

A Passivity-Based Approach for Simulating Satellite Dynamics With Robots: Discrete-Time Integration and Time-Delay Compensation

Marco De Stefano , Ribin Balachandran , and Cristian Secchi 

Abstract—This article proposes a passivity-based approach for simulating satellite dynamics on a position-controlled robot equipped with a force–torque sensor. Time delays intrinsic in the computational loop and discrete-time integration degrade the behavior of the satellite dynamics reproduced by the robot. These factors can generate an energy-inconsistent simulation and can even render the system unstable. In this article, time delay and discrete-time integration effects are analyzed from an energetic perspective and compensated through a passivity-based control strategy to ensure a faithful and stable dynamic simulation with position-controlled robots. The benefit of the proposed strategy is validated by simulations and experiments on the On-Orbit Servicing Simulator (OOS-SIM), a robotic facility used for simulating free-floating dynamics.

Index Terms—Hardware-in-the-loop simulation, passive integrator, space robotics, time-delay compensation.

I. INTRODUCTION

THE assembly and maintenance in orbit using robotic technologies began in 1981 when the Space Shuttle Manipulator was used for the first time [1]. Advances in space robotic technologies have enabled the execution of precise and complex maintenance tasks in orbit which reduce the risks associated with manned extravehicular activities. Unmanned mission demonstrators were flown in orbit to validate autonomous robot control technologies [2]. A milestone was the success of the Robot Technology Experiment (ROTEX) to demonstrate servicing prototype capabilities using robots [3]. Later, robotic experiments were conducted with the Japanese Engineering Test Satellite VII (ETS-VII) [4], including real-time collision avoidance capabilities [5].

Although the list is not exhaustive, the common aspect in all of these on-orbit experiments is the use of a manipulator

Manuscript received June 22, 2019; accepted September 2, 2019. This paper was recommended for publication by Associate Editor J. Abbott and Editor P. Rubuffo Giordano upon evaluation of the reviewers' comments. (Corresponding author: Marco De Stefano.)

M. De Stefano and R. Balachandran are with the Department of Analysis and Control of Advanced Robotic System, German Aerospace Center, 82234 Wessling, Germany (e-mail: marco.destefano@dlr.de; ribin.balachandran@dlr.de).

C. Secchi is with the Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, 41100 Modena, Italy (e-mail: cristian.secchi@unimore.it).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2019.2945883

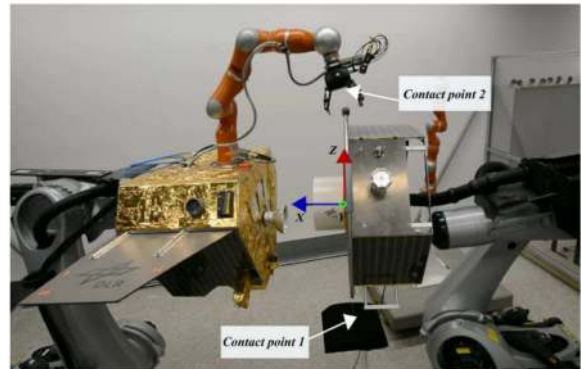


Fig. 1. OOS-SIM facility. The industrial robots are equipped with force–torque sensors. Servicer (left) and client (right).

operating in space. Robotic manipulators can be used in missions for maintenance or deorbiting of defective satellites in the context of active space debris removal. The dynamics and control of a space robot can be quite complex [6] and before the actual space missions, it is necessary to test and validate the control tasks on ground. Thus, reliable simulators which can reproduce microgravity conditions are required. To this end, several solutions can be adopted such as air bearing systems, neutral buoyancy, zero-G parabolic flights, cable off-loaders, and robotic facilities [1]. Air bearing systems can provide gravity compensation but they are limited to planar motion simulations only [7]. Neutral buoyancy can provide a very large workspace, however, the fluids introduce hydrodynamic effects which distort the dynamics to be reproduced. Zero-G parabolic flights can achieve nearly zero gravity conditions but they are limited by test duration (only 20–30 s) and cargo [8].

Robotic facilities are systems with hardware-in-the-loop where the robot end-effector moves according to a mathematical model to reproduce the desired working conditions (e.g., zero gravity). Several robotic simulators for space applications are available—INVERITAS, a facility for rendezvous and capture of satellites [9], EPOS (European Proximity Operation Simulator) to simulate rendezvous and docking [10], the SOCS (Lockheed Martin Space Operation Simulation Center) for testing the Orion rendezvous [11], and the OOS-SIM, an on-ground experimental facility for on-orbit servicing simulations [12]. In particular, the OOS-SIM is composed of two industrial robots simulating the free-floating satellites (see Fig. 1). One of the satellites (servicer)

is equipped with a manipulation arm. Each of the industrial robots has a force–torque sensor mounted on its end-effector to measure external interactions. The measured forces and torques are then provided to the mathematical model of the desired dynamics whose output (velocity) is commanded to the robot that moves in Cartesian space.

The main issues with robotic facilities are the time delay between the measured forces and the simulation driven reaction of the robot, and discrete-time dynamics integration [1]. These factors, if not properly addressed, cause energy inconsistencies and instabilities, especially during interactions with the environment.

A. Related Work

Several control algorithms for time-delay compensation have been proposed for achieving stable simulation by modifying the contact model (see, e.g., [13], [14]). However, they require identification of contact parameters. In [15] and [16], a first-order compensation model was designed, but the knowledge of contact frequency and time-delay duration is needed.

The passivity-based approaches have emerged as intuitive and effective strategies for achieving stability independent of the time delay. In fact, ensuring the passivity of the overall system is a sufficient condition for having a stable behavior [17]. The field of haptics and teleoperation has thoroughly exploited approaches based on passivity to deal with the instabilities due to time delay (see [18] for a survey). For example, wave variables ([19], [20]) have been implemented for making a bilateral communication channel passive. However, this technique introduces a characteristic impedance which might deteriorate the behavior of the teleoperation system [21]. In [22] and [23], the concept of energy tank has been exploited for dealing with time delay. But, the selection and tuning of the energy tank parameters is required. In our article, we exploit time-domain passivity approach (TDPA), [24] to address the delay problems. Within this context, [25] proposed a geometric solution for haptic devices. In [26], delay has been compensated using a time-varying damping weighted by the inertia matrix for a haptic device that, unlike industrial robots, is back-drivable. In [27], a technique for dissipating the active energy in the null space of a redundant robot is suggested. This solution is not applicable in the considered context since most of the facilities for rendering satellite dynamics do not have redundancy (e.g., [10] and [12]). In [28] and [29], TDPA has been exploited for reliably rendering satellite dynamics on a robotic simulator independent of the time delay. However the method required the knowledge of the robot dynamic model.

In hardware-in-the-loop simulators, a discrete-time integration of a mathematical model (e.g., satellite dynamics) is also required. Discrete-time integration can lead to a loss of the geometric and energetic properties characterizing physical systems and, therefore, to an unnatural and unstable behavior [17]. Geometric integrators, i.e., numerical integration methods that preserve geometric structure, such as symplectic forms, have been developed over the years (see [30]–[32]). These methods require a numerical and iterative solution of the updated equation

for each time step, which typically prohibits real-time determinism for the industrial robot. Energy-preserving integrators based on momentum conservation [33] are also available in the literature. Nevertheless, geometric integration deals mainly with isolated physical systems or systems with some damping [34], and the interaction with the external environment is not considered. In haptics control, the operator needs to interact with a virtual environment and the problem of integrating a nonisolated physical dynamics is relevant. In [35], it is shown that standard explicit integrators do not ensure passivity and, therefore, a more complex integration strategy has to be sought. An implicit integration method, based on the port-Hamiltonian formulation of the dynamics to be simulated is proposed in [20]. In [36], a fast but implicit and variable rate integration strategy for implementing mass–spring–damper systems is designed. Implicit and variable rate integration methods can prohibit real-time determinism on a standard industrial robot and, therefore, a real-time explicit method is more suitable. In [37] and [38], an explicit method for simulating the dynamics of a rigid body was proposed and validated on a robotic system for simulating the dynamics of a satellite. However, translational and rotational dynamics were treated separately.

B. Contribution

A common approach to compensate time delay in the control loop of a robotic simulator is to modify the contact parameters without addressing the effects of discrete-time dynamics integration [13]–[15] and [16].

In this article, a control strategy based on passivity criteria is proposed in order to ensure a faithful and reliable dynamic simulation. The approach does not rely on contact models and it can compensate the increase of energy due to the time delay and also due to the discrete-time dynamics integration. The contribution is based on three main aspects.

First, a novel time-delay compensation strategy based on TDPA is developed. This strategy is different from previous works [28] and [29], where a force estimator was required for the time-delay compensation control. The force estimator was a function of the robot dynamics model. While using industrial robots, the dynamic model is not always available and the absence of a joint-torque interface limits its identification. Therefore, in this article, the time-delay compensation strategy will rely only on measured data without depending on the robot dynamics model. Furthermore, in [28] and [29], the passivity controller was applied in admittance causality. In this article, the passivity controller will be designed in impedance causality where the modified force is the input to the satellite dynamics. Thus, the admittance port can be used to compensate the energy generated by the discrete-time integrator by modifying the velocity commanded to the robot.

In [37] and [38], separate passive integrators for translational and rotational dynamics were designed, respectively, without including time-delay compensation. In this article, a unified integrator is obtained for simulating the complete satellite dynamics.

Second, a unified architecture is designed, which combines the time-delay compensation and the passive integrator, and

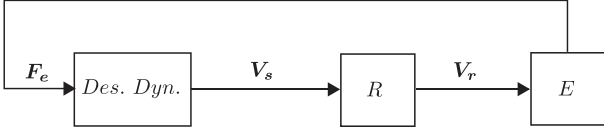


Fig. 2. Admittance architecture with the desired dynamic— R is the robot, E is the environment, $Des. Dyn.$ is the force-acceleration model of the dynamics to be simulated.

it acts as a passivity layer between the real hardware and the mathematical model. Time-delay effects are resolved with a passivity control (PC) which acts in impedance mode (force correction) and the discrete integration effects are solved with a second PC in admittance mode (velocity correction).

Third, the proposed unified control structure is validated experimentally on an industrial robot, the OOS-SIM, shown in Fig. 1.

II. PROBLEM STATEMENT

A suitable control strategy for implementing desired dynamics on industrial robots is the admittance control [39]. This strategy has been exploited in many different fields, see, e.g., [23] for a surgical application, [40] for a human–robot interaction scenario, and also [12] for robotic simulators for space applications.

The schematic of a robotic simulator controlled with an admittance architecture is illustrated in Fig. 2.

The robot R is equipped with a six DoF (degree of freedom) force–torque sensor at the end-effector. When the robot interacts with the environment E , forces and torques are generated and measured by the sensor. $\mathbf{F}_e \in \mathbb{R}^6$ is the vector composed of Cartesian forces and torques. This is the input to the desired dynamics that, after integration, will provide the twist $\mathbf{V}_s \in \mathbb{R}^6$ as a set point to the robot, which will consequently reproduce the desired behavior.

