A Novel Approach to the Simulation of On-Orbit Rendezvous and Docking Maneuvers in a Laboratory Environment Through the Aid of an Anthropomorphic Robotic Arm

Andrea Antonello*, Francesco Sansone*, Alessandro Francesconi*, Ruggero Carli†, Andrea Carron†
* Centre of Studies and Activities for Space (CISAS), Via Venezia 15, 35131 Padova, Italy.
{andrea.antonello.it, ffsansone}@gmail.com, alessandro.francesconi@unipd.it
† University of Padova, Department of Information Engineering, Via Gradenigo 6/B, 35131 Padova, Italy.
{carlirug, carronan}@dei.unipd.it

***Abstract*—Orbital robotics, due to the unfriendly environment (radiation, micro-gravity, thermal stresses, etc.) poses unique challenges to robot and robot algorithms, and sets the need for new and innovative autonomous systems. The design of servicing operations and devices is nowadays one of the most important research field in space robotics. Servicing operations range from regular inspection to the upgrade of components and re-fuelling. It is immediate to notice that, regardless of the operation to be carried out, the success is strictly linked to the way in which the chaser and the target satellites move and interact with respect to each other. The importance of relative motion for rendezvous and docking operations, calls for an approriate laboratory facility able to reproduce orbital conditions. This can be achieved only with a robotic structure that simulates the target and chaser's kinematics and dynamics.**

In this paper, a complete approach to the problem is presented, from the kinematic analysis to the modelling of the impact. In particular, a spring-dashpot model was chosen for the contact simulation, and a virtual-force control system has been adopted. Then, by considering the system's stability, we extracted the analytical expressions that link the performances of the facility with the range of orbital systems that can be simulated.

Furthermore, with the aid of a SimMechanics® numerical model, we inspected the performances of three different control strategies for the movimentation of the robot.

I. Introduction

The increasing number of human objects in space has laid the foundation of a novel class of orbital missions for servicing and maintenance. There are currently very few facilities able to simulate relative motion between orbiting objects: one of the most important is probably DLR's European Proximity Operations Simulator (EPOS) [1]. The current facility, built in 2009, consists of a 25 m long testing site with in two industrial anthropomorphic robots that can reproduce docking and berthing scenarios, taking into account dynamic contacts, gravity and even sunlight illumination for utmost realistic simulations.

The main goal of this paper is to provide a guideline for the development of a robotic manipulator for the simulation of close approach orbital maneuvers, with particular attention to docking and capture. The interesting aspect of this structure will be the integration of a force sensing device that will take into account both disturbances and contact forces between the objects. Through a dedicated algorithm, the system is able to compute in real time the consequences of these inputs in terms of trajectory modifications, which are then fed to the hardware in the loop (HIL) control system. Moreover, the software governing the manipulator can be commanded to perform active maneuvers and relocation: as a consequence, this structure can be used as the testing bench for any attitude modification system, providing a reliable, real time simulation of the orbital scenario. Furthermore, with the aid of dynamic scaling laws, the potentialities of the facility can be exponentially increased: the simulation environment is not longer bounded to be as big as the robot workspace, but could be several orders of magnitude bigger, allowing for the reproduction of otherwise preposterous scenarios in a laboratory environment. The manipulator presented in this paper will serve as the main testing facility for the reproduction and the verification of theoretical and numerical analysis at CISAS research center (Padova, Italy).

II. Manipulator Analysis

A. Preliminary design

For this particular application the 6 DOF anthropomorphic configuration seems to be the best choice in order to obtain the maximum dexterity. In this structure, the first 3 joints provide the positioning, whereas the remaining 3 take care of the attitude. Since the robot will not be an off-the-self product, a custom end effector was designed (Fig. 1), aiming to limit singular configurations and to maximize the angular range.

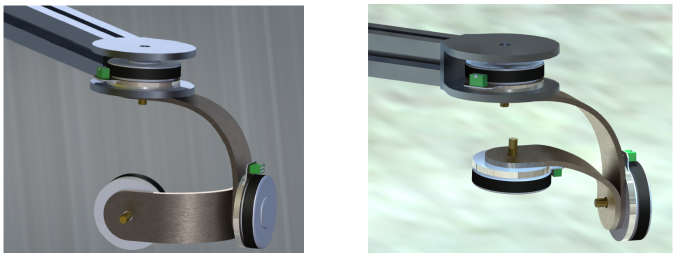


Figure 1: *End effector* design.

