Development of an Integrated Design and Simulation Environment for Concurrent Base-Arm Motion Control of Space Manipulators

Zheng Zhong[1], **M. Reza Emami**[1,2]
[1]*Institute for Aerospace Studies, University of Toronto, Canada*
[2]*Division of Space Technology, Luleå University of Technology, Sweden*
reza.emami@ltu.se

ABSTRACT

To develop and verify the motion control of space manipulators experimentally, a manipulator system with a freely-moving base is needed. In addition, an integrated design and simulation environment is required for the system to study the operation under various conditions and in different scenarios before implementing the control scheme on the experimental hardware setup. Lastly, certain experiments need to be designed to validate and compare the performance of various control schemes quantitatively. This paper discusses the development of a new free-base manipulator system with its integrated design and simulation environment, as well as the design of a set of experiments that can statistically evaluate the performance of concurrent base-arm motion control schemes.

1. INTRODUCTION

Ground verification and validation experiments play a key role in the practical implementation of space robotic systems. Many possible ground experiment methods exist for space robotic systems, such as neutral buoyancy [1,2], parabolic flight [3,4], and suspension system [5,6]. A thorough survey of different testing techniques can be found in [7].

Autonomous robotic manipulators have many possible applications in space environment, such as servicing in-orbit satellites [8] or capturing and removing space debris [9,10]. Such space manipulator systems introduce some challenges to their control, due to having free bases that react to the motion of the manipulators [11]. These systems are also difficult to test on Earth due to gravitational conditions. A suitable test method is called planar air-bearing table, where the manipulator system is supported by air-bearings on a table made of proper material. This allows for nearly frictionless motion on the table surface, thus simulating the microgravity conditions in two dimensions. There are many examples of experiment platforms utilizing the air-bearing table to test space manipulator systems [12-16]. Some platforms feature multiple manipulator arms to study cooperative operations [17,18]. A notable approach is the implementation of Dynamically Equivalent Manipulator (DEM) concept [19], which models a free-base manipulator as a fixed-base manipulator with a spherical joint located at the center of mass of the original manipulator.

The existing experiment platforms are each limited to a very specific set of experiments, and the results from these vastly different setups are difficult to compare. Thus, an experiment platform that can apply different control schemes to similar experiments is required. In addition, an integrated design and simulation environment (IDSE) is needed to facilitate the implementation and verification of different control schemes. Lastly, suitable experiments need to be designed so that the performance of different controllers can be statistically examined.

In this paper, a new free-base manipulator system with its integrated design and simulation environment is presented. In section 2, this manipulator system is described in detail. The simulation counterpart of the system is presented in section 3. Some experimental design considerations for validating control schemes are highlighted in section 4. A case study demonstrating the capabilities of hardware and software platforms is discussed in section 5.

2. HARDWARE PLATFORM

2.1 General Description

The constructed free-base manipulator system consists of a six degree-of-freedom spatial robot manipulator mounted on a circular base, as shown in Fig. 1. Such actuated degrees of freedom allows for spatial motion of the end-effector as well as concurrent motion control of both the end-effector and the base. The base uses three air bearings symmetrically mounted on the circumference to freely move and rotate on a 2×2 m^2 glass table, and react to the movements of the manipulator. The base also has a wide footprint that can support the entire manipulator system without additional air-bearings, allowing the manipulator to extend outside of the bounds of the glass table.

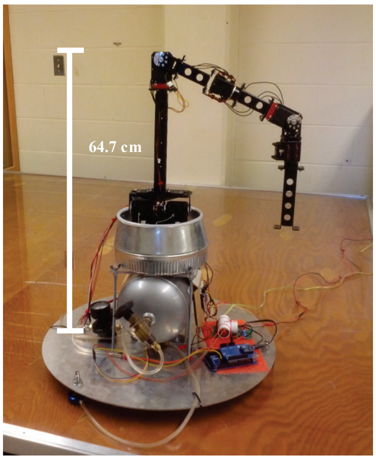


Figure 1. Assembled hardware platform

2.2 Mechanical Design

Following the concept of Dynamically Equivalent Manipulator (DEM), the range of motion for the end-effector is dependent on the location of the center of mass of the entire system. If the manipulator mass is comparable to that of the base, the end-effector would require large joint movements to produce any significant motion. On the other hand, if the base mass is significantly larger than that of the manipulator, the system will behave much like a fixed-base manipulator. Thus, mass and inertia of the manipulator system is designed to allow for large end-effector motion, large base rotational motion, and small but measurable base linear motion. A list of manipulator link properties is shown in Tab. 1. When fully extended horizontally, as shown in Fig 2, the center of mass of the system is located 11.4 cm away from the center of mass of the base. Hence, with zero initial momentum, the movement of the base is contained in a radius of 11.4 cm, while the end-effector motion is contained within a radius of 70.1 cm.

