Modular and Recursive Kinematics and Dynamics for Parallel Manipulators

WASEEM AHMAD KHAN[1], VENKAT N. KROVI[2], SUBIR KUMAR SAHA[3]
and JORGE ANGELES[1]
[1]*Centre for Intelligent Machines, McGill University*
[2]*Mechanical and Aerospace Engineering, State University of New York at Buffalo;*
E-mail: vkrovi@buffalo.edu
[3]*Department of Mechanical Engineering, IIT Delhi*

Abstract. Constrained multibody systems typically feature *multiple closed kinematic loops* that constrain the relative motions and forces within the system. Typically, such systems possess far more articulated degrees-of-freedom (within the chains) than overall end-effector degrees-of-freedom. Thus, actuation of a subset of the articulations creates mixture of active and passive joints within the chain. The presence of such passive joints interferes with the effective modular formulation of the dynamic equations-of-motion in terms of a minimal set of actuator coordinates as well the subsequent recursive solution for both forward and inverse dynamics applications.

   Thus, in this paper, we examine the development of modular and recursive formulations of equations-of-motion in terms of a minimal set of actuated-joint-coordinates for an exactly-actuated parallel manipulators. The 3 <u>R</u>RR planar parallel manipulator, selected to serve as a case-study, is an illustrative example of a multi-loop, multi-degree-of-freedom system with mixtures of active/passive joints. The concept of decoupled natural orthogonal complement (DeNOC) is combined with the spatial parallelism inherent in parallel mechanisms to develop a dynamics formulation that is both recursive and modular. An algorithmic approach to the development of both forward and inverse dynamics is highlighted. The presented simulation studies highlight the overall good numerical behavior of the developed formulation, both in terms of accuracy and lack of formulation stiffness.

Keywords: recursive kinematics, modular dynamics, decoupled natural orthogonal complement, parallel manipulators

1. Introduction

The recent few decades have witnessed an increased use of dynamic computational models for the design, analysis, parametric refinement and model-based control of a variety of multibody systems such as vehicles, heavy machinery, spacecraft and robots. The principal underlying factor for this revolution has been the improved understanding of the methodologies for dynamic modeling of these increasingly-complex multibody systems. A good overview of the wide variety of problems-of-interest, the proposed formulations as well as some of the computational methods

to address some of these problems may be seen in a number of seminal textbooks [7, 16, 20, 43, 44].

In the context of *robotic multibody systems*, the systematic formulation and efficient solution of both the *inverse-* and the *forward-* dynamics problems are of great interest. The goal of the *inverse dynamics* problem is to formulate the system's equations-of-motion (EOM) and compute the time-histories of the controlling actuated joint torques/forces, given the time-histories of all the system's actuated-joint variables. The solution process is primarily an algebraic one and typically does not require the use of numerical integration methods since the position coordinates, velocities and accelerations of the system are known. On the other hand, the *forward dynamics* problem seeks to formulate the system's EOM and compute the time histories of all the joint coordinates, given the time-histories of the actuated joint torques/forces. The solution is obtained in a two-stage process: an initial algebraic solution of the EOM to determine the accelerations which are then numerically integrated in a second stage to obtain the velocity and position time histories.

For either problem-of-interest, the critical first step remains formulation of the EOM and for which the principal formulation approaches fall into one of two broad categories: *Euler-Lagrange* methods and *Newton-Euler* methods. *Euler-Lagrange* approaches are commonly used in robotics to obtain the EOM of robotic manipulators within a configuration space defined in terms of relative joint variables. For serial chain manipulators, such relative joint variables afford a minimal-coordinate description and additionally often permit a direct mapping to actuator coordinates. *Newton-Euler* formulations on the other hand, typically are defined in terms of Cartesian configuration coordinates. Recursive formulations are created by first developing EOM for each body, which are then assembled to obtain the EOM for the entire system. Both formulation approaches have been shown to be extremely effective in generating efficient (and essentially equivalent) EOM for *serial-chain and tree-structured multibody systems*.

However, the adaptation of such formulations to efficiently generate and solve the EOM of *constrained multibody systems* has proven challenging. Such systems can range from fixed topology systems (various types of parallel manipulators, closed-loop linkages) to variable-topology systems (legged walking machines [45], multi-fingered hands [23, 41] and cooperative payload manipulation systems [25]). The characteristic (and dominant) feature of all such constrained multibody systems is the formation of closed kinematic loops. The engendered loop-closure constraints then serve to constrain the relative motions and forces within the system creating a spectrum of underactuated, exactly-actuated or redundantly-actuated dynamic systems [26].

Traditionally, the kinematic loop-closure constraints have been modeled by introducing algebraic constraints (typically nonlinear) into the dynamics formulation. The resulting systems of Differential Algebraic Equations (DAEs) offer numerous challenges for both forward dynamics and inverse dynamics. It is noteworthy that

the solution is very closely tied to the form of the EOM which, in turn, depend critically upon the method of formulation. As Ascher et.al. [6], note the form of the EOM can result in a *formulation stiffness* whose effects manifest themselves in conjunction with but distinctly differing from the effects of the *numerical stiffness* of the selected numerical integration scheme.

In this paper, we will examine the development of both forward and inverse dynamics formulations for a subclass of constrained mechanical systems using an examplary parallel manipulator. Such parallel manipulator systems possess a natural spatial parallelism and *multiple closed kinematic loops* arising due to multiple legs/branches coming in contact with the common central mobile platform. Parallel manipulators also feature *multiple sets of active and passive joints* within these loops requiring careful modelling – thus the selection of a parallel manipulator example is intended to be illustrative.

Further, we place emphasis on the *modular* and *recursive* development of both forward and inverse dynamics formulations (in terms of the minimal set of actuated-coordinates) for such parallel manipulator systems. The emphasis on *modular* development is to promote reuse of existing components – an overall system is assumed to be composed of several serial or tree structured individual subsystems plus sets of holonomic constraints. In particular, we examine exploiting the spatial parallelism [30] that is inherent in closed kinematic chains to pursue a modular composition of the overall system dynamics. Further, we note that while modular formulation may not be a critical issue for fixed-topology parallel manipulator systems, many variable-topology constrained mechanical systems would benefit from a modular formulation.

Recursive algorithms are desirable from the viewpoint of simplicity and uniformity of computation despite the ever-increasing complexity of multibody systems. More specifically, in the context of robotic systems, the recursive implementation of dynamics algorithms has been the key to efforts in real-time dynamics computations and subsequent implementation of dynamic-model-based control (ranging from computed torque to model reference adaptive control methods). Thus, it is anticipated that the ability to recursively implement dynamics algorithms in parallel manipulator systems would play a similar critical role.

It is important to note that, prior to the dynamics computation stage, a forward or inverse kinematics stage is often required. Hence, the development of efficient recursive dynamics algorithms also necessitates the careful investigation of *recursive kinematics algorithms*, which remains a challenging research problem. Thus, a recursive algorithm for the forward dynamics of *closed-chain systems* first appeared in [40] building on the recursive nature of the *Decoupled Natural Orthogonal Complement (DeNOC)* matrices. However, a limited set of examples of primarily one- and two-degree-of-freedom, single-loop planar manipulators were reported there. In the current paper, we explore the extension and application of the DeNOC approach to the case of multi-loop, multi-dof parallel manipulators with mixtures of active and passive joints.

We will bear these aspects in mind while we briefly review and discuss major methods of formulation of the EOM (for both serial and parallel manipulators) in Section 2. Section 3 presents the relevant background pertaining to the DeNOC approach which forms the basis of the recursive and modular dynamic formulation for parallel-architecture manipulators. Section 4 discusses the nuances of the implementation in the context of a case-study of a 3 <u>R</u>RR planar parallel manipulator which features both the multiple closed-loops and the mixtures of actuated and unactuated joints within the chain. The results are discussed in the context of a numerical example in Sections 5 and 6 concludes this paper.

2. Constrained Multibody Dynamics

2.1. NON-RECURSIVE EULER-LAGRANGE FORMULATIONS

In the Euler-Lagrange approach, the dynamics of constrained mechanical system with closed loops are traditionally obtained by cutting the closed loops typically at passive joints. The EOM for the resulting tree-structured articulated-system are then developed in terms of a set of extended generalized coordinates [14]. All solutions are then also required to satisfy the additional algebraic constraint equations required to close the cut-open loops which are enforced by way of *Lagrange multipliers*. The resulting formulation, in *descriptor form*, yields an often simpler albeit larger system of index-3 differential algebraic equations (DAEs) as follows:[1]

$$\mathbf{I}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u}) - \mathbf{A}(\mathbf{q})^T \boldsymbol{\lambda} \tag{1}$$
$$\mathbf{c}(\mathbf{q}) = \mathbf{0} \tag{2}$$

where $\mathbf{q}$ and $\dot{\mathbf{q}}$ are, correspondingly, the n-dimensional vector of generalized coordinates and of velocities, $\mathbf{I}(\mathbf{q})$ is the $n \times n$ dimensional inertia matrix, $\mathbf{c}(\mathbf{q})$ is the m-dimensional vector of holonomic scleronomic constraints; $\boldsymbol{\lambda}$ is the m-dimensional vector of Lagrange multipliers, $\mathbf{A}(\mathbf{q})$ is the $m \times n$ constraint Jacobian matrix, $\mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{u})$ is the n-dimensional vector of external forces and velocity-dependent inertia terms, while $\mathbf{u}$ is the vector of actuator forces or torques. The solution of such systems of index-3 DAEs by direct discretization is not possible using explicit finite difference discretization methods [7]. Instead, typically, the above system of index-3 DAEs is converted to a system of ODEs and expressed in state-space form, which may be integrated using standard numerical codes. Some of the typical methods used to achieve this are discussed below.

[1] The differential index is defined as the number of times the DAE has to be differentiated to obtain a standard set of ODEs

2.1.1. *Direct Elimination*

The surplus variables are eliminated directly, using the *position-level algebraic constraints* to explicitly reduce the index-3 DAE to an ODE in a minimal set of generalized coordinates (conversion into Lagrange's Equations of the second kind). This is also referred to as the *closed form solution* of the constraint equations – the resulting minimal-order ODE can then be integrated without worrying about stability issues. However, such a reduction cannot be done in general, and even when it can, the differential equations obtained are typically cumbersome [22].

2.1.2. *Lagrange Multiplier Computation*

All the algebraic position and velocity level constraints are differentiated and represented at the acceleration level, to obtain an augmented index-1 DAE (in terms of both unknown accelerations and unknown multipliers) [7, 32] as:

$$\begin{bmatrix} \mathbf{I(q)} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ -\dot{\mathbf{A}}(\mathbf{q})\dot{\mathbf{q}} \end{bmatrix} \tag{3}$$

which may be solved for $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$. By selecting the state of the system to be $\mathbf{x} = [\mathbf{q}^T \ \dot{\mathbf{q}}^T]^T$ the above set of equations may be converted to the standard state-space form and integrated using standard codes. The advantage is, the conceptual simplicity and simultaneous determination of the accelerations and Lagrange multipliers by solving *linear* system of equations. However, such a model requires more initial conditions than a model that uses an independent set of minimal-actuation-coordinates and tends to suffer from numerical stability issues.

