A CoppeliaSim dynamic simulator for the da Vinci Research Kit

Marco Ferro[1], Alessandro Mirante[1], Fanny Ficuciello[2] and Marilena Vendittelli[1]

***Abstract*—The design of a physics-based dynamic simulator of a robot requires to properly integrate the robot kinematic and dynamic properties in a virtual environment. Naturally, the closer is the integrated information to the real robot properties, the more accurate the simulator predicts the real robot behaviour. A reliable robot simulator is a valuable asset for developing new research ideas; its use dramatically reduces the costs and it is available to all researchers. This paper presents a dynamic simulator of the da Vinci Research Kit (dVRK) patient-side manipulator (PSM). The kinematic and dynamic properties of the simulator rely on the parameters identified in [1]. With respect to the kinematic simulator previously developed by some of the authors, this work: (i) redefines the kinematic architecture and the actuation model by modeling the double parallelogram and the counterweight mechanism, to reflect the structure of the real robot; (ii) integrates the identified dynamic parameters in the simulation model. The obtained simulator enables the design and validation of control strategies relying on the robot dynamic model, including interaction force estimation and control, that are fundamental to guarantee safety in many surgical tasks.**

***Index Terms*—Simulation and Animation; Surgical Robotics: Laparoscopy; Calibration and Identification**

I. INTRODUCTION

IN the modern surgical practice, the necessity of achieving operations safely and with elevated precision is the catalyst fostering the development of computer- and robot-assisted surgery paradigms. The da Vinci Surgical System [2] has responded with great impact to such needs, by increasing the quality and the overall outcome of the interventions in minimally-invasive surgery. The da Vinci Research Kit (dVRK) [3] has generated a further boost in this direction, providing an open-source mechatronic toolset for both clinical and research purposes.

Besides, the introduction of dedicated simulation tools in the clinical workflow has significantly supported these developments and played a key role in different stages of the medical procedures, from training and pre-operative planning to intra-operative plan adaptation. Dynamic simulators, in particular, allow to render physical contacts between surgical instruments and environment, to predict the dynamic behaviour of the robot, to validate physical interaction control laws, and algorithms for model-based interaction forces estimation and collisions/faults detection and isolation. A realistic dynamic simulation of the PSM arms of the dVRK, that integrates the dynamic parameters of the real robot, would need a detailed designing process, that consists in modelling: i) the double parallelogram structure, guaranteeing mechanically the Remote Center of Motion (RCM) constraint at the surgical tool; ii) the counterweight mechanism, balancing the motion of the surgical tool along the prismatic joint of the chain through a transmission mechanism.

An extensive survey about the research progresses with the dVRK system, including the development of dedicated simulators, has been recently provided in [4]. Several simulators are available today for training [5], [6], and simulation platforms for research have also been recently developed, with different degrees of modeling accuracy and possible dynamic simulation capabilities. The simulator UnityFlexML in [7], for example, presents a kinematic model of the PSM in the Unity game development engine, to study the interaction of the surgical tools with deformable objects in a reinforcement learning scenario. The ATAR simulator [8] replicates the surgical tools of the PSMs in a virtual training environment that is interfaced with the dVRK surgeon console. It simulates the kinematics only and it does not model the double parallelogram structure nor the counterweight mechanism. The AMBF framework [9], [10] allows dynamic simulations of the PSMs, models the parallelogram and introduces the counterweight in the simulation model. However, the dynamic parameters integrated in the model are extracted from SolidWorks, and are not those of the identified model of the real robot. In addition, no physics-based simulation appears to be linked to the counterweight since the transmission mechanism between the counterweight and the tool actuator is not modeled.

In our previous works, the dVRK simulator based on the V-REP/CoppeliaSim simulation environment [11] integrated also the possibility of interfacing virtual reality and haptic devices [12], as well as additional novel surgical tools [13]. However, it simulated a kinematically equivalent robot model, that did not include the modelling of the counterweight and with a simplified actuation model of the double parallelogram that did not correspond to the real robot.

Currently, up the the Authors' knowledge, a dynamic simulator of the dVRK combining a complete modeling of the PSMs mechanics (including the double parallelogram, the counterweight and the associated transmission model) with integrated dynamic parameters, experimentally identified and

Manuscript received: June, 17, 2022; Revised October, 2, 2022; Accepted October, 30, 2022.

This paper was recommended for publication by Editor Aniket Bera upon evaluation of the Associate Editor and Reviewers' comments *(Corresponding author: Marco Ferro).*

[1]Marco Ferro, Alessandro Mirante and Marilena Vendittelli are with Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, Via Ariosto 25, 00185, Rome, Italy, {ferro,vendittelli}@diag.uniroma1.it

[2]Fanny Ficuciello is with Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Università di Napoli Federico II, via Claudio, 21, 80125 Napoli NA, Italy, fanny.ficuciello@unina.it

Digital Object Identifier (DOI): see top of this page.

simulation framework	dynamic robot simulation	model of the doubleparallelogram	counterweight dynamic simulation	identified real-robot dynamic parameters
UnityFLexML [7]	no	yes	no	no
ATAR [8]	no	no	no	no
V-REP dVRK simulator [11]–[13]	no	partially	no	no
AMBF [9]	yes	yes	no	no[a]
proposed simulator	yes	yes	yes	yes

[a]Potentially they could be integrated upon appropriate transformations that depend on the simulation model.

TABLE I: Comparative scheme of the main dVRK simulators available to the research community. The first column reports the name of the simulation framework, the subsequent columns the functionalities that it integrates.

consistent with the real system, is lacking.

In this paper, we modify the kinematic simulator of the dVRK [13] developed in CoppeliaSim, with the final objective of enabling dynamic simulation of the PSMs in the virtual environment that can reliably predict the behaviour of the real system. The provided dynamic model corresponds to the real robot mechanical structure and actuation model and is not a kinematic equivalent of the dVRK, as in [11]–[13]. For this purpose, the model of the single PSM is first redesigned to match the kinematic formalization provided in [1], by taking into account the following features: i) the explicit modelling of the closed kinematic chain due to the double parallelogram structure; ii) the simulation of the counterweight mechanism and the associated transmission system, employed on the real system to balance the motion of the surgical tool through the prismatic joint. Then, we integrate the full set of dynamic parameters of the PSM links obtained from the convex optimization procedure illustrated in [1]. A comparison of the features of interest of our proposed simulator with the mentioned solutions available to the research community is shown in Table I.