Our main goal is to simulate a satellite dynamics that can be characterized as a free-floating rigid body. The translational dynamics is given by

$$\dot{\mathbf{v}}_s = \mathbf{M}^{-1} \mathbf{f}_e \quad (1)$$

and the rotational dynamics is described by

$$\dot{\boldsymbol{\omega}}_s = \mathbf{I}^{-1}(\mathbf{I}\boldsymbol{\omega}_s \times \boldsymbol{\omega}_s + \boldsymbol{\tau}_e) = \mathbf{I}^{-1}(\mathbf{S}(\mathbf{I}\boldsymbol{\omega}_s)\boldsymbol{\omega}_s + \boldsymbol{\tau}_e) \quad (2)$$

where the variables are defined as¹:

- 1) $\mathbf{M} \in \mathbb{R}^{3 \times 3}$ the simulated mass matrix;
- 2) $\mathbf{f}_e \in \mathbb{R}^3$ the Cartesian measured force;
- 3) $\mathbf{v}_s \in \mathbb{R}^3$ the Cartesian linear velocity;
- 4) $\dot{\mathbf{v}}_s \in \mathbb{R}^3$ the Cartesian linear acceleration;
- 5) $\mathbf{I} \in \mathbb{R}^{3 \times 3}$ the inertia matrix of the body about the center of mass;
- 6) $\boldsymbol{\omega}_s \in \mathbb{R}^3$ the angular velocity;
- 7) $\dot{\boldsymbol{\omega}}_s \in \mathbb{R}^3$ the angular acceleration;
- 8) $\mathbf{S}(\mathbf{I}\boldsymbol{\omega}_s) \in \mathbb{R}^{3 \times 3}$ the skew-symmetric matrix such that $\mathbf{S}(\mathbf{I}\boldsymbol{\omega}_s)\boldsymbol{\omega}_s = \mathbf{I}\boldsymbol{\omega}_s \times \boldsymbol{\omega}_s$;
- 9) $\boldsymbol{\tau}_e \in \mathbb{R}^3$ the Cartesian measured torques.

¹In this article, we use bold notation to indicate vectors and normal font for the individual components of the vectors.

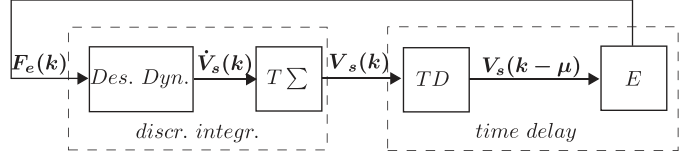


Fig. 3. Admittance architecture where $T\Sigma$ is the discrete integrator with time step T . Robot and internal controller are represented by time-delay block, TD .

The total wrench, twist, and mass matrix are defined as:

- 1) $\mathbf{F}_e = (\mathbf{f}_e, \boldsymbol{\tau}_e) \in \mathbb{R}^6$ the wrench;
- 2) $\mathbf{V}_s = (\mathbf{v}_s, \boldsymbol{\omega}_s) \in \mathbb{R}^6$ the twist;
- 3) $\mathbf{H} = (\mathbf{M}, \mathbf{0}_{3 \times 3}; \mathbf{0}_{3 \times 3}, \mathbf{I}) \in \mathbb{R}^{6 \times 6}$ the total mass matrix (mass and inertia of the body).

By integrating (1) and (2) and by applying the inverse kinematics it is possible to obtain the desired joint velocities for the industrial robot to track and the end-effector will evolve as a floating rigid body not subjected to gravity.

Fig. 3 shows the control architecture of the industrial robot where the location of the discrete-time integration and the time delay are delineated. The acceleration of the rigid body (computed by the desired dynamics block) is integrated in discrete time with sampling time T , which results in the commanded velocity $\mathbf{V}_s(k)$. We assume that the industrial robot can perfectly track a desired velocity set point. This is a common assumption and it can be achieved by properly tuning the gains of the low-level controllers of the robot [41]. However, the internal control of the robot and the inverse kinematics calculation might require several sampling steps for the computation of the desired set point. These factors introduce time delays in the corresponding simulation-driven motion of the robot. Therefore, the real velocity of the robot results to be $\mathbf{V}_s(k - \mu)$, which is the commanded velocity delayed by a quantity μ , where $\mu \in \mathbb{N}$ is the number of discrete time steps of sampling time T . Hence, the robot has been represented as a time delay TD in Fig. 3.

A. Time-Delay Effects

The effect of time delay within the admittance architecture shown in Fig. 3 can be seen in the following example. Let us consider a rigid body with mass $\mathbf{M} = \text{diag}[50, 50, 50]$ kg and inertia $\mathbf{I} = \text{diag}[18, 20, 22]$ kgm² with initial linear velocity $\mathbf{v}_s(0) = [0.1, 0.15, 0.2]$ m/s and angular velocity $\boldsymbol{\omega}_s(0) = [0.01, 0.02, 0.03]$ rad/s. We run a simulation with a time delay (TD) of 10 ms in the loop. During its motion, forces and torques are generated by colliding against virtual walls modeled using spring–damper systems. The measured forces and torques are the input to the desired dynamics block. The behavior of the system can be seen in Fig. 4 where the velocity with the time delay in the loop (solid line) is compared with the ideal case velocity (dashed line) without delay. As it can be seen, the velocity of the simulated rigid body increases after each collision which is against the energy conservation principle and it might lead to an unstable system.

B. Discrete Integration Effects

For analyzing the integration effects, we will consider the Euler integration method which is usually exploited in industrial

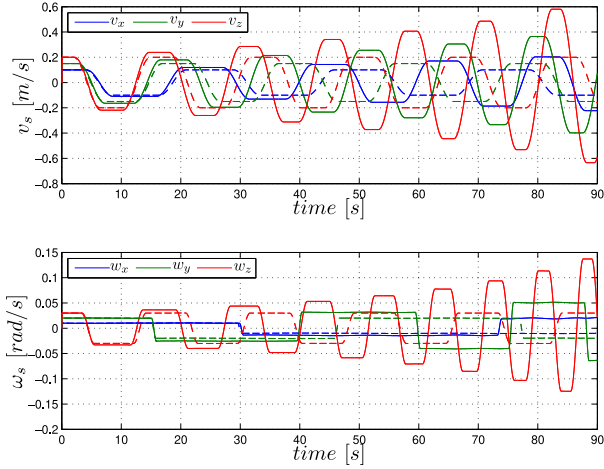


Fig. 4. Time delay causes instabilities-comparison of velocity with time delay in the loop (solid line) and ideal velocity without time delay (dashed line).

robots. The reason is that Euler integrator is fast and explicit and it can be implemented in real time. To isolate the effects of the discrete integration, let us consider that the time delay in the loop is zero. The total energy $H(\mathbf{v}_s, \boldsymbol{\omega}_s)$ of the rigid body is given by

$$H(\mathbf{v}_s, \boldsymbol{\omega}_s) = H_t(\mathbf{v}_s) + H_r(\boldsymbol{\omega}_s) \quad (3)$$

where $H_t = \frac{1}{2} \mathbf{v}_s^T \mathbf{M} \mathbf{v}_s$ is the translational kinetic energy and $H_r = \frac{1}{2} \boldsymbol{\omega}_s^T \mathbf{I} \boldsymbol{\omega}_s$ is the rotational energy. Consider now the effect of discretization on translational motions in (1). The following energy balance holds:

$$\dot{H}_t = \mathbf{f}_e^T \mathbf{v}_s. \quad (4)$$

If there is no external interaction (i.e., $\mathbf{f}_e = 0$), the energy stored in the system is constant (i.e., $\dot{H}_t = 0$).

Integrating the desired dynamics using the standard Euler method leads to the following discrete system:

$$\mathbf{v}_s(\mathbf{k}) = \mathbf{v}_s(\mathbf{k} - 1) + T\mathbf{M}^{-1} \mathbf{f}_e(\mathbf{k} - 1). \quad (5)$$

In case of free motion (i.e., $\mathbf{f}_e(\mathbf{k} - 1) = 0$), the velocity of the system is constant over time. Thus, in this simple case, standard Euler integration is energetically well posed since it allows the discretized dynamics to behave in a physically consistent way independent of the sample time. However, this relation does not hold anymore in case of external interaction. Let us consider a body with mass 30 kg subjected to a force profile shown at the top of Fig. 5. The integration of the dynamics is considered in the continuous case and compared with Euler discrete integrator for sampling times $T_1 = 0.1$ s and $T_2 = 0.01$ s. The increase in the energy which is introduced into the system with respect to the continuous time integrator is shown in Fig. 6, where H_{t_c} is the energy calculated in continuous time. Notice that the increase in energy is proportional to the sampling time and this leads to position drifts. The drift due to the integration with T_1 reaches 0.05 m between 0 and 32 s and 0.15 m between 32 and 50 s, (see Fig. 5 middle). Also, for the case with T_2 , the drift appears. Since the sampling time is smaller, it results in a drift ten times lower, as shown in Fig. 5 bottom.

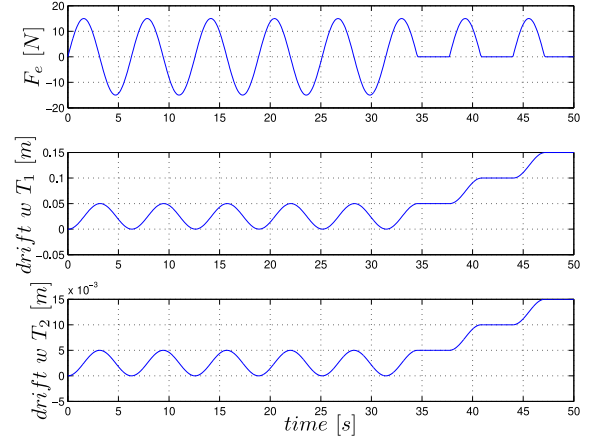


Fig. 5. Force profile, drift in position due to the discrete integration with (w) sampling times T_1 and T_2 .

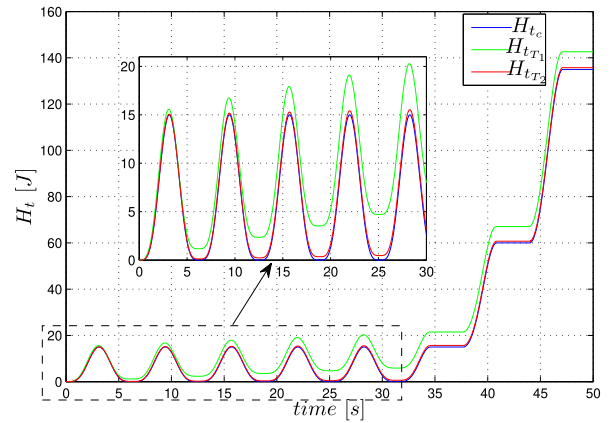


Fig. 6. Discrete integration causes drift in energy for the translational dynamics. Energy ($H_{t_{T_1}}$) calculated with sampling time T_1 and ($H_{t_{T_2}}$) calculated with sampling time T_2 . The reference energy H_{t_c} is calculated in continuous time.

Similar to the translational case, using the rotational dynamics in (2), the following energy balance holds:

$$\dot{H}_r = \boldsymbol{\omega}_s^T \boldsymbol{\tau}_e. \quad (6)$$

Thus, as evident from (6), the energy variation is due to the power exchanged with the environment.

In order to compute the discrete angular velocity set point, (2) is integrated according to the Euler method and it results in

$$\begin{aligned} \mathbf{I}(\boldsymbol{\omega}_s(\mathbf{k}) - \boldsymbol{\omega}_s(\mathbf{k} - 1))T^{-1} &= \mathbf{S}(\mathbf{I}\boldsymbol{\omega}_s(\mathbf{k} - 1))\boldsymbol{\omega}_s(\mathbf{k} - 1) \\ &\quad + \boldsymbol{\tau}_e(\mathbf{k} - 1). \end{aligned} \quad (7)$$

Consequently, the angular velocity set point to be commanded to the robot is

$$\begin{aligned} \boldsymbol{\omega}_s(\mathbf{k}) &= \boldsymbol{\omega}_s(\mathbf{k} - 1) + \mathbf{I}^{-1}T\mathbf{S}(\mathbf{I}\boldsymbol{\omega}_s(\mathbf{k} - 1))\boldsymbol{\omega}_s(\mathbf{k} - 1) \\ &\quad + T\mathbf{I}^{-1}\boldsymbol{\tau}_e(\mathbf{k} - 1). \end{aligned} \quad (8)$$

Similar to the translational case, the Euler integration generates extra energy which destroys the energetic properties of

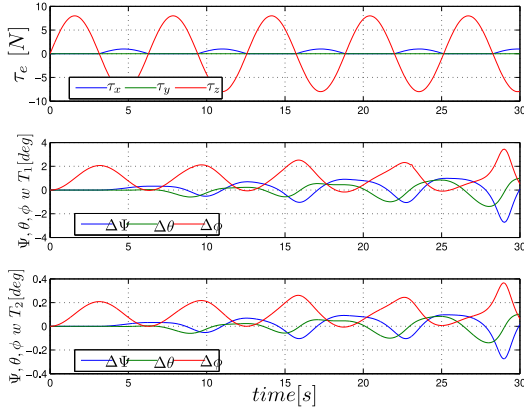


Fig. 7. Applied torques (top) and drift in roll, pitch and yaw considering sampling time T_1 (middle) and T_2 (bottom).

