Control of a Dynamic Driving Simulator: Time-Variant Motion Cueing Algorithms and Prepositioning

Diplomarbeit – Diploma Thesis

IFS 2006-0021

von

cand. kyb. Cornelius Weiß

Betreuer: Dipl.-Ing. Martin Fischer

Braunschweig, 30. November 2006

Deutsches Zentrum für Luft- und Raumfahrt e.V.

in der Helmholtz-Gemeinschaft

Institut für Verkehrsführung und Fahrzeugsteuerung

Prof. Dr.-Ing. Karsten Lemmer

Lilienthalplatz 7
38108 Braunschweig

Telefon: +49 (0)531 295 – 3401
Telefax: +49 (0)531 295 – 3402

E-Mail: ifs@dlr.de
Internet: www.dlr.de/fs

Versicherung

Hiermit bestätige ich, dass ich die vorliegende Diplomarbeit selbständig angefertigt und alle Hilfsmittel und Quellen angegeben habe.

Braunschweig, den 30. November 2006

Cornelius Weiß

Cornelius Weiß

Abstract

Die vorliegende Diplomarbeit beschäftigt sich mit dem Problem der realistischen Darstellung von Bewegungshinweisen (Motion Cues) in dynamischen Simulatoren. Der Fahrsimulator des Instituts für Verkehrsführung und Fahrzeugsteuerung am Deutschen Zentrum für Luft und Raumfahrt in Braunschweig nutzt ein Hexapod Bewegungssystem, um Beschleunigungen in allen sechs Freiheitsgraden darzustellen. Das Ziel dieser Arbeit ist es, den Fahreindruck des Probanden zu verbessern. Das Modul, welches die von einer Mehrkörpersimulation des Autos gelieferten Bewegungen in Signale umsetzt, die von der Plattform dargestellt werden können, soll sich zu diesem Zweck an die aktuelle Fahrsituation anpassen.

Zwei adaptive Lösungen wurden entwickelt, die zum einen zwischen statischen Motion Cueing Algorithmen wechseln und zum anderen zeitvariante Filterstrukturen nutzen. Eine der Fahrsituation angepaßte Vorpositionierung des Simulators verschiebt die Plattform innerhalb des Arbeitsraumes, um Bereiche, die laut prädiziertem Verhalten vorwiegend genutzt werden, zu vergrößern. Auf diese Weise ist es möglich, die Kapazität der Bewegungsplattform hinsichtlich Arbeitsraum, Geschwindigkeit des Verfahrens und Darstellung der Beschleunigungen voll auszuschöpfen.

Abstract

This thesis mainly addresses the task to provide the driver of a dynamic driving simulator with the same motion cues as would occur in reality. The DLR Driving Simulator, which is the subject matter of this thesis, uses a hexapod motion platform to generate these specific forces and accelerations. The ambition of this thesis is to enhance the realism of the simulation. This is achieved by improving the system that generates the motion commands for the simulator platform based on the simulated vehicle's motion, the *motion cueing algorithm*.

Two adaptive motion cueing algorithms were established, that switch between different static algorithms and use filters with time-varying coefficients. A prepositioning module enlarges the available motion envelope in the direction of likely motions. The current and a predicted driving situation are used to determine the adaption of the motion cueing algorithm and the prepositioning motion of the platform to exploid the capabilities of the simulation platform to a higher degree.

Contents

List of Figures

List of Tables

Denotations/Abreviations

- ADAS: advanced driving assistance system
- MCA: motion cueing algorithm
- MDA: motion drive algorithm
- WA: washout algorithm
- PU: prepositioning unit
- DS-Analyser: *d*riving *s*ituation analyser
- HMI: human-machine-interface
- Linear motions: surge,sway,heave
- Angular motions: yaw, pitch, roll
- Acceleration: 2nd time derivative of displacement, 1st time derivative of velocity
- Specific forces: external (non-gravity) forces acting on a body per unit of mass of that body; the overall acceleration acting on a body is the specific force added to the acceleration due to gravity
- dof: degree-of-freedom
- σ: the switching signal
- a_{real},ω_{real}: inputs to the motion cueing algorithm
- a_{sim},ω_{sim}: outputs of the filtering structure in motion cueing algorithm, both signals are integrated to yield the positions s and angles b that are simulated with the platform

Chapter 1

Introduction

1.1 Driving Simulators

New advanced driving assistance systems (ADAS) are developed to increase the comfort and safety of future cars. To ensure that they are accepted by the driver and that they have the desired effects on his behaviour, they need to be tested. These tests often cannot be done in real cars and traffic for the new system hasn't reached a sufficient maturity. A reasonable work-around are driving tests in simulators within a virtual reality.

The assessment of a driver regarding his behaviour is coupled with different problems. The vehicle is part of the human-machine-control-loop, with the driver taking the role of a highly non-linear and adaptive controller. Objective measurements imply different problems [Ric71]:

- Measurements with real vehicles are not always comparable, since there are disturbances like changing traffic or environmental conditions.
- The conditions during the measurements have to be reproduceable, to repeat the tests and compensate for learning effects. This includes the driving environment, for instance in terms of other vehicles.

Driving Simulators are used to generate a driving environment without facing problems like risks that are associated to drives. They provide controllable and measurable conditions for advanced driving experiments, which can be repeated at a low cost. Thus, a driving simulator can be a good tool to examine a driver's behaviour. Significant benefits are:

- The experimental conditions are determined and reproducable.
- The driving and vehicle parameters are easy to adjust.
- Significant values (e.g. velocity, steering wheel angle) are easy to measure.
- The influences of assistance systems or natural limitations like fatigue, alcohol and drugs can be examined without the risk of accidents.

The assesment of new adaptive cruise control algorithms and interfaces involved in these driving tasks is facilitated.

Driving simulators use different tools to provide the driver with a realistic simulation. The visual system (display) gives the main cues for the orientation of the driver in the simulated environment producing a sensation that is analogous to the human visual perception in real driving situations. Besides this system, modern driving simulators use motion platforms that render the movements of the simulated vehicle. Most dynamic driving simulators consists essentially of six hydraulical linear actuators that are mounted in a hexapod configuration (see figure 1.1 and section 1.2 for a description of the system used at the DLR institute for transportation systems). The objective is a driving sensation via motion cues close to that in a real drive, despite the limited capabilities of the motion platform. A description of the human motion perception is given in section 1.5.

Despite the advantages given by a simulation, the lack of a completly realistic driving simulation is still a drawback. The results of tests with simulators are

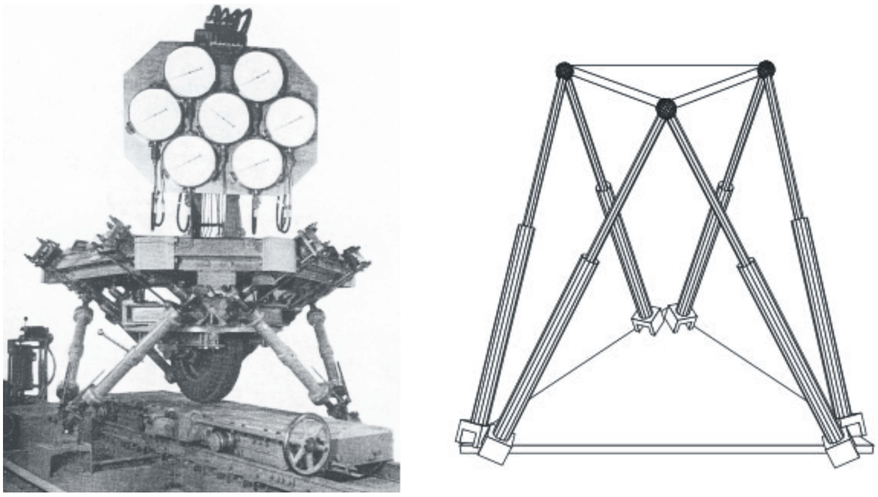


Figure 1.1: The figure shows the first Gough platform (1947) and a schematic depection of the underlying kinematics [Mer99].

only comparable to real world tests when both the vehicle and the driver react similar. Maintaining a realistic trajectory in the simulated environment with the vehicle is very difficult. Visual and kinesthetic cues that are perceived in a real test drive have to be presented to the driver. Especially lateral accelerations impart a great amount of information necessary to steer a vehicle. These motion cues aren't easy to render with a driving simulator, for it has only limited capabilities concerning the presentation of displacement, velocity and acceleration. This task is further complicated by the human's highly nonlinear perception of motions.

Due to constraints on the motion envelope and the actuator motions[1], the commonly used hexapod is not able to follow a real car's trajectory. Thus, it has to be fed with motions that do not exceed these limitations but provide a good replication of the original movement.

Since driving generally involves movements that exceed the platform's capabilities regarding displacement, velocity and acceleration, the motions created by the underlying simulation of the driving dynamics have to be modified. The modification is achieved through scaling and filtering of the platform's inputs in washout algorithms[2]. They serve as the link between the vehicle dynamics provided by the simulation and the coordinated movement of the platform through actuator displacement (usually six degrees of freedom, a good overview of other approaches can be found in [PGN98]). Certain frequencies of the motion signal are dropped and the whole drive is often scaled down. The rseulting movement commands do not cause an excitation of the actuator limits regarding position, velocity and acceleration. The aim of a motion cueing algorithm is thus to reproduce the motion cues with the highest possible fidelity, given the constraints

[1] Often, the actuators are limited regarding the control power and are subject to a low-pass response characteristic.

[2] These washout algorithms (WA) are also called motion cueing algorithms (MCA) or motion drive algorithms (MDA).

of the particular system.

1.2 Research at the Institute of Transportation Systems

Research at the Institute of Transportation Systems at the German Aerospace Center in Braunschweig focuses inter alia on the behaviour of a vehicle's driver and his interaction with existing and future advanced driver assistance systems (ADAS).

Figure 1.2: The DLR Virtual reality Lab [Sui05].

Necessary Experiments are performed in a driver assistance lab, for tests in the real environment incorporate many problems. The driver assistance lab[3] consits of a virtual reality lab (a cave with a simple cockpit mockup, see figure 1.2), a measurement vehicle[4] and a large motion based simulator (section 1.2.1). These are used with the development and evaluation of new active and passive assistance functions and the examination of human-machine-interfaces (HMI). The new functions and HMI concepts are assessed using three stages of evaluation, being the virtual reality lab, the large driving simulator with a very high degree of realism and the evaluation in real traffic. Especially the dynamic driving simulator facilitates the assesment of driving systems, since it provides a realistic driving environment. The immersion of the driver into the simula-

[3]A more detailed introduction to the work at the Institute of Transportation Systems, German Center for Aerospace, can be found in [Sui05].

[4]A real car used for the analysis of real traffic drives. It measures the vehicle's physical states, the driver's behaviour (e.g. gaze movements, the lane keeping ability) and the environmental conditions.

tion is crucial, for it is the basis of studies that yield valid results. Therefore, the driver's actions in the simulator should be as similar as possible to those observed in real traffic situations.

1.2.1 The DLR Driving Simulator

Figure 1.3: The DLR Driving Simulator.

The motion platform installed at the German Aerospace Center in Braunschweig resembles a Steward/Gough platform with six degrees-of-freedom [Sui05, BK05]. It extends the local driver assistance lab at the Institute of Transportation Systems. One of the major differences towards other platforms is the inverted cabin, hanging down from the platform (see figure 1.3). The result is an improved usability regarding the presentation of low-frequency translations with tilt-coordination (see also section 2.1), for the centroid lies below the actuator hinges. The flexibility of the system regarding the integration of new ADAS and a good fit into the existing lab environment were main demands that were met.

A high degree of realism is reached with a good presentation of motion cues. Hence, small delays, large/realistic amplitudes and a high bandwith of the cues are essential for a high-fidelity motion. The motion capabilities are given in table 1.1. Benefits are also obtained by use of the particular visual system and a vehicle that appears and acts like a real one. The incorporation of a real vehicle is important since it is the part of the simulation that is closest to the driver. All instruments can be controlled as expected by the driver. The steering wheel for instance, is equipped with a motor to render force feedback and thereby enhance the controllability of the simulated vehicle. A good visual system is also crucial for the immersion of the driver into the simulation. A large field

Constraints/Capabilities of the Simulator			
	Travel	Velocity	Acceleration
Surge	$\pm 1,5m$	$\pm 2m/s$	$\pm 10m/s^2$
Sway	$\pm 1,4m$	$\pm 2m/s$	$\pm 10m/s^2$
Heave	$\pm 1,4m$	$\pm 2m/s$	$\pm 10m/s^2$
Roll	$-20^\circ \; +21^\circ$	$\pm 50^\circ/s$	$\pm 250^\circ/s^2$
Pitch	$\pm 21^\circ$	$\pm 50^\circ/s$	$\pm 250^\circ/s^2$
Yaw	$\pm 21^\circ$	$\pm 50^\circ/s$	$\pm 250^\circ/s^2$

Table 1.1: Specifications of the DLR Dynamic Driving Simulator [DZfLuR].

Figure 1.4: Interior of the DLR Driving Simulator [Sui05].

of view with a sufficient resolution provides for the visual cues together with displays in the side-mirrors and a large display mounted on the backseats that can be seen in the rear mirror.

The software used to simulate a driving environment is formed by different modules (see figure 1.5). A traffic and world simulation establishes a virtual reality that interacts with the cabin interfaces (e.g. steering wheel, displays), a sound generator, the image generator and the part that simulates the vehicle dynamics (Carsim). The operator control inputs and the signals from the cabin interfaces drive a mathematical model of the vehicle dynamics that generates the vehicle's linear and rotational movements. The modulated motion cues are produced by passing the vehicle states through the motion cueing algorithm (emphasized in figure 1.6) to fit the simulator's constraints and capabilities. The reference simulator platform states are subsequently transformated from spatial coordinates to the space of the six actuators. The motion cueing algorithm is the subunit of the simulator's control to be enhanced within the work of this thesis.

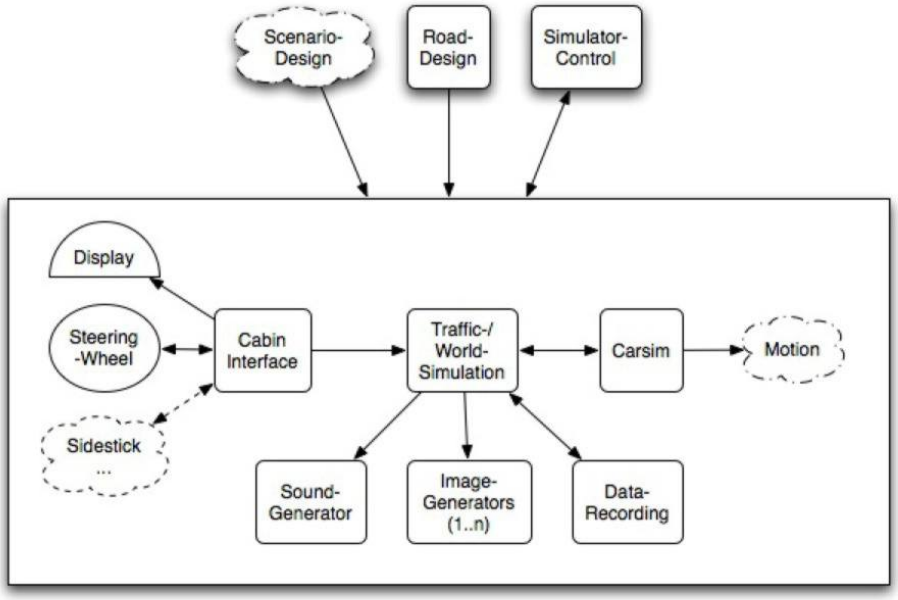


Figure 1.5: Software Modules used with the Driving Simulator [Sui05].

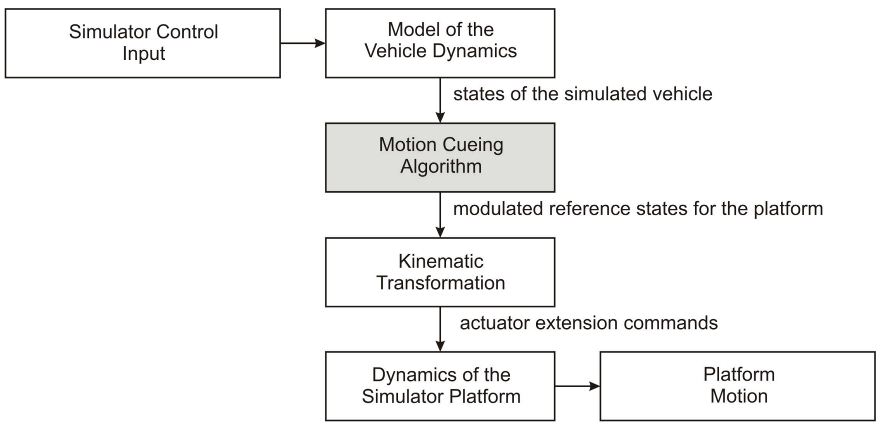


Figure 1.6: Structure of the signal flow in the motion system of driving simulators. The module that is considered in this thesis – the motion cueing algorithm – is highlighted.

The fundamental problem when providing a realistic motion to the driver is the proper control of the platform. Although the common driving simulator design is often a direct derivation from flight simulators, the design of the motion control is more difficult. The simulator at the Institute of Transportation Systems is initially driven by a washout-algorithm similar to that of flight simulators.

1.3 Validity of Driving Simulations

The major problem when conducting driving tests in simulators is the transfer of the results obtained in simulator studies towards real driving situations. The validity of the driver's behaviour during the simulated test-drive has to be shown. This is facilitated with good visual and kinaesthetic cues[5]. These found the basis for a realistic driving sensation that ensures the immersion of the driver into the simulation. Vice versa, a dynamic driving simulator with a poor tuning produces artificial motions not matching the visual cues and the original behaviour of the car. These false cues are the cause for an unrealistic driving behaviour and simulator sickness, effects worse than that of a simulation with no motion cues at all.

The results of simulated drives need to be validated. This is achieved when the physical, perceptual, relative and absolute behavioral validity of the tests are good (see also [RK00]):

- Physical validity: Regards the similarities between the rendered motions and the real vehicle's movements.
- Perceptual validity: Comparing the driver's perception in the simulator with that in a real car.
- Relative and absolute behavioral validity: Considers similarities in the actions and reactions of a driver in the simulator and one in a real vehicle.

Since the movements of the simulator are limited and a direct rendering of the complete simulated car's manoeuvres is not possible in general, a good behavioral validity is most important, and ultimately facilitated by a good perceptual one. The driver's actions in the real world and in simulations are only similar when the driver perceives just a small or no difference between the real and the simulated drive. Thus, it is sometimes suggestive to neglect the physical validity of the motion and allow for differences in the movements. An example for such artificial movements is the tilt-coordination, which uses the imperfections of the human motion perception to enrich the simulation's perceptual validity (see also sections 1.5 and 2.1).

1.3.1 Test Environment

The validity of the concepts developed in this thesis are assessed within a test environment [Fis05]. New motion cueing algorithms are inserted into the corresponding Matlab/Simulink diagram (see the third branch of figure 1.7). The signal flow can be characterized as follows:

1. Inputs to the simulink plan are either real accelerations measured with a test car[6] or artificial reference *accelerations* (e.g. sinusoidal or step-like accelerations) in case of the translational degrees of freedom. Analogously, angular *velocities* are used as inputs for the rotational degrees of freedom.

[5] Kinaesthetic cues are also called motion cues.

[6] Data of the local measurement vehicle (ViewCar) can be used.

2. The reference values are passed onto the motion cueing algorithm that is to be assessed, after they are transformed into a suitable coordinate system (the *Test MCA* block in figure 1.7).
3. The motion cueing algorithm produces reference positions and angles that are transferred to a model of the simulator platform.
4. The resulting movement of the simulated platform is passed to a model of the human motion perception system, the vestibular system.
5. The perceived motion is exported among other values.

Since the environment also contains a parallel path without the models of the MCA and the simulator and a branch with a *reference* MCA, it is possible to compare the motions perceived in the simulator with those perceived in a real drive and those perceived when using the reference MCA. This way, it is possible to tune new motion cueing concepts towards a good perceptual validity.

1.4 Motions Occuring During Driving Situations

To asses the movements that are to be simulated by motion platforms, Reymond and Kemeny recorded typical driving situations with a test car [RK00]. The

Degree of freedom	Acceleration limits	Vehicle motion
Surge	-0.6 $\rightarrow$ 0.4 g	braking, accelerating
Sway	-0.7 $\rightarrow$ 0.7 g	cornering
Heave	-0.8 $\rightarrow$ 1.1 g	suspension and road elev.
Roll	$\pm 320°/s^2$	suspension and cornering
Pitch	$\pm 360°/s^2$	suspension and cornering
Yaw	$\pm 45°/s^2$	steering

Table 1.2: The table shows the maximum accelerations Reymond and Kemeny obtained by analysing real driving situations [RK00].

analysis of the spectral distribution of the accelerations obtained in test drives (given in table 1.2) provides a good possibility to determine if the particular simulator is able to render the desired motion cues. Reymond and Kemeny state the following results:

- The spectral power density of the linear accelerations that were measured has a sharp attenuation above 1 Hz.
- The angular accelerations have a predominat high frequency content that reflects vibrations, only the yaw acceleration has a low frequency part that results from the road curvature. Hence, the natural roll and pitch parts of the motion may be rendered directly within a reasonable bandwidth.

The high frequency parts of the motion are important for a better estimation of vehicle speed and road conditions. A comparison of the capabilities of the DLR

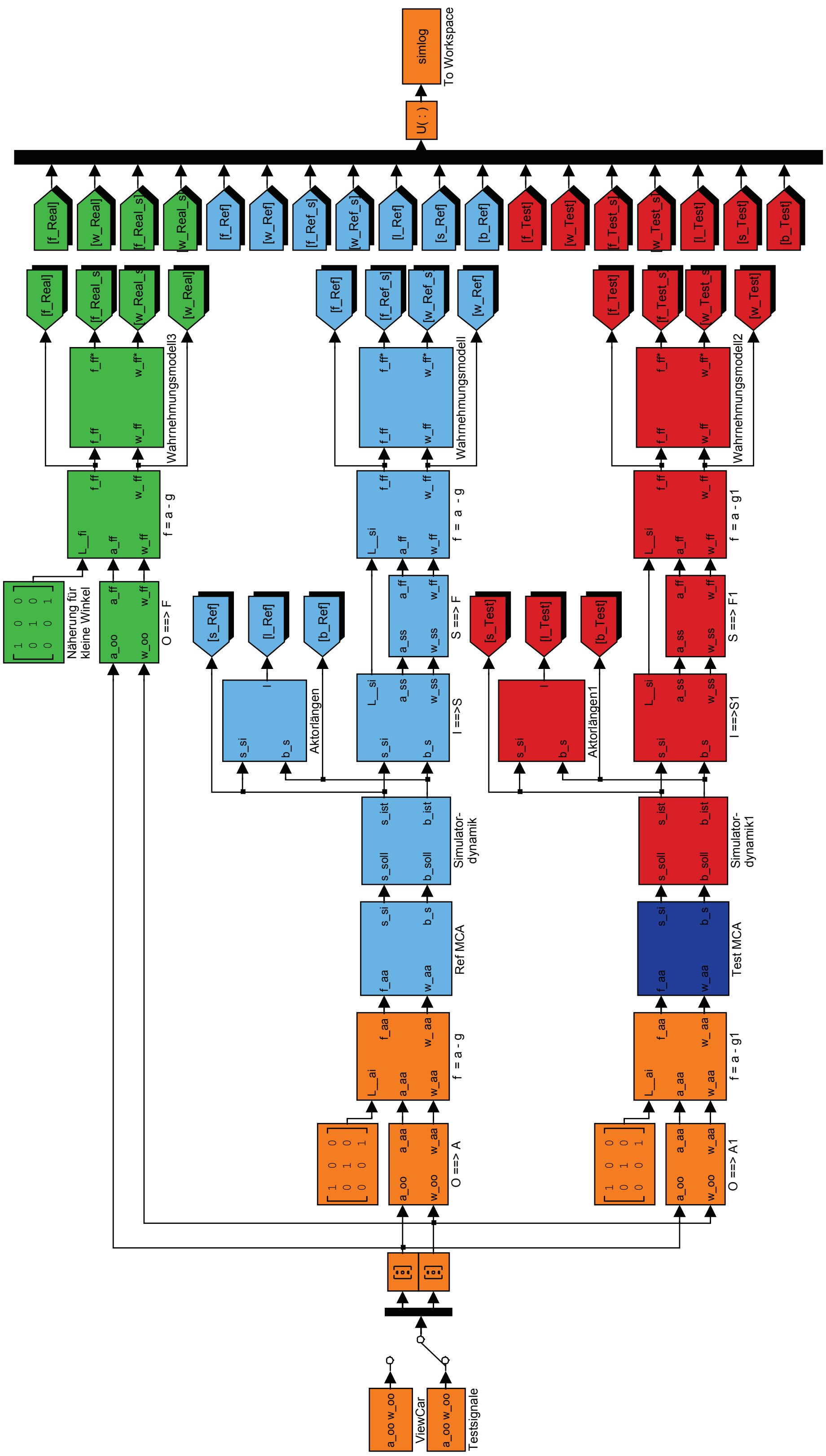


Figure 1.7: The test environment implemented in Matlab/Simulink that was used to assess the devolped algorithms. A more detailed description can be found in [Fis05].

driving simulator (table 1.1) with the maximum accelerations in real driving situations (table 1.2) yields a good correlation. As motion cueing algorithms are tuned towards holistic worst-case situations[7], a down-scaling of the movements is suggestive. Reymond and Kemeny propose to render the high frequency parts of the accelerations with a magnetic seat actuator, for typical motion platforms have a low-freqency response behaviour.

