Integrating Mimic Joints into Dynamics Algorithms – Exemplified by the Hybrid Recupera Exoskeleton

Shivesh Kumar
Robotics Innovation Center,
DFKI GmbH
Robert–Hooke Straße 1
Bremen 28359, Germany
shivesh.kumar@dfki.de

Marc Simnofske
Robotics Innovation Center,
DFKI GmbH
Robert–Hooke Straße 1
Bremen 28359, Germany
marc.simnofske@dfki.de

Bertold Bongardt
Robotics Innovation Center,
DFKI GmbH
Robert–Hooke Straße 1
Bremen 28359, Germany
bertold.bongardt@dfki.de

Andreas Müller
Institute of Robotics,
Johannes Kepler University
Altenbergerstraße 69
Linz 4040, Austria
a.mueller@jku.at

Frank Kirchner
RIC, DFKI GmbH /
AG Robotik, Universität Bremen
Robert–Hooke Straße 1
Bremen 28359, Germany
frank.kirchner@dfki.de

ABSTRACT

The design of various robots in industrial and academic contexts integrates closed loops to improve the mechanical stiffness in comparison with purely serial or tree-type topologies. In particular, planar kinematic loops as parallelograms or double parallelograms are employed in such hybrid robots. Since these systems are geometrically overconstrained in the group of spatial Euclidean motions, the computational performance and numerical accuracy of any model-based dynamics software is negatively affected. This paper introduces a novel method to avoid these numerical issues for any hybrid system with loops that can be characterized by the concept of linear mimic joints: these are passive joints which depend on an active joint in a closed loop in a linear manner. With the proposed approach, the loop closure functions are automatically composed from the robot description file and integrated into the analytical equations for solving the forward and the inverse dynamics problems. The paper illustrates the application of this method for a novel shoulder mechanism containing a planar six bar mechanism that has been designed for the Recupera whole-body exoskeleton.

KEYWORDS

Hybrid robots, dynamic modeling, mimic joints, exoskeletons.

ACM Reference format:
Shivesh Kumar, Marc Simnofske, Bertold Bongardt, Andreas Müller, and Frank Kirchner. 2017. Integrating Mimic Joints into Dynamics Algorithms – Exemplified by the Hybrid Recupera Exoskeleton. In *Proceedings of Advances in Robotics, New Delhi, India, 28 June - 02 July 2017 (AIR'17),* 6 pages.
DOI: 10.475/123_4

AIR'17, New Delhi, India

DOI: 10.475/123_4

1 INTRODUCTION

Serial and tree-type mechanisms are well known for their versatility in applications, large workspace, and a simple modeling and control. Hence, they often represent the state-of-the-art in robotic systems. However, they generally feature only limited precision, low stiffness, and poor dynamic characteristics. In contrast to serial robots, parallel devices offer higher stiffness, speed, accuracy, and payload capacity. At the downside, they provide a reduced workspace and a more complex geometry which requires careful analysis and control. To combine the various advantages of serial and parallel topologies, hybrid serial-parallel robots have been developed in industry and academia. For instance, the stiffness of a manipulator can be significantly improved by including a simple parallelogram mechanism. In particular, industrial robots as ABB's IRB4400, IRB6660, KUKA's KR 30,50-PA., KR 700-PA robots, and Comau's Smart H, NJ, NX series, SR400 utilize this design concept [7, 23].

The usage of parallel submechanisms in a robot's design introduces a new level of complexity into its modeling, simulation, and control. For describing serial robots, Denavit–Hartenberg (DH) parameters [8], and their modifications [10], have become the de-facto standard: they specify each coordinate transformation by only four parameters instead of six parameters, due to the particular placement of local coordinate systems at specific locations. Since the placement of these coordinate systems requires manual effort, work has been invested to extract the DH parameters automatically from CAD models of the serial manipulators [20].

In case of tree type robots and robots with closed loops, the traditional notion of DH parameters can not be used and hence various extensions have been proposed in the literature [11]. A comparison of various robot parameterization techniques can, for example, be found in [2]. Due to the dependence of the frame placement on the link geometries, the modeling becomes unintuitive in particular for complex link shapes (for example in exoskeletons or human-machine interfaces). For these reasons, standard open source robot description formats, as `URDF` (`ROS`), `COLLADA` (`OpenRAVE`), or `SDF` (`Gazebo`), do not rely on DH parameters (or extensions) for

Type	position	velocity	acceleration
implicit:	$\phi(\boldsymbol{q}) = \boldsymbol{0}$	$\boldsymbol{K}\dot{\boldsymbol{q}} = \boldsymbol{0}$	$\boldsymbol{K}\ddot{\boldsymbol{q}} = \boldsymbol{k}$
explicit:	$\boldsymbol{q} = \gamma(\boldsymbol{y})$	$\dot{\boldsymbol{q}} = \boldsymbol{G}\dot{\boldsymbol{y}}$	$\ddot{\boldsymbol{q}} = \boldsymbol{G}\ddot{\boldsymbol{y}} + \boldsymbol{g}$

Table 1: Loop constraints [5]

representing the coordinate transforms and, instead, store the required transformations by six parameters. These displacements requested by these description formats can be automatically extracted from computer aided design (CAD) environments by programs as `CAD-2-SIM` [1] with little manual effort: due to the automation, the workflow becomes reusable, less prone to erratic human influence, and quickly adaptable to new design iterations.