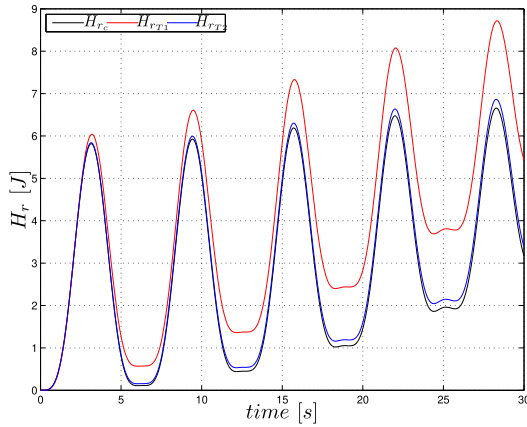


Fig. 8. Discrete integration of rotational dynamics causes energy drift. Comparison of energy calculated in continuous time (H_{r_c}) and discrete sampling times T_1 ($H_{r_{T_1}}$) and T_2 ($H_{r_{T_2}}$).

the rotational dynamics. This can be seen in the following example. Consider the torque profile in Fig. 7 (top) acting on a simulated body with inertia on the principal axis $I = \text{diag}(18, 20, 22)$ kgm². We calculate the rotational energy H_r for sampling times $T_1 = 0.1$ s and $T_2 = 0.01$ s. Fig. 8 shows the comparison of the energy calculated with the continuous time (H_{r_c}) and discrete times ($H_{r_{T_1}}$, $H_{r_{T_2}}$). It can be seen that there is an increase in energy due to the discrete integration. The increase of energy results in a variation of angular position with respect to the ideal, continuous case. Fig. 7 (middle and bottom) shows the angular error (Δ) in roll, pitch, and yaw (Ψ, θ, ϕ) calculated with T_1 and T_2 with respect to the continuous case.

Energy and pose drifts affect the physical behavior of the satellite simulated by the robot. Such drifts may lead the robot to interact with unforeseen objects that produce further (drifted) behaviors leading to a deteriorated performance of the dynamic simulation and even instabilities.

Notice that both industrial robots in Fig. 1 operate with the same control architecture and the reactive dynamics is perceived by the respective force–torque sensors, hence, the robots are susceptible to the issues presented in this section. Therefore,

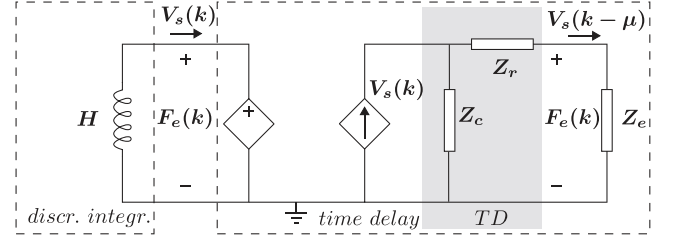


Fig. 9. Admittance architecture of the industrial robot for the hardware-in-the-loop satellite simulation represented in electrical domain.

compensating the effects of time delay and discrete-time integration is important when external force is applied in order to avoid instability issues and produce a faithful dynamic simulation with the industrial robots.

III. NETWORK MODELING AND TDPA

In this section, we design a network model of the system presented in Fig. 3. This representation will allow us to highlight the energetic structure of the system and it will help identify the energy production due to the delay and the discrete integration. Furthermore, a background on TDPA, the control strategy that will be used, is provided.

A. System Modeling

The control architecture of the industrial robot shown in Fig. 3, can be represented in electrical domain. This representation aids the definition of the power ports and the analysis of the energy behavior. Therefore, the conventional mechanical–electrical analogy is exploited to map forces into voltages and velocities into currents and the resulting circuit is shown in Fig. 9.

The industrial robot is represented in the gray box with an impedance Z_r and its internal controller Z_c . The interaction of the robot with the environment (Z_e) produces the voltage F_e which represents the force–torque sensor signals.² This is transmitted with an ideal voltage source F_e to the left side of the circuit and acts on the inductance H , which represents the simulated mass of the satellite. Using the admittance dynamics in (5) and (8), the desired velocity V_s is computed and is then commanded to the internal controller of the robot through the dependent current source on the right side of the circuit.

Usually, it might be difficult to have access to the internal controller of the industrial robot. The designed network representation allows us to consider the internal controller together with the robot impedance (Z_c and Z_r). Robot and internal controller act as a delay source (between the commanded velocity and resulting measured velocity of the robot) and therefore, they can be replaced with the time-delay block, TD (see gray box in Fig. 9.)

Two one-port networks can be identified in the electrical system which are represented by the two dashed boxes in Fig. 9.

²Note that for the numerical simulation, the measured force F_e is modeled as a spring-dashpot model, later it will be replaced with the real sensor measurements during the experiments with the industrial robot.

The output of the left-side network is \mathbf{V}_s , the velocity computed by the discrete dynamics integration. The right-side network, which contains the time delay, outputs the interaction force \mathbf{F}_e . Now, the energy behavior of the system can be analyzed from the interaction port with power conjugated variables \mathbf{V}_s and \mathbf{F}_e .

B. Time Domain Passivity Approach

TDPA is an effective strategy for enforcing passivity in a one-port network. The discrete-time domain will be considered, since it is the most suitable setting for the problems addressed in this article [24]. Consider a one-port network endowed with a power port $(\mathbf{u}(\mathbf{k}), \mathbf{y}(\mathbf{k})) \in \mathbb{R}^n \times \mathbb{R}^n$ by which it can interact (or be interconnected) with external systems.

The input $\mathbf{u}(\mathbf{k})$ and the output $\mathbf{y}(\mathbf{k})$ are power conjugated (e.g., force and velocity), i.e., their product has the unit of power. The system is passive if there is a lower bounded energy function $E(m)$ such that

$$E(m) = E(0) + \sum_{k=0}^m \mathbf{u}^T(\mathbf{k})\mathbf{y}(\mathbf{k})T \geq 0 \quad \forall m > 0 \quad (9)$$

where $E(0)$ represents the initial energy stored in the system. Equation (9) states that the system cannot produce more energy with respect to its initial storage and the input energy. Passivity is a sufficient condition for stability and, in particular, if a system is passive then it is also stable [42]. The main idea of TDPA is to monitor the energy flowing through the power port by means of a passivity observer (PO). If $E(m) < 0$, the system is producing energy and such a regenerative effect can destabilize the system [17]. In order to reestablish a passive behavior, a passivity controller (PC) is activated and it acts as a variable damper. More details about TDPA can be found in [24].

IV. TIME-DELAY COMPENSATION

In this section, we will exploit the network representation developed in Section III and TDPA in order to compensate the destabilizing effects of the time delay illustrated in Section II-A.

In our previous works ([28], [29]), the admittance PC (velocity modification) has been used and it requires the knowledge of the robot dynamics. However, for industrial robots, the dynamics are not generally available. In the current work, an impedance causality for the PC is chosen, i.e., the input variable to the admittance dynamics is a variable force signal. This will allow us to render passive the subnetwork containing the robot with time delay without the knowledge of the robot dynamics.

Consider the one-port network (dashed box in Fig. 10) which includes the time delay, with power-correlated variables $(\mathbf{F}_e, \mathbf{V}_s)$. We observe the energy at this port and if activity is detected, the PC modifies the force, thus rendering the one-port network passive. In the ideal case (where no delay is considered), the energy at one-port will be

$$\sum_{k=0}^m \mathbf{F}_e^T(\mathbf{k})\mathbf{V}_s(\mathbf{k})T = \sum_{k=0}^m \mathbf{V}_s^T(\mathbf{k})\mathbf{Z}_e\mathbf{V}_s(\mathbf{k})T. \quad (10)$$

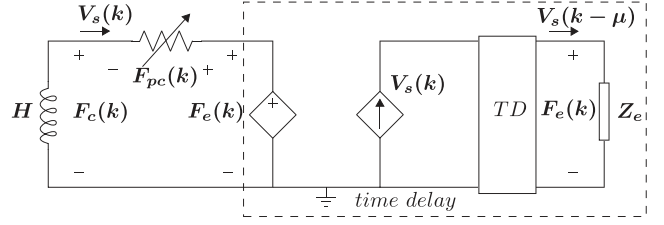


Fig. 10. Modeling in electrical domain for the time delay-variable force output.

However, when delay is considered, this condition will not hold anymore because the robot will interact with the environment with a delayed velocity $\mathbf{V}_s(\mathbf{k} - \mu)$ producing a force $\mathbf{V}_s^T(\mathbf{k} - \mu)\mathbf{Z}_e$. Then, the energy at the one-port will be different, more specifically

$$\sum_{k=0}^m \mathbf{F}_e^T(\mathbf{k})\mathbf{V}_s(\mathbf{k})T = \sum_{k=0}^m \underbrace{\mathbf{V}_s^T(\mathbf{k} - \mu)\mathbf{Z}_e}_{\mathbf{F}_e^T(\mathbf{k})}\mathbf{V}_s(\mathbf{k})T. \quad (11)$$

The delay causes a discrepancy in the velocity during the contact with the environment and the measured force is a function of it, see (11). The energy of the port $(\mathbf{F}_e, \mathbf{V}_s)$ can be monitored at each time-step with an energy observer defined as

$$E_{\text{obs}_1}(\mathbf{k}) = E_{\text{obs}_1}(\mathbf{k} - 1) - \mathbf{F}_e^T(\mathbf{k})\mathbf{V}_s(\mathbf{k})T + \mathbf{V}_s^T(\mathbf{k} - 1)\boldsymbol{\alpha}(\mathbf{k} - 1)\mathbf{V}_s(\mathbf{k} - 1)T \quad (12)$$

where $\boldsymbol{\alpha}(\mathbf{k}) \in \mathbb{R}^{6 \times 6}$ is a positive definite matrix which represents the time-varying damping of the passivity controller, later defined. Therefore, the last term in (12) is the energy associated to the passivity controller, defined as

$$\mathbf{F}_{\text{pc}}(\mathbf{k}) = \boldsymbol{\alpha}(\mathbf{k})\mathbf{V}_s(\mathbf{k}). \quad (13)$$

The impedance causality of the PC modifies the force $\mathbf{F}_e(\mathbf{k})$ by a quantity $\mathbf{F}_{\text{pc}}(\mathbf{k})$ on the left-side of Fig. 10. The modified force provided to the admittance model is

$$\mathbf{F}_c(\mathbf{k}) = \mathbf{F}_e(\mathbf{k}) - \mathbf{F}_{\text{pc}}(\mathbf{k}). \quad (14)$$

We choose the matrix $\boldsymbol{\alpha}(\mathbf{k})$ to have a diagonal form, therefore, the following energy exchange from (12) holds:

$$E_{\text{obs}_1}(\mathbf{k}) = - \sum_{m=0}^{\mathbf{k}} \sum_{i=1}^6 F_{e,i}(m)V_{s,i}(m)T + \sum_{m=1}^{\mathbf{k}} \sum_{i=1}^6 \alpha_i(m-1)V_{s,i}^2(m-1)T \quad (15)$$

where i is the i th component of the vector. This leads to

$$E_{\text{obs}_1}(\mathbf{k}) = \sum_{i=1}^6 E_{\text{obs}_{1,i}}(\mathbf{k}) \quad (16)$$

where the PO is

$$E_{\text{obs}_{1,i}}(\mathbf{k}) = E_{\text{obs}_{1,i}}(\mathbf{k} - 1) - F_{e,i}(\mathbf{k})V_{s,i}(\mathbf{k})T + \alpha_i(\mathbf{k} - 1)V_{s,i}^2(\mathbf{k} - 1)T. \quad (17)$$

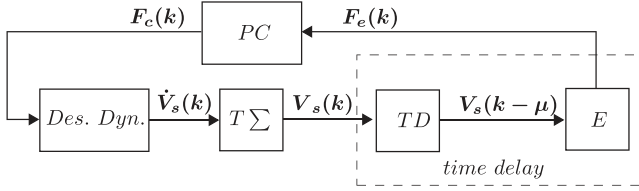


Fig. 11. Scheme with impedance PC for compensating the time-delay effects.

