

MPI Motion Simulator: Development and Analysis of a Novel Motion Simulator

Teufel, H. J.¹, H.-G. Nusseck², K. A. Beykirch³, J. S. Butler⁴, M. Kerger⁵, and H. H. Bühlhoff⁶
Max Planck Institute for Biological Cybernetics, Tübingen, Germany

This paper discusses the technical issues that were required to adapt a KUKA Robocoaster for use as a real-time motion simulator. Within this context, the paper addresses the physical modifications and the software control structure that were needed to have a flexible and safe experimental setup. It also addresses the delays and transfer function of the system. The paper is divided into two sections.

The first section describes the control and safety structures of the MPI Motion Simulator. The second section shows measurements of latencies and frequency responses of the motion simulator. The results show that the frequency responses of the MPI Motion Simulator compare favorably with high-end Stewart Platforms, and therefore demonstrate the suitability of robot-based motion simulators for flight simulation.

I. Introduction

To train and assess pilots, it has become commonplace to use motion simulators for safety, as well as economic reasons. One of the main challenges when training on a motion simulator is recreating the motion of a complex system within the limited range of motion, such that the pilot believes the dynamics are the same as that of a plane or helicopter.

Stewart platforms are the most commonly used motion simulators since they have a high payload and can achieve high accelerations¹. The drawbacks of Stewart platforms are their limited range and dexterity. With this in mind, a novel motion simulator, the MPI Motion Simulator, was developed as a viable and superior alternative¹. The MPI Motion Simulator is based on the commercial Robocoaster, manufactured by KUKA Roboter GmbH², which has been modified for use as a real-time motion simulator.

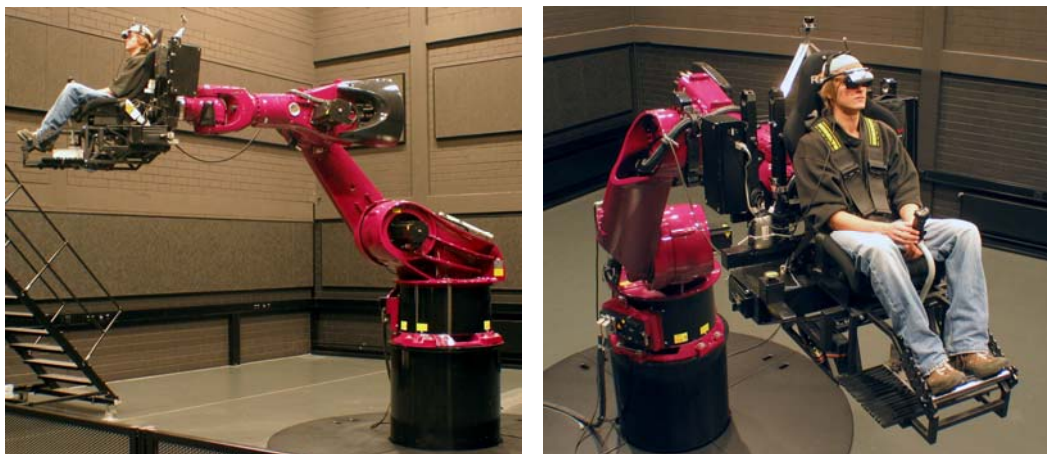


Figure 1: MPI Motion Simulator setup.

¹ Research Assistant, Department Bühlhoff, Spemannstraße 44, 72076 Tübingen.

² VR-System Administrator, Department Bühlhoff, Spemannstraße 44, 72076 Tübingen.

³ Research Scientist, Department Bühlhoff, Spemannstraße 44, 72076 Tübingen, Member AIAA.

⁴ Research Scientist, Department Bühlhoff, Spemannstraße 44, 72076 Tübingen.

⁵ VR Administrator, Department Bühlhoff, Spemannstraße 44, 72076 Tübingen.

⁶ Director, Department Bühlhoff, Spemannstraße 44, 72076 Tübingen, Member AIAA.

The Robocoaster is a 3-2-1 serial robot³ with a large motion envelope (see Figure 2). Due to the large motion envelope compared with a Stewart Platform and its dexterous nature, it is ideal for use as a flight simulator, since 1-1 motion cueing for some complex tasks, such as unusual attitude recovery or the side step maneuver in a helicopter^{4,5}, is possible.

