Noname manuscript No.
(will be inserted by the editor)

Reduced-order Forward Dynamics of Multi-Closed-Loop Systems

Majid Koul · Suril V Shah · S K Saha* · M Manivannan

the date of receipt and acceptance should be inserted later

Abstract In this work, a reduced-order forward dynamics of multi-closed-loop systems is proposed by exploiting the associated inherent kinematic constraints at acceleration level. First, a closed-loop system is divided into an equivalent open architecture consisting of several serial and tree-type subsystems by introducing cuts at appropriate joints. The resulting cut joints are replaced by appropriate constraint forces also referred to as Lagrange multipliers. Next, for each subsystem, the governing equations of motion are derived in terms of the Lagrange multipliers which are based on the Newton-Euler formulation coupled with the concept of Decoupled Natural Orthogonal Complement (DeNOC) matrices, introduced elsewhere. In the proposed forward dynamics formulation, Lagrange multipliers are calculated sequentially at subsystem level and later treated as external forces to the resulting serial or tree-type systems of the original closed-loop system, for the recursive computation of joint accelerations. Note that such subsystem-level treatment allows one to use already existing algorithms for serial and tree-type systems. Hence, one can perform the dynamic analyses relatively quickly without re-writing the complete model of the closed-loop system at hand. The proposed methodology is in contrast to the conventional approaches, where the Lagrange multipli-

Majid Koul, S K Saha
Department of Mechanical Engineering
Indian Institute of Technology Delhi
New Delhi - 16, India
*Corresponding author: saha@mech.iitd.ac.in

Suril V Shah
International Institute of Information Technology
Hyderabad - 32, AP, India

M Manivannan
Department of Applied Mechanics
Indian Institute of Technology Madras
Chennai - 36, Tamil Nadu, India

ers are calculated together at system level or simultaneously along with the joint accelerations, both of which incur higher order computational complexities and thereby more number of arithmetic operations. Due to the smaller size of matrices involved in evaluating Lagrange multipliers in the proposed methodology, and the recursive computation of the joint accelerations, the overall numerical performances like computational efficiency, etc., are likely to improve. The proposed reduced-order forward dynamics formulation is illustrated with two multi-closed-loop systems, namely, a 7-bar carpet scrapping mechanism and a 3-RRR parallel manipulator.

Keywords Forward dynamics · Lagrange multipliers · Multi-closed-loop systems · Reduced order

1 Introduction

A mechanical system with closed kinematic loops has a circuit of links or bodies such that the closed circuit can be traced through the system from one body to another and returns to the original body without traversing any of the body more than once [1]. Such kinematic loops generally occur in various mechanical systems such as robots, vehicles, spacecrafts and other heavy-duty industrial and traditional (textile and agriculture) machines. In general, closed-loop systems possess relatively few degrees-of-freedom (DOF) compared to the number of connected bodies. Moreover, multi-closed-loop systems consist of more than one independent kinematic loops, which can usually be more than the DOF of a system. Figure 1 shows an example of one such multi-closed-loop system consisting of four independent kinematic loops (11 moving bodies indicated with #1, #2, etc.) and 3-DOF. Details of Fig. 1 will be explained in Sect. 2.

Forward dynamics, as one of the aspects of dynamic analyses, predicts the joint motions based on the knowledge of joint forces and torques. Forward dynamics of multibody systems in the engineering and research arena is mainly carried out for its benefits in design verification and analyses of dynamic behaviour, before the system is actually made. A good forward dynamics algorithm leads to the fast and numerically stable simulations.

In contrast to the open-loop systems, dynamic modelling of closed-loop systems is complex mainly due to the additional loop-closure kinematic constraints that need to be satisfied throughout the simulation. These kinematic constraints relate the dependency of generalized coordinates to each other. Also, in closed-loop systems, the accumulation of round-off errors due to finite precision arithmetic causes a significant growth in constraint violation which finally leads to unstable/ unrealistic simulations. Hence, some constraint stabilization techniques need to be incorporated that adds to the overall computational cost.

A general procedure for forward dynamics of closed-loop systems is referred to as augmented formulation. It requires the division of the system into an equivalent open architecture, leading to several serial and tree-type subsystems [2], [3], [4], [5]. The division is carried out by placing cuts at appropriate joints. These cuts are replaced by appropriate Lagrange multipliers representing constraint forces and moments. For the resulting subsystems, the equations of motion along with the Lagrange multipliers are then derived using the Newton-Euler formulation [5] or the energy-based Euler-Lagrange method [6]. The equations of motion are then augmented with the independent kinematic constraint equations at acceleration level resulting into a set of Differential Algebraic Equations (DAE), which are solved simultaneously to obtain the unknown Lagrange multipliers and joint accelerations [7]. Finally, to obtain the joint velocities and positions, one needs to integrate the joint accelerations numerically. Typically, the augmented formulation is considered to be elegant and simple, as it requires lesser pre-modelling effort, however it usually results into violations in the lower-order constraint equations, i.e., velocity and position constraints. This mainly is due to the round-off errors in finite precision arithmetic associated with the larger set of DAE's [8]. Various constraint stabilization/ elimination methods have been developed, e.g., [8], [9], [10], [11] which remove/ minimize most of the numerical inefficiencies or inaccuracies related to the integration of DAE sets.

In an another approach referred to as embedding technique [2], the dynamic equations of motion are formulated in terms of the system's independent joint variables (DOF). This approach leads to a system of equations in independent accelerations commonly referred to as Ordinary Differential Equation (ODE) formulation [12]. In this approach, the dependent joint variables are expressed as a function of independent joint variables requiring complex algebraic manipulations (greater pre-modelling effort) and is generally non-recursive [8]. This approach, however, avoids the calculations of the Lagrange multipliers and therefore does not suffer from constraint violation problem. Other popular approaches leading to ODE formulation, which also avoids the Lagrange multiplier estimation, are based on an orthogonal complement matrix whose columns span the null-space of the velocity constraint matrix [13]. Premultiplication of this matrix to the unconstrained equations of motion results into a minimal-order representation, which can be solved for its independent joint accelerations, followed by the numerical integrations to obtain the joint velocities and positions. Although this technique is simple to implement, an explicit choice of the orthogonal complement matrix for a given system is sometimes difficult, and is usually determined numerically.

For the formulations described above, the equations of motion at the subsystem level i.e., for open-loop systems, can also be derived using the techniques described in [2], [13], [14], [15], [16] in addition to the Newton-Euler and Euler-Lagrange formulations. One such technique is the use of the natural orthogonal complement (NOC) matrix resulting out of the velocity constraints

written in Cartesian coordinates level along with the unconstrained Newton-Euler equations of motion [14]. This procedure for deriving the equations of motion annihilates the non-working constraint forces and moments which are otherwise determined in Newton-Euler formulation [2], [5]. This procedure is also simple to implement as the NOC matrix is obtained naturally from the velocity constraint expressions of the moving bodies in terms of the joint rates without any complex numerical computations. In [16], this NOC matrix was decoupled, leading to Decoupled Natural Orthogonal Complement (DeNOC) matrices, to obtain a series of recursive dynamics algorithms for serial chain systems, which were not only efficient but also numerically stable.

In a typical DAE formulation discussed above, the Lagrange multipliers arising out of the cut-loop methodology are solved either implicitly or explicitly [8]. In case of implicit elimination, the constraint forces are solved concurrently with the joint accelerations, resulting into an $\boldsymbol{O}(n+m)^3$ computational complexity, where n is the total number of system's generalized coordinates and m is the number of independent constraints. This would mean that the simulation-turn-around time increases in cubic power with the increase of n and m. On the other hand, in explicit elimination, the constraint forces are solved first utilizing the constraint equations and later substituted in the equations of motion to find the joint accelerations [7], [9]. This approach benefits both in terms of reduction in the computational complexity of the forward dynamics algorithm and in the application of various constraint stabilization techniques, that are usually required in the forward dynamics of closed-loop systems.

Several works towards the reduction of computational complexity of forward dynamics algorithms have been carried out in [17], [18] and others. Formulations with coordinate partitioning were presented in [19], whereas recursive coordinate reduction method was proposed in [20], [21]. A semi-recursive formulation for both the open- and closed-loop systems was proposed in [22]. Other formulations providing an $\boldsymbol{O}(n+m)$ overall performance were proposed in [18], [23]. These were based on the ability to express the dependent kinematic quantities in terms of the independent ones. Note that all the above formulations treated the cut-open closed-loop system still as one and tried to manipulate the generalized coordinates and the Lagrange multipliers to achieve efficiency, even though the original system was broken down to several small open-loop subsystems. They failed to exploit the benefits of the small subsystems that resulted out of the cut-loop methodology. This paper will focus on this aspect.

Besides, as the closed-loop system is cut-open into several serial or tree-type subsystems, why not use the well-established methodologies and algorithms for the serial and tree-type systems, e.g., [5], [14], [16]? This, however, would require the treatment of Lagrange multipliers as external forces and moments. As a result, the requirement of finding the Lagrange multipliers explicitly was

felt. This can be done using the explicit technique of augmented formulation [2], where the joint accelerations are expressed in terms of the Lagrange multipliers and solved using the acceleration-level kinematic constraints. The novelty of this paper lies in following the latter formulation by reducing the problem dimensions, by exploiting the smaller subsystems which may be existing in the cut-open tree-type system of the original closed-loop system. Hence the phrase *reduced-order* is used in the title. Such formulation is expected to provide some benefits in terms of simulation time, physical interpretations, etc. The methodology presented here is more elegant and generic, and can be applied to systems with larger multi-closed-loops and many degrees-of-freedom.

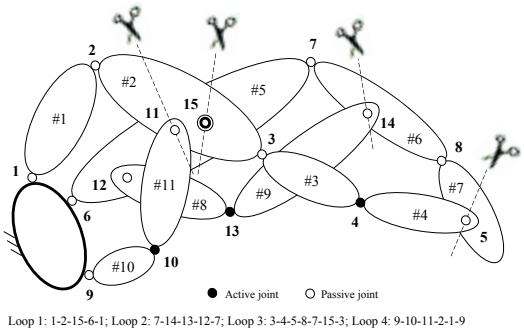
The paper is organized in the following manner: Section 2 describes the generalized procedure for deriving the equations of motion for a closed-loop system. Some of the existing procedures and the proposed methodology for the forward dynamics of the closed-loop systems are given in Sect. 3, followed by two numerical illustrations in Sect. 4. Section 5 presents the simulation results and discussion. Finally, conclusions are given in Sect. 6.

2 Equations of Motion for a Closed-loop System

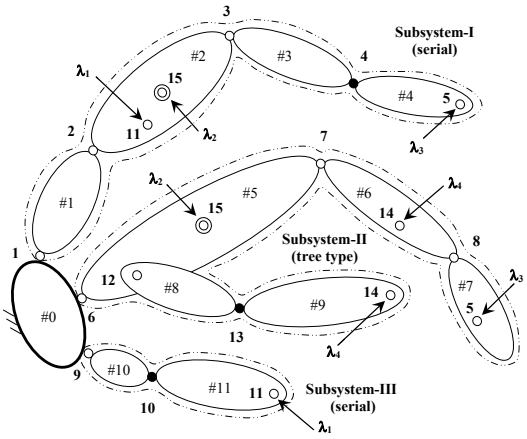
As per the general practice [2], [7], [24], [25], a closed-loop system is divided into an equivalent open-loop serial and/ or tree-type systems referred to as subsystems, by making cuts at appropriate joints and introducing Lagrange multipliers thereby. Figure 1(a) depicts a skeletal diagram of an excavator (a typical multi-closed-loop system) used for debris removal in heavy industries [26]. This 3-DOF system is actuated at joints 4, 10 and 13. The system has four closed-loops namely (i) Loop 1: 1-2-15-6-1, (ii) Loop 2: 7-14-13-12-7, (iii) Loop 3: 3-4-5-8-7-15-3 and (iv) Loop 4: 9-10-11-2-1-9. In order to convert such multi-loop system into an equivalent open-loop system, i.e., the spanning tree, the closed kinematic loops are cut at joints, as indicated in Fig. 1(a). The resulting open-loop architecture is shown in Fig. 1(b) that typically contains one or more serial and/or tree-type subsystems. In the next step, dynamic formulation of a general tree-type system is employed to these subsystems where the Lagrange multipliers are identified as external wrenches (moments and forces) to the equations of motion.

2.1 Equations of motion of a subsystem

Using the notations for tree-type systems proposed in [27], first the unconstrained Newton-Euler (NE) equations of motion of a cut-open subsystem are written. The independent generalized equations of motion for the subsystem are then derived by pre-multiplying the above NE equations with the transpose of DeNOC matrices associated to the subsystem [27].