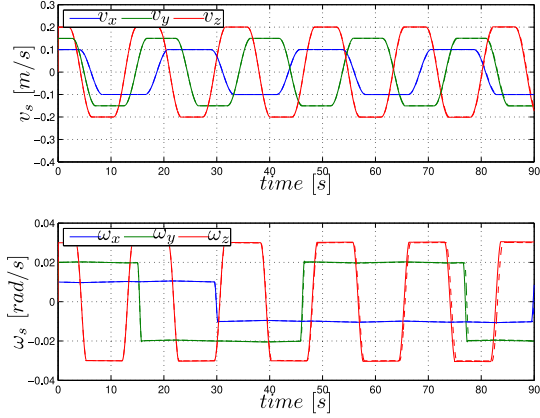


Fig. 12. Stable system—linear and angular velocity with time delay in the loop and passivity control (solid line), ideal velocity (dashed line).

The control term in (13), which dissipates at each time step the active energy, is a function of the variable damping matrix $\alpha = \text{diag}(\alpha_1, \dots, \alpha_6)$ whose components are

$$\alpha_i(k) = \begin{cases} -\frac{E_{\text{obs1},i}(k)}{V_{s,i}^2(k)T} & \text{if } E_{\text{obs1},i} < 0 \\ 0 & \text{else} \end{cases}. \quad (18)$$

Thus, if the one-port system behaves passively, $F_{\text{pc}} = 0$ and the measured force F_e is sent as an input to the admittance dynamics without modification. If some energy is produced, the damping factor is set such that the extra energy is dissipated and passivity is restored. The damper α_i will make $E_{\text{obs1},i} \geq 0 \forall i$. Therefore, the overall energy in (16) will be $E_{\text{obs1}}(k) \geq 0$ achieving passivity of the system. The schematic of Fig. 3 is adapted with the corrected force $F_c(k)$ as input to the desired dynamics and it is shown in Fig. 11.

In order to validate the control, we run the simulation with the same initial conditions and mass parameter as described in Section II-A. As it can be seen in Fig. 12, the velocity of the satellite does not increase and the overall system is stable. By applying the described approach, the damaging effect of time delay shown in Fig. 4 is resolved.

The energy exchange between the time-delay port and robot controller is not regulated explicitly, therefore, the one-port network as seen from the environment needs to be passive. A numerical analysis of the energy calculated at the environment port is performed in order to check the passivity. Fig. 13 (top) shows the energy for the translational dynamics where the time delay is in the loop and it is monitored with the proposed energy observer in (17) without the update of the passivity controller,

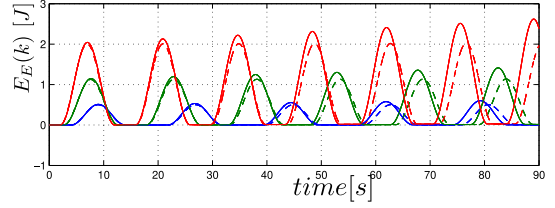
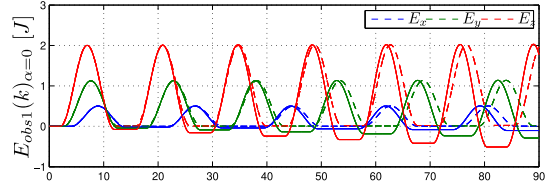


Fig. 13. Energy in ideal case (dashed line) and energy with time delay in the loop (solid line) measured at different ports. Top: Energy measured by the energy observer. Bottom: Energy exchange between the robot and the environment as measured at the interaction point.

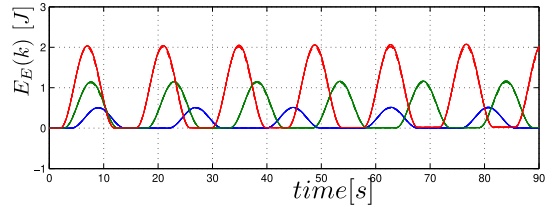
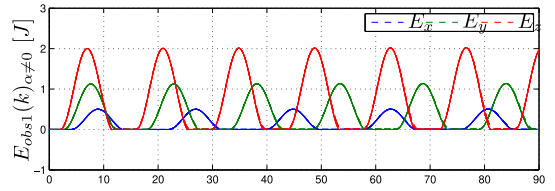


Fig. 14. Energy in ideal case (dashed line) and energy with time delay and PC (solid line) measured at different ports. Top: Energy measured by the energy observer. Bottom: Energy exchange between the robot and the environment as measured at the interaction point.

i.e., $E_{\text{obs1}}(k)_{\alpha=0}$. Fig. 13 (bottom) shows the energy measured at the interaction point, i.e., $E_E(k) = \sum F_e(k)V_s(k - \mu)T$ using the measured force $F_e(k)$ and velocity at the end-effector $V_s(k - \mu)$. In both plots, the dashed line is the energy in the ideal case (i.e., without time delay in the loop). As can be seen, violation of passivity is found at the port where the proposed energy observer is located, while at the interaction port, the system results to be passive according to definition (9).

Although the interactions are passive, it can be clearly seen that the energy increases at the interaction port. This is attributed to the time delay in the system. To prove this, we then evaluated the energy at the same ports with the passivity controller acting [i.e., $\alpha \neq 0$ in (17)]. As it can be seen in Fig. 14, the passivity is restored and the energies at both ports follow the ideal energy behavior (dashed line).

V. PASSIVE AND EXPLICIT INTEGRATOR

As shown in Section II-B, the generation of extra energy due to the discrete integration cause energy and position drifts. In this

section, the extra energy introduced by the discrete integrator is identified. This information is exploited for modifying the velocity output of the Euler integrator with an admittance causality. To have a complete passive integrator, we need to merge works performed in [37] and [38] in order to build the overall passive architecture.

A. Energy Generated by the Discrete Integration

1) *Translational Dynamics*: Consider the dynamics (1) discretized with the Euler method as reported in (5). The kinetic energy (4) in discrete time is

$$\begin{aligned} H_t(k) &= \frac{1}{2} \mathbf{v}_s(k)^T \mathbf{M} \mathbf{v}_s(k) \\ &= \frac{1}{2} [\mathbf{v}_s(k-1) + T \mathbf{M}^{-1} \mathbf{f}_e(k-1)]^T \\ &\quad \mathbf{M} [\mathbf{v}_s(k-1) + T \mathbf{M}^{-1} \mathbf{f}_e(k-1)]. \end{aligned} \quad (19)$$

Considering that the mass matrix \mathbf{M} is symmetric and positive definite, (19) can be rewritten as

$$\begin{aligned} H_t(k) &= H_t(k-1) + T \mathbf{v}_s(k-1)^T \mathbf{f}_e(k-1) \\ &\quad + \underbrace{\frac{1}{2} T^2 \mathbf{f}_e(k-1)^T \mathbf{M}^{-1} \mathbf{f}_e(k-1)}_{\Delta H_t(k)}. \end{aligned} \quad (20)$$

Equation (20) does not represent a physical and passive behavior. Indeed, the variation of energy should be only due to the power provided through the port, i.e., $\mathbf{v}_s(k-1)^T \mathbf{f}_e(k-1)$. The extra energy term

$$\Delta H_t(k) = \frac{1}{2} T^2 \mathbf{f}_e(k-1)^T \mathbf{M}^{-1} \mathbf{f}_e(k-1) \quad (21)$$

is due to Euler integration [37]. This extra energy causes energy inconsistency and drift that make the reproduced dynamics diverge from the ideal behavior as shown in Section II-B.

2) *Rotational Dynamics*: Consider the dynamics (2) discretized by means of the Euler method and reported in (8). The rotational kinetic energy in discrete time $H_r(k)$ is given by

$$H_r(k) = \frac{1}{2} \boldsymbol{\omega}_s(k)^T \mathbf{I} \boldsymbol{\omega}_s(k). \quad (22)$$

Considering that $(\mathbf{I}^{-1})^T = (\mathbf{I}^T)^{-1} \equiv \mathbf{I}^{-1}$ and exploiting the properties of the skew-symmetric matrix $\mathbf{S}(\cdot)$ after substituting (8) into (22), we obtain

$$\begin{aligned} H_r(k) &= H_r(k-1) + T \boldsymbol{\omega}_s(k-1)^T \boldsymbol{\tau}_e(k-1) \\ &\quad + \underbrace{\frac{1}{2} T^2 \boldsymbol{\omega}_s(k-1)^T \mathbf{S}(k)^T \mathbf{I}^{-1} \mathbf{S}(k) \boldsymbol{\omega}_s(k-1)}_{\Delta H_{r1}(k)} \\ &\quad + \underbrace{\frac{1}{2} T^2 \boldsymbol{\tau}_e(k-1)^T \mathbf{I}^{-1} \boldsymbol{\tau}_e(k-1)}_{\Delta H_{r2}(k)} \end{aligned} \quad (23)$$

where, for the sake of brevity, the dependency of $\mathbf{I} \boldsymbol{\omega}_s(k-1)$ in \mathbf{S} has been omitted.

As in the translational case, the discrete integration of rotational dynamics leads to a nonphysical behavior as shown in

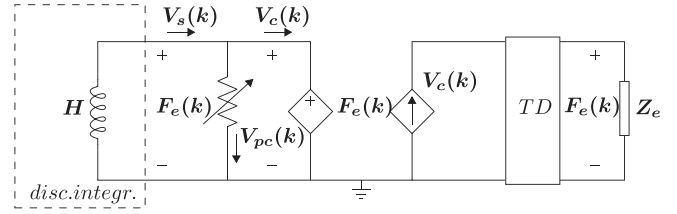


Fig. 15. Modeling in electrical domain for discretization—variable velocity input.

(23) by the term

$$\Delta H_r(k) = \Delta H_{r1}(k) + \Delta H_{r2}(k). \quad (24)$$

In fact, the energy variation should be only due to the energy provided through the power port, i.e., $T \boldsymbol{\omega}_s(k-1)^T \boldsymbol{\tau}_e(k-1)$.

In particular, ΔH_{r1} results from the integration of the rotational coupled dynamics, ΔH_{r2} results from the integration of the external torque $\boldsymbol{\tau}_e$. These energy terms introduce undesired dynamics into the system [38].

B. Passive Integration Scheme

In order to reestablish the physically consistent energetic behavior of the discretized dynamics, we need to dissipate the extra energy ΔH_t and ΔH_r . In this subsection, we will exploit TDPA for dissipating the extra energies.

