LOCALIZATION AND FUEL MANAGEMENT TECHNIQUES FOR THE NTUA SPACE SERVICER EMULATOR SYSTEM

Thaleia Flessa [(1)], Iosif S. Paraskevas [(2)], Yiorgos Rekleitis [(3)], Evangelos Papadopoulos [(4)]

National Technical University of Athens (NTUA), Mechanical Engineering Dept., Control Systems Laboratory (CSL)
9 Heroon Polytechneiou Str., 17580 Zografou, Athens, Greece
Emails: [(1)]tflessa@mail.ntua.gr, [(2)]isparas@mail.ntua.gr, [(3)]georek@central.ntua.gr, [(4)]egpapado@central.ntua.gr

ABSTRACT

The development of space robotic systems presents design challenges and their commissioning requires thorough studies and experiments. A planar space emulator has been developed to emulate the operation of free-flying robotic servicers and consists of two robotic systems that hover over a flat granite table. They have been designed to be lightweight, modular, multi-functional and analogous to space servicers. This paper presents the design challenges of the initial robot and their influence on the design of the new robot, the localization techniques, including sensors used and preliminary experimental results, and the fuel management techniques. The last two issues are important to both emulators and space robots, and correlate directly to realistic space robotics challenges.

1. INTRODUCTION

The importance of space robots in satellite servicing, EVA assistance and removal of orbital debris is increasing. Autonomous robotic exploration in the form of orbital and planetary agents, and servicing and support systems are key steps in space exploration [1]. The robotic systems designed for fulfilling the above tasks require extensive theoretical, analytical and experimental task validation and are expensive to build. Technology demonstration flights such as JAXA's ETS-VII [2], DARPAS's Orbital Express [3] and the recent Robotic Refuelling Mission (RRM) [4], are usually expected to succeed at least partially, have minimum fault tolerance and reduced flexibility.

To demonstrate and validate space robot capabilities, space simulators (hardware or software) are used to emulate a space-like environment on earth. Common methods include neutral buoyancy facilities, gravity compensation mechanisms, parabolic flights and planar or rotational systems using air-bearings. Neutrally buoyant systems allow for 3D space motion simulation, but the existence of water resistance reduces the realism. Gravity compensation mechanisms negate the effects of gravity; however this approach has several limitations due to mechanism singularities and imperfections, and gravity loading of joints. Parabolic flights are severely limited by the cost and time available for the experiments. A different approach is a planar simulator, which negates the effect of gravity by employing a practically frictionless motion of the simulated robotic system on a horizontal plane. This method has the disadvantage of not being able to emulate 3D motions. Planar simulators using air-bearings are perhaps the most versatile and less expensive in comparison to other methods, and allow for repeated and thorough testing of control algorithms and verification of dynamics [5]. They require minimum preparation compared to other simulation methods and are easier to upgrade and adapt to alternative scenarios. Facilities that simulate a space-like environment have been developed in the USA (MIT, NASA, Stanford), Europe (U. of Padova, U. of Southampton) and Japan (Tokyo Institute of Technology, Tokyo University). The space emulator developed at the NTUA's CSL, is an air-bearing planar space emulator, consisting of two robots and a granite table. Its goal is to assist in studying the behaviour of free-flying robots in a 2D lab environment.

A challenging problem of the CSL emulator is the accurate detection of the attitude and position of the robot on the granite table. This is similar to the issue of attitude and position determination of satellites. Measurements from relative sensors (e.g. INS) or from absolute sensors (e.g. deep space radars) can be fused; a similar idea applies here. Various processing methods have been proposed in the literature, including the use of redundant information [6]. The calculation of the attitude is challenging because the accuracy is directly related to the use of trigonometric functions and complicated mathematical expressions, like [7] which uses an algorithm directly related to the motion of each relative sensor. Such approaches suffer in terms of execution time and are prone to round-off errors. The least squares method, [8], requires the use of computationally intensive inverse and transpose matrices. In this paper the method presented in [6], which uses few trigonometric functions and can be adapted, is demonstrated.

Another important aspect in space systems is the use of active actuators (thrusters, momentum exchange devices) for maintaining the desired attitude and for following desired trajectories. Momentum devices use electrical power provided by solar panels and stored in batteries, and is essentially unlimited. Thrusters require fuel; its consumption severely affects the lifespan and cost of a space system and is a critical parameter in space system design. It is necessary for a space emulator to emulate fuel constraints and also to use momentum exchange devices and thrusters similar to those in real space systems. This enables the investigation of fuel consumption and fuel minimization issues, using an optimized combination of the active actuating devices.

In this paper, we present the design of the NTUA space emulator and how the challenges encountered influenced the design of a second free-flying robot for use with the emulator, the methods used for robot localization and how they compare and simulation results from fuel consumption minimization techniques.

2. EMULATOR DESIGN

2.1. Specifications and Requirements

The planar emulator consists of a granite table of minimum roughness and a robot supported on three porous air-bearings, Fig. 1. The robot is capable of horizontal frictionless motion on the table, thus allowing for 2D emulation zero gravity in a laboratory environment. The novelty of this configuration is that the robot is not only of low mass and completely self-contained but also it is composed of subsystems similar to those of a space system, therefore making the emulator more realistic. An additional second robot is currently under development, with the same design goals. The emulator is fully presented at [9].