Table 1. Dynamic parameters of hardware platform

Parameter	Symbol	Value
Base mass	m_0	8.4 kg
Link 1 length	l_1	0.058 m
Link 1 mass	m_1	0.374 kg
Link 1 mass ratio	m_1/m_0	0.044
Link 2 length	l_2	0.307 m
Link 2 mass	m_2	0.504 kg
Link 2 mass ratio	m_2/m_0	0.060
Link 3 length	l_3	0.152 m
Link 3 mass	m_3	0.412 kg
Link 3 mass ratio	m_3/m_0	0.049
Link 4 length	l_4	0.135 m
Link 4 mass	m_4	0.341 kg
Link 4 mass ratio	m_4/m_0	0.041
Link 5 length	l_5	0.079 m
Link 5 mass	m_5	0.161 kg
Link 5 mass ratio	m_5/m_0	0.019
Link 6 length	l_6	0.142 m
Link 6 mass	m_6	0.342 kg
Link 6 mass ratio	m_6/m_0	0.041

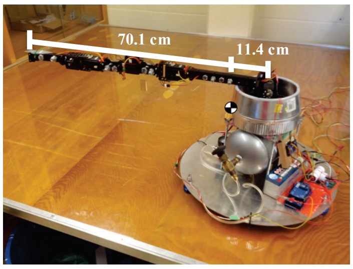


Figure 2: Fully extended manipulator

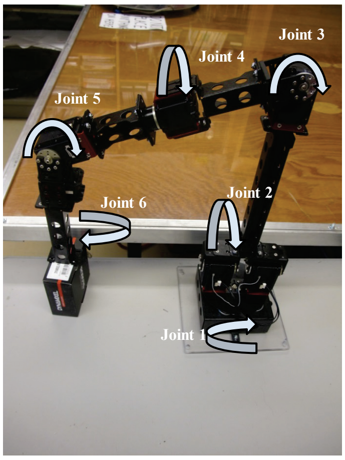


Figure 3. Hardware platform manipulator

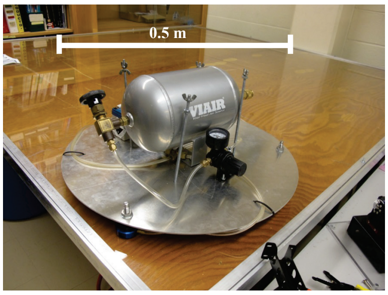


Figure 4. Hardware platform base

All joints in the 6-DOF manipulator of the hardware platform are revolute, as shown Fig. 3, and are actuated by Robotis™ MX-T servomotors. The actuators are housed inside aluminum casings, which are used as parts of the links. Aluminum girders are used to extend the links to the desired lengths, and house the wirings for the electronics. A turn-table is used to provide vertical rotation axis for the first joint, and adapters are used to align the axis of joints 4 and 6 parallel to the length of their links. The modular nature of the components gives the possibility to modify the manipulator configuration, by re-aligning and/or locking the joints, to suit different experiment scenarios. For example, joints 2, 4, and 6 can be locked to positions shown in Fig. 2. In this configuration, the manipulator system behaves as a planar manipulator with joints 1, 3, and 5 having parallel axes. The geometric and dynamic properties of the manipulator can also be adjusted by extending and attaching weights to the girders. Doing so will change the location of the center of mass of the system and the range of motion of the base and the end-effector.

The base of the system is a circular chassis made of stainless steel with an air tank attached to its center, as shown in Fig 4. The chassis is supported by three porous carbon air-bearings in a triangle formation to prevent toppling. Air is supplied to the air-bearings through flexible hoses, and is kept at operation pressure by a regulator. The air tank has sufficient capacity to operate the air-bearings for up to 2 minutes. Battery and onboard instrumentation are also attached to the base chassis. The first joint of the manipulator is connected to the top of the air tank.

2.3 Onboard Instrumentation

In order to integrate with the simulation environment, a personal computer (PC) is used as the main control unit. The PC running MATLAB/Simulink® collects data from sensors, computes the control inputs, and sends the inputs to the actuators. The PC also performs trajectory generation, data analysis, and simulations.

Each of the MX-T series servomotors is equipped with a 32Bit CORTEX-M3 microcontroller unit to control the motor. These motors have internal PID gains that can be optimized prior to or during the experiments. These motors can also be operated in torque control mode, where the PID gains are completely bypassed. The joint controllers can directly input the desired motor current, which is proportional to motor torque, as control signals. This versatility allows the manipulator system to implement both servo-based control schemes as well as dynamic model-based control algorithms.

