Interner Bericht

DLR-IB-FT-BS-2025-86

Development of a G-Seat Force Cueing Algorithm for Motion Simulation

Hochschulschrift

Davide Nazzaro

Deutsches Zentrum für Luft- und Raumfahrt

Institut für Flugsystemtechnik
Braunschweig

Deutsches Zentrum
für Luft- und Raumfahrt

Institutsbericht
DLR-IB-FT-OP-2025-86

Development of a G-Seat Force Cueing Algorithm for Motion Simulation

Davide Nazzaro

Institut für Flugsystemtechnik
Braunschweig

Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Flugsystemtechnik
Abteilung Flugregelung und Simulation

Stufe der Zugänglichkeit: I, Allgemein zugänglich: Der Interne Bericht wird elektronisch ohne Einschränkungen in ELIB abgelegt.

Braunschweig, den 14.07.2025

Institutsleitung:	Prof. Dr.-Ing. S. Levedag
Abteilungsleitung:	Dr.-Ing. Tobias Bellmann
Betreuer:in:	M.Sc. Tobias Treichl
Verfasser:in:	Davide Nazzaro

[redacted]

POLITECNICO DI TORINO

Master's Degree Course in Mechatronic Engineering

Master's Degree Thesis

Development of a G-Seat Force Cueing Algorithm for Motion Simulation

Supervisor
Dr. Martina Mammarella
Dr. Fabrizio Dabbene
Ing. Tobias Treichl

Candidate
Davide Nazzaro

april 2025

Development of a G-Seat Force Cueing Algorithm for Motion Simulation

Davide Nazzaro

Abstract

Motion simulators aim to replicate the sensation of movement to enhance realism in flight training and research applications. To achieve this goal, motion cues, including but not limited to vestibular, pressure, and visual, are typically employed. By combining the motion of robotic platforms such as hexapods and robotic arms, combined with visual and sensory feedback, motion simulators create highly immersive experiences that mimic real-world vehicle dynamics.

G-seats, equipped with pressure-based actuators to simulate acceleration and maneuvering forces, play a crucial role in these type of systems, especially when large-scale motion platforms are unavailable or have limited mobility.

Due to workspace constraints, motion simulators are unable to reproduce low-frequency sustained accelerations. To overcome this limitation, prototype flap actuators integrated into the seat have been designed to provide localized pressure cues, aiming to enhance the fidelity of perceived pressure.

This thesis focuses on the design of an optimal washout algorithm for motion cueing (also known as force cueing), tailored for the actuators to be integrated into the Robotic Motion Simulator (RMS) system at the German Aerospace Center (DLR). Specifically, a somatosensory pressure model is integrated into an LRQ-based control strategy to generate the optimal signal for the actuators, aiming to accurately replicate the pressure a pilot would experience in a real vehicle.

The effectiveness of the algorithm was evaluated through software simulations and compared with a classical washout algorithm highlighting the improvement that can be obtained.

Contents

List of Figures

Introduction

Motion simulation plays a crucial role in the research and development of technologies in the aerospace and automotive fields, as well as in other transportation domains. The human brain processes motion through several sensory channels: visual stimuli, the vestibular system, and the somatosensory system. Simulations based solely on visual stimuli are often insufficient; in fact, the mismatch between motion cues received from these channels can cause discomfort called "motion sickness" [6].

Moving platforms, such as parallel robots or capsules controlled by a robotic arm, are the most widely used methods to simulate real aircraft movements by stimulating human perception through the vestibular system.

Simulators aim to minimize the difference between the generated motion cues and the dynamics of the actual aircraft. Motion systems struggle to replicate low-frequency accelerations because of hardware constraints. One way to overcome this problem is to build larger motion systems to extend the operational bandwidth to lower frequencies, but this leads to a cost increase. Therefore, it is necessary to replicate the desired accelerations in a way that the simulator remains within its operational limits, providing the pilot with accelerations as realistic as possible. A possible approach is to design motion cues based on somatosensory perception to be integrated into the sensed cues perceived by the vestibular system. G-seat is a technology that takes advantage of this approach by simulating accelerations through somatic cues such as skin pressure. This can be achieved by using actuators like active seat belts (ASB), movable flaps, or air-inflatable padding placed in the seat ([7],[27]). In this context, the process of defining motion cues that provide the desired perceived pressures is called force cueing.

This thesis builds upon previous research work [15], which commissioned a prototype system and developed an initial control algorithm for the system. The objective of this work is to enhance that algorithm by integrating a tactile perception model, further improving the realism of force cueing.

In this thesis work, realized in collaboration with CNR-IEIIT and DLR, an optimal control algorithm has been developed for actuators to be embedded in a G-seat for the DLR Robotic Motion Simulator (RMS), to simulate the movements associated with low-frequency accelerations. The actuators used are movable flaps, integrated under the seating surface and in the backrest.

As previously mentioned, motion simulation is utilized in multiple transportation sectors; however, this research is especially pertinent to the aviation industry. Therefore, any reference to a vehicle in this context will specifically refer to an aircraft.

The rest of the thesis is structured as follows.

Chapter 1 provides an overview of the architecture of the DLR RMS, highlighting the main features and functionalities. It includes a summary of the most used force cueing algorithms and a description of the anatomical mechanism of tactile perception.

Chapter 2 describes the simulation environment and the tools used to model the system. It also delves into the modeling of the somatosensory perception system and describes the implementation procedure of the force cueing algorithm.

Chapter 3 contains the description of the methodology used to test the performance of the developed model and covers the tuning process used to optimize the response of the actuators. Additionally, this chapter presents the results obtained.

Finally, Chapter 4 summarizes the key findings of the project, discussing the implications of the results and potential improvements and suggesting possible directions for further research and optimization in G-seat technology for motion simulation.

Chapter 1

State of the art

Motion simulation plays an important role in the aviation industry. Flight testing is expensive and complex and is often conducted in the final stages of development when most of the budget is already allocated; therefore, exploiting simulated models is paramount for aircraft manufacturers. To ensure the reliability of the simulation environment, it is crucial that it accurately represents the interactions between the aircraft systems and provides the pilot with feedback that closely resembles real-life scenarios [26]. This allows pilots to train in a controlled environment while providing them the ability to perform high-risk maneuvers that would be difficult or dangerous to attempt in real-flight conditions.

This chapter describes the hardware platform employed for this work, a definition of motion cueing and the most commonly used algorithm techniques to implement it. Lastly: an overview of the perceptual models useful for this thesis is provided.

1.1 Motion platform

1.1.1 The Robotic Motion Simulator (RMS)

The Robotic Motion Simulator [19] is a 7 degree of freedom (DOFs) motion system developed by DLR, and it's the simulation framework on which this study is based. The robotic arm, i.e. KUKA KR 500 TÜV, features six DOFs, while an additional DOF is created by the $10m$ linear track on which it's mounted. The simulator capsule (gondola) is attached to the robotic arm, enabling infinite yaw and roll motions, whereas the pitch motion is restricted between $-216°$ and $93°$. Figure 1.1 and Figure 1.2 show an overview of the RMS joint DOFs and an image of the simulator in action. Figure 1.3 shows the simulator cockpit, which is modular and can host different systems to increase the simulation fidelity. Translational movement is constrained by a $3m$ limit, which can be extended using the linear axis, $2m$ on the longitude axis, and $1m$ vertically. Manufacturer-imposed constraints on torque, acceleration, and velocity limit the simulator's drives. The RMS presents several unique benefits compared to traditional simulators that utilize a Steward platform. Notably, it boasts an extensive rotational and translational workspace, positioning the RMS as one of the most agile serial kinematic simulators on the market. Additionally, it provides significant flexibility in accommodating various modular cockpit components and input devices. However, it is important to note that the RMS has lower payload capacity and acceleration capabilities in comparison to its Steward-platform based alternatives, with a maximum payload limit of $500kg$.

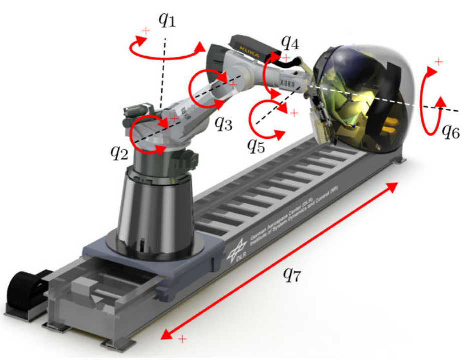


Figure 1.1. Joint Locations and Turning Directions of the Robotic Platform [5]

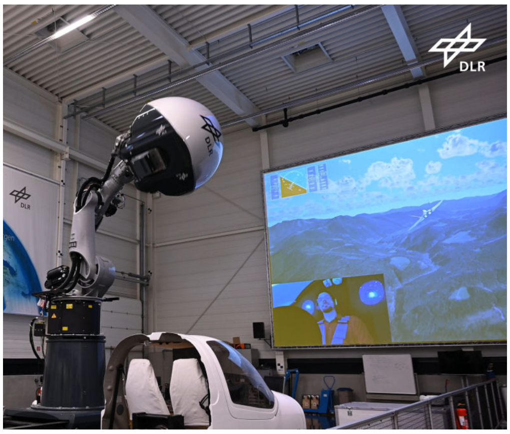


Figure 1.2. Robotic Motion Simulator (RMS) at the German Aerospace Center [19]

Inside the RMS, the pilot receives flight visual information through a display or virtual reality headset. Interaction between the pilot and the simulator is enabled by control devices such as a joystick, thrust lever, and pedals, which directly influence the flight dynamics model. The resulting accelerations, rotational velocities, and orientations are processed through a washout filter, which optimizes workspace utilization while maintaining realistic motion perception. An optimization approach based on inverse kinematics allows the position and orientation of the robotic arm to match the desired ones, ensuring that the joints' angles, velocities, and accelerations meet the required values. By incorporating sophisticated motion control technology, the simulator can deliver an exceptionally realistic and interactive setting for both pilot training and aircraft evaluation. This approach offers a viable substitute for conventional flight testing while maintaining the complexity and fidelity for such advanced stages.

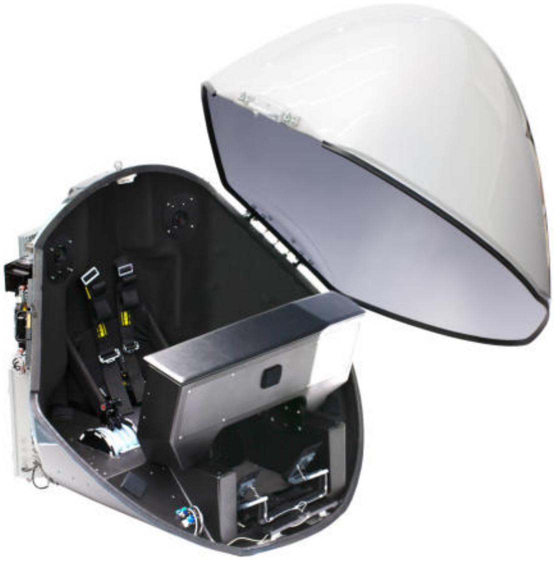


Figure 1.3. Gondola, RMS cockpit simulator [19]

1.1.2 G-seat

The G-Seat technology represents a groundbreaking advancement in the simulation of motion and acceleration for seat-based simulators, including those used in airplanes, helicopters, cars, boats, and spacecraft. Its mechanism employs a sophisticated combination of seat movement, pressure-applying components within the seat, and a tensioning harness to deliver immediate, rapidly reversible, and enduring sensations of motion and acceleration across all three spatial dimensions.

Research [2] indicates that G-seats serve as a valuable enhancement for motion simulators and, when paired with a moving platform, can significantly enhance the sensation of motion. In the investigation conducted by Chung et al. [11], various configurations of a simulator incorporating a G-seat were evaluated. Pilots operated Blackhawk military transport helicopters within a motion simulator that utilized both a G-seat independently and a G-seat in conjunction with platform movement. The pilots noted that the combination of the G-seat and platform movement provided a more pronounced sense of motion than the platform alone.

For this reason, G-seat technology was evaluated as a viable option to improve RMS performance. Metal flaps were selected to serve as actuators to be integrated within the seat cushioning.