Moreover, by designing the joints such that the last three joint axes intersect at a point, it was possible to use Piper's simplification [2]: this led to substiantial easing in the kinematics and dynamics analysis. Among the requirements that need to be satisfied in this project, there is the workspace: the manipulator, in fact, must have sufficient dexterity in a cube whose volume is at least 0.7 m × 0.7 m × 0.7 m (which can be further extended with the use of a rail).

After an iterative process (comprising dexterity, workspace, stiffness and structural analyses), a preliminary design of the robot was performed, yielding the sizing parameters presented in the table.

	link 1	*link* 2	*link* 3	*link* 4	*link* 5	*link* 6
length [m]	0.7	0.7	0.6	/	/	/
mass [kg]	1.05	1.71	1.46	0.23	0.19	0.21

B. *Kinematics analysis*

1) Differential kinematics: The kinematic analysis of the system is performed using differential kinematics, which provide the relationship between joint velocities and end effector velocities (both linear and angular).

$$\begin{bmatrix} \mathbf{v} \\ \omega \end{bmatrix} = \begin{bmatrix} \mathbf{J}_P \\ \mathbf{J}_O \end{bmatrix} \cdot \begin{bmatrix} \dot{\mathbf{q}} \end{bmatrix} \tag{1}$$

Where $\mathbf{J}_P$ and $\mathbf{J}_O$ are both 3×6 matrices. The derivation of the Jacobian can be accomplished using several methods [3]. In order to extract $\dot{\mathbf{q}}$, we can invert the equation:

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\bar{q}) \cdot \mathbf{v} \tag{2}$$

From this vector, since $\mathbf{v}$ is known from the trajectory planning, we can finally obtain the joint variable position using an integration. However, the initial position $\mathbf{q}(t = 0)$ needs to be known in order to start the integration. This value can be obtained, for example, with one of the IK methods available (for example, using Piper's solution).

C. *Dynamics analysis*

1) Euler-Newton: Euler-Newton approach is based on the balance of all the forces and torques acting on the generic link of the manipulator. The solution of this problem is well suited for a recursive approach. Following Luh-Walker [4] notation, the *outward* and the *inward* iteration blocks can be synthetized as follows:

$$\begin{cases} {}^{i+1}\dot{w}_{i+1} &= {}^{i+1}_{i}R\,{}^{i}\dot{w}_i + {}^{i+1}_{i}R\,{}^{i}w_i \times \dot{\theta}_{i+1} + \ddot{\theta}_{i+1}\,{}^{i+1}\hat{k}_{i+1} \\ {}^{i+1}\dot{v}_{i+1} &= {}^{i+1}_{i}R\,[\,{}^{i}\dot{v}_i + {}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{i+1})\,] \\ {}^{i}\dot{v}_{C_i} &= {}^{i}\dot{v}_i + {}^{i}\dot{\omega}_i \times {}^{i}P_{C_i} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{C_i}) \\ F_i &= m\,\dot{v}_{C_i} \\ N_i &= {}^{C}_{i}I\dot{\omega}_i + \omega_i \times {}^{C}_{i}I\omega i \end{cases} \tag{3}$$

$$\begin{cases} {}^{i}f_i &= {}^{i}F_i + {}_{i+1}^{i}R^{i+1}f_{i+1} \\ {}^{i}n_i &= {}^{i}N_i + {}_{i+1}^{i}R^{i+1}n_{i+1} + {}^{i}P_{C_i} \times {}^{i}F_i + {}^{i}P_{i+1} \times {}_{i+1}^{i}R^{i+1}f_{i+1} \\ \tau_i &= {}^{i}f_i^T\,\hat{Z}_i \end{cases} \tag{4}$$

2) Euler-Lagrange: Euler-Lagrange method is an energy based approach. With this technique, the equations of motion can be obtained in a systematic way independently of the reference frame. By choosing a set of generalized coordinates describing the link positions it is possible to write Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) - \left(\frac{\partial \mathcal{L}}{\partial \mathbf{q}}\right) = \tau \tag{5}$$

Where τ are the generalized forces acting on the links (mainly given by actuator torques and joint friction). Although the formulation is fairly easy to understand, its implementation is quite laborious. The Lagrange equation can be re-written as[1]:

$$\tau = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \tag{6}$$

Where $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ represents the inertia matrix and is calculated with the following formula:

$$\mathbf{M}(\mathbf{q}) = \sum_{i=1}^{n} (m_i \mathbf{J}_P^{i\,T} \mathbf{J}_P^{i} + \mathbf{J}_O^{i\,T} \mathbf{R}_i \mathbf{I}_i \mathbf{R}_i^{T} \mathbf{J}_O^{i}) \tag{7}$$