1) *Passive Integration Scheme for Translational Dynamics*: The extra energy term produced by the integration and reported in (21) is always positive and, therefore, it always corresponds to a production of energy. It is possible to consider this term in a passivity observer in order to detect a loss of passivity. Thus, we can write the energy observer as

$$\begin{aligned} E_{\text{obs}_2}(k) &= E_{\text{obs}_2}(k-1) - \Delta H_t(k) \\ &\quad + \mathbf{f}_e(k-1)^T \boldsymbol{\beta}_1(k-1) \mathbf{f}_e(k-1) T. \end{aligned} \quad (25)$$

The second term on the right side of (25) is the energy produced by the discrete integration, which needs to be dissipated. The last term represents the energy dissipated by the passivity controller with a damping matrix $\boldsymbol{\beta}_1$ defined later. The action of the PC is provided in an admittance causality as follows:

$$\mathbf{v}_{\text{pc}}(k) = \boldsymbol{\beta}_1(k) \mathbf{f}_e(k). \quad (26)$$

This provides a correction to the output velocity of the admittance dynamics, i.e., the velocity commanded to the robot, given by

$$\mathbf{v}_c(k) = \mathbf{v}_s(k) - \mathbf{v}_{\text{pc}}(k) \quad (27)$$

where $\mathbf{v}_s(k)$ is achieved from (5). The passivity controller is represented as a variable resistance in parallel to the inductor in Fig. 15. The dissipation and, consequently, the correction, get active only when passivity is violated, i.e., when $E_{\text{obs}_2}(k) < 0$ and only the excess of energy is dissipated.

We choose the matrix $\boldsymbol{\beta}_1(k) \in \mathbb{R}^{3 \times 3}$ to have a diagonal form such that $\boldsymbol{\beta}_1 = \text{diag}(\beta_{1,1}, \beta_{1,2}, \beta_{1,3})$ and from (25), the

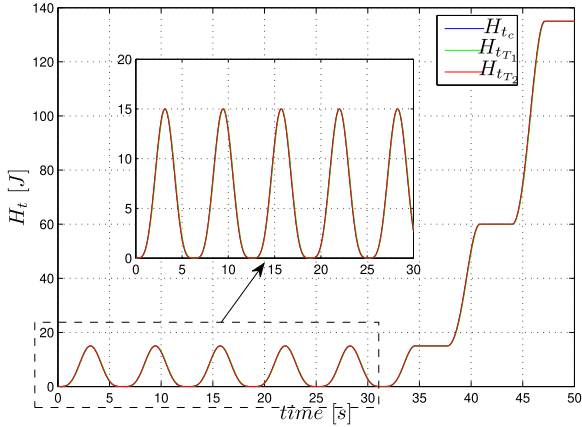


Fig. 16. Translational energy corrected with the proposed integrator considering sampling time (T_1 and T_2) and comparison with the continuous case ($H_{t,c}$).

following energy balance still holds:

$$E_{\text{obs}_2}(k) = -\Delta H_{t,i}(k) + \sum_{m=1}^k \sum_{i=1}^3 \beta_{1,i}(m-1) f_{e,i}^2(m-1)T \quad (28)$$

where i is the i th component of the vector and $\Delta H_{t,i}(k) = \frac{T^2 f_{e,i}^2}{2M_{i,i}}$. This leads to

$$E_{\text{obs}_2}(k) = \sum_{i=1}^3 E_{\text{obs}_{2,i}}(k) \quad (29)$$

where

$$E_{\text{obs}_{2,i}}(k) = E_{\text{obs}_{2,i}}(k-1) - \Delta H_{t,i}(k) + \beta_{1,i}(k-1) f_{e,i}^2(k-1)T. \quad (30)$$

Thus, the coefficients of the damping matrix $\beta_1 = \text{diag}(\beta_{1,1}, \beta_{1,2}, \beta_{1,3})$ are defined as

$$\beta_{1,i}(k) = \begin{cases} -\frac{E_{\text{obs}_{2,i}}(k)}{f_{e,i}^2(k)T} & E_{\text{obs}_{2,i}}(k) < 0 \\ 0 & \text{else} \end{cases}. \quad (31)$$

This will provide the exact amount of damping required in (26) to restore the energy behavior of the system. For validation, we run the simulation with the same conditions as described in Section II-B. Using the proposed approach the extra energy has been dissipated as it can be seen in Fig. 16. It is worth comparing Fig. 6 (problem statement) with Fig. 16.

2) *Passive Integration Scheme for Rotational Dynamics:* For the rotational dynamics, the natural power port to be considered is given by the pair $(\tau_e(k), \omega_s(k))$. Similarly to (26), it would be possible to implement a dissipative action on this port while exploiting the external torque $\tau_e(k)$.

However, in case of free motion, i.e., $\tau_e(k) = 0$, no energy can be dissipated. In particular, during free motion, $\Delta H_{r_2} = 0$ as evident from (23), but ΔH_{r_1} can be greater than zero. Thus, it is not sufficient to consider only the natural port (τ_e, ω_s) for dissipating all the energy produced by the discrete integration

for the rotational case. To overcome this problem, we design a fictitious port (χ, ω_s) where the PC is applied. The torque $\chi \in \mathbb{R}^3$ is composed of the external torque τ_e and a fictitious torque $\tau_c = f(\omega_s, I)$ expressed as

$$\chi(k) = \tau_e(k) + \underbrace{\frac{1}{2} T^2 \omega_s(k)^T S(I \omega_s(k))^T I^{-1} S(I \omega_s(k))}_{\tau_c} \quad (32)$$

which allows us to explicitly consider the contribution of ΔH_{r_1} at the port level [38]. The definition of χ allows us to measure the real and fictitious torques which can generate the extra energy terms ΔH_{r_1} and ΔH_{r_2} produced by the integration. In this way, the term τ_e takes into account the presence of ΔH_{r_2} while the term τ_c takes into account the presence of ΔH_{r_1} . Thus, even during free motion, $\chi \neq 0$ and it is possible to use the TDPA for dissipating the energy produced by the Euler integrator.

We can now exploit the PO/PC approach by defining the energy observer as

$$E_{\text{obs}_3}(k) = E_{\text{obs}_3}(k-1) - (\Delta H_{r_1}(k) + \Delta H_{r_2}(k)) + T \chi(k-1)^T \beta_2(k-1) \chi(k-1). \quad (33)$$

Unlike the translational case where the dynamics is independent along its components, the rotational dynamics is coupled. Therefore, for the dissipative action, we design a single damping coefficient (β_2) which acts in all the directions of the angular velocity. Otherwise, compensating in different directions will lead to a distortion of the reproduced dynamics. Therefore, if some generated energy is detected ($E_{\text{obs}_3} < 0$), the time-varying damper $\beta_2(k)$ will be modulated as follows:

$$\beta_2(k) = \begin{cases} -\frac{E_{\text{obs}_3}(k)}{T \|\chi(k)\|^2} & \text{if } E_{\text{obs}_3} < 0 \\ 0 & \text{else} \end{cases} \quad (34)$$

where $\|\chi(k)\|^2 = \chi(k)^T \chi(k)$.

The velocity modification provided by the passivity controller is given by

$$\omega_{\text{pc}}(k) = \beta_2(k) \chi(k). \quad (35)$$

This is used for correcting the output of the Euler integrator as

$$\omega_c(k) = \omega_s(k) - \omega_{\text{pc}}(k) \quad (36)$$

where $\omega_c(k)$ represents the velocity commanded to the robot.

As it can be seen in Fig. 17, here, the energy drifts discussed in the problem statement are removed with the proposed passive integration scheme. It is worth comparing Fig. 8 (before applying the method) and Fig. 17 (with the proposed method).

C. Algorithm of the Passive Integrator

Using the results of Sections V-A and V-B, it is possible to build a passive and explicit discrete integrator for a rigid body dynamics. We will group these results in an algorithmic form in Algorithm 1 in order to provide a pseudocode that can be easily executed in real time on industrial robots. The algorithm computes the extra energy produced and it builds the fictitious variable χ . Then, it observes if there is a violation of passivity and it computes the gains of the passivity controllers (that will be

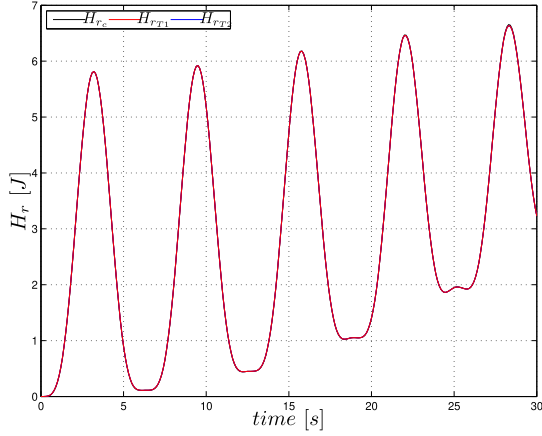


Fig. 17. Proposed integrator corrects the energy drift—rotational energy in continuous time (H_{r_c}) and discrete time ($H_{r_{T1}}$, $H_{r_{T2}}$).

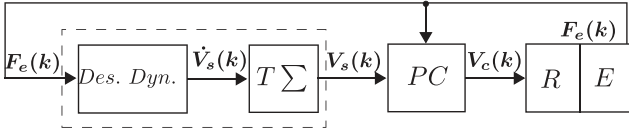


Fig. 18. Scheme with admittance PC for compensating the discrete-time integration effects.

Algorithm 1: Passive Integrator for Satellite Dynamics.

Input: $v_s(k-1)$, $\omega_s(k-1)$, $f_e(k)$, $\tau_e(k)$, M , I , T

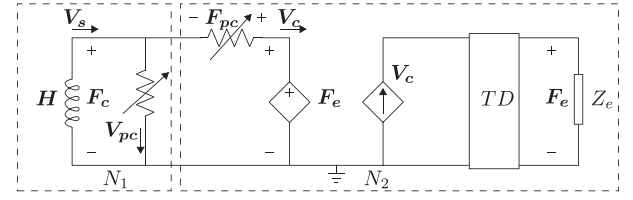
Output: $v_c(k)$, $\omega_c(k)$

- 1: Compute ΔH_t using (21)
- 2: Compute ΔH_r using (24)
- 3: Build the $\chi(k)$ variable using (32)
- 4: Compute $E_{obs2}(k)$ using (30)
- 5: Compute $E_{obs3}(k)$ using (33)
- 6: Set the damping $\beta_1(k)$ using (31)
- 7: Set the damping $\beta_2(k)$ using (34)
- 8: Compute $v_c(k) = v_s(k) - \beta_1(k)f_e(k)$
- 9: Compute $\omega_c(k) = \omega_s(k) - \beta_2(k)\chi(k)$
- 10: Output: $V_c(k) = (v_c(k), \omega_c(k))$

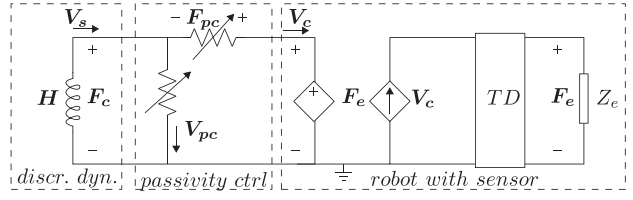
zero if passivity is not violated). Finally, the dissipative actions of the PCs are exploited for updating the velocities commanded to the robot. Fig. 18 shows the admittance scheme modified accordingly, where the PC provides the corrected velocity $V_c = (v_c, \omega_c)$ to the robot.

VI. OVERALL ARCHITECTURE

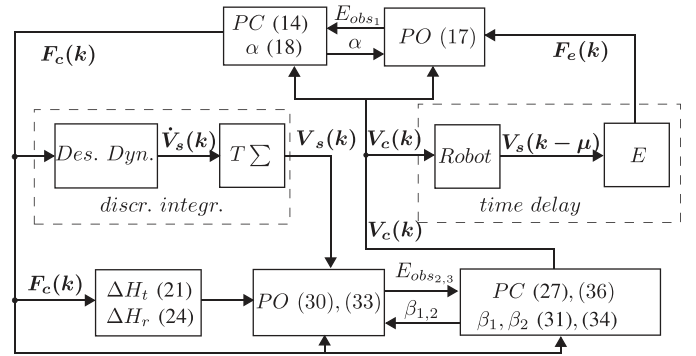
In the previous sections, passivity controllers were developed to deal with the time delay and the discrete-time integration effects separately. Here, we propose a framework which combines both controllers. As shown in Fig. 19(a), two networks have been introduced, namely, N_1 with the variable resistor where the resulting current is V_{pc} and N_2 with voltage drop F_{pc} .