Each flap is controlled by an electric motor, specifically a Nanotec Electronic DF45, which receives the target position in degrees. The driver (Somanet Node)

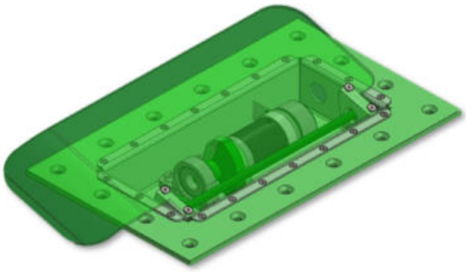


Figure 1.4. Actuator CAD model

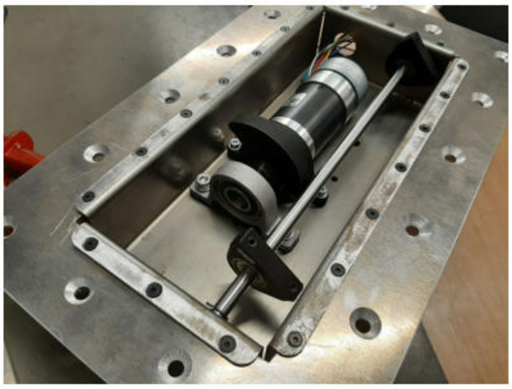


Figure 1.5. Actuator prototype [Credit to: DLR Institute of Dynamics and Control]

gets the actual position by an incremental encoder, the Same Sky AMT10, as an input and controls the motor to reach the desired position. Then, a cam disc moves the motor through a planetary gear, ensuring a linear relationship between the rotation of the motor and the angle of the disc. As the disc rotates, it pushes against the metal plate from below, causing it to deflect. The disk is mounted inside the actuator housing, allowing it to rotate around its axis. The amplitude of the flap deflection depends on the size of the disk. In this case, the selected disk allows a maximum fin deflection of about 16.5°, corresponding to a disk rotation of 136.5°. Since the relationship between disk motion and fin deflection is nonlinear, it can be mapped by interpolating the measured data points.

The actuators are engineered for installation beneath the seat, allowing them to apply pressure to the buttocks against the seat back. The actuator is shown in Figure 1.4 and Figure 1.5, while their configuration is depicted in Figure 1.6.

A Hall sensor tracks the position of the motor; in this way, position feedback is guaranteed. The initial position must be defined at the beginning of the operation, and this is achieved by a reference run each time the motor is switched on, during which the motor rotates in one direction until it detects a stop. The stop is identified when the motor current exceeds a predefined threshold, at which the zero point position is set. The Hall sensor has a resolution of only 7.5°. However, since the motor completes over 10 full rotations (3675°) to move the seat flap from 0° to its maximum deflection of 16.5°, the resulting 7.5° position deviation corresponds to an average flap deflection error of just 0.03°. This margin is negligible and does not impact operations.

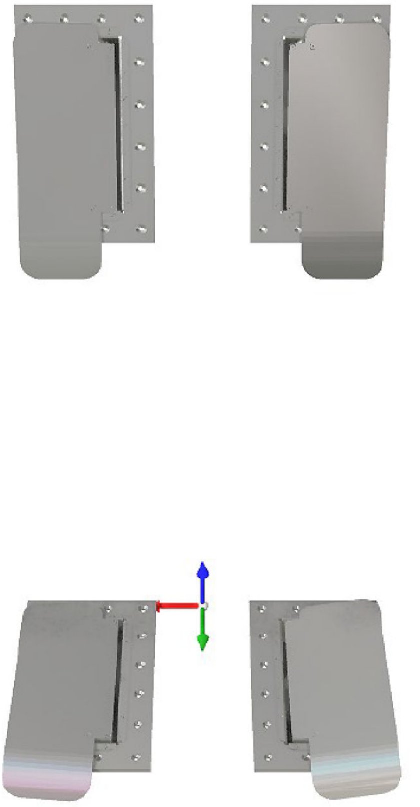


Figure 1.6. Configuration of flaps, modeled using the DLR SimVis software. [15]

1.2 Motion cueing alogrithms

In the field of flight simulation, motion cueing refers to the creation of stimuli that must be conveyed to the pilot. To achieve this, several components are essential: a physics model, a motion platform, and a Motion Cueing Algorithm (MCA) [9]. The physical model interprets the operator's control inputs, assesses the dynamics of the aircraft, and computes the simulated physical state by solving the differential equations governing motion. Then, the simulator replicates this physical state at the operator's location. This process is facilitated by MCAs, which convert the simulated physical state into corresponding motion for the simulator whenever feasible. Motion cueing involves the creation of sensory stimuli designed to replicate realistic movements within a virtual environment. The vestibular system is generally given precedence in the transmission of information due to its heightened sensitivity to alterations in movement. Conversely, the other components of the simulator can be engineered to produce stimuli across multiple perceptual channels, like visual, auditory, and somatosensory ones.

The primary objective of MCAs is to mimic the motion of an actual aircraft while adhering to the limitations imposed by the simulator. The area in which it can move is referred to as the workspace. In addition, simulated vehicles often generate very high accelerations, which cannot be replicated by the simulator. Therefore, the goal is to mislead the operator's senses, emphasizing the importance of perceived validity to ensure that the experience of controlling the vehicle feels authentic rather than merely physically accurate. Various techniques are implemented in MCA to trick the operator's sensory perception.

The MCA receives specific physical quantities (computed by the physics model) from the simulated aircraft as input and the desired motion platform pose as output. The pose is represented as a six-components vector that defines the position and the orientation of the motion platform. The inputs may vary depending on which MCA is used, although they typically include linear acceleration (expressed as specific force) and angular velocity, as these are sensed by the human vestibular system. Other parameters, such as the vehicle position, orientation, or angular acceleration may also be used.

A detailed summary of motion cueing algorithms is provided by Fischer [16] and Born [5]. The following sections recall the most commonly used MCA approaches.

1.2.1 Classical Washout

The first MCA was originally designed by NASA engineers Schmidt and Conrad in 1969 [13], and then modified and improved by other groups of researchers. However, the basic ideas remained the same [17]. This algorithm has come to be known as the classical algorithm or classical washout and is still widely used. There are some variations from different implementations, but the general principles are similar. The key features of the classical washout are:

- Scaling down the inputs relative to the physical state of the simulator in order to minimize the generated motion and reduce the risk of exceeding the platform's workspace;

- Removing low-frequency acceleration that could lead to large displacement and cause the reaching of the hardware constraints;

- Using tilt to simulate low-frequency accelerations that were filtered out since the vestibular system is not able to distinguish between translational acceleration and a tilted head position for sufficiently small angles;

- Ensuring the algorithm returns the motion platform to its neutral position for constant input, which is why the algorithm is called "washout".

Figure 1.7 shows a classical washout filter algorithm structure.

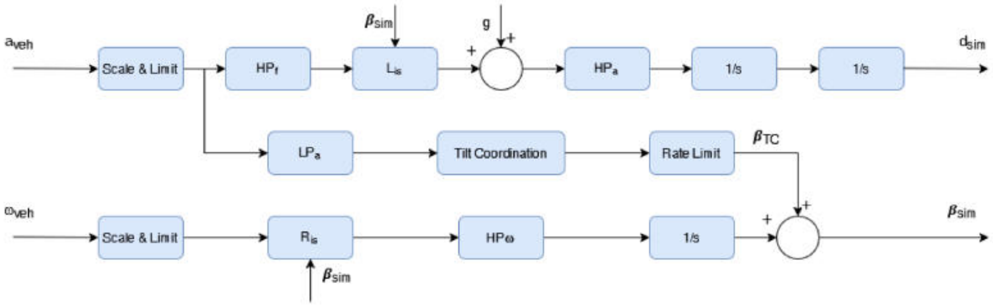


Figure 1.7. Classical Washout structure

The classical algorithm is the predominant choice in commercial simulators. It is defined by a combination of linear high-pass and low-pass filters, the parameters of which, including break frequencies and damping ratios, can be fine-tuned through an off-line trial and error process.

The initial implementation requires as inputs the three specific forces acting on the aircraft cockpit in the body axes, which are defined as

$$f_{veh} = a - g, \tag{1.1}$$

where g is the gravitational acceleration and a is body-axis components of the aircraft acceleration at the cockpit reference point, and the three angular rates ω_{veh} of the aircraft.

The translational branch generates translational motion while operating within the motion platform's workspace. This channel reproduces high-frequency displacements only. In this channel, the specific force enters as input and then is scaled and transformed into world frame acceleration [22]. Through a High-Pass Filter, the low-frequency acceleration is attenuated to reproduce only high-frequency displacement. Finally, a double integrator provides the sway, surge, and heave displacements of the motion platform.

The translational channel HP filter transfer function is

$$HP_f = \frac{s^2}{s^2 + 2\zeta_h\omega_h s + \omega_h^2} \tag{1.2}$$

Where ζ_n is the damping ratio, which determines the sharpness of the filter's response, and ω_h is the cut-off frequency.

The rotational channel works like the translational one, but it handles the angular velocities instead of specific forces. It contains blocks that have the same function of the translational branch; however, the high-pass filter output is integrated once to obtain the Euler angles (pitch, roll, and yaw). Yaw is a direct output, while pitch and roll are further combined with the coordination channel output.

The rotational channel HP filter transfer function is

$$HP_\omega = \frac{s}{s + \omega_{h,\theta}} \tag{1.3}$$

Where $\omega_{h,\theta}$ is the cut-off frequency.

The coordination channel is responsible for simulating low-frequency accelerations, previously attenuated by the translational channel. It induces rotational movements that tilt the motion platform, exploiting gravity to simulate low-frequency accelerations. The scaled-specific forces enter this branch as input. A low-pass filter is then applied to eliminate high-frequency signals. The resulting acceleration is converted into pitch and roll angles (tilt coordination is limited to lateral and longitudinal accelerations) by the tilt coordination block. In order to ensure these motions are not perceived as angular displacements, a rate limiter is applied, trying to keep the angular velocity of the tilt coordination below the operator's perceptual threshold.

The coordination channel LP filter transfer function is

$$LP_a = \frac{\omega_t^2}{s^2 + 2\zeta_t \omega_t s + \omega_t^2} \tag{1.4}$$

Where ζ_t is the damping ratio, which determines the sharpness of the filter's response, and ω_t is the cut-off frequency.

Most of the processing blocks of the classical algorithm can be parameterized, and the choice of the parameters significantly modifies its performance.

1.2.2 Adaptive Washout

The classical washout is effective and simple to implement. However, its performance is limited by two factors:

- the reachable workspace of the motion platform is not considered, which enhances flexibility but results in suboptimal adaptation to the particular hardware;
- the algorithm parameters are pre-tuned offline for a worst-case scenario, as the generated motion must stay within the platform constraints regardless of the inputs.

This conservative approach often leads to an underutilization of the motion platform, particularly when input magnitudes are small, as accelerations may be excessively downscaled. Consequently, only a fraction of the platform's workspace is utilized under typical conditions, and the resulting motion cues may be less effective.

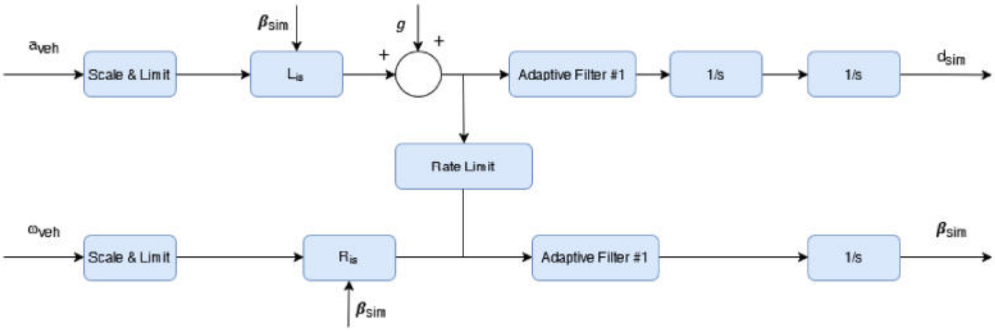


Figure 1.8. Adaptive Washout structure

Adaptive algorithms have been introduced ([21],[23])to address these limitations, allowing the MCA filters to adjust their parameters dynamically throughout execution. Classical washout algorithms, in contrast, are characterized by their use of simple linear filters and scaling elements with time-invariant coefficients combined with nonlinear limiting and transformation processes such as tilt coordination. Adaptive washout algorithms, on the other hand, incorporate time-varying filters and scaling coefficients, enabling a more efficient and responsive use of the motion platform. Figure 1.8 shows the block diagram that outlines the operation of the adaptive washout.