It is worth remarking that the work focuses on the dynamic simulation model of the PSMs only because, as a matter of fact, during surgical procedures the ECM is generally kept in a fixed configuration which is changed when necessary to improve the user's view, while keeping the PSMs fixed. The dynamic model of the ECM is therefore not as relevant as that of the PSMs in the procedures currently executed with the da Vinci Surgical System.

Summing up, the main contributions of this paper are: i) the model of the double parallelogram kinematic structure of the PSMs; ii) the model of the counterweight mechanism, completed by the transmission system simulation model; iii) the integration of the dynamic parameters of the real system, identified in [1], in the dynamic simulation model of the dVRK PSMs, made possible by the first two modeling contributions; iv) validation of the simulator against the identified model and the measures from the real robot.

Paper organization: Section II summarizes the identification procedure [1]; Section III presents the PSM simulation model; Section IV reports on the validation procedure and proposes control examples. Section V concludes the paper.

II. DYNAMIC PARAMETERS OF THE dVRK-PSM

The dVRK platform is composed, on the operator side, of a console with two Master Tool Manipulators (MTMs) used to teleoperate, at the patient side, a pair of Patient Side Manipulators (PSMs) mounted on a cart, together with an Endoscopic Camera Manipulator (ECM).

Each of the two PSMs of the dVRK platform is a 7-DoF actuated arm (see Fig. 1a), with 6 additional passive joints arranged in a double parallelogram structure that enables translation of the tool along its axis only and rotations around a fulcrum point, the Remote Center of Motion (RCM). The actuated DoF are organized according to a RRPRRRR sequence: the first two revolute joints, J_1 and J_2, provide roll and pitch rotation around the RCM point, the third prismatic joint J_3 allows translation along the instrument axis, while the last four revolute joints, J_4, J_5, J_6 and J_7, yield mobility of the *EndoWrist*, i.e., orientation and opening/closure motion of surgical tool (see Fig. 1b). The manipulator also hosts a counterweight (see Fig. 1c), balancing the tool motion along J_3 through a tendon driven transmission mechanism. A torsional spring is included at J_4, to favor the joint return to the angular zero position [14].

Denoting with $n = 7$ the number of active joints, the modelling process of the robot dynamic model in [1] relates joint positions, velocities and accelerations $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$, taking values in $\mathbb{R}^n$, to the joint torques $\boldsymbol{\tau} \in \mathbb{R}^n$ in the form:

$$\boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\delta} = \boldsymbol{\tau}, \tag{1}$$

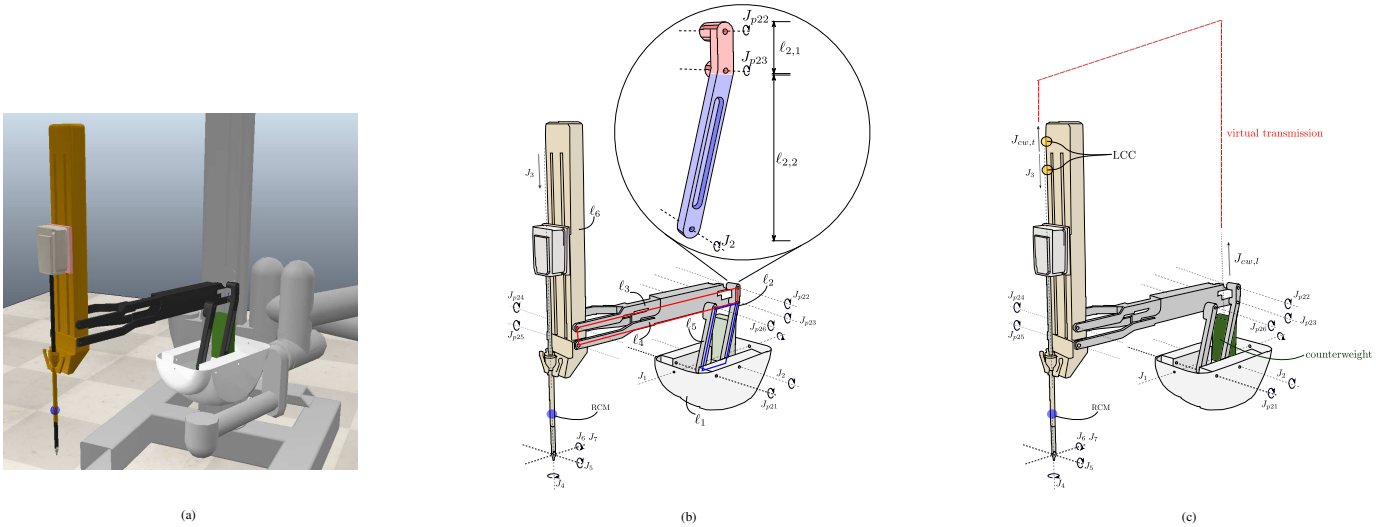
where $\boldsymbol{H} \in \mathbb{R}^{n \times p}$ is the *regressor* matrix and $\boldsymbol{\delta} \in \mathbb{R}^p$ collects the p constant dynamic parameters to be identified.

For each link ℓ_k, $k = 1, \ldots, N$, $N = 15$, denote by m_k its mass and by $\boldsymbol{r}_k \in \mathbb{R}^3$, $\boldsymbol{L}_k \in \mathbb{R}^{3\times3}$ respectively the Center of Mass (CoM) position and the symmetric inertia tensor, expressed in the link frame. The dynamic parameters of interest are collected, in barycentric form [15], in the vector: $\boldsymbol{\delta}_{L,k} = (m_k, \boldsymbol{l}_k^T, L_{xx}^k, L_{xy}^k, L_{xz}^k, L_{yy}^k, L_{yz}^k, L_{zz}^k)^T$, with $\boldsymbol{l}_k = m_k \boldsymbol{r}_k$ the first moment of inertia, and L_{ij}^k, $\{i, j\} \in \{x, y\}$, the inertia matrix entries.

A complete description of the dynamic model requires to consider also motor friction, the elastic torque contribution generated by power cables (e.g., for J_1 and J_2) and by the presence of the torsional spring at the joint J_4. The associated dynamic parameters (i.e., motor inertia, viscous and Coulomb friction constants, Coulomb friction offset, stiffness coefficient) are identified in [1] and gathered for each link in $\boldsymbol{\delta}_{A,k} \in \mathbb{R}^5$. Clearly, not all the components of $\boldsymbol{\delta}_{A,k}$ are meaningful for each link ℓ_k due to the presence of passive joints; hence, when appropriate, they have not been considered in the identification procedure.

The vector $\boldsymbol{\delta}$ in (1) is built by stacking the dynamic parameters $\boldsymbol{\delta}_{L,k}$ and $\boldsymbol{\delta}_{A,k}$ for all the N links.