(a)



(b)



(c)

Fig. 19. Overall passivity-based control for simulating satellite dynamics on a robot in presence of time delay and discretization effects. (a) Interconnection of passive networks. (b) Passivity control interface in electrical domain. (c) Block diagram of the overall architecture with the passivity controllers.

The passivity controller variables $V_{pc} = (v_{pc}, \omega_{pc})$ and F_{pc} will render the respective networks passive. Therefore, the overall system is given by the interconnection of the two passive networks (N_1, N_2). Interconnection of passive networks leads to an overall passive system that has no energy production [43]. Having proven that the overall system is now passive, a 2-port network named *passivity ctrl* is isolated from Fig. 19(a) by combining the two passivity control variables, namely F_{pc} and V_{pc} . This 2-port network acts as an interface or passivity transformation from the simulated discrete dynamics (*discr. dyn.*) network to the *robot with sensor* network and is shown in Fig. 19(b). Fig. 19(c) shows the block diagram with input–output variables and the respective energy observers and passivity controllers with references to the corresponding equations.

Notice that the corrected wrench F_c is obtained by (14), where F_{pc} in (13), $E_{obs1,i}$ in (17), and α_i in (18) are recalculated considering the corrected velocity V_c instead of V_s . Similarly, the corrected twist in network N_2 , namely, $V_c = (v_c, \omega_c)$ is obtained by the corrected velocity (27) and the corrected angular velocity (36), where v_{pc} in (26), $E_{obs2,i}$ in (30), $\beta_{1,i}$ in (31), and ΔH_t in (21) are recalculated considering the corrected force $F_c = (f_c, \tau_c)$ instead of $F_e = (f_e, \tau_e)$. Similarly E_{obs3} ,

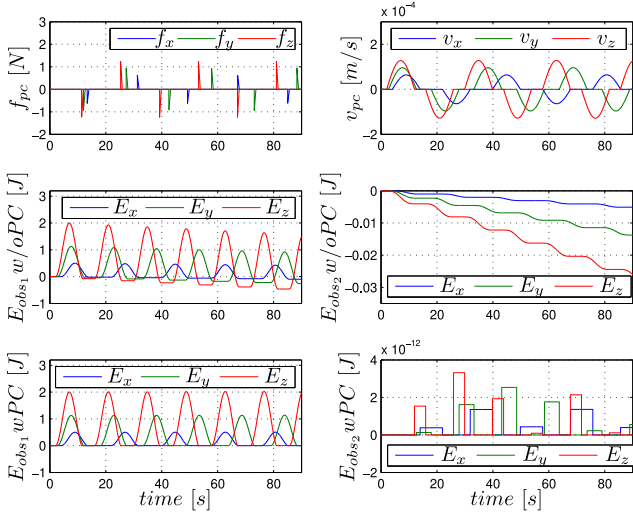


Fig. 20. Translational dynamics-force correction \mathbf{f}_{pc} and velocity correction \mathbf{v}_{pc} , energy observers without and with PC. Last row (E_{obs} w PC) indicates the passivity of the system.

$\Delta H_{r1}(k)$, $\Delta H_{r2}(k)$, $\chi(\mathbf{k})$ are recalculated from (23) and (32) with the corrected values τ_c and ω_c to recompute ω_{pc} from (35). Therefore, the robot will receive a modified twist \mathbf{V}_c (to correct the effects of the discrete integration) and the simulated dynamics receives as input a modified wrench \mathbf{F}_c (to correct the effects of time delay).

VII. RESULTS WITH THE OVERALL ARCHITECTURE

The overall architecture has been verified first in simulation and later experimentally validated.

For the simulation case, we consider the satellite mass of $M = 280$ kg and $\mathbf{I} = \text{diag}(18, 20, 22)$ kgm² (i.e., client satellite for the DEOS mission [44]). The considered sampling time is 4 ms and the time delay in the loop 40 ms. Forces–torques act on the simulated satellite as it collides against the virtual walls.

Fig. 20 shows the results for the translational dynamics. The first column shows the passivity control for time-delay compensation where the correction is provided in impedance mode with the forces \mathbf{f}_{pc} . The second column is the passivity control for compensating the effects of discrete integration which provides an admittance correction \mathbf{v}_{pc} . Both corrections are generated to remove the activities measured by the energy observer and shown in the second row of Fig. 20. As it can be seen, the energies observed without the passivity controllers, (E_{obs1} w/o PC and E_{obs2} w/o PC) become negative, indicating activity in the system. The third row shows the energies with the passivity controllers (E_{obs1} w PC and E_{obs2} w PC). The positive semidefiniteness of these energies indicate the passivity of the system for translational dynamics.

Similarly, Fig. 21 shows the corresponding results for the rotational dynamics where τ_{pc} is the torque correction related to the passivity control for the time delay. The second column is related to the passivity control which provides an angular velocity correction ω_{pc} to avoid the discrete-time integration effects. Also, for the rotation dynamics, the energies with the

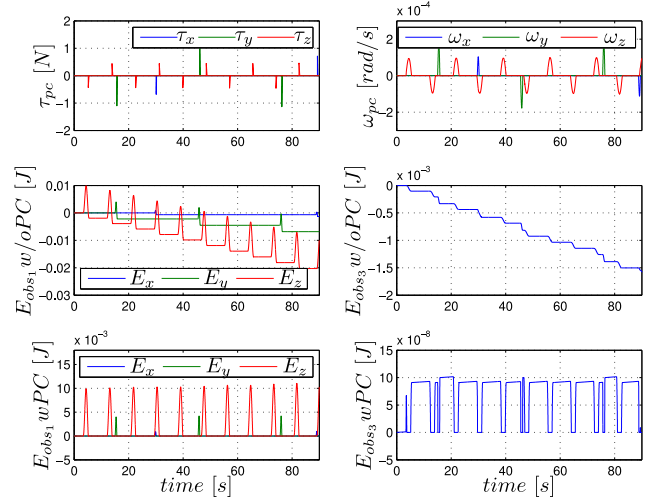


Fig. 21. Rotational dynamics-torque correction τ_{pc} , angular velocity correction ω_{pc} energy observers without and with PC. Last row (E_{obs} w PC) indicates the passivity of the system.

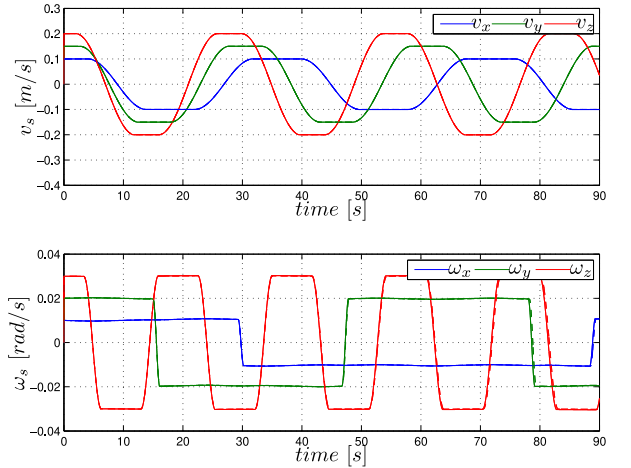


Fig. 22. Stable behavior of the system-linear and angular velocities with the passivity approach to correct time delay and integration effects. The dashed line is the ideal velocity.

passivity control (E_{obs1} w PC and E_{obs3} w PC) shown in the last row of Fig. 21 are positive semidefinite and therefore, all the activity has been dissipated. As a result, the diverging behavior are resolved and the motion of the rigid body is stable and energetically consistent. It can be seen in Fig. 22 where the translational and rotational velocities of the rigid body are shown.

A. Experiments

The proposed control architecture is applied and validated on the client robot of the OOS-SIM facility shown in Fig. 1. The industrial robot (KR-120) is equipped with a 6-DoF force–torque sensor at its end-effector to measure external interaction and the satellite mock-up is connected onto the sensor. Robot and sensor run in real time with a frequency of 250 Hz. We run the model-based dynamics in a real-time VxWorks computer, which

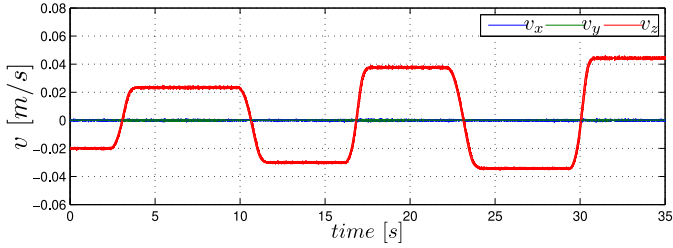


Fig. 23. Experiment-increase in the robot velocity due to the time delay and discretization without passivity control.

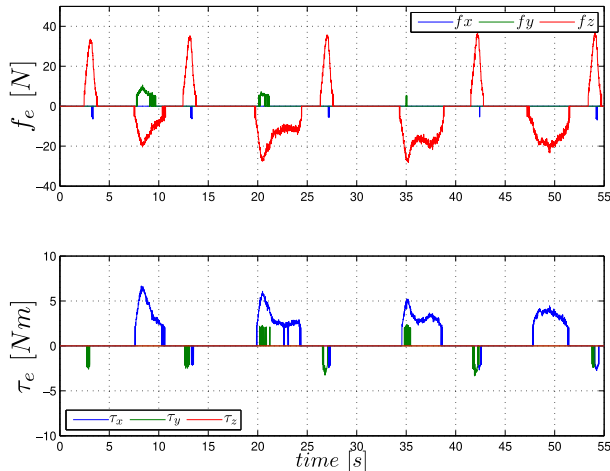


Fig. 24. Experiment-forces and torques measured by the sensor.

receives also the data from the sensors and computes the desired set point using inverse kinematics. We use the internal set-point tracking controller of the robot (KRC-4) through the RSI (Robot Sensor Interface) which enables communication with the real-time computer. More details can be found in [12].

The experiment considers a rigid body with a mass of 700 kg and $I = \text{diag}(116, 160, 160)$ kgm^2 (i.e., satellite in the DEOS mission [44]) with the intrinsic time delay of the facility. The satellite mock-up has an initial linear velocity of $[0, 0, -0.02]$ m/s and its motion is obstructed by two fixed physical constraints on which the mock-up will make contacts, see Fig. 1. The contact point 1 in the figure is a fixed surface and the contact point 2 is the gripper of the light weight robot which is stationary.

1) *Robot Motion*: Fig. 23 shows the increase in the velocity of the satellite simulated with the robot during its interaction with the environment without the proposed approach. As it can be seen the velocity diverges after each contact leading to an unstable behavior.

Now, the proposed passivity-based approach is applied. Forces and torques are measured by the sensor during the contacts as shown in Fig. 24 and the active energy is observed and dissipated during the experiment. Therefore, the velocity of the robot does not increase as can be seen in Fig. 25. The following results show the behavior of the passivity controllers and the energy observers to prove the passivity of the system

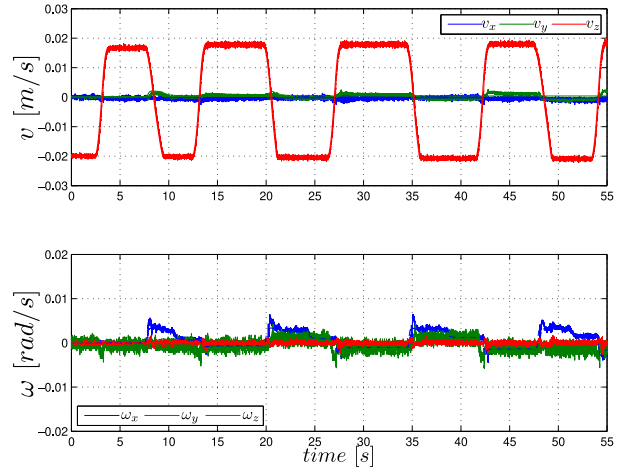


Fig. 25. Experiment-linear (top) and angular velocity (bottom) of the robot with the proposed approach. The velocity does not increase.

for the translational and rotational dynamics (Figs. 26 and 27, respectively).