1.2.3 Optimal Washout

The main limitation of both classical and adaptive algorithm is that the human perception is not considered, as both algorithms are essentially heuristic. The filter structure is predefined and seeks to optimize the parameter values of the algorithm blocks. Thus, this fixed structure does not represent the optimal approach to motion cueing, even when parameters are finely tuned.

For this reason, the optimal MCA was introduced by [25] and developed by [26]so that the motion cueing generation system is treated as a control problem, trying to minimize the difference between the output of the system and the expected output. Figure 1.9 shows the block diagram of the optimal washout algorithm. The main difference from the other presented approaches can be found in the filters W_{11}, $W12$, and W_{22}. These filters can be obtained using the Linear Quadratic Regulator (LQR) method and obtaining filters of high order. The goal of this optimization problem is to minimize perception errors by incorporating mathematical models of human perceptual systems.

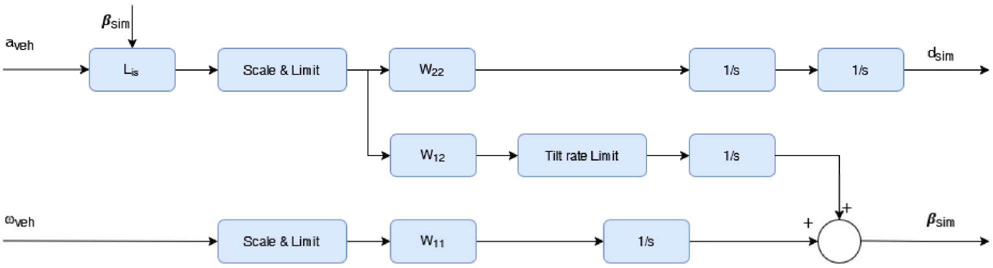


Figure 1.9. Optimal Washout structure

The potential advantage that comes from a reduced sensation error depends on the accuracy of the human perception models. Additionally, the working space usage may also be improved.

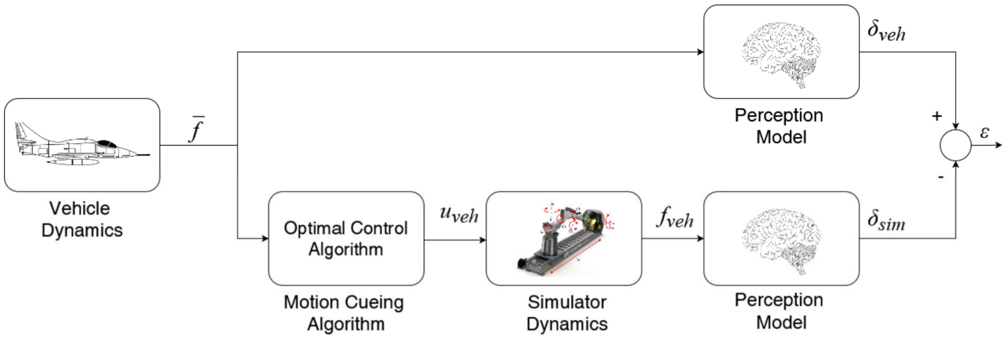


Figure 1.10. Optimization structure of the optimal control algorithm

Figure 1.10 illustrates the optimization structure of the optimal control algorithm. The input is the force F that represents the vehicle's dynamics. In the upper branch, the force perceived by the pilot in an aircraft during a real scenario is calculated using the appropriate perception model. In the lower branch, the force is first processed through the optimal motion cueing algorithm, which provides the simulator with linear and angular position vectors. The force f_{veh} generated by the simulator is then sent to the perception block, which returns the force δ_{sim} the pilot perceives as reproduced by the simulator.

The two perceived forces (the real one and the one in the simulator) are subsequently compared. The algorithm's objective is to minimize the discrepancy between these two forces, thereby enhancing the simulation experience.

In the linear Optimal MCA implementation, the filters are computed a priori, by selecting the weighting matrices in the cost function. In the Nonlinear Optimal MCA variant, the optimization problem is solved at each time step of the simulation. This approach is challenging because of the real-time implementation, due to the Riccati equation solution to be computed at each step. The optimal approach stands out by incorporating the sensation error and the simulator's physical constraints directly into the filter design. However, due to the significant computational effort required for optimization, this method demands more time for tuning and development compared to other approaches

1.2.4 MPC Washout

A method for developing motion cueing algorithms that has gained significant prominence in recent years is Model Predictive Control (MPC) developed by Dagdelen et al. [14]. This technique is capable of managing a large number of variables and complex constraints. The primary objective of MPC is to enhance a cost function through the prediction of system behavior.

The objective is to determine a viable control action and a sequence of states at each time step across a prediction horizon, facilitating the computation of the optimal movement for the simulator. The optimal movement is attained by minimizing the squared perception error, which is defined as the discrepancy between the acceleration observed in the simulator and that of the real vehicle. This methodology incorporates the physical limitations of the simulator and also factors in the human perception system during the calculations. When the algorithm predicts that the platform is approaching its limits, it starts a washout maneuver, guiding the simulator toward its neutral position while remaining below the motion perception threshold. This inevitably introduces false cueing, as the system must decelerate to stay within its physical boundaries.

The higher the control horizon, the better the tracking performance, increasing significantly the complexity of the calculations. The MPC strategy has received positive subjective evaluations[9]; however, further studies are needed to compare its performance with existing MCA methods.

1.3 Force Cueing

In the context of G-seat application, the accelerations specified by the simulation are modeled by applying pressure to the body, this method is known as force cueing [15]. This approach operates in a similar way to motion cueing but relies on actuators exerting pressure on the human body. The position and the direction of these actuators are crucial to ensure the correct perception of acceleration.

Unlike motion-based systems, the G-seat does not use tilt coordination, in fact, it works purely through pressure application. To design cueing strategies based on human perception models, a model for pressure sensation must be included. In the present work, the optimal washout algorithm has been chosen and its detailed implementation will be presented in Chapter 2. In the following section, the main pressure perception models will be discussed.

1.4 Human perception model

The operator of a motion platform has to control its dynamics, which requires perceiving its motion. The human brain combines multisensory cues through vision, the vestibular system, somatosensory inputs, and auditory cues to make dynamic estimates of self-orientation [8], [12]. This section focuses on the somatosensory system, examining its physiology and categorization from an engineering perspective.

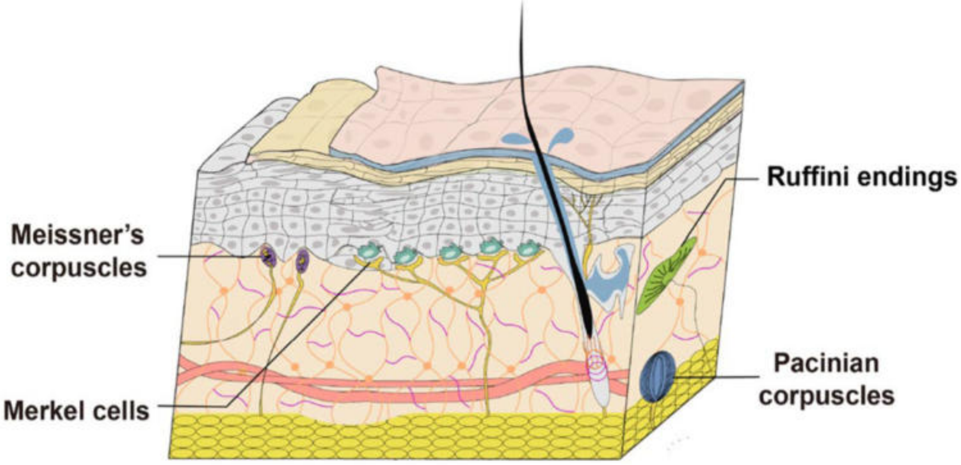


Figure 1.11. Mechanoreceptors of human skin [10]

Somatic sensors include mechanoreceptors, thermoreceptors, electromagnetic receptors, chemoreceptors, and pain receptors [3], [10]. For vehicle simulation, mechanoreceptors are the most relevant, as they detect touch, pressure, and vibration. Tactile sensors, located in the skin, are also called cutaneous sensors, which play a key role in perceiving steady-state motion. It is thought that these receptors convert physical energy into chemical alterations, leading to the specific stimulation of free nerve endings and the generation of nerve impulses.

Figure 1.11 shows the receptors of interest for this study and their placement.

1.4.1 Pacinian Corpuscles

Pacinian corpuscles are found in subcutaneous tissues of the body; they are a type of encapsulated nerve endings. They adapt to pressure within 10 milliseconds in both compression and release. They function as a high-pass filter and a rectifier. Figure 1.12 presents a schematic of an in-vitro experiment where a Pacinian corpuscle is positioned on a fixed plate and subjected to controlled compression via an applied electrical signal. The signal activates a piezoelectric crystal, causing it to deform and compress the corpuscle while removing the signal eliminates the compressive force. The bottom trace depicts crystal displacement, while the generated potential of the Pacinian corpuscle is shown just above it.

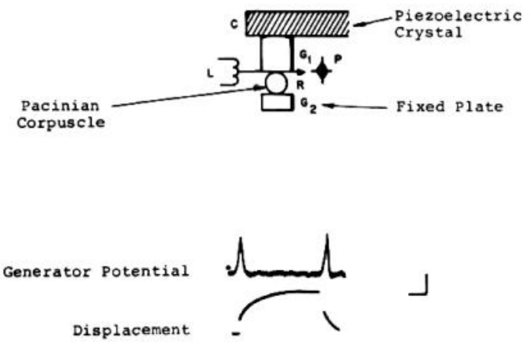


Figure 1.12. The generator potential induced by the compression of an intact Pacinian corpuscle. Calibration is $10ms$ and $50\mu V$. [4]

The Pacinian corpuscle generates a potential that consistently corresponds to the direction of the stimulus, regardless of whether it is being compressed or released, a phenomenon known as the "rectifier" effect. This suggests that it does not convey information about the direction of the applied force. Examination of the duration of the generated potential in relation to calibration marks indicates that the receptor fully adapts in approximately $10ms$, highlighting its sensitivity to high-frequency stimuli. These receptors are essential for sensing tissue vibrations and swift mechanical alterations, capable of responding to displacements as small as $10\mu m$. The transfer function of the Pacinian corpuscle, as described by Borah, characterizes its input in terms of skin deformation and its output as afferent potential, featuring a time constant of 10 milliseconds. The voltage output generated is affected by pressure, although the precise nature of this relationship necessitates additional research.

1.4.2 Type I receptors

A notable example of Type I receptors is the Merkel disc, located in the epidermis near the skin's surface. These receptors exhibit an initial strong response, followed by gradual adaptation over approximately 1 to $30s$, with a frequency sensitivity ranging from 1 to $100Hz$. They are highly attuned to detecting edges and curved surfaces, responding to skin displacements between less than $1mm$ and up to $5mm$. Merkel discs are distributed at a density of about one per square millimeter, though their static afferent firing rate is irregular.

Another Type I receptor is the Meissner corpuscle, which adapts rapidly and is found at a similar density of 1 per square millimeter. It has a broader bandwidth of 2 to $200Hz$ and detects skin displacements as small as $0.001mm$ up to $1mm$. However, its spatial resolution is lower compared to Merkel cells.

1.4.3 Type II receptors

Ruffini end organs are Type II receptors located in the dermis, as well as in connective tissues and joint capsules. Unlike Type I receptors, they are not positioned near the surface. They exhibit partial but slow adaptation, with time constants of approximately 1, 5, and $20s$, making them well-suited for detecting sustained skin deformation and skeletal joint rotation. These receptors are distributed at a density of about 10 per square centimeter and respond within a $10Hz$ bandwidth.

Ruffini endings have a continuous resting and static afferent firing rate (AFR) and are believed to play a key role in sensing skin stretch. Other Type II receptors include free nerve endings, which primarily detect pain, and hair follicle receptors, which respond to axial or bending forces with spatial resolution ranging from 1 to $3cm$.

1.4.4 Frequency response of somatosensory receptors

Figure 1.13 presents a psychophysical tuning curve illustrating the sensitivity of haptic receptors to vibrational stimuli. The graph plots the required stimulus amplitude for detection against frequency in hertz. Two data sets are shown: one under normal conditions and another with a topical anesthetic applied to the palm's base.

Each set features two distinct slopes, reflecting different receptor response characteristics. The shallower slopes correspond to Type I and Type II receptors, while the steeper slopes (falling at a rate of -2) are characteristic of Pacinian corpuscles. The anesthetized condition resulted in reduced sensitivity, meaning higher stimulus amplitudes were needed for receptors near the skin's surface to detect the vibration, while deeper receptors, primarily Pacinian corpuscles, remained unaffected.

