A Study on State-of-the-Art Motion Cueing Algorithms applied to Planar Motion with Pure Lateral Acceleration — Comparison, Auto-Tuning and Subjective Evaluation on a KUKA Robocoaster Serial Ride Simulator

von der Fakultät für Ingenieurwissenschaften, Abteilung Maschinenbau und Verfahrenstechnik der

Universität Duisburg-Essen

zur Erlangung des akademischen Grades

eines

Doktors der Ingenieurwissenschaften

Dr.-Ing.

genehmigte Dissertation

von

Duc An Pham

aus

Hanoi, Vietnam

Referent: Prof. Dr.-Ing. Andrés Kecskeméthy

Korreferent: Prof. Dr.-Ing. habil. Christoph Woernle

Tag der mündlichen Prüfung: 22. 11. 2017

Acknowledgment

Writing this dissertation has had a big impact on me. I would like to reflect on the people who have supported and helped me so much throughout this period.

Firstly, I would like to express my sincere gratitude to my advisor Prof. Dr.-Ing. Andrés Kecskeméthy for the continuous support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this dissertation.

Furthermore, I would like to thank my friend Philippe Ferreira who supported me greatly and were always willing to help me.

These acknowledgments would not be complete without mentioning my research colleagues: M.Sc. Sebastian R ottgermann and Dr.-Ing. Francisco Geu. It was a great pleasure working with them and I appreciate their ideas, help.

My deepest appreciation belongs to my family for their sacrifices and understanding. They always encourage me during the period of study.

Finally, there are my friends who are as my second family in the foreign country.

Thank you very much, everyone!

Duisburg, in 12 2017 Duc An Pham

In the dissertation, the following conversions are applied

- Vectors are described by underlined letters, e.g. $\underline{a}$
- Matrices are described by upper–case letters, e.g. $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$
- A vector from origin of frame $\mathcal{K}_X$ to origin of $\mathcal{K}_Y$ is denoted as ${}_X\underline{r}_Y$
- A decomposition of a vector $\underline{r}$ in a reference frame $\mathcal{K}_X$ is denoted as ${}^X\underline{r}$
- A rotation matrix from the reference frame $\mathcal{K}_X$ to the reference frame $\mathcal{K}_Y$ is denoted as ${}^X\mathbf{R}_Y$
- Decomposition of a relative acceleration of frame $\mathcal{K}_Y$ with respect to frame $\mathcal{K}_X$ in frame $\mathcal{K}_A$ is denoted as ${}^A_X\underline{a}_Y$
- A vector from point M to point N is denoted as ${}_M\underline{r}_N$
- Unless stated otherwise, the coordinate system at driver's head is defined such that the X axis is to the right, the Y axis is backward, and the Z axis is upward
- Unless stated otherwise, vectors are decomposed in the inertial, earth–fixed reference frame $\mathcal{K}_0$.
- The reference frame $\mathcal{K}_X$ has the origin X

VECTORS AND MATRICES (unless otherwise noted in the dissertation)

$\underline{a}$	Acceleration of a point
$\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$	Matrices representing the state-space model of a control system
$\underline{\xi} = [R, \alpha, z]$	Position vector of a point in cylindrical coordinates
$\dot{\underline{\xi}} = \left[\dot{R},\, \dot{\alpha},\, \dot{z}\right]$	Velocity vector of a point in cylindrical coordinates
$\ddot{\underline{\xi}} = \left[\ddot{R},\, \ddot{\alpha},\, \ddot{z}\right]$	Acceleration vector of a point in cylindrical coordinates
$\mathbf{C}(\xi)$	Transformation matrix that transforms the vector components from Cartesian to cylindrical coordinates
$\underline{S} = [x,\, y,\, z]^{\,T}$	Position of the cabin of the driving simulator (m)
$\underline{v} = [\dot{x},\, \dot{y},\, \dot{z}]^{\,T}$	Longitudinal, lateral and vertical velocity m/s
$\underline{a} = [\ddot{x},\, \ddot{y},\, \ddot{z}]^{\,T}$	Longitudinal, lateral and vertical acceleration (m/s^2)
$\underline{j} = [\dddot{x},\, \dddot{y},\, \dddot{z}]^{\,T}$	Longitudinal, lateral and vertical jerk (m/s^3)
$\underline{\beta} = [\varphi,\, \theta,\, \psi]^{\,T}$	Bryant angles (rad)
$\dot{\underline{\beta}} = \left[\dot{\varphi},\, \dot{\theta},\, \dot{\psi}\right]^{\,T}$	Rates of Bryant angles (rad/s)
$\underline{\omega} = [\omega_x,\, \omega_y,\, \omega_z]^{\,T}$	Angular velocity (rad/s)
$\underline{f} = [f_x,\, f_y,\, f_z]^{\,T}$	Specific forces (m/s^2)
$\mathbf{T}_S$	Transformation matrix from angular velocity to Euler angle rates
$\underline{\varsigma} = \left[\underline{f},\, \underline{\omega}\right]^{\,T}$	Specific forces and angular velocities
$\mathbf{W}(s)$	Transfer function matrix of the open-loop optimal control
$\mathbf{Q}$, $\mathbf{R}$, $\mathbf{R_c}$	Weighting matrices in cost functions

SUBSCRIPTS – (main symbol)$_{SUBSCRIPT}$
In most cases, subscripts indicate to what the main symbol is related

$(\,)_{tar}$	Relates to target signals
$(\,)_{err}$	Relates to the error
$(\,)_{max}$	Relates to the maximal value
$(\,)_{min}$	Relates to the minimal value
$(\,)_{norm}$	Relates to normalized value
$(\,)_{Oto}$	Relates to the otolith system
$(\,)_{Scc}$	Relates to the semicircular system
$(\,)_{Ve}$	Relates to the vestibular system
$(\,)_{ref}$	Relates to reference signals
$(\,)_{sc}$	Relates to scale values of signals
$(\,)_{sh}$	Relates to form-shapes of signals
$(\,)_{f}$	Relates to specific forces
$(\,)_{\omega}$	Relates to angular velocities
$(\,)_{x,y,z}$	x, y, z components
$(\,)_{HP}$	Relates to the high-pass filters
$(\,)_{LP}$	Relates to the low-pass filters
$(\,)_{cam}$	Relates to the tracking camera's reference frame $\mathcal{K}_{cam}$ defined in the KUKA Robocoaster controller
$(\,)_{T}$	Relates to the tilt coordination
$(\,)_{MC}$	Relates to the motion cueing algorithm
$(\,)_{lm}$	Relates to limit values
$(\,)_{Cly}$	Relates to cylindrical coordinates
$(\,)_{c}$	Relates to the state of simulator platform
$(\,)_{d}$	Relates to the desired values in the driving simulator

SUPERSCRIPTS – ^{SUPERSCRIPT}(main symbol) or (main symbol)^{SUPERSCRIPT}
In most cases, superscripts indicate to which frame the main symbol is in

${}^{Ps}( )$	In the pilot head frame $\mathcal{K}_{Ps}$ of the drive simulator
${}^{0}( )$	In the inertial frame $\mathcal{K}_0$
${}^{Pv}( )$	In the pilot head frame $\mathcal{K}_{Pv}$ in the vehicle
${}^{S}( )$	In the simulation frame $\mathcal{K}_S$ of the drive simulator
${}^{V}( )$	In the vehicle frame $\mathcal{K}_V$
${}^{E}( )$	In the moving frame in $\mathcal{K}_E$ at the end effector of the robot
${}^{W}( )$	In the moving, washout frame $\mathcal{K}_W$ created by rotating $\mathcal{K}_0$ an angle α around axis Z_0
$( )^{S}$	Element of a vector decomposed in the frame $\mathcal{K}_S$
$( )^{Ps}$	Element of a vector decomposed in the frame $\mathcal{K}_{Ps}$
$( )^{R}$	Relates to the reference frame $\mathcal{K}_R$ defined in the KUKA Robocoaster controller.

SYMBOLS
(unless otherwise noted in the dissertation)

$\underline{a}$	Acceleration of a point
$\underline{g}$	Acceleration vector due to gravity
$\mathbf{J}$	System cost function
$\underline{e}$	Error
f	Specific force (m/s^2)
φ	Roll angle
θ	Pitch angle
ψ	Yaw angle
$\underline{e}_{MC}$	Average error of simulated acceleration
$\underline{e}_{sc}$	Average scale error of simulated signals
$\underline{e}_{sh}$	Average shape error of simulated signals
λ_1	First performance indicator
λ_2	Second performance indicator
λ_1^*	First good criterion index
λ_2^*	Second good criterion index
$\lambda_1^{\&}$	First normalized good criterion index
$\lambda_2^{\&}$	Second normalized good criterion index
λ_{1f}^*	Average error of the specific force
$\lambda_{1\omega}^*$	Average error of the angular velocity
a_{max}	Maximum acceleration of the drive simulator
ω_{max}	Maximum angular velocity of the drive simulator
$\lambda_{1f,sc}^*$	Average scale error of the specific force
$\lambda_{1f,sh}^*$	Average shape error of the specific force
$\lambda_{1\omega,sc}^*$	Average scale error of the angular velocity
$\lambda_{1\omega,sh}^*$	Average shape error of the angular velocity
λ_{2f}^*	Average error of the first derivative of the specific force
$\lambda_{2\omega}^*$	Average error of the first derivative of angular velocity
$\lambda_{2f,sc}^*$	Average scale error of the first derivative of specific force
$\lambda_{2f,sh}^*$	Average shape error of the first derivative of specific force
$\lambda_{2\omega,sc}^*$	Average scale error of the first derivative of angular velocity
$\lambda_{2\omega,sh}^*$	Average shape error of the first derivative of angular velocity
w_X	Weighting parameters corresponding to the quantity X
$\hat{X}$	Perception (sensation) of a quantity X
$\mathbf{W}(s)$	Matrix of optimal washout filters
δ	Threshold values

$\tau_1, \tau_2, \tau_a, \tau_L$	Time coefficients of the model of the semicircular system
A_0, B_0, B_1	Time coefficients of the model of the otolith system
K_{Scc}	Gain factor of the model of the semicircular organ
K_{Oto}	Gain factor of the model of the semicircular organ
G_S	Gain factor (threshold units) of the model of the semicircular organ
G_O	Gain factor (threshold units) of the model of the otolith organ
φ_S	Tilt angle of the cabin
φ_T	Tilt angle used to simulate the correspondent specific force
x_i	State variables of a dynamic system $i = 1, 2 \ldots$
y_i	Output variables of a dynamic system $i = 1, 2 \ldots$
u_i	Input variables of a dynamic system $i = 1, 2 \ldots$
k_j	Gain values of washout filters $j \in \{LP, HP\}$ in the Y-channel of the CLRN algorithm
ζ_j	Damping factor/ratio of washout filters $j \in \{LP, HP\}$ in the Y-channel of the CLRN algorithm
$\omega_{n,j}$	Natural frequency of washout filters $j \in \{LP, HP\}$ in the Y-channel of the CLRN algorithm
ω_b	Break frequency of high-pass filters in the Y-channel of the CLRN algorithm
k_i	Gain values of high-pass filters in the $i \in \{R\,, \alpha\,, z\}$ channel of the CLG algorithm
ζ_i	Damping factor/ratio of high-pass filters in the $i \in \{R\,, \alpha\,, z\}$ channel of the CLG algorithm
$\omega_{n,i}$	Natural frequency of high-pass filters in the $i \in \{R\,, \alpha\,, z\}$ channel of the CLG algorithm
$\omega_{b,i}$	Break frequency of high-pass filters in the $i \in \{R\,, \alpha\,, z\}$ channel of the CLG algorithm
k_0	Initial gain values of a high-pass filter in the α-channel of the ADSK algorithm
ζ_0	Initial damping ratio of high-pass filters in the α-channel of the ADSK algorithm
ω_{n0}	Initial natural frequency of high-pass filters in the α-channel of the ADSK algorithm
w_a	Weighting parameter of acceleration
w_v	Weighting parameter of velocity
w_s	Weighting parameter of position
w_k	Weighting parameter of gain

w_ζ	Weighting parameter of damping ratio
w_ω	Weighting parameter of natural frequency
$\underline{u}_S$	Input signal for the dynamic system of the drive simulator
$\underline{u}_V$	Input signal for the dynamic system of the simulated vehicle
W_{ij}	Components of washout transfer function $W(s)$ $(i, j = 1, 2)$
r_{Sz}	Distance between rotation center and pilot head
t_0, t_E	Initial time and end time, respectively
γ_i	Break frequency of low-pass filters
c_i	Break frequency of low-pass filters
J_P	Penalty function for liming movement inside workspace
J_{fsh}	Penalty function for minimizing form errors
J_ω	Penalty function for eliminating angular velocity
$J_{\dot{\omega}}$	Penalty function for eliminating angular acceleration
J_{sc}	Penalty function of the ranges of scale factor
J_{tr}	Penalty function for using translational motion
J_{Wo}	Penalty function for washout effect on tilt angle
wJ_k	Weighting value of penalty function J_k with $k \in \{P, fsh, \omega, \dot{\omega}, sc, tr, Wo\}$
k_a	Ratio of translational movement to scaled specific forces
$k_{a,min}$	Minimum ratio of translational movement to scaled specific forces
k_O	Tracking factor in the shape error condition
k_{S1}	Internal scale factor
k_S	Global scale factor
$k_{S,min}$	Minimum global scale factor
$k_{S,max}$	Maximum global scale factor
$e^*_{f,sh}$	Maximum shape error of the simulated specific force
${}^C e^*_{f,sh}$	Maximum shape error of the surge simulated specific force due to the centripetal acceleration
$A_1 \cdots A_6$	Joints of the KUKA Robocoaster
$q_1^R \cdots q_6^R$	Joint angles of the KUKA Robocoaster
a_E, b_E	Distance between K_{Ps} and K_E along X_E- and Z_E-axis, respectively

ABBREVIATION	
AD	Adaptive MCA in general
ADSK	Adaptive MCA introduced by Schweig and Kammers (2011)
ADRN	Coordinated adaptive MCA introduced by Reid and Nahon (1985)
CL	Classical MCA in general
CLG	Classical MCA introduced by Giordano et al. (2010)
CLRN	Classical MCA introduced by Reid and Nahon (1985)
exMPCFK	Explicit model predictive control MCA introduced by Fang and Kemeny (2012)
HPa	High-pass filter transfer function in acceleration channel
HPω	High-pass filter transfer function in angular velocity channel
HQR	Handling Quality Ratings
LP	Low-pass filter transfer function
LQGR	Linear quadratic Gaussian regulator
MC	Motion cues
MCA	Motion cueing algorithm
MPC	Model predictive control MCA introduced by Augusto and Loureiro (2009)
MPC*	Modified Model predictive control MCA for two DoFs
MVMO	Mean variance mapping optimization
MCE	Motion-cue errors
Op	Optimal MCA in general
OpRN	Optimal MCA introduced by Reid and Nahon (1985)
OpS	Optimal MCA introduced by Sivan et al. (1982)
OTO	Otolith system
PRIO	Proprioceptive system
SCC	Semicircular system
SS	Simulator sickness
MS	Motion sickness
FOV	Field of view
OpT	Optimal MCA introduced by Telban et al. (2005)
OpTYM	Optimal MCA with Young Meiry vestibular model introduced by Telban et al. (2005)
TNO	Netherlands Organisation for Applied Scientific Research
WF	Washout filters
ZyRo	Optimal tracking MCA introduced by Zywiol and Romano (2003)
DoF	Degrees of freedom
w.r.t	With regard to

Contents

1 Introduction

1.1 Motivation

Driving and flight simulators have been very useful for different purposes, such as entertainment industry, research on human motion perception, automotive technology, pilot training, and others. The first simulator existed before the Second World War with the aim of training armies. Then, the motion-base flight simulators have been developed for a long period of time, while recently, the driving simulators have also been widely developed due to the decreased cost of the related technology, such as powerful computers, 3D head mount screen. Therefore, nowadays, the driving simulators more and more play a significant role in the training and research of human perception. The goal of the driving simulators is to resemble not only the driving environment, but also the appropriate perception of visual, auditory, proprioceptive and vestibular cues which improve the accuracy of the estimation of motion quantities such as position, velocity and acceleration.

At the beginning, the visual information was used as the primary means of sensing motion and the response of simulated vehicles. Later studies proved the important role of the non-visual sensation for feedback motion perception despite the driver conscious attention. Hence, the motion system of the flight simulator, based on 6 DoFs Stewart Hexapod, is conventional in the 1970s and still the most popularly used in the current time. Afterward, the driving simulators used various kinds of motion systems: 3, 5, 8 DoFs, and the KUKA Robocoaster, as well as the typical mechanical structure named DESDEMONA, etc. Currently, the overall strategy for a driving simulator, as shown in Fig. 1.1, illustrates that visual, auditory and motion system are primary systems equipped for a driving simulator.

The motion platform has been used to provide useful motion cues to the drivers during the simulation session. However, the restricted workspace causes the difficulty of reproducing the exact motion cues required. Thus, a MCA - motion cueing algorithm - block, shown in Fig. 1.1, modifies the target motion from the simulated dynamic vehicle to produce motions that can be implemented in a motion platform and provide drivers with similar motion perceptions as those in the realistic vehicle.

MCAs or named washout filters have been commonly applied to generate useful motions which support the drivers in giving accurate behavior. There are various types of the MCAs introduced in the literature for various driving simulators. The classical washout

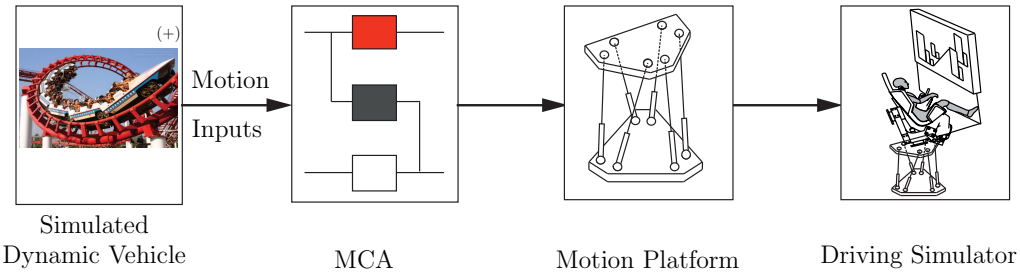


Figure 1.1: The overall strategy for driving simulator (+ Gieszl (2007))

filters are the first ones developed and still used widely. Moreover, several MCAs utilize the optimizing methods for improving the quality of the simulator's motion, most of all with the aim of finding optimized physical quantities such as accelerations and angular velocities by solving the tracking problem with or without constraint conditions.

The tracking goal is derived with the purpose that physical quantities perceived by the human sensory system, such as specific forces and angular velocities, are reproduced as precise as possible resembling those in the realistic vehicle. However, the effects of motion cues on driver behavior are not explicitly described. Furthermore, the drivers could not distinguish between various motions generated by the driving simulators. For example, regarding the simulation task with lateral and yaw rotational motions, the yaw motion was assumed to be simulated by the lateral motion and visual perception (Schroeder (1999)). Moreover, Savona et al. (2017) mentioned that the lateral acceleration was assumed to provide the main sensory information for a driver and that the yaw component seems to be without influence on the motion perception. Besides, most related studies were focused up to now on flight simulations. Therefore, the current requirements for driving simulations are still an open topic research.

The quality of motion cues (motion perception fidelity) depends on many factors, which are impacted by the simulation tasks and the parameters of a MCA. Until now, the explicit criteria for selecting parameters (tuning criteria) for MCAs with a simulation task are still not well-defined. Instead, several particular tuning criteria were proposed for particular simulation tasks solved with a classical algorithm. For example, the Sinacori's criteria (Sinacori (1977)), that have been commonly used in several flight simulation tasks, were derived from the experimental data of running a helicopter flight in the simulator, and have not been validated for other vehicles simulation tasks. Another method was based on the analysis of the simulated motion through a mathematical model of the motion perceptive system (Reid and Nahon (1985)); however, also this method is not reliable enough since the model motion perception lacks reliability. Furthermore, the opaque relationship between the numerical measures derived from simulated quantities and the subjective assessment of the motion perception fidelity is

also a serious problem of determining the tuning criteria. To overcome this problem, subjective tests have been commonly used for assessing motion perception fidelity with both experienced or inexperienced subjects, but the method takes much time to prepare and is hard to be widely applied.

Recently, research on finding the range of several common factors which have important effects on the level of motion perception fidelity, has been pursued. The most important factor in this respect is the scale factor, which scales-down the target signals in order to be reproducible in the simulators with a high level of fidelity (Jamson (2010), Berthoz et al. (2013)). The other factors are angular velocity and acceleration threshold, that limit the change of the quantities to avoid the rotational false cues in the simulation session. Moreover, the amplitude of jerk (Grant and Haycock (2008), Soyka et al. (2009)) and the ratio of translational to rotational motion were taken also into account (Stratulat et al. (2011)). The results of the research are not completely consistent with each other due to different drive tasks and different participants, but they are quite similar and useful for tuning process.

Many studies in the literature mentioned that the tuning parameters for a MCA is necessary to find the appropriate parameters for a specific drive task, which were fulfilled mostly by trial and error. Thus, this approach is very time-consuming and requires that designers have experience in the simulation field (Grant and Reid (1997a)). On the other hand, some auto-tuning methods have been introduced which use fuzzy control theory to tune the algorithms, but these methods focus on only the parameter of the classical algorithm (Song et al. (2003), Hwang et al. (2009), and Chen and Fu (2011)). Besides, in the tuning process, only angular velocity false cues and the position of the simulator are constrained in the defined ranges of these quantities.

The four main objectives of this thesis have been selected according to these findings as: 1) Adapt current MCAs to the Robocoaster robot architecture as available at LMR; 2) Investigate the possibilities of designing automatic tuning procedures that can be applied to any target motion task and robot architecture; 3) Perform a comparison of current MCAs well-tuned to the Robocoaster and a target task to see which one is better for the application; 4) Perform systematic subjective tests and correlate subjective scores with objective performance measurements in order to have a validated objective metric for later studies.

In order to systematically compare the different MCAs and their effects on the Robocoaster, this thesis regards only a simple maneuver with only lateral (left/right) accelerations, such as occur while travelling along a horizontal planar S-curve without

banking with constant velocity, such as they arise along a planar track of a Wilde House roller coaster. This means that both longitudinal (e.g. breaking or accelerating) or vertical (bump) accelerations are neglected. Moreover, it is assumed that the target yaw motions are simulated by the lateral motion and visual environment and thus do not need to be generated physically. Future research can generalize these findings to more general settings in which all acceleration directions can occur.

1.2 Literature survey on motion cueing algorithm

1.2.1 Motion cueing algorithm and types of motion cues

A motion cueing algorithm (MCA) maintains not only a driving simulator within its physical limits, but also generates such movements that the necessary motion cues of drivers on the realistic vehicle are equivalently reproduced. Thanks to the evidence from the research of the human motion perception, the sensed motion cues are represented by physical quantities, e.g. specific forces and angular velocities sensed by the human vestibular system, etc. The conventional MCA termed as "classical washout filters" splits translational acceleration inputs into high-frequency and low-frequency components. While the high-frequency components are reproduced by the translational motion of the platform, the sustained (or low-frequency) translational motions are reproduced by tilting the platform (called tilt coordination), which is the currently most-used "perceptual trick" for MCAs. The trick utilizes the ambiguity of human perception for the gravity and translational acceleration sensed by the otolith organ (a component of the vestibular system) that cannot distinguish the effect due to gravity with those due to translational acceleration when the tilt angle is smaller than 30 °. For example, the sensation of forward (lateral) acceleration can be achieved by tilting upwards (sidewards) the driving simulator with suitable angles reorienting the pilot with respect to gravity such that her/his sensory organs are similarly stimulated as under the translational acceleration. Moreover, the washout effect of the washout filters returns the simulator to its initial position over time, called the "washed-out motion".

Nevertheless, using MCA may cause false cues, weak or strong motion, and simulator sickness, etc. For instance, the tilt rate may be limited, or the driver may perceive false cues of too fast motion instead of the desired translational acceleration. Besides, the tilt acceleration and the location of rotation center may cause undesired translational acceleration cues. In addition, too weak and too strong motion cues reduce also the realism of simulation tasks, and the washout behavior can degrade realism, as stronger returning motions may be detected. Such false cues can lead to motion sickness which

degrades realism even more significantly than that in a fixed simulator. Therefore, for a high realistic simulation task, MCAs have a decisive role in reproducing the necessary motion cues and removing the uncomfortable ones.

Grant and Reid (1997a) mentioned that a necessary motion cue is a signal generated by a motion relative to inertial space which is sensed by the pilot and/or guides the pilot's behavior. Therefore, such motion cues of the real vehicle should be available in the driving simulation generated by motion platform. Earlier, Baarspul (1986) subdivided the environment motion cues into four distinct types, which were further classified by Grant and Reid (1997a) through the underlying frequency: 1) *Initial cues*: There are very high-frequency alerting and onset cues perceived as initial effects of generated motions immediately after a driver starts a maneuver. For example, such a cue appears when a forward or lateral acceleration is quickly changed. 2) *High-frequency transients cues*: There are high-frequency intermediate cues either from external disturbances or driver inputs covering the interval between the initial cues and the low-frequency transient cues. 3) *Low-frequency transient cues*: There are low-frequency intermediate cues either from external disturbances or driver inputs covering the interval between the high-frequency transient and sustained cues. 4) *Sustained cues*: There are perceived as prolonged or quasi-static effects of low-frequency motion generated by disturbances or driver's maneuvers, for example, the lateral acceleration of a car curving at a corner.

The motion cue errors (MCEs) may have severe effects on the participants during driving simulation tasks such as suffering symptoms of motion/simulator sickness. Grant and Reid (1997a) categorized the MCEs into 3 groups: 1) *False cues*, representing a wrong direction of cues or seriously distorted cues; 2) *scale or missing cue errors*, which cover wrong amplitude between the target and the simulated cues or missing cues; and 3) *phase errors*, representing time-delayed stimuli. Of all three MCEs, the false cues are the most destructive for a realistic driving behavior and the main source of simulator sickness (Grant and Reid (1997a)). Baarspul (1986) mentioned that in almost all cases, the fidelity of the simulation with false cues is even worse than with no motion cues at all (e.g. a fixed simulator). On another approach of simulating with large amplitude tasks, Zaichik et al. (2009) categorized the false cues into 4 groups and estimated the effect of each type of the false cues on the realism of the simulation session. Currently, removing all MCEs is in general not possible, so that an explicit correlation between MCEs and the level fidelity of the simulation would be an important research objective. In the meantime, a designer will still meet difficulties to deal with MCEs in a practical application.

Ideally, a precise reproduction of all motion cues in the driving simulator would preserve realism in a simulation session. However, this is in general impossible due to the restricted workspace of the motion platform. Fortunately, due to the nonlinear characteristic of the human motion receptors, the target motion cues can be suitably scaled down for a specific driving simulator. Hence, if the reproduced motion cues perfectly track the scaled target ones, they still provide the realistic motion perception.

1.2.2 State of the art of MCAs

Motion cueing algorithms are aimed at mapping vehicle motions into the limited workspace of driving simulators while preserving the perceptual realism of the simulation. In the literature, there is a variety of MCAs developed for corresponding driving simulators, such as 6 DoF Stewart platforms (Stewart (1965)); the 3 DoF VTI simulator (Nordmark et al. (2004)); the 8 DoF driving simulator of the University of Leeds (Jamson (2007)); the 6 DoF Desdemona simulator in TNO (Feenstra et al. (2007)); the 6 DoF KUKA Robocoaster (Giordano et al. (2010)), etc. The survey of driving simulators reported by Slob (2008) describes more a hundred different driving systems in the world. Most of these simulators use a 6 DoF hexapod as the motion platform, others have modifications in the mechanical structure to improve the workspace.

At the origin of MCAs, one can find the classical washout filters proposed by Conrad and Schmidt (1970). Due to the simplicity of the algorithm, it is the most widely used approach in commercial simulators and has also been further developed to match general architectures of drive simulators. For example, to exploit the workspace of KUKA Robocoaster, Giordano et al. (2010) developed a new motion cueing algorithm named cylindrical classical algorithm in which the washout filter is implemented in cylindrical instead of Cartesian coordinates as usually done in hexapod platforms. Similarly, a MCA with spherical coordinates was introduced by Wentink et al. (2005) for the Desdemona driving simulator in TNO. Furthermore, the lane MCA or the MCA for 3 DoF and 8 DoF simulators introduced in the report of Fischer et al. (2010) were derived from the original classical algorithm. Also, an alternative version of the classical algorithm called fast tilt-coordination proposed by Fischer (2009), as well as a compensated washout filters introduced by Sammet (2007) and re-described by Fischer (2009) are also based on the classical algorithm. However, the MCA exploits ineffectively the available workspace of driving simulators and potentially produces false cues due to its linear characteristics as mentioned by (Shao et al. (2009)). Therefore, Wang and Fu (2004), as well as Chen and Fu (2007) propose the integration of a workspace management in the classical algorithm to improve the exploitation of the restricted area and to reduce MCEs.

Adaptive washout filters were developed in various versions, starting with Parrish et al. (1975) and followed by Reid and Nahon (1985), Nehaoua et al. (2008), and Schweig and Kammers (2011). In general, these MCAs are also washout filters whose adaptive parameters are systematically modified in real time to minimize a cost function using steepest descent techniques. The cost function is designed to make the simulated quantities track the corresponding target ones. As mentioned in Nahon and Reid (1990), although the adaptive MCAs can reduce the false cues, their stability is strongly depending on the adaptation parameter. Altogether, however, the adaptive MCA are more flexible than those of the classical MCAs.

The optimal washout filters were firstly proposed by Sivan et al. (1982) and later continuously implemented and developed by Reid and Nahon (1985), Telban et al. (2005), Cho et al. (2007), Shao et al. (2009), and others. Compared to the classical algorithms, the optimal MCAs use higher-order washout filters that are determined by optimal control theory with or without constraints. In the control models of the optimal MCAs, the human vestibular system is integrated to minimize the motion sensation error of a driver between the aircraft and on the simulator. The cost function of the optimal control is defined to minimize the sensation errors and to have an influence on the simulator's physical quantities. The differences between the perception models make the variety of the optimal washout filters. At the beginning, Sivan et al. (1982) used the vestibular model with a lower order, then Reid and Nahon (1985) used more involved vestibular models of higher order. Later, Telban et al. (2005) introduced a new model of the vestibular system with a finer set of parameters. In the same report, the authors especially introduce a visual-vestibular model that is then integrated into a nonlinear optimal MCA. In the algorithm, neuron networks are used to solve the Ricatti equation in real time, leading to real-time optimal washout filters. Recently, Shao et al. (2009) developed a new MCA that integrates the model of the human pressure receptor together with the vestibular system model. Furthermore, in another approach, Cho et al. (2007), instead of using Linear quadratic regulator (LQR) algorithm, used quadratic programming (QP) algorithm to develop the QP-based fast algorithm to solve the optimal problem with constraint conditions for position, acceleration and angular velocity of the motion simulator Eclipse-II. As mentioned in the report of Telban et al. (2005), the optimal washout filters can clearly improve motion cues. However, the tuning process for optimal washout filters is more involved as the weighting parameters of the cost function have opaque relation with the simulated quantities. Besides, the assumption that the minimizing quadratic errors of the perceived motion cues improve the motion perception fidelity has not yet been firmly proved. Additionally, the lack of

the full understanding and the inaccuracy of the modeling of human motion perception degrade the desired effect of the optimal MCAs.

Zywiol and Romano (2003) proposed an upgraded version of an offline optimal tracking algorithm, which was developed from the optimal MCA using a nonlinear optimal tilt coordination introduced by Romano (2003). The algorithm is different from that developed by Sivan et al. (1982), because it focuses on solving the offline problem of global tracking the desired signals. Here, the models of the human motion receptor, such as the vestibular system, are not considered. Thus, the algorithm is able to find better combinations of linear acceleration and tilt angle to replicate the target specific force as the online version. However, an advantage of this approach is that the tuning process can be done with fewer parameters. This was realized in the altered parameter set of the report of Kecskeméthy and Tändl (2009).

A novel approach to MCAs, which has been developed recently, is based on a model predictive control (MPC). Dagdelen et al. (2004) first introduced this algorithm and applied it to the ULTIMATE simulator at Renault, Technical Center for simulation. The MPC integrates both the motion system dynamics and the vestibular model, hereby controlling better the trajectory of the simulator inside the workspace and reducing the sensory conflict. Later, Augusto and Loureiro (2009) described in detail another version of the MPC-MCA for the Charmers driving simulator that considers the vestibular model and very similar MPC-MCA was also described by Baseggio et al. (2011). By the MPC strategy, MCAs can handle not only limits of the workspace, but also the thresholds of motion perceptions. Thus, potential false cues can be reduced, and the exploitation of the workspace can be improved. However, these MCAs, by using implicit MPC strategies, are computationally very time-consuming, especially with a large horizon of prediction and control, which restrict their online suitability. To overcome this problem, the explicit MPC strategy was introduced by Fang and Kemeny (2012), with which the controller is first computed offline and then implemented online.

In conclusion, several MCAs have been developed in the past to improve the realism of driving simulator, which all have some strengths and weaknesses. Mainly, the tuning of the parameters is an open issue in particular for washout filters where the parameters are not intuitive for inexperienced simulator users. This is somehow better for MCAs using optimal tracking or the MPC strategy were the parameters correspond explicitly to the motion system physical limits and motion detection thresholds. However, in all cases an automated tuning would be desirable in order to alleviate the user from this cumbersome task, as explained in the next section.

1.2.3 Requirements of the tuning parameters for MCAs

To achieve realistic simulations, the MCAs or washout filters need typically extensive tuning with objective and subjective assessments. The tuning process is mostly done by experts with trial and errors, and the optimal parameters obtained depend on many different aspects such as the simulator specifications and the maneuvering commands. For example, if objective assessments are used, a MCA is tuned to both generate necessary motion cues and eliminate the false cues (see section 1.2.1). In another approach, if subjective assessments are used, the participant evaluates the level of realism based on given scores or statements, and the parameters will be modified to remove the disturbing subjective notions. In the investigation of the MCAs for the flight simulator, Reid and Nahon (1985) as well as Telban et al. (2005) firstly used the objective assessment for roughly tuning, then implemented the subjective assessment for fine tuning. This two-stage produce is necessary as due to the lack of information of the human receptors and the exact effect of the physical quantities on the level of realism in a specific simulation session, a subjective tuning process from scratch could require too many trials. Thus by the objective pre-assessment of the simulated quantities, substantial time can be saved in the trial and error, subjective scoring, as has been investigated by Hess and Marchesi (2009), White and Rodchenko (1999), and Zaichik et al. (2010).

1.3 Structure of the thesis

After *Chapter 1*, which introduces in a brief review to the development of MCAs, *Chapter 2* presents the background of human motion perception and psychophysics problems of motion sickness. Then, *Chapter 3* given a short overview and the control procedures of the KUKA Robocoaster driving simulator built up at LMR. *Chapter 4* reviews and describes the principles of the MCAs expressed by their mathematical formulas and implementing diagrams, while, *Chapter 5* reviews the methods for assessing motion perception fidelity in the literature and the tuning methods. Besides, common criteria for tuning all MCAs which is not mentioned in the literature is proposed. The chapter closes by computing the parameters of the investigated MCAs by trial and error, and the automatic tuning process. *Chapter 6* finally compares the response of the different MCAs with given parameters and correlates these with subjective assessments, such as to obtain a relationship between the subjective index and the numerical index. Finally, *Chapter 7* addresses the conclusions of this dissertation and recommendations for further research. The appendix provides extra information such as the mathematical models of motion sensory systems and responses of MCAs with target signals scaled down by a different factor, etc.

2 Fundamentals of Human Motion Perception

Motion sensory systems transduce information about our movement as patterns of activity across populations of neurons. Hereby, the self-motion sensory system including the visual, auditory, vestibular, kinesthetic, and somatosensory (proprioception) systems contribute to the perception of the stimuli in different reference frames. Therefore, the precise motion perception comprises a complex interaction several components of a multi-sensory system. This section provides a brief overview of the basic physiological principles involved in human self-motion perception and the corresponding perceived physical quantities together with the corresponding mathematical models.

2.1 Historic overview of perception modeling and evaluation

2.1.1 The auditory system

The auditory system transduces sound energy into neural signals. The sensory information (hair cells) in the cochlea in the inner ears transform mechanical forces into electrical impulses. From the cochlea, the acoustic nerve transmits the neural signals to the primary auditory cortical area. The auditory system can detect changes in overall sound intensity, changes in inter-aural time differences, inter-aural intensity differences, and changes in the frequency spectrum of the auditory scene (Blauert (1997)). In the literature of vection, which is an illusion of self-motion in stationary observers, there are only very few reports on auditory induced vection. Schulte-Pelkum (2007) restated from the works of Dodge (1923) that vection was successfully induced by a moving sound field in some attempts, even though clear data were not obtained. Then, Lackner (1977) also succeeded in inducing a circular vection in blindfolded participants. Lackner reported that depending on stimulus (different combinations of six hidden loudspeakers around seated participants were used), about $25\% - 75\%$ of the participants had the sensation of self-rotation. However, nobody reported vection when visual information about the stationary environment was accessible. Therefore, it seems that auditory information is weighted less than visual information during self-motion perception. Similar results were obtained recently in a study by Larsson et al. (2004), who used the headphone-based auralization using generic head-related transfer functions, named Head-Related Transfer Functions (HRTF).

2.1.2 The visual system

The visual system detects and interprets information from visible light falling onto the retina at the back of each eye. Two subsystems, the scotopic (dark-adapted system) and photopic (light-adapted system), operate at low and high levels of illumination, respectively. On the retina, there are two types of photoreceptor cells called rods opsins and cones opsins that absorb photons (a particle of light) and convert light into neural signals that are sent to the brain. To inform a relative motion between the eyeball and the visual scene, the change of structured light of the image on the retina, is transformed systematically. Such transformations are called as "optic flow" and are continually processed in the visual system Schulte-Pelkum (2007).

Several studies have shown that that optic flow is involved in estimating the direction and speed of self-motion (Gibson (1979), Bremmer and Lappe (1999), Schulte-Pelkum (2007)). Furthermore, Telban et al. (2005) referred to the report of Young (1978) to mention that visually induced self-motion has been used to mimic for rotations about both the earth-horizontal and earth-vertical axes, and along all three linear axes. For example, circularvection refers to visually induced rotational motion, in particular yaw, but also visually induced pitch and roll. Moreover, linearvection refers to visually induced translational motion. One common experience of linearvection is the illusion of moving backwards when seated in a stationary train as the adjacent train in the station begins to slowly move forward.

Additionally, the relative distance to the viewer can be provided by the visual cues. When judging self-motion, particular features like landmarks and can be used to estimate motion in response to "sightings" of these landmarks. However, the determination of such features is not precise as it requires regular checks and feedback (Harris et al. (2002)).

Cardullo et al. (2011) noted that the self-motion perception due to the visual system is a slow process compared to the vestibular response to head movements. Therefore, when a human without a functioning vestibular system bases on visual perception of attitude and motion alone, her/his eye stabilization and posture control have a limited range compared to healthy humans.

Young (1978) mentioned that the onset delay (or latency) of visually induced motion is highly variable among individuals, and repeated exposures will reduce this latency. The latency of onset to either circularvection or linearvection has an impact on the perception of motion in flight simulation.

Hess (2008) proposed the visual perception model that provides a simple model for visual observation (see Fig. D.1a on page 216 in appendix D). Lone and Cooke (2010) noted that the saturation limits are set to twice the variance of a zero-mean Gaussian random number generator, where the variance of this random number generator implicitly determines the visual cue quality. The values of the parameter in the model were determined by the experiment implemented e.g. by Hess (2008)

2.1.3 The vestibular system

The vestibular organs are located close to the cochlea in each ear, and in a mirror-symmetric fashion to each other in the horizontal plane and consist of the semicircular canals and the otolith organs for sensing angular and linear motion, respectively (Fig. 2.1). Each of the two organs contains five vestibular sub-organs: three semicircular canals (sensing rotational movements) and two otolithic organs – named saccule and utricle – (sensing linear acceleration originated from inertial or gravitational acceleration). The three semicircular canals are approximately orthogonal to each other and referred to as the anterior, posterior, and horizontal canals. Vestibular organs are filled with a fluid (called endolymph) that is inside the canals and can move relative to each of the surrounding tubes (Tortora and Derrickson (2008)).

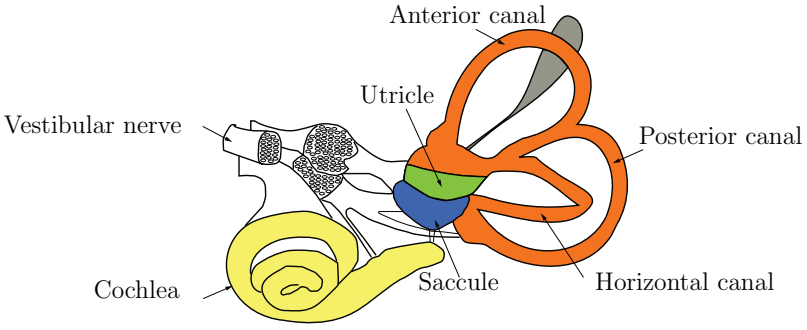


Figure 2.1: Vestibular system (adapted from Kandel et al. (2000))

Semicircular organs

The semicircular organ has a canal containing an ampulla. A gelatinous structure called cupula is situated inside the ampulla. The structure contains the sensory hair cells and bridges the width of the ampulla cavity, forming a seal through which endolymph cannot flow. Each hair cell is composed of about 70 stereocilia and one kinocilium (Boff et al. (1986)), with the stereocilia graded in length towards the kinocilium (Telban et al. (2005)). The angular acceleration along an axis of canal orientation deflects the cupula in the opposite direction and causes a displacement of the hair bundled in each hair

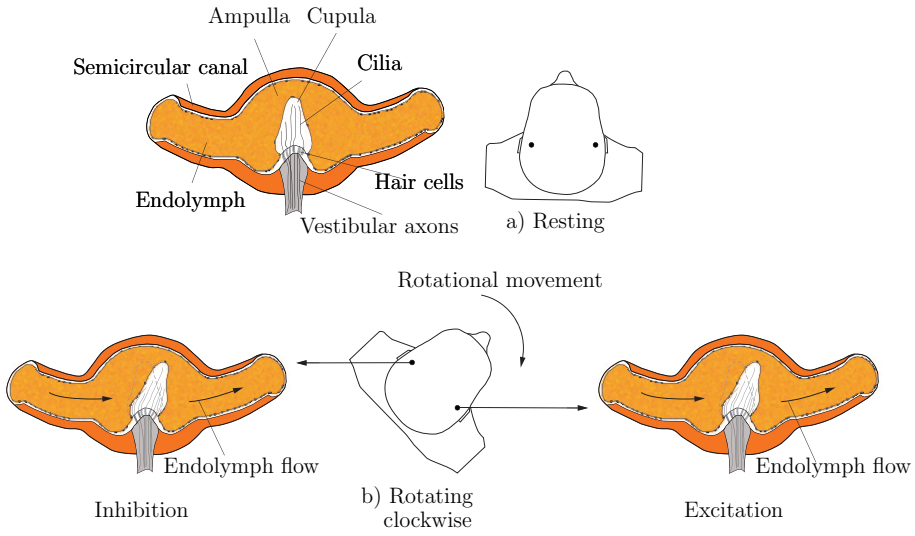


Figure 2.2: Semicircular organ in a) rest b) clockwise rotational movement (adapted from Tortora and Derrickson (2008))

cell due to the inertia of endolymph. This mechanical force is transduced into neural signals and transmitted to the brain.

Fig. 2.2a shows the semicircular organ in resting state and Fig. 2.2b shows the stimulated state when the human is under a clockwise rotational movement. In the stimulated state, the endolymph flow causes a deflection of the cupula generating rotating motion cues. For two semicircular organs, the hair cells are excited in one direction and inhibited in the opposite direction.

Fernandez and Goldberg (1971) noted that the semicircular canals, in the frequency range of normal human (and animal) head and body movements, has the neural response dynamics that are actually proportional to the angular velocity, rather than angular acceleration. The detection of the cupula and hence the firing rate of the sensory cells is proportional to the angular velocity of the head and not to the angular acceleration. Note that even though the angular acceleration (related to an inertial force) is the physical stimulus acting on the sensory organ, an angular velocity is what is sensed during head movements due to the high-pass characteristics of the semicircular organ. Therefore, the angular velocity is the filtered result, and it would be incorrect to state that the vestibular canals register angular velocity directly. Schulte-Pelkum (2007) gave an example to illustrate the statement: “After about five seconds of constant velocity rotation (e.g., on a rotating chair), the endolymph flows back to the resting position, and the sensory cells decrease the firing rate back to resting level, even though the observer is still rotating. The vestibular nuclei circuitry in the brainstem extends this time constant to about 15 seconds. If other sensory information, such

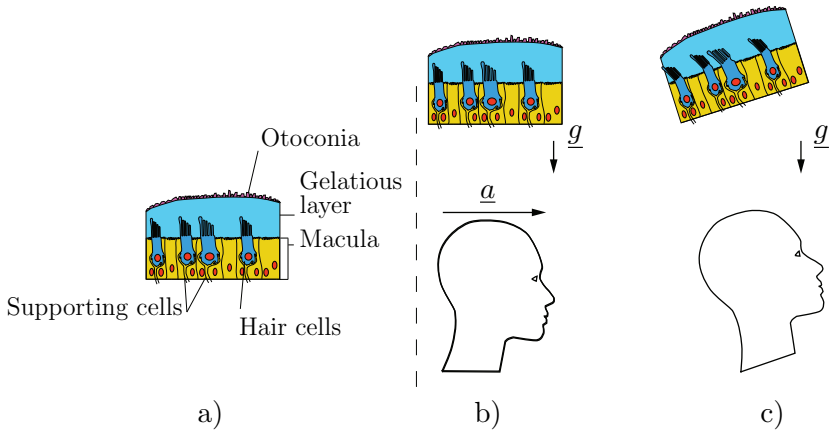


Figure 2.3: Otolith organ a) In resting state; b) Under forward acceleration; c) Under backward head tilt (adapted Tortora and Derrickson (2008))

as visual information, is missing in this case, the sensation of motion will gradually diminish."

Therefore, these high-pass characteristics of the semicircular canals are crucial, since they can cause aftereffects due to self-motion, such as the rotation aftereffect. Various dynamic models of the semicircular are introduced in the literature (see Tab. D.2 on page 217). In many existing models, the cupula deflection is modeled as a torsion pendulum. Telban et al. (2005) introduced the representation of the semicircular channels as a dynamical model. Reid and Nahon (1985) introduced the dynamic model in which ω and $\hat{\omega}$ are the angular velocity and perceived one, respectively (see Fig. D.1c on page 216). The model includes there elements in which the first block models the cupula displacement, i.e., the over damped torsional pendulum; the second block represents the threshold of angular velocity, and the last block represents the washout of human response to steady-state rotational acceleration inputs (Reid and Nahon (1985)). Later, Fischer (2009) summarized the transfer function, $H_{Scc,i}\,, i \in \{1 \cdots 4\}$, of the existing models shown in Tab. D.2.

Otolithic organs

The otolithic organs, shown in Fig. 2.3a, contain an otolithic membrane composed of the otoconial layer and the gelatinous layer. This gelatinous layer is located under the otoconical layer and is attached to the sensory cell base, known as the macula, that is incorporated into the membranous tissue walls of the inner ear. The macula can move with the head because it is rigidly attached to the skull (Telban et al. (2005)). An otolithic organ, the utricle, is oriented in a horizontal plane in the normal upright head position, while another organ, the saccule, is oriented in a vertical plane. The macula

is sensitive to linear accelerations since the inertia of the otoconical layer shifts the gelatinous layer during changes in linear velocity or head movement. For example, the resulting displacements of the otolithic membrane due to forward linear acceleration and the tilt motion are illustrated in Fig 2.3b,c, respectively. The reason of the deflection is that with a forward acceleration or backward tilting of the head the relative motion results in a deformation of the gelatinous and otoconial layers in shear. The deformation is caused by either inertial force (accompanying with a forward acceleration $\underline{a}$) pulling the gelatinous layer backward (in Fig. 2.3b) or gravity force (accompanying with gravity acceleration $\underline{g}$ pulling the layer downward (Fig. 2.3c). Depending on the direction of the deflection, the hair cells are either excited or inhibited.

Similar to the vestibular organ, the dynamic models of the otolithic organ has been for long an important topic of research in the literature (Zacharias (1978), Young and Meiry (1968) , Fernandez and Goldberg (1971), Hosman and Stassen (1999), Reid and Nahon (1985), etc). Fig. D.1b on page 216 shows the dynamic model introduced by Reid and Nahon (1985). The first block of the model stands for the otolith actual mechanical behavior, the second is the mechanical threshold, while the last one represents a neural processing activity. The input of the model is defined as specific force $\underline{f} = \underline{a} - \underline{g}$, and the output is named as perceived (or sensed) specific force $\underline{\hat{f}}$. Later, Fischer (2009) summarized various transfer functions of the otolith system ($H_{Oto,i}\,,\ i \in \{1\,,2\,,3\}$), listed in the Tab. D.1 on page 217.

2.1.4 The somatosensory (or proprioception) systems

The somatosensory system consists of various sensory receptors, namely the sensations of touch or pressure, vibrations, temperature, pain, the sensations of muscle movement and joint position. Therefore, the somatosensory systems provide information of our movement through touch (i.e., the pressure sensor with skin) and the position and movement of our body parts (proprioception) through the stimulation of muscle and joints. Moreover, the somatosensory systems also monitor the temperature of the body and collect information about pain.

The mechanical sensors of the cutaneous system perceive temperature and pressure changes on the skin, whereas spindle receptors signal changes of tension and length of the muscles and tendons as well as joint position (Schulte-Pelkum (2007)). For example, when we are accelerated forward in a vehicle, the somatosensory system registers change of pressure in the back as we are pressed into the seat due to inertia, and vibrations from the vehicle. Oftentimes, it is difficult or even impossible to disambiguate vestibular and somatosensory contributions of self-motion (Guedry (1974)).

The relative importance of the somatosensory information compared to vestibular information has been highlighted by Bringoux et al. (2003), and Schulte-Pelkum (2007) mentioned that thresholds for the detection of slow orientation changes w.r.t to gravity are significantly increased if all body pressure cues are eliminated by a pressure suit.

The modeling of the somatosensory is often ignored in the literature. Gum (1973) summarized two mathematical models for body-pressure and controlling lateral head motion. Fig. D.1d on page 216 shows the body-pressure models with the natural frequency of $34\,(rad/s)$. This bandwidth would make the body's pressure response dynamics the highest of all of human motion sensing capabilities. In the model, the high-pass filter has $1\,(s)$ time-constant due to the adaptation effect with that the receptors in the skin lose their sensitivity to sustained acceleration. Fig. D.1e shows the models of the control of lateral head motion. Here, the head is essentially an inverted pendulum w.r.t the driver's body that is trapped into a moving cockpit. The stimulated aspect to the head model includes the gravity forces, the muscle forces, and disturbance factors, such as inertial force. The muscle model is used to generate the muscle force, while the muscle spindle feedback is used to get the feedback of the tilt angle of the head.

To summarize, substantial work in human sensory modeling has provided useful, but incomplete information for predicting motion platform effects on driver-vehicle performance and behavior. Hence, the more empirical data is needed for building better models.

2.1.5 Vestibular thresholds

The threshold of the whole-body self-motion has been investigated in early studies of Mach (1875), Hosman and Van der Vaart (1978), Young et al. (2001), Soyka et al. (2011). Together with the development of mathematical models of the vestibular system, thresholds of angular velocities and linear acceleration are estimated for the semicircular and otolith organs, respectively. In the literature, two types of experimental studies estimating the thresholds have been presented: the estimation of absolute and differential thresholds. Most of the related works in the literature are on the absolute thresholds studies. There are few studies on linear motion perception at suprathreshold levels (Zaichik et al. (1999), Naseri and Grant (2011)).

From the psychophysical point of view, the absolute threshold is defined as the smallest detectable level of a stimulus. Thus the aim of such studies is to find the minimum

amplitudes of motion that subjects are capable to detect. Numerous absolute threshold values for the three translational motions and rotational motion were presented by Meiry (1965), Zacharias (1978), Hosman and Van der Vaart (1978), Reid and Nahon (1985), Benson et al. (1986), etc. Later, Fischer (2009) reviewed the threshold values with and without the visual stimulus, as shown in Tab. D.3 on page 217. Several previous studies mentioned that the absolute threshold studies have shown a decrease of threshold estimates with increasing frequency which indicates sensitivity to both acceleration and jerk (Benson et al. (1986), Gundry (1978)).

A different threshold study investigates human ability to discriminate between two different motions that are both easily detectable. In psychophysics, the smallest detectable difference between a starting and a secondary level of a particular sensory stimulus is called the difference threshold or just-noticeable-difference (JND). The Weber's law (Teghtsoonian (1971)) of psychophysics used in the studies states that the change in a stimulus that will be just noticeable is a constant ratio of the original stimulus. That means the ratio between the difference threshold magnitude ΔI and the stimulus intensity I for any given stimulus is a constant $K_{Web} = \frac{\Delta I}{I}$. Zaichik et al. (1999) prove the applicability of the Weber's law to the perception of the acceleration supra-threshold values. Recently, Naseri and Grant (2011) showed that the Weber's factors for the linear motion at supra-threshold change with respect to the frequency of the sinusoidal stimulus, such as 0.06 for a frequency of 0.4 Hz and 0.02 for a frequency of 0.6 Hz.

2.2 The nature of simulator sickness

As mentioned before, driving simulators with interactive 3D simulation environments have been effectively used for drivers training and human behavior research. However, a considerable number of simulator operators suffer from symptoms being like motion sickness (MS), or more specifically simulator sickness (SS). For example, Crowley (1987) indicated that 40% of the participants in his study reported at least one symptom of SS, while McGuiness et al. (1981) mentioned the number of 27%. The majority of the participants state that these symptoms persisted between 15 minutes and 6 hours after experiments (McGuiness et al. (1981)), and the published guidelines for pilots exposed to a simulator should avoid high-risk activities for at least 12 hours after the simulator training as recommended by Johnson (2005). Thus, SS is a serious problem and can reduce the fidelity of the simulation session. In this section, an overview of the SS as well as its causes and effects are introduced.

2.2.1 Simulator sickness

Before the appearance of driving simulators, the manifestations and symptoms of the MS were a noticeable problem for passengers traveling by train or other vehicles manifesting signs of pallor, sweating, and vomiting. In that time, medicamentations were used to prevent the participants from effects of the sickness (Reason and Brand (1975)). SS is a particular form of MS that has become problematic with the increase of simulation for aviator training in the 1980s and the current entertainment industry. For example, flight simulators, the IMAX theater, or virtual reality environments can generate SS. The discomfort experience during such simulation sessions is now universally referred to as SS. Kennedy et al. (1988) stated that while MS is caused by motion, SS is caused by the inability to simulate the motion environment accurately enough (Johnson (2005)). In fixed simulators, the observers commented that their discomfort stemmed from the lack of vestibular cues coming with simulated motion. (Miller and Goodson (1958)).

SS is polysymptomatic (Kennedy and Fowlkes (1992)) and includes symptoms such as nausea, dizziness, spinning sensations, visual flashbacks, motor dyskinesia, confusion, and drowsiness. Observable signs of SS consist of pallor, cold sweating, and emesis (vomiting) (McCauley (1984)). For example, Gower Jr and Fowlkes (1989) reported that the SS symptoms produced by the 6DoF motion simulator 2B31 were fatigue (34 % of participants), eyestrain (29%), headache (17%), difficulty focusing (13 %), sweating (11 %), nausea (9 %), and stomach awareness (9 %) (Johnson (2005)).

2.2.2 Measurement of simulator sickness

Measuring SS is a difficult task as mentioned in an extensive number of publications (Hettinger et al. (1987), Casali and Frank (1988), Kennedy and Fowlkes (1992), Kolasinski (1995)). Because SS is polysymptomatic one cannot measure a single dependent variable (Kennedy and Fowlkes (1992)). Furthermore, the individual susceptibility of the participant influences the results of the measurement of SS. Four possible ways of measuring SS are (Hettinger et al. (1987), Casali and Frank (1988)):

1. Direct observation of participants during a simulator session and identification of signs of facial pallor and sweating.

2. Self-report measures, such as the Simulator Sickness Questionnaire (SSQ), that asks the participant to note after the simulation the type and severity of symptoms cur-

rently being experienced. For example the standard measurement instrument for SS, the Simulator Sickness Questionnaire, lists 16 symptoms: general discomfort, fatigue, headache, eyestrain, difficulty focusing, increased salivation, sweating, nausea, difficulty concentrating, fullness of head, blurred vision, dizzy (eyes open), dizzy (eyes closed), vertigo, stomach awareness, and burping (Kennedy et al. (1988)).

3. Furnishing participants with wear equipment for measuring physiological conditions such as respiration rate and stomach activity. For example, heart rate, or pulse rate, have been reported to change from baseline levels as a function of simulation exposure; Sweating, a common symptom of SS, can be measured as an increase in skin conductance or a decrease in skin resistance (Casali and Frank (1988))

4. Employing tests of postural equilibrium to measure simulator-induced disorientation or ataxia after the trial. These tests have been widely used, but with mixed results. Such tests are, for example, Sharpened Romberg (SR), Stand on One Leg Eyes Closed (SOLEC), Stand On Preferred Leg Eyes Closed (SOPLEC, SOPL), Stand On Non-preferred Leg Eyes Closed (SONLEC, SONL), walk toe to heel, Walk On Floor Eyes Closed (WOFEC), Walk On Line Eyes Closed (WOLEC), and Walk On Rail Eyes Open (WOREO).

2.2.3 Potential variables associated with simulator sickness in virtual environments

Many variables related to simulator sickness can be found in the literature. Johnson (2005) reviewed and categorized those variables as individual related variables, simulator related variables, and task related variables. The information below summarizes the details not only from the report of Johnson (2005), but also from Kolasinski (1995), Johnson (2005), Kuipers (2014).

First, the **individual related variables** include: 1) *Gender*: Females are more susceptible than males (Kolasinski (1995), Pausch et al. (1992)); 2) *Age*: Several reviewers have reported that susceptibility to SS varies with ages in the same way that MS varies with age. Reason and Brand (1975) claim that below age 2 infants are generally immune. Susceptibility is at its highest level between ages 2 and about 12. There is a pronounced decline between ages 12 and 21. This decline continues, though slowlier, through adulthood until about age 50, after which SS is very rare (Johnson (2005)). 3) *Experience*: Symptoms of simulator sickness can increase with experience. That is, pilots with a greater number of flight hours reported a greater susceptibility to SS

(Braithwaite and Braithwaite (1990), Gower Jr and Fowlkes (1989), and Silverman and Slaughter (1995)); 4)*Prior history of motion sickness*: People who have a history of prior episodes of SS will be more likely to experience SS in future simulator-based training (Kennedy et al. (1987), Wright (1995), Gower Jr and Fowlkes (1989), Lampton et al. (1995)); 5) *Miscellaneous*: illness, drugs, sleep and fatigue have also effects on SS (e.g individuals, who are in ill health, are more susceptible to SS) (Kennedy et al. (1987), Kolasinski (1995)). Similarly, medication, drugs, and alcohol can increase a pilot's susceptibility to SS (Young (2003), Kennedy et al. (1987)).

Secondly, the **simulator related variables** are: 1) *Wide field of view* (FOV) displays have been associated with an increased susceptibility to SS (Kennedy and Fowlkes (1992), Kennedy and Fowlkes (1992), Kolasinski (1995), McCauley (1984), Pausch et al. (1992)); 2) *Visual factors* such as off-axis viewing, design eye point, or viewing region also influence SS; 3) *The general issue* of cue asynchrony (i.e visual delay, transport delay, and/or asynchronous visual and motion systems) has been investigated as a source of SS, but with equivocal results (Kolasinski (1995), McCauley (1984), Pausch et al. (1992)); 4) *Motion systems* with strong motion cues can increase the symptoms of SS (Kuipers (2014), Kennedy and Fowlkes (1992), Kolasinski (1995), McCauley (1984)).

Thirdly, the **task related variables** include: 1) *Session duration*: The longer the period of time spent operating the simulator, the greater the likelihood of significant discomfort (Gower Jr and Fowlkes (1989), Kennedy and Fowlkes (1992), Kolasinski (1995), McCauley (1984), Wright (1995)); 2) *Velocity and acceleration*: navigation with high speed stimulates the symptoms of simulator sickness (Reason and Brand (1975), Kuipers (2014)); 3) *Unusual or unnatural maneuvers*, such as moving forward/backwards in time or flying backwards, are also associated with an increased risk of discomfort (Kolasinski (1995)); 4) *Scene complexity*: details needed for a realistic reproduction of a 3D environment stimulate the symptoms of SS (i.e the more detailed the scenery, the higher the chances of experiencing SS)(Kuipers (2014))

2.2.4 Manifestation and theory of simulator sickness

In the literature, the investigation of SS involves the study of incidence, residual aftereffect, adaptation, and susceptibility. The general conclusions of the aspects based on the reviews of Johnson (2005) are:

- The incidence of SS varies within a large range. Depending upon the simulator, the

conditions of operation, and the definition of criteria applied, the rate of SS can vary from low to extremely high.

- The most commonly reported aftereffects are fatigue, stomach awareness, and vertigo, in that order. Most of these aftereffects appear and disappear within 2 hours after exiting the simulator, although some participants reported symptoms lasting up to "several hours after the simulator training session" (Silverman and Slaughter (1995)).

- The more prior exposure to the simulator, the less SS is experienced. For example, Crowley (1987) stated that there was a statistically significant inverse relationship between the prior number of hours spent training in the Cobra simulator and the amount of SS reported. This statement was also given by Gower Jr and Fowlkes (1989)

- SS susceptibility depends on many variables as listed in section 2.2.3.

In the literature, there are two explanations for the SS, the **sensory conflict (SC)** and **postural instability (PI)** theory. Each theory assigns a different cause to explain the aspects that influence SS.

First, the **sensory conflict** theory, that is commonly accepted to explain SS, states that sensory inputs from the eyes, semicircular canals, otoliths, proprioceptors, and somatosensors are provided in parallel both to a neural store of past sensory patterns of spatial movement and to a comparator unit (Reason (1970), Reason and Brand (1975)). This comparator unit compares the present pattern of motion information with an expected pattern based on prior motion history stored in the neural store. A mismatch between the current pattern and the stored pattern generates a mismatch signal. This mismatch signal is a conflict between the current novel motion environment and past experience and causes both SS and the process of adaptation. However, the mismatch signals can be decreased with continued simulation sessions. Therefore, the drivers gradually adapt to the simulation system. Moreover, the SC theory assumes that the conflict between motion information provided by various kinds of sensory system is an important source of SS. For example, the SC theory can explain that a driving simulator providing only visual perception is even better than one providing unrelated visual and motion information.

Secondly, the **postural instability** theory notes that the situations producing sickness are caused by their unfamiliarity to the participants. Enduring this unfamiliarity, a participant may lose postural control. This postural instability causes discomfort

until the participant adapt. in other words, a prolonged exposure to a novel motion environment causes postural instability that precedes and causes the sickness. This theory is supported by the experimental data in the reports of Stoffregen and Smart (1998) and Stoffregen et al. (2000).

The SC and PI theory make different predictions in certain instances. The SC theory states that SS must decline after the age of 12, as the participants' life experiences have an increasing number of prior sensorimotor patterns of motion memories gathered in their neural store. Another reason is that with age, the participants' receptivity or the strength of the mismatches also declines. In contrast, the PI theory explains that postural stability among adults is known to decline with increasing age (Lane et al. (1989)), and suggests that SS increases with age.

2.2.5 Preventing and treating simulator sickness

Several authors have published guidelines in an effort to reduce the rate of SS (Kennedy et al. (1987), Kolasinski (1995), McCauley (1984), Wright (1995), and Johnson (2005)). Some suggestions based on the work of Johnson (2005) to prevent potential discomfort in simulation sessions are: 1) Simulator flights should not be scheduled on the same day as aircraft flights; 2) Arrive for simulator training in a usual state of health and fitness; 3) Persons who are new to the simulator, or who have not operated it for months, are at risk; 4) Do not schedule simulator sessions for longer periods than two hours, for any reason; 5) To aid adaptation to a new simulator, aviators should begin with brief simulator hops, flying gentle maneuvers, with subsequent hops separated by one-day intervals (NTSC (1988)). In this context, "brief" means less than one hour, with breaks as needed; 6) Minimize head movement, particularly when new or dynamic maneuvers are being trained; 7) Inform the instructor if you are experiencing discomfort; 8) The instructor should turn off visual display and turn on cabin lights before asking the pilot to exit the simulator; 9) The instructor should decrease the FOV (turn off side displays) during early sessions, nauseogenic maneuvers, or if the pilot shows any symptoms of discomfort; 10) Avoid high-risk activities for at least 12 hours after simulator training.

3 LMR Ride Simulator and Kinematical Definition

Stewart platforms are widely used in driving simulators since they have a high payload and can run with high accelerations. The drawbacks of Stewart platforms are their limited range and no dexterity. The KUKA Robocoaster, manufactured by KUKA Roboter GmbH, has a large motion workspace compared to a Stewart platform. Furthermore, it is ideal to simulate the complex tasks of a flight operation, such as the side step maneuver in a helicopter. Thus, the LMR Driving Simulator was based on the KUKA Robocoaster. This section introduces the equipment of the driving simulator, the reference frames, and the tilt-coordination technique.

3.1 Structure, control equipments and virtual robot

All robot components, including hardware, path-planning, control and visualization, software and TÜV clearance, which are used in this thesis, have been kindly put at disposition by M.Sc. Sebastian Röttgermann and Dr.-Ing Francisco Geu Flores from the LMR group, which is gratefully acknowledged. Fig. 3.1 shows the physical ride simulator used in the Chair of Mechanics and Robotics (Lehrstuhl für Mechanik und Robotik – LMR – in german) at the University of Duisburg-Essen. It consists of a *KUKA KR500/1 TÜV* Robocoaster robot with a *Maurer Söhne* rollercoaster seat (Maurer Rides GmbH (2016)) mounted at the robot flange, a *Visette45* head mounted display unit, an A.R.T motion tracking system with two cameras, a visualization PC, a trajectory-generation PC and a control PC. All of these systems are interconnected via ethernet. The robot system is adapted in order to fulfill the safety rules for human passengers prescribed by the German Technical Control Board *TÜV Süd*. On the robot controller, the optional *KUKA Robot Sensor Interface* technology package is installed, which enables the interaction with the external PCs via ethernet.

The trajectory PC computes the corresponding robot target motion in configuration coordinates as a function of time from the datasets delivered (offline) by the motion cueing algorithms. The robot controller receives the prescribed axis value sets from the trajectory PC at a sampling rate of 12ms. A matching virtual reality environment is computed simultaneously on the visualization PC and displayed in stereo 3D on the head-mounted display with help of a *more3D* plugin (more3d GmbH & Co. KG (2016)). In this virtual reality environment, the viewing angle changes depending on the passenger's head movement which is tracked with the *A.R.T.* camera system. The whole simulation system is coordinated by a control PC which starts all needed

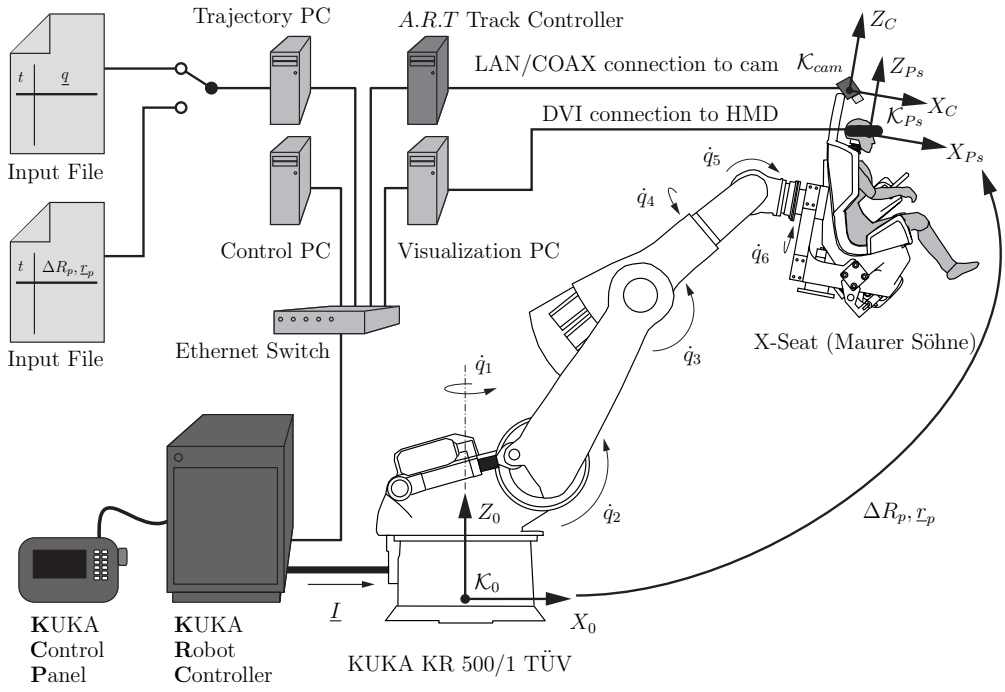


Figure 3.1: Architecture of the LMR driving simulator (Röttgermann, (2014))

devices simultaneously and allows the operator to monitor the simulation online. The program running on the trajectory PC and the visualization environment as well as the control environment program is based on the multibody library *MOBILE* developed by Kecskeméthy (2002) and his group. The KUKA Robocoaster simulation software (KRSS) is linked to the MOBILE library and both simulates the motion of the robot with the specific input trajectory files as well as verifies the boundary conditions of the robot's joints when the robot follows the given trajectory. The trajectory data entered in the robot's controller contains the time history of the relative position and orientation of the end effector (EE).

3.2 Definition of the reference frames

This section presents the reference frames used to develop a MCA in the driving simulator based on the KUKA Robocoaster. A series of five reference frames that are both in a vehicle and in the LMR driving simulator is shown in Fig. 3.2.

3.2.1 Inertial reference frame $\mathcal{K}_0$

The inertial reference frame $\mathcal{K}_0$ is defined at the robot's base with Z_0-axis pointing upwards along the first robot axis and X_0-axis pointing forwards (Fig. 3.1). The

forward direction X_0 is chosen such that the simulation task has symmetric lateral motions with respect to the first axis q_1. The initial position of the robot is given by the joint angles q_i^R, $i \in \{1 \cdots 6\}$

$$\left[\, q_1^R,\, q_2^R,\, q_3^R,\, q_4^R,\, q_5^R,\, q_6^R \right] = [\, 0,\, -21.37,\, -24.26,\, 0,\, 45.63,\, 0] \tag{3.1}$$

which align the EE parallel to the frame $\mathcal{K}_0$ (angles in degrees).

3.2.2 Reference frame $\mathcal{K}_E$ of the end effector

The reference frame $\mathcal{K}_E$ has its origin at the EE of the KUKA Robocoaster. The X_E axis is vertical downwards, the Z_E axis points forwards and the Y_E points to the left hand side. The trajectory data generated by the KUKA Robocoaster simulation software represents the relative location and orientation of the frame $\mathcal{K}_E$ with respect to its initial pose. For the KRSS, the initial position of frame $\mathcal{K}_E$ is given by:

$$^{0}_{0}\underline{r}_{E_0} = (x_{E_0}, y_{E_0}, z_{E_0}) = (2.678,\, 0,\, 2.292)\ (m). \tag{3.2}$$

Hereby, $^{C}_{A}\underline{r}_{B}$ represents the relative position vector of point B w.r.t point A decomposed in the frame $\mathcal{K}_C$.

3.2.3 Simulator reference frame $\mathcal{K}_S$

The simulator reference frame $\mathcal{K}_S$ has its origin at the point selected to match the corresponding MCA being investigated. The origin can be either at the EE of the KUKA Robocoaster or the origin of the frame $\mathcal{K}_{Ps}$ (see below). The X_S-axis points forwards and the Z_S-axis points upwards w.r.t the simulator cockpit, and the Y_S-axis points towards the driver's right hand side. The XY-plane is parallel to the floor of the cockpit.

3.2.4 Vehicle reference frame $\mathcal{K}_V$

The vehicle reference frame $\mathcal{K}_V$ has its origin at the same relative cockpit position as the simulator reference frame $\mathcal{K}_S$. The frame $\mathcal{K}_V$ has the same orientation for the X_V, Y_V, and Z_V axis w.r.t to the cockpit as the simulator frame $\mathcal{K}_S$.

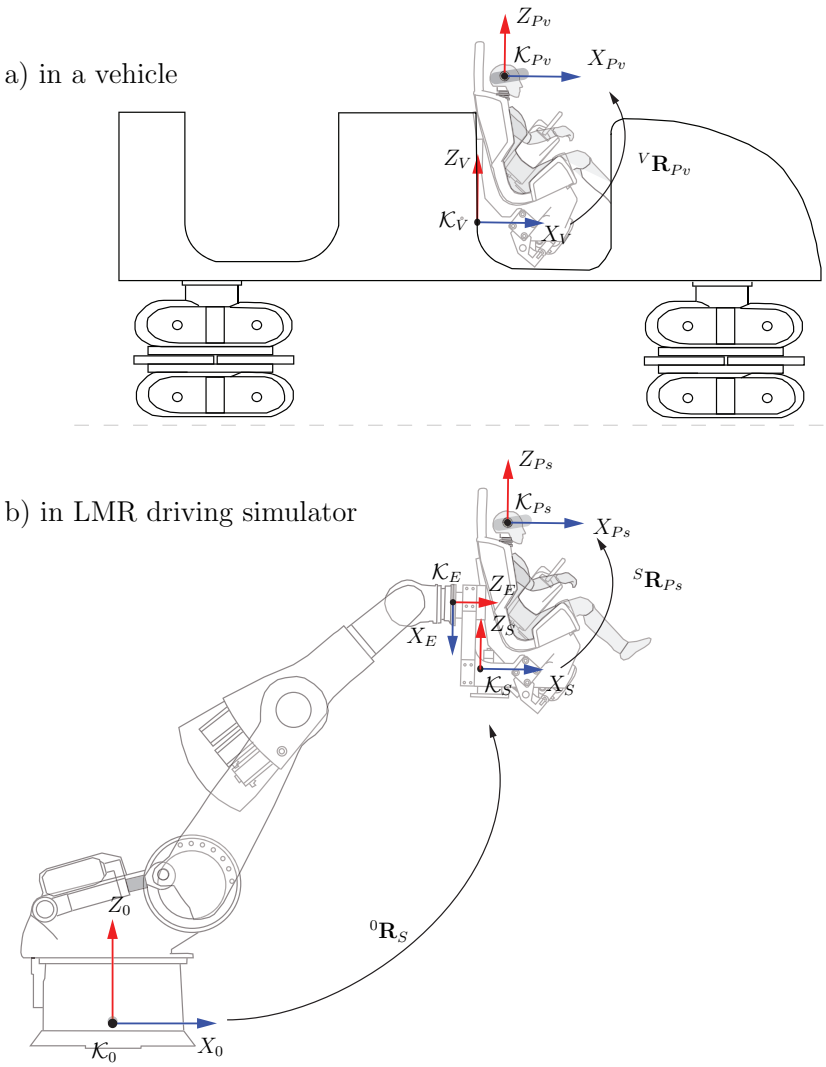


Figure 3.2: Reference frames a) in a vehicle and b) in the LMR driving simulator (adapted from Röttgermann, (2014))

3.2.5 Reference frames $\mathcal{K}_{Ps}$ and $\mathcal{K}_{Pv}$

The frames $\mathcal{K}_{Ps}$ and $\mathcal{K}_{Pv}$ are attached to driver's head in the simulator and vehicle, respectively (see Fig. 3.2). Their origins are located at the midway point between the left and right vestibular systems; their X-axes point forwards and their Z-axes point upwards. In this study, the frames $\mathcal{K}_{Ps}$ and $\mathcal{K}_{Pv}$ are parallel to the frames $\mathcal{K}_S$ and $\mathcal{K}_V$, respectively. The hypothesized position of the frame $\mathcal{K}_{Ps}$ is decomposed in frame $\mathcal{K}_E$ as ${}^{E}_{E}r_{Ps} = (-a_E, 0, b_E)$, where a_E and b_E are depended on the position of the drivers' head w.r.t seat. In this thesis, the position are assumed to be $a_E \approx 0.55\,(m)$ and $b_E \approx 0.35\,(m)$, corresponding to a driver with medium height of $1.7(m)$. The assumptions can be adapted to other drivers according to their biometric data.

3.3 Orientation Equations

The orientation between two reference frames $\mathcal{K}_0$ and $\mathcal{K}_S$ can be specified by Bryant angles $\underline{\beta}_S = [\varphi_S, \theta_S, \psi_S]^{\mathrm{T}}$, representing roll-pitch-yaw convention that defines a sequence of rotations such as

$$ {}^0\mathbf{R}_S = \mathrm{Rot}(Z, \psi_S) \cdot \mathrm{Rot}(Y, \theta_S) \cdot \mathrm{Rot}(X, \varphi_S)\,, \tag{3.3} $$

where $\mathrm{Rot}(a, b)$ is the rotation around the a axis of an angle b (Jazar (2010)). Let $\underline{\omega}_S$ and $\dot{\underline{\beta}}_S$ be angular velocity and tilt rate of Tait-Bryan angle, respectively. Thus, the kinematical Euler equation

$$ \dot{\underline{\beta}}_S = \mathbf{T}_S \, {}^S\underline{\omega}_S \tag{3.4} $$

can be obtained, where

$$ \mathbf{T}_S = \frac{1}{\cos\theta_S} \begin{bmatrix} \cos\theta_S & \sin\theta_S \sin\varphi_S & \sin\theta_S \cos\varphi_S \\ 0 & \cos\theta_S \cos\varphi_S & -\cos\theta_S \sin\varphi_S \\ 0 & \sin\varphi_S & \cos\varphi_S \end{bmatrix} \quad , \tag{3.5} $$

and $\dot{\underline{\beta}}_S = \left[\dot{\varphi},\, \dot{\theta},\, \dot{\psi}\right]^{\mathrm{T}}$. For small angles, the linearization $\dot{\underline{\beta}}_S \approx {}^S\underline{\omega}_S$ can be assumed.

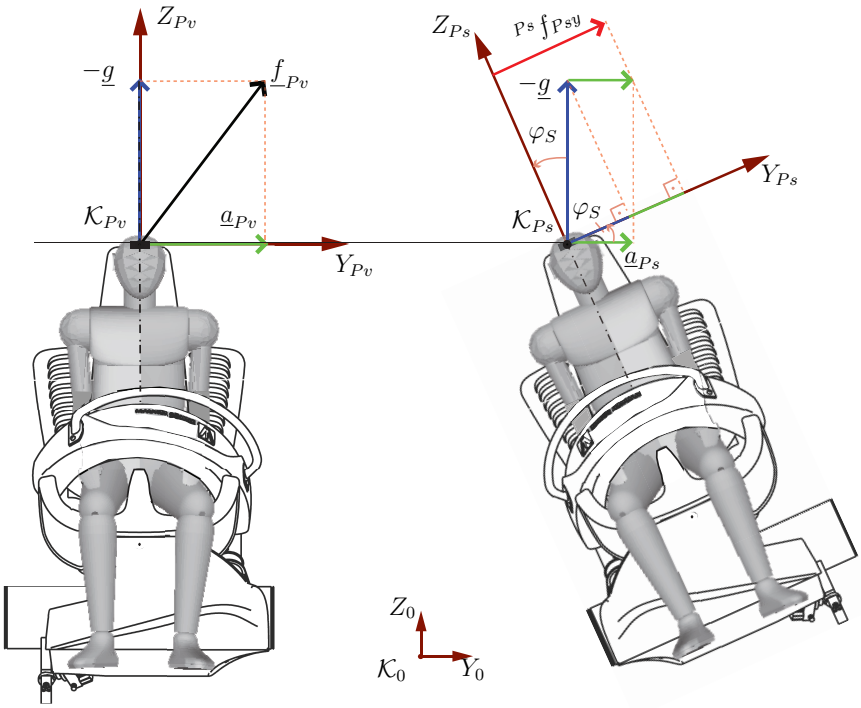
3.4 Tilt-coordination

We assume that only horizontal lateral (left/right) accelerations occur on the target vehicle. According to the literature, the specific force (or g-force) $\underline{f}_{Pv}$ is defined as the total acceleration (including gravity acceleration) the driver perceives while negotiating a given motion. If $\underline{a}_{Pv}$ is the kinematical acceleration of the vehicle, then the total specific force in the target vehicle is (Fig. 3.3a)

$$ \underline{f}_{Pv} = \underline{a}_{Pv} - \underline{g}\,. \tag{3.6} $$

where $\underline{a}_{Pv}$ is the true lateral kinematical acceleration oriented horizontally to the right of the passenger's head; ${}^0\underline{g} = [0, 0, -\mathrm{g}]^{\mathrm{T}}$ is the gravity acceleration.

In order to simulate the perceived vehicle lateral specific force, the cabin is tilted with respect to the horizontal plane, such that the perceived lateral specific force in the local simulator passenger head frame $\mathcal{K}_{Ps}$ is a combination of the kinematical acceleration and the component of the gravity acceleration in lateral direction, neglecting effects in



a) True horizontal lateral acceleration of the target vehicle

b) Simulated perceived lateral specific force at the driver's head

Figure 3.3: Tilt-coordination implemented to simulate a target lateral acceleration $\underline{a}_{Pv}$

normal (cranial/caudal) direction. Writing this in vectorial form gives

$$ {}^{Ps}\underline{f}_{Ps} = {}^{Ps}\underline{a}_{Ps} - {}^{Ps}\underline{g}\,, \tag{3.7} $$

where ${}^{Ps}\underline{a}_{Ps}$ is the kinematical acceleration of the passenger's head and ${}^{Ps}\underline{g}$ is the decomposition of the gravity acceleration in the passenger's head frame $\mathcal{K}_{Ps}$ yielding

$$ {}^{Ps}\underline{g} = {}^{Ps}R_0 = -\mathrm{g}\cdot\left[-\sin\theta_S\,,\cos\theta_S\sin\varphi_S\,,\cos\theta_S\cos\varphi_S\right]^{\mathrm{T}}\,, \tag{3.8} $$

with $\mathrm{g} = 9.81\,m/s^2$. For small Tait-Bryan angles, the coordinates of the simulated specific force ${}^{Ps}\underline{f}_{Ps}$ in the frame $\mathcal{K}_{Ps}$ are thus given by

$$ \begin{bmatrix} {}^{Ps}f_{Psx} \\ {}^{Ps}f_{Psy} \\ {}^{Ps}f_{Psz} \end{bmatrix} = \begin{bmatrix} {}^{Ps}a_{Psx} & - & \mathrm{g}\sin\theta_S \\ {}^{Ps}a_{Psy} & + & \mathrm{g}\cos\theta_S\sin\varphi_S \\ {}^{Ps}a_{Psz} & + & \mathrm{g}\cos\theta_S\cos\varphi_S \end{bmatrix} \approx \begin{bmatrix} {}^{Ps}a_{Psx} & - & \mathrm{g}\theta_S \\ {}^{Ps}a_{Psy} & + & \mathrm{g}\varphi_S \\ {}^{Ps}a_{Psz} & + & \mathrm{g} \end{bmatrix}\,, \tag{3.9} $$

where for motion cueing only the Y-component is retained in the context of the present study.

For the present study, it will be assumed that the vehicle undergoes only lateral kinematical accelerations, which are thus oriented along the left/right axis passing through the vestibular systems at the passenger's head. Uneven target tracks with up/down motion as well as non-steady varying speed (braking, accelerating) are thus neglected (Fig. 3.3a). In the simulator seat, the passenger tilt produces in addition to the component of the kinematical acceleration $\underline{a}_{Ps}$ in passenger's head Y-direction a corresponding component of the gravity acceleration added unconsciously to the kinematical acceleration at the head as an "extended" component f_{Psy} (Fig. 3.3a) in lateral direction, then perceived in total as motion acceleration. In this thesis, it will be assumed that the kinematical seat acceleration $\underline{a}_{Ps}$ is as in the target vehicle case only in horizontal direction. This means that the kinematical seat acceleration will reduce the normal (cranial/caudal) acceleration by a small magnitude $a_{Ps} \sin\varphi_S$, which can be neglected in the same way as the gravity acceleration change $(1 - \cos\varphi_S)\,\mathrm{g}$ due to the tilting of the cabin. Also, rotational effects of the passenger's head acceleration due to angular velocity and angular acceleration are assumed to be negligible due to the narrow thresholds on rotational time derivatives for avoiding false cues, as explained later in this thesis.

In the literature (e.g. Reid and Nahon (1985)), another possibility has been investigated in which the acceleration $\underline{a}_{Ps}$ is applied along the passenger's head lateral direction Y_{Ps} (with $+/-$ sign to simulate left/right lateral acceleration). However, this motion results as a non-holonomic motion prescription similar to a sledge on ice in which the motion in the configuration space becomes unpredictable and may thus yield to violations of the workspace in particular in the vertical direction. For this reason, this approach was not further investigated in this thesis. In appendix A.1 on page 138, a graphical description of the simulated motion that are generated by the classical algorithm for the two options (horizontal vs. tilted $\underline{a}_{Ps}$) is included.

4 Overview and Discussion of Motion Cueing Algorithms in the Literature

A motion cueing algorithm (MCA) uses tilt coordination to reproduce the effect of the sustained translational acceleration motion via the gravity vector such as to copy the human perception sensed by the vestibular system. In this section, a review of MCAs and their implementing diagrams is presented. Then, based on the criteria introduced by Nahon and Reid (1990), a comparison of MCAs' features is discussed.

4.1 Classical washout filters – (CL)

The classical washout filters were initially developed for flight simulators by Conrad and Schmidt (1970) and later applied for general driving simulators. They are mostly used in the commercial simulators (Reid and Nahon (1985, 1988)). Several derived versions of the algorithm were introduced according to the mechanical platforms: 3-DoFs reported by Grant et al. (2003), 8-DoFs reported by Fischer et al. (2010), Jamson (2007, 2010), Dagdelen et al. (2004), Grant et al. (2002), Chapron and Colinot (2007), Grant and Clark (2006). In addition, an algorithm with spherical coordinates was introduced by Wentink et al. (2005), Valente Pais et al. (2009), and one with cylindrical coordinates was later developed by Giordano et al. (2010). This section introduces the traditional CL algorithm for Stewart hexapod platforms that was described in detail by Reid and Nahon (1985). Furthermore, the CL algorithm with cylindrical coordinates for the KUKA Robocoaster introduced by Giordano et al. (2010) is reviewed.

4.1.1 Classical washout filters for simulators based on Stewart hexapods

In the implementing scheme of the CL algorithm (see in Fig. 4.1), the target specific force ${}^{V}\underline{f}_{V}$ and angular velocity ${}^{V}\underline{\omega}_{V}$ are scaled and filtered by high-pass filters HPa and HPω. The high-pass filter HPa fulfills a washout-effect that returns the motion platform to its neutral position (Reid and Nahon (1988)). In the cross channel, the lateral and longitudinal accelerations are firstly scaled and filtered by low-pass filters LP. Then, the sustained (low-frequency) parts of these accelerations are replaced by the tilt angle $\underline{\beta}_{T}$ composed of appropriate roll and pitch angles. These angles are computed by the tilt coordination technique in the "Tilt Coord" block. The "Rate Limit" block is used to restrict the angular velocity of the tilt motion under the threshold values of the semicircular organ.

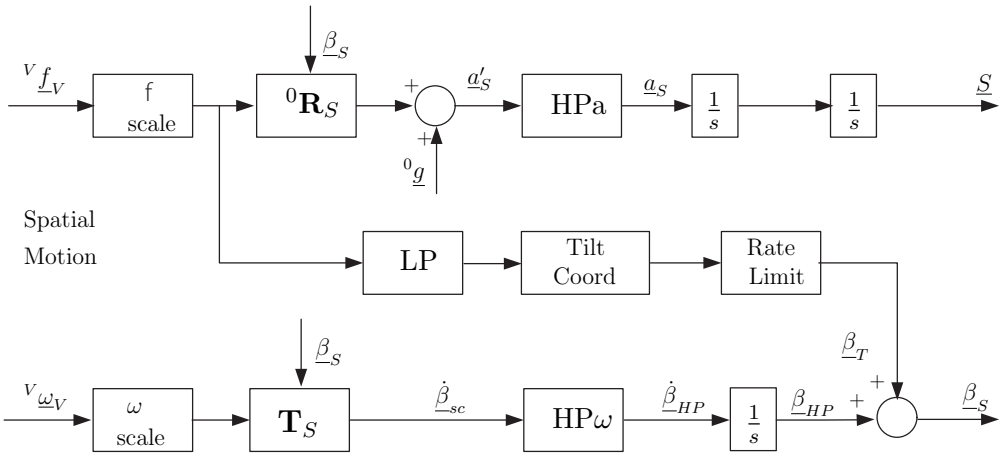


Figure 4.1: CLRN algorithm implementation based on Nahon and Reid (1990) - for Stewart motion platform (Rate limit = threshold)

The transfer functions of the washout filters are:

$$\text{HPa} = \frac{s^2}{s^2 + 2\zeta_{HP}\,\omega_{nHP} s + \omega_{nHP}^2} \cdot \frac{s}{s + \omega_b}\,, \tag{4.1}$$

$$\text{HP}\omega = \frac{s^2}{s^2 + 2\zeta_{HP}\,\omega_{nHP}\,s + \omega_{nHP}^2}\,, \tag{4.2}$$

$$\text{LP} = \frac{\omega_n^2}{s^2 + 2\zeta_{LP}\,\omega_{nLP}\,s + \omega_{nLP}^2}\,. \tag{4.3}$$

Hereby, $\omega_{nHP/nLP}$ is the natural frequency, $\zeta_{HP/LP}$ is the damping ratio, and ω_b is the break frequency.

