2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

Reproducing Physical Dynamics with Hardware-in-the-Loop Simulators: A Passive and Explicit Discrete Integrator

Marco De Stefano[1,2], Ribin Balachandran[1], Jordi Artigas[1] and Cristian Secchi[2]

Abstract— **In this paper we present a passive and reliable explicit discrete integrator, which allows to preserve the energy and dynamic properties of a physical body rendered on a hardware-in-the-loop simulator. Starting from the standard Euler integrator, we identify the energy generation that results from the integration process. This energy makes the time discrete dynamics deviate from the ideal one, resulting in position drifts or stability issues. By exploiting the time domain passivity approach, the simulated dynamics is reshaped in order to preserve its physical energy properties. The proposed integration method allows precise simulation of virtual bodies on industrial robot facilities. The method has been validated in simulation and experimentally tested on the DLR OOS-SIM facility.**

I. Introduction

The removal of space debris in orbit and maintenance tasks on defected satellites represent a significant challenge for future space missions. A popular solution for tackling this problem consists of the use of a spacecraft equipped with a manipulator arm that can safely collect the debris or operate on a satellite.

To achieve this goal, simulating the operating scenario under controlled conditions before the mission is important. One of the main difficulties to test space control on-ground is to reproduce the micro-gravity effect. Some technologies are already available such as air-bearing system, free-fall, suspended systems and robotic simulators [1]. The robotic-based simulators are Hardware In the Loop (HIL) systems where a mathematical model of the dynamics to simulate (e.g. satellite, vehicle) is exploited for feeding the robot(s) that reproduces the desired behavior. They can implement active gravity compensation and allow motion within the workspace. Some of these robotic simulators are: the SOCS (Lockheed Martin Space Operation Simulation Center) for testing the Orion rendezvous [2], the EPOS (European Proximity Operation Simulator) to simulate rendezvous and docking [3], INVERITAS a facility for rendezvous and capture of satellites [4] and the OOS-SIM, an on-ground experimental facility for on-orbit servicing simulation [5]. The OOS-SIM, for example, is composed of an industrial robot with a manipulation arm which simulates a servicer satellite and by another industrial robot which simulates the dynamics of a target satellite (see Fig. 1). In these systems, the output of a force/torque sensor is provided to a mathematical model of the dynamics to be reproduced, which is integrated for computing the desired kinematic input (position or velocity) to the robot. Finally, the robot moves in the Cartesian space emulating the motion of the desired dynamics. The most common behavior to reproduce is the rigid body dynamics, which mimics the evolution of a satellite in space both during free motion and while in contact with other bodies (e.g. the servicer during the grasping). Despite the simplicity of the dynamics to reproduce, the complexity of the robotic simulator introduces several technological problems that, if not properly addressed, prevent it from accurately implementing the desired dynamics or even cause an unstable behavior of the system.

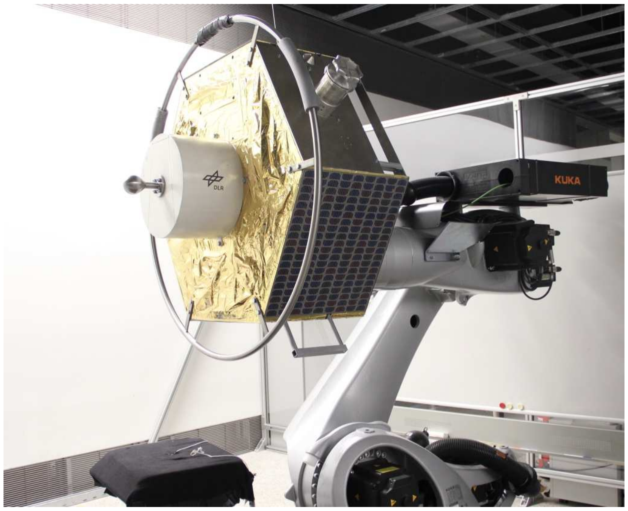

Fig. 1: Hardware-in-the-loop simulator at DLR

A. Related works

In [6] the stability of a similar system is analytically investigated considering a scaling on the mass and inertia parameters. In [7] the destabilizing effect due to the delay in the transmission of the force/torque data is analyzed and in [8] an energy-based strategy for optimally coping with this issue has been proposed.

A major problem for a stable and reliable implementation of the desired dynamics is the methodology in which the dynamics is discretized for producing the desired set-point to the industrial robot. Usually, the dynamics is achieved using standard explicit integration techniques (e.g. Euler integration) since they are simple, fast and suitable for a real-time implementation. Nevertheless, it is known (see e.g. [9]) that using standard integration techniques for implementing

[1] The authors are with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany.

[2] The authors are with the University of Modena and Reggio Emilia, 41100 Modena, Italy.

Contact: marco.destefano@dlr.de

978-1-5090-4632-4/17/$31.00 ©2017 IEEE

a dynamics subject to external forces, lead to a generation of energy and, therefore, to non-physical dynamics and to large drifts. This undesired behavior becomes more evident for large sample times, which is a common situation in industrial robots, where the control frequency is usually low.

Several geometric integrators for physical systems, i.e. numerical integration methods that preserve energy and/or other geometric quantities (e.g. symplectic structure), have been developed over the years [10], [11], [12]. Some energy-preserving integrators are also available in the literature (e.g. the energy momentum method [13]). Nevertheless, geometric integration deals mainly with isolated physical systems (e.g. galaxies or other astronomical system), or with systems with some damping [14], and the interaction with the external environment is not considered.