Pacinian corpuscles exhibit increased sensitivity with frequency, requiring only a two-micron deformation for detection at $300Hz$. Further experiments extended the frequency range, showing that these receptors respond to deformations as small as $1\mu m$ peak-to-peak between 200 and $400Hz$, with slightly reduced sensitivity up to $500Hz$.

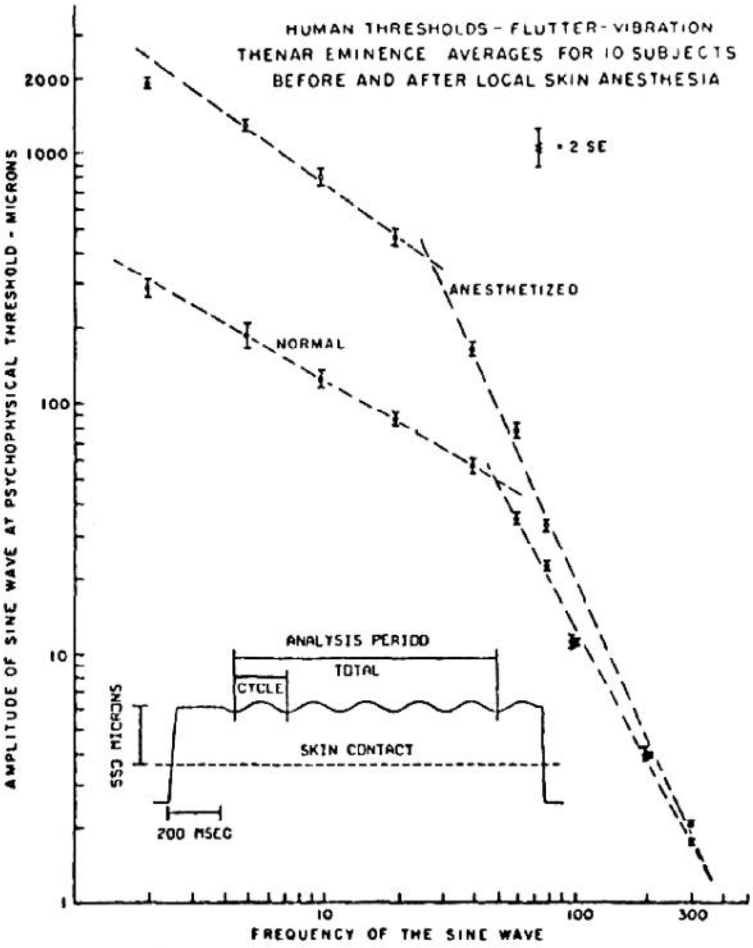


Figure 1.13. Psychophysical tuning curve for the human threshold in response to tactile stimuli on glabrous skin, specifically the thenar eminence. [4]

The somatosensory system plays a crucial role in motion perception, where mechanoreceptors provide essential feedback of touch, pressure and vibration. Among these receptors, Pacinian corpuscles demonstrate exceptional sensitivity to high-frequency stimuli, functioning as rectifiers and high-pass filters. Type I receptors, such as Merkel discs and Meissner corpuscles, contribute to edge detection and fine touch discrimination, while Type II receptors, including Ruffini endings, aid in perceiving sustained skin deformation and joint movement.

Chapter 2

System modelling

This chapter describes in detail the implementation procedure of the control strategy. In the first section, the implementation environment and software used are presented. Then, the optimal washout algorithm used to control the G-seat flaps and the haptic perception model used is described.

2.1 Simulation environment

The system modeling and controller design were performed using MATLAB/Simulink. The software was then tested on Modelica. In this paragraph, the development environments will be briefly described.

2.1.1 Modelica - an overview

Modelica is an open-source, object-oriented language designed for the modeling of extensive, intricate, and diverse physical systems [20]. It is particularly effective for multi-domain modeling, such as mechatronic systems in robotics, automotive, and aerospace sectors, which encompass mechanical, electrical, hydraulic, and control subsystems, as well as process-oriented applications and the generation and distribution of electrical power.

The design of Modelica allows it to be employed in a manner akin to how engineers construct real-world systems. Models within Modelica are represented mathematically through differential, algebraic, and discrete equations, eliminating the need for manual resolution of specific variables. A Modelica tool possesses sufficient information to automate this process. Furthermore, Modelica is structured to leverage specialized algorithms that facilitate the efficient management of large-scale models. It is well-suited for hardware-in-the-loop simulations and embedded control systems.

Modelica accommodates both high-level modeling through composition and detailed component modeling via equations. Standard component models are generally accessible in model libraries. A graphical model editor enables users to create models by constructing a composition diagram (or schematic) that involves positioning icons representing component models, establishing connections, and inputting parameter values through dialogue boxes. Additionally, features for incorporating graphical annotations in Modelica ensure that icons and composition diagrams can be easily transferred between various tools.

2.1.2 Dymola - an overview

Dymola, or Dynamic Modeling Laboratory, is a proprietary modeling and simulation software created by Dassault Systèmes, tailored to facilitate Modelica [1]. Specifically, it offers libraries of genuinely reusable components and supports connectors along with composite causal connections. Moreover, model libraries are accessible across various engineering fields.

Dymola employs an innovative modeling approach that integrates object orientation with equations, eliminating the traditional requirement for manual conversion of equations into block diagrams through the implementation of automatic formula manipulation.

Notable features of Dymola include:

- The capability to manage extensive and intricate multi-engineering models,
- Enhanced modeling efficiency through graphical model assembly,
- Accelerated simulation via symbolic pre-processing,
- Flexibility for user-defined model components,
- An open interface for integration with other software,
- Support for 3D animation,
- Real-time simulation capabilities.

The architecture of the Dymola software is depicted in figure 2.1. Dymola boasts a powerful graphical editor tailored for model composition. It works with Modelica models saved in files and can import a variety of data and graphic files. Furthermore, Dymola features a symbolic translator that transforms Modelica equations into C-code for simulation. This C-code can be exported to Simulink and hardware-in-the-loop systems. Additionally, Dymola provides sophisticated tools for experimentation, plotting, and animation. Users can employ scripts to organize experiments and perform calculations, and an automatic documentation generator is also included.

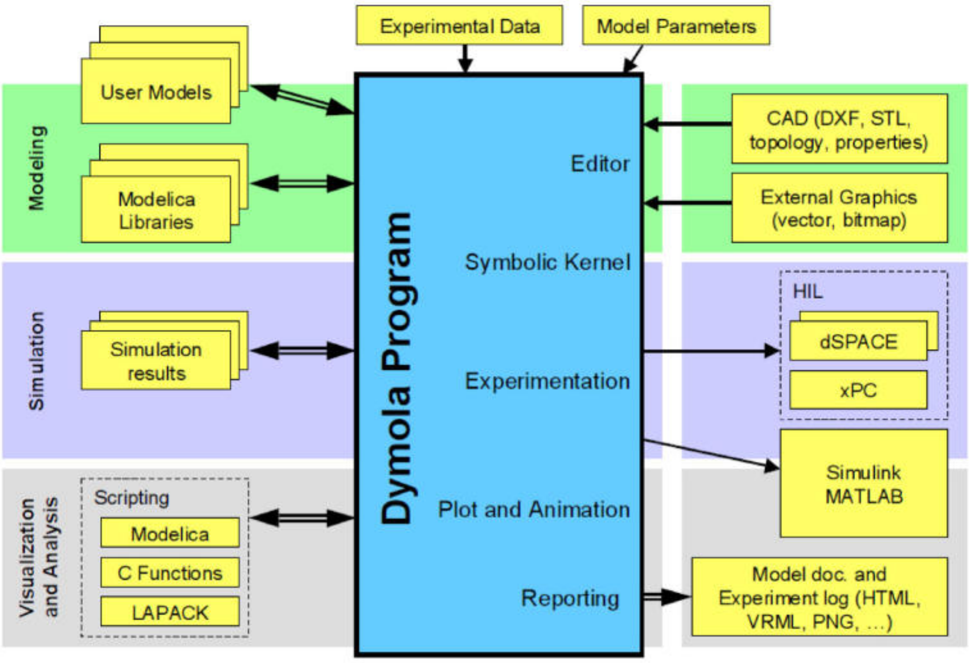


Figure 2.1. Dymola architecture [1]

2.2 Perception model

This section introduces the different mathematical models of the receptors examined in Chapter 1. These models are primarily derived from existing literature. Specifically, the models for the Pacinian corpuscle and the type I and II tactile receptors are integrated into simulations of a seated human experiencing vertical acceleration. The behavior of these models is characterized by their frequency and time responses.

Don R. Gum developed a body pressure model, depicting the body/seat dynamics as a spring mass damper system [18]. This model illustrates the skin's deflection under the ischial tuberosities, which are situated in the buttocks region. When the user is seated, the body weight causes the skin in the buttocks area to deform. The model assumes that the seat remains rigid and does not present deflection, meaning that only the skin experiences deformation. Figures 2.2 and 2.3 show how this model is derived and the mechanical circuit. The equation that describe it is given by:

$$F = m_b \frac{dx_b}{dt^2} + c_b(\frac{dx_b}{dt} - \frac{dx_c}{dt}) + k_b(x_b - x_s) \tag{2.1}$$

$$0 = -k_b(x_b - x_c) + k_c x_c - c_b(\frac{dx_b}{dt} - \frac{dx_c}{dt}) + c_s\frac{dx_b}{dt^2} \tag{2.2}$$

where m_b is one-half the mass of the body on the seat, k_b is the body flesh spring constant, c_b is the body flesh damping factor, x_b is the body displacement, k_c is the seat cushion spring constant, c_c is the seat cushion damping factor, x_c is the seat displacement.

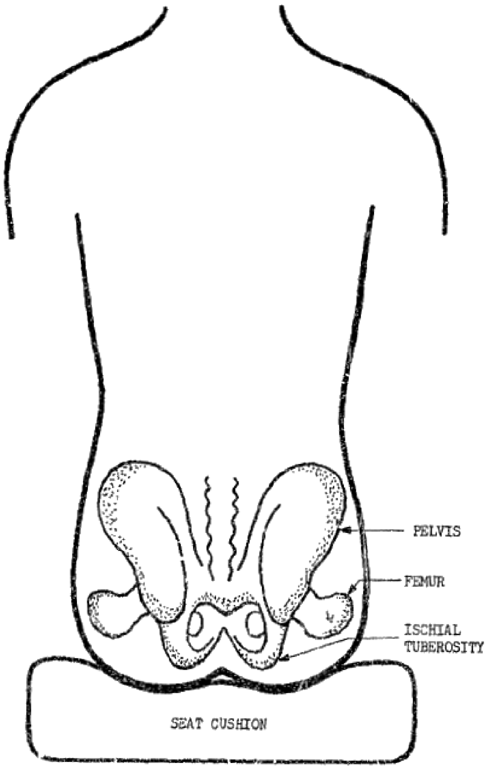


Figure 2.2. Body-seat contact [18]

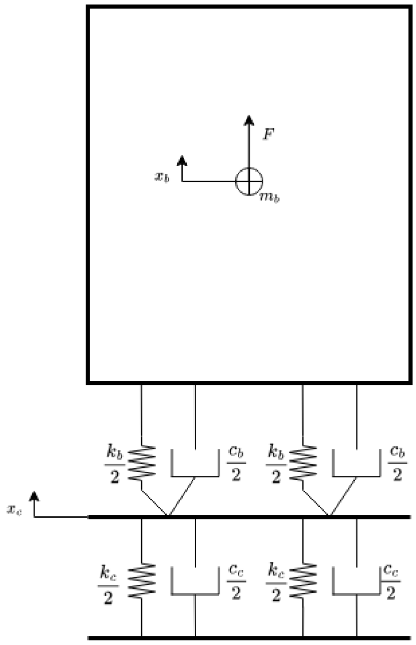


Figure 2.3. Body-seat contact mechanical model [18]

The transfer function governing the body/seat dynamics connects an external vertical force applied to the seat to the resulting skin deflection, excluding any initial deformation caused solely by the person's body weight. This transfer function indicates the output of any additional deformation resulting from the force exerted by the seat.

