A Recursive Lie-Group Formulation for the Second-Order Time Derivatives of the Inverse Dynamics of Parallel Kinematic Manipulators

Andreas Müller[1], Shivesh Kumar[2], Thomas Kordik[1]

***Abstract*—Series elastic actuators (SEA) were introduced for serial robotic arms. Their model-based trajectory tracking control requires the second time derivatives of the inverse dynamics solution, for which algorithms were proposed. Trajectory control of parallel kinematics manipulators (PKM) equipped with SEAs has not yet been pursued. Key element for this is the computationally efficient evaluation of the second time derivative of the inverse dynamics solution. This has not been presented in the literature, and is addressed in the present paper for the first time. The special topology of PKM is exploited reusing the recursive algorithms for evaluating the inverse dynamics of serial robots. A Lie group formulation is used and all relations are derived within this framework. Numerical results are presented for a 6-DOF Gough-Stewart platform (as part of an exoskeleton), and for a planar PKM when a flatness-based control scheme is applied.**

***Index Terms*—Parallel kinematic manipulator (PKM), inverse dynamics, series-elastic actuators (SEA), feedback linearization, flatness-based control, O(n)-algorithm, Lie group**

I. Introduction

Series elastic actuators (SEA) were introduced as actuation concept for serial kinematic robotic arms [1] as a means to provide inherent compliance (a key characteristics of collaborative robots (cobots)). Lightweight arms were proposed in order to reduce the moving mass. Along this line, parallel kinematics manipulators (PKM) equipped with SEA possess lower moving mass, and thus lower reflected inertia at the end-effector (EE), compared to serial robots, which serves as criteria for safety assessment [2], [3], [4]. They would hence be perfectly suited as inherently compliant agile robotic manipulators and cobots. Yet, SEA-actuated PKM (SEA-PKM) have exclusively been used as force-controlled support devices [5], [6], [7]. The dynamics of SEA-PKM is expressed as

$$\mathbf{M}_\mathrm{t}\dot{\mathbf{V}}_\mathrm{t} + \mathbf{C}_\mathrm{t}\mathbf{V}_\mathrm{t} + \mathbf{W}_\mathrm{t}^\mathrm{grav} = \mathbf{J}_\mathrm{IK}^T\mathbf{u} \tag{1}$$

$$\mathbf{M}_\mathrm{m}\ddot{\mathbf{q}}_\mathrm{m} = \boldsymbol{\tau} - \mathbf{u} \tag{2}$$

where the equations of motion (EOM) (1) govern the PKM dynamics in task space (see Sec. III), and (2) the actuator dynamics, with $\mathbf{M}_\mathrm{m} = \mathrm{diag}(m_1, \dots m_{n_\mathrm{a}})$ defined by the reduced inertia moment m_i of the ith drive unit, and the vector $\boldsymbol{\tau}$ of actuator torques/forces. Both are coupled via the elastic forces $\mathbf{u} = \mathbf{K}(\mathbf{q}_\mathrm{m} - \boldsymbol{\vartheta}_\mathrm{a})$, where $\mathbf{K} = \mathrm{diag}(k_1, \dots k_{n_\mathrm{a}})$ describes the SEA compliance, with stiffness coefficient k_i associated to drive i. Here $\boldsymbol{\vartheta}_\mathrm{a}$ denotes the coordinate vector with the n_a actuated joints, $\mathbf{q}_\mathrm{m}$ the vector of n_a motor coordinates, and $\mathbf{V}_\mathrm{t}$ the task space velocity of the platform. The PKM mechanism is actuated by the elastic forces $\mathbf{u}$, which are transformed to task space with the inverse kinematics Jacobian $\mathbf{J}_\mathrm{IK}$ (i.e. $\dot{\boldsymbol{\vartheta}}_\mathrm{a} = \mathbf{J}_\mathrm{IK}\mathbf{V}_\mathrm{t}$).

Problem: Position and trajectory tracking control of SEA-driven robots necessitate model-based feedforward control. Flatness-based exact feedback linearization control methods are well established for SEA-actuated serial robotic arms [8], [9], [10]. Since the model (1),(2) is formally identical to that of SEA-actuated serial robots (except the IK Jacobian), they can be directly adopted to SEA-PKM. The EE (task space) motion is used as flat output, and it is known that the vector relative degree of this control system with input $\boldsymbol{\tau}$ is $\{4, \dots, 4\}$, and $\{3, \dots, 3\}$ if damping is included in (2). That is, the PKM state and the input $\boldsymbol{\tau}$ can be expressed in terms of EE pose, task space velocity $\mathbf{V}_\mathrm{t}$, and its time derivatives $\dot{\mathbf{V}}_\mathrm{t}, \ddot{\mathbf{V}}_\mathrm{t}, \dddot{\mathbf{V}}_\mathrm{t}$. To this end, (1) is solved for $\mathbf{q}_\mathrm{m}$. Substituting this solution, and its second time derivative in (2) yields $\boldsymbol{\tau}$, which serves as feed-forward control. The crucial aspect of the flatness-based control is that it *involves the first and second time derivative of the EOM* (1). For serial robotic arms, recursive second-order inverse dynamics $O(n)$ algorithms were proposed [11], [12] using classical vector formulations of rigid body kinematics. Recursive Lie group formulations were proposed in [13], [14] employing compact expressions for rigid body twists complementing the inverse dynamics algorithms in [15], [16], which can be seen as generalization of the spatial vector algebra [17], [18]. Also a closed form Lie group formulation was reported in [19]. All these approaches for serial kinematics robots are direct extensions of recursive inverse dynamics formulations. In contrast, the derivatives of the inverse dynamics solution for PKM is more involved due to the presence of loop constraints. The latter can be resolved and incorporated in the EOM in various different ways, and the modeling approach dictates the complexity of the higher-order inverse dynamics algorithm, which is thus crucial to for development of SEA-PKM into robotic manipulators. The best suited modeling method is the one proposed for non-redundant fully parallel PKM with simple limbs reported in [20], [21] and [22]. In this method, each limb is regarded as a serial chain, and their motion is expressed in terms of the platform motion by means of the inverse kinematics solution of each limb. For each limb, this

Manuscript received: November 30, 2022; Revised March 2, 2023; Accepted March 28, 2023. This paper was recommended for publication by Editor Lucia Pallottino upon evaluation of the Associate Editor and Reviewers' comments. Support by LCM K2 Center for Symbiotic Mechatronics within the Austrian COMET-K2 program is acknowledged.

[1]Institute of Robotics, Johannes Kepler University, 4040 Linz, Austria; [2]Robotics Innovation Center, DFKI GmbH, 28359 Bremen, Germany

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

resembles a task space formulation of serial robots. This was developed into a *task space* formulation for general redundant PKM with simple limbs in [23] and with complex limbs in [24], [25] using a Lie group framework (although the basic concept does not depend on it). The important implication is that the higher-order inverse dynamics problem boils down to merging the higher-order inverse dynamics of the individual limbs with the higher-order inverse kinematics of the PKM.

Contribution: In this paper, for the first time in the literature, a second-order inverse dynamics algorithm for computing the second time derivative of (1) for non-redundant PKM with simple limbs is proposed. It builds upon the dynamics formulation introduced in [23], [24] and the recursive inverse dynamics algorithms for serial robots introduced in [13], [14] using a Lie group formulation. For an accessible introduction to the general Lie group formulation, the reader is referred to [26], [16], while a summary of the particular formulation used in this paper can be found in [27], [28]. Due to space limitation, the algorithm is presented for kinematically non-redundant PKM only, while it is applicable to general fully parallel PKM. The actual (flatness-based) control using the presented second-order inverse dynamics solution will be topic of a forthcoming publication. This paper provides the algorithmic foundation, as [8]-[14] do for serial robots.

Organization: Sec. II recalls the kinematic modeling of PKM emphasizing that a single limb can be treated as a serial kinematic chain. The closed form inverse dynamics formulation is summarized and expressed in a form suited for computing time derivatives in Sec. III. The new algorithm for computing the 4th time derivatives of the inverse and forward kinematics is presented in Sec. IV, which is then used for the second time derivative of the inverse dynamics in Sec. V. Numerical results are presented in Sec. VI for a 6UPS Gough-Stewart platform (GSP) and a planar 2-DOF 3RRR PKM. Outlook and suggestions for future research are given in Sec. VII.

II. KINEMATICS

A. Kinematic topology

A fully parallel PKM consists of a moving platform connected to the fixed platform (ground) by L limbs. Each limb is (in this paper) a serial kinematic chain (therefore called simple). A typical example is the GSP whose topological graph is shown in Fig. 1a). Following the common convention, technical joints are modeled as combination of 1-DOF joints, so that each limb $l = 1, \dots, L$ comprises $N_l = 6$ joints (edges). Topologically, the PKM consists of L congruent sub-graphs connected to the platform, one is shown in Fig. 1b). This gives rise to a tailored kinematics modeling, where the kinematics of each sub-graph is described in terms of the platform motion.