In haptics, where an operator has to interact with a virtual environment, the problem of passively (i.e. while preserving its energetic properties) integrating a non-isolated physical dynamics is relevant. In [15], it is shown that standard explicit integrators do not ensure passivity and that, therefore, a more complex (and harder to implement) integration strategy has to be sought. In [16], an implicit integration method, based on the port-Hamiltonian formulation of the dynamics to simulate, has been proposed and in [17], a fast but implicit and variable rate integration strategy for implementing mass-spring-damper systems is illustrated.

Implicit and variable rate integration methods can be cumbersome to implement on a standard industrial robot.

B. Contribution

Current implicit integration methods require a numerical and iterative solution of the updated equation for each time step, which typically prohibits real-time determinism. Thus, we aim at developing an explicit integrator starting from the standard Euler method by adjusting the output for meeting the passivity constraint. Specifically, we will implement a rigid body dynamics using the discrete Euler integration method; we identify the energy produced and then we will keep track of it at each integration step. The integrated value will be further updated in order to dissipate the produced energy and to make the overall system a passive integrator. In order to modulate the output of the integrator, we will exploit the Time Domain Passivity Approach (TDPA) proposed in [18] and widely used in haptics for damping out the excess energy produced by the virtual environment. Recently, TDPA has been exploited for increasing the impedance that can be rendered by admittance-type haptic interfaces [19]. In our work we will combine the TDPA with the Euler integration for getting a simple and controlled explicit passive integrator. Therefore, the main contribution of the paper is twofold. First, we develop a new controlled explicit integrator that allows to reproduce the passive behavior of a rigid body dynamics independently of the discrete-system sampling time. Such an integrator can be easily implemented in standard industrial robots. Second, we show that the proposed integrator allows to achieve the desired performance on a real satellite simulator, the OOS-SIM in Fig. 1.

The paper is organized as follows: Sec. II introduces the dynamics we aim at simulating and states the problem due to the discretization. In Sec. III the energy term, which causes the energy drift, is identified and the passive explicit integrator is presented. Simulation results are discussed in Sec. IV with a real application on an industrial robot proposed in Sec. V. Conclusions and future works are discussed in Sec. VI.

II. Problem Statement

The architecture of a generic robotic simulator is illustrated in Fig. 2. $\mathbf{F(k)}$ represents the total wrench applied to the end effector of the robot (i.e. to the object to be simulated) during the interaction with the environment. The wrench is provided as an input to the desired dynamics (in the dashed box) where the acceleration is computed and discretely integrated with a sample time T. Thus, it will provide the twist $\mathbf{v(k)}$ as a set-point to the robot which will consequently reproduce the desired behavior. We will assume that the robot can perfectly track the desired set-point. This is a common assumption with industrial robots and it can be achieved by properly tuning the gains of the low level controllers. In particular, in this paper, we will consider only the Cartesian dynamics. Thus, the desired dynamics is defined to be:

$$\mathbf{M}\dot{\mathbf{v}} = \mathbf{F}, \tag{1}$$

where $\mathbf{M} = diag(m_i) \in \mathbb{R}^{3\times 3}$ is the desired virtual mass which we want to simulate. $\mathbf{F} \in \mathbb{R}^3$ is the force applied to the robot during its interaction with the environment and $\mathbf{v} \in \mathbb{R}^3$ is the Cartesian velocity of the end-effector. Moreover, Let $H = \frac{1}{2}\mathbf{p}^T\mathbf{M}^{-1}\mathbf{p}$ be the kinetic energy of the system (1), where $\mathbf{p} = \mathbf{M}\mathbf{v} \in \mathbb{R}^3$ is the momentum. The dynamics can be reformulated in a port-Hamiltonian form as:

$$\begin{pmatrix} \mathbf{v} \\ \dot{\mathbf{p}} \end{pmatrix} = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial H}{\partial \mathbf{x}} \\ \frac{\partial H}{\partial \mathbf{p}} \end{pmatrix} + \begin{pmatrix} 0 \\ I \end{pmatrix} \mathbf{F}, \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^3$ represents the Cartesian configuration and $\dot{\mathbf{x}} = \mathbf{v}$. $0, I \in \mathbb{R}^{3\times 3}$ are the null and the identity matrix respectively. As for any port-Hamiltonian system without damping, we have that the following balance holds [9]:

$$\dot{H} = \mathbf{F}^T\mathbf{M}^{-1}\mathbf{p} = \mathbf{F}^T\mathbf{v}, \tag{3}$$

which represents the fundamental energetic property of any undamped mechanical system, namely that the power due to the interaction with the environment is energetically stored in

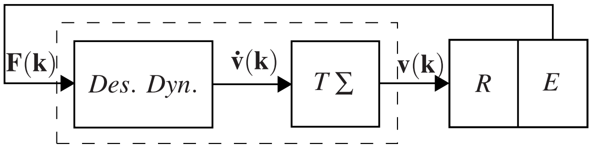

Fig. 2: Admittance architecture with the desired dynamic - R is the robot, E is the environment, *Des. Dyn.* is the force-acceleration model of the dynamics to implement, $T\Sigma$ is the discrete integrator with time step T.

the system. Notice that if there is not interaction (i.e. $\mathbf{F}=0$), then the energy stored in the system is constant (i.e. $\dot{H}=0$). The model (2) can be rewritten as:

$$\begin{cases} \mathbf{v}=\mathbf{M}^{-1}\mathbf{p} \\ \dot{\mathbf{p}}=\mathbf{F}. \end{cases} \tag{4}$$

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

$$\begin{cases} \mathbf{x(k)}=\mathbf{x(k-1)}+T\mathbf{M}^{-1}\mathbf{p(k)} \\ \mathbf{p(k)}=\mathbf{p(k-1)}+T\mathbf{F(k-1)}, \end{cases} \tag{5}$$

where the second line is equivalent to the following velocity integration strategy:

$$\mathbf{v(k)}=\mathbf{v(k-1)}+T\mathbf{M}^{-1}\mathbf{F(k-1)}, \tag{6}$$

In case of free motion (i.e. $\mathbf{F(k-1)}=0$), the momentum and, consequently, the energy of the system are constant over time. Thus, in this very simple case, straight Euler integration is energetically well posed since it allows the discretized dynamics to behave physically independently of the sample time.