The system's input consists of an externally applied force, which is converted into skin displacement in meters through the body pressure dynamics transfer function. The skin deflection correlates with perceived pressure, influenced by a central nervous delay and adaptation. The adaptation of the Pacinian

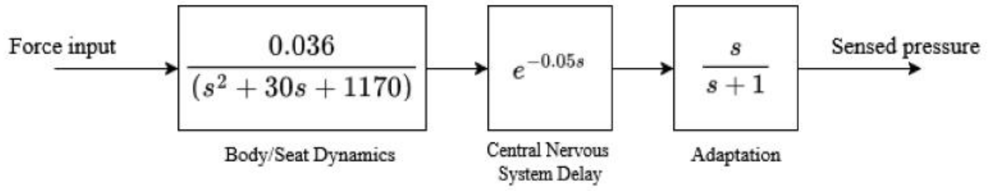


Figure 2.4. Gum Body Pressure model

corpuscle is modeled with a duration of one second, allowing it to adjust to stimuli and enhance the sensation of skin deformation. Gum notes that *"no quantitative information regarding an adaptation process is given except that it is much more rapid than the vestibular apparatus and may be around one second."* [18]. Consequently, he incorporated a time constant of one second, along with a $50ms$ delay from the central nervous system. Figure 2.4 illustrates the block diagram of this model.

Following this, Joshua Borah made modifications to the Gum model [4]. Borah's model operates under the assumption of a specific force input and considers a human mass of $70kg$, in contrast to the $28kg$ used by Gum. Gum's model posited that half of the individual's mass is supported by each ischial tuberosity, facilitating mass distribution during aircraft roll maneuvers. The increased mass in Borah's model is more representative of real-world conditions, leading to a reduction in both the damping ratio and the natural frequency of the body/seat dynamics. This revised model incorporates the features of type I and type II cutaneous receptors, along with the high-pass characteristics of the Pacinian corpuscle. Additionally, it includes a rectification effect to address the Pacinian corpuscle's inability to convey the direction of the stimulus. The output from this block diagram directs the signal to a central processor, depicted as a Kalman filter, which enables the extraction of directional information from the Pacinian corpuscle based on the rectification channel from which the signal originates.

Borah introduced an alternative body pressure model that examines receptor dynamics to develop a comprehensive integrated model.

This model features a more realistic structure compared to the one depicted in figure 2.5, as it suggests that receptor dynamics and the dynamics of the Pacinian corpuscle contribute additively rather than merging through the multiplication of the two transfer functions. It is important to note that this combination does not necessarily follow a linear pattern. The primary focus of the research cited in these references is to develop a model that integrates the entire motion-sensing apparatus using a Kalman Filter, referred to as the central processor in the subsequent text. A revised model, illustrated in Figure 2.6, is derived from the

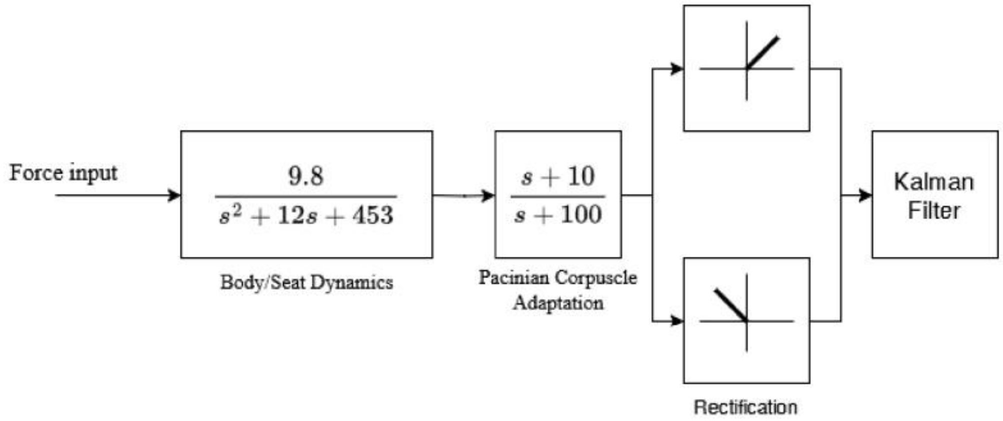


Figure 2.5. Borah Body Pressure model

frameworks established by Gum and Borah [8]. This updated model incorporates specific force sensations by utilizing the body/seat dynamics outlined in Borah's. framework. Key modifications include the introduction of a central nervous system delay term and the application of Pacinian corpuscle dynamics, replacing the combined receptor dynamics transfer function previously utilized by Borah. The Pacinian corpuscle is represented as an adaptation term with a time constant of 10 ms, selected for its capability to detect a broad spectrum of stimuli and its sensitivity to minimal skin displacements. The body dynamics model from Borah was preferred due to the use of a $70kg$ mass, which provides a more accurate average compared to the $56kg$ proposed by Gum. The $10ms$ adaptation term for the Pacinian corpuscle was also chosen, as there was limited data on its adaptation at the time of Gum's publication. An experiment referenced in Figure 1.12, conducted to assess the Pacinian corpuscle's response as reported by Borah, indicates that the generator potential of the Pacinian corpuscle adapts within a $10ms$ time frame.

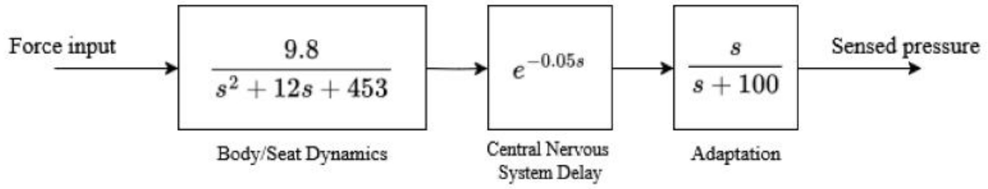


Figure 2.6. Modified Borah Body Pressure Model

2.2.1 Step Response

The analyses of step response have been performed utilizing the same input of specific force for all models. The step response for the receptor dynamics of the Borah model, the modified model, and the Gum model is presented in Figure 2.7. The Borah model features a singular transfer function that integrates the characteristics of both Type I and Type II cutaneous receptors, as well as those of the Pacinian corpuscle. In contrast, the modified model utilizes the high-pass properties of the Pacinian corpuscle, incorporating a $10ms$ adaptation period. Conversely, the Gum model applies the Pacinian corpuscle framework with an adaptation period of $1s$.

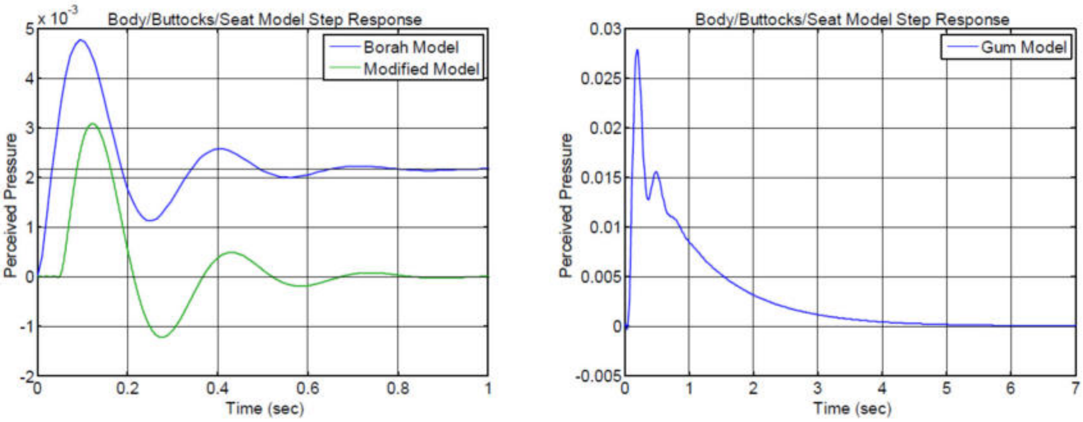


Figure 2.7. Pressure models step responses [8]

As depicted in Figure 2.7, all three models demonstrate an immediate response to stimuli, which is due to the high-pass characteristics of the Pacinian corpuscle. This suggests that a sudden force exerted on the body is detected significantly faster than by the vestibular system. These models exemplify rapid adaptation receptors, thereby enhancing the efficacy of proprioceptive sensing in high-frequency piloting tasks. The Borah model and the modified model both exhibit a settling time of $0.0391s$. In contrast, the Gum model has a significantly longer settling time of $3.91s$.

The Borah model and its modified version generally exhibit superior performance compared to the Gum model [8]. The modified model incorporates the delay of the central nervous system, yet the settling time remains nearly identical to that of the Borah model. The Borah model achieves a stable fixed value, suggesting that the dynamics of the receptor do not fully equilibrate to zero. This model is effective in identifying sustained specific force signals. Furthermore, both models demonstrate a significantly shorter settling time than the Gum model due to the absence of damping.

Given these performance characteristics, the modified Borah model was selected for implementation as the tactile perception algorithm.

2.3 Force cueing algorithm

This section describes the procedure leading to the development of the force cueing algorithm in which a dynamic model of the system and a body pressure model are included in the formulation of the optimal washout filter. This algorithm is applied to the G-seat actuator only, as an enhancement to the RMS motion cueing algorithm. In Figure 2.8 this correlation is schematized.

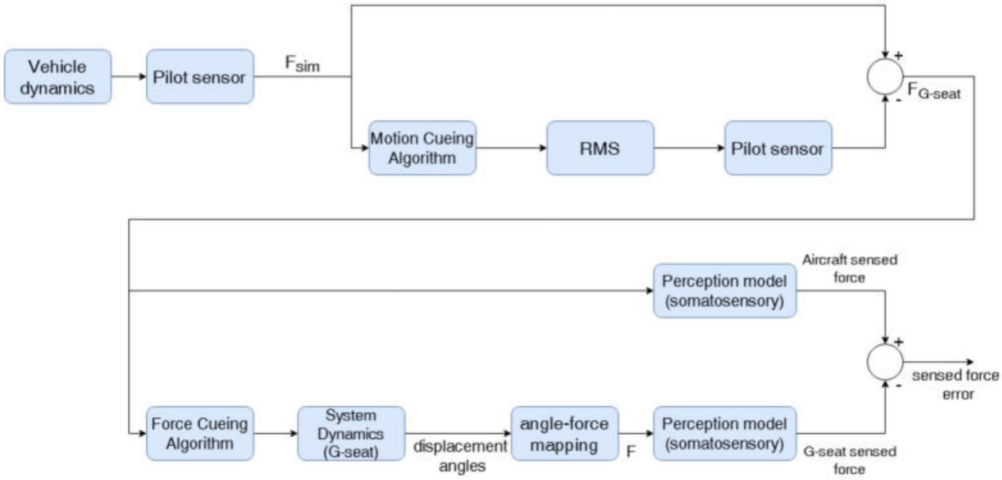


Figure 2.8. Total model

The simulation of aircraft dynamics produces an output F_{sim} from the Pilot Sensor block that reflects the inertial force experienced by the pilot in a realistic context. In the lower branch, the MCA block converts the simulated force into the desired position and attitude, which is then processed by the RMS block. This block characterizes the dynamics of the simulation platform and generates movements of the simulator based on the provided cues. Additionally, a second pilot sensor block measures the force F_{RMS} felt by the pilot within the simulator environment.

The difference between these two quantities is combined in the sum block that returns the error of these quantities. This error is processed by the force cueing algorithm considered in this paper. The upper branch processes this signal through the somatosensory perception system, which provides the optimal pressure that the pilot is expected to feel. Conversely, in the lower branch, F_{G-seat} is integrated into the force cueing algorithm, which translates these forces into angular positions via actuator dynamics. Subsequently, an additional block processes these values by establishing a relationship between flap angles and the force exerted on the body. This force is then relayed to the somatosensory perception block, yielding the G-seat sensed pressure.

Finally, a comparison between these two pressure signals determines the effectiveness of the cueing strategy in replicating realistic force perception. By combining global motion cues (via the RMS) and localized force feedback (via the G-seat), the system aims to enhance the pilot's immersion and training effectiveness, ensuring that the perceived forces closely match those expected in real-world conditions.