The identification problem consists in estimating a suitable value $\boldsymbol{\delta}^*$ of the dynamic parameters collected in $\boldsymbol{\delta}$, minimizing the error between the torques predicted by (1) and their measured values, in a least-square fashion. To enforce the



(a) (b) (c)

Fig. 1: (a) One of the two PSM arms of the da Vinci Research Kit. The location of the RCM point is also shown in the blue circle. (b) The double parallelogram mechanism highlighted with red and blue squares. Joint axes are highlighted with dashed black lines, with passive joints denoted as J_{p*}. (c) The counterweight modelling solution: the force measured at $J_{cw,l}$ is transmitted to $J_{cw,t}$, that exerts the same force on the tool through a dedicated loop closure configuration (LCC).

physical feasibility of the parameters to be identified, a set of inequality constraints is included in [1], and the identification is formalized as a semidefinite programming problem. The solution $\boldsymbol{\delta}^*$ provides a set of parameters that, consistently with the dynamic behaviour of the real robot, describe mass, inertia and center of mass of all the links of the PSM.

For control design purposes, it is worth mentioning that the dynamic model (1) has been also reformulated in [1] according to the Euler-Lagrange equations:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau}_f = \boldsymbol{\tau} \tag{2}$$

where $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is the matrix of Coriolis and centrifugal forces, $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is the gravity vector and $\boldsymbol{\tau}_f \in \mathbb{R}^n$ the friction torques. Retrieving a set of feasible parameters $\boldsymbol{\delta}^*$ provides an estimation of terms in (2), namely $\widehat{\boldsymbol{M}}(\boldsymbol{q})$, $\widehat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})$, $\widehat{\boldsymbol{g}}(\boldsymbol{q})$ and $\widehat{\boldsymbol{\tau}}_f$. This allows to implement torque control laws, as will be shown in Section IV.

III. THE DVRK-PSM DYNAMIC SIMULATOR

In minimally invasive surgery executed through robotic manipulators the link constrained to move through the incision point can only translate along its axis and rotate about a point on the link axis known as Remote Center of Motion (RCM), which should be maintained in the same position as the incision point during the surgery.

While for a serial nDoF manipulator, the RCM constraint can be enforced by proper kinematic modeling and control of the constraint and the surgical task [16], the PSM of the dVRK satisfy this constraint mechanically, through a double parallelogram mechanism equipped with passive joints (denoted by J_{p**} in Fig. 1b).

As highlighted in [14]- [13], the derivation of the robot kinematics does not necessarily require that the model of the double parallelogram reflect the actuation of the physical structure. The solution adopted in [14] to satisfy the RCM constraint was, in fact, to add two (ficticious) revolute joints at the RCM and to constraint consistently the motion of the passive joints of the parallelogram. Moreover, in the robot kinematics description the presence of the counterweight, balancing the motion of the surgical tool along its longitudinal axis, is not required.

In a dynamic simulator, however, both the kinematics and the actuation model of the double parallelogram and of the counterweight mechanisms must reflect the physical structure for proper simulation of the actuation effort.

Therefore, in this work, we provide: (i) a reformulation of the hierarchy of joints and shapes composing the double parallelogram structure which is coherent with the robot kinematics in [1] and includes the explicit formulation of the closure constraint; (ii) a simulation model of the counterweight system. This allows to integrate the dynamic parameters of the robot links identified in [1], so as to obtain a physics-based simulation.

For the sake of clarity, we first summarize some CoppeliaSim features, exploited to model the virtual robot[1].

Shapes: rigid meshes built with triangular faces, used to model and visualize the robot links in the virtual environment. More in detail, each link is characterized by a pair of shapes: i) a *respondable* shape, simulating contacts with other shapes in the environment; ii) a *visual* shape, that is graphically more accurate and reproduces a geometric structure analogous to the real link. In addition, *respondable* meshes can be set as *dynamic*, in order to simulate also the dynamics of the corresponding rigid body.

Joints: CoppeliaSim objects enabling motion of the attached links. Joint operation modes can be *passive*, *dependent*, or *torque/force*. When working in *torque/force mode*, an object

[1]A more detailed and exhaustive description of the CoppeliaSim feature can be found at the official documentation webpage: https://www.coppeliarobotics.com/helpFiles/en/designingDynamicSimulations.htm

joint behaves as active and, when connected to *dynamic respondable shapes*, it is said *dynamically enabled*. Due to their moltiple operation modes, joints can be involved in the design of *loop closure configurations*, necessary to model closed chains.

Dummy object: simple point shape with an attached reference frame, without mass nor other dynamic properties, used to represent a reference frame.

Overlap constraint: dynamic constraint imposed on a couple of dummy objects and forcing them to adopt the same position and orientation.

Loop closure configuration (LCC): a special hierarchical configuration of virtual objects, combining *dynamically enabled joints* and *overlap constraints* to connect respondable shapes that are not in a parent-child relationship. Loop closure configurations are adopted to model closed chains, either kinematic or dynamic.

A. Modelling of the double parallelogram chain

The double parallelogram structure of the dVRK-PSM is highlighted in Fig. 1b. The shapes of the links and joints are modeled as follows: the actuated revolute joints J_1 and J_2 describe the roll and pitch rotation of the structure that is anchored at the link ℓ_1. Differently from [14]– [13], their position in the kinematic hierarchy reflects here the structure of the physical robot. The first parallelogram, highlighted with a red square in the Fig. 1b, is composed by the top part of the link ℓ_2 (red part labeled as $\ell_{2,1}$ in the encircled close-up view), along with links ℓ_3, ℓ_4 and ℓ_6, and it is connected through the passive joints J_{p22}, J_{p23}, J_{p24} and J_{p25}. Similarly, the second parallelogram of the kinematic chain, highlighted with a blue square in the same figure, is composed by the bottom part of the link ℓ_2 (blue part labeled as $\ell_{2,2}$ in the encircled close-up view), along with links ℓ_1, ℓ_4 and ℓ_5, and it is connected by the joints J_2, J_{p21}, J_{p23} and J_{p26}. To enforce the two closed kinematic chains, two different *LCC*s are set: the loop LCC_1 for the first parallelogram is enforced through the joint J_{p24} and specifies an *overlap constraint* between dummy objects attached to the links ℓ_3 and ℓ_6. The second loop LCC_2 is enforced through the joint J_{p21} and specifies an *overlap constraint* between dummy objects attached to the links ℓ_1 and ℓ_5.

