2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

The DLR Robot Motion Simulator Part II: Optimization based path-planning

Tobias Bellmann, Martin Otter, Gerd Hirzinger

Abstract— **In Part I of this paper, a novel motion simulator platform is presented, the DLR Robot Motion Simulator with 7 degrees of freedom (DOF). In this Part II, a path-planning algorithm for mentioned platform will be discussed. By replacing the widely used hexapod kinematics by an antropomorhic, industrial robot arm mounted on a standard linear axis, a comparably larger workspace at lower hardware costs can be achieved. But the serial, redundant kinematics of the industrial robot system also introduces challenges for the path-planning as singularities in the workspace, varying movability of the system and the handling of robot system's kinematical redundancy. By solving an optimization problem with constraints in every sampling step, a feasible trajectory can be generated, fulfilling the task of motion cueing, while respecting the robot's dynamic constraints.**

I. INTRODUCTION

The path-planning algorithm of a motion simulater has to map the large-scale movements of a vehicle like an aircraft or a car to the limited workspace of the simulator, utilizing the gravity to display long-lasting accelerations. For classical motion simulator designs based on the Stewart platform [1], the according path-planning algorithms, mostly based on the principle of washout filters [2][3], are widely researched and optimized for several applications as flight-simulations [4] or vehicle simulations [5].

By introducing a serial mechanical configuration like an industrial robot as simulator base, several challenges for the path-planning algorithm are introduced: The movability of the end-effector depends on the actual joint configuration, because of kinematic singularities in the workspace and a varying effective lever arm leading to varying Cartesian acceleration limits. Furthermore, it is difficult to adapt the path-planning algorithm exactly to the workspace of an industrial robot, as it has the form of a shell around the robot-base and the possible Cartesian positions and orientations are non-linearly constrained by the joint limits [6].

In the last years, path-planning algorithms for interactive motion simulation with non redundant industrial robots have been the focus of several publications [7][8] and the use-case driving simulator has been implemented [9][10].

This paper focuses on the path-planning of a kinematically redundant motion simulator, where a linear axis is used to increase the available workspace of the robot, as presented in Part I of this paper, see also Fig.1. To handle the mentioned challenges, the following approach shown in Fig. 2(left) has been chosen:

The authors are with Institute of Robotics and Mechatronics, German Aerospace Center, 82234 Oberpfaffenhofen, Germany (Tobias.Bellmann@dlr.de)

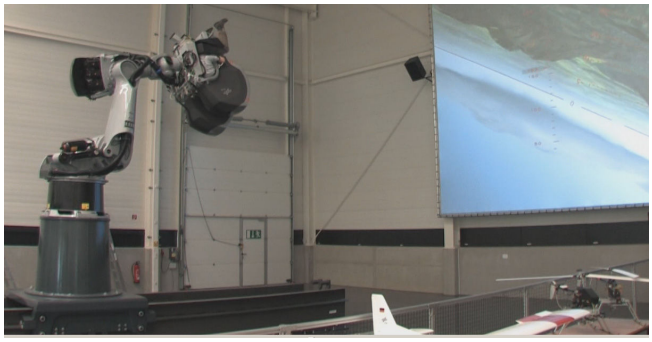

Fig. 1. The DLR Robot Motion Simulator performing an interactive roll maneuver. In the right side of the picture, a part of the flight visualization as seen by the simulator pilot is visible.

First, a Cartesian trajectory is planned, using a modified washout-filter approximately adapted to the simulator workspace. This reference trajectory represents the motion cues to create the impression of the simulated vehicle's movements. As this trajectory is planned without considering the dynamics of the robot system, it can exceed the kinematic and dynamic limits of the robot system.

Thus, in a second step, a local optimization will be performed on this reference trajectory. The result of this optimization are the joint trajectories for a feasible simulator movement following the reference trajectory as close as mechanically and dynamically possible. As the simulator is interactive, this optimization has to be performed in every sample step of the simulation.

Since the used optimization procedure introduces some implications for the design of the washout filter, the optimization step will be treated first in this paper.

II. LOCAL OPTIMIZATION BASED TRAJECTORY GENERATION

The approach for an optimization based trajectory generation presented in the next sections is based on the method of singularity handling presented in [11], and has been modified to meet the requirements of a path-planning algorithm.

A. Problem Formulation

Let

$$\underline{x} = \underline{f}(\underline{q}) \quad \underline{x} = \begin{bmatrix} \underline{r} \\ \underline{\varphi} \end{bmatrix} \quad \underline{r} \in \mathbb{R}^3 \quad \underline{\varphi} \in \mathbb{R}^3 \quad \underline{q} \in R^n \qquad (1)$$

be the forward kinematics of the robot arm ($\underline{x}$ effector position, $\underline{q}$ joint positions, see Fig 2 (right)). With $n > 6$

978-1-61284-385-8/11/$26.00 ©2011 IEEE

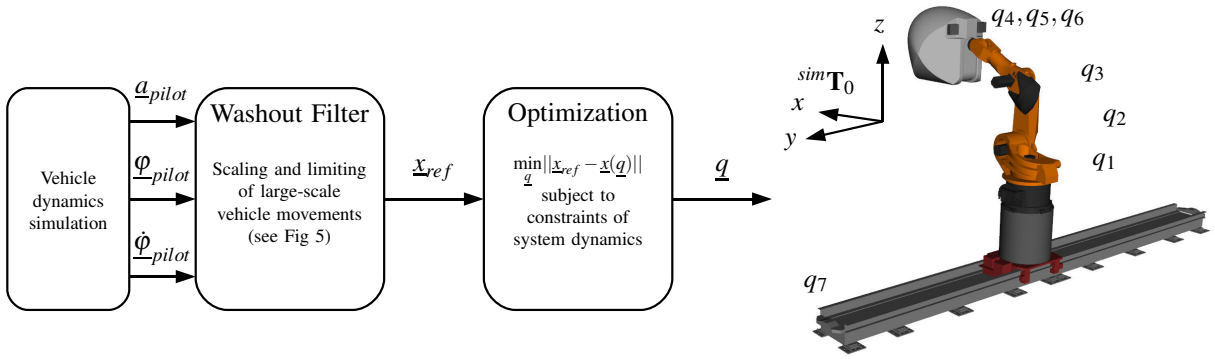

Fig. 2. Path-planning diagram (left), showing the two processing steps from real world movements to simulator joint trajectories. The washout filter maps the movements of the simulated vehicle to the restricted Cartesian workspace of the simulator. The optimization then calculates a feasible trajectory for the robot joint angles. The used simulator configuration as well as the joint numeration and coordinate system are shown (right).