2.1.3. *Lagrange Multiplier Approximation/Penalty Formulation*

In this approach the loop-closure constraints are *relaxed* and replaced using virtual springs and dampers [50]. Using such virtual springs can be considered as a form of penalty formulation [16], which incorporates the constraint equations as a dynamical system penalized by a large factor. The Lagrange multipliers are *estimated* using a compliance-based force-law (based on the extent of constraint violation and an assumed spring stiffness) and eliminated from the list of the $n + m$ unknowns leaving behind a system of $2n$ first order ODEs. While the sole initial drawback may appear to be restricted to the numerical ill-conditioning due to the selection of large penalty factors, it is important to note that penalty approaches only approximate the true constraint forces and can create unanticipated problems.

2.1.4. *Dynamic Projection on to Tangent Space*

These seek to take the constraint-reaction-containing dynamic equations into the *orthogonal* and *tangent* subspaces of the vector space of the system's generalized velocities. Let $\mathbf{S(q)}$ be an $n \times (n - m)$ full-rank matrix whose column space is in the

null space of $\mathbf{A(q)}$ i.e. $\mathbf{A(q)S(q)} = \mathbf{0}$. The orthogonal subspace is spanned by the so-called constraint vectors (forming the rows of the matrix $\mathbf{A(q)}$) while the tangent subspace *complements* this orthogonal subspace in the overall generalized velocity vector space. All *feasible* dependent velocities $\dot{\mathbf{q}}$ of a constrained multibody system necessarily belong to this tangent space, appropriately called the *space of feasible motions*. This space is spanned by the columns of $\mathbf{S(q)}$ and is parameterized by an $n - m$ dimensional vector of independent velocities, $\boldsymbol{\nu}(t)$, yielding the expression for the feasible dependent velocities as $\dot{\mathbf{q}} = \mathbf{S(q)}\boldsymbol{\nu}(t)$. A family of choices exist for the selection between dependent and independent velocities, where each choice can give rise to a different $\mathbf{S(q)}$. A popular choice, called *Coordinate Partitioning* [44], in which the generalized velocity is partitioned into m-dimensional dependent $\dot{\mathbf{q}}_d$ and $(n - m)$-dimensional independent $\dot{\mathbf{q}}_i$ velocities, i.e.,

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{\mathbf{q}}_d \\ \dot{\mathbf{q}}_i \end{bmatrix} \tag{4}$$

By selecting $\boldsymbol{\nu}(t) = \dot{\mathbf{q}}_i$ and solving the linear constraints of $\mathbf{A(q)}$ a relation between $\dot{\mathbf{q}}_d$ and $\dot{\mathbf{q}}_i$ is then obtained as $\mathbf{A}\dot{\mathbf{q}} \equiv \mathbf{A}_d\dot{\mathbf{q}}_d + \mathbf{A}_i\dot{\mathbf{q}}_i = \mathbf{0}$. While $\mathbf{A}_d$ is nonsingular, one can solve for $\dot{\mathbf{q}}_d = -\mathbf{A}_d^{-1}\mathbf{A}_i\dot{\mathbf{q}}_i \equiv \mathbf{K}\dot{\mathbf{q}}_i$ where $\mathbf{K}$ is an $m \times (n - m)$ matrix. This leads to a special form of $\mathbf{S(q)}$ (denoted by $\mathbf{T}$)

$$\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{K} \\ \mathbf{1} \end{bmatrix} \dot{\mathbf{q}}_i = \mathbf{T}\dot{\mathbf{q}}_i \tag{5}$$

The $n \times (n - m)$ matrix $\mathbf{T}$ lies in the null space of $\mathbf{A}$, i.e., $\mathbf{AT} = \mathbf{O}$, where $\mathbf{O}$ represent the $m \times (n - m)$ zero matrix; matrix $\mathbf{T}$ is commonly called the *loop-closure orthogonal complement*. Efficient methods for the numerical computation of $\mathbf{T}$ exist and are reviewed by [16]. Pre-multiplying both sides of Equation (1) by $\mathbf{T}^T$ we obtain a constraint-free second order ODE as

$$\mathbf{T}^T\mathbf{I(q)}\ddot{\mathbf{q}} = \mathbf{T}^T\mathbf{f(q, \dot{q}, u)} \tag{6}$$

The above system of equations is still under-determined, but may be successfully used in its current form for inverse dynamics applications (as it often is). Equation (6) a system of $n - m$ second order ODEs in n generalized coordinates, and may then be augmented by m constraint conditions at the acceleration level, $\mathbf{A}\ddot{\mathbf{q}} + \dot{\mathbf{A}}\dot{\mathbf{q}} = \mathbf{0}$, to form a system of n second-order ODEs in as many unknowns. Alternatively, an approach known as the *Embedding Technique* [44] is commonly used, where Equation (5) is differentiated with respect to time and embedded in Equation (6) to create

$$\mathbf{T}^T\mathbf{I(q)T}\ddot{\mathbf{q}}_i + \mathbf{T}^T\mathbf{I(q)}\dot{\mathbf{T}}\dot{\mathbf{q}}_i = \mathbf{T}^T\mathbf{f(q, \dot{q}, u)} \tag{7}$$

which is the minimal-order ODE sought and can be integrated with suitable ODE solvers. A detailed description of the velocity partitioning formulation as well as the geometrical interpretation of constrained system dynamics can be found in Blajer [11].

2.2. RECURSIVE NEWTON-EULER FORMULATIONS

Traditional Lagrange-based formulations of EOM yield algorithms that are of order $O(N^4)$ [14], in the number of floating point operations; where N is the number of rigid bodies in the manipulator. In contrast, we note that most of the fast recursive dynamics algorithms proposed in the last two decades have been based on the Newton-Euler formulation. Stepanenko and Vukobratovic [47] are credited with the development of the first recursive NE method for inverse dynamic computations for human limbs that resulted in an $O(N)$ algorithm. Orin *et al.* [33] improved the efficiency of the recursive inverse dynamics method by referring forces and moments to local link coordinates and employed it for real-time control of a leg of a walking machine. Luh *et al.* [27] developed the first implementation of an efficient Recursive Newton-Euler Algorithm (RNEA) for both forward and inverse dynamics by referring most quantities to link coordinates. Further gains have been made in the efficiency over the years, see for example [10] and [18].

The earliest $O(N)$ algorithm for *forward dynamics* was developed by Vereshchagin [48], who used a recursive formulation to evaluate the Gibbs-Appel form of the EOM, applicable to unbranched chains. This is followed by the work of Armstrong [5] who developed a more systematic $O(N)$ algorithm for mechanisms including those with spherical joints. Later, Orin and Walker [34] used RNEA for inverse dynamics as the basis for an efficient recursive forward dynamic algorithm commonly referred to as the *Composite-Rigid-Body Algorithm* (CRBA). The necessity to solve a linear system of equations of dimension N results in an algorithm of complexity $O(N^3)$. However for small N, the first-order terms dominate the computation, making the CRBA method perhaps the most efficient general purpose algorithm for serial manipulators with $N < 10$ (which includes most practical cases). Featherstone [13] developed the *Articulated-Body Algorithm* (ABA) which was followed by a more elaborate and faster model [14]. The computational complexity of ABA is $O(N)$ but is more efficient than CRBA only for $N > 9$. Further gains have been made in efficiency over the years, see for example [12] and [29].

However, extension of such NE methods to systems with closed kinematic loops has met with limitations principally due to the algebraic nature of the constraints. The most common method for dealing with kinematic loops is to cut the loop, introduce Lagrange multipliers to substitute for the cut joints and develop a recursive method within the individual open-chain systems. The resulting method works well for inverse dynamics computations and is used extensively for real-time dynamic-model based control applications. However, most of the resulting forward dynamics

algorithms for closed-loop mechanisms result in EOM described in terms of *a non-minimal set of generalized coordinates* [8, 9, 15, 42, 46]. These equations would then need to be projected onto the independent set of coordinates, typically by a numerical scheme, which destroys the recursive nature. This is the principal limitation that the Decoupled Natural Orthogonal Complement approach (discussed next) seeks to overcome.

2.3. THE DECOUPLED NATURAL ORTHOGONAL COMPLEMENT

The *Natural Orthogonal Complement (NOC)* has a rich history [2–4] of application in reducing the unconstrained Newton-Euler EOM of various bodies of a serial-chain articulated robotic system to a set of reduced-dimension Euler-Lagrange EOM in terms of the minimal set of actuated joint-coordinates. We review some of the critical underlying concepts of the NOC-based formulations and their applicability to developing the reduced equations in the rest of this sub-section. A brief overview of critical mathematical foundations is presented in Section 3 and the interested reader is referred to [2] for further details.

The key to the development of the reduced equations by the NOC method is the definition of the NOC matrix – a velocity transformation matrix that relates the Cartesian angular/translational velocities of various bodies to the joint rates defined as orthogonal to the kinematic constraint matrix. While, in general, orthogonal complement matrices tend to be non-unique, Saha and Angeles [39] showed a systematic and constructive method for computing a unique Natural Orthogonal Complement for the twist constraint matrix, eliminating the need for use of numerical SVD/eigenvalue methods. This feature is capitalized upon and plays a critical role in the development of the reduced dynamic equations. By definition, the columns of such an NOC matrix span the null space of the matrix of velocity constraints. Hence, all motions for the articulated system defined as linear combinations of the columns remain consistent (instantaneously) with the applied constraints and hence do not generate any (non-working) internal forces. Hence, projecting the unconstrained EOM onto the feasible motion space (spanned by the columns of the NOC matrix) creates a reduced system of independent EOM (in terms of the joint rates) while eliminating all the non-working constraint wrenches.

Saha [36–38] introduced a representation of this NOC matrix as the product of two matrices – a lower block triangular matrix and a block diagonal matrix – termed the Decoupled Natural Orthogonal Complement matrices. The special structure facilitates setup of a recursive forward/inverse kinematics computation and serves as a critical first step towards implementing recursive forward/inverse dynamics algorithms (that would otherwise not have been possible with the NOC formulation). Thus, although recursive kinematics algorithms for serial chains have had a long history [27, 33, 47], a recursive algorithm for the forward kinematics of *closed-chain systems* first appeared in [40]. In that work, examples of one- and
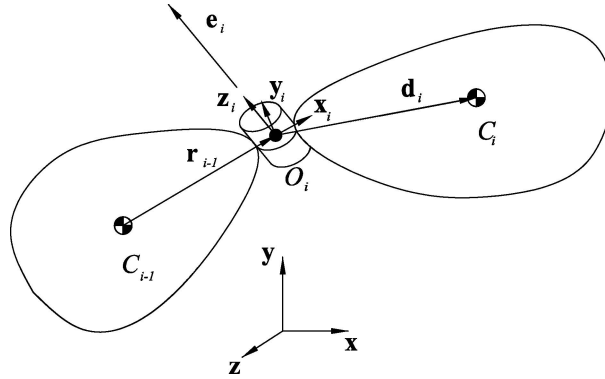
Figure 1. Two bodies connected by a kinematic pair.

two-degree-of-freedom, single-loop planar manipulators were included and various physical interpretations were reported. In this paper, we explore the extension and application of the DeNOC to the case of multi-loop, multi-dof parallel manipulators, using a *two-step* approach and emphasizing the use of geometric parallelism in such systems.