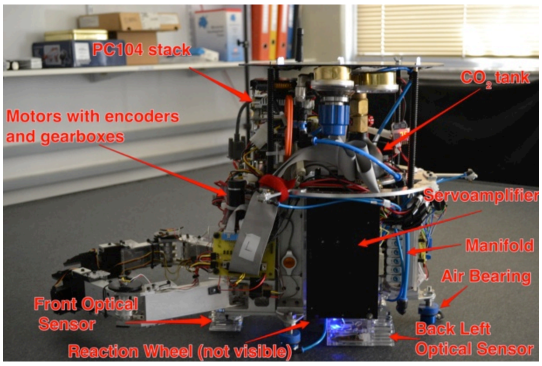


Figure 1. The robot hovering over the granite table.

The robot is mounted on top of the three air bearings and carries a CO_2 gas tank. The CO_2 gas flows through the air bearings and a 10 micron gas film between the floating system and the granite table is created, that develops a force normal to the surface, lifting the air bearing-mounted robot and cancelling the gravity force. A three-tone, special finishing granite table, with maximum anomaly height of 5μm was chosen. The table has a surface area of 2.2x1.8 m^2, standing on six adjustable height legs and the tilt is less than 0.02^O. The system has a dry weight of about 16.5 kg, well within the air bearings load limits. The full CO_2 tank increases the total weight (wet weight) to about 17 kg.

The robot is fully autonomous. Its propulsion autonomy is achieved by the on board CO_2 tank used to provide gas to the air bearings and to three couples of propulsion thrusters. The robot is also equipped with a reaction wheel to control the robot's angular momentum. Power autonomy is achieved using Li-Po batteries. The robot has two manipulators, each with two joints and one gripper. Each joint is actuated by a servoamplifier-driven Maxon DC servomotor, a gearbox and an encoder. An on-board PC104 provides computational autonomy and is responsible for controlling all the robot's functions (motors, thrusters, sensors, wireless communication with the overhead camera) and runs locally all control algorithms.

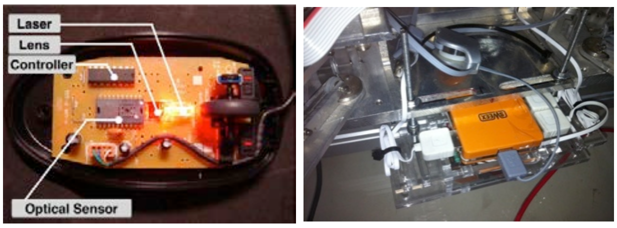
The NTUA simulator provides a low-cost, long duration and easily reconfigurable platform that allows the analytical and experimental validation of different control, dynamics, and planning schemes.

2.2. System Localization

The localization sensors and the actuators used are presented here. More details can be found in [9]:

(a) *Camera*: An overhead Matrix Vision mvBlueFOX-220 camera tracks robot position and orientation and is for absolute localization. LEDs are mounted on top of the robot at a predefined pattern and tracked by the camera. Images are sent to an off-board computer, where robot position and orientation is determined via image processing. The information is sent wirelessly via TCP/IP to the robot's processing unit (PC104). During calibration, various techniques were tried to reduce distortion; however, due to emulator requirements, the method in [10] was used. To evaluate the image processing, the PhaseSpace Mocap system was used.

(b) *Optical Sensors*: These are used for relative measurements. They use optical flow techniques; by comparing successive table surface pictures, they measure the differential displacements δx and δy at each sampling instant. Two sensors can produce four values per instant, and the three unknown parameters (x, y, and θ) can be calculated. These sensors have high sampling rate, compact size, good accuracy and low cost. However, their odometric error is accumulated over time, e.g. [11]. Three TRUST Retractable Laser Mini Mouse with 1600cpi nominal resolution optical sensors are mounted beneath the base of the robot at a distance of 2mm from the table in custom-made supports, Fig. 2. To characterize them the PhaseSpace Mocap was used.



(a) (b)

Figure 2. (a) Optical sensor and PCB (b) custom made support with HUB for receiving the data from sensors.

(c) *Thrusters:* The robot is actuated by six adjustable thrusters in three counter facing pairs, 120^0 apart. The thrusters use CO_2 at 7 bars and a pressure regulator was installed to reduce the tank pressure to 7 bar. Each thruster is actuated by a voltage actuated 2-way on-off solenoid valve. The electronic circuitry controls the gas flow using Pulse Width Modulation (PWM) allowing values of thrust in a continuous range while using the technology of on-off thrusters, which are used in actual space systems. One key parameter was determining the value of thrust from each thruster in regards to the PWM frequency. After experimentation the optimal frequency was found at 7 Hz, and the actuating force between 0 and 0.48N (depending on duty cycle).