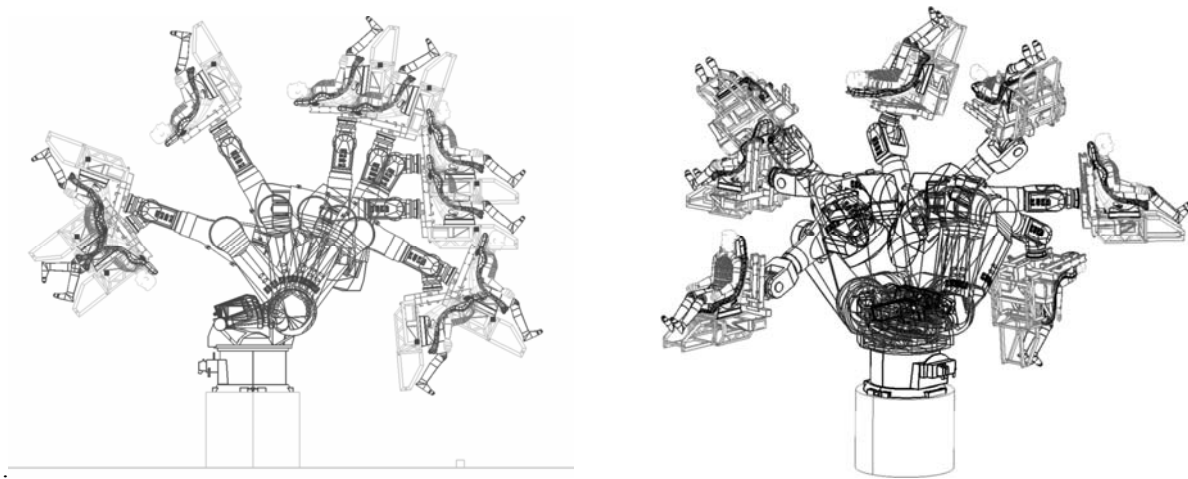


Figure 2: Graphical representation of the MPI Motion Simulator work space

This paper discusses the technical issues that were required to adapt the KUKA Robocoaster for use as a real-time motion simulator. Within this context, the paper addresses the physical modifications and the software control structure that were needed to have a flexible and safe experimental setup. It also addresses the motion simulator's velocity and acceleration limitations, as well as the delays and transfer function of the system. The paper is divided into two sections. The first section describes the control and safety structures of the MPI Motion Simulator. The second section shows measurements of latencies and frequency responses of the motion simulator.

II. Physical modifications and development of control structure of the MPI Motion Simulator.

A. Physical modifications of the MPI Motion Simulator

The MPI Motion Simulator consists of a six-joint serial robot in a 3-2-1 configuration. It is based on the commercial KUKA Robocoaster² (a modified KR-500 industrial robot³ with a 500kg payload), which was originally designed for use in amusement parks. In order to increase passenger safety and avoid collisions with the ground or the Robocoaster itself, stringent safety measures were implemented in the Robocoaster. There are aluminum hardware end-stops on four of the Robocoaster's joints. As required by the German safety association "TÜV", these hardware end-stops absorb the kinetic energy of the robot by deforming the aluminum in case of a crash. Axes 4 and 6 can rotate continuously. Continuous rotation of Axis 1 is currently being developed allowing use of the simulator as a centrifuge. Hence the system has 4 pairs of hardware end-stops limiting Axis 1, 2, 3 and 5 in both directions. The original choice of the hardware end-stops was based on a maximization of rotations, at the expense of potential linear motion. For scientific purposes, allowing large linear motions is also quite important. As a first attempt to allow more linear movement, the hardware end-stops of Axes 2 and 3 were modified. The maximum range of linear movements is strongly dependent on the position from which the movement begins. With the modified hardware end-stops, lateral movements of $\pm 1.5\text{m}$ to the left and to the right can be realized by stretching out Axis 2 and 3 and holding an upright position with the other four axes. A new set of hardware end-stops is under construction and will be mounted at the time Axis 1 is modified for continuous rotation in 2008. The current hardware end-stops of the MPI Motion Simulator are shown in Table 1. Table 2 shows the maximum linear amplitudes possible from a particular starting position which was optimized for linear left-right motion. As a consequence up-down and forward-backward motions are more restricted than from other starting positions. Nevertheless, complex motion profiles that combine lateral movements with rotations are possible (see Figure 2). Stewart platforms, which cannot rotate at the limits of their linear motion range, are less capable for certain tasks.

Payload	500kg	
Number of axes	6	
Repeatability	+/- 0.15mm	
Weight approx.	2350kg	
Motors	Electromechanical, AC-Servo Motors	
Installed engine output	41kW	
Sound level	< 75 dB(A)	
Axis data	Range [deg]	Max. velocity [deg/s]
Axis 1	-133 to +133	69
Axis 2	-128 to -42	57
Axis 3	-30 to +78	69
Axis 4	Continuous	76
Axis 5	-58 to +58	76
Axis 6	Continuous	120

Table 1: Technical specifications of the MPI Motion Simulator.

As delivered, the Robocoaster is equipped with a two heavy roller coaster-style seats, which allow rapid entry and exit of passengers in amusement parks (Figure 1, left). For basic science research on multi sensory perception the original Robocoaster seats were not suitable. Hence the original seats were replaced by a single Recaro¹⁰ seat with a Schroth¹¹ safety belt system all mounted on a platform directly at the flange (Figure 1, right). A footrest and flight stick was also mounted on the platform.