Multi-body dynamics has been an area of extensive research during the past decades. Notable works include Newton–Euler's [5, 10], the Lagrangian [10], the Decoupled Natural Orthogonal (DeNOC) [21], and Kane's method [3, 14]. Traditionally, the equations of motion were described in 3D Euclidean space – which quickly yields a large amount of equations for systems of connected bodies [15]. To address this issue, alternate elegant formulations have been developed based on screw theory which can be simply transformed into program code for present day computers [5, 9]. The Rigid Body Dynamics Library (`RBDL`) [6] is an open source library that implements such algorithms for generic tree-type systems and imports robot descriptions using the `URDF` [24] format.

The presence of closed loops in robots significantly increases the complexity of the kinematics and dynamics problems associated to the multi-body systems (MBS). In particular, planar kinematic loops impose redundant constraints on the system – that either need to be removed manually [5] or demand numerical decomposition techniques which deteriorate the computational performance and numerical accuracy of the solution [17]. In many practical instances of kinematic loops within hybrid robots, the active joint guides the other passive joints inside a loop in a linear manner. The contribution of this paper is an explicit method to compute the forward and inverse dynamics of arbitrary hybrid robots that involve planar kinematic loops which can be described by means of linear mimic joints (for example, parallelograms or chains of parallelograms). Once specified in a `URDF` file, featuring the mimic joint tag, the associated loop closure functions are detected and transferred into a suitable form to be used in the computations of the analytical the forward and inverse dynamics algorithms. The benefit of this approach is that efficient dynamics algorithms for tree type system with $O(n)$ complexity can be directly used to solve the dynamics. The results are free of numerical errors due to loop closure and free of singularities arising from redundant constraints imposed by the planar kinematic loop. The application of the dynamic modeling is demonstrated for a novel shoulder mechanism which employs a double-parallelogram mechanism at the second shoulder joint.

The organization of the paper is the following: Section 2 provides theoretical preliminaries for modeling robots with closed loops along with an introduction to the concept of loop closure functions. It derives the formulas for the forward and inverse dynamics for these mechanisms. Section 3 presents the concept of mimic joints and the novel method to derive the loop closure functions automatically from the robot description. Section 4 presents the application of this approach to the forward and inverse dynamics computation of a shoulder mechanism developed for a novel, full-body exoskeleton. Section 5 draws the conclusions and presents future work.

2 MODELING RIGID BODY SYSTEMS WITH CLOSED LOOPS

This section briefly introduces the theory of multi-body dynamics subjected to holonomic and sceleronomic constraints. It strictly adopts the notation and terminology introduced by Featherstone in [5]. Therefore, consider a rigid body system with N_B bodies, N_J joints, and $N_L = N_J - N_B$ kinematic loops. Assume that a spanning tree is defined and that the joints are enumerated using regular numbering scheme. Let n denote the degree of freedom of the selected spanning tree, computed as $n = \sum_{i=1}^{N_B} n_i$, and let n_c denote the number of loop-closure constraints, computed as $n_c = \sum_{k=N_B+1}^{N_J} n_k^c$. Further, let $\boldsymbol{q}$ indicate the vector of all joints of the spanning tree (of size n) and let $\boldsymbol{y}$ indicate the vector of all independent joints of the spanning tree (of size $n - n_c$).

2.1 Loop Constraints

Loop constraints are algebraic constraints on the motion variables of a multi-body system. Loop constraints can be expressed in an implicit and in an explicit way, they are summarized in Table 1 at position, velocity, and acceleration levels. Here let $\boldsymbol{K} = \frac{\partial \phi}{\partial \boldsymbol{q}}$, $\boldsymbol{k} = -\dot{\boldsymbol{K}}\dot{\boldsymbol{q}}$, $\boldsymbol{G} = \frac{\partial \gamma}{\partial \boldsymbol{y}}$, and $\boldsymbol{g} = \dot{\boldsymbol{G}}\dot{\boldsymbol{y}}$. If both functions ϕ and γ describe the same constraint, $\phi \circ \gamma = \boldsymbol{0}$, $\boldsymbol{KG} = \boldsymbol{0}$, and $\boldsymbol{Kg} = \boldsymbol{k}$ can be deduced. Algorithms to compute variables in Table 1 from the spanning tree are provided in [5] and skipped here for brevity.