the inverse kinematics

$$\underline{q} = \underline{f}^{-1}(\underline{x}) \tag{2}$$

becomes kinematically redundant [12]. This implies that there are a multitude of joint trajectories $\underline{q}(t)$ leading to the very same effector trajectory $\underline{x}(t)$. One approach to calculate a desired inverse for redundant-robots is the minimization of the error between reference and effector position under constraints, with the robot joint positions as optimization variables [13]

$$\min_{\underline{q}} ||\underline{x}_{ref} - \underline{x}(\underline{q})||, \tag{3}$$

where $\underline{x}_{ref}$ would be the reference position calculated by the washout filter treated in section V. As pictured in Fig. 3, the trajectory $\underline{q}^i = \underline{q}(t_i)$ is calculated as a result of an optimization in every sample step.

Utilizing additional optimization criteria, the ambiguity of the null-space can be handled. The viewed contributions in this field concentrate on the solution of this ambiguity by introducing additional criteria and constraints for the desired null-space movements.

By introducing constraints into this optimization problem, the dynamic of the robot system can be considered. Furthermore, in the vicinity of singularities, the dynamical limits of the system, as well as the stability of the solution are guaranteed by the constraints, but a positioning error is introduced for these mechanical impossible cases. This approach calculates the locally optimal movement of the robot-system to fulfill the tasks given by the optimization criteria. It hereby handles singularities, the varying movability and can solve the ambiguity of the redundant robot system.

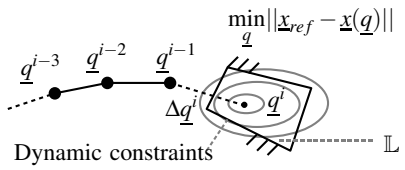

Fig. 3. Using local optimization to determine the next joint angle in every timestep. The optimization problem $\min_{\underline{q}} ||\underline{x}_{ref} - \underline{x}(\underline{q})||$ is solved, subject to the dynamic constraints of the robot (black box), defining a space of possible solutions $\mathbb{L}$.

B. Converting a nonlinear dynamic system to a time-discrete optimization problem

In order to consider the dynamic behavior of a system in the optimization problem, the differential equations, describing the system, have to be reformulated as constraints of the optimization.

A time-discrete nonlinear system is given by the system equations

$$\underline{q}^i = \quad \underline{q}^{i-1} + \Delta\underline{q}^i \quad \underline{q} \in \mathbb{R}^n \tag{4}$$

$$\Delta\underline{q}^i = \quad \underline{f}(\underline{q}^{i-1}, ...) + \mathbf{B}(q^{i-1}, ...)\underline{u}^i \tag{5}$$

This system is reformulated as optimization problem to be solved in every sample step. The requested form of the optimization problem shall be

$$\min_{\Delta\underline{q}^i} \left| \mathbf{\Phi}(\underline{q}^{i-1}, ..., \underline{q}^i_{ref}, ...) \right| \tag{6}$$

$$\text{subject to } \mathbf{G}\Delta\underline{q}^i \geq \underline{h} \tag{7}$$

The solution of the optimization problem shall be constrained by the linear inequality constraint $\mathbf{G}\Delta\underline{q}^i \geq \underline{h}$, so the system dynamics can be considered. Assuming the input $\underline{u}$ is limited to $\underline{u}_{max}, \underline{u}_{min}$, the dynamic system (5) can be transformed into inequality constraints

$$\underbrace{\begin{bmatrix} \mathbf{B}^{-1} \\ -\mathbf{B}^{-1} \end{bmatrix}}_{\mathbf{G}} \Delta\underline{q}^i \geq \underbrace{\begin{bmatrix} \underline{u}_{min} + \mathbf{B}^{-1}\underline{f}(\underline{q}^{i-1}, ...) \\ -\underline{u}_{max} - \mathbf{B}^{-1}\underline{f}(\underline{q}^{i-1}, ...) \end{bmatrix}}_{\underline{h}}, \tag{8}$$

defining a solution space $\mathbb{L} \subseteq \mathbb{R}^n$:

$$\mathbb{L} = \left\{ \Delta\underline{q} \quad | \quad \underline{u}_{min} \leq \mathbf{B}^{-1}(\Delta\underline{q} - \underline{f}(\underline{q}^{i-1}, ...) \leq \underline{u}_{max} \right\} \tag{9}$$

The solution space contains all potential results of the optimization problem (6), therefore the resulting trajectory $\underline{q}^i$ is bounded by the limitations of the system dynamics

(5). The development of the trajectory is controlled via the optimization criterion $\Phi(\underline{q}^{i-1},...,\underline{q}^{i}_{ref},...)$.

III. CONSTRAINTS INTRODUCED BY THE ROBOT SYSTEM

The robot system introduces several constraints into the path-planning process, due to mechanical or electrical limitations. The dynamics of the robot has to be considered as well as limitations like hardware stops, maximum joint accerlerations and speeds. The goal of this section is to collect and to reformulate these constraints to integrate them into the optimization problem (6).

A. Dynamic constraints

The dynamic model of the robot is described by the following equation

$$\mathbf{M}(\underline{q})\ddot{\underline{q}}+N(\underline{q},\dot{\underline{q}})+G(\underline{q})=\underline{\tau}_{joint}, \tag{10}$$

where $\mathbf{M}(\underline{q})$ is the mass inertia matrix of the robot, $N(\underline{q},\dot{\underline{q}})$ are the non-conservative forces like friction, centripetal and coriolis forces and G(q) the gravitational forces. $\underline{\tau}_{joint}$ are the torques, applied in the single joints. The components of this dynamic equation can be calculated with the euler-lagrange equations [14].
Using the discretization

$$\dot{q}^i=\frac{\Delta q^i}{\Delta T},\quad \ddot{q}^i=\frac{\Delta q^i-\Delta q^{i-1}}{\Delta T^2}, \tag{11}$$

with the sample time ΔT, (10) can be rewritten in the form of (8) by approximating M,N,G in (10) with the values from the last sample step:

$$\frac{1}{\Delta T^2}\mathbf{M}^{i-1}\Delta\underline{q}^i \geq \quad \underline{\tau}_{joint,min}-\underline{\tau}^{i-1}_{aux}+\mathbf{M}^{i-1}\tfrac{\Delta q^{i-1}}{\Delta T^2} \tag{12}$$

$$-\frac{1}{\Delta T^2}\mathbf{M}^{i-1}\Delta\underline{q}^i \geq \quad -\underline{\tau}_{joint,max}+\underline{\tau}^{i-1}_{aux}-\mathbf{M}^{i-1}\tfrac{\Delta q^{i-1}}{\Delta T^2} \tag{13}$$

with

$$\mathbf{M}^{i-1}=\mathbf{M}(\underline{q}^{i-1}),\quad \underline{\tau}^{i-1}_{aux}=N(\underline{q}^{i-1},\dot{\underline{q}}^{i-1})+G(\underline{q}^{i-1}). \tag{14}$$

