Real-Time Path Planning for an interactive and industrial robot-based Motion Simulator

Tobias Bellmann, Martin Otter, Johann Heindl, Gerhard Hirzinger

Institute of Robotics and Mechatronics, German Aerospace Center (DLR)

Münchner Straße 20, 82234 Weßling, Telefon: +49 8153 28-0, Fax: +49 8153 28-2243

Abstract

A new path-planning algorithm for a motion simulator based on an industrial robot is introduced and compared with the classical washout filter. The algorithm calculates the path of the simulator cell via the solution of an optimization problem with constraints like the workspace of the simulator or maximum joint angle accelerations. Several test results are presented in order to validate the new algorithm. A calculated trajectory is applied to the target robot in order to compare the simulated accelerations with the measured accelerations of the simulator cell.

1 Introduction

As presented on the First Motion Simulator Conference 2005 [1], an industrial robot (in this case the KUKA Robocoaster) equipped with seats and a visualizing system can be used as a platform for a Motion-Simulator. The motivation behind this combination results from the about ten times lower costs, caused by the mass-production of industrial robots. Furthermore the workspace of an industrial robot is considerably greater than the workspace of a hexapod-based motion-simulator, allowing maneuvers like rollers or loops.

The disadvantages of such a simulator are caused by the properties of the serial configuration of the industrial robot. The maximum acceleration and movability of the simulator cell depend on the actual configuration of the robot, singularities of the kinematic can prevent certain trajectories. These problems especially complicate the path-planning in cartesian space with classical washout filters.

In order to deal with these problems, a new method for path-planning has been developed, based on the solution of a multi-objective optimization.

This article first introduces the classical approach to motion-simulator path-planning, the

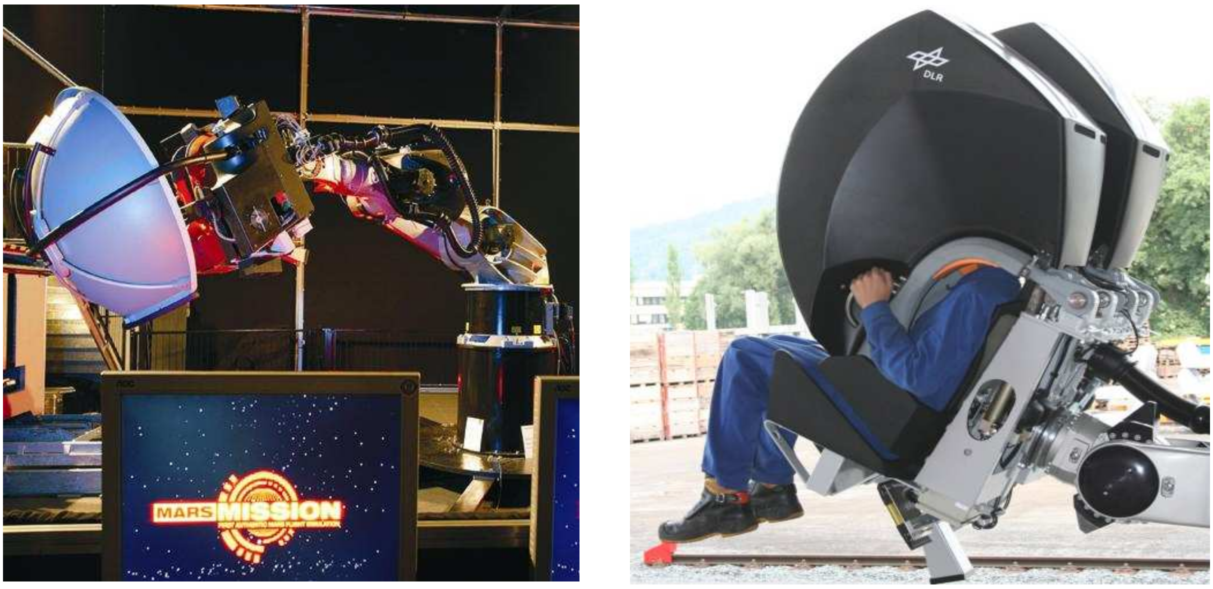


Figure 1: Robocoaster equipped with a spherical projection screen (left) and with two retractable caps, containing LCD screens and sound systems (right), source: DLR

washout filter, showing the difficulties resulting from the use of a serial kinematic. The following sections are an introduction to the new, optimization-based path-planning method, its parameterization and an analysis of the workspace and suitable inital positions. Selected simulation and experimental results demonstrate the usage of the path-planning algorithm.

2 Classical Washout Filter for Hexapod based Motion Simulators

The classical washout filter (WOF) is the standard path-planning algorithm for Hexapod Motion Simulators [4].

2.1 Functionality of the classical washout filter

The purpose of the classical washout filter is to transform the real-world trajectory of the simulated airplane or car to the simulator trajectory, which has to suit the limited workspace of the simulator. The desired result can be achieved by splitting the input acceleration into a high-pass-filtered and a low-pass-filtered component. The low-pass component of the acceleration is displayed by tilting the simulator cell relative to the gravitation vector, in order to avoid continuous movements, which would quickly exceed the simulator workspace.

The high-pass component of the input acceleration can be directly displayed via movement of the simulator cell. The correct implementation and parameterization of the high-pass Filter assures that the physical limits of the simulator will not be exceeded.

Angular velocities of the simulated trajectory also are high-pass filtered and therefore only suggested. The human rate perception organ lacks the ability to measure absolute angular velocities (only angular accelerations can be percepted and are integrated in the nerve-system), therefore it is sufficient to only begin a rotation, which will then be slowly decelerated. The perception of the rotation continues (supported by the display of visual motion cues) until a contrary rotation is initiated.

The components of the classical washout filter are displayed in figure 2 - 4.