2.2 Equations of Motion (EOM)

The equations of motion for the spanning tree of a multi-body system can be written as

$$\boldsymbol{\tau} = \boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \ddot{\boldsymbol{q}}) \tag{1}$$

where $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ are $(n \times 1)$ vectors of joint position, velocity and acceleration variables of the spanning tree, $\boldsymbol{H}(\boldsymbol{q})$ is the $(n \times n)$ mass-inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \ddot{\boldsymbol{q}})$ is a $(n \times 1)$ vector for Coriolis-centrifugal and gravity efforts, and $\boldsymbol{\tau}$ is the $(n \times 1)$ vector of force/torque variables. In case of robots with closed loops, the equivalent spanning tree of the robot system is subjected to loop constraint forces

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \ddot{\boldsymbol{q}}) = \boldsymbol{\tau} + \boldsymbol{\tau}_a + \boldsymbol{\tau}_c \tag{2}$$

where $\boldsymbol{\tau}_c$ and $\boldsymbol{\tau}_a$ are the constraint and active forces, respectively produced by the loop joints. If the selected loop joint is passive, $\boldsymbol{\tau}_a = 0$ can be substituted in Equation 2. The constraint force $\boldsymbol{\tau}_c$ is usually unknown but its value can either be calculated or eliminated from the equation following the Jourdain's principle [19] of virtual power, i.e., $\boldsymbol{\tau}_c \dot{\boldsymbol{q}} = 0$. Based on the (implicit or explicit) nature of the loop constraints, the equations of motion are developed for the entire system.

2.2.1 EOM with implicit loop constraints. The loop joints impose a set of kinematic constraints on the spanning tree which are briefly introduced in Table 1. Assuming that the position level implicit constraints have been successfully differentiated two times, the

acceleration level loop constraints can be collected in a single matrix equation of the form

$$\boldsymbol{K}\ddot{\boldsymbol{q}} = \boldsymbol{k} \tag{3}$$

where $\boldsymbol{K}$ is $(n_c \times n)$ matrix. If the system is subjected to an implicit loop constraint then it can be shown that $\boldsymbol{\tau}_c$ takes the form

$$\boldsymbol{\tau}_c = \boldsymbol{K}^T\boldsymbol{\lambda} \tag{4}$$

where $\boldsymbol{\lambda}$ is vector of unknown forces, also regarded as Lagrangian multipliers. Combining Equations 2, 3 and 4, the equation of motion of the overall system taking into account implicit loop constraints can be written as

$$\begin{bmatrix} \boldsymbol{H} & \boldsymbol{K}^T \\ \boldsymbol{K} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau} - \boldsymbol{C} + \boldsymbol{\tau}_a \\ \boldsymbol{k} \end{bmatrix} . \tag{5}$$

This is a system of $(n+n_c)$ equations in $(n+n_c)$ unknowns. The coefficient matrix of dimension $(n+n_c)\times(n+n_c)$ in Equation 5 is symmetric but not positive definite. If the rank of matrix $r = rank(\boldsymbol{K})$ is less than n_c, then coefficient matrix becomes singular and the system is said to be over-constrained. Over-constrained systems are actually very common. For example, any system containing planar kinematic loop is over-constrained. Redundant constraints usually affect the computational efficiency and accuracy of any model based dynamics software [17].

2.2.2 EOM with explicit loop constraints. Using explicit velocity level loop constraint (refer to Table 1) and Jourdain's principle of virtual power, one can establish that $\boldsymbol{\tau}_c$ will have the following property

$$G^T\boldsymbol{\tau}_c = 0 . \tag{6}$$

Similarly, one can write the explicit motion constraints at an acceleration level

$$\ddot{\boldsymbol{q}} = G\ddot{\boldsymbol{y}} + \boldsymbol{g} . \tag{7}$$

Combining Equations 2, 6 and 7, the equation of motion taking into account explicit loop constraints can be developed.

$$\begin{bmatrix} \boldsymbol{H} & -\boldsymbol{1} & \boldsymbol{0} \\ -\boldsymbol{1} & \boldsymbol{0} & G \\ \boldsymbol{0} & G^T & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{\tau}_c \\ \ddot{\boldsymbol{y}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau} - \boldsymbol{C} + \boldsymbol{\tau}_a \\ -\boldsymbol{g} \\ \boldsymbol{0} \end{bmatrix} \tag{8}$$

2.3 Loop Closure Functions

It is usually more complex to deal with robots involving closed loops. In contrast to a tree type robot, the mobility of closed loop system is dependent on r which can vary with the configuration. Also, the different assembly modes can lead to configuration ambiguities. Thus, it is useful to derive explicit functions for the modeling of closed loop systems whenever and wherever possible.