B. Modelling of the counterweight

On the identified dynamics of the dVRK PSM arm, the counterweight is linked to the prismatic joint J_3 through a transmission system, based on pulleys and tendons. The identification procedure in [1] models this mechanism, providing an estimation of the relative dynamic parameters that we have integrated in the simulation model as explained below.

Since deformable objects, like tendons, are not simulated in CoppeliaSim, the transmission system is modeled by simulating the effect at the two endpoints of the transmission chain. On the *load-side*, the actuated prismatic joint $J_{cw,l}$ makes the counterweight mirror the motion of the tool. In particular, setting the current position q_3 of the joint J_3 as a reference for $J_{cw,l}$, i.e., $q_{r_{cw,l}} = q_3$, a low-level PID controller regulates the position of $J_{cw,l}$.

On the *tool-side*, the measure of the force provided at $J_{cw,l}$ for regulating its position to $q_{r_{cw,l}}$ contributes to the actuation of the tool joint. Specifically, a fictitious prismatic joint $J_{cw,t}$ is placed above the prismatic joint J_3 so as the joint axes are coincident. To simulate the compensation mechanism, the force f_{cw} provided by $J_{cw,l}$ to move the counterweight is set as the force commanded at $J_{cw,t}$, while an additional *LCC* between $J_{cw,t}$ and J_3 is enforced. The effect of the *LCC* is to partially compensate the force due to gravity on the dynamics of J_3 with the force commanded at $J_{cw,t}$.

Note in Fig. 1c: i) $J_{cw,l}$ providing the necessary actuation to move the counterweight along a constrained direction, consistently with the motion of the tool; ii) the *respondable* counterweight load shape, characterized by its mass and inertia from [1]; iii) the fictitious prismatic joint $J_{cw,t}$.

C. Integration of the dynamic coefficient in the robot model

The full set of identified parameters $\boldsymbol{\delta}$ allows to include in the simulation the dynamic effect of the links motion according to their mass, inertia and center of mass.

On the other side, joint friction, motor inertia and springs are not simulated by the physics engine of CoppeliaSim. Therefore, only the inertial parameters $\boldsymbol{\delta}_{L,k}$ are set in the simulator for each link. The effect of the additional parameters $\boldsymbol{\delta}_{A,k}$ can be optionally taken into account as an additive torque contribution determined on the base of a mathematical model of the involved phenomena, evaluated on the current state of the system. In the proposed simulations we have not introduced this model-based contribution to report on the physics simulation only.

The assignment of mass, inertia tensor and CoM coordinates of the corresponding rigid link ℓ_i requires preliminary mathematical manipulations. In fact, while in CoppeliaSim inertia tensors and CoM coordinates are expressed in the virtual world frame $\mathcal{F}_w$, the corresponding quantities in $\boldsymbol{\delta}_{L,i}$ are expressed in the related link frame $\mathcal{F}_i$. Denoting by $< {}^w\boldsymbol{R}_0,\ {}^w\boldsymbol{p}_0 >$ the constant roto-translation between the CoppeliaSim world frame $\mathcal{F}_w$ and the robot base frame $\mathcal{F}_0$, we simply convert the CoM coordinates as

$$ {}^w\boldsymbol{r}_i = {}^w\boldsymbol{R}_i\ {}^i\boldsymbol{r}_i + {}^w\boldsymbol{p}_0, \tag{3}$$

where ${}^w\boldsymbol{R}_i = {}^w\boldsymbol{R}_0\ {}^0\boldsymbol{R}_i$ is the composed rotation expressing the orientation of the i-th link frame $\mathcal{F}_i$ in $\mathcal{F}_w$.

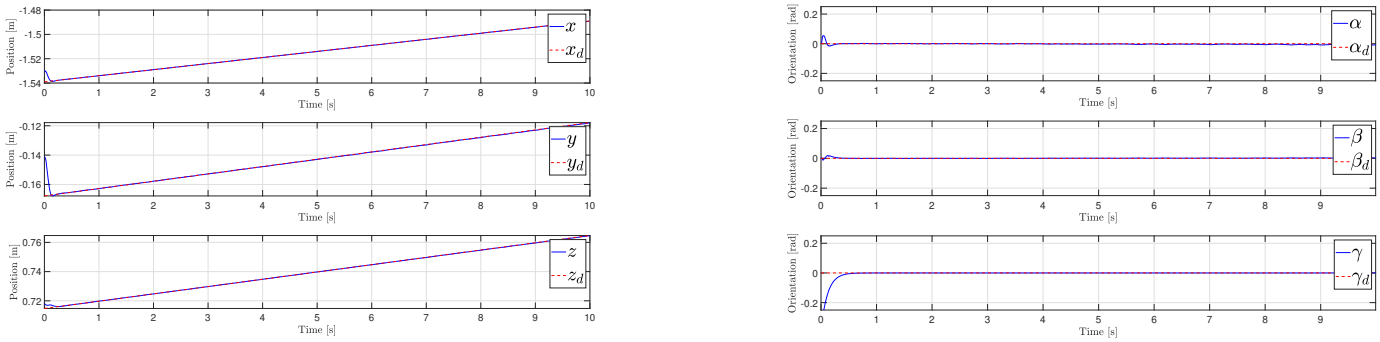
Similarly, in order to convert the elements of $\boldsymbol{\delta}_{L,i}$ related to the inertia tensor expressed in $\mathcal{F}_i$, we apply the parallel axis theorem [17] on the inertia tensor $\boldsymbol{L}_i$ so as to obtain

$$ {}^w\boldsymbol{L}_i = {}^w\boldsymbol{R}_i\boldsymbol{L}_i{}^w\boldsymbol{R}_i^T + m_i\boldsymbol{S}^T\left({}^w\boldsymbol{r}_i\right)\boldsymbol{S}\left({}^w\boldsymbol{r}_i\right), \tag{4}$$

where $\boldsymbol{S}(*)$ is the skew-symmetric operator. Evaluating (3) and (4) for all the rigid links ℓ_i generates the correct data to be integrated in the PSM simulation model.