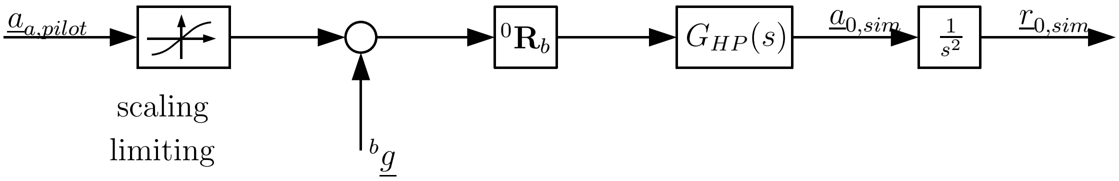


Figure 2: Signal-channel for dynamic (high-pass) accelerations

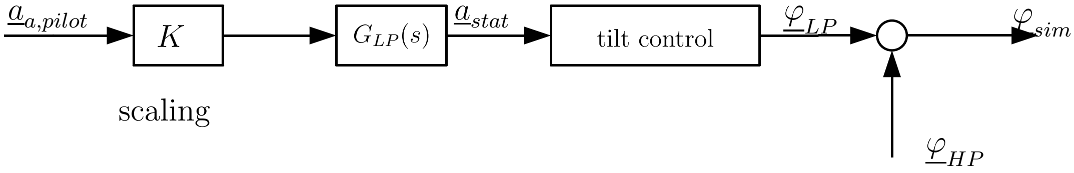

$\underline{a}_{a,pilot}$
K
scaling
$G_{LP}(s)$
$\underline{a}_{stat}$
tilt control
$\underline{\varphi}_{LP}$
$\underline{\varphi}_{sim}$
$\underline{\varphi}_{HP}$

Figure 3: Signal-channel for static (low-pass) accelerations

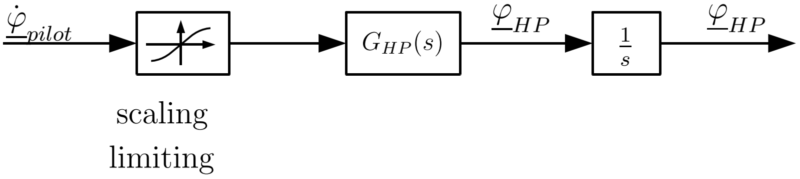

$\dot{\underline{\varphi}}_{pilot}$
scaling
limiting
$G_{HP}(s)$
$\dot{\underline{\varphi}}_{HP}$
$\frac{1}{s}$
$\underline{\varphi}_{HP}$

Figure 4: Signal-channel for angular rates

2.2 The applicability of the WOF with serial configurations

As mentioned before, the use of a serial configuration introduces some problems concerning path-planning. As the complete path is calculated in cartesian space, limitations of the robot joints like maximum and minimum joint angles or joint angle velocities can not be considered. Furthermore, the movability of the robot depends on its actual configuration. Figure 5 shows a three axis kinematic positioned in a singular configuration, meaning the degrees of freedom of the kinematic are reduced. Close to singular configurations, very high joint angular velocities are necessary to maintain the desired trajectory. With a path-planning algorithm only considering constraints in the cartesian space it can not be guaranteed that singularities will be avoided. If the planned trajectory reaches the vicinity of a singularity, the limitations of the drive speeds lead to path errors and cause a fail-save shutdown by the robot control.

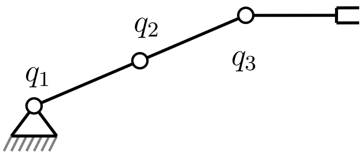


Figure 5: Example for a singular configuration

This lack of robustness of the cartesian path-planning with serial configurations disqualifies the WOF as a suitable path-planning algorithm.

3 Optimization based trajectory generation for the Robocoaster

The problems mentioned in the last section can be solved by considering the restrictions of the serial kinematic during the path-planning. This can be achieved by calculating the inverse kinematic in every calculation step, testing the feasibility of the desired path and altering the path if necessary. This method, however, is quite elaborate due to the complicated calculation of the inverse kinematic.

A new method for planning the simulators trajectory, based on the solution of an optimization problem regarding all restrictions of the kinematic, shall be introduced here.

3.1 The formulation of the optimization problem

The optimization problem consists of several subproblems performing the different tasks of path-planning. Similar to the WOF, the input signals of the optimization problem are the high- and low-pass-filtered reference accelerations of the simulated aircraft/vehicle. The subproblems of the optimization can be formulated like this:

1. The subproblem for dynamic accelerations: All acceleration which should be displayed by movement of the simulator cell, are calculated in the subproblem $\mathbf{SP}_{dyn}$:

$$\min_{\Delta \underline{q}^i} \left| \underline{\ddot{x}}_b(\underline{q}) - \underline{a}_{dyn} \right|^2 \qquad \text{with} \quad \underline{\ddot{x}}_b = \begin{bmatrix} \ddot{x}_b \\ \ddot{y}_b \\ \ddot{z}_b \end{bmatrix} \tag{1}$$

where $\underline{\ddot{x}}_b$ is the acc. from the simulator pilots view, $\underline{q}$ are the joint angles and $\underline{a}_{dyn}$ is the high-pass filtered input acceleration. This is a least squares optimization problem, stating that the difference between the high pass filtered pilot acceleration and the robot acceleration at the head of the pilot is as small as possible.