The first second-order component of HPa is used to generate the acceleration onset cues, and the first-order component is to pull a simulator motion platform back to the null position. In the translational channel, the simulated acceleration $\underline{a}_S$ is the high-frequency part, that is, then, integrated to produce the translational movement $\underline{S}$ of the simulator. In the rotation channel, the scaled rotation rate $\dot{\underline{\beta}}_{sc}$ is first filtered to get only the high-frequency part $\dot{\underline{\beta}}_{HP} = \text{HP}\omega \cdot \dot{\underline{\beta}}_{sc}$. Then, the total rotation angle of the simulator motion platform is $\underline{\beta}_S = \underline{\beta}_{HP} + \underline{\beta}_T\,.$

Nahon and Reid (1990) mentioned that the advantages of the classical algorithm are that the method is mathematically and computationally simple, and hence computationally cheap. In addition, from the designer's point of view, the method is quite transparent. On the other hand, the principal disadvantage of this method is that it includes linear washout filters, therefore it does not effectively exploit the simulator capabilities.

4.1.2 Classical algorithm with cylindrical coordinates – CLG

For the driving simulator based on KUKA Robocoaster, Giordano et al. (2010) introduced the algorithm to exploit a circular motion of the frame $\mathcal{K}_S$ around the Z_0 axis (see Fig. 3.2) to simulate the target lateral acceleration instead of using a linear motion. For the development of the algorithm, the cylindrical coordinates are defined with the unit vectors $\{\underline{e}_r, \underline{e}_\alpha, \underline{e}_z\}$ representing for radial, transversal and vertical directions, respectively. The cylindrical coordinates of a vector $\underline{\xi} = [R,\, \alpha,\, z]^T$ are hereby defined as

$$R = \sqrt{x^2 + y^2}\,; \alpha = \text{atan2}(y, x)\,; z = z \quad . \tag{4.4}$$

The reference frames (shown in Fig. 3.2) are defined with the frame $\mathcal{K}_S \equiv \mathcal{K}_{Ps}$. Moreover, a reference frame $\mathcal{K}_W$ is defined as the washout frame. The origin of the frame $\mathcal{K}_W$ is coincident with that of the frame $\mathcal{K}_0$, for the z direction it holds $Z_W \equiv Z_0$, and the X_W-axis is rotated by the angle α with respect to X_0-axis, such that it always points forwards. The rotation matrix from frame $\mathcal{K}_0$ to frame $\mathcal{K}_W$ is given by ${}^0\mathbf{R}_W = \text{Rot}\,[Z_0,\, \alpha]$.

Considering the high-pass filter for the linear acceleration (see Fig. 4.2), the input to this filter is ${}^V\underline{a}_V$, the linear acceleration of the vehicle expressed in the frame $\mathcal{K}_V$. The acceleration ${}^V\underline{a}_V$ is first scaled and transformed into the washout frame $\mathcal{K}_W$ to obtain the acceleration ${}^W\underline{a}_{V,sc}$. Then, the acceleration is transformed from Cartesian coordinates to cylindrical ones by the conversion matrix $\mathbf{C}(\underline{\xi})$ as

$$\ddot{\underline{\xi}} = \begin{bmatrix} \ddot{R} \\ \ddot{\alpha} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{R} & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^W\underline{a}_{V,sc} = \mathbf{C}(\underline{\xi})\,{}^W\underline{a}_{V,sc} \quad . \tag{4.5}$$

The acceleration $\ddot{\underline{\xi}}$ (in cylindrical coordinates) is filtered by the following high-pass filter HPa defined as

$$\text{HPa}(s) = \frac{s^2}{s^2 + 2\zeta_i\omega_{n,i}s + \omega_{n,i}^2} \cdot \frac{s}{s + \omega_{b,i}}\,, \quad \text{with} \quad i \in \{R\,, \alpha\,, z\}\,, \tag{4.6}$$

where $\omega_{n,i}$ is the natural frequency, ζ_i is the damping ratio, and $\omega_{b,i}$ is the break frequency.

Because the washout process is described in the moving frame $\mathcal{K}_W$, the simulated acceleration decomposed in the frame $\mathcal{K}_W$ includes a centripetal and a Coriolis acceleration. Therefore, expressed in cylindrical coordinates, the simulated acceleration w.r.t $\mathcal{K}_W$ is

$${}^W\underline{a}_S = \left(\ddot{R} - R\dot{\alpha}^2\right)\underline{e}_r + \left(R\ddot{\alpha} + 2\dot{R}\dot{\alpha}\right)\underline{e}_\alpha + \ddot{z}\underline{e}_z\,. \tag{4.7}$$

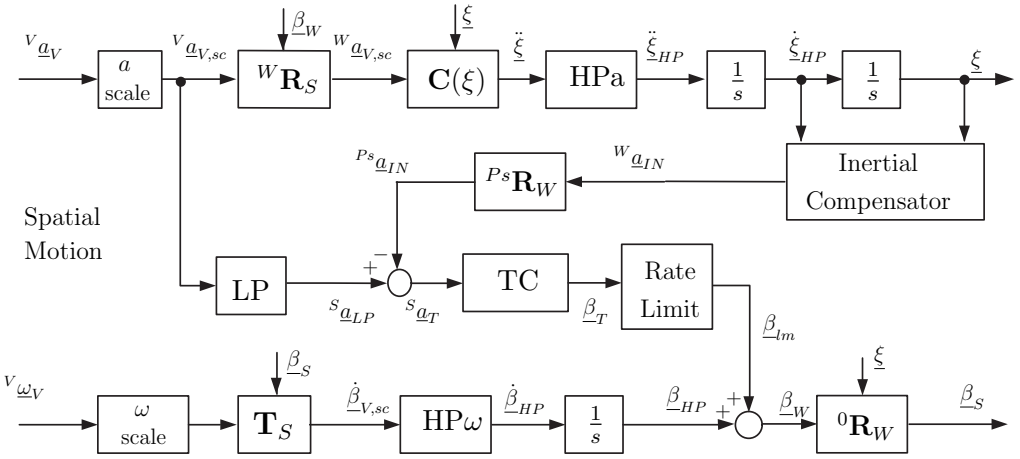


Figure 4.2: CLG algorithm implementation for KUKA Robocoaster motion platform (based on Giordano et al. (2010))

Let ${}^{W}\underline{a}_{IN}$ define the inertial acceleration components representing centripetal and Coriolis acceleration. Then this vector is expressed in the cylindrical coordinates as

$$ {}^{W}\underline{a}_{IN} = -R\dot{\alpha}^2\underline{e}_r + 2\dot{R}\dot{\alpha}\underline{e}_\alpha + 0\underline{e}_z \,. \tag{4.8} $$

In Fig. 4.2, the inertial acceleration ${}^{W}\underline{a}_{IN} = \left[-R\dot{\alpha}^2\,, 2\dot{R}\dot{\alpha}\,, 0\right]^{\mathrm{T}}$ is calculated in the Inertial Compensator block. Note that the centripetal and Coriolis acceleration are considered as the low frequency part of disturbances. Hence, the acceleration is sent to tilt coordinator (TC) block to compute the compensation tilt angle. The input of the TC block ${}^{S}\underline{a}_{T}$, that is computed from the low-pass frequency part of target and spurious acceleration, is then

$$ {}^{S}\underline{a}_{T} = {}^{S}\underline{a}_{LP} - {}^{S}\underline{a}_{IN} \quad , \tag{4.9} $$

where ${}^{S}\underline{a}_{LP}$ is the low frequency part of input acceleration in the frame $\mathcal{K}_S$. Giordano et al. (2010) used the complementary low-pass filter, described as the expression

$$ \mathrm{LPa}(s) = 1 - \mathrm{HPa}(s) \,. \tag{4.10} $$

With the selection of the filters, the rest low-frequency part of the target acceleration can be completely reproduced. Therefore, the simulated specific force can track the target one. Then, to compensate the undesired accelerations and reproduce the sustained (low-frequency) acceleration, the gravity acceleration ${}^{S}\underline{g}$ is used. The idea for the compensation is the same as that of the tilt coordinator (TC block in Fig. 4.2). Thus, the tilt angles $\underline{\beta}_T$ can be calculated such as

$$ \varphi_T = \arcsin\frac{{}^{S}a_{Ty}}{\mathrm{g}\cos\theta_T} \quad ; \theta_T = -\arcsin\frac{{}^{S}a_{Tx}}{\mathrm{g}} \quad ; \psi_T = 0 \quad . \tag{4.11} $$

The tilt angle $\underline{\beta}_T$ enters the "Rate Limit" block to generate $\underline{\beta}_{lm}$ whose tilt rate is limited. Concerning the angular high-pass channel, the input angular velocity ${}^V\underline{\omega}_V$ is first scaled and transformed into an Bryant-angle rotational rate $\dot{\underline{\beta}}_{V,sc}$ which is then filtered by a high-pass filter HPω to obtain the high-frequency part $\dot{\underline{\beta}}_{HP}$. Afterwards, the Bryant-angle rate is integrated to produce the high-frequency angle $\underline{\beta}_{HP}$. Here, $HP\omega$ are selected as a second order high-pass filter

$$\mathrm{HP}\omega(s) = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad . \tag{4.12}$$

For Bryant angles of $\mathcal{K}_S$ with respect to $\mathcal{K}_W$, it holds under the assumption of small angles: $\underline{\beta}_W = \underline{\beta}_{lm} + \underline{\beta}_{HP}$. Finally, the simulated Bryant angles $\underline{\beta}_S$ of the frame $\mathcal{K}_S$ w.r.t the frame $\mathcal{K}_0$ is determined such as $\underline{\beta}_S = \underline{\beta}_W + [0, 0, \alpha]^{\mathrm{T}}$.

Note that for the LMR driving simulator, the parameters of washout filters should be selected to keep radius R and z nearly constant in order to reduce the effect of the spurious components and to avoid potential physical violations.

4.2 Adaptive washout filters

The goal of the adaptive washout filters is to adapt parameters according to the current states of the simulator at each step of simulation time. The algorithm uses gradient-based optimization to find the adaptive parameters that minimize a cost function containing the motion errors and constraints on the motion platform states. As a further development of the coordinated adaptive washout filters introduced by Parrish et al. (1975), Reid and Nahon (1985) suggested an adaptive washout algorithm denoted in this thesis as ADRN. Later, Naseri and Grant (2005) developed a new actuator-state based adaptive algorithm (ASBA) to take advantage of ADRN by using motion system actuator states in the cost function. In an earlier research on the adaptation of MCAs to the LMR driving simulator, Schweig and Kammers (2011) developed the adaptive algorithm named in this thesis as hybrid classical-adaptive washout filters ADSK. The algorithm optimizes parameters of the high-pass filters in the cylindrical coordinate. This section introduces in detail the ADRN and ADSK algorithms.

4.2.1 Coordinated adaptive washout filters – ADRN

The adaptive washout filters are developed for four parallel problems such as: roll/sway, pitch/surge, yaw and heave. In terms of the roll/sway problem, Fig. 4.3 shows the corresponding implementation diagram of the ADRN algorithm. The input signals are

the specific force ${}^{V}\underline{f}_{V}$ and angular velocity ${}^{V}\underline{\omega}_{V}$ of the frame $\mathcal{K}_V$ in the vehicle. The input signals are first scaled and then used to compute the desired acceleration $\underline{a}_d$ and the tilt rate $\underline{\dot{\beta}}_d$ such as

$$\underline{a}_d = {}^{0}\mathbf{R}_S \, {}^{V}\underline{f}_{V,sc} + {}^{0}\underline{g} \quad ; \quad \underline{\dot{\beta}}_d = \mathbf{T}_S \, {}^{V}\underline{\omega}_{V,sc} \, . \tag{4.13}$$

For the roll/sway problem, the desired signals enter the Demux block to decouple the lateral acceleration a_{dy} and roll tilt rate $\dot{\varphi}_d$.

The two blocks "Adaptive #1" and "Adaptive #2" (see Fig. 4.3), are the adaptive washout filters for the sway and roll channel, respectively. The washout filters use the optimized (adaptive) parameters $\underline{p}$ to compute a_{Sy} and φ_S according to filter equations (control law):

$$a_{Sy} = p_1 \, a_{dy} - k_{y1} \, \dot{S}_y - k_{y2} \, S_y \, , \tag{4.14}$$

$$\dot{\varphi}_S = \mathrm{LIM}(p_2 \, a_{dy}) + p_3 \, \dot{\varphi_{dx}} \, , \tag{4.15}$$

where k_{y1} and k_{y2} are constants, and LIM() is a function (in the Limit block) that restricts the tilt rates to yield valid tilt angles to simulate specific force. Additionally, $\underline{p} = \{p_1, p_2, p_3\}$ are the adaptive parameters that minimize the cost function

$$\begin{aligned} J_{ADRN} = & 0.5 \cdot \Big\{ (a_{dy} - a_{Sy})^2 + w_1 \, (\dot{\varphi}_d - \dot{\varphi}_S)^2 + \rho \Big[w_2 \, \dot{S}_y{}^2 + w_3 \, S_y + w_4 \, \dot{\varphi}_S{}^2 \\ & + w_5 \, \varphi_S{}^2 + w_6 \, (p_1 - p_{10})^2 + w_7 \, (p_2 - p_{20})^2 + w_8 \, (p_3 - p_{30})^2 \Big] \Big\} \end{aligned} \tag{4.16}$$

by using the steepest descent method. Thus,

$$p_i = -G_i \cdot \frac{\partial J_y}{\partial p_i} \quad \text{with} \quad (i = 1 \cdots 3) \, . \tag{4.17}$$

Here, $w_{1\ldots8}$ are weighting parameters, S_y is the offset displacement of the simulator; a_{Sy} is the simulated acceleration; and p_{i0}, $i = \{1 \cdots 3\}$ are the initial values of the adaptive parameters p_i.

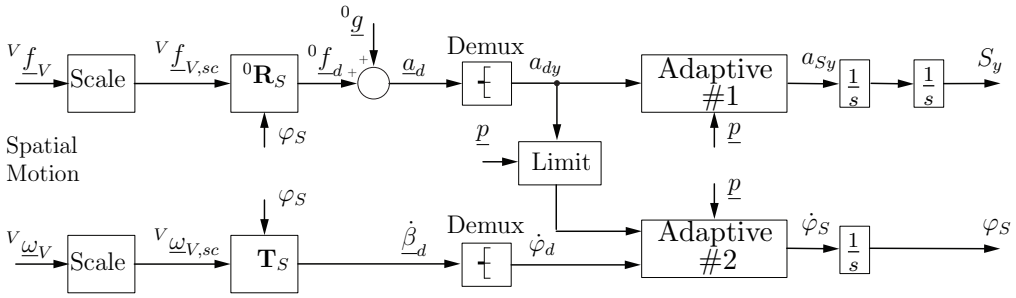


Figure 4.3: ADRN algorithm implementation for roll/sway problem (based on Reid and Nahon (1985))

The cost function is selected so that the simulated acceleration a_{Sy} tracks the desired acceleration a_{dy} and $\dot{\varphi}_S$ tracks $\dot{\varphi}_d$, while the actual motion is restricted by the weighting parameters for the variables $\dot{S}_y$, S_y, $\dot{\varphi}_S$, and φ_S. The form $w_m\,(p_i - p_{i0})$ with $m = \{6 \cdots 8\}\,,\ i = \{1 \cdots 3\}$ is introduced to restrict large deviation of the parameters p_i from the corresponding initial values p_{i0}.

Note that, for the tuning process, the selection of the initial values for parameters p_{i0}, constant G_i, and the weighting values can lead to the unstable state. Thus, the algorithm needs the experience of designers for tuning its parameters. Although the ADRN features a flexible cost function for minimizing the motion cues errors and the diversion of the parameters, the solution of the large number of differential equations is very time-consuming. Moreover, the tuning process of this algorithm is not straightforward since its behavior is difficult to adjust (Nahon and Reid (1990)).

4.2.2 Hybrid classical-adaptive washout filters – ADSK

Based on the CLG algorithm, the adaptive algorithm ADSK (see Fig. 4.4) is developed by inserting an grey "ADAPT" block that optimizes the parameters of the sub-component "high-pass filter" HPa to reduce the motion cues errors. For the algorithm, the high-pass filter is represented as $\mathrm{HPa}(s) = \mathrm{HP}_1(s) \cdot \mathrm{HP}_2(s)$ where

$$\mathrm{HP}_1(s) = \frac{s}{s + \omega_b}, \quad \mathrm{HP}_2(s) = \frac{K s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad . \tag{4.18}$$

The ADSK algorithm uses a steepest descent method in the "ADAPT" block to modify the parameters K, ω_n, and ζ of the high-pass filter HP_2 in real time for minimizing the cost function (Nehaoua et al. (2008))

$$\begin{aligned} J_{ADSK} = \frac{1}{2}\Big[& w_a(\ddot{\underline{\xi}}_F - \ddot{\underline{\xi}}_{HP})^2 + w_v \dot{\underline{\xi}}_{HP}^2 + w_S \underline{\xi}_{HP}^2 \\ & + w_k(K - K_0)^2 + w_\zeta(\zeta - \zeta_0)^2 + w_\omega(\omega_n - \omega_{n0})^2 \Big] \quad . \end{aligned} \tag{4.19}$$

Here, $\ddot{\underline{\xi}}_F$ is the filtered acceleration through the 1st-order filter HP_1; $\{K_0, \zeta_0, \omega_{n0}\}$ are the initial values of the filter parameters; $\{w_a\,, w_v\,, w_S\}$ are weighting parameters penalizing the difference between the target and the actual motion, restraining the translational velocity and displacement of the simulator, respectively. Furthermore, $\{w_k\,, w_\zeta\,, w_\omega\}$ restrict the deviation of the adaptive parameters from their initial values. The minimization process is implemented by using the function `fminunc` in MATLAB® with a quasi-Newton optimization method.

Like the ADRN, the hybrid algorithm can separate the optimal problems into the

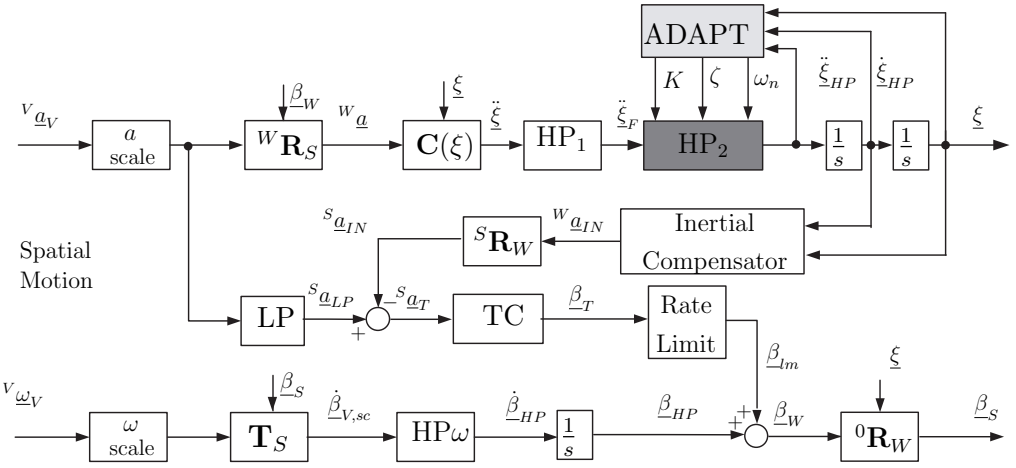


Figure 4.4: ADSK algorithm implementation for KUKA Robocoaster motion platform (based on Schweig and Kammers (2011))

sub-problems of 2-DoF (e.g. roll/sway or pitch/surge). By doing so, the optimization is implemented for channels of the cylindrical coordinates to find the adaptive parameters. However, the algorithm is more time-consuming than the ADRN due to the optimization methods. In terms of tuning parameters, the response of the algorithm is quite opaque regarding the adjusting of the weighting parameters.

4.3 Linear optimal washout filters

Linear optimal washout filters were first developed by Sivan et al. (1982) with four assumptions: 1) The vestibular system dominates the perception of motion cues in a driving simulator; 2) The discrepancy between the motion cues in the actual vehicle and in the moving simulator can be measured by the mean-square value of the vestibular error; 3) The actual vehicle motion can be modeled as a random process with rational spectrum; 4) The dynamical systems, including the vestibular systems, can be represented by linearized equations. Similar to the ADRN, the algorithm solves four problems separately: roll/sway, pitch/surge, heave, and yaw.

The problem structure of the linear optimal algorithm (see Fig. 4.5) includes two separate channels generating motion perception in an actual vehicle $\{\hat{\underline{\omega}}_V, \hat{\underline{f}}_V\}$ and in a simulator $\{\hat{\underline{\omega}}_S, \hat{\underline{f}}_S\}$. In both channels, the vestibular system presents the driver's perception. Concerning the ideal case, the output of the simulator is identical to the input $\underline{u}_S$.

The goal of the optimal algorithm is to determine a transfer function $\mathbf{W}(s)$ consisting of high-order filters. The $\mathbf{W}(s)$ filters the vehicle input $\underline{u}_V$ to obtain the simulator

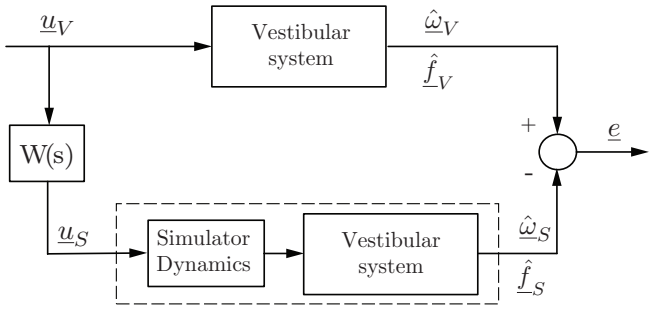


Figure 4.5: Linear optimal algorithm problem structure (based on Sivan et al. (1982))

input $\underline{u}_S$ such that a cost function containing the driver sensation error $\underline{e}$ (between vehicle and simulator) is minimized. The transfer functions $\mathbf{W}(s)$ are generated by an offline program, and then implemented online (Telban et al. (2005)).

Assume that the input $\underline{u}_V$ consists of filtered white noise, and can be expressed as

$$\begin{cases} \dot{\underline{x}}_n &= \mathbf{A}_n \underline{x}_n &+ \mathbf{B}_n \underline{n} \\ \underline{u}_V &= \mathbf{C}_n \underline{x}_n \end{cases}, \tag{4.20}$$

where $\underline{x}_n$ are the filtered white noise states, $\underline{n}$ represents white noise; for the regarded roll/sway problem $\underline{u}_V = \left[{}^V a_{Vy},\, \varphi_V\right]^{\mathrm{T}}$ (Sivan et al. (1982)), $\mathbf{A}_n = \mathrm{diag}\{-\gamma_1, -\gamma_2\}$ and $\mathbf{B}_n = \mathrm{diag}\{\gamma_1, \gamma_2\}$ are state and input matrices of the model of white noise. Here, γ_1 and γ_2 are first-order filter break frequencies for each DoF.

The dynamic model of the optimal problem is represented by the linear state equation

$$\begin{cases} \dot{\underline{x}} &= \mathbf{A}\underline{x} &+ \mathbf{B}\underline{u}_S &+ \mathbf{H}\underline{n} \\ \underline{y} &= \mathbf{C}\underline{x} &+ \mathbf{D}\underline{u}_S \end{cases}, \tag{4.21}$$

where the $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{H}$ are the state space matrices which will be described for each optimal algorithms in below parts. The washout filters $\mathbf{W}(s)$ are determined by solving the linear quadratic optimal control problem with the cost function

$$J_{Op} = E\left\{ \int_{t_0}^{t_f} \left(\underline{e}^{\mathrm{T}} \mathbf{Q} \underline{e} + \underline{u_S}^{\mathrm{T}} \mathbf{R} \underline{u}_S + \underline{x_c}^{\mathrm{T}} \mathbf{R}_c \underline{x}_c \right) dt \right\} \stackrel{!}{=} min\,. \tag{4.22}$$

Here, $E\{\}$ is the statistical mean estimate, and $[t_0, t_f]$ is the simulation time; the diagonal weighting matrices $\mathbf{Q}$, $\mathbf{R}_c$ are the semi-positive definite, while $\mathbf{R}$ is the positive definite diagonal; the output $\underline{y} = \left[\underline{e}^{\mathrm{T}}, \underline{x}_c^{\mathrm{T}}\right]^{\mathrm{T}}$ includes the sensation error $\underline{e}$ and the simulator state variables $\underline{x}_c = \left[\int a_{Sy} dt, \int v_{Sy} dt\right]^{\mathrm{T}}$; $v_{Sy} = \int a_{Sy} dt$ (Sivan et al. (1982)); the sensation errors $\underline{e} = [\Delta \hat{f}_y, \Delta \hat{\omega}_x]^{\mathrm{T}}$ (roll/sway problem) are represented in terms of the specific force and angular velocity, respectively, between the expected output of the passenger's vestibular system and the target values (with $\mathcal{K}_S \equiv \mathcal{K}_{Ps}$ & $\mathcal{K}_V \equiv \mathcal{K}_{Pv}$):

$$\Delta \hat{f}_y = {}^V \hat{f}_{Vy} - {}^S \hat{f}_{Sy}; \quad \Delta \hat{\omega}_x = {}^V \hat{\omega}_{Vx} - {}^S \hat{\omega}_{Sx}\,. \tag{4.23}$$

This thesis reviews 4 sets of optimal washout filters for the roll/sway problem. The sets were developed by Sivan et al. (1982), Reid and Nahon (1985), and Telban et al. (2005) (two sets). Note that each state space model has different state variables $\underline{x}$, system inputs $\underline{u}$, and simulator variables $\underline{x}_c$, that are included in Tab. B.3 on page 141. Furthermore, the parameters of the vestibular system models are summarized in Tab. B.1 and B.2 on page 141.

4.3.1 Linear optimal washout filters – OpS

Sivan et al. (1982) used the following models of semicircular and otolith organs:

$$H_{Scc}(s) = \frac{\hat{\omega}_{Sx}}{\omega_{Sx}} = G_S \frac{s}{(\tau_1 s + 1)(\tau_2 s + 1)}\,, \tag{4.24}$$

$$H_{Oto}(s) = \frac{{}^S\hat{f}_{Sy}}{{}^S f_{Sy}} = G_O \frac{(s + A_0)}{(s + B_0)}\,, \tag{4.25}$$

where G_S and G_O are the gain factors in threshold units for the semicircular and otolith organs, respectively; τ_1 and τ_2 are the time parameters of the semicircular organ; and A_0 and B_0 are the time parameters of the otolith organ.

Referring to Eq. 4.21, the authors define the state variables $\underline{x} = \left[\underline{x}_V^{\mathrm{T}}, \underline{x}_S^{\mathrm{T}}, \underline{x}_n^{\mathrm{T}}\right]^{\mathrm{T}}$, where $\underline{x}_n = [{}^V a_{Vy}, \varphi_V]$ is the white noise state variable; and $\underline{x}_V$ is the state variable that includes the variables of the vestibular modes, velocity, and position of the vehicle

$$\underline{x}_V = \left[(\underline{x}_V^{Oto})^{\mathrm{T}}, (\underline{x}_V^{Scc})^{\mathrm{T}}, \int v_{Vy} dt, \int a_{Vy} dt\right]^{\mathrm{T}}; v_{Vy} = \int a_{Vy} dt\,. \tag{4.26}$$

Hereby, the otolith state variable is $\underline{x}_V^{Oto} = \left[{}^V\hat{f}_{Vy} - {}^V f_{Vy} G_O B_0\right]$, and the semi-circular state variable is $\underline{x}_V^{Scc} = \left[{}^V\hat{\omega}_{Vx} - G_S b_S \varphi_V, \int {}^V\hat{\omega}_{Vx}\right]^{\mathrm{T}}$.

From the model displayed in Fig. 4.5, if the simulator dynamics are ignored, the state variable in the simulator $\underline{x}_S$ is similar to $\underline{x}_V$ (with subscript "V" is replaced by "S"):

$$\underline{x}_S = \left[(\underline{x}_S^{Oto})^{\mathrm{T}}, (\underline{x}_S^{Scc})^{\mathrm{T}}, \int v_{Sy} dt, \int a_{Sy} dt\right]^{\mathrm{T}}; \; v_{Sy} = \int a_{Sy} dt\,. \tag{4.27}$$

The matrices $\bar{\mathbf{A}}$, $\bar{\mathbf{B}}$, $\bar{\mathbf{C}}$, $\bar{\mathbf{D}}$ of the state space model of the vehicle/simulator model use the vestibular model described by state matrices $\widetilde{\mathbf{A}}$, $\widetilde{\mathbf{B}}$, $\widetilde{\mathbf{C}}$, $\widetilde{\mathbf{D}}$. For example, the approach of Sivan et al. (1982) defined

$$\bar{\mathbf{A}} = \begin{bmatrix} \widetilde{\mathbf{A}} & 0 \\ 0 & \mathbf{A_c} \end{bmatrix}, \bar{\mathbf{B}} = \begin{bmatrix} \widetilde{\mathbf{B}} \\ \mathbf{B_c} \end{bmatrix}, \bar{\mathbf{C}} = \begin{bmatrix} \widetilde{\mathbf{C}} \\ 0 \end{bmatrix}, \bar{\mathbf{D}} = \widetilde{\mathbf{D}}\,. \tag{4.28}$$

where

$$\begin{aligned}
\widetilde{\mathbf{A}} &= \begin{bmatrix} -B_0 & 0 & 0 \\ 0 & -A_S & 1 \\ 0 & -B_S & 0 \end{bmatrix}, \widetilde{\mathbf{B}} = \begin{bmatrix} G_O(A_0 - B_0) & -G_O \mathrm{g}(A_0 - B_0) \\ 0 & -A_S B_S G_S \\ 0 & -B_S^2 G_S \end{bmatrix}, \\
\widetilde{\mathbf{C}} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \widetilde{\mathbf{D}} = \begin{bmatrix} G_O & -G_O \mathrm{g} \\ 0 & G_S B_S \end{bmatrix}, \mathbf{A_c} = \begin{bmatrix} 0^* & 1 \\ 0 & 0^* \end{bmatrix}, \mathbf{B_c} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \quad ,
\end{aligned} \tag{4.29}$$

and, $A_S = (\tau_1 + \tau_2)/\tau_1\tau_2$, and $B_S = 1/\tau_1\tau_2$.

The vestibular model was computed by making the assumption: the angles φ_V and φ_S are so small that $\sin\varphi_S \approx \varphi_S$ and $\sin\varphi_V \approx \varphi_V$ along with $\cos\varphi_S \approx 1$ and $\cos\varphi_V \approx 1$. Moreover, for numerical convenience Sivan et al. (1982) defined $0^* = 10^{-5}$.

The augmented matrices of the linear optimal system (Eq. 4.21) are obtained as

$$\begin{aligned}
\mathbf{A} &= \begin{bmatrix} \bar{\mathbf{A}} & 0 & \bar{\mathbf{B}}\mathbf{C}_n \\ 0 & \bar{\mathbf{A}} & 0 \\ 0 & 0 & \mathbf{A}_n \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ \bar{\mathbf{B}} \\ 0 \end{bmatrix}, \mathbf{H} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{B}_n \end{bmatrix}, \\
\mathbf{C} &= \begin{bmatrix} -\bar{\mathbf{C}} & \bar{\mathbf{C}} & -\bar{\mathbf{D}}\bar{\mathbf{C}}_n \\ 0 & \mathbf{C}_c & 0 \end{bmatrix}, \mathbf{D} = \bar{\mathbf{D}}\,,
\end{aligned} \tag{4.30}$$

where $\mathbf{C}_c = [0,\, \mathbf{I}]$ for two output states of the simulator, velocity and displacement.

The problem of minimizing the cost function J_{Op} (Eq. 4.22) subject to the state space equation Eq. 4.21 is a standard stochastic state feedback optimization problem. The solution to this problem is (Kwakernaak and Sivan (1972))

$$\underline{u}_S = -\mathbf{F}\underline{x} = [\mathbf{F}_1\,, \mathbf{F}_2\,, \mathbf{F}_3]\,[\underline{x}_V,\, \underline{x}_S,\, \underline{x}_n]^{\mathrm{T}}\,, \tag{4.31}$$

Here, the feedback matrix $\mathbf{F} = [\mathbf{F}_1,\, \mathbf{F}_2,\, \mathbf{F}_3]$ includes the three sub-matrices corresponding to the sub-vectors $\underline{x}_V$, $\underline{x}_S$, and $\underline{x}_n$. The feedback matrix $\mathbf{F}$ is computed as

$$\begin{aligned}
\mathbf{F} &= \mathbf{R_2^{-1}}\left(\mathbf{B^T P} + \mathbf{R_{12}^T}\right), \\
\mathbf{R_1} &= \mathbf{C^T Q C}\,, \mathbf{R_{12}} = \mathbf{C^T Q D}\,, \mathbf{R_2} = \mathbf{R} + \mathbf{D^T Q D}\,,
\end{aligned} \tag{4.32}$$

with $\mathbf{P}$ being the unique non-negative definite solution of the algebraic Riccati equation

$$\begin{aligned}
\mathbf{P}\left(\mathbf{A} - \mathbf{B}\mathbf{R_2^{-1}}\mathbf{R_{12}^T}\right) + \left(\mathbf{A} - \mathbf{B}\mathbf{R_2^{-1}}\mathbf{R_{12}^T}\right)^{\mathbf{T}}\mathbf{P} + \mathbf{R_1} \\
-\mathbf{R_{12}}\mathbf{R_2^{-1}}\mathbf{R_{12}^T} - \mathbf{P}\mathbf{B}\mathbf{R_2^{-1}}\mathbf{B^T P} = 0\,.
\end{aligned} \tag{4.33}$$

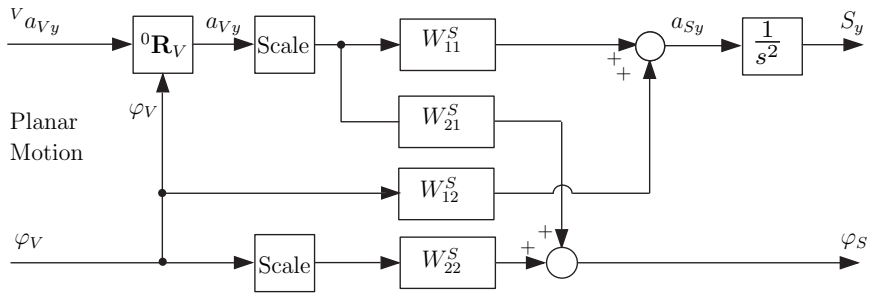


Figure 4.6: OpS algorithm implementation for roll/sway problems (based on Sivan et al. (1982))

The washout filter $\mathbf{W}(s)$ is obtained by using the expression

$$\mathbf{W}^{S}(s) = -\left(\mathbf{I} + \mathbf{F_2}(s\mathbf{I} - \bar{\mathbf{A}})^{-\mathbf{1}}\bar{\mathbf{B}}\right)^{-\mathbf{1}} \cdot \left(\mathbf{F_1}(s\mathbf{I} - \bar{\mathbf{A}})^{-\mathbf{1}}\bar{\mathbf{B}} + \mathbf{F_3}\right) \quad , \tag{4.34}$$

and its elements denote as high-order washout filters

$$W_{ij}(s) = \frac{(s + a_1^{ij})(s + a_2^{ij}) \cdots (s + a_n^{ij})}{(s + b_1^{ij})(s + b_2^{ij}) \cdots (s + b_n^{ij})} \quad \text{with} \quad i,\, j = \{1,\, 2\}. \tag{4.35}$$

Here, $a_{(\cdot)}^{ij}, b_{(\cdot)}^{ij}$ are the zeros and poles of the washout filter $W_{ij}(s)$, respectively, and n is the dimension of the matrix $\bar{\mathbf{A}}$.

Fig. 4.6 shows that target acceleration $^{V}a_{Vy}$ is transformed through the block $^{0}\mathbf{R}_{V}$ to inertial coordinates and then scaled by a suitable gain factor. Next, the scaled acceleration is filtered through W_{11}^{S} to produce the simulator acceleration a_{Sy}. This acceleration is integrated twice to generate the simulator offset S_y. The roll tilt angle φ_S is obtained as the sum of the tilt angle formed from the inertial acceleration being passed through the tilt coordination filter W_{21}^{S}, and the filtered roll target angle through W_{22}^{S}. The filter W_{12}^{S} always has extremely small gain and can, therefore, be neglected.

4.3.2 Linear optimal washout filters – OpRN

Reid and Nahon (1985) used the same models of semicircular and otolith organs as Sivan et al. (1982) (transfer functions Eq. 4.24 and 4.25). However, the authors used different parameters values for the gain factors G_O, and G_S than Sivan et al. (1982) (see Tab. B.1 and B.2). The input is $\underline{u}_V = \left[\varphi_V,\, ^{V}a_{Vy}\right]^{\mathrm{T}}$, and the simulator variables are $\underline{x}_c = \left[\int S_y dt,\, \int v_{Sy} dt,\, \int a_{Sy} dt\right]^{\mathrm{T}}$; $v_{Sy} = \int a_{Sy} dt$; $S_y = \int v_{Sy} dt$. Therefore, the state space matrices of the vestibular system model and those of the simulator (as the Eq. 4.29) are

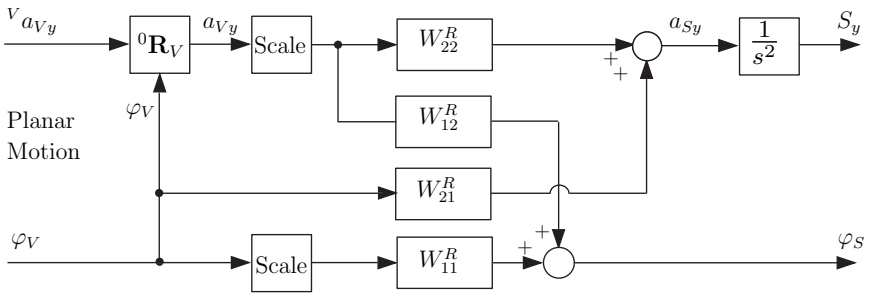


Figure 4.7: OpRN algorithm implementation for roll/sway problem (based on Reid and Nahon (1985))

$$\widetilde{\mathbf{A}} = \begin{bmatrix} -a_S & 1 & 0 \\ -b_S & 0 & 0 \\ 0 & 0 & -b_0 \end{bmatrix}, \widetilde{\mathbf{B}} = \begin{bmatrix} -a_S b_S G_S & 0 \\ -b_S^2 G_S & 0 \\ -G_O \mathrm{g}(A_0 - B_0) & G_O(A_0 - B_0) \end{bmatrix},$$
$$\widetilde{\mathbf{C}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \widetilde{\mathbf{D}} = \begin{bmatrix} G_S b_S & 0 \\ -G_O \mathrm{g} & G_O \end{bmatrix}, \mathbf{A_c} = \begin{bmatrix} 0^* & 1 & 0 \\ 0 & 0^* & 1 \\ 0 & 0 & 0^* \end{bmatrix}, \mathbf{B_c} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}. \tag{4.36}$$

The augmented matrices of the linear optimal system are identical to Eq. 4.30. In addition, the optimal washout filters are computed by Eq. 4.34 with reference to Eq. 4.33. The washout filters W_{ij}^R, with $\{i, j = 1, 2\}$ are computed as Sivan's method (see section 4.3.1), afterward the filters are implemented to generate the simulated signals, as illustrated in Fig. 4.7. Like in the case of the OpS algorithm, the small gain of W_{21}^R cause an insignificant effect of this filter on the simulated signals, and it is therefore removable. Note that the indices of W_{ij}^R, with $\{i, j = 1, 2\}$ in Fig. 4.7 are different from those of W_{ij}^S in Fig. 4.6. The reason is that Reid and Nahon (1985) used the inputs $\underline{u}_V = \left[\varphi_V, {}^{V}a_{Vy}\right]^{\mathrm{T}}$, while Sivan et al. (1982) used the inputs $\underline{u}_V = \left[{}^{V}a_{Vy}, \varphi_V\right]^{\mathrm{T}}$.

4.3.3 Linear optimal washout filters – OpT and OpTYM

Telban et al. (2005) developed two MCAs, one – OpT – which uses their own proposed model for the otolith organ and the other – OpTYM – which uses the otolith model of Young and Meiry (1968).

$$H_{Scc}(s) = \frac{\hat{\omega}_{Sx}}{\omega_{Sx}} = G_S \frac{\tau_a \tau_1 s^2 (1 + \tau_L s)}{(\tau_1 s + 1)(\tau_2 s + 1)(\tau_a s + 1)}, \tag{4.37}$$

$$H_{Oto}(s) = \frac{{}^{S}\hat{f}_{Psy}}{{}^{S}f_{Psy}} = G_O \frac{s + A_0}{(s + B_0)(s + B_1)}. \tag{4.38}$$

Here, Telban's model has different parameters from Young-Meiry's model, as shown in

Tab. B.1, and B.2. Moreover, for the two algorithms, the origin of the frame $\mathcal{K}_S$ (the center of rotation) is not at the driver's head. In terms of the LMR driving simulator, the origin of the frame $\mathcal{K}_S$ is selected at the EE.

For the selection of the frame $\mathcal{K}_S$, the y-component of the simulated acceleration decomposed in the frame $\mathcal{K}_S$ is computed for small tilt angle φ_S as ${}^S a_{Sy} = a_{Sy}\cos\varphi_S \approx a_{Sy}$ (see Fig. 3.3). The simulated specific force for roll/sway problem is then

$$ {}^S f_{Psy} = {}^S a_{Sy} + \mathrm{g}\varphi_S - r_{Sz}\ddot{\varphi}_S \approx a_{Sy} + \mathrm{g}\varphi_S - r_{Sz}\ddot{\varphi}_S \,, \tag{4.39} $$

with r_{Sz} is the distance between the center of rotation and the driver's head in the Z-axis direction. The simulator input is given by $\underline{u}_S = [u_1,\, u_2]^{\mathrm{T}} = [\dot{\varphi}_S,\, a_{Sy}]^{\mathrm{T}}$. By replacing Eq. 4.39 in 4.38, and taking derivatives and rearranging, it results in

$$ \begin{aligned} &{}^S\dddot{\hat{f}}_{Psy} + (B_0 + B_1)\,{}^S\ddot{\hat{f}}_{Psy} + B_1 B_0\,{}^S\dot{\hat{f}}_{Psy} = \\ &G_O\Big[- r_{Sz}(B_0 + B_0 - A_0)u_1 + (\mathrm{g} - r_{Sz}B_0B_1)u_1 + \mathrm{g}A_0 \int u_1 dt + \dot{u}_2 + A_0 u_2\Big], \end{aligned} \tag{4.40} $$

which can be rewritten as

$$ {}^S\dddot{\hat{f}}_{Psy} + a\,{}^S\ddot{\hat{f}}_{Psy} + b\,{}^S\dot{\hat{f}}_{Psy} = \left[cu_1 + du_1 + e\int u_1 dt + \dot{u}_2 + A_0 u_2\right], \tag{4.41} $$

or, in the state space notation, as

$$ \begin{cases} \dot{\underline{x}}_S^{Oto} &= \mathbf{A}_{Oto}\underline{x}_S^{Oto} + \mathbf{B}_{Oto}\underline{u}_S \\ \hat{f}_{Psy} &= \mathbf{C}_{Oto}\underline{x}_S^{Oto} + \mathbf{D}_{Oto}\underline{u}_S \end{cases}, \tag{4.42} $$

where $\underline{x}^{Oto}$ are the otolith states, and

$$ \mathbf{A}_{Oto} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -b & -a & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -b & -a \end{bmatrix}; \qquad \mathbf{B}_{Oto} = \begin{bmatrix} c & 0 \\ d - ac & 0 \\ e & 0 \\ 0 & f \\ 0 & h - af \end{bmatrix}; \tag{4.43} $$

$$ \mathbf{C}_{Oto} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \end{bmatrix}; \qquad \mathbf{D}_{Oto} = \begin{bmatrix} G_O r_{Sz} & 0 \end{bmatrix}. \tag{4.44} $$

Here,

$$\begin{aligned} K_{Oto} &= 0.4\frac{B_0 B_1}{A_0}\,; & G_{Oto} &= \frac{1}{\delta_{Oto}}\,; \quad G_O = G_{Oto}K_{Oto}\,; \\ a &= B_0 + B_1\,; & b &= B_0 B_1\,; \\ c &= -G_O r_{Sz}(B_0 + B_0 - A_0)\,; & d &= G_O(\mathrm{g} - r_{Sz}B_0 B_1)\,; \\ e &= \mathrm{g}G_O A_0\,; & f &= G_O\,; h = G_O A_0\ . \end{aligned} \tag{4.45}$$

For the OpTYM algorithm, the Eq. 4.37 can be rewritten as

$$\frac{\hat{\dot{\varphi}}_S}{\dot{\varphi}_S} = \frac{T_4 s^3 + T_3 s^2}{s^3 + T_2 s^2 + T_1 s + T_0}\,, \tag{4.46}$$

where

$$\begin{aligned} T_0 &= \frac{1}{\tau_a \tau_1 \tau_2}\,; & T_1 &= \frac{\tau_a + \tau_1 + \tau_2}{\tau_a \tau_1 \tau_2}\,; & T_2 &= \frac{\tau_1\tau_2 + \tau_a(\tau_1 + \tau_2)}{\tau_a \tau_1 \tau_2}\,; \\ T_3 &= \frac{G_S}{\tau_2}\,; & T_4 &= G_S\frac{\tau_L}{\tau_2} & & . \end{aligned}$$

The transfer function is rewritten in terms of state-space matrices as

$$\mathbf{A}_{Scc}^{Y} = \begin{bmatrix} -T_2 & 1 & 0 \\ -T_1 & 0 & 1 \\ -T_0 & 0 & 0 \end{bmatrix}; \qquad \mathbf{B}_{Scc}^{Y} = \begin{bmatrix} T_3 - T_2 T_4 & 0 \\ -T_1 T_4 & 0 \\ -T_0 T_4 & 0 \end{bmatrix}; \tag{4.47}$$

$$\mathbf{C}_{Scc}^{Y} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}; \qquad \mathbf{D}_{Scc}^{Y} = \begin{bmatrix} T_4 & 0 \end{bmatrix}. \tag{4.48}$$

On the other hand, the OpT algorithm uses a reduced-order transfer function of the semicircular organ, in which both time constant τ_L and τ_2 are removed. As a result, the state-space matrices for the OpT algorithm are:

$$\mathbf{A}_{Scc}^{Te} = \begin{bmatrix} -T_1^* & 1 \\ -T_0^* & 0 \end{bmatrix}; \mathbf{B}_{Scc}^{Te} = \begin{bmatrix} -G_S\tau_1 & 0 \\ -G_S\tau_0 & 0 \end{bmatrix}; \mathbf{C}_{Scc}^{Te} = \begin{bmatrix} 1 & 0 \end{bmatrix}; \mathbf{D}_{Scc}^{Te} = \begin{bmatrix} G_S & 0 \end{bmatrix}, \tag{4.49}$$

where $T_0^* = \frac{1}{\tau_a \tau_1}$ and $T_1^* = \frac{\tau_a + \tau_1}{\tau_a \tau_1}$. The human vestibular model is a combination of the otolith and the semicircular models, which is represented as

$$\widetilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_{Scc}^{k} & 0 \\ 0 & \mathbf{A}_{Oto} \end{bmatrix}; \widetilde{\mathbf{B}} = \begin{bmatrix} \mathbf{B}_{Scc}^{k} \\ \mathbf{B}_{Oto} \end{bmatrix}; \widetilde{\mathbf{C}} = \begin{bmatrix} \mathbf{C}_{Scc}^{k} & 0 \\ 0 & \mathbf{C}_{Oto} \end{bmatrix}; \widetilde{\mathbf{D}} = \begin{bmatrix} \mathbf{D}_{Scc}^{k} \\ \mathbf{D}_{Oto} \end{bmatrix}, \tag{4.50}$$

with $k \in \{Y, Te\}$.Telban et al. (2005) defined different state variables for the control system (Eq. 4.21) compared to Sivan et al. (1982). Concretely, the combined states are $\underline{x} = \left[\underline{x}_e^{\mathrm{T}}, \underline{x}_c^{\mathrm{T}}, \underline{x}_n^{\mathrm{T}}\right]^{\mathrm{T}}$. The state of errors $\underline{x}_e = \underline{x}_S - \underline{x}_V$, with $\underline{x}_S = \left[(\underline{x}_S^{Oto})^{\mathrm{T}}, (\underline{x}_S^{Scc})^{\mathrm{T}}\right]^{\mathrm{T}}$ and similarly for $\underline{x}_V$, corresponds to the respective vestibular states whose system matrices

are shown in Eq. 4.50 for the simulator and the vehicle, respectively; the simulator states are $\underline{x}_c = \left[\int S_y dt,\, \int v_{Sy} dt,\, \int a_{Sy} dt,\, \varphi_S\right]^{\mathrm{T}}$; $v_{Sy} = \int a_{Sy} dt$; $S_y = \int v_{Sy} dt$; and $\underline{x}_n$ are the filtered white noise states (see Eq. 4.20).

The combined system matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$, $\mathbf{H}$ are then given by

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \widetilde{\mathbf{A}} & 0 & -\widetilde{\mathbf{B}} \\ 0 & \mathbf{A}_c & 0 \\ 0 & 0 & \mathbf{A}_n \end{bmatrix}; \mathbf{B} = \begin{bmatrix} \widetilde{\mathbf{B}} \\ \mathbf{B}_c \\ 0 \end{bmatrix}; \mathbf{H} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{B}_n \end{bmatrix}; \\ \mathbf{C} &= \begin{bmatrix} \widetilde{\mathbf{C}} & 0 & -\widetilde{\mathbf{D}} \\ 0 & 1 & 0 \end{bmatrix}; \mathbf{D} = \begin{bmatrix} \widetilde{\mathbf{D}} \\ 0 \end{bmatrix}. \end{aligned} \tag{4.51}$$

Like the OpS algorithm, the cost function J_{Op} (Eq. 4.22) is minimized by $\underline{u}_S = -\mathbf{F}\underline{x}$, where, the feedback matrix $\mathbf{F} = [\mathbf{F}_1,\, \mathbf{F}_2,\, \mathbf{F}_3]$ includes the three sub-matrices corresponding to the sub-vectors $\{\underline{x}_e,\, \underline{x}_c,\, \underline{x}_n\}$, respectively. The feedback matrix $\mathbf{F}$ is computed as

$$\begin{aligned} \mathbf{F} = [\mathbf{F_1}\,, \mathbf{F_2}\,, \mathbf{F_3}] &= \mathbf{R_2^{-1}}\left(\mathbf{B^T P} + \mathbf{R_{12}^T}\right), \\ \mathbf{R_1} = \mathbf{C^T Q C}\,, \mathbf{R_{12}} = \mathbf{C^T G D}\,, \mathbf{R_2} &= \mathbf{R} + \mathbf{D^T Q D}\,, \mathbf{G} = \mathbf{diag}\left[\mathbf{Q}, \mathbf{R_c}\right]. \end{aligned} \tag{4.52}$$

Here, $\mathbf{P}$ is the unique non-negative definite solution of the Riccati equation (see Eq. 4.33).

The washout filter matrix $\mathbf{W}^k(s)$ with $k \in \{Y, Te\}$ is determined by

$$\mathbf{W}^k(s) = [\mathbf{F_1},\, \mathbf{F_2}] \begin{bmatrix} s\mathbf{I} - \widetilde{\mathbf{A}} + \widetilde{\mathbf{B}}\mathbf{F_1} & \widetilde{\mathbf{B}}\mathbf{F_2} \\ \mathbf{B_c F_1} & \mathbf{sI - A_c + B_c F_2} \end{bmatrix}^{-1} \begin{bmatrix} \widetilde{\mathbf{B}}(\mathbf{I} + \mathbf{F_3}) \\ \mathbf{B_c F_3} \end{bmatrix} - \mathbf{F_3}\,, \tag{4.53}$$

The washout filter matrix $\mathbf{W}^k(s)$ is implemented for online simulation as shown in Fig. 4.8. The vehicle acceleration is first transformed to the inertial frame $\mathcal{K}_0$. The nonlinear scaling block is then applied to scale down the inputs. Then, the scaled inertial acceleration is filtered by W_{22}^k to generate the simulated acceleration a_{Sy}. If the input signal is only lateral acceleration, the simulated angular velocity combines the tilt-rate that is produced by filtering a_{Vy} through W_{12}^k. The tilt rate limit is used to maintain the angular velocity under the threshold values.

Because of their fixed parameters, the optimal washout filters must be tuned for the worst-case maneuvers and often generate minimal motion under more gentle maneuvering. Note that not only the weighting parameters but also the suitable factor for the nonlinear scaling block must be tuned to archive high fidelity simulation.

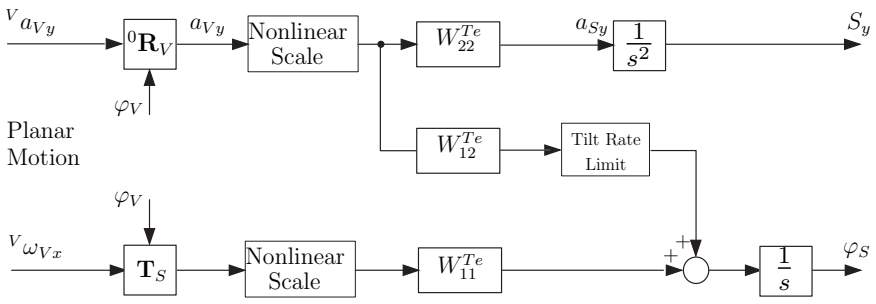


Figure 4.8: OpT algorithm implementation for roll/sway problem (based on Telban et al. (2005))

4.4 Nonlinear optimal washout filters – OpTNon

The linear optimal washout filters are usually determined for the worst-case maneuvers. They are therefore not adapted to the actual maneuvers and potential false cues have a high change of occurring. Cardullo and Kosut (1983) and Ish-Shalom (1982) suggested a nonlinear approach to overcome these difficulties. The approach combines the ideas of the adaptive and optimal washout filters to maximize motion cue fidelity in a driving simulator. Based on the approach, Telban et al. (2005) proposed a nonlinear optimal washout filter whose block diagram of control problem is illustrated in Fig. 4.9. The algorithm is to minimize the perceptual errors $\underline{e}$ by finding the optimal washout filters to generate a suitable simulator input $\underline{u}_S$. The washout filters were found at each time step with the feedback information of the simulator states. To do so, the Riccati Equation Solver needs to be implemented in real-time to find the optimal input $\underline{u}_S$ at each time step. Furthermore, both a vestibular and an optokinetic model were included in the Perceptual System block. Therefore, not only the specific forces and the angular velocity (by the vestibular system) but also the velocity of rotational $\hat{\dot{\varphi}}$ and translational perception $\hat{v}_y$ (by the visual system) were taken into account.

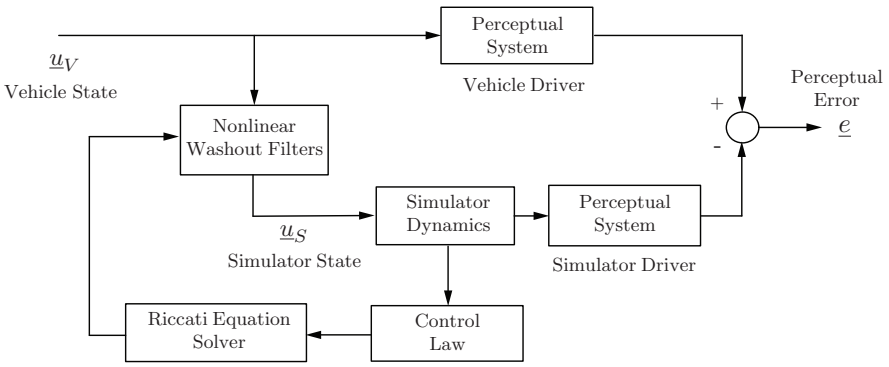


Figure 4.9: Control problem for developing the OpTNon algorithm based on Telban et al. (2005)

The reduced-order transfer functions of semicircular and otolith organs are:

$$H_{Scc}(s) = G_S \frac{\tau_a \tau_1 s^2}{(\tau_1 s + 1)(\tau_a s + 1)}; \quad H_{Oto}(s) = G_O \frac{s + A_0}{s + B_0}. \tag{4.54}$$

and the transfer function of the optokinetic influence for rotational channels is

$$H_{OK} = \frac{\dot{\varphi}_V - \hat{\dot{\varphi}}_S}{\hat{\dot{\varphi}}_{S,OK}} = \frac{1}{\tau_{OK} s + 1}, \tag{4.55}$$

where τ_{OK} is the optokinetic time constant (Van der Steen (1998)). Note that the transfer function is also used for translational channels. Therefore, for the simulator driver, the perception response of the perceptual system including the vestibular and visual system is $\underline{\hat{y}}_{S,PE} = [\hat{\dot{\varphi}}_{S,OK} + \hat{\dot{\varphi}}_S,\ \hat{v}_{S,OK} + \hat{v}_{Sy}]^{\mathrm{T}}$.

The new matrices of the otolith model integrating the visual states are

$$\begin{aligned}
\mathbf{A}_{Oto} &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -a & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -a & 0 \\ -T_2^* & 0 & 0 & -T_2^* & 0 & -T_2^* \end{bmatrix}; \mathbf{B}_{Oto} = \begin{bmatrix} c & 0 \\ d - ac & 0 \\ e & 0 \\ 0 & f \\ 0 & h - af \\ -T_2^* G_O r_{Sz} & 0 \end{bmatrix}; \\
\mathbf{C}_{Oto} &= \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}; \qquad \mathbf{D}_{Oto} = \begin{bmatrix} G_O r_{Sz} & 0 \end{bmatrix},
\end{aligned} \tag{4.56}$$

and the matrices of the semicircular system integrating the visual states are

$$\begin{aligned}
\mathbf{A}_{Scc} &= \begin{bmatrix} -T_2 & 1 & 0 \\ -T_1 & 0 & 1 \\ -T_2^* & 0 & -T_2^* \end{bmatrix}; \mathbf{B}_{scc} = \begin{bmatrix} -G_S T_1 & 0 \\ -G_S T_0 & 0 \\ -G_S {T_2}^* & 0 \end{bmatrix}; \\
\mathbf{C}_{Scc} &= \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}; \qquad \mathbf{D}_{Scc} = \begin{bmatrix} G_S & 0 \end{bmatrix},
\end{aligned} \tag{4.57}$$

where $T_2^* = 1/\tau_{OK}$. The desired system equation is given in Eq. 4.21. Telban et al. (2005) defined the state variables as $\underline{x} = \left[\underline{x}_e^{\mathrm{T}},\ \underline{x}_c^{\mathrm{T}},\ \underline{x}_n^{\mathrm{T}}\right]^{\mathrm{T}}$. Note that the state of errors $\underline{x}_e = \underline{x}_S - \underline{x}_V$, with $\underline{x}_S = \left[(\underline{x}_S^{Scc})^{\mathrm{T}},\ (\underline{x}_S^{Oto})^{\mathrm{T}}\right]^{\mathrm{T}}$ and similarly for $\underline{x}_V$, corresponds to the respective states of the model composing of the vestibular and the visual system whose system matrices are shown in Eq. 4.56 and 4.57 for the simulator and the vehicle, respectively; the simulator states are $\underline{x}_c = \left[\int S_y dt,\ \int v_{Sy} dt,\ \int a_{Sy} dt,\ \varphi_S\right]^{\mathrm{T}}$; $v_{Sy} = \int a_{Sy} dt$; $S_y = \int v_{Sy} dt$; and $\underline{x}_n$ are the filtered white noise states (see Eq. 4.20).

The cost function J_{Op} in Eq. 4.22 is augmented with the additional term $e^{2\eta t}$ proposed by Anderson and Moore (1971):

$$J_{Non} = E\left\{ \int_{t_0}^{t_f} e^{2\eta t} \left(\underline{e}^{\mathrm{T}} \mathbf{Q} \underline{e} + \underline{u}_S{}^{\mathrm{T}} \mathbf{R} \underline{u}_S + \underline{x}_c{}^{\mathrm{T}} \mathbf{R}_c \underline{x}_c \right) dt \right\} \stackrel{!}{=} min\,. \tag{4.58}$$

where, the perception error vector is

$$\underline{e} = \underline{\hat{y}}_{V,PE} - \underline{\hat{y}}_{S,PE} = [\Delta y_{PE,1},\, \Delta y_{PE,2}]^{\mathrm{T}} \tag{4.59}$$

with

$$\Delta y_{PE,1} = (\hat{\dot{\varphi}}_{V,OK} - \hat{\dot{\varphi}}_{S,OK}) + (\hat{\dot{\varphi}}_V - \hat{\dot{\varphi}}_S) = \Delta\hat{\dot{\varphi}}_{OK} + \Delta\hat{\dot{\varphi}}_S \tag{4.60}$$

$$\Delta y_{PE,1} = (\hat{v}_{V,OK} - \hat{v}_{S,OK}) + (\hat{\dot{v}}_{Vy} - \hat{\dot{v}}_{Sy}) = \Delta\hat{v}_{OK} + \Delta\hat{\dot{v}}_{Sy}\,; \tag{4.61}$$

and the input simulator vector $\underline{u}_S$ and the simulator state vector $\underline{x}_c$ are defined as in the OpT and OpTYM algorithms, respectively.

The state matrices in Eq. 4.50 with the additional motion platform and filtered white noise states are used once more. The cost function J_{Non} is minimized when

$$\underline{u}_S = \mathbf{F}(\eta)\underline{x} \quad , \tag{4.62}$$

where

$$\mathbf{F}(\eta) = \mathbf{R_2^{-1}} \left(\mathbf{B}^{\mathrm{T}} \mathbf{P}(\eta) + \mathbf{R_{12}} \right) \tag{4.63}$$

and its block sub-matrices are

$$\begin{aligned} \mathbf{F_1}(\eta) &= \mathbf{R_2^{-1}} \left[\widetilde{\mathbf{B}} \mathbf{P_{11}} + \mathbf{B_c^{\mathrm{T}}} \mathbf{P_{21}} + \widetilde{\mathbf{D}} \mathbf{Q} \widetilde{\mathbf{C}} \right] ; \\ \mathbf{F_2}(\eta) &= \mathbf{R_2^{-1}} \left[\widetilde{\mathbf{B}} \mathbf{P_{12}} + \mathbf{B_c^{\mathrm{T}}} \mathbf{P_{22}} \right] ; \\ \mathbf{F_3}(\eta) &= \mathbf{R_2^{-1}} \left[\widetilde{\mathbf{B}} \mathbf{P_{23}} + \mathbf{B_c^{\mathrm{T}}} \mathbf{P_{23}} - \widetilde{\mathbf{D}} \mathbf{Q} \widetilde{\mathbf{C}} \right] . \end{aligned} \tag{4.64}$$

Here, $\mathbf{P}_{ij}$ with $i, j = \{1..3\}$ are the block components of $\mathbf{P}(\eta)$ which results as the solution of the algebraic Riccati equation

$$(\mathbf{A}' + \eta\mathbf{I})^{\mathrm{T}} \mathbf{P}(\eta) + \mathbf{P}(\eta) (\mathbf{A}' + \eta\mathbf{I}) - \mathbf{P}(\eta)\mathbf{B}\mathbf{R_2^{-1}}\mathbf{B}^{\mathrm{T}}\mathbf{P}(\eta) + \mathbf{R}_1' = \mathbf{0}\,, \tag{4.65}$$

with $\mathbf{A}' = \mathbf{A} - \mathbf{B}\mathbf{R_2^{-1}}\mathbf{R_{12}^{\mathrm{T}}}$ and $\mathbf{R}_1' = \mathbf{R_1} - \mathbf{R_{12}}\mathbf{R_2^{-1}}\mathbf{R_{12}^{\mathrm{T}}}$. The partitions $\mathbf{P}_{ij}$ correspond to the partitions of the system matrix $\mathbf{A}$ (see Eq. 4.50). By symmetry $\mathbf{P}_{12} = \mathbf{P}_{12}^{\mathrm{T}}$. A

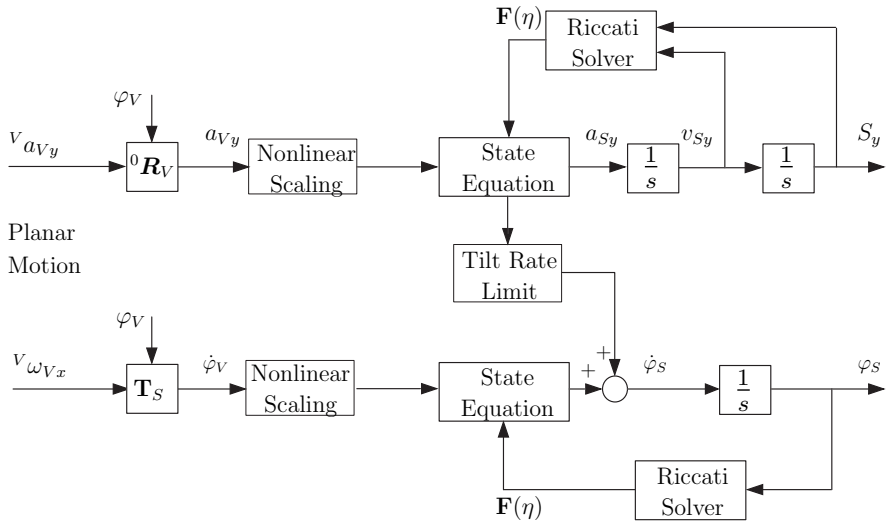


Figure 4.10: OpTNon algorithm implementation for roll/sway problem (Telban et al. (2005))

nonlinear control law is chosen to make η dependent upon the simulator states:

$$\eta = \underline{x}_c^{\mathrm{T}} \mathbf{Q}_2 \underline{x}_c \,, \tag{4.66}$$

where $\mathbf{Q}_2$ is a weighting matrix that is at least positive semi-definite. As the system states increase in magnitude, i.e. with large commanded platform displacements and velocities, then η increases, resulting in faster control action to quickly wash out the platform to its neutral state. For small commands there will be limited control action, resulting in motion cues being sustained for longer durations. The feedback matrix $\mathbf{F}(\eta)$ is then determined by solving the Riccati Eq. 4.33 in real time as a function of η. Finally, the state variables are obtained as:

$$\begin{bmatrix} \dot{\underline{x}}_e \\ \dot{\underline{x}}_c \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{A}} - \widetilde{\mathbf{B}}\mathbf{F_1}(\eta) & \widetilde{\mathbf{B}}\mathbf{F_2}(\eta) \\ -\mathbf{B_c}\mathbf{F_1}(\eta) & \mathbf{A_c} - \mathbf{B_c}\mathbf{F_2}(\eta) \end{bmatrix} \begin{bmatrix} \underline{x}_e \\ \underline{x}_c \end{bmatrix} + \begin{bmatrix} -\widetilde{\mathbf{B}}(\mathbf{I} + \mathbf{F_3}(\eta)) \\ -\mathbf{B_c}\mathbf{F_3}(\eta) \end{bmatrix} \underline{u}_V \quad . \tag{4.67}$$

The block diagram for the nonlinear optimal algorithm (see Fig. 4.10) shows the real time computation of simulated signal $\underline{u}_S$ in the "State Equation" block which is integrated to generate the simulated motions a_{Sy} and φ_S. At each time step, the feedback matrix $\mathbf{F}(\eta)$ is determined by the "Riccati Solver" block.

4.5 MCA using optimal tracking – ZyRo

This approach, first developed by Zywiol and Romano (2003), solves the MCA problem by using optimal control theory for reference tracking. The linear model of the tilt coordination circuit was developed instead of using the typical high-pass filters. In

terms of roll/sway problem, from Eq. 3.9, the simulated acceleration in the inertial frame $\mathcal{K}_0$ is linearized as

$$a_{Sy} \approx {}^{S}a_{Sy} = {}^{S}f_{Sy} - \mathrm{g}\varphi_S \,. \tag{4.68}$$

The control diagram for the method (Fig. 4.11) computes the simulated displacement and specific force as the output signals.

Let $\underline{u} = [u_1, u_2]^{\mathrm{T}}$ denote a deterministic control input vector where u_1, u_2 are filtered through a low-pass filter (PT1) to generate the specific force at the drivers's head and the tilt rate $\dot{\varphi}_S$, respectively. The control system is described in the state space equation as

$$\begin{cases} \dot{\underline{x}} = \mathbf{A}\underline{x} + \mathbf{B}\underline{u} \\ \underline{y} = \mathbf{C}\underline{x} \end{cases} , \tag{4.69}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -9.81 \\ 0 & 0 & -\gamma & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -c \end{bmatrix} ; \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \gamma & 0 \\ 0 & 0 \\ 0 & c \end{bmatrix} ; \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{4.70}$$

and the state variables $\underline{x} = \left[{}^{S}f_{Sy}, \dot{\varphi}_S, \varphi_S, \dot{S}_y, S_y\right]^{\mathrm{T}}$.

The desired tracking behavior is accomplished with the following cost function

$$J_{ZyRo} = \int_{t_0}^{t_f} \left[\left(\underline{y} - \underline{y}_{ref}\right)^{\mathrm{T}} \mathbf{Q} \left(\underline{y} - \underline{y}_{ref}\right) + \underline{u}^{\mathrm{T}} \mathbf{R} \underline{u} \right] dt \,, \tag{4.71}$$

that aims to find the appropriate time history for the control input vector $\underline{u} = [u_1, u_2]^{\mathrm{T}}$ to make the output signal $\underline{y} = [{}^{S}f_{Sy}, S_y]^{\mathrm{T}}$ track the reference output vector, which is

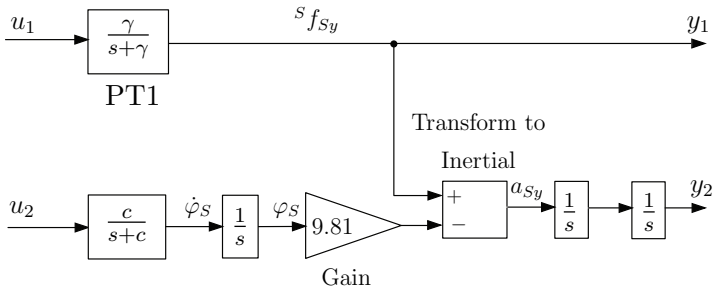


Figure 4.11: Control diagrams of the ZyRo algorithm based on Zywiol and Romano (2003)

$\underline{y}_{ref} = [^V f_{Vy}, 0]^{\mathrm{T}}$, as close as possible. The second component of the reference vector is to implement the washout effect that "pulls" the simulator to the initial position. The optimal control input signal is then estimated as

$$\underline{u} = -\mathbf{R}^{-1}\left[\mathbf{B}^{\mathrm{T}}\mathbf{U}\underline{x} + \mathbf{B}^{\mathrm{T}}\bar{\mathbf{S}}\right], \tag{4.72}$$

where the positive semi-definite matrix $\mathbf{U}$ is the solution of the differential Riccati equation

$$\dot{\mathbf{U}} = -\mathbf{U}\mathbf{A} - \mathbf{A}^{\mathrm{T}}\mathbf{U} - \mathbf{C}^{\mathrm{T}}\mathbf{Q}\mathbf{C} + \mathbf{U}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{U}, \tag{4.73}$$

while $\bar{\mathbf{S}}$ in Eq. 4.72 is the solution of the co-state equation, solved in backward time given by

$$\dot{\bar{\mathbf{S}}} = -[\mathbf{A}^{\mathrm{T}} - \mathbf{U}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^{\mathrm{T}}]\bar{\mathbf{S}} + \mathbf{C}^{\mathrm{T}}\mathbf{Q}\underline{y}_{ref}. \tag{4.74}$$

The numerical values of the weight matrices $\mathbf{Q} = \mathrm{diag}\{q_1, q_2\}$, $\mathbf{R} = \mathrm{diag}\{r_1, r_2\}$, as well as of the first break frequencies c and γ of the low-pass filter (PT1) from Fig. 4.11 are tuned according to the simulation task. While q_2 regulates how the platform is "washed out" back to its home configuration, q_1 has the effect on how the simulated specific forces track the reference ones ; r_2 is used to passively restrict the angular velocity under the threshold value; and the two break-frequencies c and γ smooth the simulated specific force and angular velocity.

4.6 MCA using model predictive control theory

Model predictive control (MPC) is an advanced control technique. It can deal with the control problems that have both large multivariable and complicated constraints. The general design objective of MPC is to optimize the future behavior of the plant output over a future fixed interval (prediction horizon and control horizon) while always respecting the previously defined constraints. Therefore, MPCs have been applied to create MCAs which can consider simulator limitations as well as human motion perception thresholds, improving the motion cueing quality. Dagdelen et al. (2004) first introduced the algorithm applied to the ULTIMATE simulator at the Renault Technical Center for Simulation. Later, Augusto and Loureiro (2009) described the details of the implicit MPC, that was also designed by Baseggio et al. (2011). The implicit MPC is based on an online optimization technique with Quadratic Programming (QP) (Wang (2009)). Recently, another approach (Fang and Kemeny (2012)) has been introduced that is based on an offline optimization technique used to compute an explicit MPC controller that can be applied to the online implementation.

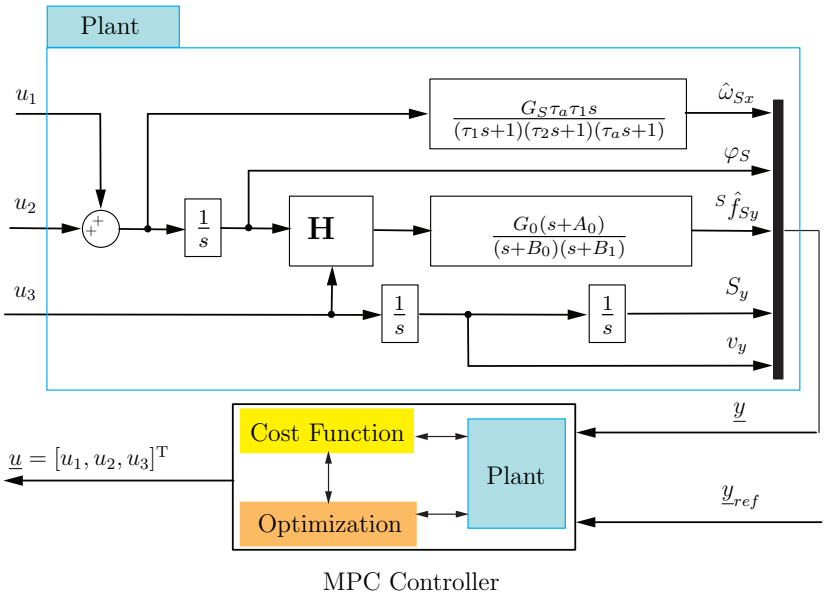


Figure 4.12: Implementation diagram of the MPC algorithm

4.6.1 Implicit MPC motion cueing algorithms

Augusto and Loureiro (2009) and Baseggio et al. (2011) designed an implicit MPC algorithm that includes a linear MPC controller to solve the MCA problems, such as removing false cues, operating within the limited workspace, and reproducing beneficial motion cues. Augusto and Loureiro (2009) regarded four operation modes pitch/surge, roll/sway, heave and yaw. In this thesis, only the roll/sway mode is regarded as illustrated in Fig. 4.12. It is clear that the MPC controller is composed of a plant, a cost function, and an optimization method.

The plant includes tilt coordination and the integration of the vestibular system (introduced by Reid and Nahon (1985)) to generate the sensed quantities and simulated motion. The control vector (input of the Plant), consists of three inputs $\underline{u} = [u_1,\, u_2,\, u_3]^{\mathrm{T}} = [\dot{\varphi}_a,\, \dot{\varphi}_r,\, a_{Sy}]^{\mathrm{T}}$ denoting the tilt roll angular rate, the actual vehicle roll angular rate, and the simulated acceleration, respectively. The optimization is designed so the output vector $\underline{y} = [\hat{\omega}_{Sx}, \varphi_S, {}^S\hat{f}_{Sy}, S_y, v_y]^{\mathrm{T}}$ tracks the reference signals $\underline{y}_{ref} = [\hat{\omega}_{Vx},\, 0,\, {}^V\hat{f}_{Vy},\, 0,\, 0]^{\mathrm{T}}$ as close as possible. The transformation matrix $\mathbf{H} = [\mathrm{g},\, 1]$ (Fig. 4.12) is for computing the simulated specific force

$$ {}^S f_{Sy} = \mathbf{H} \cdot [\varphi_S\,, a_{Sy}]^{\mathrm{T}} = [\mathrm{g},\, 1]\,[\varphi_S\,, a_{Sy}]^{\mathrm{T}} = a_{Sy} + \mathrm{g}\varphi_S \quad . \tag{4.75} $$

The MPC controller uses quadratic programming (QP) to find a suitable control signal

$\underline{u}$ that minimizes the cost function

$$J_{MPC}(\Delta\underline{u}) = \left[\underline{y} - \underline{y}_{ref}\right]^{\mathrm{T}} \mathbf{Q} \left[\underline{y} - \underline{y}_{ref}\right] + \underline{u}^{\mathrm{T}}\mathbf{S}\underline{u} + \Delta\underline{u}^{\mathrm{T}}\mathbf{R}\Delta\underline{u} \tag{4.76}$$

in order to force the output signals to track the reference signals given (Wang (2009)), where $\underline{y}_{ref}$, $\Delta\underline{u}$ are the reference output values, and input increment, respectively; while $\mathbf{Q} = \mathrm{diag}\{q_1, \ldots, q_n\}$, $\mathbf{R} = \mathrm{diag}\{r_1, \ldots, r_n\}$ and $\mathbf{S} = \mathrm{diag}\{s_1, \ldots, s_n\}$ are weighting matrices.

The constraint conditions for output signals and incremental inputs are

$$|S_y| \leq S_{y,max} \quad ; |v_y| \leq v_{y,max} \quad ; |\varphi_S| \leq \varphi_{max} \quad ; |\omega_S| \leq \delta_S \quad ; \Delta u \leq \Delta u_{max}\,. \tag{4.77}$$

The response of the algorithm is controlled by the weighting parameters. However, selecting larger values in the weighting matrix for specific force fidelity could lead to a jerky motion. To avoid these problems, this dissertation introduces an alternative algorithm, termed MPC*, for roll/sway model and inserting, as in ZyRo algorithm, two first-order low-pass filters in both the translational and rotational input channel (see Fig. 4.13) to smooth the input signals. That means that the deterministic control input vector $\underline{u} = \{u_1,\, u_2\}$ is filtered through a first-order lag block with time constants $\{c_1,\, \gamma_1\}$ to generate the simulated angular velocity and specific force $\{\omega_{Sx},\, {}^{S}f_{Sy}\}$, respectively. Furthermore, a reduced model of the vestibular system (Reid and Nahon

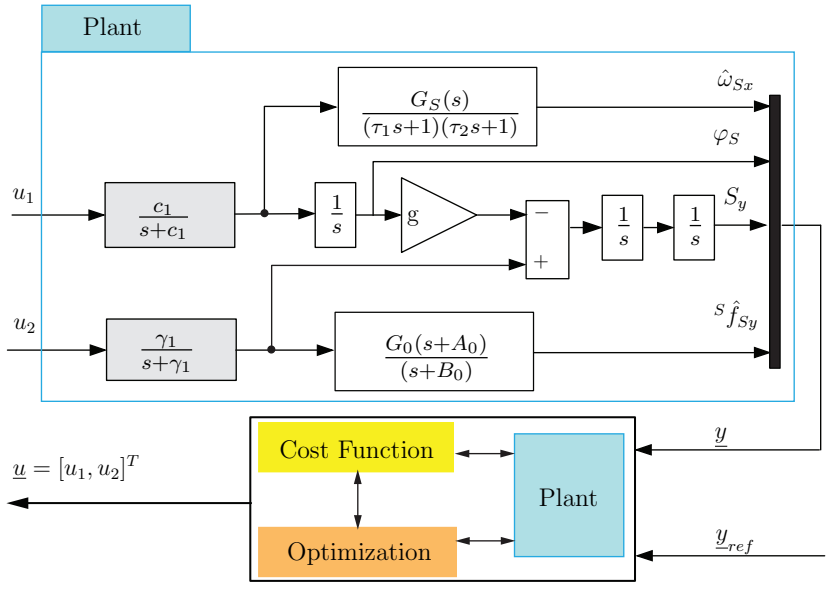


Figure 4.13: Implementation diagram of the MPC* algorithm

(1985)) is without time parameters $B_1 = 1.5(1/s)$ for the otolith model and $\tau_a = 30(s)$ for the semicircular model according to Eq. 4.24 and 4.25 was used to reduce computational effort. As a result, the state-space model of the plant (Fig. 4.13) is

$$\begin{cases} \dot{\underline{x}} = \mathbf{A}\underline{x} + \mathbf{B}\underline{u} \\ \underline{y} = \mathbf{C}\underline{x} \end{cases} . \tag{4.78}$$

Here, the matrices of the model are

$$\mathbf{A} = \begin{bmatrix} -b_0 & G_O(A_0 - B_0) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1/\gamma & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1/c & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \mathrm{g} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -G_S F & 0 & 0 & E & F \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 1/\gamma & 0 \\ 0 & 1/c \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; \tag{4.79}$$

$$\mathbf{C} = \begin{bmatrix} 1 & G_O & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.80}$$

where $E = -(1/\tau_1 + 1/\tau_2)$ and $F = -1/(\tau_1\tau_2)$. In addition, the state and output variables are defined as:

State variables $\underline{x} = [x_1 \cdots x_8]^{\mathrm{T}}$:
x_1 is a state variable of otolith model; $x_2 = {}^S f_{Psy}$ is the simulated specific force ; x_3 is the angular rate $\dot{\varphi}$, $x_4 = \varphi_S$ is the tilt angle; $x_5 = v_y$ is the velocity of the cabin; $x_6 = S_y$ is the position of the simulator; $x_7 = \hat{\omega}_{Sx}$ is the sensed angular velocity; and x_8 is a state variable of semicircular model.
Output variables $\underline{y} = [y_1\,, y_2\,, y_3\,, y_4]^{\,T}$:
$y_1 = {}^S\hat{f}_{Psy}$ is the sensed specific force at the driver's head; $y_2 = S_y$ is the position of the simulator; $y_3 = \hat{\omega}_{Sx}$ is the sensed angular tilt rate; and $y_4 = \varphi_S$ is the simulated tilt angle.

The objective of the MPC* algorithm is to make the output signal $\underline{y}$ track the reference signal $\underline{y}_{ref} = \left[{}^V\hat{f}_{Vy}\,, 0\,, 0\,, 0\right]^{\,\mathrm{T}}$ as close as possible. In addition, the cost function and constraints conditions are similar to those in Eq. 4.76 and 4.77, respectively.

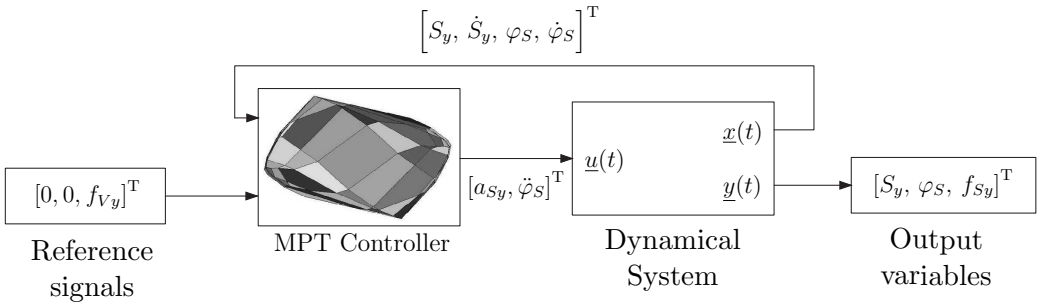


Figure 4.14: Implementation diagram of the exFKMPC algorithm

4.6.2 Explicit MPC motion cueing algorithm – exFKMPC

Based on the multi-parameter programming toolbox (MPT), developed by the Automatic Control Laboratory of ETH, Zürich, and the solution for the tracking problem introduced by Pekar and Havlena (2004), Fang and Kemeny (2012) proposed an MCA using explicit MPC for real-time driving simulators hereon referred to as exFKMPC.

Note that the plant or dynamic system block (see in Fig. 4.14) does not include the model of the vestibular system. Only the linear model of tilt coordination is considered. The plant has the input variables $\underline{u} = [a_{Sy}, \ddot{\varphi}_S]^{\mathrm{T}}$, the output variables $\underline{y} = \left[S_y,\, \varphi_S,\, {}^{S}f_{Sy}\right]^{\mathrm{T}}$, and the state variables $\underline{x} = \left[S_y,\, \dot{S}_y,\, \varphi_S,\, \dot{\varphi}_S\right]^{\mathrm{T}}$. The state space model of the plant in continuous time representation is

$$\begin{cases} \dot{\underline{x}} = \mathbf{A}\,\underline{x} + \mathbf{B}\,\underline{u} \\ \underline{y} = \mathbf{C}\,\underline{x} + \mathbf{D}\,\underline{u} \end{cases}, \tag{4.81}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}; \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \mathrm{g} & 0 \end{bmatrix}; \mathbf{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}. \tag{4.82}$$

To solve the tracking problem for the output signals, Fang and Kemeny (2012) used an approach based on work of Pekar and Havlena (2004) that transforms the optimization tracking problem into a standard regulation optimization in which a reference signal is inserted in state variables. Furthermore, the MPT controller is expressed in the discrete form given by the discrete state-space model

$$\begin{cases} \underline{x}_{k+1} = \mathbf{A}_d\,\underline{x}_k + \mathbf{B}_d\,\underline{u}_k \\ \underline{y}_{k+1} = \mathbf{C}_d\,\underline{x}_k + \mathbf{D}_d\,\underline{u}_k \end{cases}, \tag{4.83}$$

with

$$\underline{x}_k = \begin{bmatrix} S_{y,k} \\ v_{Sy,k} \\ \varphi_{S,k} \\ \omega_{Sx,k} \\ y_{ref,k} \end{bmatrix} \; ; \; \mathbf{A}_d = \begin{bmatrix} 1 & T_s & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & T_s & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \; \mathbf{B}_d = \begin{bmatrix} 0.5{T_s}^2 & 0 \\ T_s & 0 \\ 0 & 0.5{T_s}^2 \\ 0 & T_s \\ 0 & 0 \end{bmatrix} \; ; \tag{4.84}$$

$$\mathbf{C}_d = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \mathrm{g} & 0 & -1 \end{bmatrix} \; ; \; \mathbf{D}_d = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \; ; \underline{u}_k = \begin{bmatrix} a_{Sy,k} \\ \dot{\omega}_{Sx,k} \end{bmatrix} \quad . \tag{4.85}$$

where physical quantities $S_{y,k}$, $v_{Sy,k}$, $a_{Sy,k}$, $\varphi_{S,k}$, $\omega_{Sx,k}$, and $\dot{\omega}_{Sx,k}$ are the discrete state variables of simulator position S_y, velocity v_{Sy}, tilted angle φ_S, angular velocity $\omega_{Sx} \approx \dot{\varphi}_S$, respectively; $a_{Sy,k}$ and $\dot{\omega}_{Sx,k}$ are the discrete input variables of acceleration a_{Sy} and angular acceleration $\dot{\omega}_{Sx}$, respectively; the reference signal $y_{ref,k}$ is selected to track the target specific force ${}^V f_{Vy}$; and T_s is the sample time of the discrete dynamic model.

In order to compute feasible motions subject to the simulator's given physical limits, Fang and Kemeny (2012) proposed a braking control law which allows the simulator to remain in its translational and rotational workspace. The braking law is

$$\begin{aligned} S_{y,min} < \left| S_{y,k} + c_v \, t_1 \, \dot{S}_{y,k} + 0.5 \, c_u \, \ddot{S}_{y,k} \, t_1^2 \right| < S_{y,max} \\ \varphi_{min} < \left| \varphi_{S,k} + c_v \, t_2 \, \dot{\varphi}_{S,k} + 0.5 \, c_u \, \ddot{\varphi}_{S,k} \, t_2^2 \right| < \varphi_{max} \end{aligned} \; . \tag{4.86}$$

Furthermore, the constraint conditions for the plant are

$$\begin{aligned} |x_{1,k}| \leq & S_{y,max}; \quad |x_{2,k}| \leq v_{Sy,max}; \quad |u_{1,k}| \leq a_{Sy,max} \\ |x_{3,k}| \leq & \; \varphi_{max}; \quad |x_{4,k}| \leq \; \omega_{max}; \quad |u_{2,k}| \leq \dot{\omega}_{max} \end{aligned} \; , \tag{4.87}$$

The limit values $S_{y,max}$, $v_{y,max}$, a_{max}, φ_{max}, ω_{max}, and $\dot{\omega}_{max}$, are the absolute maximum values of position, velocity, lateral acceleration, tilted angle, angular velocity and angular acceleration of the simulator, respectively; c_v, c_u, t_1 and t_2 describe the braking control law and determine the behaviour of the simulator when approaching its workspace limits.