B. Joint acceleration and speed limits

In addition to the dynamic model, joint acceleration and speed limits are implemented in the robot control. Their definition

$$\dot{\underline{q}}_{min}\leq\dot{\underline{q}}\leq\dot{\underline{q}}_{max} \tag{15}$$

$$\ddot{\underline{q}}_{min}\leq\ddot{\underline{q}}\leq\ddot{\underline{q}}_{max} \tag{16}$$

can be directly reformulated using (11) to

$$\frac{1}{\Delta T}\mathbf{I}\Delta\underline{q}^i \geq \quad \dot{\underline{q}}_{min} \tag{17}$$

$$-\frac{1}{\Delta T}\mathbf{I}\Delta\underline{q}^i \geq \quad -\dot{\underline{q}}_{max} \tag{18}$$

$$\frac{1}{\Delta T^2}\mathbf{I}\Delta\underline{q}^i \geq \quad \ddot{\underline{q}}_{min}+\tfrac{\Delta q^{i-1}}{\Delta T^2} \tag{19}$$

$$-\frac{1}{\Delta T^2}\mathbf{I}\Delta\underline{q}^i \geq \quad -\ddot{\underline{q}}_{max}-\tfrac{\Delta q^{i-1}}{\Delta T^2} \tag{20}$$

C. Hardware stops

Unlike the previous constraints, the joint's hardware stops cannot be implemented in a straightforward way ($\underline{q}_{min}\leq\underline{q}\leq\underline{q}_{max}$): if a joint reaches its hardware stop with the speed $\Delta\underline{q}^{i-1}/\Delta T$, the acceleration to stop the joint in one sample step

$$\ddot{\underline{q}}_{stop}=\Delta\underline{q}^{i-1}/\Delta T^2, \tag{21}$$

can exeed the limits introduced with the dynamics model (10) and the acceleration limits (16). In this case, the solution space $\mathbb{L}$ would be empty.
To avoid inconsistent constraints, the braking acceleration must be within the limits defined by the dynamics and the acceleration limits of the robot. A possible approach, presented in [15], is to calculate the braking distance $s_{brake,j}$ of each joint j, assuming a constant deceleration $\ddot{q}_{brake,j}$:

$$s_{brake,j}=\left|\frac{\dot{q}_j^2}{2\ddot{q}_{brake,j}}\right| \tag{22}$$

Then, it is checked in every sample step, if the braking distance would exceed the remaining workspace of the joint:

$$\begin{aligned}\Delta\underline{q}_j^{i-1}>0 \quad\wedge\quad & q_j^{i-1}+s_{brake,j}\geq q_{max,j}\\ \Delta\underline{q}_j^{i-1}<0 \quad\wedge\quad & q_j^{i-1}-s_{brake,j}\leq q_{min,j},\end{aligned} \tag{23}$$

When this condition becomes true, an additional equality equation

$$\Delta\underline{q}_j\overset{!}{=}\begin{cases}-\ddot{q}_{brake,j}\Delta T^2+\Delta\underline{q}_j^{i-1} & \text{with } \dot{q}_j^{i-1}>0\\ \ddot{q}_{brake,j}\Delta T^2+\Delta\underline{q}_j^{i-1} & \text{with } \dot{q}_j^{i-1}<0\end{cases} \tag{24}$$

is activated, forcing a constant deceleration of the braking joint. (24) is deactivated, when (23) is no longer fulfilled and the joint has come to a full stop.
This approach is easy to implement, but because of the sampled evaluation of (23), the braking will result in an overshoot Δs_j. Between the last sample step without braking and the sampling step when condition (23) becomes true, the joint can move in the worst-case the distance

$$\Delta s_{max,j}=|\Delta\underline{q}^{i-1}|+\ddot{\underline{q}}_{max}\Delta T^2, \tag{25}$$

leading to an intolerable overshooting of the braking trajectory. This can be avoided by modifying the braking condition, consindering this worst-case overshoot:

$$\begin{aligned}\Delta\underline{q}_j^{i-1}>0 \quad\wedge\quad & q_j^{i-1}+s_{brake,j}+\Delta s_{max,j}\geq q_{max,j}\\ \Delta\underline{q}_j^{i-1}<0 \quad\wedge\quad & q_j^{i-1}-s_{brake,j}-\Delta s_{max,j}\leq q_{min,j},\end{aligned} \tag{26}$$

IV. CHOOSING THE OPTIMIZATION CRITERIA

While the solution space $\mathbb{L}$ is defined by the constraints of the optimization, the solution itself is calculated as result from the minimization of the optimization criterion $\Phi(..)$ from (6). The desired behaviour of the robot system is to follow the reference position and orientation from the washout filter as close as dynamically and mechanically possible, as described in (3). But the dynamic behaviour and the kinematics of the robot introduce some challenges to be handled with the formulation of the optimization problem.

A. Selecting an optimization method

For the implementation of (6), several optimization methods are suitable. In this paper, a least squares formulation

$$\min_{\Delta q} |\mathbf{A}\Delta\underline{q} - \underline{b}|^2$$
$$\text{subject to} \quad \mathbf{G}\Delta\underline{q}^i \geq \underline{h} \quad \mathbf{E}\Delta\underline{q}^i = \underline{f} \tag{27}$$

is chosen. Other formulations like quadratic or linear programming are possible. In these cases the solution of the optimization problem can be obtained within a fixed amount of time, as needed for a real-time sampled data system.
The following tasks have to be fulfilled:

- The position error defined in (3) has to be minimized to achieve the proper simulator cell movements.
- The nullspace movement of the robot has to be defined.