3. Mathematical Background

3.1. TWISTS, WRENCHES AND EQUATIONS OF MOTION

In this section, some pertinent definitions and concepts will be briefly reviewed. See [2] and [36] for further details. Figure 1 shows two rigid links of a serial chain connected by a revolute pair. The mass-center of ith link is at C_i while that of link $i - 1$ is at C_{i-1}. The axis of the ith pair is represented by a unit vector $\mathbf{e}_i$. We attach a frame $\mathcal{F}_i$ with origin O_i and axes X_i, Y_i and Z_i, to link $i - 1$, such that Z_i is parallel to $\mathbf{e}_i$. The global inertial reference frame $\mathcal{F}$ with axes X, Y and Z is attached to the base of the manipulator. Unless otherwise specified, all quantities will be represented in this global frame in the ensuing discussion. Further, we define, the three-dimensional position vectors $\mathbf{d}_i$ from the O_i to the mass center of link i and $\mathbf{r}_{i-1}$ from the mass center of link $i - 1$ to O_i. The six-dimensional twist[2] and wrench vectors of link i at its mass center C_i are now defined as

$$\mathbf{t}_i = \begin{bmatrix} \boldsymbol{\omega}_i \\ \mathbf{v}_i \end{bmatrix}; \quad \mathbf{w}_i = \begin{bmatrix} \mathbf{n}_i \\ \mathbf{f}_i \end{bmatrix} \tag{8}$$

where $\boldsymbol{\omega}_i$, $\mathbf{v}_i$, $\mathbf{n}_i$, and $\mathbf{f}_i$ are the three-dimensional angular and linear velocities of the center-of-mass and the moment and force vectors acting at the center-of-mass

---

[2] See Appendix A for further details of this hybrid twist definition.

of link i, expressed in an inertial frame of reference coincident with C_i. The Newton-Euler equations for link i are

$$\mathbf{n}_i = \mathbf{I}_i \dot{\boldsymbol{\omega}}_i + \dot{\mathbf{I}}_i \boldsymbol{\omega}_i = \mathbf{I}_i \dot{\boldsymbol{\omega}}_i + \hat{\boldsymbol{\omega}}_i \mathbf{I}_i \boldsymbol{\omega}_i$$
$$\mathbf{f}_i = m_i \dot{\mathbf{v}}_i$$

where $\mathbf{I}_i$ is the 3×3 inertia tensor about C_i, m_i is the mass of link i and $\hat{\boldsymbol{\omega}}_i$ is the cross product matrix of $\boldsymbol{\omega}_i$. The above set, in vector form, may be written as

$$\mathbf{w}_i = \underbrace{\begin{bmatrix} \mathbf{I}_i & \mathbf{O} \\ \mathbf{O} & m_i \mathbf{1} \end{bmatrix}}_{\mathbf{M}_i} \begin{bmatrix} \dot{\boldsymbol{\omega}}_i \\ \dot{\mathbf{v}}_i \end{bmatrix} + \underbrace{\begin{bmatrix} \hat{\boldsymbol{\omega}}_i & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}}_{\mathbf{W}_i} \begin{bmatrix} \mathbf{I}_i & \mathbf{O} \\ \mathbf{O} & m_i \mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_i \\ \mathbf{v}_i \end{bmatrix} \tag{9}$$

or

$$\mathbf{w}_i = \mathbf{M}_i \dot{\mathbf{t}}_i + \mathbf{W}_i \mathbf{M}_i \mathbf{t}_i \tag{10}$$

For a serial chain with n moving rigid links coupled by n revolutes, we define

$$\mathbf{t} = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_n \end{bmatrix}; \quad \mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_n \end{bmatrix} \tag{11}$$

The resulting set of Newton-Euler equations for the entire unconstrained system then takes the form

$$\mathbf{w} = \mathbf{M}\dot{\mathbf{t}} + \mathbf{W}\mathbf{M}\mathbf{t} \tag{12}$$

where $\mathbf{M}$ and $\mathbf{W}$ are block-diagonal matrices, namely,

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \ddots & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{M}_n \end{bmatrix}; \mathbf{W} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \ddots & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{W}_n \end{bmatrix} \tag{13}$$

3.2. KINEMATIC RELATIONS BETWEEN TWO BODIES COUPLED
     WITH A KINEMATIC PAIR

The twist $\tilde{\mathbf{t}}_i$ of link i at the ith joint location O_i, can be written recursively in terms of the twist of link $i-1$ at mass center C_{i-1} and including the contribution of the ith joint twist as

$$\tilde{\mathbf{t}}_i = \tilde{\mathbf{B}}_{i-1} \mathbf{t}_{i-1} + \tilde{\mathbf{p}}_i \dot{\theta}_i \tag{14}$$

where

$$\tilde{\mathbf{B}}_{i-1} = \begin{bmatrix} \mathbf{1} & \mathbf{O} \\ -\hat{\mathbf{r}}_{i-1} & \mathbf{1} \end{bmatrix} \tag{15}$$

$$\tilde{\mathbf{p}}_i = \begin{bmatrix} \mathbf{e}_i \\ \mathbf{0} \end{bmatrix}, \quad \text{for a revolute joint} \tag{16}$$

$$\tilde{\mathbf{p}}_i = \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_i \end{bmatrix}, \quad \text{for a prismatic joint} \tag{17}$$

where $\hat{\mathbf{r}}_{i-1}$ is the cross product matrix (CPM)[3] of $\mathbf{r}_{i-1}$. Further, the twist $\tilde{\mathbf{t}}_i$ of link i at the ith joint location O_i can be transformed to the subsequent mass center C_i as

$$\mathbf{t}_i = \mathbf{B}_i \tilde{\mathbf{t}}_i; \quad \mathbf{B}_i = \begin{bmatrix} \mathbf{1} & \mathbf{O} \\ -\hat{\mathbf{d}}_i & \mathbf{1} \end{bmatrix} \tag{18}$$

where $\hat{\mathbf{d}}_i$ is the cross product matrix of $\mathbf{d}_i$. Substituting the value of $\tilde{\mathbf{t}}_i$ from Equation (14) in the above equation we obtain

$$\mathbf{t}_i = \mathbf{B}_i \tilde{\mathbf{B}}_{i-1} \mathbf{t}_{i-1} + \mathbf{B}_i \tilde{\mathbf{p}}_i \dot{\theta}_i \tag{19}$$

Further, we introduce the notation

$$\mathbf{t}_i = \mathbf{B}_{i,i-1} \mathbf{t}_{i-1} + \mathbf{p}_i \dot{\theta}_i \tag{20}$$

where

$$\mathbf{B}_{i,i-1} = \begin{bmatrix} \mathbf{1} & \mathbf{O} \\ -\hat{\mathbf{a}}_i & \mathbf{1} \end{bmatrix} \tag{21}$$

$$\mathbf{p}_i = \begin{bmatrix} \mathbf{e}_i \\ -\hat{\mathbf{d}}_i \mathbf{e}_i \end{bmatrix}, \quad \text{for a revolute joint} \tag{22}$$

$$\mathbf{p}_i = \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_i \end{bmatrix}, \quad \text{for a prismatic joint} \tag{23}$$

with $\hat{\mathbf{a}}_i = \hat{\mathbf{r}}_i + \hat{\mathbf{d}}_i$. Matrix $\mathbf{B}_{i,i-1}$ is called the *twist propagation matrix* and can be seen to be the equivalent of the State Transition Matrix of Rodriguez [35], while $\mathbf{p}_i$ is called the *twist generator*. The twist $\mathbf{t}_i$ is thus the sum of twist $\mathbf{t}_{i-1}$ and the twist generated at the ith joint, both evaluated at the corresponding mass centers.

---

[3] The cross product matrix (CPM) $\hat{\mathbf{u}}$ of any three-dimensional vector $\mathbf{u}$ is defined, for any vector $\mathbf{v}$ of the same dimension, as $\hat{\mathbf{u}} = \text{CPM}(\mathbf{u}) = [\partial(\mathbf{u} \times \mathbf{v})/\partial \mathbf{v}]$.

Equation (20) is recursive in nature and is, in fact, the forward recursion part of the *recursive Newton-Euler algorithm* proposed by [27] for the efficient inverse-dynamics computation of serial chains.

4. Modeling of a 3 RRR Planar Parallel Manipulator – A Case Study

Figure 2 shows a manipulator that belongs to the class of planar parallel manipulators with three degrees of freedom [31]. For the sake of simplicity (but without any loss of generality) we restrict ourselves to one that has: i) only revolute joints, ii) identical legs and iii) a moving platform in the shape of an equilateral triangle. The three degree-of-freedom (DOF) planar manipulator consists of three identical dyads, numbered *I*, *II* and *III* coupling the platform $\mathcal{P}$ with the base such that their fixed pivots lie on the vertices of an equilateral triangle. The proximal and the distal links of each dyad are numbered 1 and 2 respectively. Joint 1 of each dyad is actuated. The centroidal moment of inertia of each link about the axis normal to the xy-plane is the scalar I_i, for $i = 1, 2$. The platform is assumed to have a mass m_P (lumped at the mass-center located at the centroid of the moving equilateral triangle P) and a centroidal moment of inertia I_P. We divide the rigid platform $\mathcal{P}$ into *three parts*, corresponding to the three serial chains, *I*, *II* and *III*, such that the end effector of *each open chain* lies at point P, the mass-center of the platform $\mathcal{P}$. Cutting the platform in this manner is advantageous because:

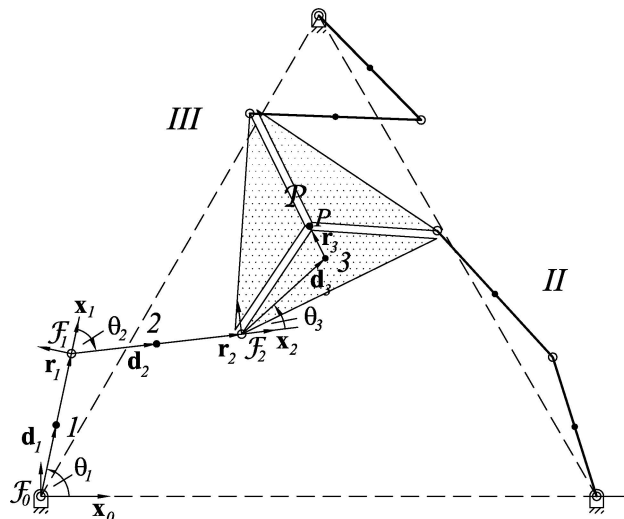– Torques may be applied to the joints that otherwise should be cut to open the chains;

Figure 2. 3-DOF planar parallel manipulator.