The MPT controller optimizes the input vector to minimize the cost function

$$J_N(\underline{x}_k) = \min_{u_0, u_1, \ldots, u_{N-1}} {\underline{x}_N}^{\mathrm{T}} \mathbf{Q_N} \underline{x}_N + \sum_{i=0}^{N-1} \underline{u}_i^{\mathrm{T}} \mathbf{R} \underline{u}_i + \sum_{i=0}^{N-1} \underline{x}_i^{\mathrm{T}} \mathbf{Q_x} \underline{x}_i + \sum_{i=0}^{N-1} {\underline{y}_i}^{\mathrm{T}} \mathbf{Q_y} \underline{y}_i \, , \tag{4.88}$$

while respecting the brake law and the constraint conditions in Eq. 4.86 and 4.87, respectively. Here, N is the prediction horizon, $\underline{x}_N$ is the vector of final values for the state variables, $\underline{u}_i$ is the i^{th} predicted input vector, $\underline{x}_i$ is the corresponding predicted state

vector, $\underline{y}_i$ is the corresponding predicted output vector, and $\mathbf{Q_N} = \text{diag}\{q_{N1}, \ldots, q_{Nn}\}$, $\mathbf{R} = \text{diag}\{r_1, \ldots, r_n\}$, $\mathbf{Q_x} = \text{diag}\{q_{x1}, \ldots, q_{xn}\}$ and $\mathbf{Q_y} = \text{diag}\{q_{y1}, \ldots, q_{yn}\}$ are positive semi-definite the weighting matrices. Additionally, Fang and Kemeny (2012) stated the necessary and sufficient condition for selecting possible parameters in order to avoid overshoots (see Eq. 4.86) as $(c_v^2 \,/\, 2\, c_u) > 1$.

4.7 Adaptation of MCAs to the LMR driving simulator

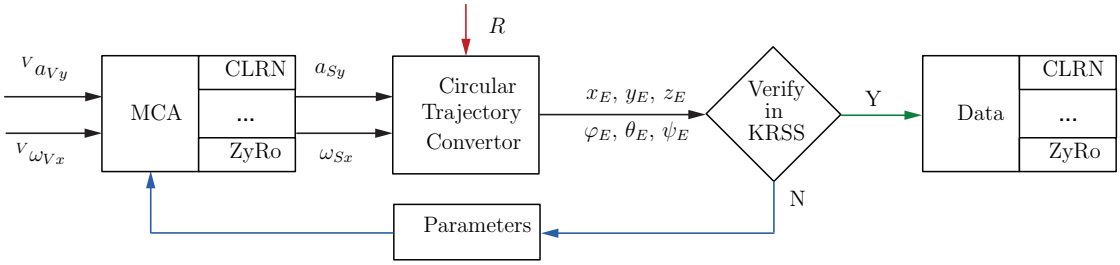


Figure 4.15: Process adapting MCAs to the LMR driving simulator

In order to adapt the MCAs to the LMR driving simulator, the motion data were created using the procedure shown in Fig. 4.15. For the roll/sway problem, the target values $^V a_{Vy}$ and $^V \omega_{Vx}$ enter a selected MCA to generate the simulated signals a_{Sy} and ω_{Sx}. These are then converted into the position and orientation of the frame $\mathcal{K}_E$ through a "Circular Trajectory Converter" block. The trajectory of the frame $\mathcal{K}_E$ is verified in the software KRSS for consistency with the robot limits. If violations exist the parameters of the MCAs must be modified for a new run. The verified data are then stored for later use in the LMR driving simulator.

4.8 Comparison of the features of the reviewed MCAs

Nahon and Reid (1990) implemented an unbiased comparison of three MCAs: classical, coordinate adaptive, and optimal washout filters. The authors developed three practical criteria (Cr_1, Cr_2, Cr_3) to evaluate the efficacy of the algorithms under a designer's perspective (see Tab. 4.1). In the current work, the features of the investigated MCAs were assessed according to the above criteria. Moreover, an additional criterion was introduced to emphasize the advantage of "exploiting workspace" and generating simulated motion within the robots physical limits.

Based on the processes of constructing MCAs and tuning parameters, the comparison of the features of the MCAs is summarized in Tab. 4.2. It can be seen that all MCAs

Criterion
Cr_1: Capability to achieve good pilot rating and the potential to achieve excellent ones **1a** : Exploiting the nonlinearities of human motion sensation. **1b** : Limiting severely motion cueing only when the motion base nears its limits. **1c** : The ability to be adjustable for different pilots. **1d** : The ability to be adjustable for different flight segments.
Cr_2: Adjustable features (how easily the MCA can be adjusted). **2a** : Number of parameters. **2b** : The change of parameters should be easy to determine a priori. **2c** : Parameters should be applied to physically meaningful quantities. **2d** : Adjustment should be possible without requirements on expertise of the adjuster's part about how the algorithm works.
Cr_3: Time of computation (execution speed).

Table 4.1: Comparison criteria proposed by Nahon and Reid (1990) for flight simulation

can satisfy the criteria **1c** and **1d**, that means that any MCA could be applied for different drivers and simulation tasks. However, not all MCAs can take the nonlinear characteristics of the motion perception into account (**1a**). Furthermore, for the criterion **1b**, most MCAs cannot manipulate the simulated motion in the place near the hard limits of the simulator, except two novel model predictive control methods (MPC* and exMPCFK). While MPC* algorithm uses the constraint conditions to force stopping the simulator before its limit, the exMPCFK algorithm smooths the motion of the simulator near its hard limits.

Regarding the criterion Cr_2, transparency of parameters indicates how easily the designer can predict changes in the simulator motion (states or variables) that would result from a change in one of the free parameters Nahon and Reid (1990). The ZyRo algorithm has the smallest numbers of the parameters. Additionally, its parameters have transparency (physical meaning) with respect to the simulated quantities, so that adjustments of parameters feature a more direct causality with respect to the corresponding simulator state/variables. As a result, the ZyRo algorithm is tuned more easily than the other algorithms. The OpS algorithm has the same number of parameters as the ZyRo; however, the tuning process for OpS algorithm is more complicated than for the ZyRo algorithm because of the more indirect parameters on the change of the simulated signals. The adaptive algorithms are the most difficult to tune because they can be unstable if inappropriate parameters are selected.

Almost all MCAs' responses have relatively poor deduction from previous results, especially, the linear optimal washout filter family. Regarding the **2b** and **2c** criteria, the five optimal washout filters seem to have transparent parameters that control the specific quantities (i.e. such as simulator's position and acceleration). However, the

Algorithms		Criteria for comparing MCAs' features									EW	Implementation	
		Cr_1				Cr_2				Cr_3			
		1a	1b	1c	1d	2a	2b	2c	2d	$t(s)$		Online	Offline
1	CLRN	–	–	x	x	15	x	–	–	4		x	x
2	CLG	–	–	x	x	15	x	–	–	4		x	x
3	ADSK	–	–	x	x	21	–	–	–	1297		x	x
4	ADRN	–	–	x	x	17	–	–	–	618		x	x
5	OpTYM	–	–	x	x	8	–	x	–	t_{Op}		x	x
6	OpT	–	–	x	x	8	–	x	–	t_{Op}		x	x
7	OpTNon	–	–	x	x	9	–	x	–	129		x	x
8	OpS	–	–	x	x	6	–	x	–	t_{Op}		x	x
9	OpRN	–	–	x	x	7	–	x	–	t_{Op}		x	x
10	MPC*	–	x	x	x	13	x	x	–	1933	+	x	x
11	exMPCFK	–	x	x	x	15	x	x	–	t_{ex}		x	x
12	ZyRo	–	–	x	x	6	x	x	–	79	+		x

Note:
- EW: Exploiting driving simulator workspace
- '–': No ; 'x': Yes
- '+': Better exploiting workspace in operation
- t_{Op} is time for calculating optimal washout filters (15s) and then implementing (3s)
- t_{ex} is time for calculating explicit MPC controller (0.5h - 12h) and then implementing (5s)
- Online: The input signals are based on the maneuver of the drivers interacting with a driving simulator
- Offline: The input signals are fixed for specific simulation tasks

Table 4.2: Asssessment of the MCAs features

effect of the parameters on these quantities is not easy to deduct from their previous responses. On the other hand, although the parameters of the two classical algorithms (CLRN and CLG) have no physical interpretation, their responses are predictable. In contrast to these, three novel algorithms (ZyRo, MPC*, and exMPCFK) can satisfy both criteria **2b** and **2c**. However, all MCAs require a designer experienced in selecting suitable parameters for the specific simulation task, all MCAs cannot, therefore, fulfill criterion **2d**.

In terms of "exploiting workspace", Fig. 4.16 shows the rotational angle α used to simulate the lateral acceleration and the tilt angles φ_S. It can be seen that the two algorithms MPC*, ZyRo use smaller rotational angles of both α and φ_S than the remaining algorithms. The reason is that these two algorithms utilize optimization methods in a specific time period to find the suitable combination of both lateral accelerations and tilt angles. On the other hand, the remaining algorithms consider the high- and low-frequency parts of the target inputs individually to give the corresponding response, and hence do not "prepare" the workspace for the motion in the future time steps.

The time-consumption (Cr_3 in Tab. 4.2) is from the simulation case with only lateral acceleration input. The simulation was implemented in Matlab® Simulink in a PC

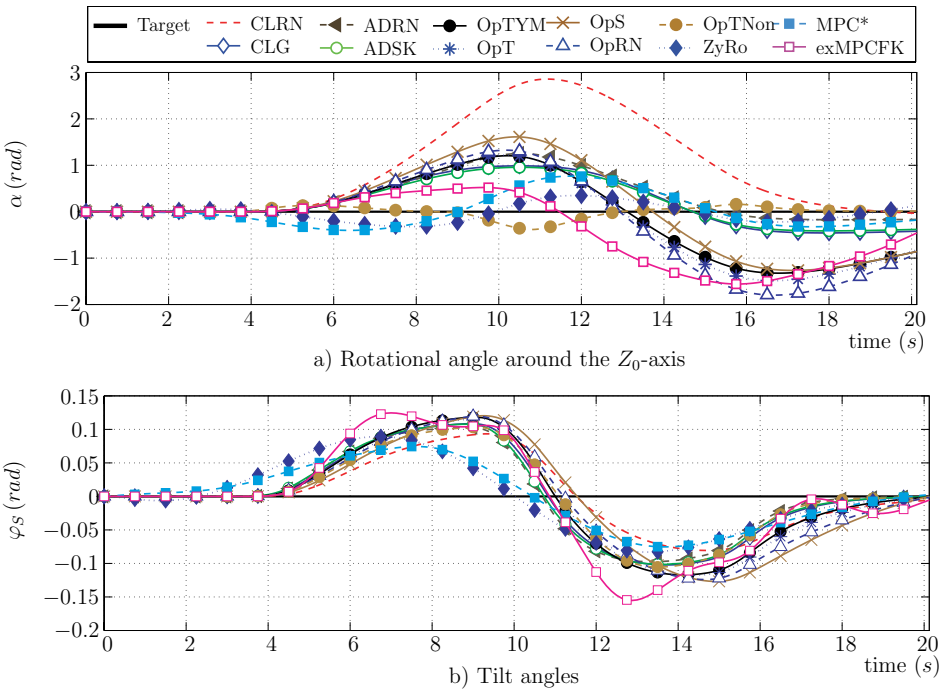


Figure 4.16: Comparison of the simulator's angles α and tilt angle φ_S

desktop with Quadcore 2.66GHz. It is easy to see that the classical algorithms and the optimal washout filters have better features than the other algorithms. Note that the optimal washout filters require two steps for generating the simulated signals. Firstly, the optimal washout filters are designed offline with optimal theory, then they are used in the implementing scheme to filter the input signals. Similarly, the exMPCFK algorithm has also two implementing steps. The first step for computing explicit MPC controller takes an exceptionally long time, from 0.5h to 12h. It is the main disadvantage point of the exMPCFK algorithm, that requires more time for testing a new set of parameters.

In summary, the ZyRo and the MPC family algorithms are the most transparent. However, regarding time-consumption, these methods are more suitable for offline implementation than on an online driving simulator. On the other hand, the classical algorithm family can operate both in offline and online mode. The optimal washout filter family has a complex design with integrated perception systems, thus the tuning process for these MCAs consumes more time than for the other ones. Out of the five optimal washout filter family, the OpTNon algorithm is the only one that can compute a set of new washout filters in real-time. This characteristic is similar to the adaptive algorithm ADRN that computes only the parameters of the high-pass washout filters.

5 Assessing Motion Perception Fidelity and Parameter-Tuning for MCAs

The term fidelity has been defined as the degree to which a device accurately reproduces a specific effect (Gerathewohl (1969)). The simulator fidelity depends on the nature of the provided cues including equipment cues and environment cues. Motion cues correspond to environment cues together with the visual out-of-the-window cues (Key (1980)). The motion perception fidelity enhances not only the perceptive fidelity but also the drivers' behavior during a simulation session.

MCAs produce the simulated motion together with other environment cues in order to provide a high fidelity simulation. However, such effects are beyond the scope of this thesis and will not be regarded here. The tuning parameters of MCAs aim to archive the desired simulated motion for simulation tasks. This section first introduces common methods assessing motion perception fidelity and reviews tuning methods in the literature. Then, the objective criteria for tuning parameters of MCAs are proposed and used to tune parameters for the MCAs adapted to the LMR driving simulator, and a comparison of responses of the MCAs in regard to several objective criteria is introduced. Finally, an auto-tuning method using proposed criteria is developed.

5.1 Motion perception fidelity assessment

Motion perception fidelity has been commonly measured by either subjective or objective assessments. On the one hand, by using numerical criteria, objective assessments are applied to an offline evaluation. The numerical criteria are obtained from analysis of the recorded simulation data including responses of vehicles, driver's behavior, subjective scores, and statements given by the experienced drivers. For example, Sinacori criteria (Sinacori (1977)) were established from data of four test points of helicopter simulation tasks (Schroeder (1999)). The criteria divide the levels of motion perception fidelity according to the range of scale factor and phase-lag of high-pass filters. On the other hand, by using technical questionnaires, subjective assessments are applied to get a straightforward evaluation of participants attending to simulation tasks. The technical questionnaires are set up for an investigated purpose, such as realism of motion perception, a symptom of motion sickness, etc. For example, Cooper and Harper Jr (1969) defined the level of motion perception fidelity (Handling Quality Ratings (HQR), see Tab. 5.1) which has been used for the subjective assessments of flight

Rating	Fidelity	Description
3	HIGH	Motion sensations are close to those of visual flight
2	MEDIUM	Motion sensation differences are noticeable but not objectionable
1	LOW	Motion sensation differences are noticeable and objectionable including loss of performance and disorientation

Table 5.1: Handling Quality Ratings for flight simulation (Cooper and Harper Jr (1969))

simulation tasks. In the HQR, the high motion perception fidelity means the motion sensations are close to the perception of the real flight as presented by the visuals, while the low motion perception fidelity means that the differences are very noticeable and objectionable because of performance loss or disorientation.

In summary, objective assessments are faster than subjective ones and can be applied for different specific tasks. However, they cannot be used widely for any simulation tasks because the objective criteria are solely defined for a particular purpose. In contrast, the subjective assessment is straightforward and consistent to assess the fidelity of a driving simulator. Furthermore, questionnaires can be used to investigate different aspects and situations including the perception feelings during braking, accelerating, simulator sickness, etc.

5.2 Parameter-Tuning for MCAs

The goal of tuning parameters for MCAs is to reproduce the feeling of driving a real vehicle and to remove simulator sickness. That means that the appropriate simulated quantities (benefit cues) are generated and the undesired motion cues (false cues) are reduced. However, the tuning of a MCA is especially difficult due to various factors, such as nontransparent parameters of a MCA, limited workspace, and nonlinear characteristics of human motion perception. Valente Pais et al. (2009) mentioned that for the classical algorithm, higher gain (scale factor), and hence more required motion, often causes lagging of the tilt angle due to limited tilt rate. Lower gain causes smaller motion cues so that it is hard to distinguish the motions with classical filters from the motion with the rumble filters. Chapron and Colinot (2007) formulated the 4 difficulties of tuning processes for MCAs: 1) tilt derivatives (angular velocity) limit cause the deformation of the specific force profile; 2) the values of tilt-derivative limit is hard to select, as the larger they are, the lower is the specific force deformation, but the worse is the somatogravic illusion; 3) the limited workspace and the problem for its exploita-

tion; 4) the selection of appropriate translational and rotation motion combination to well-reproduce the specific force with minimal deformation. Additionally, White and Rodchenko (1999) mentioned that it is opaque how well the simulated motion cues need to match the real cues. Normally, a MCA is first roughly tuned with objective criteria (ObC) to generate the desirably simulated quantities. Then, the quantities are fine-tuned with subjective assessments with driver-in-the-loop (DIL) settings involving interactive exercises in which the driver rides the simulator and comments on the experienced motions, and a designer modifies the motion parameters accordingly (Nahon et al. (1992), Valente Pais et al. (2009)).

However, the DIL method is time-consuming, and the current ObC cannot be applied for general simulation tasks. Hence, several new tuning methods, termed here "subject-objective criteria" (SOC) were introduced in the literature which include numerical conditions established from the result of subjective experiments, and which are aimed at reducing the time consumed in the tuning process. For example, a maneuver-specific tuning was introduced by Brünger-Koch et al. (2006) and an objective tuning with the criteria for the response of washout filters was introduced by Sinacori (1977), Schroeder (1999), Zeyada and Hess (2000) and Zaichik et al. (2010).

In this thesis, four main categories of tuning methods are considered. The first method (section 5.2.1) involves the DIL method using statements of participants to adjust the parameters of a MCA, the results of which can be later used within a supervisory software introduced by Nahon et al. (1992) to automatically adapt according to a simulation task. The second method (section 5.2.2) analyzes numerically simulated quantities that can be perceived by the vestibular system; the parameters are tuned here to remove as much as possible the potential false cues that are able to degrade seriously the motion perception fidelity (Grant and Reid (1997a)). The third method (section 5.2.3) concerns the effect of several parameters of a MCA on its response; for example, from the subjective experiment with experienced pilots (Sinacori (1977) and Schroeder (1999)), the relationship between frequency response of high-pass washout filters and motion perception fidelity as well as the range of parameters of high-pass filters were even determined for corresponding levels of motion perception fidelity (White and Rodchenko (1999), Zaichik et al. (2009), (2010)). The fourth method (section 5.2.4) is based on operating-pilot and motion-perception models to assess the motion perception fidelity, so that the simulated quantities can be compared with reference quantities; hereby, the better the correlation between the two response is, the better the parameters are tuned as shown by Reid and Nahon (1985) and Pouliot et al. (1998) to verify the simulated quantities in flight simulators. Note that the last three methods can be implemented in silico to find suitable parameters.

5.2.1 Subjective method using driver in-the-loop (DIL) setups

The subjective tuning method is based on the drivers' ability to rate the operating vehicles performance going through specific simulation tasks. In the literature, the tuned parameters were obtained from the assessments of participants after they experienced the simulation tasks repeatedly (Reid and Nahon (1986), Telban et al. (2005), Fischer (2009)). The rating was implemented by using questionnaires or statements about realistic levels of motion perception or comfortable levels in operating a driving simulator. Note that subjective evaluation is considered as a straightforward and consistent method for assessing the fidelity of a driving simulator. Therefore, the DIL is the most reliable tuning method for finding suitable parameters with which a MCA can generate a high level of motion perception fidelity. However, due to the non-linearity of the relationship between motion and perception and due to history effects on the subjective assessment it is by no means guaranteed that this tuning process will converge, and thus there is an imminent risk of deadlock trapping in the cycle of motion parameter variation and subjective assessment.

Nonetheless, this is an important tuning method and thus it is regarded in this context. Hereby, the subjective tuning method can be classified into two groups: individual driver assessment and average assessment in maneuvers for drive tracks.

a) Individual driver assessment

Grant and Reid (1997a) developed an expert system to adjust parameters of classical washout filters to satisfy the pilot. The tuning was implemented for specific maneuvers and focused on the "limit problem" of the simulator equipment and "pilot selected problems". The paradigm of the expert system (Fig. 5.1) assumes a crucial role of the pilot in the identification of disturbing problems after flying with the flight simulator,

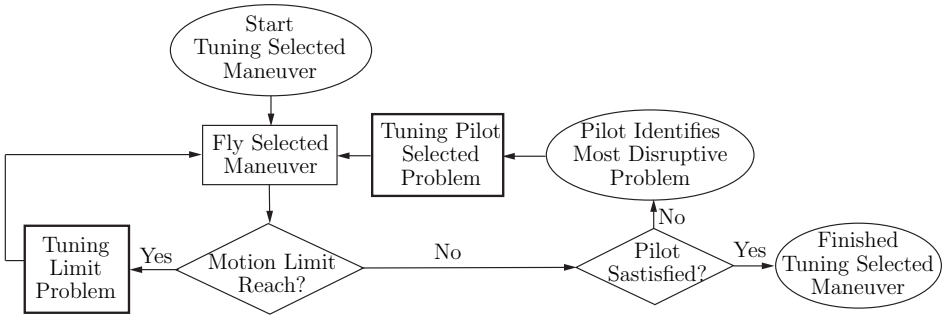


Figure 5.1: Expert tuning system introduced by Grant and Reid (1997b)

and designers must be experienced in adjusting parameters for removing the problems. Hereby, the adjustment could either increase the motion or eliminate motion cues errors from the simulated quantities, and the tuning process ends when a pilot states his satisfaction after test-maneuvers. Altogether, the method requires experienced pilots attending the tuning process and a large amount of time for trial and error.

b) Average assessment of the maneuvers for drive-track

The method is based on selecting maneuvers with isolated individual DoFs to find the optimized parameters for a MCA. Hereby, several subjects perform various settings (different sets of parameters) for different drive-tracks or maneuvers, and the final parameters are determined from the overall individual average values which describe the test subjects' evaluation for each drive-track taken into account in the tuning process (Bruenger-Koch (2005), Fischer (2009)). The method is much simpler than that for individual drivers, and it saves considerable time within the tuning process.

5.2.2 Objective method for analysis of washout filters responses

Objective methods were introduced with the assumption that simulated specific forces and angular velocities dominate the motion perception. In the literature, four objective methods have been used to tune parameters of washout filters, which are briefly summarized in the sequel.

a) Degrading effect of motion cue errors

Based on pilot statements involved to the effect of distortion of simulated quantities on the motion fidelity, Grant and Reid (1997a) introduced a set of tuning rules and

Motion cue errors	Motion cues	Source of errors	Motion fidelity	Solution
False cues	Opposite direction	Sc_1, Sc_2, Sc_5	Large decrease	$\downarrow k$ or $\uparrow \delta_S$
	Unexprected	Sc_3, Sc_6	Large decrease	$\downarrow k$
	High-frequency distortion	Sc_1, Sc_4, Sc_5, Sc_6	Large decrease	$\downarrow k$ or $\uparrow \delta_S$
Scaled/Missing cues	High-frequency transient	Sc_7	Decrease, jerkiness	$\downarrow \omega_n$
	Low-frequency transient	Sc_8	Decrease	$\downarrow k$ or $\uparrow \delta_S$
	Initial, transient, sustained	Sc_9	Decrease	$\uparrow k$
Phase errors	Phase lead	Sc_{10}	Insignificant	
	Phase lag	Sc_{11}	Significant	$\downarrow k$ or $\uparrow \delta_S$

Sc_1: Workspace limited; Sc_2: Washout effect; Sc_3: G-Tilt; Sc_4: Tilt-rate limit; Sc_5: Tilt-coordination remnant; Sc_6: Location of rotational point; Sc_7: Reproducing only very high frequency part; Sc_8: Reproducing only very low frequency part; Sc_9: Two small scale factor; Sc_{10}: High-pass filter effect; Sc_{11}: Low-pass filter effect or missing low-frequency transient.

Table 5.2: Motion cues errors and tuning rule introduced by Grant and Reid (1997a)

requirements for tuning parameters of classical washout filters. For the tuning rules, the authors consider eight types of motion cues grouped in three motion cue errors and eleven sources of false cues (Sc_i, $i \in \{1 \cdots 11\}$) listed in Tab. 5.2. It can be seen that one type of motion cue error can be due to various sources. For example, "'opposite direction" cues are caused by one of three sources of errors which are limited workspace, washout effects, and remnant of the tilt angle. Moreover, the limited workspace and the redundant tilt angle also cause relative high-frequency distortion.

The effect of parameters on the motion cue errors are marked by an up/down arrow indicating how a change of their value contributes to a reduction of cue errors, i.e. an up arrow ("↑") indicating that a reduction of motion cue error is achieved by an increase in the value and a down arrow ("↓") by a reduction of the values. For example, the opposite direction cues can be eliminated by reducing the gain k or increasing the rotational threshold value δ_S (see Tab. 5.2).

In Fig. 5.2, comparing with the target specific force f_{Vy}, the simulated specific force f_{Sy} has various false cues including opposite, unexpected and distorted cues. Moreover, the simulated specific force has a large phase-lag ($\delta t = t_1 - t_0$) and small benefit cues. As a result, motion fidelity is significantly degraded. The illustration of other errors and their sources of errors were described in detail in the report of Pham (2015).

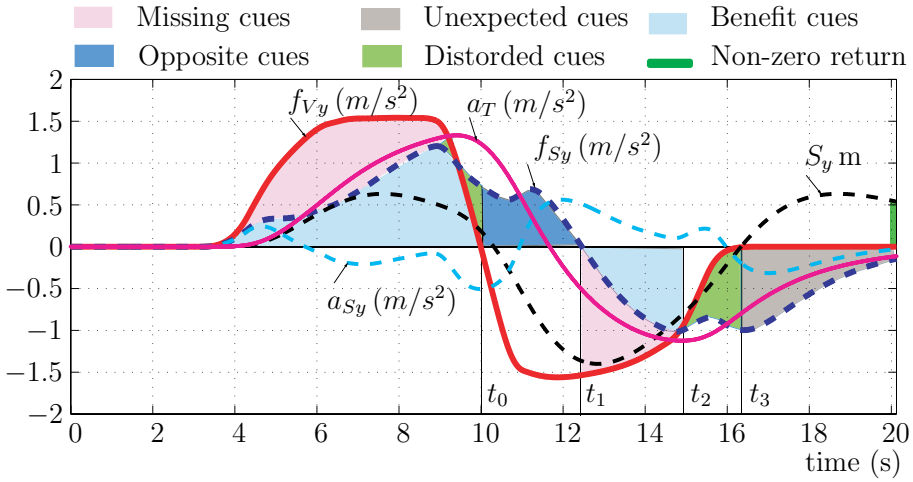


Figure 5.2: Demonstration of various types of motion cues errors

b) Parameter auto-tuning using fuzzy theory

Song et al. (2003) mentioned that fuzzy logic is effectively applicable to a system in which the inputs-outputs are not sharply defined. Thus, they designed a fuzzy control which takes displacement limit, angle limit and the low-frequency specific force as

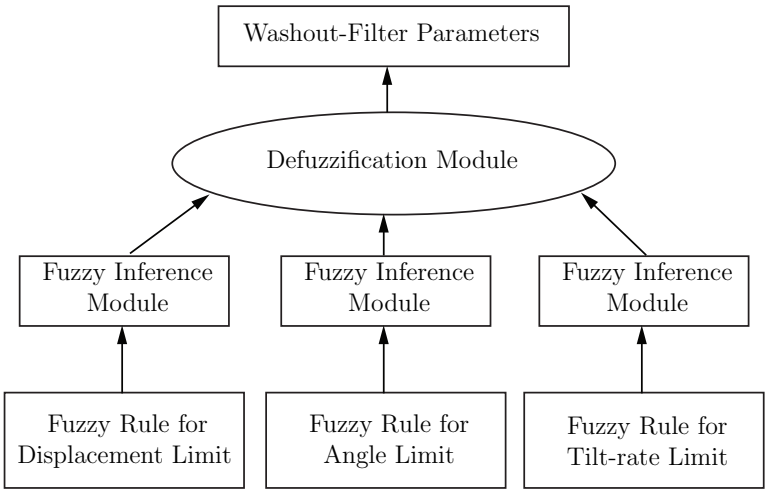


Figure 5.3: Tuning method based on fuzzy theory (Song et al. (2003))

inputs and then defines proper filter parameters as outputs. The tuning method was applied in a classical algorithm in which the adjustment of the natural frequency of washout filters ω_n was based on the state of the position and angle, while the tilt-rate limit δ_S, the scale factor k are adjusted according to the low-frequency specific force. Later, Chen and Fu (2011) used the feedback position of the simulator via a camera to design a fuzzy controller, which adjusts the break frequency of the high-pass filter approximately. In the design of a fuzzy controller, three conditions of simulator displacement and brake frequency of filters were defined: small, medium, large.

In another approach, Hwang et al. (2009) designed a fuzzy controller for optimal washout filters to produce a compensating signal $\underline{u}_F$ to eliminate the errors $\hat{\underline{e}}$ (Fig. 5.4). The $\underline{u}_F$ considered the sensation errors $\hat{\underline{e}}$ and the time derivative of the input $\underline{u}_V$. The design of the fuzzy controller was derived from the discrimination of large, medium or

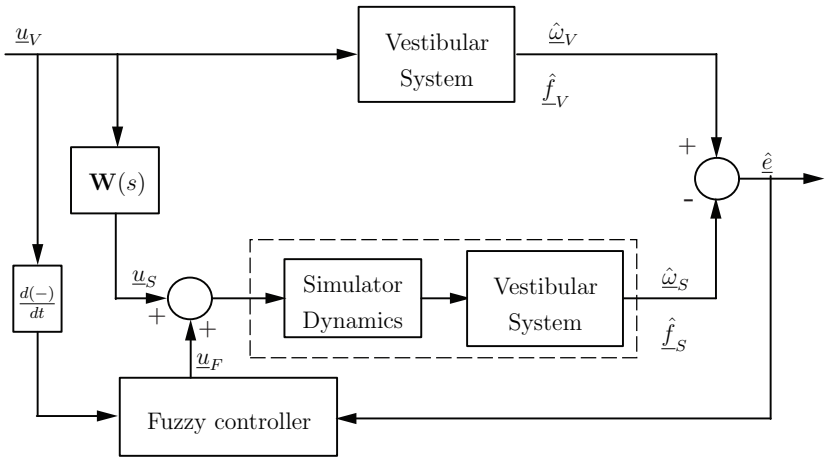


Figure 5.4: Fuzzy control for Optimal parameter Hwang et al. (2009)

small human motion perceptions, although the precise level was not defined precisely. The fuzzy logic classified the sensation errors and the time derivative of the inputs into 5 conditions: negative large, negative small, zero, positive small, and positive large, from combinations of which suitable conditions for $\underline{u}_F$ were determined.

c) Tuning method by flattening Bode plot

The method focuses on reproducing accurately the target specific force by combining the responses of a high-pass filter (HP) and a low-pass filter (LP) (Romano (2003), Fischer (2009), and Jamson (2010)). For example, with scale factor $k = 1$ and $\mathcal{K}_{Ps} \equiv \mathcal{K}_S$, according to Fig. 3.3 on page 28 the target lateral specific force is ${}^{Pv}f_{Pvy}$, and the simulated lateral specific force is ${}^{Ps}f_{Psy} = {}^{Ps}a_{Psy} + \mathrm{g}\sin\varphi_S$. Due to small tilt angles φ_S, the simulated acceleration can be expressed as ${}^{Ps}a_{Psy} = {}^{S}a_{Sy} = a_{Sy}\cos\varphi_S \approx a_{Sy}$.

From the scheme of the classical algorithm (Fig. 4.1), the output of the high-pass filter HP is the simulator acceleration $a_{Sy} = a'_{Sy}\mathrm{HP} \approx {}^{V}f_{Vy}\mathrm{HP}$, and the output of the low-pass filter at tilt-coordination level is $\mathrm{g}\sin\varphi_S = {}^{V}f_{Vy}\mathrm{LP}$. Thus, the combined transfer function is

$$\mathrm{TF} = \frac{{}^{S}f_{Sy}}{{}^{V}f_{Vy}} = \frac{a_{Sy} + \mathrm{g}\sin\varphi_S}{{}^{V}f_{Vy}} \approx \frac{{}^{V}f_{Vy}\mathrm{HP} + {}^{V}f_{Vy}\mathrm{LP}}{{}^{V}f_{Vy}} = \mathrm{HP} + \mathrm{LP}\,. \tag{5.1}$$

If TF = 1, the simulated specific force ${}^{S}f_{Sy}$ tracks perfectly the target specific force ${}^{V}f_{Vy}$. Therefore, with a high-pass filter HP, the transfer function of the ideal low-pass filter is $\mathrm{LP}^* = 1 - \mathrm{HP}$ which was used by Wentink et al. (2008), Fischer (2009), and Giordano et al. (2010). On the other hand, Reid and Nahon (1985) defined $\mathrm{HP} + \mathrm{LP} \neq 1$, thus, ${}^{S}f_{Sy}$ has a deviation from ${}^{V}f_{Vy}$.

The Bode plot of Fig. 5.5 illustrates the magnitude (as gain in dB) and phase shift (in deg or rad) against the logarithmic angular frequency of the transfer function of linear system. Hereby, the Bode plot of the combined transfer function TF can be used to predict a difference between the simulated specific force and the target specific force. Concretely, if ${}^{S}f_{Sy}$ perfectly tracks ${}^{V}f_{Vy}$, the TF has a magnitude of 0 dB and a phase shift of 0° in time history response.

Fig. 5.5 shows two possible responses of washout filters with a $\mathrm{LP}^{\&} \neq 1 - \mathrm{HP}$ (solid red line) and the LP^* (dashed black line). If the HP+$\mathrm{LP}^{\&}$ is used, there is a jump in the phase plot and a large peak in a magnitude plot. Moreover, in the phase plot, a large reduction of magnitude at $2\,rad/s$ and a large phase shift exist. As a result, the

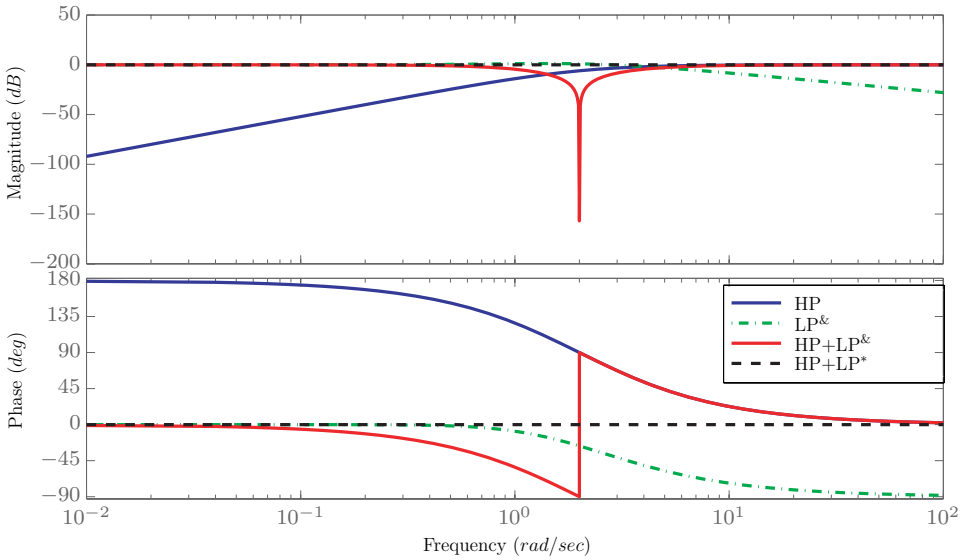


Figure 5.5: Bode plot of washout transfer functions

simulated specific force will be distorted at this frequency. In contrast, if HP + LP* is used, there is no phase shift and the magnitude value remains at $0\,dB$. Thus, the simulated specific force tracks perfectly the reference one.

In summary, flattening Bode plots aims to reduce a gap (or a sharp peak) that appears in the response of washout filters at a range of frequency. By doing that, the simulated specific force can track better the target specific force.

d) Tuning method using "performance indicator" and "good criterion"

Another way to evaluate motion perception error, more directly than the flattening Bode plots of washout filters is through a "performance indicator" and a "good criterion" which score errors of specific force and angular velocity as well as their first derivatives. In this case, the tuning method consists in minimizing the scores of the simulated signals compared with the target ones.

The "performance indicator" (Pouliot et al. (1998)) consists of two indicators λ_1 and λ_2, both split into a translational part (index "f") for the specific force, and a rotational part (index "ω") for the angular velocity, and normalized by the maximal motion acceleration a_{max} and the maximal angular velocity ω_{max} of the investigated simulator, respectively. Hereby, λ_1 describes the normalized average of the square error between the target and simulator motion signals, and λ_2 describes the normalized average square error of the corresponding rates of change, respectively:

$$\lambda_1 = 100 \cdot \left(\frac{\lambda_{1f}}{a_{max}} + \frac{\lambda_{1\omega}}{\omega_{max}} \right) \; ; \lambda_2 = 100 \cdot \left(\frac{\lambda_{2f}}{a_{max}} + \frac{\lambda_{2\omega}}{\omega_{max}} \right). \tag{5.2}$$

Similarly, the "good criterion" (Fischer (2009)), splits the translational and rotational parts into a "scale" component (index "*sc*") describing the difference of magnitudes between the target and simulated motions, and a "shape" component (index "*sh*") describing the difference of shapes when both curves are scaled similarly (Eq. 5.3). The scale and shape error components for the specific force are described as follows.

$$\lambda_{qp} = (\lambda_{qp,sc} + \lambda_{qp,sh}) \quad , q \in \{1,2\}\,;\, p \in \{f,\omega\}\,. \tag{5.3}$$

Let $f^j_{Vm}, f^j_{Sm}, m \in \{x,y,z\}$ be the target ("vehicle") and simulated specific forces, respectively, at time t_j, and let N+1 sampling points $t_0,\ldots,t_N$ be measured. With the global internal scale factor

$$k_{Sm} = \frac{\sum_{j=0}^{N} |f^j_{Sm}|}{\sum_{j=0}^{N} |f^j_{vm}|} \quad , m \in \{x,y,z\} \quad , \tag{5.4}$$

the differences at the sampling points are computed as

$$\Delta f^j_{m,sc} = f^j_{Vm}(1-k_{Sm}) \quad ; \Delta f^j_{m,sh} = k_{Sm} f^j_{Vm} - f^j_{Sm} \quad , \quad m \in \{x,y,z\} \quad , \tag{5.5}$$

which yield the partial "good" components

$$\lambda^*_{1f,sc} = \frac{1}{N}\sum_{j=0}^{N} \sqrt{\sum_{m\in\{x,y,z\}} (\Delta f^j_{m,sc})^2} \quad ; \lambda^*_{1f,sh} = \frac{1}{N}\sum_{j=0}^{N} \sqrt{\sum_{m\in\{x,y,z\}} (\Delta f^j_{m,sh})^2} \quad . \tag{5.6}$$

From this, the "good criteria" and their time derivatives are obtained as

$$\lambda^*_1 = (1/f_{\text{norm}}) \cdot \lambda^*_{1f} + \ (1/\omega_{\text{norm}}) \cdot \lambda^*_{1\omega}\,, \tag{5.7}$$

$$\lambda^*_2 = (1/\dot{f}_{\text{norm}}) \cdot \lambda^*_{2f} + (1/\dot{\omega}_{\text{norm}}) \cdot \lambda^*_{2\omega} \quad , \tag{5.8}$$

with

$$f_{\text{norm}} = \max\left(|f_{Vm}|\right) \quad (m \in \{x,y,z\})\,, \tag{5.9}$$

$$\dot{f}_{\text{norm}} = \max(|\dot{f}_{Vm}|) \quad (m \in \{x,y,z\})\,, \tag{5.10}$$

$$\omega_{\text{norm}} = \max(|\omega_{vx}+\omega_{sx}|, |\omega_{vy}+\omega_{sy}|), |\omega_{sz}|)\,, \tag{5.11}$$

$$\dot{\omega}_{\text{norm}} = \max(|\dot{\omega}_{vx}+\dot{\omega}_{sx}|, |\dot{\omega}_{vy}+\dot{\omega}_{sy}), |\dot{\omega}_{sz}|)\,. \tag{5.12}$$

In this thesis, based on Grant et al. (2003), normalized good criteria were defined as

$$\lambda_1^{\&} = (1/\delta_O) \cdot \lambda_{1f}^* + (1/\delta_S) \cdot \lambda_{1\omega}^* \quad ; \quad \lambda_2^{\&} = (1/\delta_O) \cdot \lambda_{2f}^* + (1/\delta_S) \cdot \lambda_{2\omega}^* \quad , \tag{5.13}$$

with δ_O and δ_S are the threshold values of the otolith and semicircular organs, respectively. The smaller numerical indices, the better the quality of the simulated signals.

5.2.3 Subject-objective tuning method

The subject-objective tuning methods are based on a set of numerical criteria that are derived from motion perception fidelity assessments of test subjects in the specific simulation tasks. The numerical criteria are constructed by correlating subjective assessments and parameters of classical washout filters. In the literature, two types of subject-objective numerical criteria have been proposed which are further regarded in this thesis:

a) Tuning washout filters using the Sinacori motion fidelity criteria

The method was first introduced by Sinacori (1977) and then refined with boundaries of levels of motion perception fidelity by Schroeder (1999). Based on the empirical data of subjective experiments related to motion perception fidelity, the relation between the response of the high-pass filter and a level of motion perception fidelity was established. Concretely, there are three Motion Fidelity Scales, (MFS), including high, medium and low levels which correlated pilot opinion and platform motion magnitude attenuation and phase distortion. The boundaries of the levels of motion perception fidelity were built from the natural break frequency ω_n and the gain factor k (scale factor) of the high-pass filter. According to the Sinacori/Schroeder motion fidelity criterion, high fidelity is obtained when the gain at 1 rad/s is in the range of $[0.6\,, 1]$, and the phase shift smaller than 30°. The tuning method is simple and used solely to tune parameters for the classical algorithms or optimal washout filters.

b) Tuning classical washout filters using the Zaichik criteria

Zaichik et al. (2010) introduced different motion perception fidelity criteria that independently include characteristics of washout filters and the simulation tasks (precision tasks or maneuvers) as well as the role of simulated acceleration, namely beneficial or negative effect. The beneficial effect denotes the perceived cues supporting the

operation of a simulated vehicle. On the other hand, the negative effect denotes an unpleasant feeling of pilots under the motion cues. Unlike the Sinacori criteria, these criteria feature separated effects for the scale factor and the break frequency of the high-pass filter. For example, motion perception fidelity criteria for precision control tasks – such as required for simulation of roll accelerations – are defined through the break frequency $\omega_{n\varphi}$ of the corresponding high-pass filter: 1) $\omega_{n\varphi} \leq 0.7(s^{-1}) \rightsquigarrow$ fidelity is high; 2) $\omega_{n\varphi} \geq 4(s^{-1}) \rightsquigarrow$ fidelity is medium; 3) $0.7 < \omega_{n\varphi} < 4(s^{-1}) \rightsquigarrow$ fidelity corresponds to fixed-base simulation.

A number of common parameters for all MCAs were investigated to find their possible boundary with respect to the motion perception fidelity of simulation tasks. The most significant concerned parameter is the scale factor that scales down the amplitude of the input signals. There are various suggestions for the selection of a scale factor, for example, due to either "visual-vestibular discrepancy" (Groen et al. (2001)) or the effect of a jerk (Grant and Haycock (2008) and Soyka et al. (2009)), often a unit scale factor provides too powerful motions, so it becomes necessary to scale down specific force inputs. However, Grant et al. (2003) and Berthoz et al. (2013) mentioned that driving performance is significantly degraded if the motion cues are so much reduced or absent. Several authors thus proposed the non-linear scale factors (Telban et al. (2005), Chapron and Colinot (2007)) in which the scale factor decreases with the increase of the stimuli amplitude of input signals (Feenstra et al. (2009)).

Additionally, determining threshold values of angular velocity and angular acceleration play an important role in a simulation task. Although Fischer (2009) stated that the simulation with an unrestricted angular velocity can be better with a restricted one, the restriction of rotational motion is much preferred in almost all simulation tasks in the literature. However, the accurate thresholds apparently depend on various aspects of simulation tasks, such as mental load (Hosman and Van der Vaart (1978)), simulation with or without visual cues (Valente Pais et al. (2006); Groen et al. (2006)), and amplitudes of stimulating signals (Chapron and Colinot (2007); Zaichik et al. (1999)). Moreover, an interaction of drivers with the environment during driving actions has also an effect on the threshold values. As a result, various threshold values were determined and applied in different simulation tasks.

Last but not least, a ratio of linear acceleration to target specific force has also effect on the motion fidelity. Few preliminary results were introduced in publications, such as Jamson (2010) and Stratulat et al. (2011), however, more validations are required for using the suggestion in a tuning process. In this thesis, several suggestions of these parameters mentioned in this part are shown in Tab. 5.5.

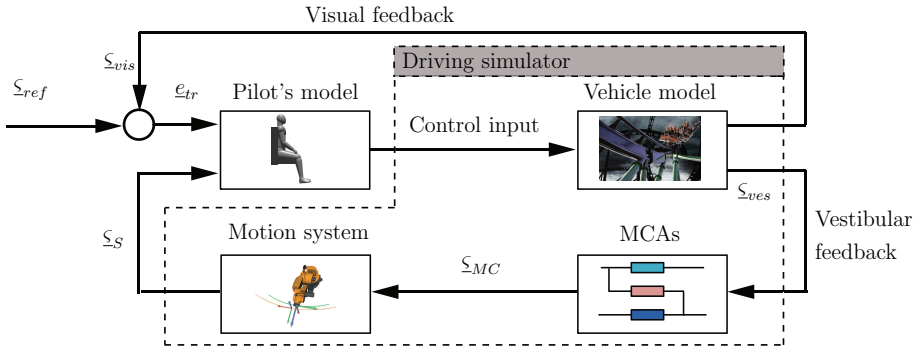


Figure 5.6: The pilot simulated-vehicle control loop (modified from Hosman and Stassen (1999))

5.2.4 Tuning methods using models of pilot behavior and self-motion perception

a) Model of pilot behaviors

Fig. 5.6 shows pilot-simulated vehicle control loop in which $\underline{\varsigma}_{vis}$ and $\underline{\varsigma}_{ves}$ are estimated the visual and vestibular quantities generated by the vehicle dynamic model, respectively. The vestibular quantities are processed by the MCAs to give the filtered quantities $\underline{\varsigma}_{MC}$ that enter into the simulator dynamics to generate the simulated vestibular feedback $\underline{\varsigma}_S$. Then, from the visual tracking error $\underline{e}_{tr}$ and motion perception $\underline{\varsigma}_S$, a pilot model behavior generates the corresponding operation.

Hosman et al. (2002) tuned the parameters of classical washout filters by minimizing the cost function

$$J = \sum \left(\underline{e}_{tr}^2 + Q \cdot {\delta_c}^2 + R \cdot {\dot{\delta}_c}^2 \right) , \tag{5.14}$$

where $\underline{e}_{tr}$ is visual tracking errors; Q and R are weighting factors of the aircraft characteristics and the tasks; δ_c is the control output as an inceptor displacement; and $\dot{\delta}_c$ is the first derivative control output as an inceptor velocity.

Currently, four pilot's models existing in the literature: are the quasi-linear model (McRuer and Krendel (1974)), the structure model (Hess (1997)), the descriptive model (Hosman and Stassen (1999)) and the optimal model (Kleinman et al. (1970)). However, the structure and the parameters of pilot models were only developed according to flight task maneuvers. Thus, the utility the pilot model to tune a MCA for a general specific drive task is still an open issue of research.

b) Self-motion perception models

Another approach for tuning MCA parameters is to use mathematical models of the motion perception, i.e. the visual and vestibular systems. The usage is based on the hypothesis that minimizing the predicted errors of motion perception in a driving simulator could improve motion fidelity. This hypothesis was also mentioned in the development of the optimal washout filters introduced by Sivan et al. (1982), Telban and Cardullo (2001), and Mayrhofer et al. (2007).

Fig. 5.7 shows the reference input signals of a vehicle $\underline{\varsigma}_{ref}$ entering both the "MCA" and the "perception" block to generate perceived motions $\hat{\underline{\varsigma}}_{ref}$ and filtered signals $\underline{\varsigma}_{MC}$. The filtered signals are fed into the "simulator" block to produce the simulator motion $\underline{\varsigma}_{sim}$ that then enters another "perception" block to generate perceived motions $\hat{\underline{\varsigma}}_{sim}$. The predicted errors $\hat{\underline{e}}$ are the difference between $\hat{\underline{\varsigma}}_{ref}$ and $\hat{\underline{\varsigma}}_{sim}$. The smaller the predicted errors are, the better parameters were obtained. Note that the perception block in Fig. 5.7 could be the vestibular model as used in Sivan et al. (1982), Reid and Nahon (1985), or the visual-vestibular interaction models such as Van der Steen (1998), Telban et al. (2005), Groen et al. (2007).

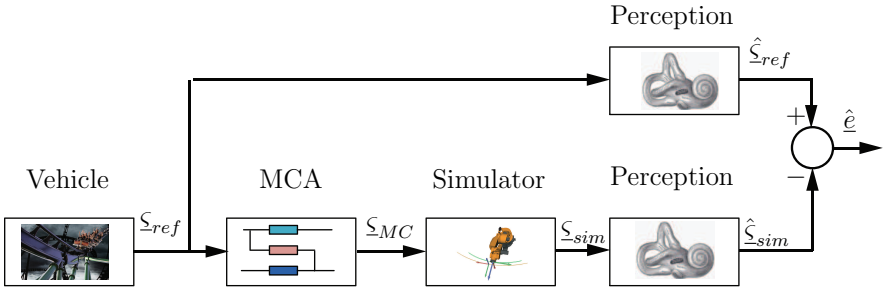


Figure 5.7: Tuning MCA based on analysis of motion perception generated by model of self-motion perception

5.2.5 Remarks on the parameter tuning methods

Fig. 5.8 summarizes the current problem of a tuning process. Normally, the tuning process depends on the used drive–track, test drivers, MCAs as well as tuning criteria (oranges ellipse). Current tuning approaches can be categorized into two major approaches (dashed box): improving motion fidelity or minimizing errors of motion cues. However, the relation between two approaches is not yet investigated (the black box in Fig. 5.8) and the approaches have been applied only independently for a given application, but not in combination of both.

Considering motion fidelity, Tab. 5.3 on page 76 summarizes advantages and disadvantages of the different existing tuning methods. The tuning criteria can be either fidelity questions to obtain the driver's assessment of a simulation session or subjective-objective criteria, such as the Sinacori criteria or model pilot behavior.

Concerning motion cues errors, simulated motion cues are compared with the target ones after each trial with a set of parameters. Errors of the simulated quantities can be tuned primarily by numerical assessments and by graphical illustrations of motion cue errors as well as by fuzzy theory. In addition, flattening the Bode plots of washout filters of classical algorithms can reduce distortion of the specific force.

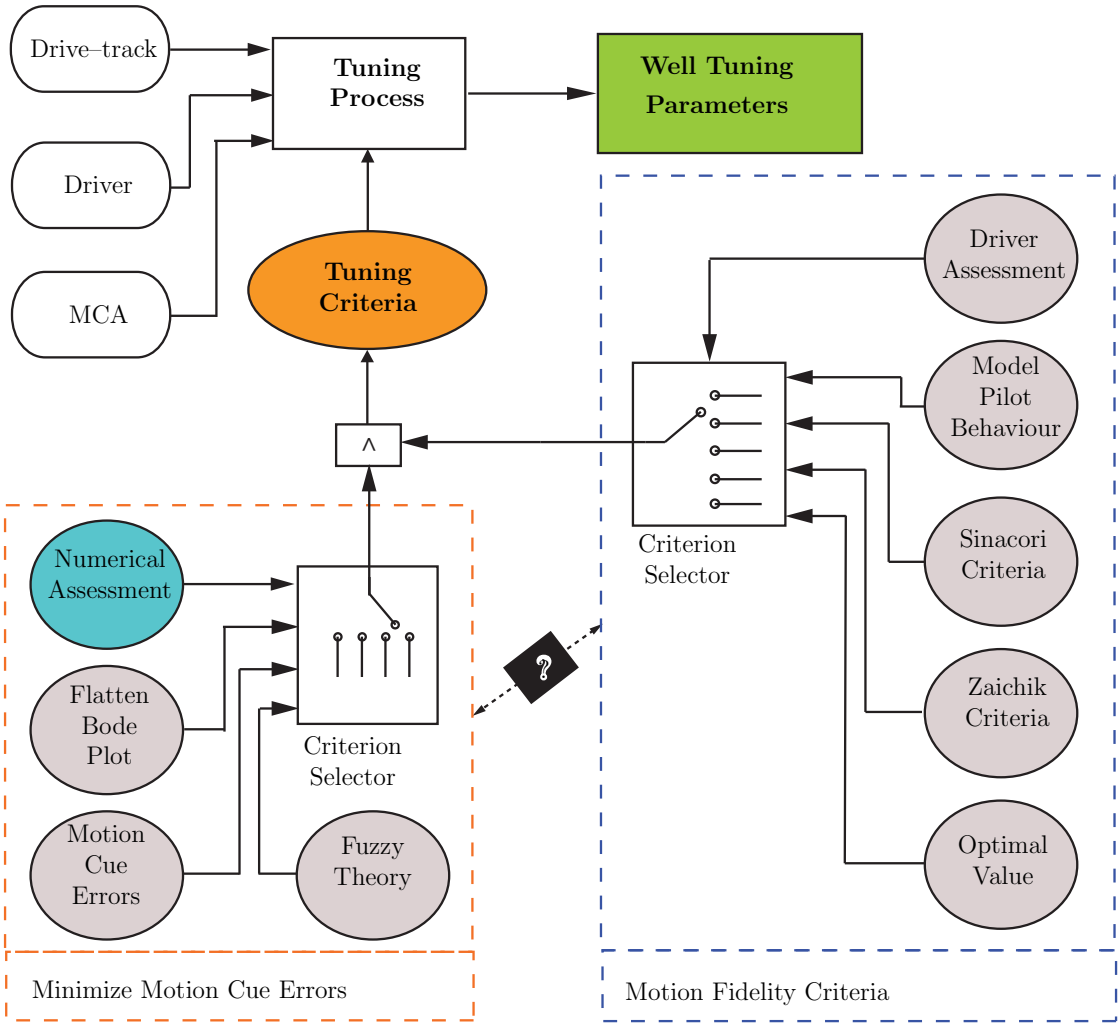


Figure 5.8: Process for finding well-tuned parameters (a cyan eclipse represents the DIL method; Gray eclipses represent the SIL methods)

TM	Advantages	Disadvantages
MDB (5.2.4a)	– Allows for a large number of parameters – Implementable for all MCAs – Objective and replicable	– Depends on the goodness of the models – Unclear if the Least-Square approach is suitable – The effects of scale and signal shape are not separated – Needs tuned parameters for the models
SPM (5.2.4b)	– Allows for a large number of parameters – Implementable for all MCAs – Objective and replicable	– Depends on the goodness of the models – Unclear if the Least-Square approach is suitable – Different environments for investigating self-motion models and validating drive simulation are necessary
AWF (5.2.2a ,c,d)	– Implementable for all MCAs – Objective and replicable – Predictability of several effects of false cues and motion fidelity	– Unclear correlation between the numerical measures and motion fidelity – The tuning objective needs further investigation – No clear definition of the effect of motion cue errors – Bode plots only for classical optimal algorithms
SiCoZ (5.2.3)	– Objective and replicable, considering the limits of the simulator – Overview of possible combination of parameters	– Questionable if the filters characteristics at 1 (rad/s) has an important influence – Used only for classical and optimal washout filters
IDr (5.2.1a)	– Exact result the test subjects – No effect of uncertain models	– Only for parameters with physical meaning – The goodness depends on the individual driver – Low consistency between drivers – Not replicable, time consuming – Unclear comparative questions
MDT (5.2.1b)	– Implementing each MCA and maneuver only one time – No effect of uncertain models – replicable	– Time consuming – Simulator study required
FTh (5.2.2b)	– Objective and replicable – Takes the limit of workspace and angular velocity into account	– No motion fidelity considered – Only for CL algorithm – Simulator study required
TM: Tuning method; MDB: Model of driver behavior (5.2.4a); SPM: Self-motion (5.2.4b); AWF: Analysis of washout filter response (5.2.2a,c,d); SiCoZ: Sinacori or Zaichik criteria (5.2.3) ; IDr: Individual driver (5.2.1a); MDT: Maneuver or drive track (5.2.1b); FTh: Fuzzy theory (5.2.2b)		

Table 5.3: Advantage and disadvantages of motion perception fidelity tuning methods

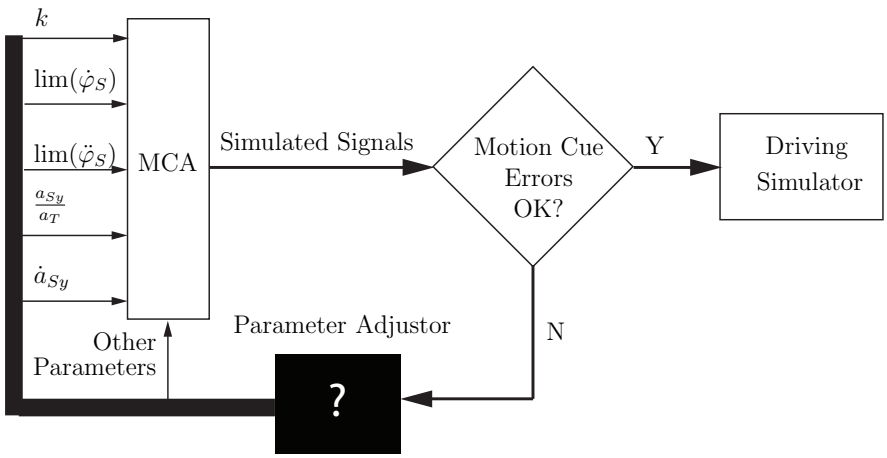


Figure 5.9: The implicit relationship between parameters and response of MCAs

As a result, the variety of tuning methods as well as the unclear relation between motion perception fidelity and motion cues errors can confuse designers in finding suitable parameters. In this setting, how serious false cues degrade motion perception fidelity and how the distortion of simulated motion effects on levels of motion fidelity are still open problems. To overcome these problems, a tuning process of MCA's parameters for high motion perception fidelity in a simulation task is addressed here in two steps:

Step 1: Rough tuning of parameters for either minimizing motion cue errors or optimizing subjective-objective criteria; **Step 2**: Validating and fine-tuning the achieved parameters by subjective tests. Suitable parameters of a MCA that generate simulated motions having good quality are commonly determined by trial and error (TnE) or automatically (AT) within the restricted requirements.

Concerning the rough tuning step, in the literature, the effects of common parameters on motion fidelity such as scale factor, jerk, and the ratio of a linear acceleration to a target acceleration along with threshold of angular velocity and angular acceleration were investigated. For example, Grant and Reid (1997a) first analyzed the effects of the washout filters' parameters on the existence of false cues. Then, the authors proposed rules to reduce the undesired cues by adjusting the scale factor k and break frequency of the washout filters. Furthermore, for the ZyRo and MPC algorithms, simulated quantities are directly influenced by a specific parameter in the weighting parameter matrices.

However, the relationship between MCA's parameters and the amount of motion cue errors generated is unclear and there is no conventional process to suitably adjust the parameters (see the black box in Fig. 5.9). Therefore, a unified approach according to designer's experience is necessary.

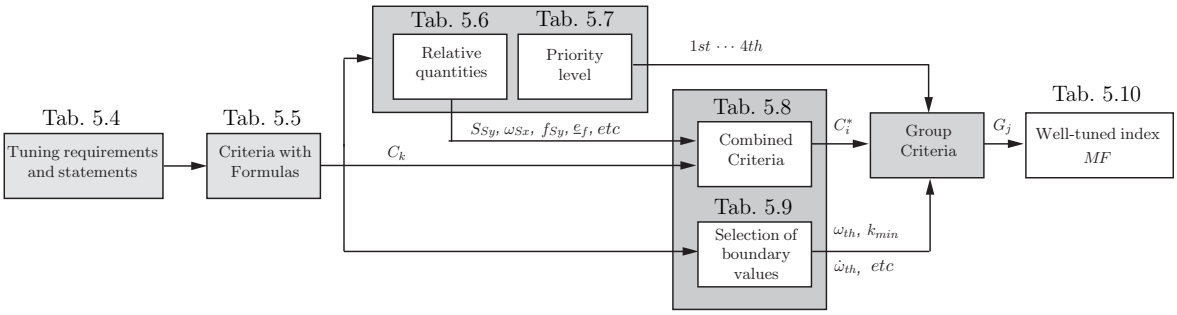


Figure 5.10: Flow diagram for proposing well-tuned criteria

5.3 Proposed objective criteria

In this thesis, a novel set of objective criteria, called the "well-tuned index" (MF), that can be used for every MCAs is proposed. The well-tuned index is based on published results of the effects of the common parameters appearing in the MCAs on motion perception fidelity, such as scale factors, threshold values, and the ratio of translational to rotational motion (Fig. 5.9).

Fig. 5.10 describes the steps through which the well-tuned index (MF) is built. In the first step, the general rules or requirements that are currently used for MCAs according to Tab. 5.4 are processed. In the second step, the statements and rules in the literature together with the descriptive expressions of the relative parameters listed in Tab. 5.5 are applied. In the third step, the relative quantities (Tab. 5.6) and the four priority or significancy levels (Tab. 5.7) are evaluated. In the fourth step, the combined criteria C_i^* with $i \in \{1 \cdots 5\}$ (Tab. 5.8) are constructed to categorize the listed criteria C_k with $k \in \{1 \cdots 14\}$ (Tab. 5.5) with respect to the relative quantities in Tab. 5.9. Additionally, the boundary values of the quantities (Tab. 5.9) are chosen to make practical criteria for the driving simulator. In the fifth step, the group criteria G_j with $j \in \{1 \cdots 4\}$ are divided into the significant levels of the combined criteria C_i^* in regard to their significant levels. In the last step, the well-tuned index MF is built from the group criteria G_j (5.3.4 on page 87). The tuning process is first explained in general and then applied to the specific task of transversal acceleration tracking with the LMR drive simulator.

5.3.1 Tuning rules and statements in the literature

Tab. 5.4 shows the rules found in the literature for either tuning of parameters. These found rules are summarized in Tab. 5.5 by 14 characteristic conditions that have been

General rules for tuning a MCA

1. Implementability in the simulator
   a. Motion inside the workspace of a driving simulator.
   b. Simulated acceleration and velocity do not exceed the limit values of the specific drive simulator.
2. Improvement of the motion fidelity
   a. Removal potential false cues
      i. Rotational false cues
      ii. Cue distortion by unexpected cues or opposite cues
      iii. Missing cues
      iv. Phase lag
   b. Identification of suitable quantities via subjective assessment
      i. Scale factor
      ii. Threshold values for angular velocity and acceleration
      iii. Break frequency
      iv. Use the tilt-coordination when there is no better choice
      v. Find the best ratio of translational acceleration and tilt acceleration

Table 5.4: General rules for tuning parameters of MCAs in the literature

used in published experimental setups. Although almost all experiments were implemented with the classical algorithm, the formulations can be applied to other MCAs since they describe conditions for common parameters appearing in almost all MCAs.

Formulation	Description and reference
Condition: C_1 – $S_y \leq S_{ylm}$ – $\varphi_S \leq \varphi_{lm}$ – $\dot{S}_y \leq \dot{S}_{ylm}$ Rules: – *1a* – *1b*	Simulated motion is restricted in the simulator workspace • Grant and Reid (1997a) • Grant and Reid (1997b) • Gouverneur et al. (2003) • Fischer (2009), etc.
Condition: C_2 – $\varphi_S \leq \varphi_{S,max}$ Rules: *2aii*	Avoid the distortions of the subjective vertical cues when the tilt angle exceeds $20° - 30°$ (Aubert effect) (Reymond and Kemeny (2000))

<table>
<tr>
<td>Condition: C_3
– $\omega_{Sx} \leq \omega_{th}$
Rules:
– 2ai
– 2bii</td>
<td>Avoid rotational false cues by threshold ω_{th} on angular velocity (different values in the literature).
• Nesti et al. (2012): (++)
• Chapron and Colinot (2007): (++)
• Zywiol and Romano (2003): (++)
• Jamson (2010): (+*) and (+)
• Reid and Nahon (1985): (*+)
• Groen and Bles (2003): (*+)
• Telban et al. (2005): (**) and (*+)
• Fischer (2009): (*+) and (*)
• White and Rodchenko (1999): (–)
• Zaichik et al. (2010): (–)
• Pouliot et al. (1998): (x)
<table>
<tr><td>Symbols</td><td>++</td><td>+*</td><td>*+</td><td>**</td><td>+</td><td>–</td><td>x</td></tr>
<tr><td>$\omega_{th}(deg/s)$</td><td>6</td><td>5.8</td><td>3</td><td>2</td><td>8.59</td><td>30</td><td>2.6 – 3.6</td></tr>
</table>
</td>
</tr>
<tr>
<td>Condition: C_4
– $\dot{\omega}_{Sx} \leq \dot{\omega}_{th}$
Rule:
– 2ai
– 2aii
– 2bii</td>
<td>Limit simulated tilt angular acceleration to prevent false rotational perception and unexpected specific force due to the distance of the driver head to the rotation center.
• Chapron and Colinot (2007):
– $\dot{\omega}_{th} = 8\,deg/s^2$ with $a = 0\,m/s^2$
– $\dot{\omega}_{th} = 11\,deg/s^2$ with $a = 1\,m/s^2$
• Fang and Kemeny (2012): $\dot{\omega}_{Sx} < 12\,deg/s^2$
• Reymond and Kemeny (2000): limit $\dot{\omega}_{Sx,y}$
• Groen et al. (2001): $\dot{\omega}_{th} = 3\,deg/s^2$</td>
</tr>
<tr>
<td>Condition: C_5
– $f_T \leq k_{fT}$
Rule: 2aii</td>
<td>Restrict the undesired specific force caused by tilt angular acceleration when the position of the rotation center is at the driver's head by a factor $k_{fT} = 0.1\,m/s^2$.
• White and Rodchenko (1999)
• Zaichik et al. (2009), (2010)</td>
</tr>
</table>

Condition: C_6 – $\frac{e^*_{f,sh}}{\delta_O} \leq k_e$ – $\frac{e^*_{\omega,sh}}{\delta_S} \leq k_e$ Rule: – *2aii* – *2aiii* – *2aiv*	Limit error cues for tilt and translation • Sivan et al. (1982) presented motion perception in threshold unit and minimized motion error below pilot's threshold. • Pouliot et al. (1998) presumed that two sensations that differ by less than the threshold will not be detected • Grant et al. (2003) normalized the shape error of the specific force and angular velocity by their threshold values
Condition: C_7 – $\downarrow e_{f,sh}$ or $\downarrow \lambda_{1f,sh}$ Rule: – *2ai* – *2aii* – *2aiii*	Flatten the Bode plot of response of the washout filters to reduce the form error of specific force. • Romano (2003) • Bruenger-Koch (2005) • Fischer (2009) • Jamson (2010)
Condition: C_8 – $\downarrow \underline{e}_f$ or $\downarrow \lambda_{1f,sh}$ Rule: – *2aii* – *2aiii* – *2bi* – *2biii*	Minimize shape errors of specific force. • Grant and Reid (1997a) minimized the distortion and missing cues of the simulated specific forces • Zaichik et al. (2012) suggested a tuning procedure that reproduces "useful cues" and minimizes the "false cues".
Condition: C_9 – $\frac{\int a_{Sy} dt}{\int a_{Vy} dt} \geq k_{a,min}$ Rule: *2biv*	Use more transversal movement in simulation of lateral acceleration to improve the realism • Jamson (2010) • Damveld et al. (2012) • Chapron and Colinot (2007) • Valente Pais et al. (2009)
Condition: C_{10} – $\frac{\int a_{Sy} dt}{\int a_{Vy} dt} \geq k_{a,min}$ Rule: *2aiii*	Avoidance of the use of the translational movement to simulate only very high-frequency part of target acceleration, Grant and Reid (1997a)
Condition: C_{11} – $\frac{\int a_{Sy} dt}{\int a_T dt} \approx \frac{65}{35}$ Rules: *2bv*	Introduce best ratio of translational acceleration and tilt acceleration due to tilt angle for half-sinusoid acceleration of simulation in brake-case, Stratulat et al. (2011)

Condition: C_{12} – $A(\text{HP}) \geq 0.6$ – $\alpha_p \leq 20^0$ Rules: – *2bi* – *2biii*	Increase the amplitude of high-pass filter HP (A(HP)) and reduce the phase lag α_p at critical frequency 1 rad/s. • Sinacori (1977) • Schroeder (1999) • Gouverneur et al. (2003)
Condition: C_{13} – $k \in [k_{min},\, k_{max}]$ Rule: – *2aiii* – *2bi*	Proper definition of scalar factor because drivers have tolerance with the scale factor, and too small motion cue degrades the motion perception fidelity • Berthoz et al. (2013): $k \in [0.4\,, 1]$, $k_{op} = 0.7$ • Grant et al. (2003): $k = 0.5$, but $k = 0.3 \downarrow$ motion fidelity • Jamson (2010): $k_{max} = 0.9$, but $k = 0.5$ is better • Feenstra et al. (2009): $k \in [0.4\,, 1]$ • Telban and Cardullo (2001); Telban et al. (2005): 3rd order nonlinear scale factor • Chapron and Colinot (2007); Feenstra et al. (2009): dynamic scale factor
Condition: C_{14} – $k \leq \frac{0.012}{\delta_O}$ Rule: *2bi*	Avoid perception as in fixed simulators with small scale factor • Zaichik et al. (2009), (2010) • White and Rodchenko (1999)

Abbreviations:

- $\underline{e}_f = \underline{f}_V - \underline{f}_S$ represents the specific force errors
- $\underline{e}_\omega = \underline{\omega}_V - \underline{\omega}_S$ represents the specific force errors
- k denotes the scale factor
- $f_T = h \cdot \omega_{Sx}$ represents false specific force errors due to the angular acceleration
- h represents distance between the center of rotation with the driver's head
- $e^*_{f,sh} = max(|\underline{e}_{f,sh}|]$ represents the maximum form error of the specific force
- $e^*_{\omega,sh} = max(|\underline{e}_{\omega,sh}|)$ represents the maximum form error of the angular velocity
- $\delta_O = 0.17\,(m/s^2)$ represents the threshold value of the otolith system
- δ_S represents the threshold value of the semicircular system
- k_e represents the maximum ratio of normalized form error to the threshold of vestibular system
- k_{fT} represents the maximum specific force due to the tilt angular acceleration
- $k_{a,min}$ represents the minimum value of k_a, that is the ratio of the simulated translational acceleration to the target translational acceleration
- $\varphi_{S,max}$ represents the maximum value of the tilt angle for coordination technique

Table 5.5: Description of the fourteen principal tuning conditions found in the literature

5.3.2 Categorizing tuning criteria and priority levels of tuning rules

The criteria in Tab. 5.5 can be categorized with respect to physical quantities (Tab. 5.6) and tuning rules. Concretely, the criteria C_1 and C_2 are related to restricting angles and displacements of a driving simulator; $C_3 \cdots C_5$ are related to false cues due to the tilt motion that has an effect on the angular velocity, angular acceleration, the extra specific force; $C_6 \cdots C_8$ are related to false cues of simulated specific forces, where false cues are represented by numerical shape errors such as $\underline{e}_{f,sh}$, $\lambda_{1f,sh}$; $C_9 \cdots C_{11}$ are related to the ratio of linear acceleration to simulated specific forces; and $C_{12} \cdots C_{14}$ are related to scale factors that are used to scale down the input signals.

Priority levels of tuning rules

In more simulation cases, it is impossible to satisfy all the tuning criteria for a MCA. Therefore, a priority order of tuning criteria (5.10) is proposed with respect to the effect of the criteria on motion perception fidelity. The most significant rule is to make simulated motion implementable. The 2nd priority rule, based on the statement of Grant and Reid (1997a), and Zaichik et al. (2009), emphasizes the significant effect of the false cues and missing cues on the motion fidelity. The 3rd priority rule is the selection of suitable scale factors for a simulation (Grant et al. (2003), Berthoz

	Requirements											**Related quantities**
	1		2									
	1a	1b	2a				2b					
			i	ii	iii	iv	i	ii	iii	iv	v	
C_1	x	x										S_y, φ_S
C_2				x								φ_S
C_3			x					x				ω_{Sx}, ω_{th}
C_4			x	x				x				$\dot{\omega}_{Sx}$, $\dot{\omega}_{th}$
C_5				x								f_T, k_{fT}, $\dot{\omega}_{Sx}$
C_6				x	x	x						$e_{f,sh}$, k_e
C_7			x	x	x							$e_{f,sh}$, $\lambda_{1f,sh}$
C_8				x	x		x		x			$\underline{e}_f$, $\lambda_{1f,sh}$, ω_n
C_9										x		a_{Sy}, k_a
C_{10}					x							a_{Sy}, k_a
C_{11}											x	a_{Sy}, a_T
C_{12}					x		x					k, ω_n
C_{13}							x		x			k_S, k_{min}, k_{max}
C_{14}							x					k

Table 5.6: Categorization of tuning criteria with respect to physical quantities

(1st) - (Rule 1a,b) Avoiding limit violation of a simulator
(2nd) - (Rule 2ai $\cdots$ 2aiv) Reducing the false cues, missing cues, phase lag
(3rd) - (Rule 2bi) Selecting suitable scale errors
(4th) - (Rule 2biv, 2bv) Using more translational movement

Table 5.7: Priority levels for tuning process

et al. (2013)) to have high motion fidelity. The reason for this is that weak or strong motion has a less serious effect than false cues. The 4th priority rule is for using more translational motion for reproducing linear acceleration (Grant et al. (2003), Jamson (2010), Stratulat et al. (2011)). Since the usage is recommended only in a few publications, the rule has been ranked here after the three previous more relevant ones.

5.3.3 Combined criteria and range of parameters

Based on the priority levels, the criteria of Tab. 5.6 can be combined in five significant criteria C_i^*, $i \in \{1 \cdots 5\}$ The combined criteria are defined as:

- $C_1^* = C_1 \vee C_2$ – (blue): Limiting the simulator's movement and avoiding the vertical false cues due to large tilt angle
- $C_2^* = C_3 \vee C_4 \vee C_5$ – (red): Avoiding false rotational cues
- $C_3^* = C_6 \vee C_7 \vee C_8$ – (light blue): Avoiding false cues of the specific force
- $C_4^* = C_9 \vee C_{10} \vee C_{11}$ – (green): Selecting the suitable scale factors
- $C_5^* = C_{12} \vee C_{13} \vee C_{14}$ – (white): Using as much translational motions as possible to simulate linear accelerations.

Here, $\vee$ represents the "or" logic operator meaning the union of the corresponding sets of conditions. Since a high-pass filter only exists in the classical algorithm, the condition C_{12} is not general for other algorithms, thus it is possible to avoid the criterion, and $C_5^* = C_{13} \vee C_{14}$. In Tab. 5.8, mathematical formulations for the conditions are summarized. The criterion C_2^* includes the three inequalities: restricted angular velocity, angular acceleration, and extra specific force. The criterion C_3^* is represented by an inequality that restricts the maximum of shape errors of simulated specific force. The criterion C_4^* is represented by two inequalities to maintain suitably scaled signals. The criterion C_5^* limits the minimum amount of linear acceleration by using an inequality of the ratio of linear acceleration to the target acceleration. Note that the criteria C_2^* and C_3^* have an interactive relation that focuses on finding the trade-off between the rotational errors and specific force errors as mentioned e.g. by Jamson (2010). The trade-off is the equivalent to finding the balance between C_2^* and C_3^* for removing the

Combined criteria	Formulations
C_1^*	$S_{Sy} \leq S_{max};\ \varphi_S \leq 30^o$
C_2^*	$\omega_{Sx} \leq \omega_{th};\ \dot{\omega}_{Sx} \leq \dot{\omega}_{th};\ h \cdot \dot{\omega}_{th} \leq k_{fT}$
C_3^*	$\frac{e^*_{f,sh}}{\delta_O} \leq k_e$
C_4^*	$k_S \in [k_{S,min},\ k_{S,max}]$
C_5^*	$\frac{\int a_{Sy} dt}{\int a_{Vy} dt} \geq k_{a,min}$

Table 5.8: Combined criteria and theirs formulas

potential false cues due to the distortion of specific force or being over threshold values of angular velocities.

The range of parameter values is defined by boundary values depending on the driving simulator, the simulation tasks, and the drivers. Various values were introduced to this respect in the literature (Tab. 5.5). Thus, the boundary values were adapted in this thesis according to the priority levels defined above.

- The objective of the first criterion C_1^* is to avoid the violation of the physical limits of a driving simulator. A tuned parameter set must satisfy the criterion, otherwise it cannot be used. Additionally, the tilt boundary prevents recognizing the rotational tilt state by the driver. In this study, the limit values of the LMR driving simulator can be found in the report of Pham (2014).

- For the second criterion C_2^* the angular velocity threshold was set to $\omega_{th} = 6\,(deg/s)$ according to the research of Nesti et al. (2012), who studied the roll-rate threshold by using subjective experiments in the MPI CyberMotion simulator based on the KUKA Robotcoaster. Chapron and Colinot (2007) in the new SHERPA simulator and Zywiol and Romano (2003) used the same value for the tilt-rate limit, which seems to justify this choice. In addition, Chapron and Colinot (2007) introduced the angular acceleration threshold $\dot{\omega}_{th} = 0.2\,(rad/s)$ by (C_4 in Tab. 5.6).

  Furthermore, as the location of the frame $\mathcal{K}_{Ps}$ at the driver' vestibular system is in general unknown, a limit on the false specific force cue at the driver's head due to the eccentricity is defined by a factor k_{fT}. Let h be the distance between the real position of the driver's vestibular system and the assumed one in the MCAs. The the roll angular acceleration $\dot{\omega}_x = \dot{\omega}_{th} = 0.2\,(rad/s)$ generates the extra specific force

δ_O	ω_{th}	$\dot{\omega}_{th}$	$k_{S,min}$	$k_{S,max}$	k_a	k_{fT}	k_e
$0.17\,(m/s^2)$	$6\,(deg/s)$	$11\,(deg/s)$	0.36	1	0.50	$0.1\,(m/s^2)$	1

Table 5.9: Selected quantites

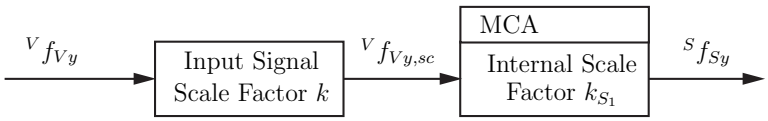


Figure 5.11: Global scale factor $k_S = k \cdot k_{S_1}$

which should be $f_T = h.\omega_x \leq \delta_O$ Pouliot et al. (1998), or as Zaichik et al. (2009) $f_T \leq 0.1\,m/s^2$ mentioned that. Assuming $h \in [0,\, 0.5]$ (m), $f_T = h \cdot \omega_x = [0,\, 0.1]$ (m/s^2), in this thesis the factor $k_{fT} = 0.1\,m/s^2$ was chosen which guarantees that the effect of the extra specific force can be negligible.

- For the third criterion C_3^*, factor $k_e = 1$ and the threshold value of the otolith organs $\delta_O = 0.17\,m/s^2$ were chosen. This selection is a little stricter than that mentioned in the study of Grant et al. (2003), which showed the classical washout filters with the best parameter set generate the maximum shape error of approximately $e_{f,sh} \approx 0.2\,m/s^2$.

- For the criterion C_4^*, the task is to define an acceptable range of scale signals by limiting the values $k_{S,min}$ and $k_{S,max}$ of the global scale factor k_S. In this thesis, the global scale factor is decomposed in two factors: a input signal scale factor k and an internal MCA scale factor k_{S1} (Fig. 5.11). The internal scale factor k_{S_1} is determined by the tuning procedure involving the possible simulation specific force during the complete simulation task as

$$k_{S_1} = \begin{cases} \frac{\int |f_{Sy}|dt}{\int |f_{Vy,sc}|dt} = \frac{\int |f_{Sy}|dt}{\int |k f_{Vy}|dt} & \text{with} \quad f_{Vy} \neq 0 \\ 0 & \text{with} \quad f_{Vy} = 0 \end{cases} \tag{5.15}$$

  An advantage of this procedure is that k_{S1} can describe better the final ratio of simulated signal to target signal for all MCAs, especially ZyRo, MPC* algorithms, and optimal washout filters.

  For the simulation task in this thesis (section 5.4.1), if $k = 0.4$, $\max(k\, f_{Vy}) \approx 1.03\,(m/s^2)$. For an acceptable shape error of $e_{f,sh}^* \leq \delta_O = 0.17\,(m/s^2)$, the factor $k_{S_1} \geq 1 - \frac{\delta_O}{k f_{Vy}} \Leftrightarrow k_{S_1} \geq 0.83$ is obtained, which in order to improve the tracking of simulated specific force was narrowed to $0.9 \leq k_{S_1} \leq 1$. Therefore, corresponding to the range of scale factor $k \in [0.4,\, 1]$ suggested by Berthoz et al. (2013), the range of k_S in this thesis was set to $k_S \in [k_{S,min},\, k_{S,max}] = [0.36,\, 1]$.

- For the fifth criterion C_5^*, Stratulat et al. (2011) found the best ratio to be 35/65%. In this thesis, the proposed minimum ratio $k_{a,min} = 50\%$ is selected so that the translational motions are used more than rotational motions.

5.3.4 The well-tuned index MF holds for a single parameter

In oder to identify the fulfillment of the combined criteria (Tab. 5.8), four binary digits G_j, $j \in \{1 \cdots 4\}$ are defined as

$$G_1 = \overline{C_1^*}\,; \quad G_2 = \overline{C_2^*} \wedge \overline{C_3^*}\,; \quad G_3 = \overline{C_4^*}\,; \quad G_4 = \overline{C_5^*}\,, \tag{5.16}$$

where

- $\overline{C_i^*} = 1$, if the criterion C_i^* is satisfied, otherwise $\overline{C_i^*} = 0$
- G_1: means no violation of the limits and protecting the tilt-coordination
- G_2: means avoiding the false cues
- G_3: means avoiding too weak cues
- G_4: means using as much translational movement as possible to simulate lateral acceleration; $\wedge$ is the "and" logic operator

Note that the combination of the criteria C_2^* and C_3^* can be described as a trade-off between false cues of specific force and rotational motions. However, according to Grant and Reid (1997a), all types of false cues seriously degrades the motion fidelity. Therefore, G_2 includes both criteria C_2^* and C_3^*.

For numerically assessing how well the parameter is tuned, the well–tuned index is proposed such as the binary number

$$MF = 2^3 \cdot G_1 + 2^2 \cdot G_2 + 2^1 \cdot G_3 + 2^0 \cdot G_4\,. \tag{5.17}$$

By using a binary number, the MF index can describe the priority order of tuning rules (Tab. 5.10), and the level of motion fidelity (Tab. 5.1). Concretely, the well-tuned index MF is categorized into 4 levels as shown in Tab. 5.10. A high MF implies that a MCA (with tuned parameters) can produce implementable motions that have no predicted false cues and uses as much as the translational motion as possible. A medium MF has a larger scale error than the high MF, which could produce weak motion perception. The last level, a low MF, is attributed to any case that produces any types of false cues.

High	Medium	Low	Not Satisfied
$MF_1 = 14, 15$	$MF_2 = 12, 13$	$MF_3 = 8 \cdots 11$	$MF_4 < 8$

Table 5.10: Level of well–tuned index MF

5.4 Tuning MCAs by trial and error

5.4.1 Description of the simulation task

In the present thesis, the simulation consists of a roller coaster following a planar S-curve trajectory at a constant velocity $v = 3.6\,\mathrm{m/s}$ as shown in Fig. 5.12a. The trajectory is composed of an entering straight rail $l_1 = 43.2\,(m)$, a S-curve rail, and an exiting straight rail $l_2 = 45\,(m)$. The roller coaster completes the S-curve in $20.125\,(s)$. Due to the constant velocity, there is only relative lateral acceleration $a_{Vy}(t) = a_R(t) = \frac{v^2}{C_S(t)}$, with $C_S(t)$ representing the curvature of the S-curve at a given time t (see Fig. 5.12b). The decomposition of constant velocity v in the X- and Y-axes is shown in Fig. 5.12c and 5.12d, respectively. The mathematical formulation of the S-curve rail is included in appendix A.4. For the simulation, the target yaw motion was assumed to be simulated by the lateral motion and visual perception (Schroeder (1999)) and was not considered here as an explicit robot signal.

The tuning process (Fig. 5.13a) aims to find a possible parameter set for a given MCA. Initially, the scale factor $k = k_{in} \in \{k_{min}, k_{max}\}$ and parameters from literature are used for the MCA to generate the simulated data. Then, the simulated data are examined by the criteria (orange block in Fig. 5.13b). The block checks the simulated signal with the proposed criteria C_i^* with $i \in \{1 \cdots 5\}$ and generates the well-tuned index MF. For $MF \in \{14, 15\}$ (being equivalent to the highest level) the current

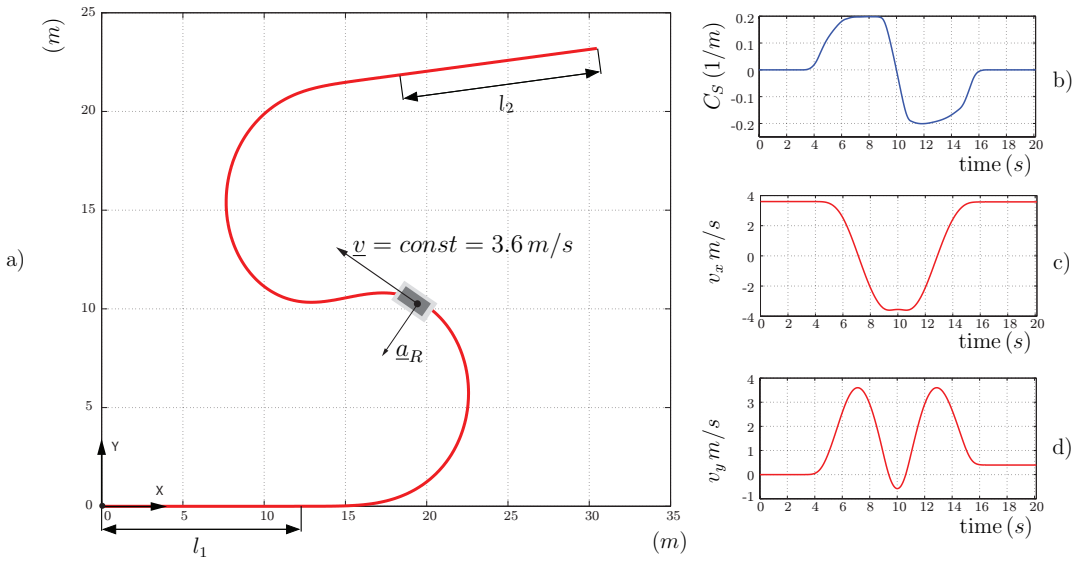


Figure 5.12: a) XY-coordinates of the S-curve; b) Curvature C_S of the S-curve trajectory; c) X-axis velocity v_x component of the simulated rollercoaster; d) Y-axis velocity v_y component of the simulated roller coaster

scale factor k_{Tn} and other parameters are saved as the set of parameters used for the MCA. In contrast, if $MF \leq 7$ (not satisfied), the scale factor is first reduced by δk. If the $k = k_{min}$, the parameters are changed to improve MF as high as possible.

For $MF \in \{8, \cdots, 11\}$ (low level), the scale factor is also first reduced until $k = k_{min}$, then the other parameters are changed to achieve a high level of MF. In this situation, if MF cannot reach the high level, the scale factor is continuously reduced to obtain the medium level of MF. Since the false cues take higher priority over the scale error, the scale factor k can be reduced to get a higher value of MF with $k < k_{min}$.