Unfortunately this well posedness does not hold anymore in case of interaction. This can be easily shown by a 1-DOF example. Consider the force profile, shown at the top of Fig. 3, which acts on a mass of 30 kg. The integration of the dynamics is considered in the continuous case and compared with the Euler discrete integrator for sampling time: $T_1=0.1$ s, $T_2=0.01$ s. Fig. 4 clearly shows the increase in the energy which is introduced into the system with respect to the continuous time integrator (H_c is the energy calculated in continuous time). Notice that the larger the sampling time, the larger is the increase of energy which leads to a drift in the position. The drift due to the integration with T_1 reaches 0.05 m (when the force profile acts between 0s and 32s) and 0.15 m between 32 s and 50 s, (see Fig. 3 middle). Also for the case with T_2, the drift appears. Since the sampling time is smaller, it results in a drift 10 times lower, as shown in Fig. 3 bottom. This drift causes inconsistency in simulating the desired dynamics with a discrete integrator that is usually implemented for rendering a desired dynamics with a robot. The robot will receive position commands accordingly but, as it has been shown, the energy properties of the simulated mass will be not preserved. Such a drift may lead the robot to interact to unforeseen objects that produce new (drifted) behaviors leading to a deteriorated performance of the system.

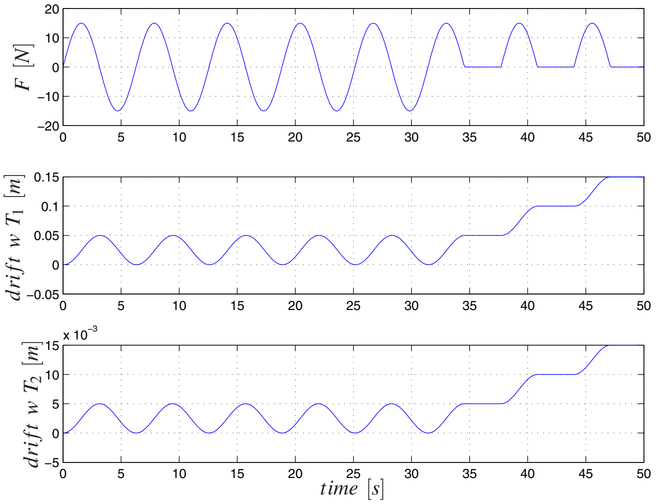

Fig. 3: Force profile, drift in position due to the discretization with (w) T_1 and T_2.

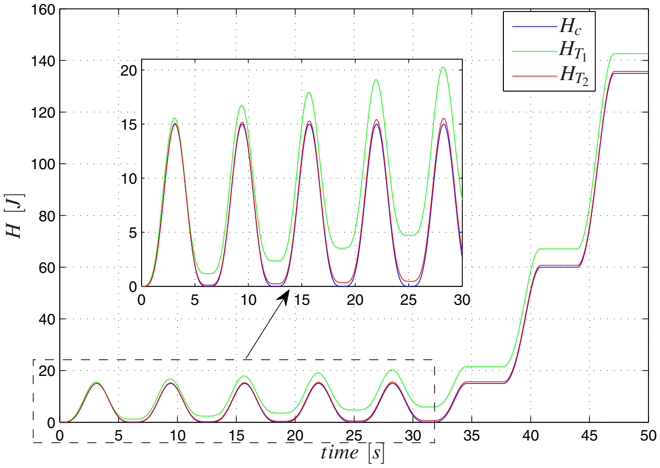

Fig. 4: Problem statement: mechanical energy considering different sampling time (H_{T_1} and H_{T_2}) and comparison with the continuous case H_c.

The goal of this work is to design a controlled Euler integration method that preserves the energetic balance in (3) in the discrete case. In this way, it will be possible to reproduce by the robotic simulator the behavior (1) while preserving its energetic properties independently of the sample time.

III. The passivity-based integration method

As shown in Sec. II, the extra energy due to the discrete integration makes the energy behavior of the discrete dynamic system different from that of its continuous counterpart.

In this section, we formally identify the extra energy introduced by the discrete integrator and we exploit this information for adjusting the velocity output of the Euler integrator using the TDPA. An analysis of the energy behavior of the continuous and discrete time systems is presented and the passivity-based integrator scheme is introduced.