- Joint friction may be accommodated directly for such joints;
- Cutting the links (platform) produces a more streamlined recursive modelling algorithm for parallel manipulators;

While we will discuss some of these issues in detail, the interested reader is also referred to a similar discussion of benefits presented in [51]. In what follows, we will first consider the development of a recursive forward kinematics formulation that will serve as the underlying basis for a recursive dynamic formulation.

4.1. RECURSIVE FORWARD KINEMATICS

The forward kinematics problem for a parallel manipulator consists of determining the position, twists and twist-rates of the platform and all the other links given only the actuated-joint angles, velocities and accelerations. Implicit in the final solution, is the intermediate determination of the unactuated (passive) joint angles, velocities and accelerations which are related to the actuated-joint angles through the loop-closure constraints. For more details, see [24].

4.1.1. *Position Analysis*

The displacement analysis is a critical first step that lies beyond the scope of this paper; we adopt the approach proposed by [28] to this end.

4.1.2. *Velocity Analysis*

Since the manipulator is planar, we use two-dimensional position vectors, three-dimensional twist vectors $\mathbf{t} = [\omega, \mathbf{v}]^T$, and three-dimensional wrench vectors $\mathbf{w} = [n, \mathbf{f}]^T$, where ω is the scalar angular velocity, $\mathbf{v}$ is the two-dimensional velocity vector, n is the scalar moment and $\mathbf{f}$ is the two-dimensional force vector. For *each chain*, we define position vectors $\mathbf{d}_i$ from the ith joint axis to the mass center of link i, $\mathbf{r}_i$ from mass center of link i to the $(i + 1)$ joint axis and $\mathbf{a}_i = \mathbf{d}_i + \mathbf{r}_i$ as shown in Figure 2. The twist of the end effector of *each chain* is given by [40] as

$$\mathbf{t}_P = \mathbf{B}_{P3}\mathbf{t}_3 \tag{24}$$

where

$$\mathbf{B}_{P3} = \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{Er}_3 & \mathbf{1} \end{bmatrix}; \quad \mathbf{E} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

For each serial chain, the twist of the ith link, $\mathbf{t}_i$ can then be computed recursively from the twist of the $(i-1)^{\text{th}}$ link and the ith twist generator as

$$\mathbf{t}_i = \mathbf{B}_{i,i-1}\mathbf{t}_{i-1} + \mathbf{p}_i\dot{\theta}_i$$

$$\mathbf{B}_{i,i-1} = \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{E}(\mathbf{r}_{i-1} + \mathbf{d}_i) & \mathbf{1} \end{bmatrix} \quad \mathbf{p}_i = \begin{bmatrix} 1 \\ \mathbf{E}\mathbf{d}_i \end{bmatrix} \tag{25}$$

where the 3×3 matrix $\mathbf{B}_{i,i-1}$ is the *twist-propagation matrix*; $\mathbf{t}_{i-1}$ is the twist of link $(i-1)$ with respect to its mass center; $\mathbf{p}_i$ is the *twist generator*; and $\dot{\theta}_i$ is the relative angular joint velocity of the ith joint, while $\mathbf{0}$ is the two-dimensional zero vector and $\mathbf{1}$ is the 2×2 identity matrix. An expression for $\mathbf{t}_3$ can be obtained based on the above equations and may be substituted into Equation (24), to yield

$$\mathbf{t}_P = \mathbf{B}_{P3}(\mathbf{B}_{32}\mathbf{t}_2 + \mathbf{p}_3\dot{\theta}_3) \tag{26}$$

Noting that the third joint is a passive joint, its contribution to the end effector twist may be eliminated by premultiplying the entire expression by $\boldsymbol{\Phi}_3$, the twist annihilator of $\tilde{\mathbf{p}}_3 = \mathbf{B}_{P3}\mathbf{p}_3$ (defined as shown in Appendix B) as

$$\boldsymbol{\Phi}_3\mathbf{t}_P = \boldsymbol{\Phi}_3\mathbf{B}_{P2}\mathbf{t}_2 \tag{27}$$

Similarly, by substituting the twist of link-2, $\mathbf{t}_2$, into Equation (27), we obtain

$$\boldsymbol{\Phi}_3\mathbf{t}_P = \boldsymbol{\Phi}_3\mathbf{B}_{P2}(\mathbf{B}_{21}\mathbf{t}_1 + \mathbf{p}_2\dot{\theta}_2) \tag{28}$$

The contribution of this second unactuated joint can similarly be eliminated from the final twist by premultiplying Equation (28) by $\bar{\boldsymbol{\Phi}}_2$, the twist annihilator of $\tilde{\mathbf{p}}_2 = \boldsymbol{\Phi}_3\mathbf{B}_{P2}\mathbf{p}_2$, to obtain

$$\boldsymbol{\Phi}_2\mathbf{t}_P = \bar{\boldsymbol{\Phi}}_2\boldsymbol{\Phi}_3\mathbf{t}_P = \bar{\boldsymbol{\Phi}}_2\boldsymbol{\Phi}_3\mathbf{B}_{P2}\mathbf{B}_{21}\mathbf{t}_1 \tag{29}$$

where the 3×3 matrix

$$\boldsymbol{\Phi}_2 = \bar{\boldsymbol{\Phi}}_2\boldsymbol{\Phi}_3 = \boldsymbol{\Phi}_3 - \tilde{\mathbf{p}}_2\tilde{\mathbf{p}}_2^T/\delta_2$$

Noting the similarity between Equations (27) and (29), the kinematics relationships may be written in recursive form (despite the presence of passive joints) as

$$\boldsymbol{\Phi}_i\mathbf{t}_P = \boldsymbol{\Phi}_i\mathbf{B}_{P,i-1}\mathbf{t}_{i-1} \tag{30}$$

where $\boldsymbol{\Phi}_i$ is evaluated recursively as

$$\tilde{\mathbf{p}}_i = \boldsymbol{\Phi}_{i+1}\mathbf{B}_{P,i}\mathbf{p}_i \quad \delta_i = \tilde{\mathbf{p}}_i^T\tilde{\mathbf{p}}_i$$

$$\boldsymbol{\Phi}_i = \left[\mathbf{1} - \tilde{\mathbf{p}}_i\tilde{\mathbf{p}}_i^T/\delta_i\right]\boldsymbol{\Phi}_{i+1} = \bar{\boldsymbol{\Phi}}_i\boldsymbol{\Phi}_{i+1} = \boldsymbol{\Phi}_{i+1} - \tilde{\mathbf{p}}_i\tilde{\mathbf{p}}_i^T/\delta_i$$

where the properties $\mathbf{B}_{P,i}\mathbf{B}_{i+1,i} = \mathbf{B}_{Pi}$ and $\mathbf{\Phi}_i^T \mathbf{\Phi}_i = \mathbf{\Phi}_i$ have been used. Finally, since joint 1 is actuated, substituting $\mathbf{t}_1 = \mathbf{p}_1 \dot{\theta}_1$ into Equation (29), we can express the twist of the platform $\mathcal{P}$ in terms of $\dot{\theta}_1$ as

$$\mathbf{\Phi}_2 \mathbf{t}_P = \mathbf{\Phi}_2 \mathbf{B}_{P1} \mathbf{p}_1 \dot{\theta}_1 \tag{31}$$

This equation illustrates a well known feature for parallel chains, i.e., in contrast to serial manipulators, parallel manipulators have two Jacobian matrices. Note that $\mathbf{\Phi}_2$ is a projection matrix and is thus singular. Hence, writing Equation (31) for *each open chain* we obtain

$$\mathbf{K}\mathbf{t}_P = \mathbf{\Phi}\mathbf{B}\mathbf{P}\dot{\theta}_A \quad \text{where}$$
$$\mathbf{K} = \left[\mathbf{\Phi}_2^I + \mathbf{\Phi}_2^{II} + \mathbf{\Phi}_2^{III} \right] \; : \; 3 \times 3$$
$$\mathbf{\Phi} = \left[\mathbf{\Phi}_2^I \quad \mathbf{\Phi}_2^{II} \quad \mathbf{\Phi}_2^{III} \right] \; : \; 3 \times 9$$
$$\mathbf{B} = \text{diag}\left(\mathbf{B}_{P1}^I, \mathbf{B}_{P1}^{II}, \mathbf{B}_{P1}^{III} \right) \; : \; 9 \times 9$$
$$\mathbf{P} = \text{diag}\left(\mathbf{p}_1^I, \mathbf{p}_1^{II}, \mathbf{p}_1^{III} \right) \; : \; 9 \times 3$$
$$\dot{\theta}_A = \left[\dot{\theta}_1^I \quad \dot{\theta}_1^{II} \quad \dot{\theta}_1^{III} \right]^T$$

where all dimensions have been stated for clarity. Finally when $\mathbf{K}$ is nonsingular,[4] we may solve for the end effector twist in terms of the actuated joint rates as

$$\mathbf{t}_P = \mathbf{K}^{-1}\mathbf{\Phi}\mathbf{B}\mathbf{P}\dot{\theta}_A \tag{32}$$

The unactuated joint rates may now be computed by back-substituting $\mathbf{t}_P$ to yield

$$\dot{\theta}_2 = \frac{\tilde{\mathbf{p}}_2^T}{\delta_2}(\mathbf{K}^{-1}\mathbf{\Phi}\mathbf{B}\mathbf{P}\dot{\theta}_A - \mathbf{B}_{P1}\mathbf{p}_1\dot{\theta}_1) \tag{33}$$

$$\dot{\theta}_3 = \frac{\tilde{\mathbf{p}}_3^T}{\delta_3}\mathbf{\Psi}_2^T(\mathbf{K}^{-1}\mathbf{\Phi}\mathbf{B}\mathbf{P}\dot{\theta}_A - \mathbf{B}_{P1}\mathbf{p}_1\dot{\theta}_1) \tag{34}$$

where the 3×3 matrix $\mathbf{\Psi}_2$ is defined as $\mathbf{\Psi}_2 = (\mathbf{1} - \mathbf{B}_{P2}\mathbf{p}_2\tilde{\mathbf{p}}_2^T/\delta_2)^T$ and $\mathbf{1}$ is the 3×3 identity matrix. We note that Equations (33) and (34) are general and applicable to *each open chain* and that the bracketed term on the right hand side of each equation is the same. Thus the final relationship between the joint rates and actuated joint rates is expressed in vector form as

$$\dot{\theta} = \begin{bmatrix} \bar{\mathbf{P}}^I & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \bar{\mathbf{P}}^{II} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \bar{\mathbf{P}}^{III} \end{bmatrix} \begin{bmatrix} \mathbf{L}^I \\ \mathbf{L}^{II} \\ \mathbf{L}^{III} \end{bmatrix} \mathbf{B}\mathbf{P}\dot{\theta}_A \tag{35}$$

---

[4] See [19] for a more detailed discussion on singularities of parallel manipulators.