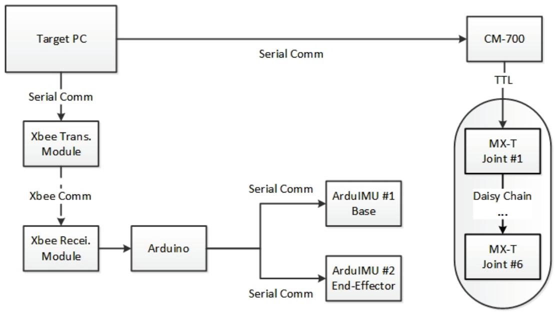


Figure 5. Onboard instrumentation scheme

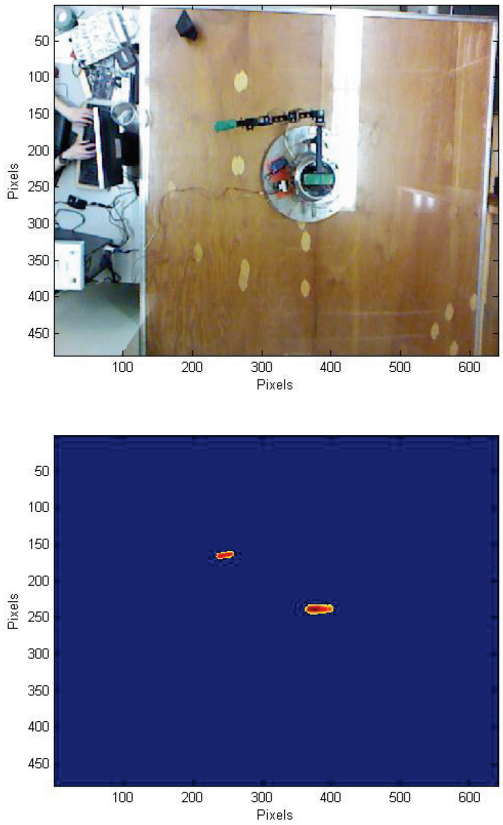


Figure 6. Captured frame (top) by Kinect® camera and processed frame (bottom) to locate visual mark

Joint positions are measured with a 12bit 360-degree contactless absolute encoder. The measured data are collected by a Robotis™ CM-700 controller via daisy chained TTL serial communication, and sent to the PC. The CM-700 also interprets commands received from MATLAB and translates the commands into control inputs for the MX-T microcontrollers. Two ArduIMU V3 inertial measurement units (IMU), with 3-axis gyro with 131 LSBs/dps sensitivity and 3-axis accelerometer, are installed at the end-effector and the base for real-time measurements of end-effector/base orientations. These measurements are monitored by an Arduino unit, and then transmitted to the PC in serial protocols via Xbee communications at the command of MATLAB. The scheme of the onboard instrumentation is presented in Fig. 5.

2.4 Visual Positioning

A Kinect® module is placed 2.5 m above the test area to provide motion tracking and real-time position measurements of the manipulator system during the experiments. From this position, each pixel in the frame has a width of approximately 4 mm.

A visual mark with a color distinctive from the background is used to identify the base of the manipulator system. Dark green color is selected for the mark to be distinguished from the predominantly red color of the test-bed. To detect the base, each frame taken by the camera on the Kinect® module is first blurred slightly by averaging the RGB values of each pixel with its surrounding pixels, in order to remove specks in the image. The blurred frame is then processed again to highlight pixels with RGB values that match the mark color within a threshold. A sample frame and processed frame is shown in Fig. 6. This processed frame is used for blob detection to locate sizable regions where highlighted pixels are grouped together. Using pre-calibrated parameters, the locations of the centroids of the blobs in the camera's frame are translated to obtain the locations of the mark in the world frame. The Kinect® module is also equipped with an IR depth sensor, allowing for spatial position measurements of the height of the blob regions in the frames.

Additional marks can be placed on the manipulator to provide position measurements on any manipulator components, such as the end-effector. On the other hand, since the position and orientation of the base and the joint angles can all be measured, one can also use forward kinematics to obtain position and orientation of the manipulator links.

3. SOFTWARE PLATFORM

3.1 General Description

The software platform consists of a physical model of the hardware platform, developed using SimMechanics™ and SimElectronics™ toolboxes in MATLAB/Simulink® environment, and a motion control scheme implemented in MATLAB® to control the hardware platform in real-time. The controller can operate both the physical model as well as the hardware platform simultaneously, creating an integrated design and simulation environment (IDSE). The control scheme can then be designed, optimized and verified using the physical model and Simulink® toolboxes before being applied to the hardware platform.

3.2 Physical Model

For a free-base manipulator system, trajectory generation and control rely on the accurate knowledge of the physical properties of manipulator system. In order to have an accurate model that can also be reconfigured in the same manner as the hardware platform, a CAD drawing of each individual component of the system is created in SolidWorks® with the same geometric and dynamic properties. The physical model of each module, such as a link, is assembled from these drawings (Fig. 7), and the full model is created using these modules to match the desired configuration of the hardware platform. Some of the physical properties that are difficult to measure, such as centers of mass and moments of inertia of each link, are also computed in SolidWorks® for using in the trajectory generation and controls. The model is then imported into SimMechanics™ and SimElectronics™ physical modeling toolboxes and 3D animation modules (Fig. 8).

3.3 Instrumentation Model

The actuator dynamics and sensor behaviors are also modeled in the software platform to closely simulate the hardware. In particular, the servomotors are implemented as mathematical objects using their transfer functions while taking into account motor dynamics, such as back-emf and gear ratio, as shown in Fig. 9. During the simulation, these mathematical objects receive input voltage signals and compute torque output signals as a real electrical object would.