Direction of linear movement	Range [m]
Left-right	3.0
Up-down	0.45
Forward-backward	0.92

Table 2: Maximum linear range based on the starting position: {0, -102, 78, 0, -23, 0} deg.

To ensure that the entry stair is at the parking position, that the safety belt is closed and that no one can enter the working space of the robot while it is in use, continuous monitoring of these areas is used to stop all motion should an unsafe state occur. For experiments performed in darkness, infrared cameras allow visual monitoring from the control room.

1. Visualization system

The current visualization system (as used in current experiments⁴) consists of an eMagin HMD (3D Visor Z800⁷). An optical head tracking device (*Natural Point TrackIR*⁸), which is mounted on the seat, tracks the head position of the pilot. The display software (based on the MPI VeLib⁹), receives the current robot position and the information from the head tracking device to calculate the virtual camera position. The virtual environment is presented with an update rate of 60Hz.

2. Simulation environment

Before any experiment is performed on the MPI Motion Simulator, each experimental motion trajectory undergoes a testing phase on a KUKA simulation PC (Office PC). The “Office PC” is a special product sold by KUKA which simulates the real robot arm and includes the identical operating system and control screen layout as the real robot. The control structure that is described below was also implemented on the “Office PC” environment.

B. Control structure of the MPI Motion Simulator

1. Overview

A schematic overview of the control system of the MPI Motion Simulator is shown in Figure 3. It is divided into two main parts: the KUKA control system, consisting of the robot and KUKA control Unit (shown on the right of Figure 3), and the two components of the MPI control system (shown on the left half of Figure 3). The KUKA control Unit sends the current joint-angles to the MPI control system at an internal rate of 12ms and receives back joint-angle differences to perform the axis movements. Both are transmitted by an Ethernet connection using the UDP protocol.

2. Nomenclature

So far the terms joint angle differences and joint angles have been introduced which need some explanation. The KUKA control system receives joint angle differences (in radians) from the MPI control system. These joint angle differences are small increments that are sent within one heartbeat (12ms interval). For a given starting position, the movement of each robot joint is determined by the sum of all joint angle differences received by the KUKA control system. Because the time interval of 12ms is constant, the small increment or joint angle difference may be interpreted as a joint velocity, hence the change of these joint angle differences over time may be interpreted as acceleration. The position feedback signal which is sent from the KUKA control system to the MPI control system is used as a heartbeat synchronization.

The term heartbeat stands for the communication between two network clusters to verify that the other cluster is still alive. Here the term heartbeat synchronization means that the KUKA control system sends position data at each 12ms interval. The MPI control system is triggered by receiving this position feedback from the KUKA control system. Within each heartbeat trigger, the MPI control system calculates the new robot position and sends the command to the KUKA control system. This procedure ensures that both control systems run with the same frequency.

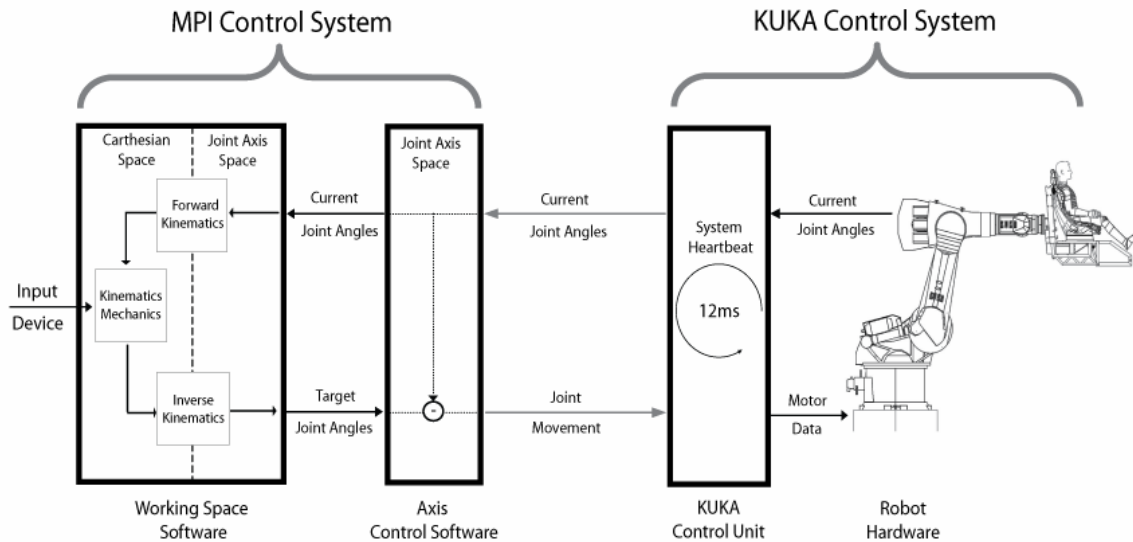


Figure 3: Schematic overview of the control system of the MPI Motion Simulator

3. Control structure

The MPI control system introduced above and shown in Figure 3 is subdivided into two parts: the Cartesian and the joint-angle control system. Forward kinematics, based on the Denavit-Hartenberg convention, is used to transform the current joint-angles to Cartesian coordinates. The vehicle model computes the new Cartesian position for the next 12ms cycle. The limits of the Cartesian working space are monitored and the new Cartesian positions are transformed back to joint-angle values by an analytic implementation of the inverse kinematics. The new joint-angles, joint velocities and accelerations are monitored by the axis control system. Joint-angle differences are finally passed to the KUKA control system. Depending on the task, different input devices, such as a helicopter cyclic stick and rudder paddles may be attached to the MPI control system.