2. The subproblem for static accelerations: The subproblem $\mathbf{SP}_{stat}$ is responsible for the correct orientation ${}^b\mathbf{R}_0$ of the simulator cell, in order to simulate static, low-frequent accelerations $\underline{a}_{stat}$ via the gravitational acceleration $\underline{g}$.

$$\min_{\Delta \underline{q}^i} \left| {}^b\mathbf{R}_0(\underline{\varphi}(\underline{q}))\underline{g} - \underline{a}_{stat} \right|^2 \quad \text{with} \quad \underline{\varphi} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} (\mathrm{RPY-angles}) \tag{2}$$

Additionally, the angular velocity of the simulator cell must be damped in order to prevent the system from overshoots during re-orientations:

$$\min_{\Delta \underline{q}^i} \left| \Delta^b \underline{g}(\underline{\varphi}(\underline{q})) - g_3 \cdot \frac{\Delta \underline{a}_{stat}}{|\Delta \underline{a}_{stat}|} \right|^2 \quad \text{mit} \quad \underline{g} = \begin{bmatrix} 0 \\ 0 \\ g_3 \end{bmatrix} \tag{3}$$

3. The subproblem for angular velocities: The simulation of angular velocities can be calculated by the subproblem for angular velocities $\mathbf{SP}_{\omega}$.

$$\min_{\Delta \underline{q}^i} \left| \underline{\dot{\varphi}}(\underline{q}) - \underline{\dot{\varphi}}_{dyn} \right|^2 \tag{4}$$

With $\underline{\dot{\varphi}}_{dyn}$ being the desired, high-pass filtered input angular velocity. This OP minimizes the difference between the desired angular velocity $\underline{\dot{\varphi}}_{dyn}$ and the angular velocity of the simulator cell $\underline{\dot{\varphi}}$ by the adaption of $\Delta\underline{q}^i$.

Additional subproblems are formulated for the radial orientation of the robot and for the return to an initial position. The constraints of the optimization are the mechanical restrictions of the robot, as well as maximum and minimum joint angles, joint velocities and joint accelerations:

$$\underline{q}_{min} < \underline{q} < \underline{q}_{max} \tag{5}$$

$$\underline{\dot{q}}_{min} < \underline{\dot{q}} < \underline{\dot{q}}_{max} \tag{6}$$

$$\underline{\ddot{q}}_{min} < \underline{\ddot{q}} < \underline{\ddot{q}}_{max} \tag{7}$$

The single robot joints may not exceed their physical limits during the simulation of motions. Therefore, the maximum joint angle velocities and accelerations must be limited depending on the remaining angle reserve in the joint. In the vicinity of the angle limit the joints acceleration constraint is calculated considering the actual angle velocity, preventing the joint reaching its stop position. Additional constraints can be formulated to limit the maximum angular velocity of the cell, in order to achieve tilting the cell below the human perception limit.

$$\underline{\omega}^i < \underline{\omega}_{max} \tag{8}$$

The subproblems and the constraints can be reformulated into the standard form of a least squares optimization problem with constraints:

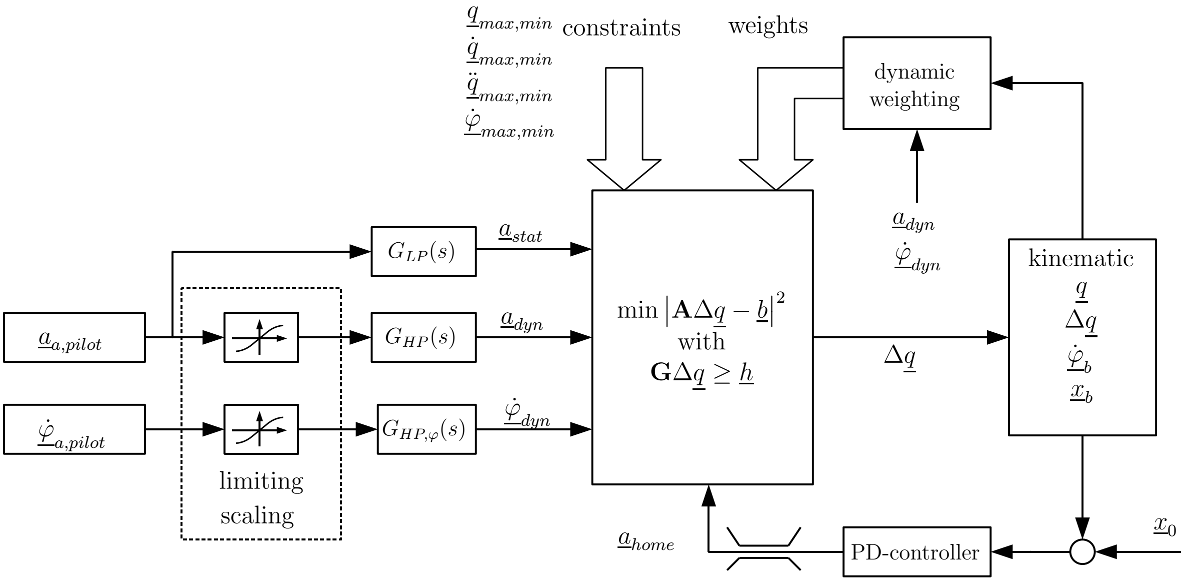
$$\min_{y} \left|\mathbf{A}\underline{y} - \underline{b}\right|^2 \quad \text{with constraints} \qquad \mathbf{G}\underline{y} \geq \underline{h} \tag{9}$$

Where $\underline{y}$ equals the joint angle increment vector $\Delta\underline{q}$. The individual subproblems can now be weighted by a factor α and combined into a single, multi-objective optimization problem:

$$\min_{y} \sum_{i} \alpha_i \left|\mathbf{A}_i\underline{y} - \underline{b_i}\right|^2 = \min_{y} \left\| \begin{bmatrix} \alpha_1\mathbf{A}_1 \\ \vdots \\ \alpha_n\mathbf{A}_n \end{bmatrix} \underline{y} - \begin{bmatrix} \alpha_1\underline{b}_1 \\ \vdots \\ \alpha_n\underline{b}_n \end{bmatrix} \right\|^2 \tag{10}$$

The solution of this optimization problem is the optimal change of the joint angles, taking into account the robots constraints.

