities resulted in errors in this documentation and in difficulties to keep track of the experiment protocol. I would thus advice against noting down the outcome immediately and focus more on retrieving this in hindsight from the data by a classifier as mentioned before.

The operator of the Rysen and treadmill brings belt to the desired speed at the start of a run. After reaching this speed, he needs to stay alert on falls. If a fall occurs, the Rysen needs to be frozen and the treadmill needs to be stopped. This task could be made less intensive if freezing the Rysen simultaneously caused the treadmill to stop and if the Rysen's automatic fall detection, which is now only trained on walking, would be trained on cycling as well.

The person in charge of the Qualisys data acquisition starts the recordings, checks the amount of time left until they end and occasionally checks the IMU signals. With the currently available computer, recordings can only have a maximum duration of ten minutes. In the near future, a better computer is coming to the lab, which should allow unlimited recording times. This would make it possible to combine this task with operation of the Rysen and treadmill.

7 Conclusion

This thesis described the design of a new experimental setup which makes it possible to conduct fall experiments with cyclists. During this experiment, a cyclist rides straight ahead at one of various possible prescribed speeds while controlled perturbations can be delivered to the handlebar, whereby relevant data to capture the bicycle-rider system is measured and safety is provided by an active safety harness.

The setting allows for *realistic* cycling dynamics, although the cyclist needs to get used to the manner of starting and the lack of visual feedback, and the tracking forces did have some influence on the steering. During the experiment, *steering* was *effective*, and no issues occurred regarding the participant *safety*. The available cycling *speeds* were limited by the capabilities of the available treadmill, which doesn't reach the upper limit of the self-stable region of $6\,\mathrm{m\,s^{-1}}$. However, according to the experiment participants, balancing the bicycle was easiest around $3\,\mathrm{m\,s^{-1}}$. The perturbations demonstrated rise times that satisfied the *responsiveness* requirement and were of sufficient *magnitude* to cause falls. The perturbation direction was determined *randomly* and the operator of the perturbation device had full *control* of when perturbations were delivered. The reference forces could be followed to a certain extent, unless when slack occurred, which was consistently followed by undesirable force peaks. For repeating perturbations, the imposed forces do however lead to very consistent angular momentums, close to the estimates. The measurements of the *bicycle dynamics* and *rider control* were consistent and reliable. The measurements of the *perturbations* went well in all cases but one, and relevant information is also available through the synchronization signal.

Overall, the development of the setup is considered successful, despite the large number of different systems that had to work together. With this experiment, we are able to expose the chances that a cyclist falls after perturbations with variable input forces and the data required to regard the corresponding EoMs of the bicycle-rider system can be measured successfully. This is valuable to validate rider models and it can serve as a baseline to compare stability performance of future bicycles with. All this suggests that a repeatable experiment has been created that enables a breakthrough in cycling research.

7.1. Recommendations

The experiment could be improved by looking into the problems that are currently present. This includes CPU overloads, which could be solved by running the control loop at lower rates or by replacing the current controller by a more powerful one. For the motor defects, I would recommend to contact the manufacturer to sort out what causes the temporary deactivations. Lastly, the controller might be able to track the desired forces better when another control strategy might be used. For the current PI feedback, the system is decoupled to create multiple single-input single-output (SISO) systems. There is however a correlation between the forces in the different ropes, which can not be taken account in this way. A controller that can handle multiple-input multiple output (MIMO) systems, might thus be able to perform better. The lack of force feedback in some of the ropes might then be compensated for when the length of the ropes could be measured, for example with encoders.

In order to decrease the workload of the experiment operators, it would be beneficial if the Rysen's fall-detection algorithm was trained to react on cycling falls and was coupled to a deactivation of the treadmill. Another step forward could be obtained if the computer connected to the Qualisys data acquisition is replaced by one that can handle recordings with a duration of the full experiment.

The last point to focus on is the development of an automatic classifier to detect falls or recoveries. For this, I would recommend using at least the cadence measurements and coordinates of the front wheel GCP.

A
Comparison with BumpEm

Since the bump'em system is designed to be a modular system with multiple configuration possibilities, the best suited option for the application was to be chosen. There is a configuration for *open-loop* and for *closed-loop* control. The main difference is that a force sensor is included for the *closed-loop* version and excluded for the *open-loop* version. For this application, it was chosen to include the force sensor to allow for *closed-loop* control, such that higher rise times and better force tracking performance can be obtained. It is also possible to include an encoder for state control, but this was not considered necessary here.