1.5 Motion Perception

The simulation's realism depends strongly on the capabilities of the motion platform and the perception of motion by the human driver. As the perceptual validity of a driving simulation is very important and might be increased making use of the nonlinearities in the perception of the motion, some notes on the capabilities of the human perception system are required.

Together with visual cues, tactile sensors in the skin and the inner organs, the vestibular system (see figure 1.8) is responsible for the perception of motions. It measures transient displacements of the human head. The organ consists of the otoliths being responsible for the sensing of translational accelerations and the semicircular system that senses rotations. Both are located in the inner ear. Motion onsets are also detected by the Paccini tactile receptors located in the skin.

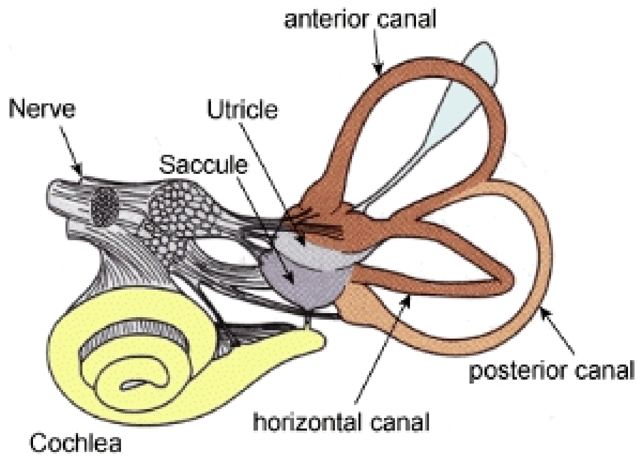


Figure 1.8: The vestibular system of humans. The three canals form the semicircular system whereas the otoliths are the Saccule and the Utricle, respectively. The picture was taken from [Enc].

Motions are only perceived as long as they exceed thresholds. These are not static and depend on the duration of the motion[8] and the over-all amount of signals perceived by the human [Ric71]. Most driver's are not likely to perceive

[7]This is the major drawback in the tuning process of simulators, as long as they are composed of static structures!

[8]The longer the signal, the lower the threshold.

actively a little bumb on the road when a car is approaching from straight ahead and a collision is to be expected. Reymond and Kemeny argue that the detection threshold for linear movements is $5cm/s^2$, the one for angular acc. is $0,3\deg/s^2$, respectively [RK00]. These values are to be taken as rough estimates for values that obey a dynamic behaviour.

Rendering transient vehicle dynamics is the primary requirement to achieve a satisfying level of perceptual validity. Low-frequency and sustained lateral accelerations cannot always be rendered directly with ordinary driving simulators, a great part of the motion sensation (e.g. in long curves) would be missing without a reasonable work-around. There exist two basic elements to modulate the vehicle's original linear accelerations to match the driving simulator's capabilities, being the tilt-coordination and the down-scaling of motions. Both elements have to be combined to yield a realistic simulation of standard driving situations on standard driving simulators:

- Scaling the amplitude is a method to make use of the non-linearities in human motion perception, for humans lack a good ability to put ongoing accelerations into context with former ones. The amplitude might be reduced down to 25% of it's original value [Ric71].
- Another method is commonly called *tilt coordination*. Since the human perception system isn't capable of detecting motions below certain thresholds, it is possible to vary the gravitational vector with respect to the human body to display long sustained accelerations without perceiving this rotational movement. The platform tilt should be done around a center of rotation close to the driver's head whereas the natural rotations should be rendered around the platforms centroid. A distortion of the driver's subjective vertical may only occur while using tilt-coordination-angles above $20-30°$. This phenomenon is called the Aubert effect. Thus, sustained accelerations simulated via tilt coordination should not exceed $0.5m/s^2$ (see also [RK00]). A rate limiting of the motion is often used to ensure the compliance with the corresponding thresholds.

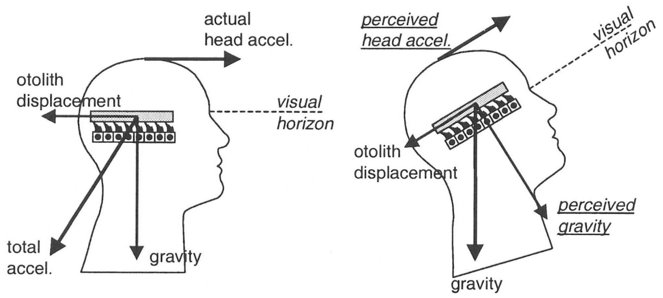


Figure 1.9: The Principle of Tilt Coordination for longitudinal accelerations. The left head needs to be moved horizontally to perceive an acceleration, whereas the right head might stay in its tilted position as long as the visual cues are appropriate.

A more important criterion for the realism of a driving simulation than the physical validity (e.g. the amplitude, see 1.3) is the delay/phase-shift of motion cues. A phase-lag of more than 18° is likely to be regognized [Ric71] and the discrepancy between between visual cues and motion perception might cause simulator sickness. Presenting the mid-frequency components of the signals to the driver is still a problem.

Chapter 2

Motion Cueing Algorithms

2.1 The Classical Approach

Driving simulators use different methods to provide the driver with a realistic simulation. The visual system gives the main cues for the orientation of the driver in the virtual environment analogously to a human's visual perception in real driving situations. Besides this system, modern driving simulators use motion systems that provide the driver with cues that are sensed by the vestibular, tactile and proprioceptive systems. Since driving generally involves movements that exceed most platform's capabilities regarding displacement, velocity and acceleration, the movements created by the underlying vehicle/driving simulation have to be modified. Hence, the aim is to reproduce the motion cues with the highest possible fidelity, given the constraints of the particular system.

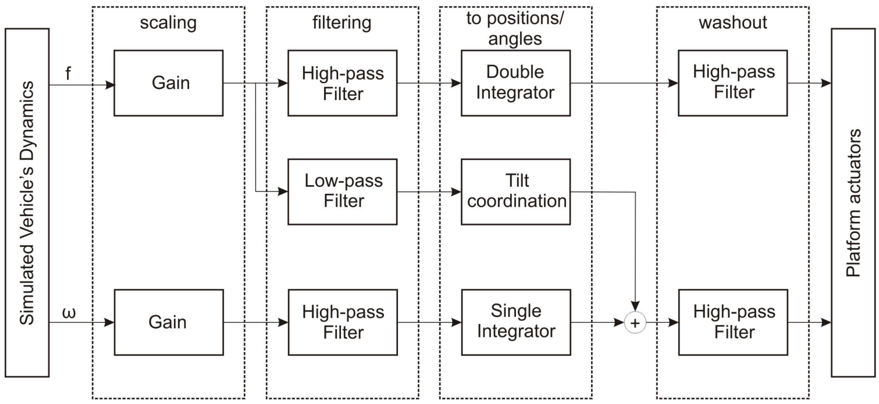


Figure 2.1: The Common Scheme for Motion Cueing Algorithms, f denotes the real linear accelerations a_{real} whereas ω denotes the angular velocities ω_{real}, calculated by the vehicle simulation. Outputs of the high-passes in the filtering module are the linear accelerations and angular velocities to be simulated $(a_{sim}\&\omega_{sim})$.

The motion cueing algorithm[1] is the part of the simulator's software that modulates the movement produced by the vehicle simulation to compensate for the simulator's limited capabilities. Although the demands of a driving simulation are different, derivatives of the software implemented in flight simulators are often used. These consists essentially of the following parts (see also figure 2.1):

1. The *scaling* block. It is used to reduce the amplitude of the motion signals, for humans are not able to differ between the real forces affecting them while driving and the slightly reduced ones presented in the simulator, as long as the difference is not to large. This way, the effects of the limitations in the simulator capabilities are attenuated, real movements with a larger amplitudes can be presented to the driver.

2. The *filtering* block is the most complex and important unit. The classical concept is a combination of different linear filter-strategies used to extract parts of the car's accelerations produced by a vehicle simulation.

[1] Also known as motion drive algorithm or washout algorithm.

These filter structures limit the directly rendered movements to the high-frequencies. Low-frequency lateral and longitudinal translations are extracted in a low-pass filter and represented by a tilt of the simulator. This mechanism is known as tilt coordination.

3. The *to positions/angles* block transforms the modified accelerations and velocities in position commands and euler angles via a single or double integration. The low-frequency components of the linear motions are not integrated but transformed into suitable angular velocities.

4. The *washout* unit. This block ensures that the platform returns to the neutral position when motions are finished.

The next sections describe the filtering unit and the washout block of the classical motion cueing algorithm in more detail.

2.1.1 Filtering

The classical motion cueing algorithm is the basic solution widely used in dynamic simulators. It consists of empirically determined high- and low-pass filters whose parameters are adjusted off-line in advance. Reid and Nahon treated the development and tuning of the algorithm in several technical reports [RN85, RN86]. This motion cueing algorithm uses the accelerations and angular velocities affecting the simulated vehicle as inputs. As seen in figure 2.2, the block-diagram consists of three paths:

1. The first path calculates the translational accelerations to be simulated from the accelerations calculated by the vehicle simulation. Especially the limitations regarding the motion envelope of the simulator impose restrictions on the direct rendering of translational accelerations. To avoid saturating the actuators, the original acceleration a_{real} has to be modulated. Sustained parts cannot be presented directly to the driver this way. Thus, applying only this method is not suitable for driving tests with a road curvature producing low-frequency accelerations. This is for instance the case in long turns, when large sustained lateral accelerations affect the driver. The original acceleration is high-pass filtered and the reference position is formed by a double integration of the acceleration a_{sim}.

2. The longitudinal and lateral specific forces are also low-pass filtered and rate limited to yield roll and pitch tilt angles. The purpose of this artificial tilt movement is to orient the driver relative to the gravity vector in a way that covers the low-frequency specific forces of the simulated vehicle that cannot be rendered directly. Hence, this mechanism allows for sustained longitudinal and lateral accelerations to be simulated. The force acting along the z-axis of the driver – the sensed gravity – is reduced, but at a not recognizable level as long as certain thresholds are maintained (see section 1.5). The tilt-coordination movement has to be limited in its velocity, to guarantee that only the desired horizontal accelerations are perceived.

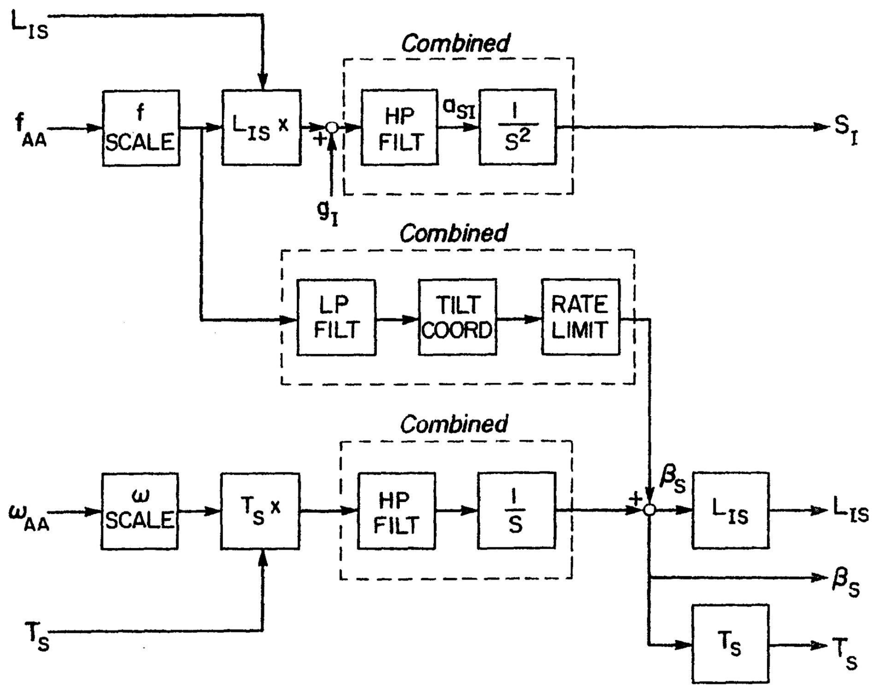


Figure 2.2: The classical motion cueing concept. The integration towards positions and angles is included in this representation (denoted by the dashed *Combined* rectangulars) [NR90].

3. The angular motions of the vehicle simulation are also high-pass filtered. This third path yields the angular accelerations that can be rendered with the simulators capabilities. The slow rotary movements of the simulated vehicle cannot be presented to the driver. This is not crucial since the roll and pitch movements are predominantly of high frequency nature. An illustrative example are long bends, that produce a slow yaw movement and a sustained lateral acceleration. The later is more important for a good controllability of the simulated vehicle.

The values produced by the second and third path are summerized to form the reference angles for the simulator. The next section gives a more detailed description of the tilt-coordination mechanism.

2.1.2 Tilt-Coordination

Tilt coordination might be used to render low-frequency/sustained accelerations in both the longitudinal and lateral direction. These components cannot be rendered directly. Hence, a great part of the motion sensation — for instance in long curves — would be missing. Since the human perception system isn't capable of detecting motions below certain thresholds, it is possible to tilt the human body with respect to the gravitational vector to display long sustained accelerations without perceiving this rotational movement. The resulting gravitational vector might then be splitted into ortohogonal forces, being the force along the vertical body axis that is still perceived as gravitation and a force that represents horizontal accelerations with respect to the driver. Fooling the driver is easy when the center of tilt is located near the vestibular system. It is resonable to place the center of tilt above the head. A position below the motion perception system results in wrong translational accelerations (false cues) when the platform is shifted from one stationary tilt-angle to another. The main problem when applying the mechanism is, that a significant value for the artificial acceleration is only formed after some time. The tilt coordination cannot be used to render realistic horizontal high frequency accelerations, for these would exceed the thresholds for the perception of rotational velocities. Thus, a simulation that uses only tilt-coordination to render the translational accelerations is not suggestive, for the large lag of the motion cues would destroy the realism of the simulation [Ric71].

The artificial platform tilt should be done around a center of rotation close to the driver's head whereas the real rotations should be rendered around the platforms centroid. A distortion of the subjective vertical may only occur while using tilt-coordination-angles above $20 - 30\,\mathrm{deg}$[2]. Reymond and Kemeny state that sustained accelerations simulated via tilt-coordination should not exceed $0.5g$. The mechanism should not be used for excessive braking, for the transport delay might create a false cue via a perceived lag. The maximum acceptable transport delay is located between 20 and 50 milliseconds. However, the tilt-coordination is especially helpful while driving long curves [RK00].

The mechanism of tilt-coordination shows good results for stationary and low-frequency accelerations, whereas the direct presentation of horizontal accelerations suits only for instationary and high-frequency components. To obtain a good realism for most of the motions to be rendered, both mechanisms are superposed. The advantages of both methods can be used while avoiding their drawbacks through a mutual compensation of their disadvantages. One problem that still remains is that the simulation with the coupled subsystems covers not the whole range of frequencies contained in the original signal. Depending on the parameters, mid-frequency signals[3] might not be rendered precisely, for the tilt-coordination needs time to build up whereas the translational accelerations decrease fast to avoid saturating the actuators. For a step-like input that should be presented to the driver, a depression in the perceived acceleration occurs after some time. Figure 2.3 plot 4 shows a simulation of the simulated acceleration (solid line) with a depression after $t = 2.5$ seconds. The time constants of the underlying filters are designed to yield a good balance between maintaining the tresholds for the perception of angular motion — this requires long time constants — and the introduction of transport delays by slow filter dynamics. To compensate for the delay, a reasonable pre-tilting of the cabin could be introduced that maintains the perception thresholds.

[2] This is called the Aubert effect.

[3]

- High frequency signals: initial cues
- Mid frequency siganls: transitional cues
- Low frequency signals: sustained cues

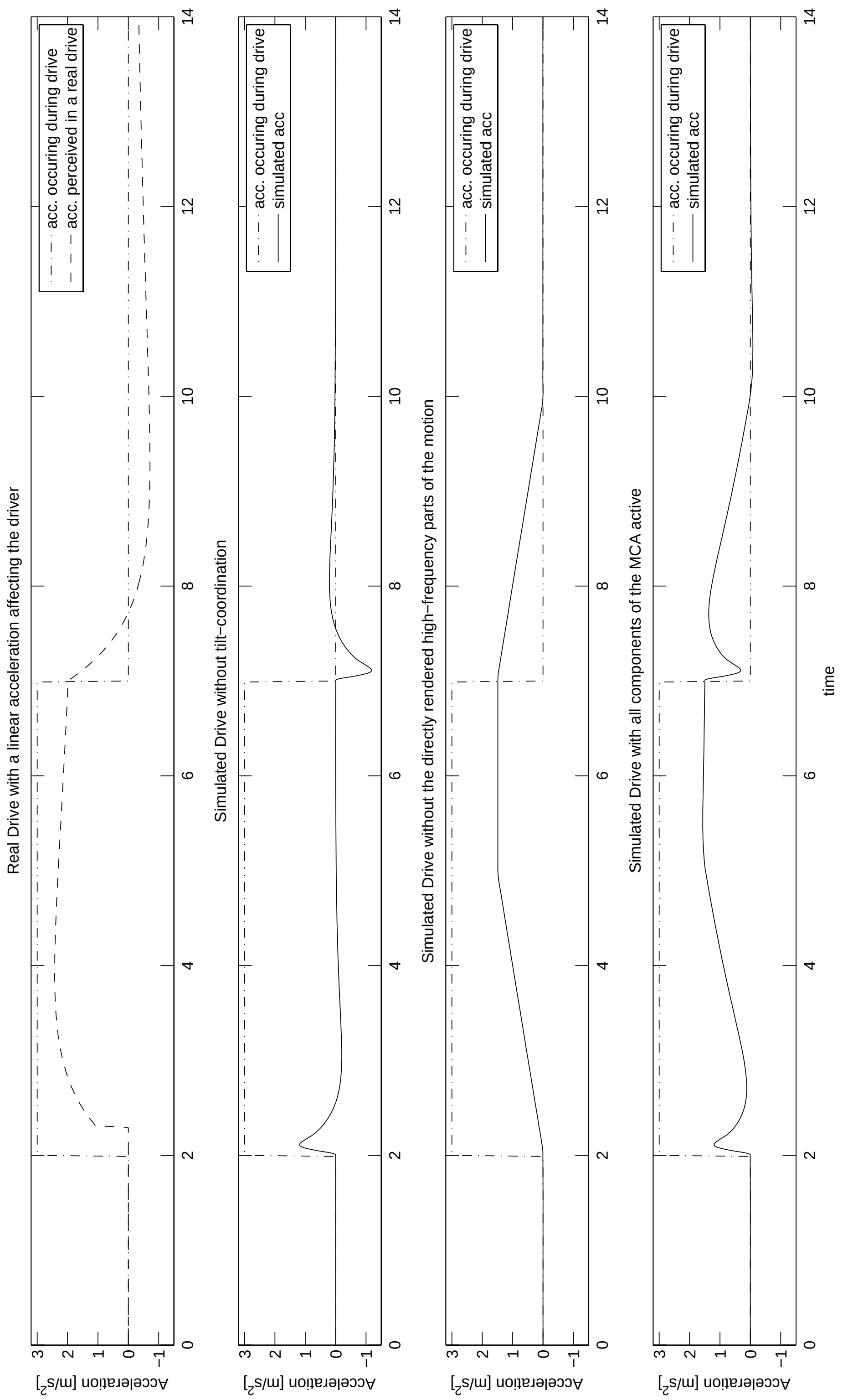


Figure 2.3: The simulation of a step-like *linear* acceleration occuring during a drive (dash-dotted lines). The resulting acceleration in the simulation platform is presented with solid lines. Note the differences while using the complete motion cueing algorithm and parts of it.

Another possibility to influence the presentation of motion cues is the mixture of both methods. The components of the original translational accelerations a_{real} have to be assigned to the two subsystems, being the direct branch and the tilt-coordination cross feed respectively. Richter describes two approaches:

- Give the acceleration to one subsystem, compare the result with the original acceleration and pass the difference to the second subsystem. Richter achieves good results with tilt-coordination as the primary subsystem [Ric71].
- Handle both systems unattached. Both paths in the filter structure get the full signal as an input and process it. It is necessary that the systems are designed/tuned according to each other to yield a good overall result that covers nearly all frequencies off the original accelerations.

The latter method is the approach mainly used today in the common motion cueing algorithms.

2.1.3 Washout

Sustained deviations of the motion platform from its neutral position reduce the working-envelope unnecessarily in one direction. These originate from nonlinearities in the filtering module (e.g. rate-limiting, tilt-coordination). The position and angle commands are once more high pass filtered to ensure that the platform returns to its neutral position. This additional system is also referred to as the *washout filter*. It avoids to reach the saturation of the actuators and ensures the return of the platform to the neutral position, for linear high-pass filters produce a zero-mean signal for input signals with a limited power. It should work at undetectable rates with respect to perception thresholds to avoid false cues.

Nahon and Reid argue that the translational movements should be modulated with third-order high-pass filters, the rotational movements should be passed through second-order high-pass filters. If nonlinear scaling and rate-limiting blocks are omitted, second order filters are generally sufficient to guarantee that the platform returns to its neutral position [NR90].

2.1.4 Discussion of the Classical Concept

The classical approach uses the common frequency splitting technique to modulate the simulated vehicle's motions. An inherent problem is always the presentation of large-amplitude and medium-frequency accelerations. The high frequency parts of the linear and angular motions are simulated directly by scaled movements. The low frequency parts of the longitudinal and lateral motion are simulated by tilting the simulator's cabin frame with respect to the graviational vector. This extraneous motion in the roll and pitch channel has to be limited, to ensure that the tilt coordination takes place below the rotational perception thresholds of humans driving a vehicle. The roll and pitch cross feed can produce significant errors in the perceived lateral and longitudinal specific force when the center of the tilt coordination rotation is not the point of the human rotational perception, being the driver's head. However, if no related visual cues are produced, the driver should not recognize the tilt coordination motion and should perceive the component of the gravitation orthogonal to the cabin's z-axis as a lateral/longitudinal acceleration. The sum of the low and high frequency parts does not always reproduce the scaled specific force with high fidelity. The next paragraphs give assets and drawbacks of the classical motion cueing algorithm as well as some gentle modifications and suggestions for the design of motion cueing algorithms.

2.1.4.1 Benefits

The classical motion drive algorithm bears advantages when it comes to the adjustment of parameters and the structural transparency of the concept. The latter is crucial when the motion cueing system has to be tuned to make use of the full motion capabilities of the simulator. Each parameter that is used has a clear physical meaning. The whole concept is mathematically and computationally simple and easy to implement.

2.1.4.2 Disadvantages

The major disadvantage is caused by the same elements that are responsible for the advantages: the linear filters. The design of the filters is a trade-off between motion platform characteristics and capabilities and the simulated vehicle's dynamics. This process also depends strongly on the driving task to be fulfilled as it determines the maximum accelerations that have to be rendered. Parameters to be tuned are gains, time constants and the applied order of the filters. As the system has to be tuned to render all occuring motions, worst cases are the basis for this adjustment of the filter parameters. Thus, the simulator is not taking advantage of its full motion capabilities when applied to standard driving situations, and the motion-envelope is not exploited to its limits. Another evident drawback is that there is no optimal parameter set due to the lack of understanding of the human motion perception. Unlike other approaches (see 2.2.1), the classical concept does not take the non-linearities of human perception into account (see section 1.5). The best parameter set is usually chosen subjectively between alternative concepts and configurations.

2.1.4.3 Modifications

There are several ways to enhance the motion yield by a classical motion cueing concept. Most of them regard a particular subsystem of the algorithm. They were often developed to compensate for problems occuring with a special simulator.

- Richter suggests to modify the filters used in the tilt-coordination path. First order filters are generally not suitable, for the initial incline of the output is to fast. Second order filters are usually sufficient, whereas nonlinear filters are even better. They produce an output that is the maximum angular velocity allowed in either positive or negative direction. This behaviour forms a switch-like output when the stationary tilt is reached.

This implies jerks in the motion that might be reduced with additional filters [Ric71].

- Nahon and Reid propose to gradually add adaptive solutions to the classical concept. This is one way to enhance the perceptual validity of the driving simulation, and leads ultimately to the adaptive solutions presented in section 2.2.

- Grant et al. suggest to switch some parameters in the motion drive algorithm during the simulation. This allows for parameter sets suiting different driving situations. The global interchange between two concepts is achieved through a structural change in the algorithm with a switching parameter that is either zero or one (see also section 2.2.3). It affects the weight of different paths in the algorithm that are hereby activated or deactivated. Possible driving situations that require a special parameter set appear for instance on roads with many turns or no turns at all (e.g. on a highway) [GAB+02].