Let us define loop closure functions such that they provide a unique mapping between the independent position variables $\boldsymbol{y}$ and position variables in the spanning tree $\boldsymbol{q}$. For this assumption to hold true, let us define a set $C \subseteq \mathbb{R}^{n-r}$ of acceptable values of $\boldsymbol{y}$, and assume $\boldsymbol{y} \in C$. For all $\boldsymbol{y} \in C$, there exists a function, γ such that

$$\begin{aligned} \boldsymbol{q} &= \gamma(\boldsymbol{y}) \\ \dot{\boldsymbol{q}} &= G\dot{\boldsymbol{y}} \\ \ddot{\boldsymbol{q}} &= G\ddot{\boldsymbol{y}} + \dot{G}\dot{\boldsymbol{y}} = G\ddot{\boldsymbol{y}} + \boldsymbol{g} \end{aligned} \tag{9}$$

It must be noted that the above relations are identical to the explicit loop constraint equations noted in Table 1 when the loop closure errors are zero. This method is less generic in nature than the ones described before because it is not always possible to find such a mapping analytically. However, the advantages outweighs the manual effort needed to derive these functions because when loop closure functions are used numerical loop closure errors can not occur. Also there is no need to introduce a constraint stabilization term unlike when practically dealing with Equation 3.

2.4 Forward and Inverse Dynamics

The equations of motion presented above could be either solved for independent joint accelerations $\ddot{\boldsymbol{y}}$ under given actuator force conditions or for the actuator forces $\boldsymbol{u}$ required to generate given acceleration. The former is called the forward dynamics problem and the latter is called the inverse dynamics problem [5, 9]. In this section, the forward and inverse dynamics formula are derived such that a link between the loop closure functions and spanning tree dynamics is established.

2.4.1 Forward dynamics solution. By using Equation 6 and multiplying by G^T on both sides of Equation 2, the loop constraint forces $\boldsymbol{\tau}_c$ can be eliminated.

$$G^T\boldsymbol{H}\ddot{\boldsymbol{q}} = G^T(\boldsymbol{\tau} - \boldsymbol{C} + \boldsymbol{\tau}_a) \tag{10}$$

Substituting Equation 7 in Equation 10 and simplifying one can arrive at the solution to the forward dynamics problem:

$$\ddot{\boldsymbol{y}} = (G^T\boldsymbol{H}G)^{-1}G^T(\boldsymbol{\tau} - \boldsymbol{C} + \boldsymbol{\tau}_a - \boldsymbol{H}\boldsymbol{g}) . \tag{11}$$

2.4.2 Inverse dynamics solution. Equation 10 could be rewritten as:

$$G^T\boldsymbol{\tau} = G^T(\boldsymbol{H}\ddot{\boldsymbol{q}} + \boldsymbol{C} - \boldsymbol{\tau}_a) = G^T\boldsymbol{\tau}_{ID} \tag{12}$$

where τ_{ID} is inverse dynamics output of a spanning tree given by

$$\boldsymbol{\tau}_{ID} = ID(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = ID(\gamma(\boldsymbol{y}), G\dot{\boldsymbol{y}}, G\ddot{\boldsymbol{y}} + \boldsymbol{g}) . \tag{13}$$

The solution to Equation 12 is not unique because G^T is an $(n-r)\times n$ matrix which imposes $(n-r)$ constraints on an n dimensional vector of unknowns, leaving r freedoms of choice. In other words, there are ∞^r different values of $\boldsymbol{\tau}$ which will produce the same acceleration. To arrive at a unique solution, the actuated degrees of freedom must be separated from the passive degrees of freedom. This can be done with the help of a matrix G_u which basically contain the rows of G corresponding to the actuated degrees of freedom. If the rank of matrix G_u is equal to $(n-r)$, then the system is properly actuated and a unique solution to the inverse dynamics problem can be found which is given by:

$$\boldsymbol{u} = G_u^{-T}G^T\boldsymbol{\tau}_{ID} \tag{14}$$

where $\boldsymbol{u}$ is a vector of actuator forces required to produce the given acceleration $\ddot{\boldsymbol{y}}$.

3 MIMIC JOINTS AND CALCULATION OF LOOP CLOSURE FUNCTIONS

Mimic joints are passive joints that linearly mimic the motion of independent active joint in a kinematic loop. They can be formally

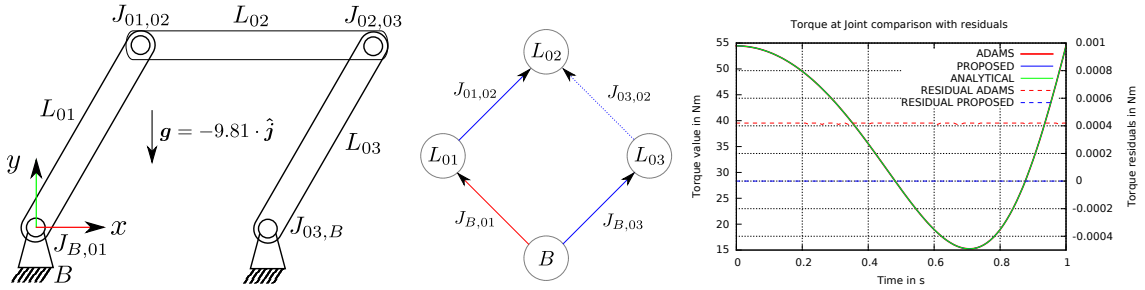

Figure 1: Parallelogram mechanism (left to right): (a) schematic, (b) topological graph, (c) comparison of actuator torque at $J_{B,01}$ between ADAMS, analytical solution and proposed method