Besides these anticipated design options, other changes with respect to the original design have been made as well. First of all, various dimensions of the custom parts have been adapted to metric sizes. Furthermore, the amount of power supplies has been reduced in order to cut on expenses. This is done by sharing one across two modules that do not pull simultaneously, instead of providing each module with an own power supply. Moreover, off-the-shelf S-type load cells were used as force sensor instead of the custom-made aluminium dogbones with strain gauges. The latter were preferred, but manufacturing was unsuccessful in the available time. Another point to consider is that the feedback control initially used a proportional and derivative gain, while we opted for proportional and integral gain. Although the damping effect that the derivative gain can produce is desirable, it also reacts strongly on measurement noise and it is directly related to motion jerk. Therefore, this is not used. Integral feedback is used, because it can adjust for persistent errors.

B Control applications

This appendix explains more about the applications created for the control. Section B.1 covers the design of the user interface and Section B.2 shows the content of the Simulink model.

B.1. User interface

The user interface to communicate with the controller during the experiments was created in MATLAB App Designer. The code at the backend of the buttons of the interface shown in Figure 3.7 is shown in the listing below.

Listing B.1: App code view behind user interface buttons

```
  % Button pushed function: StartButton
  function StartButtonPushed(app, event)
    params = evalin('base','params');
    tg = params.tg;
    F_min = params.F_min;
    F_max = params.F_max;

    % set emergency button and click detection to 0
    tg.setparam('Emergency_constant','Value',0)
    tg.setparam('In_onclick/Click_detection','Value',0)
    % set minimal and maximal pull force to initial values
    tg.setparam('In_onclick/F_min','Value',F_min)
    tg.setparam('In_onclick/F_max','Value',F_max)
    delete(timerfind)

    date = datetime;

    tg.start;

    date.Format = 'yyyyMMdd_HHmmss';
    date_str = char(date);
    assignin('base','date_str',date_str)
  end

  % Button pushed function: StopButton
  function StopButtonPushed(app, event)
    params = evalin('base','params');
    tg = params.tg;
    tg.setparam('Emergency_constant','Value',1)

    % Make a directory for the last run and go there
    date_str = evalin('base','date_str');
    cd 'C:\Users\SID␣TUDelft\Documents\TU␣Delft\MScThesis\Matlab\BumpEmulation\LogData'
    mkdir(date_str)
    cd(date_str)

    % Copy .dat files from Speedgoat to PC
    f=SimulinkRealTime.openFTP(tg);
    mget(f,'F_001.dat');
    mget(f,'C_001.dat');
    close(f)

    % Open .dat files from Speedgoat
    fsys = SimulinkRealTime.fileSystem(tg);
    file_id_force = fopen(fsys, 'F_001.dat');
    file_id_current = fopen(fsys, 'C_001.dat');
    F_data = fread(fsys, file_id_force);
    C_data = fread(fsys, file_id_current);
    fclose(fsys,file_id_force);
    fclose(fsys,file_id_current);

    Forces = SimulinkRealTime.utils.getFileScopeData(F_data);
    Currents = SimulinkRealTime.utils.getFileScopeData(C_data);

    % Save opened files to PC
    name_str_forces = strcat('Forces_',date_str);
    name_str_currents = strcat('Currents_',date_str);
    assignin('base',name_str_forces,Forces)
    assignin('base',name_str_currents,Currents)
    save(name_str_forces,'Forces')
    save(name_str_currents,'Currents')

    % plot opened files
    figure
    subplot(2,2,1)
    hold on
    grid on
    plot(Forces.data(:,7),Forces.data(:,3), 'Color', [0.623, 0.698, 0.874])
    plot(Forces.data(:,7),Forces.data(:,5), 'Color', [0.666, 0.874, 0.623])
    plot(Forces.data(:,7),Forces.data(:,1), 'Color', [0.823, 0.254, 0.254])
    ylabel('CCW␣Force␣(N)')
    legend('measured␣1','measured␣3','desired␣13')
    ylim([0 100])
    title(date_str,'Interpreter','none')

    subplot(2,2,2)
    hold on
    grid on
    plot(Forces.data(:,7),Forces.data(:,4), 'Color', [0.623, 0.698, 0.874])
```

```
      plot(Forces.data(:,7),Forces.data(:,6), 'Color', [0.666, 0.874, 0.623])
      plot(Forces.data(:,7),Forces.data(:,2), 'Color', [0.823, 0.254, 0.254])
      ylabel('CW␣Force␣(N)')
      legend('measured␣2','measured␣4','desired␣24')
      ylim([0 100])

      subplot(2,2,3)
      hold on
      grid on
      plot(Currents.data(:,5),Currents.data(:,1), 'Color', [0.623, 0.698, 0.874])
      plot(Currents.data(:,5),Currents.data(:,3), 'Color', [0.666, 0.874, 0.623])
      xlabel('Time(s)')
      ylabel('CCW␣Currents␣(A)')
      legend('I␣to␣M1','I␣to␣M3')
      ylim([0 10])

      subplot(2,2,4)
      hold on
      grid on
      plot(Currents.data(:,5),Currents.data(:,2), 'Color', [0.623, 0.698, 0.874])
      plot(Currents.data(:,5),Currents.data(:,4), 'Color', [0.666, 0.874, 0.623])
      xlabel('Time␣(s)')
      ylabel('CW␣Currents␣(A)')
      legend('I␣to␣M2','I␣to␣M4')
      ylim([0 10])

      savefig(date_str)

      cd ../..
    end

    % Button pushed function: ClicktopullButton
    function ClicktopullButtonPushed(app, event)
      if isempty(timerfind('Tag','Click'))
        params = evalin('base','params');
        tg = params.tg;

        t_min = params.t_min;
        t_max = params.t_max;
        t_next = t_min + randi(t_max-t_min);

        Direction_sign = 2*randi([0 1])-1;

        t = timer('Tag','Click','StartDelay',t_min);
        t.StartFcn = @(~,~)tg.setparam('In_onclick/Click_detection','Value',1);
        t.TimerFcn = @(~,~)tg.setparam('In_onclick/Click_detection','Value',0);
        t.StopFcn = @(~,~)delete(t);