Each of these four subdivisions incorporates its own safety procedures. The working space software monitors the Cartesian position. The axis control software monitors the joint-angles, angular velocities and angular accelerations. The KUKA control unit provides an inherent redundant supervision of these, with software end-stops for the angles. The robot itself is equipped with aluminum end-stops that would absorb the energy in case of failure of the preceding safety systems.

4. Motion Filter

The KUKA control unit is equipped with an adjustable moving average filter to smooth out the joint angle differences sent by the MPI control system. The factory default is 19 heartbeat cycles of 12ms. All data shown in this paper are collected with no filter present. For security reasons, current experiments are performed with the default filter setting.

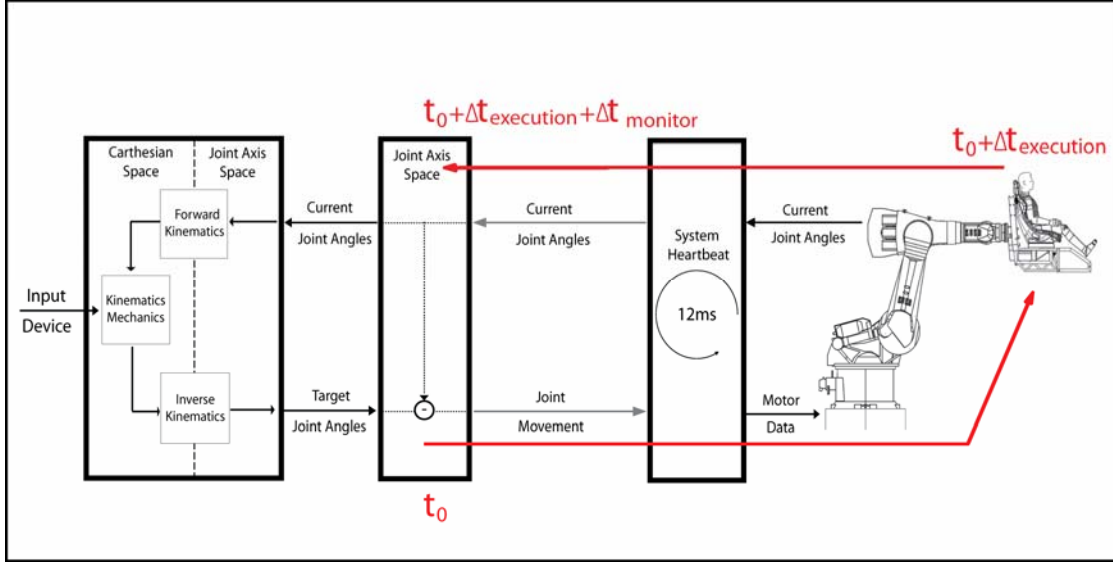


Figure 4: Estimation of latencies from the initial command at time t_0 to execution of the command at time $t_0 + \Delta t_{\text{execution}}$ and receiving the response at time $t_0 + \Delta t_{\text{execution}} + \Delta t_{\text{monitor}}$.

III. Latencies

A. Introduction

In order to control and monitor the motion simulator it is necessary to be aware of all latencies that occur. Here we measured the time that passes from the command of new joint difference angles from the MPI control software to the KUKA control (t_0) unit until the movement is finally executed ($t_0 + \Delta t_{\text{execution}}$, see Figure 4). The KUKA control unit also passes the current joint-angles to the control software. These were also monitored to compare to the time of the command (t_0). We wanted to answer the question of how much time, in ms, or internal heartbeat cycles of 12ms, passes from the moment the command is sent until the command is executed, and how much time or how many frames pass until the resulting position is received by the control software ($t_0 + \Delta t_{\text{execution}} + \Delta t_{\text{monitor}}$). To perform this measurement an external sensor had to be connected to the simulator.

B. Measurement setup

A 1 meter long linear potentiometer with 10kOhm resistance was connected to a PEAK-Micromod⁶ measurement module with 10bit AD-converters. The digitized potentiometer value was sent via CAN-bus and a USB-CAN converter at 1 kHz to the control-PC. The robot was programmed to perform a sinusoidal linear movement with a 40cm amplitude and a frequency of 0.25Hz. This linear movement incorporates all six axes of the robot and drove the potentiometer covering approximately 80% of the linear potentiometer range. The latency of the CAN-bus data acquisition system was about 1ms and hence much smaller than the internal 12ms rate of the robot. Within one cycle of the heartbeat-synchronized measurement loop, the control software receives the current robot position from the KUKA control unit ($\mathbf{P}_{\text{monitor}}$), reads the current potentiometer value via CAN-bus ($\mathbf{P}_{\text{measurement}}$), and finally computes the next desired robot position ($\mathbf{P}_{\text{target}}$), which is then transmitted to the KUKA control unit.