A complete description of the mathematical procedures for solving (27) would be far beyond the scope of this paper. The interested reader can find a description and an implementation of (27)in [16].

B. Feasible formulations of the positioning criterion

To realize the task definition (3), one could be tempted to implement this definition straightforward

$$\min_{\underline{q}} ||\underline{x}_{ref} - \underline{x}(\underline{q})|| \Rightarrow \min_{\Delta\underline{q}^i} |\mathbf{J}^{i-1}\Delta\underline{q}^i - (\underline{x}^i_{ref} - \underline{x}(\underline{q})^{i-1})|^2, \tag{28}$$

with $\mathbf{J}^{i-1} = \partial(\underline{f}(\underline{q}))/\partial(\underline{q})\,|_{t^{i-1}}$. But this approach results in large oscillations around the reference position $\underline{x}^i$, caused by the dynamic limitations of the system:
The implementation (28) at first forces the effector in the direction of x^i_{ref}. But without any predictive braking during the approach of the reference position, the velocity difference between actual and reference position has to be adapted solely in the last sample step. The acceleration necessary for this adaption must be smaller than the acceleration limit $\ddot{\underline{x}}^i_{max}$, introduced by the robot constraints to avoid overshooting:

$$(\Delta\underline{x}^i_{ref} - \Delta\underline{x}^{i-1})/\Delta T^2 \leq \ddot{\underline{x}}^i_{max} \tag{29}$$

The maximum acceleration $\ddot{\underline{x}}^i_{max}$ is limited by the maximum joint acceleration $\ddot{\underline{q}}_{max}$ and the maximum joint torques $\underline{\tau}_{joint,max}$ as described in (16) and (10). With the real-world motor torques, this joint acceleration can not be generated, so the joint will be decelerated in several sample steps after reaching the reference position. This causes overshooting, as the constraints of the dynamic model (10) will force the solution beyond the reference position.
There are several possibilities to realize a more damped tracking behaviour. [13] suggests to minimize not only the position error, but also the error between the first derivatives of the reference and the effector trajectory

$$\min_{\underline{q}} ||\underline{x}_{ref} - \underline{x}(\underline{q})|| + ||\dot{\underline{x}}_{ref} - \dot{\underline{x}}(\underline{q})||. \tag{30}$$

This works well, as long as the position error is not significant, but for large discrepancies in the position, the optimization criteria minimizing the position and the velocity error can become contradictory: If the velocity, necessary for minimizing the position error, is not equal to the first derivative of the reference trajectory, it leads to unwanted damping.
In [9] an additional discrete PID controller with the transfer function $F_{PID}(z)$ generating a velocity signal is used as input for a velocity optimization:

$$\Delta\underline{x}^i_{PID} = F_{PID}(z)(\underline{x}^{i-1} - \underline{x}^i_{ref}) \quad \Rightarrow \quad \min_{\Delta\underline{q}} |\mathbf{J}^{i-1}\Delta\underline{q} - \Delta\underline{x}^i_{PID}|^2 \tag{31}$$

Via the controller dynamics, the dynamics of the tracking behaviour of $\underline{x}^i$ can be parametrized and a predictive deceleration can be achieved to avoid overshooting.

C. Additional optimization criteria improving the null-space movement

By using the optimization criterion (31), the null-space movement is unspecified. As it only minimizes the positional error, it provides no information about the preferred null-space configuration of the redundant robot system. Furthermore, the null-space movement is undamped, so oscillations can occur (see Fig. 4). As the postion and orientation in space demand 6 DOF from the kinematics, an additional optimization criterion has to influence at least $n-6$ axes to define the complete configuration behaviour [13]. The application presented in this paper introduces a robot system with 7 DOF, so one axis has to be influenced by an additional optimization criterion. By damping the movement of the first rotational axis of the robot (axis 1), a sufficient null-space damping can be achieved. Therefore, the additional optimization criterion

$$\min_{\Delta\underline{q}} |\mathbf{K}_2\Delta\underline{q}|^2 \tag{32}$$

is introduced, with the weighting matrix $\mathbf{K}_2 = diag[1,0,0,0,0,0,0]$, minimizing the speed of axis 1, thus damping the null-space movement of the robot system. To obtain a preferred null-space configuration $\underline{q}_{ref}$, additonally a quadratic potential function

$$H(\underline{q}) = (\underline{q}_{ref} - \underline{q})\mathbf{K}_3(\underline{q}_{ref} - \underline{q}) \tag{33}$$

is minimized as presented in [17]

$$\min_{\Delta\underline{q}} |2\mathbf{K}_3(\underline{q}_{ref} - \underline{q}^{i-1})\Delta\underline{q} - (\underline{q}_{ref} - \underline{q}^{i-1})\mathbf{K}_3(\underline{q}_{ref} - \underline{q}^{i-1})|^2, \tag{34}$$

where $\mathbf{K}_3$ is the weighting matrix emphasizing the axes to be influenced. Fig. 4 shows the effect of the two additional optimization criteria.

D. Collecting and weighting the optimization criteria

So far various optimization criteria have been formulated. To integrate them into the least squares framework (27), these criteria are weighted with the diagonal matrices $\mathbf{K}_{1,2,3}$ and collected to the optimization problem

$$\min_{\Delta\underline{q}} \Bigg| \underbrace{\begin{bmatrix} \mathbf{K}_1\mathbf{J}^{i-1} \\ \mathbf{K}_2 \\ 2\mathbf{K}_3(\underline{q}_{ref} - \underline{q}^{i-1}) \end{bmatrix}}_{\mathbf{A}} \Delta\underline{q}^i - \underbrace{\begin{bmatrix} \mathbf{K}_1(\Delta\underline{x}^i_{PID}) \\ \underline{0} \\ (\underline{q}_{ref} - \underline{q}^{i-1})\mathbf{K}_3(\underline{q}_{ref} - \underline{q}^{i-1}) \end{bmatrix}}_{\underline{b}} \Bigg|^2. \tag{35}$$