```

```
            tg.setparam('In_onclick/t_next','Value',t_next);
            tg.setparam('Direction_sign','Gain',Direction_sign);

            start(t)
        end
    end
end
```

B.2. Simulink model

The block diagrams that make up the Simulink Real-Time model are shown in the figures below. Figure B.1 shows the main block diagram and Figures B.2 to B.5 represent the subsystems.

The code shown in Listing B.2 is the initialization function that is executed at the start of the compilation of the model. Listing B.3 shows the content of the ClickToPullFcn block in Figure B.2.

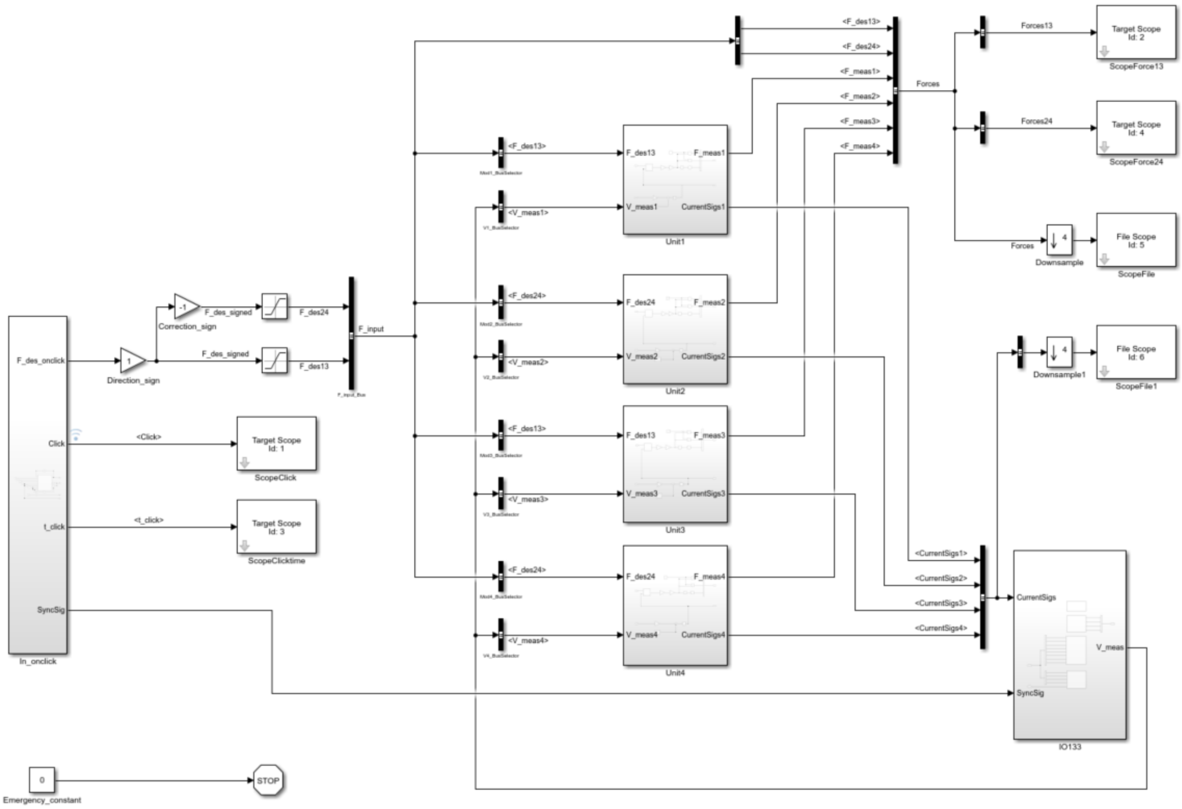


Figure B.1: Main scheme

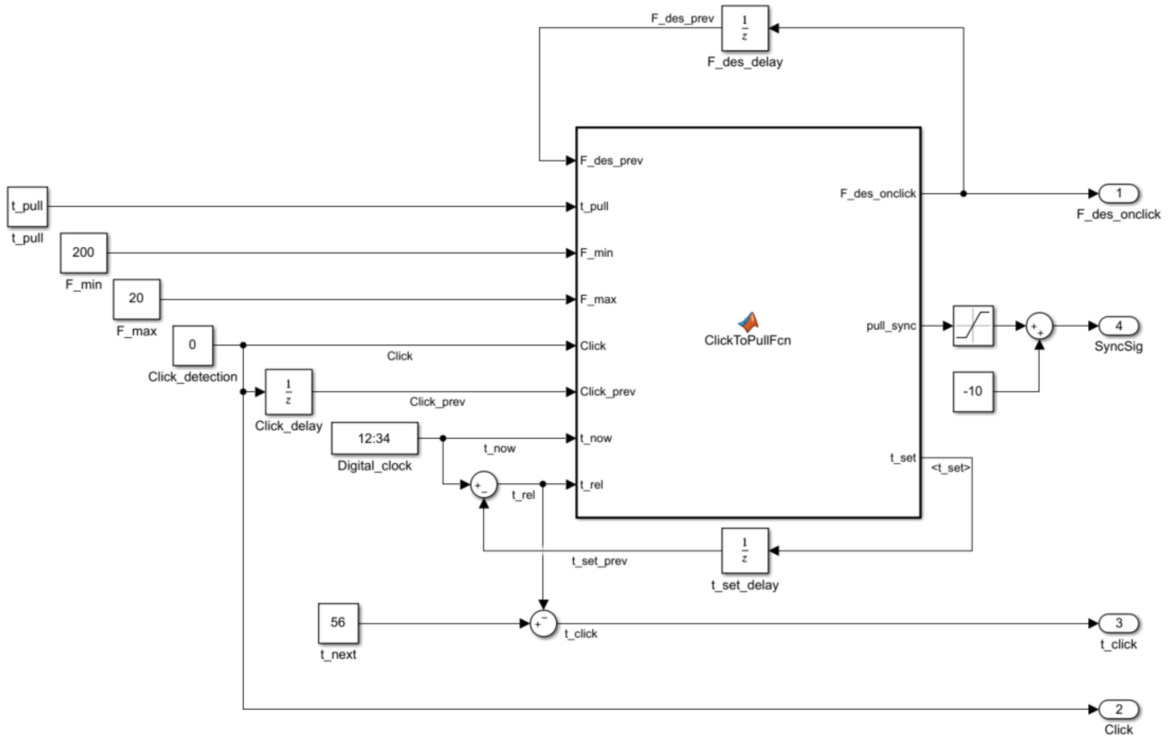


Figure B.2: In_onclick subsystem

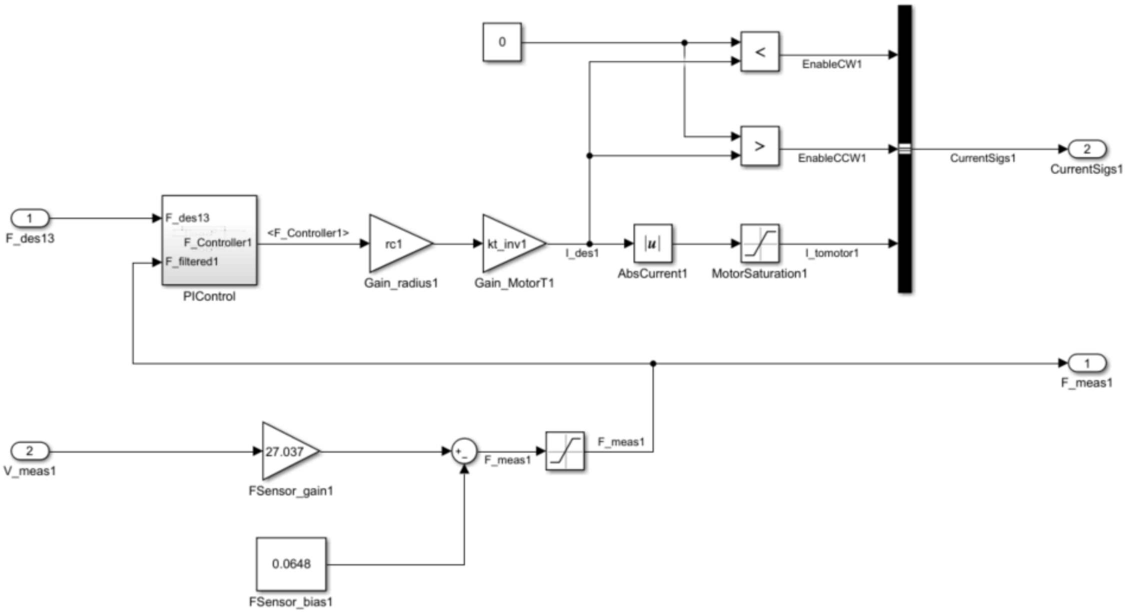


Figure B.3: Unit1 subsystem

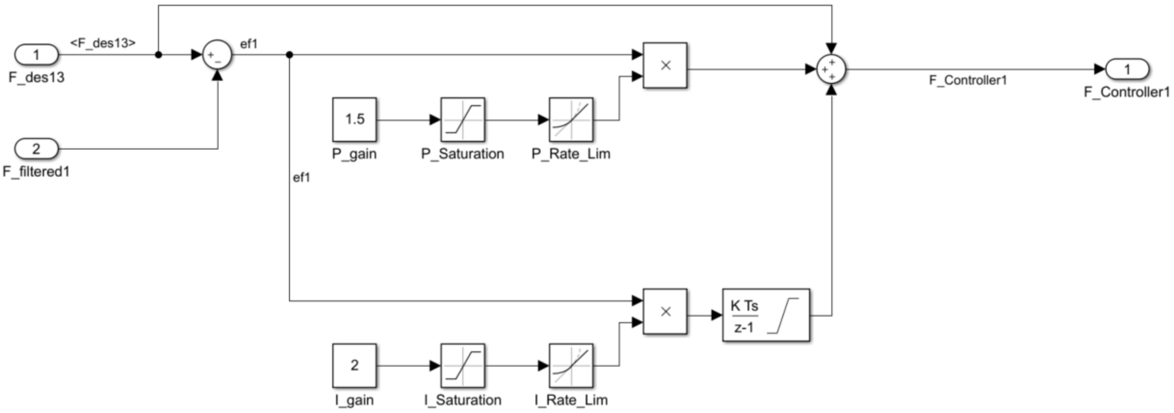


Figure B.4: PIControl subsystem

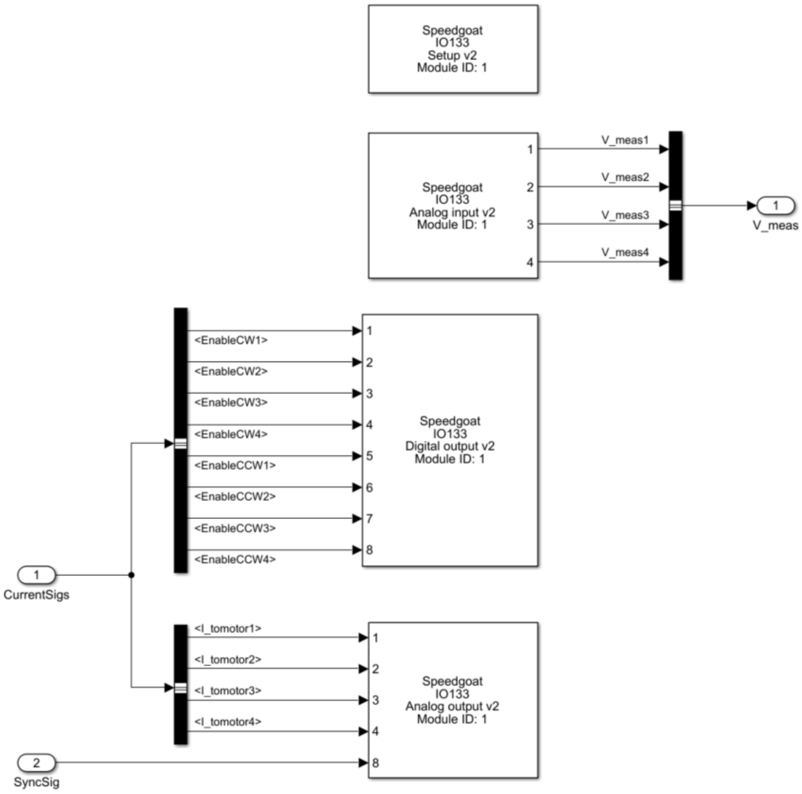