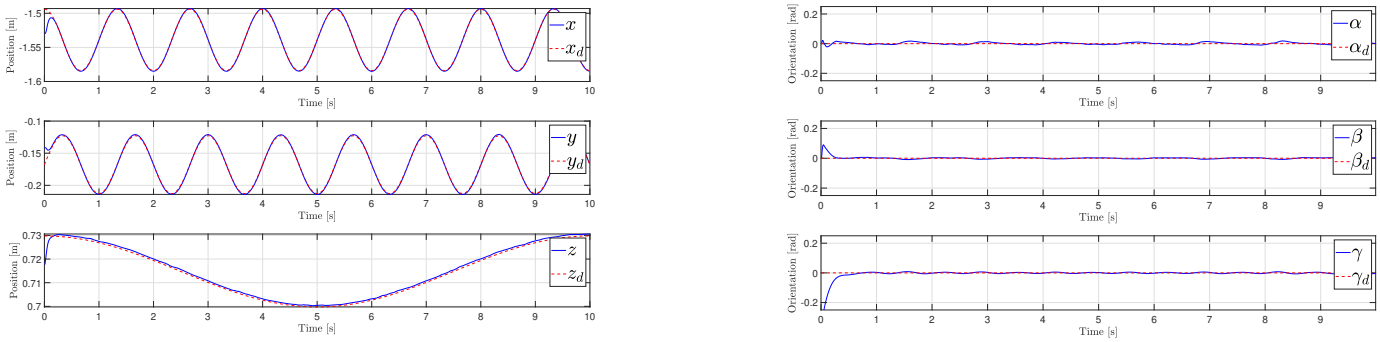
IV. SIMULATOR VALIDATION

The validation of the proposed dynamic dVRK simulator in CoppeliaSim has been accomplished through different simulation tests. First, a validation of the redesigned robot kinematics, including the double parallelogram model, through



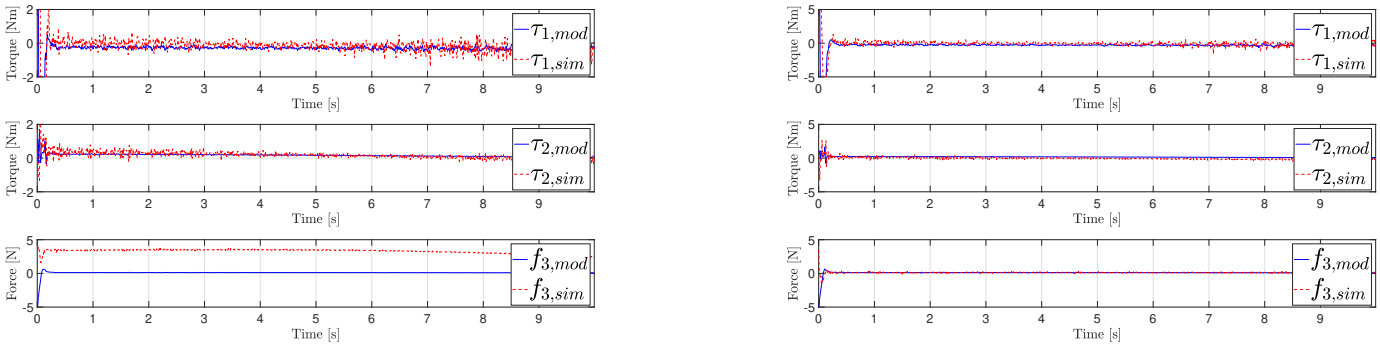
(a) (b)

Fig. 2: Validation of the kinematic model: Reference (dashed red) and actual (blue) Cartesian position (a) and orientation (b) during a rectilinear trajectory tracking task.



(a) (b)

Fig. 3: Validation of the kinematic model: Reference (dashed red) and actual (blue) Cartesian position (a) and orientation (b) during a spiral trajectory tracking task.



(a) (b)

Fig. 4: Dynamic model validation through kinematic control: comparison of model-based torques (blue) vs simulated torques (red), during a tracking task along a rectilinear trajectory. Left (a): the counterweight is not modeled and a significant force error is observed on J_3. Right (b): the counterweight is accounted in the robot modelling, resulting in a small force error.

the execution of Cartesian trajectory tracking tasks with a kinematic controller. In this first case, the CoppeliaSim dynamic engine is disabled and the tracking error is taken as a performance indicator. Next, the same conditions are used to assess also the correctness of the dynamic model, with the CoppeliaSim dynamic engine turned on. In this case, validation is accomplished by comparing the torques predicted by the dynamic model, identified in [1], with the torques simulated by the physics engine of CoppeliaSim. To further evaluate the performance of the simulator, its torque prediction capability has been compared with data collected on the real robot during the identification procedure in [1]. Finally, for demonstration purposes, we show the results of Cartesian trajectory tracking tasks which are accomplished through torque control. Control design is based on the robot dynamic model (2), using the identified matrices.

Given a Cartesian reference trajectory for the tool $\boldsymbol{r}_d$, with time derivative $\dot{\boldsymbol{r}}_d$, the kinematic controller provides the joint reference velocity:

$$\dot{\boldsymbol{q}}_d(t) = \boldsymbol{J}(\boldsymbol{q})^{-1} \left(\boldsymbol{K}_p \boldsymbol{e}(t) + \dot{\boldsymbol{r}}_d(t) \right), \tag{5}$$

where $\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{f}(\boldsymbol{q})}{\partial \boldsymbol{q}}$ is the robot Jacobian, $\boldsymbol{f}(\boldsymbol{q}) = \boldsymbol{r}(t)$ the

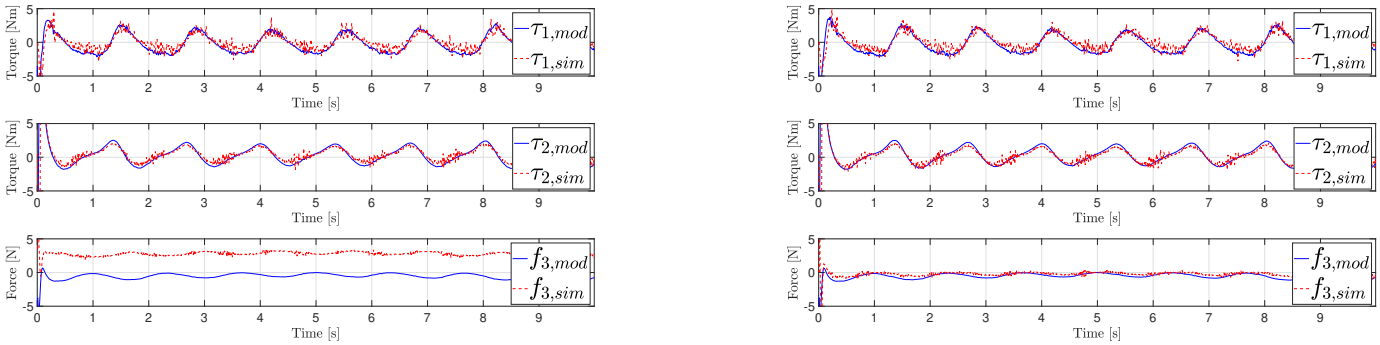


Fig. 5: Dynamic model validation through kinematic control: comparison of model-based torques (blue) vs simulated torques (red), during a tracking task along a spiral trajectory. Left (a): the counterweight is not modeled and a significant force error is observed on J_3. Right (b): the counterweight is accounted in the robot modelling, resulting in a small force error.