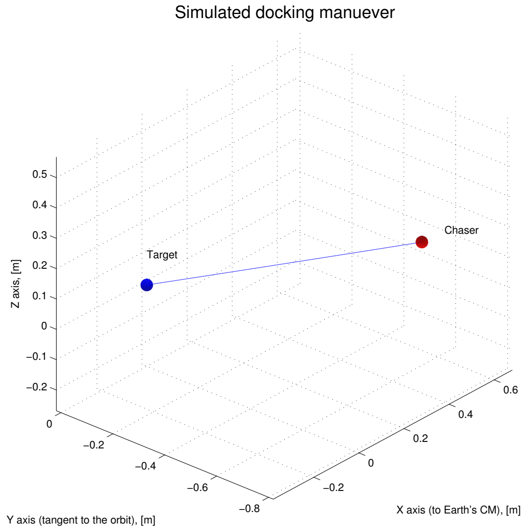


Figure 2: Simulated docking manuever

The term $\mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ accounts for centrifugal and Coriolis terms:

$$\mathbf{V}(\mathbf{q}) = \dot{\mathbf{M}}(\mathbf{q})\dot{\mathbf{q}} - \frac{1}{2}\frac{\partial}{\partial q_i}(\dot{\mathbf{q}}^{\mathbf{T}}\mathbf{M}(\mathbf{q})\dot{\mathbf{q}}) \tag{8}$$

Finally, the term $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^{n \times 1}$ accounts for the gravity compensation and is easily obtained from the derivation of the potential energy:

$$\mathbf{G}(\mathbf{q}) = \sum_{i=1}^{n} \frac{\partial \mathcal{U}}{\partial \mathbf{q_i}} \tag{9}$$

where:

$$\frac{\partial \mathcal{U}}{\partial q_i} = -\sum_{j=1}^{n} m_i \mathbf{g_0} \check{\mathbf{J}}_{P_i}^{j}(\mathbf{q}) \tag{10}$$

III. Orbital Trajectory Analysis

In close approach maneuvers, generally, one object (the target) is passive and non-maneuvering, whereas the other (the chaser), is active and trying to approach the target. With the aid of Clohessy Wilthshire's expressions (Eq. 11), it is possible to describe the relative motion of the chaser in a target-centerd frame. If the orbit is circular ($\mathbf{V} \cdot \mathbf{r}_0 = 0, \quad h = \sqrt{\mu\, r_0}$), then [5]:

$$\begin{cases} \delta\ddot{x} - 3\dfrac{\mu}{r_0^3}\delta x - 2\sqrt{\dfrac{\mu}{r_0^3}}\delta\dot{y} &= \dfrac{F_x}{m_c} \\ \delta\ddot{y} + 2\sqrt{\dfrac{\mu}{r_0^3}}\delta\dot{x} &= \dfrac{F_y}{m_c} \\ \delta\ddot{z} + \dfrac{\mu}{r_0^3}\delta z &= \dfrac{F_z}{m_c} \end{cases} \tag{11}$$

These are called Clohessy-Wiltshire (CW) equations and, with an integration, the velocity and the position equations can be obtained. By using compact notation, the instantaneous position and velocity, $\delta\mathbf{r}(t)$ and $\delta\mathbf{v}(t)$, can be written as follows, with $\delta\mathbf{r}_0$ and $\delta\mathbf{v}_0$ being the initial conditions and $\mathbf{\Psi}_{ii}(t) \in \mathbb{R}^{3\times 3}$ being orbit and time dependent matrices:

$$\begin{Bmatrix} \delta\mathbf{r}(t) \\ \delta\mathbf{v}(t) \end{Bmatrix} = \begin{bmatrix} \mathbf{\Psi}_{rr}(t) & \mathbf{\Psi}_{rv}(t) \\ \mathbf{\Psi}_{vr}(t) & \mathbf{\Psi}_{vv}(t) \end{bmatrix} \cdot \begin{Bmatrix} \delta\mathbf{r}_0 \\ \delta\mathbf{v}_0 \end{Bmatrix} \tag{12}$$

[1] In the term τ, all the nonlinear components of the system are momentarily ignored.