Figure B.5: IO133 subsystem

Listing B.2: Callback InitFcn

```
clear all;

%% Parameters

% target computer
tg = slrt;

% motor parameters
rc = 19.05e-3;          % real drum radius (m)
kt_nom = 0.231;         % nominal motor torque const (Nm/A)
kt_exp = 2.3*kt_nom;    % experimental motor torque const (Nm/A)
kt_inv = 1/kt_exp;      % inverse of experimental motor torque for gain (A/Nm)
I_stall = 56.9;         % stall current (A)

% for each motor (1,2,3,4)
% radii
rc1 = rc;
rc2 = rc;
rc3 = rc;
rc4 = rc;
% inverse motor currents
kt_inv1 = kt_inv*1.15;
kt_inv2 = kt_inv;
kt_inv3 = kt_inv;
kt_inv4 = kt_inv;
% saturation currents
I_sat1 = 10;
I_sat2 = 10;
I_sat3 = 10;
I_sat4 = 10;

% simulation parameters
t_total = 60;             % total simulation time (s)
t_samp = 0.001;           % sampling time (s)
F_max = 20;               % max pull force (N)
F_min = 200;              % min pull force (N)
F_tracking = 5;           % desired tracking force (N)
t_pull = 0.3;             % pull duration (s)
t_min = round(t_pull+3);  % minimum time between two subsequent pulls (s)
t_max = 30;               % maximum time between two subsequent pulls (s)

date = datetime;
date.Format = 'yyyyMMdd_HHmmSS';
date_str = char(date);

% parameter struct
params.tg = tg;
params.F_min = F_min;
params.F_max = F_max;
params.F_tracking = F_tracking;
params.t_min = t_min;
params.t_max = t_max;

%% initializing
```

```
% set emergency button to 0
set_param('TestForceSensor1234Motor1234/Emergency_constant','Value',
    num2str(0));

% set click detection to 0
set_param('TestForceSensor1234Motor1234/In_onclick/Click_detection','Value
    ',num2str(0));

% set min pull force to initial value
set_param('TestForceSensor1234Motor1234/In_onclick/F_min','Value',num2str(
    F_min));

% set max pull force to initial value
set_param('TestForceSensor1234Motor1234/In_onclick/F_max','Value',num2str(
    F_max));

% calculate a random t_next and set it
t_next = t_min + randi(t_max-t_min);
set_param('TestForceSensor1234Motor1234/In_onclick/t_next','Value',num2str
    (t_next));

% delete all timerfunctions
delete(timerfind);
```

Listing B.3: ClickToPullFcn

```
function [F_des_onclick,pull_sync,t_set] = ClickToPullFcn(F_des_prev,
    t_pull, F_min, F_max, Click, Click_prev, t_now, t_rel)

  % initializing t's as type double
  t1 = 0; t2 = 0; t_set = 0;

  % assiging values from digital clock
  t1 = t_now; t2 = t_rel;

  if Click > Click_prev
    if F_max > F_min
      F_des_onclick = F_min + randi(F_max-F_min);
    else
      F_des_onclick = F_max;
    end
    t_set = t1;
    pull_sync = F_des_onclick/10;
  elseif (Click > 0 ) && (t2<t_pull)
    F_des_onclick = F_des_prev;
    t_set = t1-t2;
    pull_sync = F_des_onclick/10;
  else
    F_des_onclick = 0;
    t_set = t1-t2;
    pull_sync = 0;
  end
end
```