- The use of high pass filters can produce disturbing artefacts, for instance a forward sag after finishing a braking manouevre. These effects are also due to the zero-mean output of linear filters after limited-power inputs. Similar effects can be observed during lane-change tasks and are interpreted as a steering instability of the simulated vehicle. To compensate for these disadvantages, Reymond and Kemeny [RK00] introduced modifications of the classical MCA. A non-linear adaptive gain is inserted behind the filters to anticipate artefacts and compensate for them via a control of the motion output (see figure 2.4). This process is critical for reasonable motions might be attenuated in their onset. A reduced perceptual validity is the result.

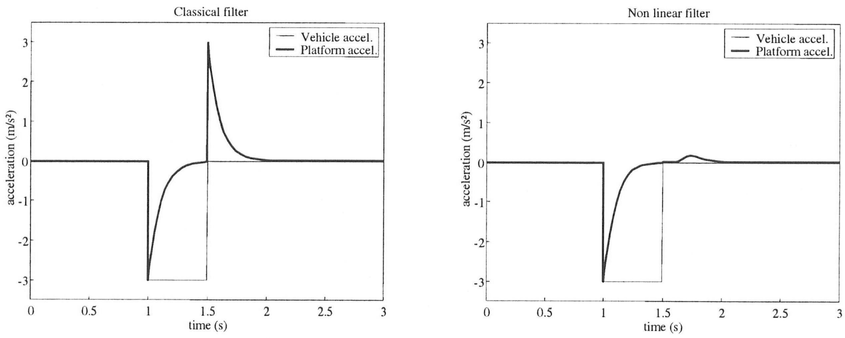


Figure 2.4: Filter output with nonlinear gain to anticipate and reduce false cues.

The approaches given above allow for a better realism of the driving simulation when the classical motion cueing concept adopted from flight simulators is used. But benefits are also to be expected when the classical concept is omitted and especially the filtering part is modified. The next section gives an overview of more sophisticated methods to modulate the motion of the simulated vehicle to suit the simulator's capabilities.

2.2 Advanced Concepts

2.2.1 The Optimal Control Approach

Nahon and Reid [NR90, RN85] developed an *optimal* motion cueing algorithm for a flight simulator based on the work of Sivan et al. [SISH82, IS82, KS72]. This approach works similar to the classical washout algorithm, since linear filters are applied. However, the main difference is that the filter parameters are obtained in advance through a linear quadratic optimisation process, for which the structure and a cost functional have to be given. Here, four transfer function matrices from the two inputs – the aircraft cockpit accelerations and euler angles – to the two outputs being the translational accelerations and the rotational angles are computed. Figure 2.5 shows the structure of the motion cueing algorithm.

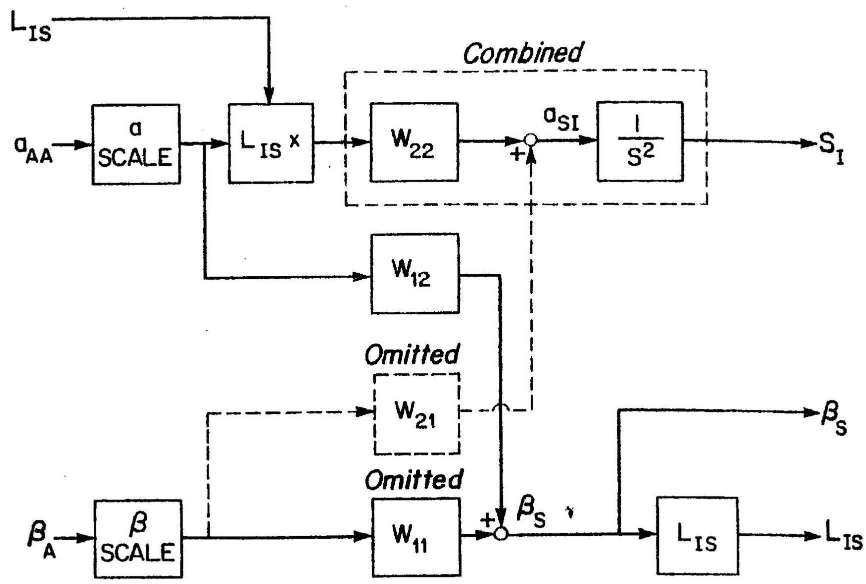


Figure 2.5: The Optimal Control Algorithm. The double integration towards position commands is integrated in the linear path (denoted *Combined*). Since the algorithm uses the Euler angles as inputs, no integration block is used in the rotational path [NR90].

The optimal control problem is to select the input to the motion platform so as to minimize the cost functional, that imposes

- a cost to the differences between the sensed motion in reality and in the simulator
- penalties on a motion that might violate the simulator constraints (see section 6 of [RN85] for a detailed treatment of the cost functions applied).

Figure 2.6 shows the structure of the system subject to the optimisation.

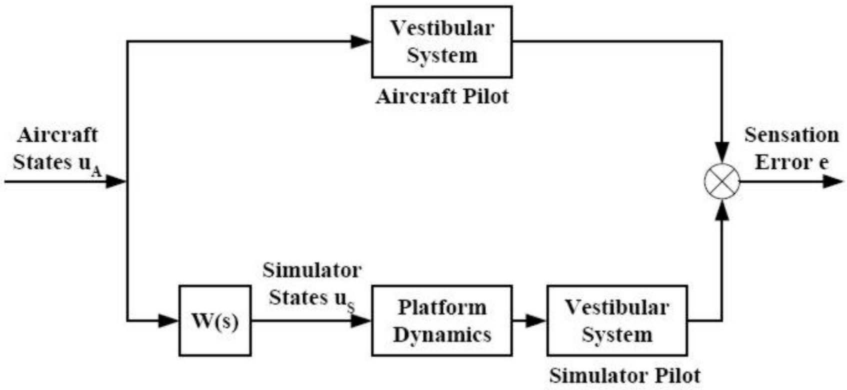


Figure 2.6: The structure of the optimal control problem for a flight simulator. *W(s)* denotes the motion cueing algorithm to be optimized. The motion sensation error e between the motions perceived in reality and those perceived in a simulator is minimized in the optimal control problem through a penalization of high values in the cost function [Tel02].

The time-invariant optimal control problem generates four transfer function matrices, the one resembling the first path in the classical washout approach (denoted W_{22} in figure 2.5) obeys a high-pass behaviour to attenuate the low-frequency translational movements. The longitudinal and lateral accelerations are fed through a transfer function denoted W_{12} that has the form of a low-pass filter. This path works similar to the tilt coordination path in the classical structure. The transfer function from euler-angles to translations (W_{21}) is dropped after the optimisation[4] for it produces no benefits. The euler angles are passed through a filter W_{11} to generate rotational motions. The filters formed by the optimisation tend to have a unity-structure, at least for the pitch and roll channels. This results in a direct rendering of the moves in these channels for uncoordinated flying manoeuvres while scaling and limiting are neglected. Coordinated turns with aircrafts do not produce low-frequency forces that have to be rendered[5] and the special structure of the optimal filters produces signals with W_{12} and W_{11} that cancel each other out in this case. The yaw filter obeys the high pass character also found in classical algorithms, for sustained rotations around the z-axis cannot be rendered with common hexapod motion systems. The high pass character of the transfer functions results mainly from the penalization of large actuator extensions whereas the low pass filters result from the cost imposed on the difference between the motion sensed in reality and in the simulator. Nahon and Reid recommend to reduce the order of the obtained filters sufficiently to simplify the computational complexity.

The main advantage of this design approach is that it uses a model of the human vestibular system during the optimisation (see figure 2.6). This way, the overall motion-sensation error is minimized. The non-linearities in motion perception can therefor be included and a better exploitation of the motion capabilities can

[4]The resulting transfer function matrix has no content different from zero.

[5]The only variation is a sustained higher accelaration along the cockpit's z-axis, but this cannot be presented.

be achieved.

Another benefit could be drawn upon the utilization of cost-functions in the optimisation. These functions can be designed to reflect the needs in motion simulation, thus tuning the optimal filters can be handled by changing the minimization objective. However, Nahon and Reid [NR90] showed that this theoretical advantage does not hold in practice, as the tuning process is neither easy nor transparent with the cost functions used.

The major disadvantage is that the optimal control approach yields fixed parameters just like the classical washout algorithm does. These are calculated in advance to met the simulator capabilities in the motion situations used with the a-priori optimisation. Hence, the motion is not exploiting the performance limitations of the motion platform in ordinary situations, when worst-case situations were used during the design process. Supplementary problems are caused by the missing tilt rate limiting and the perception model that might be used with the optimisation. Also modern mathematical descriptions often take only the mechanical perception into account and neglect the influences a visual cues.

2.2.2 The Adaptive Approach

The adaptive motion cueing algorithm for a flight simulator build by Reid and Nahon [RN85] consists of an empirically determined combination of high- and low-pass filters similar to that of a classical washout algorithm. Though, the great difference is that some coefficients in the transfer functions are varied systematically according to an online optimisation result. This optimisation imparts a great amount of flexibility for it uses a cost function that can be imbued with 'unlimited intelligence' [NR90], for instance a sophisticated vestibular model.

Reid and Nahon use the aircraft cockpit specific forces and angular rates as input to this motion cueing algorithm (see figure 2.7). The first are high-pass filtered with an adaptive gain. The longitudinal and lateral specific forces are also adaptively scaled and fed to the pitch and roll channels. This second path works analogously to the tilt-coordination path in classical algorithms. The angular motion is adaptively scaled and added to the cross-feed part from the second path. Both signal components are filtered together to yield the simulator's angles. The transfer functions for pitch and roll do not show explicitly a high-pass character. For uncoordinated manoeuvres, this eventually produces a direct rendering of the aircraft angular motion that is only scaled. For coordinated manoeuvres, tilt signals and angular motion cancel each other out, the simulator roll angle will stay at zero. A supplementary tilt-rate limiting is included in the cross-feed path, to keep the rates of the produced motion below the perception thresholds [RN85].

A vast advantage of this method is the more realistic behaviour of the simulator for non-worst-case situations. The motion fidelity is only reduced near the system limits and additional lags are omitted. The capabilities of the system are exploited more efficiently. The flight simulator platform used by Reid and Nahon is able to respond in the best way throughout the whole flight envelope. Great disadvantages of this technique are the laborious adjustment process of the cost functions and the high execution time that is due to the great number

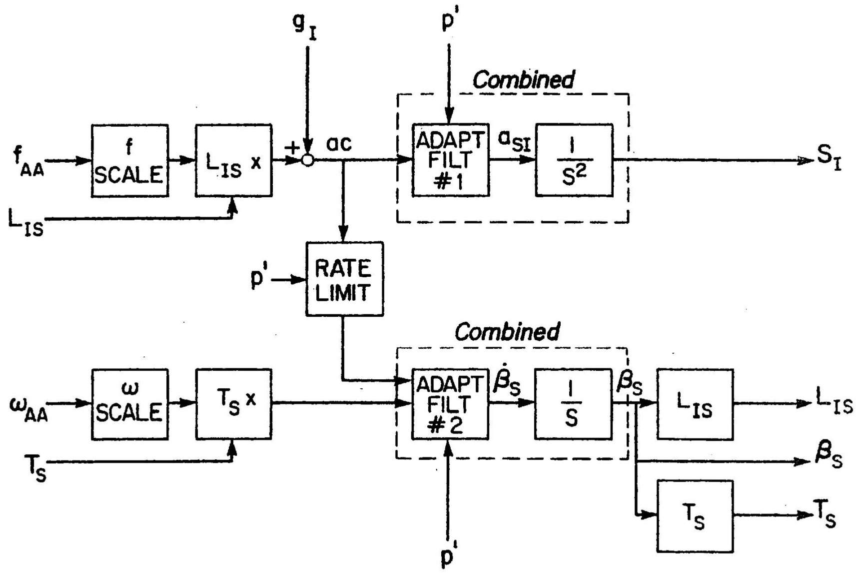


Figure 2.7: The Coordinated Adaptive Algorithm. The figure was taken from [NR90].

of differential equations that have to be solved in real time. Nahon and Reid argue that the number of differential equations is about three times higher than for the classical approach [NR90]. Anyway, this number depends strongly on the parameters that are varied and determined in the adaption process, and often the computational cost can be neglected while using modern computers.

2.2.3 An Algorithm Based on the Lateral Lane Position

Grand et al. developed a motion cueing algorithm that is able to switch between two different concepts for the modulation of the motion produced by a vehicle simulation [GAB$^+$02]. The global interchange between the two concepts is achieved through a structural change with a switching parameter K_y that is either zero or one. The first concept, denoted the lane position based approach, uses a static scaling for the lateral motion that is calculated from the lateral position of the vehicle on the road. The second concept is similar to the classical approach. Both concepts are superposed (see figure 2.8 regarding the linear movements. The roll movements are based on the rolling calculated by the driving simulation and the sustained lateral accelerations presented through the tilt-coordination that are due to road curvature and not due to lane position changes.

Additionally, parameters of interest can be switched during the simulation. Parameters used with the classic approach and lane position based concept can be changed to new values using linear ramps. The parameters used for the lane based approach are additionally modified by a second order filter. Both concepts cannot prevent the occurence of lateral and roll motion errors due to the

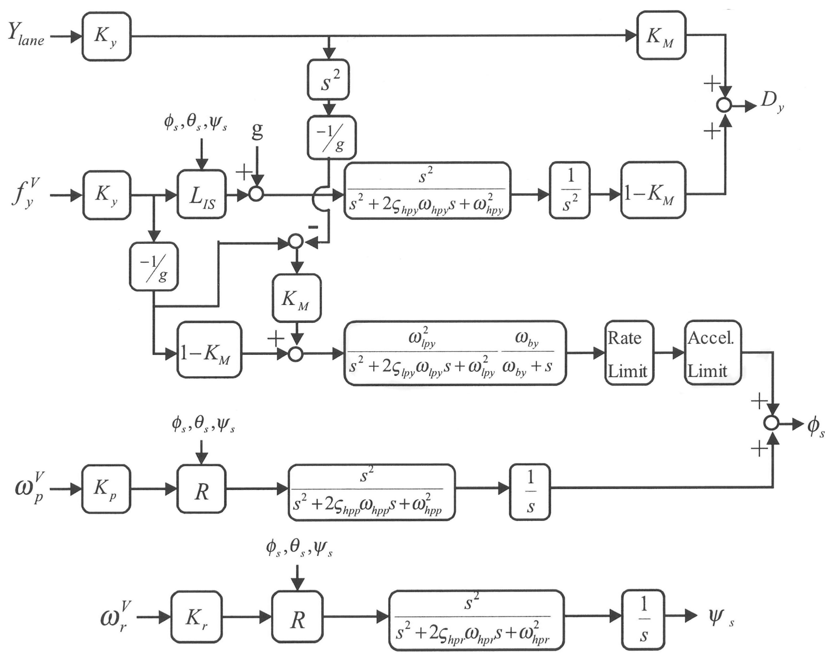


Figure 2.8: The lateral, roll and yaw channels of the motion drive algorithm used with the VIRTTEX simulator [GAB+02]. Besides the traditional inputs f_y^V and ω_p^V, the algorithm uses the lateral position in the current driving lane Y_{lane}.

limitations of the platform Grant et al. use. The parameters constituting the algorithms can be used to adjust the relativ size of these errors.

2.2.4 A Driving Task Adaptive Motion-Cueing Algorithm with Dynamic Scaling

To determine the hexapod's actions, one has to compute the prevailing position of its centroid and the corresponding euler angles in real time. This is done by the motion cueing algorithm that calculates those values out of the vehicle acceleration and angular velocities. In the approach proposed by Tajima et al. [TMY06], both the longitudinal and lateral acceleration are scaled down (see figure 2.9). All other inputs from the vehicle simulation are not modified through scaling. These values are subsequently passed through the washout algorithm to obtain the necessary attenuations and integrated over time to yield position and angles.

Tajima et al. assume that the initial segment of the motion is of great importance regarding the driver's sensation and should be simulated as accurately as possible. The restrictions produced by the simulator may be compensated by augmenting the initial part of the transient response of the motion platform. The static scaling gains used in washout algorithms to reduce the motion might be substituted by frequency dependent filters that obey a high-pass character.

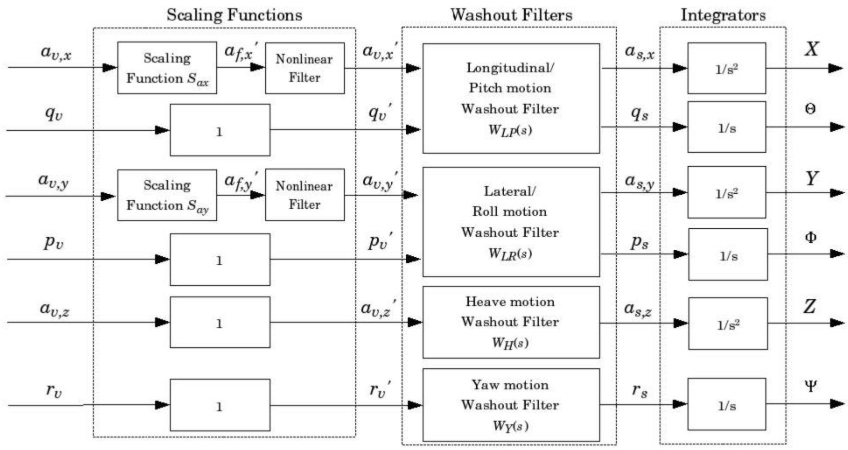


Figure 2.9: The motion cueing structure proposed by Tajima et al. [TMY06]. The scaling blocks for the longitudinal and lateral motion are high-pass filters to augment the onset of the movements in these directions.

They are of the the following form:

$$S_i = \frac{s + \omega_{L,i}}{s + \omega_{H,i}} \text{ with } \omega_{L,i} < \omega_{H,i} \text{ for } i = \text{longitudinal(x) and lateral(y).} \quad (2.1)$$

This preliminary filter is a simple lead-lag network with a high band-pass characteristic. Its steady state gain $K_{L,i} = \omega_{L,i}/\omega_{H,i}$ should be choosen as high as possible to take advantage of the whole working envelope[6] and thereby improve the fidelity of the simulation. Tajima et al. determine $\omega_{H,i}$ experimentally, $\omega_{L,i}$ is adjusted to achieve a proper steady state gain. The resulting phase lead characteristics constitute some inadequate motion cues, when the filters' inputs change at a high rate. This effect is reduced by a nonlinear transformation of the filters' outputs.

The washout filters itself are designed by the use of three optimal control problems[7]. These yield the filters for the longitudinal and pitch motion ($W_{LP}(s)$), for the lateral and roll motion ($W_{LR}(s)$) and for the yaw motion ($W_Y(s)$). The cost function takes the motion sensation error and the platform deviation from its origin into account. The resulting filters are linear. A remaining problem is a reasonable choice for the parameters used during the optimisation, e.g. weights in cost functions. These strongly influence the results for they are able to emphasize the direction of optimality the filters show (e.g. optimality with respect to constraints or towards sensation error).

Tajima et al. [TMY06] assigned the cut-off frequencies of the implemented filters by subject evaluation. Two configurations are tested against each other, the better one is kept[8]. They conclude that the proposed new algorithm is

[6]mind $\omega_{L,i} < \omega_{H,i}$

[7]Tajima et al. refer to [TCH99] for a good description. The optimal control method is based on LQG theory (LQG = linear quadratic Gaussian). In LQG theory the input signals are assumed to be stochastic or alternatively impulses in a deterministic setting, and the expected value of the output variance (or the 2-norm) is minimized [SP05].

[8]This process is also called *one pair comparison method*.

superior to the old one, since the vehicle's behaviour could be predicted more accurately by the driver.

The main observation of Tajima et al. is that the desirable filter design depends strongly on the current driving task. A motion drive algorithm that adapts to the present situation should provide the driver with more realistic motion cues. That is why they argue that the cut-off frequencies should be properly scheduled corresponding to the driving task and the vehicle's speed. The main design principles attained from the subject evaluation were:

- The high frequency components of the longitudinal motion should be presented as realistic as possible.
- The lateral acceleration is more important than the yaw rate at high speed during a lane change task.
- The yaw rate should be augmented at low speed[9].

To render a realistic simulation possible, Tajima et al. introduce a new variable LD that is filtered and afterwards used to judge whether the vehicle is making a lateral-directional manoeuvre. This decision variable influences the gain and cut-off frequencies used in the W_{LP}-filter. The velocity of the simulated vehicle affects the yaw movement as well as the gain of the W_{LR}-filter. Its cut-off frequency depends on the velocity and the lateral acceleration. The adjustment rules are formulated as piecewise constant or linear functions depending on the decision criteria given above.

[9]Mind the difference between a sustained bend and a sharp corner.

2.3 Discussion of the Concepts, General Problems and Approaches to Further Enhancements

Recent years have seen advances in the maturity of driving simulation platforms. The driver is provided with at least good visual and motion cues and the realism of the simulation is significantly increased by the coherence of the different sensory informations that contribute to the driver's perception of motion. But this realism still depends strongly on the capabilities of the motion platform and it is still to be enhanced, since it is the immersion of the driver into the simulation that founds the basis for successful driving tests. The results of the tests are only valid if both the actions and reactions of the vehicle and the driver are similar to the ones in reality. This is strongly corrupted when the driver is concerned with motion sickness or irritated because of false cues affecting him during the drive and especially in dangerous or complex situations (e.g. a high amount of cars on the road).

Evaluating the validity of test-drives in simulators is a difficult task. As a test-drive generally consists of different driving situations that have to be rendered with a good perceptual validity, different manoeuvres are used in the tuning process. This adjustment should also take the simulator capabilities into account, as well as the characteristics of human perception. The result are manoeuvre-specific parameter sets for the filters that are implemented in the motion-cueing algorithm. However, the objective for static motion cueing concepts is one parameter-set that produces both a good behavioral validity and a realistic perceived motion. The level of maturity reached in this process can be assessed with test drives consisting of curve-driving or braking manoeuvres. Afterwards, the driver has to answer questions like the following [BK05]:

- How realistic was the feeling of driving?
- How accurately could the car be handled?
- How well do the simulator's movements match reality?
- How safe did you feel?

The influence of learning has to be avoided in these test-drives, an a-priori simulator training is suggestive.

Maintaining a realistic trajectory in the simulated environment with the vehicle is very difficult. The visual and kinesthetic cues that are perceived in a real test drive impart the information necessary to steer the vehicle. Especially the linear accelerations occuring for instance during turns or braking manoeuvres are important. These motion cues aren't easy to render with a driving simulator, for it has only limited capabilities considering the presentation of displacement, velocity and acceleration. This task is further complicated by the human's nonlinear perception of motions.

Given that the motion cues presented to the test driver are very limited due to a small displacement envelope and limitations in actuator power, the system cannot render the full dynamics of a real car. Some discrepancies between the motion and the visually simulated motion are to be expected. Hence, the validity of the motion cueing is affected. The primary obstacle regarding the generation of realistic motions is the software that translates the simulated vehicle's motion to position commands that do not violate the motion platform's constraints, the motion drive algorithm. The physical validity of the simulation generated with motion platforms is generally limited. The high-pass filtered components are transferred to the motion platform, which has typically a low-pass frequency response due to resonance damping mechanism and limitations in power. Sustained accelerations and high amplitudes cannot be rendered properly. Limitations in the actuator stroke are responsible for a reduced physical validity.

One of the main problems that still remains is that the simulation of lateral accelerations with the system consisting of the direct path and the tilt-coordination path covers not the whole range of frequencies contained in the original signal. Transitional signals might not be rendered precisely for the tilt-coordination needs time to build up whereas the transient accelerations decrease fast, thereby avoiding the saturation of the actuators. For a step-like input that should be presented to the driver, a depression in the perceived acceleration occurs after some time (see figure 2.3).

Scaling the amplitude and rate limiting of the motion are two methods to make use of the non-linearities and imperfections in the human motion perception. Grant et al. [GAB$^+$02] state that a large scaling might be less realistic since it yields a large phase lag for high lateral simulator velocities result in a pressure drop in the hydraulic system. Tests on their particular simulation platform suggest to reduce the roll errors at the expense of lateral specific force errors.

The *classical motion cueing concept* is the approach often used with simulator platforms. The structure is clear and the parameters constituting the modules have a clear physical influence. Therefore, it is easy to implement and to tune. The major disadvantage of this concept is based on the same elements that are responsible for the advantages, the linear filters. As the system has to be tuned to render all occuring motions, worst cases are the basis for this adjustment of the filter parameters. Thus, the simulator is not taking advantage of its full motion capabilities when applied to standard situations and the motion-envelope is not exploited to its limits. In addition, this motion-cueing algorithm does not take the non-linearities of the human perception into account.

The *optimal control approach* is a derivative of the classical concept. The main difference besides some minor structural changes lies in the paramaters constituting the filters. These are determined in advance to yield an optimal solution for a specific set of inputs that is used with the optimisation that minimizes a cost function. The major disadvantage is that the optimal control approach yields fixed parameters just like the classical washout algorithm does. These are calculated to meet the simulator capabilities even in worst-case situations. Hence, the motion is not exploiting the performance limitations of the motion platform in ordinary situations. Supplementary problems can be caused by a missing tilt rate limiting [NR90].