A. Energy produced by the Euler integration method

Consider the dynamics (1) discretized by means of the Euler method and reported in (5). The discrete kinetic energy $H(k)$ is given by:

$$H(k)=\frac{1}{2}\mathbf{p(k)}^T\mathbf{M}^{-1}\mathbf{p(k)}=$$
$$\frac{1}{2}[\mathbf{p(k-1)}+T\mathbf{F(k-1)}]^T\mathbf{M}^{-1}[\mathbf{p(k-1)}+T\mathbf{F(k-1)}], \tag{7}$$

where the discrete dynamics has been considered. By simply reordering the terms we obtain:

$$H(k) = \frac{1}{2}\mathbf{p}(\mathbf{k-1})^T\mathbf{M}^{-1}\mathbf{p}(\mathbf{k-1}) + T\mathbf{v}(\mathbf{k-1})^T\mathbf{F}(\mathbf{k-1}) \\ + \frac{1}{2}T^2\mathbf{F}(\mathbf{k-1})^T\mathbf{M}^{-1}\mathbf{F}(\mathbf{k-1}), \qquad (8)$$

where $\mathbf{v}(\mathbf{k-1}) = \mathbf{M}^{-1}\mathbf{p}(\mathbf{k-1})$. We can then write:

$$H(k) = H(k-1) + T\mathbf{v}(\mathbf{k-1})^T\mathbf{F}(\mathbf{k-1}) \\ \underbrace{+\frac{1}{2}T^2\mathbf{F}(\mathbf{k-1})^T\mathbf{M}^{-1}\mathbf{F}(\mathbf{k-1})}_{\Delta\mathbf{H}} \qquad (9)$$

which is not a physical and passive behavior. In fact, the energy variation should be due only to the energy provided through the power port, i.e. $\mathbf{v}(\mathbf{k-1})^{\mathbf{T}}\mathbf{F}(\mathbf{k-1})T$, and the extra energy term $\Delta\mathbf{H} = \frac{1}{2}T^2\mathbf{F}(\mathbf{k-1})^T\mathbf{M}^{-1}\mathbf{F}(\mathbf{k-1})$ is just due to Euler integration.

This causes two main problems. First, as shown in Sec. II, the extra energy will cause a drift that makes the reproduced dynamics drifting with respect to the ideal one. Second, as evident from (9), the discrete dynamics is not passive and, therefore, it may happen that during the interaction with the environment, the system becomes unstable [9].

In order to simplify the presentation, since the desired dynamics (1) is decoupled, in the following analysis and in the design of the explicit passive integration scheme we will consider a single component. Thus, we will remove the bold notation, that has characterized vectors and matrices so far, and with a regular font we will indicate the generic i^{th} component of the vectors involved in the analysis. For example, the extra energy term $\Delta\mathbf{H}$ corresponding to the i^{th} component in (9) will be written as:

$$\Delta H = \frac{T^2F(k-1)^2}{2m}. \qquad (10)$$

B. Relation between the continuous and the discrete dynamics

In this subsection, we aim at understanding how accurate is ΔH, which can be computed in real time and that will be used for adjusting the output velocity of the Euler integrator (6). It is as an estimate of the energy produced when discretizing the continuous dynamics, namely of the difference between the discrete energy (E_d) and the continuous energy (E_c).

We consider the dynamics in (1) where a discrete force input $F(k)$ is commanded to the mass m and its velocity is derived using both continuous and discrete integration. The difference between energy increments of the dynamic system in one sampling cycle between the two integration methods is $\Delta E = E_d - E_c$. The analytical value of the additional energy due to discrete integration ΔH is also calculated in the simulation. Fig. 5 shows the difference between ΔE and ΔH for different sampling rates. It can be seen that the difference is small and it tends to zero as the sampling time tends to zero.

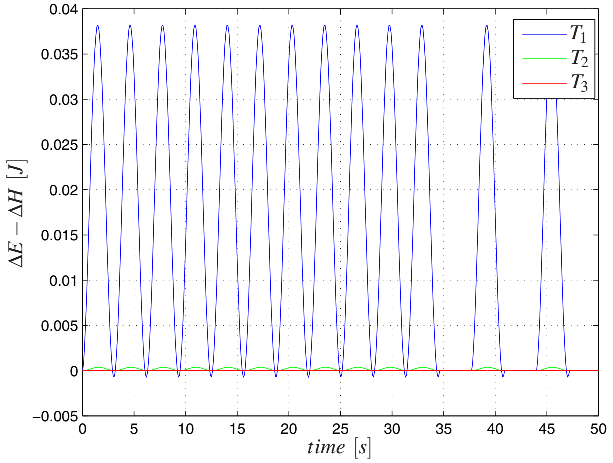

Fig. 5: Difference between analytical and identified values of additional energy: $T_1 = 0.1s$, $T_2 = 0.01s$, $T_3 = 0.001s$.

The reason for the difference between ΔE and ΔH is analyzed here. A graphical representation of the power plots for both continuous (dashed curve) and discrete (samples) dynamic systems are shown in Fig. 6. The difference between the energy increase per sampling cycle in both the systems is analyzed in the lower figures (positive, increasing power in A and negative, decreasing power in B). In part A of Fig.6, the power of the continuous system varies linearly between the samples $k-1$ and k from $P(t-T)$ to $P(t)$ since the input force $F(k-1)$ is constant during this time. The area of the shaded regions (quadrilateral $acde$) is the extra energy ΔE in the sampling time T produced by the discrete system than the continuous one. This area is the sum of the areas of the rectangle $abde$ and the triangle bcd. If $Area(Q)$ function is

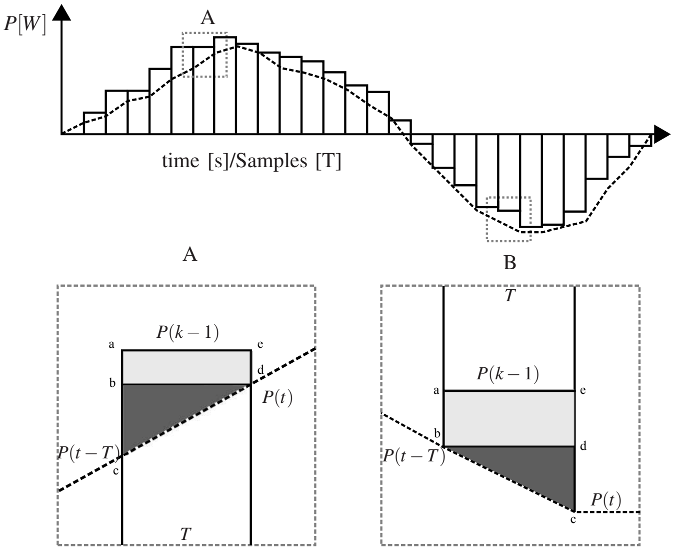

Fig. 6: Power difference between discrete and continuous time domains.