B. Forward and inverse kinematics of limbs

Denote with $\boldsymbol{\vartheta}_{(l)} \in \mathbb{V}^{N_l}$ the vector of N_l joint coordinates of limb l when connected to the platform. The velocity 'forward kinematics' of limb l gives the platform twist $\mathbf{V}_\mathrm{p}$ in terms of the $6 \times N_l$ *forward kinematics Jacobian of limb* l

$$\mathbf{V}_\mathrm{p} = \mathbf{J}_{\mathrm{p}(l)} \dot{\boldsymbol{\vartheta}}_{(l)}. \tag{3}$$

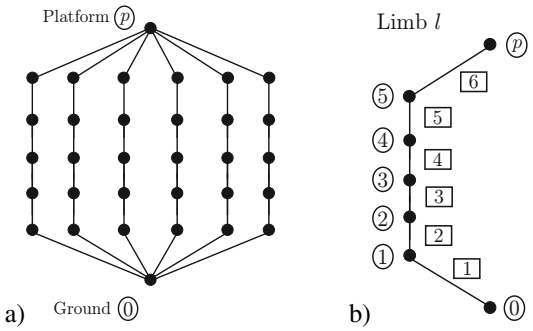


Fig. 1: a) Topological graph of a Gough-Stewart platform. b) Sub-graph corresponding to one limb including platform.

This relates platform twist and joint rates $\dot{\boldsymbol{\vartheta}}_{(l)}$ when the platform is connected only to the separated limb l.

The platform of the PKM, i.e. when connected to all limbs, has DOF $\delta_\mathrm{p} \leq N_l$, and only δ_p components of $\mathbf{V}_\mathrm{p}$ are independent. The *task space velocity* vector $\mathbf{V}_\mathrm{t}$ is introduced accordingly comprising the δ_p relevant components of the platform twist $\mathbf{V}_\mathrm{p}$. This is formally expressed as

$$\mathbf{V}_\mathrm{p} = \mathbf{P}_\mathrm{p} \mathbf{V}_\mathrm{t} \tag{4}$$

with a unimodular $6 \times \delta_\mathrm{p}$ velocity distribution matrix $\mathbf{P}_\mathrm{p}$, which assigns the δ_p components of the task space velocity to the components of the platform twist. This relates platform twist and task space velocity. The latter is used in the task space formulation of EOM. The intermediate step via the platform twist is crucial to account for general PKM.

Consider the platform when only connected to limb l. The platform has DOF $\delta_{\mathrm{p}(l)} \geq \delta_\mathrm{p}$, which is the generic rank of $\mathbf{J}_{\mathrm{p}(l)}$, whereas the serial chain has DOF $N_l \geq \delta_{\mathrm{p}(l)}$. If $\delta_{\mathrm{p}(l)} = \delta_\mathrm{p}$, the PKM is called *equimobile* [23], [24], i.e. there is a $\delta_{\mathrm{p}(l)} \times N_l$ submatrix $\mathbf{J}_{\mathrm{t}(l)}$ of $\mathbf{J}_{\mathrm{p}(l)}$ so that $\mathbf{V}_\mathrm{t} = \mathbf{J}_{\mathrm{t}(l)} \dot{\boldsymbol{\vartheta}}_{(l)}$. For a non-equimobile PKM ($\delta_{\mathrm{p}(l)} > \delta_\mathrm{p}$, i.e. the platform has a different mobility when connected to the kinematic chain of a single limb and when connected to all limbs), only $\delta_{\mathrm{p}(l)}$ rows correspond to the task space velocity, while the remaining $\delta_{\mathrm{p}(l)} - \delta_\mathrm{p}$ rows represent constraints on the platform motion. This is expressed with help of a $\delta_{\mathrm{p}(l)} \times \delta_\mathrm{p}$ velocity distribution matrix $\mathbf{D}_{\mathrm{t}(l)}$ so that

$$\mathbf{D}_{\mathrm{t}(l)} \mathbf{V}_\mathrm{t} = \mathbf{J}_{\mathrm{t}(l)} \dot{\boldsymbol{\vartheta}}_{(l)}. \tag{5}$$

The $\delta_{\mathrm{p}(l)} \times N_l$ matrix $\mathbf{J}_{\mathrm{t}(l)}$ is the *task space Jacobian of limb* l, formally defined as $\mathbf{J}_{\mathrm{t}(l)} := \mathbf{P}_{\mathrm{t}(l)} \mathbf{J}_{\mathrm{p}(l)}$, with $\delta_{\mathrm{p}(l)} \times 6$ selection matrix $\mathbf{P}_{\mathrm{t}(l)}$. The latter simply extracts the relevant rows from the forward kinematics Jacobian. Throughout the paper, it is assumed that the PKM is kinematically non-redundant, i.e. $\delta_{\mathrm{p}(l)} = N_l$, and $\mathbf{J}_{\mathrm{t}(l)}$ is a full rank $N_l \times N_l$ matrix, implying that the mechanism DOF δ is equal to δ_p.

Introducing the *inverse kinematics Jacobian of limb* l, $\mathbf{F}_{(l)} := \mathbf{J}_{\mathrm{t}(l)}^{-1} \mathbf{D}_{\mathrm{t}(l)}$, the solution of the inverse kinematics problem at velocity and acceleration level is, respectively,

$$\dot{\boldsymbol{\vartheta}}_{(l)} = \mathbf{F}_{(l)} \mathbf{V}_\mathrm{t}, \quad \text{with} \quad \mathbf{F}_{(l)} := \mathbf{J}_{\mathrm{t}(l)}^{-1} \mathbf{D}_{\mathrm{t}(l)} \tag{6}$$

$$\ddot{\boldsymbol{\vartheta}}_{(l)} = \mathbf{F}_{(l)} \dot{\mathbf{V}}_\mathrm{t} + \dot{\mathbf{F}}_{\mathrm{t}(l)} \mathbf{V}_\mathrm{t}. \tag{7}$$

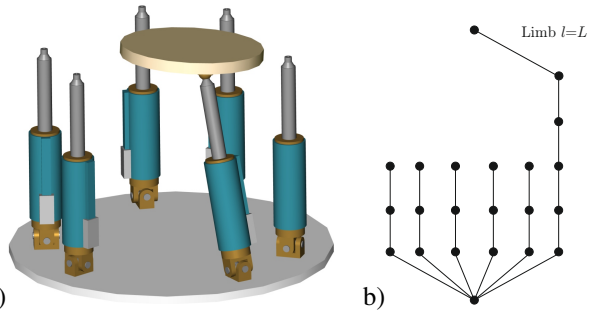


Fig. 2: a) Tree-topology model of Gough-Stewart PKM. Platform connected to one limb. b) Corresponding spanning tree.

C. Kinematics of associated tree-topology system

A tree-topology system is introduced by eliminating cut-joints, which is a standard approach in multibody dynamics [29], [30], [18]. A fully parallel PKM possesses $L-1$ fundamental cycles. Taking into account the special topology of PKM, the $L-1$ joints connecting the platform to the respective limb are selected as cut-joints. That is, a spanning tree is introduced so that the platform is the leaf of one serial chain, i.e. the platform remains attached to one limb while it is disconnected from the remaining $L-1$ limbs. W.l.o.g., limbs $l=1,\dots,L-1$ are cut off from the platform. In the tree-topology system, the platform is then connected to limb L, shown in Fig. 2 for the 6-DOF GSP.

Denote with $\bar{\boldsymbol{\vartheta}}_{(l)} \in \mathbb{V}^{n_l}$ the n_l tree-joint coordinates of limb $l=1,\dots,L-1$ when disconnected from the platform (this is $\boldsymbol{\vartheta}_{(l)}$ with variables of the cut joint connecting to the platform removed), and with $\bar{\mathbf{F}}_{(l)}$ the $n_l \times \delta_\mathrm{p}$ submatrix of $\mathbf{J}_{\mathrm{t}(l)}^{-1}$ obtained by eliminating the rows corresponding to the variables of the cut-joint connecting the limb to the platform. Then $\dot{\bar{\boldsymbol{\vartheta}}}_{(l)} = \bar{\mathbf{F}}_{(l)}\mathbf{V}_\mathrm{t}$. In case of the GSP, the 3-DOF spherical joints are cut, and the remaining $L-1$ chains have $n_l=3$ joint coordinates. Limb L does not contain cut-joints, and comprises the platform in the tree-topology, so that $\bar{\boldsymbol{\vartheta}}_{(L)} = \boldsymbol{\vartheta}_{(L)}$ are the corresponding N_L tree-joint coordinates.

Summarizing the tree-joint coordinates of all limbs, i.e. $\dot{\bar{\boldsymbol{\vartheta}}}_{(l)}, l=1,\dots,L-1$ and $\dot{\boldsymbol{\vartheta}}_{(L)}$, in $\dot{\bar{\boldsymbol{\vartheta}}}$, and the corresponding inverse kinematics Jacobians, i.e. $\bar{\mathbf{F}}_{(l)}, i=1,\dots,L-1$ and $\mathbf{F}_{(L)}$, of the limbs in $\bar{\mathbf{F}}$, gives rise to the velocity inverse kinematics solution of the tree-topology mechanism

$$\dot{\bar{\boldsymbol{\vartheta}}} = \bar{\mathbf{F}}\mathbf{V}_\mathrm{t} \tag{8}$$

which determines all tree-joint velocities in terms of the task space velocity satisfying the loop constraints.