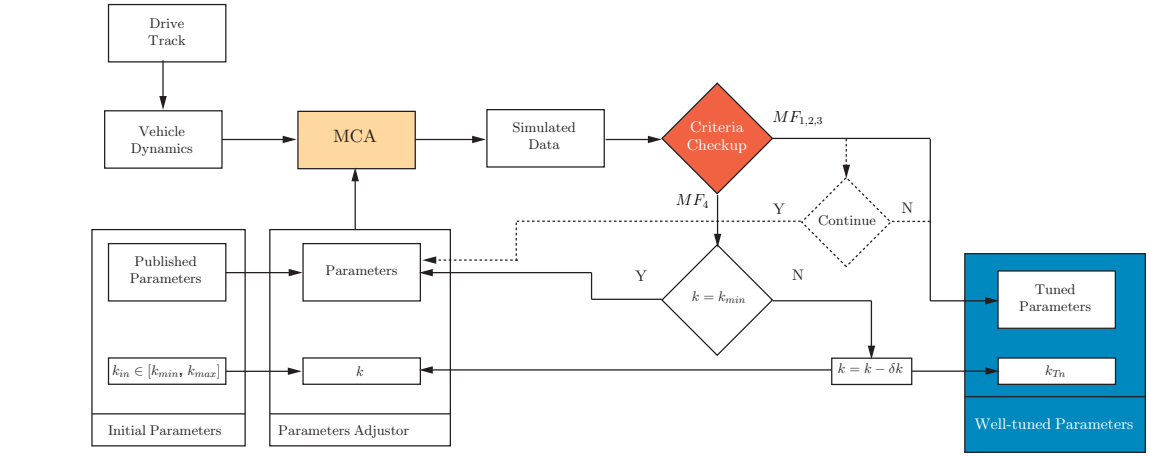
For $MF \in \{12, 13\}$ and scale factor $k \leq k_{min}$, the parameters can finely be adjusted to increase a internal scale factor k_{S1}. By doing that, the values of MF could be increased.

The parameters of the MCAs were manually tuned by trial and error. In the case of the CLRN and CLG algorithms, the break frequency and damping factor were adjusted, while for the other MCAs the weighting parameters in their cost functions were suitably changed. Since there are no general rules that apply to all MCAs, the tuning process is time-consuming and different for each MCA. Furthermore, it depends significantly on the designer's experience. The dashed line in Fig. 5.13a represents the designer's decision to continue or halt the tuning process when the MF reaches a satisfying level.

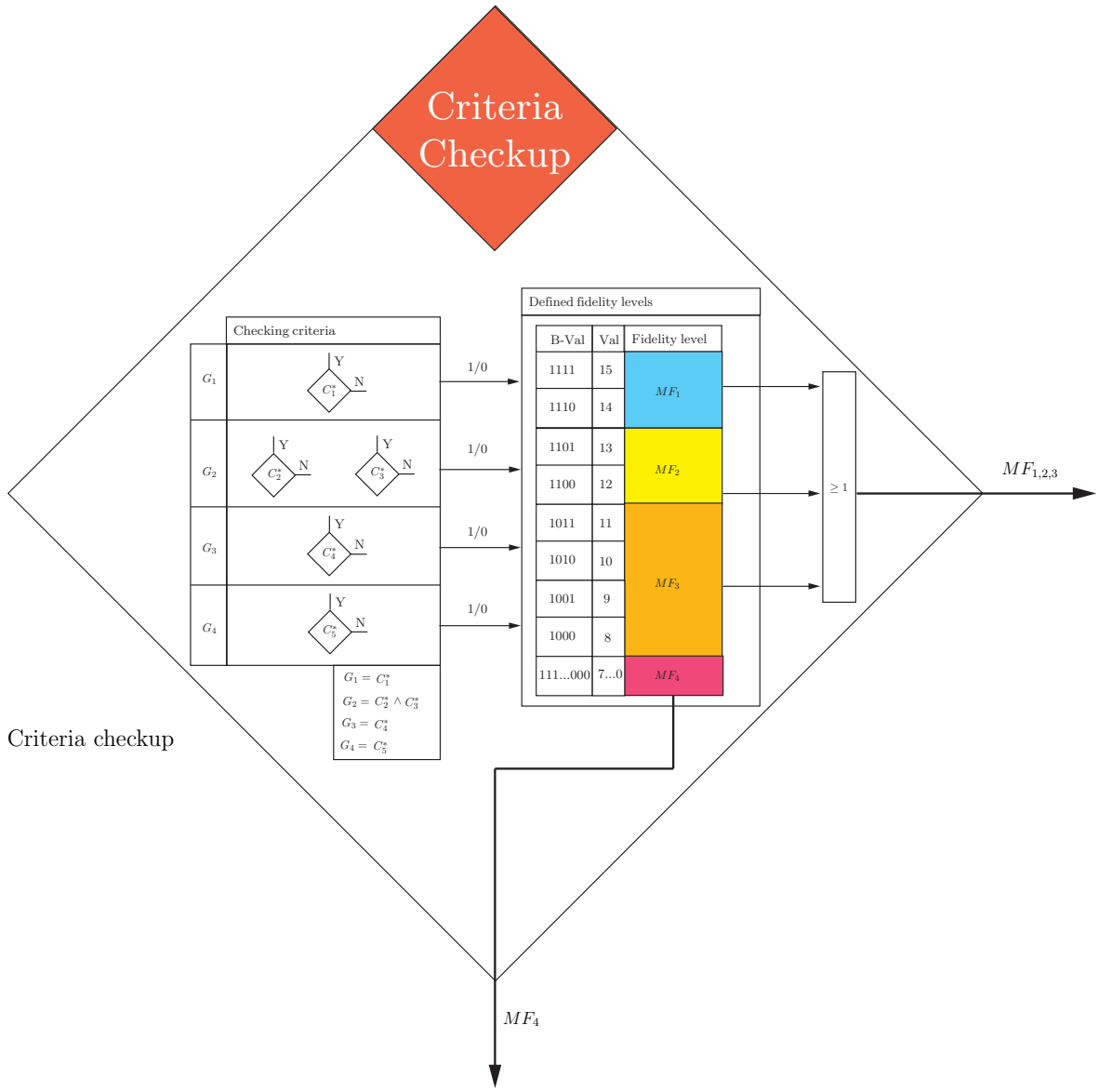
Given the fact that the scale factor can vary in the range $[k_{min}, k_{max}]$, in this thesis, the smallest scale factor k_{min} was selected. All MCAs were tuned in order to obtain a value of MF as high as possible. The parameters for all MCAs presented in the following section show that almost all MCAs can be tuned to reach the best MF in the simulation test.

5.4.2 Tuning of MCAs parameters by trial and error

In a first approach, the MCA parameters were manually tuned by trial and error according to the procedure in the literature. Hereby, the washout effect of a MCA was considered only for the simulated tilt angle, so that the seat of the LMR driving simulator had no remnant tilt angle at the end of the simulation. The process of manual tuning is shown in Fig. 5.14. Here, the user evaluates the criteria and changes by trial and error the MCA parameters until an acceptable well-tuned index MF.



a) Tuning procedure



b) Criteria checkup

Figure 5.13: Tuning procedure with proposed criteria

For the **CLRN** algorithm, the natural break frequency of high-pass filters is firstly chosen to exploit the available workspace and to make the simulated acceleration a_{Sy} approximately zero at the end. Then, the natural break frequency of the low-pass filter is selected to generate a suitable angular velocity. The fine-tuning process involved changing the damping factor and/or the scale factor to obtain the highest MF.

The **CLG** algorithm requires the tuning of 3 high-pass filters. In order to eliminate the effect of the spurious components, the break frequency and damping factor of the R- and Z-channel were chosen to keep radius the R and the Z-coordinate almost constant. The natural break frequency of the α-channel was then tuned to eliminate the false cues of the angular velocity and angular acceleration. Finally, the washout break frequency $\omega_{b\alpha}$ and the damping factor ζ_α were modified in the fine–tuning step to obtain the highest MF and to pull the tilt angle to zero. A reduction of the scale factor was selected if altering the break frequencies and the damping factor could not remove all false cues.

For the **ADRN** algorithm, the tuning consisted in first selecting constant values k_1 and k_2 according to the damping factor and natural break-frequency of a second high-pass washout filter of a translational channel (as in the classical algorithm). Then, the initially adaptive factors $\{p_{10}, p_{20}, p_{30}\}$ and weighting parameters $w_{1\cdots8}$ were tuned. Note that, for the simulation task, the restriction of the angular velocity and the roll angle could be ignored by setting the weighting factors to zero, since the false cues of the angular velocity are controlled by a nonlinear module in the adaptive process.

In the case of the **ADSK** algorithm, all the parameters of a second high-pass filter including the scale factor, the damping factor, and the natural break frequency are

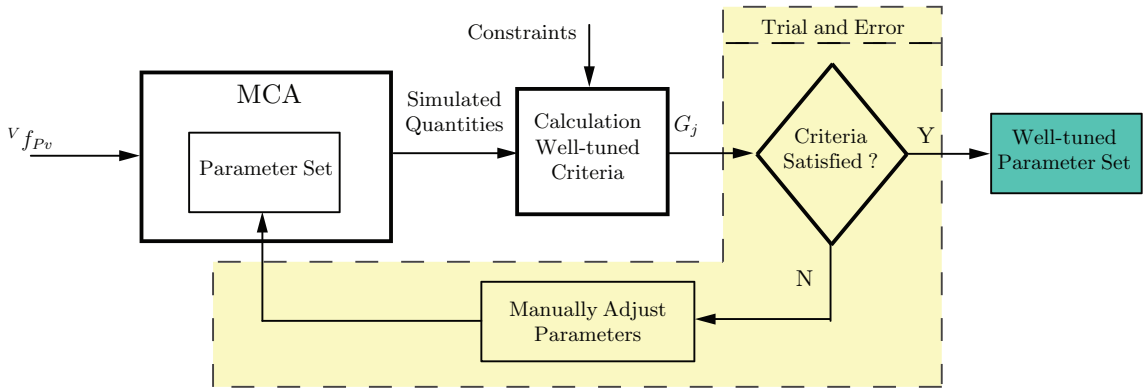


Figure 5.14: Concept of the manual trial and error tuning method for a MCA

adapted in real-time. For the studied simulation, only the parameters of the washout filter of the α-channel were adapted. The other channels were fixed as for the CLG algorithm. The natural break-frequency was restricted because the simulated signals were more influenced by the natural break-frequency than by the other parameters. Additionally, the initial values of the adaptive parameters were selected to reduce the oscillation appearing in simulated signals due to a large change of the parameters at a time. The restriction of the change of a parameter was implemented by selecting the corresponding weighting parameter $w_m\,, m \in \{k\,, \zeta\,, \omega\}$ according to its significant level of effect. Similarly, w_a restricts the difference between the simulated and the desired acceleration. In order to pull the simulated position to zero at the end of a simulation (implementing washout effect), the weighting value of w_S was increased.

For the **optimal washout filters**, the tuning consisted in choosing appropriate values for the weighting parameters. For example, increasing the value of r_1 or q_1 (OpTYM, OpT, OPRN, OpTNon) reduces the simulated angular velocity, while increasing r_2 or r_{c3} (OpTYM, OpT, OpTNon) reduced the simulated acceleration, and the weighting factor q_2 (OpT, OpRN, OpTYM, OpTNon) had an effect on both the simulated specific force and the simulated acceleration. From the tuning experience, we found that increasing r_{c1} pulls the simulated acceleration to zero at the end of the simulation, and r_{c2} affects the shape of the simulated acceleration.

The **ZyRo** approach was relatively easy to tune because it has only 4 weighting parameters that directly influence simulated quantities. For example, to increase the simulated acceleration or displacement, the parameter q_2 can be reduced. In addition, increasing q_1 makes the simulated specific force track better the target one, and increasing r_2 restricts indirectly the values of the angular velocity. By a proper selection of the break frequencies λ and c, the oscillation of the first derivative of the simulated specific force and angular velocity could be reduced. For instance, $c = 1.25$ (Tab. B.8) leads to similar changes for angular acceleration and angular velocity.

Also, the **MPC*** algorithm was simple to tune because the weighting parameters are directly correlated with the simulated quantities. As in the ZyRo algorithm tuning, increasing q_1 makes the simulated specific force track better the target one, and increasing q_4 pulls the simulated angle back to zero. In addition, increasing s_2 or q_3 reduces the angular velocity, and increasing the prediction horizon p and the control horizon m increases the tracking quality of simulated specific forces, but comes at the cost of increasing the computation time.

For the tunning of the **exMPCFK** algorithm, a similar procedure as for the MPC*

algorithm could be used. The selection of the parameters for the brake law followed the guidelines mentioned by Fang and Kemeny (2012). Unlike, the ZyRo and MPC* algorithms, the constrained conditions used in the exMPCFK algorithm remove the false cues of the rotational motion. Therefore, the tuning process focuses on making the simulated specific force track the target one, and q_{y3} is much larger than the other parameters. Additionally, the simulated acceleration could be pulled to zero at the end of the simulation with a suitable value of r_1.

5.4.3 MCAs response comparison for different input signal scale factors

This section compares the performance of the different MCA for the target ride under study in this thesis. As is known, the response of the different MCA approaches depends strongly on the dynamics of the target motion. For this reason, a systematic study was carried out by choosing seven signal scale factors $k \in \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ and manually tuning the parameters of each MCA for the smallest value of k so that the MCA follows as good as possible the target motion in terms of the well-tuned index MF. We discuss here in detail three scale factors $k \in \{0.4, 0.5, 0.7\}$ and give an overview of the well-tuned index for all scale factors. The detail of all scale factor is given in the Appendix.

Scale factor $k = 0.4$

When the input signal scale factor is chosen at its lower limit, all MCAs behave similarly. This is shown in Fig. 5.15 for the time histories and in Fig. 5.16 for the Pouliot' "performance indicator" along with the Fischer's "good criterion" and "normalized good criterion". The time histories include also simulated pitch angular velocities (Fig. 5.15c) needed to control the pitch tilt angles for compensating the spurious centripetal accelerations arising in the rotational trajectory of the LMR KUKA Robocoaster simulating the transversal accelerations. Furthermore, the time histories of simulated yaw angular velocities of the rotational trajectory are presented in Fig. 5.15e. Besides, Fig. 5.17 shows the partial components of Fischer' "good criterion".

Fig. 5.15a shows that the simulated specific forces generated by all MCAs are similar and track well the scaled input specific forces. Furthermore, the angular velocities of both roll (Fig. 5.15b) and pitch (Fig. 5.15c) remain within the threshold limits described by the dashed black line. Note that the simulated yaw angular velocities have a large deviation from the target yaw angular velocity (Fig. 5.15e), however, the errors were assumed to be negligible according to the literature (Schroeder (1999)). Future research should verify if this is true in the present setting.

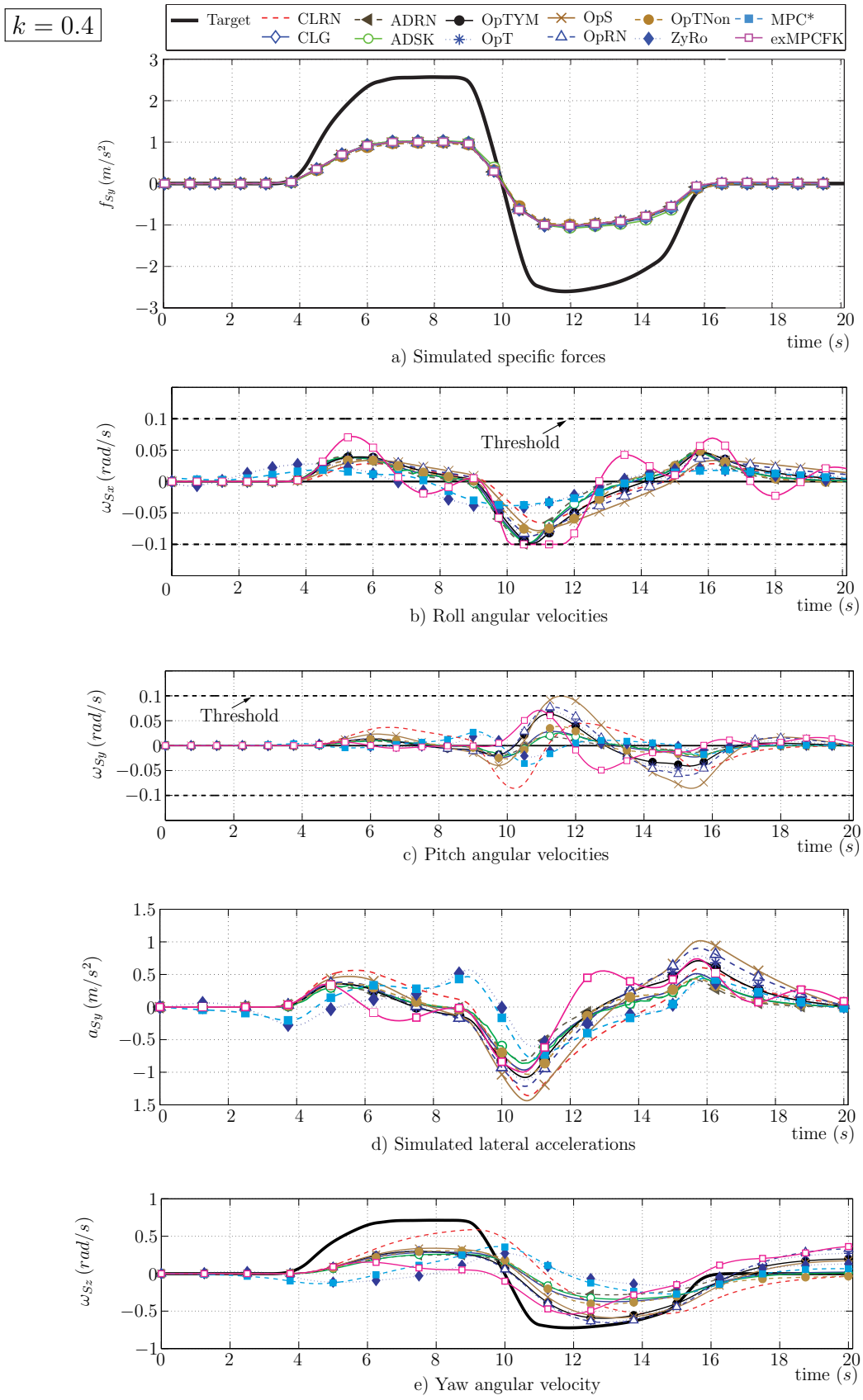
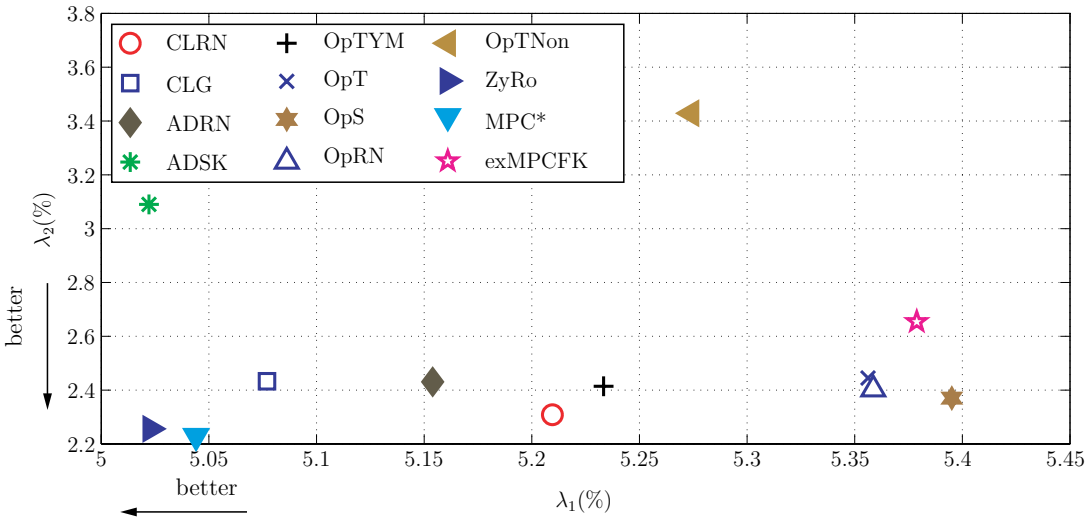
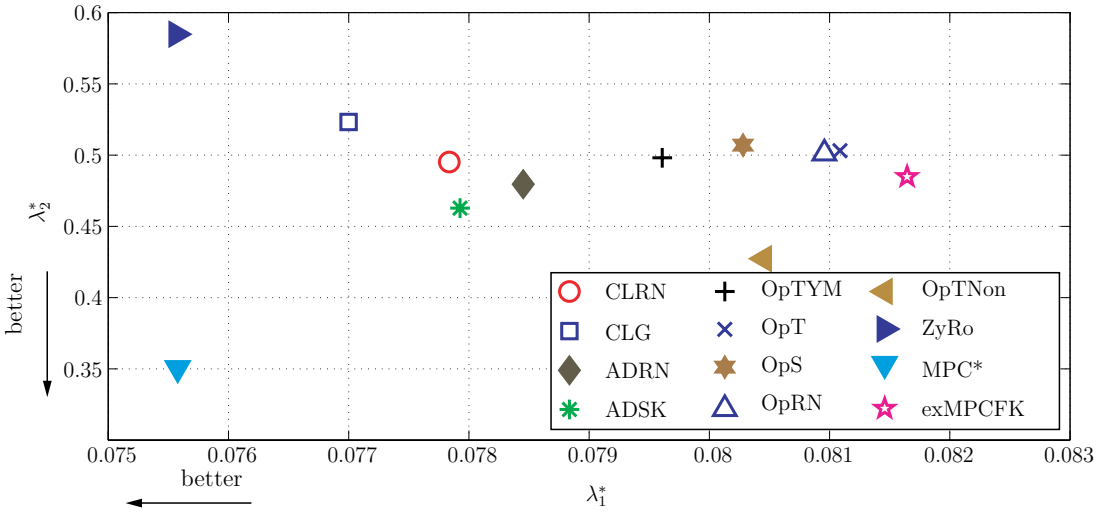


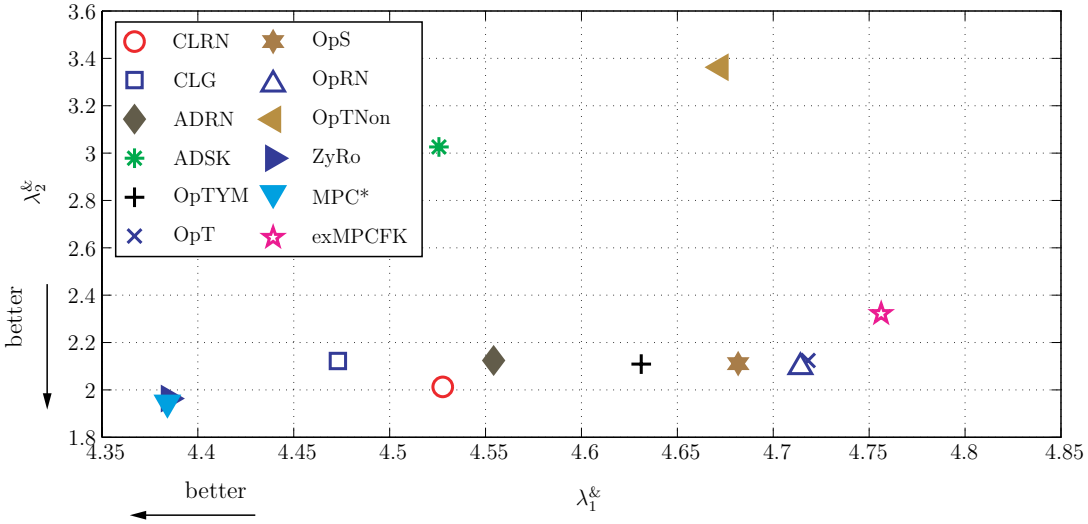
Figure 5.15: Comparison responses of all MCAs for $k = 0.4$



a) Pouliot's performance indicators λ_1, λ_2 for $k = 0.4$

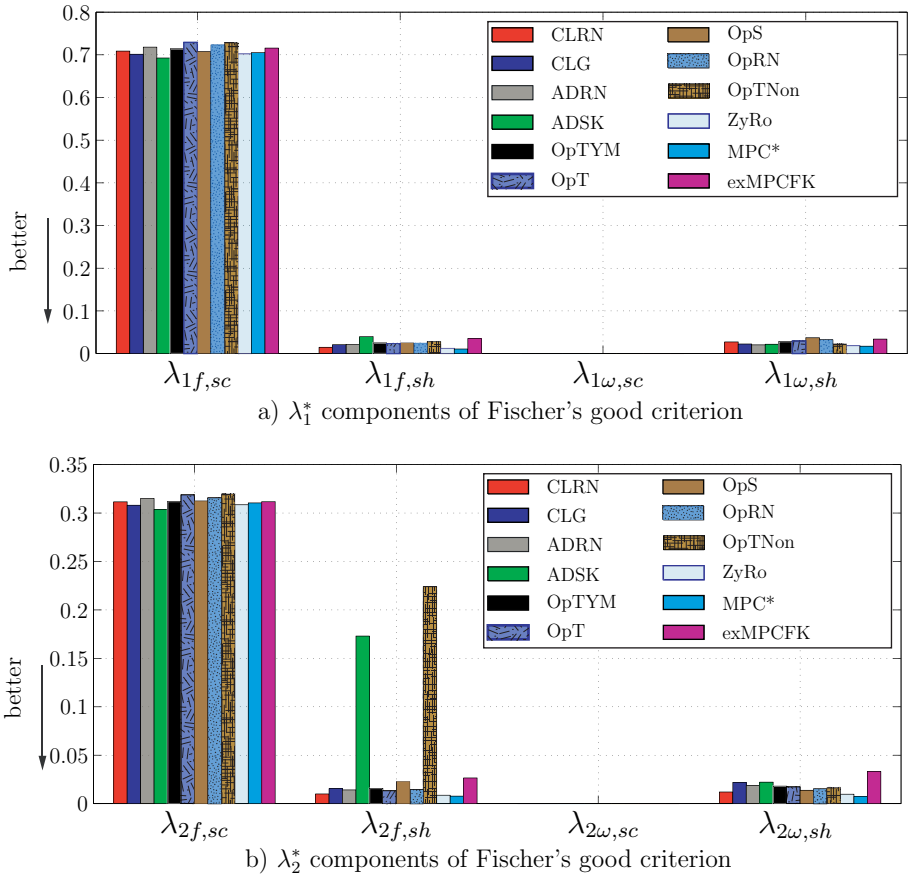


b) Fischer's good criterion λ_1^*, λ_2^* for $k = 0.4$



c) Normalized good criterion $\lambda_1^\&$, $\lambda_2^\&$ for $k = 0.4$

Figure 5.16: Comparison of numerical performance measures for $k = 0.4$



a) λ_1^* components of Fischer's good criterion

b) λ_2^* components of Fischer's good criterion

Figure 5.17: Comparison components of Fischer's good criterion for $k = 0.4$

With respect to the proposed well-tuned index, all MCAs present a high index MF (Tab. 5.11 on page 104). Most MCAs generate similar simulated lateral accelerations and angular velocities (5.15b and Fig. 5.15d) except the ZyRo, MPC* algorithms which generate simulated lateral accelerations before the time instant at which the target signal changes. These pre-movements are aimed at preparing the workspace for the quick change of the specific forces. For the ZyRo and MPC* algorithms, the simulated roll angular velocity reaches only half of the threshold.

The average errors expressed in the "good criterion" show that the ZyRo algorithm presents the largest λ_2^* and the smallest λ_1^*, while the OpTNon and ADSK algorithms have the second and the third lowest of the first derivative. However, after the normalizing the average sub-errors by the threshold values of the vestibular system (Grant et al. (2003)), the ZyRo algorithm generates the minimum values of the goodness criterion, as shown by the "normalized good criteria" $\lambda_1^{\&}$ and $\lambda_2^{\&}$, while the worst $\lambda_2^{\&}$ belongs to the ADSK and OpTNon algorithms. The reason is that the algorithms (ADSK and OpTNon) present the largest value of error in shape errors of the first derivative of the simulated specific forces (5.17b). Note that the average errors expressed in the

"performance indicator" have relatively similar distribution to those expressed in the "normalized good criterion".

The components of "good criterion" (Fischer (2009)) are shown in Fig. 5.17a and 5.17b on page 96. The ZyRo and MPC* algorithms have the smallest error in most sub-indices, while the exMPCFK algorithm has similar sub-indices with optimal washout filters but it has the largest angular acceleration error $\lambda^*_{2\omega,sh}$.

In conclusion, although all MCAs were tuned to have a high index MF, the ZyRo and MPC* algorithms produce the smallest average errors. The online adaptive ADSK and OpTNon algorithms can produce oscillations that result in high errors of the first derivative. The exMPCFK algorithm has the potential to reduce the error by increasing the prediction horizon but computing the explicit controller is very time-consuming. Overall, for this input signal with a scale factor of $k = 0.4$, the type of MCA does not have a significant effect on the level of the well-tuned index MF (Tab. 5.11), which is in conformity with the research of Nahon and Reid (1990).

Scale factor $k = 0.5$

When the input signal scale factor is 0.5, the responses of MCAs are apparently different. This is shown in Fig. 5.18 for the time histories and in Fig. 5.19 for the Pouliot' "performance indicator" along with the Fischer's "good criterion" and "normalized good criterion". The time histories include also simulated pitch (Fig. 5.18c) and yaw angular velocities (Fig. 5.18e). Furthermore, Fig. 5.20 shows the components of Fischer' "good criterion".

Fig. 5.18a shows that the simulated specific forces generated by all MCAs are similar and track well the scaled input specific forces. Only the ADSK algorithm generates a small overshoot. However, there are only the ADRN, MPC* and ZyRo algorithms generating both roll (Fig. 5.18b) and pitch (Fig. 5.18c) angular velocities within the threshold limits (dashed black line). Therefore, regarding the proposed well-tuned index, the three MCAs present a high index MF (Tab. 5.11 on page 104). Moreover, the CLG, ADSK and OpTNon algorithms present a low index MF, while the rest of MCAs generate a "not satisfied" index MF since these MCAs generates the simulated lateral accelerations (Fig. 5.18d) leading incompatible rotational motions of the LMR KUKA Robocoaster.

With respect to yaw angular velocity, the CLG, OpRN and exMPCFK algorithms present large amplitudes which are even over the peak amplitudes of target yaw angular velocity. However, the effects of the errors were assumed to be negligible.

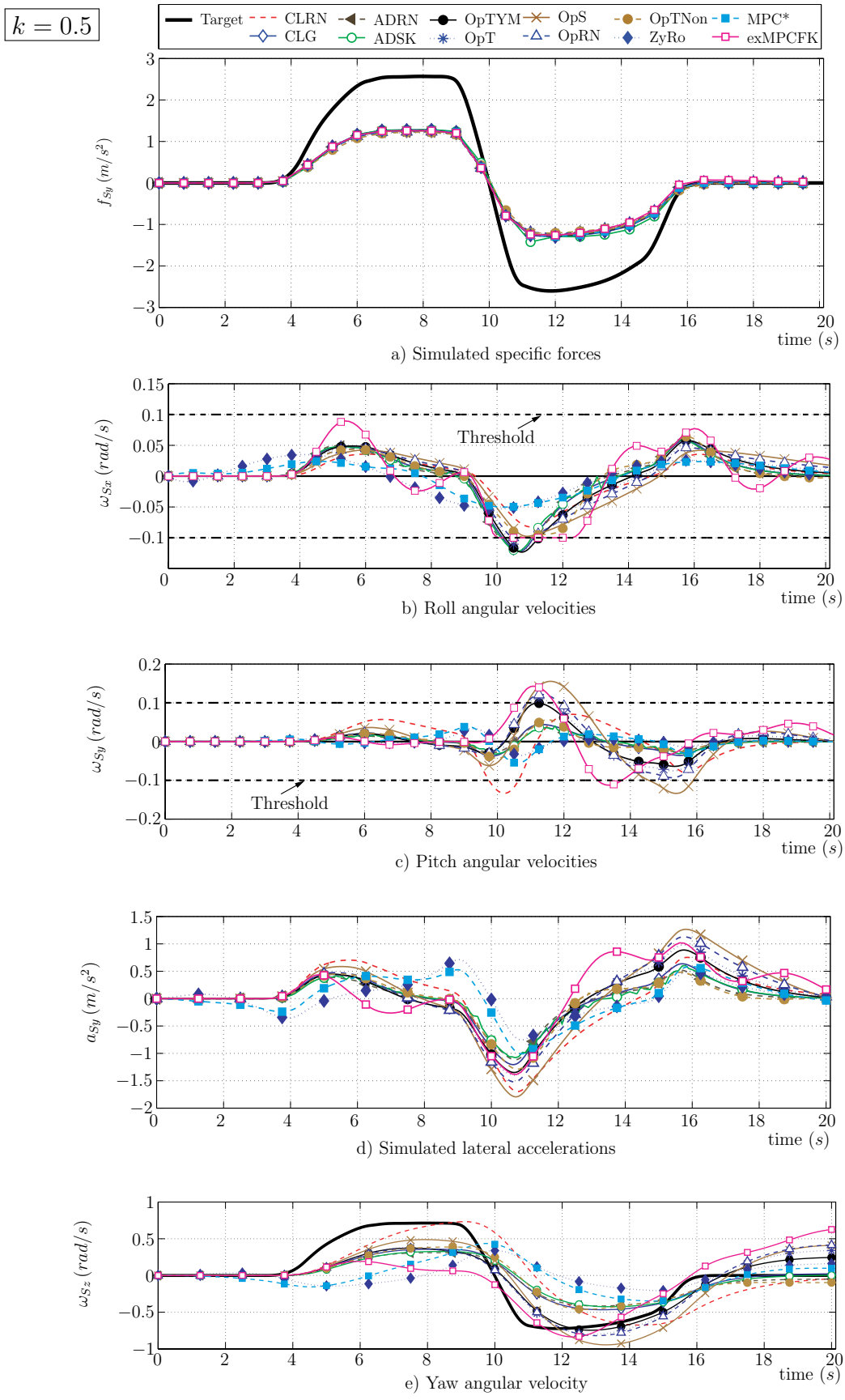
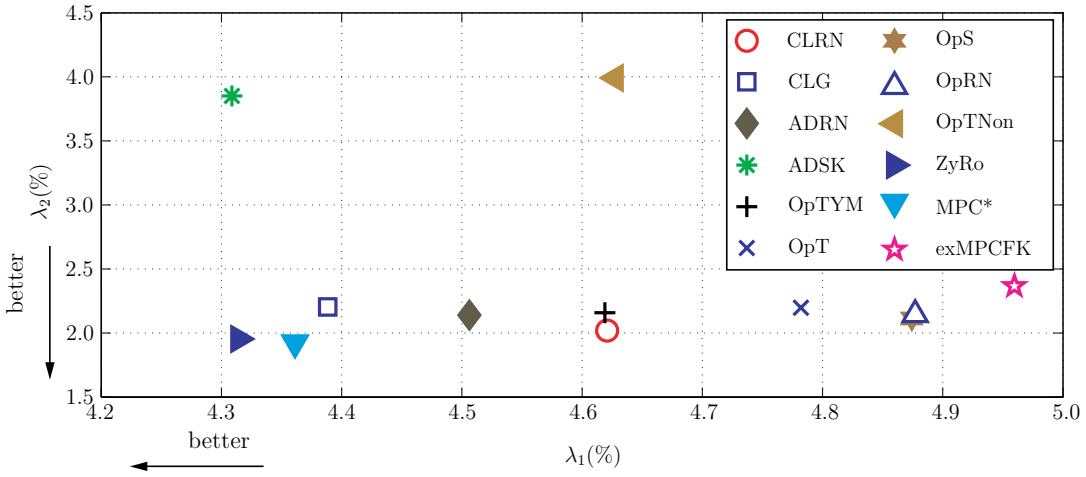
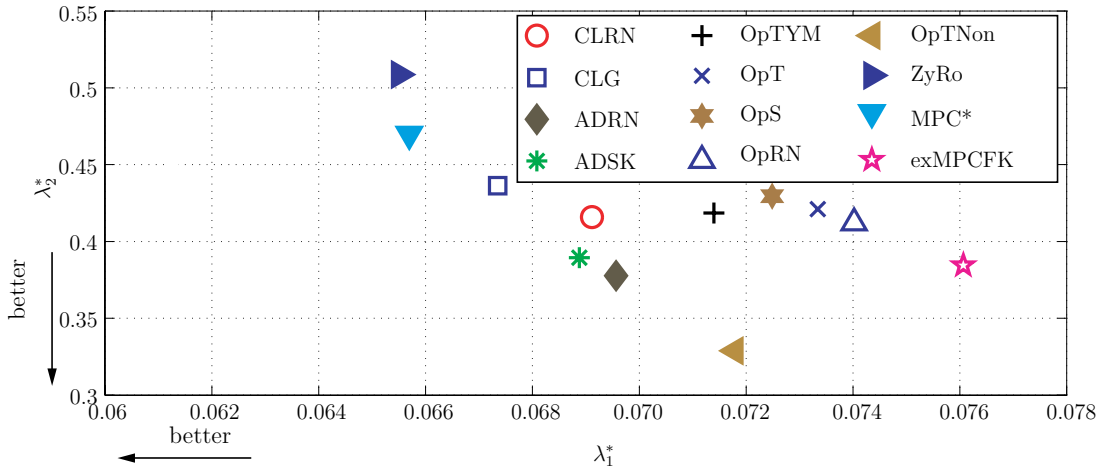


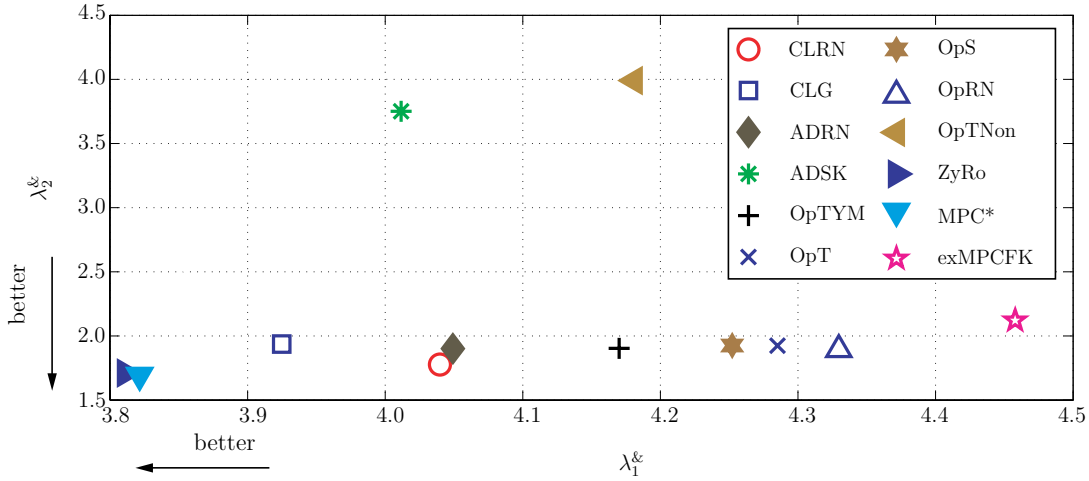
Figure 5.18: Comparison responses of all MCAs for $k = 0.5$



a) Pouliot's indicator λ_1, λ_2 for $k = 0.5$

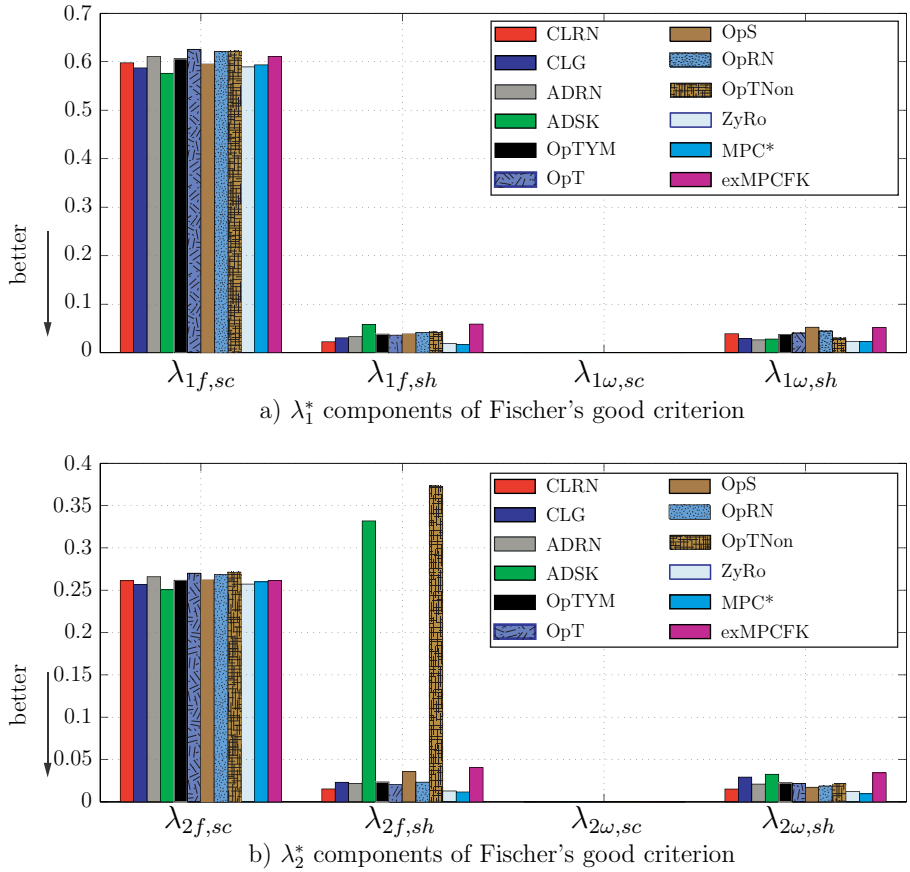


b) Fischer's good criterion λ_1^*, λ_2^* for $k = 0.5$



c) Normalized good criterion $\lambda_1^\&$, $\lambda_2^\&$ for $k = 0.5$

Figure 5.19: Comparison of numerical performance measures for $k = 0.5$



a) λ_1^* components of Fischer's good criterion

b) λ_2^* components of Fischer's good criterion

Figure 5.20: Comparison components of Fischer's good criteria for $k = 0.5$

The average errors expressed in the "good criterion" show that the ZyRo and MPC* algorithms present the smallest λ_1^* and the largest λ_2^*, while the OpTNon and ADSK algorithms have the first and the fourth lowest λ_2^*. However, concerning the "normalized good criteria" $\lambda_1^{\&}$ and $\lambda_2^{\&}$, the ZyRo algorithm generates the minimum values of the goodness criterion, while the worst $\lambda_2^{\&}$ belongs to the ADSK and OpTNon algorithms. Moreover, the exMPCFK algorithm presents the largest $\lambda_1^{\&}$. Fig. 5.20a and 5.20b show in detail the sub-indices of the average errors. The ZyRo and MPC* algorithms have the smallest error in most sub-indices, while the algorithms (ADSK and OpTNon) present the largest shape errors of the first derivative of the simulated specific forces. The exMPCFK algorithm has the worst shape error for the specific force $\lambda_{1f,sh}^*$ and the angular acceleration $\lambda_{2\omega,sh}^*$. In conclusion, for this input signal with a scale factor of $k = 0.5$, the type of MCA has a strong effect on the level of the well-tuned index MF (Tab. 5.11). Concerning simulated specific force, the average scale errors decreases, but the shape errors increase.

Scale factor $k = 0.7$

When the input signal scale factor is 0.7, the responses of MCAs are strongly different. This is shown in Fig. 5.21 for the time histories and in Fig. 5.22 for the Pouliot'

performance indicator along with the Fischer's "good criterion" and "normalized good criterion". The time histories include also simulated pitch (Fig. 5.21c) and yaw angular velocities (Fig. 5.21e). Furthermore, Fig. 5.23 shows the components of Fischer' "good criterion".

Fig. 5.21a shows that the simulated specific forces generated by eleven MCAs are similar and track well the scaled input specific forces. Only the exMPCFK algorithm generates large distortions of simulated specific force due to the limited roll angular velocity. Further, only the ZyRo algorithm generates both roll (Fig. 5.21b) and pitch (Fig. 5.21c) angular velocities within the threshold limits (dashed black line). Therefore, regarding the proposed well-tuned index, only the ZyRo algorithm presents a high index MF (Tab. 5.11 on page 104). Moreover, the MPC* algorithm presents a low index MF, while the rest of MCAs generate a "not satisfied" index MF. Note that all simulated yaw angular velocity have larger amplitude than the target yaw angular velocities (Fig. 5.21e), however, these errors were assumed to be negligible (see above).

The average errors expressed in the "normalized good criterion" show that the ZyRo and MPC* algorithms present the smallest $\lambda_1^{\&}$ and $\lambda_2^{\&}$, while the worst $\lambda_2^{\&}$ belongs to the ADSK and OpTNon algorithms. Moreover, the exMPCFK algorithm presents the largest $\lambda_1^{\&}$. Fig. 5.23a and 5.23b show in detail the sub-indices of the average errors. The ZyRo and MPC* algorithms have the smallest error in most sub-indices, while the algorithms ADSK and OpTNon algorithms present the largest shape errors of the first derivative of the simulated specific forces. The exMPCFK algorithm has worst form error of the specific force $\lambda_{1f,sh}^{*}$ and angular acceleration error $\lambda_{2\omega,sh}^{*}$.

In conclusion, for this input signal with a scale factor of $k = 0.7$, the type of MCA has a significant effect on the level of the well-tuned index MF (Tab. 5.11). Due to the large lateral simulated accelerations, the simulated motions were not within the workspace and the pitch angular velocities of eleventh MCAs exceed the threshold level.

In summary, when the scale factor is increased, false cues of the angular velocity exist in all MCAs, such as the ZyRo with $k \geq 0.9$, the MPC* with $k \geq 0.7$, and the ADSK with $k \geq 0.5$. However, only the ADSK and OpTNon algorithms generate false cues of the angular acceleration. Regarding the well-tuned index MF, six MCAs present "not satisfied" indices for $k \geq 0.5$. The ZyRo algorithm presents a high index MF for only $k \in \{0.4, \cdots, 0.8\}$, while MPC* and ADRN algorithms present high indices for $k \in \{0.4, 0.5, 0.6\}$. The CLG, ADSK and OpTNon algorithms present low indices for $k \geq 0.5$. In addition, for $k = 1.0$, only the ZyRo algorithm generates implementable motions in LMR KUKA Robocoaster.

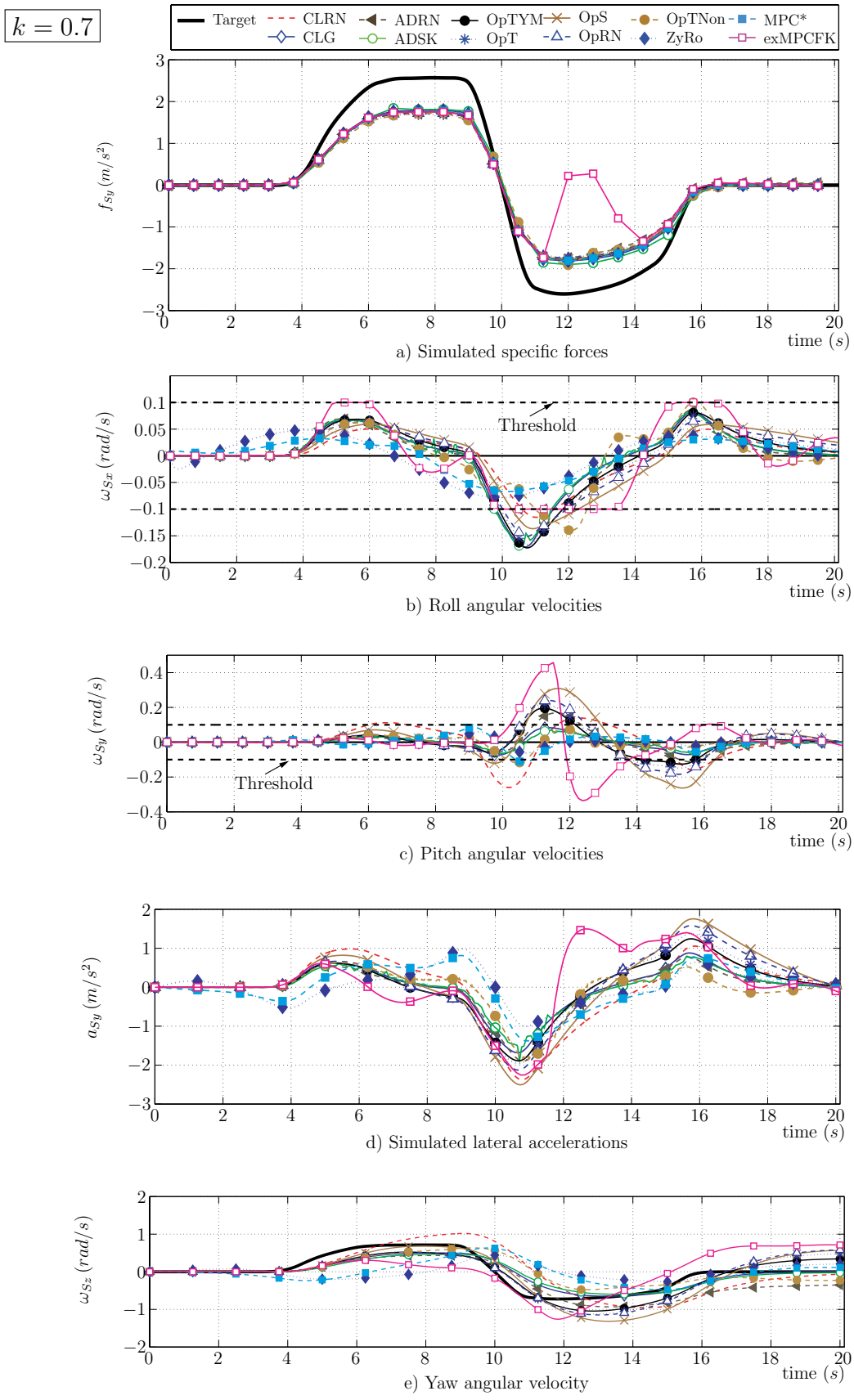
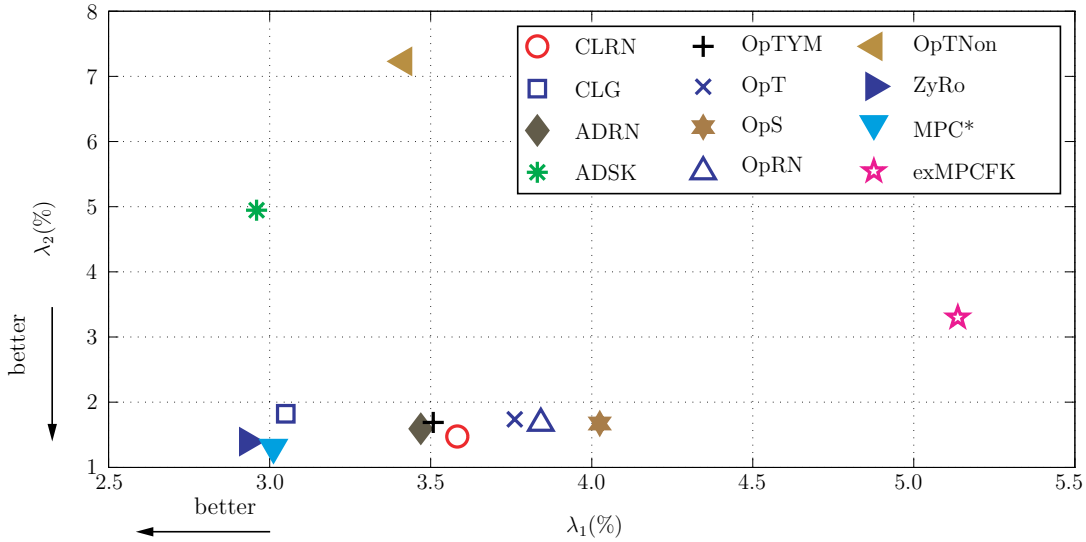
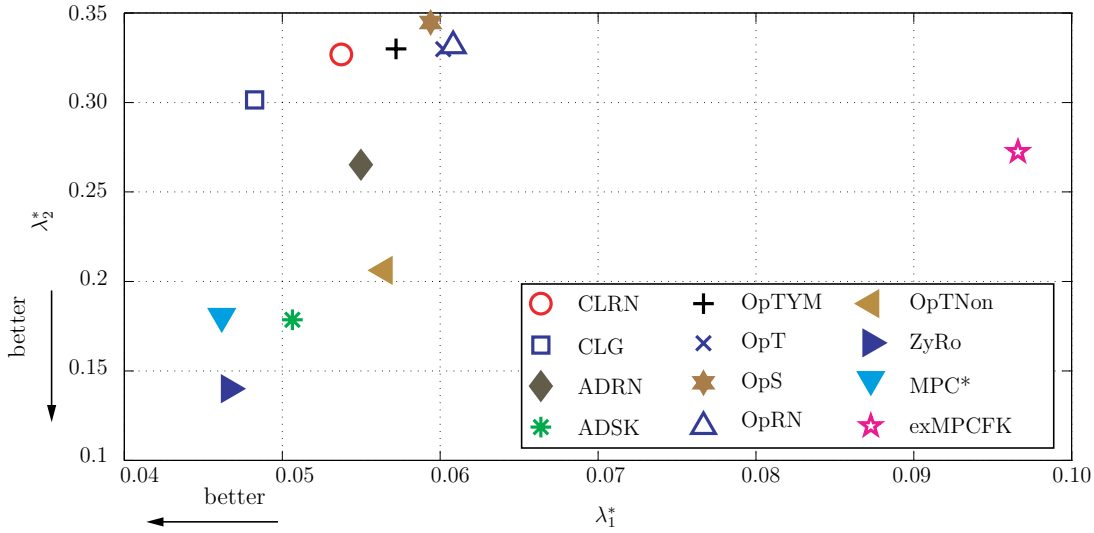


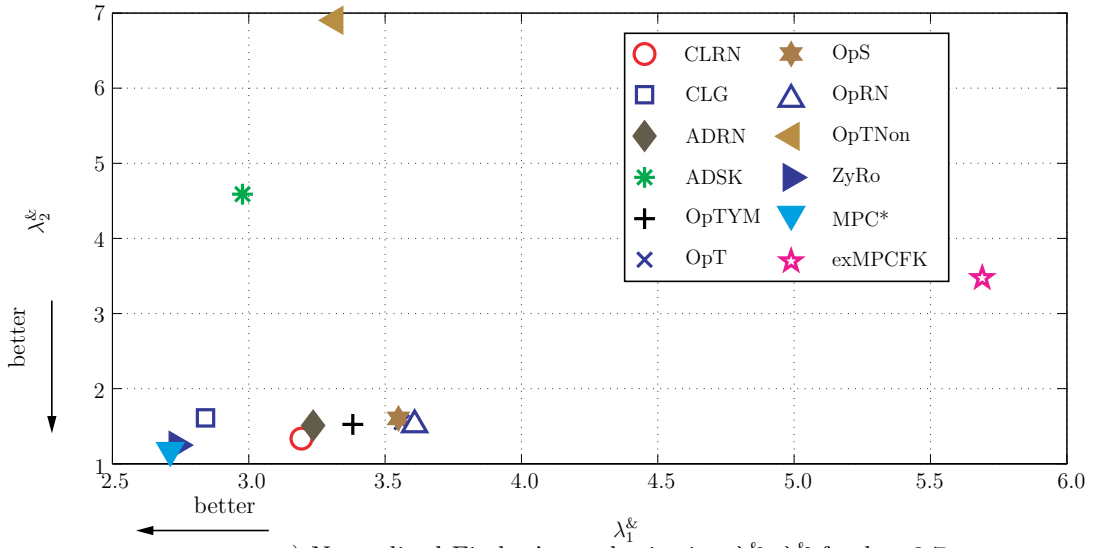
Figure 5.21: Comparison responses of all MCAs with scale factor $k = 0.7$



a) Pouliot's performance indicator λ_1, λ_2 for $k = 0.7$



b) Fischer's good criterion λ_1^*, λ_2^* for $k = 0.7$



c) Normalized Fischer's good criterion $\lambda_1^\&$, $\lambda_2^\&$ for $k = 0.7$

Figure 5.22: Comparison of numerical performance measures for $k = 0.7$

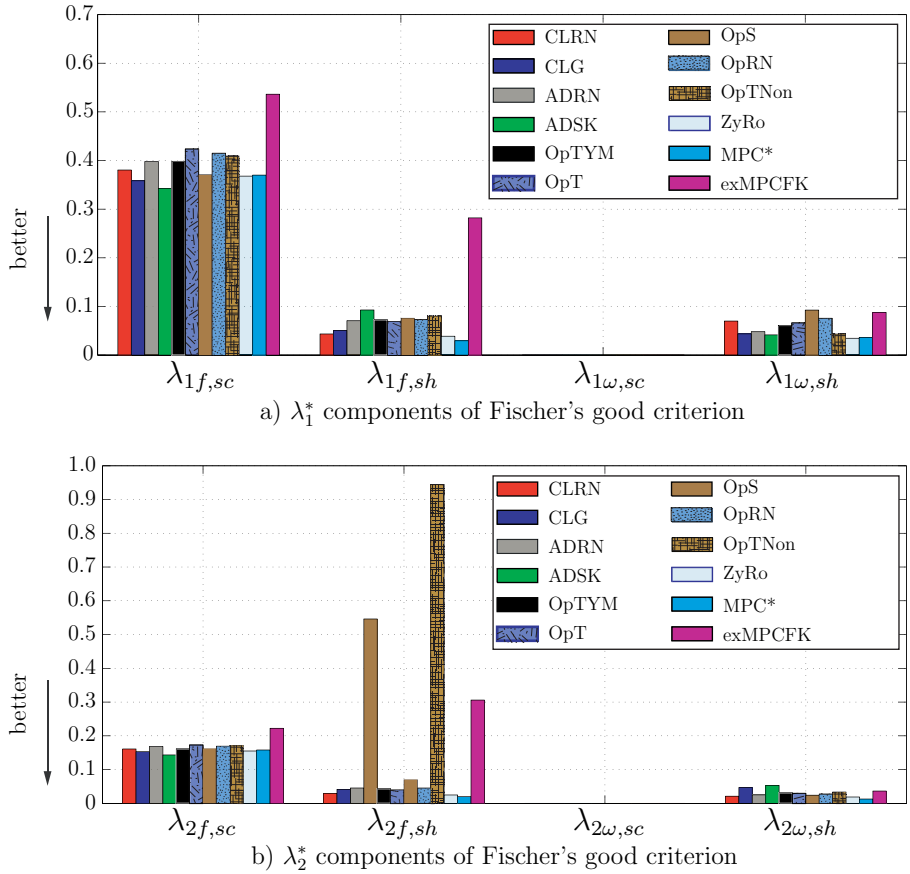


Figure 5.23: Comparison components of Fischer's good criteria for $k = 0.7$

		Scale factor k						
		0.4	0.5	0.6	0.7	0.8	0.9	1.0
MCAs	CLRN	15	3	3	3	3	3	3
	OpS	15	3	3	3	3	3	3
	OpRN	15	3	3	3	3	3	3
	OpTYM	15	3	3	3	3	3	3
	OpT	15	3	3	3	3	3	3
	exMPCFK	15	3	3	3	3	3	3
	CLG	14	10	2	2	2	2	2
	ADSK	14	10	2	2	2	2	0
	OpTNon	14	10	10	3	2	2	2
	ADRN	14	14	14	3	3	3	3
	MPC*	15	15	15	11	11	11	7
	ZyRo	14	14	14	14	14	10	10

Table 5.11: Resulting "best" MF for manually-tuned MCAs depending on the outer scale factor $k \in \{0.4 \cdots 1.0\}$

5.5 Auto-tuning with the MVMO method for the example of the ZyRo algorithm

As can be seen from the previous section, tuning by trial and error requires much effort and experience from the designers. To overcome the problem of the manual trial and error, an auto-tuning method can be sought for which automatically finds the suitable parameter set. In this section, we analyze the application of the mean-variance mapping optimization (MVMO) for the case of the offline tuning of the ZyRo algorithm. A similar approach can be applied to the other MCA approaches, which is left for further work. The main concept is shown in Fig. 5.24. Here, the block "MVMO" replaces the trial and error feedback used in manual tuning. We describe first the basic concept of the MVMO approach, followed by definition of the cost function components used for the auto-tuning method. Based on these, three tuning criteria are evaluated for the ZyRo algorithm, showing the practicability of the approach.

5.5.1 Summary of the MVMO method

Mean-variance mapping optimization (MVMO) is a novel optimization algorithm mapping method developed by Erlich et al. (2010) The method is a member of the family of so-called population-based stochastic optimization techniques, which incorporates the information of the performance of a specific number of best individuals to minimize a particular cost function. The approach employs the concepts of selection, mutation and crossover from evolutionary computation algorithms, and develops its own transformation strategy for mutated genes of the offsprings based on the mean-variance of the n-best population (Erlich et al. (2010)).

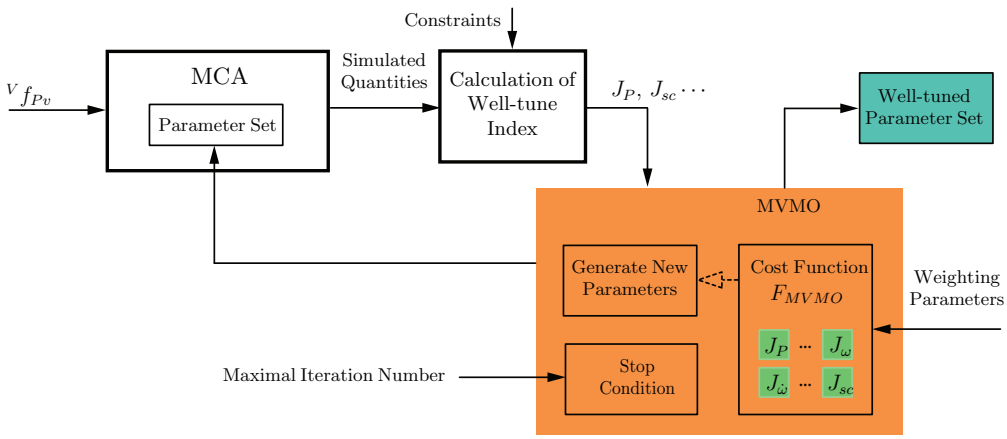


Figure 5.24: Concept of the auto tuning method for a MCA

5.5.2 Selected cost function for the MVMO approach

The optimization method MVMO requires a suitable cost function that is to be minimized by automatically variating the MCA parameter set at each optimizing cycle. In this study, the cost function is defined as a weighted sum of penalty function, which is described in the sequel.

Fig. 5.24 shows the MVMO block, which consists of a "Stop Condition" block controlled by the "Maximal iteration count", and a "Cost Function" block that is manipulated by the "Weighting parameters" assigning a level of priority for the penalty functions J_k of the global cost function.

$$F_{MVMO} = \sum_{k \in \{P,\, fsh,\, \omega,\, \dot{\omega},\, sc,\, tr,\, Wo\}} w_k J_k \,. \tag{5.18}$$

The penalty functions J_k (see Tab. 5.12) are built from the proposed criteria C_i^* described in the Section 5.3.3 together with the weighting factors w_i with $i \in \{P,\, fsh,\, \omega,\, \dot{\omega},\, sc,\, tr,\, Wo\}$. Hereby, the following penalty functions are used:

- J_P: Position penalty function that limits movement inside workspace
- J_{fsh}: Shape error penalty function that restricts the maximum shape errors
- J_ω: Angular velocity penalty function that restricts the maximum angular velocity
- $J_{\dot{\omega}}$: Angular acceleration penalty function that restricts the maximum angular acceleration
- J_{sc}: Scale error penalty function that maintains the scale factor in a specific range
- J_{tr}: Translational motion penalty function that controls the amount of translational motion
- J_{Wo}: Washout penalty function that determines the usage of washout effect on a tilt angle

The weighting parameters can be changed according to the aim of the tuning process. For example, to obtain a high well-tuned index MF, the weighting parameters can be adjusted to match the values in Eq. 5.17.

Penalty function	Definition
J_P	$\sum_{j\in\{1\cdots6\}} \left\{ e^{\left\langle z_j^L - z_i^{min} \right\rangle^2} + e^{\left\langle z_j^{max} - z_i^U \right\rangle^2} - 2 \right\}$
J_{fsh}	$e^{\left\langle \lvert e_{f,sh}\rvert^{max} - k_O \delta_O \right\rangle^2} - 1$
J_ω	$e^{\langle \lvert\omega\rvert^{max} - \omega_{th} \rangle^2} - 1$
$J_{\dot\omega}$	$e^{\langle \lvert\dot\omega\rvert^{max} - \dot\omega_{th} \rangle^2} - 1$
J_{sc}	$k_{Jsc}\left[e^{\langle k_{min} - k_S \rangle^2} + e^{\langle k_S - k_{max} \rangle^2} - 1 \right] - 1$ $+ \langle k_{max} - k_S \rangle^0 \langle k_S - k_{min} \rangle^0 \left[e^{\langle k_{max} - k_S \rangle^2} + k_{Jsc} \right]$
J_{Wo}	$e^{g^2 \lvert\varphi_S(t_f)\rvert^2} - 1$
J_{tr}	$e^{\langle k_{a,min} - k_a \rangle^2} - 1$

Remarks:

1) $\langle f(x) \rangle^n$ denotes the Heaviside function

$$\langle f(x) \rangle^n = \begin{cases} 0 & \text{for } f(x) < 0 \\ f(x)^n & \text{for } f(x) \geq 0 \end{cases}$$

2) For J_P, $\underline{z} = [x_E,\ y_E,\ z_E,\ \varphi_E,\ \theta_E,\ \psi_E]$ denotes the global coordinates of the end effector, $\underline{z}^L$ and $\underline{z}^U$ denote the lower and upper limits of the feasible motion box within the workspace, and $\underline{z}^{min}$ and $\underline{z}^{max}$ denote the minimal and maximal values attained through the task trajectory, respectively. Note that, if there is no violation of the lower or upper physical limits , $J_P = 0$
3) $\lvert\cdots\rvert^{max}$ denotes the maximum of the absolute value of the argument
4) $k_O \in [0,\ 1]$ is a shaping parameter for the shape error penalty function. A small value means that the shape error is "punished" more strongly
5) $[k_{S,min},\ k_{S,max}]$ is the range of desired scale factor, and k_{Jsc} is a shaping parameter that controls the amplitude of the penalty function when k_S is within the desired range.
6) $k_{a,min}$ denotes the minimum desired ratio between simulated and target translational accelerations.

Table 5.12: Definition of penalty functions

Selected limit-values							
	$k_O\,(-)$	$\delta_O\,(m/s^2)$	$\delta_S\,(rad/s)$	$\dot{\omega}_{th}\,(rad/s^2)$	$k_{S,min}\,(-)$	$k_{S,max}\,(-)$	$k_{a,min}\,(-)$
AT1	1	0.17	0.1	0.2	0.4	1	0.4
AT2	1	0.17	0.1	0.2	0.4	0.4	0.4
AT3	1	0.17	0.1	0.2	0.4	0.4	0.5
Searching ranges of the tuned parameters							
	$q_1\,(-)$	$q_2\,(-)$	$r_1\,(-)$	$r_2\,(-)$	$c\,(1/s)$	$\gamma\,(1/s)$	k_S
AT1	[0.001 , 0.1]	[1 , 100]	[1 , 300]	[0.001 , 10]	[0.001 , 1]	[0.01 , 1]	[0.4 , 1]
AT2	[0.001 , 0.1]	[1 , 400]	[1 , 300]	[0 , 20]	[0.001 , 1]	[0.01 , 1]	0.4
AT3	[0.001 , 0.1]	[1 , 400]	[1 , 300]	[0 , 20]	[0.001 , 1]	[0.01 , 1]	0.4
Weighting parameters							
	wJ_P	wJ_ω	$wJ_{\dot{\omega}}$	wJ_{sc}	wJ_{tr}	wJ_{fsh}	wJ_{Wo}
AT1	2^{10}	2^9	2^6	2^4	0	2^6	2^6
AT2	2^8	2^6	2^6	2^2	1	2^6	2^6
AT3	2^8	2^6	2^6	2^2	2^2	2^6	2^6
Optimized parameters found by the auto-tuning with MVMO & well-tuned index							
	$q_1^*\,(-)$	$q_2^*\,(-)$	$r_1^*\,(-)$	$r_2^*\,(-)$	$1/c^*\,(s)$	$1/\gamma^*\,(s)$	k_S^*	MF
AT1	0.0553	93.1597	57.0096	7.4812	0.8169	0.0229	0.92	14
AT2	0.0550	388.19	56.8747	0.3644	0.8	0.3458	0.4	14
AT3	0.1432	183.8770	241.5727	0.0230	0.9922	0.5295	0.4	14

Table 5.13: Limited values, searching ranges, weighting parameters, optimized parameters, and well-tuned index of the **ZyRo** algorithm

5.5.3 Application to the ZyRo algorithm

In order to test the MVMO approach, three optimization trials were carried out

1. [**AT1**] Optimizing all parameters including k_S.
2. [**AT2**] Optimizing all parameters except k_S which is held fixed at $k_S = 0.4$.
3. [**AT3**] Optimizing all parameters except $k_S = 0.4$ and increasing the translational acceleration ratio $k_{a,min}$ from 0.4 to 0.5.

The corresponding results are displayed in Tab. 5.13. More explicit plots on the achieved trajectories are shown in Appendix C.

For the run **AT1**, the optimizer could find a quite high scale factor $k_S^* = 0.92$, while obeying well to the constraints, as can be seen by a high well-tuned index of 14. Moreover, all angular velocities and accelerations are under their threshold values and, thanks to the washout penalty function J_{Wo}, the simulated angle φ_S is pulled to zero at the end of the simulation.

For the run **AT2** the target specific force was first scaled down by a scale factor of $k = 0.4$ before applying the ZyRo algorithm and a minimal ratio of simulated translational

acceleration to target translational acceleration of $k_a = 0.4$. The simulated specific force tracks the scaled target signal very well with maximum shape error $e^*_{f,sh} \approx 0$. Also, one can see that the automated tuning renders quite similar values to those of the manual trial and error tuning but of course at a tiny fraction of the required time.

Finally, a run **AT3** was performed with the same scale factor $k_S = 0.4$ as **AT2**, but with a higher minimal ration of $k_{a,min} = 0.5$. Tab. 5.13 displays that the MCA produces only a slightly higher ratio of $k_a = 0.43$, which shows that the optimization of run **AT2** already is close to the best tuning that can be achieved with the ZyRo algorithm for a scale factor of $k_S = 0.4$. The time plots in the appendix Fig. C.10 show that the more translational motions are used, the later the simulated angle returns to zero at the end of the simulation.

In summary, the test runs show that the auto tuning with the MVMO method has a number of advantages compared to manual tuning methods proposed in the literature. Firstly, the method is time-efficient, easily implementable and flexible. A designer only determines the purpose of the optimization and adjusts the weighting parameters of the corresponding penalty functions. Secondly, the MVMO tuning method allows one to consider various physical quantities instead of regarding solely the position, and angular velocity. Finally, while this has not been investigated in this context, the auto-tuning method can be applied both to various types of MCAs and to different types of drive-tracks in offline mode which can be tackled in future research.

6 Example Study of Correlation Between Numerical and Subjective Perception Quality

As mentioned above, various objective assessments were introduced to evaluate the level of motion perception fidelity or realism of a simulation session. However, the relationship between the level of motion perception fidelity evaluated by an objective assessment and that by a subjective assessment remains an open question. In this Chapter, a study investigating the correlation between objective and subjective assessment of motion perception fidelity is carried out. The procedure consists of 3 steps as described in Fig. 6.1. First, the simulated quantities (signals) were generated by the selected MCAs. Then, the simulated quantities were used to generate the trajectory for the KUKA Robocoaster and to compute the numerical indices of the objective assessment. Lastly, the trajectories were implemented in the driving simulator to obtain a assessment by a set of participants, and the subjective results were used to construct the numerical functions of motion perception fidelity. For the study, we used not well-tuned algorithms to purposedly generate false cues that could be related to subjective assessment. The individual steps of the study are described in detail in sections 6.1, 6.2 and 6.3 along with a fitting procedure in section 6.4 to correlate the numerical measures (MF, "performance indicator", "good criterion", and "penalty functions") with subjective rating.

6.1 Generating the simulated signals based on the reference motion

In the first step, ten simulated signals (Fig. 6.2) were deliberately generated by nine MCAs considering the following aspects:

a) simulated signals must satisfy the criteria C_1^*
b) motion cues errors exist in simulated signals
c) global scale factor k_S vary from 0.33 to 1
d) the centripetal accelerations are not compensated

Note that the condition a) guarantees the implementability of the simulated data in a driving simulator; the conditions b), c), and d) generate various types of motion cue errors (i.e. rotational false cues, scale errors and shape errors), that have effects on the level of motion perception fidelity.

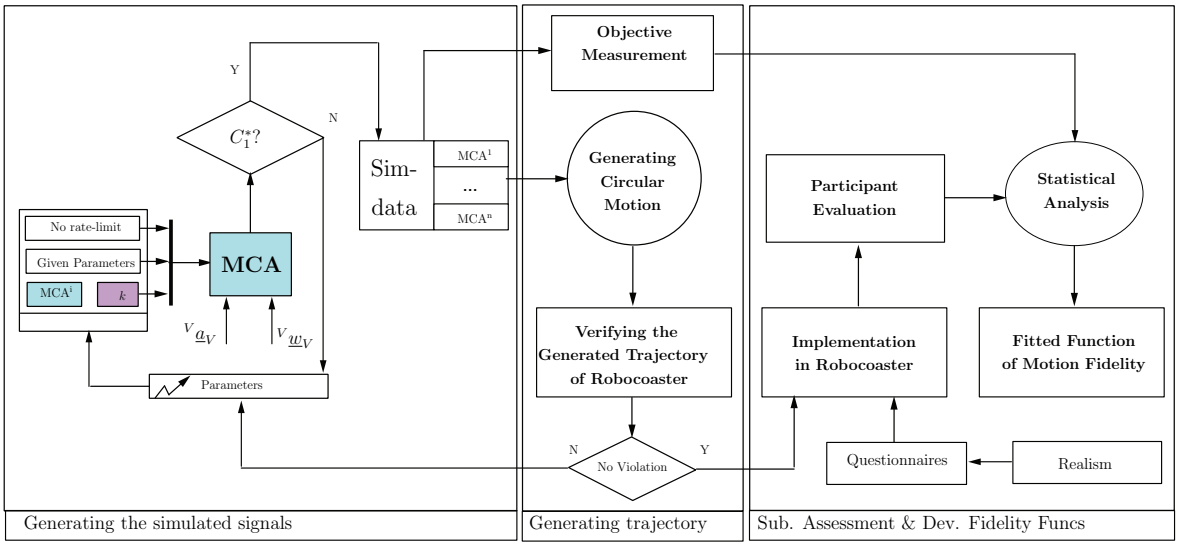
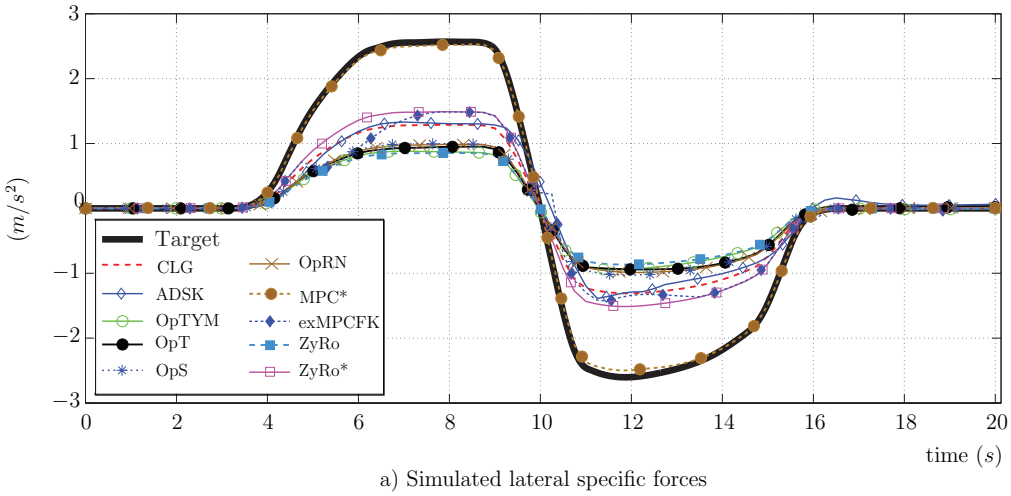


Figure 6.1: Procedure finding correlation between objective and subjective assessment of simulated signals generated by various MCAs

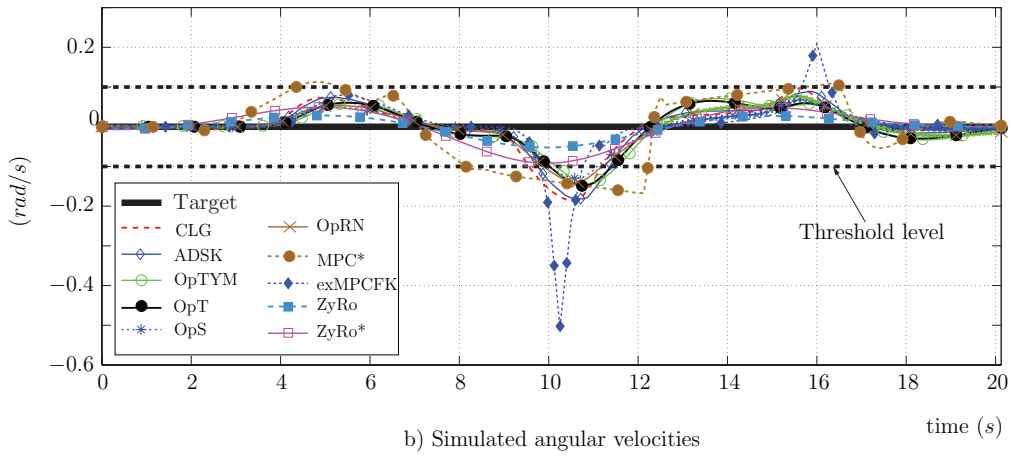
The time history responses of simulated signals, denoted by the corresponding MCAs, are illustrated in Fig. 6.2a. Most algorithms display strong scale errors and small shape errors of simulated specific forces. The exMPCFK has a peak in angular velocity false cues, while the algorithms – OpS, OpT, and OpRN – have small rotational false cues and the ZyRo and ZyRo* algorithms have no false cues at all (ZyRo* represents the ZyRo algorithm with hand-tuned parameters). Although the CLG and ADSK algorithms have quite the large false cues, the exMPCFK algorithm has the largest. Furthermore, 5 algorithms generate distortions of simulated surge specific force (centripetal acceleration) that are larger than the otolith threshold level δ_O (Fig. 6.2c). The reason is that the algorithms use wide circular motions to simulate the lateral acceleration. However, the largest distortion, which appears in the OpS algorithm, reaches only about $0.25\,m/s^2$.

6.2 Objective assessments and trajectory generation

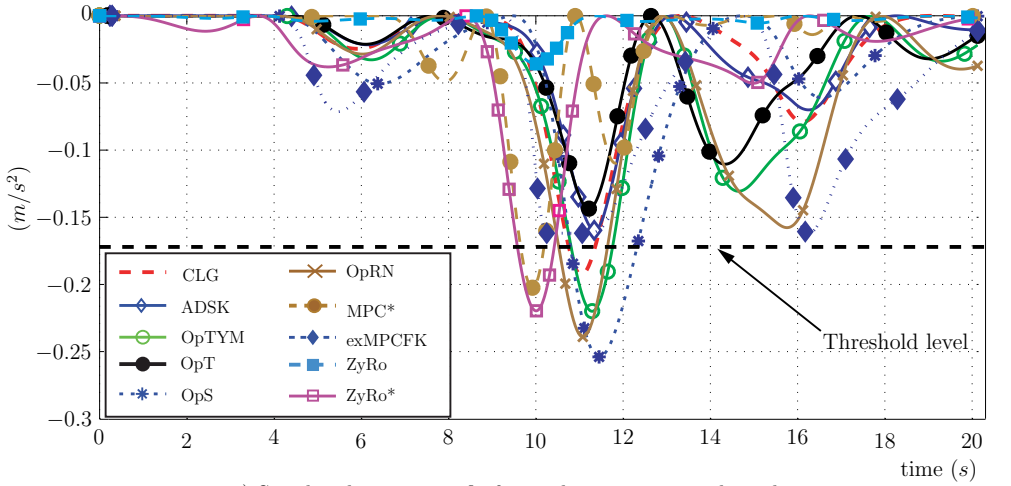
The objective assessments of the tested trajectories from Fig. 6.2 are shown Fig. 6.3. Hereby, the ZyRo algorithm displays the smallest shape errors $\lambda_{1f,sh}$ and $\lambda_{2f,sh}$, but the largest scale errors $\lambda_{1f,sc}$ and $\lambda_{2f,sc}$. Besides, the optimal washout filters OpS, OpRN, OpT, OpTYM have large shape errors $\lambda_{1f,sh}$ and $\lambda_{2f,sh}$ due to the distortion of the simulated surge specific forces.



a) Simulated lateral specific forces

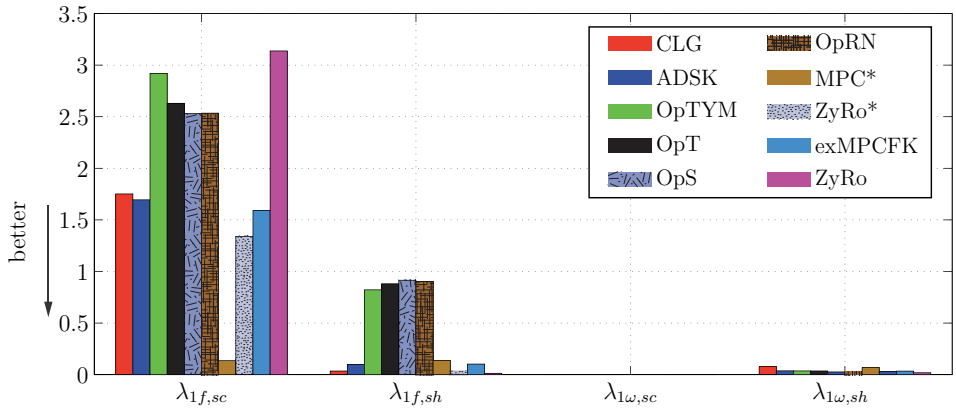


b) Simulated angular velocities

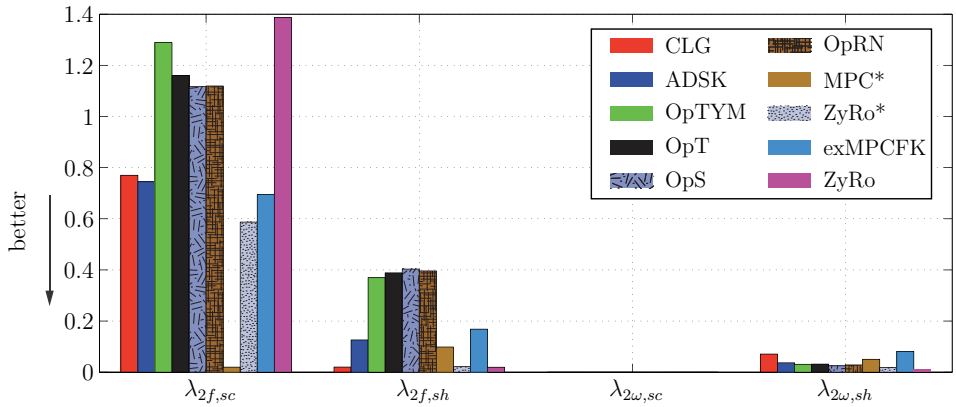


c) Simulated surge specific forces due to centripetal accelerations

Figure 6.2: Comparison of the simulated signals: a) lateral specific forces, b) angular velocity, and c) surge specific force



a) λ_1^* components of 'good criterion'



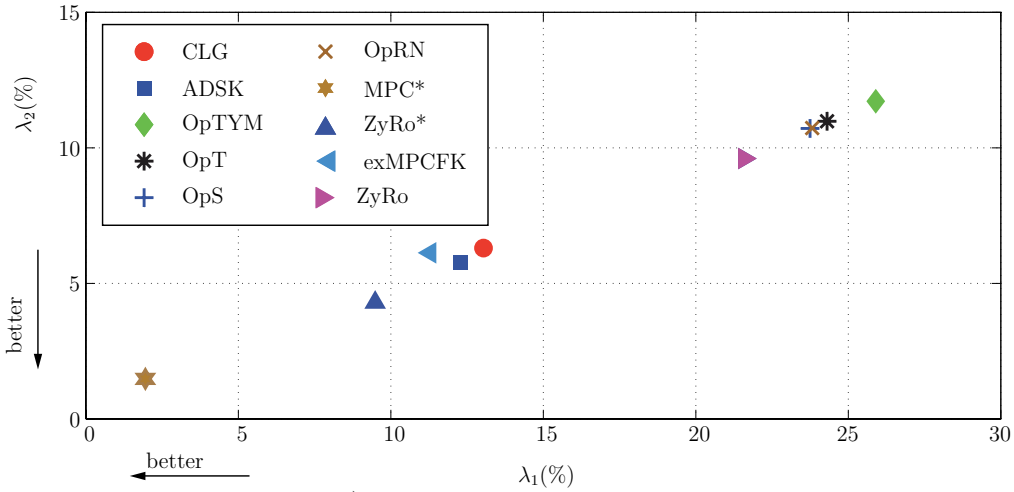
b) λ_2^* components of 'good criterion'

Figure 6.3: Comparison of numerical performance measures for ten simulated MCAs

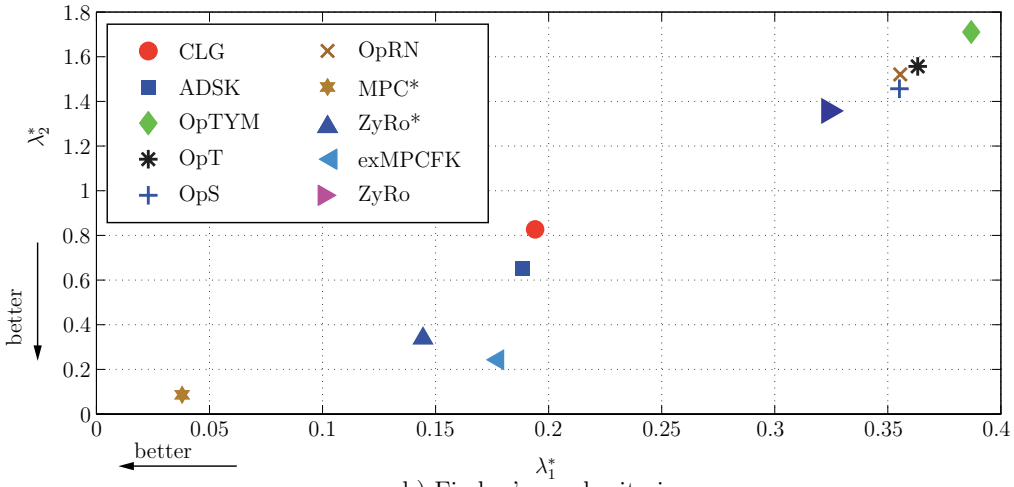
	CLG	ADSK	OpTYM	OpT	OpS	OpRN	MPC*	exMPCFK	ZyRo	ZyRo*
k_S	0.5	0.51	0.38	0.35	0.39	0.39	0.98	0.53	0.33	0.58
$e^*_{f,sh}$	0.1927	0.1597	0.2200	0.1270	0.2539	0.2390	0.2024	0.1618	0.0358	0.2196
${}^{C}e^*_{f,sh}$	0.0218	0.4463	0.0805	0.0836	0.0301	0.0795	0.0990	0.4210	0.0065	0.0124
$\overline{C_1^*}$	1	1	1	1	1	1	1	1	1	1
$\overline{C_2^*}$	0	0	0	0	0	0	0	0	1	0
$\overline{C_{2,1}^*}$	0	0	0	0	0	0	0	0	1	1
$\overline{C_{2,2}^*}$	0	1	1	1	1	1	0	0	1	0
$\overline{C_3^*}$	0	0	0	1	0	0	0	0	1	0
$\overline{C_{3,1}^*}$	1	0	1	1	1	1	1	0	1	1
$\overline{C_{3,2}^*}$	0	1	0	1	0	0	0	1	1	0
$\overline{C_4^*}$	1	1	1	0	1	1	1	1	0	1
$\overline{C_5^*}$	0	0	1	0	0	1	0	0	0	0
MF	10	10	9	10	10	11	10	10	12	10
SR	2.24	2.35	2.24	2.12	2.06	2.06	2.24	2.82	2.47	2.18

– ${}^{C}e^*_{f,sh}$ represents the maximum shape error of the specific force due to centripetal acceleration
– $\overline{C_{2,1}^*}$, $\overline{C_{2,1}^*}$ relates to angular velocity and angular acceleration, respectively
– $\overline{C_{3,1}^*}$, $\overline{C_{3,1}^*}$ relates to lateral and surge specific forces,, respectively
– $\overline{C_2^*} = \overline{C_{2,1}^*} \wedge \overline{C_{2,2}^*}$ and $\overline{C_3^*} = \overline{C_{3,1}^*} \wedge \overline{C_{3,2}^*}$

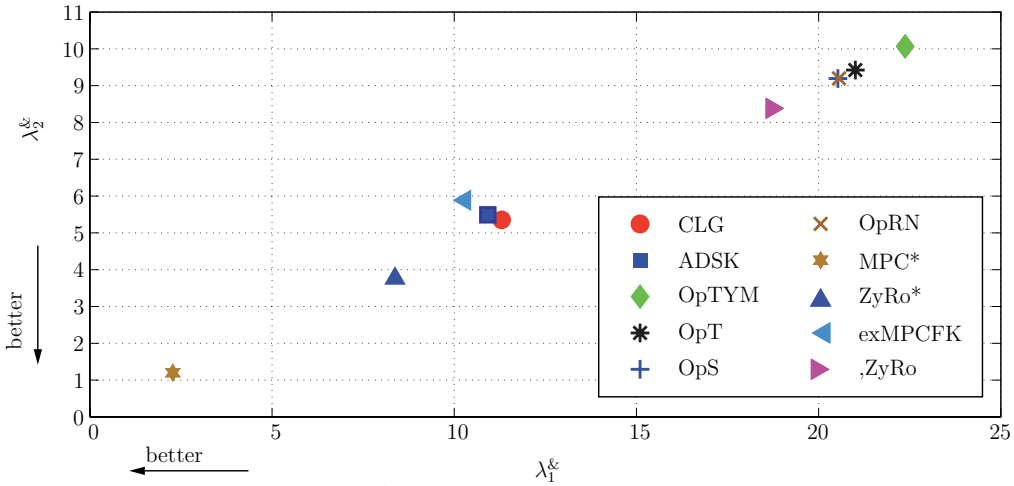
Table 6.1: Average subjective scores and criteria of the simulated MCAs



a) Pouliot's performance indicators



b) Fischer's good criterion



c) Normalized good criterion

Figure 6.4: Comparison of numerical performance measures

The numerical indices of two objective measures, λ_i and $\lambda_i^{\&}$ (Fig. 6.4a and Fig. 6.4c) yield a similar distribution, the ZyRo* algorithm displays the second-best numerical scores after the MPC* algorithm. The exMPCFK algorithm has larger indices than the ZyRo* algorithm for λ_i and $\lambda_i^{\&}$, but has smaller index λ_2^* than the ZyRo* algorithm. Furthermore, the OpTYM algorithm has the worst scores for λ_1 and λ_1^*. The MPC* algorithm seems to produce signals with better motion perception fidelity than the OpTYM and ZyRo* algorithms.