C. Results:

The blue curve in Figure 5 shows the target robot position; the red curve shows the robot's position feedback and the green curve the measured potentiometer position.

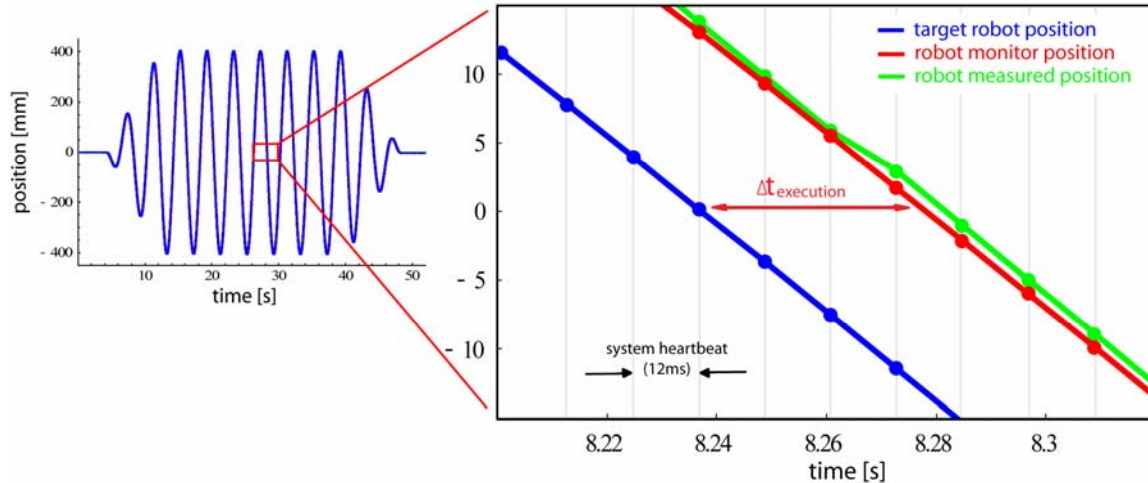


Figure 5: Results of the latency measurements. The left part of the figure shows the complete sinusoidal trajectory. The abscissa shows time in seconds and the ordinate the lateral position in mm. The right part shows a magnification of 120ms out of the full trajectory. *Blue*: target robot position ($\mathbf{P}_{\text{target}}$), *Red*: robot position feedback ($\mathbf{P}_{\text{monitor}}$), *Green*: measured potentiometer position ($\mathbf{P}_{\text{measurement}}$).

The latency between the position feedback of the KUKA control unit and the measured potentiometer feedback ($\Delta t_{\text{monitor}}$) is approximately 0.5ms. The latency between the target robot position ($\mathbf{P}_{\text{target}}$, blue curve) and the robot position feedback ($\mathbf{P}_{\text{monitor}}$, red curve and $\mathbf{P}_{\text{measurement}}$, green curve) is approximately 42ms (3.5 heartbeat cycles).

We calculated cross correlations between the command signals and the position feedback data ($\mathbf{P}_{\text{monitor}}$, $\mathbf{P}_{\text{measurement}}$). The measurements were performed with various settings of the moving average filter size in the KUKA control system. The correlation coefficients (Table 3) show that the ramped sinusoid input signals are perfectly reproduced by the MPI Motion Simulator (mean correlation coefficient = 1.00). The potentiometer values do not show perfect correlation, because jerks occur at the transition from static to kinetic friction (mean correlation coefficient = 0.96). Larger values of the moving average filter setting simply cause a larger latency of the execution.

Filter	Command vs. potentiometer [ms]	Command vs. pos feedback [ms]
1	41.21 (0.98)	40.77 (1.00)
3	53.15 (0.97)	52.77 (1.00)
6	71.19 (0.93)	70.77 (1.00)
10	95.23 (0.94)	94.78 (1.00)
19	149.21 (1.00)	148.78 (1.00)

Table 3: Latencies for different filter settings. Correlation coefficients are shown in brackets.

By comparing the two columns of Table 3, we notice that the time difference between execution of the command and position feedback of the response ($\Delta t_{\text{monitor}}$) is less than 1ms. From these measurements, we established that our system has a latency of 41ms. We have also shown that the robot movement and position feedback occur within one heartbeat cycle.

IV. System analysis

A. Introduction

To estimate the transfer functions of the system, joint-angle responses for sinusoidal trajectories of different frequencies and amplitudes were recorded. Both the signal sent to the motion simulator and the signal recorded by the position feedback was Fourier transformed, and the gain and phase were computed for the respective frequency of the original sinusoid.