defined as the area of the polygon Q, ΔE is given by:

$$\begin{aligned}\Delta E &= Area(acde)\\ &= Area(abde) + Area(bcd)\\ &= [P(k-1) - P(t)]T + \frac{1}{2}T[P(t) - P(t-T)]\\ &= F(k-1)T[v(k-1) - v(t)]\\ &\quad + \frac{1}{2}TF(k-1)[v(t) - v(t-T)],\end{aligned}$$

where $[v(t) - v(t-T)] = \frac{F(k-1)}{m}T$. Then, it is possible to write:

$$\begin{aligned}\Delta E &= F(k-1)T[v(k-1) - v(t)] + \frac{1}{2}TF(k-1)\frac{F(k-1)}{m}T\\ &= F(k-1)T[v(k-1) - v(t)] + \frac{T^2F(k-1)^2}{2m}\\ &= F(k-1)T[v(k-1) - v(t)] + \Delta H,\end{aligned} \tag{11}$$

which results in a difference:

$$\Delta E - \Delta H = F(k-1)T[v(k-1) - v(t)]. \tag{12}$$

Similarly, in part B of Fig. 6, ΔE is given by the sum of the areas of rectangle *abde* and triangle *bcd*, which results in:

$$\Delta E - \Delta H = F(k-1)T[v(k-1) - v(t-T)]. \tag{13}$$

Equation (12) and (13) represent analytically the error between ΔE and ΔH (shown in Fig. 5). The equations clearly show that as the sampling rate increases, ΔE gets closer to ΔH since $v(k-1)$ gets closer to $v(t)$ and $v(t-T)$. In the graphical analysis, this turns to be a smaller area of rectangle *abde*.

This analysis shows that there is always a difference between what we can estimate in real time, namely ΔH, and the real difference between the energetic behavior in the discrete case and in the real case. This error is due to the loss of information related to the discretization process and it can not be avoided. Such a difference gets smaller as the sample time gets lower.

However, adjusting the output velocity for recovering the passivity of the discrete model has several advantages. First, a physical behavior of the discrete dynamics is ensured. The evolution will be close to the ideal one in the limits reported in (12). Second, a stable interactive behavior is achieved thanks to the passivity of the discrete dynamics.

C. Passive Integration scheme

TDPA is a passivity ensuring tool widely applied in the fields of haptics and time-delayed teleoperation. The underlying principle of TDPA is to observe the input and output energy flow (with the Passivity Observer) of a single-port network, (the virtual environment, in case of haptics) or a 2-port network (the communication channel with delay, in case of teleoperation) [20]. The passivity condition for a two-port network is given by:

$$\sum_{k=0}^{nT}(F_1(k)v_1(k)T + F_2(k)v_2(k)T) + E(0) \geq 0, \tag{14}$$

where (F_i, v_i) and $E(0)$ are the power correlated variable sets of port $i = 1, 2$, and the initial energy storage of the network respectively. If condition (14) holds, the system is defined to be passive. The extra energy generated in the port that violates the passivity condition is dissipated with a time-varying damper, the Passivity Controller (PC). In an admittance like architecture, the equation for the observed energy is:

$$\begin{aligned}E(k) = E(k-1) &+ \sum_{k=0}^{nT}(F_1(k)v_1(k)T + F_2(k)v_2(k)T)\\ &+ \beta(k-1)F(k-1)^2T,\end{aligned} \tag{15}$$

where β is a time-varying damper later discussed. We can represent the discretization problem with a network analogy proposed in Fig. 7 where E_c can be seen as the energy in continuous time which, due to the discretization, assume a value of E_d. Considering the 2-port as shown in Fig. 7, the energy observer then becomes,

$$\begin{aligned}E_{obs}(k) &= E_{obs}(k-1) + E_c(k) - E_d(k)\\ &\quad + \beta(k-1)F(k-1)^2T,\\ &= E_{obs}(k-1) - \Delta E(k) + \beta(k-1)F(k-1)^2T,\\ &\approx E_{obs}(k-1) - \Delta H(k) + \beta(k-1)F(k-1)^2T,\end{aligned} \tag{16}$$

where $\Delta E(k)$ has been approximated to $\Delta H(k)$ as per equation (11). Notice that the energy observer will measure an active energy as soon as there is an external force (which causes ΔH).

At each integration step E_{obs} must be greater than zero for ensuring the passivity. Therefore, it is possible to define the time-varying damper $\beta(k)$, function of the observed energy (16), as follows:

$$\beta(k) = \begin{cases} -\frac{E_{obs}(k)}{F(k)^2T} & E_{obs}(k) < 0\\ 0 & else. \end{cases} \tag{17}$$

The velocity corrected by the PC is given by the following quantity:

$$v_{pc}(k) = \beta(k)F(k). \tag{18}$$

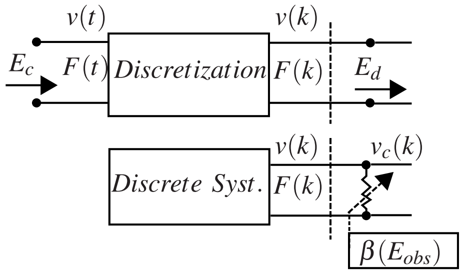

Fig. 7: Continuous energy (E_c) and discrete energy (E_d). Analogue discrete system with the designed variable damper.