4 Parameterization of Optimization Problem



$\underline{q}_{max,min}$
$\underline{\dot{q}}_{max,min}$
$\underline{\ddot{q}}_{max,min}$
$\underline{\dot{\varphi}}_{max,min}$
constraints
weights
dynamic weighting
$\underline{a}_{dyn}$
$\underline{\dot{\varphi}}_{dyn}$
kinematic
$\underline{q}$
$\Delta\underline{q}$
$\underline{\dot{\varphi}}_b$
$\underline{x}_b$
$G_{LP}(s)$
$\underline{a}_{stat}$
$\underline{a}_{a,pilot}$
$G_{HP}(s)$
$\underline{a}_{dyn}$
$\min\left|\mathbf{A}\Delta\underline{q}-\underline{b}\right|^2$
with
$\mathbf{G}\Delta\underline{q}\geq\underline{h}$
$\Delta\underline{q}$
$\underline{\dot{\varphi}}_{a,pilot}$
$G_{HP,\varphi}(s)$
$\underline{\dot{\varphi}}_{dyn}$
limiting
scaling
$\underline{a}_{home}$
PD-controller
$\underline{x}_0$

Figure 6: System overview, with its single components

The complete path-plannnig system depends on a multiplicity of parameters, which are either constant, or dynamically adapted from iteration to iteration.

4.1 The input filters

In case of ideal movability of the robot, the simulator cell exactly follows the output of the input filters G_{HP}, G_{LP}, $G_{HP,\omega}$. Hence the input filter parameters can be the same ones as those of the classical washout filter.

4.2 The dynamic weighting of the optimization problem

Several subproblems of the optimization interfere among each other. For example, the subproblem for the static acceleration $\mathbf{SP}_{stat}$ competes with the subproblem of angular velocity $\mathbf{SP}_{\omega}$ for the correct orientation. In order to improve the behaviour of the path-planning, the weightings α_{xxx} have to be calculated dynamically for every iteration. The weight function $\alpha_{xxx}(\eta)$ is defined as a s-form like function, which continuously switches between zero and the maximum value $\alpha_{xxx,max}$ of the specific weight (see fig. 7). In the case of the interference between $\mathbf{SP}_{stat}$ and $\mathbf{SP}_{\omega}$, η_{limit} would be the perception threshold for angular velocities. If the angular velocity to be simulated is greater than the perception threshold, the subproblem $\mathbf{SP}_{\omega}$ will be more weighted than the subproblem $\mathbf{SP}_{stat}$. On the contrary, if no angular velocities have to be displayed, the subproblem $\mathbf{SP}_{\omega}$ must not be weighted to avoid unnecessary damping of the cell orientation.

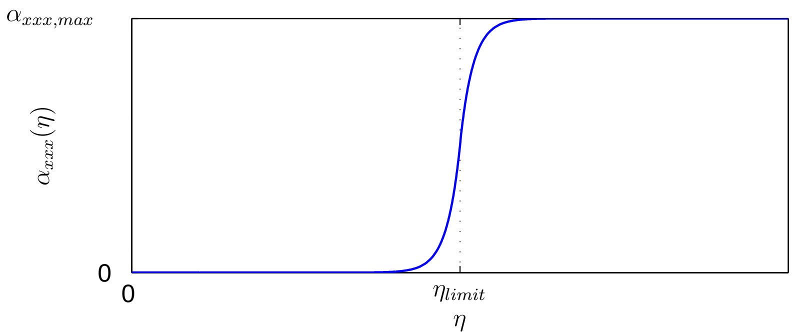


Figure 7: S-function for weighting the single subproblems

The same consideration applies to the other subproblems. In general, subproblems in the translational domain interfere among themselves, and rotational subproblems interfere with other rotational subproblems.

5 Workspace and Suitable Starting Configuration

The acceleration and movability of the simulator cell depend on the actual configuration of the robot. Therefore, the choice of the initial position of the robot has a high influence on the capabilities of the motion-simulation. In order to avoid singularities of the kinematic, which would reduce the movability of the robot, an initial position as shown in Figure 8 has been chosen. This position combines good acceleration values in all three dimensions with the possibility to tilt the cell around its three axis without changing the configuration.

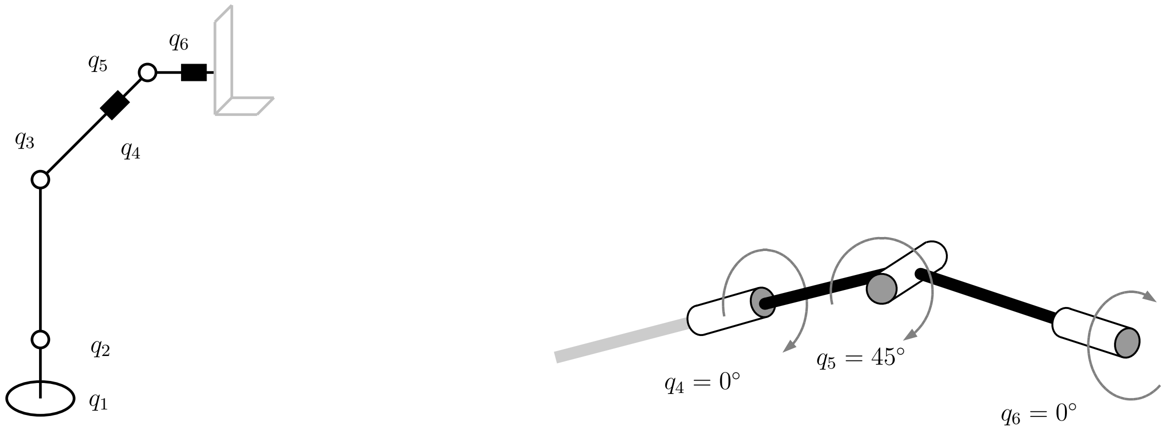


Figure 8: Configuration for improved movability of the simulator cell

The workspace of the used industrial robot is a spherical shell around the robot base. The serial configuration allows movements and static accelerations, which can not be reached by a classical hexapod motion-simulator. The possibility to approach nearly every orientation within the workspace allows i.e. overhead flights and the simulation of negative accelerations.

6 Selected Simulation and Experimental Results

Several simulations and robot tests has been performed in order to validate the correctness of the resulting simulator trajectories. The results will be compared with the results of the classical washout filter and criteria like the angle-error between the reference acceleration and the simulated acceleration will be considered.