In addition to the servomotors, the signals in the software platform are processed to behave like step functions instead of smooth curves. This simulates time

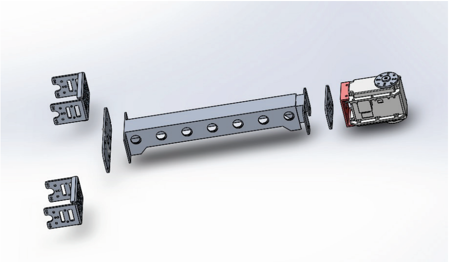
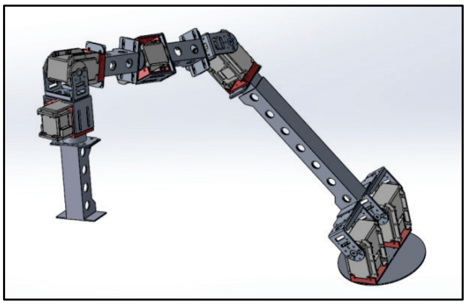


Figure 7. Components used to assemble Link 2



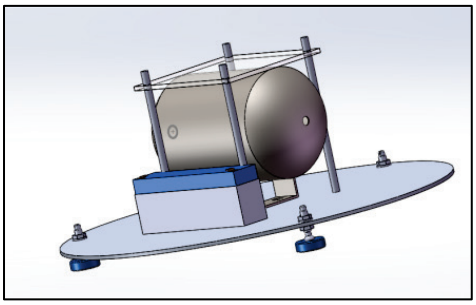


Figure 8. Assembled manipulator (top) and the base (bottom)

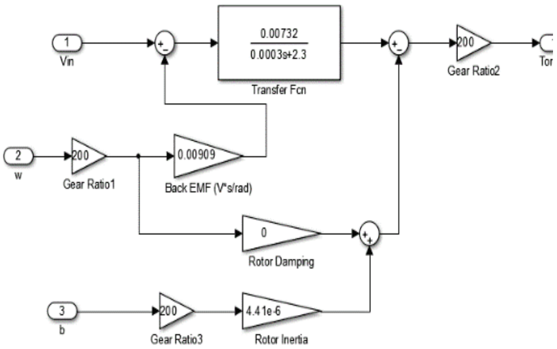


Figure 9. Simulink™ model of an electric motor

delays during communications between different hardware controller units.

3.4 Platform Integration

To integrate the hardware and software platforms, the joint trajectory generator and motion control modules are implemented in MATLAB™, so that they can be used directly by both platforms.

The joint trajectory generator utilizes the Generalized Jacobian Matrix (GJM) approach [20]. For a free-base spatial manipulator system, the rate of manipulator joints $\dot{q}_m$ at a given configuration and end-effector motion can be calculated as

$$\dot{q}_m = J^e(q_b, q_m)^{-1} \begin{bmatrix} v_e \\ \omega_e \end{bmatrix}, \qquad (1)$$

where v_e and ω_e are the end-effector linear and angular velocities, respectively, expressed in the world coordinate frame, and J^e is the Generalized Jacobian Matrix (GJM) for the end-effector motion (as denoted by the superscript e) as a function of the base orientation q_b and joint angles q_m, and is given by

$$J^e(q_b, q_m) = \begin{bmatrix} J_t^e \\ J_r^e \end{bmatrix} = \begin{bmatrix} J_{mt}^e \\ J_{mr}^e \end{bmatrix} - \begin{bmatrix} J_{bt}^e \\ J_{br}^e \end{bmatrix} M_b^{-1} M_m, \quad (2)$$

where J_m^e and J_b^e are the Jacobian matrices that relate the manipulator joint speeds and base movements, respectively, to the end-effector motion. The subscripts t and r denote translation and rotation components of the Jacobian matrices, respectively. The matrix M_b is the inertia matrix of the base, and M_m is the inertia matrix of the manipulator system. The inertia matrices relate the base angular motion and joint speeds through the conservation of momentum:

$$\begin{bmatrix} \omega_{bx} \\ \omega_{by} \\ \omega_{bz} \end{bmatrix} = \begin{bmatrix} -(M_b^{-1} M_m)_x \\ -(M_b^{-1} M_m)_y \\ -(M_b^{-1} M_m)_z \end{bmatrix} \begin{bmatrix} \dot{q}_{m1} \\ \dot{q}_{m2} \\ \vdots \\ \dot{q}_{m6} \end{bmatrix}, \qquad (3)$$

In the case of concurrent base-arm motion control of a 6-DOF manipulator with a free planar base, where as an example the task is to control the motion of the end-effector point as well as the translation and orientation of the base, the kinematic equations become:

$$\begin{bmatrix} \dot{q}_{m1} \\ \dot{q}_{m2} \\ \vdots \\ \dot{q}_{m6} \end{bmatrix} = \begin{bmatrix} J_{mtx}^e - J_{btx}^e M_b^{-1} M_m \\ J_{mty}^e - J_{bty}^e M_b^{-1} M_m \\ J_{mtz}^e - J_{btz}^e M_b^{-1} M_m \\ J_{tx}^0 \\ J_{ty}^0 \\ [-M_b^{-1} M_m]_z \end{bmatrix}^{-1} \begin{bmatrix} v_{ex} \\ v_{ey} \\ v_{ez} \\ v_{bx} \\ v_{by} \\ \omega_{bz} \end{bmatrix}, \qquad (4)$$

where J^0 is the Generalized Jacobian Matrix for the base motion (as denoted by the superscript 0). The subscripts x, y, and z denote the specific components of the matrices corresponding to the direction of translations.