(d) *Reaction wheel*: A reaction wheel directly driven by a torque controlled DC servomotor has been installed along the vertical axis. The reaction wheel provides an additional controllable rotational degree of freedom, decoupled from the thrusters. The reaction wheel was chosen to provide at a range of operating conditions the required torque to correctly orient the robot at a reasonable time. Motor saturation may still occur when high angular accelerations are needed in the presence of high wheel angular velocity. The reaction wheel selected weights 1.69 kg and has inertia of 0,0013kgm^2.

2.3. Challenges and Troubleshooting

During the robot development several design and operation challenges have arisen: complexity, interaction of many components, confined volume and limited budget. These challenges must be tackled, while maintaining and improving the quality and fidelity of the emulator. An update of the most critical ones, which have been presented in [9], follows.

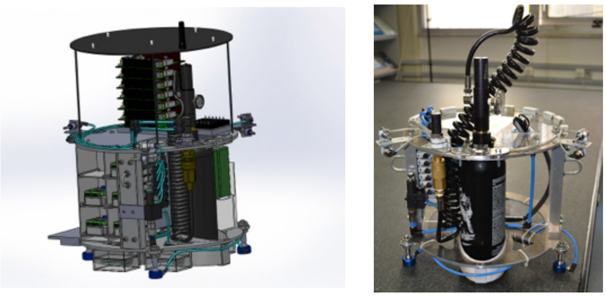
(a) For the image processing to determine the position and orientation of the robot it should detect at least 3 LEDs. However in some areas the camera fails to detect some of them. Therefore, a pattern with 3 LEDs, collinear to the X-axis direction of the Robot Body Coordinate System (RBCS) is selected and a 4th LED off the X-axis line has been installed for redundancy. This way, it is easier and computationally efficient to determine the attitude and position of the robot, a method verified by all preliminary tests. Also important is the minimization of non-LED "shines", which may confuse the image processing. To counter this, a black mat PVC board has been installed on top of the robot.

(b) The confined volume and the complex interconnections between the subsystems posed several difficulties during the development of the robot. There was no particular attention in the requirements for cabling and piping, e.g. specialized routings on the chassis. Due to this fact the installation of cables and CO_2 pipes was not a trivial task and it was a common problem during experiments, for subsystems to stop functioning properly, due to tensions to the cables.

(c) During integration a large EMI noise which affected the readings of the encoders was present. The source of the noise has been identified in the switching DC/DC converter necessary to provide the 24V DC for motors and solenoid valves. The original DC/DC converter had limited EMI filtering and for this reason a new converter is used (XPPower QSB35024S24), which complies with the standard EN55022, level A and is suitable for our requirements.

2.4. New Robot Design

A second robotic system is under development for use with the existing emulator, see Fig 3. A number of requirements have been set for its design: (i) Similar dimensions and inertia characteristics as the first one, (ii) Provision for similar subsystems as the first one, but with reduced size and weight when possible, (iii) Provision for routing the cabling and piping through tracks on the chassis and (iv) Low cost design which can be completed in the laboratory's workshop using Off-The-Shelf parts. The design should allow various operational scenarios in cooperation with the first robot, such as: docking procedures, satellite servicing emulation, swarm operations. Fig. 3a presents the CAD design, and Fig. 3b presents the current appearance of the chassis with the CO_2 subsystem installed (which includes both the hovering subsystem and thrusters). Note that in contrast to the first robot, the subsystems' preliminary selections on the new robot include: smaller pressure regulator, digital servoamplifiers, fewer PC104 boards and use of Matlab's xPC Target software.



(a) (b)

Figure 3. (a) The new robot in CAD (b) current form.

3. LOCALIZATION

The emulator uses two different localization sensors, an overhead camera for absolute measurements and on-board optical sensors for relative measurements. A third sensor type, the PhaseSpace Mocap system, was used to calibrate and characterize the other. A number of challenges must be taken into account: (a) correspondence of raw data to the actual motion (geometric problem), (b) sensor accuracy (noise interference) and (c) use of the redundant information in in reducing errors (data fusion). Next, the algorithms tested and preliminary experimental results from both sensors are presented and discussed.

3.1. Theory for localization problems

For the planar case, the position (x, y) and angle (θ) with respect to the World Coordinate System (WCS) must be determined. For the case of the absolute sensor (camera) 3 LEDs are required, unless they can be uniquely identified, in which case 2 LEDs are enough. In the case of the relative sensors (optical sensors) only 2 are required. Additional LED or sensors increase the redundancy of the system.

i) *Geometry for the absolute sensor.* The pattern of the LEDs on top of the robot is known, and image processing recognizes the pattern's position and attitude. The image processing technique used is beyond the scope of this paper.