C
Constant angle evaluation

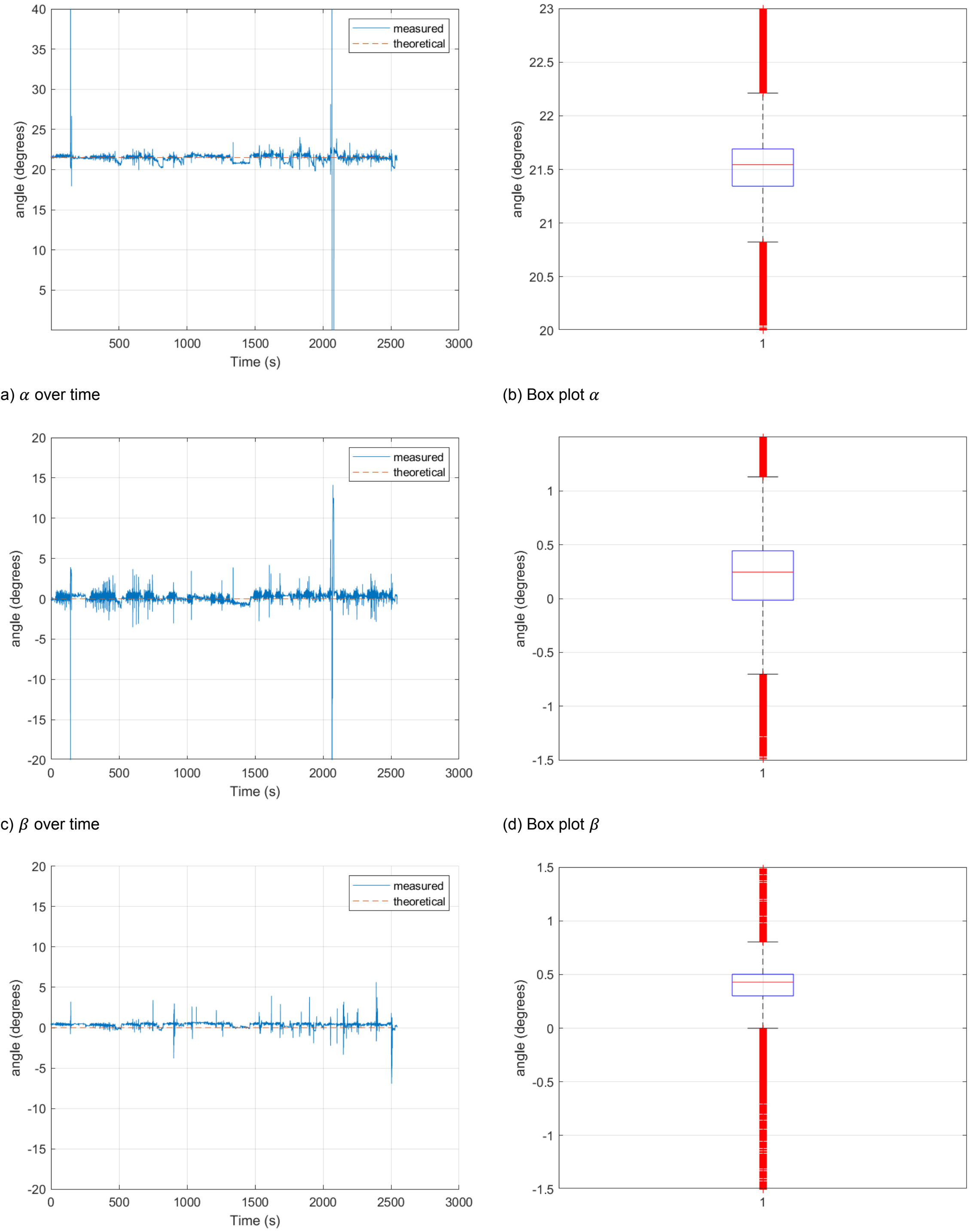
Angles that are part of the cans-in-series rotations from Figure 3.10, but are not of direct interest, can be assessed as sanity checks. These angles are the pitch of the rear frame χ, which is determined to be $0°$ by the wheel GCPs during cycling, and the front frame tilt α and its lean β, which are defined by the bicycle design to be $21.5°$ and $0°$ respectively.

This analysis is visualized with the help of Figure C.1. The plots on the left show the results for the angles over all experiment time concatenated, together with the theoretical value. The plots on the right show the box plots of this data. Table C.1 shows the according values for the means, standard deviations (STD), medians and interquartile ranges (IQR). It stands out that the mean and median values are very close to the theoretical values. The median for χ deviates most from it, with $0.4281°$. The standard deviations are all below $2.5°$ and that of χ is even an order of magnitude lower. The interquartile ranges are all below $0.5°$.

In Figure C.1 (a) and (b), two large deviations from the theoretical values are shown at the same times. Inspection of the video material at these instants reveals that markers had fallen to the ground here. This contributes to the higher standard deviations for α and β. The smaller, but more frequent peaks for χ are often the result of the subject positioning the bicycle back to the middle of the belt after a fall.

	α	β	χ
Mean	$21.3°$	$0.200°$	$0.363°$
STD	$2.41°$	$1.75°$	$0.348°$
Median	$21.5°$	$0.246°$	$0.428°$
IQR	$0.347°$	$0.459°$	$0.201°$

Table C.1: Statistic quantities for the front frame tilt α, the front frame bank β and the rear frame pitch χ.



(a) α over time

(b) Box plot α

(c) β over time

(d) Box plot β

(e) χ over time

(f) Box plot χ

Figure C.1: Analysis of front frame tilt α, front frame bank β and rear frame pitch χ, which are supposed to be constant over time. The left plots show the angles over time, together with the theoretical value. The right plots show the corresponding box plots. The red line in each box indicates the median and the box edges represent the first (Q1) and third (Q3) quartiles. The whiskers visualize Q1-1.5*IQR and Q3+1.5*IQR. Red plus signs beyond the whiskers represent outliers.