The *coordinated adaptive approach* lacks the problem of fixed parameter sets. Parameters of interest can be adjusted online during the simulation, using a cost function and a repeated optimisation process similar to the optimal control approach. Great disadvantages of this technique are the laborious adjustment process of the cost function and the high execution time that is due to the great number of differential equations that have to be solved in real time.

Since motion simulation platforms are generally unique installations that meet the particular interests of the research group working with them, there is no common solution to the problem of a lack of realism of the simulation. The motion cueing algorithm that is used always has to be tuned towards a sufficiently good setting, which is often an extensive process. Though, there are some promising suggestions concerning the structure of the used concept.

2.3.1 Desireable Features of Motion Cueing Algorithms

The primary desire/measure when rating a simulator and its motion cueing algorithm is always its realism. The capability to reach an excellent or at least a good perceptual validity (see section 1.3) is founded by the following characteristics of the underlying motion cueing algorithm:

- The whole system should exploit the non-linearities in both human perception (thresholds, adaptation of sensing) and in the simulator's capabilities to render a realistic simulation.

- The simulated motion should only be scaled down or limited when the capability constraints regarding displacement, velocity and acceleration are reached. A static down-scaling of the motions' amplitudes should be as moderate as possible, a dynamic scaling that depends on the current state of the platform and the motions to be expected can yield better results (see also section 2.2).

- Nahon and Reid state that it is a great advantage when the system is adjustable for different pilots in flying simulators, as those may prefer different parameter settings for the algorithm [NR90]. However, this adaptibility to the drivers' preferences is not suggestive for driving simulators since the test-drivers are usually employed short term and recruited from the population, thus not being test-driving experts. Additionally, the comparability of the test might be attenuated without increasing the behavioral validity.

- A great feature is the adjustability of the algorithm for different situations. This implies that the system is not tuned for an over-all worst-case situation, but for the different maneuvres occuring in every segment of the whole drive.

Another criteria is the amount of work that has to be accomplished when tuning a motion cueing algorithm. That is to say, the algorithm should be easy to adjust. This is facilitated when the algorithm is based on a minimum of free parameters, which are related to meaningful physical quantities. The effect of parameter changes is then easy to determine in advance.

The last desireable feature is a high execution speed which is directly related to a small number of differential equations in the filter system. This is crucial since the algorithms generally have to be solved in real time.

2.3.2 Approaches to Overcome the Problems

The problems faced when designing a motion cueing algorithm for a particular system depend especially on the capabilities of the particular system. Yet, there are some attempts to improve the algorithm and thereby the simulation. The ability to behave non-linear is necessary to exploit both the simulator's and the human perception's capabilities. This is more likely to be achieved with adaptive algorithms. Though, the classical algorithm bears advantages as well, for the structure is transparent. An element that is especially important are the cross-fed channels of tilt-coordination. They render the sustained translational accelerations possible that are necessary for a good perceptual validity. However, these elements confer a great amount of complexity to the algorithm.

Reymond and Kemeny argue that the motion cueing algorithm's output should be scaled depending on the context, and that a reduction of the transport delay is suggestive. The integration of visio-vestibular perception models in the design process bears a promising potential [RK00].

Nahon and Reid suggest a lot of improvements for motion cueing algorithms. One concept is to combine the benefits of different motion cueing approaches. A classical layout with adaptive filters (a hybrid design) could yield a good overall result. An elementary task is to decide which blocks are to be made adaptive, and which ones are kept in the static form. The adaption part of the algorithm is also a good starting point for future work. More sophisticated cost functions could be used to make the adaptiveness more intelligent [NR90].

Richter makes several suggestions to improve the realism of the simulation with advanced motion cueing systems. Generally, the motion envelope and human motion perception can be exploited to a higher degree with the use of more information on the current and future driving situations. The disadvantages of the used methods can be compensated with knowledge on the track to be driven. The current velocity, the curvature of the road ahead and an expected lateral acceleration could for instance influence the artificial tilting mechanism, that is likely to produce a significant phase lag that might be compensated. Filters can be tuned towards the demands of certain driving situations and tracks. As long as only more or less straight roads are driven, the mechanism of tilt coordination ought not to be used [Ric71]. Tajima et al. emphasize the same approach, stating that the desirable filter design depends strongly on the current driving task. A washout algorithm that adapts to the present situation should provide the driver with a more realistic motion cueing [TMY06]. This is the starting point for this thesis. As it is possible to gain information about the ongoing driving situation, it should be possible to adapt the motion cueing algorithm of the driving simulator subject to this thesis[10]. The solutions to this task are described in the next chapters.

[10]The driving simulator is located at the Institute for Transportation Systems, German Aerospace Center, Braunschweig.

Chapter 3

Time-Variant Motion Cueing Algorithms

Driving simulations can be seen as hybrid systems, that involve both continous-valued variables and discrete phenomena. The simulation evolves according to equations that depend on both discrete variables like changes in the surrounding environment (sunshine, rain, fog, dirty roads, the road surface) and continously distributed variables like the physical states of the vehicle. The current road condition beneath the car for instance is given as a sequence of situations that is continous in time but rather discrete in its characteristic. The behaviour of

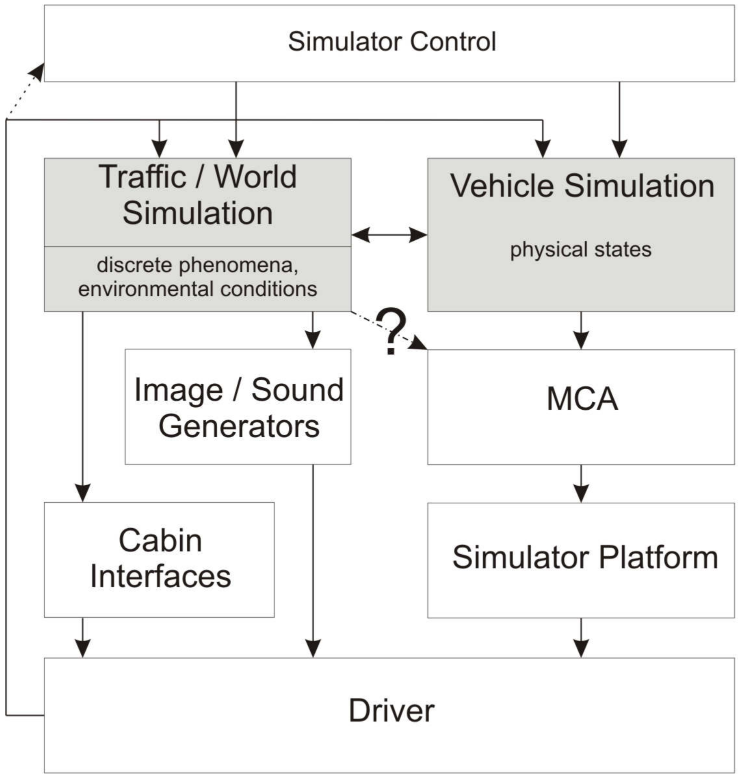


Figure 3.1: Information flow in the driving simulation. The influences of discrete phenomena and environmental conditions on the motion cueing algorithm are examined in this thesis, denoted with a question mark at the corresponding dashed arrow.

the car is controlled by the driver, and the driving situation is also eventually determined by decisions the driver makes, like choosing the speed or turning at intersections. Figure 3.1 shows the signal flow associated with the driving simulation.

As the simulated vehicle's mechanic states are continous variables, and the driving situation is rather a variable that jumps between discrete values, it is reasonable to see this system as a hybrid system. Hybrid systems that consist of several subsystems and a rule that governs the switching among them are called *switched systems*.

3.1 Switched Systems

The switched system is a special form of the hybrid dynamic system that contains both continous dynamics and discrete elements. It is of multi-model nature. The discrete variable dynamics of the hybrid system, are determined by systems like a digital automaton or input-output transition systems with a countable number of states [Bra97].

The component models constitute the low-level or local dynamics, they are also denoted as modes of the whole system (see figure 3.2). The transition between modes is achieved with a high-level coordinator, denoted the supervisor[1] which produces a switching signal[2] σ. This signal may depend on its own past values, the time, states or outputs of the system and even on external signals. It determines the currently active mode of the system.

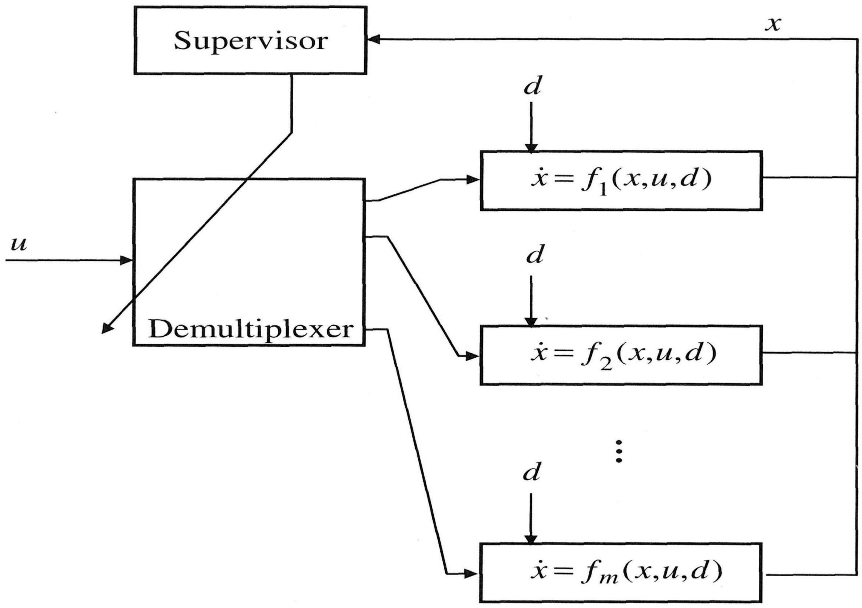


Figure 3.2: A switched system consists of the low-level dynamics $\dot{x} = f_i(x, u, d)$ and a supervisor that coordinates the switches between the different modes. Usually, the output $y = h_i(x, u, d)$ is switched, whereas the input is given to all subsystems [SG05].

Switched systems can be used to model systems with known or unknown sudden parameter variations. Examples for systems in the class of switched systems are:

- synchronously switched systems
- periodically switched systems
- systems with changes in their structure

[1]Also: mode changer, gain scheduler, reference governor
[2]Also: *switching rule* or *switching law*

- systems with partwise failures

The switching among different subsystems hence is an essential feature of many engineering and real world systems. In case of linear subsystems, they serve as a link between common linear systems and complex or uncertain systems, for the switching may produce complex system behaviour such as chaos or multiple limit cycles [SG05]. Nevertheless, switched linear systems are easy to handle with tools from linear and multilinear analysis.

Logical decision-making is often incorporated in controls. Many control systems perform logical checks that determine the mode the continous-variable system is operating under. Controllers whose subsystems are not only familar dynamical systems like integrators, transfer functions, summers and gains but also event-driven logical elements and switches are often called logic-based switching controllers [Mor97]. Complex systems (e.g. nonholonomic systems) are often not stabilizable with a single (linear) continous state feedback controller. Though, they might be stabilized with a multi-controller architecture that switches between several candidate controllers, as the one given in figure 3.3 [SG05]. Benefits are also obtained regarding the control-performance of simple LTI systems. These might be for instance controlled in turn by a linear controller and a time-optimal one, depending on the deviation from the desired set point.

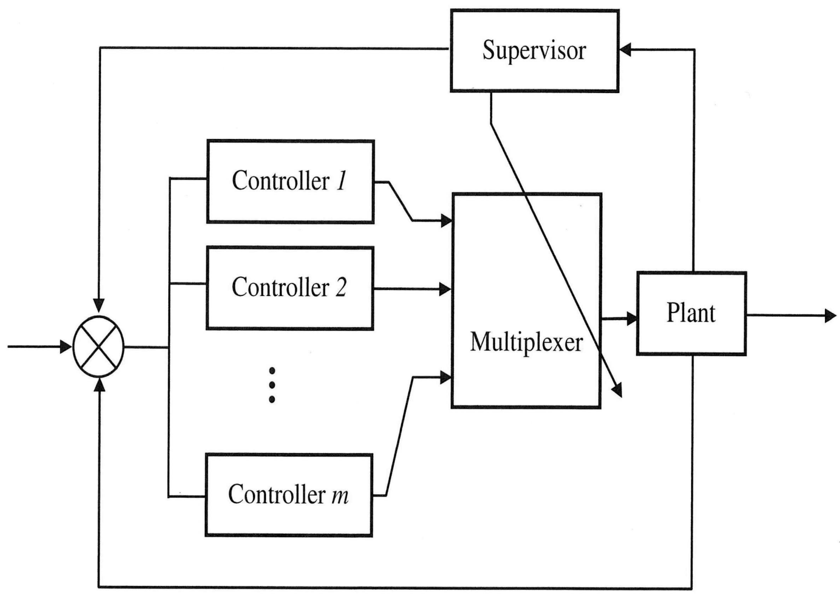


Figure 3.3: In a multi controller architecture, the supervisor shifts the input of the plant by choosing between different controllers [SG05].

The switching rule σ is a piecewise constant signal that takes values from a set $M = \{1, ..., m\}$ where m is the number of modes. Different switching laws generally produce totally different system behaviours and outputs. A classification of switching rules can be made upon their dependence on several states and variables:

- A *switching path* is a pre-determined switching law that depends only on the time.
- *Time-driven* switching law: The signal depends only on its own past values and the time.
- *Event-driven* switching law: The signal depends on its past values, the states and outputs of the system and on some external signals. It is not explicitly time-dependent.
- The pure state or output feedback law is a special case of the event-driven rule that depends only on the system's states or outputs.

The switching signal is denoted *determinant* over the time-interval $[t_0, t_1]$ when it generates a unique switching path for the initial state x_0 over this interval. Sun and Ge give a useful collection of properties for both continous and discrete-time switched linear systems [SG05].

The system under consideration can be seen as a switched system with the driving situation being the switching signal. Thus, it is reasonable to use a motion cueing algorithm that adapts to this signal, and not a static algorithm that shows the same behaviour for all driving tasks.

3.2 Structural Changes in the Motion Cueing Algorithm

The task of a motion cueing algorithm is to modulate the vehicle's movements produced by a simulation to fit a motion platforms capabilities. Main in- and outputs are thus the original and modulated movements of the simulated car. The concept presented in this thesis uses the current and a predicted driving situation to adapt the motion cueing algorithm. This adaption should eventually lead to an improved realism of the driving situation and a better exploitation of the motion platform's capabilities.

As the simulated vehicle's physical states are continous variables, and the driving situation is a variable that jumps between discrete values, it is reasonable to see this system as a hybrid system. The driving situation can be thought of as a event-driven switching signal that is influenced by the vehicle's physical states, for instance position and velocity of the car (see figure 3.1). The particular situations are defined as follows:

1. The *default* situation. This situation is not a special driving situation, but rather a collection of all other defined and undefined situations.
2. *Straight*: This situation is characterized by a road environment that lacks a severe curvature. Illustrative examples are highways or rural roads in shallow landscapes.
3. *Intersection*: This driving situation is likely to be present while driving through an urban environment. It is characterized by junctions, intersections and turns that are generally driven at a small velocity.

4. *Left bend*: A sustained turn predominately driven at medium velocities. This situation is more likely to be encountered in rural environments and produces sustained lateral accelerations that are to be simulated.

5. *Right bend*: Analogously to the left bend.

As the vehicle travels through the simulated environment with a continously changing velocity, the driving situation changes from one type to another.

The hybrid nature of the dynamic system under consideration suggest to use a concept that adapts the system and in particular its structure to the governing variable, the driving situation that is to be seen as the switching signal σ. In this case, it is produced by an external supervisor that observes and recognizes some environmental conditions (mainly the road curvature) and potentially some vehicle states like the velocity. The adaption of the structure can be achieved

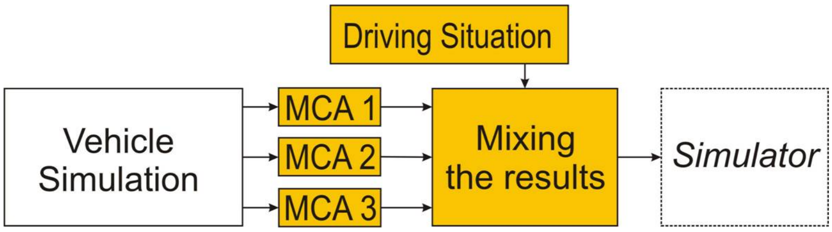


Figure 3.4: The concept for the adaption of the motion cueing algorithm. The output is calculated out of the subsystems' outputs depending on the driving situation as the switching signal.

with different subsystems (see figure 3.4). Each of these *sub motion cueing algorithms* suits the expected motions of one driving situation especially good with respect to the limitations of the motion platform[3]. The overall output of this structural changing motion cueing concept is formed out of the subsystems' outputs, depending on the driving situation. The next sections describe the implementation of an adaptive motion cueing concept and give a discussion of assets and problems that occured during the development.

3.2.1 Implementation

The hybrid nature of the system under consideration suggest to use a motion cueing concept that changes in its structure. These structural changes lead to a parallel architecture of the adaptive motion cueing algorithm, that uses

- several sub-algorithms at the same time to calculate the modulated motions that are later on passed to the motion platform,
- a weighting device that produces the final output of the adaptive algorithm out of the single sub-algorithms' outputs

[3]The default algorithm is an exception, it is tuned towards a good fit with all driving situations.

- and a subsystem that analyses the current driving situation and produces the weights for the weighting device.

The three modules of the adaptive motion cueing algorithm are described in the next sections. Figure 3.5 shows the implemented Simulink plan.

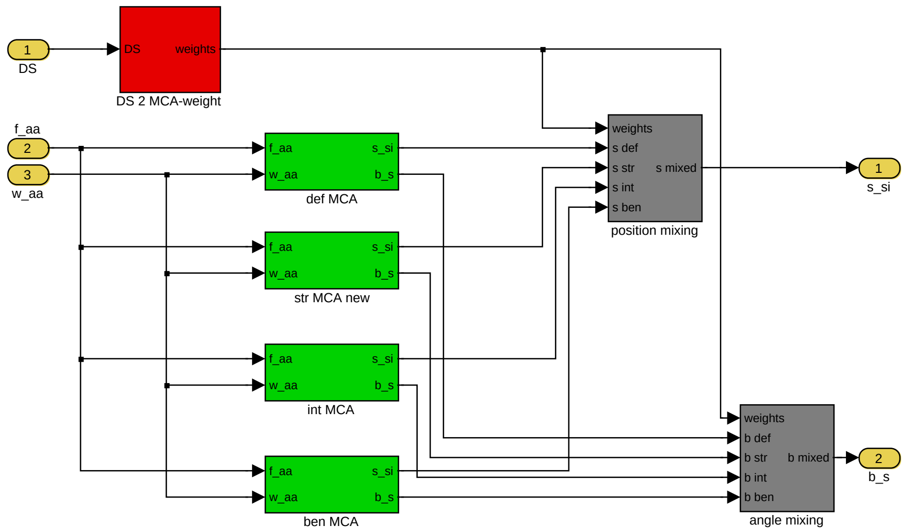


Figure 3.5: The basic adaptive motion cueing algorithm. The output is calculated as the sum of the subsystems' weighted outputs. The sub-MCAs are highlighted with a green background whereas the summing units are grey. The subsystems are shown in detail in figures 3.6, 3.7, 3.8 and 3.9.

3.2.1.1 Sub-Motion Cueing Algorithms

The adaptive motion cueing algorithm uses several sub-motion cueing algorithms that all suit a particular driving situation especially well. All subsystems are hitherto implemented as static classical motion cueing algorithms, that differ mainly in the used parameter sets. Though, it is also possible to use completely different concepts as those presented in chapter 2 or even completely different ones. At the moment, only the sub-algorithm matching the *straight*-driving situation shows a slightly different structure, for it uses a splitted scaling for the linear motions. The single MCAs are:

- The *default* MCA: A Motion cueing algorithm that suits all driving situations. Since the DLR driving simulator currently runs with classical algorithms that were tuned towards a generally good presentation of motion cues, one of the resulting parametersets is used. Figure 3.6 on page 48 shows the Simulink realization.
- The *straight* MCA: Driving on a straight road involves lateral short term accelerations. Thus, this motion cueing algorithm emphasizes the use of the direct rendering of the lateral motions. This is achieved with changed

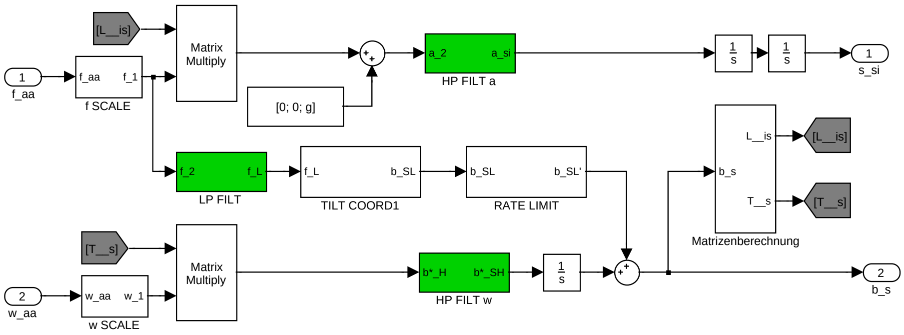


Figure 3.6: The *default* sub-MCA that is tuned to suit all driving situations.

break frequencies in the high and low-pass filters that extract components of the linear motions. In particular, the frequency splitting of the linear motions in the lateral channel are changed. A larger band of signals is fed into the direct rendering path, for the break frequency of the high pass filter is smaller. Vice versa, the break frequency of the low-pass filter in the tilt-coordination path is smaller, to render only the sustained and very low frequency motions with this artificial movement. The changes in the parameters are reasonable, since there are no lateral motions and accelerations to be expected like those in sustained turns that would harm the simulators capabilities. The main lateral accelerations on straight roads are likely to be caused by lane change manoeuvres and a weak curvature. The shift towards a more direct rendering is additionally supported by a splitted scaling of the linear motions. Figure 3.7 shows the Simulink plan. To compensate for the limitations of lateral displacement, a prepositioning

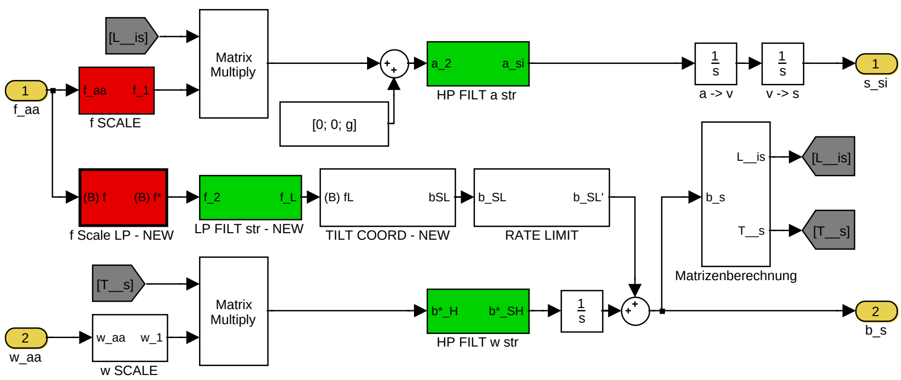


Figure 3.7: The *straight* sub-MCA that suits especially good for driving situations lacking sustained lateral accelerations. Differences towards figure 3.6 result from the splitted scaling (red blocks) and different parameters used with the the filters (green) and scaling blocks.

device could be used that acts upon certain conditions of the drive (see also chapter 4). The simulator cabin is biased from its original position to either the left or the right side of the motion envelope, to allow for larger movements in the other direction. The *straight MCA* should take this enhanced capabilities of the platform into account perhaps with a reduced down-scaling (this would result in larger amplitudes) or shifted filter break frequencies.

- The *intersection* MCA: The intersection MCA should be tuned towards a good rendering of motions occuring with the third driving situation. Urban environments are characterized by a lower driving velocity and sharper turns, although there are of course exeptions like roads with several lanes in both directions (inner city highways). The yaw movement could be emphasized to increase the realism of the simulation.

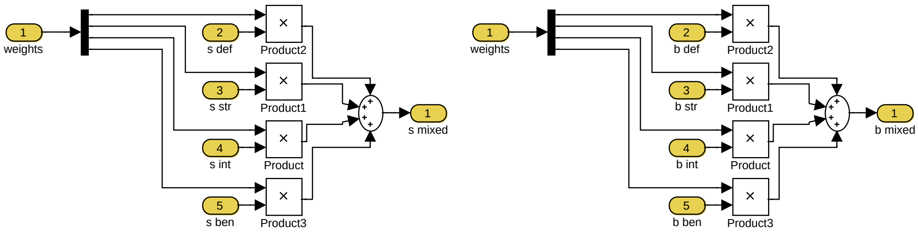
- The *bend* MCA: This MCA should emphasize the good presentation of sustained lateral and longitudinal accelerations, that are one of the challenges an engineer faces when tuning the parameters of a motion cueing algorithm towards a good realism of the simulation. This is due to the limited travel in the horizontal degrees of freedom that imposes limitations on the accelerations that can be presented. The common work-around is the artificial tilt-coordination motion that unfortunately implies several drawbacks, for instance a severe lag in the rendering of accelerations. Some of the rendering problems imposed by the simulator platform might be attenuated with the prepositioning mentioned with the straight MCA above. However, this mechanism is only suggestive as long as the resulting motions are mainly unidirectional. The motion concept currently used with the DLR simulator feeds the tilt-coordination movements back into the linear path by use of the transformation matrix *[L_is]* (see figure 3.6 and 3.7). This results in a washout of the linear motions, and the overall motions cover both sides of the motion envelope. Chapter 4 gives a more detailed description of the prepositioning method.