(a) A closed-loop system



(b) Subsystems after cuts

Fig. 1 A multi-closed-loop system before and after introduction of cuts

For n coupled bodies in a fixed-base tree-type subsystem, the unconstrained equations of motion can be written as

$$\mathbf{M}\dot{\mathbf{t}} + \mathbf{\Omega}\mathbf{M}\mathbf{E}\mathbf{t} = \mathbf{w} \tag{1}$$

where the $6n$-dimensional vectors $\mathbf{t}$, $\dot{\mathbf{t}}$ and $\mathbf{w}$ are the generalized twist, twist-rate and wrench respectively, defined as $\mathbf{t} \equiv [\mathbf{t}_1^T, ..., \mathbf{t}_n^T]^T$, $\dot{\mathbf{t}} \equiv [\dot{\mathbf{t}}_1^T, ..., \dot{\mathbf{t}}_n^T]^T$ and $\mathbf{w} \equiv [\mathbf{w}_1^T, ..., \mathbf{w}_n^T]^T$. Moreover, the 6-dimensional twist, twist rate and the wrench associated with i^{th} link are defined as $\mathbf{t} \equiv [\boldsymbol{\omega}_i^T \ \mathbf{v}_i^T]^T$, $\dot{\mathbf{t}} \equiv [\dot{\boldsymbol{\omega}}_i^T \ \dot{\mathbf{v}}_i^T]^T$ and $\mathbf{w}_i \equiv [\mathbf{n}_i^T \ \mathbf{f}_i^T]^T$ — $\boldsymbol{\omega}_i$, $\mathbf{v}_i$, $\mathbf{n}_i$ and $\mathbf{f}_i$ being the angular velocity, linear velocity,

moment and force applied on the i^{th} link about its origin. The matrices $\mathbf{\Omega}$, $\mathbf{M}$ and $\mathbf{E}$ are the $6n \times 6n$ generalized matrices of angular velocity, extended mass and coupling term for the subsystem, respectively, which are defined as

$$\mathbf{M} \equiv diag[\mathbf{M}_1..\mathbf{M}_i..\mathbf{M}_n], \mathbf{\Omega} \equiv diag[\mathbf{\Omega}_1..\mathbf{\Omega}_i..\mathbf{\Omega}_n], \mathbf{E} \equiv diag[\mathbf{E}_1..\mathbf{E}_i..\mathbf{E}_n] \quad (2)$$

where, $\mathbf{\Omega}_i$, $\mathbf{M}_i$ and $\mathbf{E}_i$ are the 6×6 angular velocity, extended mass and coupling matrices for the i^{th} link respectively. They are given below

$$\mathbf{\Omega}_i \equiv \begin{bmatrix} \tilde{\boldsymbol{\omega}}_{\boldsymbol{i}} & \mathbf{O} \\ \mathbf{O} & \tilde{\boldsymbol{\omega}}_{\boldsymbol{i}} \end{bmatrix}, \ \mathbf{M}_i \equiv \begin{bmatrix} \mathbf{I}_i & m_i\tilde{\mathbf{d}}_i \\ -m_i\tilde{\mathbf{d}}_i & m_i\mathbf{1} \end{bmatrix}, \ \mathbf{E}_i \equiv \begin{bmatrix} \mathbf{1} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \quad (3)$$

In (3), $\tilde{\boldsymbol{\omega}}_{\boldsymbol{i}}$ and $\tilde{\mathbf{d}}_i$ are the 3×3 skew-symmetric matrices associated with the 3-dimensional vectors, $\boldsymbol{\omega}_i$ and $\mathbf{d}_i$, respectively, where $\mathbf{d}_i$ is the vector from origin of i^{th} link to its mass centre. Also $\mathbf{I}_i$ and m_i are the 3×3 inertia tensor about the origin of the i^{th} link and the mass of the i^{th} link, respectively. Note that the wrench $\mathbf{w}_i$, is composed of $\mathbf{w}_i^E$, the wrench due to externally applied moments and forces, $\mathbf{w}_i^C$, the wrench due to the constraint moments and forces at the uncut joints and $\mathbf{w}_i^\lambda$, the wrench due to Lagrange multipliers at the cut joints, i.e., $\mathbf{w}_i \equiv \mathbf{w}_i^E + \mathbf{w}_i^C + \mathbf{w}_i^\lambda$. Hence (1) can also be rewritten as

$$\mathbf{M\dot{t}} + \mathbf{\Omega MEt} = \mathbf{w}^E + \mathbf{w}^C + \mathbf{w}^\lambda \quad (4)$$

where, $\mathbf{w}^E$, $\mathbf{w}^C$ and $\mathbf{w}^\lambda$ denote the $6n$-dimensional vectors of the corresponding moments and forces associated with a subsystem. The generalized twist for an n-link subsystem is written in compact form as $\mathbf{t} \equiv \mathbf{N\dot{q}}$, where $\mathbf{N}$ and $\dot{\mathbf{q}}$ are the corresponding $6n \times n$ NOC matrix of the velocity constraints [28] and the vector of joint rates of the subsystem respectively. This NOC is an orthogonal complement matrix to the velocity constraint matrix, denoted with $\mathbf{D}$, that relates the twists of the interconnected links in the form of $\mathbf{Dt} = \mathbf{0}$, i.e., $\mathbf{DN} \equiv \mathbf{0}$. Since $\mathrm{w}^C = \mathbf{D}^T\boldsymbol{\lambda}^c$, where $\boldsymbol{\lambda}^c$ is the set of Lagrange multipliers representing reactions in the joints of the open-loop system, it can then be easily shown, $\mathbf{N}^T\mathbf{D}^T\boldsymbol{\lambda}^c = \mathbf{0}$ [28]. This result will be used next to eliminate constraint wrenches. The NOC matrix $\mathbf{N}$ when written in the decoupled form [16] leads to the generalized twist $\mathbf{t} \equiv \mathbf{N}_l\mathbf{N}_d\dot{\mathbf{q}}$, where $\mathbf{N} \equiv \mathbf{N}_l\mathbf{N}_d$. Its time derivative's $\dot{\mathbf{t}}$ $\equiv \mathbf{N}_l\mathbf{N}_d\ddot{\mathbf{q}} + (\dot{\mathbf{N}}_l\mathbf{N}_d + \mathbf{N}_l\dot{\mathbf{N}}_d)\dot{\mathbf{q}}$, where $\mathbf{N}_l$ and $\mathbf{N}_d$ are the lower block triangular and block diagonal DeNOC matrices, respectively. Pre-multiplying (4) by the transpose of these matrices, i.e., $\mathbf{N}_d^T\mathbf{N}_l^T$, yields a minimal set of equations of motion eliminating the constraint wrenches [27], i.e.,

$$\mathbf{N}_d^T\mathbf{N}_l^T(\mathbf{M\dot{t}} + \mathbf{\Omega MEt}) = \mathbf{N}_d^T\mathbf{N}_l^T(\mathbf{w}^E + \mathbf{w}^\lambda) \quad (5)$$

Note that the constraint wrenches at the uncut joints do not perform any work [28], as pointed out before (5). Substituting the expression of the generalized twist $\mathbf{t}$ and its time derivative $\dot{\mathbf{t}}$ into (5), and rearranging the terms, one obtains the following constraint equations in minimal order

$$\mathbf{I\ddot{q}} + \mathbf{C\dot{q}} = \boldsymbol{\tau}^E + \boldsymbol{\tau}^\lambda \quad (6)$$

where the expressions of the generalized inertia matrix (GIM) $\mathbf{I}$, matrix of convective inertia (MCI) terms $\mathbf{C}$, and vectors of the generalized external forces $\boldsymbol{\tau}^E$, and the generalized force due to Lagrange multipliers $\boldsymbol{\tau}^\lambda$ are given as

$$\mathbf{I} \equiv \mathbf{N}^T\mathbf{M}\mathbf{N}, \;\; \mathbf{C} \equiv \mathbf{N}^T(\mathbf{M}\dot{\mathbf{N}} + \boldsymbol{\Omega}\mathbf{M}\boldsymbol{\mathcal{E}}\mathbf{N}),$$
$$\boldsymbol{\tau}^E \equiv \mathbf{N}^T\mathbf{w}^E \;\text{ and }\; \boldsymbol{\tau}^\lambda \equiv \mathbf{N}^T\mathbf{w}^\lambda \tag{7}$$

where $\mathbf{N}$ has been defined before (5). Equation (6) represents the equations of motion for a cut-open subsystem (serial/ tree-type) subjected to Lagrange multipliers, and external wrenches (driving forces and torques). In fact, similar minimal-order formulations arise by using projection-based methods [12], Kane's method [13] etc. by exploiting the relationships between dependent and independent velocities and accelerations. Now, similar to (6), the independent equations of motion for other subsystems can also be derived, which will be the basis to obtain the expressions for the generalized forces due to Lagrange multipliers and later to carry out simulation of closed-loop systems.

2.2 Loop-closure constraints

The equations of motion of all the subsystems in the form of (6) can be put together for the solution of inverse dynamics, where the independent driving joint forces and torques along with the generalized forces due to Lagrange multipliers, $\boldsymbol{\tau}^\lambda$, are simultaneously solved using the approach described in [7], [25]. Alternatively, subsystems which have smaller number of links and joints in each of them, can be considered one at a time to see if they are *determinate* or *indeterminate*, as proposed in [29]. For a *determinate* subsystem, the joint forces and torques along with the Lagrange multipliers can be solved uniquely from its dynamic equations of motion in the form of (6), which will make one of the *indeterminate* subsystems *determinate*. The above step is then repeated. Such subsystem-level recursion for inverse dynamics algorithm is quite elegant and computationally beneficial [29]. A similar concept is attempted here for the forward dynamics problem as well, in which the joint accelerations for all subsystems are evaluated recursively along with the forces due to Lagrange multipliers, $\boldsymbol{\tau}^\lambda$. Unfortunately, unlike the inverse dynamics problem, the number of dynamic equations of motion available in the form of (6) are not sufficient to solve for both the joint accelerations and Lagrange multipliers. Hence, additional set of equations resulting out of the kinematic constraints satisfying the loop-closures of a closed-loop system are used for the solution of $\ddot{\mathbf{q}}$ and $\boldsymbol{\tau}^\lambda$. For that, all the loop-closure constraint equations are written in the acceleration level, which can be expressed as

$$\mathbf{J}\ddot{\mathbf{q}} = -\dot{\mathbf{J}}\dot{\mathbf{q}} \tag{8}$$

where $\mathbf{J}$ is the constraint Jacobian matrix of the closed-loop system at hand. Depending on the number of independent kinematic loops and the suitable

choice of the generalized coordinates, the sizes of $\mathbf{J}$ and $\ddot{\mathbf{q}}$ will vary. It is worth noting that in (7), $\boldsymbol{\tau}^{\lambda} \equiv \mathbf{N}^T\mathbf{w}^{\lambda}$ can also be represented as $\boldsymbol{\tau}^{\lambda} \equiv \mathbf{J}^T\boldsymbol{\lambda}$, where the wrench $\mathbf{w}^{\lambda}$ due to the Lagrange multipliers is the generalized constraint reaction force vector arising out of cutting the joints to make the closed-loop system open [7], [8], [12].

Note that, at the generalized twist level, the loop-closure constraint equations in terms of $\mathbf{t}$ can be expressed, say as $\hat{\mathbf{J}}\mathbf{t} = \mathbf{0}$, where $\hat{\mathbf{J}}$ is the respective $m{\times}6n$-dimensional constraint Jacobian matrix. The wrench $\mathbf{w}^{\lambda}$ is then $\mathbf{w}^{\lambda} = \hat{\mathbf{J}}^T\boldsymbol{\lambda}$. By projecting the dynamic equations (4) into the directions of $\dot{\mathbf{q}}$ (by pre-multiplying them with $\mathbf{N}^T$), one obtains $\boldsymbol{\tau}^{\lambda} \equiv \mathbf{N}^T\mathbf{w}^{\lambda} = \mathbf{N}^T\hat{\mathbf{J}}^T\boldsymbol{\lambda} = \mathbf{J}^T\boldsymbol{\lambda}$, from which $\mathbf{J} = \hat{\mathbf{J}}\mathbf{N}$. The same can also be deduced from the loop-closure constraints at the generalized twist level, i.e., $\hat{\mathbf{J}}\mathbf{t} = \mathbf{0}$, which after using $\mathbf{t} = \mathbf{N}\dot{\mathbf{q}}$ becomes $\hat{\mathbf{J}}\mathbf{N}\dot{\mathbf{q}} = \mathbf{J}\dot{\mathbf{q}} = \mathbf{0}$. Hence, $\boldsymbol{\tau}^{\lambda} \equiv \mathbf{J}^T\boldsymbol{\lambda}$.

The proposed reduced-order forward dynamics formulation of a closed-loop system is based on the use of (6) and (8), which is presented next.

