Published in the Int. J. of Mechanics of Structures and Machines, V. 29, N. 2, Aug., 2001, pp. 143--175.

Recursive Kinematics and Dynamics for Closed Loop Multibody Systems*

Subir Kumar Saha†
Dept. of Mech. Eng., IIT Delhi
Hauz Khas, New Delhi 110 016, INDIA
Email: saha@mech.iitd.ernet.in

and

Werner O. Schiehlen
Institut B für Mechanik, Universität Stuttgart
Pfaffenwaldring 9, 70569 Stuttgart, GERMANY
Email: wos@mechb.uni-stuttgart.de.

ABSTRACT

A kinematic constraint formulation of closed loop systems, say, mechanisms, parallel manipulators, etc., in the velocity level is proposed for the development of *recursive* inverse and forward dynamics algorithms to be used in control and simulation, respectively. Recursive algorithms are desired mainly for efficiency, and numerical stability in simulation. In the case of open loop mechanical systems, e.g., a serial manipulator, such formulations are now well established. However, for closed loop systems, e.g., a four-bar mechanism, Stewart platform, existing methodologies using a recursive algorithm, first, transform the closed loop to an open loop by cutting the loop, and then, either introduce Lagrange multipliers or write the loop constraint equations relating the dependent and independent coordinates, respectively. In the former case, even if complete recursion of the open-loop is applied, one is confronted with differential algebraic equations (DAE) of

*This research work has been carried out during the first author's term as Humboldt Fellow at the Institut B für Mechanik, Universität Stuttgart, Germany.
†Communicating author.

motion which are numerically more difficult to solve. The latter, on the contrary, leads to *minimum order* dynamic equations of motion that avoid the difficulties of the DAE, however, is not completely recursive as the loop constraint equations are solved in a conventional way, i.e., using the LU decomposition. In the proposed method, first, closed-loop constraints are solved *recursively* and the recursive dynamic algorithms are obtained similar to the open loop systems. As a result, complete recursive algorithms are obtained in a *uniform* manner rather than decoupling and solving the problem separately, as it is the case with the existing approaches. The method is illustrated with two examples, namely, an one degree of freedom (DOF) four-bar linkage, and a two DOF five-bar planar parallel manipulator.

Keywords: Closed loop, Dynamics, Recursive Algorithm.

I. INTRODUCTION

A dynamic model, i.e., the equations of motion, of a mechanical system is essential for accurate control and realistic prediction of its behavior or simulation, respectively. Today, the task of generating a dynamic model even for a complex systems like vehicles, robotic systems, etc., has become a regular practice. However, control or simulation of an industrial robot based on its dynamic model is still difficult in real time. Thus, there is a thrust on simultaneously improving the modeling techniques and the computing power so that both speed and accuracy improve. Similar needs were felt in the modeling of other systems as well. As a result, different dynamic methods for complex mechanical systems, also known as *Multibody Systems*, have emerged, for example, in [1]–[10]. These methodologies generally give rise to *recursive* $\mathcal{O}(n)$ algorithms for the *inverse dynamics* problems of open loop serial-chain multibody systems, where n is the number of degrees of freedom of the system under study, see e.g., [11]–[13]. Inverse dynamics is defined as *given the trajectory of the end-effector, compute the controlling torques and forces.* This is an essential step in control. On the contrary, *forward dynamics*, a step required in simulation, is defined as *to compute the joint acclerations for given actuator torques and forces.* In simulation, forward dynamics is followed by numerical integration to obtain the joint velocity and position successively.

The traditional formulations [1]–[10] produce *non-recursive* $\mathcal{O}(n^3)$ forward dynamics algorithms, as shown in [15, 16]. However, with special considerations in writing the kinematic constraints and the dynamic equations of motion, it is also possible to generate recursive $\mathcal{O}(n)$ forward dynamics algorithms using the same basic principles, e.g., the Virtual Work, the Newton-Euler or Lagrange equations, see [17]-[28]. Contrary to the $\mathcal{O}(n^3)$ forward dynamics algorithms, the recursive $\mathcal{O}(n)$ algorithms are apparently advantageous if n is large, normally, $n > 7$ [29, 30]. Recursive algorithms are, however, less sensitive to high frequency disturbance due to the discretization applied in numerical integration schemes, which is evident from the studies made in Ref. [31]. As a result, the total simulation time that includes the times for both forward dynamics and numerical integration may be less. Moreover, the recursive algorithms provide physical interpretations like *articulated-body-inertia* [17], *state transition matrix* [21], *twist propagation matrix* [32], etc., which are helpful in debugging the associated computer programs and study the influences of inertia parameters. Furthermore, the recursive algorithms form the basis for the development of *parallel* $\mathcal{O}[log(N)]$ forward dynamics algorithms [33]-[37], where N is the total number of bodies. The latter require special computer hardware, namely, parallel computer processors. Due to the above benefits, the recursive $\mathcal{O}(n)$ algorithms are gaining popularity in multibody system dynamics.

Another consideration in simulation is numerical stability during integration. In this context, a *minimum order* representation of the dynamic equations, i.e., in terms of the independent generalized coordinates, leading to ordinary differential equations is always preferable over alternative representation of the dynamic systems using differential algebraic equations [8, 38, 39]. The minimum order representation is also useful from control point of view, as the existing wealth of control results for open loop systems can be extended to closed loop systems [40].

Now, realizing the benefits of the *recursive* and *minimum order* formulations, an attempt is made in this paper to obtain a *recursive minimum order* algorithm for closed loop systems, which will be suitable for both inverse and forward dynamics required in control and simulation, respectively. Whereas, such a representation for open loop systems is available, e.g., in [17, 19, 20, 32], the same for the closed loop systems is not yet known to exist. For example, in [18, 20, 24, 28], the recursive algorithms for closed loop systems in non-minimal order are obtained, whereas the non-recursive minimum order formulations are reported,

e.g., in [34, 38, 41]. In this paper, the recursive forward dynamics formulation for the serial manipulators, reported in [27], is extended to closed loop systems. The key issue here is to find equivalent decoupled natural orthogonal complement matrices that will allow one to *analytically* write the elements of the generalized inertia matrix (GIM) associated with the dynamic equations of motion of the closed loop system. Once the analytical expression of the GIM is possible, the forward dynamics algorithm of [27, 32] for serial chain systems can be used directly.

This paper is organized as follows: Section II outlines the exisiting dynamic formulations of closed loop systems; Section III introduces some related definitions; Section IV presents the new kinematic constraint formulation for closed loop systems; Section V shows how to obtain the dynamic equations of motion, whereas Section VI report some results for an one degree of freedom four-bar linkage, and a two degrees of freedom five-bar planar parallel manipulator. Finally, concluding remarks are given in Section VII.

II. EXISTING FORMULATIONS

In this section, we review some formulations towards the development of the *minimum order recursive* dynamic formulation of the closed loop systems. As pointed out in the Introduction, we have not found any such algorithm. What is available in the literature, can be broadly classified as either *recursive* DAE (differential algebraic equations) or *non-recursive minimum order* dynamic equations. For example, the algorithms by [20, 23, 26, 28] and others write the equations of motion of a closed loop system in the following manner:

- Cut the closed chains of the system to open into a serial or tree structure;
- Introduce Lagrange multipliers to substitute for the cut joints;
- Use a recursive scheme for the open chain system, e.g., [17, 19] etc., to obtain a recursive algorithm.

The above formulation does not cause any difficulty in inverse dynamics. However, in forward dynamics and simulation, one has to deal with a set of differential algebraic equations of motion, which are difficult to solve and violates kinematic constraints.

On the contrary, the formulations by [38, 40, 41] follow the steps given below:

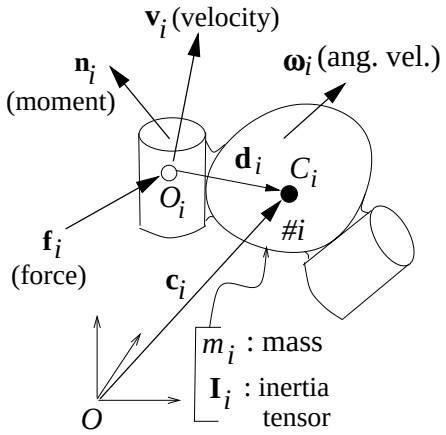

Fig. 1 Motion of the ith body.

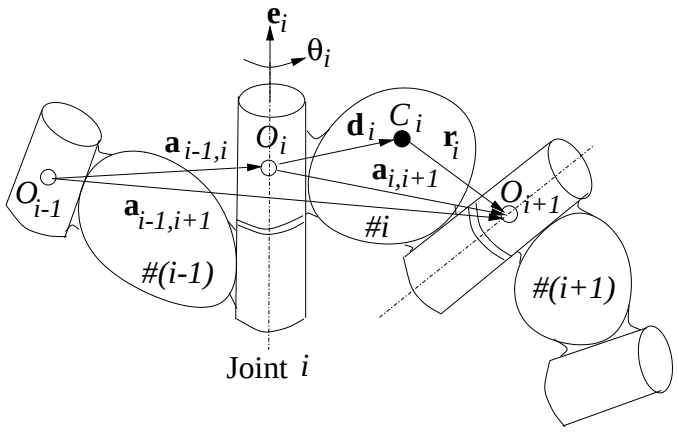

Fig. 2 A coupled system of three bodies.

- Open the closed chains to make it an open chain or tree structure, as above;
- Write the equations of motion of the new system recursively in terms of a larger number of generalized coordinates [17, 19];
- Since the above generalized coordinates are not independent for the original closed loop system, the loop closure conditions are introduced. This enables one to write the larger set of generalized coordinates in terms of an independent set. This step is, normally, done using a numerical method, e.g., using LU decomposition [42]. As a result the algorithm remains no longer recursive;
- Transform the equations of motion of the open/tree system to the closed system, which is also not recurisve.

Due to the last two steps above, the advantages of the recursive algorithm in the second step are lost. In this connection, [4] uses Lagrange's penalty formulation, similar to the numerical stabilization method, which requires iterative solution for the coeffecients of the penalty function.

III. KINEMATICS AND DYNAMICS DEFINITIONS

In this section, some definitions associated with the kinematics and dynamics of the rigid body motion copuled by kinematic pairs or joints are introduced, which will be used throughout this paper. Referring to the free motion of the ith rigid body, as shown in Fig. 1, the following terms are defined:

$\mathbf{t}_i$ and $\mathbf{n}_i$: the six-dimensional vectors of *twist* and *wrench* of the ith link, i.e.,

$$\mathbf{t}_i \equiv \begin{bmatrix} \boldsymbol{\omega}_i \\ \mathbf{v}_i \end{bmatrix} \qquad \text{and} \quad \mathbf{w}_i \equiv \begin{bmatrix} \mathbf{n}_i \\ \mathbf{f}_i \end{bmatrix} \tag{1}$$

where $\boldsymbol{\omega}_i$ and $\mathbf{v}_i$ are, respectively, the three-dimensional vectors of angular velocity and the linear velocity of the *origin point*, O_i, of the ith body where it is coupled with the $(i-1)$st body, as indicated in Fig. 2. Note here that any point on the ith body can be chosen for the definition of the twist and wrench, e.g., the mass-center, C_i, [3]. Here, point O_i is chosen to obtain simplified expressions for the *twist propagation* matrix, $\mathbf{A}_{i,i-1}$, and the *joint rate propagation* vector, $\mathbf{p}_i$, introduced later in eq.(7a). The vectors, $\mathbf{n}_i$ and $\mathbf{f}_i$, are the three-dimensional vectors of the moment about O_i and the force at O_i, respectively.