The motion cueing algorithms given above are designed to match the driving situations given in section 3.2. As it is suggestive to divide these situations into more specific ones, it is also reasonable to design motion cueing algorithms to match the more specific demands of the resulting situations. This might require to leave the framework of the classical motion cueing concept with its fixed parameters towards more sophisticated concepts, like those given in chapter 2.2. Especially optimal concepts allow for a directed tuning towards special situations. The optimization process often imposes a cost on the difference between original and modulated accelerations. Therfore, a sequence of original motions has to be given that can either include a special sub-set of driving situations or a mixture of all movements occuring during a drive. The utilized sequence strongly influences the resulting motion cueing algorithm.

3.2.1.2 Calculating the Output - The Mixing Unit

All sub-MCAs work in parallel and produce simultaneously motion signals. The output of the adaptive motion cueing algorithm has to be computed out of these single results. This is achieved with two weighting units that are highlighted with a gray background in figure 3.5. One is used with the linear movements, the other one with the rotational motions. Both are similar in design and function, see figure 3.8 for a comparison.

The wheigthing modules multiplicate each of the sub-MCA signals with a time-varying weight that is delivered by a unit that calculates these values out of the switching signal, viz. the driving situation. It is described in the next section. The weights vary between zero and one, with the latter value indicating that the output of the particular MCA is taken fully into account. All weighted sub-MCA signals are summed up and passed to the output of the adaptive motion cueing algorithm.



(a) Weighting the Positions (b) Weighting the Angles

Figure 3.8: The two weighting units. The input values – the positions *(s)* and angles *(b)* of the motion – produced by the different sub-algorithms are multiplicated with weights corresponding to the driving situation/switching signal.

3.2.1.3 Calculating the Weights

The weights that are passed to the mixing unit have to be calculated out of the driving situation. This is achieved with a logical unit, that compares the input signal – an indicator of the active driving situation – to the values given in section 3.2 that indicate the different situations. The comparison yields a boolean one or zero. This value cannot be used directly as a weight, it has to be converted to a floating point number in a data type conversion block. Figure 3.9 shows the Simulink implementation.

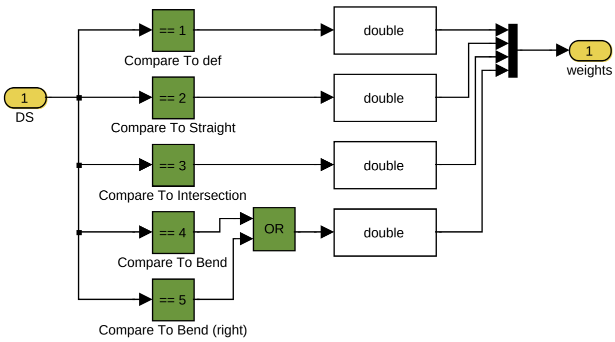


Figure 3.9: The module that forms the weights for the sub-MCAs from the driving situation

The *straight*-algorithm for instance should be active as long as the driving situation shows no sustained lateral accelerations. Thus, the output signal should attain a weight of one for the *straight*-MCA when the input to the unit equals two, this way indicating that the straight driving situation is active. Vice Versa, the *bend-MCA* should be active while the driving situation is either a left (=4) or a right turn (=5). Therefore, the corresponding situations and comparisons in the weighting unit are coupled with a logical or. Using tight comparison blocks and a discrete-valued driving situation ensures that only one of the sub-MCAs is weighted fully. The resulting values are passed onto the mixing unit. Figure 3.10 shows the resulting weights for a time-varying switching signal.

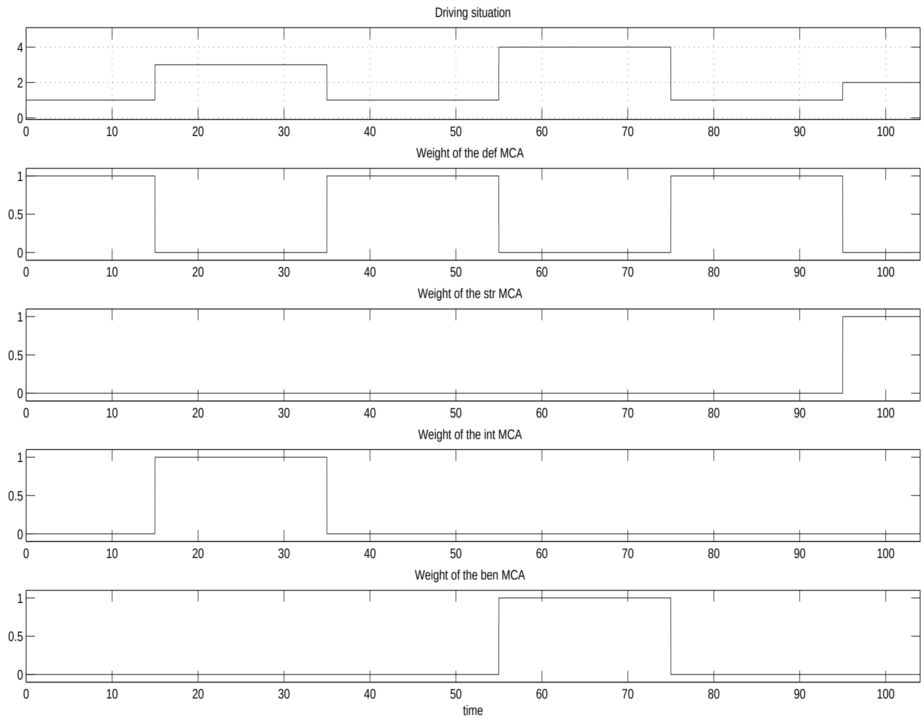


Figure 3.10: The weights for the sub-MCAs evolving in time.

3.2.2 Discussion and Improvements to the Basic Solution

The output of the adaptive motion cueing algorithm is shifted from one sub-MCA to another depending on the current driving situation. However, the reference positions and angles produced by the subsystems may differ greatly, as long as the switching does not occur when the platform is located at the neutral position and no accelerations are to be presented. Hence, the switching between the results of different static motion cueing algorithms generally yields jumps between two sets of concurring reference values. This jumping occurs in one integration step size as long as the test-environment is applied. While the adaptive motion cueing algorithm is used with the driving simulator, it occurs in one sample-time instant. To follow the changing reference values, the actuators have to execute a great power that results in jerks and short but large accelerations until the new reference values are reached. Thus, the corresponding uncontrolled transition between reference states will certainly distract the test-driver and reduce the realism, viz. the perceptual validity. Figure 3.11 on page 52 shows the outputs of two different linear filters. In case of motion cueing algorithms however, the filtered outputs are twice integrated and mixed afterwards. The second subfigure shows the resulting reference position that is comparable to that produced by the adaptive motion cueing algorithm. Plots of the results obtained with the original solution tend to mask the important effects behind effects induced by nonlinear elements and the simulator and perception dynamics included in the test-environment that was used with the development.

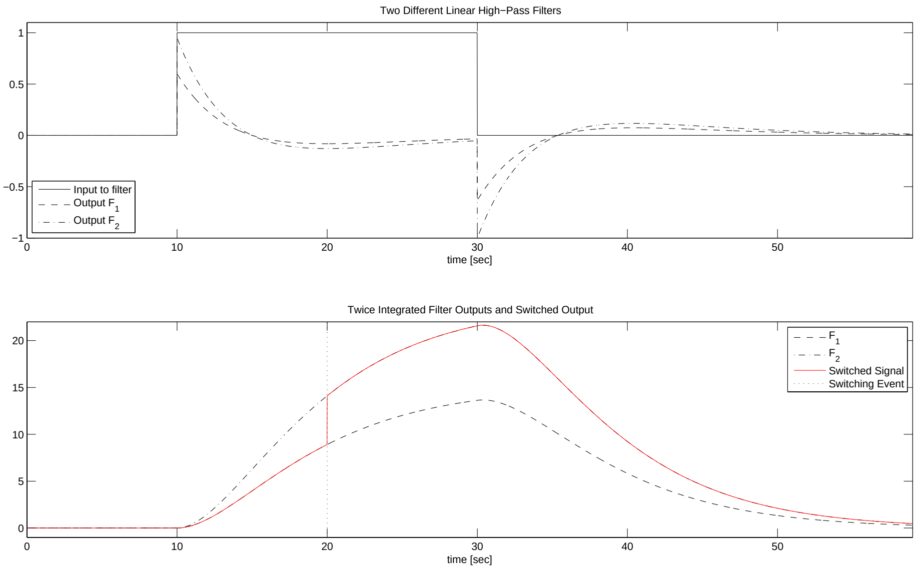


Figure 3.11: The first figure shows the output of two linear high-pass filters. Note that both signals show the zero-mean characteristic. In the case of the linear channels of the basic adaptive motion cueing algorithm, both filter outputs are twice integrated and mixed afterwards (second figure). The zero mean characteristic is also present in the switched position signal (red signal). This ensures that the platform returns to the neutral position. As the dynamics of the motion platform are slow, the jump will result in an uncontrolled transition between the reference positions. This is likely to be perceived by the driver.

One way to attenuate the severity of the switching is to shift the double integration towards reference positions in each sub-MCA behind the unit that calculates the output of the adaptive motion cueing algorithm, the mixing unit. Mixing accelerations instead of positions is reasonable since the hydraulic actuators are able to execute jumps in the accelerations[4]. Though, this is not possible for the rotational channels. The artificial motions created by the tilt-coordination are not integrated, and jumps between two different outputs will

[4]Increasing the pressure in the actuators within short time is possible, but changing the elongation of the actuator takes time for the hydraulic flow is limited.

always be passed directly to the driving simulator. This intrinsic problem also cannot be compensated by adding an additional low-pass filter, for this would certainly modulate the motion cues in a fashion decreasing the realism of the simulation (additional phase lag). Tajima et al. stated that especially the motion onsets are providing for a realistic driving simulation, and these would be attenuated by the low-pass filters [TMY06].

The switching between accelerations in the linear channel yields another drawback. The linear high-pass filters in the direct paths produce zero mean signals for inputs with a limited power. This ensures that the platform returns to the neutral position when no accelerations are to be rendered as long as the filter outputs are only integrated and not rate limited or modified in any nonlinear way. Mixing the accelerations and integrating the result destroys this characteristic. Although the accelerations produced by the sub-MCAs still equal zero after the input is finished, the mixed output shows an offset in the velocity after being integrated when the switch occured while the input was nonzero. This results from the different magnitudes of the two different filter outputs. Additional elements are necessary to compensate for this effect. Additional washout filters (section 2.1.3 and figure 2.1) are a reasonable possibility to obtain a movement that eventually drives the platform to the neutral position. Figure 3.12 on page 53 shows the obtained position signal.

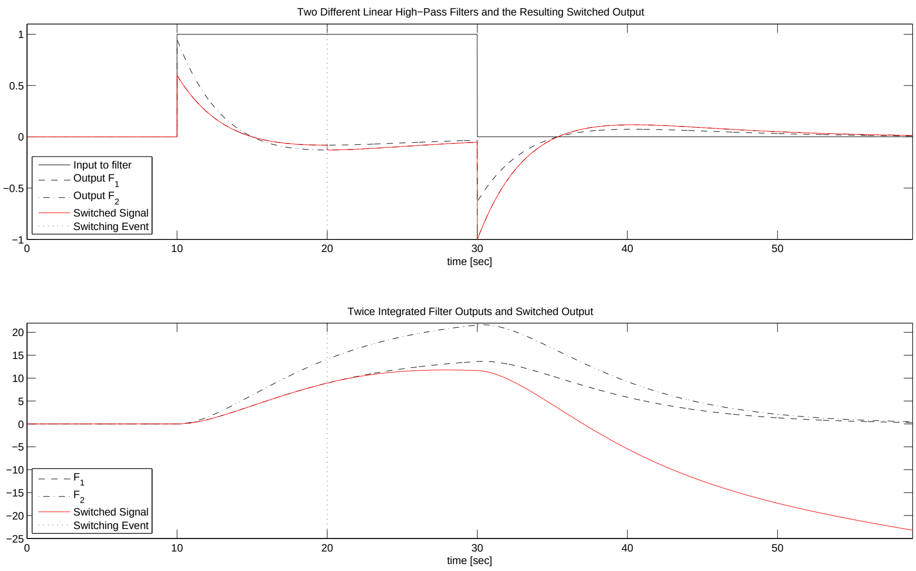


Figure 3.12: The first plot shows the switched output of two linear filters. The twice integrated signal (red signal, second plot) shows an ever decreasing position that results from the velocity offset induced by the switching while the input is nonzero.

Another way to eliminate the uncontrolled jumping behaviour, is to shift the simulator platform in a controlled way, with switching the reference value not within one sampling instant but rather in a controlled way using a short time.

3.2.2.1 Soft Switching

To obtain a smooth development of the weights, they are low-pass filtered after the conversion to floating point values. This slow switching is further on referred to as *soft-switching*, whereas the switching of reference values in one sampling instant is denoted *hard-switching*. It is important that the final weight, in this case the value one, is not changed in magnitude and reached after limited time, for the simulated drive shows fast dynamics. However, when a soft-switching is applied, the new MCA is not fully active when the new situation is encountered. So it is suggestive to start the switching between different sub-MCA outputs when no or just small motion cues have to be rendered. That is mostly the case some time before the new situation starts, so the switching should be initiated by a predicted driving situation. The prediction time also founds the basis for the design of the filters that form the soft-switched weights. Figure 3.13 shows the modificated *Ds-2-weights* module. The description of the original module is given in 3.2.1.3. Good results were obtained with a prediction time $t_{predict}$

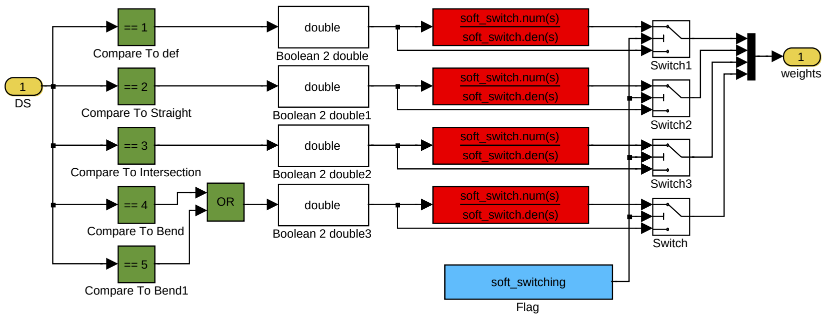


Figure 3.13: This figure shows the enhanced module that calculates the weights for the sub-motion cueing algorithms. Soft-switching is indicated by a flag. While soft-switching is used, the switching should not happen upon the current driving situation, but rather on the predicted one. See figure 3.9 for the original module.

of three seconds. One filter that showed satisfying results and was therefore applied to the adaptive motion cueing algorithm is:

$$G_{LP-softswitching} = G_{ss} = \frac{3}{s^2 + 3s + 3} \tag{3.1}$$

This filter obeys the demanded steady state amplification of one and reaches a steady value at a time short enough, figure 3.14 shows the step response of the filter. Figure 3.15 shows a plot of the filtered weights evolving in time. As the switching is now initiated by a predicted driving situation that leads by $t_{prediction} = 3[sec]$, also the switch at the end of the new situation starts three seconds ahead. This may lead to problems, for important accelerations are expected to happen at the end and the beginning of each new situation. Therefore, the corresponding motion cueing algorithm should be weighted fully in this cases. A more differentiated solution that analyses the driving situation properly is necessary. An approach to this problem is given in chapter 5. Figure 3.16 shows the reference position resulting from soft-switching the output of two

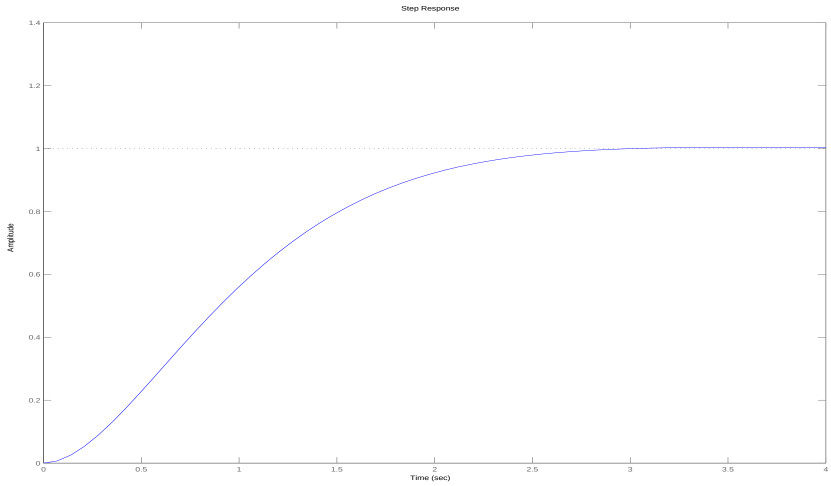


Figure 3.14: The step response of the linear soft-switching filter.

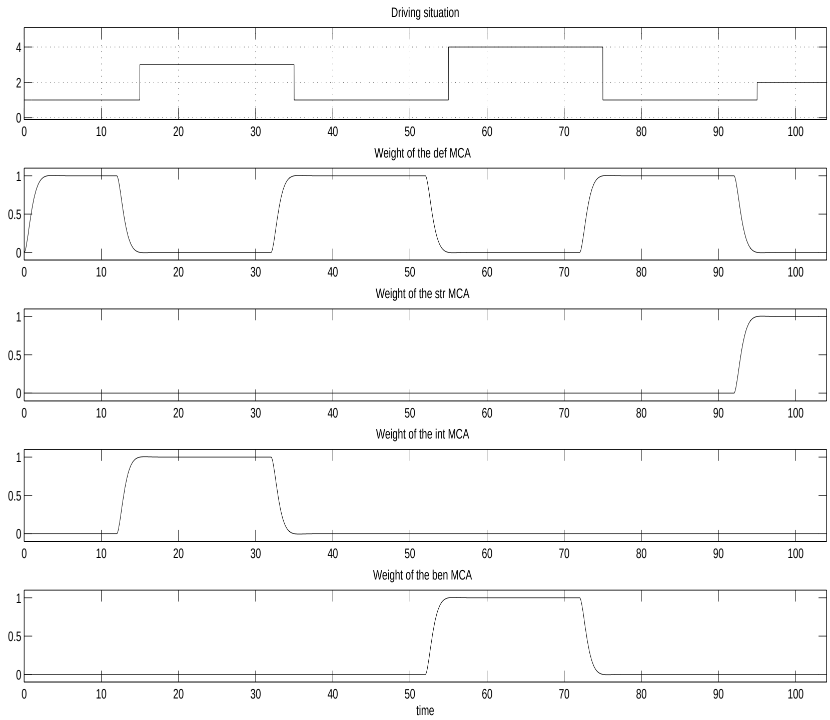


Figure 3.15: The weights for the sub-MCAs evolving in time. Note that the switching is initiated by a predicted driving situation to yield a full weighting when the first accelerations in the new situation are to be rendered. The switching could also be initiated by the current situation to ensure that the end of each situation is presented correctly.

linear high-pass filters. The simulator is able to follow the controlled transition between the two reference positions.

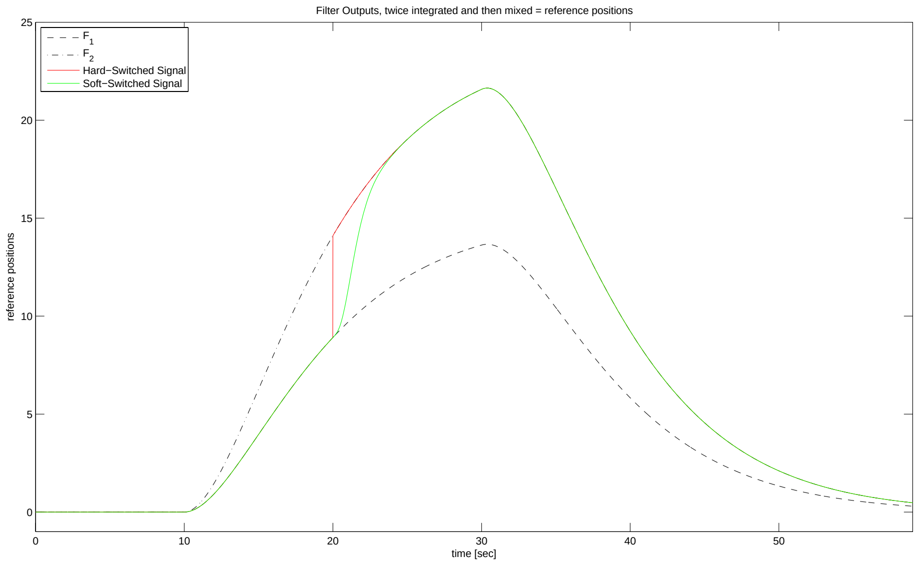


Figure 3.16: The reference position produced by a soft switching between the twice integrated results of two linear filters (green line). The transition can be seen as controlled for the platform should be able to follow the trajectory. However, the motion might be perceived as unnatural by the driver. The return to the neutral position is preserved.

3.3 Timevariant Filters in the Motion Cueing Algorithm

The structurally changing adaptive motion cueing concept applies several sub-MCAs to produce simultaneously reference positions and angles. This parallel design results in a computational complexity that increases linear with the number of different switching signal values and the resulting number of sub-MCAs. Although the mathematical problem to be solved is not NP-hard[5], it is suggestive to examine possibilities to reduce the computational cost.

The concept of the time-varaint motion-cueing algorithm is very similar to the classical approach. The design of the filter structure and other elements like scaling blocks and rate limiters was adopted. The evident difference of the approch presented in this section is founded by the time-variant nature of some of the coefficients. These are in detail:

- The parameters of the linear filters are varied by changing the coefficients

[5]exponential growth of computational complexity

a_i, b_j in the transfer functions online:

$$G_{High/LowPass} = \frac{a_n s^n + ... + a_1 s + a_0}{b_m s^m + ... + b_1 s + b_0} \text{ with } n \leq m \tag{3.2}$$

This results in changing break frequencies.

- The parameters of the scaling blocks are also varied.

The following sections present the submodules of an adaptive motion cueing concept that uses time-variant coefficients. Figure 3.17 shows the Simulink realization.

3.3.1 Dynamic Filtering

The implementation of the time varying MCA is very similar to the classical concept. The three combined acceleration signals are splitted up and seperately filtered with linear filters and passed to the output. The linear filters however are not initialized with static parameters at the start of the simulation but rather dynamically filled with coefficients. These values are provided by a subunit corresponding to the switching signal, the driving situation. They are combined with the acceleration in each channel and passed to the dynamic filters, that are implemented as C-MEX s-functions provided by Matlab/Simulink. The initial values for the transfer funtions are specified according to the default parameter-set. Figure 3.18 shows the Simulink implementation of the low-pass filter used with the tilt coordination. The other filters were implemented analogously.

3.3.2 Determining the Filter Coefficients

This block is a part of the time-variant motion cueing algorithm and determines the filter parameters and the scaling coefficients. The values are chosen from a set of constants with a multiport-switch for each of the scaling and filtering blocks. There exists a parameter set for each of the driving situations defined in section 3.2 on page 45. The switching signal governs the function of the switches.

The analysis of the structurally changing motion cueing algorithm showed that switches may result in jumps of the output. To compensate for such jumping signals that might drive the simulator into an uncontrolled transition between reference positions, it is suggestive to implement a *soft-switching* for the determination of the filter coefficients. The time-varying scaling values and filter coefficients are passed through a low pass filter located in the *determining the filter coefficients module*

$$G_{LP-softswitching} = G_{ss} = \frac{3}{s^2 + 3s + 3} \tag{3.3}$$

when a flag indicates an active soft-switching. As in the case of the structurally changing adaptive motion cueing algorithm the switch should no longer be initiated according to the current driving situation but rather by a predicted one. A drawback is that the switch at the end of a situation is also initiated by this predicted driving situation. Hence, it is suggestive to use an advanced switching signal (see chapter 5).