3 Forward Dynamics

In general, the joint accelerations $\ddot{\mathbf{q}}$, and the Lagrange multipliers $\boldsymbol{\lambda}$, for a closed-loop system can be solved using the system of equations in (6) and (8). In particular, the coefficients of (6) for a closed-loop system, are defined at subsystem level as $\mathbf{I} \equiv \text{diag}\,[\mathbf{I}_1, \ldots, \mathbf{I}_v]$; $\mathbf{C} \equiv \textit{diag}\,[\mathbf{C}_1, \ldots, \mathbf{C}_v]$; $\mathbf{q} \equiv [\mathbf{q}_1^T, \ldots, \mathbf{q}_v^T]^T$; $\boldsymbol{\tau}^E \equiv [(\boldsymbol{\tau}_1^E)^T, \ldots, (\boldsymbol{\tau}_v^E)^T]^T$ and $\boldsymbol{\tau}^{\lambda} \equiv [(\boldsymbol{\tau}_1^{\lambda})^T, \ldots, (\boldsymbol{\tau}_v^{\lambda})^T]^T$ — v being the number of subsystems. Moreover, the vectors, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$, have the usual meaning of joint rate and joint acceleration vectors of v subsystems in the closed-loop system under study. Equations (6) and (8) can now be combined as follows

$$\begin{bmatrix} \mathbf{I} & \mathbf{J}^T \\ \mathbf{J} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi} \\ -\dot{\mathbf{J}}\dot{\mathbf{q}} \end{bmatrix}, \quad \text{where } \boldsymbol{\varphi} \equiv \boldsymbol{\tau}^E - \mathbf{C}\dot{\mathbf{q}} \tag{9}$$

where $\boldsymbol{\tau}^{\lambda}$ in (6) is substituted with $\mathbf{J}^T\boldsymbol{\lambda}$ as discussed in the previous section. Using (9), $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$ can be solved simultaneously by analytical/ explicit inversion of the coefficient matrix on the Left Hand Side (LHS), followed by numerical integration to obtain joint motions. We call this approach of forward dynamics as *system approach* and will use the term hereafter. This approach is known to have some of the disadvantages like constraint violation, inversion of larger dimension matrices etc. as discussed in [7], [8] and [9]. Besides, this approach does not allow one to exploit the already available recursive algorithms and computer programs for the open-loop systems which may be very efficient and numerically stable [30]. For utilizing the algorithms of open-loop systems, one needs to estimate Lagrange multipliers first and then solve the equations of motion of subsystems individually or together. To achieve this, as proposed in [7], [9], [31], $\ddot{\mathbf{q}}$ is expressed first from (6) as

$$\ddot{\mathbf{q}} = \mathbf{I}^{-1}(\mathbf{J}^T\boldsymbol{\lambda} + \boldsymbol{\varphi}) \tag{10}$$

Then, (10) is substituted into (8) to express $\boldsymbol{\lambda}$ as

$$\boldsymbol{\lambda} = -\tilde{\mathbf{I}}^{-1}(\mathbf{J}\mathbf{I}^{-1}\boldsymbol{\varphi} + \dot{\mathbf{J}}\dot{\mathbf{q}}), \text{ where } \tilde{\mathbf{I}} \equiv \mathbf{J}\mathbf{I}^{-1}\mathbf{J}^T \tag{11}$$

For numerical computations, (11) is used first to solve for $\boldsymbol{\lambda}$, which is then treated as external forces to obtain the joint accelerations, $\ddot{\mathbf{q}}$ from (10) using any algorithm already existing for open- and tree-type systems. In fact, the expressions in (11) can also be obtained by block analytical inversion of the coefficient matrix in (9), as detailed in Appendix A.3. Quite possibly, in this approach, after obtaining the Lagrange multipliers, one may solve for $\ddot{\mathbf{q}}$ recursively, as reported in [24] and others, since the coefficient matrix in (10) is a generalized inertia matrix of a tree-type system for which the recursive algorithms are well established. However, from (11), it is clear that the computation of Lagrange multipliers for the whole system would require inversions of the matrices $\mathbf{I}$ and $\tilde{\mathbf{I}}$ which are $n\times n$ and $m\times m$ matrices, respectively. This approach is referred here as *System-level Lagrange Multiplier* (SLM) approach, as the Lagrange multipliers $\boldsymbol{\lambda}$ for the complete system have been evaluated first using (11), before solving for the joint accelerations $\ddot{\mathbf{q}}$.

An important observation is made in the above two formulations, i.e., *system approach* and the SLM approach. They both require inversions of larger size matrices than those associated with the subsystems. Even though the latter approach allows recursive forward dynamics calculations of the joint accelerations, i.e., $\ddot{\mathbf{q}}$, it needs the inversion of matrix $\mathbf{I}$ and $\tilde{\mathbf{I}}$, whose size depends on the total number of generalized coordinates and the independent constraints, respectively. In order to avoid inversions of large dimensional $\mathbf{I}$ and $\tilde{\mathbf{I}}$, requiring $\boldsymbol{O}(n^3)$ and $\boldsymbol{O}(m^3)$ computations [16], which is very inefficient for large n,m [32], it is proposed here that the Lagrange multipliers are solved at subsystem level (smaller dimensions), instead of the whole system in (9) and (11). The proposed formulation thus calculates the Lagrange multipliers at a much reduced-order. These Lagrange multipliers are then substituted in (6) for each subsystem, where the joint accelerations are finally solved utilizing any of the already existing dynamics algorithms for open-loop systems, may be an efficient recursive one like *ReDySim* (Recursive Dynamics Simulator) proposed in [27]. We refer this approach as *Subsystem-level Lagrange Multiplier* (SSLM) approach. In contrast to SLM approach, the Lagrange multipliers acting on the subsystems, which together form the complete system, will be evaluated at subsystem-level before solving for the joint accelerations of each subsystem.

For the proposed SSLM approach, first the generalized joint constraint matrix of the closed-loop system is formed by writing the loop-closure equations at the joint-rate level as

$$\mathbf{J}\dot{\mathbf{q}} = \mathbf{0} \tag{12}$$

Note that (8) is nothing but the time derivative of (12). In (12), the matrix $\mathbf{J}$ and vector $\dot{\mathbf{q}}$ contain block elements represented as $\mathbf{J}_{i,j}$ and $\dot{\mathbf{q}}_j$, respectively. For a closed-loop system having u independent loops and v subsystems, $i =$

$\{1, 2, ..., u\}$ and $j = \{1, 2, ..., v\}$. Note that depending on the DOF of each serial or tree-type subsystem, the sizes of block matrix $\mathbf{J}_{i,j}$ and block element $\dot{\mathbf{q}}_j$ will vary. Quite interestingly, for a typical multi-closed-loop system, some of the block matrices in (12) vanishes, i.e., $\mathbf{J}_{i,j} \equiv \mathbf{O}$. This is the case when an independent Loop i and subsystem j do not share any link (in other words, $\mathbf{J}_{i,j} \equiv \mathbf{O}$ whenever subsystem j does not have any contribution in the formation of Loop i). This observation is further illustrated as follows: Fig. 1(b) represents a tree-type system that resulted after placing cuts at various joints of the multi-closed-loop system given in Fig. 1(a). This tree-type system consists of three subsystems with subsystem I and III having serial, while subsystem II having a tree-type architecture, respectively. Now, as can be seen from Fig. 1(a), Loop 1: 1-2-15-6-1 shares links #1 and #2 with subsystem I and link #6 with subsystem II. Whereas Loop 1 does not have any link common with subsystem III. Hence, according to the above observation, $\mathbf{J}_{1,1}$, $\mathbf{J}_{1,2}$, $\neq \mathbf{O}$, while $\mathbf{J}_{1,3} \equiv \mathbf{O}$. Similarly Loop 2: 7-14-13-12-7 is formed out of links (#5, #6, #8 and #9) in subsystem II and does not have any link common to subsystem I and subsystem III. Hence, $\mathbf{J}_{2,2} \neq \mathbf{O}$, while $\mathbf{J}_{2,1}$, $\mathbf{J}_{2,3} \equiv \mathbf{O}$. Similarly the whole joint constraint matrix for the system in Fig. 1, i.e., the excavator in [26], would be given as

$$\mathbf{J} \equiv \begin{bmatrix} \mathbf{J}_{1,1} & \mathbf{J}_{1,2} & \mathbf{O} \\ \mathbf{O} & \mathbf{J}_{2,2} & \mathbf{O} \\ \mathbf{J}_{3,1} & \mathbf{J}_{3,2} & \mathbf{O} \\ \mathbf{J}_{4,1} & \mathbf{O} & \mathbf{J}_{4,3} \end{bmatrix} \tag{13}$$

Now, for a general i^{th} independent loop in (12), the loop-closure equations could also be expressed as

$$\overline{\mathbf{J}}_i \dot{\mathbf{q}} = \mathbf{0} \tag{14}$$

where $\overline{\mathbf{J}}_i \equiv [\mathbf{J}_{i,1}\ \mathbf{J}_{i,2}\ ...\ \mathbf{J}_{i,v}\ ]$ and $\dot{\mathbf{q}} \equiv [\dot{\mathbf{q}}_1^T\ \dot{\mathbf{q}}_2^T\ ...\ \dot{\mathbf{q}}_v^T]^T$. The latter denotes the time derivative of the generalized coordinates associated with v subsystems. Physically, $\overline{\mathbf{J}}_i$ is defined as an entity that contains information about the relation of Loop i with various subsystems of a closed-loop system. Similarly, $\overline{\mathbf{J}}_j \equiv [\mathbf{J}_{1,j}^T\ \mathbf{J}_{2,j}^T\ ...\ \mathbf{J}_{u,j}^T]$ is defined as an entity that contains information about the relation of subsystem j with various loops of a closed-loop system. Now, differentiating (12), the constraints at acceleration-level i.e., (8), are obtained. Denoting -$\dot{\mathbf{J}}\dot{\mathbf{q}}$ as $\mathbf{\Psi}$ for compact representation, (8) is re-written as $\mathbf{J}\ddot{\mathbf{q}} = \mathbf{\Psi}$, where the matrix $\mathbf{J}$ and the vectors $\dot{\mathbf{q}}$ and $\mathbf{\Psi}$ are defined as

$$\mathbf{J} \equiv \begin{bmatrix} \mathbf{J}_{1,1} & \mathbf{J}_{1,2} & \cdots & \mathbf{J}_{1,j} & \cdots & \mathbf{J}_{1,v} \\ \mathbf{J}_{2,1} & \mathbf{J}_{2,2} & & & & \mathbf{J}_{2,v} \\ \vdots & & \ddots & & & \vdots \\ \mathbf{J}_{i,1} & & & \mathbf{J}_{i,j} & & \mathbf{J}_{i,v} \\ \vdots & & & & \ddots & \vdots \\ \mathbf{J}_{u,1} & \mathbf{J}_{u,2} & \cdots & \mathbf{J}_{u,j} & \cdots & \mathbf{J}_{u,v} \end{bmatrix}, \quad \ddot{\mathbf{q}} \equiv \begin{bmatrix} \ddot{\mathbf{q}}_1 \\ \ddot{\mathbf{q}}_2 \\ \vdots \\ \ddot{\mathbf{q}}_j \\ \vdots \\ \ddot{\mathbf{q}}_v \end{bmatrix}, \quad \mathbf{\Psi} \equiv \begin{bmatrix} \mathbf{\Psi}_1 \\ \mathbf{\Psi}_2 \\ \vdots \\ \mathbf{\Psi}_j \\ \vdots \\ \mathbf{\Psi}_u \end{bmatrix} \tag{15}$$

For a multi-closed-loop system, typically the matrix $\mathbf{J}$ of (15) is sparse, as evident from (13) and later in (25) and (36). Next, to solve the forward dynamics

problem, the constrained independent equations of motion at subsystem level for v–subsystems, are written from (6) as

$$\mathbf{I}\ddot{\mathbf{q}} = \boldsymbol{\varphi} + \mathbf{J}^T\boldsymbol{\lambda} \tag{16}$$

where the matrix $\mathbf{I}$ and the vectors $\boldsymbol{\varphi}$, $\boldsymbol{\lambda}$ are defined as

$$\mathbf{I} \equiv \begin{bmatrix} \mathbf{I}_1 & & \mathbf{O}'s \\ & \mathbf{I}_2 & \\ & \ddots & \\ \mathbf{O}'s & & \mathbf{I}_v \end{bmatrix}, \boldsymbol{\varphi} \equiv \begin{bmatrix} \boldsymbol{\varphi}_1 \\ \boldsymbol{\varphi}_2 \\ \vdots \\ \boldsymbol{\varphi}_v \end{bmatrix}, \boldsymbol{\lambda} \equiv \begin{bmatrix} \boldsymbol{\lambda_1} \\ \boldsymbol{\lambda_2} \\ \vdots \\ \boldsymbol{\lambda_u} \end{bmatrix} \tag{17}$$

where $\boldsymbol{\lambda}_i$ denotes the vector of Lagrange multipliers associated with the i^{th} loop. These Lagrange multipliers are acting on the v–subsystems and will be treated as external forces for the forward dynamics of open-loop subsystems. In order to detail the efficacy of the proposed approach, the expression for the joint accelerations of j^{th} subsystem can be written using the matrices and vectors defined after (14), and in (15) and (17) as

$$\ddot{\mathbf{q}}_j = \mathbf{I}_j^{-1}(\boldsymbol{\varphi}_j + \overline{\mathbf{J}}_j^T\boldsymbol{\lambda}) = \hat{\boldsymbol{\phi}}_j + \hat{\mathbf{I}}_j^T\boldsymbol{\lambda} \tag{18}$$

where $\boldsymbol{\varphi}_j$, $\hat{\boldsymbol{\phi}}_j$ and $\hat{\mathbf{I}}_j$ are defined as

$$\boldsymbol{\varphi}_j \equiv \boldsymbol{\tau_j}^E - \mathbf{C}_j\dot{\mathbf{q}}_j, \;\; \hat{\boldsymbol{\phi}}_j \equiv \mathbf{I}_j^{-1}\boldsymbol{\varphi}_j, \;\; \hat{\mathbf{I}}_j \equiv \overline{\mathbf{J}}_j\mathbf{I}_j^{-T} \equiv [\hat{\mathbf{I}}_{1,j}^T\hat{\mathbf{I}}_{2,j}^T \ldots \hat{\mathbf{I}}_{u,j}^T] \tag{19}$$

in which $\hat{\mathbf{I}}_{i,j}^T \equiv \mathbf{J}_{i,j}\mathbf{I}_j^{-T}$. Note that to obtain (18), one requires only the inversions of the generalized inertia matrices $\mathbf{I}_j$'s of the subsystems. Using (18) and (19), joint accelerations for all the v subsystems are written in compact form as

$$\ddot{\mathbf{q}} = \hat{\boldsymbol{\phi}} + \hat{\mathbf{I}}^T\boldsymbol{\lambda} \tag{20}$$

where $\hat{\boldsymbol{\phi}}$ and $\hat{\mathbf{I}}$ are defined in terms of their block elements as

$$\hat{\boldsymbol{\phi}} \equiv \begin{bmatrix} \hat{\boldsymbol{\phi}}_1 \\ \hat{\boldsymbol{\phi}}_2 \\ \vdots \\ \hat{\boldsymbol{\phi}}_v \end{bmatrix}, \;\; \hat{\mathbf{I}} \equiv \begin{bmatrix} \hat{\mathbf{I}}_{1,1} & \hat{\mathbf{I}}_{1,2} & \cdots & \hat{\mathbf{I}}_{1,v} \\ \hat{\mathbf{I}}_{2,1} & \ddots & & \vdots \\ \vdots & & & \vdots \\ \hat{\mathbf{I}}_{u,1} & \cdots & \cdots & \hat{\mathbf{I}}_{u,v} \end{bmatrix} \tag{21}$$

where the matrix $\hat{\mathbf{I}}$ has vanishing and non-vanishing block elements exactly at the same locations as those of $\mathbf{J}$ in (15). This will be evident from (25), (29) and (36), (40). Substituting the vector of joint accelerations from (20) into the constraints at acceleration level given after (14) leads to

$$\overline{\mathbf{I}}\boldsymbol{\lambda} = \overline{\boldsymbol{\Psi}}, \;\; \text{where } \overline{\mathbf{I}} \equiv \mathbf{J}\hat{\mathbf{I}}^T \;\; \text{and } \overline{\boldsymbol{\Psi}} \equiv \boldsymbol{\Psi} - \mathbf{J}\hat{\boldsymbol{\phi}} \tag{22}$$