The reference docking manuever analyzed in this paper is based on a rendezvous approach between a target and chaser satellites which are on the same, circular 300 km LEO orbit, with a relative distance[2] of $\mathbf{\Delta x}$ = [0.3 -0.8 0.3] m and a null relative initial velocity. Suppose we want to complete the manuever in 6 s: using a two-impulse approach technique, Eq. 12 yields the initial and final $\mathbf{\Delta v}$ burns. Once the trajectory was defined, the $\delta\mathbf{r}(t)$ and $\delta\mathbf{v}(t)$ vectors were used for the simulations presented in Section VI. The trajectory is pictured in Fig. 2.

IV. Impact Modeling

When using a robotic facility for the simulation of orbital manuevers, it is fundamental to reproduce the contact dynamics. Given that the relative motion is simulated correctly with the aid of the CW expressions, the dynamic response of the satellites is strictly dependent on the inertial properties of the bodies: the simulated system, in general, will have different inertial properties and will consequently behave with its own, characteristic dynamics; since the robotic system cannot be subjected to drastic inertial modifications, a software strategy has to be implemented.

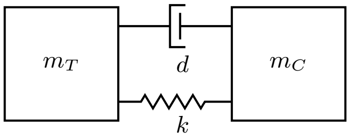


Figure 3: Spring-dashpot model.

First of all, it is mandatory to model the dynamics of the contact[3]. Over the years, several techniques have been proposed: one of the most used is certainly the spring-dashpot model [6], which models the contact between satellites as a parallel spring-damper system, as pictured in Fig. 3, where m_T and m_C are the target and chaser mass respectively. The differential equations describing the system are:

$$\begin{bmatrix} m_T & 0 \\ 0 & m_C \end{bmatrix} \cdot \begin{Bmatrix} \ddot{x_T} \\ \ddot{x_C} \end{Bmatrix} + \begin{Bmatrix} 1 \\ -1 \end{Bmatrix} \cdot f(t) = \mathbf{0} \tag{13}$$

where, if we define the relative position $x = x_C - x_T$, the force is expressed by:

$$f(t) = -kx - b\dot{x} \tag{14}$$

By using the equivalent mass[4] m, the system becomes:

$$m\ddot{x}(t) = kx + b\dot{x} \tag{15}$$

This fully defines the 1D approximated orbital behavior of the impact. In the laboratory case, however, the impact force will be characterized by different parameters, k_L and b_L: these parameters are hardware dependent, and respresent the facility's equivalent stiffness and damping (in the hypothesis of a 2^{nd} order approximation of the system). Referring to Fig. 4, the laboratory installation has been modeled with 4 main blocks: A represents the spring-dashpot impact model (which depends on the mockup of the docking system mounted on the end effector), B is a compliance system that will be further discussed, C is the stiffness of the force/torque (F/T) sensor and D is the manipulator (whose 2^{nd} order lumped parameters are a function of both the mechanics and of the control architecture). The compliance stiffness B is introduced in order to simplify the analysis: in a mechanical system, the dominant frequency of the contact is governed by the most compliant part. Thus, the insertion of a spring with a stiffness $k_{spr} << min\{k_C, k_{T/F}, k_R\}$ allows the system's overall stiffness to be approximated with that of the spring. This hypothesis is certainly true for $k_{T/F}$ (typical values for F/T sensors are on the $10^6 \div 10^7 N/m$ range); also, in this preliminary analysis, we suppose that the robot is infinitely rigid and presents no damping. As far as k_C is concerned, its value depends on the docking interface mockup and no valid approximation can be made upon it. The simplified system is represented in the circled area of Fig. 4. The force, in this case, has the following expression:

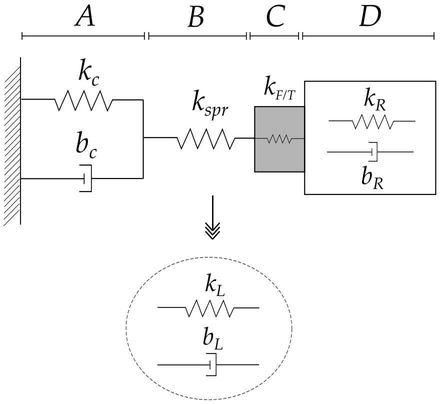


Figure 4: Lumped parameters laboratory model and its approximation.

$$f_L(t) = -k_L x - b_L \dot{x} \tag{16}$$

where:

$$k_L = k_C + k_{spr} \tag{17}$$

$$b_L = b_C \tag{18}$$

In order to simulate the actual orbital impact force (Eq. 14) in a laboratory environment (which is clearly subjected to a different dynamics, Eq. 16) it is mandatory to introduce a software artifice. With reference to [7], it is possible to implement a virtual force $f_V(t)$ in order to satisfy the following:

$$f(t) = f_L(t) + f_V(t) \tag{19}$$

Expliciting the lumped parameters, we have:

$$f(t) = -(k_L + k_V)x - (b_L + b_V)\dot{x} \tag{20}$$

Hence, the value of the virtual parameters can be computed as:

$$k_V = k - k_L \tag{21}$$

$$b_V = b - b_L + \epsilon \tag{22}$$

These parameters are computed upfront[5] and can be finely tuned in order for the laboratory dynamics to be an accurate representation of the orbital scenario.