$\mathbf{W}_i$ and $\mathbf{M}_i$: the 6×6 matrices of *angular velocity* and *mass* of the ith body, respectively, namely,

$$\mathbf{W}_i \equiv \begin{bmatrix} \boldsymbol{\Omega}_i & \mathbf{O} \\ \mathbf{O} & \boldsymbol{\Omega}_i \end{bmatrix} \quad \text{and} \qquad \mathbf{M}_i \equiv \begin{bmatrix} \mathbf{I}_i & m_i\mathbf{D}_i \\ -m_i\mathbf{D}_i & m_i\mathbf{1} \end{bmatrix} \tag{2}$$

where $\boldsymbol{\Omega}_i$ is the 3×3 cross-product tensor associated with vector $\boldsymbol{\omega}_i$. When operateing on any three-dimensional Cartesian vector, $\mathbf{x}$, the tensor, $\boldsymbol{\Omega}_i$, results in a cross-product vector, i.e., $\boldsymbol{\Omega}_i\mathbf{x} \equiv \boldsymbol{\omega}_i \times \mathbf{x}$. Moreover, $\mathbf{I}_i \equiv \mathbf{I}_i^C - m_i\mathbf{D}_i^2$, is the 3×3 inertia tensor about point, O_i, where $\mathbf{I}_i^C$ being the inertia tensor about the mass center of the ith body, C_i. Furthermore, m_i is the mass of the ith body, and the cross-product tensor $\mathbf{D}_i$ is associated with the vector, $\mathbf{d}_i$, that represents the position of the mass center C_i with respect to origin point, O_i, as shown in Figs. 1 and 2. Note in eq.(2), $\mathbf{O}$ and $\mathbf{1}$ are the 3×3 matrix of zeros and the identity matrix, respectively. Henceforth, their dimensions should be understood as of dimensions that are compatible with the dimensions of the matrix expressions where they appear.

Referring to the pair of bodies, say, $\#(i-1)$ and $\#i$, coupled by the ith joint, as shown in Fig. 2,

$\mathbf{a}_{i,i+1}$: the three-dimensional Cartesian vector denoting the position of the origin of the $(i+1)$st body, O_{i+1}, with respect to the ith one, O_i. From Fig. 2, vector $\mathbf{a}_{i,i+1}$ is equal to

$$\mathbf{a}_{i,i+1} = \mathbf{d}_i + \mathbf{r}_i \tag{3}$$

where $\mathbf{d}_i$ and $\mathbf{r}_i$ are also the three-dimensional Cartesian vectors representing, respectively, the mass center of ith body, C_i, with respect to its origin, O_i, and the origin of the $(i+1)$st body, O_{i+1}, with respect to C_i.

θ_i: the displacement of the ith joint. According to Fig. 2 that depicts a revolute joint, θ_i is the joint angle about the axis parallel to the unit vector $\mathbf{e}_i$. In case of a prismatic joint, θ_i is the joint displacement along the axis parallel to $\mathbf{e}_i$. Higher degrees of freedom joints, like universal, cylindrical, spherical, etc., are not treated separately, as they are kinematically equivalent to the combination of revolute and/or prismatic joints. For example, a universial joint is equivalent to two revolute joints whose axes are intersecting.

Finally, referring to the parallel type closed loop system, Fig. 3(a), the system is assumed to have a fixed platform, namely, the *base*, and a moving platform, i.e., the *end effector*. As indicated in Fig. 3(a), the end effector is connected to the base by n legs, which has m rigid bodies, denoted by $\#1, \cdots, \#m$, coupled by m one degree of freedom joints, denoted by $1, \cdots, m$, Fig. 3(b). Note the end-effector represents n rigidly connected end bodies of all the legs, as indicated in Fig. 3(a). Referring to Fig. 3(a), other definitions are as follows:

$\mathbf{t}$ and $\mathbf{w}$: the $6mn$-dimensional vectors of generalized twist and wrench, respectively, i.e.,

$$\mathbf{t} \equiv \begin{bmatrix} \mathbf{t}_1^{(1n)} \\ \vdots \\ \mathbf{t}_m^{(1n)} \end{bmatrix} \text{ and } \mathbf{w} \equiv \begin{bmatrix} \mathbf{w}_1^{(1n)} \\ \vdots \\ \mathbf{w}_m^{(1n)} \end{bmatrix}, \quad \text{where} \quad \mathbf{t}_i^{(1n)} \equiv \begin{bmatrix} \mathbf{t}_i^{(1)} \\ \vdots \\ \mathbf{t}_i^{(n)} \end{bmatrix} \text{ and } \mathbf{w}_i^{(1n)} \equiv \begin{bmatrix} \mathbf{w}_i^{(1)} \\ \vdots \\ \mathbf{w}_i^{(n)} \end{bmatrix} \tag{4}$$

Vectors $\mathbf{t}_i^{(1n)}$ and $\mathbf{w}_i^{(1n)}$, for $i = 1, \cdots, m$, being the $6n$-dimensional vectors, whose six dimensional vector components, $\mathbf{t}_i^{(l)}$ and $\mathbf{w}_i^{(l)}$, for $l = 1, \cdots, n$, respectively, are the twist and wrench of the ith body belonging to the lth leg, as defined in eq.(1).

$\mathbf{W}$ and $\mathbf{M}$: the $6mn \times 6mn$ matrices of generalized angular velocity and mass, respectively, namely,

$$\mathbf{W} \equiv \begin{bmatrix} \mathbf{W}_1^{(1n)} & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \mathbf{W}_m^{(1n)} \end{bmatrix} \quad \text{and} \quad \mathbf{M} \equiv \begin{bmatrix} \mathbf{M}_1^{(1n)} & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \mathbf{M}_m^{(1n)} \end{bmatrix} \tag{5a}$$

where the $6n \times 6n$ matrices, $\mathbf{W}_i^{(1n)}$ and $\mathbf{M}_i^{(1n)}$, for $i = 1, \cdots, m$, are defined as

$$\mathbf{W}_i^{(1n)} \equiv \begin{bmatrix} \mathbf{W}_i^{(1)} & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \mathbf{W}_i^{(n)} \end{bmatrix} \quad \text{and} \quad \mathbf{M}_i^{(1n)} \equiv \begin{bmatrix} \mathbf{M}_i^{(1)} & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \mathbf{M}_i^{(n)} \end{bmatrix} \tag{5b}$$

in which the 6×6 matrices, $\mathbf{W}_i^{(l)}$ and $\mathbf{M}_i^{(l)}$, for $l = 1, \cdots, n$, respectively, for the ith body of the lth leg are defined in eq.(2). In eqs.(5a) and (5b), the off-diagonal block elements of the matrices are zeros.

$\dot{\boldsymbol{\theta}}$: the n-dimensional vector of independent joint rates. In controlled mechanical systems, the number of driving actuators are normally taken as equal to the degrees of freedom (DOF) of the system under consideration, i.e., no actuation redundancy. Thus, actuated joint motions can be considered independent. For the system shown in Fig. 3, one joint per leg is assumed to be actuated. Hence, the DOF of the parallel closed loop system under study is n, and the independent generalized joint rate vector, $\dot{\boldsymbol{\theta}}$, can be taken as

$$\dot{\boldsymbol{\theta}} \equiv [\dot{\theta}_k^{(1)}, \cdots, \dot{\theta}_k^{(n)}]^T \tag{6}$$

where $\theta_k^{(l)}$, for $1 \leq k \leq m$; $l = 1, \cdots, n$, is the displacement of the actuated joint, k, of leg l.

IV. KINEMATIC CONSTRAINT FORMULATION

In this paper, we consider closed loop systems with parallel structure, as shown in Fig. 3(a). These types of systems are also known as *parallel manipulators*, which find applications as Flight Simulators, Hexapod Machine, etc. They are preferred over serial manipulators or conventional machine tool structures, mainly, due to their higher force-to-weight ratio, rigidity of the structure, and accuracy of motion. In order to derive the kinematic relations, the following assumptions are made:

- The system is non-redundant, i.e., the degrees of freedom (DOF) is equal to the number of driving actuators;
- Number of legs is equal to the DOF;
- Each leg has only one actuator;

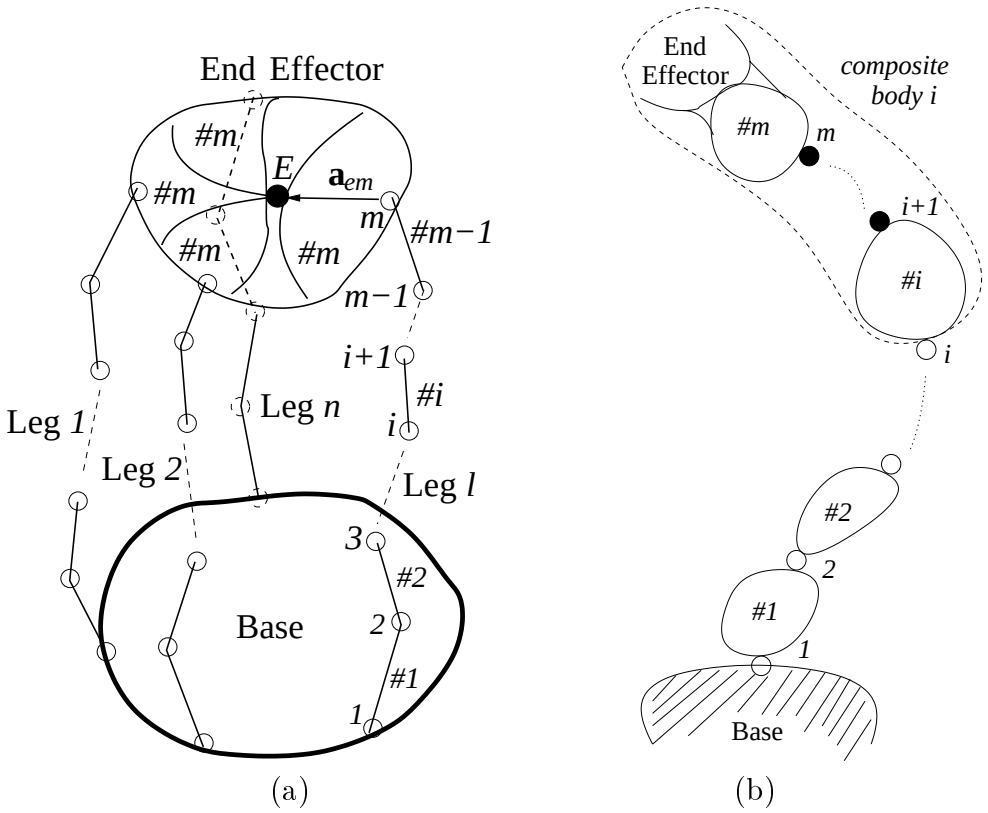

Fig. 3 (a) Parallel closed loop system; (b) The lth leg.

- Joint positions are known either from the sensor data, or from a suitable algorithm, e.g., using the one proposed in [43];
- No friction or damping;
- Gravity is taken into account by adding negative acceleration due to gravity to the base body of the system, as proposed in [15].