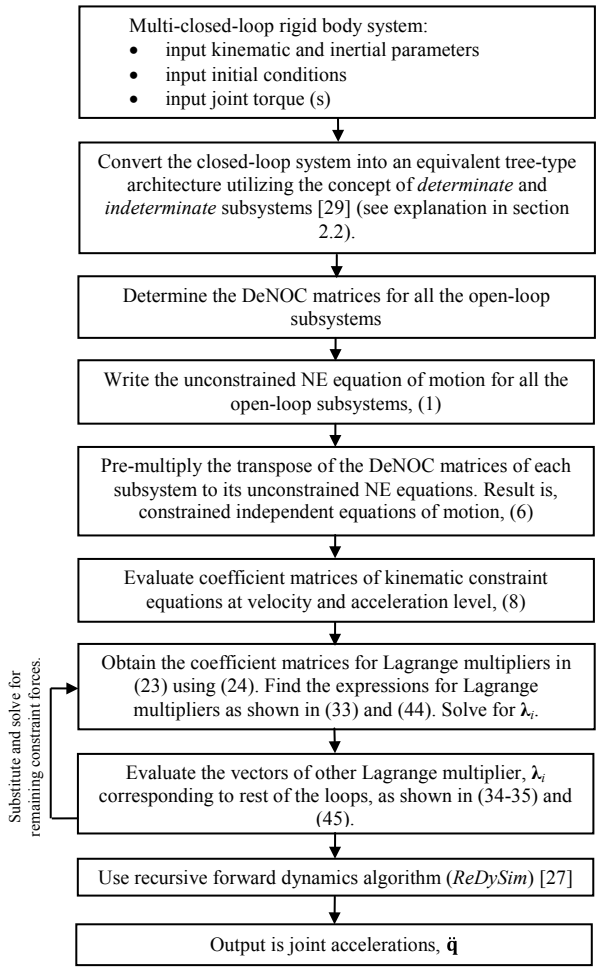


Fig. 2 Proposed forward dynamics algorithm

The block representations of the symmetric matrix $\overline{\mathbf{I}}$ and vector $\overline{\mathbf{\Psi}}$ are given next as follows

$$\overline{\mathbf{I}} \equiv \begin{bmatrix} \overline{\mathbf{I}}_{1,1} & & & sym \\ \overline{\mathbf{I}}_{2,1} & \overline{\mathbf{I}}_{2,2} & & \\ \vdots & & \ddots & \\ \overline{\mathbf{I}}_{u,1} & \cdots & \cdots & \overline{\mathbf{I}}_{u,u} \end{bmatrix}, \quad \overline{\mathbf{\Psi}} \equiv \begin{bmatrix} \overline{\mathbf{\Psi}}_1 \\ \overline{\mathbf{\Psi}}_2 \\ \vdots \\ \overline{\mathbf{\Psi}}_u \end{bmatrix} \tag{23}$$

In (23), "*sym*" means symmetric elements, where the block elements of matrix $\bar{\mathbf{I}}$ and vector $\overline{\boldsymbol{\Psi}}$ are represented as

$$\bar{\mathbf{I}}_{r,s} \equiv \sum_{j=1}^{v} \bar{\mathbf{I}}_j^{rs} \text{ and } \overline{\boldsymbol{\Psi}}_i \equiv \boldsymbol{\Psi}_i - \sum_{j=1}^{v} \mathbf{J}_{i,j} \hat{\boldsymbol{\phi}}_j, \text{ where } \bar{\mathbf{I}}_j^{rs} \equiv \mathbf{J}_{r,j} \mathbf{I}_j^{-1} \mathbf{J}_{s,j}^T \quad (24)$$

where r, $s = \{1, 2, ..., u\}$ denote the row and column indices of matrix $\bar{\mathbf{I}}$. The matrix $\bar{\mathbf{I}}$ referred here as *Generalized Inertia Constraint Matrix* (GICM). Typically, each block element of the GICM $\bar{\mathbf{I}}$, i.e., $\bar{\mathbf{I}}_{r,s}$ has interesting interpretation. For instance, in a multi-closed-loop system, if Loop i is formed using subsystems j_1 and j_2, the corresponding block diagonal element of $\bar{\mathbf{I}}$ in (23) indicated with $\bar{\mathbf{I}}_{i,i}$, will comprise of the matrices associated to the subsystems j_1 and j_2 only, i.e., $\bar{\mathbf{I}}_{i,i} = \bar{\mathbf{I}}_{j_1}^{ii} + \bar{\mathbf{I}}_{j_2}^{ii}$. Similarly, if two loops i_1 and i_2 share a common link belonging to a subsystem j, then one can obtain $\bar{\mathbf{I}}_{i_1,i_2} \equiv \bar{\mathbf{I}}_j^{i_1 i_2}$. These interpretations will be much clearer after the examples of Sect. 4.

Using (22), expressions for the Lagrange multipliers are derived next by carrying out the block Gaussian elimination of $\bar{\mathbf{I}}$, as explained in Appendix A.4 for one of the examples reported in this paper, namely, the 3-RRR parallel manipulator. For the proposed SSLM approach explained above, estimation of Lagrange multipliers would require only the explicit inversions of the subsystem inertia matrices $\mathbf{I}_j$ and the intermediate matrices $\bar{\mathbf{I}}_{r,s}$ only, which have generally much smaller dimensions compared to the complete tree-type system arising out of the original closed-loop system. As a result, opening the original closed-loop system into several serial and tree-type systems was fully exploited. Moreover, the expressions for the vector of Lagrange multipliers corresponding to each independent loop are obtained in a *reduced-order* form. This allows one to perform subsystem-level calculations instead of considering the whole system together, as done by other researchers in [7], [24]. Inversion of smaller-sized matrices is very advantageous in terms of computational speed required. In many instances they will have even explicit expressions, particularly, when the matrix sizes are 2× 2 or 3×3, without requiring any numerical inversions that are prone to numerical sensitivity. This is one of the primary contributions of this work. Additionally, note that once vectors $\boldsymbol{\lambda}_i$'*s* are known, one can solve for the forward dynamics of all serial- and/ or tree-type sub-systems using parallel computations, which is not discussed here due to possible diversion from the main focus of the present paper, i.e., the subsystem-level, reduced-order forward dynamics. For the solution of forward dynamics, i.e., to find the joint accelerations, one can now easily use already available well-established algorithms for open-loop systems e.g., [5], [6], [27], etc. In this paper, the recursive algorithm *ReDySim* for tree-type systems reported in [27] was used to solve for the forward dynamics of the subsystems. Figure 2 shows the complete scheme of the proposed forward dynamics algorithm.

4 Examples

In this section, the calculation of Lagrange multipliers at *reduced-order* level is demonstrated with an industrial carpet scrapping mechanism and a 3-RRR parallel manipulator. The solution for their joint accelerations $\ddot{\mathbf{q}}$ and the numerical integrations were performed using an already existing algorithm for tree-type systems *ReDySim*. Whereas this section evaluates the Lagrange multipliers, the simulation results, i.e., the forward dynamics and numerical integration, using *ReDySim* are reported in Sect. 5.

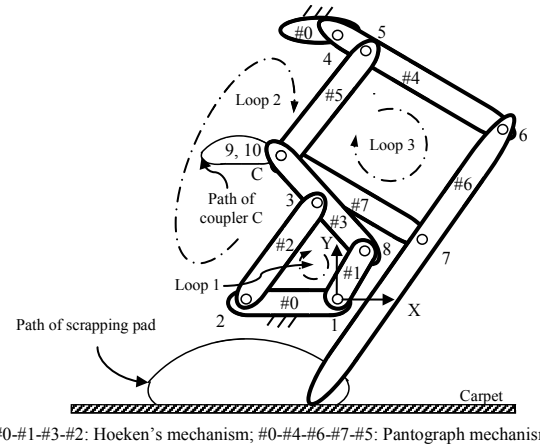


Fig. 3 The multi-loop carpet scrapping mechanism [27]

4.1 An industrial carpet scrapping mechanism

Figure 3 shows an industrial carpet scrapping mechanism used to clean a hand-knotted carpet [27]. It has a multi-closed-loop architecture. The mechanism consists of seven links which are numbered appropriately with the symbol '#'. The joints are simply numbered as 1, 2 etc. To solve the problem of forward dynamics, the system was divided into three subsystems by placing cuts at joints 8, 9 and 10. While cutting the joints, the concept of obtaining *determinate* and *indeterminate* subsystems, as briefly mentioned in Sect. 2.2 and explained in detail in [29], was kept in mind. The resulting subsystems after replacing the appropriate joints with Lagrange multipliers are shown in Fig. 4. These Lagrange multipliers are the two-dimensional vectors that contain the horizontal and vertical scalar components of the joint reaction forces at the cut joints. The constraint equations for the system were then derived by considering three independent loops shown in Fig. 3, namely, i)

Loop 1: 1-2-3-8-1, ii) Loop 2: 2-3-9-5-4-2, and iii) Loop 3: 5-6-7-10-5. The loop-closure equations for the three subsystems and three independent loops in the form of (12) have the following definitions of $\mathbf{J}$ and $\dot{\mathbf{q}}$

$$\mathbf{J} \equiv \begin{bmatrix} \mathbf{J}_{1,1} & \mathbf{J}_{1,2} & \mathbf{O} \\ \mathbf{O} & \mathbf{J}_{2,2} & \mathbf{J}_{2,3} \\ \mathbf{O} & \mathbf{O} & \mathbf{J}_{3,3} \end{bmatrix}, \quad \dot{\mathbf{q}} \equiv \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \\ \dot{\mathbf{q}}_3 \end{bmatrix} \tag{25}$$

where, for the planar motion of the mechanism, $\mathbf{J}_{1,1}$, $\mathbf{J}_{1,2}$, $\mathbf{J}_{2,2}$, $\mathbf{J}_{2,3}$ and $\mathbf{J}_{3,3}$ are the 2×1, 2×2, 2×2, 2×4 and 2×4 matrices, respectively, whose scalar elements are given in Appendix A.1, while $\dot{\mathbf{q}}_1$,$\dot{\mathbf{q}}_2$ and $\dot{\mathbf{q}}_3$ are a scalar, 2- and 4-dimensional vectors, respectively, which are defined as