B. Methods

The input signals for the estimation of the system transfer functions were sinusoidal trajectories of different frequencies. When possible, each trajectory consisted of 25 full cycles. Joint-angle position was measured with the KUKA position feedback. During all robot movements, the KUKA control system monitors the velocity, acceleration and energy status of the system. To avoid the acceleration limits of the KUKA control system all input signals was faded in and out via a ramp function of 4 periods.

The KUKA control system incorporates a current monitor to avoid excessive energy use. This might interrupt the trajectory in some cases (especially axis 1, which has to move the largest mass). This occurred frequently with trajectories with high amplitudes and high frequencies. To overcome this problem, these trajectories were shortened to 2 ramp-cycles and 12 (8 in some cases) periods of the full sinusoids. The trajectories of the lowest frequencies were also set to these values simply because the measurement would have taken too much time.

C. Results

The transport delay was computed in two ways. At the lowest frequencies (0.05Hz, 0.10Hz and 0.25Hz), the frequency dependent component of the delay (phase) is expected to be quite small and the system shows just the transport delay. Additionally, frequency-domain system identification methods were used to estimate the best-fit transfer functions, which included a transport delay term. As in the latency measurements performed with the potentiometer, the transport delay was found to be 41ms. For the lowest frequencies and the lowest amplitudes, the transport delay was a bit longer, because the position increments of the sinusoidal input signal was smaller than the smallest increment the MPI Motion Simulator could perform.