where the 3×9 matrix $\bar{\mathbf{P}}_i$ is defined as $\bar{\mathbf{P}}_i = [\mathrm{diag}(\tilde{\mathbf{p}}_1^T/\delta_1, \tilde{\mathbf{p}}_2^T/\delta_2, \tilde{\mathbf{p}}_3^T \boldsymbol{\Psi}_2^T/\delta_3)]^i$, while $\tilde{\mathbf{p}}_1^i$ as explicitly $\tilde{\mathbf{p}}_1^i = (\mathbf{B}_{P1}\mathbf{p}_1)^i$ for $i = I, II$ and III, and the 9×9 matrices $\mathbf{L}^i$ are defined for *each open chain* as

$$\mathbf{L}^I = \begin{bmatrix} \mathbf{1} & \mathbf{O} & \mathbf{O} \\ [\mathbf{K}^{-1}\boldsymbol{\Phi}_2 - \mathbf{1}]^I & [\mathbf{K}^{-1}\boldsymbol{\Phi}_2]^{II} & [\mathbf{K}^{-1}\boldsymbol{\Phi}_2]^{III} \\ [\mathbf{K}^{-1}\boldsymbol{\Phi}_2 - \mathbf{1}]^I & [\mathbf{K}^{-1}\boldsymbol{\Phi}_2]^{II} & [\mathbf{K}^{-1}\boldsymbol{\Phi}_2]^{III} \end{bmatrix}$$

$$\mathbf{L}^{II} = \begin{bmatrix} \mathbf{O} & \mathbf{1} & \mathbf{O} \\ [\mathbf{K}^{-1}\boldsymbol{\Phi}_2]^I & [\mathbf{K}^{-1}\boldsymbol{\Phi}_2 - \mathbf{1}]^{II} & [\mathbf{K}^{-1}\boldsymbol{\Phi}_2]^{III} \\ [\mathbf{K}^{-1}\boldsymbol{\Phi}_2]^I & [\mathbf{K}^{-1}\boldsymbol{\Phi}_2 - \mathbf{1}]^{II} & [\mathbf{K}^{-1}\boldsymbol{\Phi}_2]^{III} \end{bmatrix}$$

$$\mathbf{L}^{III} = \begin{bmatrix} \mathbf{O} & \mathbf{O} & \mathbf{1} \\ [\mathbf{K}^{-1}\boldsymbol{\Phi}_2]^I & [\mathbf{K}^{-1}\boldsymbol{\Phi}_2]^{II} & [\mathbf{K}^{-1}\boldsymbol{\Phi}_2 - \mathbf{1}]^{III} \\ [\mathbf{K}^{-1}\boldsymbol{\Phi}_2]^I & [\mathbf{K}^{-1}\boldsymbol{\Phi}_2]^{II} & [\mathbf{K}^{-1}\boldsymbol{\Phi}_2 - \mathbf{1}]^{III} \end{bmatrix}$$

where $\mathbf{O}$ and $\mathbf{1}$ are 3×3 zero and identity matrices. Equation (35) can be written in compact form as

$$\dot{\boldsymbol{\theta}} = \bar{\mathbf{P}}\mathbf{L}\mathbf{B}\mathbf{P}\dot{\boldsymbol{\theta}}_A \tag{36}$$

where the 9×27 matrix $\bar{\mathbf{P}}$ is defined as $\bar{\mathbf{P}} = \mathrm{diag}(\bar{\mathbf{P}}^I, \bar{\mathbf{P}}^{II}, \bar{\mathbf{P}}^{III})$ and the 27×9 matrix $\mathbf{L}$ is defined as $\mathbf{L} = \left[(\mathbf{L}^I)^T \ (\mathbf{L}^{II})^T \ (\mathbf{L}^{III})^T \right]^T$. Note that, except for $\mathbf{L}$, which is full but still retains a special form, all other matrices are block-diagonal.

4.1.3. *Acceleration Analysis*

The acceleration terms, for any chain, are obtained by differentiating Equation (26) as

$$\dot{\mathbf{t}}_P = \dot{\mathbf{B}}_{P3}\mathbf{t}_3 + \mathbf{B}_{P3}(\dot{\mathbf{B}}_{32}\mathbf{t}_2 + \mathbf{B}_{32}\dot{\mathbf{t}}_2 + \dot{\mathbf{p}}_3\dot{\theta}_3) + \mathbf{B}_{P3}\mathbf{p}_3\ddot{\theta}_3 \tag{37}$$

Adopting a process similar to the one discussed for the velocity analysis, we may eliminate the passive joint acceleration contributions as

$$\boldsymbol{\Phi}_3\dot{\mathbf{t}}_P = \boldsymbol{\Phi}_3[\dot{\mathbf{B}}_{P3}\mathbf{t}_3 + \mathbf{B}_{P3}(\dot{\mathbf{B}}_{32}\mathbf{t}_2 + \mathbf{B}_{32}\dot{\mathbf{t}}_2 + \dot{\mathbf{p}}_3\dot{\theta}_3)] \tag{38}$$

Substituting $\mathbf{t}_3$ and $\dot{\theta}_3$ into Equation (38) and re-arranging,

$$\boldsymbol{\Phi}_3\dot{\mathbf{t}}_P = \boldsymbol{\Phi}_3\mathbf{B}_{P2}\dot{\mathbf{t}}_2 + \boldsymbol{\Phi}_3\dot{\mathbf{B}}_{P2}\mathbf{t}_2 - \left[\boldsymbol{\Phi}_3(\dot{\mathbf{B}}_{P3}\mathbf{p}_3 + \mathbf{B}_{P3}\dot{\mathbf{p}}_3)\frac{\tilde{\mathbf{p}}_3^T}{\delta_3} \right]\mathbf{B}_{P2}\mathbf{t}_2$$

$$+ \left[\boldsymbol{\Phi}_3(\dot{\mathbf{B}}_{P3}\mathbf{p}_3 + \mathbf{B}_{P3}\dot{\mathbf{p}}_3)\frac{\tilde{\mathbf{p}}_3^T}{\delta_3} \right]\mathbf{t}_P \tag{39}$$

where the property $(\dot{\mathbf{B}}_{P3}\mathbf{B}_{32} + \mathbf{B}_{P3}\dot{\mathbf{B}}_{32}) = \dot{\mathbf{B}}_{P2}$ has been used. An expression for $\dot{\boldsymbol{\Phi}}_i$ can be computed as:

$$\dot{\boldsymbol{\Phi}}_i = \boldsymbol{\Phi}_i(\dot{\mathbf{B}}_{P,i}\mathbf{p}_i + \mathbf{B}_{P,i}\dot{\mathbf{p}}_i)\frac{\tilde{\mathbf{p}}_i^T}{\delta_i} = -\boldsymbol{\Phi}_i\frac{\dot{\tilde{\mathbf{p}}}_i\tilde{\mathbf{p}}_i^T}{\delta_i} \tag{40}$$

Similarly, noting the general form for $\mathbf{t}_i$ we may derive

$$\dot{\mathbf{t}}_i = \dot{\mathbf{B}}_{i,i-1}\mathbf{t}_{i-1} + \mathbf{B}_{i,i-1}\dot{\mathbf{t}}_{i-1} + \mathbf{p}_i\ddot{\theta}_i + \dot{\mathbf{p}}_i\dot{\theta}_i \tag{41}$$

Substituting expressions for $\dot{\boldsymbol{\Phi}}_3$, $\mathbf{t}_2$ and $\dot{\mathbf{t}}_2$ into Equation (39) we obtain

$$(\dot{\boldsymbol{\Phi}}_3\mathbf{t}_P + \boldsymbol{\Phi}_3\dot{\mathbf{t}}_P) = \boldsymbol{\Phi}_3\mathbf{B}_{P2}\mathbf{p}_2\ddot{\theta}_2 + [\boldsymbol{\Phi}_3\mathbf{B}_{P2}(\dot{\mathbf{B}}_{21}\mathbf{t}_1 + \mathbf{B}_{21}\dot{\mathbf{t}}_1 + \dot{\mathbf{p}}_2\dot{\theta}_2)$$
$$+ \boldsymbol{\Phi}_3\dot{\mathbf{B}}_{P2}\mathbf{t}_2 + \dot{\boldsymbol{\Phi}}_3\mathbf{B}_{P2}\mathbf{t}_2] \tag{42}$$

The passive joint acceleration contributions from the second axis can be eliminated by premultiplying the Equation (42) with $\bar{\boldsymbol{\Phi}}_2$, the twist annihilator of $\tilde{\mathbf{p}}_2 = \boldsymbol{\Phi}_3\mathbf{B}_{P2}\mathbf{p}_2$ as

$$\bar{\boldsymbol{\Phi}}_2(\dot{\boldsymbol{\Phi}}_3\mathbf{t}_P + \boldsymbol{\Phi}_3\dot{\mathbf{t}}_p) = \bar{\boldsymbol{\Phi}}_2[\boldsymbol{\Phi}_3\mathbf{B}_{P2}(\dot{\mathbf{B}}_{21}\mathbf{t}_1 + \mathbf{B}_{21}\dot{\mathbf{t}}_1 + \dot{\mathbf{p}}_2\dot{\theta}_2)$$
$$+ \boldsymbol{\Phi}_3\dot{\mathbf{B}}_{P2}\mathbf{t}_2 + \dot{\boldsymbol{\Phi}}_3\mathbf{B}_{P2}\mathbf{t}_2] \tag{43}$$

Noting that $\bar{\boldsymbol{\Phi}}_2\boldsymbol{\Phi}_3 = \boldsymbol{\Phi}_2$ and rearranging Equation (43) leads to:

$$\ddot{\theta}_2 = \frac{\tilde{\mathbf{p}}_2^T}{\delta_2}[\boldsymbol{\Phi}_3(\dot{\mathbf{t}}_P - \mathbf{a}_1) - \mathbf{a}_2] \tag{44}$$

$$\boldsymbol{\Phi}_2\dot{\mathbf{t}}_P = \boldsymbol{\Phi}_2\mathbf{a}_1 + \bar{\boldsymbol{\Phi}}_2\mathbf{a}_2 \tag{45}$$

where $\mathbf{a}_1 = \mathbf{B}_{P2}(\dot{\mathbf{B}}_{21}\mathbf{t}_1 + \mathbf{B}_{21}\dot{\mathbf{t}}_1 + \dot{\mathbf{p}}_2\dot{\theta}_2) + \dot{\mathbf{B}}_{P2}\mathbf{t}_2$ and $\mathbf{a}_2 = \dot{\boldsymbol{\Phi}}_3(\mathbf{B}_{P2}\mathbf{t}_2 - \mathbf{t}_P)$. Finally adding Equation (45) for *each open chain* and solving for $\dot{\mathbf{t}}_P$,

$$\dot{\mathbf{t}}_P = \mathbf{K}^{-1}([\boldsymbol{\Phi}_2\mathbf{a}_1 + \bar{\boldsymbol{\Phi}}_2\mathbf{a}_2]^I + [\boldsymbol{\Phi}_2\mathbf{a}_1 + \bar{\boldsymbol{\Phi}}_2\mathbf{a}_2]^{II} + [\boldsymbol{\Phi}_2\mathbf{a}_1 + \bar{\boldsymbol{\Phi}}_2\mathbf{a}_2]^{III}) \tag{46}$$