Bibliography

Amazon. (n.d.). SALVIMAR Dyneema 2mm, Unisex Adult. Retrieved March 3, 2022, from https://www.amazon.co.uk/SALVIMAR-Dyneema-2mm-Unisex-Adult/dp/B08824BP1G?th=1%7B%5C%%7D5C%7B%5C&%7Dpsc=1

Challis, J. H. (1994). A procedure for determining rigid body transformation parameters. *28*(6).

Dialynas, G., Happee, R., & Schwab, A. L. (2018). Design and implementation of a steer-by-wire bicycle. *International Cycling Safety Conference*.

Fietsersbond. (n.d.). Fietspaden. Retrieved March 1, 2022, from https://www.fietsersbond.nl/ons-werk/infrastructuur/fietspaden/

Kooijman, J. D. G., Schwab, A. L., & Moore, J. K. (2009). Some Observations on Human Control of a Bicycle. *Engineering Conference*.

Kurtzer, I. L. (2015). Long-latency reflexes account for limb biomechanics through several supraspinal pathways. *Frontiers in Integrative Neuroscience*, *8*(JAN), 1–19. https://doi.org/10.3389/fnint.2014.00099

Meijaard, J. P., Papadopoulos, J. M., Ruina, A., & Schwab, A. L. (2007). Linearized dynamics equations for the balance and steer of a bicycle: A benchmark and review. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *463*(2084), 1955–1982. https://doi.org/10.1098/rspa.2007.1857

Moore, J. K., Hubbard, M., Kooijman, J. D., & Schwab, A. L. (2009). A method for estimating physical properties of a combined bicycle and rider. *Proceedings of the ASME Design Engineering Technical Conference*, *4*(PARTS A, B AND C), 2011–2020. https://doi.org/10.1115/DETC2009-86947

Moore, J. K., Kooijman, J. D., Schwab, A. L., & Hubbard, M. (2011). Rider motion identification during normal bicycling by means of principal component analysis. *Multibody System Dynamics*, *25*(2), 225–244. https://doi.org/10.1007/s11044-010-9225-8

Moore, J. K. (2012). Human Control of a Bicycle. http://moorepants.github.com/dissertation

Plooij, M., Keller, U., Sterke, B., Komi, S., Vallery, H., & Von Zitzewitz, J. (2018). Design of RYSEN: An Intrinsically Safe and Low-Power Three-Dimensional Overground Body Weight Support. *IEEE Robotics and Automation Letters*, *3*(3), 2253–2260. https://doi.org/10.1109/LRA.2018.2812913

Reijne, M. M., Van der Helm, F. C., Boele-Vos, M. J., & Schwab, A. L. (2021). *Why Do Cyclists Fall ? A Newtonian-Mechanics based Set of Fall Mechanisms in Cycling for Accident Analysis and Prevention .*

Schepers, P., & Klein Wolt, K. (2012). Single-bicycle crash types and characteristics. *Cycling Research International*, *2*(October), 119–135.

Schwab, A. L., De Lange, P. D., Happee, R., & Moore, J. K. (2013). Rider control identification in bicycling using lateral force perturbation tests. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, *227*(4), 390–406. https://doi.org/10.1177/1464419313492317

Schwab, A. L., & Meijaard, J. P. (2013). A review on bicycle dynamics and rider control. *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, *51*(7), 1059–1090. https://www.tandfonline.com/action/journalInformation?journalCode=nvsd20

Schwab, A. L., Meijaard, J. P., & Kooijman, J. D. (2012). Lateral dynamics of a bicycle with a passive rider model: Stability and controllability. *Vehicle System Dynamics*, *50*(8), 1209–1224. https://doi.org/10.1080/00423114.2011.610898

SWOV. (2020a). *Ernstig verkeersgewonden in Nederland [Fact sheet]* (tech. rep.). SWOV. Den Haag. https://www.swov.nl/feiten-cijfers/factsheet/ernstig-verkeersgewonden-nederland

SWOV. (2020b). *Verkeersdoden in Nederland [Fact sheet]* (tech. rep.). SWOV. Den Haag. https://www.swov.nl/feiten-cijfers/factsheet/verkeersdoden-nederland

Tan, G. R., Raitor, M., & Collins, S. H. (2020). Bump'em: An Open-Source, Bump-Emulation System for Studying Human Balance and Gait. *Proceedings - IEEE International Conference on Robotics and Automation*, 9093–9099. https://doi.org/10.1109/ICRA40945.2020.9197105

Vallery, H., & Schwab, A. L. (2018). *Advanced Dynamics* (2nd ed.). Delft University of Technology.

Whipple, F. J. W. (1899). The stability of the motion of a bicycle. *The Quarterly Journal of Pure and Applied Mathematics*, *30*, 312–348.