ii) *Geometry for the relative sensors.* The measurements correspond to incremental displacements of each sensor at sampling instant "k" with respect to each sensor's body reference frame at sampling instant "k-1". On average, an optical sensor can attain sampling frequencies higher than 100Hz. The optical sensors used have a sampling frequency of 125 Hz, while the motion of the robot during the sampling interval does not exceed a few mm per second. Therefore, it is reasonable to assume that the incremental displacement of each sensor "si" between two consecutive sampling instants, "k-1" to "k", is very small and corresponds to a vector with origin the origin of the Sensor Body Coordinate System at instant "k-1", CS{si, k-1} and it can be mapped to the WCS as

$$ {}^{w}\delta\mathbf{s}_{si}(k) = {}^{w}\mathbf{R}_{si}(\theta_{k-1})\cdot {}^{si}\delta\mathbf{s}_{si}(k) \tag{1}$$

where θ_{k-1} is the attitude of the robot with respect to the WCS at sampling instant "k-1" and

$$ {}^{w}\mathbf{R}_{si}(\theta_{k-1}) = \begin{bmatrix} \cos(\theta_{k-1}) & -\sin(\theta_{k-1}) \\ \sin(\theta_{k-1}) & \cos(\theta_{k-1}) \end{bmatrix} \tag{2}$$

The initial conditions are known. Each sensor incremental displacement is added to the previous position of the sensor with respect to the WCS, $\mathbf{p}_{si}$, and the robot base position $\mathbf{c}$ is then derived from

$$\mathbf{c} = \sum_{i=1}^{n} w_i \cdot \mathbf{p}_{si} \quad \textit{with} \quad \sum_{i=1}^{n} w_i = 1 \tag{3}$$

iii) *Attitude Determination.* The attitude of the robot was assumed to be θ_{k-1}. To define the new attitude, a number of approaches have been proposed, see also [6]. The proposed method in [6], NM-2D, requires the detection of the points of a polygon. In the case of the CSL emulator the polygon is defined from the readings of the overhead camera. The orientation can be found by computing either (a) the angle between the initial and the rotated side or (b) the angle between the normal vector of the initial and the rotated side. The latter can be applied directly to 3D body attitude determination, where the normal vector refers to the normal vector of each plane of a polyhedron corresponding to the body.

An angle can be calculated for each sensor pairs, although theoretically one pair is enough. Any redundant information can be used to minimize noise. An option is the calculation of the angle for each pair and their mean value. For 3 sensors, see Fig. 4:

$$\varphi = \frac{\varphi_{12} + \varphi_{23} + \varphi_{31}}{3} \tag{4}$$

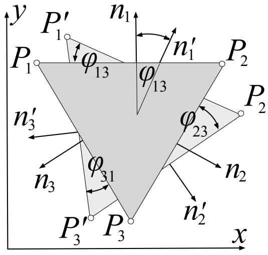


Figure 4. Two planar motion snapshots with three point position measurements

The difference between the alternatives with respect to the calculation, is the trigonometric function employed. The first method, uses the atan (or atan2) function,

$$\varphi_{ij} = \operatorname{atan2}\left(\frac{{}^{w}x_{sj,k} - {}^{w}x_{si,k}}{{}^{w}y_{sj,k} - {}^{w}y_{si,k}}\right), \quad i,j = 1,2,...,n \tag{5}$$

while the second method uses the arccosine,

$$\varphi_{ij} = \cos^{-1}\frac{(\tilde{\mathbf{p}}'_j - \tilde{\mathbf{p}}'_i)\cdot(\tilde{\mathbf{p}}_j - \tilde{\mathbf{p}}_i)}{\left\|\tilde{\mathbf{p}}'_j - \tilde{\mathbf{p}}'_i\right\|\left\|\tilde{\mathbf{p}}_j - \tilde{\mathbf{p}}_i\right\|}, \quad i,j = 1,2,...,n \tag{6}$$

Both Eqs. (5) and (6) can be used to find the accumulated change of angle of rotation, either by utilizing an initial position and attitude as a reference, or by finding the incremental change of angle with respect to the previous instant and adding this angle to the accumulated angle. Note that due to the way trigonometric functions are defined, one must be careful when using them. To this end, the following applies (the angles refer to the WCS)

$$\begin{aligned} &\textit{if} \quad \theta_k \cdot \theta_{k-1} < 0 \quad \textit{and} \quad \cos\theta_k < 0 \\ &\textit{then} \begin{Bmatrix} \delta\theta = \theta_k - \theta_{k-1} + 2\pi & \textit{for} \;\; \theta_k > 0 \\ \delta\theta = \theta_k - \theta_{k-1} - 2\pi & \textit{for} \;\; \theta_k < 0 \end{Bmatrix} \\ &\textit{else} \quad \delta\theta = \theta_k - \theta_{k-1} \end{aligned} \tag{7}$$

3.2. Experimental Results

To characterize the camera and the optical sensor performance, a number of characteristic motions were performed and the results are presented. For the purposes of this task, the robot was forced to move into a straight line and to follow a "Π" motion along the three sides of the granite table. In order to minimize contact with the surface, a specialized mechanical setup was used to ensure that the robot's motion follows a straight line. The overhead camera is located above the center of the granite table, the optical sensors below the robot's base creating a triangular pattern and the PhaseSpace cameras around the granite table.

(a) *PhaseSpace Mocap vs Optical Sensors*. The post-processed data are presented in Fig. 5. The errors for *x* direction have mean value 0.0024mm and standard deviation of 0.1120mm, for *y* direction -0.0022mm and 0.1874mm respectively, and for rotation angle 0.0011° and 0.1010° respectively.