4.1.4. *Summary of Forward Kinematics*

The overall process of computing the forward kinematics can be summarized as:

1. With the data, calculate $\mathbf{B}_{21}$, $\mathbf{B}_{32}$, $\mathbf{B}_{31}$, $\mathbf{B}_{P3}$,$\mathbf{B}_{P2}$, $\mathbf{B}_{P1}$, $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$. For each chain,

$$\tilde{\mathbf{p}}_3 = \mathbf{B}_{P3}\mathbf{p}_3$$

$$\delta_3 = \tilde{\mathbf{p}}_3^T \tilde{\mathbf{p}}_3$$

$$\Phi_3 = \left[\mathbf{1} - \frac{\tilde{\mathbf{p}}_3 \tilde{\mathbf{p}}_3^T}{\delta_3} \right]$$

$$\tilde{\mathbf{p}}_2 = \Phi_3 B_{P2}\mathbf{p}_2$$

$$\delta_2 = \tilde{\mathbf{p}}_2^T \tilde{\mathbf{p}}_2$$

$$\bar{\Phi}_2 = \left[\mathbf{1} - \frac{\tilde{\mathbf{p}}_2 \tilde{\mathbf{p}}_2^T}{\delta_2} \right]$$

$$\Phi_2 = \bar{\Phi}_2 \Phi_3$$

2. Form matrices $\mathbf{K}$, Φ, $\mathbf{B}$ and $\mathbf{P}$ with values received from each chain and calculate the platform twist $\mathbf{t}_P$ from:

$$\mathbf{K}\mathbf{t}_P = \Phi\mathbf{B}\mathbf{P}\dot{\theta}_A$$

3. Obtain the twists and joint rates recursively for each chain, using $\mathbf{t}_P$ calculated by the central processor

$$\mathbf{t}_1 = \mathbf{p}_1 \dot{\theta}_1$$

$$\dot{\theta}_2 = \frac{\tilde{\mathbf{p}}_2^T}{\delta_2}(\mathbf{t}_P - \mathbf{B}_{P1}\mathbf{t}_1)$$

$$\mathbf{t}_2 = \mathbf{B}_{21}\mathbf{t}_1 + \mathbf{p}_2 \dot{\theta}_2$$

$$\dot{\theta}_3 = \frac{\tilde{\mathbf{p}}_3^T}{\delta_3}(\mathbf{t}_P - \mathbf{B}_{P2}\mathbf{t}_2)$$

$$\mathbf{t}_3 = \mathbf{B}_{32}\mathbf{t}_2 + \mathbf{p}_3 \dot{\theta}_3$$

Now calculate the twist rates and joint accelerations. First, calculate $\dot{\mathbf{B}}_{21}$, $\dot{\mathbf{B}}_{32}$, $\dot{\mathbf{B}}_{31}$, $\dot{\mathbf{B}}_{P3}$,$\dot{\mathbf{B}}_{P2}$, $\dot{\mathbf{B}}_{P1}$, $\dot{\mathbf{p}}_1$, $\dot{\mathbf{p}}_2$ and $\dot{\mathbf{p}}_3$ using the joint rates calculated above

$$\dot{\mathbf{t}}_1 = \mathbf{p}_1 \ddot{\theta}_1 + \dot{\mathbf{p}}_1 \dot{\theta}_1$$

$$\mathbf{a}_1 = \mathbf{B}_{P2}(\dot{\mathbf{B}}_{21}\mathbf{t}_1 + \mathbf{B}_{21}\dot{\mathbf{t}}_1 + \dot{\mathbf{p}}_2 \dot{\theta}_2) + \dot{\mathbf{B}}_{P2}\mathbf{t}_2$$

$$\dot{\Phi}_3 = -\Phi_3(\dot{\mathbf{B}}_{P3}\mathbf{p}_3 + \mathbf{B}_{P3}\dot{\mathbf{p}}_3)\frac{\tilde{\mathbf{p}}_3^T}{\delta_3}$$

$$\mathbf{a}_2 = \dot{\Phi}_3(\mathbf{B}_{P2}\mathbf{t}_2 - \mathbf{t}_P)$$

$$\bar{\Phi}_2 = \mathbf{1} - \frac{\tilde{\mathbf{p}}_2 \tilde{\mathbf{p}}_2^T}{\delta_2}$$

4. Using $\boldsymbol{\Phi}_2$, $\bar{\boldsymbol{\Phi}}_2$, $\mathbf{a}_1$ and $\mathbf{a}_2$ from each chain, calculate $\dot{\mathbf{t}}_P$ from:

$$\mathbf{K}\dot{\mathbf{t}}_P = [\boldsymbol{\Phi}_2\mathbf{a}_1 + \bar{\boldsymbol{\Phi}}_2\mathbf{a}_2]^I + [\boldsymbol{\Phi}_2\mathbf{a}_1 + \bar{\boldsymbol{\Phi}}_2\mathbf{a}_2]^{II} + [\boldsymbol{\Phi}_2\mathbf{a}_1 + \bar{\boldsymbol{\Phi}}_2\mathbf{a}_2]^{III}$$

5. Calculate the joint accelerations and twist rates for each chain:

$$\ddot{\theta}_2 = \frac{\tilde{\mathbf{p}}_2^T}{\delta_2}[\boldsymbol{\Phi}_3(\dot{\mathbf{t}}_P - \mathbf{a}_1) - \mathbf{a}_2]$$

$$\dot{\mathbf{t}}_2 = \dot{\mathbf{B}}_{21}\mathbf{t}_1 + \mathbf{B}_{21}\dot{\mathbf{t}}_1 + \dot{\mathbf{p}}_2\dot{\theta}_2 + \mathbf{p}_2\ddot{\theta}_2$$

$$\ddot{\theta}_3 = \frac{\tilde{\mathbf{p}}_3^T}{\delta_3}[\dot{\mathbf{t}}_P - \dot{\mathbf{B}}_{P3}\mathbf{t}_3 - \mathbf{B}_{P3}(\dot{\mathbf{B}}_{32}\mathbf{t}_2 + \mathbf{B}_{32}\dot{\mathbf{t}}_2 + \dot{\mathbf{p}}_3\dot{\theta}_3)]$$

$$\dot{\mathbf{t}}_3 = \mathbf{B}_{32}\dot{\mathbf{t}}_2 + \dot{\mathbf{B}}_{32}\mathbf{t}_2 + \mathbf{p}_3\ddot{\theta}_3 + \dot{\mathbf{p}}_3\dot{\theta}_3$$

4.2. INVERSE DYNAMICS

The inverse-dynamics problem is defined as: Given the time-histories of all the system degrees-of-freedom, compute the time-histories of the controlling actuated joint torques and forces. As in the case of the kinematics calculations, we again cut the platform into three parts and assign cut sections of platform $\mathcal{P}$ to each open chain. Each cut section thus becomes the "third link" of the corresponding chain. Further, we divide the mass of the platform (including any tool carried by the platform) and assign its corresponding moment of inertia, with respect to the mass center of the platform, to the "third link" of each chain. The Newton-Euler equations for *each open chain* are thus,

$$\mathbf{M}\dot{\mathbf{t}} + \dot{\mathbf{M}}\mathbf{t} = \mathbf{w}^A + \underbrace{\mathbf{w}^W + \mathbf{w}^g}_{\mathbf{w}^G} + \mathbf{w}^C \tag{47}$$

where $\mathbf{M}$ is the 9×9 mass matrix, $\mathbf{t}$ is the 9-dimensional twist vector of the whole chain, $\mathbf{w}^A$ is the wrench applied by the actuators, $\mathbf{w}^W$ is the working wrench applied at the platform, $\mathbf{w}^g$ is the gravity wrench and $\mathbf{w}^C$ are the constraint wrenches, all these being 9-dimensional vectors. The friction forces have been neglected for the sake of simplicity, but, these can be readily incorporated into the model by means of a Rayleigh dissipation function [2]. In particular, the twist vector $\mathbf{t}$ may now be written as [38]

$$\mathbf{t} = \mathbf{N}_l\mathbf{N}_d\dot{\boldsymbol{\theta}} \tag{48}$$

where $\mathbf{N}_l\mathbf{N}_d$ is the decoupled orthogonal complement and $\dot{\boldsymbol{\theta}}$ is the joint-rate vector of the chain. These matrices $\mathbf{N}_l$ and $\mathbf{N}_d$ can also be identified with the spatial operators of Rodriguez and Kreutz-Delgado [35] and are "anti-causal" and "memory-less" respectively. For our manipulator, and for *each open chain*, Equation (48) becomes,

in block form,

$$
\underbrace{\begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{t}_3 \end{bmatrix}}_{\mathbf{t}} = \underbrace{\begin{bmatrix} \mathbf{1} & \mathbf{O} & \mathbf{O} \\ \mathbf{B}_{21} & \mathbf{1} & \mathbf{O} \\ \mathbf{B}_{31} & \mathbf{B}_{32} & \mathbf{1} \end{bmatrix}}_{\mathbf{N}_l} \underbrace{\begin{bmatrix} \mathbf{p}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{p}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{p}_3 \end{bmatrix}}_{\mathbf{N}_d} \underbrace{\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}}_{\dot{\boldsymbol{\theta}}} \tag{49}
$$

where all terms have been previously defined. Now, the constraint wrenches $\mathbf{w}^C$ do not develop any power, and hence, $\mathbf{t}^T \mathbf{w}^C$ is 0; by virtue of Equation (48), $\mathbf{w}^C$ lies in the nullspace of $\mathbf{N}_d^T \mathbf{N}_l^T$. To eliminate joint constraint wrenches, we pre-multiply both sides of Equation (47) by $\mathbf{N}_d^T \mathbf{N}_l^T$, and noting that, for planar manipulators $\dot{\mathbf{M}} = \mathbf{O}$ [2],

$$
\mathbf{N}_d^T \mathbf{N}_l^T \mathbf{M} \dot{\mathbf{t}} = \tilde{\boldsymbol{\tau}} + \mathbf{N}_d^T \mathbf{N}_l^T \mathbf{w}^G \tag{50}
$$

Time differentiating Equation (48) and substituting the expression for $\dot{\mathbf{t}}$ in Equation (50),

$$
\mathbf{N}_d^T \mathbf{N}_l^T [\mathbf{M} \mathbf{N}_l \mathbf{N}_d \ddot{\boldsymbol{\theta}} + (\mathbf{M} \dot{\mathbf{N}}_l \mathbf{N}_d + \mathbf{M} \mathbf{N}_l \dot{\mathbf{N}}_d) \dot{\boldsymbol{\theta}}] = \tilde{\boldsymbol{\tau}} + \mathbf{N}_d^T \mathbf{N}_l^T \mathbf{w}^G \tag{51}
$$

where $\tilde{\boldsymbol{\tau}} = \mathbf{N}_d^T \mathbf{N}_l^T \mathbf{w}^A$ is the joint torque vector for the chain and is given by $\tilde{\boldsymbol{\tau}} = \begin{bmatrix} \tilde{\tau}_1 & \tilde{\tau}_2 & \tilde{\tau}_3 \end{bmatrix}^T$ for *each open chain*. We can write the above equation in compact form as

$$
\mathbf{I} \ddot{\boldsymbol{\theta}} + \mathbf{C} \dot{\boldsymbol{\theta}} = \tilde{\boldsymbol{\tau}} + \boldsymbol{\tau}^G \tag{52}
$$

where $\mathbf{I} = \mathbf{N}_d^T \mathbf{N}_l^T \mathbf{M} \mathbf{N}_l \mathbf{N}_d$ and $\mathbf{C} = \mathbf{N}_d^T \mathbf{N}_l^T (\mathbf{M} \dot{\mathbf{N}}_l \mathbf{N}_d + \mathbf{M} \mathbf{N}_l \dot{\mathbf{N}}_d)$, $\mathbf{I}$ being the *generalized joint-space inertia matrix of the chain* and $\mathbf{C}$ the joint-space matrix of coriolis and centrifugal forces.