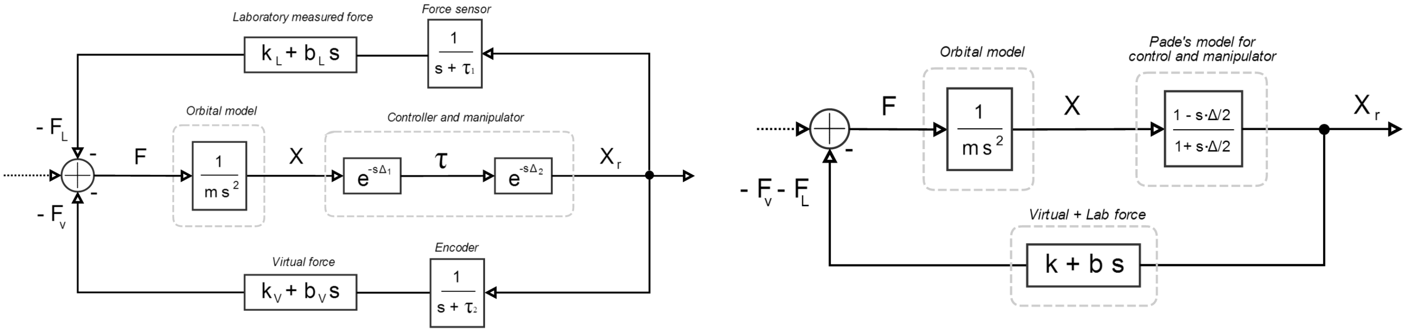
The concept of the contact simulation technique is represented in Fig. 5a: note that both the inverse dynamics loop and the actuation phase are simulated as delays (Δ_1 and Δ_2 respectively). The total delay block, with $\Delta = \Delta_1 + \Delta_2$, can be approximated with a rational

[2] Expressed in the CW frame.

[3] For this preliminary analysis, we will focus on a 1D model, which can then be extended to a more general 3D case.

[4] The equivalent mass is defined as: $m = \dfrac{m_C m_T}{m_C + m_T}$.

[5] Since the damping coefficient is influenced by the delay Δ of the simulation system, the parameter $\epsilon(\Delta)$ is introduced in Eq. 22.



(a) (b)

Figure 5: Virtual-force based control loop for contact dynamics simulation.

Padé's function in order to perform a frequency response analysis. In this case, we chose Padé's first order approximation [8]:

$$e^{-s\Delta} \approx \frac{1-\dfrac{\Delta}{2}s}{1+\dfrac{\Delta}{2}s} \tag{23}$$

Moreover, the sensors' transfer functions were inserted in the diagram: for a preliminary analysis, however, since the dynamics of the system is largely within their cut-off frequency, these blocks can be ignored. Hence, the overall system can be simplified as the one pictured in Fig. 5b, where Eq. 20 allowed for further compactness.

We then performed a frequency analysis of the system: its transfer function, for a generic input $U(t)$, is:

$$T(s) = \frac{X_r(s)}{U(s)} = \frac{1}{\dfrac{2+s\cdot\Delta}{2-s\cdot\Delta}ms^2 + bs + k} \tag{24}$$

The characteristic equation is:

$$\Delta\cdot ms^3 + (2m - b\cdot\Delta)s^2 + (2b - k\cdot\Delta)s + 2k = 0 \tag{25}$$

According to Routh-Hurwitz's criterion [9], the system is stable if:

$$\left\{ m - b\frac{\Delta}{2}, b - k\frac{\Delta}{2}, \left(m - b\frac{\Delta}{2}\right)\cdot\left(b - k\frac{\Delta}{2}\right) - km\Delta \right\} > 0 \tag{26}$$