The velocity inverse kinematics of the PKM is expressed by means of the inverse kinematics Jacobian $\mathbf{J}_\mathrm{IK}$ as

$$\dot{\boldsymbol{\vartheta}}_\mathrm{a} = \mathbf{J}_\mathrm{IK}\mathbf{V}_\mathrm{t} \tag{9}$$

where vector $\boldsymbol{\vartheta}_\mathrm{a}$ comprises the coordinates $\vartheta_{i(l)}$ corresponding to the actuated joints. For non-redundantly actuated PKM, $\boldsymbol{\vartheta}_\mathrm{a}$ are generalized coordinates. For many PKM with simple limbs (serial chains), the inverse kinematics Jacobian can be determined easily [31], [21] (see Table 6 in [32]). For PKM with complex limbs, this is more involved [24], [25]. With the above inverse kinematics solution of the mechanism (8), the inverse kinematics of the PKM is already available, however. The IK Jacobian consists of the rows of $\bar{\mathbf{F}}$ corresponding to the actuated joints. This will be exploited when computing derivatives.

III. Closed-Form Inverse Dynamics of PKM

To derive the dynamics EOM of the PKM, the EOM of the tree-topology system introduced above are derived first, and the constraints are imposed using the above inverse kinematics solution. The EOM of the tree-topology system split into the EOM of the L individual limbs.

A. Joint Space Formulation of EOM of Individual Limbs

The EOM of limb $l=1,\dots,L$ can be written in the standard form, as for any tree-topology MBS, as

$$\bar{\mathbf{M}}_{(l)}\ddot{\bar{\boldsymbol{\vartheta}}}_{(l)} + \bar{\mathbf{C}}_{(l)}\dot{\bar{\boldsymbol{\vartheta}}}_{(l)} + \bar{\mathbf{Q}}_{(l)}^\mathrm{grav} = \bar{\mathbf{Q}}_{(l)} \tag{10}$$

where $\bar{\mathbf{M}}_{(l)}(\bar{\boldsymbol{\vartheta}}_{(l)})$ is the generalized mass matrix, $\bar{\mathbf{C}}_{(l)}(\bar{\boldsymbol{\vartheta}}_{(l)}, \dot{\bar{\boldsymbol{\vartheta}}}_{(l)})$ the generalized Coriolis/centrifugal matrix, $\bar{\mathbf{Q}}_{(l)}^\mathrm{grav}(\bar{\boldsymbol{\vartheta}}_{(l)})$ generalized forces due to gravity, and $\bar{\mathbf{Q}}_{(l)}$ represents all applied forces including actuation forces. Friction, contact, and other forces are omitted, for simplicity. Expression (10) is indeed the EOM of a serial robot. These equations possess compact closed form expressions that were formalized by means of the spatial operator algebra [33], [18], [34], which is conceptually similar to the natural orthogonal complement approach [35], [20]. Using matrix Lie group methods, all expressions are intrinsically given in terms of the screw coordinates and frame transformation matrices [16], [28]. These equations can also be evaluated with $O(n_l)$ complexity using recursive Lie group algorithms [15], [16] that will be employed for the higher-order inverse dynamics in Sec. IV and V.

B. Task Space Formulation of EOM in Closed Form

If the PKM is kinematically non-redundant, $\delta = \delta_\mathrm{p}$, the PKM motion is determined by the platform motion. Then the EOM (10) govern the dynamics of a separated limb. The overall task space formulation of EOM of the PKM is (omitting EE-loads) given in (1) with the $\delta \times \delta$ generalized mass matrix and Coriolis/centrifugal matrix

$$\mathbf{M}_\mathrm{t}(\boldsymbol{\vartheta}) := \sum_{l=1}^{L} \bar{\mathbf{F}}_{(l)}^T \bar{\mathbf{M}}_{(l)} \bar{\mathbf{F}}_{(l)} \tag{11}$$

$$\mathbf{C}_\mathrm{t}(\boldsymbol{\vartheta}, \dot{\boldsymbol{\vartheta}}) := \sum_{l=1}^{L} \bar{\mathbf{F}}_{(l)}^T (\bar{\mathbf{C}}_{(l)} \bar{\mathbf{F}}_{(l)} + \bar{\mathbf{M}}_{(l)} \dot{\bar{\mathbf{F}}}_{(l)}). \tag{12}$$

The generalized forces due to gravity are

$$\mathbf{W}_\mathrm{t}^\mathrm{grav}(\boldsymbol{\vartheta}) := \sum_{l=1}^{L} \bar{\mathbf{F}}_{(l)}^T \bar{\mathbf{Q}}_{(l)}^\mathrm{grav}. \tag{13}$$

C. Inverse Dynamics Formulation

The inverse dynamics problem is to compute the actuation forces $\mathbf{u}$ for given motion of the PKM and applied wrenches. Instead of using the closed form EOM (1), the inverse dynamics solution is expressed as

$$\mathbf{u} = \mathbf{J}_{\mathrm{IK}}^{-T} \sum_{l=1}^{L} \bar{\mathbf{F}}_{(l)}^{T} \bar{\mathbf{Q}}_{(l)}(\boldsymbol{\vartheta}_{(l)}, \dot{\boldsymbol{\vartheta}}_{(l)}, \ddot{\boldsymbol{\vartheta}}_{(l)}) \tag{14}$$

using the inverse kinematics solution (6) and (7), with the EOM (10) of the limbs. The advantage of this form is that it allows for separate evaluation of the dynamics EOM of the limbs, of their inverse kinematics solution, and of the inverse kinematics Jacobian $\mathbf{J}_{\mathrm{IK}}$, respectively the forward kinematics Jacobian $\mathbf{J}_{\mathrm{IK}}^{-1}$. This holds true for their time derivatives, which allows for an efficient evaluation of the higher-order inverse dynamics as described in the next section. Note that all terms in (10), and (14), depend on $\boldsymbol{\vartheta}$ and its derivatives.

IV. Fourth-Order Forward/Inverse Kinematics

Time derivatives of the inverse dynamics solution (14) necessitate derivatives of the EOM (10) as well as of the inverse kinematics solution (6) of the limbs. When solving the ordinary inverse dynamics problem of serial robots, the EOM of the form (10) are evaluated using a recursive $O(n_l)$ algorithm. Any such inverse dynamics algorithm, which takes joint variables $\boldsymbol{\vartheta}$, velocities $\dot{\boldsymbol{\vartheta}}$, and accelerations $\ddot{\boldsymbol{\vartheta}}$ as inputs, consists of a forward kinematics loop and an inverse dynamics loop. In the forward kinematics run, the configurations, velocities, and accelerations of all bodies are computed. Solving this collectively for all limbs is called the *forward kinematics of the mechanism*. The task space formulation (10) additionally involves solving the *inverse kinematics problem of the mechanism*, i.e. computing $\boldsymbol{\vartheta}, \dot{\boldsymbol{\vartheta}}, \ddot{\boldsymbol{\vartheta}}$ from given platform motion, which means solving the inverse kinematics of all limbs. The second-order inverse dynamics additionally involves computing the third- and fourth-order inverse and forward kinematics solution. To this end, a combined fourth-order forward/inverse kinematics $O(N_l)$ algorithm is introduced. Computation of derivatives of the inverse kinematics Jacobian in (14) is discussed in Sec. V-B.

A. Derivatives of the inverse kinematics solution of limbs

Taking derivatives of (5), and solving for the highest derivative of $\boldsymbol{\vartheta}_{(l)}$, yields the ν-th time derivative of the inverse kinematics solution (6) of limb l

$$\frac{d^{\nu}}{dt^{\nu}}\boldsymbol{\vartheta}_{(l)} = \mathbf{F}_{(l)} \frac{d^{\nu-1}}{dt^{\nu-1}} \mathbf{V}_{\mathrm{t}} - \mathbf{J}_{\mathrm{t}(l)}^{-1} \mathbf{P}_{\mathrm{t}(l)} \mathbf{c}_{(l)}^{\nu} \tag{15}$$

$$\text{with} \quad \mathbf{c}_{(l)}^{\nu} := \sum_{k=1}^{\nu-1} \tbinom{\nu-1}{k} \frac{d^{k}}{dt^{k}} \mathbf{J}_{\mathrm{p}(l)} \frac{d^{\nu-k}}{dt^{\nu-k}} \boldsymbol{\vartheta}_{(l)}. \tag{16}$$

For derivatives up to order $\nu = 4$, these are

$$\mathbf{c}_{(l)}^{2} := \dot{\mathbf{J}}_{\mathrm{p}(l)} \dot{\boldsymbol{\vartheta}}_{(l)}, \;\; \mathbf{c}_{(l)}^{3} := \ddot{\mathbf{J}}_{\mathrm{p}(l)} \dot{\boldsymbol{\vartheta}}_{(l)} + 2\dot{\mathbf{J}}_{\mathrm{p}(l)} \ddot{\boldsymbol{\vartheta}}_{(l)} \tag{17}$$

$$\mathbf{c}_{(l)}^{4} := \dddot{\mathbf{J}}_{\mathrm{p}(l)} \dot{\boldsymbol{\vartheta}}_{(l)} + 2\ddot{\mathbf{J}}_{\mathrm{p}(l)} \ddot{\boldsymbol{\vartheta}}_{(l)} + 3\dot{\mathbf{J}}_{\mathrm{p}(l)} \dddot{\boldsymbol{\vartheta}}_{(l)}. \tag{18}$$

Computing inverse kinematics derivatives boils down to the inversion of the task space Jacobian and computing derivatives of the forward kinematics Jacobians of the limbs. They admit the closed form expressions (23-25), see appendix.