Using the GJM, for a given end-effector trajectory or a combination of end-effector and base trajectories, a set of joint trajectories can be computed. The software platform can then simulate the behavior of the manipulator system during trajectory tracking. This simulation is useful in selecting a feasible end-effector and base trajectories for the hardware platform to perform by predicting events such as singularity or infeasible joint movements. Once a desirable set of end-effector and base trajectories is selected, the generated joint trajectories can be directly supplied to the hardware platform for experiments.

In addition to joint trajectory generation, motion control schemes can be designed, implemented, and optimized using the software platform and then applied directly to the hardware platform. As an example, the PID control gains for the servomotors can be designed by analyzing motor dynamics. Beginning with PD control, the control law is given by:

$$v_a(s) = K_P\big(q_d(s) - q(s)\big) - K_D\, s\, q(s), \qquad (5)$$

where v_a is the input voltage to the motor, K_P and K_D are proportional and derivative gains, respectively. The subscript d denotes reference (desired) joint angle. For a system that consists of an arm attached to a direct-drive (DC) actuator driven by control voltage from Eq. 5, the closed-loop natural frequency ω and damping ratio ζ are given by [21]:

$$\omega = \sqrt{\frac{KK_P}{I_{eff}}}, \quad \zeta = \frac{b_{eff} + KK_D}{2K\,K_P}\,\omega, \qquad (6)$$

where K is the ratio between the torque constant and the electrical resistance of the motor, I_{eff} and b_{eff} are effective inertia and effective viscous damping, respectively. To reduce steady state error of the PD control response, a high K_P value that does not cause over-current of the system or trigger structural resonance from joint flexibility is selected. The derivative gain K_D is then selected using Eq. 6 by taking $\zeta = 1$, so that the response of the system is critically damped.

The computed K_P and K_D are then chosen as the initial values in the Simulink® PID Tuner, where the control gains are optimized based on design criteria, and an integral gain K_I is selected to complete the PID controller.

4. EXPERIMENT DESIGN

Concurrent base-arm motion control of a space manipulator involves simultaneous tracking reference trajectories for both the arm and the base. A typical example is moving the end-effector in a specified path while keeping the base undisturbed, also known as the zero-disturbance motion control [22].

In order to assess the performance of a control scheme, several criteria are defined. The tracking accuracy, or tracking error, is a measure of the difference between the real trajectory and the reference trajectory. In order to obtain valid data for the tracking error, the reference trajectory should involve sufficiently large movements such that tracking errors are significantly larger than measurement errors. Hence, trajectories that can considerably disturb an uncontrolled base, particularly its orientation, are good examples of suitable reference trajectories.

Power consumption and computation time are also important criteria in assessing the performance of a control scheme. The power consumption can be measured directly from the servomotor current and voltage readings; however, a high sampling rate is required to accurately measure the extra power consumption due to high frequency chattering. Computation time can be easily obtained by using MATLAB timer functions to measure the time between output control signals.

Since tracking accuracy, power consumption, and computation time can all be measured quantitatively, they can be statistically compared between different control schemes using t-statistics:

$$t = \frac{|\bar{y}_A - \bar{y}_B| - \delta}{\sigma \cdot \sqrt{\frac{1}{n_A} + \frac{1}{n_B}}}, \qquad (7)$$

where $\bar{y}_A$ and $\bar{y}_B$ are means of two quantities, σ and n are the standard deviation and the sample size of a quantity. Under the null hypothesis:

$$(H_0): |\bar{y}_A - \bar{y}_B| \leq \delta, \qquad (8)$$

if the t-statistic in Eq. 7 is greater than the critical t-value $t_{crit}(\alpha, n_A + n_B - 2)$, then there is statistically significant evidence against Eq. 5 at α significance level, that is the difference between A and B is greater than δ.

In addition, the robustness of a control scheme, which is a measure of its performance in the presence of disturbances and uncertainties, is another important factor in practical applications. Using the integrated design and simulation environment, simulated disturbances and/or model errors can be injected into the control loop. By repeatedly measuring tracking accuracy against different levels of disturbance and/or model errors, the robustness of a control scheme can be assessed.

5. CASE STUDY

In this case study, PID joint controllers are used to track a horizontal XY trajectory of the end-effector point, while having an uncontrolled planar free-base. Since the reference trajectory is strictly in XY plane, two of the parallel joints, joint 1 and 3, are used for the control, and the remaining joints are locked. For this system, Eq. 1 is simplified to:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} J^e_{mtx} - J^e_{btx} M_b^{-1} M_m \\ J^e_{mty} - J^e_{bty} M_b^{-1} M_m \end{bmatrix}^{-1} \begin{bmatrix} v_{ex} \\ v_{ey} \end{bmatrix}, \qquad (9)$$

By utilizing the software platform to preview experimental outcomes, a straight line is found feasible (i.e., singularity free) as a reference for the end-effector trajectory (Fig. 10). Such an end-effector trajectory also covers a large distance, and causes significant rotational movement for the uncontrolled base.