The maximum shape errors $e_{f,sh}^*$, ${}^C e_{f,sh}^*$ (for lateral and surge specific force), numerical criteria $\overline{C_i^*}$, and well-tuned index MF are computed as shown in Tab. 6.1. It can be seen that the exMPCFK algorithm has the second largest shape error of simulated lateral specific force, while the shape errors of almost all algorithms are under the threshold level δ_O of the otolith organ. The exMPCFK algorithm has no shape errors of the surge specific force, whereas the ZyRo* and MPC* algorithms have the large shape errors. Additionally, the ZyRo* and MPC* algorithms produce no false cue of angular velocity ($\overline{C_{2,1}^*} = 1$), but have false cues of angular acceleration $\overline{C_{2,2}^*} = 0$, hence $\bar{C}_2^* = 0$. On the opposite side, the signals of the algorithms (ADSK and Op-family) have false cues of angular velocity and no false cues of angular acceleration. The CLG and exMPCFK algorithms have both types of false cues. Furthermore, most simulated signals satisfy the C_4^* criterion except for the OpT and ZyRo algorithms. The ZyRo algorithm has the smallest global scale factor $k_S = 0.33$ which potentially causes weak motion cues (C_4^* criterion).

Concerning the form error criterion C_3^*, only the OpT and ZyRo algorithms have small distortion of specific forces in both lateral and surge channels, which is reflected in both sub-criteria $\overline{C_{3,1}^*}$ and $\overline{C_{3,2}^*}$. Other algorithms have either large lateral shape error (e.g. ADSK and exMPCFK) or surge shape error (e.g CLG, OpS, MPC*, and ZyRo*).

In summary, the ZyRo algorithm has medium level well-tuned index ($MF = 12$), while the OpTYM algorithms has the smallest index ($MF = 9$), and the rest have low level ones ($MF \in \{10, 11\}$). With the various errors in the simulated signals, it is possible to investigate the significance of each type of errors and false cues on motion fidelity by subjective assessment.

6.3 Subjective test assessments

Preliminary subjective tests were carried out with real passenger rides to validate the effect of the types of false cues on motion fidelity or realism of the simulation. A total of

17 raters in the age range 20–30 years (5 female and 12 male) were recruited and asked to rate 10 different rides using the German school mark scale of 1=very good, 2=good, 3=satisfactory, 4=sufficient, 5= failed with +0.3 for a "minus" (all apart 5) and −0.3 for a "plus" (all apart 1). This scaling was used as it is intuitively well-perceived by all students and thus no additional instructions were needed. Each participant tried randomly once each of the 10 rides. An interclass correlation was computed as a two-way Anova in which each target ride was rated by each the 17 participants ("judges"). These were classified as the only judges of interest, i.e. ICC(3,k) with k=17 according to Shrout and Fleiss (1979). We obtained an ICC of 0.4937 with a false-rejection probability of p=0.0462, which is a fair reliability. The same problem with the subjective assessments were mentioned in studies of Reid and Nahon (1988), and Grant et al. (2003).

The average subjective ratings (SR) are shown in Tab. 6.1 and Fig. 6.5, while the individual subjective ratings are shown in Fig. 6.6 by a "bubble plot" displaying with the radius of the circles to the number of hits for an individual score. One can see that there is a great variability of the scores, were raters rate one and the same ride apart from the best two in the full scale (1-5). Even for the worst ride (exMPCFK), that has average subjective score 2.82 (Fig. 6.5), one rater rated this as "very good". We do not know if this positive rating was done to be "nice" or if this is due to a different false angular velocity cue threshold for this person.

6.4 Correlating numerical and subjective scores

The objective of the numerical scores is to have a means to predict the subjective perception of a ride prior to its actual implementation. However, the analysis of the numerical scores (numerical rating = NR) and the average subjective scores (SR) shows that there is no evident relationship between the two when single numerical scores are used. For example, ZyRo is second worst in SR, but it has the best MF. Vice versa, OpS algorithm has the best SR but an MF index similar to others. Likewise, the "performance indicators" and the "good criterion" (Fig. 6.4) display a different ranking for the MCAs than that of the SR. For example, the MPC* algorithm features the best "performance indicators" and the "good criterion", but has the same level of SR as OpTYM, which has the worst score both for the "performance indicators" and the "good criterion". This shows that the single numerical score measures only the causes of discomfort, but how much discomfort affects the motion perception fidelity is not clear.

An alternative approach to better predict the subjective perception of rides by numerical scores is to use multiple linear regression to find the best fit of numerical scores to subjective ratings. In order to analyse the effect of the different numerical scores on the prediction of the subjective perception, six best-fit multiple regressions using the regress function of MATLAB®, were carried out:

BF1a Best fit with the "good criterion"
BF1b Best fit with the "normalized good criterion"
BF1c Best fit with the partial components of the "good criterion"
BF2 Best fit with the "performance indicator"
BF3 Best fit with the "well-tuned index" MF
BF4 Best fit with the individual penalty functions J_ω, $J_{\dot{\omega}}$, J_{sc}, J_{fsh}.

We describe in the following the best fit by functions F_i^{GC}, $i = \{1 \cdots 3\}$ for the case **BF1a**; by function F_i^{NGC}, $i = \{1 \cdots 3\}$ for the case **BF1b**; by functions F_i^{PGC}, $i =$ {1, 2} for the case **BF1c**; by functions F_i^{PI}, $i = \{1 \cdots 3\}$ for the case **BF2**; by function F^{MF} for the case **BF3**; and by functions F_i^{PF}, $i = \{1 \cdots 15\}$ for the case of **BF4** for different choices of regarded individual numerical scores.

BF1a: "good criterion"

In this case the normalized indices $\bar{\lambda}_1^* = \lambda_1^*/\max(\lambda_1^{*(j)})$ and $\bar{\lambda}_2 = \lambda_2^*/\max(\lambda_2^{*(j)})$ are fitted in three functions

$$F_{\{1,2,3\}}^{GC} = w_0 + w_{\bar{\lambda}_1^*}\bar{\lambda}_1^* + w_{\bar{\lambda}_2^*}\bar{\lambda}_2^*\,, \tag{6.1}$$

in which the individual scores are regarded separately and in their combination. The fitted coefficients w_0, $w_{\bar{\lambda}_1^*}$, $w_{\bar{\lambda}_2^*}$ together with the corresponding root mean square error

	BF1a: coefficients for "good criterion"						**BF1b**: coefficients for "normalized good criterion"				
	w_0	$w_{\bar{\lambda}_1^*}$	$w_{\bar{\lambda}_2^*}$	RMSE	CORR		w_0	$w_{\bar{\lambda}_1^*}$	$w_{\bar{\lambda}_2^*}$	RMSE	CORR
F_1^{GC}	2.2716	0.0075	0	0.2167	0.0056	F_1^{NGC}	2.4339	-0.2384	0	0.2054	0.3196
F_2^{GC}	2.4314	0	-0.2199	0.2091	0.2624	F_2^{NGC}	2.3933	0	-0.1708	0.2116	0.2169
F_3^{GC}	1.8270	1.2780	-0.9127	0.1825	0.5396	F_3^{NGC}	2.1421	-4.7505	4.8034	0.1162	0.8441

	BF1c: coefficients for partial components of the "good criterion"								
	$w_{1f,sc}$	$w_{2f,sc}$	$w_{1f,sh}$	$w_{2f,sh}$	$w_{1\omega,sh}$	$w_{2\omega,sh}$	w_0	RMSE	CORR
F_1^{PGC}	41.1217	-38.8255	-3.6247	3.3605	0.9879	-0.0673	-	0.2336	0.7421
F_2^{PGC}	14.6261	-14.0553	-0.7363	0.4109	-0.6579	0.6760	1.9641	0.0250	0.9933

	BF2: coefficients for "performance indicator"						**BF3**: coefficients for MF				
	w_0	$w_{\bar{\lambda}_1}$	$w_{\bar{\lambda}_2}$	RMSE	CORR		w_0	w_{MF}	-	RMSE	CORR
F_1^{PI}	2.4339	-0.4957	0	0.2948	0.4488	F^{MF}	1.9064	0.0364	-	0.2150	0.1258
F_2^{PI}	2.8734	0	-0.4650	0.3033	0.3938	-	-	-	-	-	-
F_3^{PI}	2.2848	-9.7426	9.9156	0.1948	0.8071	-	-	-	-	-	-

Table 6.2: Coefficient of fitted numerical functions for {**BF1, BF2, BF3**}

(RMSE) and Pearson correlation (CORR) are shown in Tab. 6.2. One can see that neither $\bar{\lambda}_1^*$ nor $\bar{\lambda}_2^*$ alone correlate well with the average subjective measure, and that even the best fit of $\bar{\lambda}_1^*$, $\bar{\lambda}_2^*$ together yields only a fair correlation.

BF1b: "normalized good criterion"

A similar proceduce is implemented for the normalized indices of "normalized good criterion", $\bar{\lambda}_1^{\&}$, $\bar{\lambda}_2^{\&}$. The indices have a better correlation than the indices of the original "good criterion", especially, the best fit of the combined $\bar{\lambda}_1^{\&}$ and $\bar{\lambda}_2^{\&}$ yields a good correlation of 84.41%.

BF1c: "good criterion" with partial components

Two further functions

$$F_1^{PGC} = \sum_{p\in\{1,2\}} \sum_{q\in\{f,\omega\}} \sum_{n\in\{sc,sh\}} w_{qp,n} \cdot \bar{\lambda}_{qp,n} \quad , \tag{6.2}$$

$$F_2^{PGC} = \sum_{p\in\{1,2\}} \sum_{q\in\{f,\omega\}} \sum_{n\in\{sc,sh\}} w_{qp,n} \cdot \bar{\lambda}_{qp,n} + w_0 \quad . \tag{6.3}$$

were best fitted. The coefficients $w_{qp,n}$ are listed in the Tab. 6.2. One can see that by best fitting of the partial components of the "good criterion", a good correlation with the average subjective score can be achieved. The function F_1^{PGC} yields Pearson correlation of 74.21% and a root mean square error of 0.2336. It is surprising that the function F_2^{PGC} yields nearly perfect correlation of 99.33% (see also Fig. 6.6). One can see that the numerical scores represent much better the average subjective score than the individual ones (crosses in Fig. 6.6). This indicates that the partial components of the "good criterion" may be a good measure of motion perception.

BF2: "performance indicator"

A similar procedure of **BF1a** is implemented for the numerlized indices of "performance indicator", $\bar{\lambda}_1$, $\bar{\lambda}_2$. The best fitted function F_3^{PI} which combines $\bar{\lambda}_1$ and $\bar{\lambda}_2$ yields a good correlation of 80.71%.

BF3: "well-tuned index" MF

By defining a fitted function

$$F^{MF} = w_{MF}\, MF + w_0 \,, \tag{6.4}$$

a low correlation of 12.58% is obtained (Tab. 6.2). Thus, the well-tuned index has a poor correlation with the average subjective measure.

	$w_{\bar{J}_\omega}$	$w_{\bar{J}_{\dot{\omega}}}$	$w_{\bar{J}_{sc}}$	$w_{\bar{J}_{esh}}$	w_0	RMSE	CORR
F_1^{PF}	0.6145	0	0	0	2.2076	0.1184	0.8378
F_2^{PF}	0	0.3795	0	0	2.1956	0.1717	0.6103
F_3^{PF}	0	0	0.2113	0	2.2545	0.2074	0.2906
F_4^{PF}	0	0	0	0.6433	2.1910	0.1050	0.8749
F_5^{PF}	0.6301	-0.0178	0	0	2.2097	0.1183	0.8380
F_6^{PF}	0.6584	0	0.3046	0	2.1688	0.3054	0.2906
F_7^{PF}	-0.3756	0	0	1.0124	2.1840	0.1027	0.8807
F_8^{PF}	0	0.4454	0.3321	0	2.1444	0.1421	0.7550
F_9^{PF}	0	0.0030	0	0.6408	2.1906	0.1050	0.8749
F_{10}^{PF}	0	0	0.2731	0.6691	2.1571	0.0667	0.9515
F_{11}^{PF}	-0.4738	0.0517	0	1.0664	2.1767	0.1020	0.8823
F_{12}^{PF}	0.6091	0.0576	0.3132	0	2.1610	0.0759	0.9367
F_{13}^{PF}	0	0.0830	0.2894	0.6024	2.1463	0.0636	0.9560
F_{14}^{PF}	-0.4738	0.0517	0	1.0664	2.1767	0.3743	0.8794
F_{15}^{PF}	-0.1894	0.1007	0.2832	0.7734	2.1417	0.0628	0.9571

Table 6.3: Coefficients of fitted numerical functions $\{F_1^{PF}, \cdots, F_{15}^{PF}\}$

BF4: Best fit with the individual penalty functions J_ω, $J_{\dot{\omega}}$, J_{sc}, J_{fsh}

In this best-fit run, fifteen functions

$$F_i^{PF} = w_0 + w_{\bar{J}_\omega}\bar{J}_\omega + +w_{\bar{J}_{\dot{\omega}}}\bar{J}_{\dot{\omega}} + +w_{\bar{J}_{sc}}\bar{J}_{sc} + +w_{\bar{J}_{fsh}}\bar{J}_{fsh} \quad i = \{1 \cdots 15\} \tag{6.5}$$

were analysed in which the penalty functions are regarded individually, in combinations of two, in combinations of three, and in a combination of all four. The coefficients for fifteen combinations are listed together with the respective root mean square error and the Pearson's correlation in Tab. 6.3. Clearly, the functions F_{10}^{PF}, F_{13}^{PF} and F_{15}^{PF} have best correlation, as shown in the Fig. 6.5.

Individually, from Tab. 6.3, the values of penalty functions $\bar{J}_\omega$ and of $\bar{J}_{fsh}$ have significant impact on motion perception fidelity, while the values of $\bar{J}_{\dot{\omega}}$ and $\bar{J}_{sc}$ have only poor effect on motion perception fidelity. This result is in agreement with publications related to the effect of false cues on motion perception fidelity.

However, in combination, the penalty functions provide the best correlation cases. Therefore, the impact of combined scale error and shape error of the simulated specific forces on motion perception fidelity is most significant. This means that the angular velocity and acceleration is not critical as long as the motion follows the target shape, and the angular velocity and acceleration are close to their limits. The reason may be that the thresholds of angular velocity and acceleration in the simulation task are

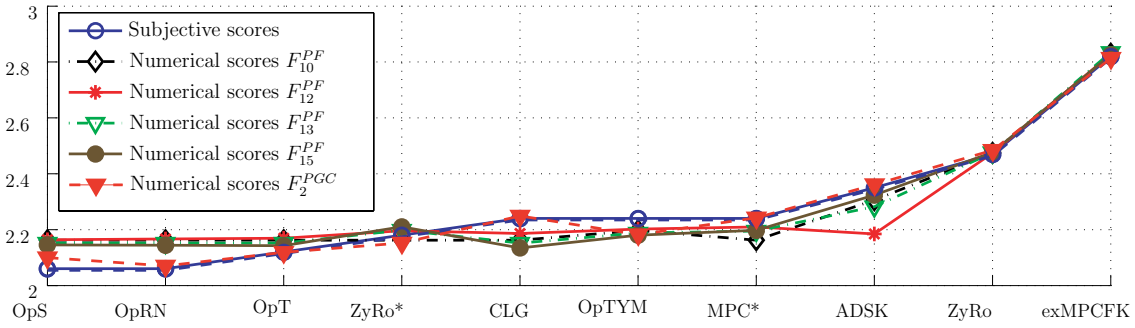


Figure 6.5: Correlation of subjective and numerical scores for F_2^{PGC} and $F_{\{10,12,13,15\}}^{PF}$

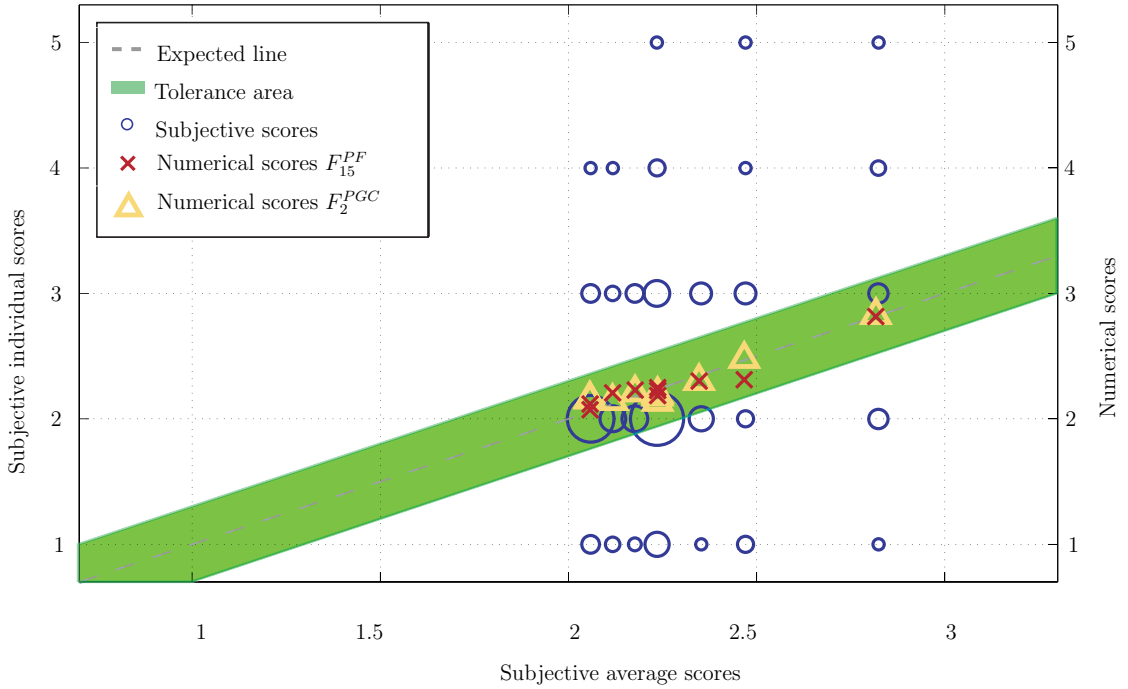


Figure 6.6: Individual subjective and numerical scores for best fitted functions F_2^{PGC} and F_{15}^{PF}

too strict. If the assumption is right, the Op-family set of simulated signals probably best cues. Another implication is that the false cues of angular velocity dominate the motion perception fidelity only if there is no shape error and scale error of the simulated specific force. Otherwise, the rotational false cues are masked by the shape errors.

In summary, the partial components of the "good criterion" yield as such a good measure as the best fit of the combined penalty functions. This may be of use for future development of numerical score for assessment of perception quality in ride simulators.

7 Conclusion and Further Research

This thesis compares existing motion-cueing algorithms (MCAs) and adapts them to the LMR driving simulator for a planar S-curve maneuver. Furthermore, an auto-tuning method is proposed for offline tuning. In order to find a correlation between numerical indices and subjective assessment of motion perception fidelity, a preliminary subjective test was performed with ten arbitrary simulated signals, and then fitted functions according to numerical indices are computed. From this work, the main conclusions are:

(1) For a simulation task with only lateral acceleration, all MCAs studied can be adapted with high well-tuned indices

(2) For a serial robot, circular motion of the cabin can be well-compensated by pitch angles

(3) Among all MCAs, optimal tracking (Zywiol-Romano algorithm = ZyRo) and model-predictive control (MPC*) algorithms can simulate a large amplitude input signal while keeping a high well-tuned index. Furthermore, these algorithms exploit better simulator's workspace than other MCAs and are easily tuned

(4) The ZyRo algorithm produces comparable results as the MPC* algorithm, while the MPC* algorithm requires more computational time

(5) By analyzing simulated signals, the usage of the motion perception model in the optimal washout filters and MPC* algorithm do not have any advantages compared to other MCAs. Additionally, the responses of all MCAs studied are similar to the scale $k = 0.4$. However, if a larger scale factor is used, the responses of the MCAs have apparent differences

(6) An auto-tuning process is proposed using a mean-variance mapping optimization (MVMO) that automatically tunes parameters to obtain a high well-tuned index MF. A case study is performed for determining the maximum global scale factor $k_{S,max}$. Remarkably, the pitch rotational motion that is used to compensate for the centripetal acceleration is also kept under the threshold value by the penalty functions

(7) A subjective for test checking motion perception fidelity of 10 simulated signals was implemented. From these measurements, a good correlation between average subjective and objective measures can be achieved.

However, due to the large variability of the individual scores, further research is still necessary to better understand the subjective ratings of simulator rides. This involves:

- The optimal selection of the initial position of the simulator to improve the usage of simulator's workspace for a particular simulation task.
- Further subjective experiments focusing on the effect of false cues and scale errors on the motion perception fidelity.
- The effects of the target yaw rate fell outside the context of this thesis, but the errors that it generates in the simulation task could be a topic of future studies.

Bibliography

Anderson, B. D. and Moore, J. B. (1971). *Linear optimal control*, Volume 197. Prentice-Hall Englewood Cliffs.

Augusto, B. and Loureiro, R. (2009). Motion cueing in the Chalmers driving simulator: A model predictive control approach. Master's thesis, Chalmers University of Technology.

Baarspul, M. (1986). Flight simulation techniques with emphasis on the generation of high fidelity 6 DoF motion cues. Technical report, Delft University of Technology.

Baseggio, M., Beghi, A., Bruschetta, M., Maran, F., and Minen, D. (2011). An MPC approach to the design of motion cueing algorithms for driving simulators. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 692–697. IEEE.

Benson, A. J., Spencer, M. B., and Stott, J. R. (1986). Thresholds for the detection of the direction of whole-body, linear movement in the horizontal plane. *Aviation, space, and environmental medicine*.

Berthoz, A., Bles, W., Bulthoff, H. H., Correia Gracio, B. J., Feenstra, P., Filliard, N., Huhne, R., Kemeny, A., Mayrhofer, M., and Mulder, M. (2013). Motion scaling for high-performance driving simulators. *Human-Machine Systems, IEEE Transactions on*, 43(3):265–276.

Blauert, J. (1997). *Spatial hearing: The psychophysics of human sound localization*. MIT press.

Boff, K., Kaufman, L., and Thomas, J. (1986). Handbook of perception and human performance. Volume 1 : Sensory processes and perception. Volume 2: Cognitive processes and performance. *New York, Wiley-Interscience, 1986.*, 1.

Borah, J., Young, L. R., and Curry, R. E. (1988). Optimal estimator model for human spatial orientation. *Annals of the New York Academy of Sciences*, 545(1):51–73.

Braithwaite, M. G. and Braithwaite, B. D. (1990). Simulator sickness in an army simulator. *Occupational Medicine*, 40(3):105–110.

Bremmer, F. and Lappe, M. (1999). The use of optical velocities for distance discrimination and reproduction during visually simulated self motion. *Experimental Brain Research*, 127(1):33–42.

Bringoux, L., Nougier, V., Marin, L., Barraud, P., and Raphel, C. (2003). Contribution of somesthetic information to the perception of body orientation in the pitch dimension. *The Quarterly Journal of Experimental Psychology Section A*, 56(5):909–923.

Bruenger-Koch, M. (2005). Motion parameter tuning and evaluation for the DLR automotive simulator. In *Driving Simulator Conference-North America*.

Brünger-Koch, M., Briest, S., and Vollrath, M. (2006). Do you feel the difference? A motion assessment study. In *Proc. Driving Simulator Conference*.

Cardullo, F. and Kosut, R. (1983). Old problem/new solutions—Motion cuing algorithms revisited. In *Flight Simulation Technologies Conference*, page 1082.

Cardullo, F., Sweet, B., Hosman, R., and Coon, C. (2011). The human visual system and its role in motion perception. In *AIAA Modeling and Simulation Technologies Conference*, page 6422.

Casali, J. G. and Frank, L. H. (1988). Manifestation of visual/vestibular disruption in simulators: Severity and empirical measurement of symptomatology. *Motion Cues in Flight Simulation and Simulator Induced Sickness*, pages 6.1–6.11.

Chapron, T. and Colinot, J. (2007). The new psa Peugeot-Citroen advanced driving simulator overall design and motion cue algorithm. In *Proceedings of Driving Simulation Conference*.

Chen, S. and Fu, L. (2007). Predictive washout filter design using the forward kinematics and a Kalman filter. In *2007 IEEE International Conference on Control Applications*, pages 420–425. IEEE.

Chen, S. and Fu, L. (2011). An optimal washout filter design with fuzzy compensation for a motion platform. In *18th IFAC World Congress Milano, Italy August*.

Cho, Y. M., Kim, H. S., Kim, I. K., Woo, J. J., and Kim, J. (2007). Fast design of the QP-based optimal trajectory for a motion simulator. *Journal of mechanical science and technology*, 21(12):1973–1985.

Conrad, B. and Schmidt, S. F. (1970). *Motion drive signals for piloted flight simulators*, Volume 1601. National Aeronautics and Space Administration.

Cooper, G. E. and Harper Jr, R. P. (1969). The use of pilot rating in the evaluation of aircraft handling qualities. Technical report, Advisory group for aerospace research and development Neuilly-sur-Seine (France).

Crowley, J. S. (1987). Simulator sickness: A problem for army aviation. *Aviation, space, and environmental medicine*.

Dagdelen, M., Reymond, G., Kemeny, A., Bordier, M., and Maïki, N. (2004). MPC based motion cueing algorithm: Development and application to the ultimate driving simulator. In *Conférence simulation de conduite*, pages 221–233.

Damveld, H. J., Wentink, M., van Leeuwen, P. M., and Happee, R. (2012). Effects of motion cueing on curve driving. *Actes INRETS*, pages 11–22.

Dodge, R. (1923). Thresholds of rotation. *Journal of Experimental Psychology*, 6(2):107.

Erlich, I., Venayagamoorthy, G. K., and Worawat, N. (2010). A mean-variance optimization algorithm. In *WCCI 2010 IEEE World Congress on Computational Intelligence*, pages 1–6.

Fang, Z. and Kemeny, A. (2012). Motion cueing algorithms for a real-time automobile driving simulator. In *Driving Simulation Conference*, pages 159–174.

Feenstra, P., Wentink, M., Correia Grácio, B. J., and Bles, W. (2009). Effect of simulator motion space on realism in the Desdemona simulator. In *Proceedings of Driving Simulation Conference*.

Feenstra, P. J., Wentink, M., Roza, Z. C., and Bles, W. (2007). Desdemona, an alternative moving base design for driving simulation. In *Proceedings of the North America-Simulation Driving Conference, Iowa City*.

Fernandez, C. and Goldberg, J. M. (1971). Physiology of peripheral neurons innervating semicircular canals of the squirrel monkey. II. response to sinusoidal stimulation and dynamics of peripheral vestibular system. *J Neurophysiol*, 34(4):661–675.

Fischer, M. (2009). *Motion-Cueing-Algorithmen für eine realitätsnahe Bewegungssimulation*. Ph.D. thesis, Deutsches Zentrum für Luft und Raumfahrt in der Helmholz Gemeinschaft, DLR.

Fischer, M., Sehammer, H., and Palmkvist, G. (2010). Motion cueing for 3-, 6-and 8-degrees-of-freedom motion systems. *Actes INRETS*, pages 121–134.

Gerathewohl, S. J. (1969). Fidelity of simulation and transfer of training: A review of the problem. Department of Transportation, Federal Aviation Administration, Office of Aviation Medicine.

Gibson, J. J. (1979). *The ecological approach to visual perception*. Houghton Miffin, Boston.

Gieszl, E. (2007). Great American scream machine Photo courtesy of ultimate roller-coaster. Website. [Online; accessed 3-June-2015].

Giordano, P. R., Masone, C., Tesch, J., Breidt, M., Pollini, L., and Bülthoff, H. H. (2010). A novel framework for closed-loop robotic motion simulation. Part II: Motion cueing design and experimental validation. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3896–3903. IEEE.

Gouverneur, B., Mulder, J. A., Van Paassen, M. M., Stroosma, O., and Field, E. J. (2003). Optimisation of the Simona research simulator's motion filter settings for handling qualities experiments. In *Proceedings of the AIAA Modelling and Simulation Technologies Conference, Austin (TX), August 11*, Volume 13.

Gower Jr, D. W. and Fowlkes, J. (1989). Simulator sickness in the AH-1S (Cobra) flight simulator. Technical report.

Grant, P., Artz, B., Blommer, M., Cathey, L., and Greenberg, J. (2002). A paired comparison study of simulator motion drive algorithms. *DSC Europe 02 Proceedings*.

Grant, P., Artz, B., Blommer, M., Cathey, L., and Greenberg, J. (2003). Analysing classes of motion drive algorithms based on paired comparison techniques. In *DSC North America, Dearborn, Michigan*.

Grant, P. and Clark, A. (2006). Motion drive algorithm development for a large displacement simulator architecture with redundant degrees of freedom. *DSC Asia/Pacific 06 Proceedings, Tsukuba, Japan*.

Grant, P. R. and Haycock, B. (2008). Effect of jerk and acceleration on the perception of motion strength. *Journal of Aircraft*, 45(4):1190–1197.

Grant, P. R. and Reid, L. D. (1997a). Motion washout filter tuning: Rules and requirements. *Journal of Aircraft*, 34(2):145–151.

Grant, P. R. and Reid, L. D. (1997b). Protest: An expert system for tuning simulator washout filters. *Journal of Aircraft*, 34(2):152–159.

Groen, E. L. and Bles, W. (2003). How to use body tilt for the simulation of linear self motion. *Journal of Vestibular Research: Equilibrium & Orientation*, 14(5):375–385.

Groen, E. L., Smaili, M. H., and Hosman, R. J. (2007). Perception model analysis of flight simulator motion for a Decrab maneuver. *Journal of aircraft*, 44(2):427–435.

Groen, E. L., V. Valenti Clari, M. S., and AW Hosman, R. J. (2001). Evaluation of perceived motion during a simulated takeoff run. *Journal of Aircraft*, 38(4):600–606.

Groen, E. L., Wentink, M., Valente Pais, A., Mulder, M., and Van Paassen, M. M. (2006). Motion perception thresholds in flight simulation. In *Proceedings of the AIAA Modelling and Simulation Technologies Conference & Exhibit, Keystone, CO*.

Guedry, F. E. (1974). Psychophysics of vestibular sensation. In *Handbook of Sensory Physiology: The vestibular system*, Volume VI(2), pages 2–154. Springer.

Gum, D. R. (1973). Modeling of the human force and motion-sensing mechanics. Technical report, Advanced Systems Division, Air Force Human Resources Laboratory, Wright-Patterson AFB, Ohio 45433.

Gundry, A. J. (1978). Thresholds of perception for periodic linear motion. *Aviation, space, and environmental medicine*.

Ham, F. M. and Collins, E. G. (1996). A neurocomputing approach for solving the algebraic matrix Riccati equation. In *Neural Networks, 1996., IEEE International Conference on*, Volume 1, pages 617–622. IEEE.

Harris, L. R., Jenkin, M. R., Zikovitz, D., Redlick, F., Jaekl, P., Jasiobedzka, U. T., Jenkin, H. L., and Allison, R. S. (2002). Simulating self-motion I: Cues for the perception of motion. *Virtual Reality*, 6(2):75–85.

Hess, R. A. (1997). Unified theory for aircraft handling qualities and adverse aircraft-pilot coupling. *Journal of Guidance, Control, and Dynamics*, 20(6):1141–1148.

Hess, R. A. (2008). Obtaining multi-loop pursuit-control pilot models from computer simulation. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 222(2):189–199.

Hess, R. A. and Marchesi, F. (2009). Analytical assessment of flight simulator fidelity using pilot models. *Journal of Guidance, Control, and Dynamics*, 32(3):760–770.

Hettinger, L. J., Nolan, M. D., Kennedy, R. S., Berbaum, K. S., Schnitzius, K. P., and Edinger, K. M. (1987). Visual display factors contributing to simulator sickness. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Volume 31, pages 497–501. SAGE Publications.

Hosman, R., Advani, S., and Haeck, N. (2002). Integrated design of flight simulator motion cueing systems. In *Royal Aeronautical Society Conference on Flight Simulation, London*.

Hosman, R. and Stassen, H. (1999). Pilot's perception in the control of aircraft motions. *Control engineering practice*, 7(11):1421–1428.

Hosman, R. J. A. W. and Van der Vaart, J. C. (1978). Vestibular models and thresholds of motion perception. Results of tests in a flight simulator. Technical report, Delft University of Technology.

Hwang, T. S., Yeh, S. K., Lin, J. R., and Su, W. P. (2009). Adaptive motion washout filter design by using self-tuning fuzzy control. In *Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on*, pages 811–815. IEEE.

Ish-Shalom, J. (1982). *Design of optimal motion for flight simulators*. PhD thesis, Massachusetts Institute of Technology.

Jamson, A. H. J. (2010). *Motion cueing in driving simulators for research applications*. PhD thesis, University of Leeds.

Jamson, H. (2007). Driving me round the bend – Behavioural studies using the new University of Leeds driving simulator. In *2nd Motion Simulator Conference, Braunschweig, Germany*.

Jazar, R. N. (2010). *Theory of applied robotics: Kinematics, dynamics, and control*. Springer Science & Business Media.

Johnson, D. M. (2005). Introduction to and review of simulator sickness research. Technical report, U.S. Army Research Institute for the Behavioral and Social Sciences.

Kandel, E. R., Schwartz, J. H., Jessell, T. M., Siegelbaum, S. A., and Hudspeth, A. J. (2000). *Principles of neural science*, Volume 4. McGraw-hill New York.

Kecskeméthy, A. (2002). MOBILE 1.3 User's Guide. Technical report, Lab of Mechanics und Robotics, University of Duisburg Essen.

Kecskeméthy, A. and Tändl, M. (2009). Physiologisch optimierte Interfaces für die hochdynamische Bewegungsgenerierung in physikalischen Simulatorsystemen. Technical report, Lab of Mechanics und Robotics, University of Duisburg Essen.

Kennedy, R. S., Berbaum, K. S., Allgood, G. O., Lane, N. E., Lilienthal, M. G., and Baltzley, D. R. (1988). Etiological significance of equipment features and pilot history in simulator sickness. *AGARD, Motion Cues in Flight Simulation and Simulator Induced Sickness 22 p(SEE N 89-12171 03-52)*.

Kennedy, R. S., Berbaum, K. S., Lilienthal, M. G., Dunlap, W. P., and Mulligan, B. E. (1987). Guidelines for alleviation of simulator sickness symptomatology. Technical report, Naval training systems center orlando FL.

Kennedy, R. S. and Fowlkes, J. E. (1992). Simulator sickness is polygenic and polysymptomatic: Implications for research. *The International Journal of Aviation Psychology*, 2(1):23–38.

Key, D. L. (1980). Fidelity of simulation for pilot training. *AGARD Advisory Report*, 159.

Kleinman, D. L., Baron, S., and Levison, W. H. (1970). An optimal control model of human response part I: Theory and validation. *Automatica*, 6(3):357–369.

Kolasinski, E. M. (1995). Simulator sickness in virtual environments. Technical report, Army research inst for the behavioral and social sciences Alexandria VA.

Kuipers, J. (2014). Multi variable strategy reduces symptoms of simulator sickness. In *Proceedings of Measuring Behavior 2014*.

Kwakernaak, H. and Sivan, R. (1972). *Linear optimal control systems*, Volume 1. Wiley-interscience New York.

Lackner, J. R. (1977). Induction of illusory self-rotation and nystagmus by a rotating sound-field. *Aviation, Space, and Environmental Medicine*, pages 129–131.

Lampton, D. R., Kraemer, R. E., Kolasinski, E. M., and Knerr, B. W. (1995). An investigation of simulator sickness in a tank driver trainer. Technical report, (ARI Research Rep. 1684). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Lane, R. L., Ousland, J. G., and Abrass, I. B. (1989). *Essentials of clinical geriatrics*. McGraw Hill.

Larsson, P., Västfjäll, D., and Kleiner, M. (2004). Perception of self-motion and presence in auditory virtual environments. In *Proceedings of seventh annual workshop presence*, pages 252–258.

Lone, M. M. and Cooke, A. K. (2010). Review of pilot modelling techniques. In *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, pages 2010–297.

Mach, E. (1875). *Grundlinien der Lehre von den Bewegungsempfindungen*. W. Engelmann.

Mayne, R. (1974). A systems concept of the vestibular organs. In *Vestibular System Part 2: Psychophysics, Applied Aspects and General Interpretations*, pages 493–580. Springer.

Mayrhofer, M., Langwallner, B., Schlüsselberger, R. Â· Bles, W., and Wentink, M. (2007). An innovative optimal control approach for the next generation simulator motion platform Desdemona.

McCauley, M. E. (1984). *Research issues in simulator sickness: Proceedings of a workshop*. National Academies.

McGuiness, J., Bouwman, J. H., and Forbes, J. M. (1981). *Simulator Sickness Occurrences in the 2E6 Air Combat Maneuvering Simulator (ACMS)*. Person-Systems Integration.

McRuer, D. T. and Krendel, E. S. (1974). Mathematical models of human pilot behavior. Technical report, Advisory group for aerospace research and developement Neuilly-sur-Seine (France).

Meiry, J. L. (1965). *The vestibular system and human dynamic space orientation.* PhD thesis, Massachusetts Institute of Technology.

Mesland, B. S. M., Bles, W., and Wertheim, A. H., G. E. L. (1998). The influence of expectation on the perception of linear horizontal motion. Technical report, TM-98-A010. Soesterberg, TNO Human Factors Research Institute.

Miller, J. W. and Goodson, J. E. (1958). A note concerning motion sickness in the 2FH2 hover trainer. *Pensacola, FL: Naval School of Aviation Medicine*.

more3d GmbH & Co. KG (accessed 10-Dec-2016]).

Nahon, M. A. and Reid, L. D. (1990). Simulator motion-drive algorithms — A designer's perspective. *Journal of Guidance, Control, and Dynamics*, 13(2):356–362.

Nahon, M. A., Reid, L. D., and Kirdeikis, J. (1992). Adaptive simulator motion software with supervisory control. *Journal of Guidance, Control, and Dynamics*, 15(2):376–383.

Naseri, A. and Grant, P. (2005). An improved adaptive motion drive algorithm. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, page 6500.

Naseri, A. and Grant, P. R. (2011). Difference thresholds: Measurement and modeling. In *AIAA modeling and simulation technologies conference exhibition*, pages 1–10.

Nehaoua, L., Mohellebi, H., Amouri, A., Arioui, H., Espié, S., and Kheddar, A. (2008). Design and control of a small-clearance driving simulator. *Vehicular Technology, IEEE Transactions on*, 57(2):736–746.

Nesti, A., Masone, C., Barnett-Cowan, M., Giordano, P. R., Bülthoff, H., and Pretto, P. (2012). Roll rate thresholds and perceived realism in driving simulation. *Actes INRETS*, pages 23–31.

Nordmark, S., Jansson, H., Palmkvist, G., and Sehammar, H. (2004). The new VTI driving simulator. Multi purpose moving base with high performance linear motion. In *Conférence simulation de conduite*, pages 45–55.

NTSC (1988). Simulator sickness field manual Mod 3. Technical report.

Parrish, R. V., Dieudonne, J. E., and Martin Jr, D. J. (1975). Coordinated adaptive washout for motion simulators. *Journal of Aircraft*, 12(1):44–50.

Pausch, R., Crea, T., and Conway, M. (1992). A literature survey for virtual environments: Military flight simulator visual systems and simulator sickness. *Presence: Teleoperators & Virtual Environments*, 1(3):344–363.

Pekar, J. and Havlena, V. (2004). Design and analysis of model predictive control using MPT toolbox. In *url: http://dsp. vscht. cz/konference matlab/matlab04/pekar. pdf.*

Pham, D. A. (2014). Trajectory and limits of the LMR driving simulator based on KUKA Robocoaster, Internal technical report, Duisburg Essen University.

Pham, D. A. (2015). Review of parameter tuning method for MCAs, Internal technical report, Duisburg Essen University.

Pouliot, N. A., Gosselin, C. M., and Nahon, M. A. (1998). Motion simulation capabilities of three-degree-of-freedom flight simulators. *Journal of Aircraft*, 35(1):9–17.

Reason, J. T. (1970). Motion sickness: A special case of sensory rearrangement. *Advancement of science*, 26(130):386.

Reason, J. T. and Brand, J. J. (1975). *Motion sickness.* Academic press.

Reid, L. D. and Nahon, M. A. (1985). Flight simulation motion-base drive algorithms. Part 1: Developing and testing the equations. Technical report, Institute for Aerospace Studies, University of Toronto.

Reid, L. D. and Nahon, M. A. (1986). Flight simulation motion-base drive algorithms: Part 2. Selecting the system parameters. Technical report, University of Toronto.

Reid, L. D. and Nahon, M. A. (1988). Response of airline pilots to variations in flight simulator motion algorithms. *Journal of Aircraft*, 25(7):639–646.

Reymond, G. and Kemeny, A. (2000). Motion cueing in the Renault driving simulator. *Vehicle System Dynamics*, 34(4):249–259.

Reymond, G., Kemeny, A., Droulez, J., and Berthoz, A. (1999). Contribution of a motion platform to kinesthetic restitution in a driving simulator. In *Driving simulation conference*, pages 123–136.

Romano, R. (2003). Non-linear optimal tilt coordination for washout algorithms. In *2003 AIAA Modeling and Simulation Technologies Conference and Exhibit*.

Sammet, T. (2007). Motion-Cueing-Algorithmen für die Fahrsimulation.

Savona, F., Stratulat, A. M., Diaz, E., Honnet, V., Houze, G., Vars, P., Masfrand, S., Roussarie, V., and Bourdin, C. (2017). The influence of lateral, roll and yaw motion gains on driving performance on an advanced dynamic simulator.

Schroeder, J. A. (1999). *Helicopter flight simulation motion platform requirements*. National Aeronautics and Space Administration, Ames Research Center.

Schulte-Pelkum, J. (2007). Perception of self-motion: Vection experiments in multisensory virtual environments.

Schweig, S. and Kammers, H. (2011). Bewegungssteuerung eines Robocoaster KUKA Roboters zur Ride Simulation mit Hilfe von Washout Filtern. Project report, University of Duisburg Essen, Duisburg, Germany.

Shao, H., Guan, L., Wang, J., Wang, L., and Fu, Y. (2009). An optimal force cueing algorithm for dynamic seat. In *Control and Automation, 2009. ICCA 2009. IEEE International Conference on.IEEE*, pages 2217–2222.

Shrout, P. E. and Fleiss, J. L. (1979). Intraclass correlations: Uses in assessing rater reliability. *Psychological bulletin*, 86(2):420.

Silverman, D. R. and Slaughter, R. A. (1995). An exploration of simulator sickness in the MH-60G operational flight trainer, An advanced wide field-of-view helicopter trainer. Technical report, Mesa, AZ: Aircrew Training Research Division. Human Resources Directorate.

Sinacori, J. B. (1977). The determination of some requirements for a helicopter flight research simulation facility.

Sivan, R., Ish-Shalom, J., and Huang, J. K. (1982). An optimal control approach to the design of moving flight simulators. *Systems, Man and Cybernetics, IEEE Transactions on*, 12(6):818–827.

Slob, J. J. (2008). State-of-the-art driving simulators: A literature survey. *DCT Report*, 107.

Song, J., Jung, U., and Ko, H. (2003). Washout algorithm with fuzzy-based tuning for a motion simulator. *KSME international journal*, 17(2):221–229.

Soyka, F., Giordano, P., Beykirch, K., and Bülthoff, H. H. (2011). Predicting direction detection thresholds for arbitrary translational acceleration profiles in the horizontal plane. *Experimental brain research*, 209(1):95–107.

Soyka, F., Teufel, H. J., Beykirch, K. A., Giordano, P. R., Butler, J. S., Nieuwenhuizen, F. M., and Bülthoff, H. H. (2009). Does jerk have to be considered in linear motion simulation? In *Proceedings of the AIAA modeling and simulation technologies conference and exhibit, Chicago (IL), AIAA-2009-6245*.

Steinhausen, W. (1931). Über den Nachweis der Bewegung der Cupula in der intakten Bogengangsampulle des Labyrinthes bei der natürlichen rotatorischen und calorischen Reizung. *Pflüger's Archiv für die gesamte Physiologie des Menschen und der Tiere*, 228(1):322–328.

Stewart, D. (1965). A platform with six degrees of freedom. *Proceedings of the institution of mechanical engineers*, 180(1):371–386.

Stoffregen, T. A., Hettinger, L. J., Haas, M. W., Roe, M. M., and Smart, L. J. (2000). Postural instability and motion sickness in a fixed-base flight simulator. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 42(3):458–469.

Stoffregen, T. A. and Smart, L. J. (1998). Postural instability precedes motion sickness. *Brain research bulletin*, 47(5):437–448.

Stratulat, A. M., Roussarie, V., Vercher, J., and Bourdin, C. (2011). Does tilt/translation ratio affect perception of deceleration in driving simulators? *Journal of Vestibular Research*, 21(3):127.

Teghtsoonian, R. (1971). On the exponents in Stevens' law and the constant in Ekman's law.

Telban, R. J. and Cardullo, F. M. (2001). An integrated model of human motion perception with visual-vestibular interaction. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, page 4249.

Telban, R. J., Cardullo, F. M., and Houck, J. A. (2005). Motion cueing algorithm development: Human-centered linear and nonlinear approaches. Technical report, NASA/CR-2005-213747.

Tortora, G. J. and Derrickson, B. H. (2008). *Principles of Anatomy and Physiology*. John Wiley & Sons, New Jersey, USA.

Valente Pais, A., Wentink, M., van Paassen, M., and Mulder, M. (2009). Comparison of three motion cueing algorithms for curve driving in an urban environment. *Presence*, 18(3):200–221.

Valente Pais, A. R., Mulder, M., Van Paassen, M. M., Wentink, M., and Groen, E. (2006). Modeling human perceptual thresholds in self-motion perception. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*.

Van der Steen, H. (1998). *Self-Motion Perception*. PhD thesis, Delft University of Technology: Delft, The Netherlands.

Wang, L. (2009). *Model Predictive Control System Design and Implementation using MATLAB®*. Springer.

Wang, S. and Fu, L. (2004). Predictive washout filter design for VR-based motion simulator. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, Volume 7, pages 6291–6295. IEEE.

Wentink, M., Bles, W., Hosman, R., and Mayrhofer, M. (2005). Design & evaluation of spherical washout algorithm for Desdemona simulator. *Proc. of AIAA Modeling and Simulation Technologies*.

Wentink, M., Valente Pais, R., Mayrhofer, M., Feenstra, P., and Bles, W. (2008). First curve driving experiments in the Desdemona simulator.

White, A. D. and Rodchenko, V. V. (1999). Motion fidelity criteria based on human perception and performance. In *Proceedings of the AIAA Modelling and Simulation Technologies Conference, Portland (OR), August 9*, Volume 11, pages 485–493.

Wright, R. H. (1995). Helicopter simulator sickness: A state-of-the-art review of its incidence, causes, and treatment. *Alexandria, VA: US Army Research Institute for the Behavioral and Social Sciences*.

Young, L. R. (1978). Visually induced motion in flight simulation. In *AGARD Symposium on Flight Simulation*. Brussels, Belgium.

Young, L. R. (2003). Spatial orientation. *Principles and practice of aviation psychology*, pages 69–113.

Young, L. R., Henn, V., and Scherberger, H. (2001). Fundamentals of the theory of movement perception by Dr. Ernst Mach.

Young, L. R. and Meiry, J. L. (1968). A revised dynamic otolith model. *Aerospace medicine*, 39(6):606–608.

Zacharias, G. L. (1978). Motion Cue Models for Pilot-Vehicle Analysis. Technical report, AMRL-TR-78-2, Department of Defense, Alexandria, VA.

Zaichik, L. E., Rodchenko, V., Rufov, I. V., Yashin, Y. P., and White, A. D. (1999). Acceleration perception. In *AIAA Modeling and Simulation Technologies Conference and Exhibit, Portland, OR*, pages 512–520.

Zaichik, L. E., Yashin, Y. P., Desyatnik, P., and Smaili, H. (2012). Some aspects of upset recovering simulation on hexapod simulators. In *AIAA Modeling and Simulation Technologies Conference*, page 4949.

Zaichik, L. E., Yashin, Y. P., and Desyatnik, P. A. (2009). Motion fidelity criteria for large-amplitude tasks. *AIAA Paper*, (2009-5916).

Zaichik, L. E., Yashin, Y. P., and Desyatnik, P. A. (2010). Peculiarities of motion cueing for precision control tasks and maneuvers. *ICAS. Nice, France. ICAS Paper*, (602).

Zeyada, Y. and Hess, R. A. (2000). Modeling human pilot cue utilization with applications to simulator fidelity assessment. *Journal of aircraft*, 37(4):588–597.

Zywiol, H. J. and Romano, R. (2003). Motion drive algorithms and simulator design to study motion effects on infantry soldiers. In *Driving Simulation Conference, North America 2003 (DSC-NA 2003).*

Appendices

A Miscellaneous Fundamental Information

A.1 Effect of a filter frame in CL algorithm

Reid and Nahon (1985) introduced two options of selecting reference frames, $\mathcal{K}_S$ or $\mathcal{K}_0$, for implementing washout process. Fig. A.1, and A.2 show in detail the effect of the washout process in each reference frame to the simulated quantities.

For example, if there is only lateral acceleration input, the center of rotation is at the driver's head ($\mathcal{K}_S \equiv \mathcal{K}_{Ps}$), and the washout filters are implemented in the frame $\mathcal{K}_S$ (Fig. A.1) that is oriented an non-zero roll angle with respect to $\mathcal{K}_0$ at this time. The target specific force ${}^{V}\underline{f}_{V}$ is reproduced completely by the desired acceleration $\underline{a}'_S$ and $-{}^{0}\underline{g}$. The Y-axis low-pass component of the desired acceleration ${}^{S}a'_{Sy}$ is through the Tilt coordinator to generate additional rotation angle, $\Delta\varphi_S$ regarding the current orientation of $\mathcal{K}_S$. On the other hand, after the simulated acceleration $\underline{a}'_S$ is filtered by the high-pass filters and translated to the inertial frame $\mathcal{K}_0$, the residual part of the simulated acceleration in Z_0 axis, ${}^{0}a_{szHP}$, can produce the vertical movement, that cause the simulator violation with the vertical physical limits.

In contrast, when the washout filters are implemented in the frame $\mathcal{K}_0$ (Fig. A.2), three components of the desire acceleration ${}^{0}\underline{a}'_S$ are filtered separately, thus, with the suitable parameters the vertical high-frequency component ${}^{0}\underline{a}'_{SzHP}$ can be completely remove, and only the ${}^{0}\underline{a}'_{SyHP}$ produce the lateral motion. However, the vertical component of the simulated specific force is reduced a small amount, because it is reproduced only by a part of the gravity acceleration $\underline{g}$.

A.2 Real Time Solution of the Riccati Equation

Solving the nonlinear Riccati Eq. 4.65 is required at each time step. Telban et al. (2005) use the structured neural network developed by Ham and Collins (1996) that finds the $\mathbf{P}(k+1)$ from previous solution of $\mathbf{P}(k)$ as equation

$$\mathbf{P}(k+1) = \mathbf{P}(k) + \mu\Delta\mathbf{P}(k), \tag{A.1}$$

with $\mu > 0$ is the learning rate parameter and the term $\Delta\mathbf{P}$ given as

$$\Delta\mathbf{P}(k) = \left[\mathbf{A}'_{\eta}(k)\underline{e}^{*}(k)\underline{z}(k) + \underline{e}^{*}(k)\underline{z}^{\mathrm{T}}(k)\mathbf{A}'_{\eta}(k) - \underline{e}^{*}(k)\underline{p}^{\mathrm{T}}(k)\tilde{\mathbf{S}}\right], \tag{A.2}$$

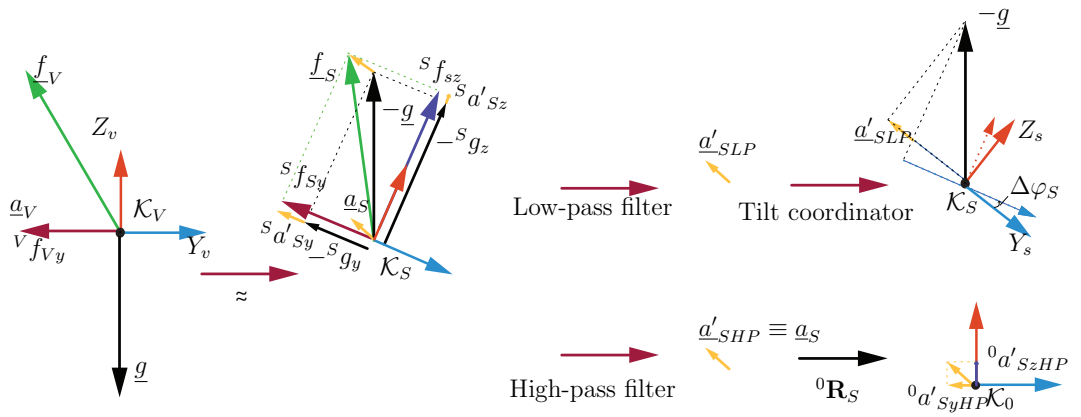


Figure A.1: Washout filters in the frame $\mathcal{K}_S$

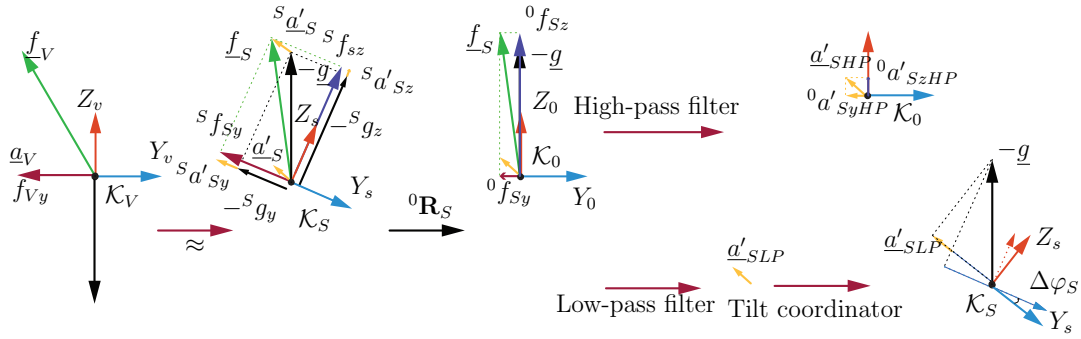


Figure A.2: Washout filters in the frame $\mathcal{K}_0$

where $\mathbf{A}'_\eta = \mathbf{A}' + \eta\mathbf{I}$, and $\tilde{\mathbf{S}} = \mathbf{B}^{\mathrm{T}}\mathbf{R}_{\mathbf{2}}^{-\mathbf{1}}\mathbf{B}$. Moreover, $\underline{z}(k)$ is the excitatory signal at the k time step, and $\underline{p}(k) = \mathbf{P}(k)\underline{z}(k)$. The error signal $\underline{e}^*(k)$ is given as

$$\mathbf{e}^*(k) = \left[\mathbf{P}(k)\tilde{\mathbf{S}}\mathbf{P}(k) - \mathbf{A}'^{\mathrm{T}}_{\eta}\mathbf{P}(k) - \mathbf{P}\mathbf{A}'_{\eta} - \mathbf{R}'_1\right] \tag{A.3}$$

Ham and Collins (1996) noted that the external excitatory vector input signals $\underline{z}(k)$ are a set of linearly independent bi-polar vectors given as

$$\underline{z}^i = [z_{11} z_{12} \cdots z_{1n}]\ , i = 1 \cdot n\,, \tag{A.4}$$

here $z_{1j} = -1$ if $\{j = 1 \cdots n\,, j \neq i\}$ and $z_{1i} = 1$.

A.3 Nonlinear scale

The nonlinear scale block used in the OpT, OpTNon algorithms is the three-order polynomial curve (cubic Hermite spline), defining the relate between input magnitude

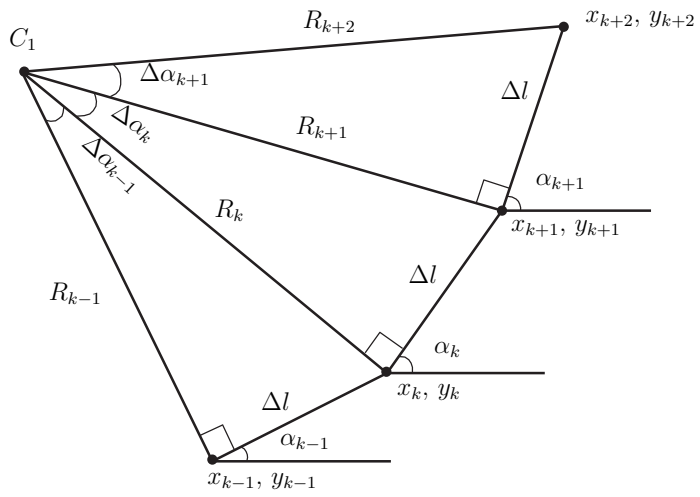


Figure A.3: S-curve construction

a and output magnitude b, that is expressed as

$$b = c_3a^3 + c_2a^2 + c_1a + c_0, \tag{A.5}$$

where,

$$c_0 = 0\,;\ c_1 = s_0\,;\ c_2 = a_{max}^{-2}(3\,b_{max} - 2\,s_0a_{max} - s_1a_{max})\,; \tag{A.6}$$

$$c_3 = a_{max}^{-3}(s_0\,a_{max} - 2\,b_{max} + s_1a_{max}). \tag{A.7}$$

Here, s_0 and s_1 are the slopes at $a = 0$, and $a = a_{max}$. And a_{max} and b_{max} are expected maximum input and output magnitude, respectively.

A.4 Planar S-curve trajectory construction

The experiment was implemented with the simulation task in which a virtual roller coaster run with a constant amplitude $v = 3.6(m/s^2)$ along a planar S-curve and has only a lateral acceleration a_y depending on time. If it is assumed that there are only two center $C_1\,, C_2$ of rotation being both sides of the curve, and the S-curve trajectory is approximated by multiple small straight segments Δl A.3. Due to the constant amplitude of the velocity, $\Delta l = vT_s$, here, $T_s = 0.0125\,(s)$ is the sample time of the simulation. Due to small $\Delta\alpha_k$, it can be calculated approximately as Eq. A.8

$$\Delta\alpha_k = \Delta l/R_{k+1} \quad ;\alpha_{k+1} = \alpha_k + \Delta\alpha_k \tag{A.8}$$

The position of the point in the S-curve, $(x_{k+1},\ y_{k+1})$, was computed by recursively

$$x_{k+1} = x_k + \Delta l \cdot \cos\alpha_k \quad ;y_{k+1} = y_k + \Delta l \cdot \sin\alpha_k\,; \tag{A.9}$$

B Parameters and Response of MCAs

In the section 5.4.3 the input acceleration is scaled from the target acceleration by the factor $k = 0.4$. All MCAs with the tuned parameters have the response satisfying the highest level of fidelity index MF. In this part, to describe more clearly the capability of the reviewed MCAs the response of the MCAs with higher amplitude (higher scale factor $k \in FSset = \{0.5, 0.6, 0.7, 0.8, 0.9\}$ used) are illustrated from Fig. B.49 to B.240 and all the fidelity index MF are shown from Tab. B.11 to B.22. In the tables, the violating the restricted conditions is presented by 0, and the satisfying these conditions is represented by 1. Referred to the response data, it can be seen that if the parameters are kept, various types false cues and physical violation appear accompanying with the higher scale factor applied.

	$A_0(s^{-1})$	$B_0(s^{-1})$	$B_1(s^{-1})$	$G_O(s^2/m)$	$\delta_{Oto}(m/s^2)$
OpS	0.076	0.19	-	2.16	0.47
OpRN	0.076	0.19	-	5.86	0.17
OpTYM	0.076	0.19	1.5	8.82	0.17
OpT	0.1	0.2	62.5	294.12	0.17
OpTNon	0.1	0.2	-	294.12	0.17

Note: '-': the parameter is not used.

Table B.1: The otolith parameters (sway direction)

	$\tau_1(s)$	$\tau_2(s)$	$\tau_a(s)$	$\tau_l(s)$	$G_S(s/rad)$	$\delta_{Scc}(deg/s)$
OpS	5.9	0.003	-	-	$233^{\$}$	1.45^{*}
OpRN	6.1	0.1	-	-	118.55	3
OpTYM	5.73	0.005	80	0.06	28.6479	2
OpT	5.73	0.005	80	0.06	28.6479	2
OpTNon	5.73	-	80	-	28.6479	2

Note: '*': the unit is (deg/s^2) and '$': the unit is (s^2/rad) .

Table B.2: Semicircular parameters (roll angle)

	$\underline{x}_c$				$\underline{u}_S$	
	$\iiint a_{Sy}dt^3$	$\iint a_{Sy}dt^2$	$\int a_{Sy}dt$	φ_S	u_1	u_2
OpS	-	+	+	-	a_{Sy}	φ_S
OpRN	+	+	+	-	φ_S	a_{Sy}
OpTYM	+	+	+	+	$\dot{\varphi}_S$	a_{Sy}
OpT	+	+	+	+	$\dot{\varphi}_S$	a_{Sy}
OpTNon	+	+	+	+	$\dot{\varphi}_S$	a_{Sy}

Note: '-': the parameter is not used; '+': the parameter is used.

Table B.3: Dynamic states used in the optimal washout algorithms

	CLRN		CLG					
	HP	LP	HP_α	LP_α	HP_r	LP_r	HP_z	LP_z
Order(-)	3	2	3	3	3	3	3	3
$\omega_n(rad/s)$	0.001	0.005	0.03	0.03	0.6	0.6	0.6	0.6
$\zeta(kg/s)$	1.4	1.4	1.4	1.4	1	1	1	1
$\omega_b(rad/s)$	0.01	-	0.9	0.9	20	20	20	20

Table B.4: Parameters of washout filters CLRN and CLG

G_1	G_2	G_3	p_{10}	p_{20}	p_{30}	ρ	k_{y1}	k_{y2}
0.47	0.00579	0.0108	1	-0.121	0.7	1	0.0001	0.04
w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	-
0.06	0.002	0.0001	0	0	0.194	17.26	46.45	-

Table B.5: Parameters of ADRN

K_0	ζ_0	$\omega_{n0}\,(s^{-1})$	$\omega_b\,(s^{-1})$	w_a	w_k	w_ζ	w_ω	w_v	w_s
0.9	1.4	0.03	0.9	5	10	65	75	30	1.5

Table B.6: Parameters of in ADSK

	q_1	q_2	r_1	r_2	rc_1	rc_2	rc_3	rc_4	γ_1	γ_2
OpS	60	1	0.377	0.999	-	0.00177	0.0477	-	0.2	20
OpRN	0.25	0.8	0.33	0.2	0.0009	0.00001	0.00001	-	0.1	0.2
OpTYM	5	42	1	20	0.0008	0.001	0.0001	5	0.1	0.2
OpT	1	20	1	20	0.005	0.005	1	10	1	1.2

Table B.7: Weighting parameters Optimal washout filters

	$\gamma\,(s^{-1})$	$c\,(s^{-1})$	q_1	q_2	r_1	r_2
ZyRo	20	1.25	199	0.05	0.01	69

Table B.8: Parameters of ZyRo

	s_i	r_i	q_i	p	m
MPC	0.5 | 60	1 | 5	199 | 4.5 | 55 | 175	400	200

Table B.9: Weighting values of MPC*

Model		Limit Values		Law brake variances	
$T_s\,(s)$	0.125	$[S_{y\min},\, S_{y\max}]\,(m)$	$[-10.86,\, 10.86]$	$c_u\,(-)$	0.45
$\{q_{yi}\}\,(-)$	$\{7,\, 1,\, 79\}$	$[v_{y\min},\, v_{y\max}]$ m/s	$[-100,\, 100]$	$c_v\,(-)$	1
$\{r_i\}$ (-)	$\{1,\, 3\}$	$[a_{y\min},\, a_{y\max}]\,(m/s^2)$	$[-14,\, 14]$	$T_1\,(s)$	2
$\{q_{xi}\}$ (-)	$\{7,\, 1,\, 1,\, 1\}$	$[\varphi_{\max},\, \varphi_{\max}]\,(rad)$	$[-0.5,\, 0.5]$	$T_2\,(s)$	2
N (-)	4	$[\omega_{\min},\, \omega_{\max}]\,(rad/s)$	$[-0.1,\, 0.1]$	-	-
Nc (-)	4	$[\dot{\omega}_{\max},\, \dot{\omega}_{\max}]\,(rad/s^2)$	$[-0.2,\, 0.2]$	-	-