B. Recursive forward/inverse kinematics algorithm

The second-order inverse dynamics involves solving the 4th-order inverse kinematics of the mechanism (computing $\dot{\boldsymbol{\vartheta}}_{(l)}, \ldots, \ddot{\ddot{\boldsymbol{\vartheta}}}_{(l)}$ from $\mathbf{V}_{\mathrm{t}}, \ldots, \ddot{\ddot{\mathbf{V}}}_{\mathrm{t}}$) and the forward kinematics of the mechanism (computing all body twists $\mathbf{V}_{i(l)}, \ldots, \ddot{\ddot{\mathbf{V}}}_{i(l)}$ from $\dot{\boldsymbol{\vartheta}}_{(l)}, \ldots, \ddot{\ddot{\boldsymbol{\vartheta}}}_{(l)}$). Both are computed together in a single run. Recursive algorithms for computing the fourth-order forward kinematics of serial chains were reported in [11], [12] using classical vector formulations. In the following, the algorithm introduced in [13] that uses the Lie group formulation from [27] is complemented with the inverse kinematics. An introduction to the Lie group formulation of serial chains in general can be found in [16]. A brief summary of the notation is given in the appendix. The homogenous transformation $\mathbf{C}_{i(l)}$ (not to be confused with $\mathbf{C}_{\mathrm{t}}$ in (12)) represents the absolute configuration of body i of limb l relative to the inertial frame, $\mathbf{C}_{i,i-1}$ is the relative configuration of body i and $i-1$, and $\mathbf{C}_{\mathrm{p}}$ denotes the absolute platform configuration relative to the inertial frame.

The algorithm consists of four subsequent loops. Input to the first loop is the current PKM state $(\boldsymbol{\vartheta}, \mathbf{V}_{\mathrm{t}})$. Solving the geometric inverse kinematics problem, computing $\boldsymbol{\vartheta}$ from given platform motion configuration $\mathbf{C}_{\mathrm{p}}$, is not the subject here, and $\boldsymbol{\vartheta}$ is assumed to be given. Denote with $\mathbf{g}$ the vector of gravitational acceleration expressed in inertial frame, then the corresponding acceleration screw of the ground is $\dot{\mathbf{V}}_{0(l)} = (\mathbf{0}, \mathbf{g})^{T}$. The algorithm is derived by splitting the forward kinematics run presented in [13], necessary as the inverse kinematics must be solved for each order first. Matrix $\mathbf{Ad}_{\mathbf{C}_{i,i-1}}$ transforms a twist from body $i-1$ to body i, and $\mathbf{ad}_{\mathbf{X}}\mathbf{Y}$ yields the Lie bracket of $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^6$, also called 'spatial cross product' ´[34], [18]. Notice that $n_L = N_L$.

1st-order Inverse and 0th-order Forward Kinematics

- Input: $\boldsymbol{\vartheta}, \mathbf{V}_{\mathrm{t}}$
- FOR $l = 1, \ldots, L$ DO (possible in parallel)
  - FOR $i = 1, \ldots, N_l$ (omitting subscript (l))
    - $\mathbf{C}_i = \mathbf{C}_{i-1}\mathbf{B}_i \exp(\mathbf{X}_i \vartheta_i)$
  - compute $\mathbf{J}_{\mathrm{p}(l)}$ with (22)
  - $\mathbf{F}_{(l)} = \mathbf{J}_{\mathrm{t}(l)}^{-1} \mathbf{D}_{\mathrm{t}(l)}$
  - $\dot{\boldsymbol{\vartheta}}_{(l)} = \mathbf{F}_{(l)} \mathbf{V}_{\mathrm{t}}$ $\quad (*)$
- Output: $\mathbf{J}_{\mathrm{p}(l)}, \mathbf{F}_{(l)}, \dot{\boldsymbol{\vartheta}}, \mathbf{C}_{i(l)}$

2nd-order Inverse and 1st-order Forward Kinematics

- Input: $\boldsymbol{\vartheta}, \dot{\boldsymbol{\vartheta}}, \dot{\mathbf{V}}_{\mathrm{t}}, \mathbf{J}_{\mathrm{p}(l)}, \mathbf{F}_{(l)}, \mathbf{C}_{i(l)}$
- FOR $l = 1, \ldots, L$ DO (possible in parallel)
  - FOR $i = 1, \ldots, N_l$ (omitting subscript (l))
    - $\mathbf{V}_i = \mathbf{Ad}_{\mathbf{C}_{i,i-1}} \mathbf{V}_{i-1} + \mathbf{X}_i \dot{\vartheta}_i$
    - compute $\dot{\mathbf{J}}_{\mathrm{p}(l),i}$ with (23)
  - compute $\mathbf{c}_{(l)}^{2}$ with (17)
  - $\ddot{\boldsymbol{\vartheta}}_{(l)} = \mathbf{F}_{(l)} \big(\dot{\mathbf{V}}_{\mathrm{t}} - \mathbf{J}_{\mathrm{t}(l)}^{-1} \mathbf{P}_{\mathrm{t}(l)} \mathbf{c}_{(l)}^{2} \big)$ $\quad (*)$
- Output: $\ddot{\boldsymbol{\vartheta}}, \mathbf{V}_i$

3rd-order Inverse and 2nd-order Forward Kinematics

- Input: $\boldsymbol{\vartheta}, \dot{\boldsymbol{\vartheta}}, \ddot{\boldsymbol{\vartheta}}, \ddot{\mathbf{V}}_{\mathrm{t}}, \mathbf{J}_{\mathrm{p}(l)}, \mathbf{F}_{(l)}, \mathbf{C}_{i(l)}, \mathbf{V}_{i(l)}$

- FOR $l = 1, \dots, L$ DO (possible in parallel)
  FOR $i = 1, \dots, N_l$ (omitting subscript (l))
  $\dot{\mathbf{V}}_i = \mathbf{Ad}_{\mathbf{C}_{i,i-1}} \dot{\mathbf{V}}_{i-1} - \dot{\vartheta}_i \mathbf{ad}_{\mathbf{X}_i} \mathbf{V}_i + \mathbf{X}_i \ddot{\vartheta}_i$
  compute $\ddot{\mathbf{J}}_{\mathrm{p}(l),i}$ with (24)
  compute $\mathbf{c}^3_{(l)}$ with (17)
  $\dddot{\boldsymbol{\vartheta}} = \mathbf{F}_{(l)} \big(\dot{\mathbf{V}}_{\mathrm{t}} - \mathbf{J}^{-1}_{\mathrm{t}(l)} \mathbf{P}_{\mathrm{t}(l)} \mathbf{c}^3_{(l)} \big)$ $(*)$
- Output: $\dddot{\boldsymbol{\vartheta}}, \dot{\mathbf{V}}_{i(l)}$

4th-order Inverse and 3rd-order Forward Kinematics

- Input: $\boldsymbol{\vartheta}, \dot{\boldsymbol{\vartheta}}, \ddot{\boldsymbol{\vartheta}}, \dddot{\boldsymbol{\vartheta}}, \ddot{\mathbf{V}}_{\mathrm{t}}, \mathbf{J}_{\mathrm{p}(l)}, \mathbf{F}_{(l)}, \mathbf{C}_{i(l)}, \mathbf{V}_{i(l)}, \dot{\mathbf{V}}_{i(l)}$
- FOR $l = 1, \dots, L$ DO (possible in parallel)
  FOR $i = 1, \dots, N_l$ (omitting subscript (l))
  $\ddot{\mathbf{V}}_i = \mathbf{Ad}_{\mathbf{C}_{i,i-1}} \ddot{\mathbf{V}}_{i-1} + \mathbf{X}_i \dddot{\vartheta}_i$
  $-\mathbf{ad}_{\mathbf{X}_i} (\dddot{\vartheta}_i \mathbf{V}_i + 2\ddot{\vartheta}_i \dot{\mathbf{V}}_i) - \dot{\vartheta}_i^2 \mathbf{ad}^2_{\mathbf{X}_i} \mathbf{V}_i$
  compute $\dddot{\mathbf{J}}_{\mathrm{p}(l),i}$ with (25)
  compute $\mathbf{c}^4_{(l)}$ with (18)
  $\ddddot{\boldsymbol{\vartheta}} = \mathbf{F}_{(l)} \big(\ddot{\mathbf{V}}_{\mathrm{t}} - \mathbf{J}^{-1}_{\mathrm{t}(l)} \mathbf{P}_{\mathrm{t}(l)} \mathbf{c}^4_{(l)} \big)$ $(*)$
- Output: $\ddddot{\boldsymbol{\vartheta}}, \ddot{\mathbf{V}}_{i(l)}$