In addition to generating joint trajectories, the PID gains of the joint actuators are designed and optimized as described in Section 3.4, and this set of PID gains is loaded directly onto the hardware platform for the experiment.

5.1 Experiment Results

During the experiment, the two actuated joints on the hardware platform are tasked with following the reference joint trajectories generated in the software platform, using the optimized PID control gains. These joint trajectories cause the end-effector to follow the desired trajectory. No feedback from the end-effector position was used.

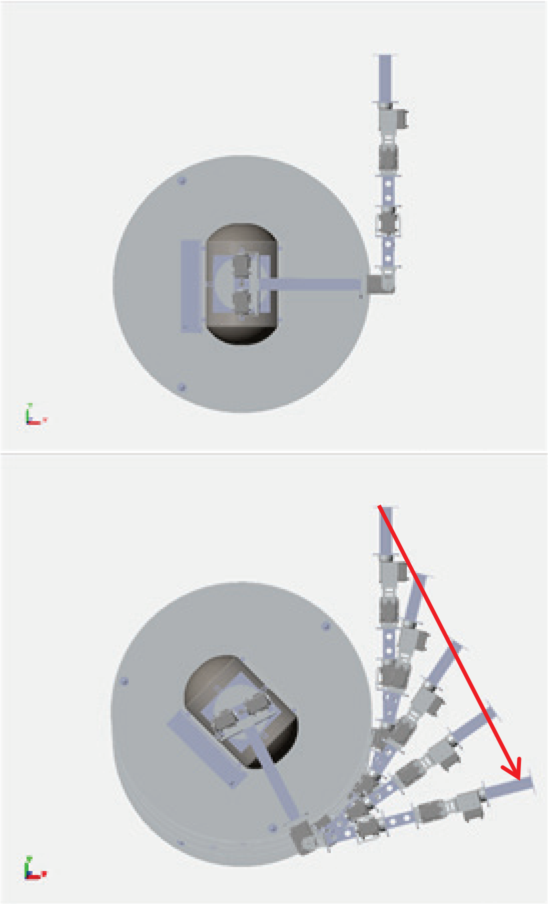


Figure 10. Initial configuration (top) and the reference trajectory (bottom)

The trajectory tracking performed by the two actuated joints during the experiment, along with the reference joint trajectories, are presented in Fig. 11 and Fig. 12. Rotational and linear motions of the uncontrolled base in reaction to the manipulator movements are presented in Fig. 13 and Fig. 14. As shown in Fig. 13 and 14, the manipulator base undergoes significant rotational movement, over 45 degrees, and minor but measurable linear movement, approximately 5cm. Finally, the end-effector position during the experiment is presented in Fig 15. The end-effector moves over 60cm during the experiment.