An optimal washout filter is applied such that the following relationship

$$u_s = W(s) \cdot u_a \tag{2.3}$$

is satisfied, where u_s is the acceleration vector used to generate the desired force, and u_a is the acceleration vector entering both the aircraft dynamic system and the simulator system. The following steps describe the procedure to obtain the transfer function $W(s)$ of the filter [24], [25]. The system S_a is the perception model that describes the relationship between the F_a and the displacement δ_a; the system S_{G-seat} system includes the seat dynamics, the angle-force mapping function, and the same tactile perception system. The following transfer function describes the body pressure model

$$\bar{f} = \frac{9.8s}{(s^2 + 12s + 453)(s + 1)} e^{-0.05s} F(s) \tag{2.4}$$

due to the continuous-time delay in the transfer function, it can be applied a third-order Pade approximation, obtaining

$$\bar{f} \approx \frac{3.289s}{(s^3 + 44.47s^2 + 828.3s + 15230)} F(s). \tag{2.5}$$

The pressure force F acting on the body is

$$F_a = m(u_a - g). \tag{2.6}$$

The actuator dynamics is described by the force-angle mapping function, which exploits the following mechanical relationship

$$F(t) = kx(t) + c_s\dot{x}(t) \tag{2.7}$$

where k [N/m] is the cushion stiffness, c_s [Ns/m] is the cushion damping factor, x is the linear displacement, F is the force exerted by the actuator. Substituting $x(t) = r\dot{\theta}(t)$ and $\dot{x} = r\dot{theta}$, the equation becomes

$$F(t) = kr\theta(t) + c_s r\dot{\theta}(t) \tag{2.8}$$

Since the goal is to precisely map forces, and the application can require mapping sudden accelerations, the acceleration has been selected as input instead of angular position, so that in the Laplace domain we have

$$\alpha(t) = \ddot{\theta}(t) \quad \Rightarrow \quad \theta(s) = \frac{\alpha(s)}{s^2}. \tag{2.9}$$

Substituting this into the force equation, we obtain

$$F(s) = kr\frac{\alpha(s)}{s^2} + crs\frac{\alpha(s)}{s^2}, \tag{2.10}$$

factoring out $\alpha(s)$ the transfer function becomes

$$\frac{F(s)}{\alpha(s)} = \frac{r(k + c_s s)}{s^2}. \tag{2.11}$$

This function describes how the flap's angular acceleration translates into force on the cushion, incorporating both stiffness and damping effects. The inertia of the flap has been neglected under the assumption that the mass of the system does not significantly affect the force transmission dynamics since the dominant terms are the spring and damping forces. This approach simplifies the system while still capturing the primary dynamic behavior.

Then, the force F acting on the pilot is

$$F(s) = \frac{r(k + c_s s)}{s^2}\alpha(s) + \frac{m}{s^2}(a - g) \tag{2.12}$$

where g is the gravity acceleration (that can be seen as a disturbance) and a is the vertical component of the aircraft acceleration.

Combining equations (2.5) and (2.12), the transfer functions that describe the G-seat dynamic system and the aircraft become

$$\bar{f}_a = \frac{3.289s}{(s^3 + 44.47s^2 + 828.3s + 15230)} m u_a(s). \tag{2.13}$$

$$\bar{f}_s = \frac{3.289s(ms^2 + rc_s s + rk)}{(s^3 + 44.47s^2 + 828.3s + 15230)(s^2)} u_s(s). \tag{2.14}$$

Then, the state space system equations for the two systems can be defined as

$$S_a \begin{cases} \dot{x}_a = A_a x_a + B_a u_a \\ y_a = C_a x_a + D_a u_a \end{cases}, \tag{2.15}$$

$$S_s \begin{cases} \dot{x}_s = A_s x_s + B_s u_s \\ y_s = C_s x_s + D_s u_s \end{cases}, \tag{2.16}$$

where x_a and x_s are the aircraft and seat system states, and the state matrices are defined as

$$A_a = \begin{bmatrix} 44.47 & -828.34 & -15225.66 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad B_a = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad C_a = 3.289 \begin{bmatrix} 0 & m & 0 \end{bmatrix} \quad D_a = 0$$

$$A_s = \begin{bmatrix} 44.47 & -828.34 & -15225.66 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad B_s = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$C_s = 3.289 \begin{bmatrix} 0 & 0 & m & rc_s & rk \end{bmatrix} \quad D_s = 0.$$

The actuator operates within a constrained workspace due to its angle operating range. To limit the movement of the flaps, it is essential to incorporate the displacement and the linear velocity of the seat into the cost function. To achieve this, supplementary seat states, denoted as $x_d = [\iint u_s^T dt \int u_s^T dt]^T$, are introduced, and the equation governing the seat motion state is formulated as

$$\dot{x}_d = A_d x_d + B_d u_s, \tag{2.17}$$

with

$$A_d = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The perception sensation error can also be defined as

$$e = y_s - y_a. \tag{2.18}$$

Finally, the equations (2.15), (2.16), and (2.17) can be integrated to create a comprehensive system:

$$S_T \begin{cases} \dot{x}_t = A_t x_t + B_t u \\ y_t = C_t x_t + D_t u \end{cases} \tag{2.19}$$

with

$$A_t = \begin{bmatrix} A_a & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & A_d \end{bmatrix}, \quad B_t = \begin{bmatrix} B_a & \mathbf{0} \\ \mathbf{0} & B_s \\ \mathbf{0} & B_d \end{bmatrix}, \quad C_t = \begin{bmatrix} -C_a & C_s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix},$$

$$D_t = \begin{bmatrix} -D_a & D_s \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad u = \begin{bmatrix} u_a \\ u_s \end{bmatrix}$$

where $\mathbf{0}$ and $\mathbf{I}$ are dimensionally suited zero and identity matrices.

The similarity between equation (2.15) and equation (2.16) results in the presence of common poles and zeros, leads to an ill-defined optimization problem. Hence, it is essential to eliminate the uncontrollable modes by canceling the pole-zero pairs from the transfer function, recovering the *minimal realization* of the total system S_t. The resultant system, denoted as S_{min} is

$$S_{min} \begin{cases} \dot{x}_{min} = A_{min} x_{min} + B_{min} u \\ y_{min} = C_{min} x_{min} + D_{min} u \end{cases}, \tag{2.20}$$

where B_{min} and D_{min} can be divided in accordance with the partitioning of u

$$u = \begin{bmatrix} u_a \\ u_s \end{bmatrix}, \quad B_{min} = \begin{bmatrix} B_{min1} & B_{min2} \end{bmatrix}, \quad D_m = \begin{bmatrix} D_{min1} & D_{min2} \end{bmatrix}$$

The signal u_a is modeled as the output of a linear system driven by white noise n, i.e.

$$N = \begin{cases} x_n & = A_n x_n + B_n n, \\ u_a & = C_n x_n. \end{cases} \tag{2.21}$$

Here, x_n represents the internal state of the filter, while A_n, B_n, and C_n are the system parameters that define the shaping filter. These parameters can be specified as $A_n = -\gamma$, $B_n = \gamma$, and $C_n = 1$, where γ denotes the first-order filter break frequency.

The minimized system (2.20) and the noise system (2.21) can be combined to form the desired system equation

$$\begin{aligned} \dot{x} &= Ax + Bu_s + Hw, \\ y &= Cx + Du_s. \end{aligned}$$

where $x = \begin{bmatrix} x_m \\ x_n \end{bmatrix}$ represents the combined state vector. While A, B, C, D, and H are defined as follows

$$A = \begin{bmatrix} A_{min} & B_{min1} C_n \\ 0 & A_{min} \end{bmatrix}, \quad B = \begin{bmatrix} B_{min2} \\ 0 \end{bmatrix}, \quad H = \begin{bmatrix} 0 \\ B_n \end{bmatrix},$$

$$C = C_{min} + D_{min1} C_n, \quad D = D_{min2}.$$

Given the actual system S_a, the simulator system S_s and the noise filter N, the goal is to find a transfer function which generates u_s gievn $u_a(\tau), -\infty < \tau \leq t$, for all t, minimizing the cost function J, which is defined as

$$J = E\left[\int_0^\infty \left(e^T Q e + x^T R_d x + u^T R u\right) dt\right]. \tag{2.22}$$

The cost function J can be reformulated into an optimal control framework using the following equation

$$J = E\left[\int_0^\infty \left(x^T R_1 x + 2x^T R_{12} x + u_s^T R_2 u_s\right) dt\right], \tag{2.23}$$

where

$$R_1 = C^T \begin{bmatrix} Q & 0 \\ 0 & R_d \end{bmatrix} C, \quad R_{12} = C^T \begin{bmatrix} Q & 0 \\ 0 & R_d \end{bmatrix} D, \quad R_1 = R + D^T \begin{bmatrix} Q & 0 \\ 0 & R_d \end{bmatrix} D,$$

The challenge of minimizing (2.23) while adhering to the system state equation (2.3)constitutes a typical optimal control problem, and the objective function J reaches its minimum when

$$u_s = -Kx, \tag{2.24}$$

where

$$K = R_2^{-1}(B^T P + R_{12}^T) \tag{2.25}$$

and P is the solution of the Riccati equation

$$P(A - BR_2^{-1}R_{12}^T) + (A - BR_2^{-1}R_{12}^T)P + R_1 - R_{12}R_2^{-1}R_{12}^T - PBR_2^{-1}B^T P = 0 \tag{2.26}$$

Sivan details the solution to the design challenge related to an optimal linear and quadratic simulator [25], achieved in this study through computer simulations conducted with MATLAB. Furthermore, Sivan describes the methodology for deriving the optimal washout filter to satisfy the relationship of the equation 2.3. In this thesis, a corresponding Simulink model was utilized to monitor constraints, allowing for modifications to the weight matrices of the function designated for minimization.

Chapter 3

Force cueing algorithm implementation

This chapter provides a comprehensive overview of the methodology employed to validate the optimal control algorithm and elaborates on the tuning process involved in fine-tuning the algorithm to achieve the desired performance metrics. Additionally, the chapter presents the results obtained from a systematic comparison between the optimal control algorithm and a conventional washout algorithm developed in the previous work on this system [15], highlighting the differences in performance and effectiveness.

3.1 Force cueing algorithm implementation

The optimal control filter has been implemented in the Modelica simulation environment, to evaluate its effectiveness and improvements over the classical washout algorithm. Figure 3.1 shows the structure of the model implemented in Modelica [15], which handles force cueing:

- The orange block is responsible for generating test values for the acceleration vector, which serves as input for the Modelica model. This block allows for the configuration of various test functions, including a step function, square wave function, or sine wave. Moreover, this block can facilitate the definition of more intricate flight trajectories. To achieve this, accelerations are extracted from the log file of a simulated flight maneuver;

- The red block contains the washout filter, which will be discussed in the next section, and the calculation of the directions of the forces applied by the actuator on the seat. To accurately ascertain the directions of the forces, it is essential to convert the unit vector in the z-direction from the specific flap coordinate system to the global coordinate system. This conversion is influenced by the spatial position of the actuator and the current deflection of its flap, denoted as ϕ, which changes during operation. In Modelica, the transformation matrix $\mathbf{T}$ can be directly derived from the spatial position of an object, which is necessary for transitioning from the global coordinate system to the relevant object coordinate system;

- The green block contains the simulation of the G-seat, consisting of four flap actuators, two positioned in the seat and two positioned in the backrest (Figure 1.6). This setup allows for the testing and visualization of the G-seat's behavior without the need for a physical prototype. Furthermore, any configuration of the G-seat, incorporating various numbers of actuators, can be created at any location within the simulation environment;

- The 'Hardware Recalculation' block transforms the position values into an input suitable for the hardware. In this case, EtherCAT (Ethernet for Control Automation Technology) is used to let the controller communicate with the servo drives. It necessitates that the target positions to be expressed in tenths of a degree;

- The blue block handles the connection to the Modelnet server.

3.2 Tuning and testing

The software-in-the-loop evaluation of the force cueing algorithm aims to select suitable parameters that meet the required performance (tuning) and verify that the desired motion is generated without exceeding physical constraints. This section outlines the procedure for tuning the parameters of the optimal washout algorithm.

3.2.1 Tuning

The simulation process involves the configuration of the software environment, the setup for the close loop simulation, and the iterative adjustment of the parameters to achieve optimal performance. Part of the work focused on determining the appropriate weights for the LQR matrices. It is a crucial procedure to achieve the desired trade-off between performance and control effort. The process involves iterative adjustments based on system response, stability, and real-world constraints.