The most important steps in writing the *minimum order* dynamic equations of motion is the ability to write the generalized twist of the system, $\mathbf{t}$ of eq.(4), as a linear transformation of the independent joint rates, namely, $\dot{\boldsymbol{\theta}}$ defined by eq.(6) [3, 16, 32]. The above miminum order representation can lead to *recursive* inverse and forward dynamics algorithms if the associated transformation matrix is represented as decopuled natural orthogonal complement (DeNOC) matrix, as shown in [27, 32] for the open loop mechanical systems. In this paper, the DeNOC matrices are derived for the closed loop parallel systems. This, first, requires the recursive solution of the passive or unactuated joint rates, which is obtained next.

A. Recursive Solution of Unactuated Joint Rates

Consider the rigid body motions of the lth leg, as shown in Fig. 3(b),

1. Write the twist of the ith body from the previous one, i.e.,

$$\mathbf{t}_i = \mathbf{A}_{i,i-1}\mathbf{t}_{i-1} + \mathbf{p}_i\dot{\theta}_i \tag{7a}$$

where superscripts to indicate the legs are omitted to reduce the clumsiness. Moreover, the terms, $\mathbf{A}_{i,i-1}$ and $\mathbf{p}_i$, are the 6×6 *twist propagation matrix* and the six-dimensional *joint-rate propogation vector*, respectively [32], whereas the scalar, $\dot{\theta}_i$, is the time derivative of the ith joint displacement, θ_i. The matrix $\mathbf{A}_{i,i-1}$ and the vector $\mathbf{p}_i$ are given as

$$\mathbf{A}_{i,i-1} \equiv \begin{bmatrix} \mathbf{1} & \mathbf{O} \\ \mathbf{\Lambda}_{i,i-1} & \mathbf{1} \end{bmatrix}; \text{ and } \mathbf{p}_i \equiv \begin{bmatrix} \mathbf{e}_i \\ \mathbf{0} \end{bmatrix} : \text{if revolute, or } \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_i \end{bmatrix} : \text{if prismatic} \tag{7b}$$

in which $\mathbf{\Lambda}_{i,i-1}$ is the cross-product tensor associated with the vector, $\mathbf{a}_{i,i-1} \equiv -\mathbf{a}_{i-1,i}$. The vector, $\mathbf{a}_{i-1,i}$, representing the position of the origin of the ith body, O_i, from the origin of the $(i-1)$st body, O_{i-1}, is shown in Fig. 2. Moreover, the vector, $\mathbf{e}_i$, for a revolute joint, is shown in Fig. 2. Note that, for the successively coupled rigid bodies, say, $\#(i-1)$, $\#i$, and $\#(i+1)$, the twist propagation matrices obey the following properties:

$$\mathbf{A}_{i-1,i}\mathbf{A}_{i,i+1} = \mathbf{A}_{i-1,i+1}; \quad \mathbf{A}_{ii} = \mathbf{1}; \quad \text{and} \quad \mathbf{A}_{i-1,i}^{-1} = \mathbf{A}_{i,i-1}, \text{ or } \mathbf{A}_{i,i+1}^{-1} = \mathbf{A}_{i+1,i} \tag{7c}$$

where $\mathbf{A}_{i-1,i+1}$ is the function of the vector, $\mathbf{a}_{i-1,i+1}$, denoting the distance of the origin of the $(i+1)$st body, O_{i+1}, from the origin of the $(i-1)$st one, O_{i-1}, is shown in Fig. 2. Hence, $\mathbf{A}_{ii}$ is an identity matrix, as the distance of the origin of ith body from itself vanish.

2. Using eqs.(7a) and (7b), for $i = 1, \cdots, m$, the twist of the moving platform or the end-effector, $\mathbf{t}_e$, can be obtained as

$$\mathbf{t}_e = \mathbf{A}_{em}\mathbf{t}_m \tag{8}$$

where the velocity component of $\mathbf{t}_e$ is defined with respect to point E of the end-effector, as indicated in Fig. 3. Moreover, the 6×6 matrix, $\mathbf{A}_{em}$, defined similar to

$\mathbf{A}_{i,i-1}$ of eq.(7b), is the function of the vector, $\mathbf{a}_{em}$, joining the mth joint with point E, as shown in Fig. 3(a) along leg l.

3. Since all the legs are meeting at the end-effector, its twist is taken as constraint twist, namely, eq.(8). Unactuated joint rates are then solved as follows: If the mth joint is not acutated, its joint rate, $\dot{\theta}_m$, is solved by substituting the expression of $\mathbf{t}_m$ from eq.(7a), in terms of $\mathbf{t}_{m-1}$ and $\dot{\theta}_m$, into eq.(8) as

$$\mathbf{t}_e = \mathbf{A}_{em}(\mathbf{A}_{m,m-1}\mathbf{t}_{m-1} + \mathbf{p}_m\dot{\theta}_m) \tag{9a}$$

which, after premultiplication by $\tilde{\mathbf{p}}_m^T$, results in

$$\dot{\theta}_m = \frac{1}{\delta_m}\tilde{\mathbf{p}}_m^T(\mathbf{t}_e - \mathbf{A}_{e,m-1}\mathbf{t}_{m-1}), \quad \text{where} \quad \tilde{\mathbf{p}}_m \equiv \mathbf{A}_{em}\mathbf{p}_m, \quad \delta_m \equiv \tilde{\mathbf{p}}_m^T\tilde{\mathbf{p}}_m \tag{9b}$$

and, according to the first property of eq.(7c), $\mathbf{A}_{e,m-1} = \mathbf{A}_{em}\mathbf{A}_{m,m-1}$.

4. Substituting the expression of $\dot{\theta}_m$, eq.(9b), into eq.(9a), a new constraint, similar to eq.(8), is formulated as

$$\mathbf{\Phi}_m\mathbf{t}_e = \mathbf{\Phi}_m\mathbf{A}_{e,m-1}\mathbf{t}_{m-1}, \quad \text{where} \qquad \mathbf{\Phi}_m \equiv \mathbf{1} - \frac{1}{\delta_m}\tilde{\mathbf{p}}_m\tilde{\mathbf{p}}_m^T \tag{10a}$$

The 6×6 symmetric matrix, $\mathbf{\Phi}_m$, projects the twists along the axis perpendicular to $\tilde{\mathbf{p}}_m$, which has the following property:

$$\mathbf{\Phi}_m^T\mathbf{\Phi}_m = \mathbf{\Phi}_m\mathbf{\Phi}_m^T = \mathbf{\Phi}_m \tag{10b}$$

Equation (10a) is used to solve for $\dot{\theta}_{m-1}$, similar to that of $\dot{\theta}_m$ from eq.(9a).

5. Continuing the above step successively, it is possilbe to write a constraint to solve for the ith joint rate, $\dot{\theta}_i$, as

$$\mathbf{\Phi}_{i+1}\mathbf{t}_e = \mathbf{\Phi}_{i+1}\mathbf{A}_{ei}\mathbf{t}_i \tag{11a}$$

which gives

$$\dot{\theta}_i = \frac{1}{\delta_i}\tilde{\mathbf{p}}_i^T(\mathbf{t}_e - \mathbf{A}_{e,i-1}\mathbf{t}_{i-1}) \tag{11b}$$

where the six diminsional vector, $\tilde{\mathbf{p}}_i$, and the scalar, δ_i, are defined as

$$\tilde{\mathbf{p}}_i \equiv \mathbf{\Phi}_{i+1}\mathbf{A}_{ei}\mathbf{p}_i, \quad \text{and} \qquad \delta_i \equiv \tilde{\mathbf{p}}_i^T\tilde{\mathbf{p}}_i \tag{11c}$$

in which the 6×6 symmetric matrix, $\mathbf{\Phi}_{i+1}$, can be obtained recursively, i.e.,

$$\mathbf{\Phi}_{i+1} \equiv \mathbf{\Phi}_{i+2} - \frac{1}{\delta_{i+1}} \tilde{\mathbf{p}}_{i+1} \tilde{\mathbf{p}}_{i+1}^T, \quad \text{where} \qquad \mathbf{\Phi}_{m+1} = \mathbf{1} \tag{11d}$$

Matrix $\mathbf{\Phi}_{i+1}$, being the projection matrix, also satisfies the properties given in eq.(10b). Equation (11b) is used to solve for the unactuated joint rates from $i = m$ to the one above the acutated joint, say, k, which need not be solved. Then, for $i = k-1, \cdots, 1$, eqs.(11a) to (11d), need modifications, which are given below:

$$\mathbf{\Phi}_{i+2} \mathbf{t}_e{}' = \mathbf{\Phi}_{i+2} \mathbf{A}_{ei} \mathbf{t}_i, \quad \text{where} \qquad \mathbf{t}_e{}' \equiv \mathbf{t}_e - \mathbf{A}_{ek} \mathbf{p}_k \dot{\theta}_k; \tag{12a}$$

$$\dot{\theta}_i = \frac{1}{\delta_i} \tilde{\mathbf{p}}_i^T (\mathbf{t}_e{}' - \mathbf{A}_{e,i-1} \mathbf{t}_{i-1}); \tag{12b}$$

$$\tilde{\mathbf{p}}_i \equiv \mathbf{\Phi}_{i+2} \mathbf{A}_{ei} \mathbf{p}_i, \quad \text{and} \qquad \delta_i \equiv \tilde{\mathbf{p}}_i^T \tilde{\mathbf{p}}_i; \tag{12c}$$

$$\mathbf{\Phi}_{i+2} \equiv \mathbf{\Phi}_{i+3} - \frac{1}{\delta_{i+1}} \tilde{\mathbf{p}}_{i+1} \tilde{\mathbf{p}}_{i+1}^T \tag{12d}$$

6. Since all the joint rates can be obtained only if the end-effector twist, $\mathbf{t}_e$, is known, it needs to be expressed in terms of the independent actuated joint rates. This is done as follows: Using $i = 0$ in eq.(12a), the following constraint is obtained:

$$\mathbf{\Phi}_2 \mathbf{t}_e{}' = \mathbf{0}, \quad \text{since} \quad \mathbf{t}_0 = \mathbf{0} \tag{13a}$$

That is the body #0 is fixed to the base. Substituting the expression for $\mathbf{t}_e^{'}$, eq.(12a), into eq.(13a), the following is obtained:

$$\mathbf{\Phi}_2 \mathbf{t}_e = \mathbf{\Phi}_2 \mathbf{A}_{ek} \mathbf{p}_k \dot{\theta}_k \tag{13b}$$

Writing eq.(13b) for all legs, i.e., $l = 1, \cdots, n$, and adding them, a relation of $\mathbf{t}_e$ in terms of the actuated joint rates, $\dot{\boldsymbol{\theta}}$, as defined in eq.(6), is obtained as follows:

$$\mathbf{\Phi} \mathbf{t}_e = \sum_{l=1}^{n} [\mathbf{\Phi}_2 \mathbf{A}_{ek} \mathbf{p}_k \dot{\theta}_k]^{(l)}, \quad \text{where} \qquad \mathbf{\Phi} \equiv \sum_{l=1}^{n} \mathbf{\Phi}_2^{(l)} \tag{14a}$$

The superscript (l) implies that each term inside $[\,\cdot\,]$ has the same superscript associated with the leg number. From eq.(14a), $\mathbf{t}_e$, is solved as

$$\mathbf{t}_e = \mathbf{J} \dot{\boldsymbol{\theta}}, \quad \text{where} \qquad \mathbf{J} \equiv \mathbf{\Phi}^{-1} [[\mathbf{\Phi}_2 \mathbf{A}_{ek} \mathbf{p}_k]^{(1)}, \cdots, [\mathbf{\Phi}_2 \mathbf{A}_{ek} \mathbf{p}_k]^{(n)}] \tag{14b}$$

The $6 \times n$ matrix, $\mathbf{J}$, is the well-known Jacobian matrix, which exists as long as the 6×6 symmetric matrix, $\mathbf{\Phi}$, is non-singular.