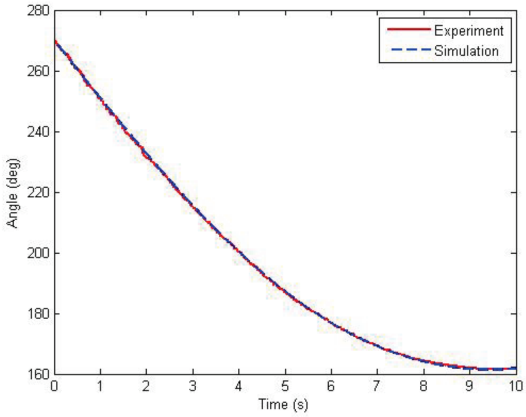


Figure 11. Joint 1 displacement

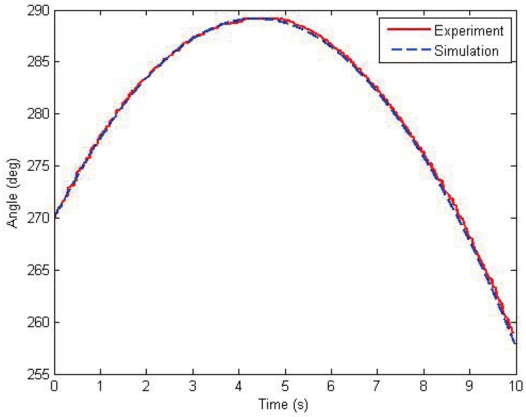


Figure 12. Joint 2 displacement

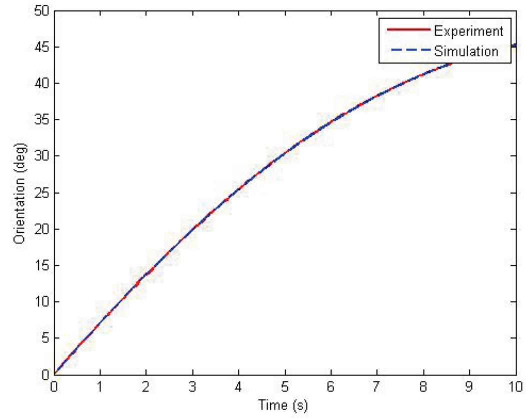


Figure 13. Manipulator base orientation

By computing the t-statistics using Eq. 7 and choosing the critical t-value with a 90% confidence level ($\alpha = 0.9$), the experimental results are compared with simulation results by setting y_A to be the difference between the two results, y_B to zero, and δ to be the measurement error. The results of the comparison are shown in Tab. 2. From the table, the difference between experiment and simulation results for joint displacements and base rotation is statistically insignificant. The experiment and simulation results for the position of base and end-effector point are statistically different. The simulation error for base position has a mean of 5.2 mm and standard deviation of 8.9 mm, and for the end-effector point has a mean of 13 mm with a standard deviation of 14 mm. This is expected since there was no feedback of the end-effector position to the controller in this experiment.

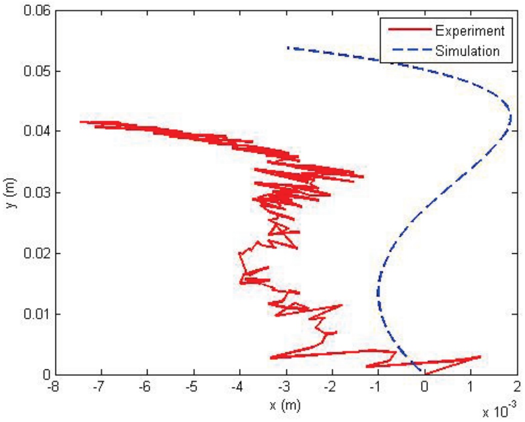


Figure 14. Manipulator base position

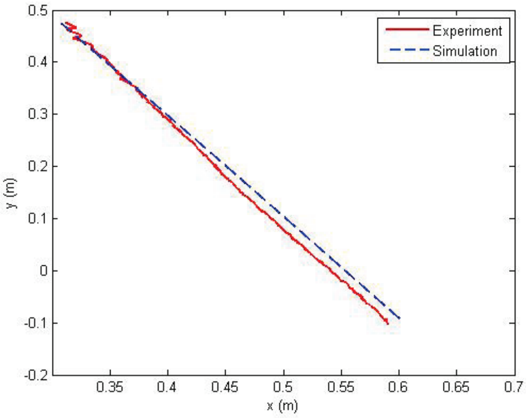


Figure 15. End-effector point position

6. CONCLUSIONS

In this paper, an experimental free-base manipulator system with its IDSE was presented. The platform has two major features: First, six actuated joins can be used for potential spatial movement of the arm as well as concurrent base-arm motion control. The modular nature of the joints allows for a wide range of system configurations. Secondly, the mass distribution of the manipulator system keeps the center of mass of the close to the base even with long manipulator length. This allows for large end-effector movements, while maintaining considerable linear and rotational motion for the base. The geometric and dynamic properties of each individual link can also be adjusted to change the effects of the free base on the manipulator motion.

Using the integrated design and simulation environment, the end-user can first preview the behavior of the hardware platform in simulation, and select reference trajectories that are suitable for the experiment. The fact that simulation and hardware platforms are both controlled in MATLAB/Simulink™ environment allows the end-user to develop and optimize a control scheme for the software platform and expect comparable performances on the hardware setup. It also allows one to easily implement different control schemes onto the platforms for verification and validation. This opens up the possibility of conducting comparative studies of various control algorithms. To achieve statistically significant comparisons between experiment results, some experimental design considerations, such as trajectory selection and comparison methods, were discussed.

Table 2: Statistical comparison of results

	$\bar{y}_A$	σ	t
Joint 1	0.154 deg	0.42	1.27
Joint 2	-0.181 deg	0.23	0.53
Base Orientation	0.155 deg	0.092	0.65
Base X	3.5 mm	2.7	3.97
Base Y	6.9 mm	8.5	2.22
End-Effector X	3.8 mm	11	1.74
End-Effector Y	22 mm	8.0	8.69

To illustrate the utilization of the platform, a case study was discussed, where the platforms were tasked with tracking a planar XY end-effector with two actuated joints while having an uncontrolled base. The use of the IDSE to select reference trajectory, generate joint trajectories, and optimize PID controllers was demonstrated. The results of the case study were statistically analyzed.