Thus, if active energy is detected (i.e. $E_{obs} < 0$), the corrected velocity is sent to the robot as follows:

$$v_c(k) = \begin{cases} v(k) - \beta(k)F(k) & E_{obs} < 0 \\ v(k) & else. \end{cases} \qquad (19)$$

where $v(k)$ is achieved by (6). Therefore, the observed energy will be $E_{obs}(k) \geq 0$ making the network, i.e. the integrator, passive. A schematic of the integration scheme is in Fig. 8. The variable damping β is modulated by the observed active energy and the applied force which will activate the *PC* (in (18), (19)) to provide the corrected velocity v_c to the robot.

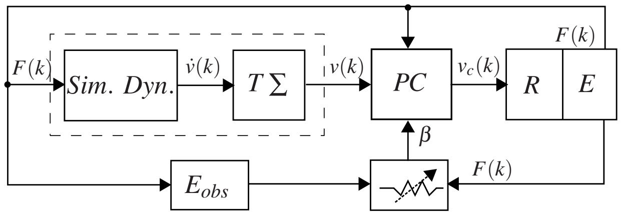

Fig. 8: The passivity-based integration scheme.

IV. Simulations

The method is firstly verified in simulation where the proposed integration scheme is applied. We consider the same case as in Sec. II (i.e. mass and force profile) for the same sampling time $T_1 = 0.1s$ and $T_2 = 0.01s$. The top of Fig. 9 shows the active energy observed in the system running with T_1. This leads to a drift in the energy as it has been described in Sec. II.

However, this active energy is compensated by the PC that adjusts the velocity (see middle of Fig. 9 for the velocity corrected by the *PC*).

Thus, the passivity proof is provided at the bottom of Fig. 9 which shows that the observed energy has always a non-negative value according to the passivity condition in (14).

A second simulation has been performed for the case with T_2 as it is shown in Fig. 10. The active energy is here dissipated by virtue of the PC and the system results to be passive (see Fig. 10 bottom).

In order to verify that the mechanical properties are preserved, a comparison with the continuous time integrator is shown in Fig. 11. H_c is the energy calculated in continuous time and H_{T1}, H_{T2} is the energy calculated as results of the correction with the proposed method with T_1 and T_2, respectively. The energy drift discussed in the problem statement in Fig. 4 is now corrected as illustrated in Fig. 11. This proves that the discrete dynamics with the proposed integration method behaves passively in discrete time and it preserves the energy properties of the simulated rigid body.

V. Experiment

The experiment is carried out on an industrial robot which is part of the OOS-SIM facility [5] and can simulate the dynamics of a satellite. The robot is equipped with a force sensor at its end-effector to measure external interaction. Both, plant and sensor run in real-time with a frequency of 250 Hz, thus the considered sampling time is 4 *ms*. The experiment is performed for the presented dynamics where the simulated mass is 250 *kg*, typical value of a satellite. Fig. 12, Fig. 13 and Fig. 14 show the data along the components (z, y, x) measured during the experiment. Each figure shows the energy observed without (w/o) PC, the energy with (w) PC, the damping coefficient β, the velocity corrected by the *PC*, the measured force F and the relative position of the robot, respectively. As it can be seen, the E_{obs} w/o the PC results in a negative energy which can produce an active behavior. However, this active energy is corrected by the PC which commands a velocity correction v_{pc} as a function of the damping coefficient β to preserve the passivity condition. Indeed, the passivity proof of the method is described by the energy observed with the PC in each figure which results to be positive, thus, passive

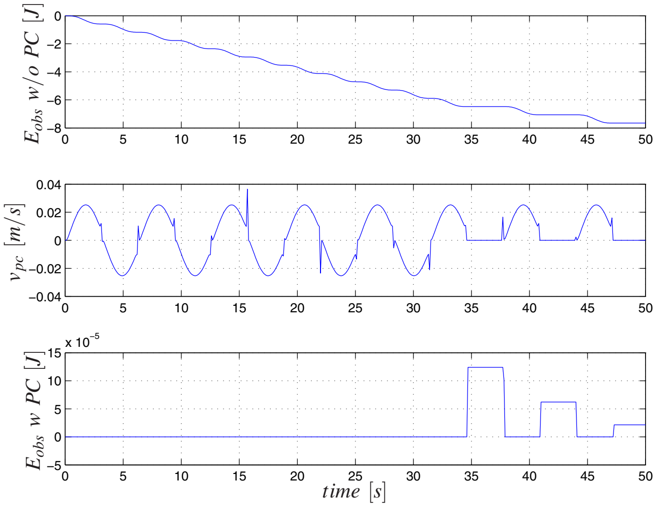

Fig. 9: Sim. 1 - E_{obs} without (w/o) PC, velocity corrected by the *PC* and E_{obs} with (w) PC for $T_1 =\ 0.1s$.