From the above derivations, i.e., eqs.(11a)-(14b), it is clear that, except the inverseion of the 6×6 symmetric matrix, $\mathbf{\Phi}$, that can be done with fixed number of operations, the unactuated joint rates can be calculated *recursively* using eq.(11b) or eq.(12b). The recursive solution is possible because the projection matrices of eq.(11d) or (12d) can be computed recursively. Hence, the major problem in developing a recursive algorithm for the closed loop systems is overcome. This is the key contribution of this paper. Note here that the above projection matrices are similar to those reported in [44] without any outline of how to compute them.

B. Derivation of the DeNOC Matrices

The derivation of the decoupled natural orthogonal complement (DeNOC) matrices, which was introduced in Ref. [27] for the open loop system, e.g., serial manipulators, for the closed loop parallel system, Fig. 3(a), is shown in the following:

1. Knowing the analytical expressions of the joint rates from eq.(11b) or eq.(12b), the twist of all the rigid bodies in leg l—Leg indications with superscripts are omitted to reduce the clumsiness of the expressions—are written as

$$\mathbf{t}_1 = \mathbf{p}_1\dot{\theta}_1 = \frac{1}{\delta_1}\mathbf{p}_1\tilde{\mathbf{p}}_1^T\mathbf{t}_e{}' \tag{15a}$$

$$\mathbf{t}_2 = \mathbf{A}_{21}\mathbf{t}_1 + \mathbf{p}_2\dot{\theta}_2 = \mathbf{\Psi}_2\mathbf{A}_{21}\mathbf{t}_1 + \frac{1}{\delta_2}\mathbf{p}_2\tilde{\mathbf{p}}_2^T\mathbf{t}_e{}' \tag{15b}$$

$$\vdots$$

$$\mathbf{t}_k = \mathbf{A}_{k,k-1}\mathbf{t}_{k-1} + \mathbf{p}_k\dot{\theta}_k, \quad \text{i.e.,} \quad \mathbf{\Psi}_k = \mathbf{1} \tag{15c}$$

$$\mathbf{t}_{k+1} = \mathbf{A}_{k+1,k}\mathbf{t}_k + \mathbf{p}_{k+1}\dot{\theta}_{k+1} = \mathbf{\Psi}_{k+1}\mathbf{A}_{k+1,k}\mathbf{t}_k + \frac{1}{\delta_{k+1}}\mathbf{p}_{k+1}\tilde{\mathbf{p}}_{k+1}^T\mathbf{t}_e \tag{15d}$$

$$\vdots$$

$$\mathbf{t}_m = \mathbf{A}_{m,m-1}\mathbf{t}_{m-1} + \mathbf{p}_m\dot{\theta}_m = \mathbf{\Psi}_m\mathbf{A}_{m,m-1}\mathbf{t}_{m-1} + \frac{1}{\delta_m}\mathbf{p}_m\tilde{\mathbf{p}}_m^T\mathbf{t}_e \tag{15e}$$

where the 6×6 matrix, $\mathbf{\Psi}_i$, for $i = 2, \cdots, m$ $i \neq k$, is non-symmetric and defined as

$$\mathbf{\Psi}_i \equiv \mathbf{1} - \frac{1}{\delta_i}\mathbf{p}_i\tilde{\mathbf{p}}_i^T\mathbf{A}_{ei}, \quad \text{where} \quad \mathbf{\Psi}_k = \mathbf{1} \tag{15f}$$

The matrix, $\mathbf{\Psi}_k$, is the coefficient matrix of $\mathbf{A}_{k,k-1}$ which is associated with the twist of the body, k, that is driven directly by the actuator placed on the kth joint. Hence, $\mathbf{\Psi}_k$ is identity.

2. Substituting the expressions for $\mathbf{t}_e^{'}$ and $\mathbf{t}_e$, from eqs.(12a) and (14b), respectively, into eqs.(15a)–(15e) for all legs, i.e., $l = 1, \cdots, n$, one obtains

$$\mathbf{t} = \mathbf{A}\mathbf{t} + \mathbf{N}_c\mathbf{N}_d\dot{\boldsymbol{\theta}} \tag{16a}$$

where $\mathbf{t}$ is the $6mn$-dimensional generalized twist vector written according to the definition given by eq.(4), and the n-dimensional vector, $\dot{\boldsymbol{\theta}}$, is defined in eq.(6). Moreover, the $6mn \times 6mn$ lower block band-diagonal matrix, $\mathbf{A}$, the $6mn \times 6n$ full matrix, $\mathbf{N}_c$, and the $6n \times n$ block diagonal matrix, $\mathbf{N}_d$, are defined as

$$\mathbf{A} \equiv \begin{bmatrix} \mathbf{O} & & & & & \mathbf{O} \\ \tilde{\mathbf{A}}_{21} & \mathbf{O} & & & & \\ & \ddots & \ddots & & & \\ & & \tilde{\mathbf{A}}_{k,k-1} & \mathbf{O} & & \\ & & & \ddots & \ddots & \\ \mathbf{O} & & & & \tilde{\mathbf{A}}_{m,m-1} & \mathbf{O} \end{bmatrix}^{(1n)} \tag{16b}$$

$$\mathbf{N}_c \equiv \begin{bmatrix} \mathbf{U}\bar{\mathbf{A}}_{ek} - \mathbf{A}_{ek} \\ \vdots \\ \mathbf{U}\bar{\mathbf{A}}_{ek} - \mathbf{A}_{ek} \\ \mathbf{1} \\ \mathbf{U}\bar{\mathbf{A}}_{ek} \\ \vdots \\ \mathbf{U}\bar{\mathbf{A}}_{ek} \end{bmatrix}^{(1n)} \quad \text{and} \quad \mathbf{N}_d \equiv \begin{bmatrix} \mathbf{p}_k^{(1)} & & & & \mathbf{0} \\ & \ddots & & & \\ & & \mathbf{p}_k^{(k)} & & \\ & & & \ddots & \\ \mathbf{0} & & & & \mathbf{p}_k^{(n)} \end{bmatrix} \tag{16c}$$

where the superscript $(1n)$ of the matrix notation imply that the non-zero element matrices are to be read with the superscript $(1n)$, namely, $\tilde{\mathbf{A}}_{i,i-1}^{(1n)}$, for $i = 2, \cdots, m$, $\bar{\mathbf{A}}_{ek}^{(1n)}$ and $\mathbf{A}_{ek}^{(1n)}$, which are defined similar to the definitions given in eq.(5b), i.e, $\tilde{\mathbf{A}}_{i,i-1}^{(1n)} \equiv \text{diag}[\tilde{\mathbf{A}}_{i,i-1}^{(1)}, \cdots, \tilde{\mathbf{A}}_{i,i-1}^{(1n)}]$, etc. In eqs.(16b) and (16c), $\mathbf{O}$ and $\mathbf{1}$ represent the $6n \times 6n$ zero and identity matrices, respectively, whereas $\mathbf{0}$ in the expression of $\mathbf{N}_d$ is the six-dimensional zero vector. Other empty cells imply zeros, and the $6n \times 6n$ constant matrix, $\mathbf{U}$, is full of identity matrices, i.e.,

$$\mathbf{U} \equiv \begin{bmatrix} \mathbf{1} & \cdots & \mathbf{1} \\ \vdots & \ddots & \vdots \\ \mathbf{1} & \dots & \mathbf{1} \end{bmatrix} \tag{16d}$$

The 6×6 matrices, $\tilde{\mathbf{A}}_{i,i-1}^{(l)}$ and $\bar{\mathbf{A}}_{ek}^{(l)}$, for $l = 1, \cdots, n$, are defined as

$$\tilde{\mathbf{A}}_{i,i-1}^{(l)} \equiv [\boldsymbol{\Psi}_i \mathbf{A}_{i,i-1}]^{(l)}, \quad \text{and} \quad \bar{\mathbf{A}}_{ek}^{(l)} \equiv \boldsymbol{\Phi}^{-1}[\boldsymbol{\Phi}_2 \mathbf{A}_{ek}]^{(l)} \tag{16e}$$

From the above defintion of $\tilde{\mathbf{A}}^{(l)}_{i,i-1}$, note that the matrix, $\tilde{\mathbf{A}}_{k,k-1}$, appearing in the kth block row of $\mathbf{A}$, eq.(16b), and associated with the actuated joint is simply, $\tilde{\mathbf{A}}_{k,k-1} = \mathbf{A}_{k,k-1}$, since $\mathbf{\Psi}_k = \mathbf{1}$, as defined in eq.(15f). Also, the identity matrix, $\mathbf{1}$, in the k-th block row of $\mathbf{N}_c$, eq.(16c), is associated with the actuated joints.

3. From eq.(16a), the generalized twist is then written as,

$$\mathbf{t} = \mathbf{N}_l \mathbf{N}_c \mathbf{N}_d \dot{\boldsymbol{\theta}}, \quad \text{where} \qquad \mathbf{N}_l \equiv (\mathbf{1} - \mathbf{A})^{-1} \tag{17a}$$

The inverse of the $6mn \times 6mn$ matrix, $(\mathbf{1} - \mathbf{A})$, is simple, a lower block triangular matrix, i.e.,

$$\mathbf{N}_l \equiv (\mathbf{1} - \mathbf{A})^{-1} = \begin{bmatrix} \mathbf{1} & & & & & \\ \tilde{\mathbf{A}}_{21} & \mathbf{1} & & & & \\ & & \ddots & & & \\ \tilde{\mathbf{A}}_{k1} & \cdots & \tilde{\mathbf{A}}_{k,k-1} & \mathbf{1} & & \\ \vdots & & \vdots & & \ddots & \\ \tilde{\mathbf{A}}_{m1} & & \cdots & & \tilde{\mathbf{A}}_{m,m-1} & \mathbf{1} \end{bmatrix}^{(1n)} \tag{17b}$$

where $\tilde{\mathbf{A}}_{ij}$ means $\tilde{\mathbf{A}}^{(1n)}_{ij}$, for $i = 2, \cdots, m;\ j = 1, \cdots, i-1$, which are defined similar to eq.(5b), in which $\tilde{\mathbf{A}}^{(l)}_{ij}$, for $l = 1, \cdots, n$, are as follows:

$$\tilde{\mathbf{A}}^{(l)}_{ij} = [\mathbf{\Psi}_i \mathbf{A}_{i-1} \mathbf{\Psi}_{i-1} \mathbf{A}_{i-2} \cdots \mathbf{\Psi}_{j+1} \mathbf{A}_j]^{(l)} \tag{17c}$$

The $6mn \times 6mn$ $\mathbf{N}_l$, the $6mn \times 6n$ $\mathbf{N}_c$, and the $6n \times n$ $\mathbf{N}_d$, are the desired decoupled natural orthogonal complement (DeNOC) matrices for the closed loop system under study. The differences of these matrices compared to those for the serial chain systems are as follows:

- For serial chain systems [27, 32], the DeNOC matrices are two, namely, $\mathbf{N}_l$ and $\mathbf{N}_d$, which are lower block triangular and block diagonal, respectively. Moreover, the matrix, $\mathbf{N}_l$, is a function of $\mathbf{A}_i$ matrices only.