2) *Translational Dynamics*: Fig. 26(a) shows the impedance correction f_{pc} to avoid the activity due to the time delay which is observed with the $E_{\text{obs}1}$ w/o pc. The activity due to the discretization is shown in Fig. 26(b) (middle) which shows a negative trend of energy. This is corrected with the passivity controller which provides a correction v_{pc} [see Fig. 26(b) top] to restore the passivity. Passivity proof is given by the positive semidefiniteness of E_{obs} w pc in Fig. 26(a) and (b).

3) *Rotational Dynamics*: The action of the passivity control dealing with the time delay for the rotational dynamics is shown in Fig. 27(a). In particular, τ_{pc} is the passivity correction due to the energy generated by the time delay, which is shown in Fig. 27(a) middle. The activity due to the discretization is corrected by the admittance passivity control which provides a velocity correction ω_{pc} shown in Fig. 27(b) top. This allows to restore the passivity properties of the system as proved in Fig. 27(a) and (b) bottom. The positive semidefiniteness of the energy indicates the passivity of the system.

VIII. DISCUSSION

The results obtained with the passivity-based approach are promising in both simulation and experiment. The experiments proved that the developed method can run on a real-time robot and can also deal with sensor noise which is intrinsic in measured data. In this article, a simple rigid body dynamics has been considered (as representative of the satellite) without any flexibility or disturbance which might require a deeper analysis for the discrete-time integrator. The method relies on the observability of the 1-port (F_e, V_s) which can be obtained from force-torque sensor and commanded velocity signals, both at high sampling rate.

The extra energies introduced by the Euler integration as reported in (21) and (24) are calculated in discrete time. These values might be slightly different with respect to the real energetic disparity between discrete and continuous-time cases. This

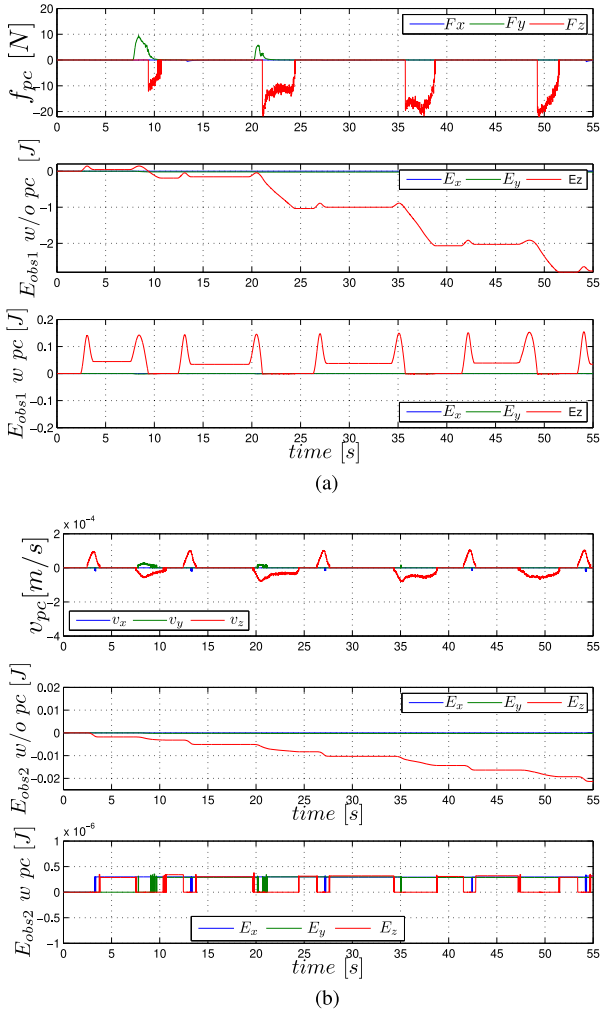


Fig. 26. Experiment-passivity controller for compensating (a) time delay and (b) discretization effects for the translational dynamics. Last rows in (a) and (b) indicate the passivity of the system. (a) Force corrected by the PC, energy observer without the PC and energy with PC. (b) Linear velocity corrected by the PC, energy observer without PC and energy with PC.

error is given by the loss of information due to discretization and cannot be avoided. However, this difference becomes smaller as the sampling time decreases [45].

The sampling time of the considered setup is 4 ms and the correction of the PC results to be in the order of 10^{-4} m/s and 10^{-4} rad/s, as can be seen in Fig. 20 (top-right) and Fig. 21 (top-right). However, the correction of the PC in force (to compensate the time delay) also has an effect on the final velocity. A numerical analysis has been performed to evaluate the total velocity error (linear \tilde{v} and angular $\tilde{\omega}$) between the desired velocity (ideal case without time delay) and the velocity with time delay and passivity controllers acting. This error can be seen in Fig. 28 where the maximum error in velocity is below 0.001 m/s and -0.3 deg/s. These values adhere, for instance, to the requirements of 0.01 m/s and 0.5 deg/s for a rendezvous and synchronization space scenario, as reported in [46, Sec. 8]. Furthermore, the deviation from the nominal desired velocity

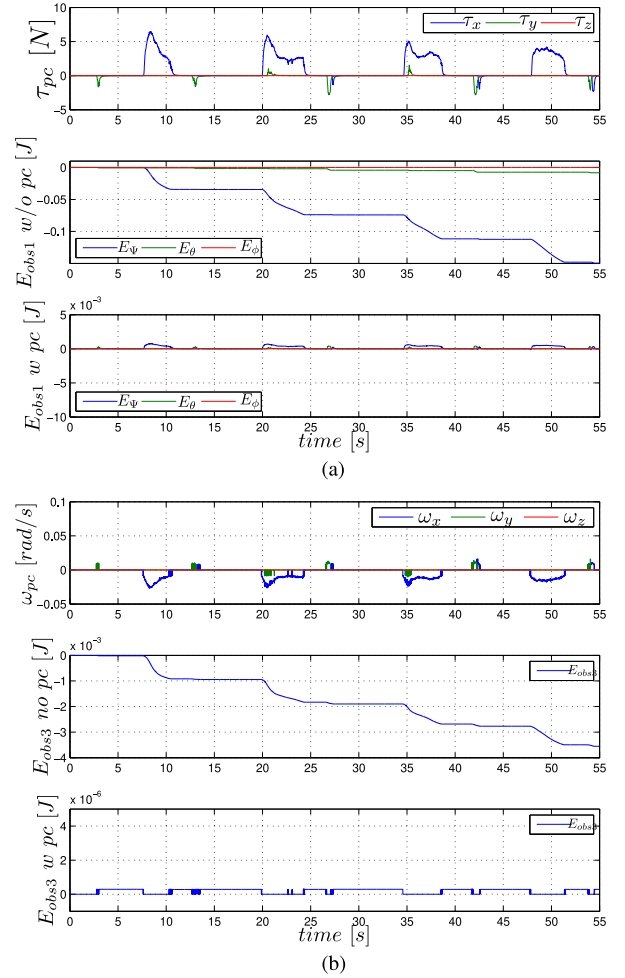


Fig. 27. Experiment-passivity controller for compensating (a) time delay and (b) discretization effects for the rotational dynamics. Last rows in (a) and (b) indicate the passivity of the system. (a) Torque corrected with the PC, energy observer without the PC and energy with PC. (b) Angular velocity corrected by the PC, energy observer without PC and energy with PC.

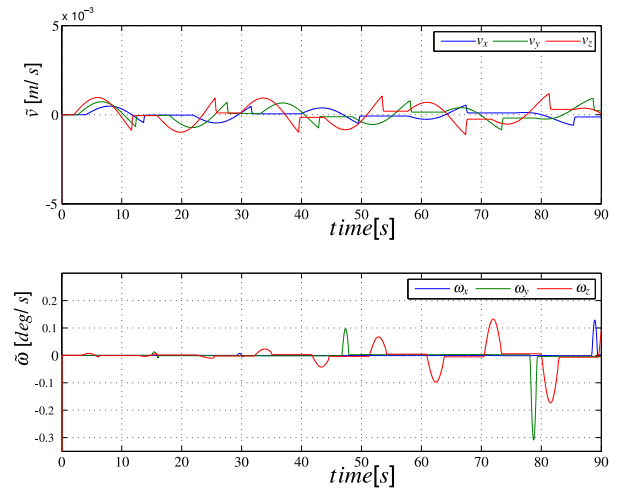


Fig. 28. Velocity correction (linear \tilde{v} and angular $\tilde{\omega}$) between the desired velocity (ideal case without time delay) and the velocity with time delay and passivity controllers acting.

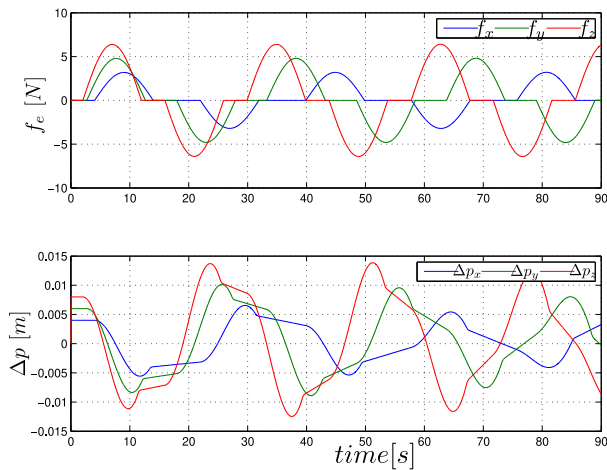


Fig. 29. Forces (top) and error between desired and measured position (bottom).

trajectory has also been calculated while using the overall architecture. This can be seen in Fig. 29. The plot at the top shows the forces applied and the plot at the bottom shows the variation of position caused by the compensation with respect to the ideal case. The error values are in the order of 0.015 m.

High-frequency force and velocity modifications are generally reported as a limitation of the TDPA damping injection [47]. The high-frequency damping generates an effect which might modify the natural dynamic behavior of the system. This effect can be minimized by using passive filters in the force and velocity signals to remove the chattering effects. However, in our approach, extra filters are not required since the system dynamics already act as filters implicitly. The chattering in the force modification for time-delay compensation is filtered out by the integration of the dynamics in (1) and (2). The chattering in the velocity correction is already small due to the low sampling times used in our facility. Furthermore, the velocity commands are sent to the internal controller of the industrial robot (in most cases, a controller with integral action) which also filters out the chattering.

IX. CONCLUSION

Reproducing satellite dynamics using on-ground robotic facilities is a challenging task. From one side, robotic simulators have the advantage of reproducing effective gravity compensation on ground and allow motion in large workspaces which is needed for a space mission. On the other side, the motion replicated by the robot is affected by intrinsic time delays in the loop and discrete data integration. This will affect the fidelity of the simulation. As it was shown in the first part of this article, time delay in the loop and discrete integration of the dynamics might cause system instability and an inconsistent simulation from the energy point of view.

In this article, passivity-based approaches were exploited to faithfully reproduce rigid-body dynamics with robotic systems in spite of these inevitable factors. The proposed model-free methods can also be applied in industrial robots whose dynamics are not known or cannot be easily identified. In order to compensate the damaging effects of intrinsic time delays,

a passivity controller which modulates the force at the input of the admittance dynamics was proposed. A second passivity controller which deals with the effects of the discrete integration was proposed and this modulated the velocity output of the integrator in order to preserve the energetic properties. Both controllers were used in a unified architecture which acted as a passivity layer between the discrete system and the real robot. The proposed methods avoid unstable behavior in reproducing satellite dynamics while ensuring an energy consistent simulation, as validated on an industrial robot.