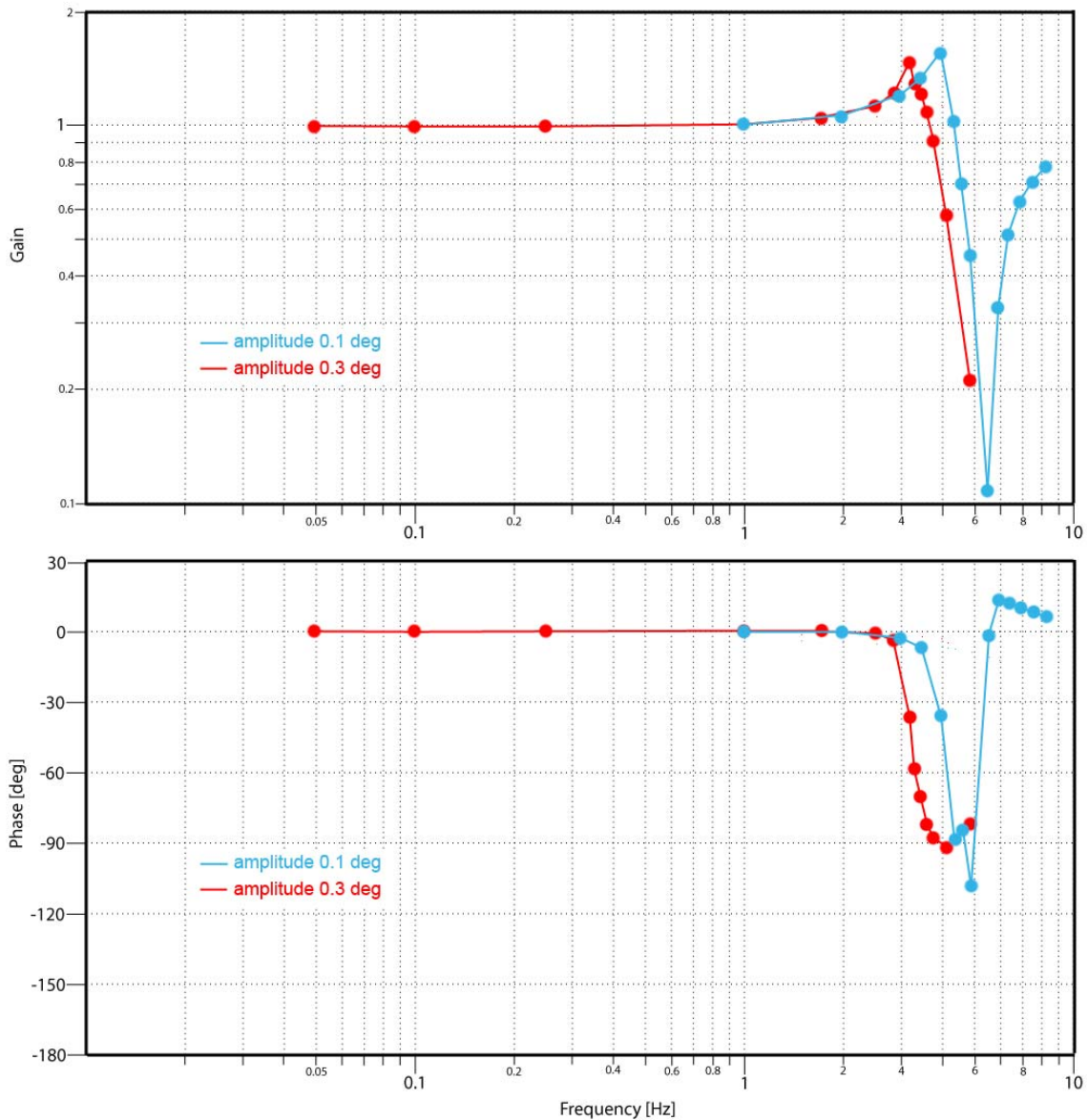


Figure 6: Bode plots of axis 1 measurements

Figure 6 shows the individual measurements of the frequency response of axis 1 for the two smallest amplitudes. Figure 7 shows the results obtained from measurements of all six axes of the motion simulator. The frame delay of 41 milliseconds was removed from all phase diagrams.

All axes were measured with amplitudes of 0.1 deg (blue dots, Figure 6) and 0.3 deg (red dots, Figure 6). Axis 1 was also measured with amplitudes of 3 deg and 30 deg (not shown here, because the measurement showed the same results as obtained for the lower amplitudes). Frequencies were measured from 0.05 Hz to the maximum frequency that could be measured depending on the amplitude. The highest frequency that could be measured varies from axis to axis. As a consequence the full resonance can be seen for some axis but not for others.

Figure 8 shows the response of axis 1 to a ramp of $1 \cdot 10^{-8}$ rad ($5.7 \cdot 10^{-7}$ deg) per system cycle (12ms) versus the command input. The blue curve shows the simulated results from the Office PC. The red curve show the results obtained by the real MPI Motion Simulator. As can be seen, the two data sets show the same minimal step size, but differ in vibrations of the real motion simulator which are of the same magnitude as the minimal incremental step size. This also shows that the current joint angles are transmitted from the KUKA control system to the MPI control system.

Instead of a linear slope, which would be expected if each step of 10^{-8} rad is performed sequentially, the robot does not move until a movement command of approx. $2 \cdot 10^{-6}$ rad (0.00012 deg) is reached. Depending on the axis this minimal increment is of the order of $1.9 \cdot 10^{-6}$ rad (0.00011 deg) for axis 3, to $3.1 \cdot 10^{-6}$ rad (0.00018 deg) for axis 6. In all measurements reported in this paper the simulator was run with a command and monitor accuracy of 10^{-8} rad.