- In case of closed loop systems, the DeNOC matrices are three, i.e., $\mathbf{N}_l$, $\mathbf{N}_c$, and $\mathbf{N}_d$, as in eq.(17a), which are lower block triangular, a full, and a block diagonal, repectively. Moreover, the matrix $\mathbf{N}_l$ is not only the function of $\mathbf{A}_i$ matrices but also of $\mathbf{\Psi}_i$ that takes care of the effect of the unactuated joint rates on the twist of rigid bodies of a leg. Furthermore, the matrix, $\mathbf{N}_c$, accounts for the interaction of all leg motions through the the twist of the end-effector, $\mathbf{t}_e$. Finally, the matrix, $\mathbf{N}_d$, has the same structure as that of serial chain sytems [27, 32].

V. DYNAMIC MODEL

In order to obtain the dynamic model of closed loop parallel systems, the uncoupled Newton-Euler equations of motion of all the rigid bodies in the system are expressed as [29, 32]

$$\mathbf{M}\dot{\mathbf{t}} + \mathbf{WME} = \mathbf{w} \tag{18a}$$

where the $6mn \times 6mn$ generalized angular velocity matrix, $\mathbf{W}$, the generalized mass matrix, $\mathbf{M}$, are defined in eqs.(5a), whereas the $6mn \times 6mn$ constant matrix, $\mathbf{E}$, is defined as

$$\mathbf{E} \equiv \mathrm{diag}[\mathbf{E}_1, \cdots, \mathbf{E}_m]^{(1n)}, \quad \text{where} \qquad \mathbf{E}_i^{(1n)} \equiv \mathrm{diag}[\mathbf{E}_i^{(1)}, \cdots, \mathbf{E}_i^{(n)}] \tag{18b}$$

for $i = 1, \cdots, m;\ l = 1, \cdots, n$, in which the 6×6 matrix, $\mathbf{E}_i^{(l)}$ is simply

$$\mathbf{E}_i^{(l)} \equiv \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{18c}$$

Now, two different problems in dynamics, namely, inverse and forward dynamics, are treated separately in the following subsections.

A. Inverse Dynamics

Inverse dynamics is defined as *given the trajectory of the end-effector, compute the controlling actuated joint torques and forces.* Here it is not necessary to obtain the explicit expression for the dynamic equations of motion. The actuated joint forces and torques are computed as follows: Since the generalized twist lies in the nullspace of the decopuled natural orthogonal complement (DeNOC) matrices, $\mathbf{N} \equiv \mathbf{N}_l\mathbf{N}_c\mathbf{N}_d$, as derived in eqs.(16c) and (17b), premultiplication of its transpose, $\mathbf{N}^T$, with the uncoupled Netwon-Euler equations of motion, eq.(18a), produces the required actuated joint torques and forces, $\boldsymbol{\tau} = \mathbf{N}^T\mathbf{w}$, as done in [16, 32] for the open loop system, i.e.,

$$\boldsymbol{\tau} = \mathbf{N}^T\mathbf{w}, \quad \text{where } \mathbf{N} \equiv \mathbf{N}_l\mathbf{N}_c\mathbf{N}_d \tag{19}$$

Vector, $\boldsymbol{\tau}$, being n-dimensional, which is obtained using the steps shown below, where the term, IDA, in the items stands for Inverse Dynamics Algorithm:

IDA-1 First, compute the $6mn$-dimensional vector, $\tilde{\mathbf{w}} \equiv \mathbf{N}_l^T\mathbf{w}$, where $\tilde{\mathbf{w}}$ is given by

$$\tilde{\mathbf{w}} \equiv [\tilde{\mathbf{w}}_1^T, \cdots, \tilde{\mathbf{w}}_m^T]^{(1n)^T} \tag{20}$$

in which the $6n$-dimensional vectors, $\tilde{\mathbf{w}}_i^{(1n)}$, for $i = 1, \cdots, m$, is defined similar to vector $\mathbf{w}_i^{(1n)}$, eq.(4). The six dimensional vector components of $\tilde{\mathbf{w}}_i^{(1n)}$, $\tilde{\mathbf{w}}_i^{(l)}$, obtained recursively as

$$\tilde{\mathbf{w}}_i^{(l)} = [\mathbf{w}_i + \tilde{\mathbf{A}}_{i+1,i}^T \tilde{\mathbf{w}}_{i+1}]^{(l)}, \quad \text{where} \quad \tilde{\mathbf{w}}_m^{(l)} = \mathbf{w}_m^{(l)} \tag{21}$$

for $i = m, \cdots, 1;\ l = 1, \cdots, n$.

IDA-2 Next, compute the $6n$-dimensional vector, $\bar{\mathbf{w}}_k \equiv \mathbf{N}_c^T \tilde{\mathbf{w}}$, for $l = 1, \cdots, n$, as

$$\bar{\mathbf{w}}_k^{(l)} = \tilde{\mathbf{w}}_k^{(l)} + \bar{\mathbf{A}}_{mk}^{(l)^T} [\tilde{\mathbf{w}}_{(1m)}^{(+)} - \tilde{\mathbf{w}}_k^{(+)}] - \mathbf{A}_{mk}^{(l)^T} \tilde{\mathbf{w}}_{(1,k-1)}^{(l)} \tag{22}$$

where the six-dimensional vectors, $\tilde{\mathbf{w}}_{(1,k-1)}^{(l)}$, $\tilde{\mathbf{w}}_{(1m)}^{(+)}$, and $\tilde{\mathbf{w}}_k^{(+)}$, are defined according eq.(23). Note the subscript, (ij), which denotes summation from body numbers, i to j, whereas the superscript $(+)$ means the summation over leg numbers, $1, \cdots, n$, i.e.,

$$\mathbf{w}_{(ij)}^{(l)} \equiv [\tilde{\mathbf{w}}_i + \tilde{\mathbf{w}}_{i+1} + \cdots + \tilde{\mathbf{w}}_{j-1} + \tilde{\mathbf{w}}_j]^{(l)}, \quad \text{and} \qquad \mathbf{w}_{ij}^{(+)} \equiv \tilde{\mathbf{w}}_{ij}^{(1)} + \cdots + \tilde{\mathbf{w}}_{ij}^{(n)} \tag{23}$$

Vector $\tilde{\mathbf{w}}_k^{(+)}$ being also defined as $\mathbf{w}_k^{(+)} \equiv \mathbf{w}_k^{(1)} + \cdots + \mathbf{w}_k^{(n)}$.

IDA-3 Finally, the actuated joint torques, $\boldsymbol{\tau}$=$\mathbf{N}_d^T \bar{\mathbf{w}}_k$, are obtained as

$$\boldsymbol{\tau}_k^{(l)} = [\mathbf{p}_k^T \bar{\mathbf{w}}_k]^{(l)} \tag{24}$$

for $l = 1, \cdots, n$. As per as the nature of the computation is concerned, it is recursive because step IDA-1 is computed recursevely for m joints in a leg, as shown in eq.(21), and steps IDA-2 and IDA-3 require a fixed number of operations for n legs. However, no computational count—a traditional way to compare the efficiency of algorithms—is done here, as the objective of this paper is to investigate whether a recursive minimal order dynamic models for closed loop systems is possible or not. The efficiency of the algorithm will be studied separately and will be reported in future.

B. Forward Dynamics

In forward dynamics, which is defined as *to compute the joint acclerations for given actuator torques and forces*, it is necessay to find an explicit expression for the generalized inertia

matrix (GIM) associated with the equations of motion of the system at hand [3, 16, 27, 32] that are expressed as

$$\mathbf{I}\ddot{\boldsymbol{\theta}} + \mathbf{C}\dot{\boldsymbol{\theta}} = \boldsymbol{\tau} \tag{25}$$

where the $n \times n$ matrices, $\mathbf{I}$ and $\mathbf{C}$, are the GIM and the generalized matrix of convective inertia terms, respectively, whereas $\boldsymbol{\tau}$, also appears in eq.(24), is the n-dimensional vector of generalized forces due to driving torques and those resulting from gravity, and ohter forces. Note that the explicit evaluation of $\mathbf{C}$ is not required for the forward dynamics, as $(\boldsymbol{\tau} - \mathbf{C}\dot{\boldsymbol{\theta}})$ can be computed together using the above inverse dynamics algorithm while $\ddot{\boldsymbol{\theta}} = \mathbf{0}$ [15]. However, the explicit evaluation of the GIM, $\mathbf{I}$ of eq.(25), is required to solve for the joint accelerations, $\ddot{\boldsymbol{\theta}}$. The GIM is written in a manner similar to the open loop systems [27, 32], as

$$\mathbf{I} \equiv \mathbf{N}^T\mathbf{M}\mathbf{N}, \quad \text{where} \quad \mathbf{N} \equiv \mathbf{N}_l\mathbf{N}_c\mathbf{N}_d \tag{26}$$

in which $\mathbf{I}$ is the $n \times n$ matrix, and the decoupled natural orthogonal complement matrices, $6mn \times 6mn$ $\mathbf{N}_l$, $6mn \times 6n$ $\mathbf{N}_c$, and $6n \times n$ $\mathbf{N}_d$, are shown in eqs.(17b) and (16c). Moreover, the $6mn \times 6mn$ generalized mass matrix, $\mathbf{M}$, is defined in eq.(5a). In this paper, only the derivation of the GIM is shown, which can be directly used by the recursive forward dynamics algorithm propsed in [27, 32]. The GIM, $\mathbf{I}$, is obtained in the following steps, where the term FDA in the items below stands for Forward Dynamics Algorithm:

FDA-1 First, the $6mn \times 6mn$ matrix, $\tilde{\mathbf{M}} = \mathbf{N}_l^T\mathbf{M}\mathbf{N}_l$, is computed as,

$$\tilde{\mathbf{M}} \equiv \begin{bmatrix} \tilde{\mathbf{M}}_{11} & & \text{sym} & \\ \tilde{\mathbf{M}}_{21} & \tilde{\mathbf{M}}_{22} & & \\ \vdots & \vdots & \ddots & \\ \tilde{\mathbf{M}}_{m1} & \tilde{\mathbf{M}}_{m2} & \cdots & \tilde{\mathbf{M}}_{mm} \end{bmatrix}^{(1n)} \tag{27}$$

where "sym" represents the symmetric elements, and the $6n \times 6n$ matrices, $\tilde{\mathbf{M}}_{ij}$, which are actually $\tilde{\mathbf{M}}_{ij}^{(1n)}$, defined similar to eq.(5a). The 6×6 matrices, $\tilde{\mathbf{M}}_{ij}^{(l)}$, for $i = 1, \cdots, m;\ j = 1, \cdots, i$, are given by

$$\tilde{\mathbf{M}}_{ij}^{(l)} \equiv [\tilde{\mathbf{M}}_i\tilde{\mathbf{A}}_{ij}]^{(l)}, \quad \text{where} \qquad \tilde{\mathbf{A}}_{11}^{(l)} = \mathbf{1} \tag{28}$$

and the 6×6 matrix, $\tilde{\mathbf{A}}_{ij}^{(l)}$, is shown in eq.(17c). Note that the 6×6 matrix, $\tilde{\mathbf{M}}_i^{(l)}$, is the well-known mass matrix of the *composite body*, i, [32] of the lth leg that consists of the bodies $\#m, \cdots, \#i$, of leg l, Fig. 3(b).