Which yields the following two conditions:

$$\Delta < min\left\{\frac{2m}{b}, \frac{2b}{k}\right\} \tag{27}$$

$$4mb - 4mk\Delta - 2b^2\Delta + kb\Delta^2 > 0 \tag{28}$$

From these equations, by fixing one of the 4 parameters (m, b, k, Δ), 3D plots can be extracted for the system design; for example, by fixing Δ (which is known once the robot control architecture has been tested), we can get the minimum mass m required for simulation stability (Fig. 6), which is analytically defined as:

$$m > max\left\{\frac{b\Delta}{2}, \frac{2b^2\Delta - kb\Delta^2}{4b - 4k\Delta}\right\} \tag{29}$$

V. Complete Trajectory Analysis

In order to simulate the complete trajectory, 3 different phases must be simulated: the initial trajectory, the impact and the consequent trajectory (in the hypothesis of a non-zero coefficient of restitution[6], $\gamma \neq 0$). The first part has been discussed in Section III, and allows for the simulation up to the impact point. Then, with the technique presented in Section IV, if the stability conditions (Eq. 26) are met, the impact can then be simulated; this translates into the knowledge of the impact force, or, in other words, of the coefficient of restitution γ of the contact. Hence, the computation of the Δv caused by the impact is straightforward.

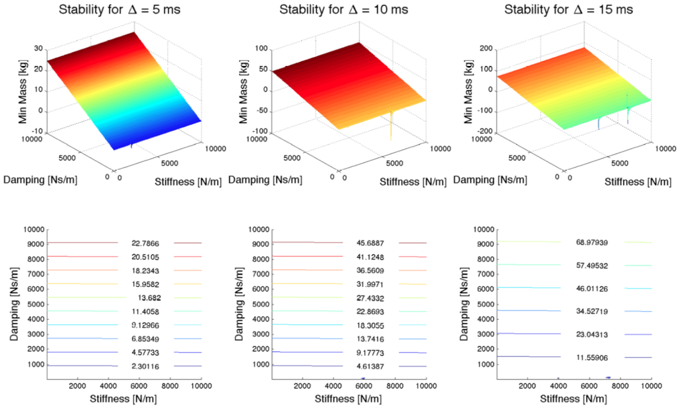


Figure 6: Minimum mass required for simulation stability, for $\Delta = \{5, 10, 15\}\, ms$

$$\Delta v = \int_{t_{imp}} \frac{|F|}{m}\cdot dt \tag{30}$$

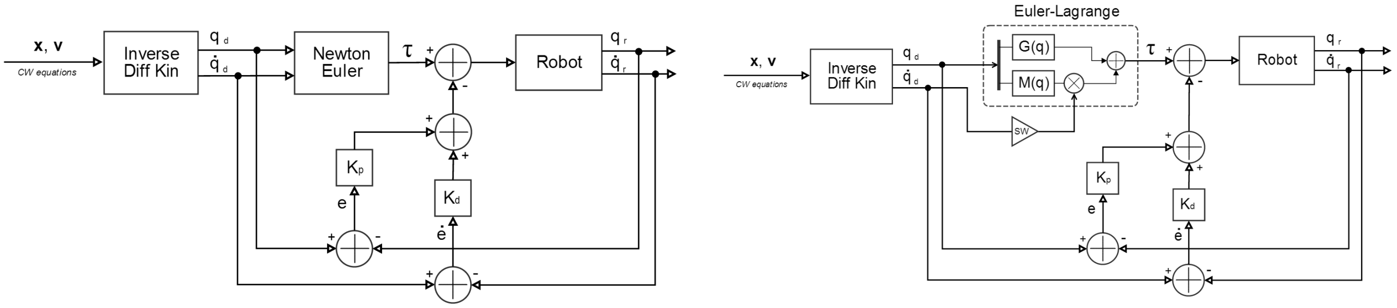
By plugging this into Eq. 26 and supposing that within the duration of the impact the relative position of the spacecrafts has not changed, it is possible to compute the third part of the trajectory, which is trivial. The procedure, in fact, is the same as the one used for the computation of the initial trajectory. In the future, we are planning to implement an autonomous control system that computes, if $\gamma \neq 0$, the new approaching trajectory after the impact, optimizing time schedule and fuel consumption.

VI. Control Techniques

In this section several control techniques for the manipulator are analyzed: the first technique, based on Newton-Euler dynamics, is the most correct in the sense that no approximations are taking place[7].

[6] The coefficient of restitution γ of two colliding objects is a positive real number between 0 and 1 representing the ratio of speeds after and before an impact, taken along the line of the impact. Pairs of objects with $\gamma = 1$ collide elastically, while objects with $\gamma < 1$ collide inelastically. For $\gamma = 0$, the objects effectively "stop" at the collision, not bouncing at all.