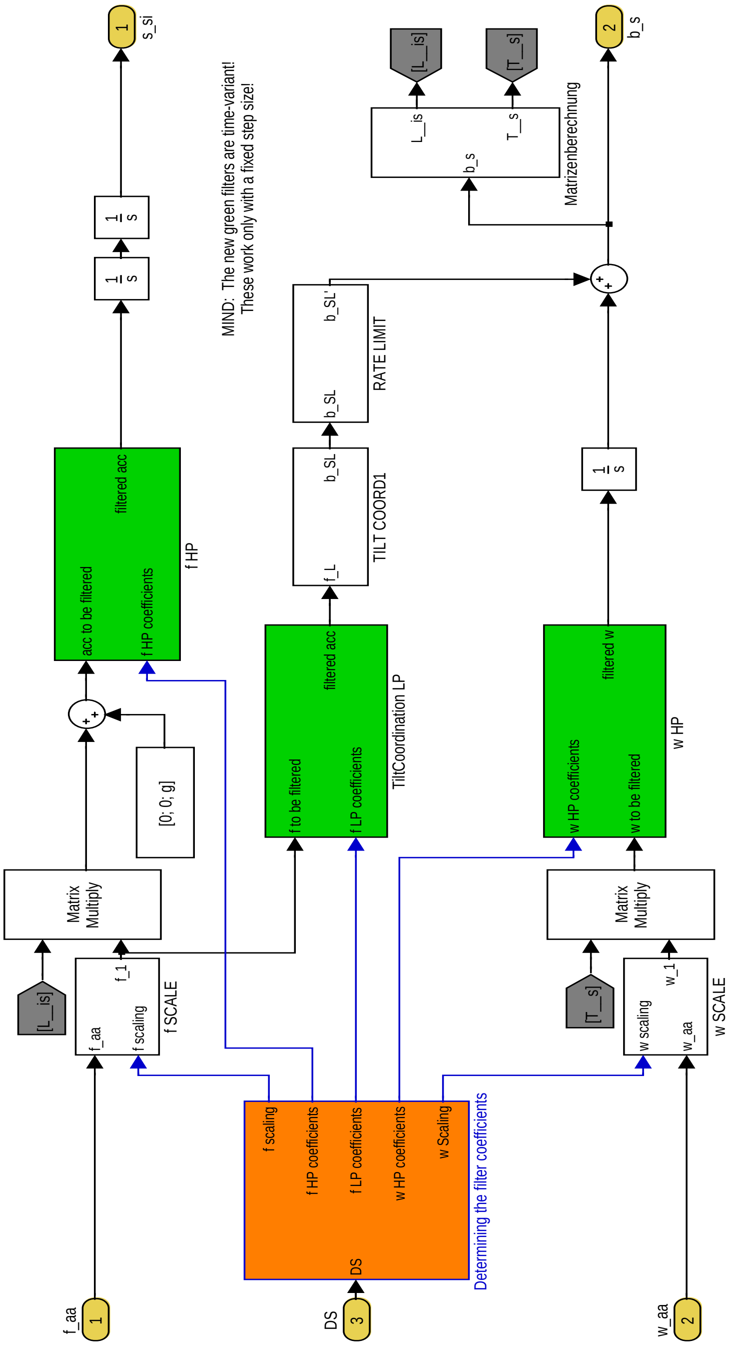


Figure 3.17: The time-variant motion cueing algorithm. The structure is smilar to that of the classical concept. Differences result from the constituting coefficients that are no longer static and have to be passed to the corresponding subsystems. This is achieved with the block *Determining the Filter Coefficients* - given in orange.

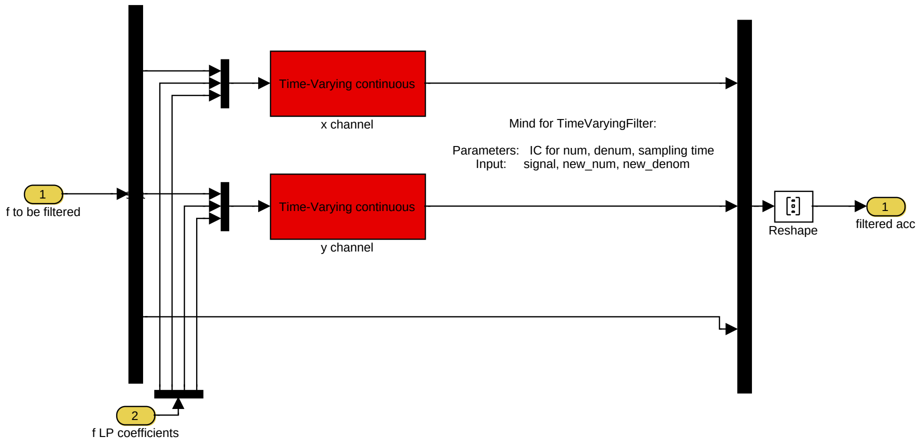


Figure 3.18: The simulink plan of the time-variant low-pass filter used with the tilt-coordination. The heave movements are not filtered, since they are cancelled out in the succeeding blocks.

3.3.3 Dynamic Scaling

The dynamic scaling modules are very similar to the mixing units of the structural changing motion cueing algorithm (see 3.2.1.2). The three acceleration signals in the rotational and linear channels are splitted and individually multiplied with a scaling coefficient. These coefficients are delivered by the unit that determines also the filter parameters upon the current or predicted driving situation.

3.3.4 Discussion

The adaptive motion cueing concept with a variable structure obeys a great computational complexity, for several sub-MCAs that are among other elements constituted of ordinary differential equations have to be solved in real time. The solution presented in this section uses the same number of dynamic systems as the classical concept, these are filled with time-varying coefficients. The computational cost is significantly reduced.

The time-variant filters yield the same outputs as the structural changing MCA with the integrators shifted behind the mixing module, since the mixing-unit is implicitly included in the filters. This is also the reason for the main drawback of the solution. Mixing accelerations destroys their zero-mean characteristic. The resulting assymetric acceleration yields a velocity offset and an increasing position when switches occured while accelerations had to be rendered and the platform was biased from its neutral position. Figure 3.19 *(c)* shows an exemplatory resulting position offset. The positions and angles have to be passed through an additional washout unit to compensate for this effect.

Another disadvantage is the lack of a variable structure, although one can compensate for this drawback with a bloated motion cueing concept that allows several different forms of use. This is not suggestive since one of the main

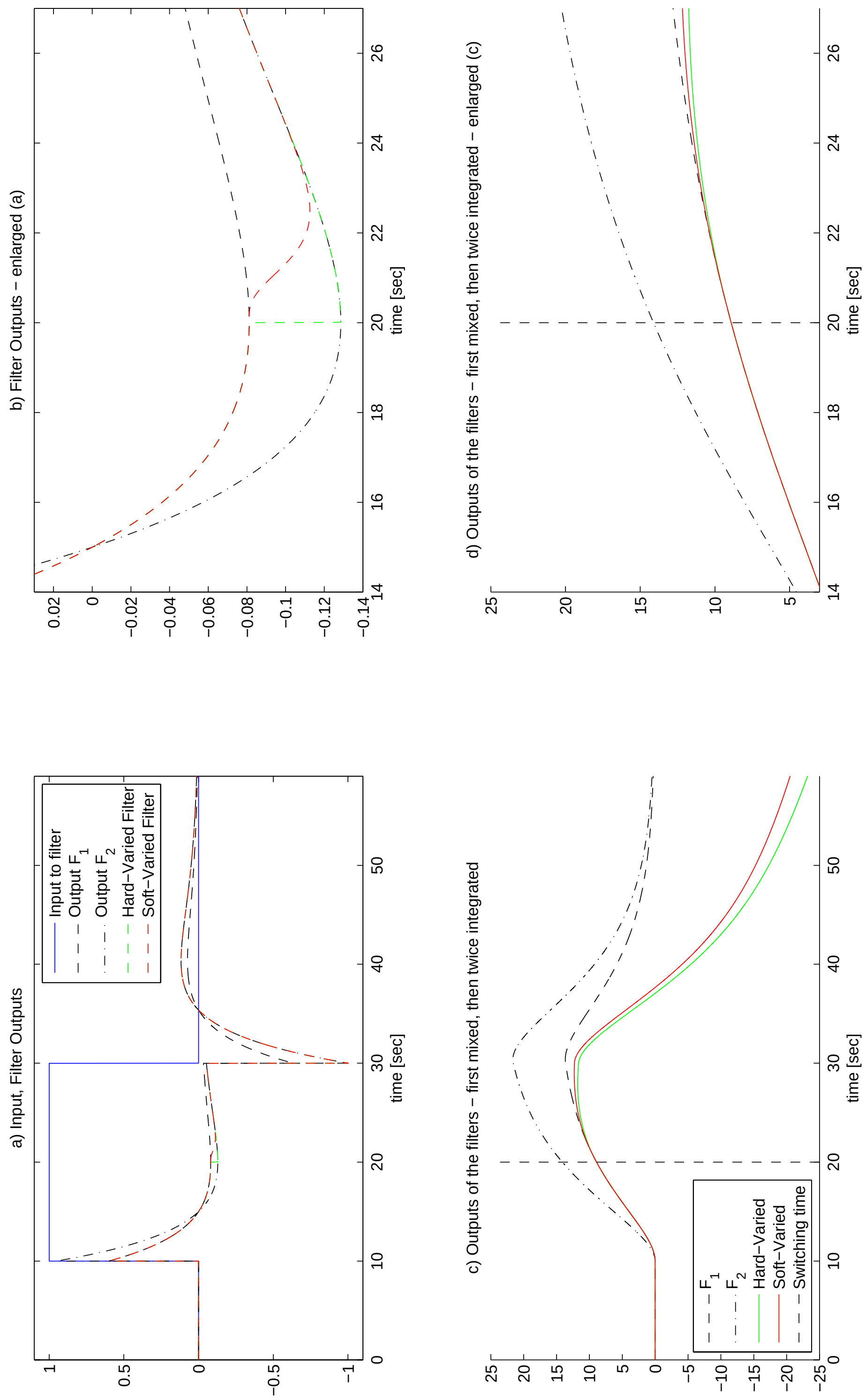


Figure 3.19: Output of two linear filters and the output of a time-variant filter (green = normal switch of the coefficients, red = soft-switch of the cefficients). The green and red line are identical except around the switching time $t = 20$ seconds. A position offset remains when the switch ocurred while the input was non-zero. This offset is smaller in case of the soft switching since the output signal that is integrated stays longer at the pre-switch filter-ouput. The assymetricity is less severe although the overall effect is lethal.

reasons for the use of the time varying MCA was the reduced computational complexity.

Chapter 4

Prepositioning

Some of the motions rendered with the platform are almost unidirectional with respect to the linear degrees of freedom. This is for instance the case for high-frequency lateral and longitudinal accelerations. The simulator will then leave its neutral position – the center of the motion envelope – and move to either the right or left side to present these accelerations and return to the origin of the motion envelope while rendering the end of the input signal.

The main obstacle when presenting motion cues are the limited capabilities of the platform. The next sections present a method to attenuate the limitations and provide for a better rendering of the motion cues. Since the lateral accelerations ocuring during a drive are especially important for the controllability of the vehicle, this part of the work concentrates on these motion cues although all results are equally valid for the longitudinal degree of freedom.

4.1 Concept

Certain driving situations produce movements of the simulator platform that are unidirectional in one linear degree of freedom and often also located near the limitations of the motion envelope. This is for instance the case in the aboved mentioned short term lateral accelerations. These occur predominantly in lane change manouvres and similar driving tasks. Vice versa, a long sustained right turn will result in a travel towards the right border of the motion envelope that is washed out by the feedback of the tilt angle after some seconds. In situations like this one, the platform is using the displacement capabilities of the whole motion envelope, for the end of the acceleration to be simulated produces an inverse movement. Figures 4.1 ,4.2 ,4.3 and 4.4 show the simulated movement of the platform for artifical lateral input signals and a classical motion cueing algorithm. As the duration of the acceleration to be simulated increases, the afore unidirectional movement becomes more bidirectional. The same observations can be made for the x/pitch-movement that presents braking and accelerating.

Since it is possible to predict the motions that are likely to be rendered in the next few seconds, it is suggestive to start some of the upcomming movements not at the center but rather at some point in the motion envelope in the direction that will not be used predominantly. This way, the motion envelope is *virtually* enlarged in the direction invers to the displacement. This displacement is denoted *prepositioning*, as it should occur prior to the expected motions. As the limitations in the expected direction are not that servere as before, it is possible to make use of the larger motion capabilities with a modified motion cueing algorithm. Modifications can include a less smaller scaling that would result in larger amplitudes of the motion or different break frequencies in the filters constituting the MCA. The latter changes are likely to yield a more realistic presentation of the occuring accelerations, for a greater amount of linear motions might be rendered directly and the use of the tilt-coordination that introduces perception delays is attenuated. Thus, it might also be reasonable to use a special MCA-structure or at least a parameter set for the motion cueing algorithm that takes the enhanced capabilities regarding the displacement of the cabin into account.

But there are some restrictions to the use of a prepositioning device. It is only

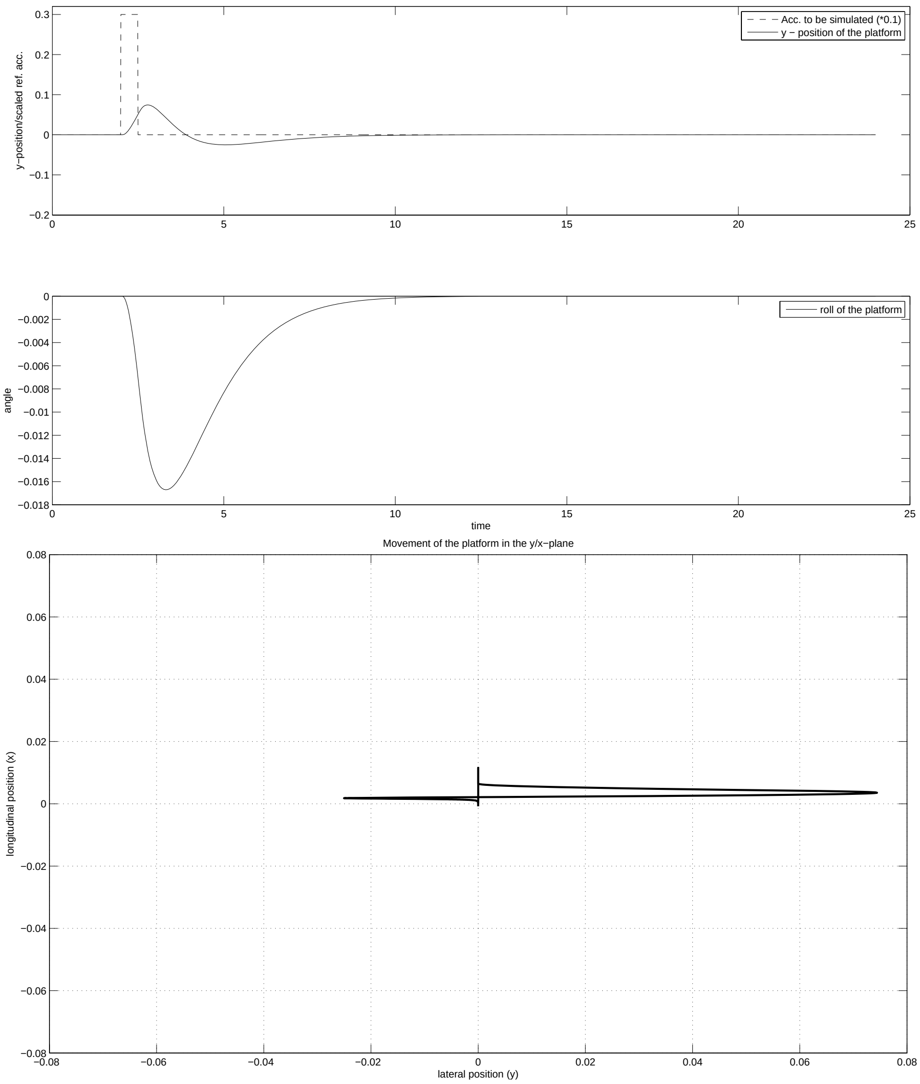


Figure 4.1: Motion of the platform for an input acceleration of $t_{sim} = 0.5[sec]$

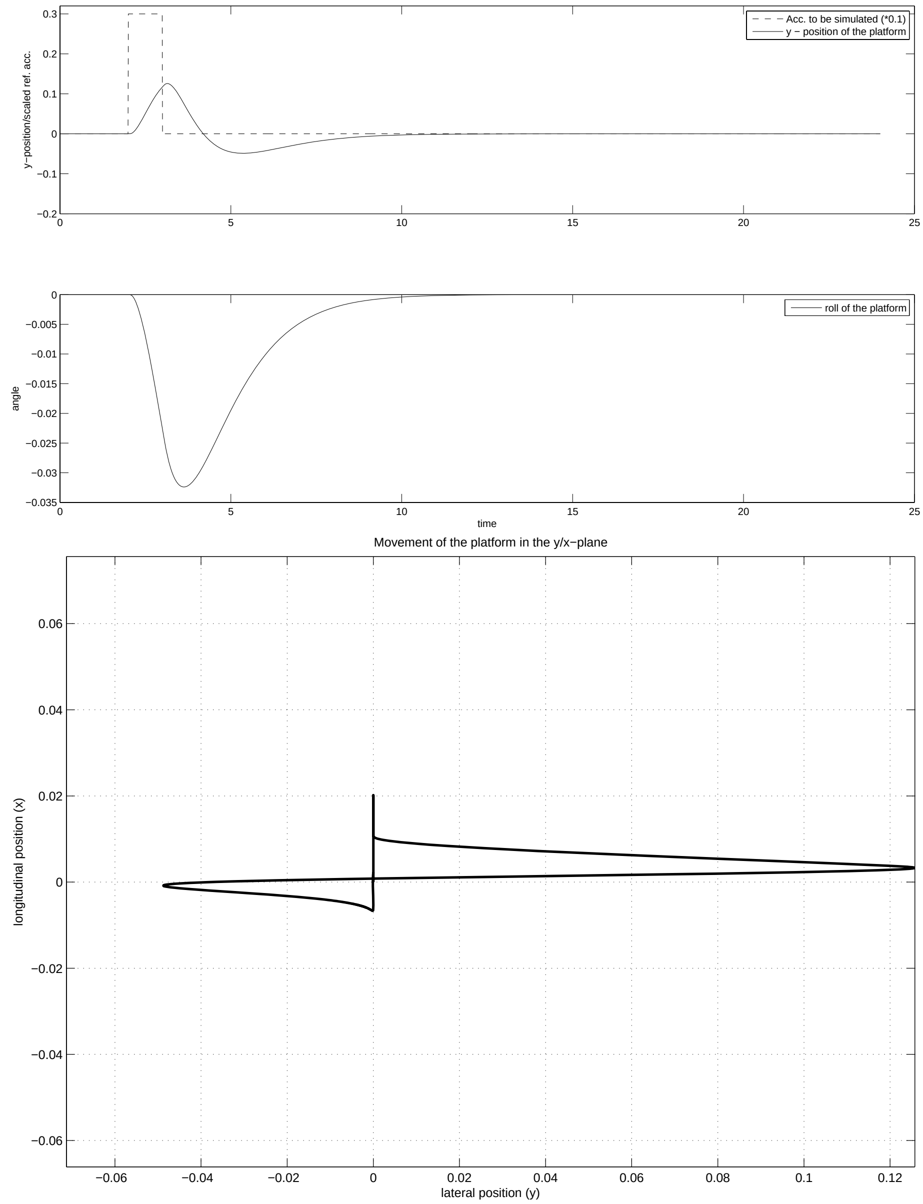


Figure 4.2: Motion of the platform for an input acceleration of $t_{sim} = 1[sec]$

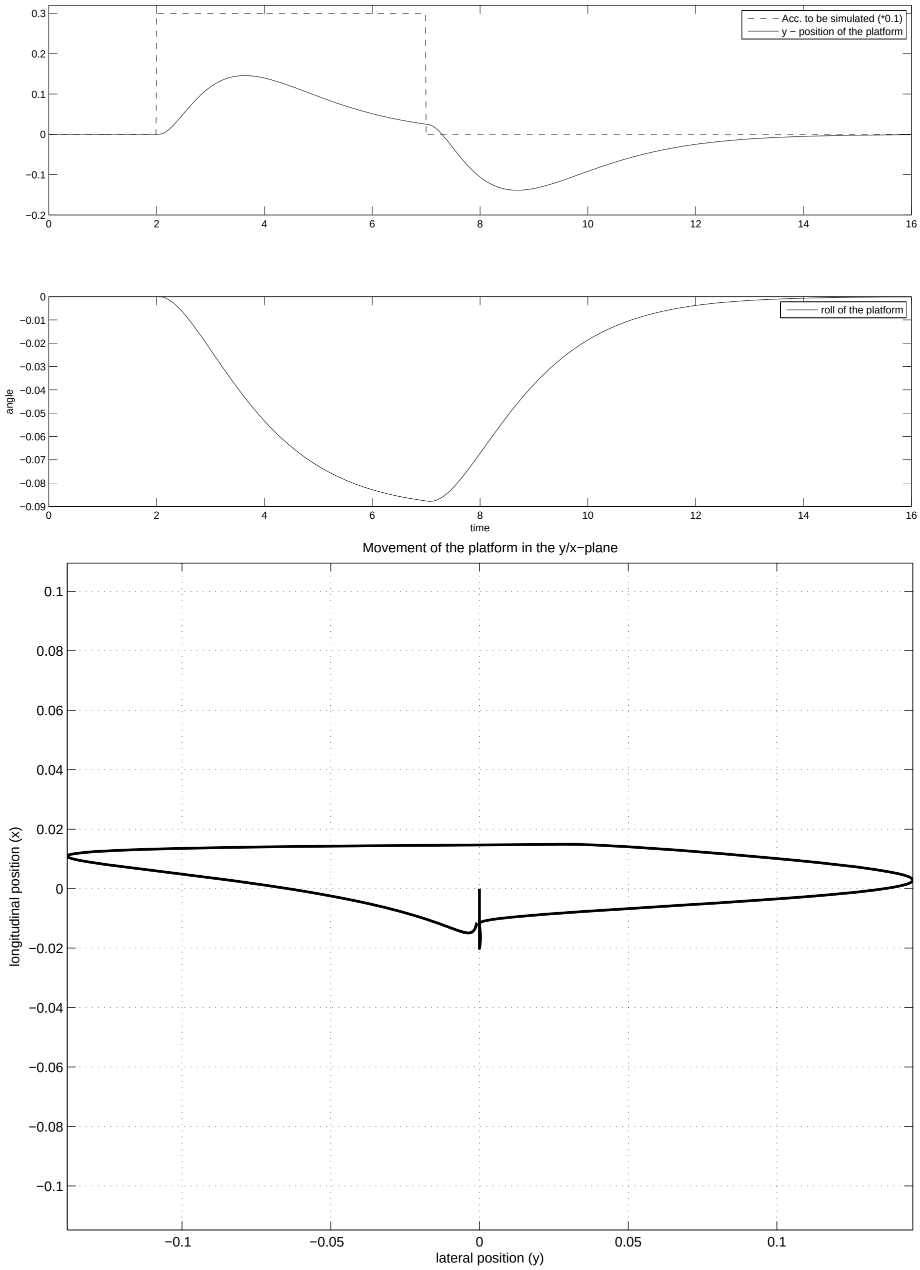


Figure 4.3: Motion of the platform for an input acceleration of $t_{sim} = 5[sec]$

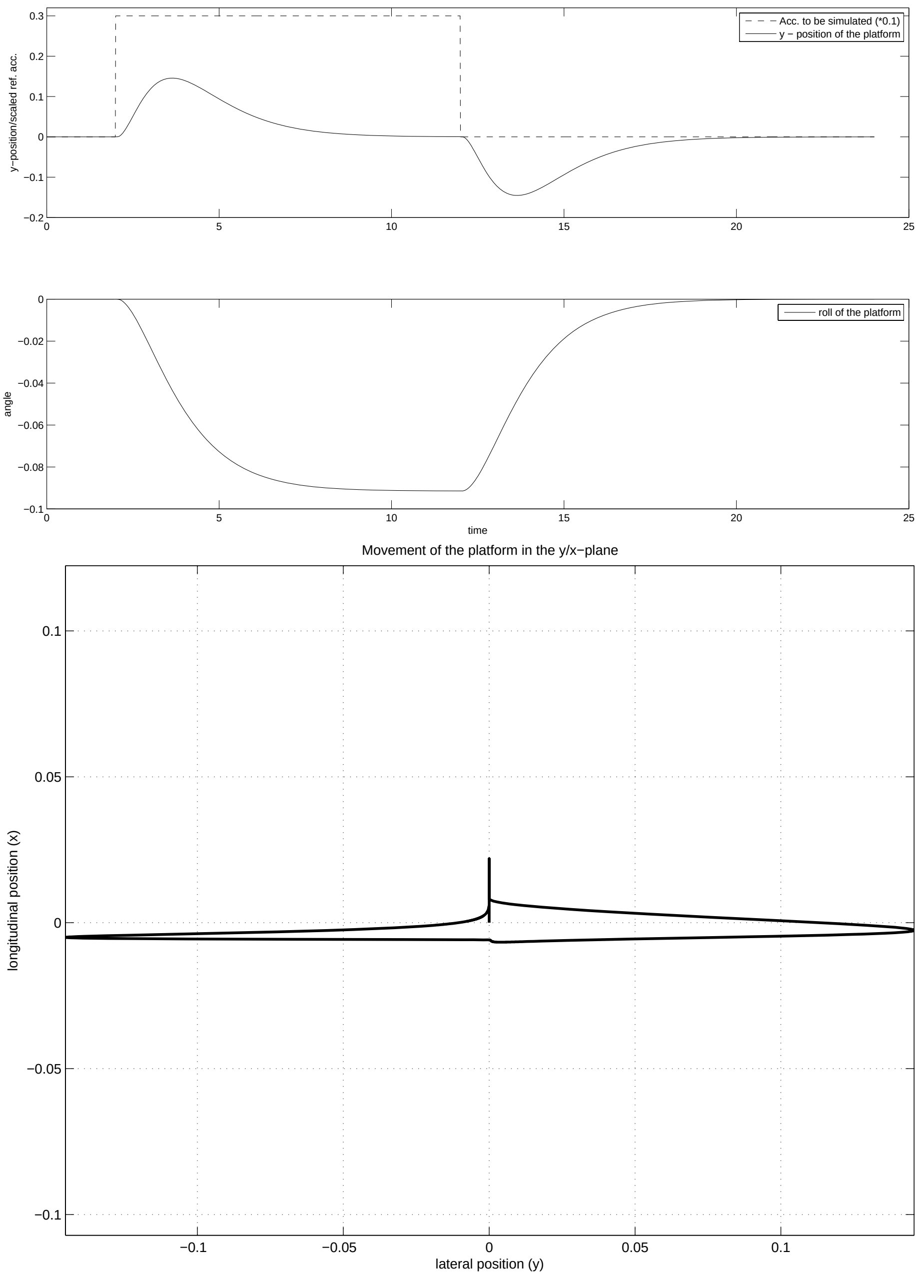


Figure 4.4: Motion of the platform for an input acceleration of $t_{sim} = 10[sec]$

applicable when no movements in the direction of prior displacement are to be expected. This is crucial since the motion cueing algorithm is mainly composed of linear elements, the filters. Using a different scaling or filter design that results in more realistic cues will require an enlarged motion envelope in all directions that are covered with the movements of the vehicle. Hence, motions in the direction of prepositioning are lethal. They will drive the simulator into its limitations, for the motion envelope is reduced in this area. Restrictions are also to be expected in the directions of travel orthogonal to the prepositioning directing, for the motion envelope resembles a ball in rough approximation. A lateral displacement away from the neutral position yields always a smaller displacement capability in the longitudinal and heave channel and vice versa. However, less severe limitations are only to be expected when the platform travels near the neutral position while the original motion cues are rendered after the simulator was prepositioned, for the displacement capabilities are largest at this point.