6.1 Step-response of the path-planning

The first simulation determines the reaction of the path-planning to an acceleration step of 1 m/s^2 in x-, y-, z-direction. Figure 9 shows the acceleration to be simulated, in Figure 10 the response of the system and the acceleration angle-error are displayed. In figure 11, the good correlation between the results and the output of the compared washout-filter can be seen.

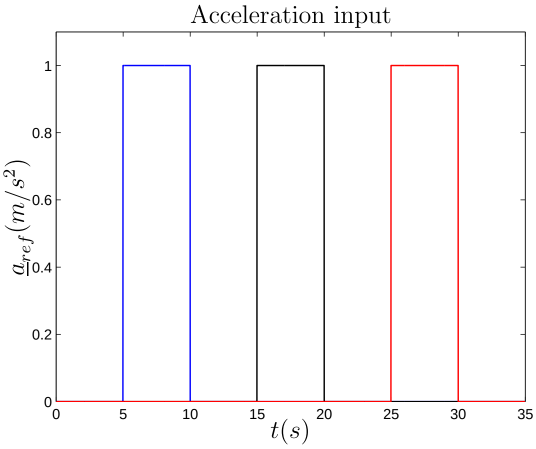


Figure 9: acc. step in x-, y-, z-direction (x: blue , y: black. , z: red)

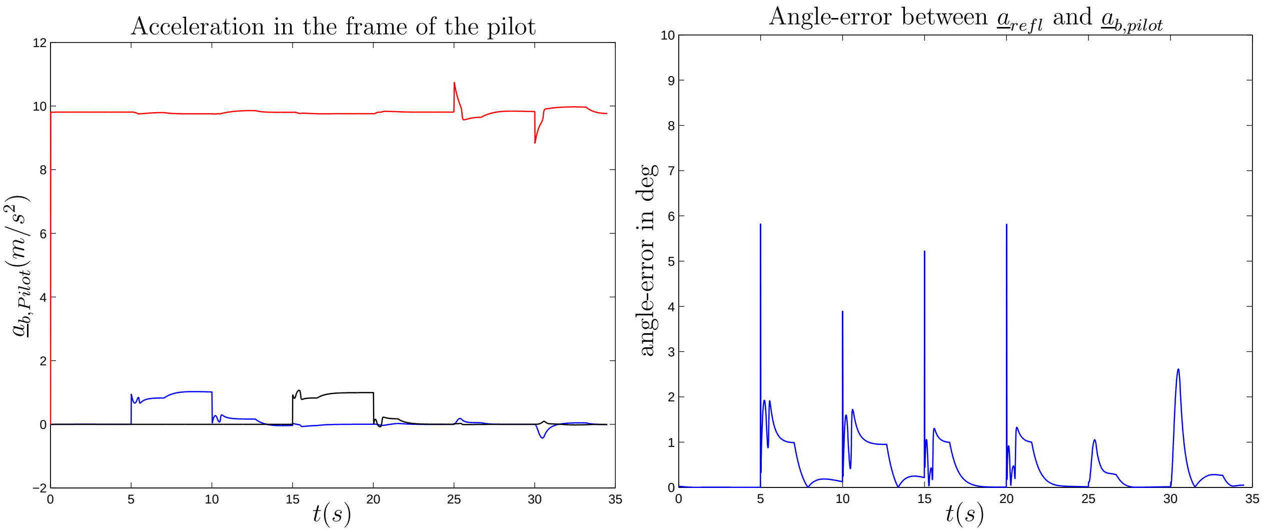


Figure 10: Acceleration in the frame of the pilot and angle error between reference acceleration and simulated acceleration

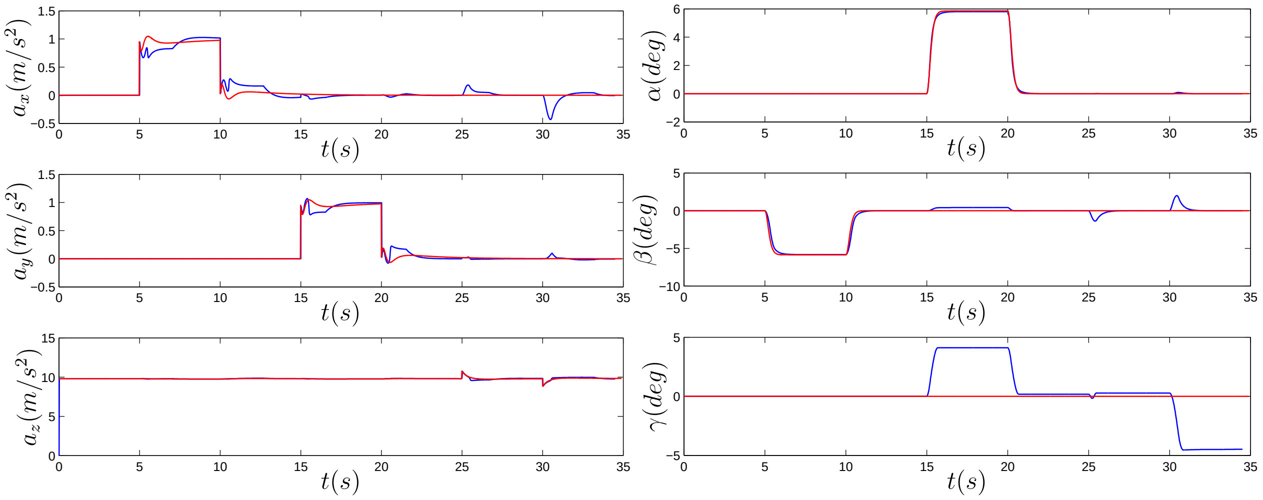


Figure 11: Comparison between washout-filter (red) and optimization problem (blue). left: acceleration, right: orientation

6.2 Aircraft simulation - inverted flying

One of the main advantages of the serial configuration over the hexapod configuration is the considerable greater workspace. To demonstrate the significant enhanced workspace of the robot based motion-simulator, a trajectory performing an inverse flight has been defined. The trajectory can be seen in figure 12. The motion simulator can follow the trajectory, generating the correct motion cues as seen in figure 13.