(b) *Overhead Camera vs Optical Sensors*. The post-processed data are presented in Figs. 6-7 for two different motions. Note that due to the mechanical setup the rotation near the vertices could not be an exact arc of a circle. The errors for *x* direction have mean value -1.1320mm and standard deviation of 2.4044 mm, for *y* direction 0.7453mm and 5.1657mm respectively, and for rotation angle 0.1269° and 0.6575° respectively.

In the "Π" motion, although optical sensors follow the motion, small algorithmic/ hardware errors accumulate, and eventually become large. In Fig. 7a, the drift in sensors' measurements create an increase of the optical sensors' triangle during the motion.

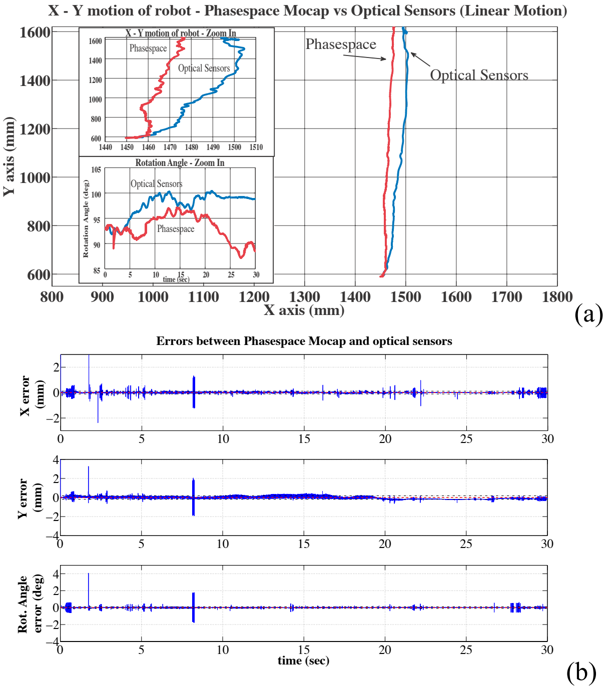


Figure 5. Use of PhaseSpace and Optical Sensors (a) X-Y position and (b) Errors (with red line the mean value).

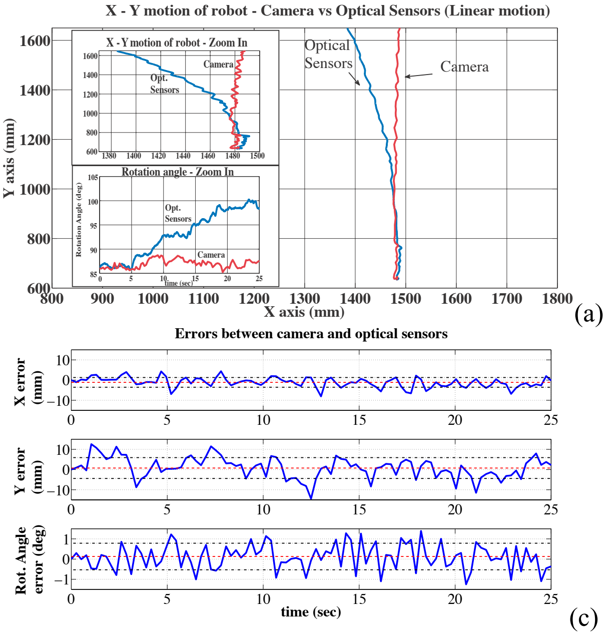


Figure 6. Camera and Optical Sensors (a) Robot X-Y position (b) Errors (red: mean black: stand. deviation).

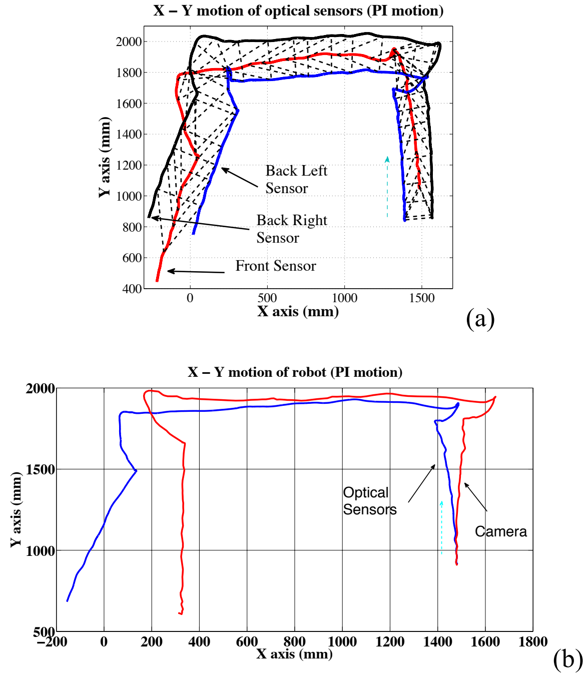


Figure 7. "Π" Motion from raw data (a) sensors' position and (b) robot's position.