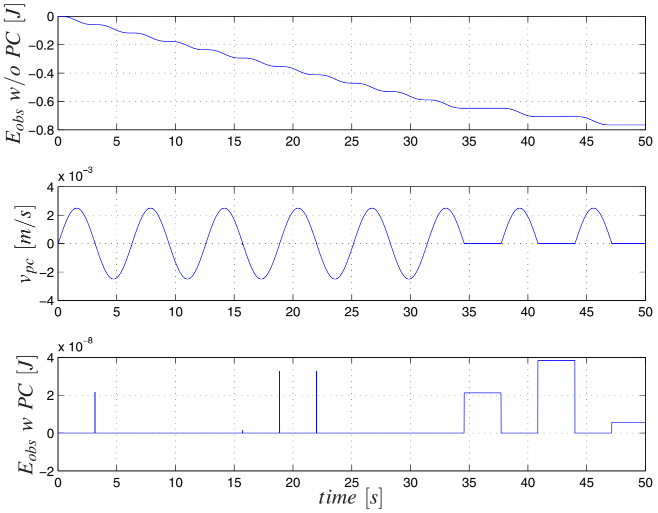

Fig. 10: Sim. 2 -E_{obs} without (w/o) PC, velocity corrected by the *PC* and E_{obs} with (w) PC for $T_2 =\ 0.01s$.

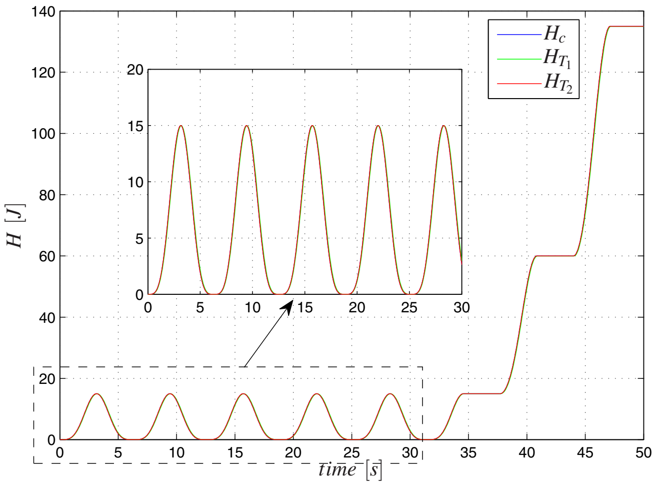

Fig. 11: Mechanical energy with the proposed method considering sampling time (T_1 and T_2) and comparison with the continuous case (H_c).

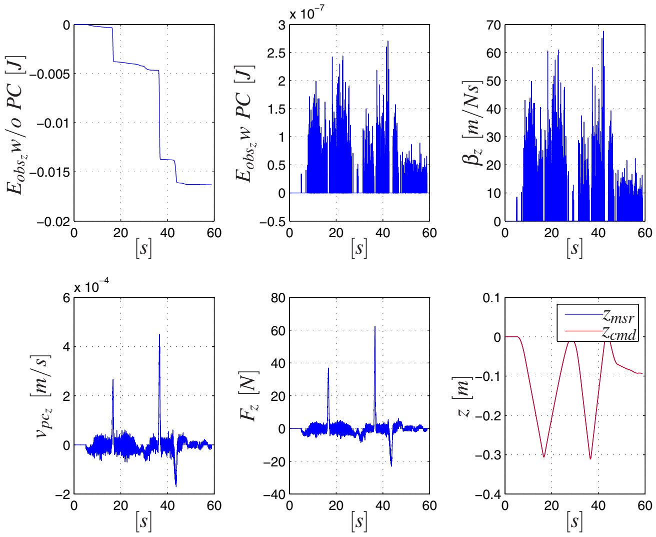

Fig. 12: Experiment results - Energy observed (without and with PC), damping factor β_z, velocity corrected by the *PC*, forces measured F_z and motion of the robot in z.

accordingly with (14). The difference in the observed energy (and consequently the v_{pc}) for each direction is due to the different magnitudes of applied forces, e.g. along the z the force reaches 60 N, lower values for x and y, therefore, the extra energy term ΔH will be different. The experiment results prove that the integration method can deal also with sensor noise and model uncertainties, typical issues intrinsic in the hardware. An additional experiment can be also seen in the accompanying video.

VI. Conclusions and Future Work

A new explicit and passive integration method for reproducing a Cartesian rigid body dynamics on a robot has been proposed. Starting from the simple and standard

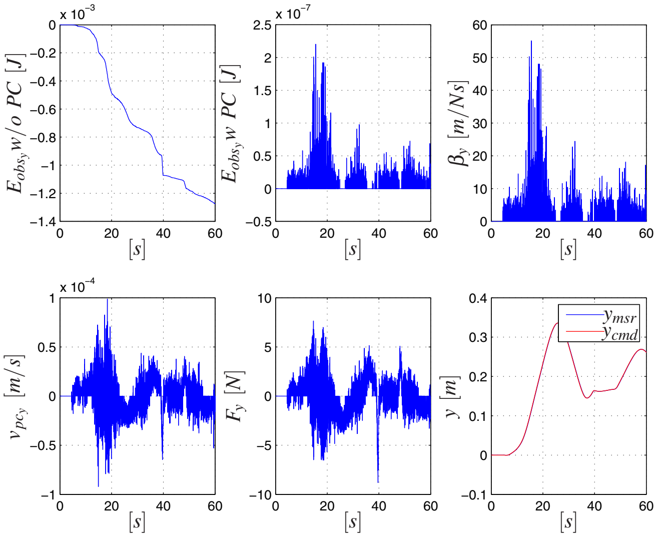

Fig. 13: Experiment results - Energy observed (without and with PC), damping factor β_y, velocity corrected by the *PC*, forces measured F_y and motion of the robot in y.