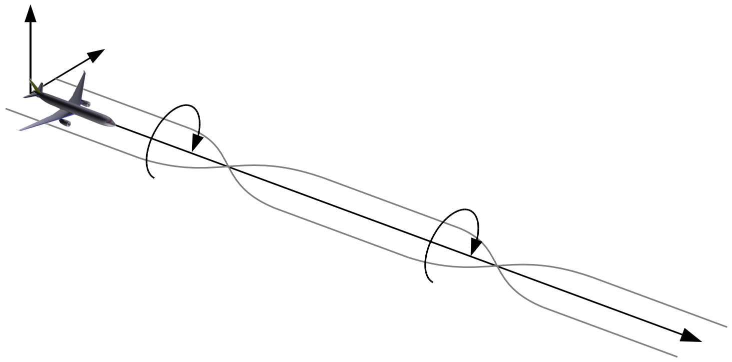


Figure 12: Trajectory: inverted flight

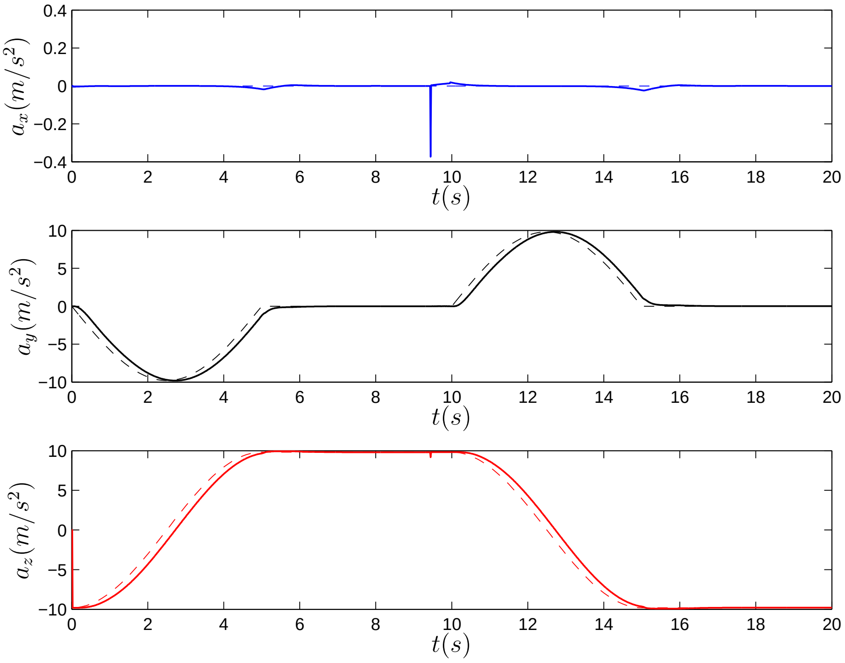


Figure 13: Acceleration from the pilots view

6.3 Car simulation - double bend

This trajectory, generated with the driving dynamic simulator SIMPACK (Intec GmbH, [3]), simulates a double bend as shown in Figure 14. The simulated run is divided into several phases. The vehicle accelerates during the first 7 seconds, then braking abruptly

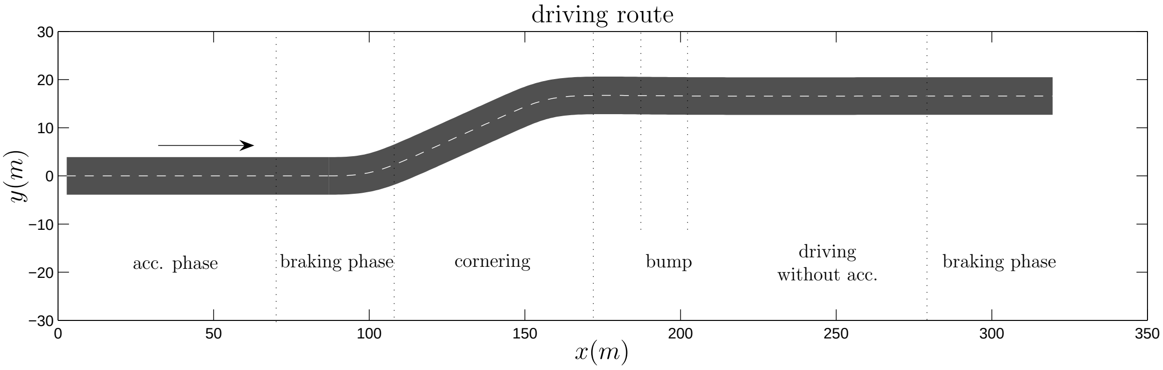


Figure 14: Reference trajectory and driving phases

before entering the double bend. After exiting the bend, the car drives straight ahead, passing a bump in the road. Finally, another braking phase follows. Figure 15 shows the reference accelerations and angular velocities. The resulting simulator accelerations can be seen in the left part of Figure 17. The missing motion cues in the y-acceleration can be explained with the display of angular velocities at that moment, inevitably resulting in changing the orientation of the cell according to the gravitational vector. Another constraint is visible when axis 5 reaches the end of its workspace, limiting the braking acceleration at 25 sec.

The simulated trajectory has been tested on the target platform, an industrial robot (KUKA KR500/1) equipped with three acceleration sensors. The results are shown in the right part of Figure 17, the measured accelerations correlating well with the simulated accelerations. The comparison between the washout filter and the new path-planning in Figure 18 shows the positive correspondence between their results.