described at position, velocity and acceleration levels by the following equation:

$$\begin{aligned} q_p &= mq_a + b \\ \dot{q}_p &= m\dot{q}_a \\ \ddot{q}_p &= m\ddot{q}_a \end{aligned} \tag{15}$$

where m is a constant multiplier term, b is a constant bias/offset term, $(q_p, \dot{q}_p, \ddot{q}_p)$ represent the position, velocity and acceleration of the passive joint and $(q_a, \dot{q}_a, \ddot{q}_a)$ represent the position, velocity and acceleration of the active joint that is being followed. These joints are defined in an input URDF[1] file with the help of mimic joint tags as shown below:

```
<joint name=PASSIVE_JNAME type=JTYPE >
    <parent link=PARENT />
    <child link=CHILD />
    <origin xyz=COORDS rpy=ANGLES />
    <axis xyz=DIRECTION />
    <mimic joint=ACTIVE_JNAME multiplier=m offset=b />
</joint>
```

3.1 Derivation of Loop Closure Functions

Let us choose a spanning tree G_t for a hybrid robot such that no loop joint has active torque ($\boldsymbol{\tau}_a = 0$) and the resulting system has n_a active joints. For this spanning tree, the relation between the tree joints and active joints can be collected in the following matrix equations:

$$\begin{aligned} \boldsymbol{q} &= \boldsymbol{M}\boldsymbol{y} + \boldsymbol{b} \\ \dot{\boldsymbol{q}} &= \boldsymbol{M}\dot{\boldsymbol{y}} \\ \ddot{\boldsymbol{q}} &= \boldsymbol{M}\ddot{\boldsymbol{y}} \end{aligned} \tag{16}$$

where $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ are $(n \times 1)$ vectors of position, velocity and acceleration of the tree joints, $(\boldsymbol{y}, \dot{\boldsymbol{y}}, \ddot{\boldsymbol{y}})$ are $(n_a \times 1)$ vectors of position, velocity and acceleration of the active joints in the tree, $\boldsymbol{M}$ is a constant $(n \times n_a)$ multiplier matrix (not to be confused with mass-inertia matrix, $\boldsymbol{H}(\boldsymbol{q})$), $\boldsymbol{b}$ is a constant $(n \times 1)$ bias/offset vector. Following are the guidelines for generating the multiplier matrix $\boldsymbol{M}$ and offset vector $\boldsymbol{b}$:

- Multiplier matrix $\boldsymbol{M}$ contains 1 in positions where tree joints are corresponding to the active joint and multiplier value m in positions corresponding to passive joints in the column corresponding to the active joint that is being followed. The rest of the positions are filled with zeros.
- Offset vector $\boldsymbol{b}$ contains 0 in positions corresponding to active joints and offset value b corresponding to passive joints.

Comparing Equation 16 with Equation 9, the following could be deduced: $\gamma(\boldsymbol{y}) = \boldsymbol{M}\boldsymbol{y} + \boldsymbol{b}$, $\boldsymbol{G} = \boldsymbol{M}$ and $\boldsymbol{g} = \boldsymbol{0}$. A matrix $\boldsymbol{M}_u$ can be defined by selecting rows of $\boldsymbol{M}$ corresponding to active degrees of freedom. This corresponds to the matrix $\boldsymbol{G}_u$ in Equation 14 for computing inverse dynamics.

3.1.1 Solution to forward dynamics. These expressions for loop closure functions can be substituted in Equation 11 to arrive at solution to forward dynamics problem

$$\ddot{\boldsymbol{y}} = (\boldsymbol{M}^T\boldsymbol{H}\boldsymbol{M})^{-1}\boldsymbol{M}^T(\boldsymbol{\tau} - \boldsymbol{C}) \,. \tag{17}$$

Equation 18 can also be expressed as a function of input actuator forces using Equation 14:

$$\ddot{\boldsymbol{y}} = (\boldsymbol{M}^T\boldsymbol{H}\boldsymbol{M})^{-1}\boldsymbol{M}^T(\boldsymbol{M}^{-T}\boldsymbol{M}_u^T\boldsymbol{u} - \boldsymbol{C}) \,. \tag{18}$$

While $\boldsymbol{u}$ is the input to Equation 18, the terms $\boldsymbol{H}, \boldsymbol{C}$ can be computed efficiently using RBDL [6] for any generic tree type system.

3.1.2 Solution to inverse dynamics. Similarly, substituting loop closure function expressions to Equation 14, the solution to inverse dynamics problem can be derived:

$$\boldsymbol{u} = \boldsymbol{M}_u^{-T}\boldsymbol{M}^T\boldsymbol{\tau}_{ID} \tag{19}$$

While $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ is the input to Equation 19, the term $\boldsymbol{\tau}_{ID}$ can be computed efficiently using RBDL [6] for any generic tree type system.