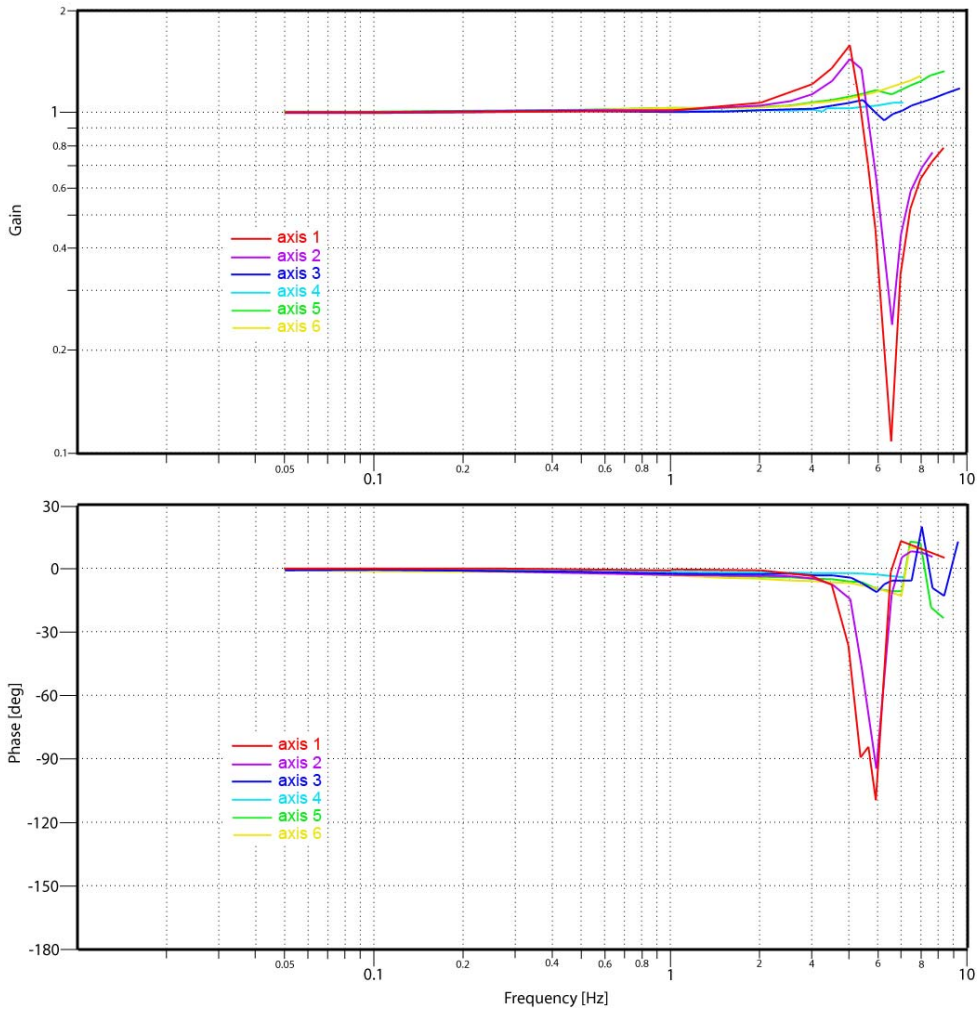


Figure 7: Bode plots of measurements of all six axes

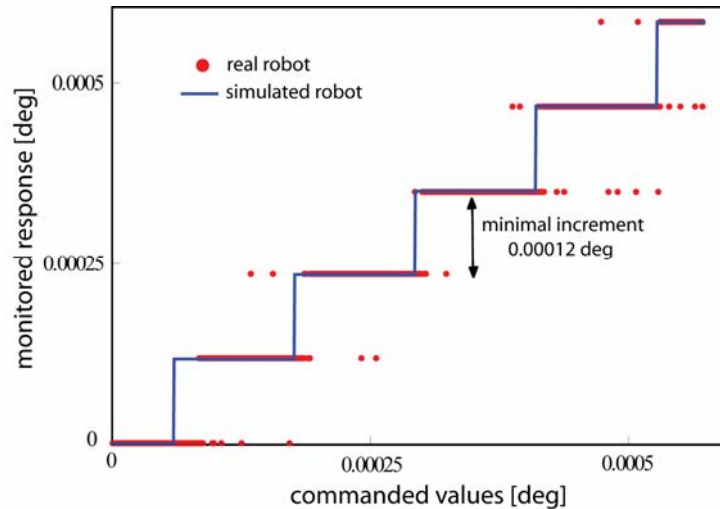


Figure 8: Minimal increment of axis 1. The abscissa shows commanded joint position, ordinate shows response of MPI Motion Simulator. Units in degrees.

V. Discussion

Two independent methods of measuring the transport delay of the MPI Motion Simulator yielded similar results. The transport delay was computed by cross correlation of the input signal and the recorded simulator movement as assessed by the potentiometer measurement. The same transport delay of 41ms was also established in measurements of the transfer functions of the motion simulator for low frequencies and large amplitude signals.

The system shows mechanical resonances at frequencies depending on the chosen axis. The lowest resonance frequency occurs for axis 1 which was expected due to the large mass that has to be driven by this axis. However, the payload capacity of 500 kg is relatively low compared to different systems. Therefore, the visualization and control loading system are limited in size and weight. The complex control structure required to safely perform the desired motions was a challenge that has been sufficiently addressed and the overall system was approved by the German safety agency (TÜV).

The main advantage of this type of simulator is that the motion envelope is large and that the different degrees of freedom are not fully coupled. Hence, larger rotation angles are possible than with traditional Stewart platforms, especially when using linear and rotational degrees of freedom simultaneously. This allows for unusual attitudes and therefore the simulator can be used for a variety of flight scenarios, including upset recovery and aerobatics training, and also for our current helicopter hover experiments⁴.

The low transport delay of the MPI Motion Simulator enables high-fidelity motion simulation and basic research on multi-sensory perception in virtual environments. The motion envelope will be improved further by modifications of the hardware end-stops, and later by implementing a double redundant software motion envelope monitoring system, increasing the motion envelope even further.

References

- ¹Pollini, L., Innocenti, M., and Petrone, A., "Study of a Novel Motion Platform for Flight Simulators using an Anthropomorphic Robot," *Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit, Keystone, Colorado*, AIAA 2006-6360 August 21–24, 2006
- ²<http://www.robocoaster.com/> [cited 13 August 2007].
- ³http://www.kuka.com/germany/en/products/industrial_robots/heavy/kr500_2/start.htm [cited 13 August 2007].
- ⁴Karl Beykirch et al., "Control of a Lateral Helicopter Sidestep Maneuver on an Anthropomorphic Robot", *Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit, Hilton Head, SC* August 20–23 2007
- ⁵Schroeder, J. A., "Helicopter Flight Simulation Motion Platform Requirements," Tech. Rep. NASA/TP-1999-208766, NASA, July 1999.
- ⁶http://www.peak-system.com/db/gb/pcanmikromod_gb.html [cited 13 August 2007].
- ⁷<http://www.3dvisor.com/> [cited 13 August 2007]
- ⁸<http://www.naturalpoint.com/trackir/> [cited 13 August 2007]
- ⁹<http://velib.kyb.tuebingen.mpg.de/> [cited 13 August 2007]
- ¹⁰<http://www.recaro.com/> [cited 13 August 2007]
- ¹¹<http://www.schroth.com/> [cited 13 August 2007]