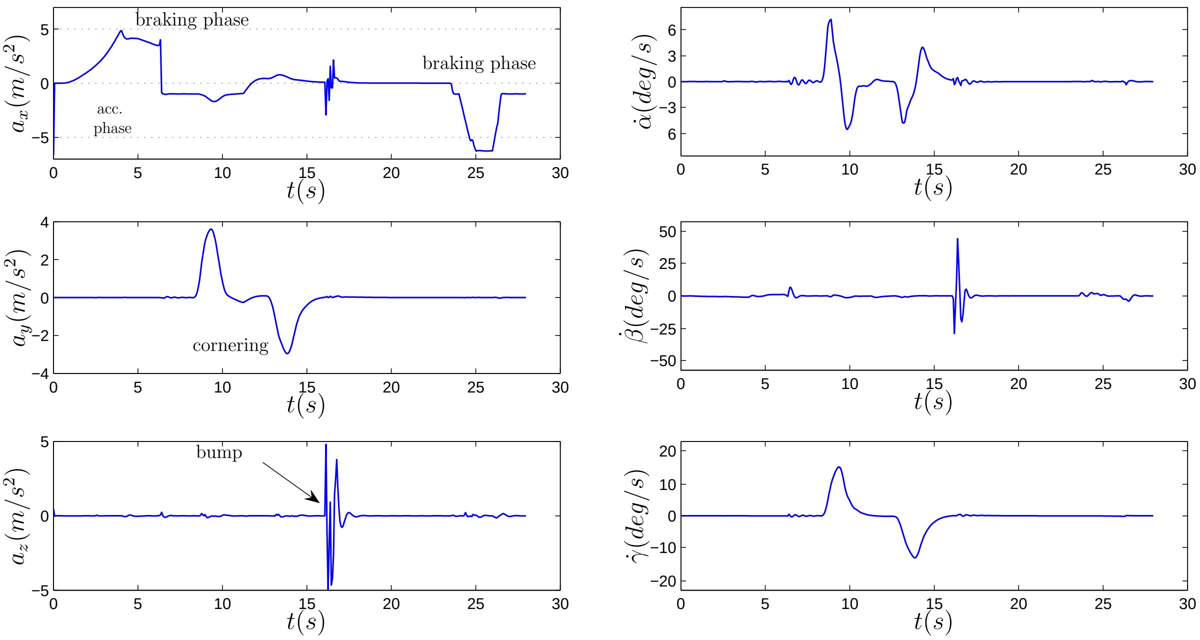


Figure 15: Reference acceleration and angular velocities of the vehicle trajectory