[7] In the hypothesis of a perfect knowledge of the geometrical and inertial parameters.



(a)

(b)

Figure 7: Control systems based on Euler-Newton and Lagrange-Euler dynamic models respectively. Note that it is possible to switch from the two Lagrange-Euler based systems by setting the switch block SW of Fig. 5b to 0 for gravity only, and to 1 to gravity and inertia.

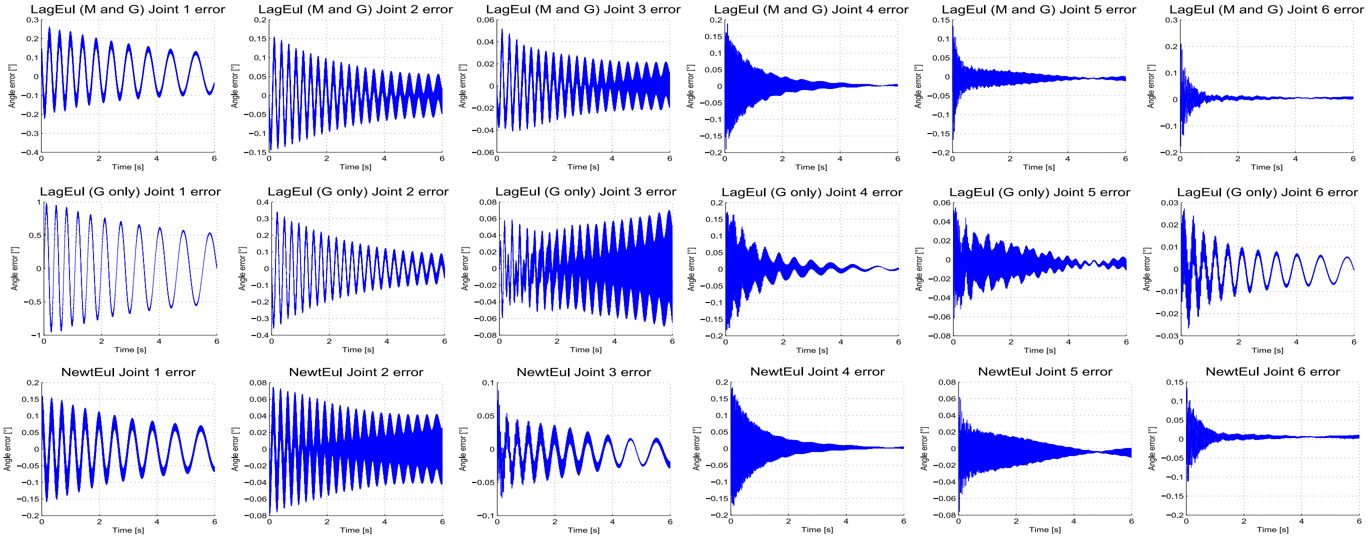


Figure 8: Joint errors for the simulated trajectory presented in Section III. The control loops are: Lagrange-Euler with M and G for the 1^{st} row, Lagrange-Euler with G only for the 2^{nd} row, Newton-Euler for the 3^{rd} row.

However, this approach requires a relatively long computation time, leading to delays that could eventually give rise to instability to the discrete digital loop. In order to find a trade-off between the correctness of the model and the computation time, we analyzed approximate controls based on Lagrange's equation. While Newton-Euler's formulation, due to its iterative, non intuitive form, is not easy to manipulate, Lagrange's expression, on the other hand, thanks to an immediate physical meaning of its components, is definitely more prone to tailoring and approximations. A SimMechanics® model was used to simulate the trajectory and to compute the joint errors, which are displayed in Fig. 8. The system block diagrams are presented in Fig. 7. The presence of computation delays is taken into account by feeding the simulated system with a quantized torque, whose step-size is equal to the self iteration time. The gains used for these simulations are constant and equal to $K_p = 50$ and $K_d = 0.01$. In the future, the authors plan to design a robust controller, able to find a good set of constant gains such that, despite possible variations, the poles are guaranteed to stay in favorable locations.

A. *Newton-Euler feedforward control*

This control technique (Fig. 7a) provides the actuators with the exact torque vector τ computed with Newton-Euler's approach. Though this system could theoretically work in an open chain fashion, there is, nonetheless, a feedback compensator that rejects external disturbances.