7. REFERENCES

1. C.R. Carignan, D.L. Akin, (2000). The reaction stabilization of on-orbit robots, *IEEE Control Systems Magazine* 19–23

2. Menon, C., Busolo, S., Cocuzza, A., Aboudan, A., Bulgarelli, A., Bettanini, C., Marchesi, M. and Angrilli, F. (2007). Issues and solutions for testing free-flying robots. *Acta Astronautica* 60 (12), 957-965

3. H. Sawada, K. Ui, M. Mori, (2004). Micro-gravity experiment of a space robotic arm using parabolic flight, *Advanced Robotics 18 (3)* 247–267

4. C. Menon, A. Aboudan, S. Cocuzza, A. Bulgarelli, F. Angrilli, (2005). Free-flying robot tested on parabolic flights: kinematic control, *Journal of Guidance, Control, and Dynamics 28 (4)* 623–630

5. H. Ueno, Y. Wakabayashi, Y. Ohkami, E. al, (2000). Ground testbed of a reconfigurable brachiating space robot, *Advanced Robotics 14 (5)* 355–358

6. Wenfu Xu, Bin Liang, Yangsheng Xu (2011). Survey of Modeling, Planning, and Ground

Verification of Space Robotic Systems, Acta Astraunautica 68, 1629-1649

7. H.A. Fujii, K. Uchiyama, (1996). Ground-based simulation of space manip- ulators using test bed with suspension system, *Journal of Gui- dance, Control, and Dynamics 19 (5)* 985–991

8. Wenfu Xu, Bin Liang, Dai Gao, and Yangsheng Xu (2010). A Space Robotic System Used for On-Orbit Servicing in the Geostationary Orbit, In: *Proceedings of 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan

9. Smith, D.A., Martin, C., Kassebom, M., Petersen, H., Shaw, A., Skidmore, B., Smith, D., Stokes, H. and Willig, A. (2004). A Mission to Preserve the Geostationary Region. *Advances in Space Research* 34, 1214 – 1218

10. Rebele, B., Krenn, R. and Schäfer, B. (2002). Grasping Strategies and Dynamic Aspects in Satellite Capturing by Robotic Manipulator. In: *Proceedings of 7th ESA Workshop on Advanced Space Technologies for Robotics and Automation 'ASTRA 2002'*, ESTEC, Noordwijk, The Netherlands

11. V. Yu. Rutkovskii, V. M. Sukhanov, and V. M. Glumov (2013). Some Issues of Controlling the Free-flying Manipulative Space Robot, *Automation and Remote Control*

12. Marchesi, M., Angrilli, F. and Bettanini, C. (2001). On Ground Experiments of Free-flyer Space Robot Simulator in Intervention Missions. In: *Proceedings of 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space 'i-SAIRAS 2001'*, St-Hubert, Quebec, Canada

13. Yoshida, K. (2003). Engineering Test Satellite VII Flight Experiments For Space Robot Dynamics and Control: Theories on Laboratory Test Beds Ten Years Ago, Now in Orbit. *The International Journal of Robotics Research* 22 (5), 321 – 335

14. Schwarz, J., Peck, M. and Hall, C. (2003). Historical Review of Air-Bearing Spacecraft Simulators, *Journal of Guidance, Control and Dynamics* 26 (4), 513 - 522

15. Papadopoulos, E., Paraskevas, I.S., Flessa, T., Nanos, K., Rekleitis, G. and Kontolatis, I. (2008). The NTUA Space Robot Simulator: Design & Results. In *Proc. 10th ESA Workshop on Advanced Space Technologies for Robotics and Automation 'ASTRA 2008'*, ESTEC, Noordwijk, The Netherlands

16. Tomasz Rybus, Janusz Nicolau-Kukliński, Karol Seweryn, et al. (2013), New Planar Air-bearing Microgravity Simulator for Verification of Space Robotic Numerical Simulations and Control Algorithms. In: *Proceedings of 12th Symposium on Advanced Space Technologies in Robotics and Automation ASTRA 2013*

17. Marc A. Ullman, Robert H. Cannon, Jr., Stephen M. Rock (1993). A Modular System Architecture for Multi-Manipulator, Free-Flying Space Robots. In: *Proceedings of the 1993 International Symposium on Intelligent Control*, August 1993, Chicago, Illinois, USA

18. S. Matunaga, K. Yoshihara, T. Takahashi, S. Tsurumi, K. Ui, (2000). Ground experiment system for dual-manipulator-based capture of damaged satellites. In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 1847–1852. Takamatsu, Japan

19. Tatiana F.P.A.T. Pazelli, Marco H. Terra, Adriano A.G. Siqueira (2011). Experimental Investigation on Adaptive Robust Controller Designs Applied to a Free-Floating Space Manipulator, *Control Engineering Practice* 19, 395-408

20. Yoji Umetani and Kazuya Yoshida, (1989). Resolved Motion Rate Control of Space Manipulators with Generalized Jacobian Matrix, *IEEE Transactions on Robotics and Automation,* Vol 5, No. 3

21. Ahmad A. Mahfouz, Mohammed M. K., Farhan A. Salem, (2013). Modeling, Simulation and Dynamics Analysis Issues of Electric Motor, for Mechatronics Applications, Using Different Approaches and Verification by MATLAB/Simulink, *International Journal of Intelligent Systems and Applications,* 05, 39-57

22. Fuhai Zhang, Yili Fu, Lei Hua, Hongwei Chen, Shuguo Wang, Bin Guo, (2012). Point-to-point Planning for Free-floating Space Manipulator with Zero-disturbance Spacecraft Attitude, In: *Proceeding of the IEEE International Conference on Information and Automation,* Shenyang, China