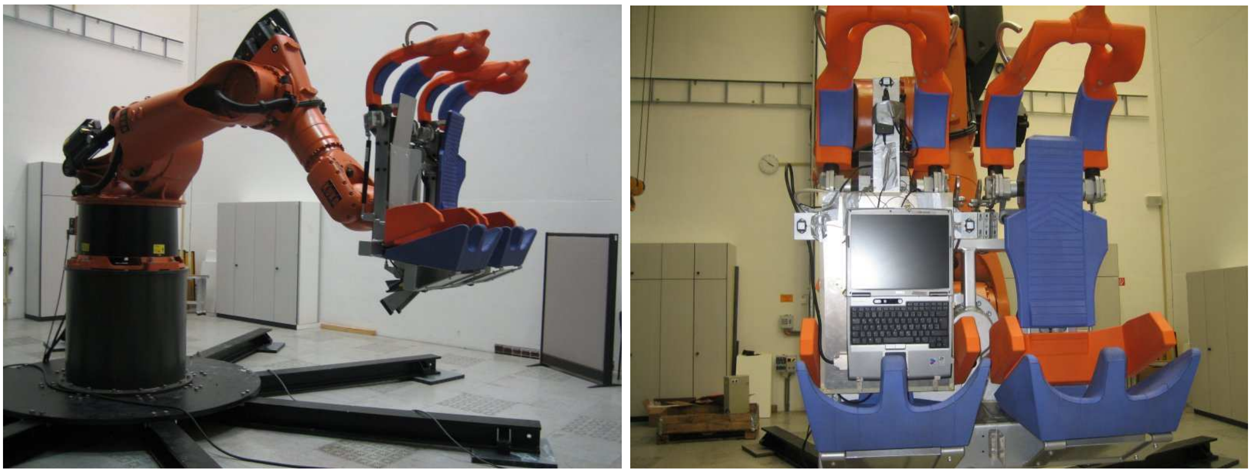


Figure 16: Test platform and measurement system

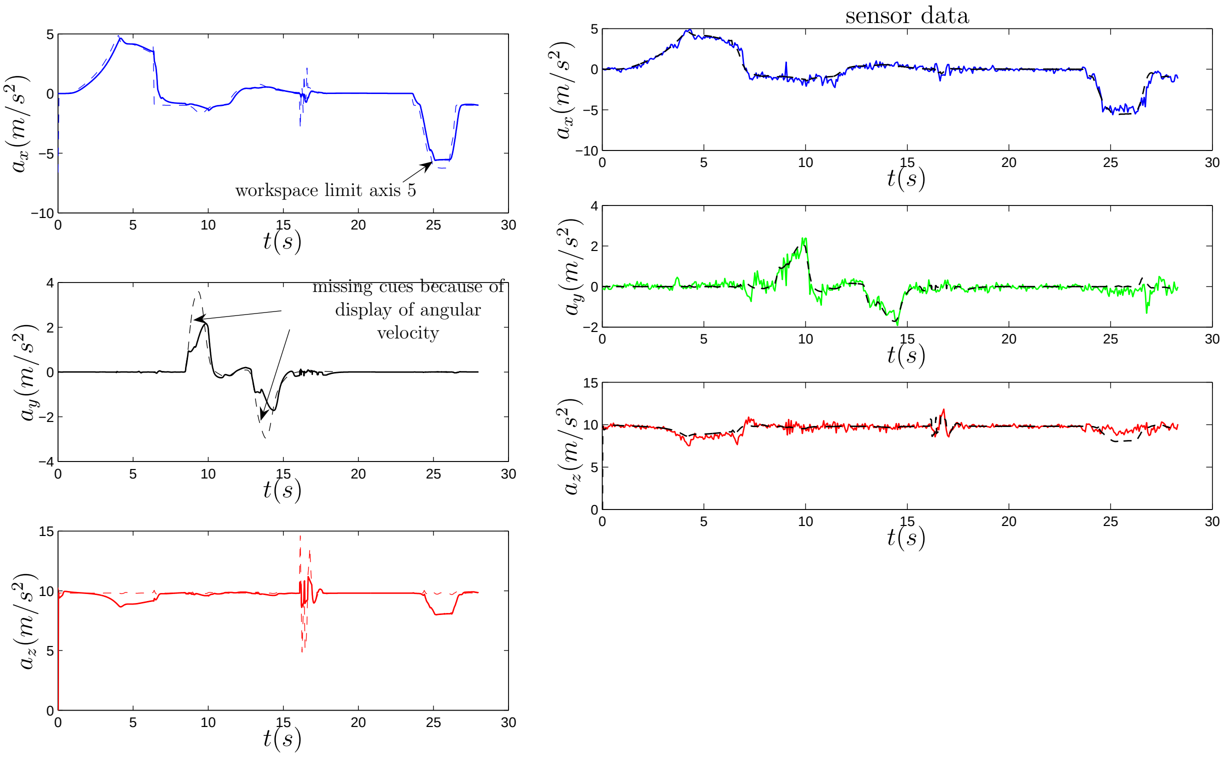


Figure 17: Comparison between input acc. and simulated acc. (left) and between simulated and measured acc. (right)

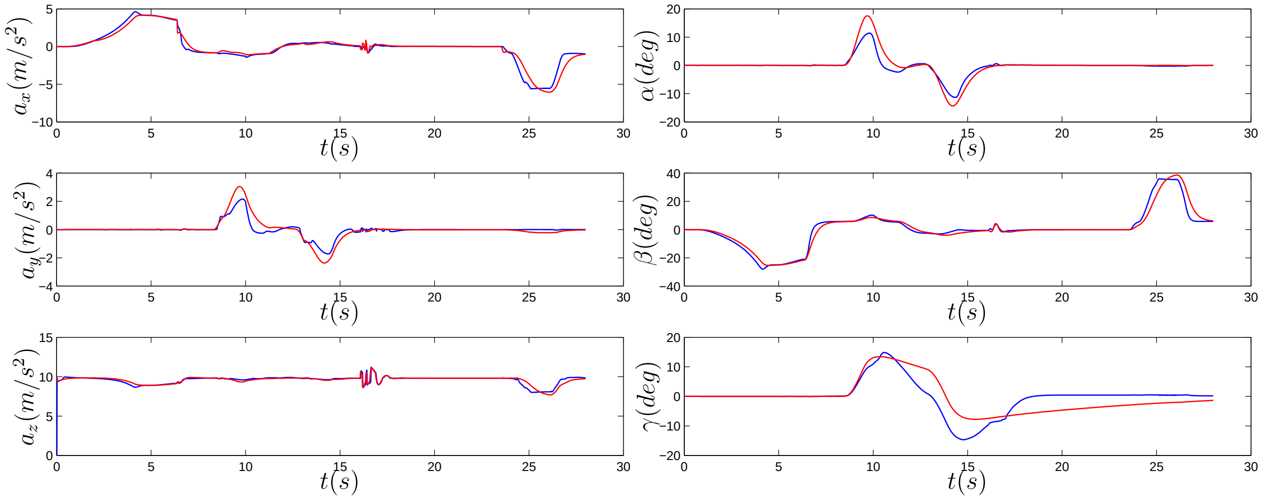


Figure 18: Comparison between washout-filter results (red) and the results of the optimization problem (blue)

7 Summary and Outlook

The new method for motion simulator path-planning based on an optimization turned out to be a robust method for the generation of simulator trajectories for industrial robots. Simulated trajectories match the results of the compared washout filter and the robot can follow the planned trajectories without getting in danger of violating one of the robot joint restrictions. The calculation of the path can be done in real-time on an actual personal computer. A suitable initial configuration improving the acceleration possibilities and movability of the robot has been defined.

Many topics still have to be investigated, such as the integration of an additional linear axis in the path-planning algorithm, the validation of the results with test pilots (the actual test robot is not certified by the German TÜV) and the construction of the simulator cell equipment, such as controllers, gauges etc.

Acknowledgement

We would like to thank

- Dr. Gertjan Looye for the preparation and provision of aircraft flight simulation data, especially for roll maneuvers,
- Dr. Jakub Tobolar for the preparation and provision of vehicle simulation data for the double bend maneuver,
- KUKA AG and KUKA Robotics for their support on the KR500 robot and on the Robocoaster.

References

[1] Johann Heindl, Martin Otter, Heiko Hirschmüller, et al., The Robocoaster as Simulation Platform – Experiences from the ”First Authentic Mars Flight Simulation”, *DLR, 1. Motion Simulator Conference, Braunschweig*, Sept 2005.

[2] Specification KUKA KR500/1: *http://www.kuka.com/*

[3] SIMPACK Product Homepage: *http://www.simpack.com/*

[4] L.D. Reid, M. Nahon, ”Flight Simulator Motion-Base Drive Algorithms: Part 1 - Developing and Testing the Equations”, *Univ. of Toronto, Inst. for Aerospace Studies Rept 296, Toronto, ON, Canada* 1985.

[5] P.R. Grant, L.D. Reid, ”Motion Washout Filter Tuning: Rules and Requirements”, *Journal of Aircraft*, Vol 34, No.2 March-April 1997

[6] M. Brünger-Koch, S. Briest, M. Vollrath, ”Do you feel the difference? A motion assessment study”, *DSC Asia/Pacific 2006 - Tsukuba*, May/June 2006

[7] C.L. Lawson, R.J. Hanson, Solving Least Squares Problem. *Prentice Hall*, 1974

[8] M. Papageorgiou, Optimierung, *Oldenbourg Verlag GMBH, München*, 1991