3.3. Discussion on localization experiments

During these preliminary experiments, all sensors follow the motion of the robot. The PhaseSpace system is considered as the most accurate. It can be seen that the optical sensors, try to follow the same line, however after 4 to 5 seconds start to accumulate large errors. This is also true versus the camera system. In the more complicated "Π" motion the accumulation of errors are large after the second vertex. The errors in positions are a consequence of the errors in the calculation of rotation angles, due to sensors drift and to arithmetic errors. On the other hand, the overhead camera seems to follow the motion profile sufficiently.

The errors can be divided into two categories. The first category is the algorithmic errors, especially when rotations take place. The analysis of the data of optical sensors will be further improved by eliminating computational errors and use of filters like EKF. The second category is errors produced from hardware, which include the height of optical sensors from the surface, the mechanical setup which allows the execution of a straight line, the effects of velocity in accuracy of measurements and calibration errors among other. The experiments are still in progress in order to fully and accurately characterize all sensors and eliminate systematic errors for all sensors.

These experiments show that the most efficient combination of both sensors, is by using the optical sensors for small time intervals (e.g. 1-4 sec) and resetting their position and attitude via the camera readings. The exact parameters of the procedure and the required filtering is currently in progress.

4. FUEL CONSUMPTION

Reducing propellant consumption is important in space and also in the laboratory since it allows for a realistic emulation of the system, reduces the need for refilling the CO_2 tank and the experiment setup time is faster. The question then arises as to how we can combine thrusters and reaction wheel to reduce fuel consumption.

4.1. Theory

The robot is modelled as a free-flying cylindrical base with two two-joint manipulators, and is actuated by six thrusters in three counter facing pairs, 120^0 apart and a reaction wheel, see Fig. 8. Fig. 8 also shows the robot body fixed frame (x_b, y_b) and the WCS frame, (x,y). The manipulators are not shown for simplicity. The directions of the forces $F_1,\ldots,F_6$ are opposite to thruster plume direction.

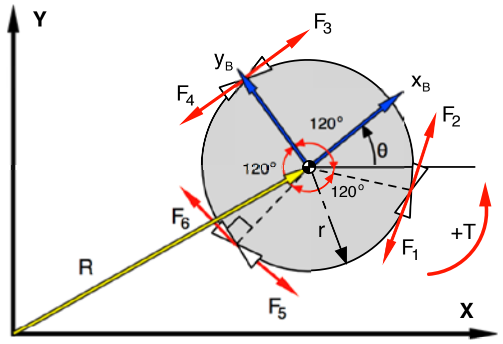


Figure 8. Free body diagram of the robot's base.

The manipulators are considered braked and the equations of motion for the robot's base are:

$$\sum f_x = m_b \cdot \ddot{R}_x, \sum f_y = m_b \cdot \ddot{R}_y, \sum M_z = J \cdot \ddot{\theta} \quad (8)$$

where m_b is the base's mass and J is the base's moment of inertia. Eq. (8) can be rewritten as:

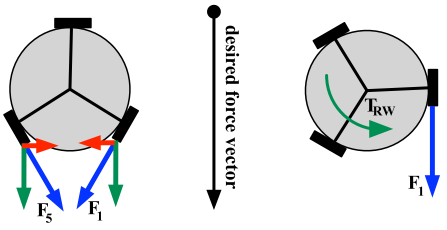
$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

$$\mathbf{A} = \begin{pmatrix} \cos\theta_1 & -\cos\theta_1 & -\sin\theta & \sin\theta & \cos\theta_3 & -\cos\theta_3 & 0 \\ -\sin\theta_1 & \sin\theta_1 & \cos\theta & -\cos\theta & \sin\theta_3 & -\sin\theta_3 & 0 \\ -r & r & -r & r & r & -r & 1 \end{pmatrix} \quad (9)$$

$$\mathbf{x} = \begin{pmatrix} f_1 & f_2 & f_3 & f_4 & f_5 & f_6 & T \end{pmatrix}^T$$

$$\mathbf{b} = \begin{pmatrix} m_b\ddot{R}_x & m_b\ddot{R}_y & J\ddot{\theta} \end{pmatrix}^T$$

where $\theta_1 = 30^0 - \theta,\ \theta_3 = 30^0 + \theta,$ T is the reaction wheel torque and $f_i > 0,\ i = 1,\ldots,6$ are the thrusters forces. The rank of $\mathbf{A}$ is 3, the linear independent solutions are three. This is expected, because the robot has additional thrusters than the minimum required for moving and rotating, and the reaction wheel provides an additional controllable rotational degree of freedom, decoupled from the rest. Fig. 9a shows that the horizontal components of the thruster forces counteract each other, while the components in the direction of the desired force vector move the robot.



without reaction wheel
(a)

with reaction wheel
(b)

Figure 9. Force generation models.