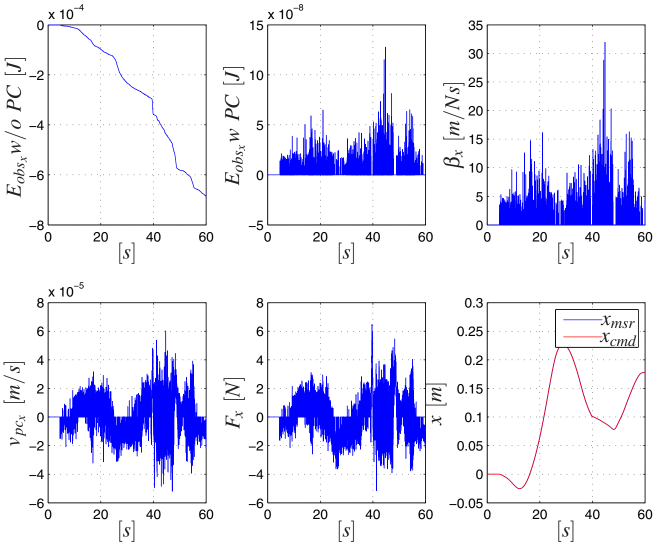

Fig. 14: Experiment results - Energy observed (without and with PC), damping factor β_x, velocity corrected by the *PC*, forces measured F_x and motion of the robot in x.

Euler integrator, we proposed a strategy based on TDPA for modifying the velocity of the Euler integrator and to render the discrete dynamics passive. The method does not consider the model of the robot which simulates the desired dynamics and, therefore, it can be applied to any robotic simulator. The effectiveness of the proposed approach has been validated in simulations and on the OOS-SIM, a robotic system that simulates a satellite dynamics.

Future work aims at extending the proposed integrator for considering the full rigid body dynamics, i.e. rotations and translations. Furthermore, we aim at extending the proposed approach to more generic scenarios (e.g. coupling between the servicer robot and the satellite).

REFERENCES

[1] J. L. Schwartz, M. A. Peck, and C. D. Hall, “Historical review of air-bearing spacecraft simulators,” in *Journal of Guidance, Control and Dynamics*, 2003, pp. AAS 03–125.

[2] Z. Milenkovic and C. D’Souza, *The Space Operations Simulation Center (SOSC) and Closed-loop Hardware Testing for Orion Rendezvous System Design*, ser. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, aug 2012, doi:10.2514/6.2012-5034.

[3] T. Boge, T. Wimmer, O. Ma, and T. Tzschichholz, *EPOS - Using Robotics for RvD Simulation of On-Orbit Servicing Missions*, ser. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, aug 2010, doi:10.2514/6.2010-7788.

[4] J. Paul, A. Dettmann, B. Girault, J. Hilljegerdes, F. Kirchner, I. Ahrns, and J. Sommer, “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.

[5] J. Artigas, M. De Stefano, W. Rackl, R. Lampariello, B. Brunner, W. Bertleff, R. Burger, O. Porges, A. Giordano, C. Borst, and A. Albu-Schaeffer, “The oos-sim: An on-ground simulation facility for on-orbit servicing robotic operations,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 2854–2860.

[6] F. Aghili and M. Namvar, “Scaling inertia properties of a manipulator payload for 0-g emulation of spacecraft.” *I. J. Robotic Res.*, vol. 28, no. 7, pp. 883–894, 2009.

[7] M. De Stefano, J. Artigas, W. Rackl, and A. Albu-Schaeffer, “Passivity of virtual free-floating dynamics rendered on robotic facilities,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 781–788.

[8] 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 *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 5419–5426.

[9] C. Secchi, S. Stramigioli, and C. Fantuzzi, *Control of Interactive Robotic Interfaces: a port-Hamiltonian Approach*, ser. Springer Tracts in Advanced Robotics. Springer, 2007.

[10] E. Hairer, G. Wanner, and C. Lubich, *Geometric Numerical Integration*, ser. Computational Mathematics. Springer, 2006.

[11] S. Blanes and F. Casas, *A Concise Introduction to Geometric Numerical Integration*. CRC Press, 2016.

[12] S. Ober-Bloebaum, “Discrete mechanics and optimal control,” *Phd thesis, University of Paderborn, Germany*, 2008.

[13] P. Betsch and S. Uhlar, “Energy-momentum conserving integration of multibody dynamics,” *Multibody System Dynamics*, vol. 17, no. 4, pp. 243–289, 2007. [Online]. Available: http://dx.doi.org/10.1007/s11044-007-9043-9

[14] K. Modin and G. Söderlind, “Geometric integration of hamiltonian systems perturbed by rayleigh damping,” *Bit Numer Math*, 2011.

[15] J. Brown and J. Colgate, “Minimum mass for haptic display simulations,” in *Proceedings of ASME International Mechanical Engineering Congress and Exposition*, 1998, pp. 249–256.

[16] S. Stramigioli, C. Secchi, A. J. van der Schaft, and C. Fantuzzi, “Sampled data systems passivity and discrete port-hamiltonian systems,” *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 574–587, Aug. 2005.

[17] D. Lee and K. Huang, “On passive non-iterative varying-step numerical integration of mechanical systems for haptic rendering,” in *Proceedings of the ASME Dynamic Systems and Control Conference*, Ann Arbor (MI), USA, Oct. 2008.

[18] B. Hannaford and J.-H. Ryu, “Time-domain passivity control of haptic interfaces,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 1–10, Feb. 2002.

[19] M. Nabeel, J. Lee, U. Mehmood, A. Jafari, J.-H. Hwang, and J.-H. Ryu, “Increasing the impedance range of admittance-type haptic interfaces by using time domain passivity approach,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept. 2015, pp. 585–590.

[20] J.-H. Ryu, D.-S. Kwon, and B. Hannaford, “Stable teleoperation with time-domain passivity control,” *IEEE Transactions on robotics and automation*, vol. 20, no. 2, pp. 365–373, 2004.