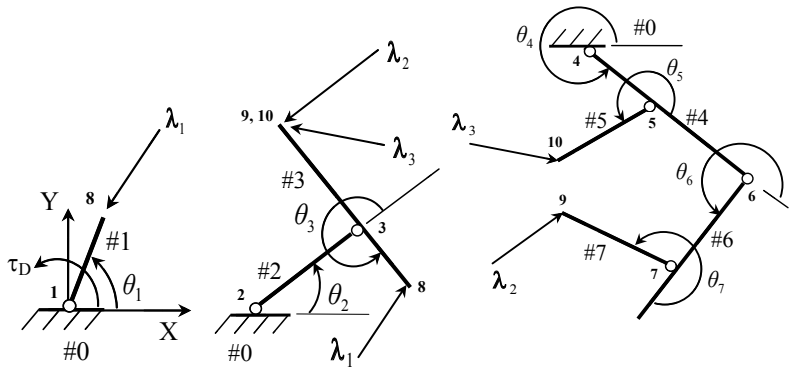
$$\dot{\mathbf{q}}_1 \equiv \dot{\theta}_1, \quad \dot{\mathbf{q}}_2 \equiv \begin{bmatrix} \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \text{ and } \dot{\mathbf{q}}_3 \equiv \begin{bmatrix} \dot{\theta}_4 \\ \dot{\theta}_5 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \end{bmatrix} \tag{26}$$

where θ_i's, for $i = \{1, 2, ..., 7\}$, are the joint angles of the revolute joints present in the scrapping mechanism. They are indicated in Fig. 4. Note that $\mathbf{O}$ in (25) represents the matrix of zeros with compatible sizes depending upon where it appears. For example, $\mathbf{O}$ in the first block row of (25) denotes the 2×4 matrix of zeros, because there are two rows corresponding to the two scalar constraints of the first loop of Fig. 3, giving rise to the first block row of (25), and four columns associated to the joint variables of the third subsystem. As mentioned in (13), several block matrices resulted from the non-sharing of any link between a loop and a subsystem must vanish. Hence, in (25), matrices $\mathbf{J}_{1,3}$, $\mathbf{J}_{2,1}$, $\mathbf{J}_{3,1}$ and $\mathbf{J}_{3,2}$ are matrices of zeros because Loop 1 does not share any link from subsystem 3, and similarly for others. In (26), $\dot{\mathbf{q}}_1$, $\dot{\mathbf{q}}_2$ and $\dot{\mathbf{q}}_3$ represent the vectors of joint-rates for the three subsystems, namely, subsystem 1, 2 and 3, respectively. The constrained equations of motion for all the three subsystems together can then be expressed in the form of (16), in which matrix $\mathbf{I}$, and vectors $\boldsymbol{\varphi}$ and $\boldsymbol{\lambda}$ have the following representations

$$\mathbf{I} \equiv \begin{bmatrix} \mathbf{I}_1 & & sym \\ & \mathbf{I}_2 & \\ \mathbf{O}'s & & \mathbf{I}_3 \end{bmatrix}, \quad \boldsymbol{\varphi} \equiv \begin{bmatrix} \boldsymbol{\varphi}_1 \\ \boldsymbol{\varphi}_2 \\ \boldsymbol{\varphi}_3 \end{bmatrix}, \quad \boldsymbol{\lambda} \equiv \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \\ \boldsymbol{\lambda}_3 \end{bmatrix} \tag{27}$$

where $\boldsymbol{\varphi}_1 \equiv \boldsymbol{\tau_D}$ - $\mathbf{C}_1\dot{\mathbf{q}}_1$, $\boldsymbol{\varphi}_2 \equiv$ - $\mathbf{C}_2\dot{\mathbf{q}}_2$ and $\boldsymbol{\varphi}_3 \equiv$ - $\mathbf{C}_3\dot{\mathbf{q}}_3$. Moreover, $\ddot{\mathbf{q}}$ is the time derivative of $\dot{\mathbf{q}}$ defined in (25), and the matrix $\mathbf{J}$ is shown in (25). The associated three loop-closure constraint equations (25) are then expressed in the form of $\mathbf{J}\ddot{\mathbf{q}} = \boldsymbol{\Psi}$, where $\boldsymbol{\Psi} \equiv -\dot{\mathbf{J}}\dot{\mathbf{q}}$ is expressed in block form as

$$\boldsymbol{\Psi} \equiv \begin{bmatrix} \boldsymbol{\Psi}_1 \\ \boldsymbol{\Psi}_2 \\ \boldsymbol{\Psi}_3 \end{bmatrix} \equiv - \begin{bmatrix} \dot{\mathbf{J}}_{1,1} & \dot{\mathbf{J}}_{1,2} & \mathbf{O} \\ \mathbf{O} & \dot{\mathbf{J}}_{2,2} & \dot{\mathbf{J}}_{2,3} \\ \mathbf{O} & \mathbf{O} & \dot{\mathbf{J}}_{3,3} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \\ \dot{\mathbf{q}}_3 \end{bmatrix} \tag{28}$$



(a) Subsystem 1 (one-link) (b) Subsystem 2 (two-links) (c) Subsystem 3 (four-links)

Fig. 4 Subsystem-level representation of carpet scrapping mechanism

Now, the joint accelerations for all subsystems can be obtained in the form of (20), where

$$\hat{\boldsymbol{\phi}} \equiv \begin{bmatrix} \hat{\boldsymbol{\phi}}_1 \\ \hat{\boldsymbol{\phi}}_2 \\ \hat{\boldsymbol{\phi}}_3 \end{bmatrix}, \ \hat{\mathbf{I}} \equiv \begin{bmatrix} \hat{\mathbf{I}}_{1,1} & \hat{\mathbf{I}}_{1,2} & \mathbf{O} \\ \mathbf{O} & \hat{\mathbf{I}}_{2,2} & \hat{\mathbf{I}}_{2,3} \\ \mathbf{O} & \mathbf{O} & \hat{\mathbf{I}}_{3,3} \end{bmatrix} \tag{29}$$

$\hat{\boldsymbol{\phi}}_i$ and $\hat{\mathbf{I}}_{i,j}$ being defined in (19) and after (19), respectively. As pointed out after (21), the locations of the vanishing and non-vanishing block elements of the matrices $\mathbf{J}$ and $\hat{\mathbf{I}}$ appearing in (25) and (29), respectively, are same. This is due to the block diagonal representation of matrix $\mathbf{I}$ of (27) using which $\hat{\mathbf{I}}$ is derived. Further, for the scrapping mechanism the symmetric matrix $\overline{\mathbf{I}}$ and vector $\overline{\boldsymbol{\Psi}}$ of (23) are obtained as

$$\overline{\mathbf{I}} \equiv \begin{bmatrix} \overline{\mathbf{I}}_{1,1} & & sym \\ \overline{\mathbf{I}}_{2,1} & \overline{\mathbf{I}}_{2,2} & \\ \mathbf{O} & \overline{\mathbf{I}}_{3,2} & \overline{\mathbf{I}}_{3,3} \end{bmatrix}, \ \overline{\boldsymbol{\Psi}} \equiv \begin{bmatrix} \overline{\boldsymbol{\Psi}}_1 \\ \overline{\boldsymbol{\Psi}}_2 \\ \overline{\boldsymbol{\Psi}}_3 \end{bmatrix} \tag{30}$$

where the non-zero 2×2 matrices $\overline{\mathbf{I}}_{1,1}$, $\overline{\mathbf{I}}_{2,1}$, $\overline{\mathbf{I}}_{2,2}$, $\overline{\mathbf{I}}_{3,2}$ and $\overline{\mathbf{I}}_{3,3}$ are obtained using (24) as

$$\overline{\mathbf{I}}_{1,1} \equiv \overline{\mathbf{I}}_1^{11} + \overline{\mathbf{I}}_2^{11}, \ \overline{\mathbf{I}}_{2,1} \equiv \overline{\mathbf{I}}_2^{21}, \ \overline{\mathbf{I}}_{2,2} \equiv \overline{\mathbf{I}}_2^{22} + \overline{\mathbf{I}}_3^{22}, \overline{\mathbf{I}}_{3,2} \equiv \overline{\mathbf{I}}_3^{32}, \overline{\mathbf{I}}_{3,3} \equiv \overline{\mathbf{I}}_3^{33} \tag{31}$$

whereas the 2-dimensional vectors $\overline{\boldsymbol{\Psi}}_1$, $\overline{\boldsymbol{\Psi}}_2$ and $\overline{\boldsymbol{\Psi}}_3$ are obtained using (24) as

$$\begin{aligned} &\overline{\boldsymbol{\Psi}}_1 \equiv \boldsymbol{\Psi}_1 - \mathbf{J}_{1,1}\hat{\boldsymbol{\phi}}_1 - \mathbf{J}_{1,2}\hat{\boldsymbol{\phi}}_2, \\ &\overline{\boldsymbol{\Psi}}_2 \equiv \boldsymbol{\Psi}_2 - \mathbf{J}_{2,2}\hat{\boldsymbol{\phi}}_2 - \mathbf{J}_{2,3}\hat{\boldsymbol{\phi}}_3, \ \ \overline{\boldsymbol{\Psi}}_3 \equiv \boldsymbol{\Psi}_3 - \mathbf{J}_{3,3}\hat{\boldsymbol{\phi}}_3 \end{aligned} \tag{32}$$

Using the expressions of (30 – 32), the 2-dimensional vectors of Lagrange multipliers $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$ and $\boldsymbol{\lambda}_3$ are obtained after carrying out the block Gaussian elimination, as explained after (24). The expressions of $\boldsymbol{\lambda}_i$'s, for i = 3, 2, 1, are given below

$$\boldsymbol{\lambda}_3 \equiv \tilde{\mathbf{I}}_{3,3}^{-1}\tilde{\boldsymbol{\Psi}}_3 \tag{33}$$

$$\boldsymbol{\lambda}_2 \equiv \tilde{\mathbf{I}}_{2,2}^{-1}(\tilde{\boldsymbol{\Psi}}_2 - \bar{\mathbf{I}}_{3,2}^{T}\boldsymbol{\lambda}_3) \tag{34}$$

$$\boldsymbol{\lambda}_1 \equiv \bar{\mathbf{I}}_{1,1}^{-1}(\overline{\boldsymbol{\Psi}}_1 - \bar{\mathbf{I}}_{2,1}^{T}\boldsymbol{\lambda}_2) \tag{35}$$

where the 2-dimensional matrices $\tilde{\mathbf{I}}_{2,2}$ and $\tilde{\mathbf{I}}_{3,3}$ are given by

$\tilde{\mathbf{I}}_{2,2} \equiv \bar{\mathbf{I}}_{2,2} - \bar{\mathbf{I}}_{2,1}\bar{\mathbf{I}}_{1,1}^{-1}\bar{\mathbf{I}}_{2,1}^{T}$ and $\tilde{\mathbf{I}}_{3,3} \equiv \bar{\mathbf{I}}_{3,3}$ - $\bar{\mathbf{I}}_{3,2}\tilde{\mathbf{I}}_{2,2}^{-1}\bar{\mathbf{I}}_{3,2}^{T}$

and the 2-dimensional vectors $\tilde{\boldsymbol{\Psi}}_2$ and $\tilde{\boldsymbol{\Psi}}_3$ are given by

$\tilde{\boldsymbol{\Psi}}_2 \equiv \overline{\boldsymbol{\Psi}}_2$ - $\bar{\mathbf{I}}_{2,1}\ \bar{\mathbf{I}}_{1,1}^{-1}\overline{\boldsymbol{\Psi}}_1$ and $\tilde{\boldsymbol{\Psi}}_3 \equiv \overline{\boldsymbol{\Psi}}_3$ - $\bar{\mathbf{I}}_{3,2}\tilde{\mathbf{I}}_{2,2}^{-1}\tilde{\boldsymbol{\Psi}}_2$.

Once the solutions for vectors $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$, and $\boldsymbol{\lambda}_3$ are obtained, joint accelerations can be solved using (16) by treating the three subsystems of Fig. 4 as three independent tree-type systems subjected to external forces $\boldsymbol{\lambda}_i$ using any existing algorithm for tree-type systems. In many algorithms, namely, those of recursive in nature, e.g., *ReDySim* of [27], the solutions of joint accelerations require only $\boldsymbol{O}(n)$ computations – n being the DOF of the tree-type system at hand. Hence, computational benefits from two sides namely from the proposed reduced-order computation of $\boldsymbol{\lambda}$ requiring inversions of smaller matrices of subsystems, and from the $\boldsymbol{O}(n)$ computations of the joint accelerations using *ReDySim* are expected. Some comparison on these aspects will be reported in Sect. 5.

One important observation in (30) and (31) is the one based on the explanations given after (23) - (24), i.e., the expressions of the GICM of (30) contains information about the architecture of the closed-loop system at hand. For example, the (i,i) block diagonal element holds information about the subsystems that are involved in the formation of Loop i. Referring to Fig. 3, Loop 1 is formed utilizing subsystems 1 and 2. Therefore, the block element at {1,1} is $\bar{\mathbf{I}}_1^{11} + \bar{\mathbf{I}}_2^{11}$. Similarly Loop 2 is formed using subsystems 2 and 3, and thereby the block element at {2,2} is $\bar{\mathbf{I}}_2^{22} + \bar{\mathbf{I}}_3^{22}$. Since Loop 3 is formed using subsystem 3 only, the block element at {3,3} is $\bar{\mathbf{I}}_3^{33}$. Also the non-diagonal block elements could be deduced by noting if any link is shared between the two loops and the subsystem that carries the link. For example, in Fig. 3, Loops 1 and 2 share a common link #2 belonging to subsystem 2. Therefore the block element at {1,2} is $\bar{\mathbf{I}}_2^{12}$, which is same as $(\bar{\mathbf{I}}_2^{21})^T$, because matrix $\bar{\mathbf{I}}$ is symmetric. Since Loops 1 and 3 do not share any subsystem, the block element at {1,3} or {3,1} would be a null matrix. Similarly, block element at {2,3} or {3,2} can be obtained as $\bar{\mathbf{I}}_3^{23} = (\bar{\mathbf{I}}_3^{32})^T$, because Loops 2 and 3 share a common link #5 belonging to subsystem 3. As a result, the complete GICM, $\bar{\mathbf{I}}$ of (23),was

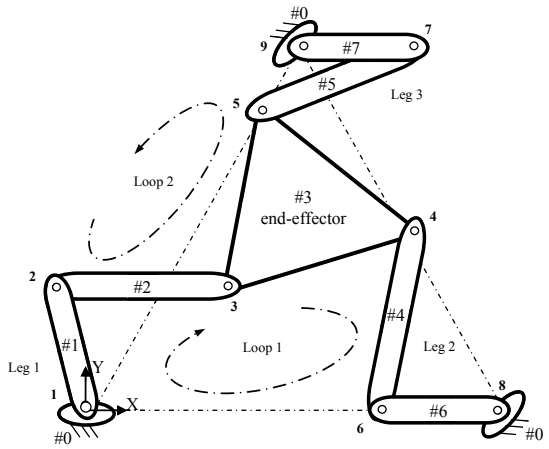


Fig. 5 The 3-RRR parallel manipulator

generated leading to (30). This is a very powerful observation interlinking the interaction between the loops and subsystems, which, as per the authors knowledge was never reported in the literature.