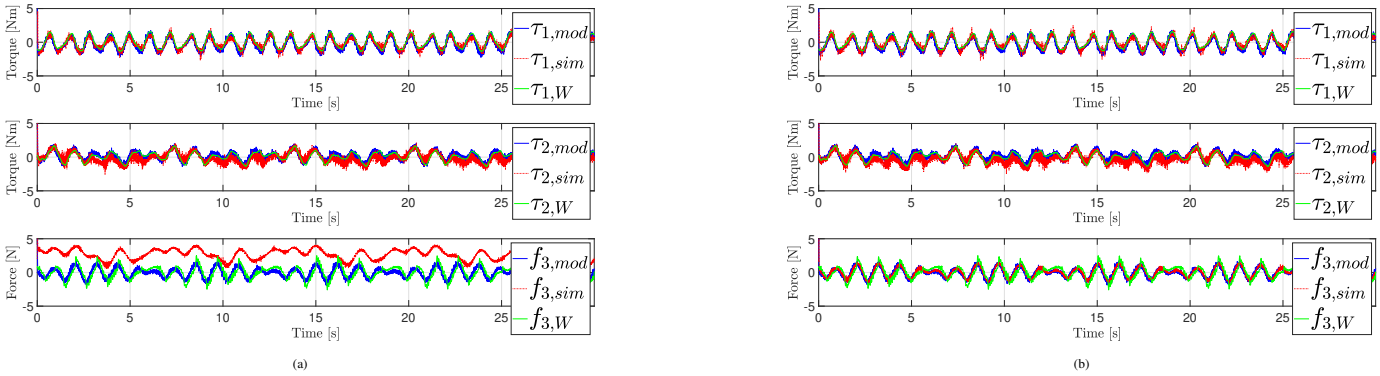


Fig. 6: Dynamic model validation through kinematic control along the optimal excitation joint trajectories used in [1]: comparison of the model-based torques (blue) vs simulated torques (red) vs measured torques (green) retrieved from online database[2]. Left (a): the counterweight is not modeled and a significant force error is observed on J_3. Right (b): the counterweight is accounted in the robot modelling, resulting in a small force error.

direct kinematics function, built consistently with [1], $\boldsymbol{e} = \boldsymbol{r}_d - \boldsymbol{r}$ the tracking error, $\boldsymbol{K}_p > 0$ a gain matrix.

In the first, kinematics only, validation, the joints of the simulated robot are set to the desired value $\boldsymbol{q}_d$ obtained from $\dot{\boldsymbol{q}}_d$ through simple integration. In the second validation scenario, when the dynamic engine of the CoppeliaSim is turned on, the joints reference velocity $\dot{\boldsymbol{q}}_d$ computed through (5) is sent to the low-level controllers simulated by CoppeliaSim. The corresponding control torques/forces are therefore generated by the physics simulation. In particular, by setting a very high value for the maximum torque/force of the joints, the target velocities are instantaneously reached, that is equivalent to let the joints operate in velocity control.

Section IV-A and Sect. IV-B respectively, show the validation of the robot simulator in these two operative conditions. The results relative to torque control are given in Sect. IV-C.

A. Validation of the novel kinematic model

In the first simulation, we consider a tracking problem for the gripper of the PSM along a rectilinear trajectory, i.e., $\boldsymbol{r}_d(t) = L\frac{t}{T}$ and $\dot{\boldsymbol{r}}_d(t) = \frac{L}{T}$, where $L = 0.09$ m is the length of the path and $T = 10$ s is the path traveling interval. The desired orientation of the gripper is set constant along the path. The tracking results, shown in Fig. 2 and comparing the pose of the PSM gripper over time with the desired quantities, prove good tracking performance in terms of Root Mean Square (RMS) along the trajectory ($\text{RMS}_p = (7 \cdot 10^{-4}, 3 \cdot 10^{-3}, 4 \cdot 10^{-4})$ [m] in position, $\text{RMS}_o = (4 \cdot 10^{-3}, 2 \cdot 10^{-2}, 3 \cdot 10^{-2})$ [rad] in orientation).

The second simulation consists in tracking of a spiral trajectory with the PSM gripper. In this case $\boldsymbol{r}_d(t) = (A\cos(2\pi f_A t), A\sin(2\pi f_A t), h\cos(2\pi f_h t))^T$, where $A = 0.05$ m is the radius of the spiral, $h = 0.03$ m is its height, $f_A = 1.5$ Hz and $f_h = 0.5$ Hz are the trajectory frequencies. Results of this simulation are reported in Fig. 3. Also in this simulation, the high accuracy of the tracking proves the consistency of the re-modeled kinematic chain and of the double parallelogram explicit simulation, with $\text{RMS}_p = (2 \cdot 10^{-3}, 1 \cdot 10^{-3}, 9 \cdot 10^{-4})$ [m] for the position, $\text{RMS}_o = (3 \cdot 10^{-3}, 2 \cdot 10^{-3}, 3 \cdot 10^{-2})$ [rad] for the orientation.

B. Validation of the dynamic model

For this validation, the Open Dynamics Engine available in CoppeliaSim is enabled, with a simulation time step $\Delta T = 10$ ms and a dynamic calculation step of 5 ms. The validation of the dynamic properties of the proposed dVRK simulator is

accomplished at the torque-level. Specifically, the joint torques $\boldsymbol{\tau}_{sim}$, simulated through the CoppeliaSim physics engine and measured by the dedicated software API functions, are compared with torques $\boldsymbol{\tau}_{mod} = \boldsymbol{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$, predicted by the model identified in [1]. The computational load to retrieve the identified dynamic model has been verified to be negligible, as the response time of the queries to read the inertia matrix and the gravity and Coriolis/centrifugal vectors is of the order of microseconds (6-7 μs). In the kinematic control, we set $\boldsymbol{K}_p = 15$. As before, two control tasks are considered to be accomplished through kinematic control: tracking of a rectilinear and of a spiral Cartesian trajectory specified for the tip of the PSM gripper. The difference between measured and predicted torques is evaluated in terms of RMS error. To better highlight the contribution of the counterweight, we show the comparison in two different cases: in the first case (nCW), the counterweight is removed from the simulated model of the PSM robot; in the second case (wCW), the counterweight is accounted.