The reference trajectory to be followed is calculated from CW equations (refer to Section III) and is istantaneously expressed as a Cartesian vector of position and velocity that is further converted into general coordinates $\mathbf{q}$ and $\dot{\mathbf{q}}$ by using the differential kinematics technique. The average iteration time[8] is $\sim$2.3 ms. Although this technique is certainly not the fastest (it is almost two times longer than approach B), the controller allows for an optimal rejection of noise and system uncertainties, with a maximum error of 0.2° at $Joint$ 4 (Fig. 8).

B. *Lagrange-Euler (gravity compensation) feedforward control*

This approach consists of the calculation of Lagrange equation's gravity term only, $\mathbf{G}(\mathbf{q})$. This means ignoring the effect of the

[8] This has been calculated using an Intel® Core i7-3770, CPU @ 3,40 GHz, 8 GB RAM computer.

inertia and of the centrifugal and Coriolis acceleration. For relatively slow dynamics (like a docking approach manuever), the gravity term accounts for most of the torque that needs to be produced by the motors for a correct trajectory tracking. Since the gravitional term is the most straightforward and less time-consuming part to be calculated in Lagrange's equation, a feedforward control that uses only $\mathbf{G}(\mathbf{q})$ has been designed, and can be seen in Fig. 7b (when the switch block value SW=0). The average iteration time is $\sim$1.1 ms. Though this is the fastest technique, the neglectance of the inertia and centrifugal terms gives rise to important errors, expecially at $Joint$ 1, where a drift of 1° was observed (Fig. 8).

C. Lagrange-Euler (inertia and gravity compensation) feedforward control

This technique can be seen as the natural evolution of the gravity compensation feedforward control. In addition, in fact, the inertia contribution $\mathbf{M}(\mathbf{q})$ is considered. At a price of an additional term to be computed, this system is able to track more accurate trajectories in which the kinematics gives rise to consistent inertial forces. The system can be seen in Fig. 7b (when the switch block value SW=1). The average iteration time is $\sim$2.1 ms. With respect to the previous system, errors are clearly mitigated, but the performances are still worse than Newton-Euler's approach (Fig. 8).

VII. Conclusions and Future Work

In this paper, we discussed a step-to-step approach to the simulation of rendezvous and docking manuevers, from the kinematics analysis to the modelling of the contact. By adopting a spring-dashpot model, it was possible to analyze the dynamics of the impact and to implent, using the virtual force approach [7], a control loop for the simulation of the orbital scenario. By inspecting the system stability with Routh-Hurwitz's criterion, analytical conditions relating the facility performances and the scenarios that can be possibly simulated were extracted. These expressions take into account the facility delay Δ and the satellites parameters m_T, m_C, k, b.

As far as the control system is concerned, results showed that the most efficent technique is based on Newton-Euler feedforward control, allowing to obtain an angular joint error $< 0.1^\circ$ with a computation time of $\sim$2.3 ms. By implementing the code in a C^{++} environment, we expect to drastically improve this Matlab$^{®}$ based result.

Future work will mainly consist of the construction of the facility and in the validation of the simulated models. In particular, the effects of non-linearities, such as friction, will be analyzed and inserted in the model; moreover, for the simulation of impacts, the parameters of Fig. 4 will be fully characterized and the approximations will be verified; once the manipulator has been built and the control software implemented, a full-manuever (rendezvous and impacts) simulation and validation campaign will take place.

References

[1] T. Boge, T. Wimmer, O. Ma, and M. Zebenay, "Epos - a robotics-based hardware-in-the-loop simulator for simulating satellite rvd operations," *i-SAIRAS, Sapporo, Japan*, 2010.

[2] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. Springer Verlag, 2010.

[3] ——, *Robotics: Modelling, Planning and Control*. Springer Verlag, 2010.

[4] J. Y. S. Luh, M. W. Walker, and R. P. Paul, "On-line computational scheme for mechanical manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, 1980.

[5] H. D. Curtis, *Orbital Mechanics for Engineering Students*. Elsevier, 2010.

[6] G. Gilardi and I. Sharf, "Literature survey of contact dynamics modelling," *Mechanism and Machine Theory*, 2002.

[7] M. Zebenay, R. Lampariello, T. Boge, and D. Choukroun, "A new contact dynamics model tool for hardware-in-the-loop docking dimulation," *i-SAIRAS, Turin, Italy*, 2012.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1989.

[9] R. C. Dorf and B. R. H., *Modern Control Systems*, 12th ed. Prentice Hall, 2012.