4.2 A 3-RRR parallel manipulator

In this subsection, another application of the proposed SSLM approach is illustrated using the 3-RRR parallel manipulator shown in Fig. 5, where 3 stands for the number of symmetric legs or the DOF of the parallel manipulator. Also, RRR stands for all the three joints in each leg being "Revolute" (R). The given parallel manipulator has 7 movable links with two independent loops. The manipulator is converted into a tree-type architecture by placing cuts according to the concept of *determinate* and *indeterminate* subsystems explained in [29]. In Fig. 5, joints 6 and 7 are cut, resulting into subsystem 1 having five links and subsystems 2 and 3 having single link each. They are shown in Fig. 6. The Lagrange multipliers are then introduced at the cut joints, and the constraint equations for the system are derived by considering the two independent loops, namely i) Loop 1: 1-2-3-4-6-8-1 and ii) Loop 2: 1-2-3-5-7-9-1. The loop closure equations are written next in matrix form of (12) where the Jacobian $\mathbf{J}$ and the vector of joint-rates $\dot{\mathbf{q}}$ are defined as

$$\mathbf{J} \equiv \begin{bmatrix} \mathbf{J}_{1,1} & \mathbf{J}_{1,2} & \mathbf{O} \\ \mathbf{J}_{2,1} & \mathbf{O} & \mathbf{J}_{2,3} \end{bmatrix}, \quad \dot{\mathbf{q}} \equiv \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \\ \dot{\mathbf{q}}_3 \end{bmatrix} \tag{36}$$

For the planar motion of the mechanism, $\mathbf{J}_{1,1}$, $\mathbf{J}_{2,1}$ are the 2×5 matrices and $\mathbf{J}_{1,2}$, $\mathbf{J}_{2,3}$ are the 2-dimensional vectors, respectively, whose elements are given

in Appendix A.2. Also, $\dot{\mathbf{q}}_1$ is the 5-dimensional vector, and $\dot{\mathbf{q}}_2$, $\dot{\mathbf{q}}_3$ are scalar quantities. They are defined as

$$\dot{\mathbf{q}}_1 \equiv \begin{bmatrix} \dot{\theta_1} \\ \dot{\theta_2} \\ \dot{\theta_3} \\ \dot{\theta_4} \\ \dot{\theta_5} \end{bmatrix}, \quad \dot{\mathbf{q}}_2 \equiv \begin{bmatrix} \dot{\theta_6} \end{bmatrix}, \quad \dot{\mathbf{q}}_3 \equiv \begin{bmatrix} \dot{\theta_7} \end{bmatrix} \tag{37}$$

The constrained equations of motion of all the three subsystems are expressed in the form of (16), where

$$\mathbf{I} \equiv \begin{bmatrix} \mathbf{I}_1 & & sym \\ & \mathbf{I}_2 & \\ \mathbf{O}'s & & \mathbf{I}_3 \end{bmatrix}, \quad \boldsymbol{\varphi} \equiv \begin{bmatrix} \boldsymbol{\varphi}_1 \\ \boldsymbol{\varphi}_2 \\ \boldsymbol{\varphi}_3 \end{bmatrix}, \quad \boldsymbol{\lambda} \equiv \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \end{bmatrix} \tag{38}$$

in which $\boldsymbol{\varphi}_1 \equiv \boldsymbol{\tau}_{D1}$ - $\mathbf{C}_1\dot{\mathbf{q}}_1$, $\boldsymbol{\varphi}_2 \equiv \boldsymbol{\tau}_{D2}$ - $\mathbf{C}_2\dot{\mathbf{q}}_2$ and $\boldsymbol{\varphi}_3 \equiv \boldsymbol{\tau}_{D3}$ - $\mathbf{C}_3\dot{\mathbf{q}}_3$. The constraint equations at acceleration level are then expressed in the form of $\mathbf{J}\ddot{\mathbf{q}} = \boldsymbol{\Psi}$, for which

$$\boldsymbol{\Psi} \equiv \begin{bmatrix} \boldsymbol{\Psi}_1 \\ \boldsymbol{\Psi}_2 \end{bmatrix} \equiv - \begin{bmatrix} \dot{\mathbf{J}}_{1,1} & \dot{\mathbf{J}}_{1,2} & \mathbf{O} \\ \dot{\mathbf{J}}_{2,1} & \mathbf{O} & \dot{\mathbf{J}}_{2,3} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \\ \dot{\mathbf{q}}_3 \end{bmatrix} \tag{39}$$

Joint accelerations are then calculated from (20), where

$$\hat{\boldsymbol{\phi}} \equiv \begin{bmatrix} \hat{\boldsymbol{\phi}}_1 \\ \hat{\boldsymbol{\phi}}_2 \\ \hat{\boldsymbol{\phi}}_3 \end{bmatrix}, \quad \hat{\mathbf{I}} \equiv \begin{bmatrix} \hat{\mathbf{I}}_{1,1} & \hat{\mathbf{I}}_{1,2} & \mathbf{O} \\ \hat{\mathbf{I}}_{2,1} & \mathbf{O} & \hat{\mathbf{I}}_{2,3} \end{bmatrix} \tag{40}$$

Note the same locations of the vanishing and non-vanishing block elements of the matrices $\mathbf{J}$ and $\hat{\mathbf{I}}$ given by (36) and (40), respectively. Further, the matrix $\overline{\mathbf{I}}$ and vector $\overline{\boldsymbol{\Psi}}$ of (23) for the 3-RRR parallel manipulator are expressed as

$$\overline{\mathbf{I}} \equiv \begin{bmatrix} \overline{\mathbf{I}}_{1,1} & sym \\ \overline{\mathbf{I}}_{2,1} & \overline{\mathbf{I}}_{2,2} \end{bmatrix}, \quad \overline{\boldsymbol{\Psi}} \equiv \begin{bmatrix} \overline{\boldsymbol{\Psi}}_1 \\ \overline{\boldsymbol{\Psi}}_2 \end{bmatrix} \tag{41}$$

where the 2×2 matrices $\overline{\mathbf{I}}_{1,1}$, $\overline{\mathbf{I}}_{2,1}$ and $\overline{\mathbf{I}}_{2,2}$ are obtained as

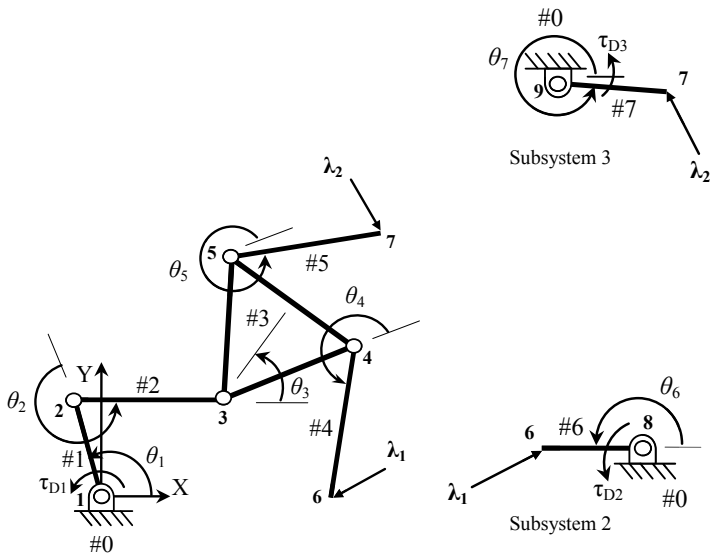
$$\overline{\mathbf{I}}_{1,1} \equiv \overline{\mathbf{I}}_1^{11} + \overline{\mathbf{I}}_2^{11}, \ \overline{\mathbf{I}}_{2,1} \equiv \overline{\mathbf{I}}_1^{21}, \ \overline{\mathbf{I}}_{2,2} \equiv \overline{\mathbf{I}}_1^{22} + \overline{\mathbf{I}}_3^{22} \tag{42}$$

whereas the 2-dimensional vectors $\overline{\boldsymbol{\Psi}}_1$ and $\overline{\boldsymbol{\Psi}}_2$ are given by

$$\begin{aligned} \overline{\boldsymbol{\Psi}}_1 &\equiv \boldsymbol{\Psi}_1 - \mathbf{J}_{1,1}\hat{\boldsymbol{\phi}}_1 - \mathbf{J}_{1,2}\hat{\boldsymbol{\phi}}_2, \\ \overline{\boldsymbol{\Psi}}_2 &\equiv \boldsymbol{\Psi}_2 - \mathbf{J}_{2,1}\hat{\boldsymbol{\phi}}_1 - \mathbf{J}_{2,3}\hat{\boldsymbol{\phi}}_3 \end{aligned} \tag{43}$$

Using the expressions of (41 – 43), the 2-dimensional vectors $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are obtained after carrying out the block Gaussian elimination, as shown in Appendix A.4. They are given below

$$\boldsymbol{\lambda}_2 = (\overline{\mathbf{I}}_{2,2} - \overline{\mathbf{I}}_{2,1}\overline{\mathbf{I}}_{1,1}^{-1}\overline{\mathbf{I}}_{2,1}^{T})^{-1}(\overline{\boldsymbol{\Psi}}_2 - \overline{\mathbf{I}}_{2,1}\overline{\mathbf{I}}_{1,1}^{-1}\overline{\boldsymbol{\Psi}}_1) \tag{44}$$



(a) Subsystem 1 (five links)

(b) Subsystems 2, 3 (one link each)

Fig. 6 Subsystems in a 3-RRR parallel manipulator

$$\boldsymbol{\lambda}_1 = \overline{\mathbf{I}}_{1,1}^{-1}(\overline{\boldsymbol{\Psi}}_1 - \overline{\mathbf{I}}_{2,1}^{T}\boldsymbol{\lambda}_2) \tag{45}$$

Once the solutions for vectors $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are obtained, joint accelerations can be similarly calculated using the recursive algorithms of *ReDySim*[27], which is basically based on (16) of the 3-RRR parallel manipulator, where $\boldsymbol{\lambda}_i$'*s* are treated as external forces. Note that the solutions for $\boldsymbol{\lambda}_i$'*s* given by (44 – 45) can also be obtained by block Gaussian elimination of (9) written at subsystem-level, as explained in Appendix A.5.

As mentioned after (24) and (35), the expressions of $\overline{\mathbf{I}}$, i.e., the GICM of the 3-RRR manipulator given by (41) can be obtained from its three subsystems and two closed loops.

5 Simulation Results and Discussions

In order to generate simulation results, an existing recursive forward dynamics algorithm for tree-type systems, called *ReDySim*[27], was used, to which the Lagrange multipliers were input by implementing external subroutines to *ReDySim*. Initially, free simulation was carried out, i.e., no external torque

Table 1 Link parameters used for the simulation

	Scrapping mechanism			**3-RRR parallel manipulator**		
Sub-system	**Link** #	**Length** (m)	**Mass** (kg)	**Link** #	**Length** (m)	**Mass** (kg)
I	1	0.038	1.5	1	0.4	3
				2, 4, 5	0.6	4
				3	0.4*	8
II	2	0.1152	3	6	0.4	3
	3	0.2304	5			
III	4	0.3346	4.2	7	0.4	3
	5	0.239	3			
	6	0.8365	10.5			
	7	0.239	3			

* represents the side of equilateral triangular link.

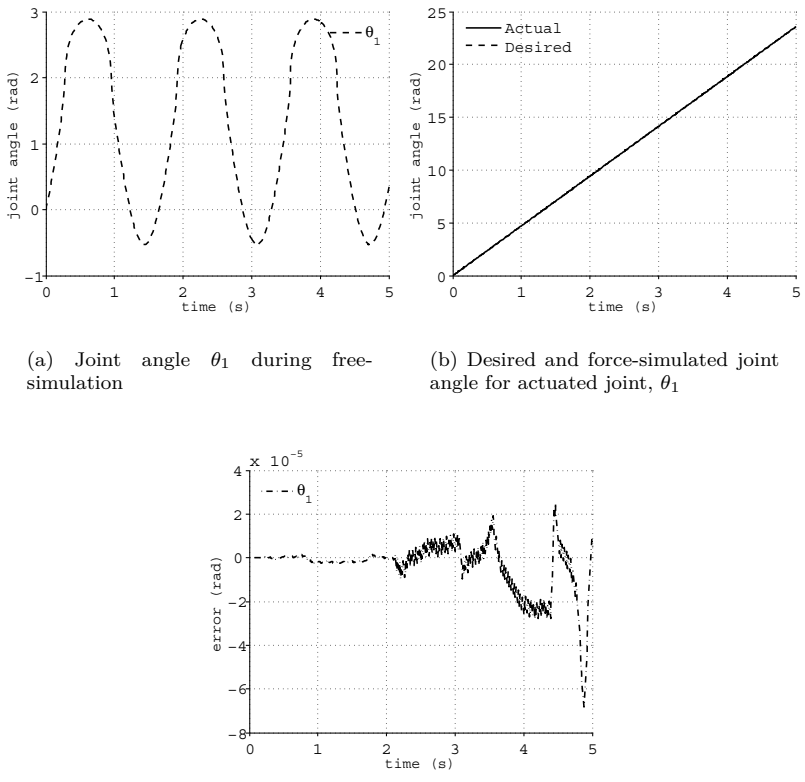
was given to the system. It was let fall freely due to gravity only. Later forced simulation, where input torques were provided at the actuated joints, was carried out. Figures 7 and 8 depict simulation results for the 1-DOF scrapping mechanism and the 3-RRR parallel manipulator, respectively, whose link parameters are given in Table 1. The initial configurations are given in Table 2. Whereas the free-fall simulation results of Fig. 7(a) were verified with those obtained using RecurDyn [33] software, the correctness of the forced simulation was judged from the deviation of the simulated joint angles from the desired angles, shown in Fig. 7(b – c). Since the deviation was very less, the simulation results produced by the proposed algorithm were considered correct. Similar comments can be made for the simulation results of the 3-RRR parallel manipulator shown in Fig. 8(a–c).