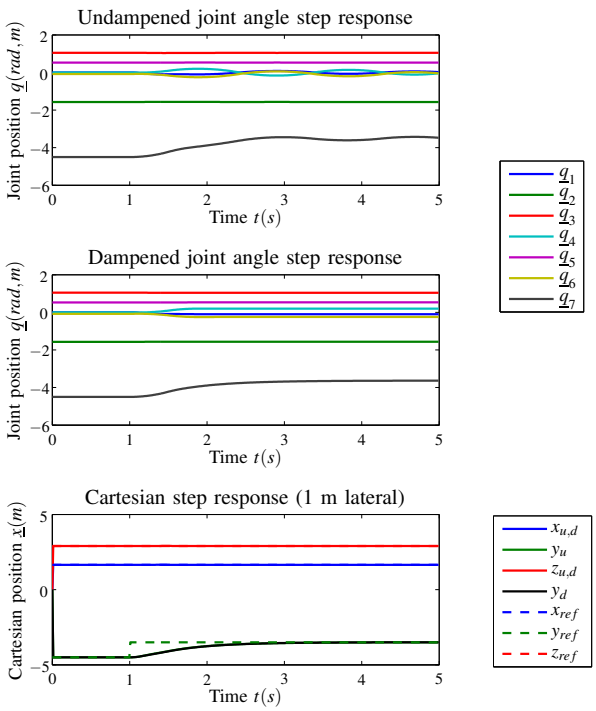

Fig. 4. Step response on a lateral Cartesian position step. The upper diagram shows the undampened null-space movement of the axes $\underline{q}$, caused by an incomplete null-space task definition. The middle diagram shows the effect of an additional optimization criterion, influencing the null-space behaviour. In the lowest diagram, the Cartesian position is visible, with both methods plotted in the same graph (Nearly no difference, undampened(u), dampened(d)).

The values of $\mathbf{K}_{1,2,3}$ define the impact of the according optimization criterion on the result. The parametrized magnitude of $\mathbf{K}_{2,3}$ is normally significantly smaller than the magnitude of $\mathbf{K}_1$ to avoid unnecessary damping of the positioning behaviour.

V. MODIFIED WASHOUT FILTER

Like most washout filter designs the implemented washout filter is based on the classical washout filter developed by Reid and Nahon [2]. Nevertheless, the path-planning method presented in this paper, as well as the application, require some modifications to the standard washout filter approach. The optimization based path-planning relies on the linear interpolation between two positions and orientations $\underline{x}_1$ and $\underline{x}_2$, but with euler angles used in most washout filters, this linear interpolation can be troublesome.

Furthermore, the application as a flight simulator makes it necessary for the washout filter to handle negative G-Forces properly, thus no unwanted movements are introduced by the handling of these forces. Fig. 5 gives an overview of the filter structure.

The specific force f experienced in the simulator or vehicle is the comibination of the gravity and the accelerations of the simulator or vehicle itself, as percepted from within the simulator / vehicle:

$$\underline{f}_{sim} = -\underline{a}_{sim} + {}^{sim}\mathbf{T}_0\underline{g} \quad (36)$$

$$\underline{f}_{pilot} = -\underline{a}_{pilot} + {}^{pilot}\mathbf{T}_0\underline{g} \quad (37)$$

A. Translational channel

This channel generates the Cartesian reference position of the simulator. As the basic principles of washout filtering are well documented in literature, in-depth explanations about the translational channel are omitted for brevity. The accelerations of the pilot $\underline{a}_{pilot}$ are scaled, transformed in the inertial coordinate system of the simulator and limited using a nonlinear, sigmoid limiter. This limited acceleration signal is now filtered with a high-pass filter to obtain the high-frequent accelerations $\underline{a}_{sim}$, displayed by the simulator mechanics. The low-frequent accelerations are handled by the tilt coordination, using the gravity as a source for low-frequent acceleration.

B. Why using quaternions as orientation description

Until this point, the orientation of the cell has been described as euler angles $\underline{\varphi} = (\alpha, \beta, \gamma)$ for clarity. But the usage of Euler angles introduces difficulties in the optimization of the orientation, due to the fact, that a linear interpolation between euler angles does not lead to the desired direct reorientation. As the optimization process of (3) leads to a linear transition between the actual orientation and the reference orientation, this problem applies also to this approach. Unlike linear interpolation between two vectors in $\mathbb{R}^3$ the three axes components (α, β, γ) are not decoupled during an ongoing rotation. So a linear interpolation over the Euler angles will result in a 'tumbling' reorientation behaviour, and not necessarily in the topological direct transition between two orientations. (E.g. thinking of two orientations $\underline{\varphi}_{cell} = (10°, 10°, 10°)$ and $\underline{\varphi}_{ref} = (350°, 350°, 350°)$ the approach presented above using euler angles would lead to a cell movement going all the way from 10° to 350° instead of moving in a direct rotation from 10° over 0° to 350°). By using quaternions as orientation representation, these problems can be avoided, as (spherical linear) interpolations between quaternions describe the direct rotation from one orientation to another. A quaternion $\mathbf{Q} \in \mathbb{R}^4$ is defined as

$$\mathbf{Q} = [w, (x, y, z)] = [\cos\frac{\Theta}{2}, (\underline{n} \cdot \sin\frac{\Theta}{2})] \quad (38)$$

where $\underline{n} \in \mathbb{R}^3$ is a unity vector describing the rotational axis for a rotation about the angle Θ. [18][19]

C. Quaternion tilt coordination

The tilt coordination of washout filters has the purpose to reorientate the cell, so the low-frequent accelerations felt by the pilot of a real vehicle are displayed by the gravity, felt by the simulator pilot. The classical washout filter design from [2] uses the following way to calculate the necessary cell orientation for this task:

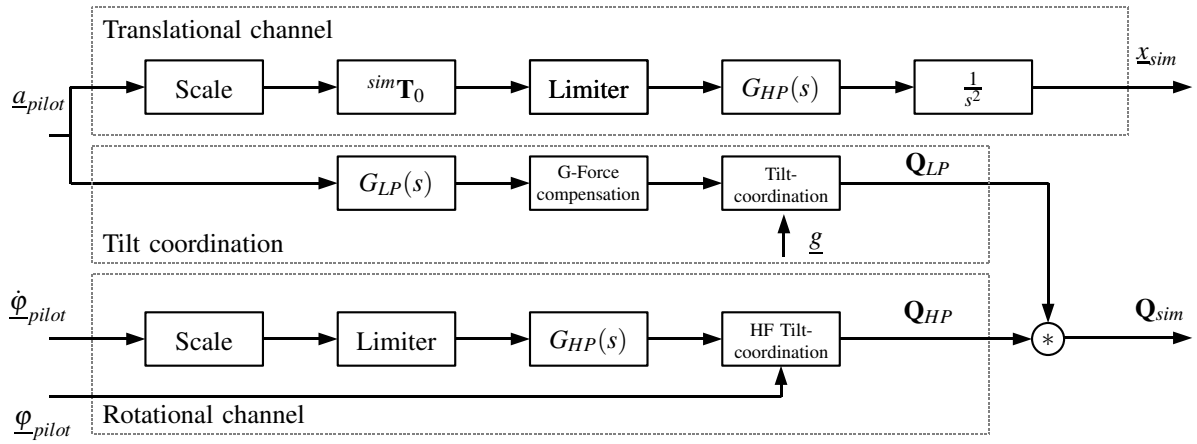

Fig. 5. Washout filter utilizing Quaternions for orientation representation. The tilt coordinations are implemented as quaternions, allowing continous interpolations between orientations. Additionally a G-force compensation is implemented to avoid the display of negative G-forces with the tilt-coordination.

Deriving from the equations

$$\underline{f}_{pilot}^{LP} = -\underline{a}_{pilot}^{LP} + {}^{pilot}\mathbf{T}_0\underline{g} \tag{39}$$

$$\frac{\underline{f}_{pilot}^{LP}}{|\underline{f}_{pilot}^{LP}|} = \mathbf{R}_x(\alpha)\mathbf{R}_y(\beta)\,[0,0,1]$$

$$= \quad [sin(\beta), -sin(\alpha)cos(\beta), -cos(\alpha)cos(\beta), 0] \tag{40}$$

the orientation $\underline{\varphi}_{LP}$, aligning the low-pass filtered specific force felt by the pilot of the real aircraft $\underline{f}_{pilot}^{LP}$ and the gravity felt by the simulator pilot, can be formulated as

$$\underline{\varphi}_{LP} = [atan2(-n_2, n_3), asin(n_1), 0]\,. \tag{41}$$

This approach works well with Stewart-platform based simulators, as $|\beta|$ never can be $> 90°$.
For a simulator mechanics capable of Cartesian orientations with $|\beta| > 90°$, this approach fails because the domain of the *asin* function is limited to $\pm 90°$. One possible solution is modifying the upper approach by replacing the *asin* expression using the term

$$\underline{\varphi}_{LP} = [\underbrace{atan2(-n_2, n_3)}_{\alpha}, atan2(n_1, n_3/cos(\alpha)), 0]. \tag{42}$$

Another approach is to use a quaternion based tilt coordination. To tilt the simulator cell, the rotation

$$\mathbf{Q}_{tilt} = [\cos\frac{\Theta}{2}, (\underline{n}\cdot\sin\frac{\Theta}{2})] \tag{43}$$

from the simulator's inertial system to the orientation where f_{pilot}^{LP} and $\underline{g}$ are aligned is calculated with

$$\underline{n} = \frac{\underline{g}\times\underline{f}_{pilot}^{LP}}{\left|\underline{g}\times\underline{f}_{pilot}^{LP}\right|} \text{ and } \theta = \arccos\frac{\underline{g}\cdot\underline{f}_{pilot}^{LP}}{|\underline{g}|\cdot\left|\underline{f}_{pilot}^{LP}\right|}. \tag{44}$$

As this rotation does not define the orientation in space completly (Every rotation aligning f_{pilot}^{LP} and $\underline{g}$ is a possible result, rotation around $\underline{g}$ is an unused degree of freedom), an additional correcting rotation can be applied to define the direction the simulator cell is facing:

Using the conversion matrix from a quaternion $\mathbf{Q}$ to a orientation matrix $\mathbf{R}$

$$\mathbf{R}(\mathbf{Q}) = \begin{bmatrix} w^2+x^2-y^2-z^2 & -2wz+2xy & 2wy+2xz \\ 2wz+2xy & w^2-x^2+y^2-z^2 & -2wx+2yz \\ -2wy+2xz & 2wx+2yz & w^2-x^2-y^2+z^2 \end{bmatrix}, \tag{45}$$

the correcting orientation $\mathbf{Q}_{corr}$ is defined as

$$\mathbf{Q}_{corr} = [\cos\frac{\sigma}{2}, (\underline{g}\cdot\sin\frac{\sigma}{2})] \tag{46}$$

with

$$\sigma = atan2(\mathbf{R}_{11}, \mathbf{R}_{12}) \tag{47}$$

This quaternion rotates the tilted cell so that the x-axis of the cell coordinate system faces the direction $[1,0,0]$. The complete rotation $\mathbf{Q}_{LP}$ is then the combination of the rotations $\mathbf{Q}_{tilt}$ followed by the correction of the cell view direction $\mathbf{Q}_{corr}$ using the definition for quaternion multiplication from [20]:

$$\mathbf{Q}_{LP} = \mathbf{Q}_{corr} * \mathbf{Q}_{tilt} \tag{48}$$

D. Rotational channel

The rotational channel simulates the angular velocities of the real vehicle. The vehicle orientation (as Euler angles) $\underline{\varphi}$ and its derivative $\dot{\underline{\varphi}}$ are the inputs to this channel. The last one, is scaled, limited and filtered by a high pass filter (similar to the translational channel) to obtain a signal containing the high frecuency componentes $\underline{\dot{\varphi}}_{pilot}^{HP}$, actually perceived by the human equilibrium sense. The tilt coordination block used in this channel also calculates a quaternion to define the orientation change resulting from the angular velocity, but with some differencies to the previous tilt coordination block. The angular velocity $\underline{\omega}_{pilot}^{HP}$ is obtained from $\underline{\varphi}_{pilot}$ and $\underline{\dot{\varphi}}_{pilot}^{HP}$ as follows:

$$\underline{\omega}_{pilot}^{HP} = [0,0,\dot{\varphi}_{pilot,z}^{HP}] + \mathbf{T}_z(\varphi_3)\cdot[0,\dot{\varphi}_{pilot,y}^{HP},0] + \mathbf{T}_z(\varphi_3)\cdot\mathbf{T}_y(\varphi_2)\cdot[\dot{\varphi}_{pilot,x}^{HP},0,0] \tag{49}$$

Where $\mathbf{T}_{x,y,z}$ is the transformation matrix, which describes the rotation around the x,y,z-axis.