4.2.1. *Projecting Joint Torques onto Minimal-Coordinate Space*

As a second step, we write the dynamics equation for *each open chain* and couple them with Lagrange multipliers, thereby obtaining the dynamics equation of the whole manipulator as

$$
\begin{bmatrix} [\mathbf{I} \ddot{\boldsymbol{\theta}} + \mathbf{C} \dot{\boldsymbol{\theta}}]^I \\ [\mathbf{I} \ddot{\boldsymbol{\theta}} + \mathbf{C} \dot{\boldsymbol{\theta}}]^{II} \\ [\mathbf{I} \ddot{\boldsymbol{\theta}} + \mathbf{C} \dot{\boldsymbol{\theta}}]^{III} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}^I \\ \boldsymbol{\tau}^{II} \\ \boldsymbol{\tau}^{III} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\tau}_G^I \\ \boldsymbol{\tau}_G^{II} \\ \boldsymbol{\tau}_G^{III} \end{bmatrix} - \mathbf{A}^T \boldsymbol{\lambda} \tag{53}
$$

where $\mathbf{A}$ is the *loop-closure constraint Jacobian* of the constraints in differential form $\mathbf{A} \dot{\boldsymbol{\theta}} = \mathbf{0}$. Now, by defining $\dot{\boldsymbol{\theta}} = \mathbf{J} \dot{\boldsymbol{\theta}}_A$ and substituting in the above equation

we obtain

$$\mathbf{A}\mathbf{J}\dot{\boldsymbol{\theta}}_A = \mathbf{0} \tag{54}$$

Since values for $\dot{\boldsymbol{\theta}}_A$ can be assigned arbitrarily, to satisfy Equation (54), $\mathbf{J}$ must lie in the null space of $\mathbf{A}$ and may be called the *loop-closure orthogonal complement*. Pre-multiplying both sides of Equation (53) by $\mathbf{J}^T$ we obtain

$$\mathbf{J}^T \begin{bmatrix} [\mathbf{I}\ddot{\theta} + \mathbf{C}\dot{\theta} - \boldsymbol{\tau}_G]^I \\ [\mathbf{I}\ddot{\theta} + \mathbf{C}\dot{\theta} - \boldsymbol{\tau}_G]^{II} \\ [\mathbf{I}\ddot{\theta} + \mathbf{C}\dot{\theta} - \boldsymbol{\tau}_G]^{III} \end{bmatrix} = \mathbf{J}^T \begin{bmatrix} \tilde{\boldsymbol{\tau}}^I \\ \tilde{\boldsymbol{\tau}}^{II} \\ \tilde{\boldsymbol{\tau}}^{III} \end{bmatrix} = \boldsymbol{\tau}_A \tag{55}$$

where $\boldsymbol{\tau}_A$ is the vector of actuator torques. Notice that the bracketed terms are nothing more than $\tilde{\boldsymbol{\tau}}^j$, which can be found for *each open chain*, for $j = I, II$ and III, recursively [38]. The $\mathbf{J}$ noted in Equation (36) i.e. $\mathbf{J} = \bar{\mathbf{P}}\mathbf{L}\mathbf{B}\mathbf{P}$ can be re-written in a slightly different and expanded form as

$$\mathbf{J} = \underbrace{\begin{bmatrix} \bar{\mathbf{P}}^I & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \bar{\mathbf{P}}^{II} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \bar{\mathbf{P}}^{III} \end{bmatrix}}_{\bar{\mathbf{P}}} \underbrace{\begin{bmatrix} \mathbf{1} & \mathbf{O} & \mathbf{O} \\ (\mathbf{S}^I - \mathbf{1}) & \mathbf{S}^{II} & \mathbf{S}^{III} \\ (\mathbf{S}^I - \mathbf{1}) & \mathbf{S}^{II} & \mathbf{S}^{III} \\ \mathbf{O} & \mathbf{1} & \mathbf{O} \\ \mathbf{S}^I & (\mathbf{S}^{II} - \mathbf{1}) & \mathbf{S}^{III} \\ \mathbf{S}^I & (\mathbf{S}^{II} - \mathbf{1}) & \mathbf{S}^{III} \\ \mathbf{O} & \mathbf{O} & \mathbf{1} \\ \mathbf{S}^I & \mathbf{S}^{II} & (\mathbf{S}^{III} - \mathbf{1}) \\ \mathbf{S}^I & \mathbf{S}^{II} & (\mathbf{S}^{III} - \mathbf{1}) \end{bmatrix}}_{\mathbf{LB}} \underbrace{\begin{bmatrix} \tilde{\mathbf{p}}_1^I & 0 & 0 \\ 0 & \tilde{\mathbf{p}}_1^{II} & 0 \\ 0 & 0 & \tilde{\mathbf{p}}_1^{III} \end{bmatrix}}_{\mathbf{P}}$$

where $\mathbf{S}^j = \mathbf{K}^{-1}\boldsymbol{\Phi}_2^j$ for $j = I, II$ and III. Substituting $\mathbf{J}$ into Equation (55) and rearranging we obtain the actuated torques as

$$\begin{aligned}
\tau_1^I &= \left(\tilde{\mathbf{p}}_1^I\right)^T \big[(\mathbf{S}^I)^T \big(\check{\mathbf{p}}_2^I + \check{\mathbf{p}}_3^I + \check{\mathbf{p}}_2^{II} + \check{\mathbf{p}}_3^{II} + \check{\mathbf{p}}_2^{III} + \check{\mathbf{p}}_3^{III}\big) \\
&\quad + \check{\mathbf{p}}_1^I - \check{\mathbf{p}}_2^I - \check{\mathbf{p}}_3^I \big] \\
\tau_1^{II} &= \left(\tilde{\mathbf{p}}_1^{II}\right)^T \big[(\mathbf{S}^{II})^T \big(\check{\mathbf{p}}_2^I + \check{\mathbf{p}}_3^I + \check{\mathbf{p}}_2^{II} + \check{\mathbf{p}}_3^{II} + \check{\mathbf{p}}_2^{III} + \check{\mathbf{p}}_3^{III}\big) \\
&\quad + \check{\mathbf{p}}_1^{II} - \check{\mathbf{p}}_2^{II} - \check{\mathbf{p}}_3^{II} \big] \\
\tau_1^{III} &= \left(\tilde{\mathbf{p}}_1^{III}\right)^T \big[(\mathbf{S}^{III})^T \big(\check{\mathbf{p}}_2^I + \check{\mathbf{p}}_3^I + \check{\mathbf{p}}_2^{II} + \check{\mathbf{p}}_3^{II} + \check{\mathbf{p}}_2^{III} + \check{\mathbf{p}}_3^{III}\big) \\
&\quad + \check{\mathbf{p}}_1^{III} - \check{\mathbf{p}}_2^{III} - \check{\mathbf{p}}_3^{III} \big]
\end{aligned}$$

where $\check{\mathbf{p}}_k^j = [\tilde{\tau}_k \boldsymbol{\Psi}_{k-1} \tilde{\mathbf{p}}_k / \delta_k]^j$ for $k = 1$, 2 and 3 and $j = I$, II and III, where $\boldsymbol{\Psi}_0$ and $\boldsymbol{\Psi}_1$ are identical with the three-dimensional identity matrix. But $\tilde{\mathbf{p}}_1^T \check{\mathbf{p}}_1 = \tilde{\tau}_1$, and hence, the above equation set is written finally as

$$\boldsymbol{\tau}_A = \begin{bmatrix} \tilde{\tau}_1^I + \left(\tilde{\mathbf{p}}_1^I\right)^T [(\mathbf{S}^I)^T (\bar{\mathbf{p}}^I + \bar{\mathbf{p}}^{II} + \bar{\mathbf{p}}^{III}) - \bar{\mathbf{p}}^I] \\ \tilde{\tau}_1^{II} + \left(\tilde{\mathbf{p}}_1^{II}\right)^T [(\mathbf{S}^{II})^T (\bar{\mathbf{p}}^I + \bar{\mathbf{p}}^{II} + \bar{\mathbf{p}}^{III}) - \bar{\mathbf{p}}^{II}] \\ \tilde{\tau}_1^{III} + \left(\tilde{\mathbf{p}}_1^{III}\right)^T [(\mathbf{S}^{III})^T (\bar{\mathbf{p}}^I + \bar{\mathbf{p}}^{II} + \bar{\mathbf{p}}^{III}) - \bar{\mathbf{p}}^{III}] \end{bmatrix} \qquad (56)$$

where $\bar{\mathbf{p}} = \check{\mathbf{p}}_2 + \check{\mathbf{p}}_3$ for corresponding chains.