In order to see the effect of the proposed methodology on the overall efficiency measured in terms of CPU time, two sets of simulations were performed. One with the conventional methodology based on [7] and [9], where $\boldsymbol{\lambda}$ was solved using (11) before $\ddot{\mathbf{q}}$ was solved using (10) followed by the numerical integrations. Equations (10) and (11) were solved using '\' or *mldivide* command of MATLAB software that uses Gaussian elimination [34] followed by forward

Table 2 Initial configuration parameters

System	**Actuated joint angles**	**Initial position** (rad)	**Initial velocity** (rad/s)
1-DOF Scrapping mechanism*	θ_1	0	4.7124
3-RRR parallel manipulator*	θ_1	1.04	0
	θ_6	4.1883	0
	θ_7	5.7637	0

*Driving torque at actuated joints was obtained from inverse dynamics program with similar initial configurations for 5 sec.



(a) Joint angle θ_1 during free-simulation

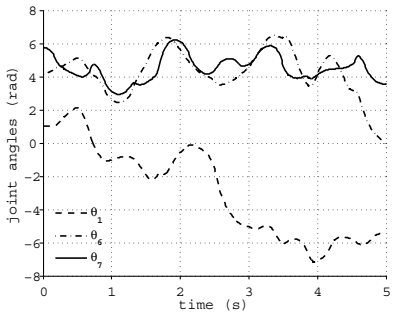
(b) Desired and force-simulated joint angle for actuated joint, θ_1

(c) Error in θ_1 for forced simulation

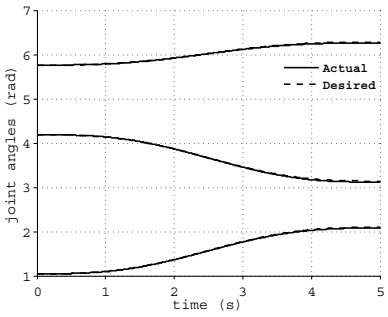
Fig. 7 Simulation results for the scrapping mechanism

and backward substitutions. Another one is the methodology based on (33 – 35), (44 – 45) and the recursive algorithm of *ReDySim*. For each simulation, again two sets of simulation parameters were used. One with a lower tolerance value and larger simulation time, another for a higher tolerance value and smaller simulation time.

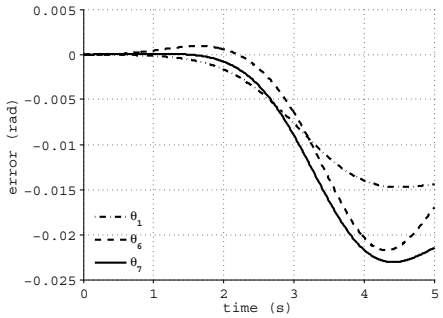
The overall CPU times for both free- and forced-simulation are shown in Table 3, whereas the time taken to calculate the Lagrange multiplies using SLM and SSLM approaches are shown in Fig. 4. Both show improvement with the proposed SSLM approach. The average time taken to compute the Lagrange multipliers using the proposed SSLM approach was about 20 – 30% lesser compared to the SLM approach, as shown in Table 4. This improvement was mainly attributed to the reduced-order computations associated with the pro-



(a) Joint angles $\theta_1, \theta_6, \theta_7$, during free-simulation



(b) Desired and force-simulated joint angles for actuated joints, $\theta_1, \theta_6, \theta_7$



(c) Error in θ_1, θ_6, θ_7 for forced simulation

Fig. 8 Simulation results for the 3-RRR parallel manipulator

posed SSLM approach. The corresponding improvement in terms of CPU time has been reflected in the total CPU time, as observed in Table 3.

6 Conclusions

A *reduced-order* forward dynamics formulation based on several subsystems of a multi-closed-loop system has been proposed in this paper, where a closed-loop system is cut open yielding one or more serial and tree-type subsystems. The Lagrange multipliers used to replace the cut-joints are then computed using the constrained dynamic equations of motion and the the loop-closure kinematic constraints associated with the original closed-loop system. The

Table 3 CPU time for overall simulation

(on Intel Core 2 Duo 2.2 GHz processor, using *ode45* solver for integration)

Mechanism	**SLM**	**SSLM**
1-DOF Scrapping mechanism	2.8 sec[a]	2.72 sec[a]
	6.51 sec[b]	6.41 sec[b]
	9.33 sec[c]	9.25 sec[c]
3-RRR parallel manipulator	3.07 sec[a]	2.96 sec[a]
	10.25 sec[b]	10.12 sec[b]
	12.91 sec[c]	12.81 sec[c]

[a] free-simulation time = 15 sec, step size = 0.01 sec, abs. and rel. tolerance = 10^{-3}
[b] free-simulation time = 5 sec, step size = 0.01 sec, abs. and rel. tolerance = 10^{-8}
[c] forced-simulation time = 5 sec, step size = 0.01 sec, abs. and rel. tolerance = 10^{-8}

Table 4 CPU time for $\boldsymbol{\lambda}$ evaluation

Mechanism	**SLM**	**SSLM**
1-DOF Scrapping mechanism	0.3045 sec[a]	0.2205 sec[a]
3-RRR parallel manipulator	0.2055 sec[a]	0.1575 sec[a]

[a] free-simulation time = 15 sec, step size = 0.01 sec, abs. and rel. tolerance = 10^{-3}

methodology has the following benefits; 1) Introduction of the definition of Generalized Inertia Constraint Matrix (GICM) in (23), which provides the important physical interpretations of the loop-subsystem interactions; 2) Improved efficiency in computing the Lagrange multipliers; 3) Ability to use already existing algorithms for tree-type systems, e.g., *ReDySim*.

In a way, the subsystems approach to inverse dynamics, proposed in [29], has been extended successfully to forward dynamics algorithms as well.

Acknowledgements

The authors would like to thank the Department of Science & Technology, Government of India for financial support to the first author under the sponsored project SR/ S3 /MERC /001 /2008.

References

1. Roberson, R. E., Schwertassek, R.: Dynamics of multibody systems, Springer-Verlag, Berlin Heidelberg (1988)
2. Shabana A.: Forms of the Dynamic Equations and Constrained Dynamics, In: Shabana A. (**3** ed.) Computational Dynamics, 178-209 and 284-349, Wiley, UK (2010)

3. Wittenburg J.: Systems with closed kinematic chains : General Multibody Systems, In: Wittenburg J. (**2** ed.) Dynamics of Multibody Systems, 129-150, Springer, New York (2008)
4. Featherstone R.: Closed Loop Systems, In: Featherstone R. (ed.) Rigid Body Dynamics Algorithms, 141-161. Springer, New York (2008)
5. Walker, M. W., Orin, D. E.: Efficient dynamic computer simulation of robotic mechanisms. J. Dyn. Systems, Measurements, and Control. **104**, 205-211 (1982)
6. Hollerbach, J. M.: A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. IEEE Trans. Systems, Man, and Cybernetics SMC-**10**(11), 730-736, (1980)
7. Blajer, W.: Dependent Variable Formulations, In: Ambrósio, J. A. C., Eberhard P., Maier G., Rammerstorfer F. G., Salençon J., Advanced Design of Mechanical Systems: From Analysis to Optimization, 83-105, CISM Courses and Lectures, Springer Vienna (2009)
8. Blajer, W.: Elimination of Constraint Violation and Accuracy Aspects in Numerical Simulation of Multibody Systems. Multibody Sys. Dyn. **7**(3), 265-284, (2002)
9. Baumgarte, J., Stabilization of constraint and integrals of motion in dynamical systems. Comp. Methods in Appl. Mech. and Eng. **1**, 1-16 (1972)
10. Braun, D., J., Goldfarb, M.: Eliminating constraint drift in the numerical simulation of constrained dynamical systems. Computer Methods in Applied Mechanics and Engineering. **198**, 3151–3160 (2009)
11. Blajer W.: Methods for constraint violation suppression in the numerical simulation of constrained multibody systems A comparative study. Computer Methods in Applied Mechanics and Engineering. **200**, 1568-1576 (2011)
12. Blajer, W.: Independent Variable Formulations, In: Ambrósio, J. A. C., Eberhard P., Maier G., Rammerstorfer F. G., Salençon J., Advanced Design of Mechanical Systems: From Analysis to Optimization, 107-129, CISM Courses and Lectures, Springer Vienna (2009)
13. Kane, T. R., Levinson, D. A.: Dynamics: Theory and Applications, McGraw-Hill, New York (1986)
14. Angeles, J., Ma, O.: Dynamic simulation of n-axis serial robotic manipulators using a natural orthogonal complement. Int. J. of Robot. Res. **7**(5), 32-47 (1988)
15. Brauchli, H., Weber, R.: Dynamical equations in natural coordinates. Comput. Meth. in Appl. Mech. and Eng., **91**, 1403-1414, (1991)
16. Saha, S. K.: Dynamics of Serial Multibody Systems Using the Decoupled Natural Orthogonal Complement Matrices. ASME J. Appl. Mech. **66**, 986-996 (1999)
17. Featherstone, R.: Robot Dynamics Algorithms. Kluwer Academic Publishers, Boston (1987)
18. Saha, S. K., Schiehlen, W. O.: Recursive kinematics and dynamics for parallel structured closed-loop multibody systems. Mech. Struct. Machines **29**(2), 143-175 (2001)
19. Wehage, R. A., Haug, E. J.: Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems. J. Mech. Design **104**, 247-255 (1982)
20. Anderson, K. S., Critchley, J.H.: Improved order-n performance algorithm for the simulation of constrained multirigid- body systems. Multibody Syst. Dyn. **9**, 185-212 (2003)
21. Blajer, W., Schiehlen, W., Schirm, W.: A projective criterion to the coordinate partitioning method for multibody dynamics. Archive of Appl. Mech. **64**, 215-222 (1994)
22. Avello, A., Jimenez, J.M., Bayo, E., Jalon, J.G.: A simple and highly parallelizable method for realtime dynamic simulation based on velocity transformations. Comput. Methods Appl. Mech. Eng. **107**, 313-339 (1993)
23. Critchley, J.H., Anderson, K.S.: A generalized recursive coordinate reduction method for multibody system dynamics. J. Multiscale Comput. Eng. **1** (2,3), 181-200 (2003)
24. Bae, D. S., Haug, E. J.: A Recursive Formulation for Constrained Mechanical System Dynamics: Part II. Closed-Loop Systems. Mech. Struct. Mach. **15**(4), 481-506 (1987)
25. Nikravesh, P. E., Gim, G.: Systematic Construction of the Equations of Motion for Multibody Systems Containing Closed Kinematic Loops. ASME J. Mech. Des., **115**, 143-149 (1993)
26. Janssen, B., Nievelstein, M.: Multiloop linkage dynamics via geometric methods – a case study on a RH200 hydraulic excavator. A Report, Technische Universiteit Eindhoven, Eindhoven (2005) *http://repository.tue.nl/612753*

27. Shah, S. V., Saha S. K., Dutt J. K.: Dynamics of Tree-type Robotic Systems. Intelligent Systems, Control and Automation: Science and Engineering Book series, Springer, Netherlands (2013), *http://www.redysim.co.nr/*
28. Angeles, J.; Fundamentals of Robotic Mechanical Systems. Second Edition, Springer-Verlag, New York (2002)
29. Chaudhary, H., Saha, S.K.: Constraint Force Formulation for Closed-loop Systems Using Two-Level Recursions. Trans. ASME, Journal of Mechanical Design. **129**, 1234–1242 (2007)
30. Mohan, A., Saha, S.K.: A recursive, numerically stable, and efficient simulation algorithm for serial robots. Multibody Syst. Dyn. **17**(4), 291-319 (2007)
31. Stejskal V., Valasek M.: Kinematics and Dynamics of Machinery. Marcel Dekker, Inc., New York, (1996)
32. Miller, E. K.: A computational study of the effect of matrix size and type, condition number, coefficient accuracy and computation precision on matrix-solution accuracy. Antennas and Propagation Society Int. Sym., AP-S. Digest. **2**, 1020–1023 (1995)
33. *http://functionbay.de/en/*
34. *http://www.mathworks.in/help/matlab/ref/mldivide.html*

Appendix A

A.1 Block Jacobian matrices for carpet scrapping mechanism

The analytical expressions for the scalar elements of the block Jacobian matrices obtained in (25) from the loop-closure equations are given below:

$$\mathbf{J}_{1,1} \equiv \begin{bmatrix} a_1 s_1 \\ a_1 c_1 \end{bmatrix}, \quad \mathbf{J}_{1,2} \equiv \begin{bmatrix} -a_2 s_2 - a_3 s_3 & -a_3 s_3 \\ -a_2 c_2 - a_3 c_3 & -a_3 c_3 \end{bmatrix}, \quad \mathbf{J}_{2,2} \equiv \begin{bmatrix} -a_2 s_2 + a_{39} s_3 & a_{39} s_3 \\ -a_2 c_2 - a_{39} c_3 & -a_{39} c_3 \end{bmatrix},$$

$$\mathbf{J}_{2,3} \equiv \begin{bmatrix} a_{59} s_5 + (a_{46} - a_{56}) s_4 & a_{59} s_5 & 0 & 0 \\ -a_{59} c_5 - (a_{46} - a_{56}) c_4 & -a_{59} c_5 & 0 & 0 \end{bmatrix},$$