The results relative to first task are shown in Fig. 4 that reports the comparison between the simulated torques $\boldsymbol{\tau}_{sim}$ (red) and the model-based predicted torques $\boldsymbol{\tau}_{mod}$ (blue): when the counterweight is not modeled, we observe a relevant force error on J_3, with $\text{RMS}_{\text{nCW}} = (0.66$ [Nm], 0.13 [Nm], 2.17 [N]). On the contrary, when the counterweight is accounted, the error on J_3 is dramatically reduced, while the effect on J_1 and J_2 is minor, having $\text{RMS}_{\text{wCW}} = (0.65$ [Nm], 0.1 [Nm], 0.048 [N]).

The second simulation, shown in Fig. 5, highlights a similar behaviour: despite the higher excitation of the trajectory, simulated torques are consistent with the model-based prediction, with a large error on J_3 when the counterweight is not modeled ($\text{RMS}_{\text{nCW}} = (0.59$ [Nm], 0.38 [Nm], 3.3 [N]), see Fig. 5a), compared to the scenario with the counterweight ($\text{RMS}_{\text{wCW}} = (0.57$ [Nm], 0.27 [Nm], 0.3 [N]), see Fig. 5b).

To conclude this part, we show also the validation results along the optimal excitation joint trajectories that have been used in [1] in the experimental sessions on the real robot. In this case, the joint velocities measured along the trajectories retrieved from the identification database[2] , are set as joint velocity reference $\dot{\boldsymbol{q}}_d$ for the simulator. Results, shown in Figure 6, are consistent with previous simulations, with lower RMS errors observed when the counterweight is accounted ($\text{RMS}_{\text{nCW}} = (0.58$ [Nm], 0.8 [Nm], 2.89 [N]), $\text{RMS}_{\text{wCW}} = (0.55$ [Nm], 0.8 [Nm], 0.59 [N])).

C. Torque-based control

In this validation, since only the first three joints of the PSM are dynamically enabled, only position tracking tasks are considered. For all tasks, specified by a reference position trajectory $\boldsymbol{r}_d(t)$, a control law with feedback linearization in the Cartesian space generates the commanded torques $\boldsymbol{u}$:

$$\boldsymbol{u} = \widehat{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{J}_L^{-1}\boldsymbol{a} + \widehat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \widehat{\boldsymbol{g}}(\boldsymbol{q}) - \widehat{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{J}_L^{-1}\dot{\boldsymbol{J}}_3\dot{\boldsymbol{q}} \quad (6)$$

where $\widehat{*}$ denote estimated quantities of the dynamic model (2) and $\boldsymbol{J}_L$ is the 3×3 Jacobian matrix relating the Cartesian linear velocity of the PSM end-effector to the motion of the first three joints. The Cartesian acceleration $\boldsymbol{a}$ is designed to stabilize the position error $\boldsymbol{e}_p = \boldsymbol{r}_d - \boldsymbol{r}$ through decoupled chains of double integrators:

$$\boldsymbol{a} = \ddot{\boldsymbol{r}}_d + \boldsymbol{K}_p\boldsymbol{e}_p + \boldsymbol{K}_d\dot{\boldsymbol{e}}_p \quad (7)$$

with $\boldsymbol{K}_p$, $\boldsymbol{K}_d > 0$. Results of the tracking tasks with rectilinear and spiral trajectories are shown in Fig. 7 and Fig. 8, respectively, with $\boldsymbol{K}_p = 8{\cdot}10^2\boldsymbol{I}_3$ and $\boldsymbol{K}_d = 6{\cdot}10\boldsymbol{I}_3$. In all the presented scenarios, since the torque-controlled PSM robot is non-redundant for the considered tasks, the dynamic behavior at the Cartesian level is exactly linearized and decoupled, showing an exponentially stable transient behavior of the Cartesian error along the desired trajectory.

It is worth noticing that, for all the presented cases, a discrepancy is present, in the transient phase, between the simulated and the model-based predicted force at joint J_3. Such discrepancy is due to the way in which the transmission mechanism of the counterweight has been modeled, as described in Sect. III-B. In particular, the low-level PID controller of joint $J_{cw,l}$, that mirrors the motion of J_3, modifies the dynamic response of the counterweight with respect to the identification model.

A videoclip showing the presented simulations is included in the multimedia material of this Letter. The software of the dynamic dVRK simulator, composed by the CoppeliaSim scenes, the dynamic model of the PSM and the example program replicating the presented simulations, is available at: https://github.com/marcofer/dynamic-dvrk-coppeliasim-simulator. We highlight that, similarly to the *portable* version presented in [12], the provided simulator has not been developed through the ROS interface. However, it could be easily adapted, in an analogously to the former version described in [11].

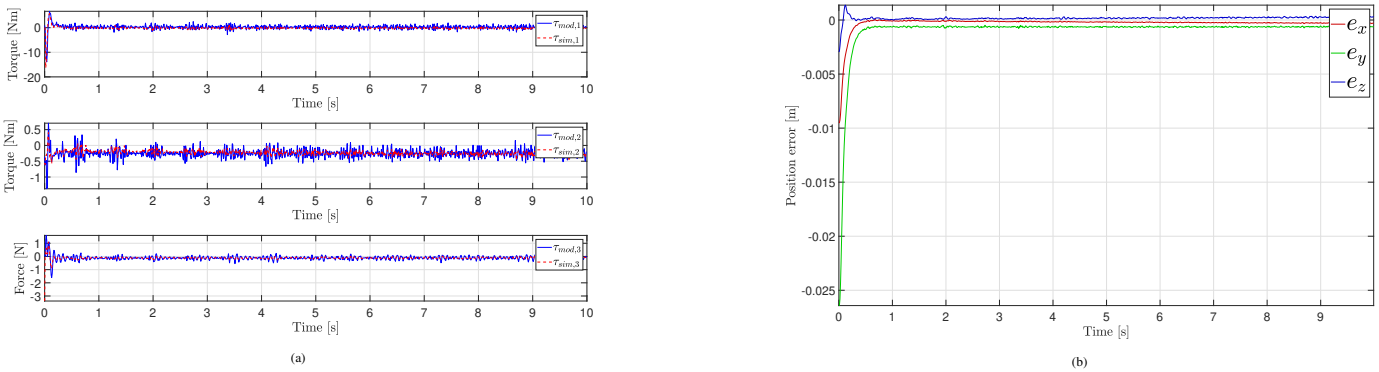
V. CONCLUSIONS

In this paper, we presented a dynamic simulator of the da Vinci Research Kit Patient-Side Manipulator, in the CoppeliaSim robotic simulation environment. The presented simulator relies on a refined version of a previously developed *kinematic* simulator in the V-REP/CoppeliaSim environment that reflects the mechanical structure of the real robot and its actuation. It is the first dynamic simulator integrating the dynamic parameters of the real system identified in [1].