FDA-2 Next, the $6n \times 6n$ matrix, $\bar{\mathbf{M}} = \mathbf{N}_c^T \tilde{\mathbf{M}} \mathbf{N}_c$, is found as,

$$\bar{\mathbf{M}} \equiv \begin{bmatrix} \bar{\mathbf{M}}_{11} & & \text{sym} & \\ \bar{\mathbf{M}}_{21} & \bar{\mathbf{M}}_{22} & & \\ \vdots & \vdots & \ddots & \\ \bar{\mathbf{M}}_{m1} & \bar{\mathbf{M}}_{m2} & \cdots & \bar{\mathbf{M}}_{mm} \end{bmatrix} \tag{29}$$

where $\bar{\mathbf{M}}_{ij}$, for $i = 1, \cdots, m;\ j = 1, \cdots, i$, is the 6×6 mass matrix with some composite effect of other leg bodies, which is given by

$$\bar{\mathbf{M}}_{ij} \equiv \begin{cases} \tilde{\mathbf{M}}_{kk}^{(i)} + \mathbf{A}_{mk}^{(i)^T} \tilde{\mathbf{M}}_{(11:k-1,k-1)}^{(i)} \mathbf{A}_{mk}^{(j)} \\ -(\mathbf{A}_{mk}^{(i)^T} \tilde{\mathbf{M}}_{(k1:k,k-1)}^{(j)^T} + \tilde{\mathbf{M}}_{(k1:k,k-1)}^{(j)} \mathbf{A}_{mk}^{(i)}) \\ +\bar{\mathbf{A}}_{mk}^{(i)^T} \mathbf{X} \bar{\mathbf{A}}_{mk}{}^{(j)} + \bar{\mathbf{A}}_{mk}^{(i)^T} \mathbf{Y}^{(j)} + \mathbf{Y}^{(j)^T} \bar{\mathbf{A}}_{mk}^{(i)} & \text{if} \quad i = j \\ \bar{\mathbf{A}}_{mk}^{(i)^T} \mathbf{X} \bar{\mathbf{A}}_{mk}{}^{(j)} + \bar{\mathbf{A}}_{mk}^{(i)^T} \mathbf{Y}^{(j)} + \mathbf{Y}^{(j)^T} \bar{\mathbf{A}}_{mk}^{(i)} & \text{otherwise} \end{cases} \tag{30}$$

The 6×6 matrices, $\mathbf{X}$ and $\mathbf{Y}^{(l)}$, are shown below:

$$\mathbf{X} \equiv \tilde{\mathbf{M}}_{(11:k-1,k-1)}^{(+)} + \tilde{\mathbf{M}}_{k+1,1:m,k-1}^{(+)} + \tilde{\mathbf{M}}_{k+1,1:m,k-1}^{(+)^T} + \tilde{\mathbf{M}}_{k+1,k+1:mm}^{(+)} \tag{31}$$

$$\mathbf{Y}^{(l)} \equiv [\tilde{\mathbf{M}}_{k1:k,k-1}^T + \tilde{\mathbf{M}}_{k+1,k:mk} + (\tilde{\mathbf{M}}_{11:k-1,k-1} \tilde{\mathbf{M}}_{k+1,1:m,k-1}) \mathbf{A}_{mk}]^{(j)}, \tag{32}$$

Matrix, $\mathbf{X}$, is the sum of all mass matrices, $\tilde{\mathbf{M}}_{ij}^{(l)}$ of eq.(28), over all the values of i, j, and l, excepts those appearing in the kth rows and columns. Similarly, it can be observed that other matrix sums in the form of $\tilde{\mathbf{M}}_{(pq:rs)}^{(l)}$, eq.(30), are associated with the sum of several blocks of the matrix $\tilde{\mathbf{M}}_{ij}^{(l)}$, given by eq.(28). The subscript of $\tilde{\mathbf{M}}_{(pq:rs)}^{(l)}$ and $\mathbf{M}_{(pq:rs)}^{(+)}$ imply the sum over the body and leg numbers, i.e.,

$$\tilde{\mathbf{M}}_{(pq:rs)}^{(l)} \equiv \sum_{i=p}^{r} [\sum_{j=q}^{\min\{i,s\}} \tilde{\mathbf{M}}_{ij} + \sum_{j=i+1}^{s} \tilde{\mathbf{M}}_{ji}^T]^{(l)} \quad \text{and} \qquad \mathbf{M}_{(pq:rs)}^{(+)} \equiv \sum_{l=1}^{n} \mathbf{M}_{(pq:rs)}^{(l)} \tag{33}$$

where p to q are corresponding to the row and column of matrix $\tilde{\mathbf{M}}$, eq.(27), which varies upto r and s, respectively.

FDA-3 Finally, each element of the GIM, i_{ij}, can be written as

$$i_{ij} = \mathbf{p}_k^{(i)^T} \bar{\mathbf{M}}_{ij} \mathbf{p}_k^{(j)} \tag{34}$$

for $i = 1, \cdots, n;\ j = 1, \cdots, i$. Equation (34) is in a format required by the recursive forward dynamics algorithm of [27, 32], which can be used directly becasue as long as the form of eq.(34) is available the forward dynamics steps are independent of the type of systems, i.e., closed or open loop.

From the above steps the question may arise how a recursive forward dynamics algorithm is possible when the above steps are not recursive. Note that the matrices associated with representation of the elements i_{ij} of the GIM, $\mathbf{I}$, as in eq.(34), need not be evaluated at this stage. On the contrary, the GIM is first analytically decomposed using a suitable scheme, e.g., using the Gaussian elimination (GE) rules [27, 32]. Again one may wonder how an $\mathcal{O}(n)$ algorithm is possible using the scheme which is of $\mathcal{O}(n^3)$. This is explained as follows: During the decomposition of the GIM, whose elements are given by eq.(34), only the rules of the GE are applied to obtain the modified expressions of the elements of the reduced matrices. No attempt is made to compute them. Next substitutions formulas, as required to solve the joint accelerations, are applied to the elements of the decomposed GIM, whose analytical expressions are known from the previous step, i.e., analytical GE. Here also one need not do the substitutions term by term, which would be of order $\mathcal{O}(n^2)$. The final expressions for the solutions of the joint accelerations then reveal the recursive relations, which is not so evident if the GE and the substitutions are performed independently. See Refs. [27, 32] for more details.

VI. RESULTS

The proposed dynamic algorithms, namely, the inverse and forward dynamics required for control and simulation, respectively, are applied to planar closed loop systems, a four-bar linkage and a planar parallel manipulator, shown in Fig. 4 and 9, respectively. In order to perform the analyses, link 1 of the four-bar linkage, and the first links of the two legs of the planar manipulator are assumed to be driven or actuated. The trajectory for an actuated joint is taken as

$$\theta_i(t) = \theta_i(0) + [\theta_i(T) - \theta_i(0)]s(\bar{t}), \quad \text{where} \qquad s(\bar{t}) = \bar{t} - \frac{1}{2\pi}\sin 2\pi\bar{t} \tag{35}$$

and $\bar{t} = t/T$. The function, $s(\bar{t})$, being the normalized cycloidal motion [16], and T and t are the total time of traverse and the time instant at which the results are desired, respectively. Moreover, $\theta_i(0)$ and $\theta_i(T)$ are the given initial and final values of the actuated joints, respectively, i.e., θ_1 for the four-bar linkage, and $\theta_1^{(1)}$ and $\theta_1^{(2)}$ for the planar manipulator. The trajectory, eq.(35), is chosen due to its smooth characteristics, namely, the velocity

and acceleration vanish at the start and end of the motion, as evident from Figs. 5(b) and Figs. 5(c), respectively.

A. Four-bar Linkage

The four-bar linkage under study is shown in Fig. 4. The parameters associated with the linkage are given by

$$a_1 = .3 \text{ m};\ a_2 = 1 \text{ m};\ a_3 = .6 \text{ m};\ a_4 = 1 \text{ m}$$
$$m_1 = 1.5 \text{ kg};\ m_2 = 5 \text{ kg};\ m_3 = 3 \text{ kg};\ g = 9.81 \text{ m/sec}^2$$

where the link lengths, $a_1, \cdots, a_4$, are defined by Fig. 4, whereas m_1, m_2, and m_3 are the masses of links 1, 2, and 3, respectively. Moreover, g is the value for the acceleration due to gravity. It is the magnitude of the gravity vector, $\mathbf{g}$, indicated in Fig. 4.

Other values associated with the trajectory, eq.(35), of the actuated joint, i.e., joint 1, are taken as

$$\theta_1(0) = 30^o;\ \theta_1(T) = 90^o;\ T = 20 \text{ sec};\ \bar{t} = .02 \text{ sec} \tag{36}$$

The steps to perform the inverse and forward dynamics and simulation are given below:

1. The unactuated joints, namely, θ_2 and θ_3, are calculated using the concept of *characteristic pair of joints* [43], which are shown in Fig. 5(a). The variables, θ_1, θ_2 and θ_3 in the plot represent the angles θ_1, θ_2 and θ_3, respectively. The unactuated joint angle, θ_4, is simply $\theta_4 = \pi - (\theta_1 + \theta_2 + \theta_3)$.

2. The unactuated joint rates and accelerations for joints 2 and 3, i.e., $\dot{\theta}_2,\ \dot{\theta}_3$, and $\ddot{\theta}_2,\ \ddot{\theta}_3$, are evaluated from eq.(11b), and the time derivatives of the expressions for the twists, eqs.(7a), (8), and others, with necessary algebraic manipulation, as done to find $\dot{\theta}_i$ in eq.(11b), respectively. The results are shown in Figs. 5(b) and Figs. 5(c), respectively. For joint 4, $\dot{\theta}_4 = -(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3);\ \ddot{\theta}_4 = -(\ddot{\theta}_1 + \ddot{\theta}_2 + \ddot{\theta}_3)$. The variables for the fourth joint do not affect the dynamics of the four-bar linkage.

3. The actuated joint torque, τ_1, of the vertical four-bar linkage, i.e., the linkage is under gravity, following the trajectory given by eqs.(35) and (36), is computed using the algorithm given in Section V(A). The variation of the joint torque, τ_1, is shown in Fig. 6, where τ_1 means τ_1.

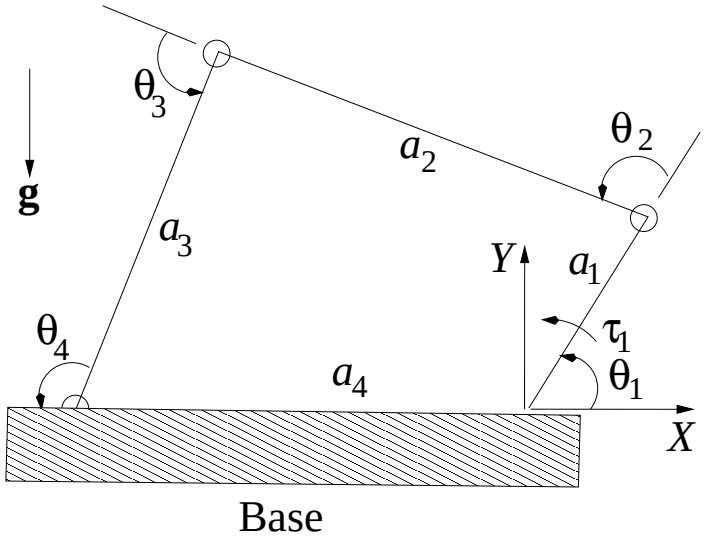

Fig. 4 A four-bar linkage.