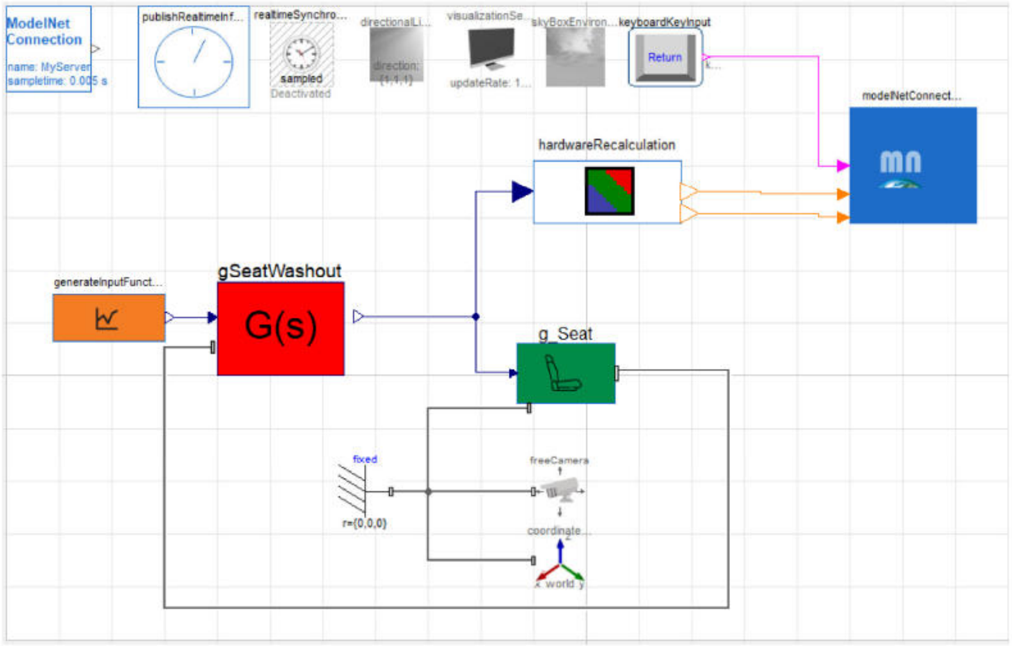


Figure 3.1. Modelica schematics of the G-seat.

The tuning process was conducted using MATLAB, where the sensed forces from model S_a and model S_s were compared to minimize the error. After identifying the optimal tuning parameters, the algorithm for the optimal washout filter was integrated into the G-seat model within Modelica to assess its performance. The weight matries in eq. (2.26) have been defined as

$$Q = q, \quad R_d = \begin{bmatrix} r' & 0 \\ 0 & r'' \end{bmatrix}, \quad R = r.$$

The Q and R_d matrices entries represent the cost associated with deviating from the desired value of the state or output. A higher value means that deviation from that particular state is highly undesirable, forcing the controller to prioritize minimizing it, while a lower value means that deviation in that state is tolerated more. The same reasoning can be applied to the value R, which acts on the control action. Table 3.1 shows the weighted entries of Q,R_d, and R.

Increasing the values of the Q matrix enhances the overall performance of the system, ensuring that the pressure perceived by the pilot on the G-seat closely replicates the sensations experienced in the actual aircraft. A similar improvement in performance can be achieved by reducing the contribution of the control effort penalization, represented by the R matrix.

Regarding the states that define the actuator constraints, increasing the weighting associated with angular position and angular velocity in the R_d matrix

Entry	Weighted variable
q_1	sensed pressure error
r'	actuator angular position
r''	actuator angular velocity
r	control action

Table 3.1. Physical meaning of the LQR weighting matrices entries

generally results in a decline in overall system performance. However, the impact of these constraints is comparatively smaller than the influence of the primary state cost terms in Q. Consequently, higher weightings are assigned to these states to maintain actuator movements within their physical limitations while improving steady-state tracking.

To further optimize the system response, the cost associated with actuator velocity is kept relatively low. This allows the algorithm to more effectively track the acceleration profiles typical of aircraft dynamics, ensuring a balance between actuator compliance and motion fidelity.

The final values, chosen at the end of this iterative process are

$$Q = 5{,}5 \cdot 10^6, \quad R_d = \begin{bmatrix} 1.25 \cdot 10^4 & 0 \\ 0 & 10^3 \end{bmatrix}, \quad R = 10^{-4}.$$

Figure 3.2 shows the enhancements realized by the optimal washout algorithm (b) in comparison to the traditional washout algorithm (a), specifically regarding the perceived force experienced through the G-seat system. The optimal algorithm demonstrates a remarkable ability to closely replicate the perceived force generated by the aircraft system while adhering to the physical limitations, as illustrated in Figure 3.3, of position described in section 1.1.2 and velocity, which are about $600 rad/s$ for the used motor.

Figure 3.4 illustrates more distinctly the significant reduction in the discrepancy between the forces experienced in the aircraft system and those calculated using both the classical washout algorithm and the optimal washout algorithm, with the error decreasing by an order of magnitude. Additionally, the classical washout algorithm exhibits a degree of lag, which hinders its ability to follow changes in acceleration accurately.

3.3 Results

Once the algorithm was fine-tuned, it was integrated into the Modelica model, which simulates the G-seat at the software level. This tool facilitated tests to

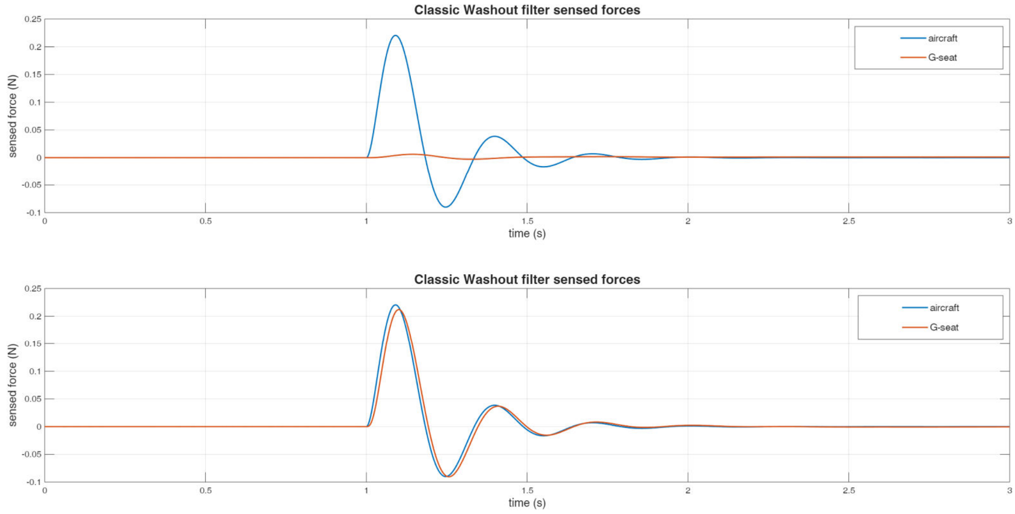


Figure 3.2. a) Comparison of the perceived forces in the classical washout algorithm, b) comparison of the perceived forces in the optimal washout algorithm

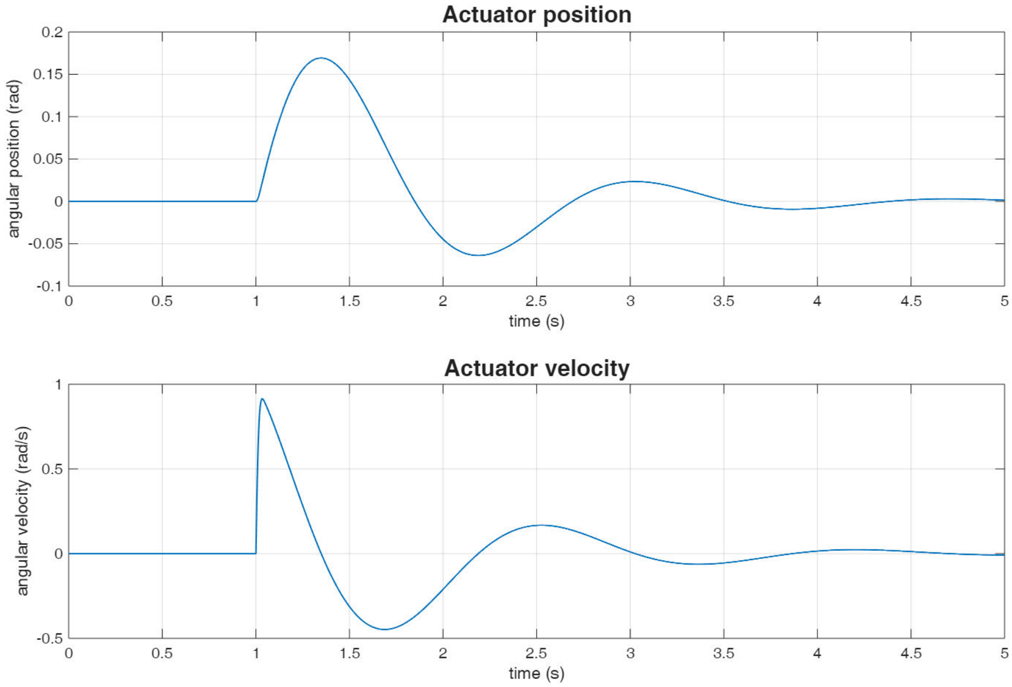


Figure 3.3. G-seat constraints x_d states

determine whether any enhancements had been realized. The actuators angular position performance was tested by applying a step acceleration input along

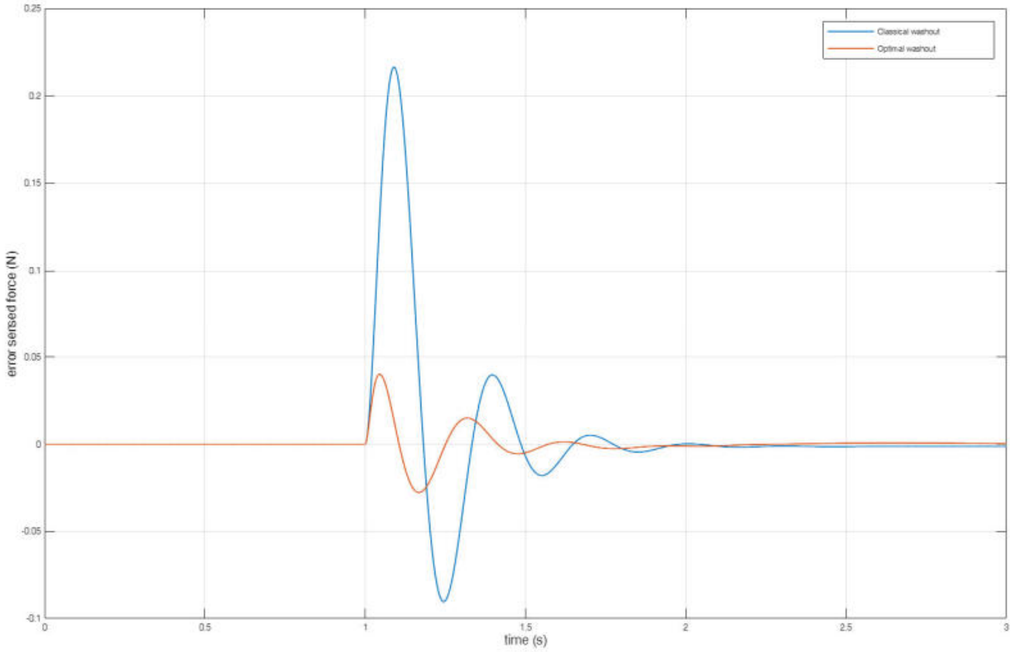


Figure 3.4. Sensed force error comparison between the classical washout algorithm and the optimal washout algorithm

the z-axis, as these accelerations exert greater stress on the flap movement and influence the algorithm performance. Furthermore, the flaps response to a trajectory input was assessed by examining its z-component for the same reasons mentioned earlier.

Then, the outcomes were compared to those obtained using the traditional washout filter.

In this experiment, a step input of $2m/s^2$ was applied along the z-axis at time $t = 1s$ to evaluate the system response to rapid changes. As shown in Figure 3.5, analyzing the system response using the optimal washout algorithm (represented by the blue line) reveals a rise time of approximately $0.3s$. The actuator stabilizes at a lower value compared to the classic washout method and does not exhibit overshoot, indicating a well-controlled response. Additionally, the actuator achieves stabilization swiftly.

Conversely, the classic washout (red line) demonstrates a more gradual response, taking a longer duration to reach steady state and stabilizing at a higher value than the optimal washout. This results in increased actuator movement. The optimal washout algorithm effectively minimizes unwanted oscillations and excessive movements by considering physical constraints, whereas the classic washout algorithm may lead to greater mechanical strain over time.