Once $\underline{\omega}_{pilot}^{HP}$ is obtained, it is possible to implement the desired orientation change as quaternion Q_{HP}:

$$\mathbf{Q}_{HP} = \quad [\cos\tfrac{\rho}{2}, (\underline{s}\cdot\sin\tfrac{\rho}{2})] \tag{50}$$

with

$$\underline{s} = \frac{\int \underline{\omega}_{HP}\,dt}{\max(\varepsilon, |\int \underline{\omega}_{HP}\,dt|)} \qquad \rho = \left|\int \underline{\omega}_{HP}\,dt\right| \tag{51}$$

The reference orientation of the cell is then calculated by applying the rotations $\mathbf{Q}_{HP}, \mathbf{Q}_{LP}$ in the order

$$\mathbf{Q}_{ref} = \mathbf{Q}_{HP} * \mathbf{Q}_{LP}. \tag{52}$$

E. Compensation of negative G-forces

During simulation, especially flight simulations, negative G-forces can occur, for example during a rapid declining maneuver. This means, a low-frequent force acting upwards from the pilot's view has to be displayed by the simulator, but this is only possible by turing the simulator cell upside-down. While this can be a reasonable behaviour during maneuvers like rolls or loops, the turning of the cell massivly irritates the pilot during a decline by introducing a non existent rotational motion cue. So it seems appropriate to neglect negative G-forces induced by the movement of the vehicle and keeping the cell upright in order to maintain a consistent perception between the visual- and the motion cues. To achieve this goal, the low-passed filtered acceleration of the real-world pilot $\underline{a}_{pilot}^{LP}$ has to be downscaled, if they are about to compensate the gravitational forces in the real-world vehicle. Let be

$$k = (-\underline{a}_{pilot}^{LP} \cdot {}^{pilot}\mathbf{T}_0 \underline{g})/|\underline{g}|^2 \tag{53}$$

the percentage share of the acceleration projected on the gravitational force as felt in the real vehicle. Then the acceleration used for the tilt-coordination can be limited with the term

$$\underline{a}_{limited}^{LP} = \begin{cases} \underline{a}_{pilot}^{LP} & \text{for } k < \eta \\ \underline{a}_{pilot}^{LP} \cdot \eta/k & \text{for } k \geq \eta \end{cases}, \tag{54}$$

with $\eta \in [0,1]$ being the threshold for the maximum percentage of $\underline{a}_{pilot}^{LP}$ from $\underline{g}$.

VI. APPLICATION: ARTISTIC FLIGHT SIMULATION

As a first application for the new motion simulator system, an artistic flight simulation has been chosen. The possibilities of the workspace and the path-planning can be demonstrated effectively by an airplane flying maneuvers as over-head flights and other artistic flight maneuvers. For all tests, the following parameters have been used:
Washout-filter: Edge frequency of high-pass and low-pass filter: $\omega_{HP} = 1.5 rad/s$, acceleration scaling: $K_{acc} = 1$. Optimization: Parameters of PID controller: $K_P = 8, K_D = 2, K_I = 0.01$.
First of all, Fig. 6 shows the step response of the washout Filter and the optimization. The differences between the washout filter, calculating the reference trajectory for the simulator cell and the actual movement of the simulator are caused by the consideration of the dynamic constraints of the robot system.

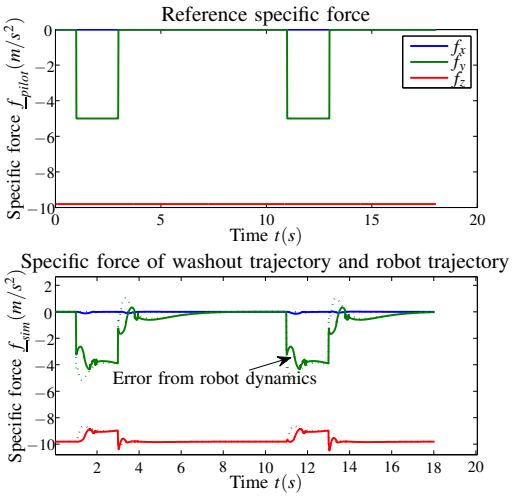

Fig. 6. Lateral step response (y-direction) of the path-planning algorithm: The upper diagram shows the reference specific force, while the lower diagram shows the response of both the washout filter (dotted line) and the robot.

As a first flight experiment, a dive maneuver is performed to show the effect of the Negative-G compensation. In Fig. 7, the result of the maneuver is shown without and with compensation of negative G-forces as described in section V-E. Please note the unwanted behaviour in the uncompensated case, as the simulator cell rolls to an over-head position within one second. As the airplane makes no roll movement at all during the maneuver, this is a very irritating behaviour for the pilot. In the compensated case, the cell stays upright and only performs the pitching movements of the dive maneuver.
The next experiment shows the result of an overhead flight (Fig. 8 and Fig. 1). The specific forces acting on the airplane are properly reproduced by the simulator, as the simulator cell follows the roll maneuver. This time, the negative G-forces are caused by the overhead flight and not by a dive maneuver. The simulator can reproduce these forces, as they are gravitational forces introduced by the rolling of the aircraft into an over-head flight. To get a better impression of the simulator's abilities, this paper includes a short video, demonstrating several interactive flight maneuvers not possible with Stewart-platforms or standard path-planning algorithms.

VII. CONCLUSIONS AND FUTURE WORKS

In this paper, an interactive path-planning algorithm for kinematically redundant motion simulators has been presented and validated. The special challenges, introduced by the serial configuration of the mechanics, such as singularities in the workspace and a varying dynamic behaviour have been defined and handled by an optimization based approach for trajectory generation. Together with modifications of the classical washout filter design, a path-planning framework has been created able to handle the large, but complicated workspace of an anthropomorphic robot arm, based on an