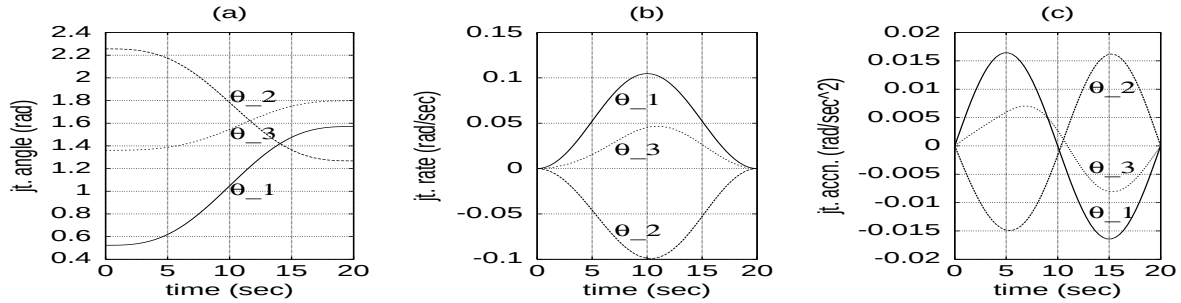

Fig. 5 (a) Joint angles, $\theta_1, \theta_2, \theta_3$; (b) Joint rates, $\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3$; (c) Joint accelerations, $\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3$

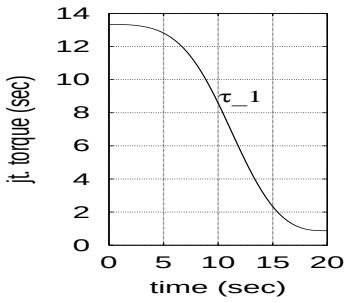

Fig. 6 Torque, τ_1.

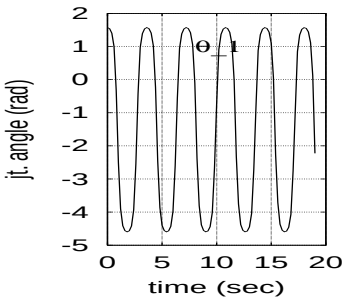

Fig. 7 Free simulation: θ_1.

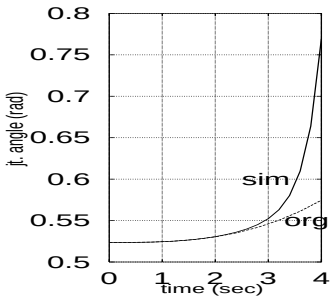

Fig. 8 Forced simulation: θ_1.

4. Forward dynamics, i.e., the joint acceleration for the actuated joint, $\ddot{\theta}_1$, is obtained using the formulation given in Section V(B). For the four-bar linkage, the inertia matrix is a scalar.

5. Based on the joint acceleration computed in the above step, simulation is performed using NEWMOS[1] software. The *SHAMGOR* algorithm of NEWMOS based on the Adams-Bashforth-Moulton formula [45] is used. Time step and tolerance for the numerical integration for all the simulation carried out here are taken as .2 sec and 10^{-6}, respectively. The result of the free-simulation, i.e., the linkage is freed under gravity from the initial vertical position of link 1, i.e., $\theta_1(0) = 90^o(1.5708 \text{ rad})$ and $\dot{\theta}_1(0) = 0$ rad/sec, is shown in Fig. 7.

6. The result of the forced simulation, i.e., with external joint torques and the gravity are shown in Fig. 8, where "org" and "sim" in the plot denote the desired original and simulated results, respectively. The external torque is the one computed from the inverse dynamics algorithm, described above. The simulated joint angle diverges from the desired angle after about 2.25 sec, Fig. 8. This is attributed to the corresponding zero eigenvalue instability, which is overcome by using a feedback control. For example, consider the following joint torque input, τ_1 [46], as

$$\tau_1 = I\tau_e + C\dot{\theta}, \quad \text{where} \qquad \tau_e = \ddot{\theta}_d - 2\lambda(\dot{\theta} - \dot{\theta}_d) - \lambda^2(\theta - \theta_d) \tag{37}$$

Scalars, I and C, follow from eq.(25), respectively. The subscript "d" stands for the desired trajectory, and λ is a positive number. Using eq.(37) with $\lambda = 1$, it is verified that the desired trajectory, as shown in Fig. 5(a), is followed exactly.

B. Five-bar Planar Parallel Manipulator

A five-bar planar parallel manipulator, also known as planar Delta robot [41], is shown in Fig. 9, in which the actuated joints are the first joints of both the legs, indicated with pointing arcs. The parameters associated with the manipulator are given by

$$a_1^{(1)} = .6 \text{ m};\ a_2^{(1)} = 1.1 \text{ m};\ a_1^{(2)} = .7 \text{ m};\ a_2^{(2)} = .8 \text{ m};\ a_0 = 1.2 \text{ m}$$

$$m_1^{(1)} = 3 \text{ kg};\ m_2^{(1)} = 1.6 \text{ kg};\ m_1^{(2)} = 2 \text{ kg};\ m_2^{(2)} = 1.1 \text{ kg};\ g = 9.81 \text{ m/sec}^2$$

[1] A simulation software in C at the Institute of B Mechanics, University of Stuttgart, Germany.

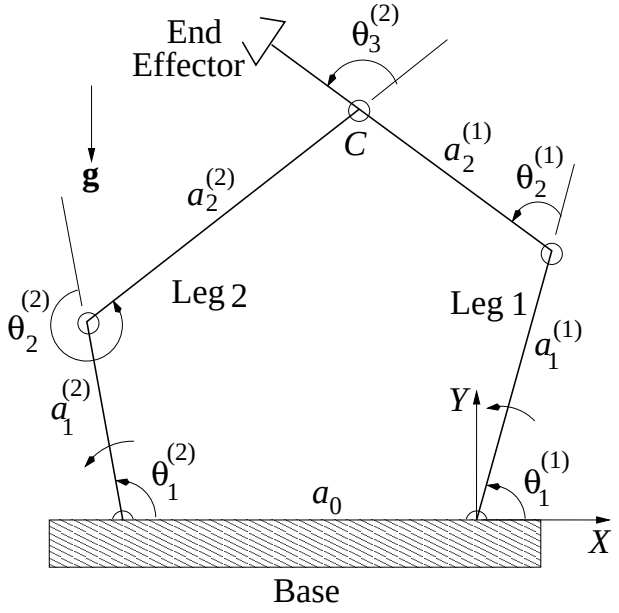

Fig. 9 A five-bar planar parallel manipulator.

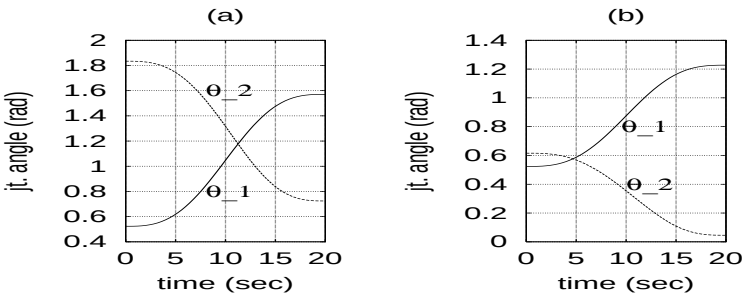

Fig. 10 Joint angles: (a) Leg 1; (b) Leg 2.

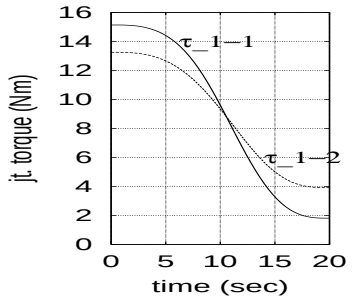

Fig. 11 Torques, $\tau_1^{(1)}$, $\tau_1^{(2)}$.

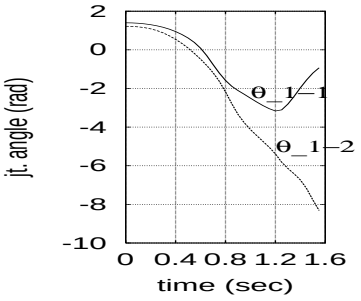

Fig. 12 Free simulation: $\theta_1^{(1)}$, $\theta_1^{(2)}$.

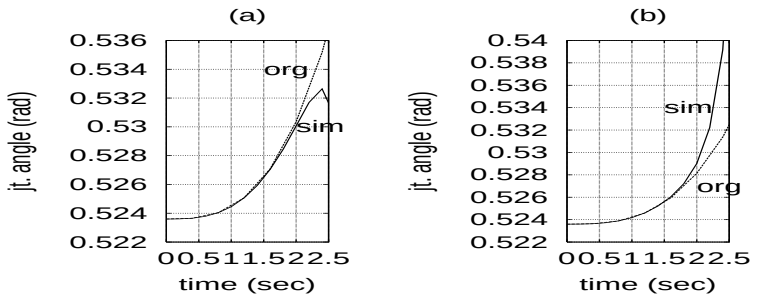

Fig. 13 Forced simulation: (a) Leg 1, $\theta_1^{(1)}$; (b) Leg 2, $\theta_1^{(2)}$.

where $a_i^{(l)}$, for $i, l = 1, 2$, and a_0 are the link lengths, as shown in Fig. 9, and where $m_i^{(l)}$, for $i, l = 1, 2$, are the masses of the link i of lth leg. Other values associated with the trajectory, eq.(35), of the actuated joints, i.e., joints 1 of legs 1 and 2, are as follows:

$$\theta_1^{(1)}(0) = 30^o;\ \theta_1^{(1)}(T) = 90^o;\ \theta_1^{(2)}(0) = 30^o;\ \theta_1^{(2)}(T) = 70.3^o \tag{38}$$

$$T = 20 \text{ sec};\ \bar{t} = .02 \text{ sec} \tag{39}$$

The steps to perform the inverse and forward dynamics, and simulation is described below:

1. The unactuated joints, namely, $\theta_2^{(l)}$, for $l = 1, 2$, are calculated using the concept of *characteristic pair of joints* [43] again, where $\theta_3^{(2)}$ is simply

$$\theta_3^{(2)} = [\theta_1 + \theta_2]^{(1)} - [\theta_1 + \theta_2]^{(2)}$$

   The actuated and unactuated joint angles for leg 1 are shown in Fig. 10(a)—θ_1, θ_2 represent the angles θ_1 and θ_2, respectively. Similar results for leg 2 are shown in Fig. 10(b).

2. The unactuated joint rates and accelerations are evaluated in a manner similar to the four-bar linkage, as described in item 2 of Section VI(A). The patterns are same, i.e., Fig. 5(b) and 5(c).