3.2 Parallelogram Example

To demonstrate the calculation of loop closure functions using mimic joints, let us take an example of a simple parallelogram mechanism (see Figure 1 (a)). The chosen spanning tree of this mechanism is shown in Figure 1 (b) where B is the fixed base, $J_{02,03}$ is the loop joint and $J_{B,01}$ is the active joint in the loop. For this spanning tree, the relation between the tree joints vector $(\boldsymbol{q})$ and active joints vector $(\boldsymbol{y})$ can be collected in the following equation:

$$\boldsymbol{q} = \begin{bmatrix} q_{B,01} \\ q_{01,02} \\ q_{B,03} \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} q_{B,01} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{20}$$

On comparison with position part of Equation 16, it could be immediately deduced that: $\boldsymbol{M} = [1, -1, 1]^T$. Matrix $\boldsymbol{M}_u$ can be calculated by selecting rows corresponding to active joints from $\boldsymbol{M}$ matrix

[1] http://wiki.ros.org/urdf/XML/joint

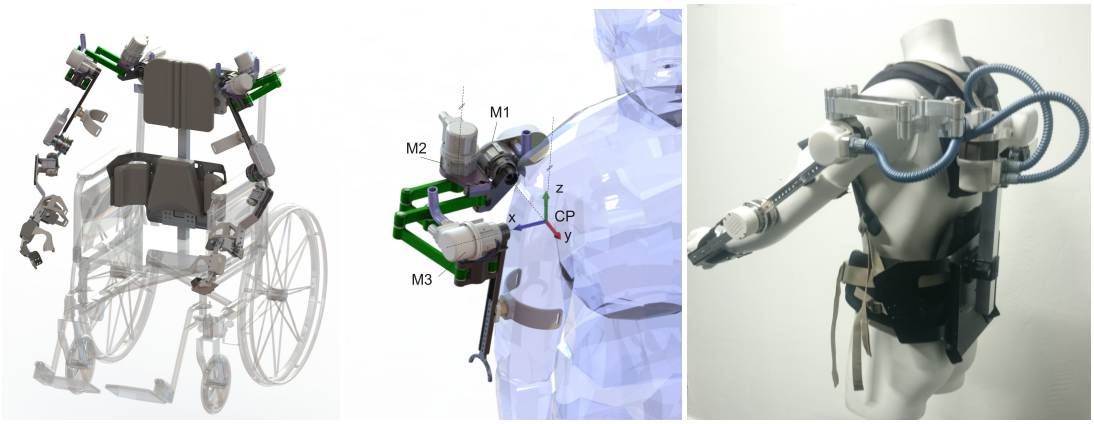

Figure 2: Recupera upper body system (left to right): (a) mounted on wheelchair, (b) shoulder mechanism, (c) built up prototype

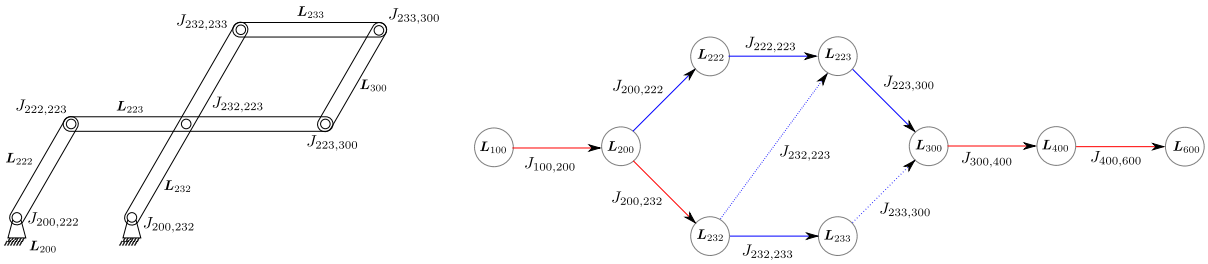

Figure 3: Recupera shoulder mechanism (left to right): (a) Six bar mechanism schematic, (b) topological graph of prototype

which in this case is simply equal to $[1]^T$. The expressions for $\boldsymbol{M}$ and $\boldsymbol{M}_u$ can now be substituted in Equation 19 to compute the actuator forces required to produce a given motion trajectory. An input motion trajectory $q_{B,01} = 2\pi t^2$ is provided from $t = 0$ to $t = 1$ second in 100 time steps. The length of the links is equal to 2 m and center of mass lies at the middle of the links. Masses of the three moving links are set to 1 kg, and the inertias are set to $4/3$ kg m^2. The comparison of torque required at $J_{B,01}$ to produce the given motion trajectory between ADAMS, analytical solution and proposed approach described in this section is shown in Figure 1 (c). It can be observed that the residual between the analytical solution and proposed method is exactly zero.

4 APPLICATION TO EXOSKELETON CONTROL

The concept of the Recupera-Reha full body exoskeleton (with 34 active degrees of freedom) that is being designed for upper body rehabilitation of stroke patients is introduced in [12]. The design is modular and upper body of the exoskeleton system can also be mounted on a wheel chair. In this section, the design and dynamic modeling of the novel shoulder mechanism from one of its arm is presented. As an outlook, the CAD design of the Recupera upper body exoskeleton system is also shown which is under physical construction presently (see Figure 2 (a)).