From Eq. (9):

$$\left.\begin{aligned} \sum f_x &= f_1 \cos\theta_1 + f_5 \cos\theta_3 \\ \sum f_y &= -f_1 \sin\theta_1 + f_5 \sin\theta_3 = 0 \\ \sum M_z &= -rf_1 + rf_2 = 0 \end{aligned}\right\} \theta = 0^0,\ f_1 = f_5 \quad (10)$$

This loss of thrust and fuel, is due to the existence of a null space in matrix $\mathbf{A}$, $null(\mathbf{A}) = 4$. Comparing with Fig. 9b, it is obvious that minimum fuel consumption should occur when a single thruster parallel to the desired force vector is used and the reaction wheel counteracts the undesired torque. From Eq. (9):

$$\begin{gathered}\sum f_x = f_1, \sum f_y = 0 \Rightarrow \cos\theta_1 = 30^0 - \theta \rightarrow \theta = 30^0 \\ \sum M_z = -rf_1 + T = 0\end{gathered} \tag{11}$$

The robot must first turn by $\theta = 30^0$ with the reaction wheel, therefore without using fuel and then move parallel to the desired force vector using only thruster 1 and the wheel. In the general case, the robot should first turn using the wheel so that the single thruster will be to the closest parallel direction to the desired force vector. A drawback of using this methodology is that the maximum total force that can be applied to the base is the maximum force that a single thruster can generate. Therefore, if maximum acceleration is needed, some fuel must be spent without providing full useful thrust.

4.2. Simulation results

The simulation is using a PD model based closed loop control algorithm and a given desired trajectory. The robot's dynamic model (cylindrical base and two manipulators each with two joints) using Lagrange equations and omitting the gravitational terms is:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{F} \tag{12}$$

$\mathbf{H}$ is a 10x10 inertia matrix and is positive definite and $\mathbf{C}$ is a 10x1 vector containing non-linear velocity terms. The robot movement is 2D and it has 7 DOF:

$$\mathbf{q} = \left[R_x, R_y, R_z, d_x, d_y, d_z, th_{11}, th_{12}, th_{21}, th_{22}\right]^T \tag{13}$$

$\mathbf{R} = \left[R_x, R_y, R_z = 0\right]^T$ is the base's centre of mass position vector with respect to the reference frame (x, y), $\mathbf{d} = \left[d_x = 0, d_y = 0, d_z = \theta\right]^T$ is the rotation of the base and $th_{ij}, i = 1,2, j = 1,2$ is the angle of each joint. The vector $\mathbf{F}$ is a 10x1 forces and torque vector:

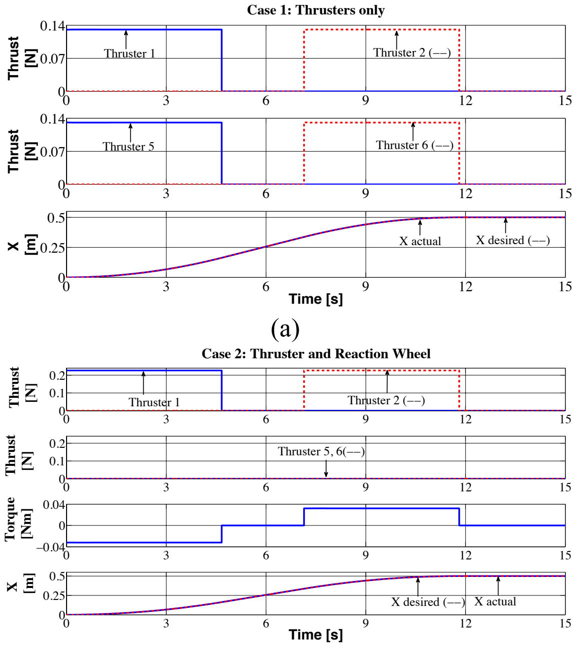
$$\begin{gathered}\mathbf{F} = \left[F_x, F_y, F_z, n_x, n_y, n_z, t_{11}, t_{12}, t_{21}, t_{22}\right]^T \\ with\ F_z = 0,\ n_x = 0,\ n_y = 0\end{gathered} \tag{14}$$

The torques t_{ij} are applied by the joint motors and here all variables related to the manipulators are zero, since we examine the base's translation. F_x, F_y are responsible for the translation and are actuated by the six thrusters. The torque n_z is actuated by the six thrusters and the reaction wheel. Appropriate constraints for the operational limits of the thrusters and the reaction wheel's motor and the avoidance of wheel saturation are taken into account. Next, we compare the two actuation methods, seen in Fig. 9: Case 1 using only thrusters and Case 2 using one thruster with the wheel, with the same trapezoidal velocity profile:

$$x = 0.5m,\ \dot{x} = 0.07m/s,\ \ddot{x} = 0.015m/s^2 \tag{15}$$

equal acceleration and de-acceleration time $t_1 = 4.67s$, constant velocity time $t_2 = 2.48s$, total time of motion $t_f = 11.81s$. The robot follows the desired trajectory parallel to the x reference frame axis. In Case 1 the base's angle is $\theta = 0^\circ$ and in Case 2 $\theta = 30^0$, Fig. 10. All other initial conditions are set to zero. The criterion for measuring fuel consumption is the total thrusters' impulse, Fig. 11:

$$I = \sum_{i=1,..,6} \int_0^{t_f} f_i\, dt \tag{16}$$



(a)

(b)

Figure 10. (a) Case 1 and (b) Case 2.