3. The required actuated joint torques, $\tau_1^{(1)}$, and $\tau_1^{(2)}$, of the vertical planar parallel manipulator following the trajectory given by eqs.(35) and (39) are shown in Fig. 11—$\tau\_1 - 1$, $\tau\_1 - 2$ represent the torques $\tau_1^{(1)}$ and $\tau_1^{(2)}$, respectively.

4. Forward dynamics is performed using the recursive algorithm of [27], where the 2×2 generalized inertia matrix is evaluated using the algorithm given in Section V(B). Even though its inversion is simple, the recursive forward dynamics steps of [27] are followed to verify the applicability of eq.(34).

5. Free-fall simulation with initial conditions of $\theta_1^{(1)}(0) = 80^o$ (1.4 rad), $\theta_1^{(2)}(0) = 70^o$(1.22 rad); $\dot{\theta}_1^{(1)}(0) = \theta_1^{(2)}(0) = 0$ rad/sec is carried out whose results are shown in Fig. 12, where $\theta\_1 - 1$ and $\theta\_1 - 2$ mean $\theta_1^{(1)}$ mean $\theta_1^{(2)}$, respectively. After about 1.4 sec of simulation time the program stopped executing because the manipulator comes to an extreme position when it is singular.

6. For the forced simulation, as defined for the four-bar linkage, the program stops after the simulation time of 3.2 sec as it apparently hits a singular configuration. This is certainly not possible as the joint torques are evaluated from the inverse dynamics algorithm for a feasible trajectory shown in Figs 10(a) and (b). The simulated joint trajectories are shown in Fig. 13(a) an (b) for legs 1 and 2, respectively, where "org" and "sim" imply the original and simulated results, respectively. This behavior is due to the corresponding zero eigenvalue instability, as in the four-bar linkage case, which can be stabilized using feedback laws similar to eq.(37) too.

VII. CONCLUSIONS

A kinematic formulation for the closed loop parallel platform type mechanical system, as shown in in Fig 3(a), is presented in this paper, which is *recursive* in nature. This also leads to the *minimum order* representation of the dynamic equations of motion. Two recursive algorithms, one for the inverse and another for the forward dynamics, are proposed. The overall complexity of the either problem is $\mathcal{O}(m)+\mathcal{O}(n)$, where m and n are the number of rigid bodies in each leg and the number of legs, respectively. Hence, for the proposed formulation exploits both the advantages of minimum order representation and recursive algorithms, which earlier were available only for the open loop systems like serial manipulators.

ACKNOWLEDGEMENT

The authors sincerely acknowledge the financial support of the Alexander von Humboldt foundation, Germany, granted to the first author.

REFERENCES

[1] R.A. Wehage and E.J. Haug, "Generalized coordinate partitioning for dimension reduction in analysis of constrained systems," *ASME J. Mech. Des.*, V. 104, Jan., pp. 247–255, 1982.

[2] J.W. Kamman and R.L. Huston, "Dynamics of constrained multibody systems," *ASME J. Appl. Mech.*, V. 51, Dec., pp. 899–903, 1984.

[3] J. Angeles and S. Lee, "The formulation of dynamical equations of holonomic mechanical systems using a natural orthogonal complement," *ASME J. Appl. Mech.*, V. 55, Mar., pp. 243–244, 1988.

[4] E. Bayo, J. García de Jalón and M.A. Serna, "A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems," *Computer Meth. in Appl. Mech. and Eng.*, V. 71, pp. 183–195, 1988.

[5] E.J. Haug, *Computer-Aided Kinematics and Dynamics of Mechanical Systems*, Allyn and Bacon, Boston, 1989.

[6] R.L. Huston, *Multibody Dynamics*, Butterworth-Heinemann, Boston, 1990.

[7] W. Schiehlen, *Multibody Systems Handbook*, Springer-Verlag, New York, 1990.

[8] F.M.L. Amirouche, *Computational Methods in Multibody Dynamics*, Prentice Hall, NJ, 1992.

[9] J. Garcia de Jalon and E. Bayo, *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge*, Springer-Verlage, New York, 1994.

[10] A.A. Shabana, *Dynamics of Multibody Systems*, Cambridge Univ. Press, Cambridge, 1998.

[11] J.Y.S. Luh, M.W. Walker and R.P.C. Paul, "Resolved acceleration control of mechanical manipulators," *IEEE Trans. on Automatic Control*, V. 25, N. 3, pp. 468–474, 1980.

[12] Y. Nakamura and M. Ghodoussi, "Dynamics computation of closed-link robot mechanisms with nonredundant and redundant actuators," *IEEE Trans. on R&A*, V.5, N.3, pp. 294-302, 1989.

[13] S. Lin, "Dynamics of the manipulator with closed chains," *IEEE Trans. on R&A*, V.6, N.4, pp. 496-501, 1990.

[14] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*, McGraw-Hill, New York, 1996.

[15] M.W. Walker and D.E. Orin, "Efficient dynamic computer simulation of robotic mechanisms," *ASME J. Dyn. Sys., Meas., and Cont.*, V. 104, Sept., pp. 205–211, 1982.

[16] J. Angeles, *Fundamentals of Robotic Mechanical Systems*, Second Edition, Springer-Verlag, New York, 1997.

[17] R. Featherstone, "The calculation of robot dynamics using articulated-body inertias," *Int. J. Rob. Res.*, V. 2, N. 1, pp. 13–30, 1983.

[18] R. Featherstone, *Robot Dynamics Algorithms*, Kluwer Academic Publishers, Boston, 1987.

[19] D. Bae and E.J. Haug, "A recursive formulation for constrained mechanical system dynamics: Part I. Open loop systems," *Mech. Struct. & Mach.*, V. 15, N. 3, pp. 359–382, 1987.

[20] D. Bae and E.J. Haug, "A recursive formulation for constrained mechanical system dynamics: Part II. Closed loop systems," *Mech. Struct. & Mach.*, V. 15, N. 4, pp. 481–506, 1987.

[21] G. Rodriguez, "Kalman filtering, smoothing, and recursive robot arm forward and inverse dynamics," *IEEE Trans. on R&A*, V. RA-3, N. 6, pp. 624–639, 1987.

[22] W. Schiehlen, "Computational aspects in multibody system dynamics," *Computer Meth. in Appl. Mech. and Eng.*, V. 90, N. 1–3, pp. 569–582, 1990.

[23] W. Schiehlen, "Multibody systems and robot dynamics," *RoManSy 8* (Proc. 8th CISM-IFToMM Symp. on Theory and Practice of Robots and Manipulators), Ed: A. Morecki, et. al., Warsaw Univ. of Tech. Publ., Warsaw, pp. 14–21, 1990.

[24] D. Bae, R. Hwang and E.J. Haug, "A recursive formulation for real-time dynamic simulation of mechanical systems," *ASME J. Mech. Des.*, V. 113, pp. 158–166, 1991.

[25] G. Rodriguez, A. Jain and K. Kreutz-Delgado, "A spatial operator algebra for manipulation modeling and control," *Int. J. Rob. Res.*, V. 10, N. 4, pp. 371–381, 1991.

[26] V. Stejskal and M. Valasek, *Kinematics and Dynamics of Machinery*, Marcel Dekker, Inc., New York, 1996

[27] S.K. Saha, "A decomposition of the manipulator inertia matrix, *IEEE Trans. on R&A*, V. 13, N. 2, Apr., pp. 301–304, 1997.

[28] D. Bae, J.M. Han and H.H. Yoo, "A generalized recursive formulation for constrained mechanical system dynamics," *Mech. Struct. & Mach.*, V. 27, N. 3, pp. 293-315, 1999.

[29] W. Stelzle, A. Kecskeméthy and M. Hiller, "A comparative study of recursive methods," *Archive of Appl. Mech.*, V. 66, pp. 9–19, 1995.

[30] M. Valasek, *On the Efficient Implementation of Multibody Systems Formalisms*, Institutbericht IB-17, Univ. of Stuttgart, Germany, 1990.

[31] U.M. Ascher, D.K. Pai and B.P. Cloutier, "Forward dynamics, elimination methods, and formulation stiffness in robot simulation," *Int. J. Rob. Res.*, V. 16, N. 6, pp. 749–758, 1997.

[32] S.K. Saha, "Dynamic modeling of serial multi-body systems using the decoupled natural orthogonal complement matrices, *ASME J. Appl. Mech.*, V. 66, N. 4, pp. 986–996, 1999.

[33] D. Bae and E.J. Haug, "A recursive formulation for constrained mechanical system dynamics: Part III. Parallel processor implementation," *Mech. Struct. & Mach.*, V. 16, N. 2, pp. 249–269, 1988.

[34] A. Avello, J.M. Jiménez, E. Bayo and J. García de Jalón, "A simple and highly parallelizable method for real-dynamic simulation based on velocity transformations," *Computer Meth. in Appl. Mech. and Eng.*, V107, pp. 313-339, 1993.

[35] A. Fijany, I. Sharf and M.T.D. D'Eleuterio, "Parallel $\mathcal{O}(logN)$ algorithms for computation of manipulator forward dynamics," *IEEE Trans. on R&A*, V. 11, N. 3, 1995.

[36] R. Featherstone, "A divide-and-conquer articulated-body algorithm for parallel $\mathcal{O}(log(n))$ calculation of rigid-body dynamics. Part 1: Basic algorithm," and "—, Part 2: Trees, loops, and accuracy," *Int. J. Rob. Res.*, V. 18, N. 9, pp. 867–892, 1999.

[37] J.F. Liu and K.A. Abdel-Malek, "On the problem of scheduling parallel computations of multibody dynamic analysis," *ASME J. Dyn. Sys., Meas., and Cont.*, V. 121, Sept., pp. 370–376, 1999.

[38] W. Blajer, W. Schiehlen and W. Schirm, "Dynamic analysis of constrained multibody systems using inverse kinematics," *Mech. and Mach. Th.*, V. 28, N. 3, pp. 397–405, 1993.

[39] J. Cuadrado, J. Cardenal and E. Bayo, "Modeling and solution methods for efficient real-time simulation of multibody dynamics," *Mult. Sys. Dyn.*, V. 1., pp. 259–280, 1997.

[40] F. Ghorbel, O. Chételat and R. Longchamp, "A reduced model for constrained rigid bodies with application to parallel robots," *IFAC Robot Control* (Proc. 4th Symp. on Robot Control, Capri, Italy), pp. 45–50, 1994.

[41] H. Brauchli and R. Weber, "Dynamical equations in natural coordinates," *Computer Meth. in Appl. Mech. and Eng.*, V. 91, pp. 1403–1414, 1991.

[42] G.H. Golub and C.F. Van Loan, *Matrix Computations*, John Hopkins Univ. Press, Balitmore, Maryland, 1983.

[43] M. Hiller, "Multiloop kinematic chains," *Kinematics and Dynamics of Multi-body Systems* (Ch. 4), in J. Angeles and A. Kecskeméthy (editors), Springer-Verlag, New York, 1995.

[44] H. Brauchli, "Mass-orthogonal formulation of equations of motion for multibody systems," *J. Appl. Math. and Phy.* (ZAMP) V. 42, March, pp. 169–182, 1991.

[45] L.F. Shampine, *Numerical Solution of Ordinary Differential Equations*, Chapman a& Hall, NY, 1994.

[46] J.E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice-Hall, NJ, 1991.