Table B.10: MPC parameters tuned for the LMR driving simulator

Response of MCA	CLRN	$k = 0.4$

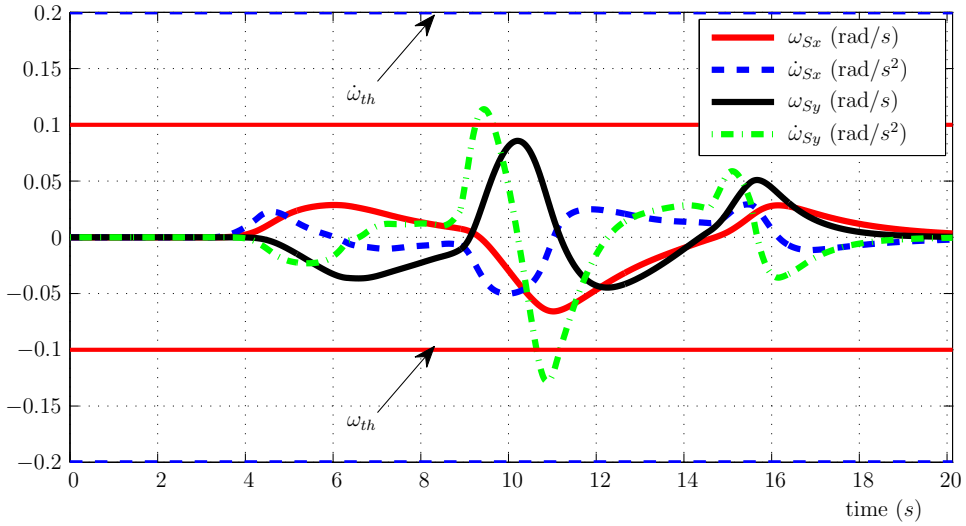


Figure B.1: CLRN - Simulated rotational quantities

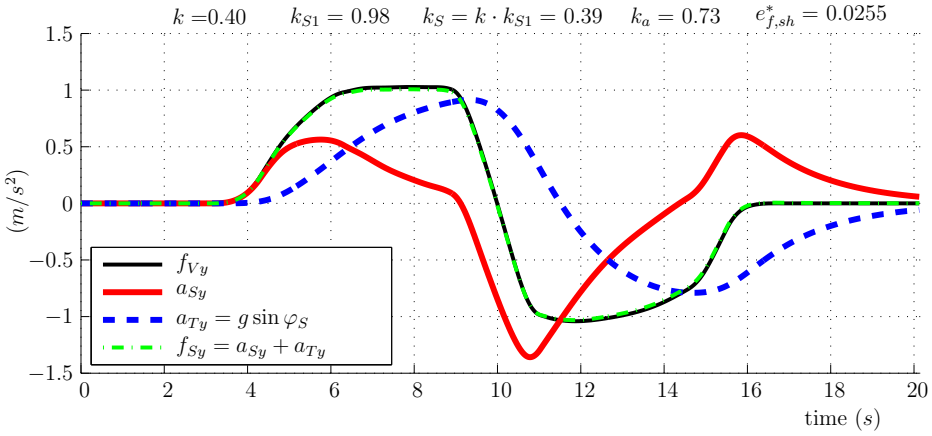


Figure B.2: CLRN - Simulated specific forces and acceleration

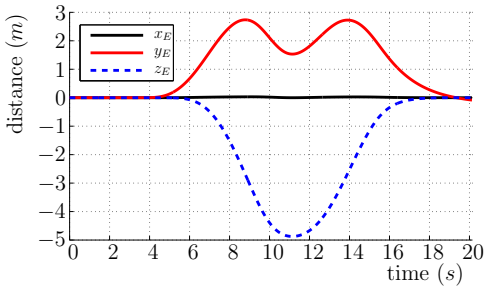


Figure B.3: CLRN - Position

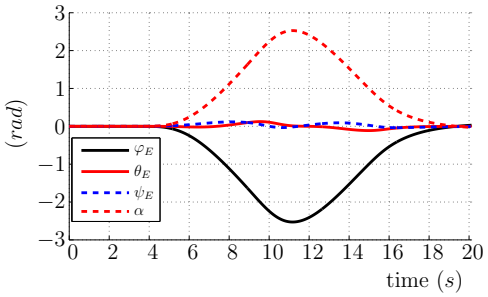


Figure B.4: CLRN - Euler angle

Response of MCA	CLG	$k = 0.4$

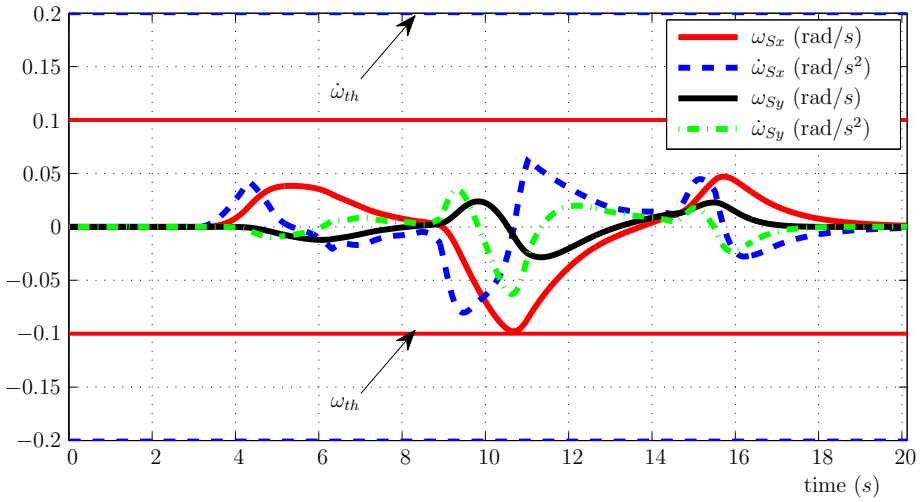


Figure B.5: CLG - Simulated rotational quantities

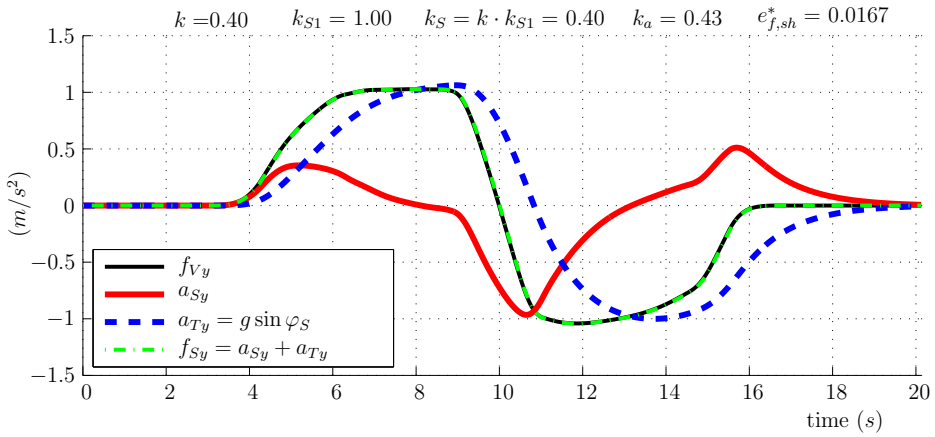


Figure B.6: CLG - Simulated specific forces and acceleration

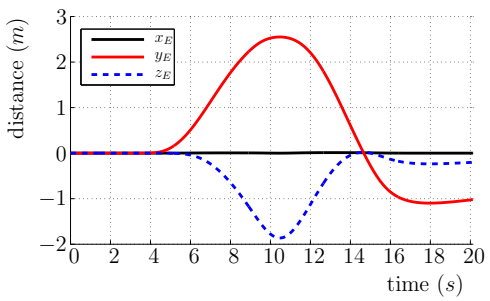


Figure B.7: CLG - Position

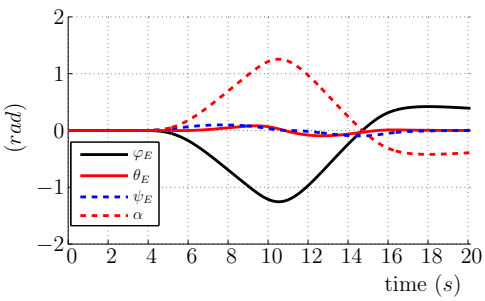


Figure B.8: CLG - Euler angle

Response of MCA	ADRN	$k = 0.4$

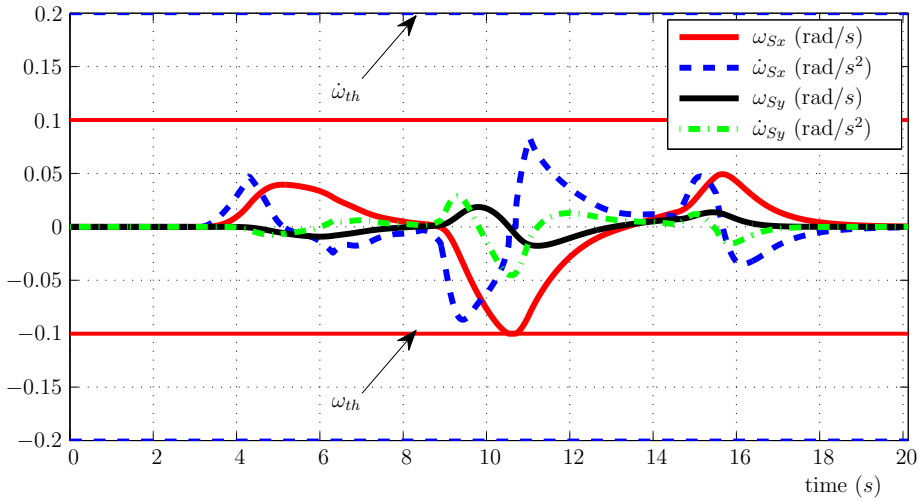


Figure B.9: ADRN - Simulated rotational quantities

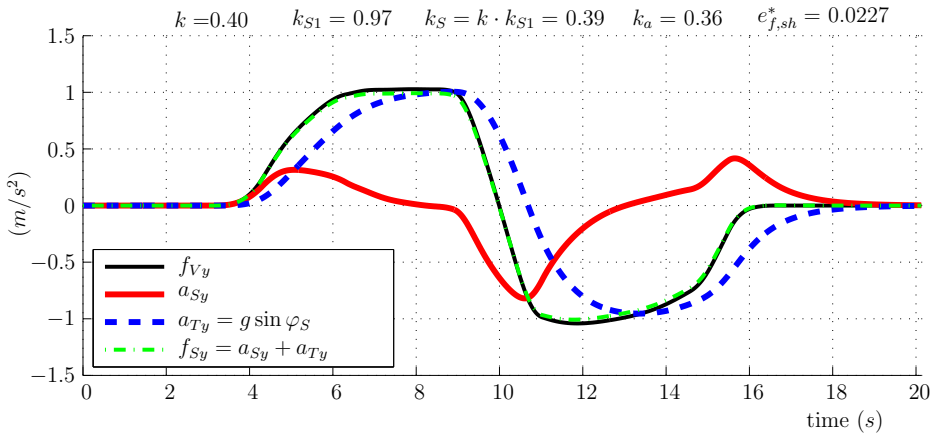


Figure B.10: ADRN - Simulated specific forces and acceleration

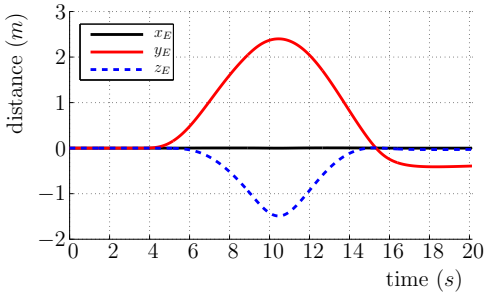


Figure B.11: ADRN - Position

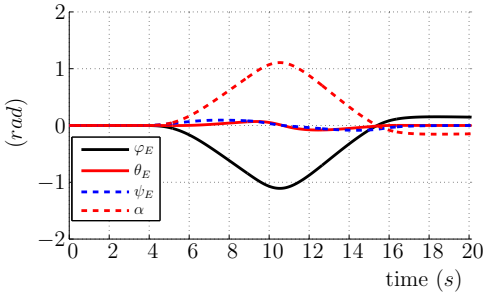


Figure B.12: ADRN - Euler angle

Response of MCA	ADSK	$k = 0.4$

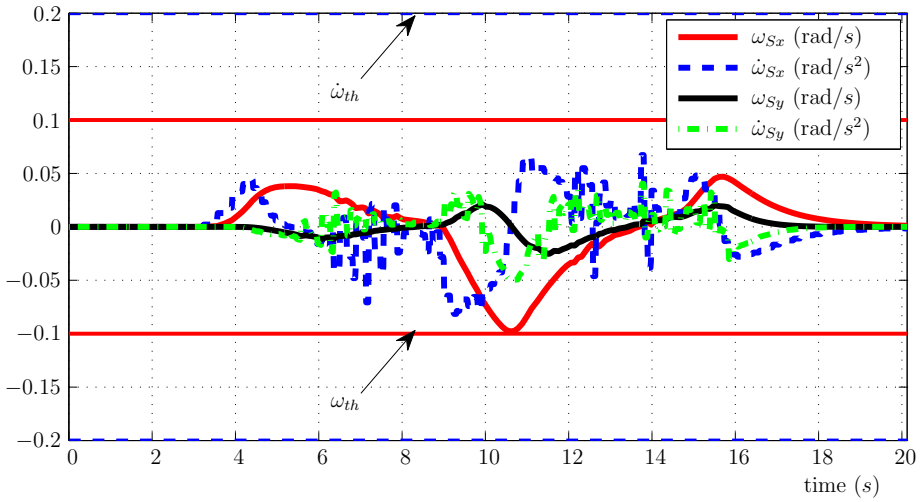


Figure B.13: ADSK - Simulated rotational quantities

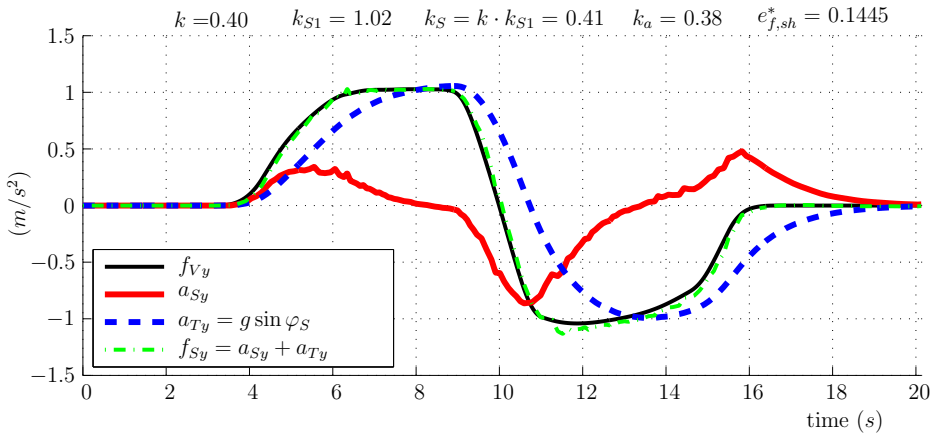


Figure B.14: ADSK - Simulated specific forces and acceleration

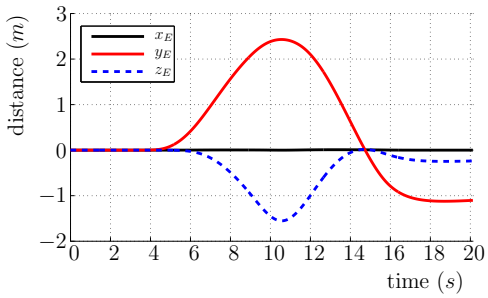


Figure B.15: ADSK - Position

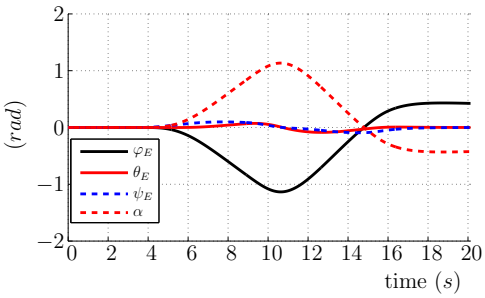


Figure B.16: ADSK - Euler angle

Response of MCA	OpTYM	$k = 0.4$

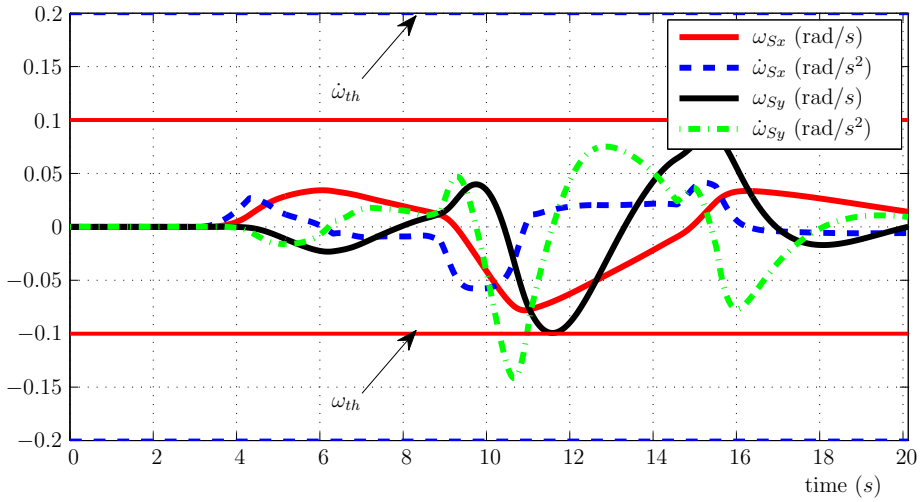


Figure B.17: OpS - Simulated rotational quantities

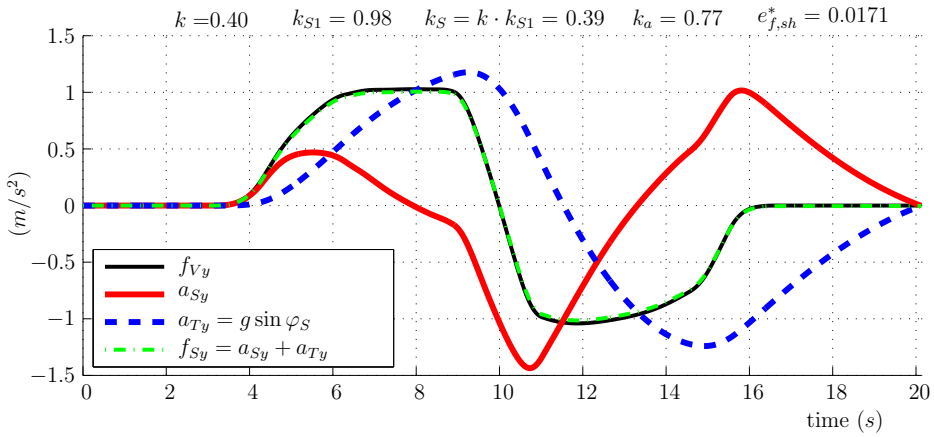


Figure B.18: OpS - Simulated specific forces and acceleration

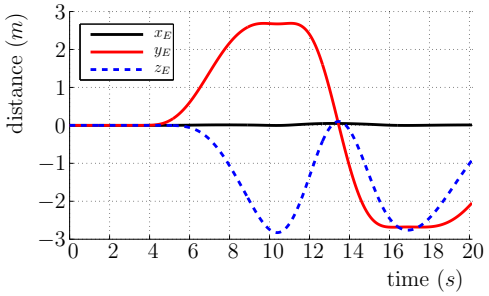


Figure B.19: OpS - Position

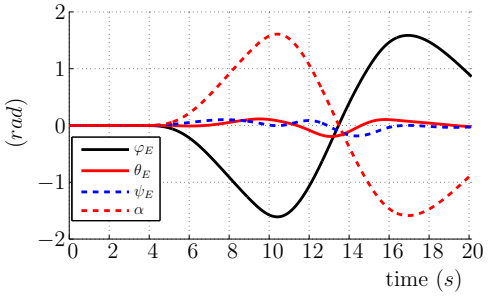


Figure B.20: OpS - Euler angle

Response of MCA	OpT	$k = 0.4$

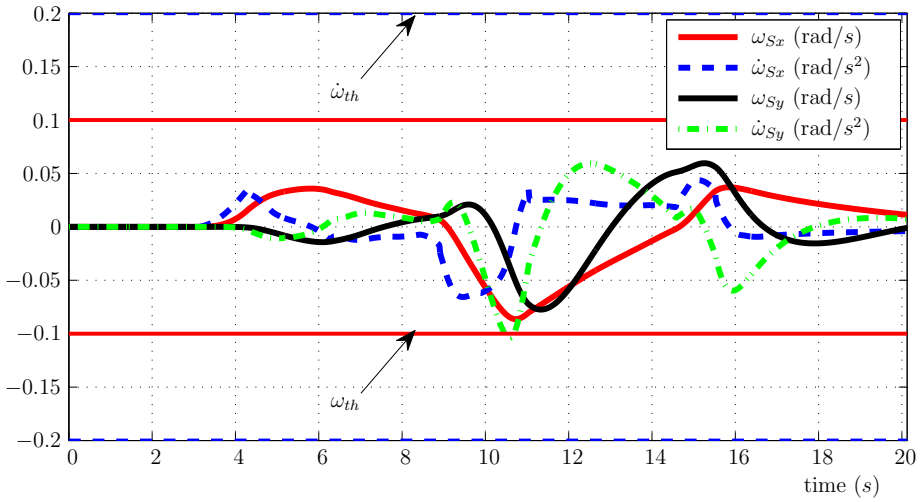


Figure B.21: OpRN - Simulated rotational quantities

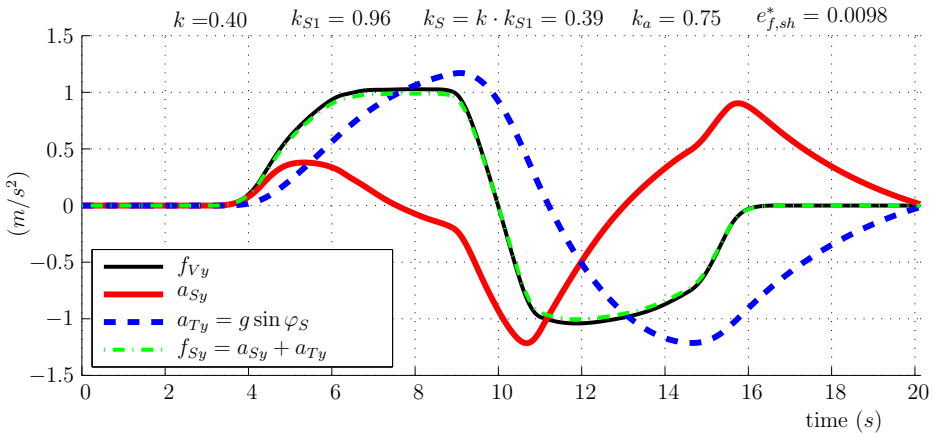


Figure B.22: OpRN - Simulated specific forces and acceleration

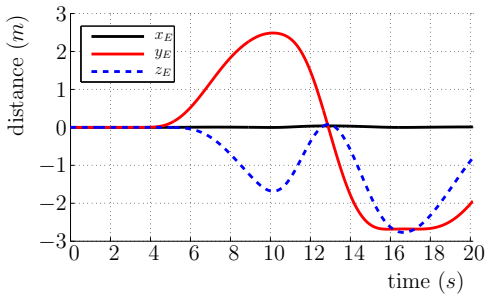


Figure B.23: OpRN - Position

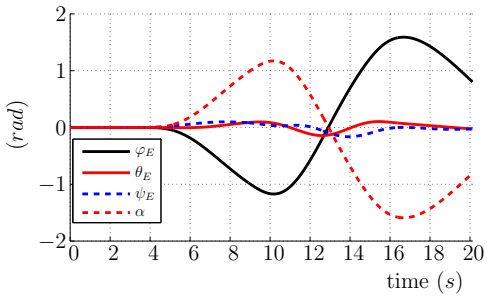


Figure B.24: OpRN - Euler angle

Response of MCA	OpS	$k = 0.4$

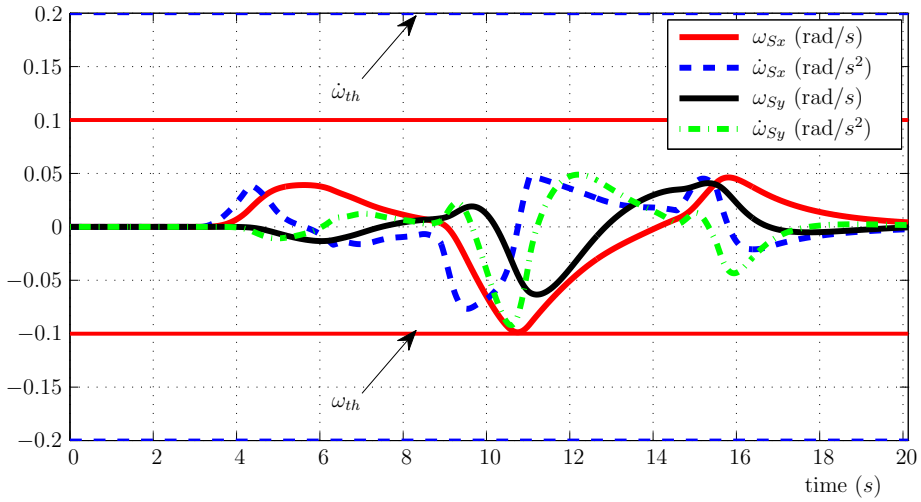


Figure B.25: OpTYM - Simulated rotational quantities

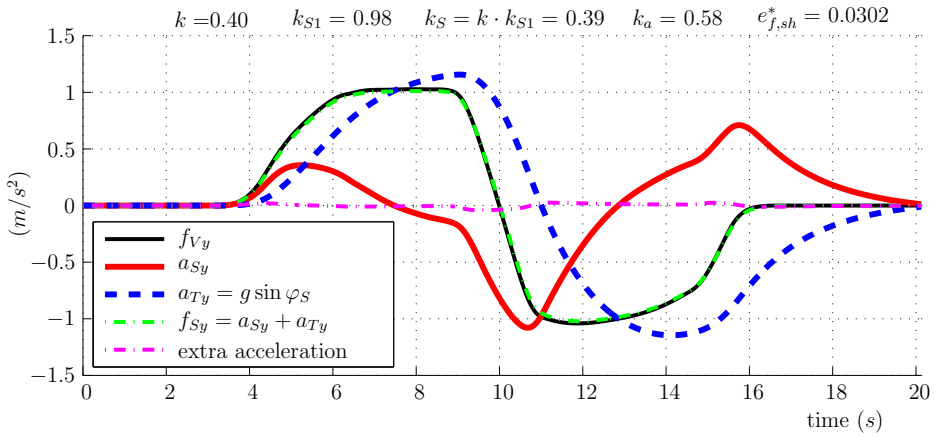


Figure B.26: OpTYM - Simulated specific forces and acceleration

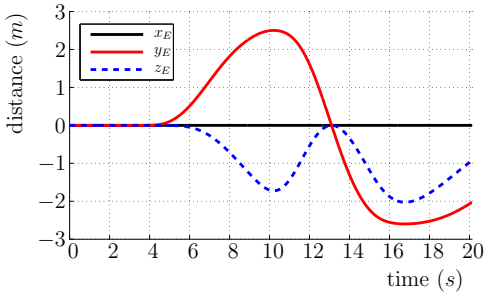


Figure B.27: OpTYM - Position

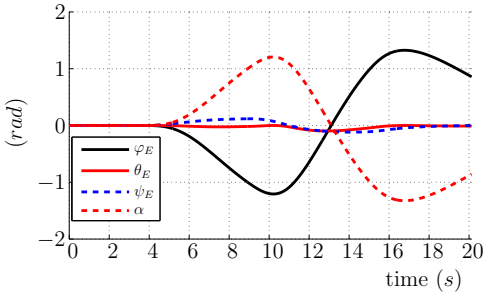


Figure B.28: OpTYM - Euler angle

Response of MCA	OpRN	$k = 0.4$

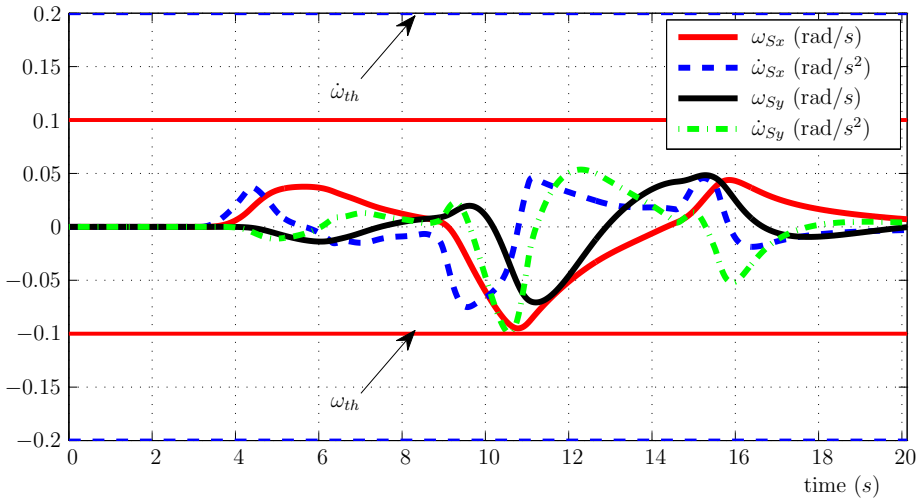


Figure B.29: OpT - Simulated rotational quantities

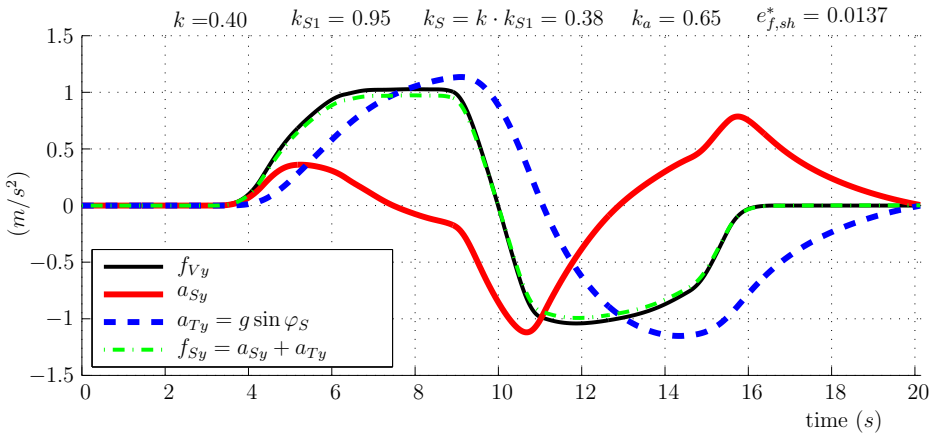


Figure B.30: OpT - Simulated specific forces and acceleration

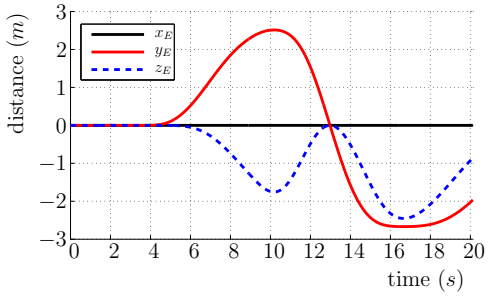


Figure B.31: OpT - Position

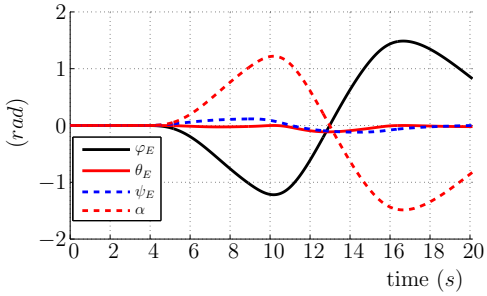


Figure B.32: OpT - Euler angle

Response of MCA	OpTNon	$k = 0.4$

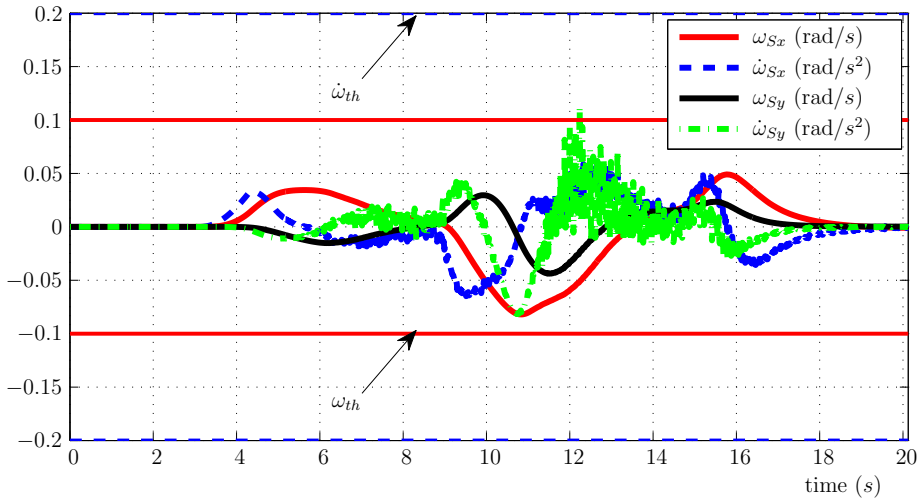


Figure B.33: OpTN - Simulated rotational quantities

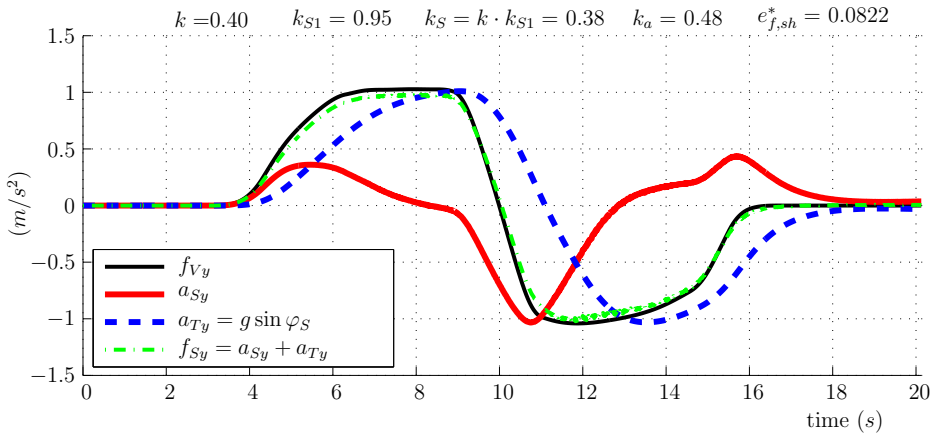


Figure B.34: OpTN - Simulated specific forces and acceleration

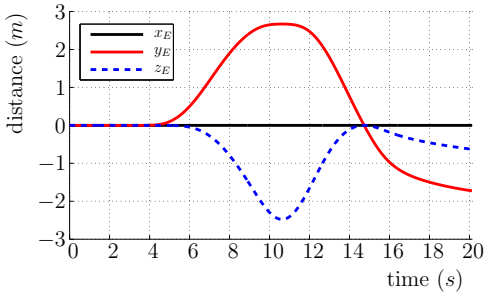


Figure B.35: OpTN - Position

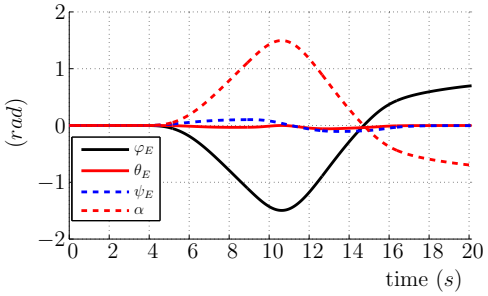


Figure B.36: OpTN - Euler angle

Response of MCA	ZyRo	$k = 0.4$

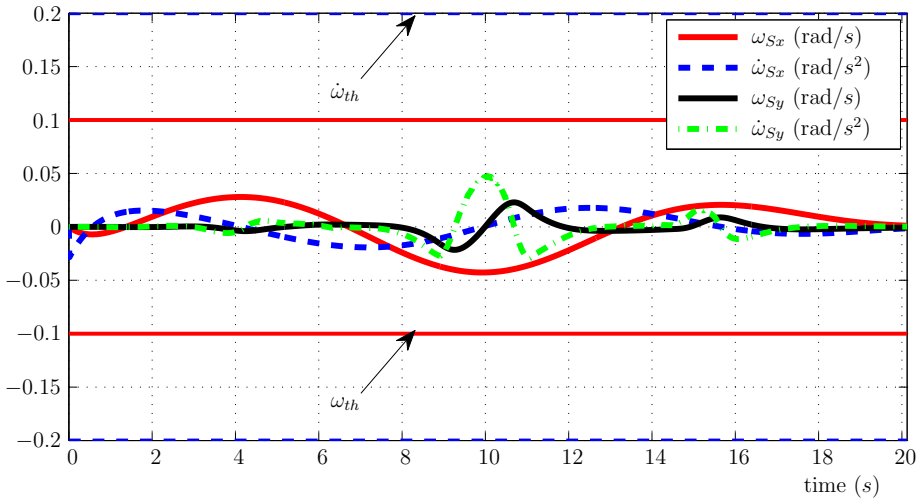


Figure B.37: ZyRo - Simulated rotational quantities

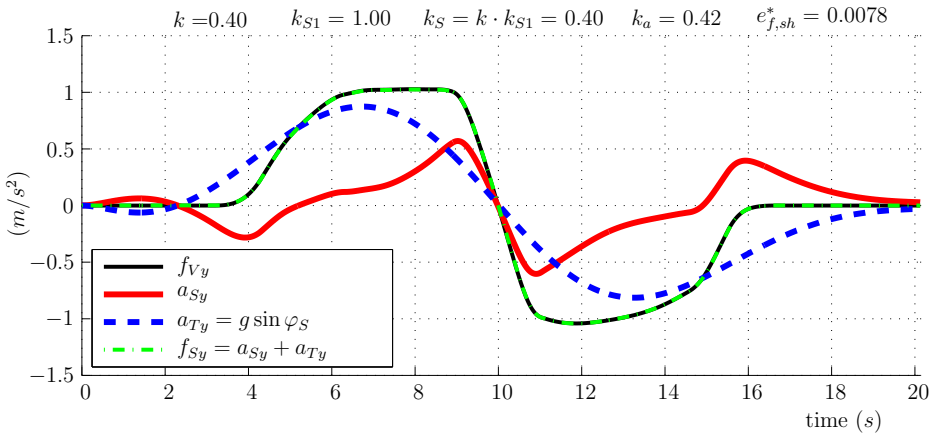


Figure B.38: ZyRo - Simulated specific forces and acceleration

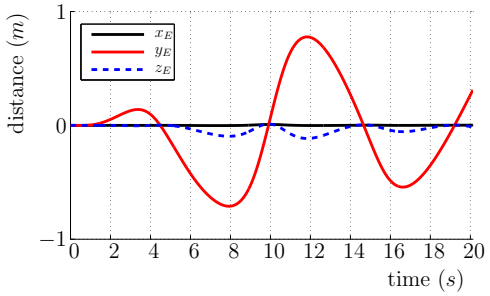


Figure B.39: ZyRo - Position

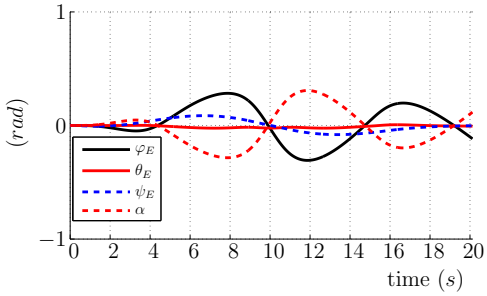


Figure B.40: ZyRo - Euler angle

Response of MCA	MPC*	$k = 0.4$

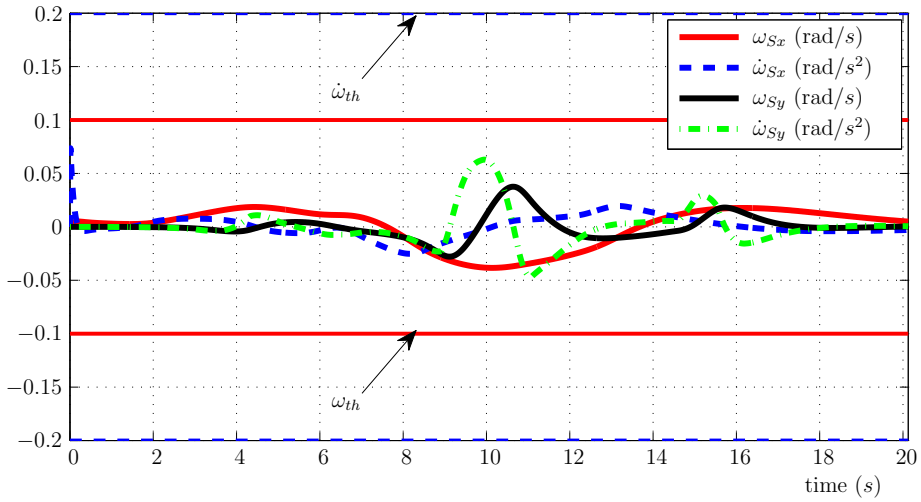


Figure B.41: MPC - Simulated rotational quantities

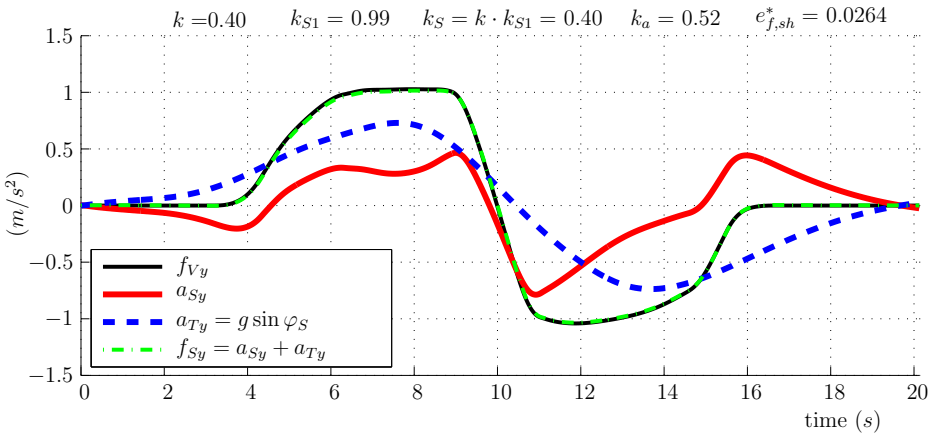


Figure B.42: MPC - Simulated specific forces and acceleration

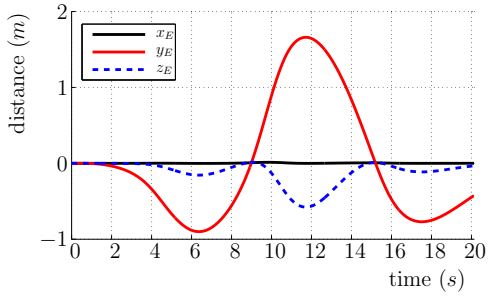


Figure B.43: MPC - Position

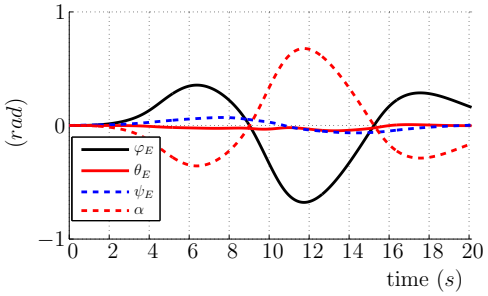


Figure B.44: MPC - Euler angle

Response of MCA	exMPCFK	$k = 0.4$

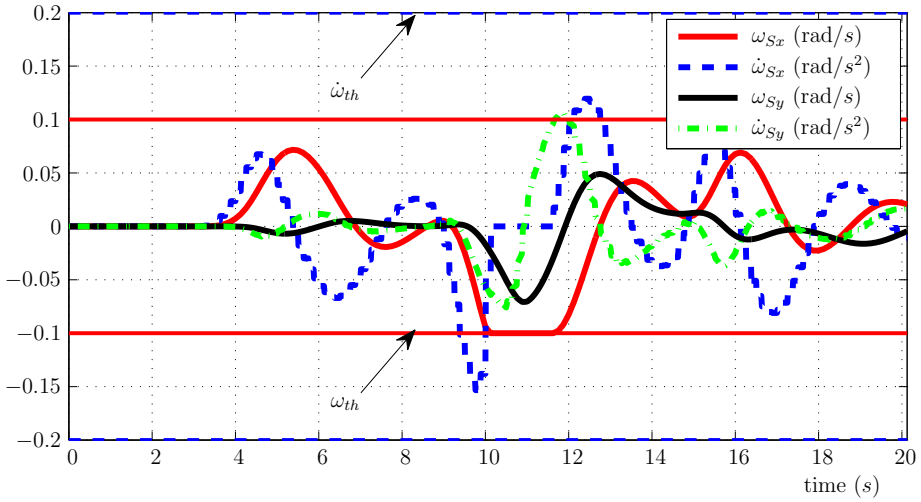


Figure B.45: exMPCFK - Simulated rotational quantities

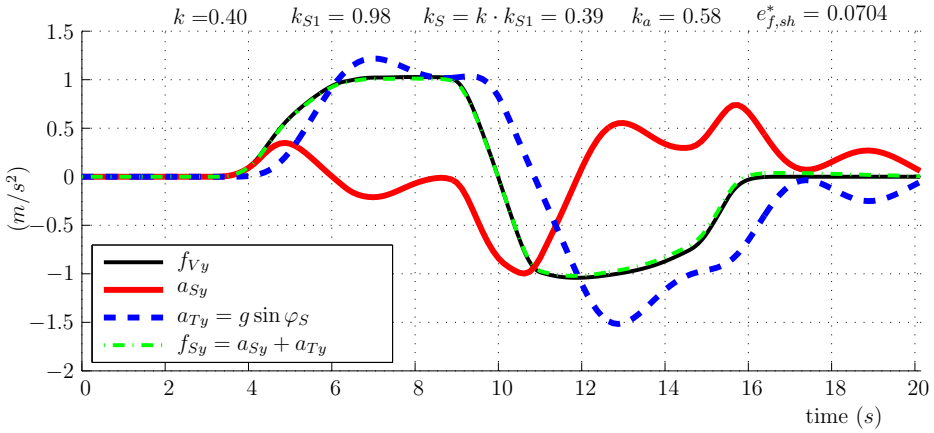


Figure B.46: exMPCFK - Simulated specific forces and acceleration

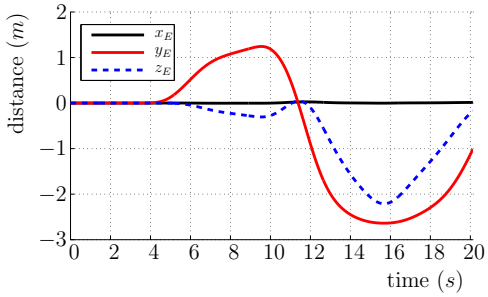


Figure B.47: exMPCFK - Position

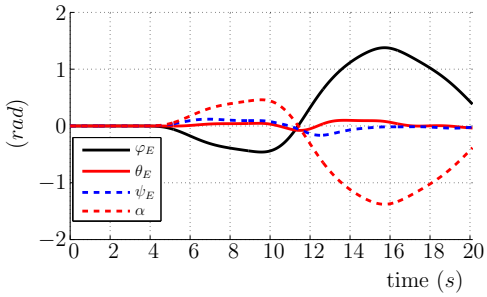


Figure B.48: exMPCFK - Euler angle

Response of MCA	CLRN	$k = 0.5$

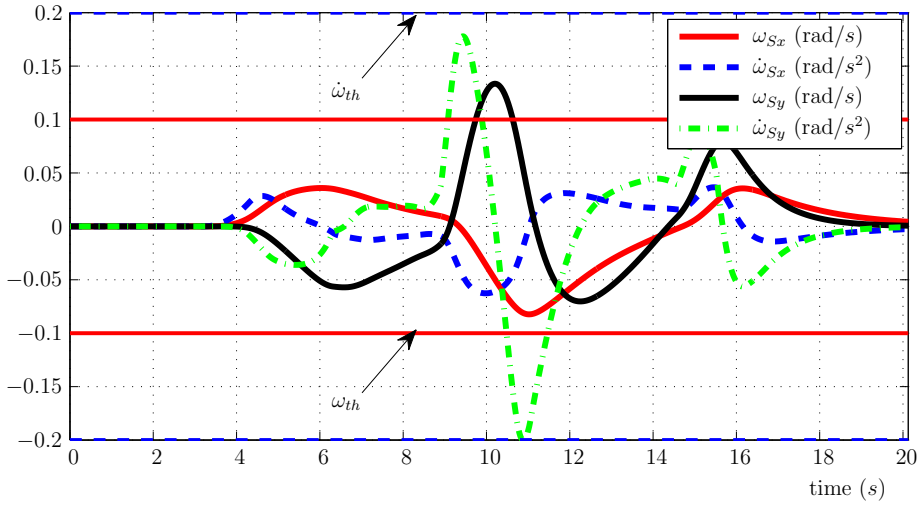


Figure B.49: CLRN - Simulated rotational quantities

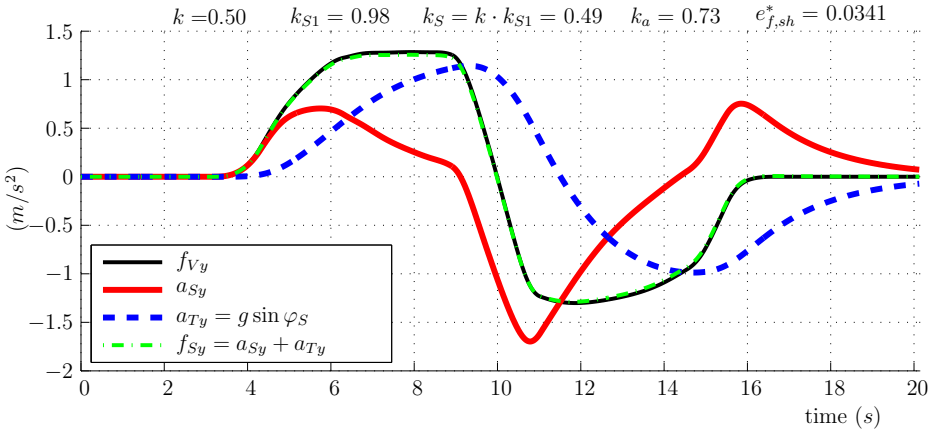


Figure B.50: CLRN - Simulated specific forces and acceleration

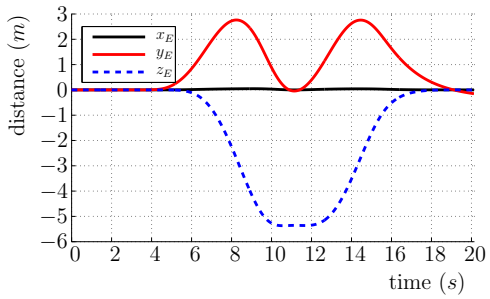


Figure B.51: CLRN - Position

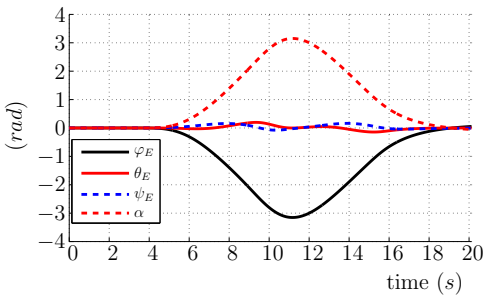


Figure B.52: CLRN - Euler angle

Response of MCA	CLG	$k = 0.5$

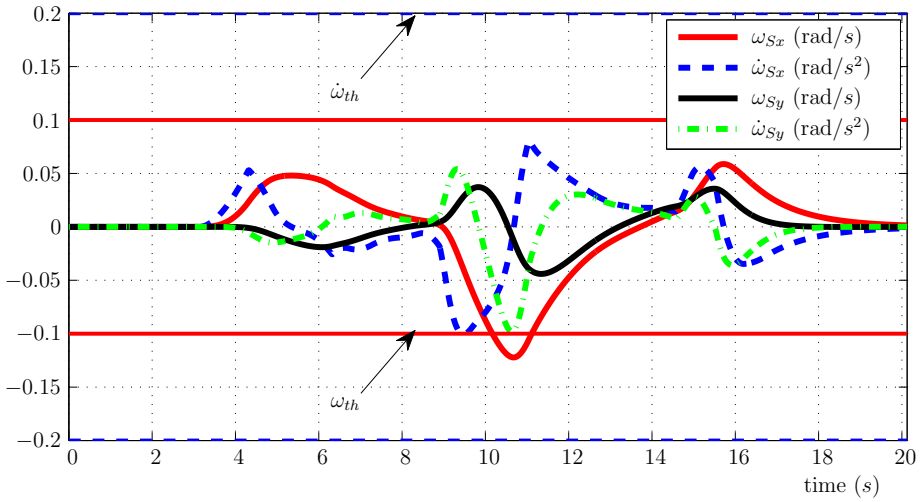


Figure B.53: CLG - Simulated rotational quantities

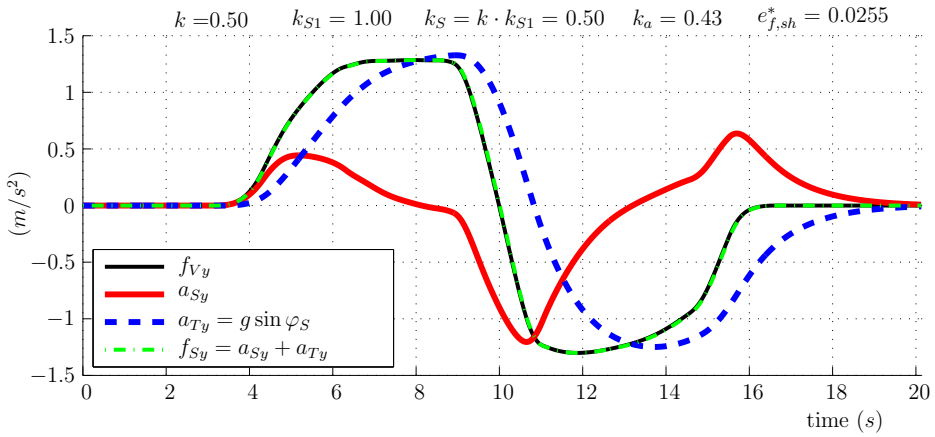


Figure B.54: CLG - Simulated specific forces and acceleration

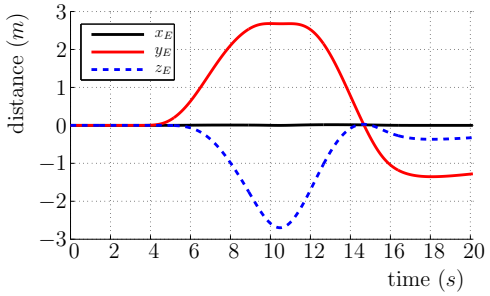


Figure B.55: CLG - Position

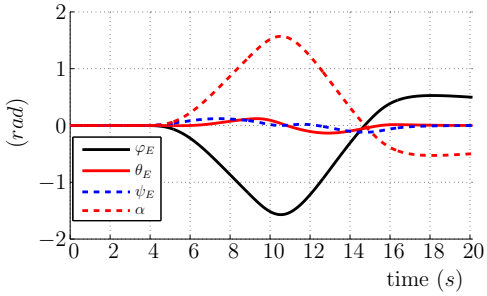


Figure B.56: CLG - Euler angle

Response of MCA	ADRN	$k = 0.5$

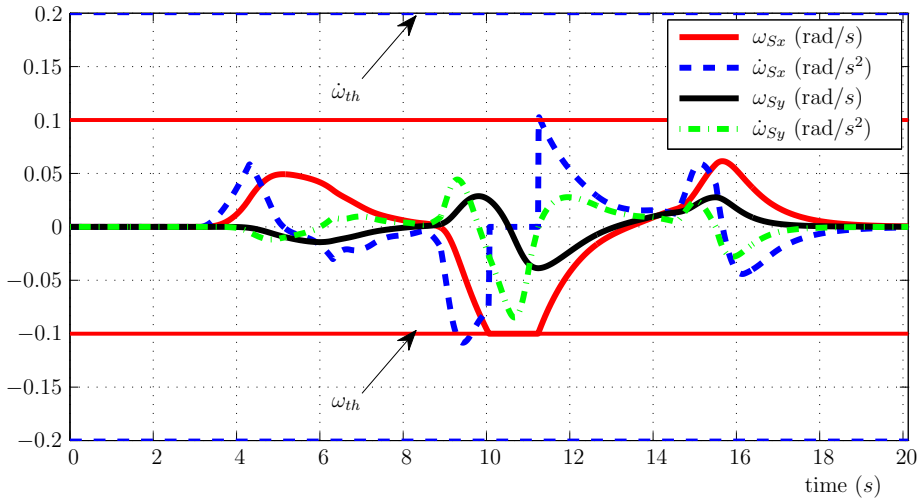


Figure B.57: ADRN - Simulated rotational quantities

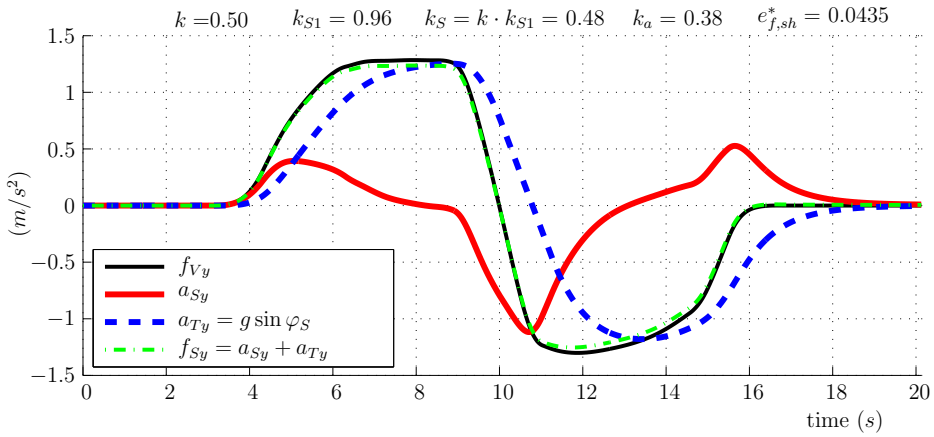


Figure B.58: ADRN - Simulated specific forces and acceleration

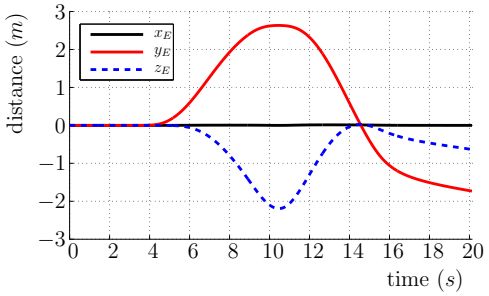


Figure B.59: ADRN - Position

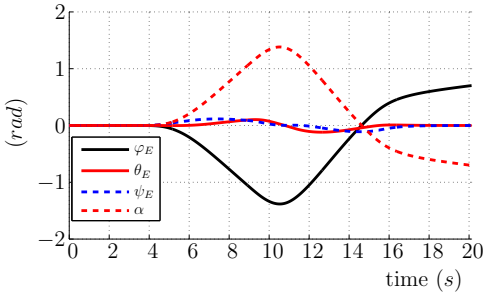


Figure B.60: ADRN - Euler angle

Response of MCA	ADSK	$k = 0.5$

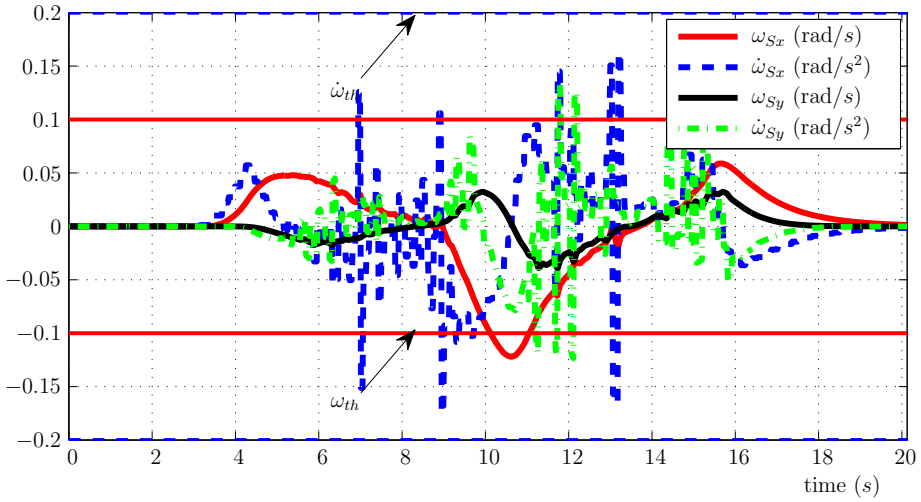


Figure B.61: ADSK - Simulated rotational quantities

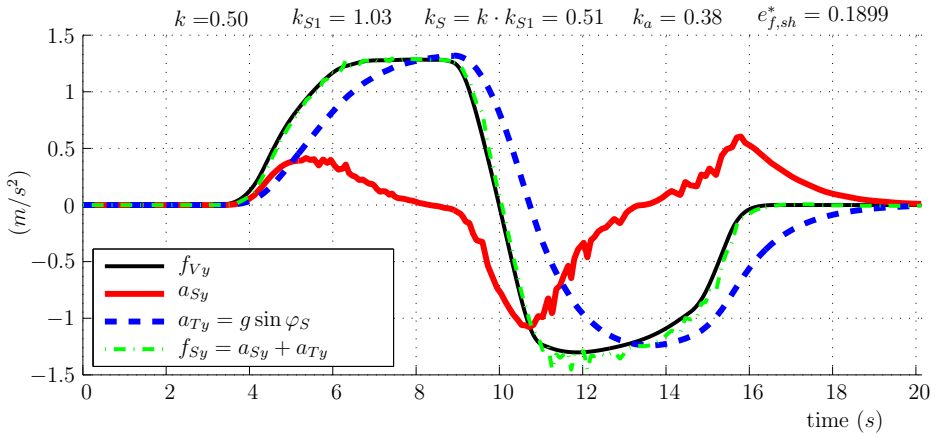


Figure B.62: ADSK - Simulated specific forces and acceleration

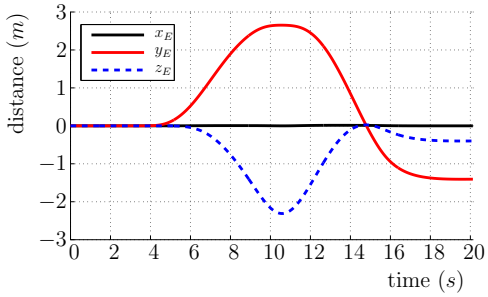


Figure B.63: ADSK - Position

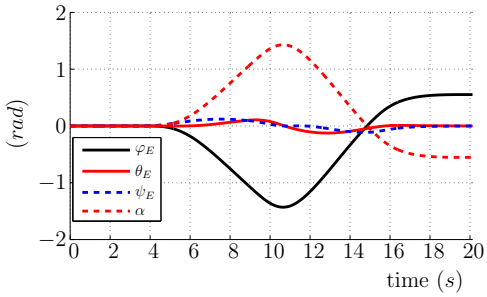


Figure B.64: ADSK - Euler angle

Response of MCA	OpTYM	$k = 0.5$

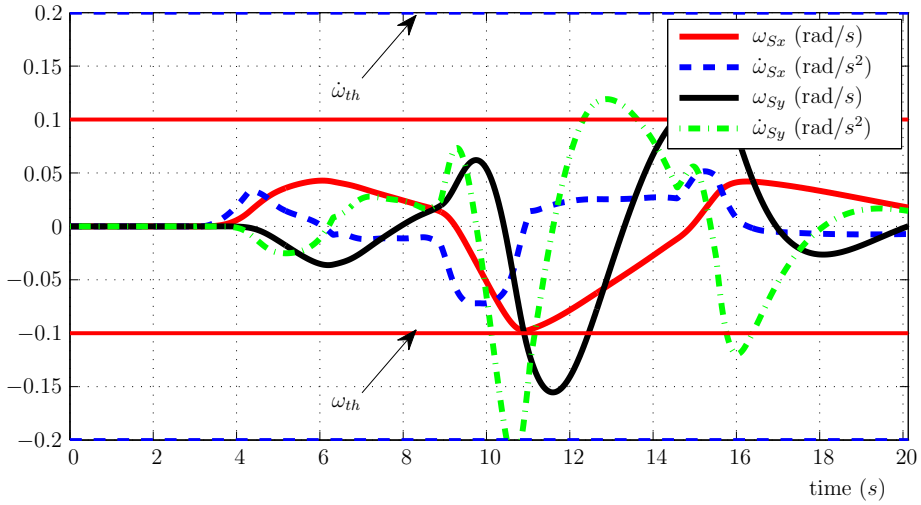


Figure B.65: OpS - Simulated rotational quantities

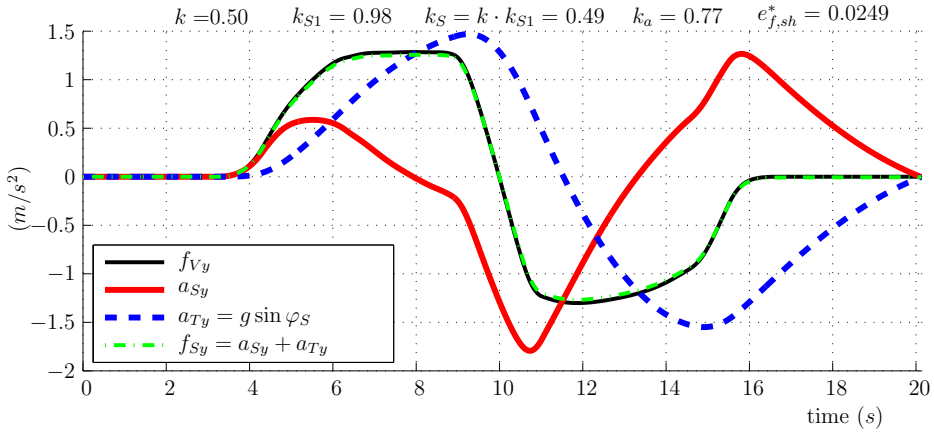


Figure B.66: OpS - Simulated specific forces and acceleration

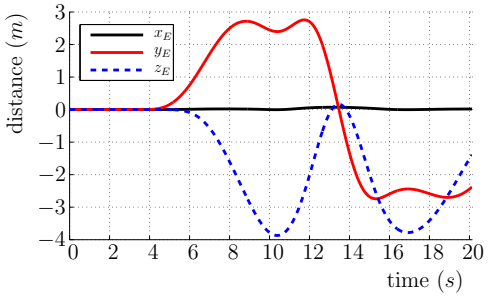


Figure B.67: OpS - Position

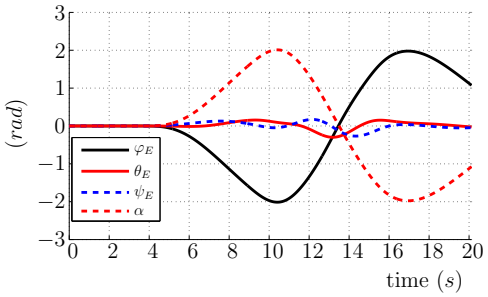


Figure B.68: OpS - Euler angle

Response of MCA	OpT	$k = 0.5$

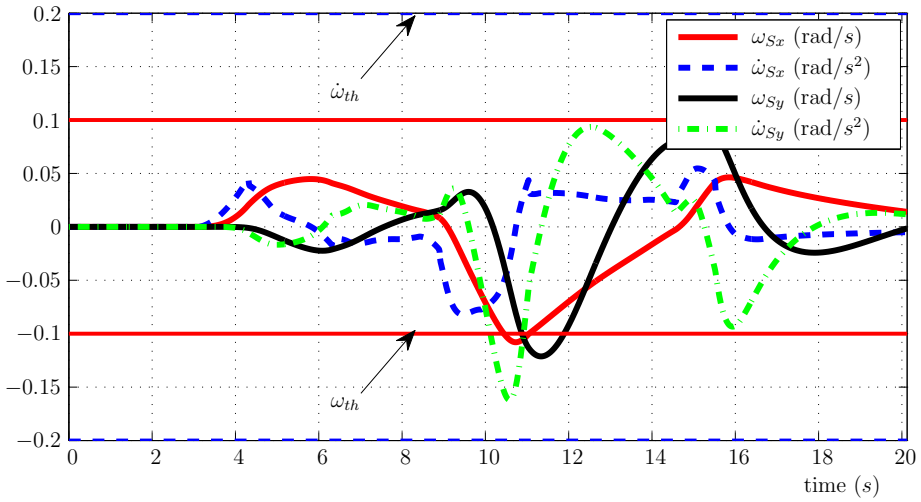


Figure B.69: OpRN - Simulated rotational quantities

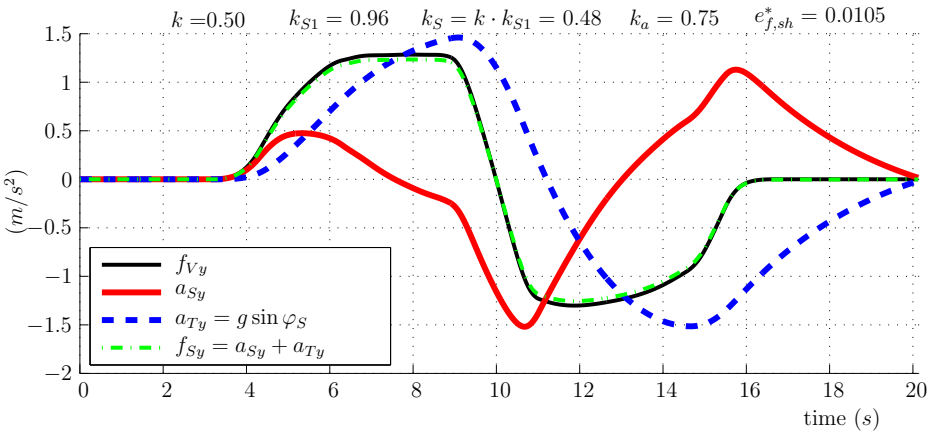


Figure B.70: OpRN - Simulated specific forces and acceleration

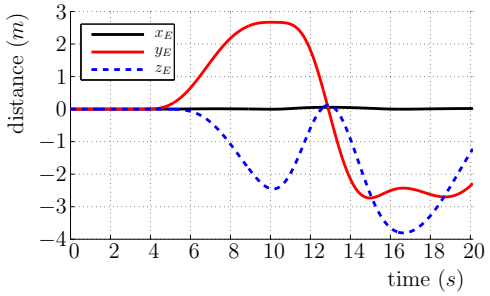


Figure B.71: OpRN - Position

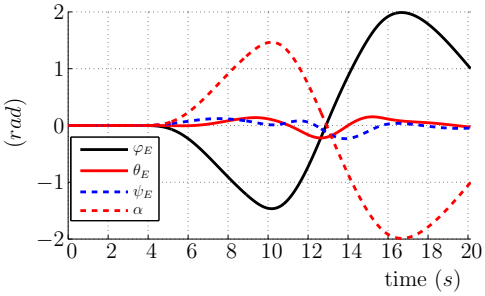


Figure B.72: OpRN - Euler angle

Response of MCA	OpS	$k = 0.5$

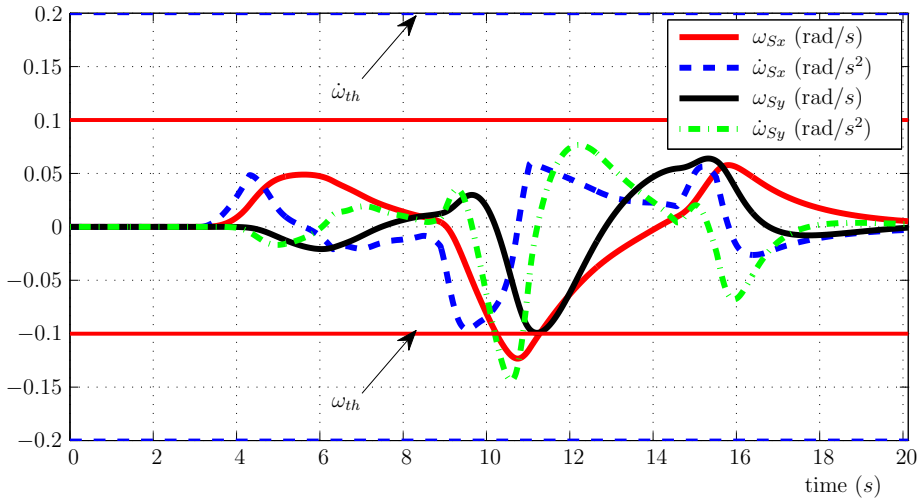


Figure B.73: OpTYM - Simulated rotational quantities

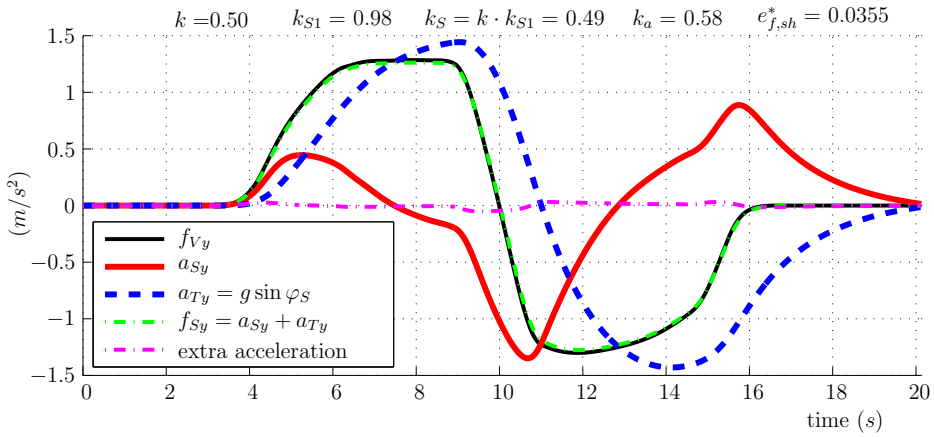


Figure B.74: OpTYM - Simulated specific forces and acceleration

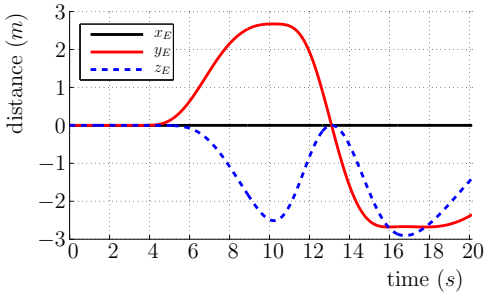


Figure B.75: OpTYM - Position

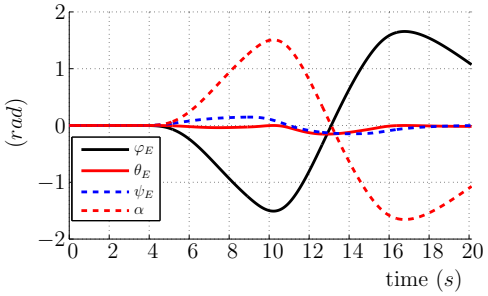


Figure B.76: OpTYM - Euler angle

Response of MCA	OpRN	$k = 0.5$

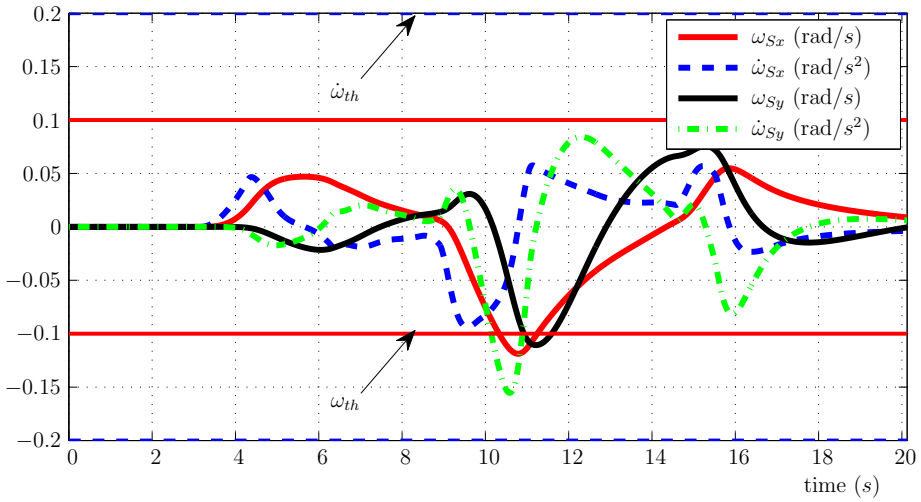


Figure B.77: OpT - Simulated rotational quantities

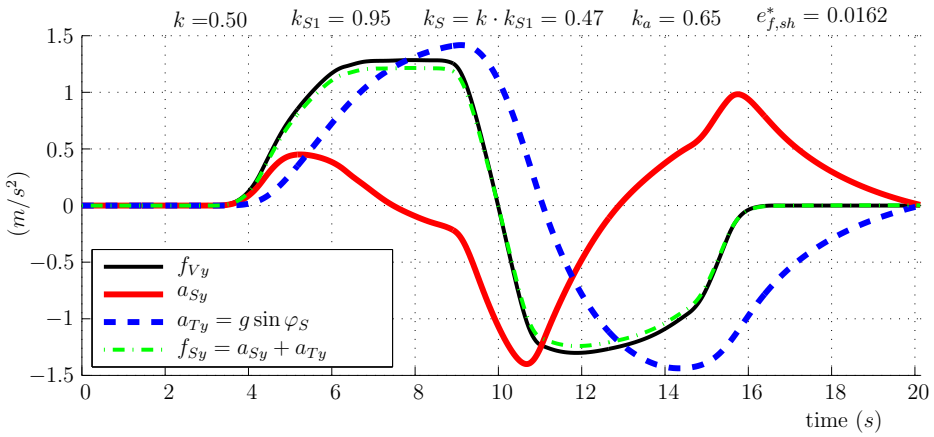


Figure B.78: OpT - Simulated specific forces and acceleration

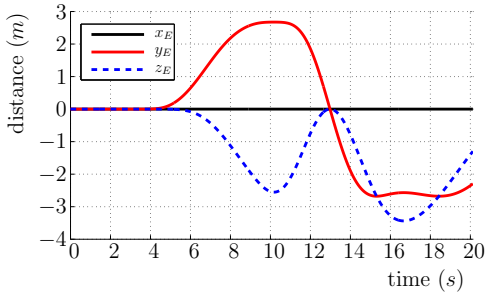


Figure B.79: OpT - Position

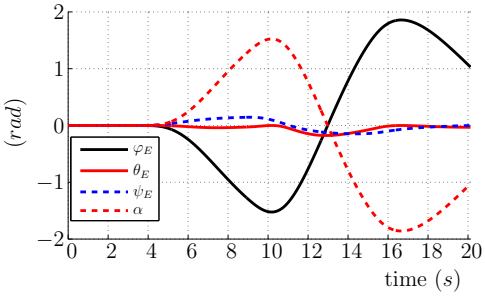


Figure B.80: OpT - Euler angle

Response of MCA	OpTNon	$k = 0.5$

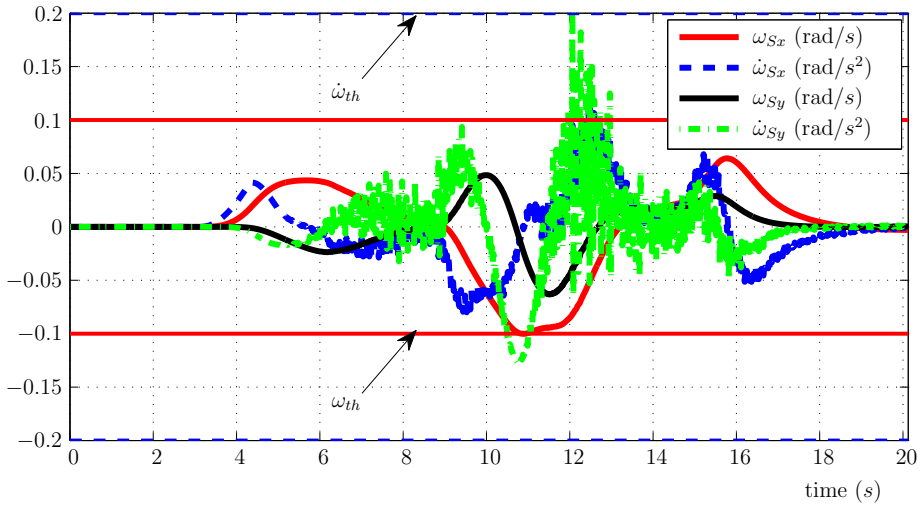


Figure B.81: OpTN - Simulated rotational quantities

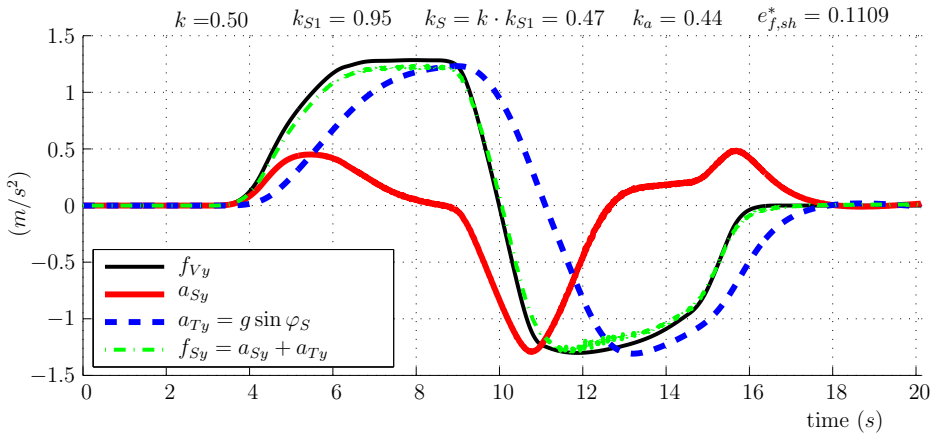


Figure B.82: OpTN - Simulated specific forces and acceleration

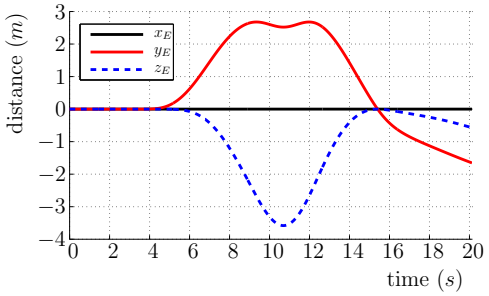


Figure B.83: OpTN - Position

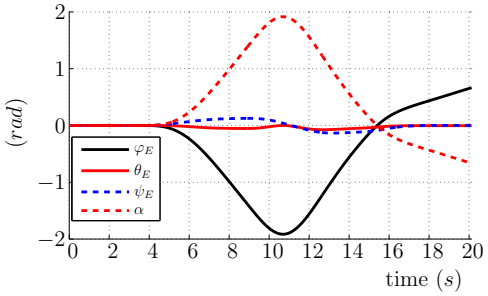


Figure B.84: OpTN - Euler angle

Response of MCA	ZyRo	$k = 0.5$

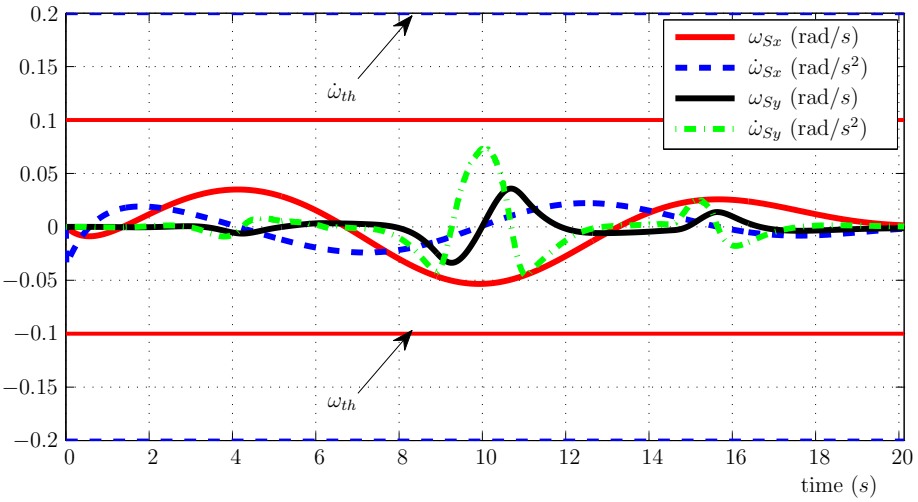


Figure B.85: ZyRo - Simulated rotational quantities

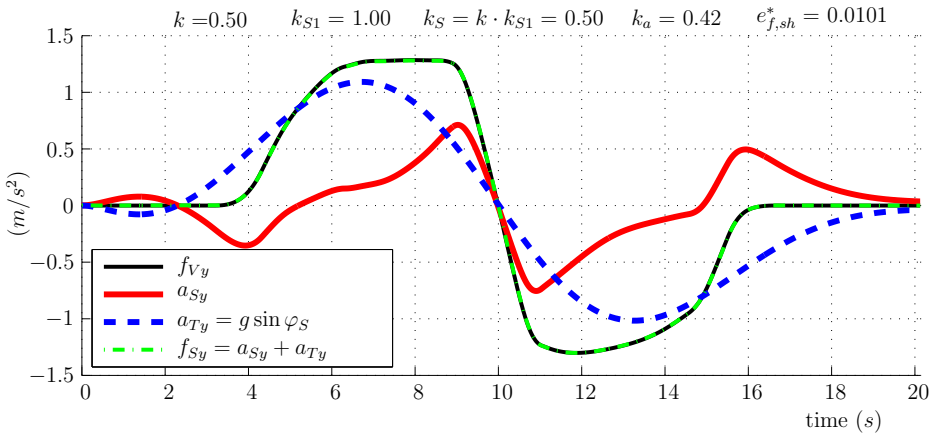


Figure B.86: ZyRo - Simulated specific forces and acceleration

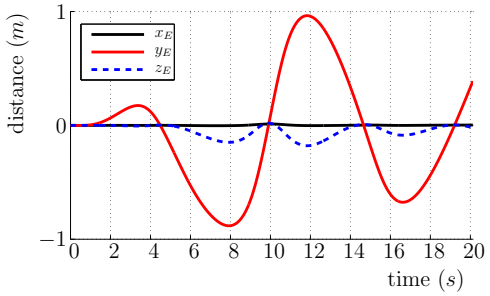


Figure B.87: ZyRo - Position

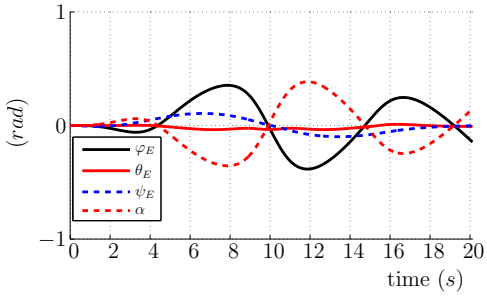


Figure B.88: ZyRo - Euler angle

Response of MCA	MPC*	$k = 0.5$

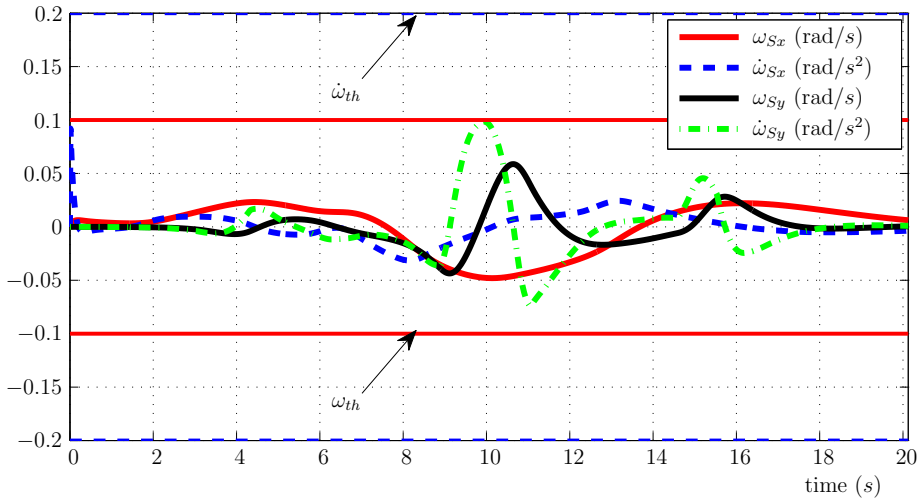


Figure B.89: MPC - Simulated rotational quantities

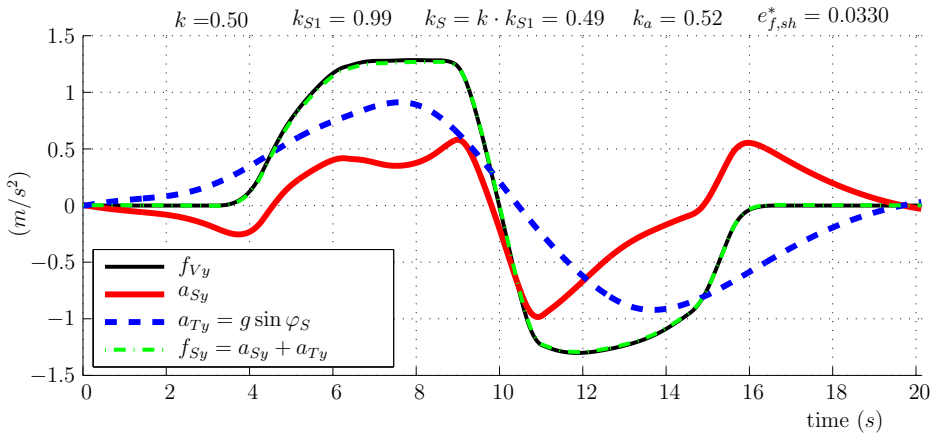


Figure B.90: MPC - Simulated specific forces and acceleration

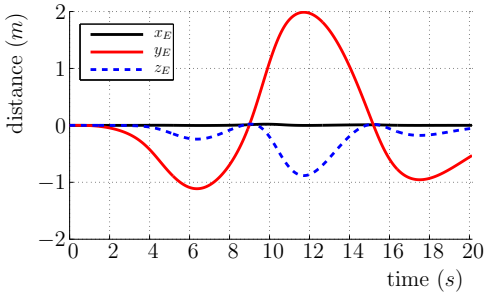


Figure B.91: MPC - Position

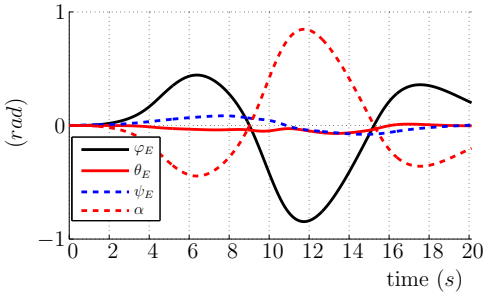


Figure B.92: MPC - Euler angle

Response of MCA	exMPCFK	$k = 0.5$

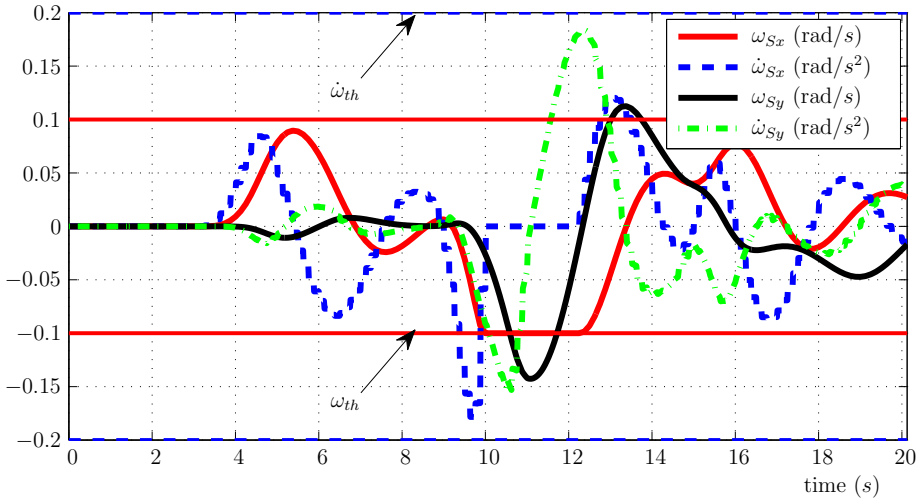


Figure B.93: exMPCFK - Simulated rotational quantities

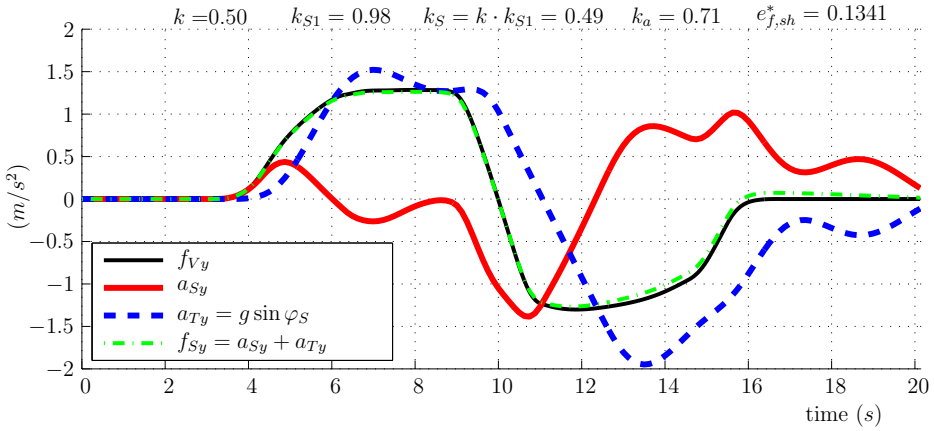


Figure B.94: exMPCFK - Simulated specific forces and acceleration

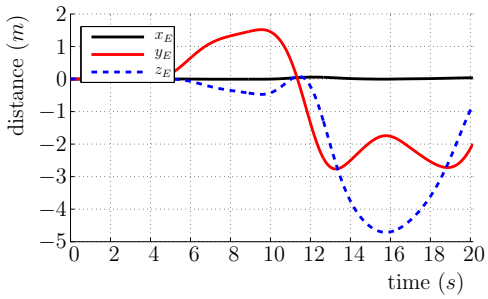


Figure B.95: exMPCFK - Position

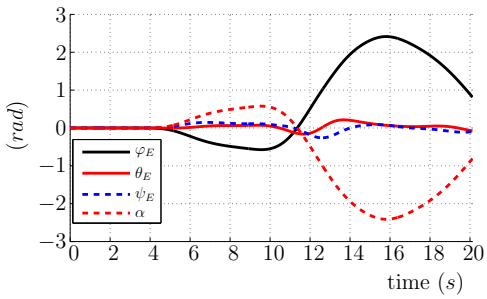


Figure B.96: exMPCFK - Euler angle

Response of MCA	CLRN	$k = 0.6$

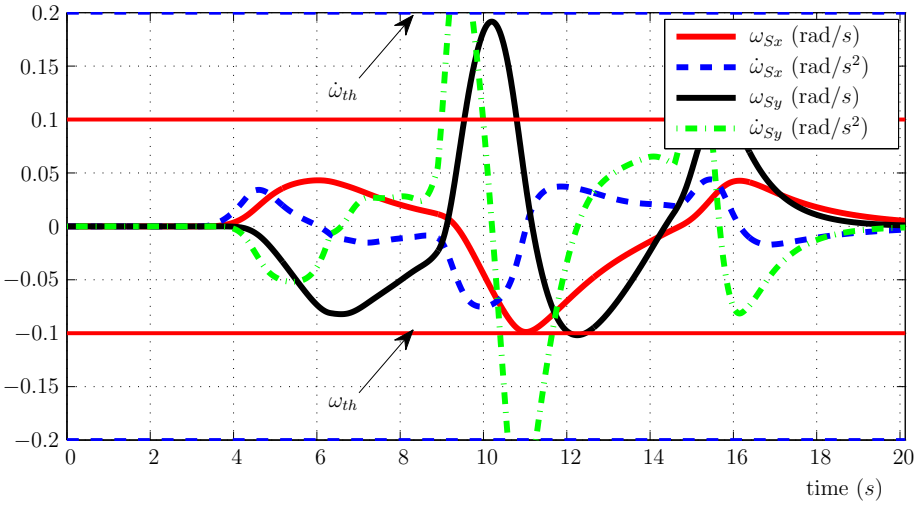


Figure B.97: CLRN - Simulated rotational quantities

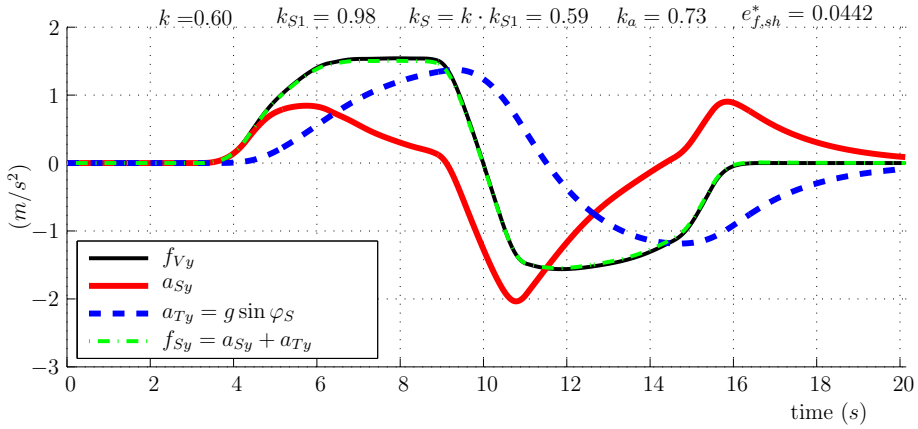


Figure B.98: CLRN - Simulated specific forces and acceleration

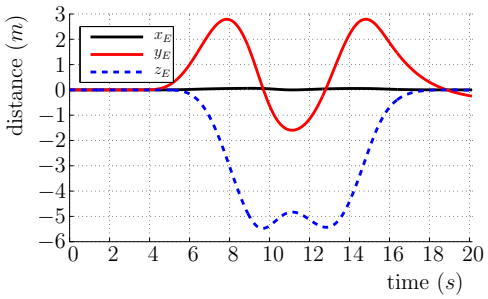


Figure B.99: CLRN - Position

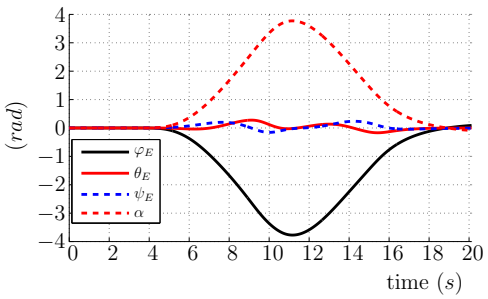


Figure B.100: CLRN - Euler angle

Response of MCA	CLG	$k = 0.6$

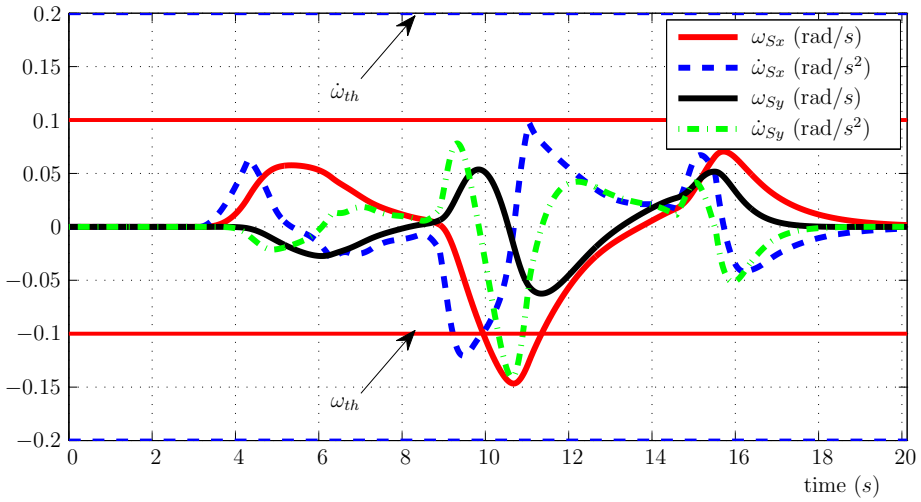


Figure B.101: CLG - Simulated rotational quantities

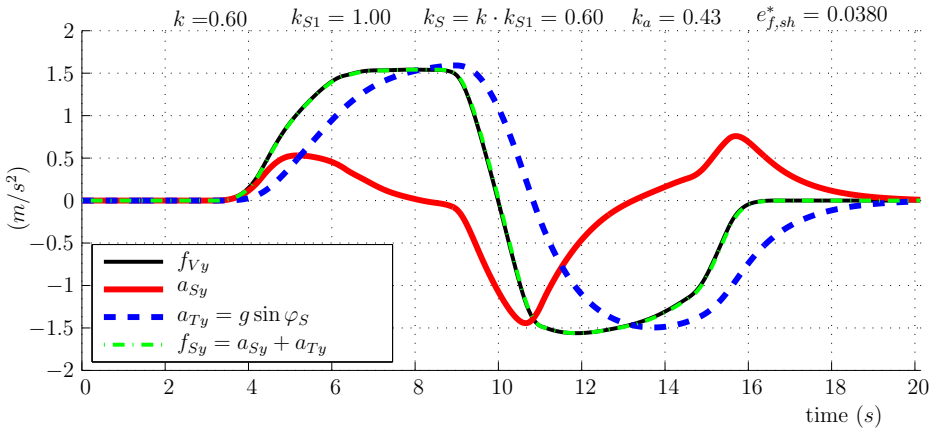


Figure B.102: CLG - Simulated specific forces and acceleration

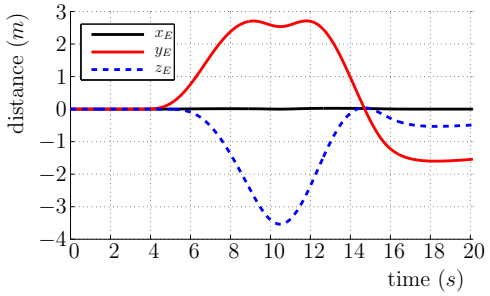


Figure B.103: CLG - Position

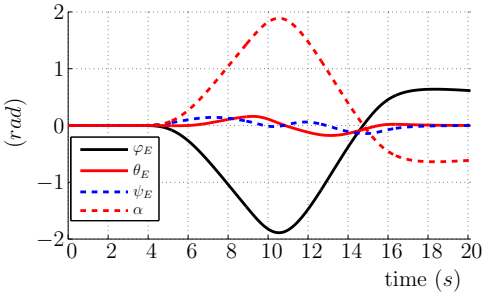


Figure B.104: CLG - Euler angle

Response of MCA	ADRN	$k = 0.6$

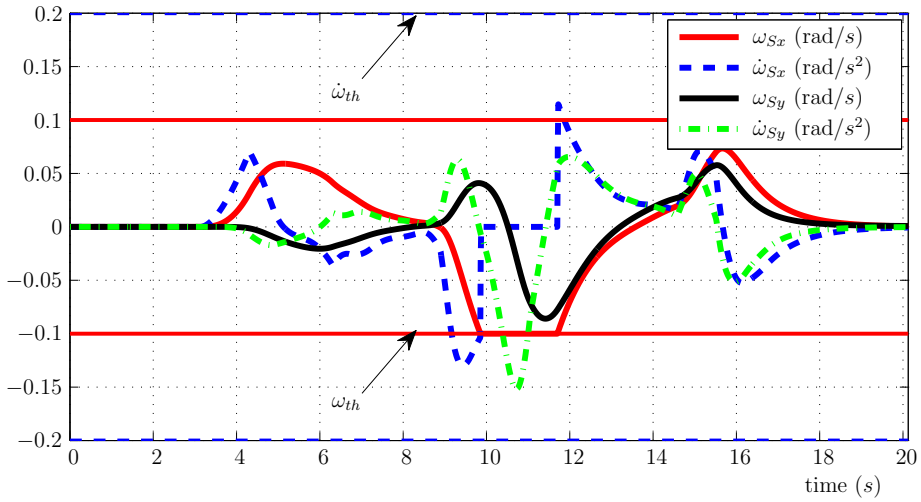


Figure B.105: ADRN - Simulated rotational quantities

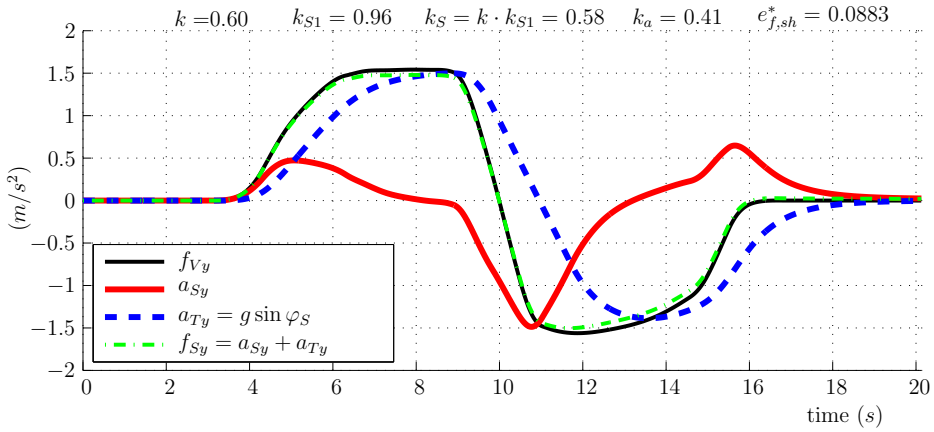


Figure B.106: ADRN - Simulated specific forces and acceleration

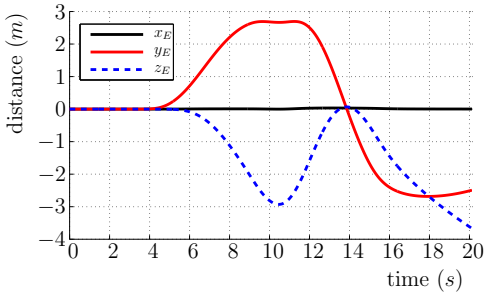


Figure B.107: ADRN - Position

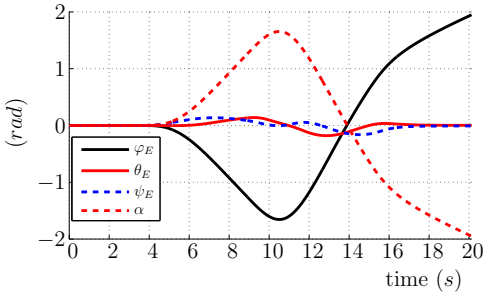


Figure B.108: ADRN - Euler angle

Response of MCA	ADSK	$k = 0.6$

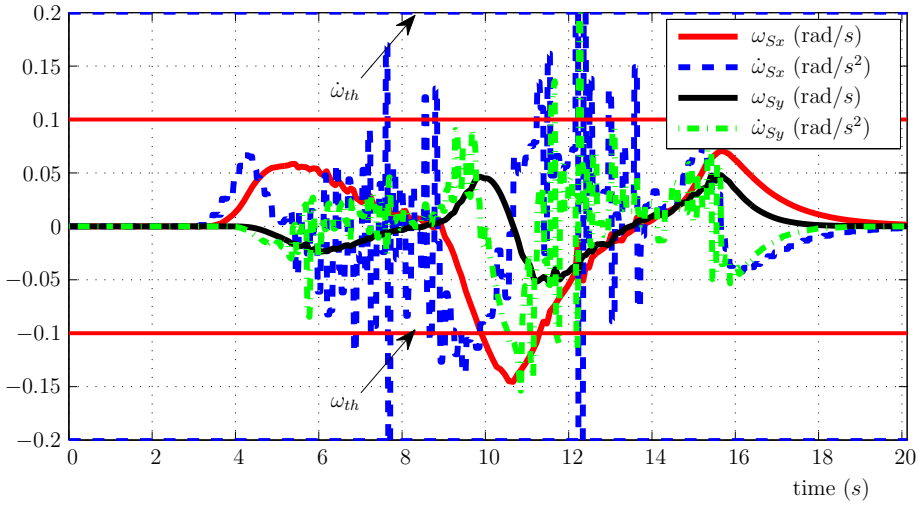


Figure B.109: ADSK - Simulated rotational quantities

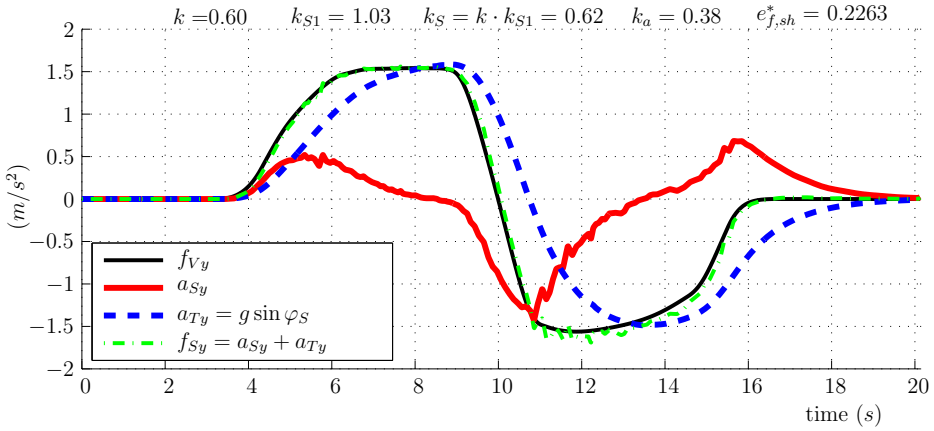


Figure B.110: ADSK - Simulated specific forces and acceleration

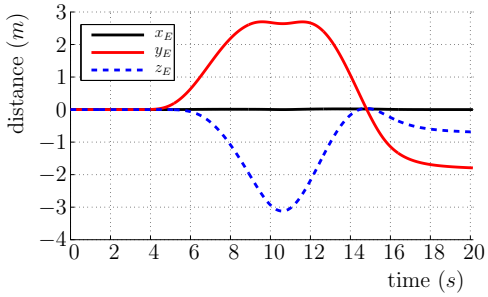


Figure B.111: ADSK - Position

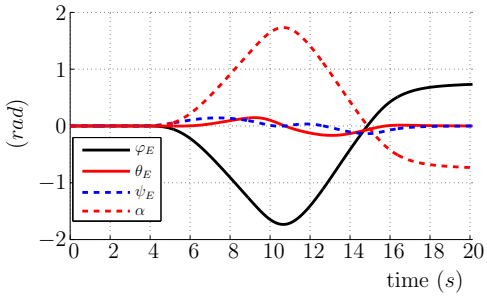


Figure B.112: ADSK - Euler angle

Response of MCA	OpTYM	$k = 0.6$

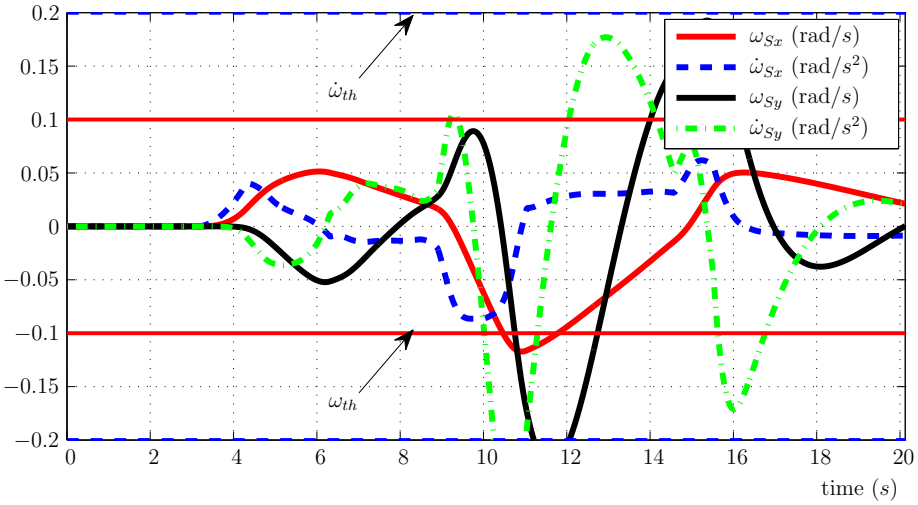


Figure B.113: OpS - Simulated rotational quantities

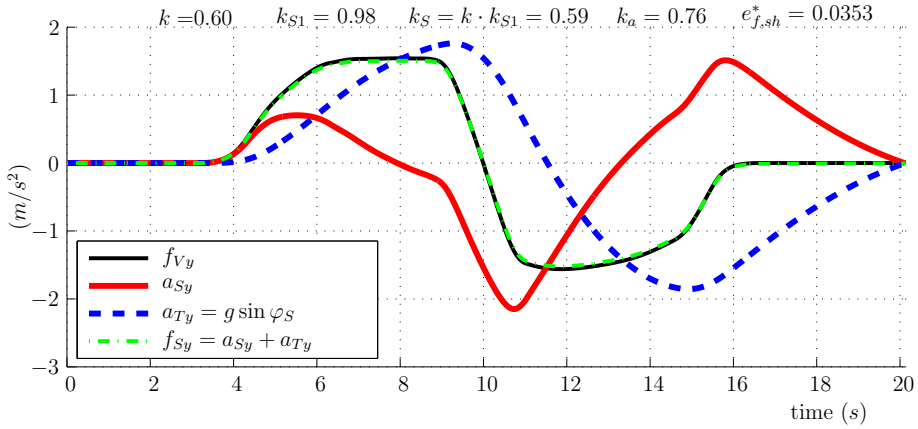


Figure B.114: OpS - Simulated specific forces and acceleration

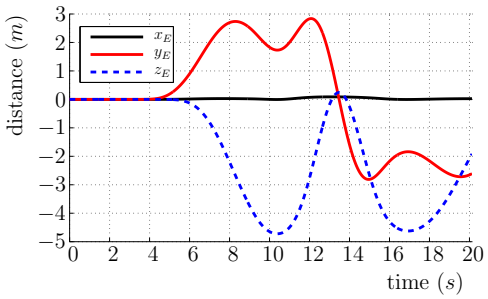


Figure B.115: OpS - Position

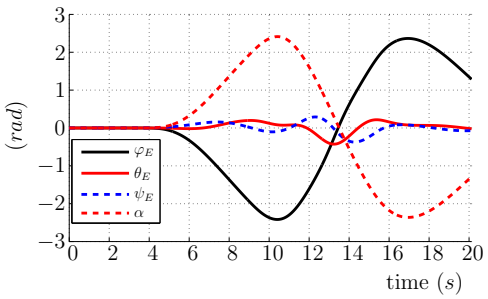


Figure B.116: OpS - Euler angle

Response of MCA	OpT	$k = 0.6$

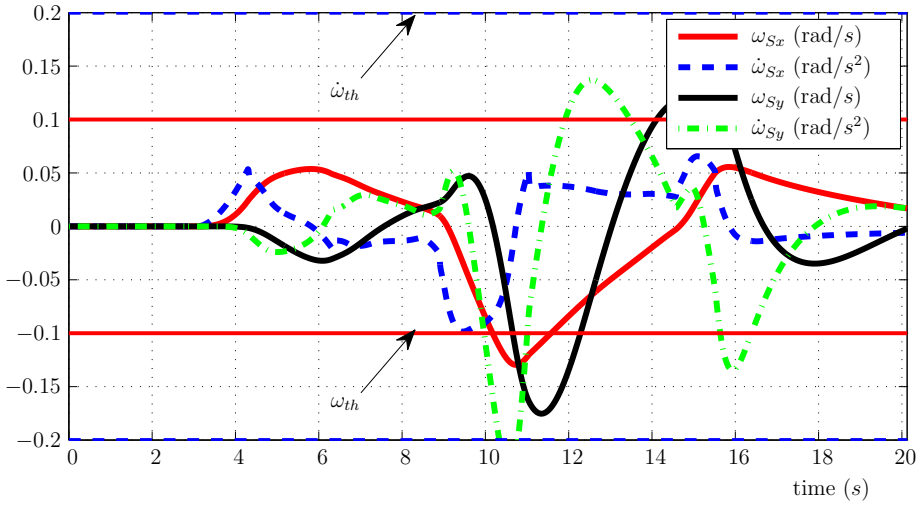


Figure B.117: OpRN - Simulated rotational quantities

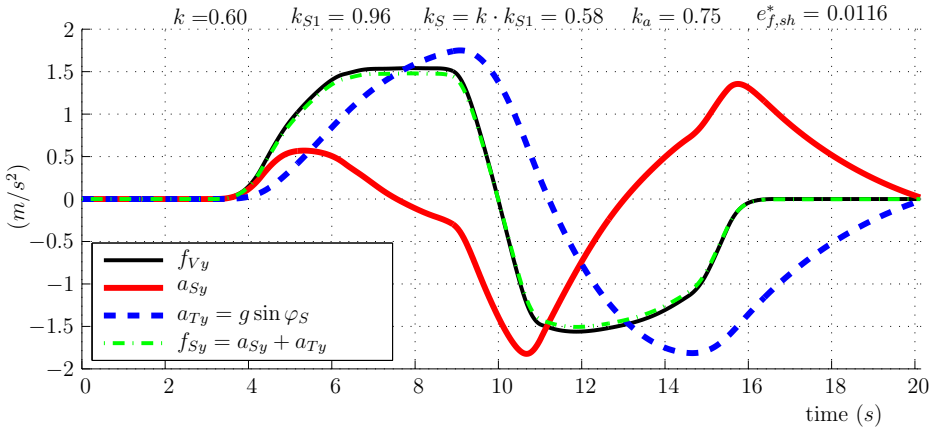


Figure B.118: OpRN - Simulated specific forces and acceleration

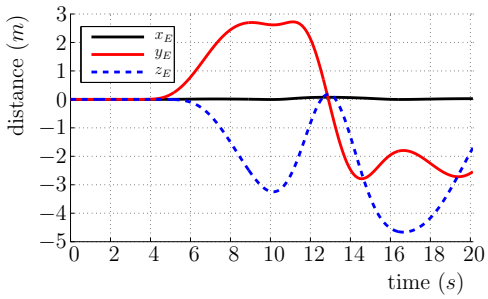


Figure B.119: OpRN - Position

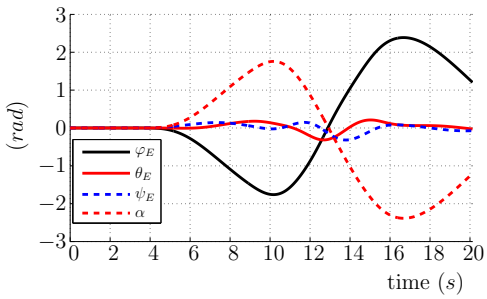


Figure B.120: OpRN - Euler angle

Response of MCA	OpS	$k = 0.6$

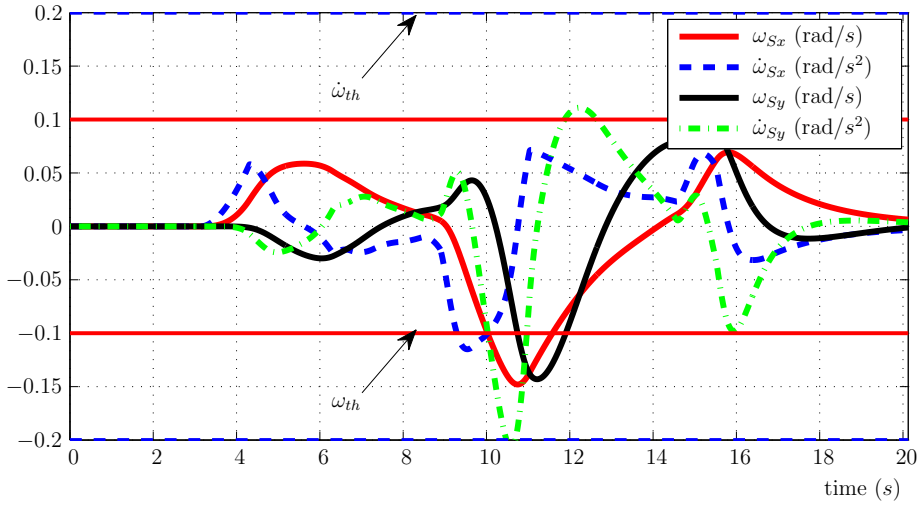


Figure B.121: OpTYM - Simulated rotational quantities

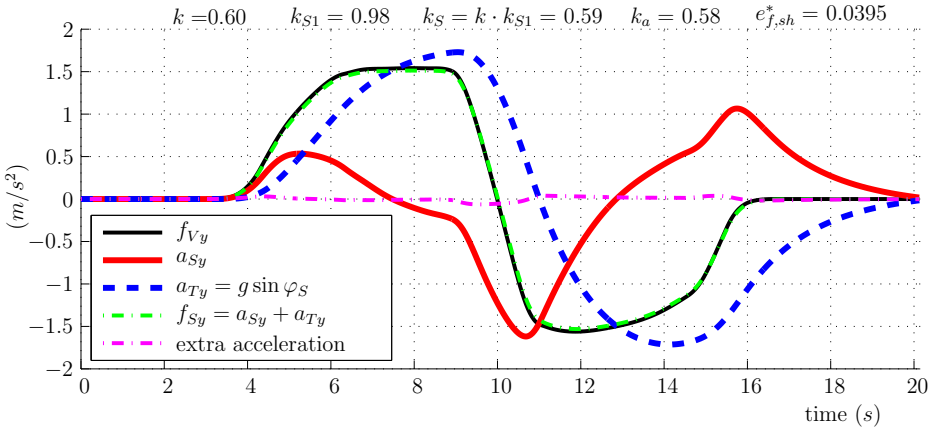


Figure B.122: OpTYM - Simulated specific forces and acceleration

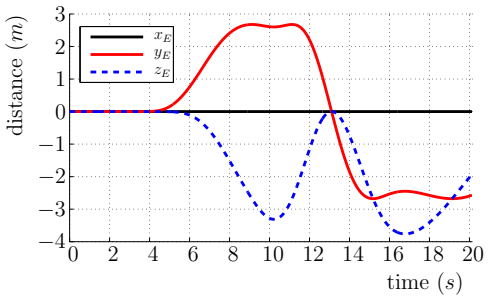


Figure B.123: OpTYM - Simulator Pos

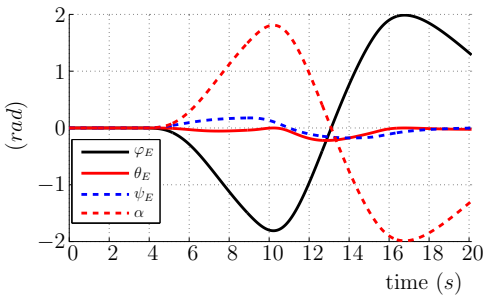


Figure B.124: OpTYM - Euler angle

Response of MCA	OpRN	$k = 0.6$

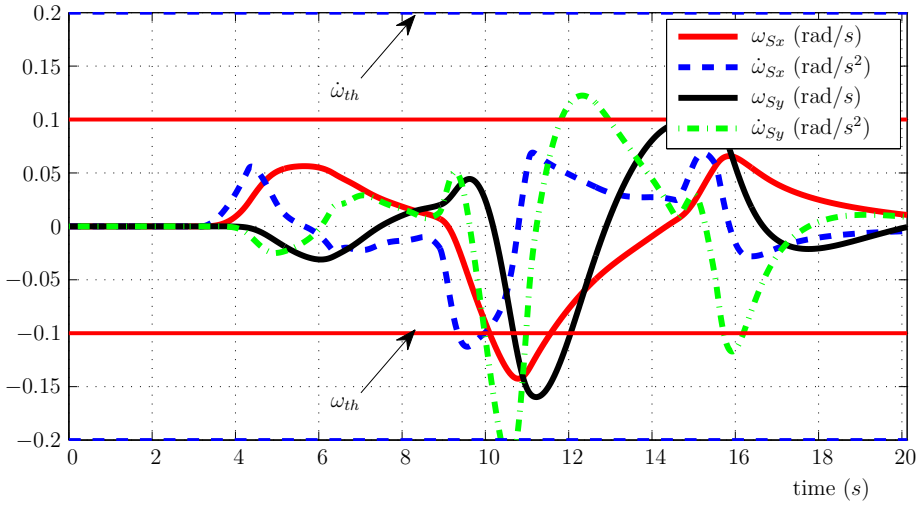


Figure B.125: OpT - Simulated rotational quantities

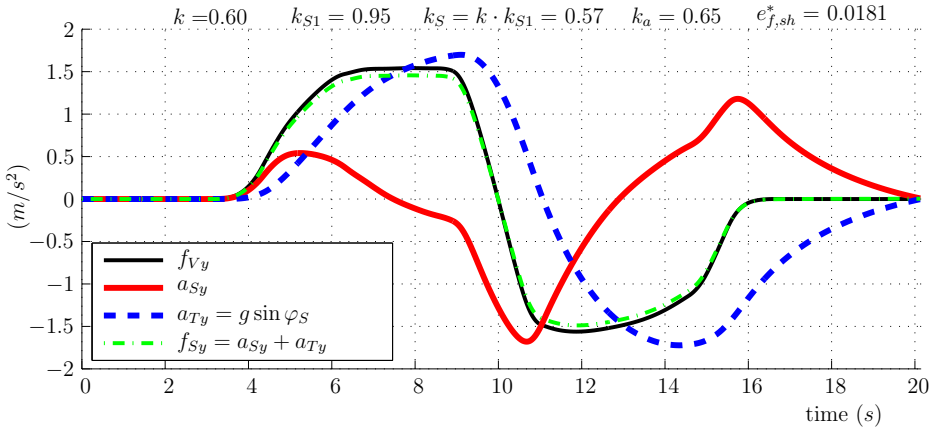


Figure B.126: OpT - Simulated specific forces and acceleration

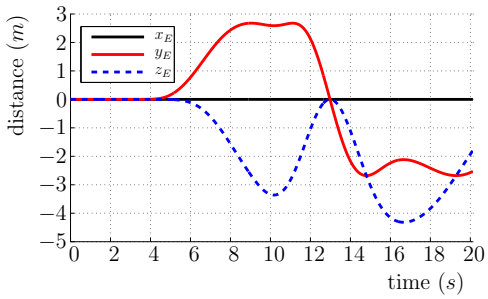


Figure B.127: OpT - Position

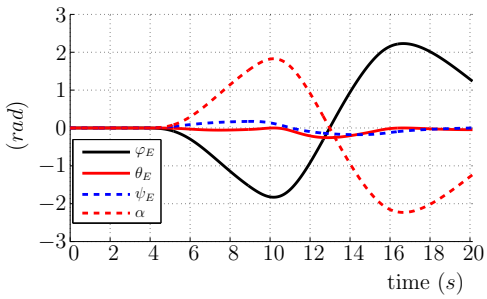


Figure B.128: OpT - Euler angle

Response of MCA	OpTNon	$k = 0.6$