4.1 Shoulder Mechanism Design

The motivation of the design is to achieve a light weight fully actuated 3 degrees of freedom mechanism with a workspace similar to human shoulder. Also, the three joint axes should intersect at a central point (CP) in the human shoulder such that no two joint axes become coincident in the allowed motion range. This ensures a singularity free operation.

In contrast to purely serial shoulder designs [16, 18] where a collision of the motors with the patient is possible, the novel Recupera shoulder design, as shown in Figure 2 (b), hides the two motors behind the shoulder by employing a variant of Watt six bar guiding mechanism [4] which creates a virtual center of rotation in the shoulder. The closed loop mechanism (actuated by motor M2) has a limited workspace which is similar to the human shoulder. For the other two motors (M1 and M3), mechanical adjustable end stops are integrated to limit the range of motion for safety purposes. To realize system level modularity, the shoulder motors are identical with the ones used in the Active Ankle joint presented in [13, 22]. A first prototype of the shoulder is already built up and commissioned as shown in Figure 2 (c).

4.2 Dynamic Modeling

The variant of Watt six bar mechanism used in Recupera exoskeleton can also be interpreted as a double parallelogram mechanism

(see Figure 3 (a)) which can easily be modeled using the concept of linear mimic joints. The topological graph of the Recupera exoskeleton arm prototype (Figure 2 (c)) is shown in Figure 3 (b). The red edges ($J_{100,200}$, $J_{300,400}$, $J_{400,600}$, $J_{200,232}$) correspond to the active degrees of freedom of the system and their joint position values constitute the independent joint position vector $\boldsymbol{y}$. The tree joints (excluding the loop joints shown by dotted blue edges) are collected in tree joint position vector $\boldsymbol{q}$. The relation between $\boldsymbol{q}$ and $\boldsymbol{y}$ is shown in Equation 21 from which the multiplier matrix $\boldsymbol{M}$ and bias vector $\boldsymbol{b}$ can be deduced.

$$\boldsymbol{q} = \begin{bmatrix} q_{100,200} \\ q_{200,222} \\ q_{222,223} \\ q_{223,300} \\ q_{300,400} \\ q_{400,600} \\ q_{200,232} \\ q_{232,233} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} q_{100,200} \\ q_{300,400} \\ q_{400,600} \\ q_{200,232} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (21)$$

Matrix $\boldsymbol{M}_u$ can be calculated by selecting rows corresponding to active joints from $\boldsymbol{M}$ matrix which results in (4×4) identity matrix in this case. The expressions for $\boldsymbol{M}$ and $\boldsymbol{M}_u$ can be substituted in Equation 19 to solve for the inverse dynamics.

5 CONCLUSIONS

A novel method to determine the loop closure functions from the definitions of linear mimic joints in a robot description file has been presented that is suitable for inclusion in forward and inverse dynamics computations. The resulting formulae are derived such that efficient $O(n)$ dynamics algorithms for tree-type systems can still be utilized. Also, they are not affected by any loop constraints stabilization errors. The proposed approach is applicable to all hybrid robots with closed loops involving linear mimic joints – that are used in considerable number of practical robot designs for example with parallelogram or double parallelogram linkages. The design of a novel hybrid exoskeleton is presented and the proposed dynamic modeling approach is applied to it. Since the robot's `URDF` specification can be generated in an automated manner (using `CAD-2-SIM`), the proposed method closes the gap for obtaining dynamics solvers automatically for specific classes of mechanisms. The method has been implemented in a `RBDL`-based `C++` library, named as Hybrid Robot Dynamics (`HyRoDyn`). For the future, it is planned to embed the obtained dynamic model into higher-level control paradigms, as assist-as-needed control.

ACKNOWLEDGMENTS

The authors are thankful to Jan Frederik Bode for verifying the proposed approach on example mechanisms using `ADAMS`.

The work presented in this paper was performed within the projects Recupera-Reha and D-Rock, funded by the Federal Ministry of Education and Research (BMBF) under grant numbers 01-IM-14006A and 01-IW-15001 respectively. The fourth author acknowledges that this work has been partially supported by the Austrian COMET-K2 program of the Linz Center of Mechatronics.

REFERENCES

[1] Bertold Bongardt. 2011. *CAD-2-SIM – Kinematic Modeling of Mechanisms Based on the Sheth-Uicker Convention.* Springer Berlin Heidelberg, Berlin, Heidelberg, 465–477. DOI: http://dx.doi.org/10.1007/978-3-642-25486-4_47

[2] Bertold Bongardt. 2013. Sheth Uicker convention revisited. *Mechanism and Machine Theory* 69 (2013), 200 – 229. DOI: http://dx.doi.org/10.1016/j.mechmachtheory.2013.05.008

[3] Keith W. Buffinton. 2005. *Robotics and Automation Handbook.* CRC Press, Chapter Kane's Method in Robotics.