4th-order Forward Kinematics

- Input: $\boldsymbol{\vartheta}, \dot{\boldsymbol{\vartheta}}, \ddot{\boldsymbol{\vartheta}}, \dddot{\boldsymbol{\vartheta}}, \ddddot{\vartheta}_i, \mathbf{J}_{\mathrm{p}(l)}, \mathbf{F}_{(l)}, \mathbf{C}_{i(l)}, \mathbf{V}_{i(l)}, \dot{\mathbf{V}}_{i(l)}, \ddot{\mathbf{V}}_{i(l)}$
- FOR $l = 1, \dots, L$ DO (possible in parallel)
  FOR $i = 1, \dots, N_l$

$$\begin{aligned}\dddot{\mathbf{V}}_i = \mathbf{Ad}_{\mathbf{C}_{i,i-1}} \dddot{\mathbf{V}}_{i-1} + \mathbf{X}_i \ddddot{\vartheta}_i \\ - \mathbf{ad}_{\mathbf{X}_i} (\dddot{\vartheta}_i \mathbf{V}_i + 3\ddot{\vartheta}_i \dot{\mathbf{V}}_i + 3\dot{\vartheta}_i \ddot{\mathbf{V}}_i) \\ - 3\dot{\vartheta}_i \mathbf{ad}^2_{\mathbf{X}_i} (\ddot{\vartheta}_i \mathbf{V}_i + \dot{\vartheta}_i \dot{\mathbf{V}}_i) - \dot{\vartheta}_i^3 \mathbf{ad}^3_{\mathbf{X}_i} \mathbf{V}_i\end{aligned}$$

- Output: $\dddot{\mathbf{V}}_{i(l)}$

C. Computational complexity

In contrast to the forward kinematics of serial robots, prior to the νth-order forward kinematics, the νth-order inverse kinematic problem is solved (for which the (ν-1)st time derivative of $\mathbf{J}_{\mathrm{t}(l)}$ is computed) separately for each order. The νth-order kinematics run ($D^{(\nu)}\boldsymbol{\vartheta} \mapsto D^{(\nu-1)}\mathbf{V}_{i(l)}$) for limb l has complexity $O(N_l)$. Each run involves an expression $\mathbf{a} = \mathbf{F}_{(l)}\mathbf{b}$ with $\mathbf{F}_{(l)} := \mathbf{J}^{-1}_{\mathrm{t}(l)} \mathbf{D}_{\mathrm{t}(l)}$, indicted with (*), solving the νth-order inverse kinematics problem ($D^{(\nu-1)}\mathbf{V}_{\mathrm{t}} \mapsto D^{(\nu)}\boldsymbol{\vartheta}_{(l)}$). Inversion of the $\delta_{\mathrm{p}} \times \delta_{\mathrm{p}}$ task space Jacobians is avoided by solving $\mathbf{J}_{\mathrm{t}(l)}\mathbf{b} = \mathbf{D}_{\mathrm{t}(l)}\mathbf{a}$ for $\mathbf{b}$. LU-decompositions of $\mathbf{J}_{\mathrm{t}(l)}$ are computed before the kinematics runs, and reused for solving the four equations (*). The total number of operations is $L\left(\frac{2}{3}\delta_{\mathrm{p}}^3 + 8\delta_{\mathrm{p}}^2\right)$. The overall complexity of the kinematics run is $O(L \cdot N_l)$, except for solving (*).

V. Second-Order Inverse Dynamics

A. Derivatives of the inverse dynamics solution

The first and second time derivative of the inverse dynamics solution (14) are readily found as (omitting EE loads)

$$\dot{\mathbf{u}} = \mathbf{J}^{-T}_{\mathrm{IK}} \Big(\sum_{l=1}^{L} \big(\mathbf{F}^T_{(l)} \dot{\bar{\mathbf{Q}}}_{(l)} + \dot{\mathbf{F}}^T_{(l)} \bar{\mathbf{Q}}_{(l)} \big) - \dot{\mathbf{J}}^T_{\mathrm{IK}} \mathbf{u} \Big) \tag{19}$$

$$\begin{aligned}\ddot{\mathbf{u}} = \mathbf{J}^{-T}_{\mathrm{IK}} \Big(\sum_{l=1}^{L} \big(\mathbf{F}^T_{(l)} \ddot{\bar{\mathbf{Q}}}_{(l)} + 2\dot{\mathbf{F}}^T_{(l)} \dot{\bar{\mathbf{Q}}}_{(l)} + \ddot{\mathbf{F}}^T_{(l)} \bar{\mathbf{Q}}_{(l)} \big) \\ - \ddot{\mathbf{J}}^T_{\mathrm{IK}} \mathbf{u} - 2\dot{\mathbf{J}}^T_{\mathrm{IK}} \dot{\mathbf{u}} \Big)\end{aligned} \tag{20}$$

with $\dot{\mathbf{F}}_{(l)} = -\mathbf{F}_{(l)} \dot{\mathbf{J}}_{\mathrm{t}(l)} \mathbf{F}_{(l)}$ and $\ddot{\mathbf{F}}_{(l)} = 2\dot{\mathbf{F}}_{(l)} \dot{\mathbf{J}}_{\mathrm{t}(l)} \dot{\mathbf{F}}_{(l)} - \mathbf{F}_{(l)} \ddot{\mathbf{J}}_{\mathrm{t}(l)} \mathbf{F}_{(l)}$. Noting that $\mathbf{J}_{\mathrm{t}(l)} := \mathbf{P}_{\mathrm{t}(l)} \mathbf{J}_{\mathrm{p}(l)}$, the latter are available with derivatives of $\mathbf{J}_{\mathrm{p}(l)}$ in (23)-(25) in appendix.

B. Derivatives of manipulator inverse kinematics Jacobian

The manipulator inverse kinematics Jacobian $\mathbf{J}_{\mathrm{IK}}$ consists of the δ rows of the inverse kinematics Jacobians of the l limbs. That is, each row of $\mathbf{F}_{(l)}$ that corresponds to an actuated joint delivers one row of $\mathbf{J}^{-1}_{\mathrm{IK}}$. If each limb comprises one actuator, then $L = \delta$, and each $\mathbf{F}_{(l)}$ contributes one row. Thus the derivatives of $\mathbf{J}_{\mathrm{IK}}$ are already available with the derivatives of $\mathbf{F}_{(l)}$ above.

C. Recursive 2nd-Order $O(n_l)$ Inverse Dynamics Algorithm

Evaluating (20) involves computing derivatives of $\bar{\mathbf{Q}}_{(l)}$, i.e. of the second-oder inverse dynamics solution of a serial kinematic chain. The advantage of the proposed higher-order inverse dynamics method for PKM is that this evaluation is separated from the kinematics modeling. Thus any computation scheme or software that delivers the inverse dynamics solution derivatives can be used (which may have any level of modeling detail including flexibilities etc). Various recursive algorithms have been proposed to evaluate the second time derivatives of the EOM (10) of the limbs, e.g. [11], [12]. Lie group formulations were proposed in [13], [14], which are coordinate-invariant and compact, and thus easy to implement. In the following, the inverse dynamics run of the algorithm in [13] is adopted. The forward kinematics run is accomplished by the algorithm in Sec. IV. Denoting with $\mathbf{M}_i$ the 6×6 mass matrix of body i expressed in the (arbitrary) body-fixed frame, and the interbody wrenches with $\bar{\mathbf{W}}_i$ (merely algorithmic variables), the inverse dynamics run is:

Inverse Dynamics

- Input: $\mathbf{C}_{i(l)}, \mathbf{V}_{i(l)}, \dot{\mathbf{V}}_{i(l)}, \ddot{\mathbf{V}}_{i(l)}, \dddot{\mathbf{V}}_{i(l)}$
- FOR $l = 1, \dots, L$ DO (possible in parallel)
  FOR $i = N_l - 1, \dots, 1$ (omitting subscript (l))

$$\begin{aligned}\bar{\mathbf{W}}_i &= \mathbf{Ad}^T_{\mathbf{C}_{i+1,i}} \bar{\mathbf{W}}_{i+1} + \mathbf{M}_i \dot{\mathbf{V}}_i - \mathbf{ad}^T_{\mathbf{V}_i} \mathbf{M}_i \mathbf{V}_i \\ \dot{\bar{\mathbf{W}}}_i &= \mathbf{Ad}^T_{\mathbf{C}_{i+1,i}} (\dot{\bar{\mathbf{W}}}_{i+1} - \dot{\vartheta}_{i+1} \mathbf{ad}^T_{\mathbf{X}_{i+1}} \bar{\mathbf{W}}_{i+1}) \\ &\quad + \mathbf{M}_i \ddot{\mathbf{V}}_i - \mathbf{ad}^T_{\mathbf{V}_i} \mathbf{M}_i \dot{\mathbf{V}}_i - \mathbf{ad}^T_{\dot{\mathbf{V}}_i} \mathbf{M}_i \mathbf{V}_i \\ \ddot{\bar{\mathbf{W}}}_i &= \mathbf{Ad}^T_{\mathbf{C}_{i+1,i}} \Big(\ddot{\bar{\mathbf{W}}}_{i+1} - 2\dot{\vartheta}_{i+1} \mathbf{ad}^T_{\mathbf{X}_{i+1}} \dot{\bar{\mathbf{W}}}_{i+1} \\ &\quad + (\dot{\vartheta}^2_{i+1} \mathbf{ad}^{2T}_{\mathbf{X}_{i+1}} - \ddot{\vartheta}_{i+1} \mathbf{ad}^T_{\mathbf{X}_{i+1}}) \bar{\mathbf{W}}_{i+1} \Big) \\ &\quad + \mathbf{M}_i \dddot{\mathbf{V}}_i - \mathbf{ad}^T_{\mathbf{V}_i} \mathbf{M}_i \ddot{\mathbf{V}}_i - \mathbf{ad}^T_{\dot{\mathbf{V}}_i} \mathbf{M}_i \mathbf{V}_i \\ &\quad - 2\mathbf{ad}^T_{\dot{\mathbf{V}}_i} \mathbf{M}_i \dot{\mathbf{V}}_i \\ \bar{Q}_i = \mathbf{X}_i^T \bar{\mathbf{W}}_i, \quad \dot{\bar{Q}}_i &= \mathbf{X}_i^T \dot{\bar{\mathbf{W}}}_i, \ \ddot{\bar{Q}}_i = \mathbf{X}_i^T \ddot{\bar{\mathbf{W}}}_i\end{aligned}$$