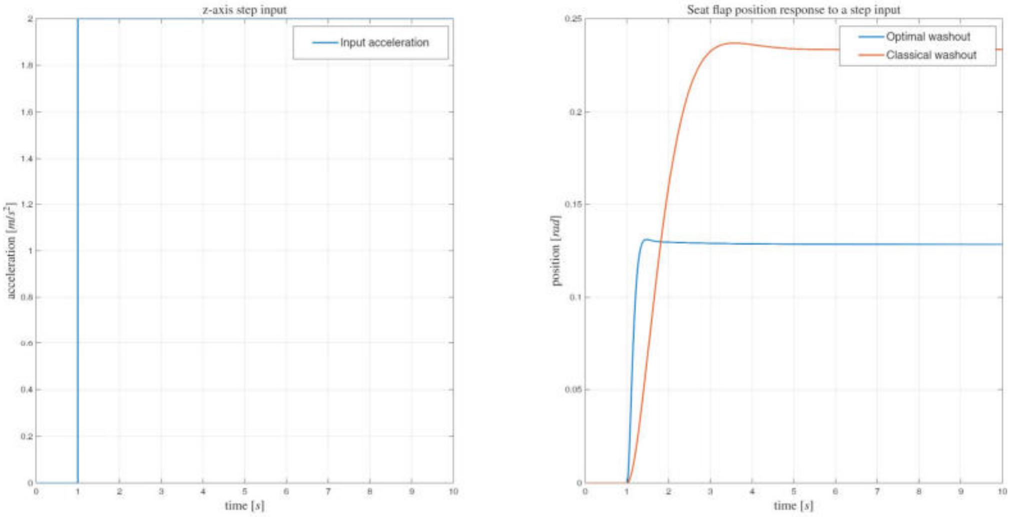


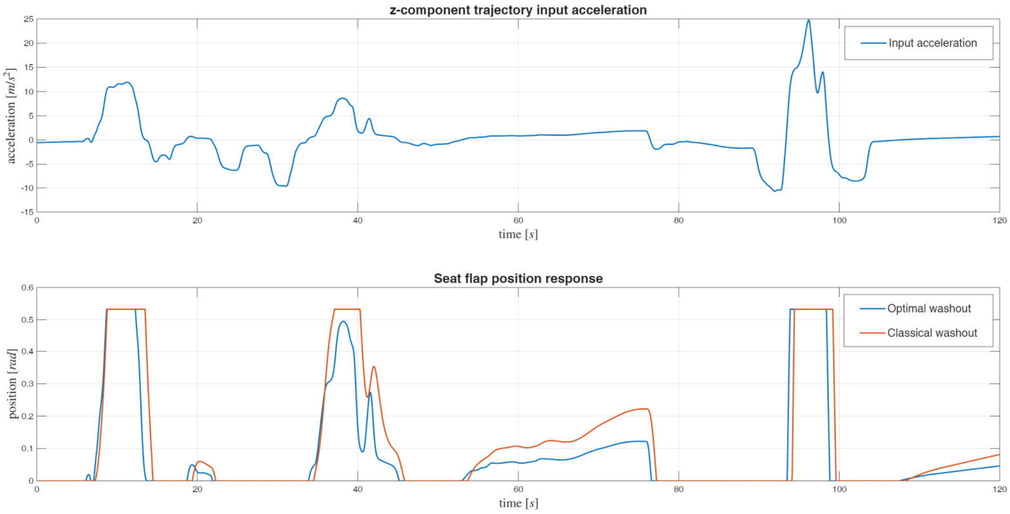
Figure 3.5. a) step input acceleration of $2m/s^2$, b) actuator position response

Figure 3.6 depicts the response of the G-seat flap in response to a trajectory, specifically focusing on its z-axis component. This trajectory encompasses both high-frequency fluctuations and low-frequency patterns, rendering the test particularly suitable for evaluating the algorithm capabilities.

The input is marked by frequent acceleration changes, evident in the numerous peaks and troughs, with peaks surpassing $20m/s^2$, which is higher than the step input value. Additionally, it is observed that the optimal washout algorithm demonstrates a quicker response to these rapid acceleration changes. It is important to highlight that negative accelerations result in a position of zero, as the actuator cannot simulate negative pressure sensations.

Despite the primary drawback of the optimal washout algorithm is its computational cost and processing time, this issue is mitigated in the present application. This is largely because the weights of the LQR are determined offline, significantly reducing the real-time computational burden. As a result, the algorithm remains efficient while preserving its ability to enhance motion cueing fidelity.

Furthermore, this test illustrates that the optimal washout algorithm, which inherently accounts for constraints, relies less on the position limiter compared to the traditional washout method. The traditional washout exhibits a slower response, with delays of up to $1s$ in some instances, whereas the optimal washout algorithm offers superior filtering, which is essential for delivering an accurate response to the pilot.



z-component trajectory input acceleration

Seat flap position response

Figure 3.6. a)trajectory input z-axis acceleration, b) actuator position response

Chapter 4

Conclusion and future works

4.1 Conclusion

This thesis focuses on the development, integration, and assessment of a force cueing algorithm aimed at enhancing the functionality of a G-seat system intended for implementation within the DLR's RMS. It examined various methodologies for motion cueing algorithms, analyzing their respective advantages and disadvantages to determine whether enhancements could be made to the currently employed algorithm. After identifying the optimal washout algorithm as a viable option, it was tailored to fit the specific application. Notably, this application is restricted to the G-seat rather than the entire mobile platform, positioning this algorithm as a supplementary tool to the primary motion cueing algorithm that regulates the RMS.

The selected optimal washout algorithm incorporates the physical limitations of the system alongside an assessment of the perception model. Consequently, the most appropriate perceptual system for this application was scrutinized, leading to the adoption of a haptic system. A review of the literature was conducted to understand how this system has been modeled, ultimately resulting in the selection of the most accurate model currently available, specifically the one identified by Cardullo as the *modified Borah.*

The washout algorithm was developed and evaluated using MATLAB, demonstrating a notable enhancement in performance characterized by quicker and more precise responses, along with a substantial reduction in sensation error. Subsequently, the algorithm was assessed within a simulator developed in Modelica and compared against the traditional washout algorithm. The findings indicated that the optimal washout method markedly enhances the accuracy and stability of force cueing, leading to decreased response times and minimized mechanical stress. These enhancements in performance directly influence the overall fidelity of motion cueing, thereby providing a more immersive and physically realistic experience.

A key limitation of the optimal washout algorithm is its high computational cost and processing time. However, this issue is mitigated in the current application through the offline determination of LQR weights, significantly reducing computational demands during real-time operation. As a result, the algorithm remains efficient while enhancing motion cueing fidelity.

4.2 Future works

While the findings are encouraging and Modelica provides precise software-level representations, this research does not include hardware validation; the actual performance of the G-seat has not been assessed. By implementing the optimal force cueing algorithm within a real G-seat system, researchers can assess its effectiveness by contrasting it with the traditional washout algorithm under genuine operational conditions. Conducting tests with pilots or trained individuals would facilitate the assessment of subjective realism, confirming that the enhanced cues correspond with human perception and expectations. To achieve the optimal MCA, it is essential to examine the impact of the weighting matrices on the filter characteristics in more detail, including an analysis of the sensitivity to minor variations in the weighting values. Furthermore, the incorporation of a more comprehensive perception model is a viable consideration. Another potential approach to enhance performance is to adopt a model predictive control (MPC) algorithm that considers the physical constraints discussed in section 1.1.2. By applying the findings of this work to practical scenarios and investing time in testing campaigns, it is possible to create a generation of force cueing systems that offer improved realism and performance.

Bibliography

[1] Dynasim AB. *Dymola Dynamic Modeling Laboratory User's Manual.* 1992-2004.

[2] Billy R Ashworth, Burnell T McKissick, and Russell V Parrish. *Effects of motion base and g-seat cueing of simulator pilot performance.* Tech. rep. 1984.

[3] Kenneth R Boff, Lloyd Kaufman, and James P Thomas. *Handbook of perception and human performance.* Vol. 1. Wiley New York, 1986.

[4] I Borah. "Sensory mechanism modeling". In: *Air Force Human Resources Laboratory Report* (1977).

[5] Julia Born et al. "Modified Optimal Motion Cueing Algorithm with Haptic Feedback for Full Motion Simulation for a Reduced Gravity Environment". In: *2024 International Conference on Space Robotics (iSpaRo)*. IEEE. 2024, pp. 307–314.

[6] Mattia Bruschetta, Fabio Maran, and Alessandro Beghi. "A fast implementation of MPC-based motion cueing algorithms for mid-size road vehicle motion simulators". In: *Vehicle system dynamics* 55.6 (2017), pp. 802–826.

[7] Mattia Bruschetta et al. "A nonlinear MPC based motion cueing strategy for a high performance driving simulator with active seat". In: *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*. IEEE. 2018, pp. 23–28.

[8] Frank Cardullo, Anthony Stanco, and Ruud Hosman. "The Somatosensory System: Physiology and Models". In: *AIAA Modeling and Simulation Technologies Conference.* 2010, p. 8357.

[9] Sergio Casas, Ricardo Olanda, and Nilanjan Dey. "Motion cueing algorithms: a review: algorithms, evaluation and tuning". In: *International Journal of Virtual and Augmented Reality (IJVAR)* 1.1 (2017), pp. 90–106.

[10] Si Chen et al. "Vibrotactile sensation: a systematic review of the artificial pacinian corpuscle". In: *Journal of Bionic Engineering* 20.4 (2023), pp. 1401–1416.

[11] William WY Chung et al. *Investigation of effectiveness of the dynamic seat in a Black Hawk flight simulation.* Citeseer, 2001.

[12] Torin K Clark et al. "Mathematical models for dynamic, multisensory spatial orientation perception". In: *Progress in brain research* 248 (2019), pp. 65–90.

[13] B Conrad and SF Schmidt. *The calculation of motion drive signals for piloted flight simulators.* Tech. rep. 1969.

[14] Mehmet Dagdelen et al. "Model-based predictive motion cueing strategy for vehicle driving simulators". In: *Control Engineering Practice* 17.9 (2009), pp. 995–1003.

[15] Simon Dammann. "Entwicklung eines Force Cueing Algorithmus für Sitzpolster Druckaktuatoren zum Einsatz im DLR Robotic Motion Simulator". PhD thesis. Technische Universität München, 2023.

[16] Martin Fischer. "Motion-cueing-algorithmen für eine realitätsnahe bewegungssimulation". PhD thesis. DLR Braunschweig, 2009.

[17] Nikhil JI Garrett and Matt Best. "Driving simulator motion cueing algorithms–a survey of the state of the art". In: (2010).

[18] Don R Gum. *Modeling of Human Force and Motion Sensing Mechanisms.* Citeseer, 1973.

[19] Miguel Neves et al. "A Novel Human-in-the-Loop Testing Facility for Space Applications". In: *2nd International Conference on Flight Vehicles, Aerothermodynamics and Re-entry Missions & Engineering (FAR).* 2022.

[20] Martin Otter and Hilding Elmqvist. "Modelica overview". In: *URl: https://www.modelica. org/education/educationalmaterial/lecture-material/english/ModelicaOverview. pdf* (2001).

[21] Russell V Parrish et al. "Coordinated adaptive washout for motion simulators". In: *Journal of aircraft* 12.1 (1975), pp. 44–50.

[22] Duc An Pham. "A study on state-of-the-art motion cueing algorithms applied to planar motion with pure lateral acceleration–comparison, autotuning and subjective evaluation on a KUKA robocoaster serial ride simulator". PhD thesis. Dissertation, Duisburg, Essen, Universität Duisburg-Essen, 2017, 2017.

[23] Lloyd D Reid and Meyer A Nahon. "Flight simulation motion-base drive algorithms. Part 1: Developing and testing the equations". In: (1985).

[24] Hua Shao et al. "An optimal force cueing algorithm for dynamic seat". In: *2009 IEEE international conference on control and automation.* IEEE. 2009, pp. 2217–2222.

[25] Raphael Sivan, Jehuda Ish-Shalom, and Jen-Kuang Huang. "An optimal control approach to the design of moving flight simulators". In: *IEEE Transactions on Systems, Man, and Cybernetics* 12.6 (1982), pp. 818–827.

[26] Robert J Telban and Frank M Cardullo. *Motion cueing algorithm development: Human-centered linear and nonlinear approaches.* Tech. rep. 2005.

[27] A White. "G-seat heave motion cueing for improved handling in helicopter simulators". In: *Flight Simulation Technologies Conference and Exhibit.* 1989, p. 3337.