REFERENCES

- [1] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, "A review of space robotics technologies for on-orbit servicing," *Prog. Aerosp. Sci.*, vol. 68, pp. 1–26, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0376042114000347>
- [2] B. Wilcox, D. B. Gennery, B. Bon, and T. Litwin, "Real-time model-based vision system for object acquisition and tracking," *Proc. SPIE*, vol. 0754, pp. 276–281, 1987.
- [3] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl, "Sensor-based space robotics-ROTEX and its telerobotic features," *IEEE Trans. Robot. Autom.*, vol. 9, no. 5, pp. 649–663, Oct. 1993.
- [4] K. Yoshida, "Engineering test satellite VII flight experiments for space robot dynamics and control: Theories on laboratory test beds ten years ago, now in orbit," *Int. J. Robot. Res.*, vol. 22, no. 5, pp. 321–335, 2003. [Online]. Available: <https://doi.org/10.1177/0278364903022005003>
- [5] E. Freund, J. Rossmann, and M. Schluse, "Real-time collision avoidance in space: The GETEX experiment," *Proc. SPIE*, vol. 4196, pp. 255–267, 2000.
- [6] S. Dubowsky and E. Papadopoulos, "The kinematics, dynamics, and control of free-flying and free-floating space robotic systems," *IEEE Trans. Robot. Autom.*, vol. 9, no. 5, pp. 531–543, Oct. 1993.
- [7] J. L. Schwartz, M. A. Peck, and C. D. Hall, "Historical review of air-bearing spacecraft simulators," in *J. Guid., Control, Dyn.*, vol. 26, no. 4, pp. 513–522, 2003.
- [8] C. Menon *et al.*, "Issues and solutions for testing free-flying robots," *Acta Astronautica*, vol. 60, no. 12, pp. 957–965, 2007.
- [9] J. Paul *et al.*, "INVERITAS: A facility for hardware-in-the-loop long distance movement simulation for rendezvous and capture of satellites and other autonomous objects," *Acta Astronautica*, vol. 116, pp. 1–24, 2015.
- [10] T. Boge, T. Wimmer, O. Ma, and T. Tzschichholz, "EPOS—Using robotics for RvD simulation of on-orbit servicing missions," in *Proc. Guid., Navigat., Control Co-located Conf.*, Aug. 2010, pp. 1–15.
- [11] Z. Milenkovic and C. D'Souza, "The Space Operations Simulation Center (SOSC) and closed-loop hardware testing for Orion Rendezvous system design," in *Proc. Guid., Navigat., Control Co-located Conf.*, Aug. 2012, doi:10.2514/6.2012-5034.
- [12] J. Artigas *et al.*, "The OOS-SIM: An on-ground simulation facility for on-orbit servicing robotic operations," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 2854–2860.
- [13] C. Qi, X. Zhao, F. Gao, A. Ren, and Q. Sun, "Contact stiffness and damping identification for hardware-in-the-loop contact simulator with measurement delay compensation," *Acta Astronautica*, vol. 123, no. C, pp. 171–180, 2016.
- [14] M. Zebenary, R. Lampariello, T. Boge, and D. Choukroun, "A new contact dynamics model tool for hardware-in-the-loop docking simulation," in *Proc. 11th i-SAIRAS*, Turin, Italy, 2012.
- [15] K. Osaki, A. Konno, and M. Uchiyama, "Delay time compensation for a hybrid simulator," *Adv. Robot.*, vol. 24, no. 8/9, pp. 1081–1098, 2010.
- [16] S. Abiko, Y. Satake, X. Jiang, T. Tsujita, and M. Uchiyama, "Delay time compensation based on coefficient of restitution for collision hybrid motion simulator," *Adv. Robot.*, vol. 28, no. 17, pp. 1177–1188, 2014.
- [17] C. Secchi, S. Stramigioli, and C. Fantuzzi, *Control of Interactive Robotic Interfaces: A Port-Hamiltonian Approach* (Springer Tracts in Advanced Robotics). Berlin, Germany: Springer, 2007.
- [18] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, pp. 2035–2057, 2006.
- [19] G. Niemeyer and J.-J. E. Slotine, "Telemanipulation with time delays," *Int. J. Robot. Res.*, vol. 23, no. 9, pp. 873–890, 2004. [Online]. Available: <https://doi.org/10.1177/0278364904045563>

- [20] S. Stramigioli, C. Secchi, A. J. van der Schaft, and C. Fantuzzi, "Sampled data systems passivity and discrete port-hamiltonian systems," *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 574–587, Aug. 2005.
- [21] C. Secchi, F. Ferraguti, and C. Fantuzzi, "Catching the wave: A transparency oriented wave based teleoperation architecture," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 2422–2427.
- [22] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli, "Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 741–756, Aug. 2011.
- [23] F. Ferraguti *et al.*, "An energy tank-based interactive control architecture for autonomous and teleoperated robotic surgery," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1073–1088, Oct. 2015.
- [24] B. Hannaford and J.-H. Ryu, "Time domain passivity control of haptic interfaces," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2001, vol. 2, pp. 1863–1869.
- [25] C. Preusche, G. Hirzinger, J.-H. Ryu, and B. Hannaford, "Time domain passivity control for 6 degrees of freedom haptic displays," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2003, vol. 3, pp. 2944–2949.
- [26] K. Hertkorn, T. Hulin, P. Kremer, C. Preusche, and G. Hirzinger, "Time domain passivity control for multi-degree of freedom haptic devices with time delay," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 1313–1319.
- [27] C. Ott, J. Artigas, and C. Preusche, "Subspace-oriented energy distribution for the time domain passivity approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 665–671.
- [28] M. De Stefano, J. Artigas, W. Rackl, and A. Albu-Schaeffer, "Passivity of virtual free-floating dynamics rendered on robotic facilities," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 781–788.
- [29] M. De Stefano, J. Artigas, and C. Secchi, "An optimized passivity-based method for simulating satellite dynamics on a position controlled robot in presence of latencies," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2016, pp. 5419–5426.
- [30] E. Hairer, G. Wanner, and C. Lubich, *Geometric Numerical Integration (Computational Mathematics)*. Berlin, Germany: Springer, 2006.
- [31] S. Blanes and F. Casas, *A Concise Introduction to Geometric Numerical Integration*. Boca Raton, FL, USA: CRC Press, 2016.
- [32] S. Ober-Bloebaum, "Discrete mechanics and optimal control," Ph.D. dissertation, University of Paderborn, Paderborn, Germany, 2008.
- [33] P. Betsch and S. Uhlar, "Energy-momentum conserving integration of multibody dynamics," *Multibody Syst. Dyn.*, vol. 17, no. 4, pp. 243–289, 2007.
- [34] K. Modin and G. Söderlind, "Geometric integration of Hamiltonian systems perturbed by Rayleigh damping," *Bit Numer. Math.*, 2011.
- [35] J. Brown and J. Colgate, "Minimum mass for haptic display simulations," in *Proc. ASME Int. Mech. Eng. Congr. Expo.*, 1998, pp. 249–256.
- [36] D. Lee and K. Huang, "On passive non-iterative varying-step numerical integration of mechanical systems for haptic rendering," in *Proc. ASME Dyn. Syst. Control Conf.*, Ann Arbor, MI, USA, Oct. 2008.
- [37] M. De Stefano, R. Balachandran, J. Artigas, and C. Secchi, "Reproducing physical dynamics with hardware-in-the-loop simulators: A passive and explicit discrete integrator," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2017, pp. 5899–5906.
- [38] M. De Stefano, J. Artigas, and C. Secchi, "A passive integration strategy for rendering rotational rigid-body dynamics on a robotic simulator," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2017, pp. 2806–2812.
- [39] L. Villani and J. de Schutter, "Force control," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., Berlin, Germany: Springer, 2016.
- [40] C. T. Landi, F. Ferraguti, L. Sabattini, C. Secchi, M. Bonf, and C. Fantuzzi, "Variable admittance control preventing undesired oscillating behaviors in physical human-robot interaction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2017, pp. 3611–3616.
- [41] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2008.
- [42] A. van der Schaft, *L₂-Gain and Passivity Techniques in Nonlinear Control (Communication and Control Engineering)*. Berlin, Germany: Springer-Verlag, 2000.
- [43] H. Khalil, *Nonlinear Systems*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.
- [44] "DEOS functional specification," DLR and SpaceTech, Immenstaad, Germany, Jan. 2009. [Online]. Available: <https://spacetechnology.com/products/missions-satellites/deos>
- [45] M. De Stefano, "Energy-based control for simulation of multi-body dynamics using robotic facilities," Ph.D. dissertation, Univ. Modena Reggio Emilia, Modena, Italy, Mar. 2019.
- [46] J. Telaar *et al.*, "GNC architecture for the e.Deorbit mission," *Proc. 7th Eur. Conf. Aeronaut. Space Sci.*, Jul. 2017, doi: [10.13009/EUCASS2017-317](https://doi.org/10.13009/EUCASS2017-317).
- [47] Y. S. Kim and B. Hannaford, "Some practical issues in time domain passivity control of haptic interfaces," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (Cat. No. 01CH37180)*, 2001, vol. 3, pp. 1744–1750.



Marco De Stefano received the bachelor's degree in aerospace engineering and the master's degree cum laude in aeronautical engineering from the University of Rome La Sapienza, Rome, Italy, in 2008 and 2011, respectively, and the Ph.D. degree in industrial innovation engineering (summa cum laude) from the University of Modena and Reggio Emilia, Modena, Italy, in 2019.

Since 2012, he has been a Researcher with the Department for Analysis and Control of Advanced Control System, German Aerospace Center (DLR), Wessling, Germany. His research interests include hardware-in-the-loop simulation, stability analysis, and control of space robots.



Ribin Balachandran received the bachelor's degree in electrical and electronics engineering from the National Institute of Technology, Surathkal, India, in 2009, and the master's degree in automation and robotics from the Technical University of Dortmund, Dortmund, Germany, in 2013. He is currently working toward the Ph.D. degree in the field of teleoperation and shared control with the Technical University of Munich (TUM), Munich, Germany.

Since 2013, he has been a Researcher with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Oberpfaffenhofen, Germany. His main research areas are time-delayed teleoperation, haptics, and control for robotics.



Cristian Secchi received the M.Sc. degree in computer science engineering from the University of Bologna, Bologna, Italy, in 2000 and the Ph.D. degree in information engineering from the University of Modena and Reggio Emilia, Modena, Italy, in 2004.

He has been Visiting Student with the University of Delft (NL) and the University of Twente (NL), in 2000 and 2002, respectively. He is currently an Associate Professor of Robotics with the University of Modena and Reggio Emilia. He has authored or coauthored more than 200 papers on international journals and conferences. His research interests include human–robot physical interaction, telerobotics, mobile robotics, and surgical robotics.

Dr. Secchi was a participant of the CROW (Coordination of AGVs in Automatic Warehouses) Project that has been selected as a Finalist for the 2010 EUROP/EURON Technology Transfer Award for the Best Technology Transfer Project in Europe. His Ph.D. thesis has been selected as one of the three finalists of the Fifth Georges Giralt Award for the Best Ph.D. Thesis on Robotics in Europe. He is an Associated Editor for the IEEE ROBOTICS AND AUTOMATION LETTERS. He has been an Associate Editor for the IEEE TRANSACTIONS ON ROBOTICS from 2012 to 2017 and for the IEEE ROBOTICS AND AUTOMATION MAGAZINE from 2005 to 2008. He has been Co-Chair of the IEEE Robotics and Automation Society Technical Committee on Telerobotics from 2007 to 2012. He has been an Associate Editor for IROS2012 and the ICRA2013 and ICRA2015 Conference Editorial Boards.