- Output: $\bar{\mathbf{Q}}_{(l)}, \dot{\bar{\mathbf{Q}}}_{(l)}, \ddot{\bar{\mathbf{Q}}}_{(l)}$

D. Computational Aspects

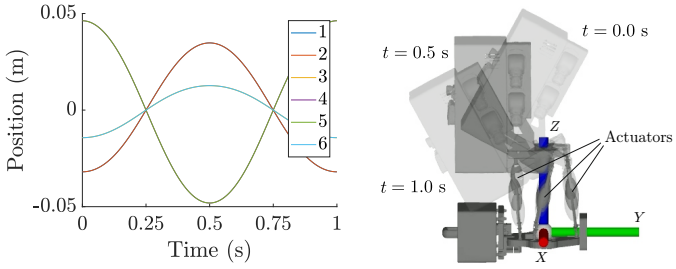
The computational effort for evaluating the inverse dynamics of limb l is of order $O(n_l)$, with $n_L = N_L$. The inverse dynamics run is thus of order $O(L \cdot n_l)$. The special topology of PKM with structurally identical limbs can be exploited for an efficient modular modeling, where the kinematic and dynamic EOM of a prototypical limb are reused for all limbs [23]. The recursive algorithm, it can be implemented once for a prototypical limb, and L instances are used during the computation. A simple example is shown in [36].

VI. Examples and Simulation Results

Implementation results are presented when the proposed algorithm is applied to a 6 DOF GSP and 2 DOF planar PKM. The algorithm was implemented in MATLAB.

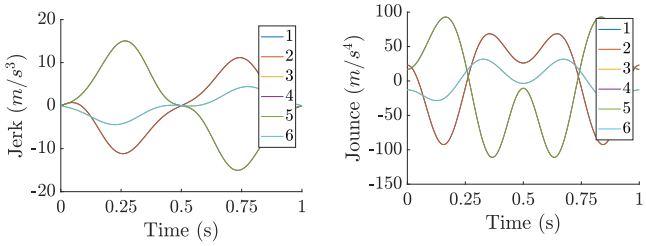
A. Gough-Stewart Platform

The proposed method is applied to compute the second-order inverse dynamics of a 6UPS GSP, which is used within the torso of the Recupera-Reha exoskeleton [37]. The topology information, joint screw coordinate vectors $\mathbf{X}_i$, the relative reference configurations $\mathbf{B}_i$ of all links, and the inertia data $\mathbf{M}_i$ w.r.t. to the body-fixed frames are extracted from the URDF description of the PKM. A pure roll motion of the platform parameterized as $\theta(t) = -A\cos(\omega t) + A + \theta_{\text{min}}$ with magnitude $A = (\theta_{\text{min}} - \theta_{\text{max}})/2$ and frequency $\omega = \frac{2\pi}{T}$, where $\theta_{\text{min}} = -0.5\,\text{rad}$, and $\theta_{\text{max}} = 0.5\,\text{rad}$ are the minimum and maximum of the roll angle θ, and $T = 1\,\text{s}$ is the cycle time. The actuator trajectories $\boldsymbol{\vartheta}_\text{a}(t)$ obtained via the inverse kinematics and the visualization of mechanism motion are shown in Fig. 3a and 3b, respectively, and their 3rd and 4th time derivatives in Fig. The actuator forces $\mathbf{u}(t)$ and their second time derivatives $\ddot{\mathbf{u}}(t)$ are shown in 4. Fig. 5, respectively. The solutions $\mathbf{u}(t)$ and $\boldsymbol{\vartheta}_\text{a}(t), \dot{\boldsymbol{\vartheta}}_\text{a}(t), \ddot{\boldsymbol{\vartheta}}_\text{a}(t)$ of the inverse kinematics respectively inverse dynamics are computed with the already existing and validated model. The higher order inverse kinematics and inverse dynamics results were verified by numerically differentiating $\mathbf{u}(t)$ and $\ddot{\boldsymbol{\vartheta}}(t)$ twice. The $O(n)$-solution and the numerical derivatives agree up to machine precision. As an indication of the computational performance, the total CPU time spent for 10000 evaluations of the MATLAB implementation of the second order inverse dynamics was measured on a standard laptop computer with Intel Core i7-4810MQ CPU at 2.8 GHz. Executing 10000 calls, the average CPU time per call was 0.75 ms, which is sufficient for real-time control applications. An order of magnitude reduction is expected from an optimized C++ implementation. It is to be noted that no parallelization opportunities were exploited. It will also be interesting to compare the performance with dedicate MBS codes. When modeled as MBS in terms of relative coordinates, this example has a spanning tree with 21 DOF subjected to 15 independent constraints.



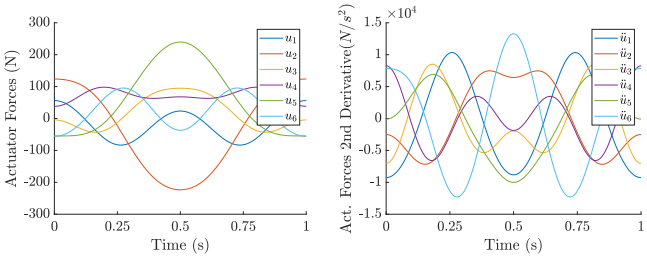
(a) Inv. kinematics solution $\boldsymbol{\vartheta}_\text{a}(t)$ (b) Motion visualization

Fig. 3: Inverse kinematics solution and animation of Gough-Stewart module of the Recupera-Reha exoskeleton.



(a) Jerk $\dddot{\boldsymbol{\vartheta}}_\text{a}$ (b) Jounce $\ddddot{\boldsymbol{\vartheta}}_\text{a}$

Fig. 4: Higher-order inverse kinematics solution.



(a) Actuator forces $\mathbf{u}(t)$ (b) 2nd time derivatives $\ddot{\mathbf{u}}(t)$

Fig. 5: Higher-order inverse dynamics results.

B. 2-DOF planar 3RRR PKM

The SEA driven 2-DOF planar PKM in Fig. 6 is considered to demonstrate the flatness-based control. The position of the EE in the plane of motion is the output of this PKM, described by its (x, y)-coordinates relative to the shown inertial frame (IFR). The PKM is controlled by three actuators, and would thus be redundantly actuated if the actuators were kinematically affixed to the arms. Instead, each arm is mounted on the output shaft of a SEA. The elastic coupling of output shaft and motor-gear unit is modeled by a torsion spring with stiffness constant k_i as indicated in Fig. 6. Geometric and inertia parameter are deduced from the CAD model, and motor specifications. The desired trajectory represents a pick and place task, where the EE alternates between two operating points as shown in Fig. 6. The EE trajectory follows a $\sin^2$ time profile of the jerk. A quantization of the joint angle encoder of 1/2000 is assumed, and white noise with amplitude of $3 \cdot 10^{-4}$m is added to model sensor noise. Fig. 7 shows the EE tracking error and the actuation torques of the flatness-based controller. When exact measurement and feedback is assumed, perfect tracking is achieved. Experimental results are reported in [38].

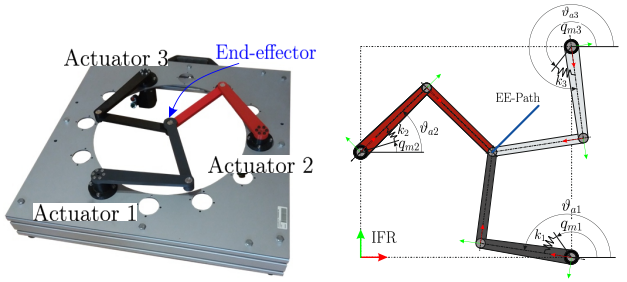


Fig. 6: Planar 2-DOF 3-RRR SEA-PKM. Physical prototype and schematic drawing.

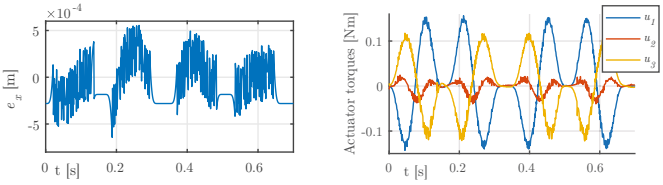


Fig. 7: Results of flatness-based control in task space. Tracking error e_x, e_y, and actuator torques u_1, u_2, u_3,

VII. Conclusion and Outlook

A recursive algorithm for efficient computation of first and second time derivatives of the inverse dynamics solution of non-redundant PKM with simple limbs (each limb a serial chain) is presented, which also solves the higher-order inverse kinematics problem. The Lie group framework is employed to this end as it offers compact coordinate invariant expressions. The formulation separates the overall kinematics of the PKM and the dynamics of the individual limbs. It is exploits inverse dynamics formulations for serial robots and a dedicated PKM modeling approach. This algorithm has $O\left(L \cdot n_l\right)$ complexity, except for the inversion of the forward kinematics Jacobian. The geometric inverse kinematics was not addressed as this depends on the particular PKM. A Newton-Raphson can always be used, however [31], [25].