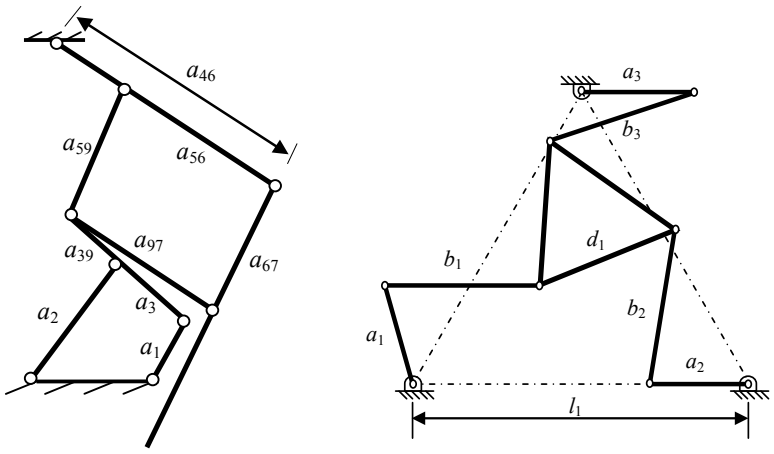
$$\mathbf{J}_{3,3} \equiv \begin{bmatrix} -a_{46} s_4 - a_{67} s_6 - \\ a_{97} s_7 + a_{59} s_5 + (a_{46} - a_{56}) s_4 & a_{59} s_5 & -a_{67} s_6 - a_{97} s_7 & -a_{97} s_7 \\ \\ a_{46} c_4 + a_{67} c_6 + \\ a_{97} c_7 - a_{59} c_5 - (a_{46} - a_{56}) c_4 & -a_{59} c_5 & a_{67} c_6 + a_{97} c_7 & a_{97} c_7 \end{bmatrix}$$

where a_i or a_{ij} represents link length (kinematic parameters), as shown in Fig. 9(a), and s_i and c_i represent $\sin\alpha_i$ and $\cos\alpha_i$, respectively. The relations between α_i's and θ_i's are given by $\alpha_1 = \theta_1$, $\alpha_2 = \theta_2$, $\alpha_3 = \pi + \theta_2 + \theta_3$, $\alpha_4 = \theta_4$, $\alpha_5 = \theta_5 + \theta_4 - 2\pi$, $\alpha_6 = \theta_6 + \theta_4 - 2\pi$ and $\alpha_7 = \theta_7 + \theta_6 + \theta_4 - 4\pi$, where α_i represents the angle subtended by a link with the horizontal line and θ_i's are the relative joint angles between the links at the joint locations indicated in the subscripts of θ_i, as shown in Fig. 4.

A.2 Block Jacobian matrices for 3-RRR parallel manipulator

The analytical expressions for the scalar elements of the block Jacobian matrices obtained in (36) from the loop-closure equations are given below

$$\mathbf{J}_{1,1} \equiv \begin{bmatrix} -a_1 s_1 - b_1 s_4 - d_1 s_7 + b_2 s_5 & -b_1 s_4 - d_1 s_7 + b_2 s_5 & -d_1 s_7 + b_2 s_5 & b_2 s_5 & 0 \\ a_1 c_1 + b_1 c_4 - d_1 c_7 - b_2 c_5 & b_1 c_4 - d_1 c_7 - b_2 c_5 & d_1 c_7 - b_2 c_5 & -b_2 c_5 & 0 \end{bmatrix},$$



(a) 1-DOF Scrapping mechanism

(b) 3-RRR parallel manipulator

Fig. 9 Kinematic architectures

$$\mathbf{J}_{2,1} \equiv \begin{bmatrix} -a_1 s_1 - b_1 s_4 - d_1 s_7' + b_3 s_6 & -b_1 s_4 - d_1 s_7' + b_3 s_6 & -d_1 s_7' + b_3 s_6 & 0 & b_3 s_6 \\ a_1 c_1 + b_1 c_4 + d_1 c_7' - b_3 c_6 & b_1 c_4 + d_1 c_7' - b_3 c_6 & d_1 c_7' - b_3 c_6 & 0 & -b_3 c_6 \end{bmatrix},$$

$$\mathbf{J}_{1,2} \equiv \begin{bmatrix} a_2 s_2 \\ -a_2 c_2 \end{bmatrix}, \quad \mathbf{J}_{1,3} \equiv \mathbf{J}_{2,2} \equiv \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{J}_{2,3} \equiv \begin{bmatrix} a_3 s_3 \\ -a_3 c_3 \end{bmatrix}$$

where a_i, b_i and d_i represent the kinematic parameters of the 3-RRR parallel manipulator shown in Fig. 9(b). The relations between α_i's and θ_i's are given by $\alpha_1 = \theta_1$, $\alpha_2 = \theta_6$, $\alpha_3 = \theta_7$, $\alpha_4 = \theta_1 + \theta_2$, $\alpha_5 = \theta_1 + \theta_2 + \theta_3 + \theta_4 - \pi$, $\alpha_6 = \theta_1 + \theta_2 + \theta_3 + \theta_5 - (\pi - \pi/3)$, $\alpha_7 = \theta_1 + \theta_2 + \theta_3$ and $\alpha_7' = \alpha_7 + \pi/3$.

A.3 Block Gaussian Elimination of (9)

After carrying out the forward block Gaussian elimination of (9), we arrive at

$$\begin{bmatrix} \mathbf{I} & \mathbf{J}^T \\ \mathbf{O} & -\mathbf{J}\mathbf{I}^{-1}\mathbf{J}^T \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi} \\ -\dot{\mathbf{J}}\dot{\mathbf{q}} - \mathbf{J}\mathbf{I}^{-1}\boldsymbol{\varphi} \end{bmatrix} \tag{46}$$

The vector of Lagrange multipliers is thus obtained as

$$\boldsymbol{\lambda} = -(\mathbf{J}\mathbf{I}^{-1}\mathbf{J}^T)^{-1}(\dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\mathbf{I}^{-1}\boldsymbol{\varphi}) \tag{47}$$

which is same as (11).

A.4 Block Gaussian Elimination of (22) for the 3-RRR parallel manipulator

Using the expressions of (41), (22) for the 3-RRR parallel manipulator is reproduced here as

$$\begin{bmatrix} \bar{\mathbf{I}}_{1,1} & \bar{\mathbf{I}}_{2,1}^T \\ \bar{\mathbf{I}}_{2,1} & \bar{\mathbf{I}}_{2,2} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{\Psi}}_1 \\ \overline{\boldsymbol{\Psi}}_2 \end{bmatrix} \tag{48}$$

Applying the forward block Gaussian elimination using the block pivot as $\bar{\mathbf{I}}_{2,1}\bar{\mathbf{I}}_{1,1}^{-1}$, on (48) the following block upper triangular matrix on the LHS is obtained

$$\begin{bmatrix} \bar{\mathbf{I}}_{1,1} & \bar{\mathbf{I}}_{2,1}^T \\ \mathbf{O} & \bar{\mathbf{I}}_{2,2} - \bar{\mathbf{I}}_{2,1}\bar{\mathbf{I}}_{1,1}^{-1}\bar{\mathbf{I}}_{2,1}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{\Psi}}_1 \\ \overline{\boldsymbol{\Psi}}_2 - \bar{\mathbf{I}}_{2,1}\bar{\mathbf{I}}_{1,1}^{-1}\overline{\boldsymbol{\Psi}}_1 \end{bmatrix} \tag{49}$$

Using the backward substitutions, one can then solve easily for $\boldsymbol{\lambda}_2$ and $\boldsymbol{\lambda}_1$, which are given by

$$\boldsymbol{\lambda}_2 \equiv \tilde{\mathbf{I}}_{2,2}^{-1}\tilde{\boldsymbol{\Psi}}_2 \tag{50}$$

$$\boldsymbol{\lambda}_1 \equiv \tilde{\mathbf{I}}_{1,1}^{-1}(\overline{\boldsymbol{\Psi}}_1 - \bar{\mathbf{I}}_{2,1}^T\boldsymbol{\lambda}_2) \tag{51}$$

where $\tilde{\mathbf{I}}_{2,2} \equiv \bar{\mathbf{I}}_{2,2} - \bar{\mathbf{I}}_{2,1}\bar{\mathbf{I}}_{1,1}^{-1}\bar{\mathbf{I}}_{2,1}^T$ and $\tilde{\boldsymbol{\Psi}}_2 \equiv \overline{\boldsymbol{\Psi}}_2 - \bar{\mathbf{I}}_{2,1}\bar{\mathbf{I}}_{1,1}^{-1}\overline{\boldsymbol{\Psi}}_1$

A.5 Block Gaussian Elimination of the subsystem-level representation of (9) for the 3-RRR parallel manipulator

To illustrate that the results of (50 – 51) can also be obtained from the fundamental formulation given by (9), but using the subsystem-level expressions for the 3-RRR parallel manipulator, i.e., (36 – 39), (9) is re-written as

$$\begin{bmatrix} \mathbf{I}_1 & \mathbf{O} & \mathbf{O} & \mathbf{J}_{1,1}^T & \mathbf{J}_{2,1}^T \\ \mathbf{O} & \mathbf{I}_2 & \mathbf{O} & \mathbf{J}_{1,2}^T & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{I}_3 & \mathbf{O} & \mathbf{J}_{2,3}^T \\ \mathbf{J}_{1,1} & \mathbf{J}_{1,2} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{J}_{2,1} & \mathbf{O} & \mathbf{J}_{2,3} & \mathbf{O} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_1 \\ \ddot{\mathbf{q}}_2 \\ \ddot{\mathbf{q}}_3 \\ -\boldsymbol{\lambda}_1 \\ -\boldsymbol{\lambda}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi}_1 \\ \boldsymbol{\varphi}_2 \\ \boldsymbol{\varphi}_3 \\ \boldsymbol{\Psi}_1 \\ \boldsymbol{\Psi}_2 \end{bmatrix} \tag{52}$$

Applying the block Gaussian elimination, the final upper block triangular on the LHS is given by

$$\begin{bmatrix} \mathbf{I}_1 & \mathbf{O} & \mathbf{O} & \mathbf{J}_{1,1}^T & \mathbf{J}_{2,1}^T \\ \mathbf{O} & \mathbf{I}_2 & \mathbf{O} & \mathbf{J}_{1,2}^T & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{I}_3 & \mathbf{O} & \mathbf{J}_{2,3}^T \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & -\bar{\mathbf{I}}_1^{11} - \bar{\mathbf{I}}_2^{11} & -\bar{\mathbf{I}}_1^{12} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & -\bar{\mathbf{I}}_1^{22} - \bar{\mathbf{I}}_3^{22} + \bar{\mathbf{I}}_1^{21}(\bar{\mathbf{I}}_1^{11} + \bar{\mathbf{I}}_2^{11})^{-1}(\bar{\mathbf{I}}_1^{21})^T \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_1 \\ \ddot{\mathbf{q}}_2 \\ \ddot{\mathbf{q}}_3 \\ -\boldsymbol{\lambda}_1 \\ -\boldsymbol{\lambda}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi}_1 \\ \boldsymbol{\varphi}_2 \\ \boldsymbol{\varphi}_3 \\ \tilde{\boldsymbol{\Psi}}_1 \\ \tilde{\boldsymbol{\Psi}}_2 \end{bmatrix} \tag{53}$$

where $\tilde{\boldsymbol{\Psi}}_1 \equiv \boldsymbol{\Psi}_1 - \mathbf{J}_{1,1}\mathbf{I}_1^{-1}\,\boldsymbol{\varphi}_1 - \mathbf{J}_{1,2}\mathbf{I}_2^{-1}\,\boldsymbol{\varphi}_2$ and $\tilde{\boldsymbol{\Psi}}_2 \equiv \boldsymbol{\Psi}_2 - \mathbf{J}_{2,1}\mathbf{I}_1^{-1}\,\boldsymbol{\varphi}_1 - \mathbf{J}_{2,3}\mathbf{I}_3^{-1}\,\boldsymbol{\varphi}_3 - \mathbf{I}_1^{21}(\mathbf{I}_1^{11} + \mathbf{I}_2^{11})^{-1}\tilde{\boldsymbol{\Psi}}_1$. The Lagrange multipliers are computed next using backward substitutions as

$$\boldsymbol{\lambda}_2 \equiv [\bar{\mathbf{I}}_1^{22} + \bar{\mathbf{I}}_3^{22} - \bar{\mathbf{I}}_1^{12}(\bar{\mathbf{I}}_1^{11} + \bar{\mathbf{I}}_2^{11})^{-1}(\bar{\mathbf{I}}_1^{21})^T]^{-1}\tilde{\boldsymbol{\Psi}}_2 \tag{54}$$

$$\boldsymbol{\lambda}_1 \equiv (\bar{\mathbf{I}}_1^{11} + \bar{\mathbf{I}}_2^{11})^{-1}[\overline{\boldsymbol{\Psi}}_1 - (\bar{\mathbf{I}}_1^{21})^T\boldsymbol{\lambda}_2] \tag{55}$$

Substituting the expressions of (42) in (54 – 55), we get exactly the same expressions for $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ to that obtained in (50 – 51) and reported in (44 – 45). Note here that one could continue with the back substitutions to solve for the joint accelerations required in the forward dynamics. This is, however, not recommended here, mainly due to the fact that one is not able to exploit the already existing algorithms for the tree-type systems, which may be of recursive in nature, e.g., *ReDySim* of [27], with good efficiency and numerical stability.