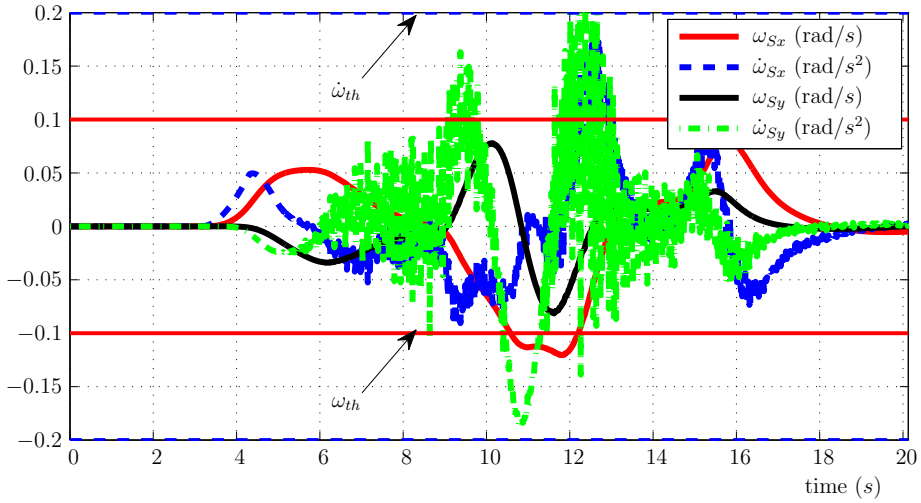


Figure B.129: OpTN - Simulated rotational quantities

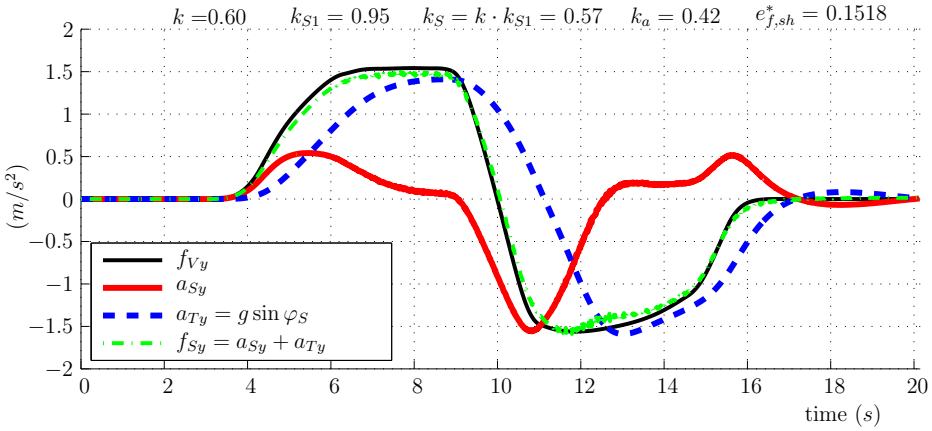


Figure B.130: OpTN - Simulated specific forces and acceleration

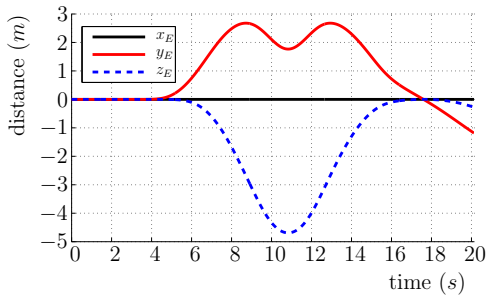


Figure B.131: OpTN - Position

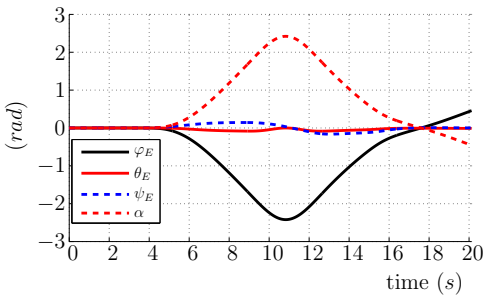


Figure B.132: OpTN - Euler angle

Response of MCA	ZyRo	$k = 0.6$

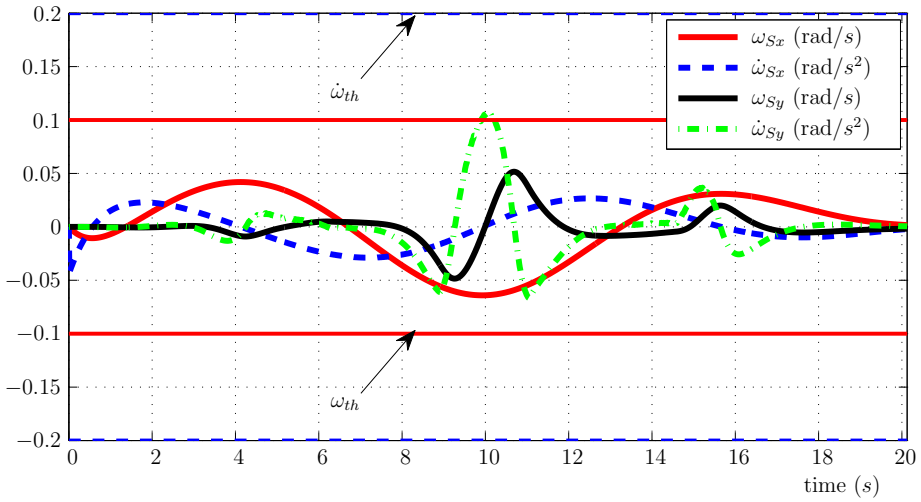


Figure B.133: ZyRo - Simulated rotational quantities

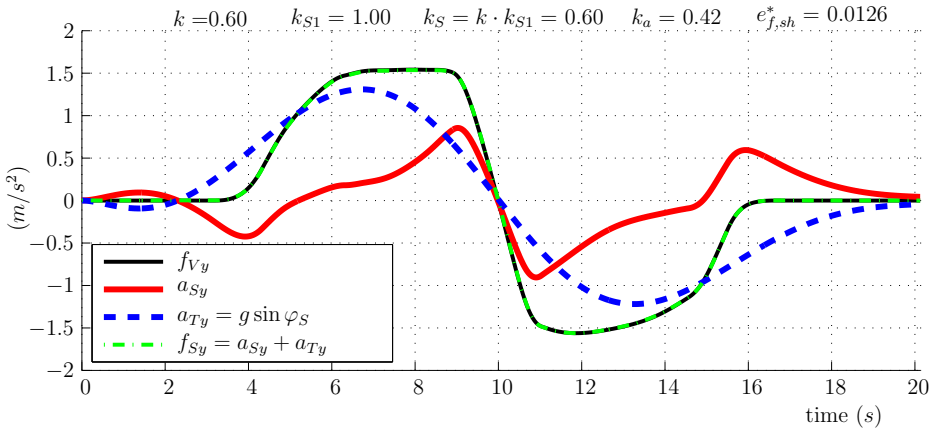


Figure B.134: ZyRo - Simulated specific forces and acceleration

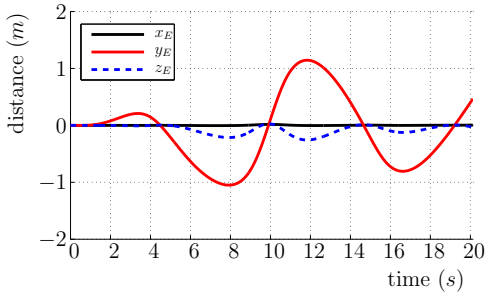


Figure B.135: ZyRo - Position

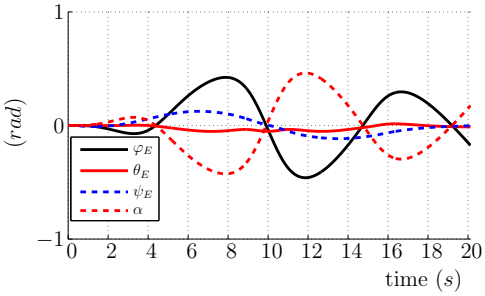


Figure B.136: ZyRo - Euler angle

Response of MCA	MPC*	$k = 0.6$

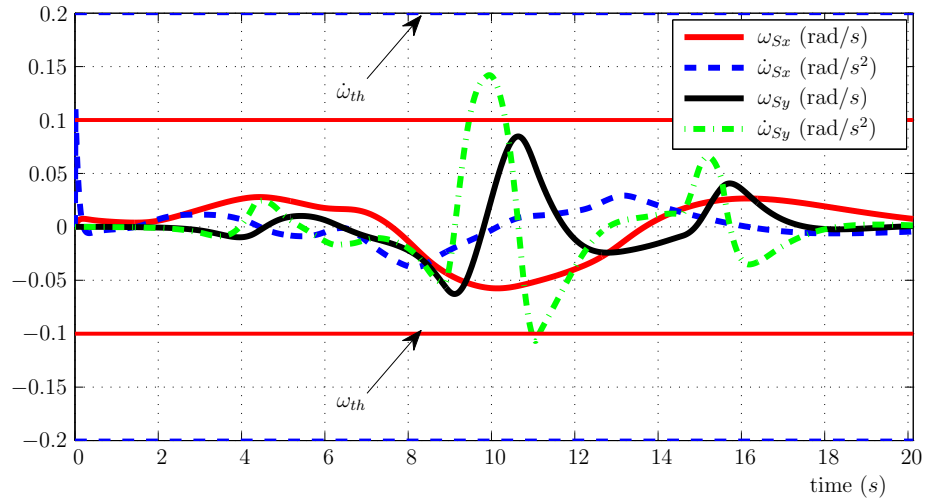


Figure B.137: MPC - Simulated rotational quantities

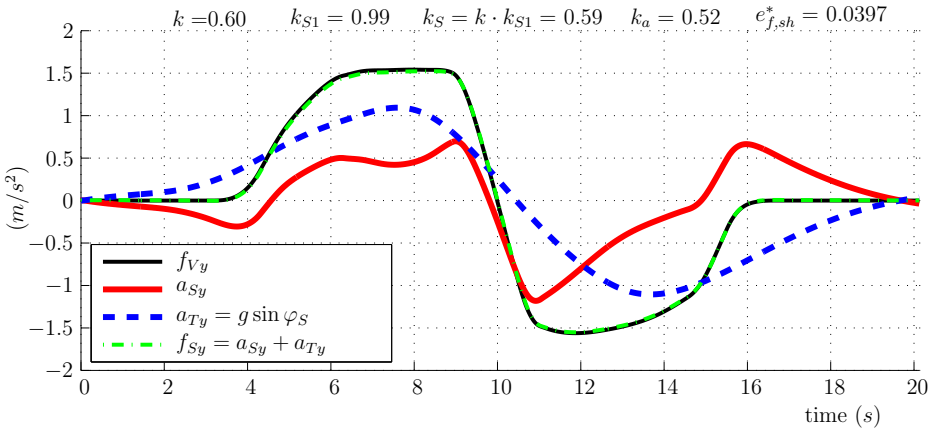


Figure B.138: MPC - Simulated specific forces and acceleration

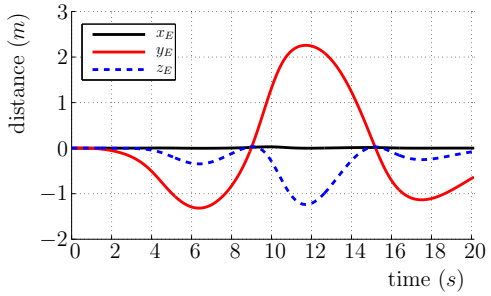


Figure B.139: MPC - Position

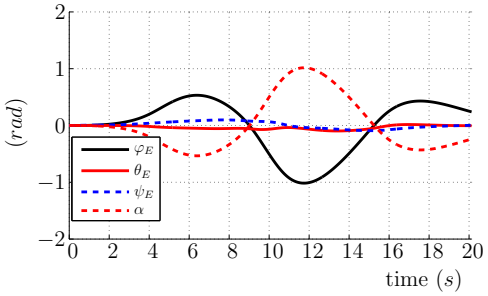


Figure B.140: MPC - Euler angle

Response of MCA	exMPCFK	$k = 0.6$

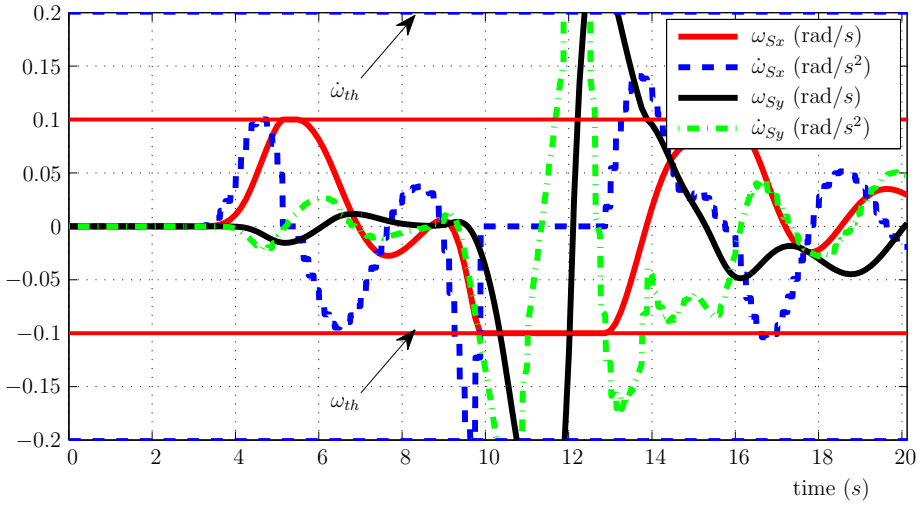


Figure B.141: exMPCFK - Simulated rotational quantities

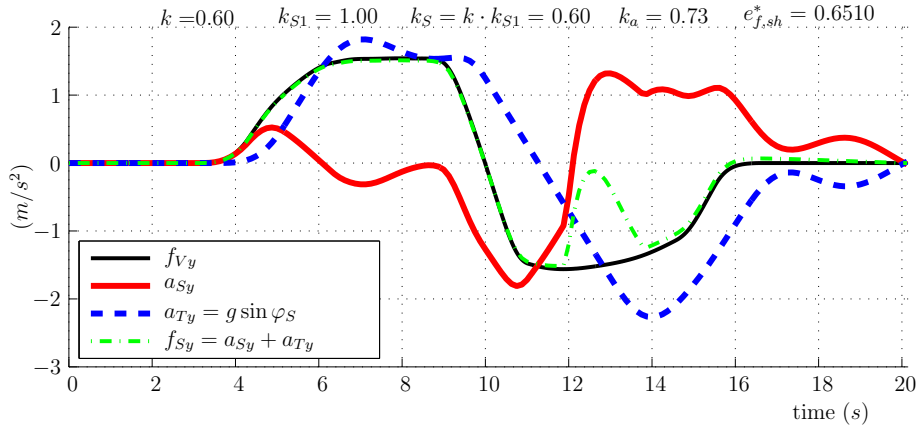


Figure B.142: exMPCFK - Simulated specific forces and acceleration

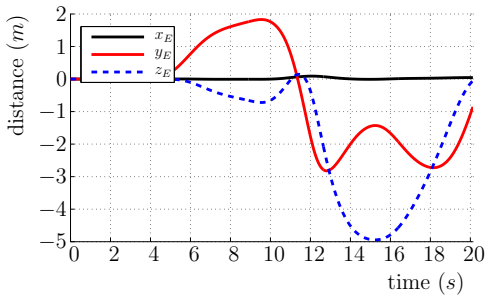


Figure B.143: exMPCFK - Position

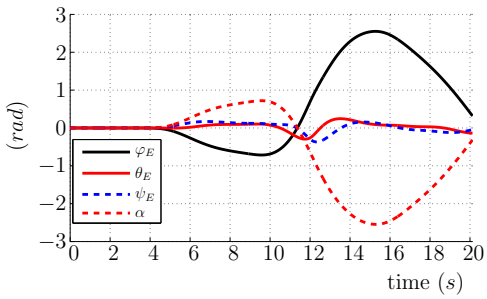


Figure B.144: exMPCFK - Euler angle

Response of MCA	CLRN	$k = 0.7$

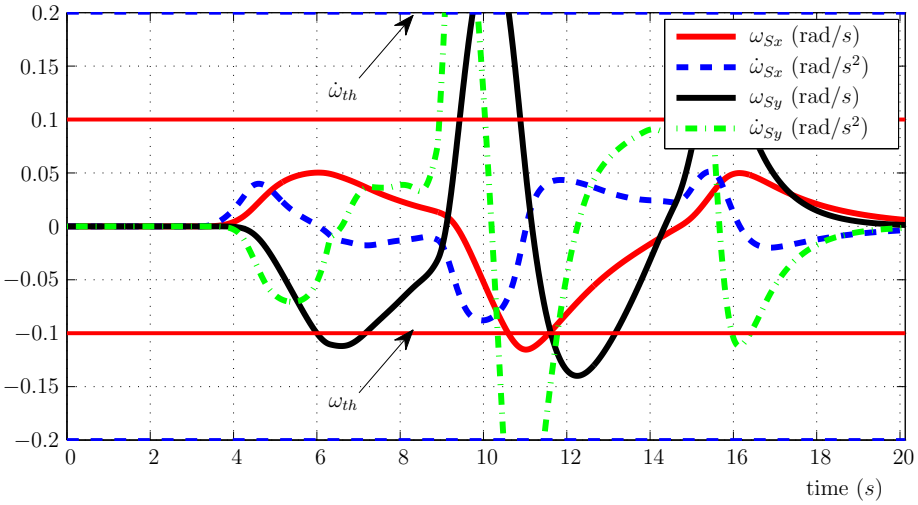


Figure B.145: CLRN - Simulated rotational quantities

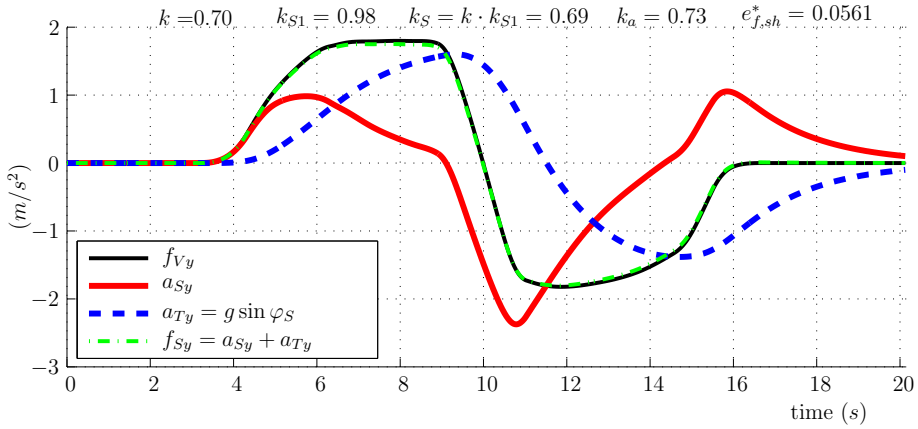


Figure B.146: CLRN - Simulated specific forces and acceleration

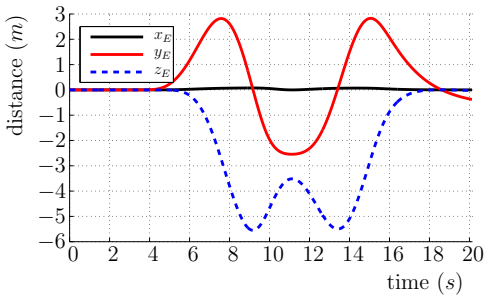


Figure B.147: CLRN - Position

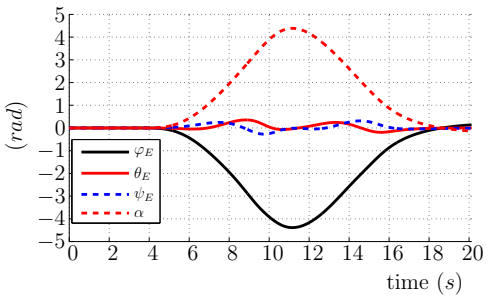


Figure B.148: CLRN - Euler angle

Response of MCA	CLG	$k = 0.7$

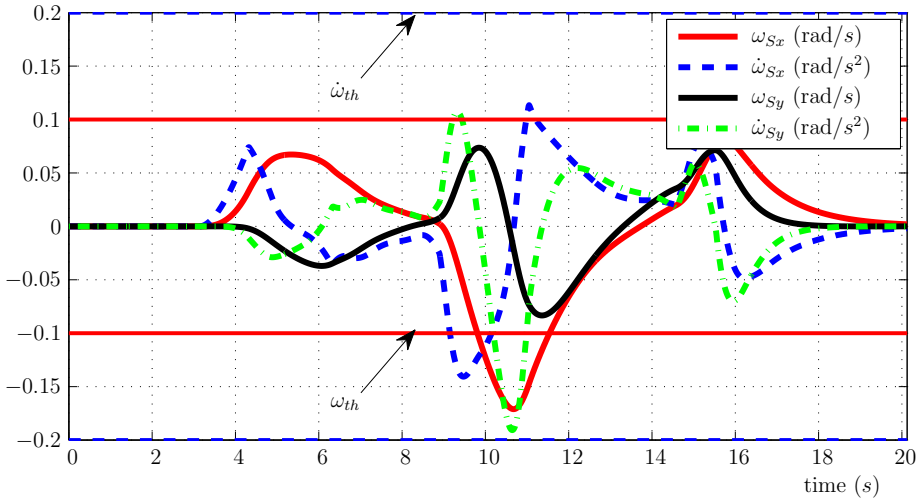


Figure B.149: CLG - Simulated rotational quantities

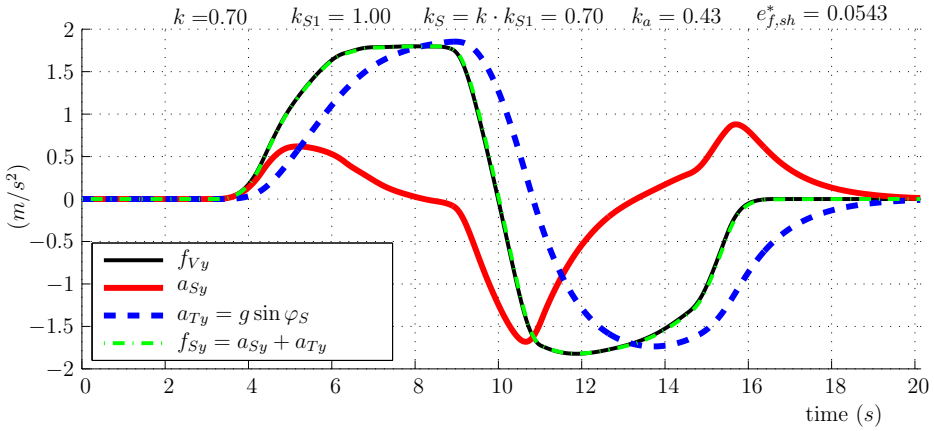


Figure B.150: CLG - Simulated specific forces and acceleration

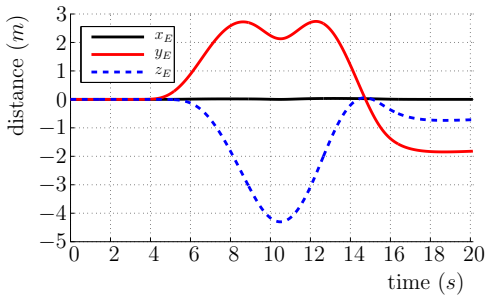


Figure B.151: CLG - Position

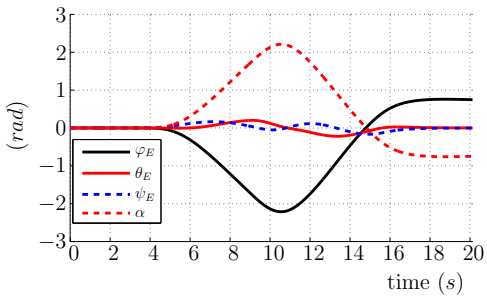


Figure B.152: CLG - Euler angle

Response of MCA	ADRN	$k = 0.7$

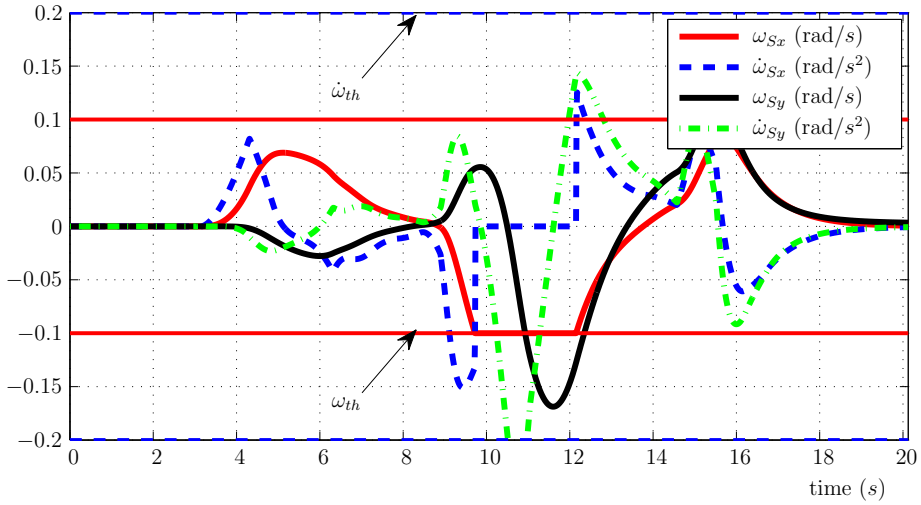


Figure B.153: ADRN - Simulated rotational quantities

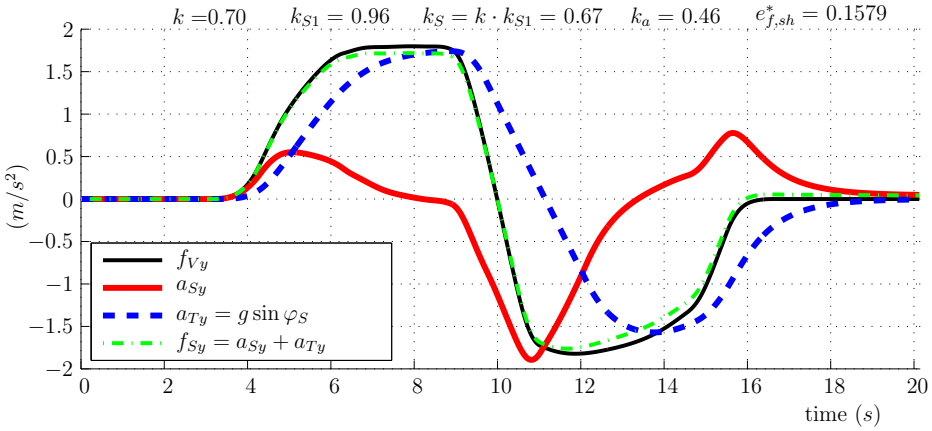


Figure B.154: ADRN - Simulated specific forces and acceleration

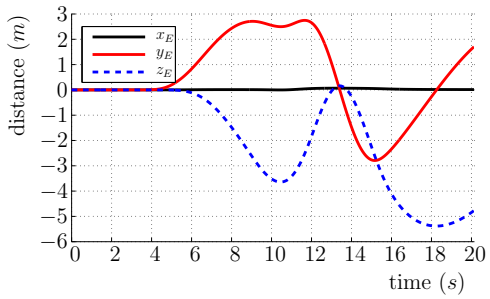


Figure B.155: ADRN - Position

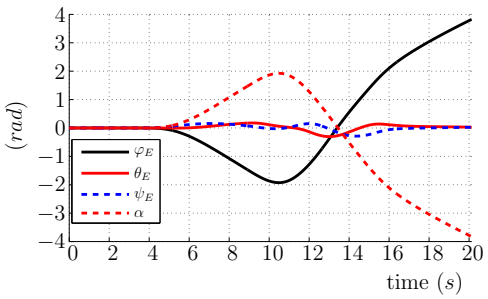


Figure B.156: ADRN - Euler angle

Response of MCA	ADSK	$k = 0.7$

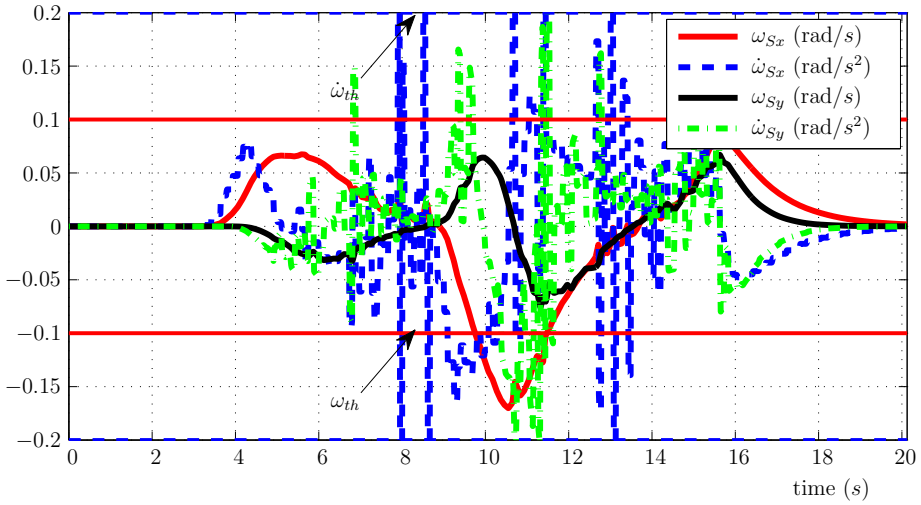


Figure B.157: ADSK - Simulated rotational quantities

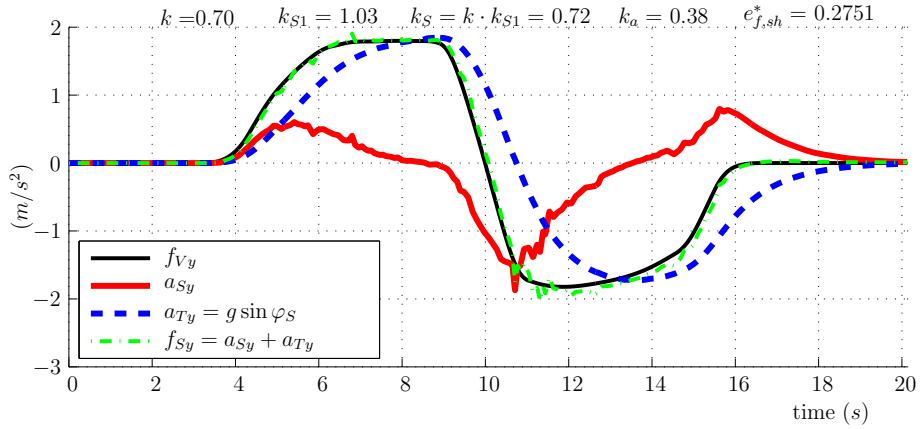


Figure B.158: ADSK - Simulated specific forces and acceleration

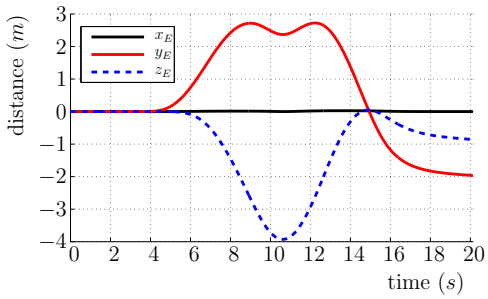


Figure B.159: ADSK - Position

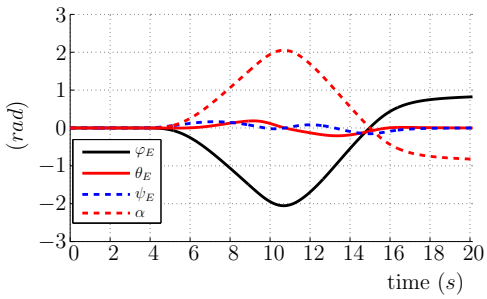


Figure B.160: ADSK - Euler angle

Response of MCA	OpTYM	$k = 0.7$

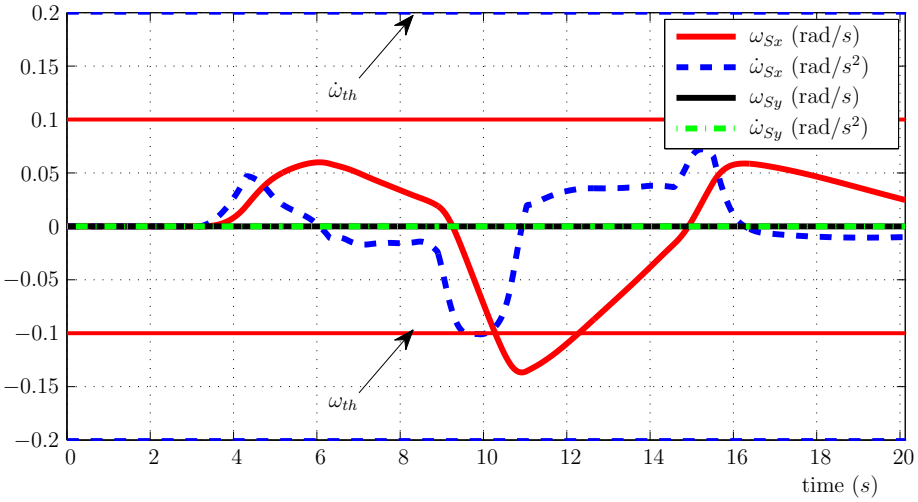


Figure B.161: OpS - Simulated rotational quantities

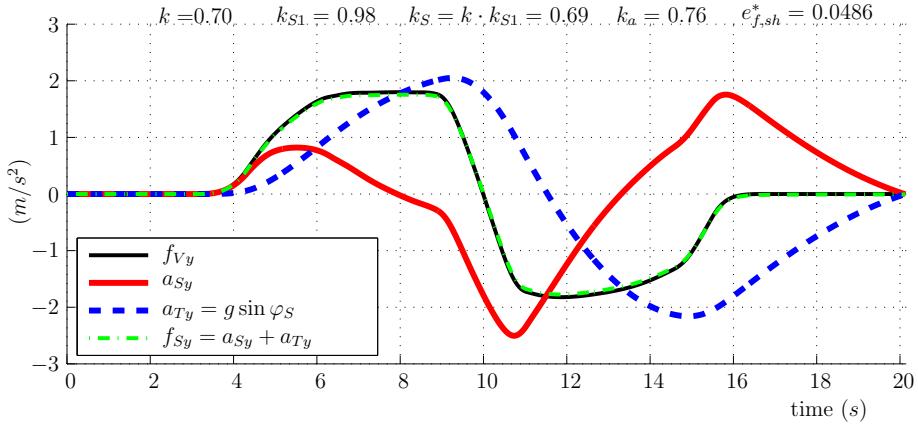


Figure B.162: OpS - Simulated specific forces and acceleration

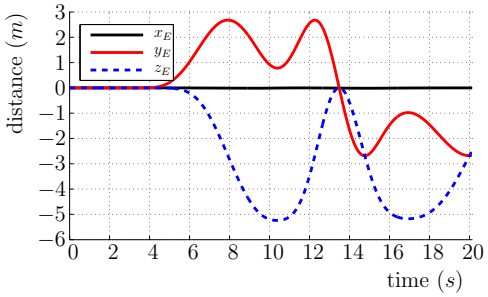


Figure B.163: OpS - Position

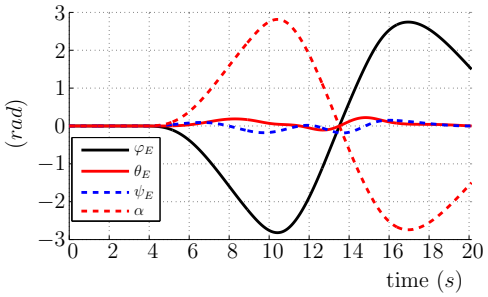


Figure B.164: OpS - Euler angle

Response of MCA	OpT	$k = 0.7$

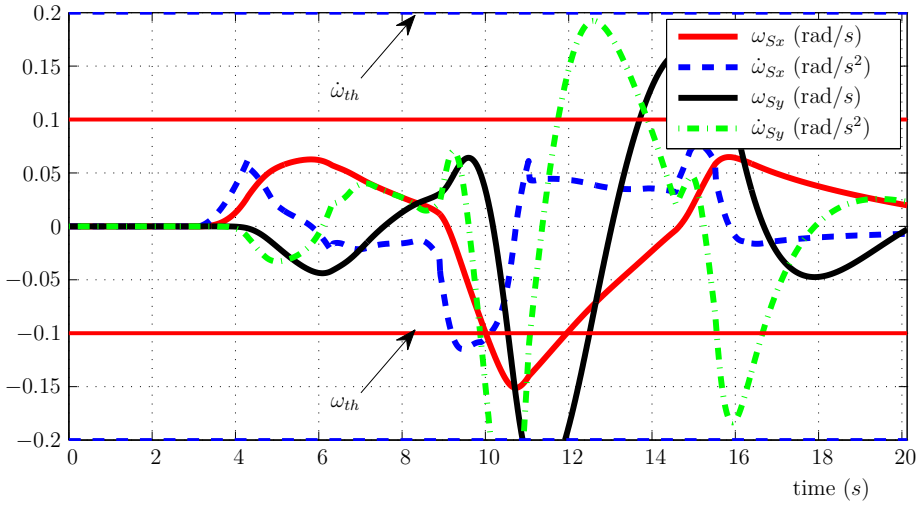


Figure B.165: OpRN - Simulated rotational quantities

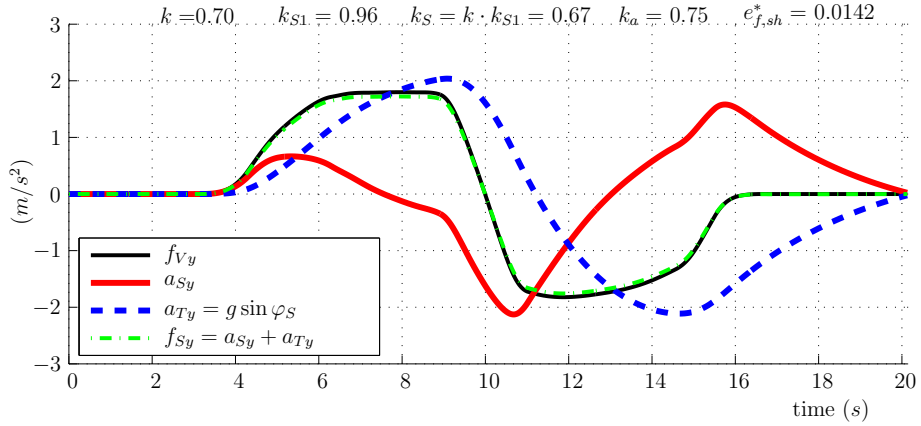


Figure B.166: OpRN - Simulated specific forces and acceleration

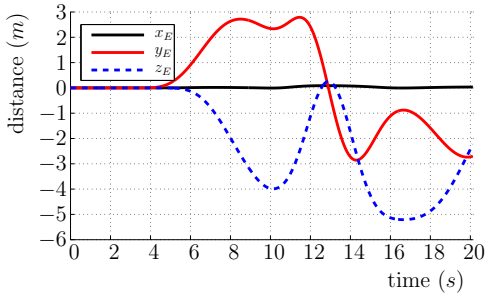


Figure B.167: OpRN - Position

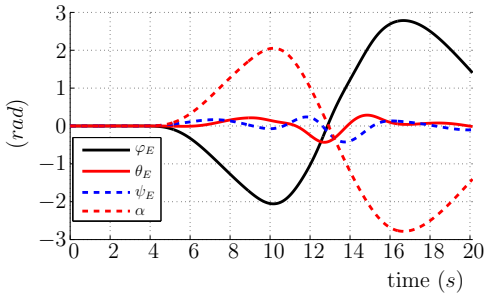


Figure B.168: OpRN - Euler angle

Response of MCA	OpS	$k = 0.7$

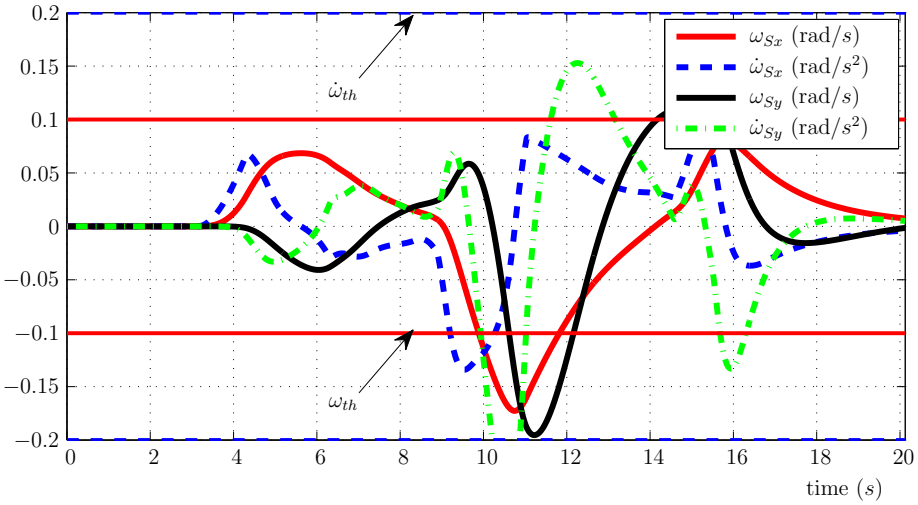


Figure B.169: OpTYM - Simulated rotational quantities

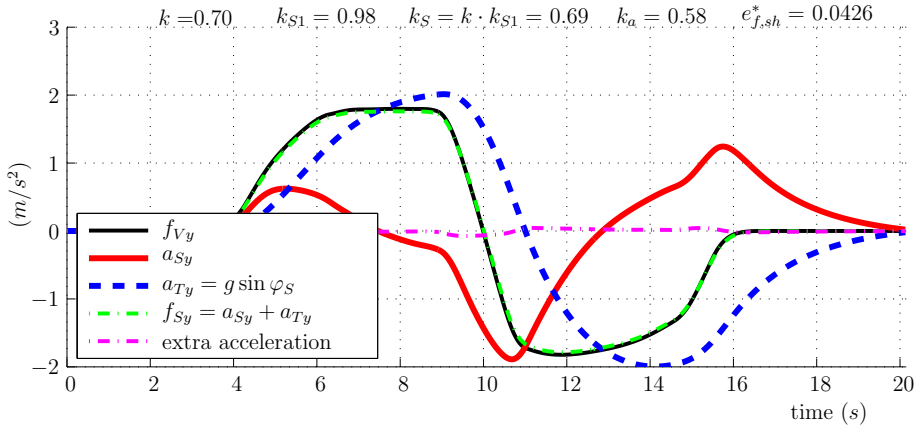


Figure B.170: OpTYM - Simulated specific forces and acceleration

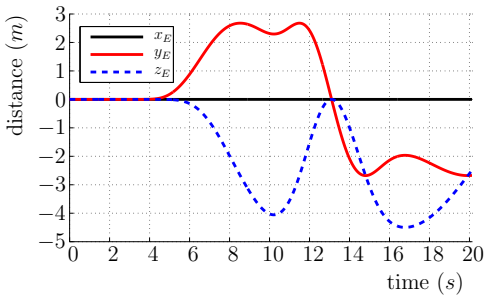


Figure B.171: OpTYM - Position

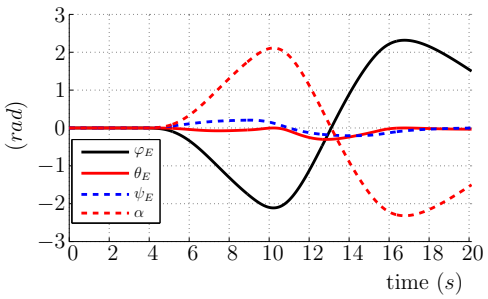


Figure B.172: OpTYM - Euler angle

Response of MCA	OpRN	$k = 0.7$

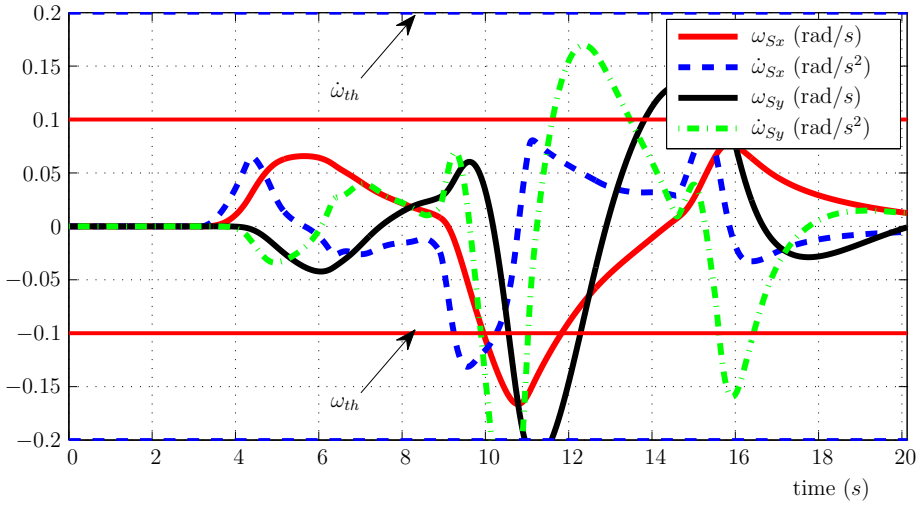


Figure B.173: OpT - Simulated rotational quantities

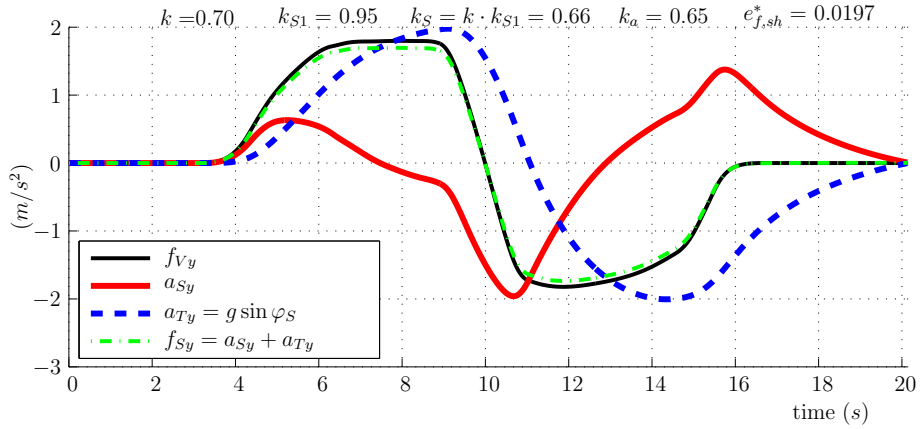


Figure B.174: OpT - Simulated specific forces and acceleration

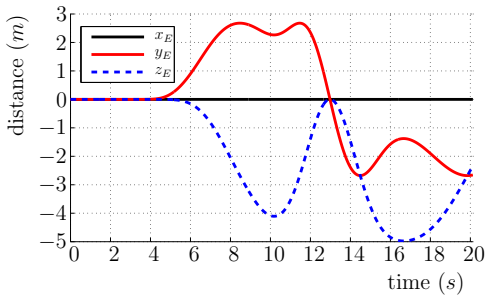


Figure B.175: OpT - Position

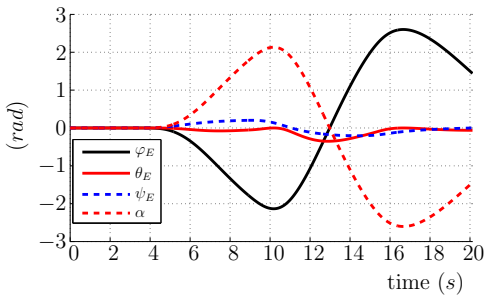


Figure B.176: OpT - Euler angle

Response of MCA	OpTNon	$k = 0.7$

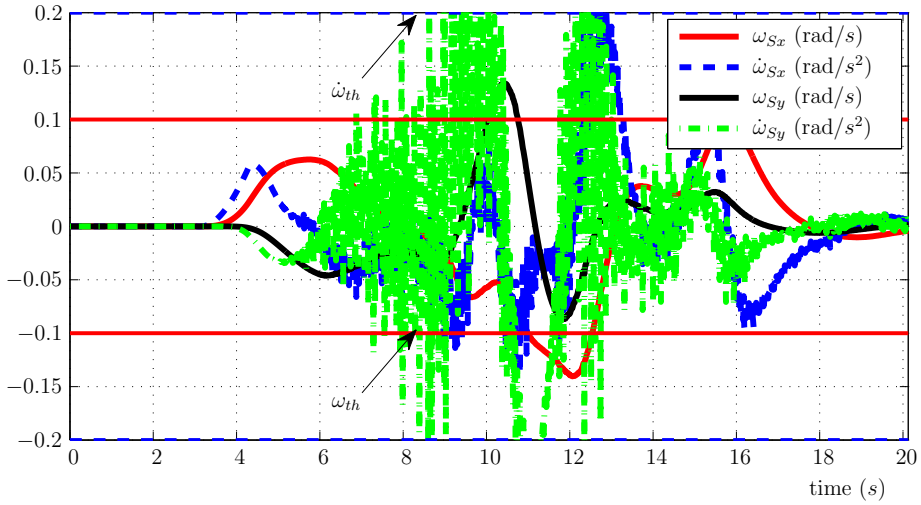


Figure B.177: OpTN - Simulated rotational quantities

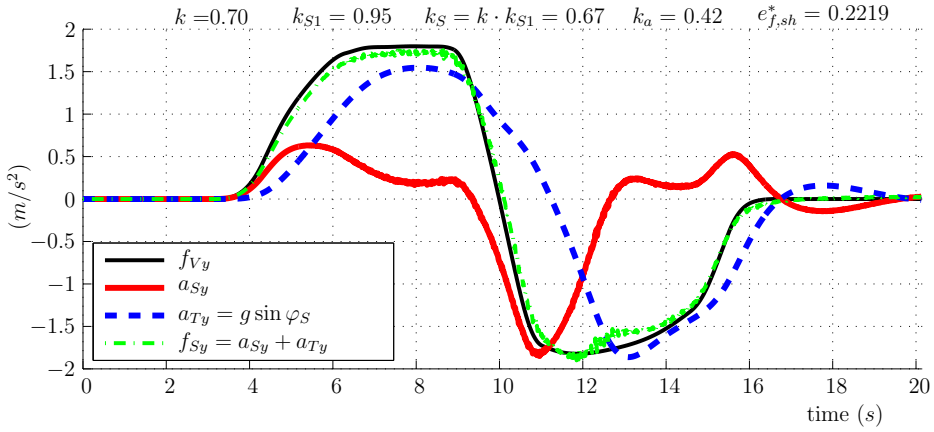


Figure B.178: OpTN - Simulated specific forces and acceleration

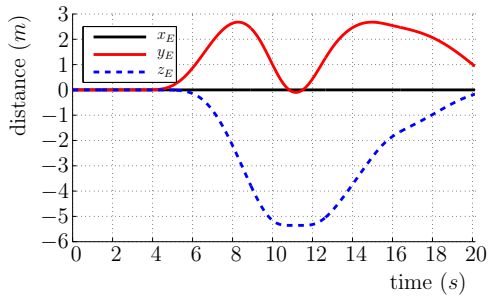


Figure B.179: OpTN - Position

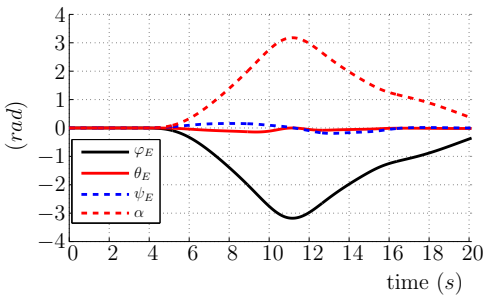


Figure B.180: OpTN - Euler angle

Response of MCA	ZyRo	$k = 0.7$

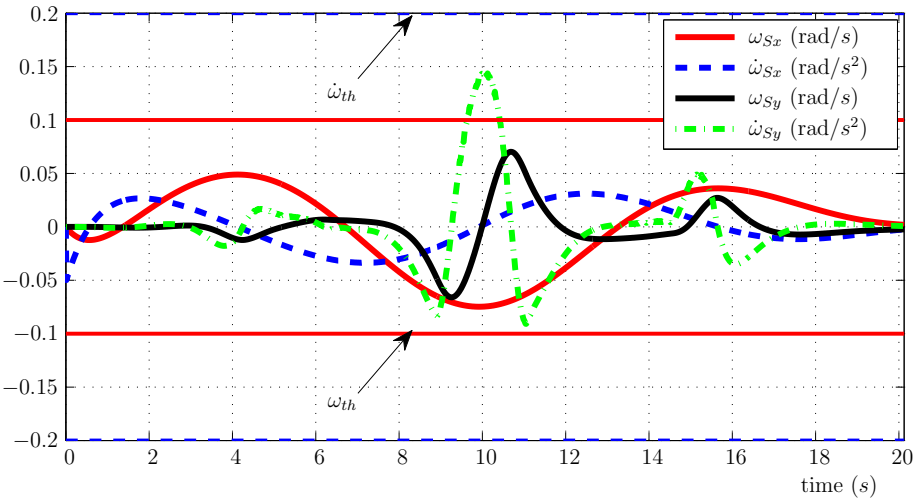


Figure B.181: ZyRo - Simulated rotational quantities

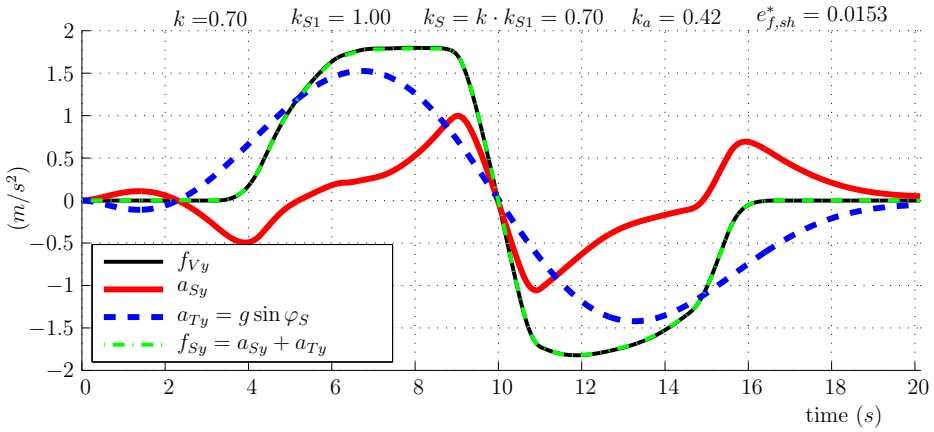


Figure B.182: ZyRo - Simulated specific forces and acceleration

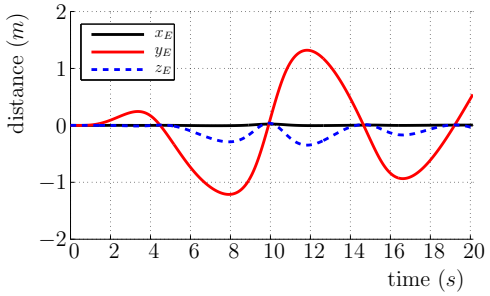


Figure B.183: ZyRo - Position

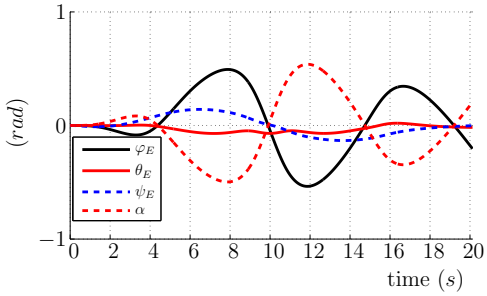


Figure B.184: ZyRo - Euler angle

Response of MCA	MPC*	$k = 0.7$

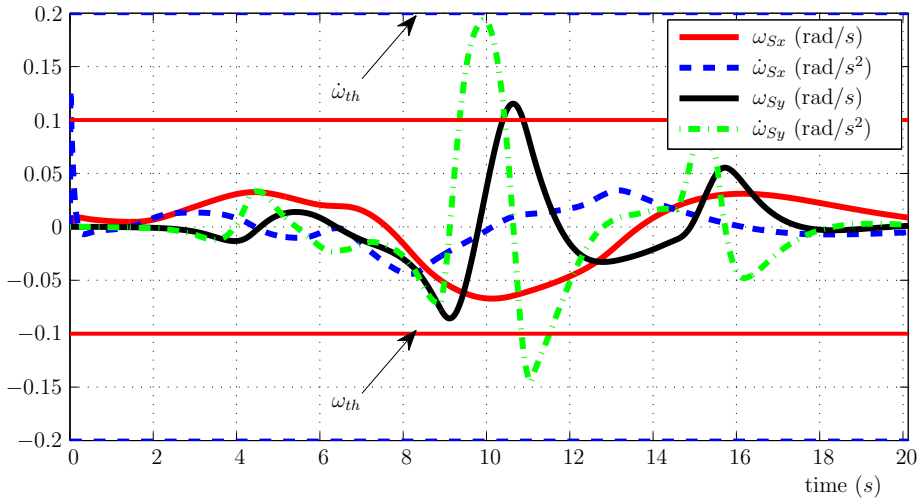


Figure B.185: MPC - Simulated rotational quantities

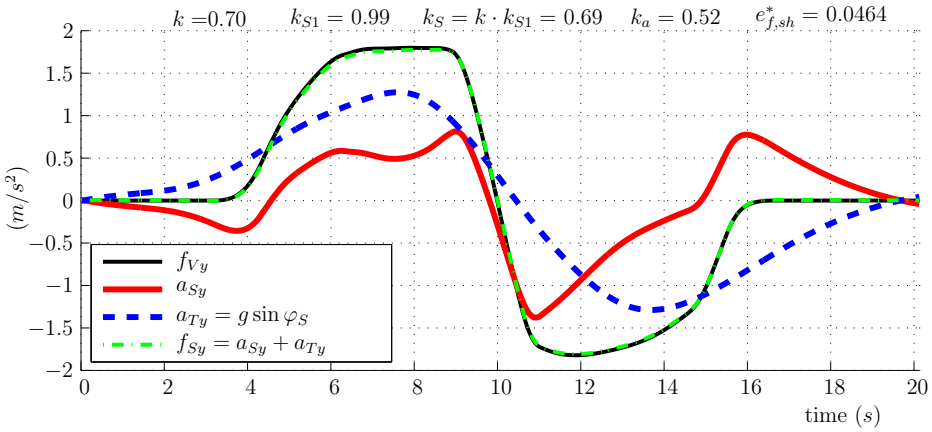


Figure B.186: MPC - Simulated specific forces and acceleration

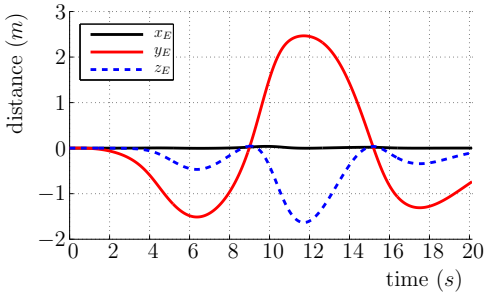


Figure B.187: MPC - Position

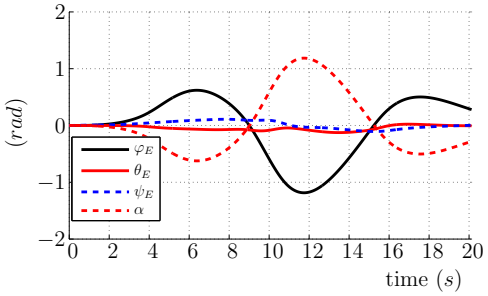


Figure B.188: MPC - Euler angle

Response of MCA	exMPCFK	$k = 0.7$

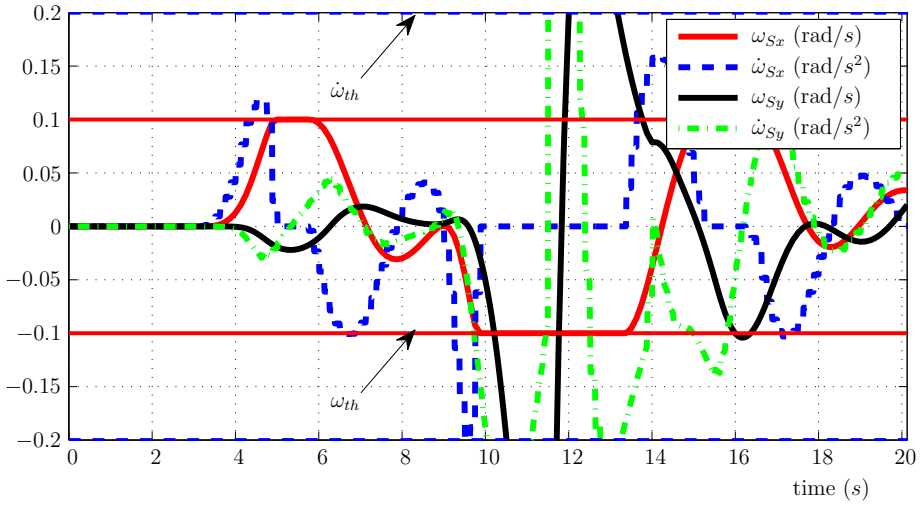


Figure B.189: exMPCFK - Simulated rotational quantities

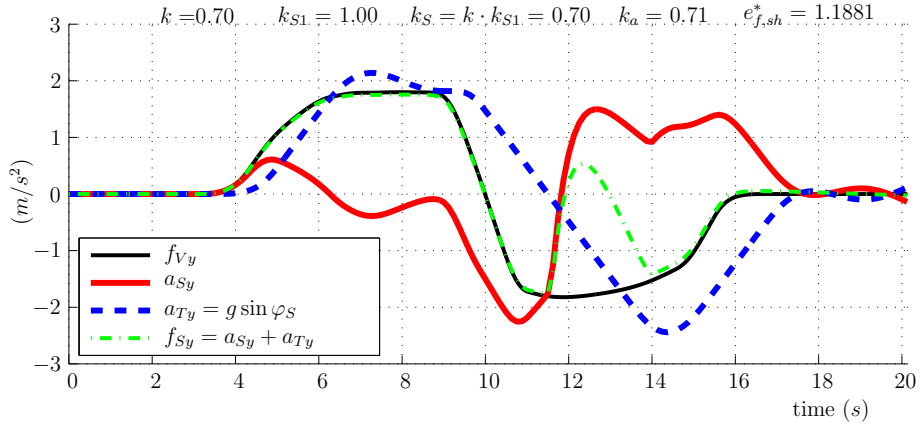


Figure B.190: exMPCFK - Simulated specific forces and acceleration

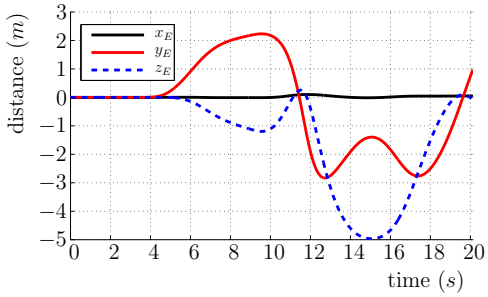


Figure B.191: exMPCFK - Position

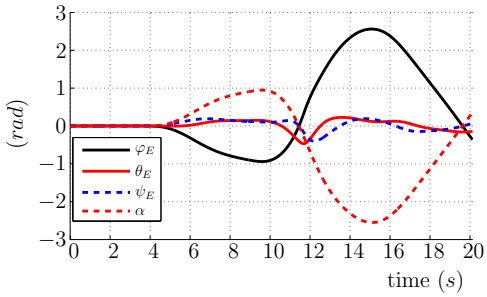


Figure B.192: exMPCFK - Euler angle

Response of MCA	CLRN	$k = 0.9$

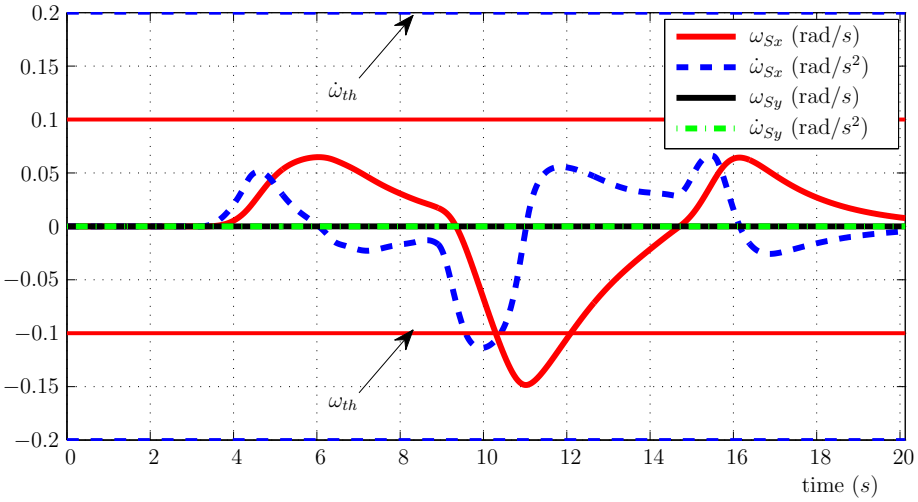


Figure B.193: CLRN - Simulated rotational quantities

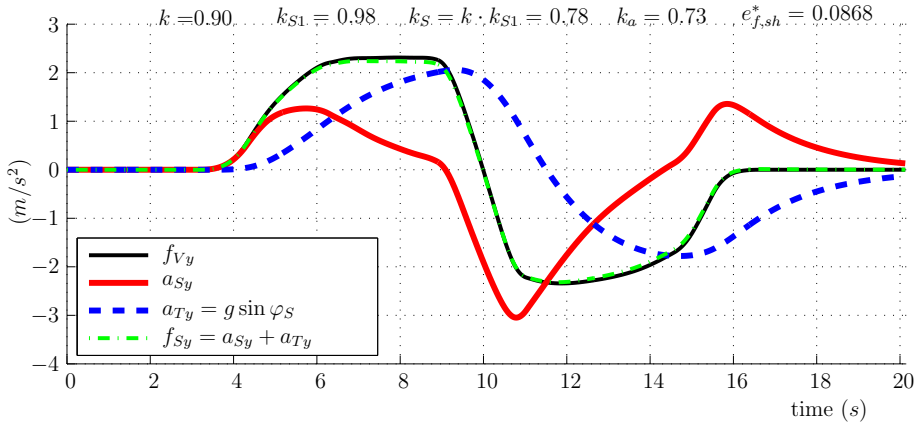


Figure B.194: CLRN - Simulated specific forces and acceleration

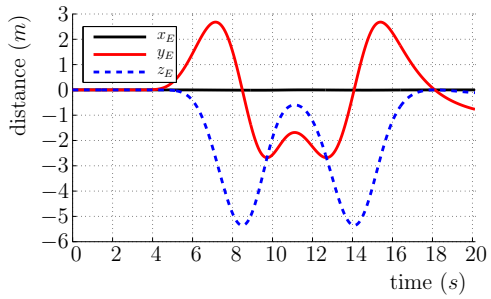


Figure B.195: CLRN - Position

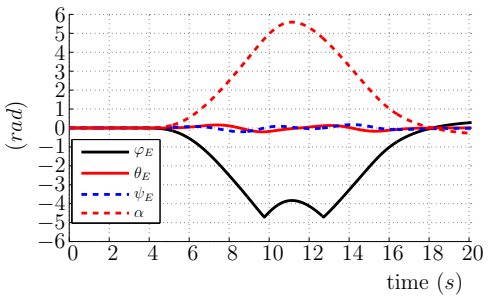


Figure B.196: CLRN - Euler angle

Response of MCA	CLG	$k = 0.9$

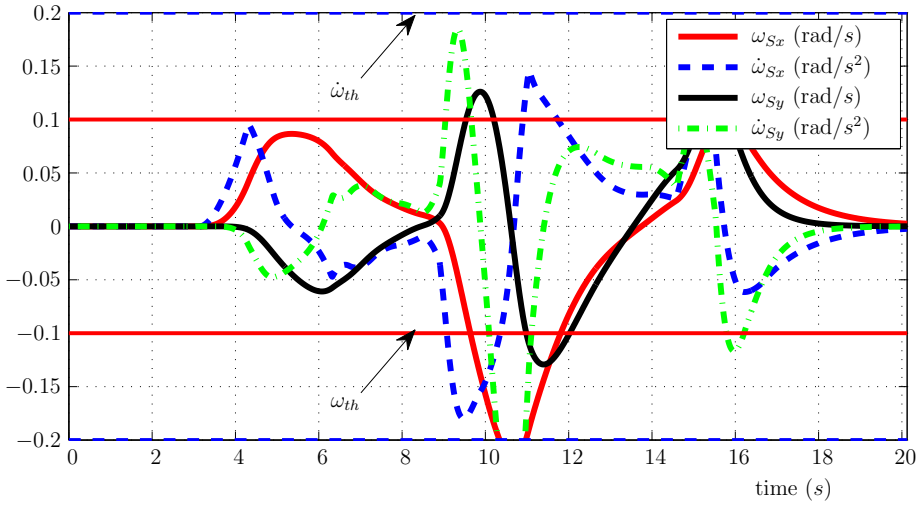


Figure B.197: CLG - Simulated rotational quantities

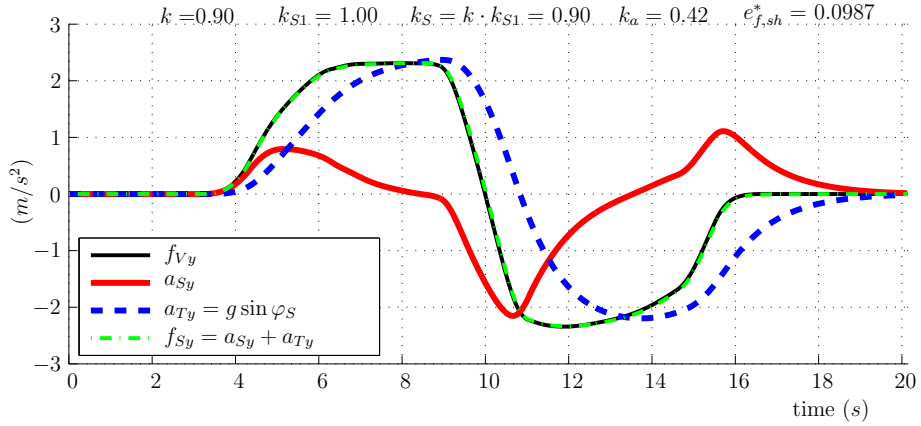


Figure B.198: CLG - Simulated specific forces and acceleration

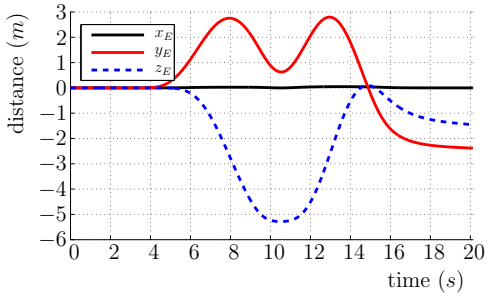


Figure B.199: CLG - Position

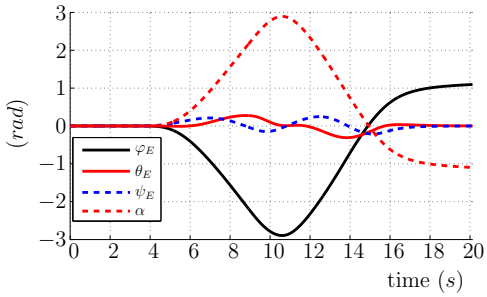


Figure B.200: CLG - Euler angle

Response of MCA	ADRN	$k = 0.9$

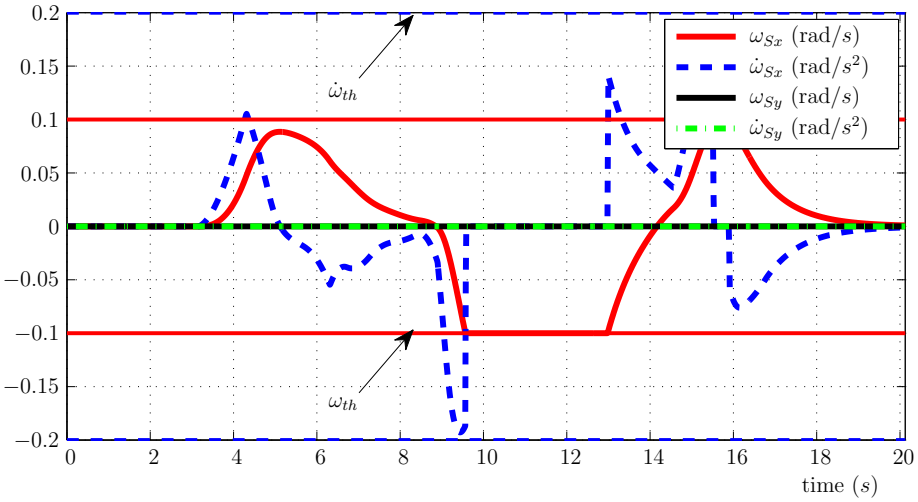


Figure B.201: ADRN - Simulated rotational quantities

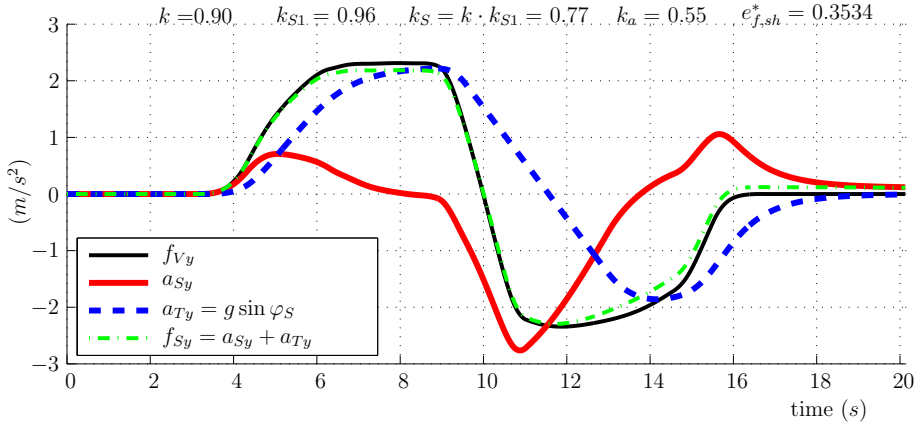


Figure B.202: ADRN - Simulated specific forces and acceleration

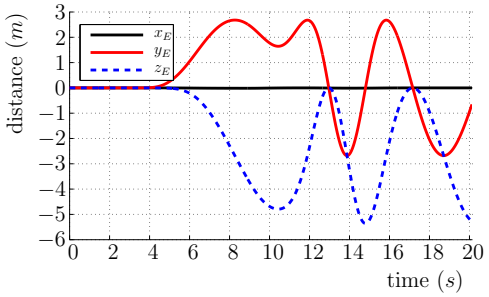


Figure B.203: ADRN - Position

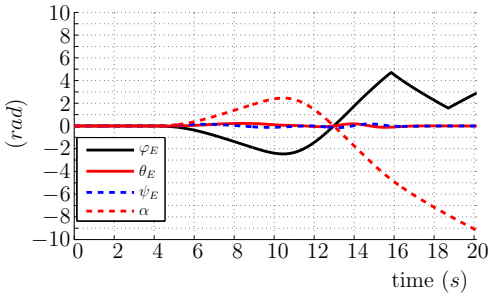


Figure B.204: ADRN - Euler angle

Response of MCA	ADSK	$k = 0.9$

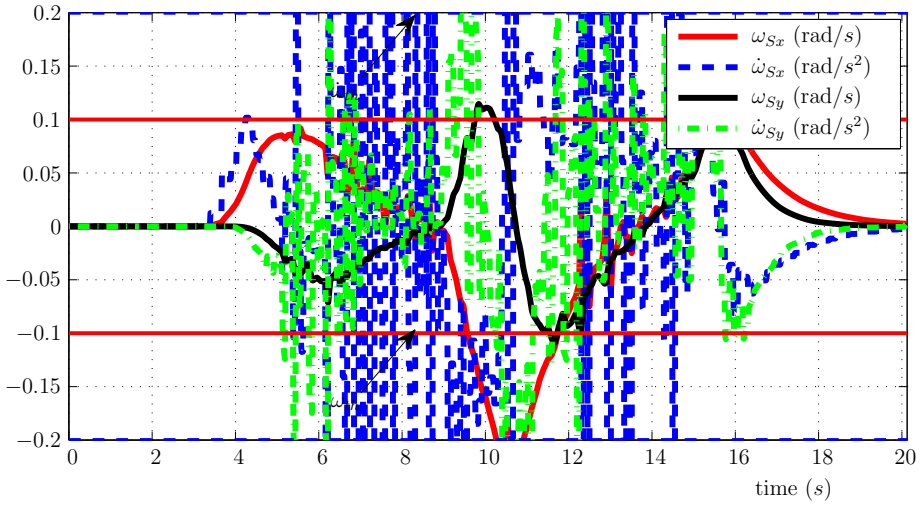


Figure B.205: ADSK - Simulated rotational quantities

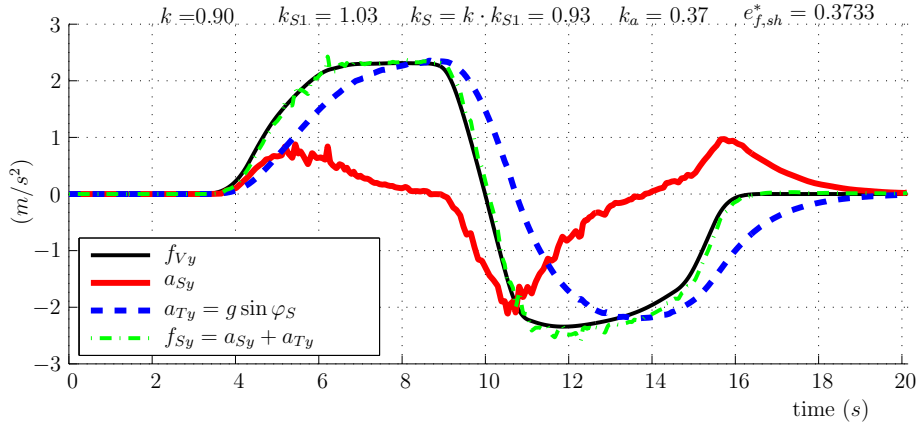


Figure B.206: ADSK - Simulated specific forces and acceleration

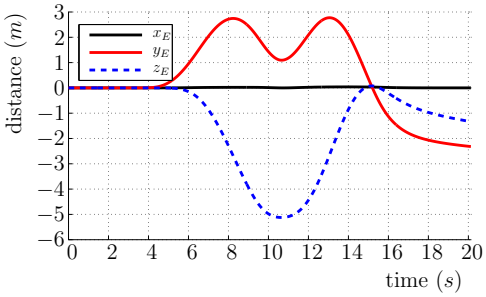


Figure B.207: ADSK - Position

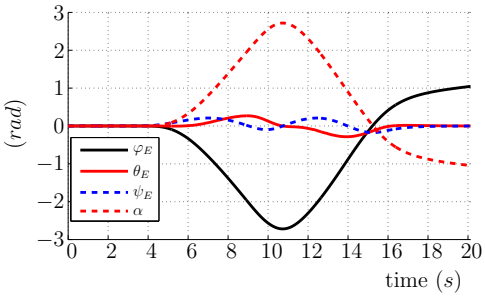


Figure B.208: ADSK - Euler angle

Response of MCA	OpTYM	$k = 0.9$

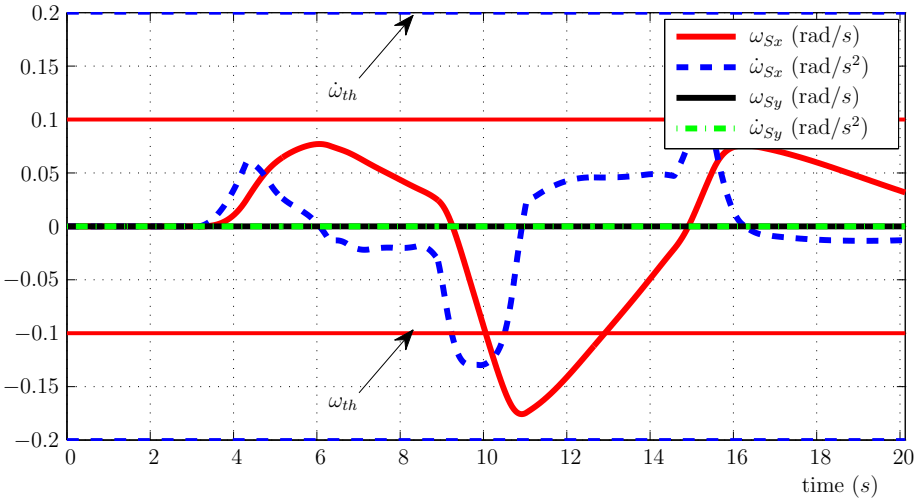


Figure B.209: OpS - Simulated rotational quantities

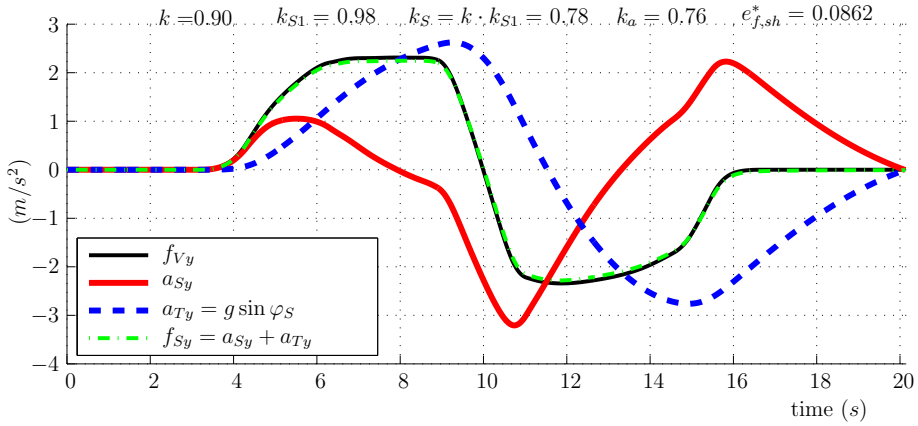


Figure B.210: OpS - Simulated specific forces and acceleration

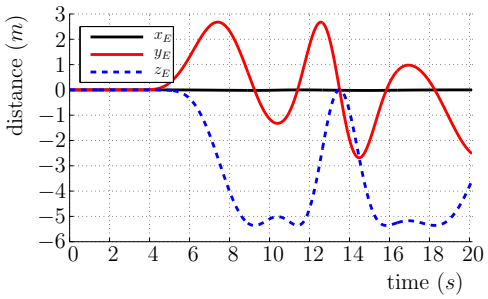


Figure B.211: OpS - Position

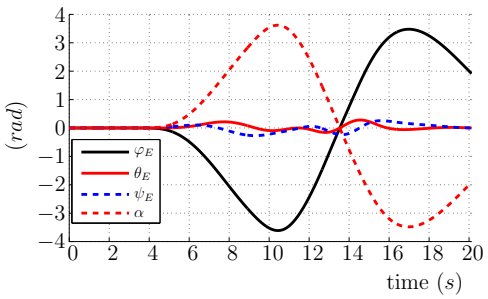


Figure B.212: OpS - Euler angle

Response of MCA	OpT	$k = 0.9$

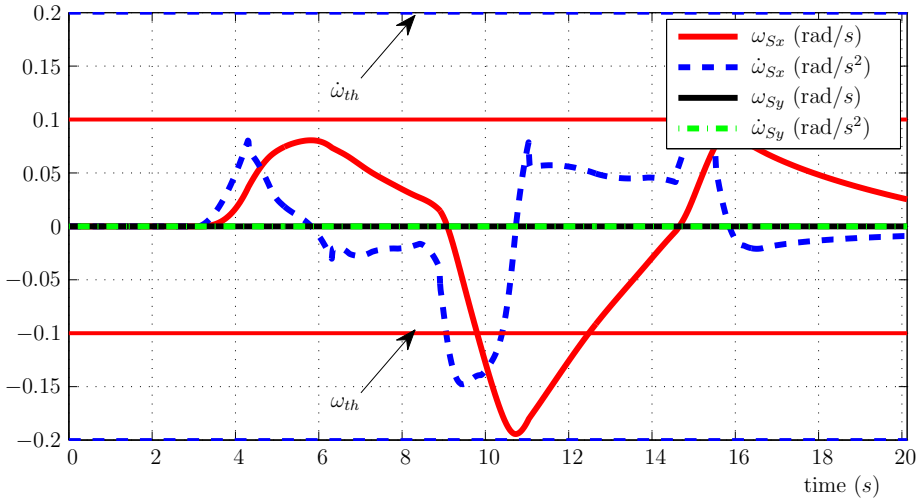


Figure B.213: OpRN - Simulated rotational quantities

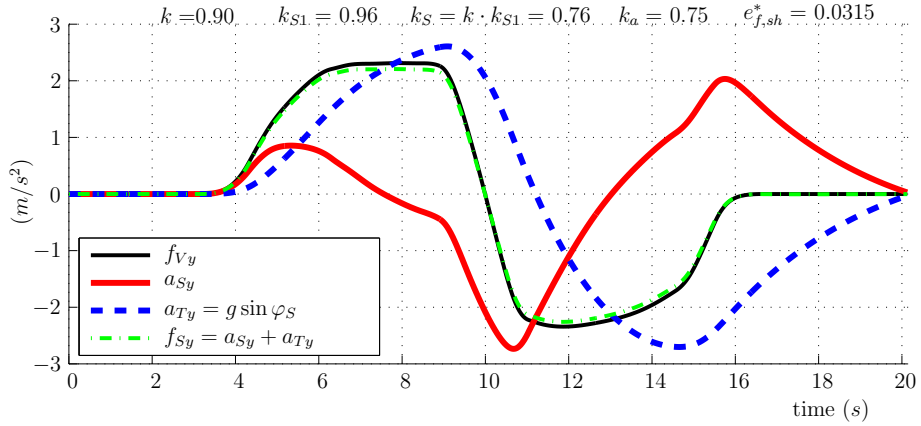


Figure B.214: OpRN - Simulated specific forces and acceleration

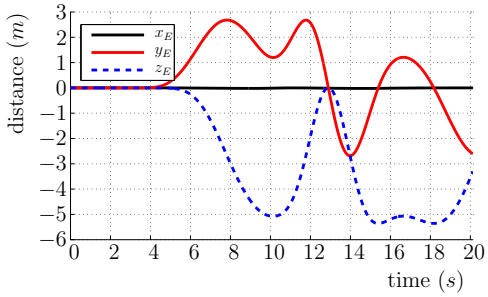


Figure B.215: OpRN - Position

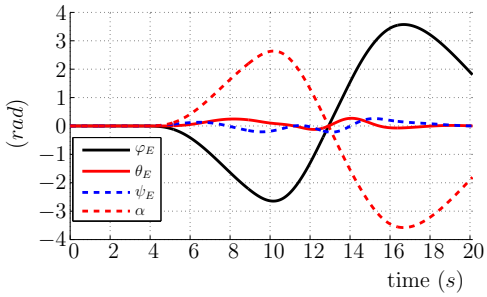


Figure B.216: OpRN - Euler angle

Response of MCA	OpS	$k = 0.9$

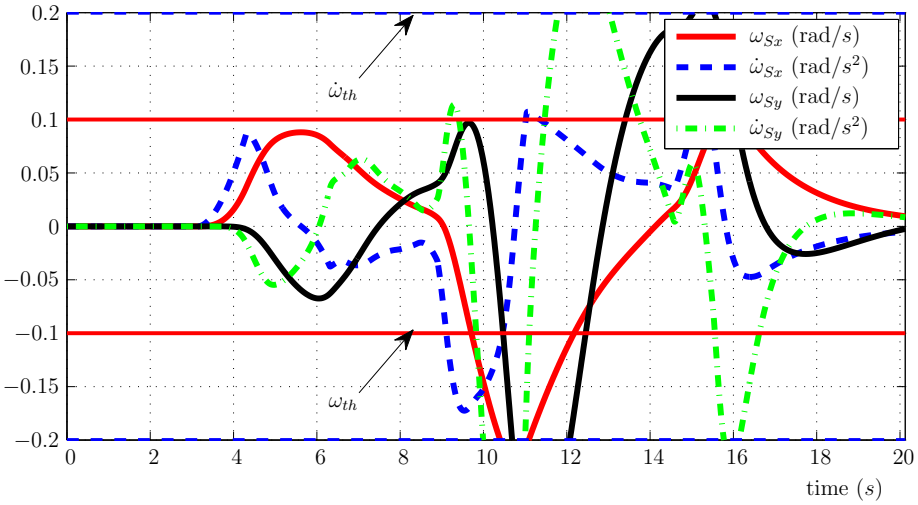


Figure B.217: OpTYM - Simulated rotational quantities

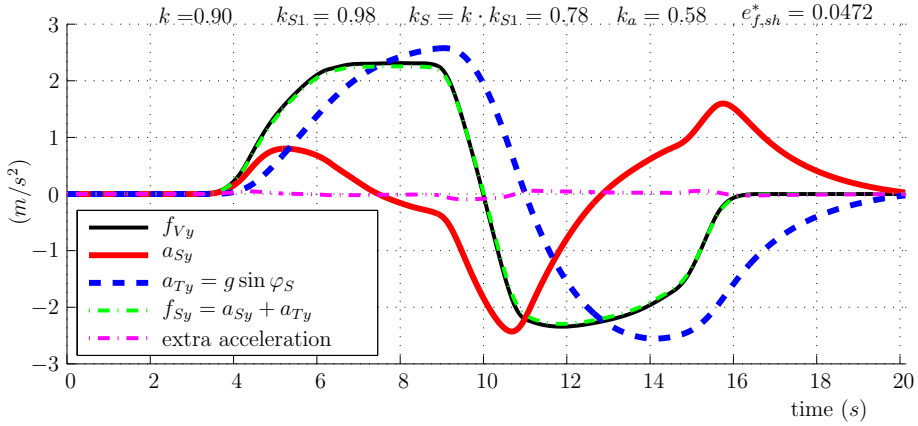


Figure B.218: OpTYM - Simulated specific forces and acceleration

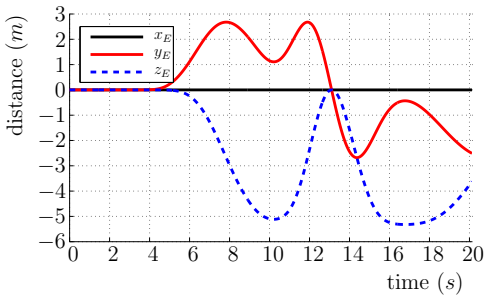


Figure B.219: OpTYM - Position

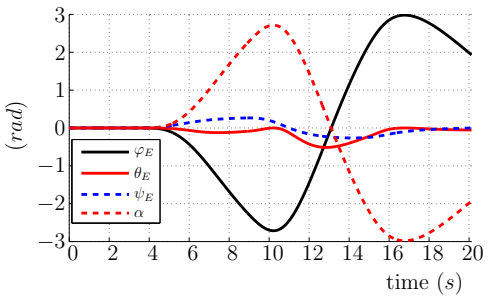


Figure B.220: OpTYM - Euler angle

Response of MCA	OpRN	$k = 0.9$

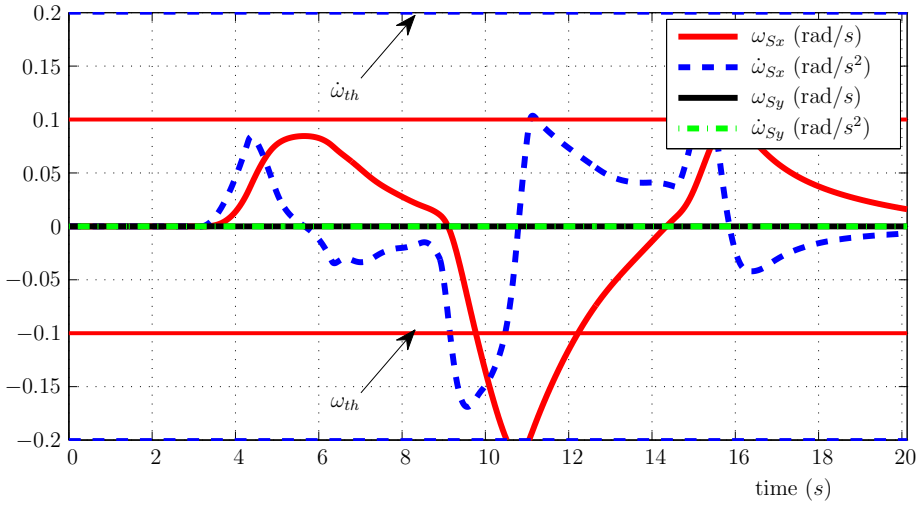


Figure B.221: OpT - Simulated rotational quantities

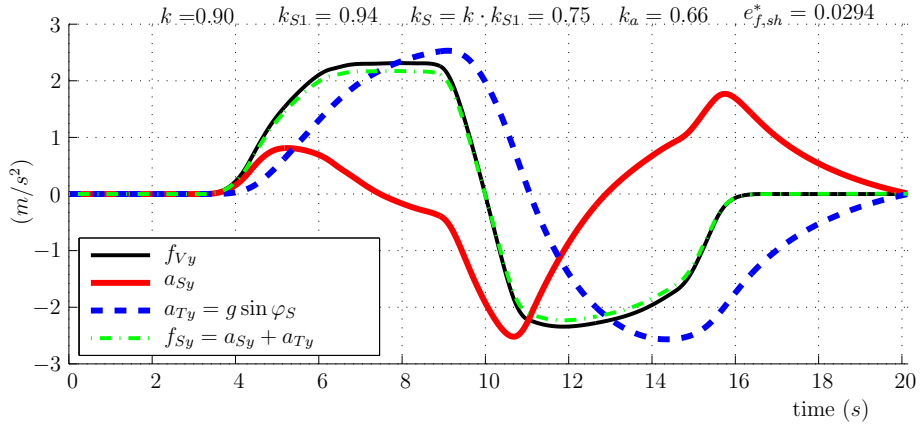


Figure B.222: OpT - Simulated specific forces and acceleration

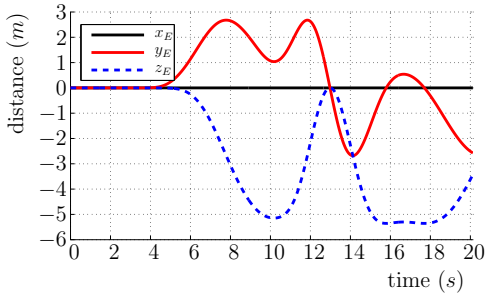


Figure B.223: OpT - Position

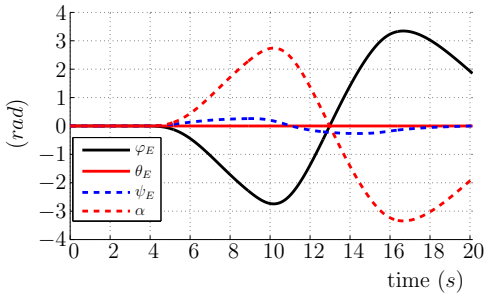


Figure B.224: OpT - Euler angle

Response of MCA	OpTNon	$k = 0.9$

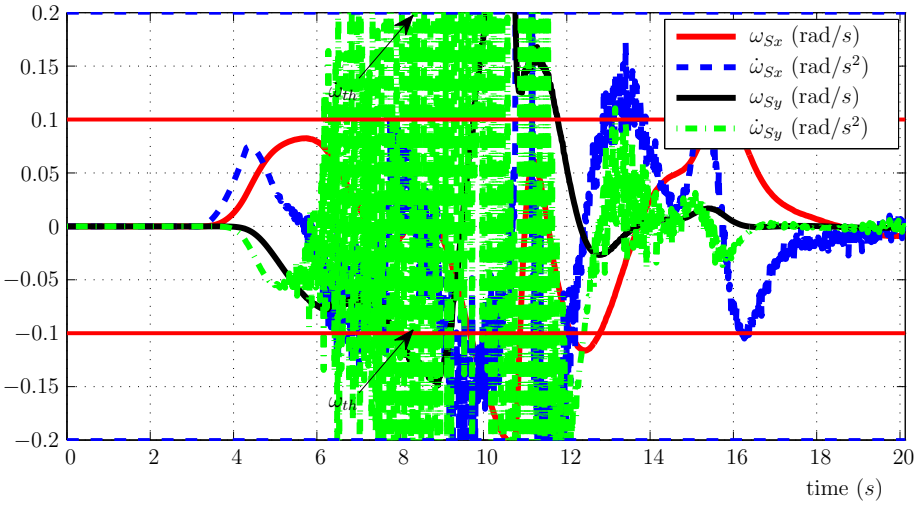


Figure B.225: OpTN - Simulated rotational quantities

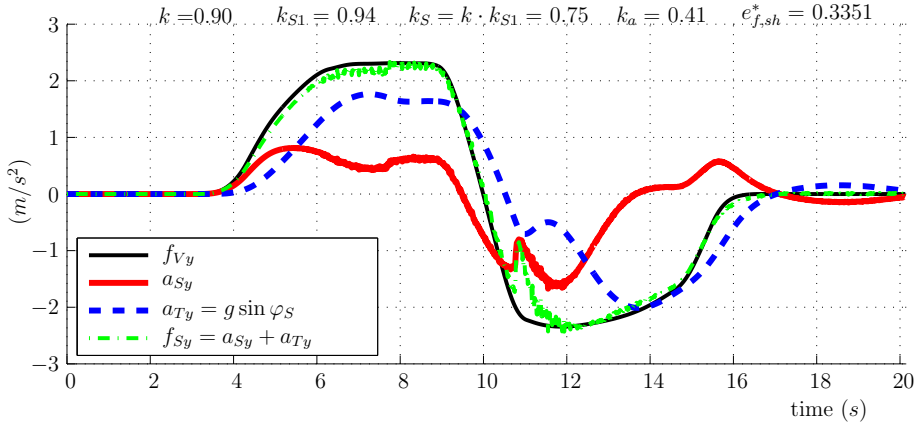


Figure B.226: OpTN - Simulated specific forces and acceleration

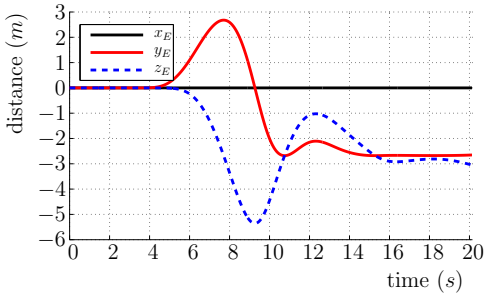


Figure B.227: OpTN - Position

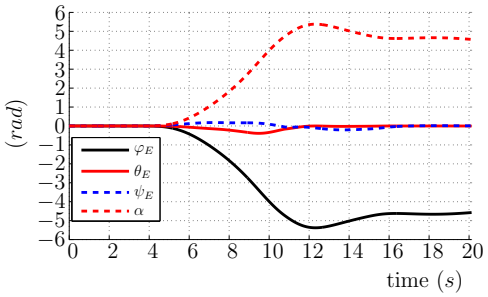


Figure B.228: OpTN - Euler angle

Response of MCA	ZyRo	$k = 0.9$

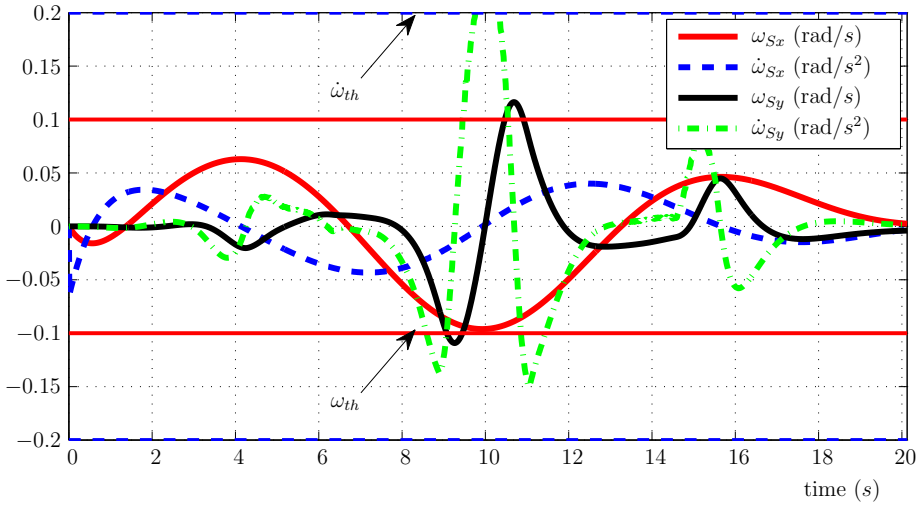


Figure B.229: ZyRo - Simulated rotational quantities

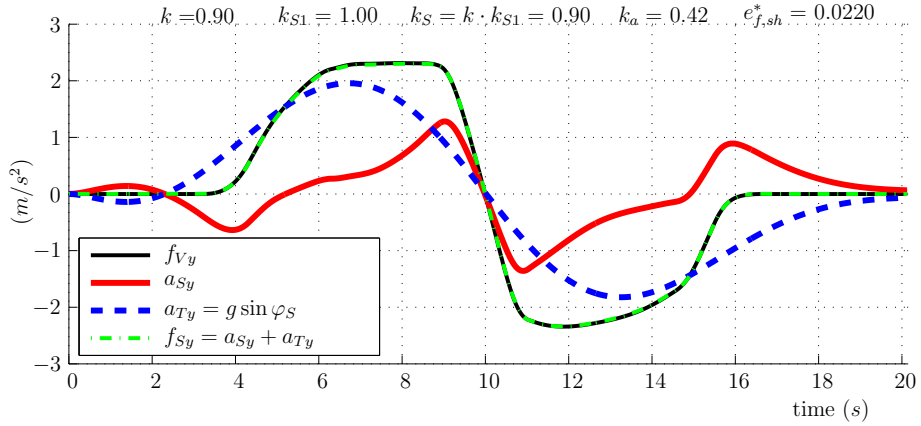


Figure B.230: ZyRo - Simulated specific forces and acceleration

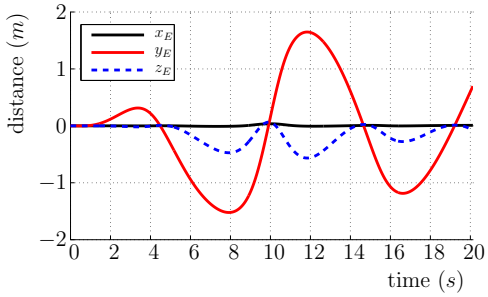


Figure B.231: ZyRo - Position

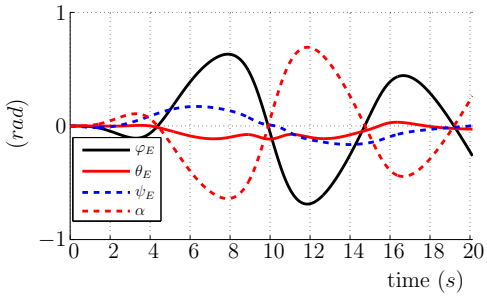


Figure B.232: ZyRo - Euler angle

Response of MCA	MPC*	$k = 0.9$

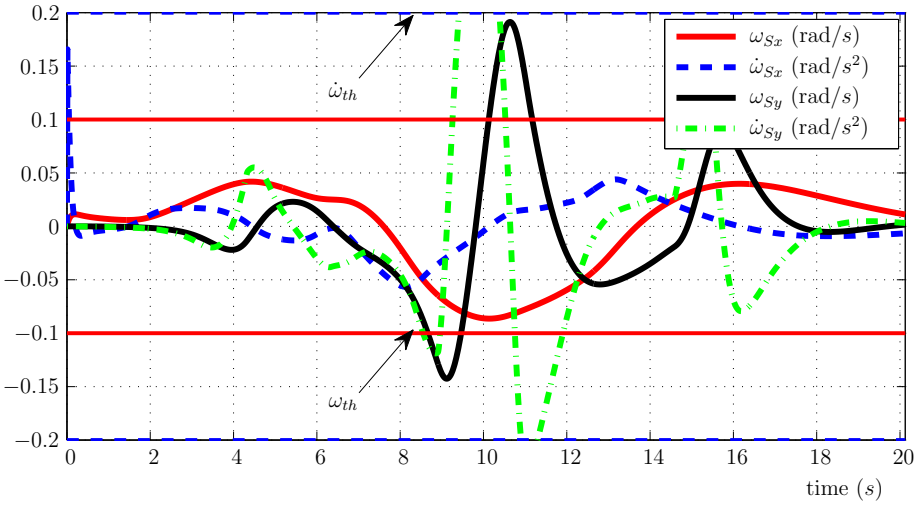


Figure B.233: MPC - Simulated rotational quantities

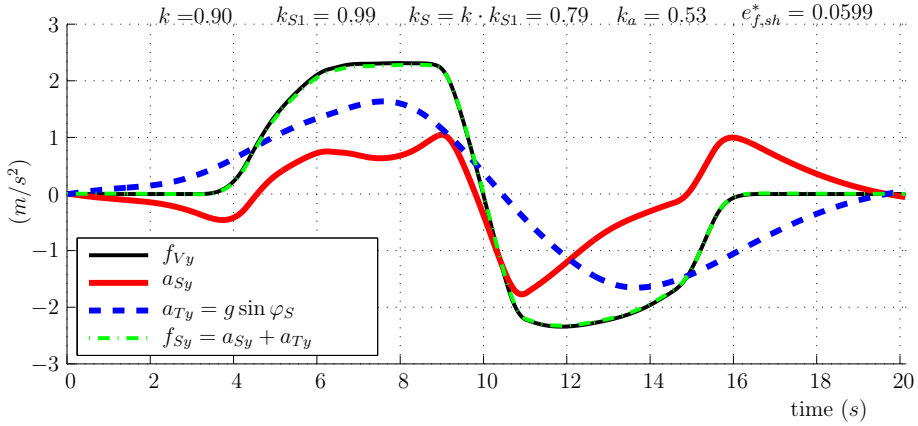


Figure B.234: MPC - Simulated specific forces and acceleration

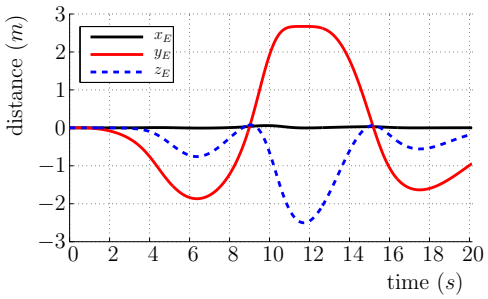


Figure B.235: MPC - Position

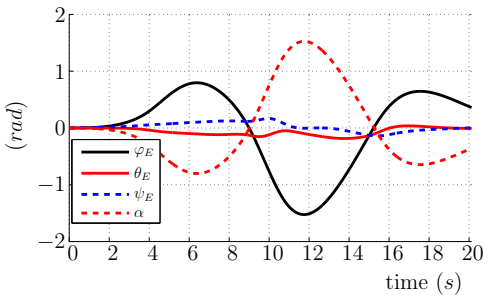


Figure B.236: MPC - Euler angle

Response of MCA	exMPCFK	$k = 0.9$

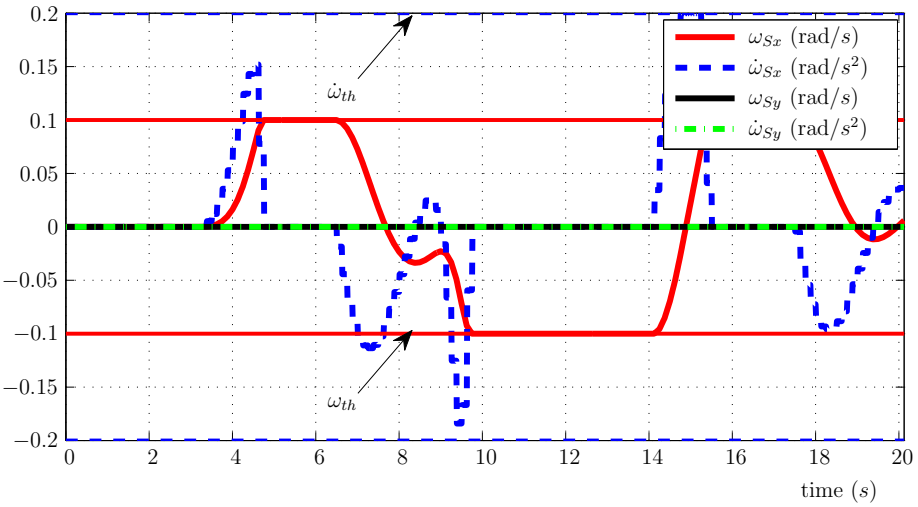


Figure B.237: exMPCFK - Simulated rotational quantities

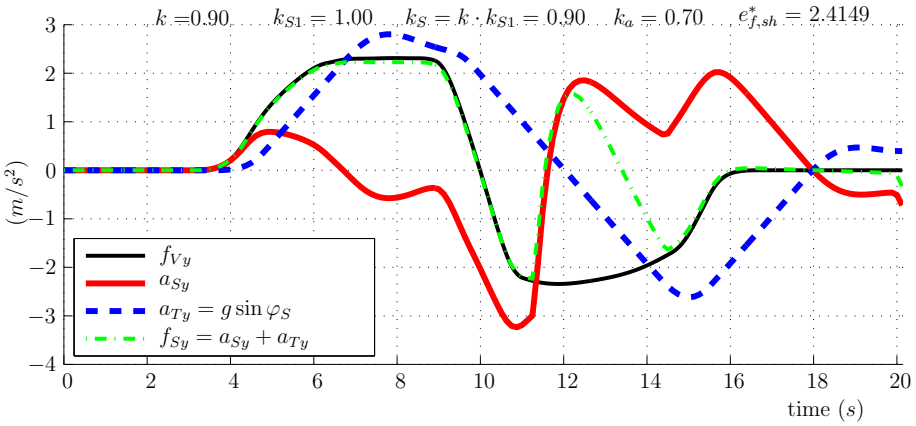


Figure B.238: exMPCFK - Simulated specific forces and acceleration

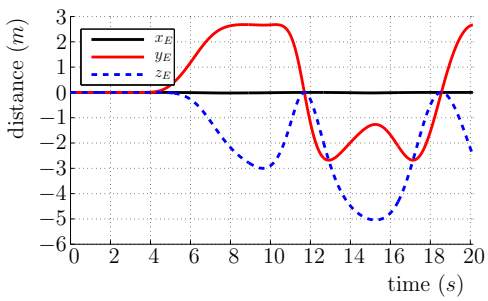


Figure B.239: exMPCFK - Position

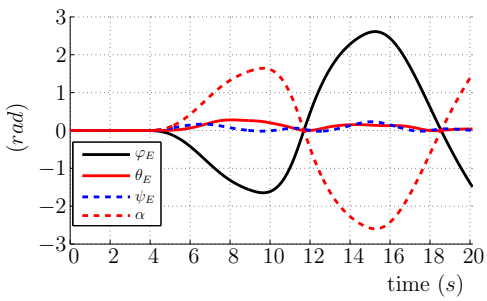


Figure B.240: exMPCFK - Euler angle

Scale	C_1^*							C_2^*		C_3^*	C_4^*	C_5^*	MF
	x_E	y_E	z_E	φ_E	θ_E	ψ_E	α	ω_{Sx}	$\dot{\omega}_{Sx}$	e_{sh}^*	k_S	k_a	
0.40	1	1	1	1	1	1	1	1	1	1	1	1	15
0.50	1	1	1	1	0	1	0	0	1	1	1	1	3
0.60	1	1	1	0	0	1	0	0	1	1	1	1	3
0.70	1	1	1	0	0	1	0	0	1	1	1	1	3
0.80	1	1	1	0	0	1	0	0	1	1	1	1	3
0.90	1	1	1	0	0	1	0	0	1	1	1	1	3
1.00	1	1	1	0	0	1	0	0	1	1	1	1	3

Table B.11: CLRN - Result of verifying criteria

Scale	C_1^*							C_2^*		C_3^*	C_4^*	C_5^*	MF
	x_E	y_E	z_E	φ_E	θ_E	ψ_E	α	ω_{Sx}	$\dot{\omega}_{Sx}$	e_{sh}^*	k_S	k_a	
0.40	1	1	1	1	1	1	1	1	1	1	1	0	14
0.50	1	1	1	1	1	1	1	0	1	1	1	0	10
0.60	1	1	1	1	0	1	1	0	1	1	1	0	2
0.70	1	1	1	1	0	1	1	0	1	1	1	0	2
0.80	1	1	1	1	0	1	1	0	1	1	1	0	2
0.90	1	1	1	1	0	1	1	0	1	1	1	0	2
1.00	1	1	1	1	0	1	0	0	1	1	0	0	0

Table B.12: CLG - Result of verifying criteria

Scale	C_1^*							C_2^*		C_3^*	C_4^*	C_5^*	MF
	x_E	y_E	z_E	φ_E	θ_E	ψ_E	α	ω_{Sx}	$\dot{\omega}_{Sx}$	e_{sh}^*	k_S	k_a	
0.40	1	1	1	1	1	1	1	1	1	1	1	0	14
0.50	1	1	1	1	1	1	1	1	1	1	1	0	14
0.60	1	1	1	1	1	1	1	1	1	1	1	0	14
0.70	1	1	1	0	0	1	0	0	1	1	1	0	2
0.80	1	1	1	0	0	1	0	1	1	1	1	1	7
0.90	1	1	1	0	0	1	0	1	1	0	1	1	3
1.00	1	1	1	0	0	1	0	1	0	0	1	1	3

Table B.13: ADRN - Result of verifying criteria

Scale	C_1^*							C_2^*		C_3^*	C_4^*	C_5^*	MF
	x_E	y_E	z_E	φ_E	θ_E	ψ_E	α	ω_{Sx}	$\dot{\omega}_{Sx}$	e_{sh}^*	k_S	k_a	
0.40	1	1	1	1	1	1	1	1	1	1	1	0	14
0.50	1	1	1	1	1	1	1	0	1	0	1	0	10
0.60	1	1	1	1	0	1	1	0	0	0	1	0	2
0.70	1	1	1	1	0	1	1	0	0	0	1	0	2
0.80	1	1	1	1	0	1	1	0	0	0	1	0	2
0.90	1	1	1	1	0	1	1	0	0	0	1	0	2
1.00	1	1	1	1	0	1	0	0	0	0	0	0	0

Table B.14: ADSK - Result of verifying criteria