The proposed formulation is a solution to a key challenge in trajectory tracking control of SEA-PKM. The modularity of the approach allows for further efficiency improvements, in particular as it immediately allows for parallelization. The exact computational effort of the recursive inverse dynamics algorithm will be investigated in future research. The complexity will be compared when using the closed form expression for the inverse dynamics of the limbs [19]. Future work includes extension of the second-order inverse dynamics algorithms to PKM with complex limbs (limbs with closed loops). Further possible extension of this work includes dealing with series-parallel hybrid robots, e.g. in humanoids with series elastic actuators [39], [40]. In addition to the Matlab implementation, a C++ implementation of the generic version of the algorithm is planned in Hybrid Robot Dynamics (HyRoDyn) [41] software framework.

Acknowledgement

Support by LCM K2 Center for Symbiotic Mechatronics within the Austrian COMET-K2 program is acknowledged.

Appendix: Lie-Group Modeling of Kinematics

The kinematics is described using the Lie group formulation introduced in [27], [24], which is a variant of that reported in [16]. For notation and background of the Lie group modeling, refer to [26], [42], [16]. Denote with $\mathbf{C}_{i(l)} \in SE\left(3\right)$ the configuration (pose) of body $i = 1, \ldots, N_l$ of limb l relative to a spatial inertial frame, which is represented as 4×4 homogenous transformation matrices. It is determined by the product of exponentials (omitting limb index (l))

$$\mathbf{C}_i\left(\boldsymbol{\vartheta}\right) = \mathbf{B}_1 \exp(\mathbf{X}_1 \vartheta_1) \cdot \ldots \cdot \mathbf{B}_i \exp(\mathbf{X}_i \vartheta_i) \tag{21}$$

with the screw coordinate vector $\mathbf{X}_i \in \mathbb{R}^6$ of joint i represented in the reference frame at body i, and $\mathbf{B}_i \in SE\left(3\right)$ is the configuration of body i relative to body $i-1$ in the zero reference $\boldsymbol{\vartheta}_{(l)} = \mathbf{0}$. The platform is the terminal body of each limb l with N_l bodies, and its configuration is $\mathbf{C}_{\mathrm{p}(l)} \equiv \mathbf{C}_{N_l(l)}$. The configuration of body i relative to the platform is $\mathbf{C}_{\mathrm{p},i} := \mathbf{C}_{\mathrm{p}}^{-1}\mathbf{C}_i$. The instantaneous screw coordinate vector of joint $i = 1, \ldots, N_l$ expressed in platform frame is $\mathbf{J}_{\mathrm{p}(l),i} = \mathbf{Ad}_{\mathbf{C}_{\mathrm{p},i}}\mathbf{X}_i$. This is the i-th column of the $6 \times N_l$ geometric forward kinematics Jacobian of limb l in (3)

$$\mathbf{J}_{\mathrm{p}(l)} = \begin{pmatrix} \mathbf{Ad}_{\mathbf{C}_{\mathrm{p},1}}\mathbf{X}_1 & \mathbf{Ad}_{\mathbf{C}_{\mathrm{p},2}}\mathbf{X}_2 & \cdots & \mathbf{X}_{N_l} \end{pmatrix}. \tag{22}$$

Therein, $\mathbf{Ad}_{\mathbf{C}_{i,j}}$ is the 6×6 matrix transforming twist/screw coordinates when expressed in the frame on body j to those when expressed in the frame on body i. It is also called 'composite body transformation operator' [33] or 'twist propagation matrix' [22]. Denote with $\mathbf{ad}$ the 6×6 'spatial cross product' matrix [33], [34] describing the Lie bracket.

Lemma 1: The time derivatives of the geometric Jacobian admit the closed form expressions (omitting limb index (l))

$$\dot{\mathbf{J}}_{\mathrm{p},i} = -\mathbf{ad}_{\Delta\mathbf{V}_{\mathrm{p},i}}\mathbf{J}_{\mathrm{p},i} \tag{23}$$

$$\ddot{\mathbf{J}}_{\mathrm{p},i} = \left(\mathbf{ad}^2_{\Delta\mathbf{V}_{\mathrm{p},i}} - \mathbf{ad}_{\Delta\dot{\mathbf{V}}_{\mathrm{p},i}}\right)\mathbf{J}_{\mathrm{p},i} \tag{24}$$

$$\begin{aligned}\dddot{\mathbf{J}}_{\mathrm{p},i} = \big(&- \mathbf{ad}^3_{\Delta\mathbf{V}_{\mathrm{p},i}} + 2\mathbf{ad}_{\Delta\dot{\mathbf{V}}_{\mathrm{p},i}}\mathbf{ad}_{\Delta\mathbf{V}_{\mathrm{p},i}} \\ &+ \mathbf{ad}_{\mathbf{ad}_{\Delta\mathbf{V}_{\mathrm{p},i}}\Delta\dot{\mathbf{V}}_{\mathrm{p},i}} - \mathbf{ad}_{\Delta\ddot{\mathbf{V}}_{\mathrm{p},i}}\big)\mathbf{J}_{\mathrm{p},i}\end{aligned} \tag{25}$$

with the twist of body i relative to the platform, represented in the platform frame, and its derivatives given as

$$\Delta\mathbf{V}_{\mathrm{p},i} = \mathbf{V}_{\mathrm{p}} - \mathbf{Ad}_{\mathbf{C}_{\mathrm{p},i}}\mathbf{V}_i \tag{26}$$

$$\Delta\dot{\mathbf{V}}_{\mathrm{p},i} = \dot{\mathbf{V}}_{\mathrm{p}} - \mathbf{Ad}_{\mathbf{C}_{\mathrm{p},i}}\dot{\mathbf{V}}_i + \mathbf{ad}_{\Delta\mathbf{V}_{\mathrm{p},i}}\mathbf{V}_{\mathrm{p}} \tag{27}$$

$$\begin{aligned}\Delta\ddot{\mathbf{V}}_{\mathrm{p},i} = &\ddot{\mathbf{V}}_{\mathrm{p}} - \mathbf{Ad}_{\mathbf{C}_{\mathrm{p},i}}\ddot{\mathbf{V}}_i \\ &+ \mathbf{ad}_{\Delta\dot{\mathbf{V}}_{\mathrm{p},i}}\left(\mathbf{V}_{\mathrm{p}} - \Delta\mathbf{V}_{\mathrm{p},i}\right) - \mathbf{ad}^2_{\Delta\mathbf{V}_{\mathrm{p},i}}\mathbf{V}_{\mathrm{p}}.\end{aligned} \tag{28}$$

Proof 1: The relation $\dot{\mathbf{Ad}}_{\mathbf{C}_{\mathrm{p},i}} = -\mathbf{ad}_{\Delta\mathbf{V}_{\mathrm{p},i}}\mathbf{Ad}_{\mathbf{C}_{\mathrm{p},i}}$ [27] applied to $\mathbf{J}_{\mathrm{p}(l),i} = \mathbf{Ad}_{\mathbf{C}_{\mathrm{p},i}}\mathbf{X}_i$, with constant $\mathbf{X}_i$, yields (23). Repeated application yields (23) and (24). With (26) follows $\Delta\dot{\mathbf{V}}_{\mathrm{p},i} = \dot{\mathbf{V}}_{\mathrm{p}} - \dot{\mathbf{Ad}}_{\mathbf{C}_{\mathrm{p},i}}\mathbf{V}_i - \mathbf{Ad}_{\mathbf{C}_{\mathrm{p},i}}\dot{\mathbf{V}}_i = \dot{\mathbf{V}}_{\mathrm{p}} - \mathbf{Ad}_{\mathbf{C}_{\mathrm{p},i}}\dot{\mathbf{V}}_i - \mathbf{ad}_{\mathbf{V}_{\mathrm{p}}}\left(\mathbf{V}_{\mathrm{p}} - \mathbf{Ad}_{\mathbf{C}_{\mathrm{p},i}}\mathbf{V}_i\right)$ and hence (27). The second derivative $\Delta\ddot{\mathbf{V}}_{\mathrm{p},i} = \ddot{\mathbf{V}}_{\mathrm{p}} - \mathbf{Ad}_{\mathbf{C}_{\mathrm{p},i}}\ddot{\mathbf{V}}_i - \mathbf{ad}_{\Delta\mathbf{V}_{\mathrm{p},i}}\mathbf{Ad}_{\mathbf{C}_{\mathrm{p},i}}\dot{\mathbf{V}}_i + \mathbf{ad}_{\Delta\dot{\mathbf{V}}_{\mathrm{p},i}}\mathbf{V}_{\mathrm{p}} + \mathbf{ad}_{\Delta\mathbf{V}_{\mathrm{p},i}}\dot{\mathbf{V}}_{\mathrm{p}}$, can be reformulated to (28).