[4] Günter Dittrich and Volker Wehn. 1989. Öffnungsmechanismus einer PKW-Motorhaube : 602. *Der Konstrukteur: Magazin für Konstruktion und Entwicklung* (1989).

[5] Roy Featherstone. 2008. *Rigid Body Dynamics Algorithm.*

[6] Martin L. Felis. 2017. RBDL: an efficient rigid-body dynamics library using recursive algorithms. *Autonomous Robots* 41, 2 (2017), 495–511.

[7] M. Gautier, W. Khalil, and P. P. Restrepo. 1995. Identification of the dynamic parameters of a closed loop robot. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, Vol. 3. 3045–3050 vol.3. DOI: http://dx.doi.org/10.1109/ROBOT.1995.525717

[8] Denavit Hartenberg and Richard Scheunemann. 1955. A kinematic notation for lower-pair mechanisms based on matrices. *Trans ASME J. Appl. Mech.* (1955).

[9] Abhinandan Jain. 2011. *Robot and Multibody Dynamics: Analysis and Algorithms.* Springer Verlag.

[10] Wisama Khalil and Etienne Dombre. 2002. *Modeling, Identification and Control of Robots* (3rd ed.). Taylor & Francis, Inc., Bristol, PA, USA.

[11] W. Khalil and J. Kleinfinger. 1986. A new geometric notation for open and closed-loop robots. In *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, Vol. 3. 1174–1179. DOI: http://dx.doi.org/10.1109/ROBOT.1986.1087552

[12] Elsa Andrea Kirchner, Niels Will, Marc Simnofske, Luis Manuel Vaca Benitez, Bertold Bongardt, Mario Michael Krell, Shivesh Kumar, Martin Mallwitz, Anett Seeland, Marc Tabie, Hendrik Wöhrle, Mehmed Yüksel, Anke Heß, Rüdiger Buschfort, and Frank Kirchner. 2016. Recupera-Reha: Exoskeleton Technology with Integrated Biosignal Analysis for Sensorimotor Rehabilitation. In *Transdisziplinäre Konferenz SmartASSIST*. 504–517.

[13] Shivesh Kumar, Abhilash Nayak, Bertold Bongardt, Andreas Mueller, and Frank Kirchner. 2017. Kinematic analysis of Active Ankle using computational algebraic geometry. In *7th IFToMM International Workshop on Computational Kinematics (CK 2017)*. Springer.

[14] Martin Lesser. 1992. A Geometrical Interpretation of Kane's Equations. *Proceedings: Mathematical and Physical Sciences* 436, 1896 (1992), 69–87.

[15] Paul RC Luh JS, Walker MW. 1980. On-Line Computational Scheme for Mechanical Manipulators. *ASME. J. Dyn. Sys., Meas., Control.* (1980).

[16] Martin Mallwitz, Niels Will, Johannes Teiwes, and Elsa Andrea Kirchner. 2015. The CAPIO Active Upper Body Exoskeleton and its Application for Teleoperation. In *Proceedings of the 13th Symposium on Advanced Space Technologies in Robotics and Automation. ESA/Estec Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA-2015)*. ESA.

[17] Andreas Mueller. 2014. Implementation of a Geometric Constraint Regularization for Multibody System Models. *Arch. Mech. Eng.* (2014).

[18] D. Naidu, R. Stopforth, G. Bright, and S. Davrajh. 2011. A 7 DOF exoskeleton arm: Shoulder, elbow, wrist and hand mechanism for assistance to upper limb disabled individuals. In *AFRICON, 2011.* 1–6. DOI: http://dx.doi.org/10.1109/AFRCON.2011.6072065

[19] J. C. Piedboeuf. 1993. Kane's equations or Jourdain's principle?. In *Proceedings of 36th Midwest Symposium on Circuits and Systems.* 1471–1474 vol.2. DOI: http://dx.doi.org/10.1109/MWSCAS.1993.343389

[20] CG Rajeevlochana, SK Saha, and S Kumar. 2012. Automatic extraction of DH parameters of serial manipulators using line geometry. In *The 2nd International Conference on Multibody System Dynamics.*

[21] A.B. Koteswara Rao, S.K. Saha, and P.V.M. Rao. 2006. Dynamics Modelling of Hexaslides using the Decoupled Natural Orthogonal Complement Matrices. *Multibody System Dynamics* 15, 2 (2006), 159–180.

[22] Marc Simnofske, Shivesh Kumar, Bertold Bongardt, and Frank Kirchner. 2016. Active Ankle - an Almost-Spherical Parallel Mechanism. In *47th International Symposium on Robotics (ISR).*

[23] Minh To and Phil Webb. 2012. An Improved Kinematic Model for Calibration of Serial Robots Having Closed-chain Mechanisms. *Robotica* 30, 6 (Oct. 2012), 963–971. DOI: http://dx.doi.org/10.1017/S0263574711001184

[24] A. Yousuf, W. Lehman, M. A. Mustafa, and M. M. Hayder. 2015. Introducing Kinematics with Robot Operating System (ROS). In *ASEE Annual Conference & Exposition.*