The development of the simulator has required: i) the redefinition of the robot kinematic structure to model the double parallelogram mechanism and its actuation and by explicitly simulating the presence of a counterweight and the associated transmission system; ii) the proper integration of the set of dynamic parameters identified.

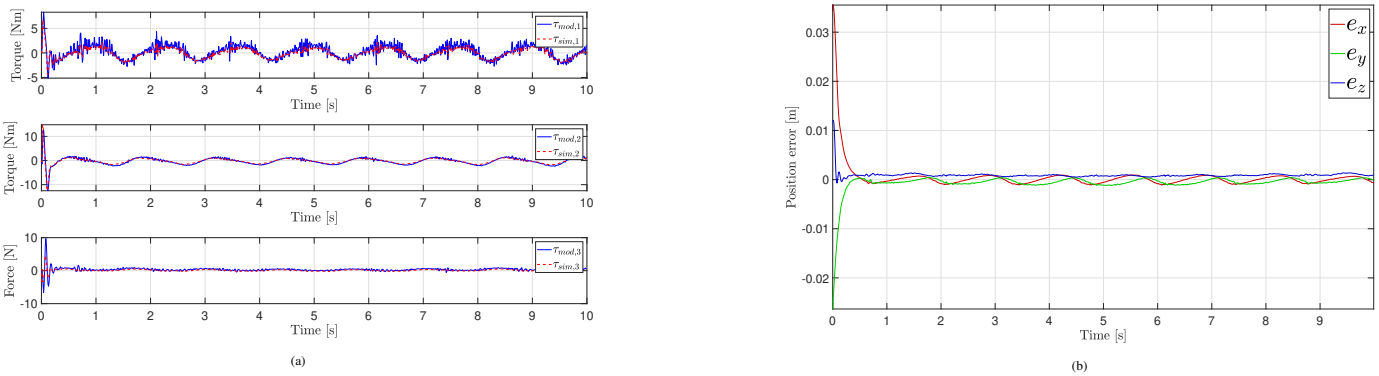
The resulting simulator has been validated through a set of control tasks, by evaluating the trajectory tracking error and the error between simulated, model-predicted, and measured joint torques. The torque comparison shows that the design choices capture consistently the dynamic behaviour described by the identified dynamic model and the behaviour of the real robot.

[2]https://github.com/WPI-AIM/dvrk_dynamics_identification



(a) (b)

Fig. 7: Dynamic model validation through torque control: commanded input torques generated by the control (a) and resulting Cartesian position error (b) during a tracking task along a linear trajectory.



(a) (b)

Fig. 8: Dynamic model validation through torque control: commanded input torques generated by the control (a) and resulting Cartesian position error (b) during a tracking task along a spiral trajectory.

REFERENCES

[1] Y. Wang, R. Gondokaryono, A. Munawar, and G. S. Fischer, "A convex optimization-based dynamic model identification package for the da vinci research kit," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3657–3664, 2019.

[2] Intuitive surgical. [Online]. Available: https://www.intuitive.com

[3] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da vinci® surgical system," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6434–6439.

[4] C. D'Ettorre, A. Mariani, A. Stilli, F. Rodriguez y Baena, P. Valdastri, A. Deguet, P. Kazanzides, R. H. Taylor, G. S. Fischer, S. P. DiMaio, A. Menciassi, and D. Stoyanov, "Accelerating surgical robotics research: A review of 10 years with the da vinci research kit," *IEEE Robotics Automation Magazine*, vol. 28, no. 4, pp. 56–78, 2021.

[5] A. Moglia, V. Ferrari, L. Morelli, M. Ferrari, F. Mosca, and A. Cuschieri, "A systematic review of virtual reality simulators for robot-assisted surgery," *European urology*, vol. 69, no. 6, pp. 1065–1080, 2016.

[6] F. Bovo, G. De Rossi, and F. Visentin, "Surgical robot simulation with bbz console," *Journal of Visualized Surgery*, vol. 3, 2017.

[7] E. Tagliabue, A. Pore, D. Dall'Alba, E. Magnabosco, M. Piccinelli, and P. Fiorini, "Soft tissue simulation environment to learn manipulation tasks in autonomous robotic surgery," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 3261–3266.

[8] N. Enayati, A. M. Okamura, A. Mariani, E. Pellegrini, M. M. Coad, G. Ferrigno, and E. De Momi, "Robotic assistance-as-needed for enhanced visuomotor learning in surgical robotics training: An experimental study," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6631–6636.

[9] A. Munawar, Y. Wang, R. Gondokaryono, and G. S. Fischer, "A real-time dynamic simulator and an associated front-end representation format for simulating complex robots and environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019, pp. 1875–1882.

[10] A. A. Gondokaryono RA, "An approach to modeling closed-loop kinematic chain mechanisms, applied to simulations of the da vinci surgical system," *Acta Polytechnica Hungarica*, vol. 16, no. 8. [Online]. Available: https://par.nsf.gov/biblio/10113819

[11] G. A. Fontanelli, M. Selvaggio, M. Ferro, F. Ficuciello, M. Vendittelli, and B. Siciliano, "A v-rep simulator for the da vinci research kit robotic platform," in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, 2018, pp. 1056–1061.

[12] M. Ferro, D. Brunori, F. Magistri, L. Saiella, M. Selvaggio, and G. A. Fontanelli, "A portable da vinci simulator in virtual reality," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, 2019, pp. 447–448.

[13] G. A. Fontanelli, M. Selvaggio, M. Ferro, F. Ficuciello, M. Vendittelli, and B. Siciliano, "Portable dvrk: an augmented v-rep simulator of the da vinci research kit," *Acta Polytechnica Hungarica*, vol. 16, no. 8, pp. 79–98, 2019.

[14] G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano, "Modelling and identification of the da vinci research kit robotic arms," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1464–1469.

[15] P. Maes, J.-C. Samin, and P.-Y. Willems, "Linearity of multibody systems with respect to barycentric parameters: Dynamics and identification models obtained by symbolic generation," *Mechanics of Structures and Machines*, vol. 17, no. 2, pp. 219–237, 1989.

[16] N. Aghakhani, M. Geravand, N. Shahriari, M. Vendittelli, and G. Oriolo, "Task control with remote center of motion constraint for minimally invasive robotic surgery," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 5807–5812.

[17] A. Abdulghany, "Generalization of parallel axis theorem for rotational inertia," *American Journal of Physics*, vol. 85, no. 10, pp. 791–795, 2017.