LIST OF MAIN SYMBOLS

- $\boldsymbol{\vartheta} \in \mathbb{R}^N$ - joint coordinate vector of PKM mechanism
- $\boldsymbol{\vartheta}_\mathrm{a} \in \mathbb{R}^{n_\mathrm{a}}$ - joint variable vector of SEA-actuated joints
- $\mathbf{q}_\mathrm{m} \in \mathbb{R}^{n_\mathrm{a}}$ - vector of joint variables of SEA-drive units
- $\mathbf{u} \in \mathbb{R}^{n_\mathrm{a}}$ - drive torques/forces at SEA-actuated joints
- $\bar{\mathbf{Q}}_{(l)} \in \mathbb{R}^{n_l}$ - vector of generalized forces of limb l
- $\mathbf{J}_\mathrm{IK}$ - $n_\mathrm{a} \times \delta_\mathrm{p}$ inverse kinematics Jacobian PKM
- $\mathbf{F}_{(l)}$ - $n_l \times \delta_\mathrm{p}$ inverse kinematics Jacobian of limb l
- $\mathbf{C}_i \in SE(3)$ - absolute configuration of body i, i.e. relative to the inertial frame (IFR)
- $\mathbf{B}_i \in SE(3)$ - reference configuration of body i relative to its predecessor
- $\mathbf{X}_i \in \mathbb{R}^6$ - screw coordinate vector (ray coordinates) of joint i represented in the reference frame on body i
- $\mathbf{V}_i \in \mathbb{R}^6$ - twist coordinate vector (ray coordinates) of body i represented in body-frame i
- $\mathbf{V}_\mathrm{t} \in \mathbb{R}^{\delta_\mathrm{p}}$ - task space velocity vector expressed in platform frame
- $\mathbf{W}_i \in \mathbb{R}^6$ - wrench coordinate vector (axis coordinates) represented in body-frame i
- $\mathbf{Ad}_{\mathbf{C}_{i,j}}$ - 6×6 twist transformation matrix, body j to i
- $\mathbf{M}_i$ - 6×6 body-fixed mass matrix of body i

n_a - # actuated joints, N - total # joint variables, N_l - # joint variables of limb l when connected to platform, n_l - # tree-joint variables of limb l (when disconnected from platform). δ_p - DOF of PKM platform. δ - DOF of mechanism.

REFERENCES

[1] J. Pratt, B. Krupp, and C. Morse, "Series elastic actuators for high fidelity force control," *Industrial Robot: An International Journal*, vol. 29, no. 3, pp. 234–241, 2002.

[2] T. Worsnopp, M. Peshkin, K. Lynch, and J. E. Colgate, "Controlling the apparent inertia of passive human-interactive robots," 2006.

[3] T. P. Huck, N. Münch, L. Hornung, C. Ledermann, and C. Wurll, "Risk assessment tools for industrial human-robot collaboration: Novel approaches and practical needs," *Safety Science*, vol. 141, 2021.

[4] R. J. Kirschner, J. Jantalia, N. Mansfeld, S. Abdolshah, and S. Haddadin, "Csm: Contact sensitivity maps for benchmarking robot collision handling systems," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3590–3596.

[5] F. Sergi, M. M. Lee, and M. K. O'Malley, "Design of a series elastic actuator for a compliant parallel wrist rehabilitation robot," in *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2013, pp. 1–6.

[6] A. Erdogan, B. Celebi, A. C. Satici, and V. Patoglu, "Assist on-ankle: a reconfigurable ankle exoskeleton with series-elastic actuation," *Autonomous Robots*, vol. 41, no. 3, pp. 743–758, 2017.

[7] H. Lee and S. Oh, "Series elastic actuators-driven parallel robot with wide-range impedance realization for balance assessment and training," *IEEE/ASME Transactions on Mechatronics*, 2022.

[8] A. De Luca, "Decoupling and feedback linearization of robots with mixed rigid/elastic joints," *Int. Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 8, no. 11, pp. 965–977, 1998.

[9] H. Gattringer, B. Oberhuber, J. Mayr, and H. Bremer, "Recursive methods in control of flexible joint manipulators," *Multibody system dynamics*, vol. 32, no. 1, pp. 117–131, 2014.

[10] G. Palli, C. Melchiorri, and A. De Luca, "On the feedback linearization of robots with variable joint stiffness," in *2008 IEEE international conference on robotics and automation*. IEEE, 2008, pp. 1753–1759.

[11] G. Buondonno and A. De Luca, "Efficient computation of inverse dynamics and feedback linearization for VSA-based robots," *IEEE robotics and automation letters*, vol. 1, no. 2, pp. 908–915, 2016.

[12] C. G. L. Bianco, "Evaluation of generalized force derivatives by means of a recursive Newton–Euler approach," *IEEE transactions on robotics*, vol. 25, no. 4, pp. 954–959, 2009.

[13] A. Müller, "Recursive second-order inverse dynamics for serial manipulators," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2483–2489.

[14] ——, "An O(n)-Algorithm for the Higher-Order Kinematics and Inverse Dynamics of Serial Manipulators Using Spatial Representation of Twists," *IEEE Rob. and Aut. Let.*, vol. 6, no. 2, pp. 397–404, 2020.

[15] F. Park, J. Bobrow, and S. Ploen, "A Lie Group Formulation of Robot Dynamics," *The International Journal of Robotics Research*, vol. 14, no. 6, pp. 609–618, 1995.

[16] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.

[17] G. Rodriguez, A. Jain, and K. Kreutz-Delgado, "A spatial operator algebra for manipulator modeling and control," *The International Journal of Robotics Research*, vol. 10, no. 4, pp. 371–381, 1991.

[18] A. Jain, *Robot and Multibody Dynamics*. Springer US, 2011.

[19] A. Müller and S. Kumar, "Closed-form time derivatives of the equations of motion of rigid body systems," *Multibody System Dynamics*, vol. 53, no. 3, pp. 257–273, 2021.

[20] J. Angeles, *Fundamentals of robotic mechanical systems*, 3rd ed. Springer, 2007.

[21] S. Briot and W. Khalil, *Dynamics of Parallel Robots*. Springer, 2015.

[22] S. K. Saha and W. O. Schiehlen, "Recursive kinematics and dynamics for parallel structured closed-loop multibody systems," *Mechanics of Structures and Machines*, vol. 29, no. 2, pp. 143–175, 2001.

[23] A. Müller, "Dynamics modeling of topologically simple parallel manipulators: A geometric approach," *ASME Applied Mechanics Review*, vol. 72(3), p. 27 pages, 2020.

[24] ——, "Dynamics of parallel manipulators with hybrid complex limbs —Modular modeling and parallel computing," *Mechanism and Machine Theory*, vol. 167, p. 104549 (36 pages), 2022.

[25] A. Müller, "A constraint embedding approach for dynamics modeling of parallel kinematic manipulators with hybrid limbs," *Robotics and Autonomous Systems*, vol. 155, p. 104187, 2022.

[26] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[27] A. Müller, "Screw and Lie group theory in multibody kinematics," *Multibody System Dynamics*, vol. 43, no. 1, pp. 37–70, 2018.

[28] ——, "Screw and Lie group theory in multibody dynamics," *Multibody System Dynamics*, vol. 42, no. 2, pp. 219–248, 2018.

[29] P. E. Nikravesh, *Computer-aided analysis of mechanical systems*. Prentice Hall, 1988.

[30] J. Wittenburg, *Dynamics of Multibody Systems*. Springer, 2008.

[31] J. Merlet, *Parallel Robots*. Springer, 2006.

[32] S. Kumar, H. Wöhrle, J. de Gea Fernández, A. Müller, and F. Kirchner, "A survey on modularity and distributivity in series-parallel hybrid robots," *Mechatronics*, vol. 68, p. 102367, 2020.

[33] A. Jain, "Unified formulation of dynamics for serial rigid multibody systems," *Journal of Guidance, Control and Dynamics*, vol. 14, pp. 531–542, 1991.

[34] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer, 2008.

[35] J. Angeles and S. Lee, "The formulation of dynamical equations of holonomic mechanical systems using a natural orthogonal complement," *ASME J. Appl. Mech.*, vol. 9, no. 5, pp. 243–244, 1988.

[36] A. Müller, "A Modular Geometric Approach to Dynamics Modeling of Fully-Parallel PKM by Example of a Planar 3RPR Mechanism," in *8th Europ. Conf. Mech. Sci. (EUCOMES)*, 2020, pp. 289–296.

[37] S. Kumar, H. Wöhrle, M. Trampler, M. Simnofske, H. Peters, M. Mallwitz, E. A. Kirchner, and F. Kirchner, "Modular Design and Decentralized Control of the Recupera Exoskeleton for Stroke Rehabilitation," *Applied Sciences*, vol. 9, no. 4, 2019.

[38] T. Kordik, C. Zauner, T. Marauli, H. Gattringer, and A. Müller, "Time-optimal trajectory planning of a redundantly actuated planar parallel robot driven with series elastic actuators," in *GAMM Annual Meeting, Proc. in Applied Mathematics and Mechanics*. Wiley, in press, 2023.

[39] M. Boukheddimi, S. Kumar, H. Peters, D. Mronga, R. Budhiraja, and F. Kirchner, "Introducing RH5 Manus: A Powerful Humanoid Upper Body Design for Dynamic Movements," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 01–07.

[40] N. Radford, "Valkyrie: Nasa's first bipedal humanoid robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, 2015.

[41] S. Kumar, K. A. v. Szadkowski, A. Mueller, and F. Kirchner, "An Analytical and Modular Software Workbench for Solving Kinematics and Dynamics of Series-Parallel Hybrid Robots," *Journal of Mechanisms and Robotics*, vol. 12, no. 2, 02 2020, 021114.

[42] J. M. Selig, *Geometric fundamentals of robotics*. Springer, 2004.