Scale	C_1^*							C_2^*		C_3^*	C_4^*	C_5^*	MF
	x_E	y_E	z_E	φ_E	θ_E	ψ_E	α	ω_{Sx}	$\dot{\omega}_{Sx}$	e_{sh}^*	k_S	k_a	
0.40	1	1	1	1	1	1	1	1	1	1	1	1	15
0.50	1	1	1	1	0	1	1	0	1	1	1	1	3
0.60	1	1	1	1	0	1	1	0	1	1	1	1	3
0.70	1	1	1	1	0	1	1	0	1	1	1	1	3
0.80	1	1	1	1	0	1	0	0	1	1	1	1	3
0.90	1	1	1	0	0	1	0	0	1	1	1	1	3
1.00	1	1	1	0	0	1	0	0	1	1	1	1	3

Table B.15: OpS - Result of verifying criteria

Scale	C_1^*							C_2^*		C_3^*	C_4^*	C_5^*	MF
	x_E	y_E	z_E	φ_E	θ_E	ψ_E	α	ω_{Sx}	$\dot{\omega}_{Sx}$	e_{sh}^*	k_S	k_a	
0.40	1	1	1	1	1	1	1	1	1	1	1	1	15
0.50	1	1	1	1	0	1	1	0	1	1	1	1	3
0.60	1	1	1	1	0	1	1	0	1	1	1	1	3
0.70	1	1	1	1	0	1	1	0	1	1	1	1	3
0.80	1	1	1	1	0	1	0	0	1	1	1	1	3
0.90	1	1	1	0	0	1	0	0	1	1	1	1	3
1.00	1	1	1	0	0	1	0	0	1	1	1	1	3

Table B.16: OpRN - Result of verifying criteria

Scale	C_1^*							C_2^*		C_3^*	C_4^*	C_5^*	MF
	x_E	y_E	z_E	φ_E	θ_E	ψ_E	α	ω_{Sx}	$\dot{\omega}_{Sx}$	e_{sh}^*	k_S	k_a	
0.40	1	1	1	1	1	1	1	1	1	1	1	1	15
0.50	1	1	1	1	0	1	1	0	1	1	1	1	3
0.60	1	1	1	1	0	1	1	0	1	1	1	1	3
0.70	1	1	1	1	0	1	1	0	1	1	1	1	3
0.80	1	1	1	1	0	1	1	0	1	1	1	1	3
0.90	1	1	1	1	0	1	0	0	1	1	1	1	3
1.00	1	1	1	1	1	1	0	0	1	1	1	1	3

Table B.17: OpTYM - Result of verifying criteria

Scale	C_1^*							C_2^*		C_3^*	C_4^*	C_5^*	MF
	x_E	y_E	z_E	φ_E	θ_E	ψ_E	α	ω_{Sx}	$\dot{\omega}_{Sx}$	e_{sh}^*	k_S	k_a	
0.40	1	1	1	1	1	1	1	1	1	1	1	1	15
0.50	1	1	1	1	0	1	1	0	1	1	1	1	3
0.60	1	1	1	1	0	1	1	0	1	1	1	1	3
0.70	1	1	1	1	0	1	1	0	1	1	1	1	3
0.80	1	1	1	1	0	1	0	0	1	1	1	1	3
0.90	1	1	1	1	1	1	0	0	1	1	1	1	3
1.00	1	1	1	0	1	1	0	0	1	1	1	1	3

Table B.18: OpT - Result of verifying criteria

Scale	C_1^*							C_2^*		C_3^*	C_4^*	C_5^*	MF
	x_E	y_E	z_E	φ_E	θ_E	ψ_E	α	ω_{Sx}	$\dot{\omega}_{Sx}$	e_{sh}^*	k_S	k_a	
0.40	1	1	1	1	1	1	1	1	1	1	1	0	14
0.50	1	1	1	1	1	1	1	0	1	1	1	0	10
0.60	1	1	1	1	1	1	1	0	1	1	1	0	10
0.70	1	1	1	1	0	1	0	0	0	0	1	0	2
0.80	1	1	1	0	0	1	0	0	0	0	1	0	2
0.90	1	1	1	0	0	1	0	0	0	0	1	0	2
1.00	1	1	1	0	1	0	0	0	0	0	1	1	3

Table B.19: OpTN - Result of verifying criteria

Scale	C_1^*							C_2^*		C_3^*	C_4^*	C_5^*	MF
	x_E	y_E	z_E	φ_E	θ_E	ψ_E	α	ω_{Sx}	$\dot{\omega}_{Sx}$	e_{sh}^*	k_S	k_a	
0.40	1	1	1	1	1	1	1	1	1	1	1	0	14
0.50	1	1	1	1	1	1	1	1	1	1	1	0	14
0.60	1	1	1	1	1	1	1	1	1	1	1	0	14
0.70	1	1	1	1	1	1	1	1	1	1	1	0	14
0.80	1	1	1	1	1	1	1	1	1	1	1	0	14
0.90	1	1	1	1	1	1	1	0	1	1	1	0	10
1.00	1	1	1	1	1	1	1	0	1	1	1	0	10

Table B.20: ZyRo - Result of verifying criteria

Scale	C_1^*							C_2^*		C_3^*	C_4^*	C_5^*	MF
	x_E	y_E	z_E	φ_E	θ_E	ψ_E	α	ω_{Sx}	$\dot{\omega}_{Sx}$	e_{sh}^*	k_S	k_a	
0.40	1	1	1	1	1	1	1	1	1	1	1	1	15
0.50	1	1	1	1	1	1	1	1	1	1	1	1	15
0.60	1	1	1	1	1	1	1	1	1	1	1	1	15
0.70	1	1	1	1	1	1	1	0	1	1	1	1	11
0.80	1	1	1	1	1	1	1	0	1	1	1	1	11
0.90	1	1	1	1	1	1	1	0	1	1	1	1	11
1.00	1	1	1	1	0	1	1	1	1	1	1	1	7

Table B.21: MPC - Result of verifying criteria

Scale	C_1^*							C_2^*		C_3^*	C_4^*	C_5^*	MF
	x_E	y_E	z_E	φ_E	θ_E	ψ_E	α	ω_{Sx}	$\dot{\omega}_{Sx}$	e_{sh}^*	k_S	k_a	
0.40	1	1	1	1	1	1	1	1	1	1	1	1	15
0.50	1	1	1	1	0	1	1	0	1	1	1	1	3
0.60	1	1	1	1	0	1	1	0	1	0	1	1	3
0.70	1	1	1	1	0	1	1	0	1	0	1	1	3
0.80	1	1	1	1	0	1	1	0	1	0	1	1	3
0.90	1	1	1	1	0	1	1	1	1	0	1	1	3
1.00	1	1	1	1	0	1	1	0	1	0	1	1	3

Table B.22: exMPCFK - Result of verifying criteria

C Auto-tuning method examples

In the section, responses of the ZyRo algorithm with the auto-tuned parameters in section 5.5.3 are illustrated. Furthermore, the optimized parameters obtained from auto-tuning process for CLRN and CLG are described with their responses.

Selected limit-values						
$k_O\,(-)$	$\delta_O\,(m/s^2)$	$\delta_S\,(rad/s)$	$\dot{\omega}_{th}\,(rad/s^2)$	$k_{min}\,(-)$	$k_{max}\,(-)$	$k_{a,min}\,(-)$
1	0.17	0.1	0.2	0.4	1	0.6

Searching ranges of the tuned parameters					
	$k_S\,(-)$	$\omega_{ny}\,(rad/s)$	$\zeta_y\,(-)$	$\omega_{by}\,(rad/s)$	ω_{LPy}
AT1	[0.4 , 1]	[0 , 1]	[0 , 1]	[0 , 1]	[0 , 3]
AT2	[0.4 , 0.4]	[0 , 2]	[0 , 1]	[0 , 1]	[0 , 3]

Weighting parameters							
	wJ_P	wJ_ω	$wJ_{\dot{\omega}}$	wJ_{sc}	wJ_{tr}	wJ_{fsh}	wJ_{Wo}
AT1	2^8	2^3	2^3	2^2	1	2^3	0
AT2	2^8	2^3	2^3	2^2	1	2^1	0

Optimized parameters found by the auto-tuning with MVMO					
	$k^*\,(-)$	$\omega^*_{ny}\,(rad/s)$	$\zeta^*_y\,(-)$	$\omega^*_{by}\,(rad/s)$	ω^*_{LPy}
AT1	0.54	0.1563	0.0166	0.3437	1.6583
AT2	0.4	1.6583	0.6183	0.6183	2.7865

Table C.1: Limited values, searching ranges, and weighting parameters as well as optimized parameters of CLRN algorithm

Selected limit-values						
$k_O\,(-)$	$\delta_O\,(m/s^2)$	$\delta_S\,(rad/s)$	$\dot{\omega}_{th}\,(rad/s^2)$	$k_{min}\,(-)$	$k_{max}\,(-)$	$k_{a,min}\,(-)$
1	0.17	0.1	0.2	0.4	1	0.6

Searching ranges of the tuned parameters				
	$k_S\,(-)$	$\omega_{ny}\,(rad/s)$	$\zeta_y\,(-)$	$\omega_{by}\,(rad/s)$
AT1	[0.4 , 1]	[0.001 , 1]	[0.01 , 10]	[0.01 , 1]
AT2	[0.4 , 0.4]	[0 , 1]	[0.01 , 2]	[0.3 , 1]

Weighting parameters							
	wJ_P	wJ_ω	$wJ_{\dot{\omega}}$	wJ_{sc}	wJ_{tr}	wJ_{fsh}	wJ_{Wo}
AT1	2^8	2^3	2^3	2^2	1	2^3	0
AT2	2^8	2^3	2^3	2^2	1	2^1	0

Optimized parameters found by the auto-tuning with MVMO				
	$k^*\,(-)$	$\omega^*_{ny}\,(rad/s)$	$\zeta^*_y\,(-)$	$\omega^*_{by}\,(rad/s)$
AT1	0.4796	0.1248	1.2183	0.2978
AT2	0.4	0.1128	1.3385	0.3968

Table C.2: Limited values, searching ranges, and weighting parameters as well as optimized parameters of CLG algorithm

Tuning method	CLG			CLRN		
	$k_S\,(-)$	k_a	e^*_{sh}	$k_S\,(-)$	k_a	e^*_{sh}
Trial-and-error parameters	0.4	0.43	0.0167	0.39	0.73	0.0255
Auto-tuned parameters	0.4	0.62	0.0214	0.39	0.43	0.0525

Table C.3: Comparison of key elements of the results of two set parameters

Response of ZyRo	Auto-tuning	**AT1**

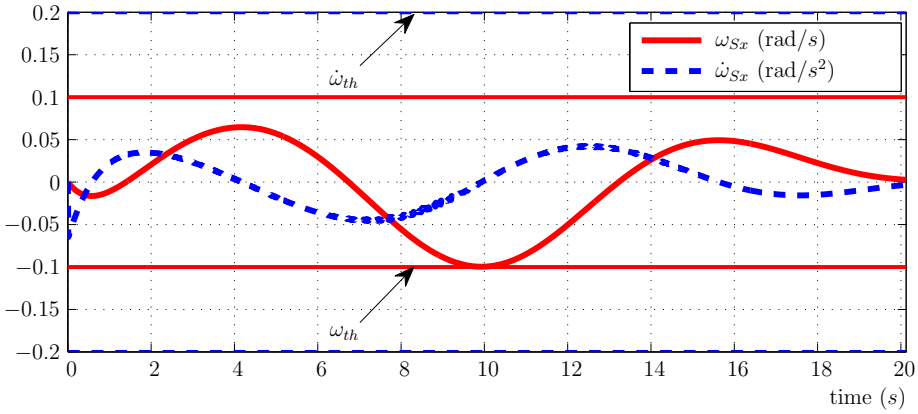


Figure C.1: ZyRo - **AT1** - Simulated rotatinal quantities

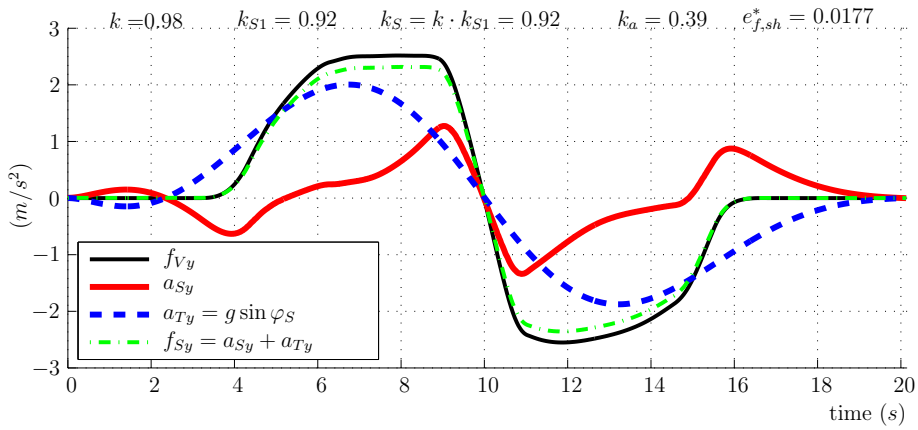


Figure C.2: ZyRo - **AT1** - Simulated specific forces and acceleration

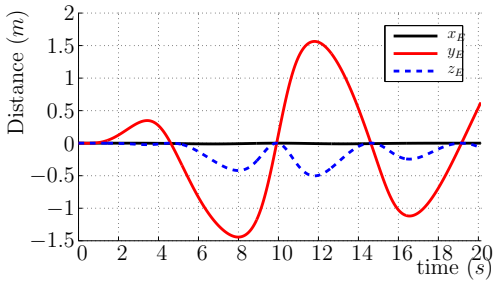


Figure C.3: ZyRo - **AT1**

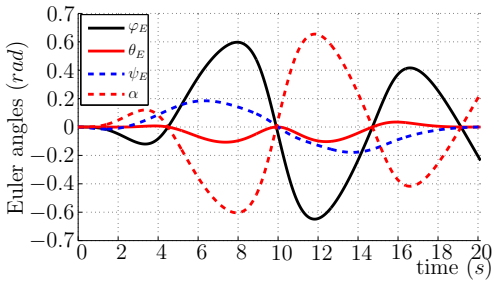


Figure C.4: ZyRo - **AT1**

Response of ZyRo	Auto-tuning	**AT2**

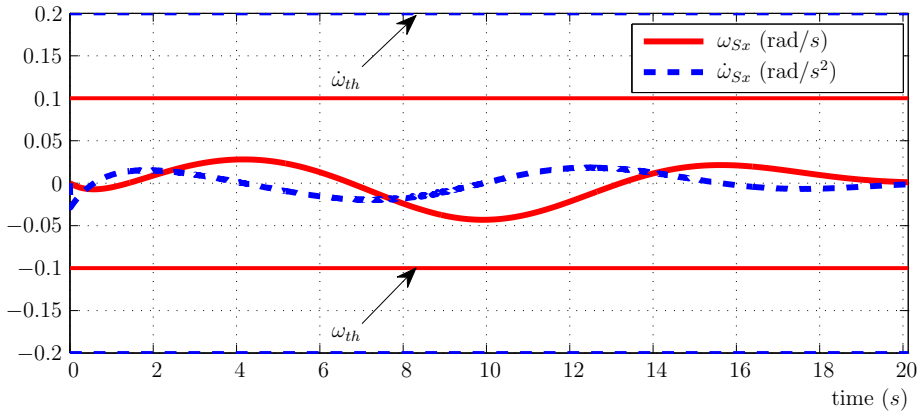


Figure C.5: ZyRo - **AT2** - Simulated angular velocity and acceleration

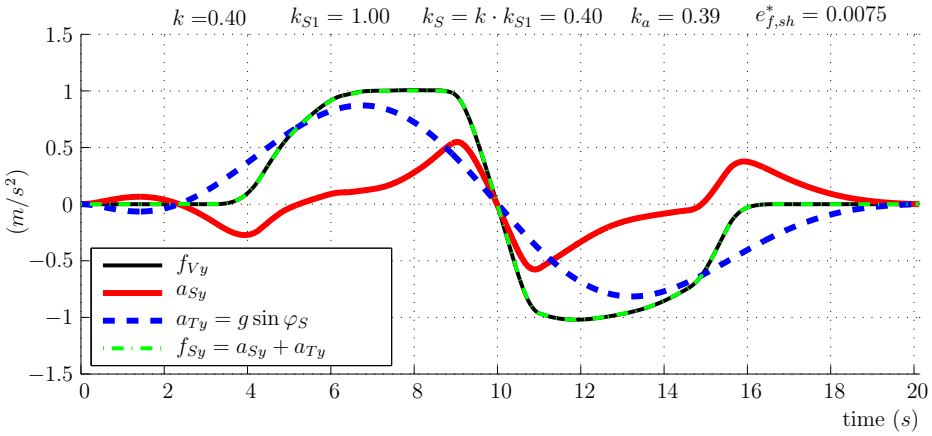


Figure C.6: ZyRo - Simulated specific forces and acceleration

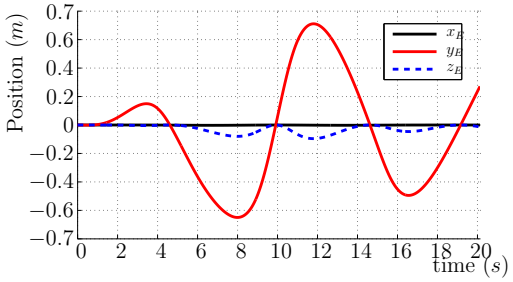


Figure C.7: ZyRo - **AT2**

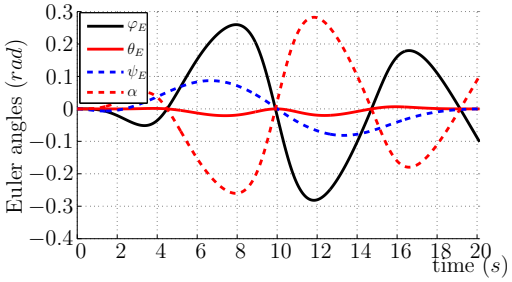


Figure C.8: ZyRo - **AT2**

Response of ZyRo	Auto-tuning	**AT3**

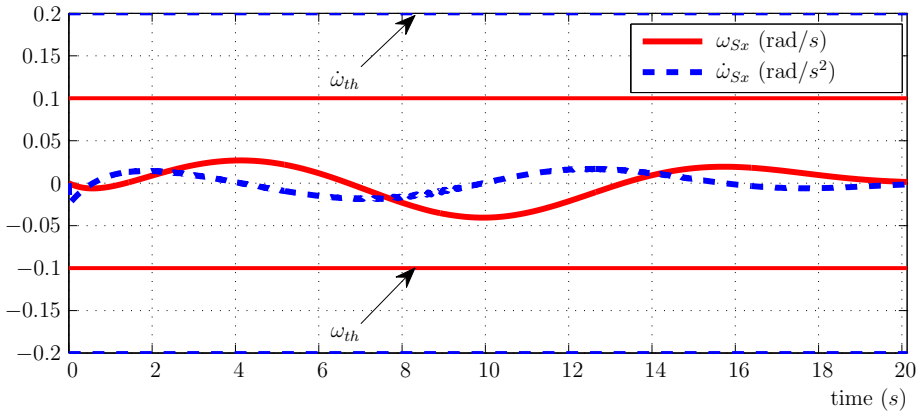


Figure C.9: ZyRo - **AT3** - Simulated rotational quantities

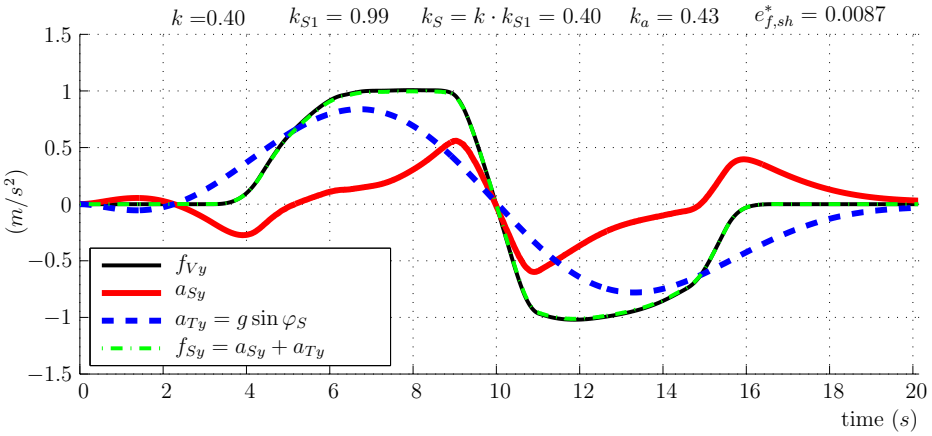


Figure C.10: ZyRo - **AT3** - Simulated specific forces and acceleration

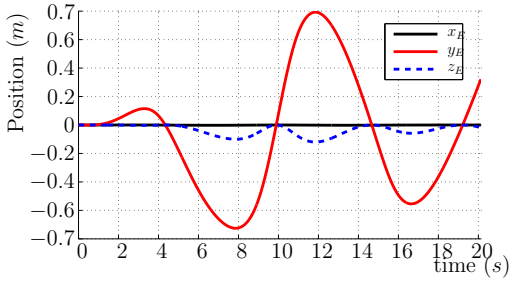


Figure C.11: ZyRo - **AT3**

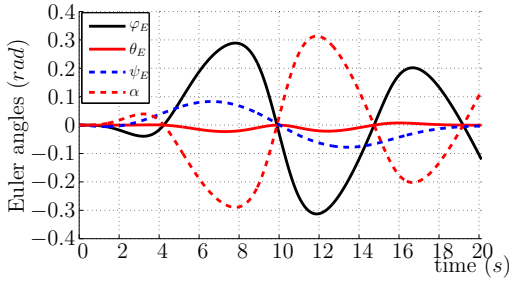


Figure C.12: ZyRo - **AT3**

Response of CLRN	Auto-tuning	**AT1**

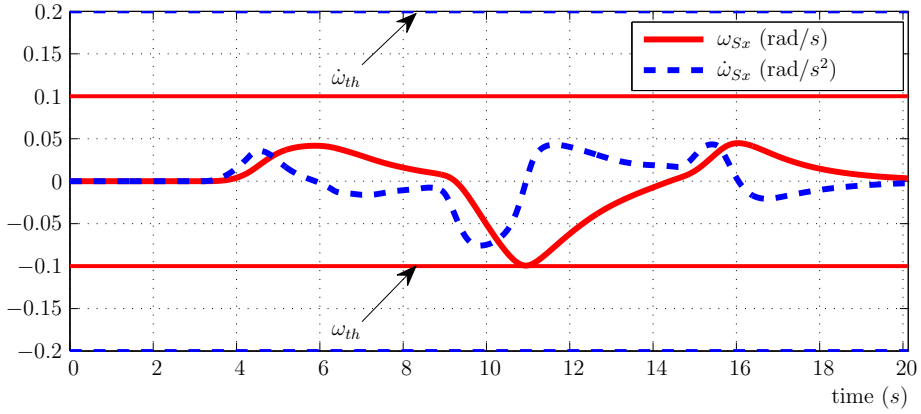


Figure C.13: CLRN - **AT1** Simulated rotational quantities

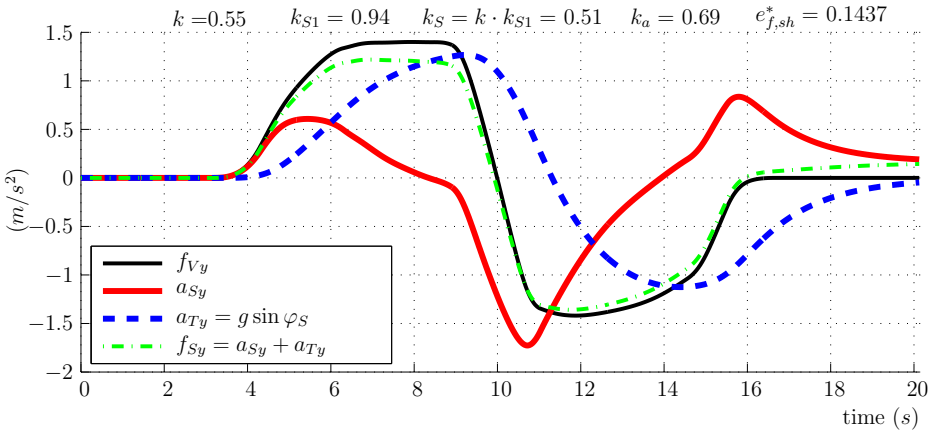


Figure C.14: CLRN - **AT1** - Simulated specific forces and acceleration

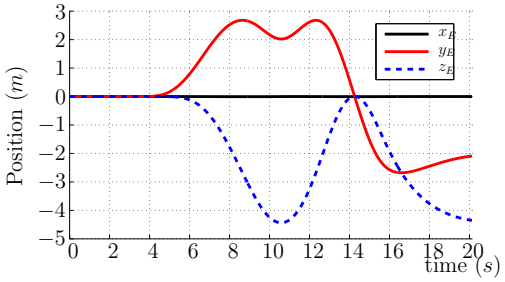


Figure C.15: CLRN - **AT1**

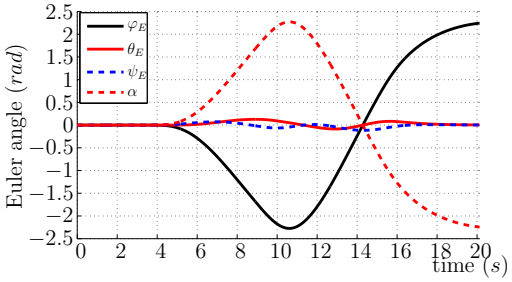


Figure C.16: CLRN - **AT1**

Response of CLRN	Auto-tuning	**AT2**

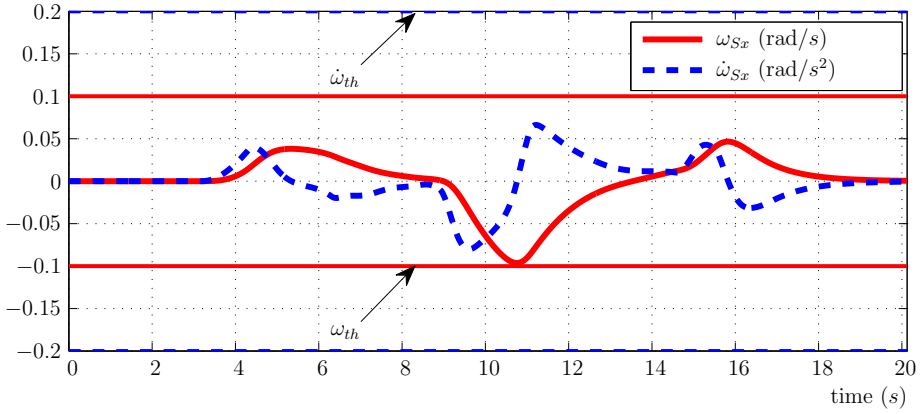


Figure C.17: CLRN - **AT2** - Simulated rotational quantities

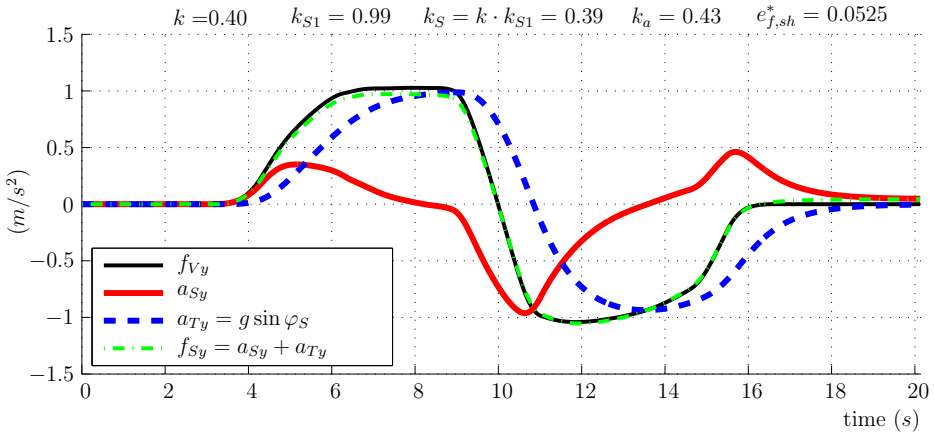


Figure C.18: CLRN - **AT2** - Simulated specific forces and acceleration

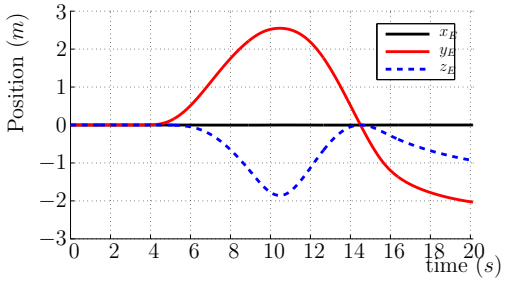


Figure C.19: CLRN - **AT2**

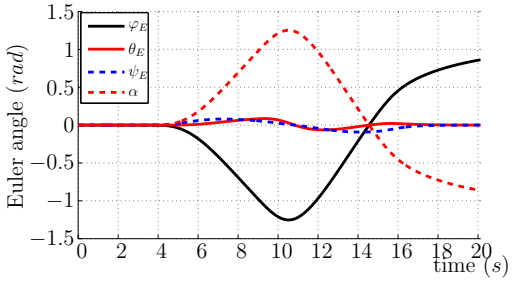


Figure C.20: CLRN - **AT2**

Response of CLG	Auto-tuning	**AT1**

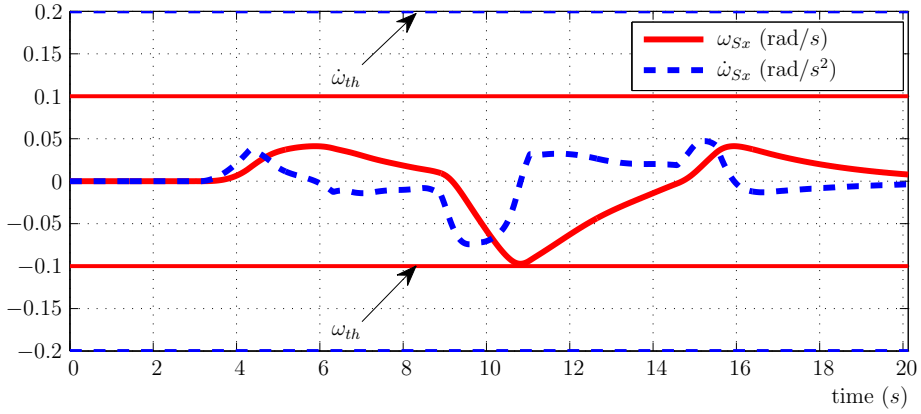


Figure C.21: CLG - **AT1** - Simulated rotational quantities

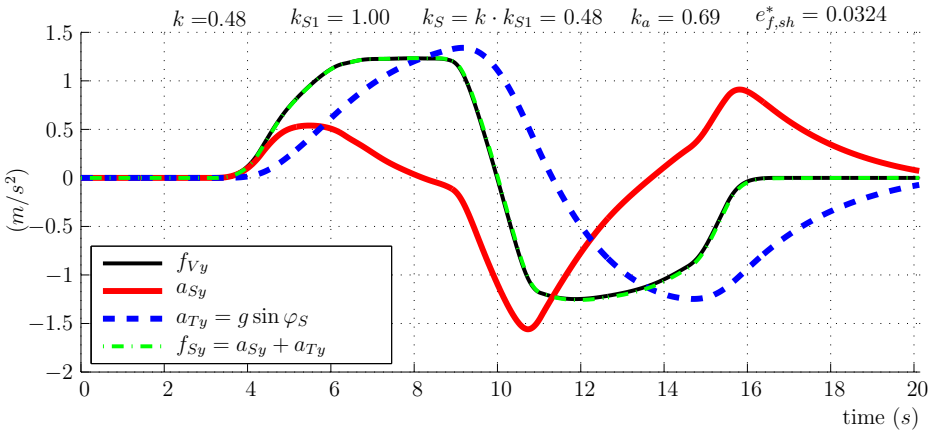


Figure C.22: CLG - **AT1** - Simulated specific forces and acceleration

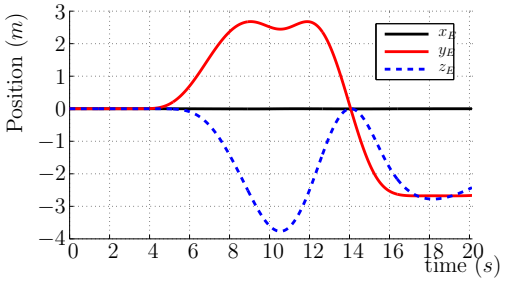


Figure C.23: CLG - **AT1**

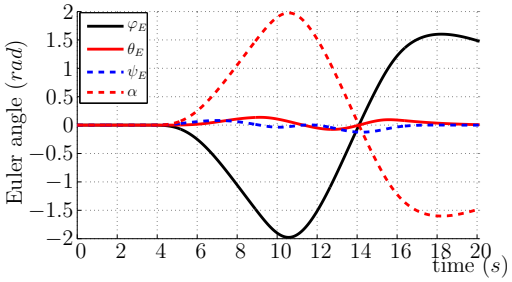


Figure C.24: CLG - **AT1**

Response of CLG	Auto-tuning	**AT2**

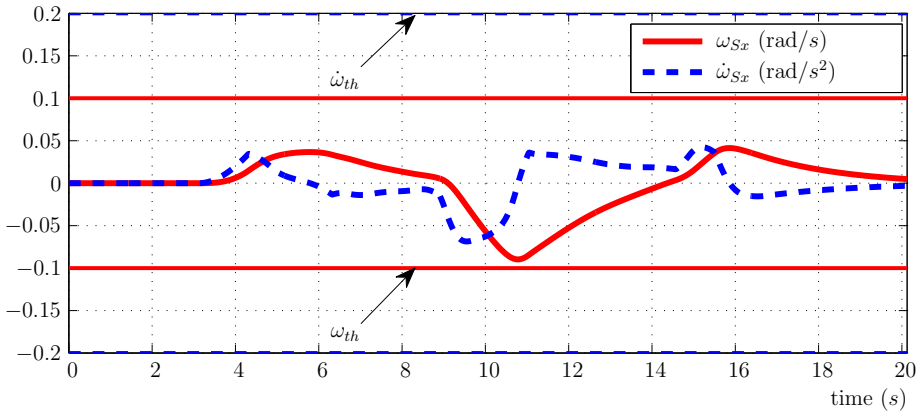


Figure C.25: CLG - **AT2** - Simulated rotational quantities

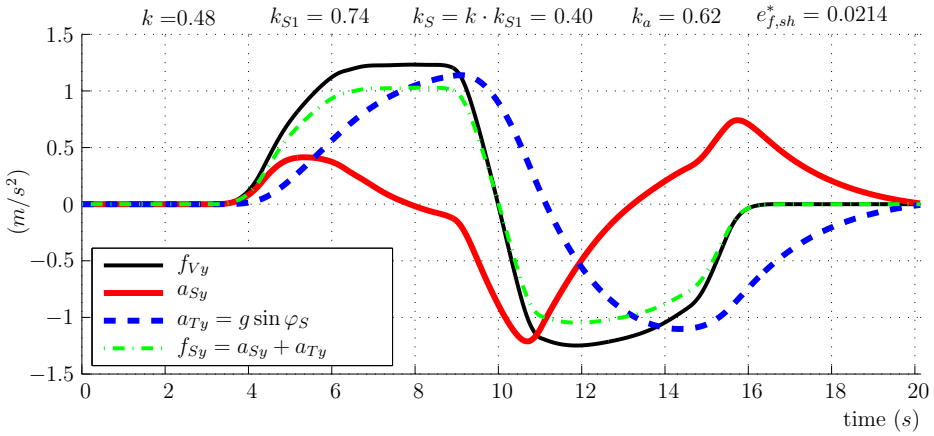


Figure C.26: CLG - **AT2** - Simulated specific forces and acceleration

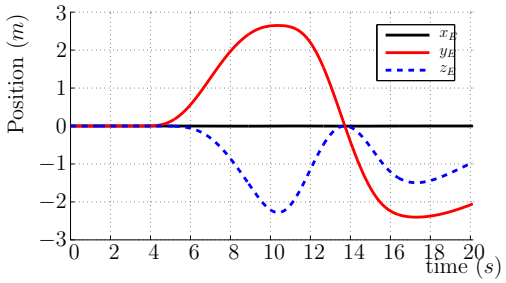


Figure C.27: CLG - **AT2**

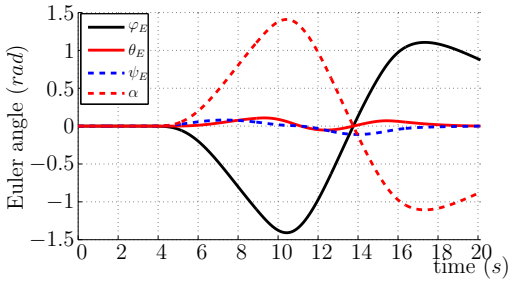


Figure C.28: CLG - **AT2**

D Model of self-motion perception

This part introduces the mathematical models of the self-motion perceptive systems that were mentioned in the section 2. The models were constructed from the empirical results in the literature. They are visual, vestibular, proprioceptive system models which were commonly applied in particular studies related to the MCAs. The part provides the most commonly used models mentioned in driving simulation field. The explanation of these model were in the section 2. Linear transfer functions of these otolith and semicircular organs are shown in Tab. D.1 and D.2, such as:

$$H_{Oto,1} = \frac{\widehat{f}}{f} = \frac{K_{Oto}}{(T_1^O s + 1) \cdot (T_2^O s + 1)}\,, \tag{D.1}$$

$$H_{Oto,2} = \frac{\widehat{f}}{f} = \frac{K_{Oto} \cdot (T_L^O s + 1)}{(T_1^O s + 1) \cdot (T_2^O s + 1)}\,, \tag{D.2}$$

$$H_{Oto,3} = \frac{\widehat{f}}{f} = \frac{K_{Oto} \cdot (T_L^O s + 1)}{(T_1^O s + 1)}\,, \tag{D.3}$$

$$H_{Scc,1} = \frac{\widehat{\omega}}{\omega} = \frac{K_{Scc} \cdot \tau_1}{(\tau_1 s + 1) \cdot (\tau_2 s + 1)}\,, \tag{D.4}$$

$$H_{Scc,2} = \frac{\widehat{\omega}}{\omega} = \frac{K_{Scc} \cdot \tau_1 \tau_a s}{(\tau_1 s + 1) \cdot (\tau_2 s + 1) \cdot (\tau_a s + 1)}\,, \tag{D.5}$$

$$H_{Scc,3} = \frac{\widehat{\omega}}{\omega} = \frac{K_{Scc} \cdot \tau_1 \tau_a s \cdot (\tau_L s + 1)}{(\tau_1 s + 1) \cdot (\tau_2 s + 1) \cdot (\tau_a s + 1)} \tag{D.6}$$

$$H_{Scc,4} = \frac{\widehat{\omega}}{\omega} = \frac{K_{Scc} \cdot \tau_1}{(\tau_L s + 1) \cdot (\tau_1 s + 1)}\,. \tag{D.7}$$

where, K_{Oto} is gain factor of the model of the otolith organ; T_1^O, T_2^O, T_L^O are time coefficients of the model of otolith models; K_{Scc} is gain factor of the model of the semicircular organ; k_φ, k_θ, k_ψ are gain factor for different rotational DoFs; $\tau_1, \tau_2, \tau_a, \tau_L$ are time coefficients of the model of semicircular system.

The threshold values of the vestibular system is the important factor for implementing tilt coordination technique in a MCA. These values were differently obtained from empirical experiments with different situation. Tab. D.3 lists several threshold values in the literature.

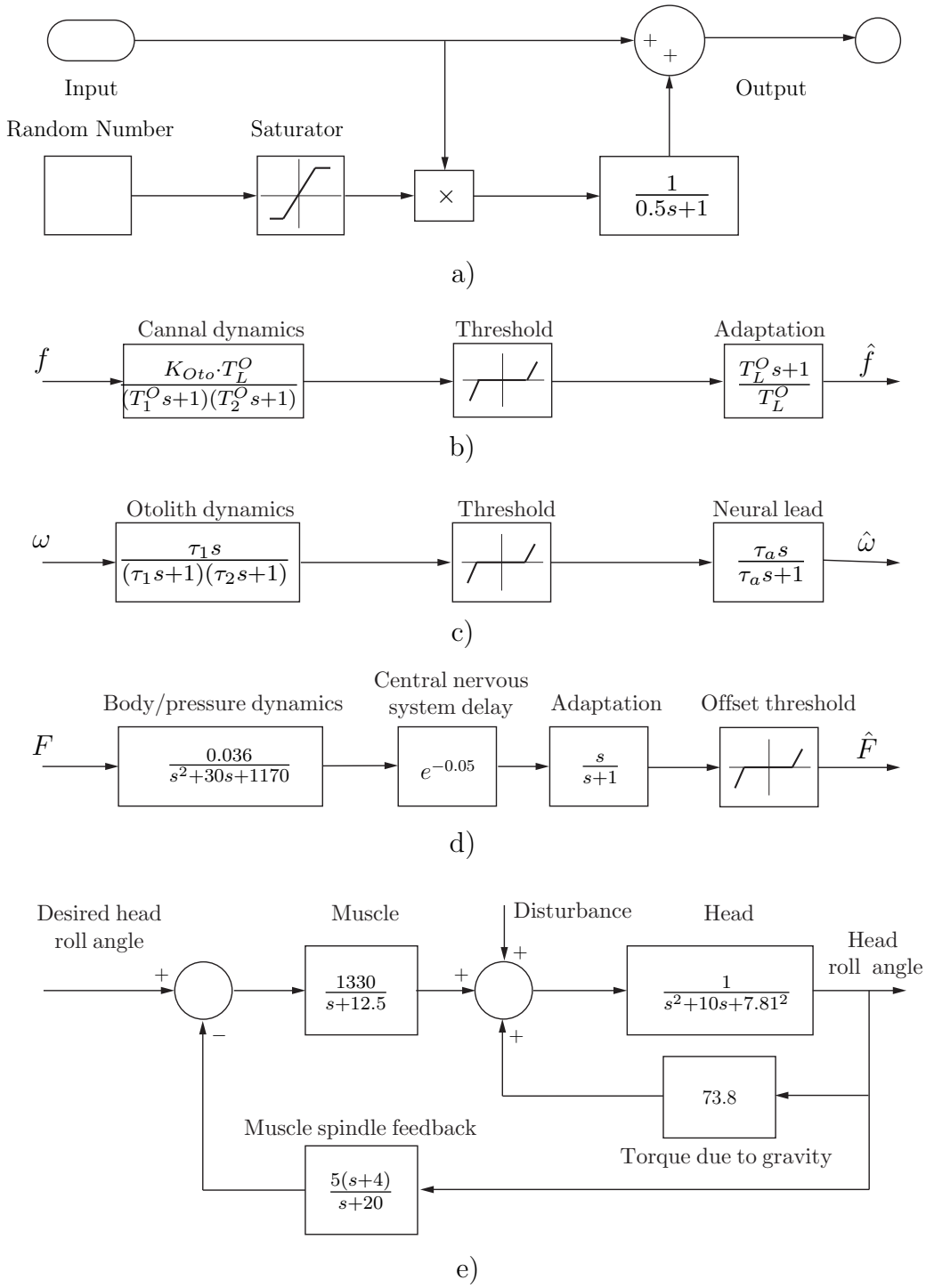


Figure D.1: The mathematical models of motion perception a) Visual cues model (Hess (2008)); b) Otolith dynamic model (Reid and Nahon (1985)); c) Semicircular model (Reid and Nahon (1985)); d) Pressuru model (Gum (1973)); e) Lateral head motion control model Gum (1973) visual cues introduced by Hess (2008)

Source	Stein-hausen (1931)	Mayne (1974)	Borah et al. (1988)	Hosman and Van der Vaart (1978)	Reid and Nahon (1985)	Wentink et al. (2005)
Model	$H_{Oto,1}$	$H_{Oto,1}$	$H_{Oto,2}$	$H_{Oto,3}$	$H_{Oto,1}$	$H_{Oto,3}$
$K_{Oto}\,(-)$	k_F	0.4	45	1	0.4	1
$T_1^O\,(s)$	10	5.26	5	0.5	5.33	0.12
$T_2^O\,(s)$	0.66	0.66	-	0.016	0.66	-
$T_L^O\,(s)$	-	13.16	10.0	1.0	13.2	0.3

Table D.1: Parameters of models of otolith organ (based on Fischer (2009))

Source	Stein-hausen (1931)	Mayne (1974)	Borah et al. (1988)	Hosman and Van der Vaart (1978)	Reid and Nahon (1985)	Wentink et al. (2005)
Model	$H_{Scc,1}$	$H_{Scc,1}$	$H_{Scc,2}$	$H_{Scc,4}$	$H_{Scc,2}$	$H_{Scc,4}$
$K_{Scc}\,(-)$	1	1	1	1	1	1
$\tau_1\,(s)$	5	$8 \cdots 15$	10	5.9	$\tau_{1,\varphi} = 6.1$ $\tau_{1,\theta} = 5.3$ $\tau_{1,\psi} = 10.2$	5.9
$\tau_2\,(s)$	0.004	$0.01 \cdots 0.08$	0.005	-	0.1	-
$\tau_a\,(s)$	-	-	30	(80)	30	-
$\tau_L\,(s)$	-	-	-	0.1	-	0.11

Table D.2: Parameters of models of the semicircular organ (based on Fischer (2009))

	In darkness			With vision			
Quant-ities	Hosman and Van der Vaart (1978)	Benson et al. (1986)	Mes-land et al. (1998)	Reid and Nahon (1985)	Nahon and Reid (1990)	Rey-mond et al. (1999)	Rey-mond and Ke-meny (2000)
$\ddot{x}\,(m/s^2)$	$0.04 \cdots 0.085$	0.063	–	0.17	–	–	0.05
$\ddot{y}\,(m/s^2)$		0.057	–	0.17			
$\ddot{z}\,(m/s^2)$		0.154	–	0.28			
$\dot{\varphi}\,(rad/s)$	–	2.04	–	3.0	3.0	2.04	–
$\dot{\theta}\,(rad/s)$		2.07	0.5	3.6			
$\dot{\psi}\,(rad/s)$		1.2	–	2.6			
$\ddot{\varphi}$, $\ddot{\theta}$, $\ddot{\psi}$ (rad/s^2)	$0.03 \cdots 0.065$	–	–	–	–	0.3	–

Table D.3: Threshold values with and without vision (based on Fischer (2009))