At the end of the driving situation that initiated the prepositioning, the simulator has to return to the former neutral position and all modifications to the motion cueing algorithm that were due to the different capabilities have to be reversed. When necessary, the simulator might enter a new prepositioning position. This is the case while driving on a two-laned highway. While travelling on the left lane, movements to the right are likely and the simulator might be displaced to the left prior to these movements. Vice-versa, travelling on the right lane could initiate a displacement to the right, since the next lane change will occur to the left.

4.2 Implementation

The prepositioning unit is divided into three subunits for both the lateral and longitudinal direction. These are an input processing logic and two alternative modules that eventually produce the new reference position for the simulator. Figure 4.5 gives the Simulink implementation.

Inputs to the lateral and longitudinal path of the unit are triggers that are activated when a prepositioning in this direction should occur and a flag that indicates the direction of prepositioning. The solutions presented in this section allow for two prepositioning reference positions[1] in each horizontal degree of freedom. Surge and sway prepositioning works analogously.

A logical unit was implemented that produces a boolean one at the first port when the trigger is active and a prepositioning in the positive direction should be produced. The second port obtains a boolean one when the trigger is active and a prepositioning to the other (negative) side should be produced. These two outputs cannot be used directly for the new reference positions for the simulator, for an uncontrolled motion would result from the jumping value. Therefore, both signals are fed forward into the two alternative subsystems that produce a motion that can be rendered with the driving platform. They are described in the following sections. As they are to be seen as alternatives that both show assets as well as disadvantages, the signal that is passed to the output

[1]Prepositioning to the left or right side and to the front or rear.

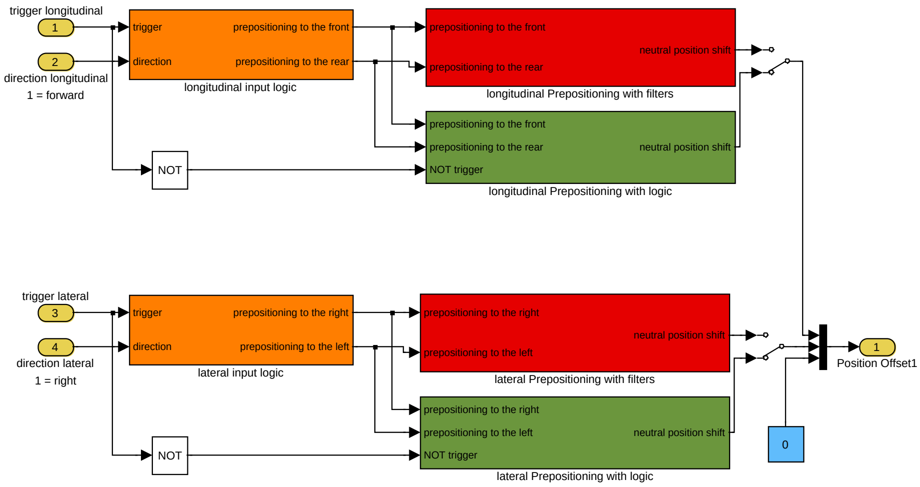


Figure 4.5: The prepositioning module.

can be chosen with a switch. The resulting position offsets from both the surge and sway path are subsequently added to the linear movements produced by the motion cueing concept. This way both the MCA and the prepositioning modules work in parallel.

4.2.1 Filtering a New Prepositioning Reference Position

The two boolean values produced by the input logic can be seen as starting points for the new reference positions. They produce a positive edge when the simulator should reach an artificial neutral position and a negative one when it should return to the original center of the motion envelope. These values cannot be used directly, since adding them to the positions resulting from the MCA would yield an uncontrolled transition. The travel to the new reference position has to happen in a controlled way that is not perceived by the driver. Especially the onset and finish of the artificial movement are expected to disturb the driver, and should therefore take place as soft as possible. Hence, the edge-like changes in the reference position are low-pass filtered and multiplied with the value for the total lateral prepositioning displacement[2]. Figure 4.6 shows the Simulink implementation.

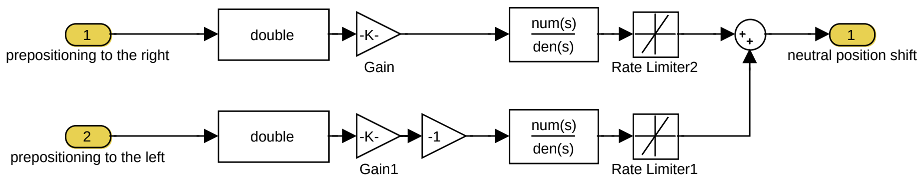


Figure 4.6: The module that filters the reference positions produced by the input logic.

[2]Prepositioning displacements of 30 to 70 cm might be reasonable.

4.2.2 Filters Used with the Prepositioning Unit

The filters used with the prepositioning unit determine the transition between the original center of the motion envelope and the artificial neutral position. The main constraint that hampers the design of the filters is the motion perception of the driver, for the resulting motion should not be detected by the driver.

Crucial design parameters for the filters are:

- The time available to reach the desired position within the workspace.
- The shape of the step-response of the filter.

The available time is determined by the time the driving situation predictor looks ahead. The shape of the step response of the filter[3] is primarily governed by the order of the filter and the time constants used. The latter values are also determined by the available time and the filter order.

Two sets of filters were developed for the prediction time being three and five seconds, respectively. To attain a prepositioning motion that is not perceivable, the whole transitioning motion has to be sufficiently smooth. Especially the onset and finish of the displacement have to be non-recognizable. As the reference-position offset is calculated and humans sense accelerations, the resulting movement has to be at least smooth in the second time-derivative of the position. Therefore filters with an order greater than two are applied. Significant changes in the step response are observed up to the fifth order. Hence, each developed filter set contains filters from second to fifth order. The filters given in equations 4.1 to 4.8 were obtained by concatening linear first order filters, using the Matlab file given in the appendix (A).

$$G_{32} = \frac{1}{0.36s^2 + 1.2s + 1} \tag{4.1}$$

$$G_{33} = \frac{1}{0.1016s^3 + 0.6533s^2 + 1.4s + 1} \tag{4.2}$$

$$G_{34} = \frac{1}{0.01978s^4 + 0.2109s^3 + 0.8437s^2 + 1.5s + 1} \tag{4.3}$$

$$G_{35} = \frac{1}{0.003355s^5 + 0.05243s^4 + 0.3277s^3 + 1.024s^2 + 1.6s + 1} \tag{4.4}$$

$$G_{52} = \frac{1}{1.103s^2 + 2.1s + 1} \tag{4.5}$$

$$G_{53} = \frac{1}{0.4506s^3 + 1.763s^2 + 2.3s + 1} \tag{4.6}$$

$$G_{54} = \frac{1}{0.1526s^4 + 0.9766s^3 + 2.344s^2 + 2.5s + 1} \tag{4.7}$$

$$G_{55} = \frac{1}{0.04592s^5 + 0.4252s^4 + 1.575s^3 + 2.916s^2 + 2.7s + 1} \tag{4.8}$$

The optimization objective during the design process was to reach a maximum possible displacement in the given prediction time without producing perceivable motions, all filters reach at least 95% of the desired output within the prediction

[3] The step response is the position offset that is later added to the MCA output.

time. Filters of increasing order differed mainly in the resulting motion onset and finish.

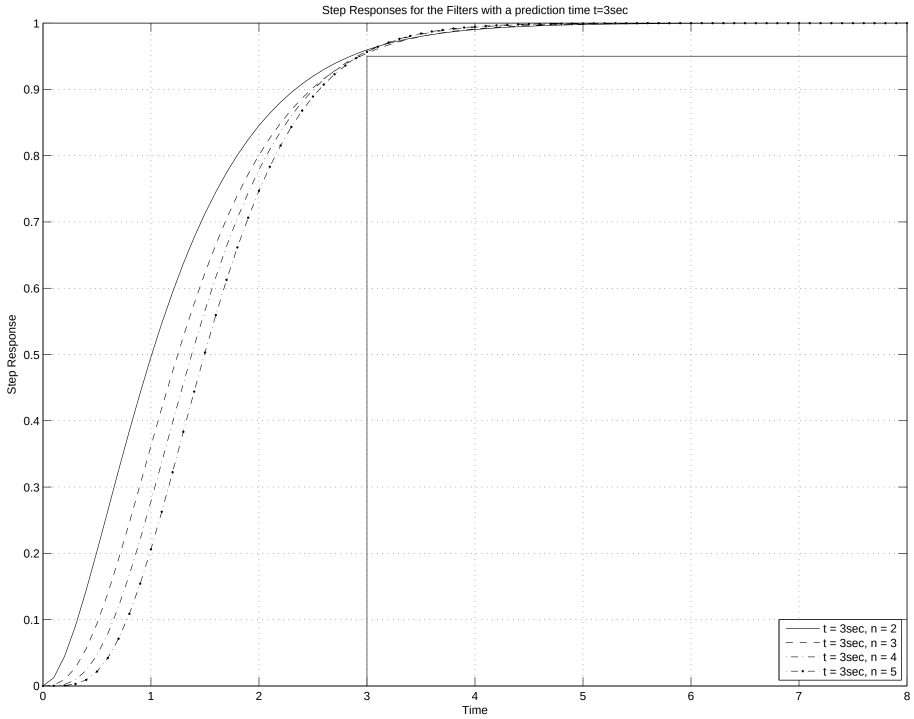


Figure 4.7: The step responses for filters of different order n and a prediction time $t = 3$ sec.

Figures 4.7 and 4.8 show the horizontal position for different filters plotted over time. Since linear filters are used, both directions of travel are produced in the same fashion.

The over-all objective of the filter included in the prepositioning unit is to yield a displacement within the motion envelope that is not perceived by the driver. Simulations with the test-environment showed that the main cause of motion perception is not the onset or finish of the movement, but the slope of the displacement in the middle of the prepositioning. This is essentially the speed of travelling. To reduce this velocity, it is necessary to reach the maximum velocity in rather short time. Table 4.1 shows the used filters and gives the maximum force perceived during the displacement. As the test were conducted with the test environment that uses a model for the motion perception, these values can only give small insight to the useability of the filters in real driving simulations. At least the filters performing sufficiently good (all three seconds filters for a displacement of 30 centimeters and all five seconds filters for a displacement of 30 to 50 centimeters) should be tested with the driving simulator. Filters of lower order are suggestive since higher order filters need more time to reach an output with the desired velocity. A second problem is the slow finish of the movement.

The results with the filters given in equations 4.1 to 4.8 were not satisfactory, for

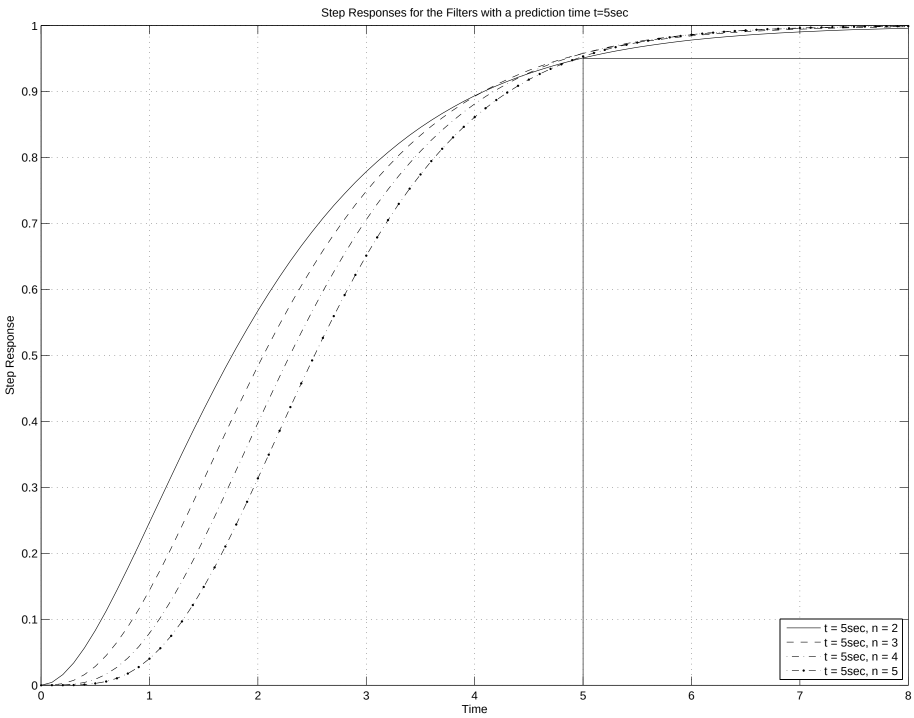


Figure 4.8: The step responses for filters of different order n and a prediction time $t = 5$ sec.

Prediction Time	Displacement	Filter Order	max. overshoot	max perceived fy
3s	30cm	2		0.1336
3s	30cm	3		0.1114
3s	30cm	4		0.1300
3s	30cm	5		0.1416
3s	50cm	2		0.4277
3s	50cm	3		0.3562
3s	50cm	4		0.3530
3s	50cm	5		0.3520
3s	70cm	2		0.6272
3s	70cm	3		0.5124
3s	70cm	4		0.5030
3s	70cm	5		0.5051
5s	30cm	2		0
5s	30cm	3		0
5s	30cm	4		0
5s	30cm	5		0
5s	50cm	2		0.1306
5s	50cm	3		0.1227
5s	50cm	4		0.1230
5s	50cm	5		0.1211
5s	70cm	2		0.2651
5s	70cm	3		0.2293
5s	70cm	4		0.2164
5s	70cm	5		0.2084
3s	30cm	2	1%	0
3s	30cm	2	3%	0
3s	30cm	2	5%	0
3s	50cm	2	1%	0.2804
3s	50cm	2	3%	0.2402
3s	50cm	2	5%	0.1518
3s	70cm	2	1%	0.4271
3s	70cm	2	3%	0.3719
3s	70cm	2	5%	0.2807

Table 4.1: Results of the test of different filters for the prepositioning unit. The filters are either concatenations of linear first order filters or second order filters with a damping ratio smaller than one.

the test-environment showed perceived forces while the simulator was displaced. Linear second order filters (equations 4.9 to 4.11) that use a damping ratio smaller than one were developed.

$$G_{1\%} = \frac{1}{0.7056s^2 + 1.3944s + 1} \tag{4.9}$$

$$G_{3\%} = \frac{1}{0.9025s^2 + 1.4250s + 1} \tag{4.10}$$

$$G_{5\%} = \frac{1}{1.0816s^2 + 1.9968s + 1} \tag{4.11}$$

An overshoot in the position offset is the consequence (see figure 4.9). This drawback might be tolerable as long as it is not to high, since the perceived motions should be significantly reduced with the use of these filters. The maximal traveling velocity is reduced while the onset and finish are still shallow enough. Table 4.1 gives the results of the test of the overshooting filters. An overshooting smaller than 3% is expected to yield good results with the real driving simualtor[4]. Displacements up to 50 centimeters might be tested with the corresponding filters (equations 4.9 and 4.10)

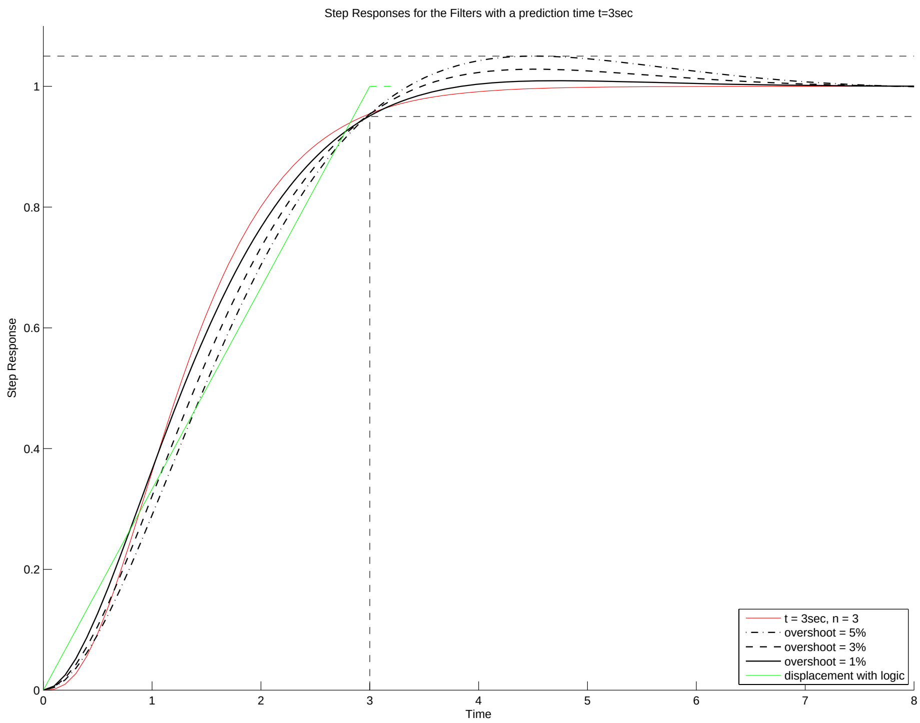


Figure 4.9: Step responses of the filters with a damping ratio smaller than one. The displacement resulting from the logical unit presented in section 4.2.3 is also showed (green line).

[4]A reasonable limit has to be found in tests with the simulation platform.

4.2.3 Establishing a New Prepositioning Reference Position with a Logical Subunit

The second approach allots a constant velocity for the travel to the deviated neutral position and the return to the original one. This is reasonable since the concepts presented in this chapter are validated with a test-environment for MCAs that contains a model for the human motion perception. This model emphasizes the drawbacks introduced by a high velocity regarding the perception of the displacement. Thus, it is suggestive to travel with the least speed possible. The new neutral position is incrementally reached by integrating this velocity over time. Two approaches were used to determine the constant velocity:

- The first approach works similar to the one Richter used for the tilt-coordination in his motion cueing concept [Ric71]. The constant velocity equals a (static) value resembling the motion detection threshold and therefore claims to guarantee that no motion is perceived. Though, this is a rather spongous argument towards the use of this method since the thresholds for motion perception are neither equal for all humans nor static (see section 1.5). An additional disadvantage is the time of travelling, that might be larger than the prediction time that is used to initiate the movement. The new reference position might be reached to late.

- The second approach produces a reference position out of a constant travelling velocity. This velocity is calculated as the ratio of the prepositioning deviation y_{prepos} over the prediction time $t_{predict}$.

  $$v_{prepos} = \frac{y_{prepos}}{t_{predict}} \tag{4.12}$$

  The advantage of this method is that the new neutral position is reached after $t = t_{predict}$.

The latter solution is used with the final logical subsystem, since it is important that the prepositioning movement is finished before any motion cues are presented. Otherwise, these motion cues might harm the motion envelope for it is not yet as large as assumed while tuning a correponding MCA.

If the driver perceives a movement while the prepositioning reference position is changed, either the prediction time has to be enlarged or the prepositioning deviation y_{prepos} has to be reduced.

A logical unit is used to shift the neutral position to the right or to the left side by adding a position offset to the MCA's output. This offset might be located on the left or right side, or in case of the surge channel at the front or in the back of the motion envelope. Figure 4.10 shows the Simulink plan. Travelling to the left for instance has to occur when

1. the new reference prepositioning position is located on the left, or
2. a return from the right deviated position should be carried out.

In the first case, the constant velocity is assigned to the integrator when a prepositioning to the left should happen (first input port active) and the position

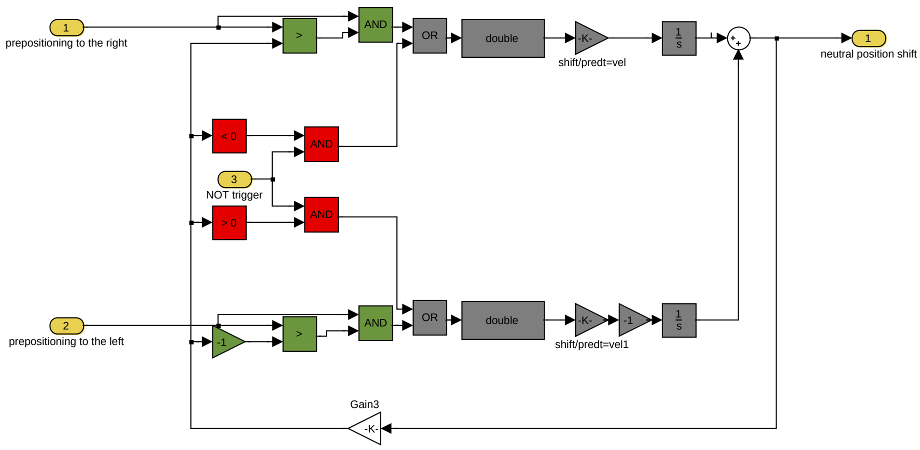


Figure 4.10: Prepositioning - Logical Unit

is not reached yet. This is observed with the dark green blocks in figure 4.10. In the latter case, the constant velocity is assigned to the integrator when the prepositioning is not active and the original neutral position is not yet reached (achieved with the red blocks in figure 4.10). Travelling in the inverse direction works analogously.

Prediction Time	Displacement with logic	max perceived fy
3	30cm	0
3	50cm	0.2997*
3	70cm	0.5521*
5	30cm	0
5	50cm	0
5	70cm	0.0639
Stars indicate that the value is only reached for a short time at the end of the perceived motion		

Table 4.2: Results of the test of different filters for the prepositioning unit.

The simulations with the logical unit generally yield good results, for perceived forces occur only for short time instants (see table 4.2). The motion onset and finish however is likely to be perceived by the driver. The validity of the results has to be shown with tests using the real driving platform.

4.3 Discussion

The reason for the use of a prepositioning unit is the enlarged motion envelope in the direction the next motions are expected. The development of the prepositioning device was accomplished with the help of the test environment for motion cueing algorithms (see section 1.3.1 and [Fis05]). As it contains a model for the human motion perception and the major constraint for the use of a prepositioning is the non-detectability of the motion, the assessment of the two different concepts depends strongly on the applied motion perception model. This is a significant drawback, since the model used with the test-environment is rather basic although similar to state-of-the-art solutions.

The filtering concept shows assets regarding the onset and finish of the motion whereas the logical unit bears the advantage of the lowest possible travelling velocity. The method finally used with the driving simulator has to be worked out in test focusing on the perception of the artificial motion.