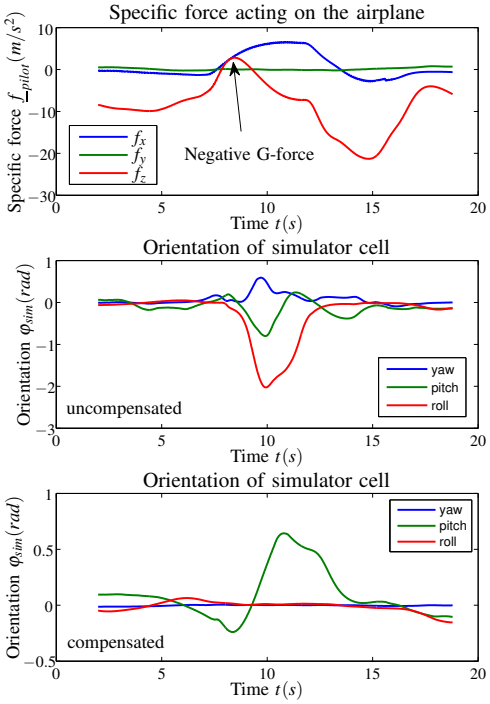

Fig. 7. Effect of negative G-force compensation. Upper diagram: The specific force acting on the aircraft during a dive maneuver, creating negative G-Forces. Middle diagram: Without compensation of the negative G-Forces the cell rolls as fast as possible to an overhead position to simulate the negative G-forces of the dive maneuver. Lower diagram: With compensation of those forces, the cell stays upright during the dive maneuver and only performs the according pitching movements

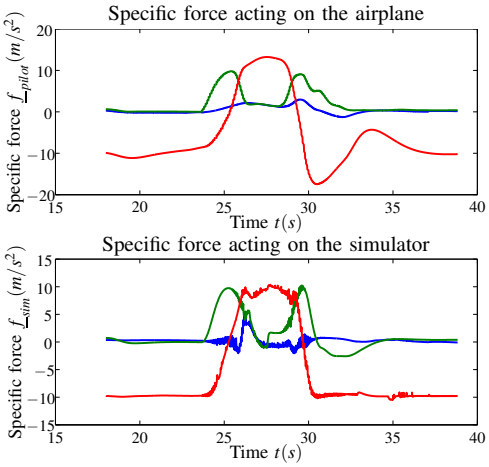

Fig. 8. Overhead flight: The airplane rolls in an over-head flight state at t=25s. The simulator follows this movement also performing a roll maneuver and reproducing the specific force felt in the airplane properly.

additional linear axis.

To show the effectiveness of the path-planning algorithm, several non-standard flight-maneuvers have been performed and their simulated results have been discussed. The attached video shows the DLR Robot Motion simulator during operation, performing interactive artistic flight maneuvers. As this is the first time a kinematically redundant robot system is used for the task of motion simulation, there are still plenty of points to be analyzed and improved. In the next steps, an automatic workspace fitting for the washout filter is planned, as well as further improvement of the optimization strategies and adaption to the special needs of driving simulations.

VIII. ACKNOWLEDGMENTS

The authors would like to thank KUKA Roboter GmbH for funding and supporting this work.

REFERENCES

[1] D. Stewart. A platform with six degrees of freedom (Platform with six degrees of freedom for flight simulation in pilot training). *Institution of mechanical engineers*, 180(15):371–378, 1965-1966.

[2] LD Reid and MA Nahon. Flight simulation motion-base drive algorithms. Part 1: Developing and testing the equations. 1985.

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

[4] R. Hosman, S. Advani, and N. Haeck. Integrated design of flight simulator motion cueing systems. In *Presented at the Royal Aeronautical Society Conference on Flight Simulation. London*, 2002.

[5] F. Barbagli, D. Ferrazzin, C.A. Avizzano, M. Bergamasco, S.S.S. Anna, and I. Pisa. Washout filter design for a motorcycle simulator. *IEEE Virtual Reality, 2001. Proceedings*, pages 225–232, 2001.

[6] S. Schaetzle, C. Preusche, and G. Hirzinger. Workspace optimization of the Robocoaster used as a motion simulator. In *Proceedings of the 14th IASTED. International Conference on Robotics and Applications*, volume 664, page 470, Cambridge, USA, 2009.

[7] T. Bellmann, M. Otter, J. Heindl, and G. Hirzinger. Real-time path planning for an interactive and industrial robot-based motion simulator. In *Proc. of the 2nd Motion Simulator Conference*, Braunschweig, 2007.

[8] L. Pollini, M. Innocenti, and A. Petrone. Novel motion platform for flight simulators using an anthropomorphic robot. *Journal of Aerospace Computing, Information, and Communication*, 5, 2008.

[9] T. Bellmann. An Innovative Driving Simulator: Robocoaster. In *Proc. of FISITA, World Automotive Congress*, Munich, 2008.

[10] P.R. Giordano, C. Masone, J. Tesch, M. Breidt, L. Pollini, and HH Bülthoff. A Novel Framework for Closed-Loop Robotic Motion Simulation-Part I: Inverse Kinematics Design. In *Proc. of 2010 IEEE International Conference on Robotics and Automation*, 2010.

[11] G. Schreiber, M. Otter, and G. Hirzinger. Solving the singularity problem of non-redundant manipulators by constraint optimization. In *Proc. of International Conference on Intelligent Robots and Systems*, pages 1482–1488, Kyongju, Korea, 1999.

[12] E.S. Conkur and R. Buckingham. Clarifying the definition of redundancy as used in robotics. *Robotica*, 15(5):586, 1997.

[13] G. Schreiber. *Steuerung für redundante Robotersysteme: Benutzer-und aufgabenorientierte Verwendung der Redundanz*. PhD thesis, Chair for mechanical engineering, University Stuttgart, 2004.

[14] J.R. Taylor. *Classical mechanics*. Univ Science Books, 2005.

[15] T. Bellmann. Echtzeitbahnplanung für einen innovativen, roboterbasierten, interaktiven Bewegungssimulator. Master's thesis, Lehrstuhl für elektrische Antriebssysteme, TU Munich, 2007.

[16] C.L. Lawson and R.J. Hanson. *Solving least squares problems*. Prentice-Hall, 1974.

[17] A. De Luca, L. Lanari, and G. Oriolo. Control of redundant robots on cyclic trajectories. In *1992 IEEE International Conference on Robotics and Automation, 1992. Proceedings.*, pages 500–506, 1992.

[18] K. Shoemake. Animating rotation with quaternion curves. *ACM SIGGRAPH computer graphics*, 19(3):245–254, 1985.

[19] E.B. Dam, M. Koch, and M. Lillholm. Quaternions, interpolation and animation. 1998.

[20] W.R. Hamilton. LXXVIII. On quaternions; or on a new system of imaginaries in Algebra. *Philosophical Magazine Series 3*, 25(169): 489–495, 1844.