Higher thrust forces appear during translational motion with acceleration and no thruster/wheel activation occurs in the absence of acceleration, which is expected since no friction or gravitational forces act on the robot. In both cases, the robot follows the desired trajectory. In Case 1, both thrusters 1, 5 during acceleration are equally activated at 0.13 N and similarly 2, 6 during de-acceleration, whereas the wheel is not activated at all. In Case 2, only thruster 1, parallel to the desired trajectory, is activated at 0.225 N and simultaneously the wheel at 0.038 Nm, and similarly thruster 2 and the wheel during de-acceleration. Thrusters 3 and 4 are not activated in any of the two cases. In both cases the disturbances observed at the other degrees of freedom (along the y axis and at the joints) where in the order of 10^{-5} and 10^{-7}. The developed thrust and torque are well within the operational limits of the actuators and no wheel saturation occurred. Comparing the impulse between the two cases, the total impulse in Case 1 is 2.430 N s and in Case 2 is 2.103N s, a difference of 0.327 N s. The use of reaction wheel reduces fuel consumption, while achieving the desired trajectory.

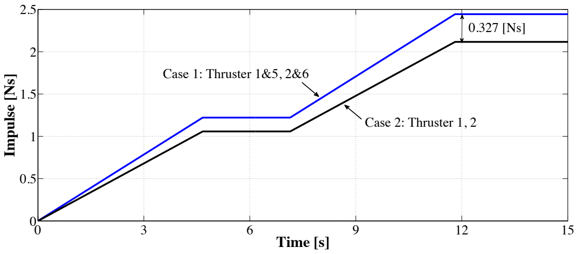


Figure 11. Impulses for both cases.

5. CONCLUSIONS

This paper presented the design of the NTUA space emulator and the challenges encountered during the development. These influenced the design of the second free-flying robot for use on the emulator. Localization and fuel consumption strategies are important issues to space emulators. Preliminary experiments of the installed localization sensors validate their performance. The camera can successfully track the motions of the robot, while the optical sensors accumulate errors over time that must be regularly corrected. Possible solutions to the error accumulation are discussed, however this is still a work in progress. The fuel consumption optimization strategy is critical, for minimizing CO2 use and increasing the experiment time. The minimization technique by combing thrusters and the reaction wheel was validated by simulation, with experiments on the emulator to follow.

6. ACKNOWLEDGEMENTS

Part of this research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitos II. Investing in knowledge society through the European Social Fund. We thank Stelios Andreou, CSL MSc Student, for his valuable assistance during the experiments.

7. REFERENCES

1. Hufenbach, B, et. al. (2011). *The Global Exploration Roadmap*. 62nd International Astronautical Congress, Capetown, SA.

2. Yoshida, K. (1999). *Space Robot Dynamics and Control: To Orbit, From Orbit, and Future.* Robotics Research, the Ninth International Symposium, ed. by J. Hollerbach and D. Koditschek, Springer, pp. 449-456.

3. Ogilvie, A., Justin Allport, Michael Hannah & John Lymer, (2008). *Autonomous Satellite Servicing Using the Orbital Express Demonstration Manipulator System*. Proc. of the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS '08).

4. http://ssco.gsfc.nasa.gov/

5. Schwartz, J., Peck M. & Hall C. (2003). *Historical Review of Air-Bearing Spacecraft Simulators.* Journal of Guidance, Control, and Dynamics, vol. 26, no. 4, pp. 513-522, 2003.

6. Tegopoulou, A., & Papadopoulos, E. G. (2011). Determination of Rigid-Body Pose from Imprecise Point Position Measurements. *IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)* (pp. 2922-2927). San Francisco, CA, USA.

7. Bonarini, A., Matteucci, M., & Restelli, M. (2005). Automatic Error Detection and Reduction for an Odometric Sensor based on Two Optical Mice. *IEEE Int. Conference on Robotics & Automation (ICRA)* (pp. 1675-1680). Barcelona, Spain.

8. Wu, P.-L., & Jeng, S.-L. (2010). Least Squares Approach to Odometry based on Multiple Optical Mouse Sensors. *5th IEEE Conf. on Industrial Electronics and Applications* (pp. 1573-1578).

9. Papadopoulos, E., Paraskevas, I. S., Rekleitis, G., & Flessa, T. (2011). The NTUA Space Robotic Emulator: Design and Experiments. *nternational Conference on Intelligent Robots and Systems (IROS '11): Workshop on Space Robotics Simulation* (pp. 1-14). San Francisco, CA, USA.

10. Bailey, D.G. (2002). *A New Approach to Lens Distortion Correction*, Proceedings of Image and Vision Computing New Zealand, Auckland, New Zealand, pp. 59-64.

11. Minoni, U., & Signorini, A. (2006). *Low-cost optical motion sensors: An experimental characterization,* Sensors & Actuators, 128(2), 402-408.