4.3.1 General Suggestions for the Use of the Positioning Method

It is not suggestive that the prepositioning is used together with long sustained accelerations, for these produce motions not only in one direction but in inverse directions. In the currently used classical MCA, the tilt-coordination and other rotations are fed back into the direct linear rendering path through a transformation matrix (see figure 3.6 on page 48). This feed-back ultimately results in a washout of the linear travel (see figure 4.4) and an inverse movement at the end of the sustained acceleration. Thus, it is not suggestive to shift the motion envelope in case of these accelerations being predicted. The corresponding driving situation is that of sustained turns. A work around could be achieved with a missing feed-back of the rotational movement. In this case, the platform would leave the neutral position in one direction and stay there until the inverse linear movement appears when the acceleration is finished. Figure 4.11 shows the resulting displacement for an artificial sustained acceleration that has to be rendered.

Though, the prepositioning can be used with driving situations that merely produce unidirectional motions. This is predominantly the case with rather short as to say high frequency motions. These are mainly rendered with linear movements of the platform, and these motions could be enhanced with an improved displacement capacity. Driving situations that favor such accelerations are:

- Driving on highways. Lateral accelerations on straight roads are expected mainly for lane-change manouvres. The resulting accelerations are of high-frequency nature and generally exceed the limits of the platform, for the lateral travel of at least three meters during this manoeuvre is larger than the displacement capability. In case of the DLR dynamic driving simulator, it is limited to approximately one meter. A lateral displacement prior to the manouevre of the cabin yields less severe limitations in this case albeight a down-scaling would still be necessary[5]. Thus, driving on the right lane should result in a shift of the neutral position to the right, whereas driving on the left lane for long times increases the probability of a lane change to the right side and an a-priori displacement to the left side of the motion envelope is suggestive.

- Braking and accelerating. The probability of high-frequency braking manouvres increases

[5]The down-scaling is not only necessary to match the limitations of the platform. Large amplitudes like the real ones are perceived as unnaturally high when rendered with a platform, thus a reasonable down scaling is always suggestive.

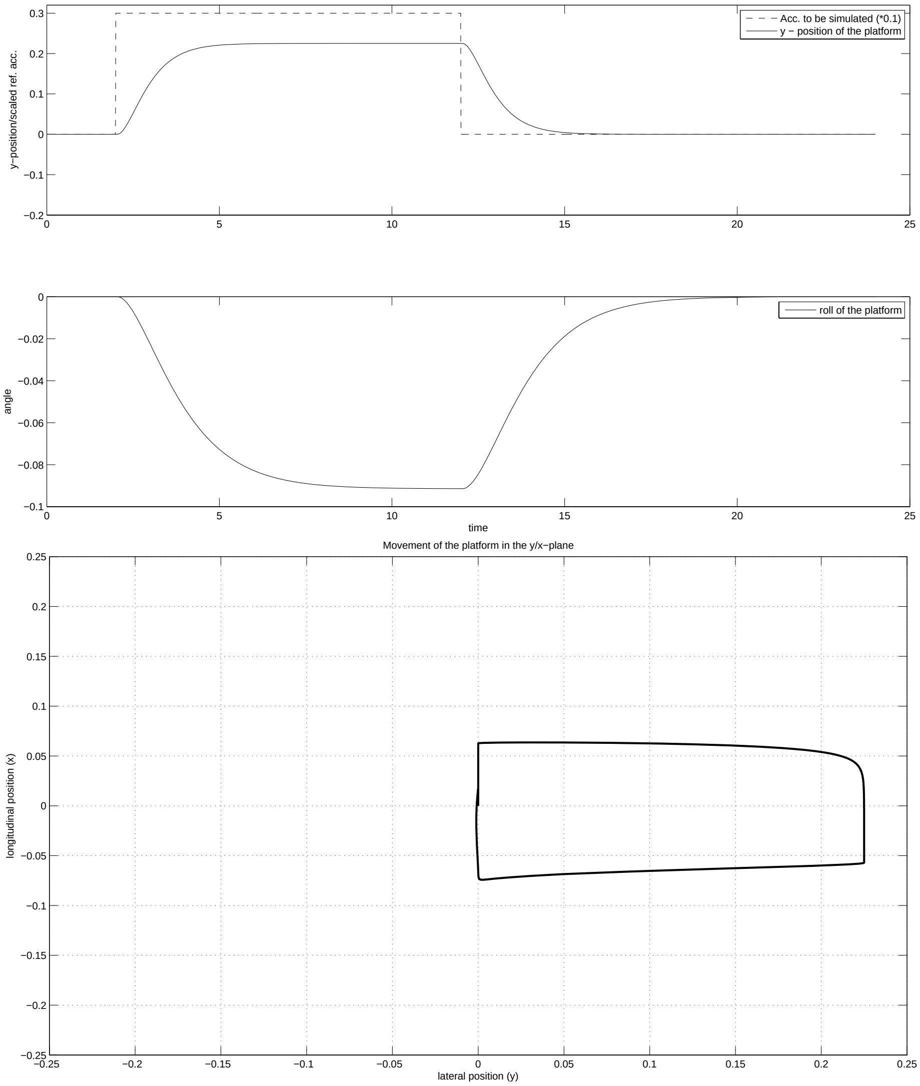


Figure 4.11: The same input signal as in figure 4.4 produces no longer a bidirectional travel but an unidirectional one.

- while driving on roads with many other vehicles.
- on a road with a great curvature, for turns are driven at a lower speed.
- when the vehicle travels at a speed near or above the speed limit.

Vice versa it is more likely that accelerations occur when the speed is low or a turn is near its finish. The simulator cabin could be shifted to the front or to the rear of the motion envelope in these cases.

The evident problem in the cases mentioned above is the huge amount of information that is necessary for a reasonable prepositioning of the simulator. As long as this information is available, this displacement yields a larger *virtual workspace* that might render a more realistic presentation of motion cues possible.

Chapter 5

Perception of the Switching and Prepositioning Motion

To obtain a MCA that reacts according to the driving task, a device that analyses the current and future situation is necessary. Switching between different motion cueing concepts and parameter-sets may produce jumps between different reference positions and angles resulting from different sub-motion cueing algorithms. These jumps are likely to result in an uncontrolled transition between two platform states. Although they can be compensated with a controlled switching between two states that takes some time to be finished, it is especially the 'distance' between two concurring sets of reference states that complicates the switching. It is important that the MCA is already adapted to the comming situation when the first cues are to be rendered. Otherwise, false cues would result from the switch.

5.1 Concept

The main task of the driving situation analyser is to extract information on the current and future driving situation and process it to make it suitable for the adaptive motion cueing algorithms and the prepositioning device. The desired output of the module under consideration is denoted *active situation*. Only changes that might be rendered with the platform without producing false cues should be passed onto the following modules. Hence, it is suggestive to minimize the distance between the state previous to the switch and the one afterwards. This is likely to be achieved when

- the switch is executed while the simulator is in its neutral position and no motions are to be rendered
- one of the active driving situations between that is switched is the default one that covers all driving situations that might occur (see 3.2 for a characterization of driving situations).

Vice versa, switches in the driving situation that are likely to produce false cues are to be eliminated or at least attenuated. Switching between different concepts that are tuned towards a good rendering of a special motion-type are not suggestive, for a distance between two simulator states then occurs not only in one or two but rather in all degrees of freedom. Thus, these switches have to be eliminated at least as long as the simulator is deviated from the neutral position.

Switching towards a special motion cueing concept should be finished when new motion cues are to be presented. Though, the reverse switch to the default concept should only start when the special driving situation is finished. A switch within a turn for instance is likely to be recognized by the driver.

5.2 Implementation

The future and current driving situations are used as inputs with the driving situation analyser. Switching towards the new situation should happen upon a change in the predicted future situation whereas the reverse switch is initiated by an ongoing situation change. This switching behaviour ensures that the special motion cueing algorithm is present for the whole non-default driving situation.

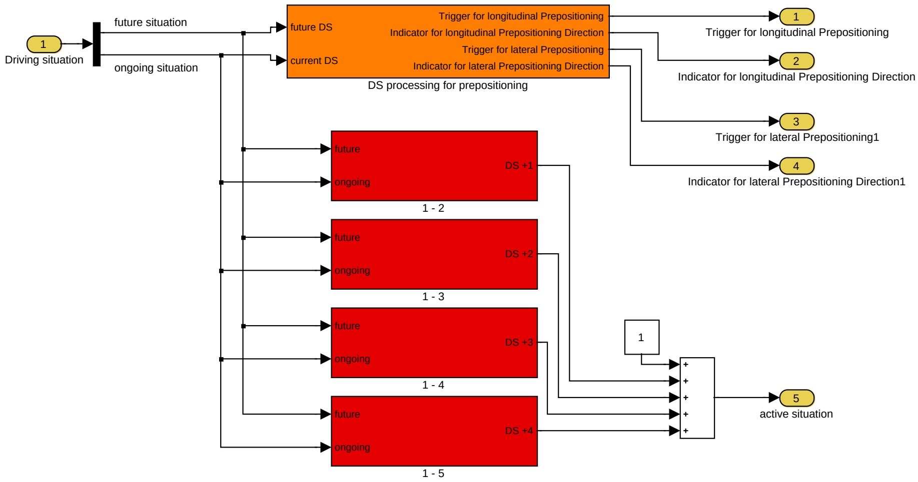


Figure 5.1: The Driving Situation Analyser. It consists of the processing unit for the prepositioning device (orange) and the parts calculating the active situation (red blocks and summation). The content of block (1-3) is presented as an example in figure 5.2

The analysing module (figure 5.1) inhabits four submodules for each special situation described in section 3.2. Each of these subsystems detects changes from the ongoing default situation denoted *1* to the special situations denoted *2* to *5*. They also detect when the special situation is neither currently present nor predicted. The two comparisons produce boolean values that are passed to a *relay* block that toggles from on to off when a switch towards the special situation is detected and maintains this state as long as the situation is no longer present in the current *and* predicted driving situation channel. The output of the relay blocks is used to calculate the active situation. As long as no relay in the modules (1-2) to (1-5) is in the active state, the default situation denoted *1* is passed as a constant through the summation block to the output of the driving situation analyser. When one of the relays is active, it passes the value denoting the special situation decreased by one to the subunit's output. This value is added to the default value of one. This way, the situation analyser produces an active situation different from the default one only when the switch happened directly from default to non-default and not when it resulted from a change between different special situations. It is assumed that each driving situation last at least as long as the predictor looks ahead. Figure 5.2 gives the Simulink plan for the unit that compares the driving situation to the *intersection* situation, block *1-3* in figure 5.2.

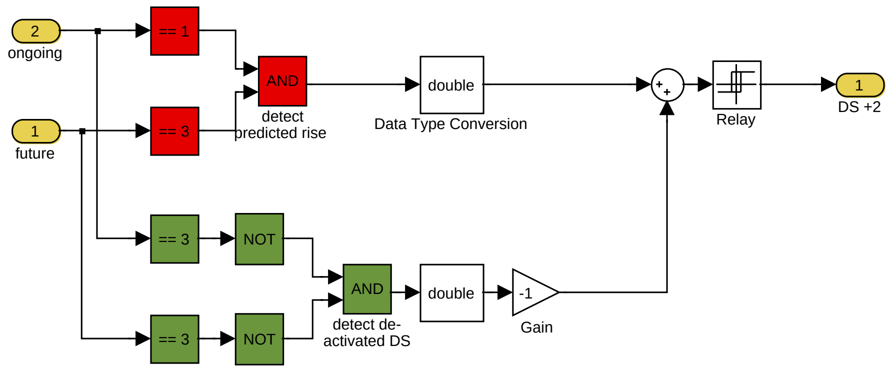


Figure 5.2: The submodule (1-3) of the driving situation analyser. The red blocks are used to detect a change from default to special situation whereas the green blocks detect that the special situation is neither active nor predicted. In this case of this special situation being detected, the activated relay passes a $3 - 1 = 2$ to the output. Vice-versa, a deactivated one yields a zero.

5.3 Discussion

The analysing module extracts switches between driving situations that are likely to produce no false cues. The resulting active situation is passed to the adaptive motion cueing algorithm that switches in a time-consuming controlled way to the new active situation. Figure 5.3 gives an exemplatory time evolution of the active driving situation.

A second component of the analyser forms trigger values for the prepositioning device. This module is denoted *DS processing for prepositioning* in figure 5.1. Though, the functionality could not be implemented for a reasonable prepositioning requires additional information on the state of the simulated vehicle, for instance

- the lane the vehicle travels in,
- its velocity and
- information on the speed limits.

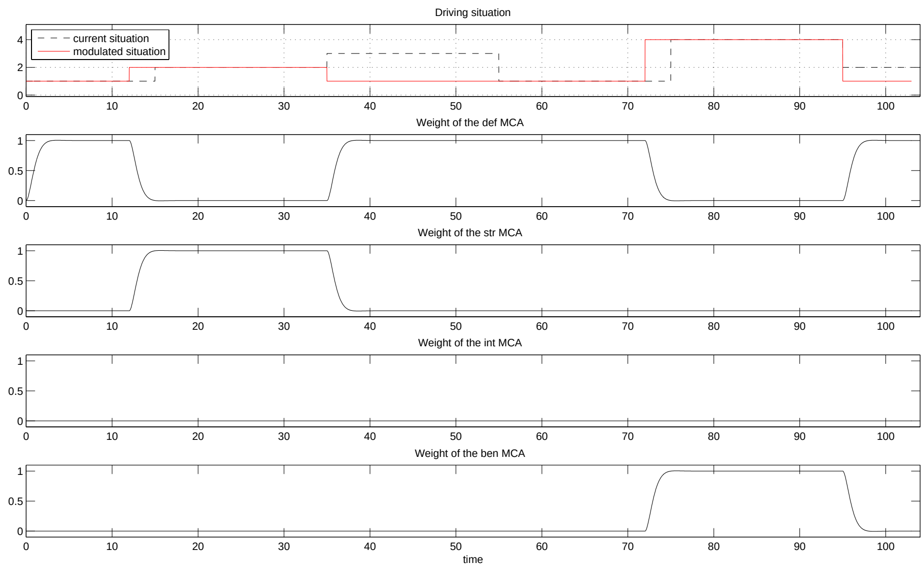


Figure 5.3: The weights produced by the driving situation analyser. Notice that the change from situation two to three is omitted. Shifting away from the default situation is initiated by the predicted switching signal whereas the back switch occurs only when the non-default situation is finished.

Chapter 6

Summary and Perspectives

The rendering of accelerations occuring during drives on motion platforms still faces problems regarding the realsim of the simulation. The most intuitive solution to a problem involving hybrid systems is a solution based on switching and structural changes, in our case a structurally changing motion cueing algorithm that changes its behaviour depending on discrete phenomena like the driving situation (chapter 3). It bears several advantages although it produces a significant amount of differential equations that have to be solved in real time. The major asset of the presented solution results from the use of different subsystems. These only have to be equivalent regarding their inputs and outputs, in this case positions and angles. Different approaches presented in chapter 2 and even completely new concepts may be used together in a single test-drive. Nonlinear filters [RK00] might be a usefull tool, especially when these are used together with the developed prepositioning unit.

The pragmatic use of the software developed in this thesis is to apply different parameter-sets for the same motion cueing structure. As this approach results in parallel static motion cueing algorithms that have to be solved at the same time, it produces a large computational cost that might be neglected with the computational power available today as long as the number of equations is not to high. It may be reduced by using only one motion drive algorithm with dynamic coefficients, for instance the scaling values or break frequencies of the filters (section 3.3). This approach implies the drawback that the return of the platform to the neutral position is no longer guaranteed. An additional washout module is necessary.

It is possible to mix both concepts to make use of the advantages of both solutions. As long as the subsystems of the switching MCA differ only in parameters and not in structure, they might be summarized in a single time-variant MCA. The dynamic motion cueing algorithm might then be combined through a switching device with other motion cueing concepts obeying different structures.

Unfortunately, it was not possible to test the developed software with a real simulator. Hence, the primary task to be accomplished in the future is the validation of the implemented modules with the driving simulator.
Especially the generation of false cues should be attenuated with the new concept, as it is possible to use motion cueing algorithms and parameters that suit the current driving situation better than solutions that were developed with respect to the whole drive. These take other situations and even worst-cases, that are not likely to occur, into account. The test-drivers should not be affected by motion-sickness that often anymore, and the validity of the tests should be increased.

The solutions given in this thesis migth be extended in different ways. The prepositioning that enlarges the motion envelope in the direction the next movements are expected to take place, might also be used to compensate for the large delay in the perception of lateral and longitudinal accelerations resulting from the use of tilt-coordination. It should be possible to predict especially the accelerations occuring in sustained turns from the driving velocity and the shape of a turn ahead (diameter and length). The simulation platform could be tilted in advance without violating the thresholds of motion perception to reduce the lag of the tilt-angle and enhance the presentation of sustained and mid-frequency motion cues.

Another problem that needs further attention is the switching that produces jumps in the outputs. These were attenuated by use of a soft-switching that gradually shifts from one output to another in reference to a modified switching signal. This solution however requires time before the second subsystem is completely active. The result might be a reduced realism of the simulation or even false cues. To abolish unnecessary restrictions on the use of the switched motion cueing algorithm (see chapter 5), it is suggestive to implement a mechanism that checks in advance whether a normal or soft switching will result in a suboptimal simulator motion. This module would predict the movements resulting from a switch while a standard motion is performed and facilitate the decision if the switch is to be performed or not.

Appendix A

Developing The Filters Used with the Prepositioning Unit

Different sets of filters were developed for the use with the prepositioning unit. This appendix gives the Matlab m-file used to develop the filters given in equations 4.1 to 4.8 and 4.9 to 4.11.

```
% M-file for the design of the preopsitioning filter
% C.Weiß - Do 23/11/2006
%
% The filters of different order should reach at least 95% of the desired
% output(tested with a step-response, thus min_amp = 0.95*1 = 0.95)

close all
clear all

timevector = linspace(0,8,81);
min_amp = 0.95;

response = zeros(84,1);
response(4:84) = timevector;


for predictiontime=[3]
    for n=[2,3,4,5]
        timeC = predictiontime;
        timeconstant = timeC/n;
        lowpass = tf([1],[timeconstant 1]);
        lp=(lowpass)^n;
        y = step(lp,timevector);
        while y((predictiontime*10)+1) <= min_amp
            timeC = timeC - 0.1;
            timeconstant = timeC/n;
            lowpass = tf([1],[timeconstant 1]);
            lp=(lowpass)^n;
            y = step(lp,timevector);
        end %of while
        response = [response , [predictiontime;n;timeconstant;y]];
    end %of for
end %of for


% shaping of a 2nd-order [n = 2,3,4,5] filter with minimum slope at
% Prediction_time/2, increase the timeconstant until the specifications
% are met
response2 = zeros(84,1);
response2(4:84) = timevector;
for predictiontime=[3]  % or [3,5]
    for n=[2] % or [2,3,4,5]
        timeconstant = 0.01;
        lowpass = tf([1],[timeconstant 1]);
        lp=(lowpass)^n;
```

```
        y = step(lp,timevector);
        while y((predictiontime*10)+1) >= min_amp
            response2 = [response2 , [predictiontime;n;timeconstant;y]];
            timeconstant = timeconstant + 0.01;
            lowpass = tf([1],[timeconstant 1]);
            lp=(lowpass)^n;
            y = step(lp,timevector);
        end %of while
        time_constant_for_third_shape = response2(3,size(response2,2));
    end %of for
end %of for

% % shaping of a 2nd-order [n = 2,3,4,5] filter with minimum slope at
% Prediction_time/2
% 1. decrease damping as long as overshooting not to large,
% then increase time constant as long as step-response of filter at
% t=prediction time > 95% of desired output; for sake of simplicity and
% computational reasons we start with the best filter from the last
% section.
response3 = zeros(86,1);
response4 = zeros(86,1);
response3(6:86) = timevector;
response4(6:86) = timevector;
start_damping = 1;
max_overshoot = [1.05,1.03,1.01];

for overshoot = max_overshoot
    overshoot
    for predictiontime=[3]
        for n=[2]
            d = start_damping;
            timeconstant = time_constant_for_third_shape;
            lp = tf([1],[timeconstant^2 2*d*timeconstant 1]);
            y = step(lp,timevector);
            while max(y) <= overshoot
                response3 = [response3 , [predictiontime;n;timeconstant;d;overshoot;y]];
                d = d - 0.01;
                lp_with_lower_d = tf([1],[timeconstant^2 2*d*timeconstant 1]);
                y = step(lp_with_lower_d,timevector);
            end %of while
            d = response3(4,size(response3,2));
            lp = tf([1],[timeconstant^2 2*d*timeconstant 1]);
            y = step(lp,timevector);
            while y((predictiontime*10)+1) >= min_amp
                response4 = [response4 , [predictiontime;n;timeconstant;d;overshoot;y]];
                timeconstant = timeconstant + 0.01;
                lp = tf([1],[timeconstant^2 2*d*timeconstant 1]);
                y = step(lp,timevector);
            end %of while
        end %of for
```

```
    end %of for
    disp(response4(1:5,size(response4,2)))
end %of for 'overshoot'
```

Bibliography

[BK05] Martin Brünger-Koch. Motion Parameter Tuning and Evaluation for the DLR Automotive Simulator. Driving Simulator Conference 2005, North America, Orlando, November 2005.

[Bra97] Michael S. Branicky. Hybrid Dynamical Systems, or HDS: The Ultimate Switching Experience. *Lecture Notes in Control and Information Sciences 222*, pages 1–12, 1997.

[DZfLuR] Institut für Verkehrsführung und Fahrzeugsteuerung Deutsches Zentrum füt Luft-und Raumfahrt. Dynamischer Fahrsimulator. URL: `http://www.dlr.de/fs/`.

[Enc] Wikipedia The Free Encyclopedia. Categories: Nervous system, vestibular system. URL: `http://en.wikipedia.org/wiki/Vestibular_system`.

[Fis05] Marc Fischer. Testumgebung für motion-cueing-konzepte im bereich fahrsimulation. Studienarbeit, Deutsches Zentrum für Luft- und Raumfahrt e.V., Institut für Verkehrsführung und Fahrzeugsteuerung, 2005.

[GAB+02] Peter Grant, Bruce Artz, Mike Blommer, Larry Cathey, and Jeff Greenberg. A Paired Comparison Study of Simulator Motion Drive Algorithms. *DSC2002 Paris*, September 2002.

[IS82] J. Ish-Shalom. Design of Optimal Motion for Flight Simulators. Ph.D. Thesis, 1982.

[KS72] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. Wiley-Interscience, 1972.

[Mer99] Jean-Pierre Merlet. *Parallel Robots*. Kluwer Academic Publishers, 1999.

[Mor97] A. Stephen Morse, editor. *Control using Logic-Based Switching*, volume 222 of *Lecture Notes in Control and Information Sciences*. Springer, 1997.

[NR90] Meyer A. Nahon and Lloyd D. Reid. Simulator motion-drive algorithms: A designer's perspective. *J. Guidance*, 13(2):356–362, March-April 1990.

[PGN98] Nicolas A. Pouliot, Clément M. Gosselin, and Meyer A. Nahon. Motion simulation capabilities of three-degree-of-freedom flight simulators. *Journal of Aircraft*, 35(1):9–17, January-February 1998.

[Ric71] Bernd Richter. Beitrag zum Problem der Beschleunigungssimulierung an Fahrsimulatoren. Dissertation, Fachbereich Verkehrswesen der Technischen Universität Berlin, 1971. Very good introduction to driving simulators!

[RK00] Gilles Reymond and Andras Kemeny. Motion Cueing in the Renault Driving Simulator. *Vehicle System Dynamics*, 34:249–259, 2000.

[RN85] L.D. Reid and M.A. Nahon. Flight Simulation Motion-Base Drive Algorithms: Part 1 - Developing and Testing the Equations. Technical Report UTIAS Report No. 296, University of Toronto, Institute for Aerospace Studies, 1985.

[RN86] L.D. Reid and M.A. Nahon. Flight Simulation Motion-Base Drive Algorithms: Part 2 - Selecting the System Parameters. Technical Report UTIAS Report No. 307, University of Toronto, Institute for Aerospace Studies, 1986.

[SG05] Zhendong Sun and Shuzi Sam Ge. *Switched Linear Systems: Control and Design*. Communications and Control Engineering. Springer, 2005.

[SISH82] R. Sivan, J. Ish-Shalom, and J-K. Huang. An Optimal Control Approach to the Design of Moving Flight Simulators. *IEEE Transactions on Systems, Man and Cybernetics*, SMC - 12(6):818–827, Nov./Dec. 1982.

[SP05] Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. John Wiley & Sons, 2005.

[Sui05] Rainer Suikat. The New Dynamic Driving Simulator at DLR. Driving Simulator Conference 2005, North America, Orlando, November 2005.

[TCH99] R.J. Telban, F.M. Cardullo, and J.A. Houck. Developments in Human Centered Cueing Algorithms for Control of Flight Simulator Motion Systems. *American Institute of Aeronautics and Astronautics*, (AIAA-99-4328), 1999.

[Tel02] Rober J. Telban. A Nonlinear Motion Cueing Algorithm with a Human Perception Model. Technical report, Department of Mechanical Engineering, State University of New York at Binghamton, 2002.

[TMY06] Jun Tajima, Kouhei Maruyama, and Naohiro Yuhara. Driving Task Adaptive Motion-Cueing Algorithm for Driving Simulators. *DSC Asia/Pacific 2006 - Tsubuka*, May/June 2006.