4.2.2. *Summary of Inverse Dynamics*

To summarize the process of computation of the inverse dynamics,

1. From Equation (50),

$$\tilde{\boldsymbol{\tau}} = \mathbf{N}_d^T \mathbf{N}_l^T (\mathbf{M}\dot{\mathbf{t}} + \mathbf{w}^G)$$

which can be calculated recursively for each chain as follows

$$\boldsymbol{\gamma}_3 = \left(\mathbf{M}_3 \dot{\mathbf{t}}_3 + \mathbf{w}_3^G\right)$$
$$\boldsymbol{\gamma}_2 = \left(\mathbf{M}_2 \dot{\mathbf{t}}_2 + \mathbf{w}_2^G\right) + \mathbf{B}_{32}^T \boldsymbol{\gamma}_3$$
$$\boldsymbol{\gamma}_1 = \left(\mathbf{M}_1 \dot{\mathbf{t}}_1 + \mathbf{w}_1^G\right) + \mathbf{B}_{21}^T \boldsymbol{\gamma}_2$$
$$\tilde{\tau}_3 = \mathbf{p}_3^T \boldsymbol{\gamma}_3$$
$$\tilde{\tau}_2 = \mathbf{p}_2^T \boldsymbol{\gamma}_2$$
$$\tilde{\tau}_1 = \mathbf{p}_1^T \boldsymbol{\gamma}_1$$

Now we calculate $\bar{\mathbf{p}}$ for each open chain

$$\bar{\mathbf{p}} = \tilde{\tau}_2 \frac{\tilde{\mathbf{p}}_2}{\delta_2} + \tilde{\tau}_3 \frac{\boldsymbol{\Psi}_2 \tilde{\mathbf{p}}_3}{\delta_3}$$

2. Add all $\bar{\mathbf{p}}$ from each open chain
3. Calculate the actuated joint torques

   Chain I:

$$\tau_1^I = \tilde{\tau}_1^I + \left(\tilde{\mathbf{p}}_1^I\right)^T [(\mathbf{S}^I)^T (\bar{\mathbf{p}}^I + \bar{\mathbf{p}}^{II} + \bar{\mathbf{p}}^{III}) - \bar{\mathbf{p}}^I]$$

Chain II:

$$\tau_1^{II} = \tilde{\tau}_1^{II} + \left(\tilde{\mathbf{p}}_1^{II}\right)^T[(\mathbf{S}^{II})^T(\bar{\mathbf{p}}^I + \bar{\mathbf{p}}^{II} + \bar{\mathbf{p}}^{III}) - \bar{\mathbf{p}}^{II}]$$

Chain III:

$$\tau_1^{III} = \tilde{\tau}_1^{III} + \left(\tilde{\mathbf{p}}_1^{III}\right)^T[(\mathbf{S}^{III})^T(\bar{\mathbf{p}}^I + \bar{\mathbf{p}}^{II} + \bar{\mathbf{p}}^{III}) - \bar{\mathbf{p}}^{III}]$$

4.3. FORWARD DYNAMICS

Now we obtain expressions for the forward dynamics of the manipulator. Re-writing Equation (55) in a slightly different form, we obtain

$$\mathbf{J}^T \left(\underbrace{\begin{bmatrix} \mathbf{I}^I & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{I}^{II} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{I}^{III} \end{bmatrix}}_{\bar{\mathbf{I}}} \begin{bmatrix} \ddot{\theta}^I \\ \ddot{\theta}^{II} \\ \ddot{\theta}^{III} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{C}^I & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{C}^{II} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & C^{III} \end{bmatrix}}_{\bar{\mathbf{C}}} \begin{bmatrix} \dot{\theta}^I \\ \dot{\theta}^{II} \\ \dot{\theta}^{III} \end{bmatrix} - \underbrace{\begin{bmatrix} \tau_G^I \\ \tau_G^{II} \\ \tau_G^{III} \end{bmatrix}}_{\bar{\tau}_G} \right) = \tau_A$$

$$\mathbf{J}^T(\bar{\mathbf{I}}\ddot{\theta} + \bar{\mathbf{C}}\dot{\theta} - \bar{\tau}_G) = \tau_A$$

Moreover, $\dot{\theta} = \mathbf{J}\dot{\theta}_A$, and hence $\ddot{\theta} = \mathbf{J}\ddot{\theta}_A + \dot{\mathbf{J}}\dot{\theta}_A$. Substituting these expressions into the above equation, we obtain

$$(\mathbf{J}^T\bar{\mathbf{I}}\mathbf{J})\ddot{\theta}_A = -\mathbf{J}^T(\bar{\mathbf{I}}\dot{\mathbf{J}}\dot{\theta}_A + \bar{\mathbf{C}}\mathbf{J}\dot{\theta}_A - \bar{\tau}_G) + \tau_A \qquad (57)$$

which is the minimal-order dynamics equation sought. The left-hand side matrix coefficient is the *generalized inertia matrix* of the manipulator. The right-hand side of the equation may be gathered in a single vector $\bar{\tau}$ and may be computed recursively by using the above inverse dynamics algorithm for $\ddot{\theta}_A = \mathbf{0}$, as originally suggested by [49]. Each diagonal block of $\bar{\mathbf{I}}$ is the *generalized inertia matrix of each serial chain* and may be computed recursively, as suggested by [38], using the DeNOC matrices. The *loop-closure orthogonal complement* $\mathbf{J} = \bar{\mathbf{P}}\mathbf{L}\mathbf{B}\mathbf{P}$ is, in turn,

$$\mathbf{J} = \underbrace{\begin{bmatrix} \bar{\mathbf{P}}^I & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \bar{\mathbf{P}}^{II} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \bar{\mathbf{P}}^{III} \end{bmatrix}}_{\mathbf{T}} \begin{bmatrix} \mathbf{L}^I \\ \mathbf{L}^{II} \\ \mathbf{L}^{III} \end{bmatrix} \underbrace{\mathbf{B}\mathbf{P}}_{\bar{\mathbf{B}}}$$

or

$$\mathbf{J} = \begin{bmatrix} \mathbf{T}^I \\ \mathbf{T}^{II} \\ \mathbf{T}^{III} \end{bmatrix} \bar{\mathbf{B}}$$

Substituting $\mathbf{J}$ into Equation (57), we obtain

$$\bar{\mathbf{B}}^T \left[(\mathbf{T}^I)^T \ (\mathbf{T}^{II})^T \ (\mathbf{T}^{III})^T \right] \begin{bmatrix} \mathbf{I}^I & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{I}^{II} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{I}^{III} \end{bmatrix} \begin{bmatrix} \mathbf{T}^I \\ \mathbf{T}^{II} \\ \mathbf{T}^{III} \end{bmatrix} \bar{\mathbf{B}}\ddot{\theta}_A = \bar{\tau}$$

or

$$\bar{\mathbf{B}}^T \left(\sum_{i=I}^{III} [\mathbf{T}^T \mathbf{I} \mathbf{T}]^i \right) \bar{\mathbf{B}}\ddot{\theta}_A = \bar{\tau} \tag{58}$$

The terms in parentheses are the contributions from the separate chains and can be distributed. The 9×9 matrix $[\mathbf{T}^T\mathbf{IT}]^j$, for $j = I, \ II$ and III, is written in block form as

$$[\mathbf{T}^T\mathbf{IT}]^j = \begin{bmatrix} \mathbf{L}_{11} & & \text{sym} \\ \mathbf{L}_{21} & \mathbf{L}_{22} & \\ \mathbf{L}_{31} & \mathbf{L}_{32} & \mathbf{L}_{33} \end{bmatrix}^j \tag{59}$$

where each $\mathbf{L}_{(k,l)}$ is a 3×3 block as given below, for each chain

Chain I
$$\mathbf{L}_{11}^I = \mathbf{I}_{1,1}^I \mathbf{a}_1^I (\mathbf{a}_1^I)^T + \mathbf{a}_1^I (\check{\mathbf{a}}_1^I)^T \mathbf{A}^I + (\mathbf{A}^I)^T \check{\mathbf{a}}_1^I (\mathbf{a}_1^I)^T + (\mathbf{A}^I)^T \bar{\mathbf{A}}^I \mathbf{A}^I$$
$$\mathbf{L}_{21}^I = (\mathbf{S}^{II})^T [\check{\mathbf{a}}_1^I (\mathbf{a}_1^I)^T + \bar{\mathbf{A}}^I \mathbf{A}^I]$$
$$\mathbf{L}_{22}^I = (\mathbf{S}^{II})^T \bar{\mathbf{A}}^I \mathbf{S}^{II}$$
$$\mathbf{L}_{31}^I = (\mathbf{S}^{III})^T [\check{\mathbf{a}}_1^I (\mathbf{a}_1^I)^T + \bar{\mathbf{A}}^I \mathbf{A}^I]$$
$$\mathbf{L}_{32}^I = (\mathbf{S}^{III})^T \bar{\mathbf{A}}^I \mathbf{S}^{II}$$
$$\mathbf{L}_{33}^I = (\mathbf{S}^{III})^T \bar{\mathbf{A}}^I \mathbf{S}^{III}$$
Chain II
$$\mathbf{L}_{11}^{II} = (\mathbf{S}^I)^T \bar{\mathbf{A}}^{II} \mathbf{S}^I$$
$$\mathbf{L}_{21}^{II} = [\mathbf{a}_1^{II} (\check{\mathbf{a}}_1^{II})^T + (\mathbf{A}^{II})^T \bar{\mathbf{A}}^{II}] \mathbf{S}^I$$
$$\mathbf{L}_{22}^{II} = \mathbf{I}_{1,1}^{II} \mathbf{a}_1^{II} (\mathbf{a}_1^{II})^T + \mathbf{a}_1^{II} (\check{\mathbf{a}}_1^{II})^T \mathbf{A}^{II} + (\mathbf{A}^{II})^T \check{\mathbf{a}}_1^{II} (\mathbf{a}_1^{II})^T$$
$$+ (\mathbf{A}^{II})^T \bar{\mathbf{A}}^{II} \mathbf{A}^{II}$$

$$\mathbf{L}_{31}^{II} = (\mathbf{S}^{III})^T \bar{\mathbf{A}}^{II} \mathbf{S}^I$$
$$\mathbf{L}_{32}^{II} = (\mathbf{S}^{III})^T \left[\breve{\mathbf{a}}_1^{II} \left(\mathbf{a}_1^{II} \right)^T + \bar{\mathbf{A}}^{II} \mathbf{A}^{II} \right]$$
$$\mathbf{L}_{33}^{II} = (\mathbf{S}^{III})^T \bar{\mathbf{A}}^{II} \mathbf{S}^{III}$$

Chain III
$$\mathbf{L}_{11}^{III} = (\mathbf{S}^I)^T \bar{\mathbf{A}}^{III} \mathbf{S}^I$$
$$\mathbf{L}_{21}^{III} = (\mathbf{S}^{II})^T \bar{\mathbf{A}}^{III} \mathbf{S}^I$$
$$\mathbf{L}_{22}^{III} = (\mathbf{S}^{II})^T \bar{\mathbf{A}}^{III} \mathbf{S}^{II}$$
$$\mathbf{L}_{31}^{III} = \left[\mathbf{a}_1^{III} \left(\breve{\mathbf{a}}_1^{III} \right)^T + (\mathbf{A}^{III})^T \bar{\mathbf{A}}^{III} \right] \mathbf{S}^I$$
$$\mathbf{L}_{32}^{III} = \left[\mathbf{a}_1^{III} \left(\breve{\mathbf{a}}_1^{III} \right)^T + (\mathbf{A}^{III})^T \bar{\mathbf{A}}^{III} \right] \mathbf{S}^{II}$$
$$\mathbf{L}_{33}^{III} = \mathbf{I}_{1,1}^{III} \mathbf{a}_1^{III} \left(\mathbf{a}_1^{III} \right)^T + \mathbf{a}_1^{III} (\breve{\mathbf{a}}_1^{III})^T \mathbf{A}^{III} + (\mathbf{A}^{III})^T \breve{\mathbf{a}}_1^{III} \left(\mathbf{a}_1^{III} \right)^T$$
$$+ (\mathbf{A}^{III})^T \bar{\mathbf{A}}^{III} \mathbf{A}^{III}$$

where $\mathbf{a}_k^j = [\mathbf{\Psi}_{(k-1)} \tilde{\mathbf{p}}_k / \delta_k]^j$, $\breve{\mathbf{a}}_k^{\,j} = [\mathbf{I}_{(2,k)} \mathbf{a}_2^T + \mathbf{I}_{(3,k)} \mathbf{a}_3^T]^j$, $\bar{\mathbf{A}}^j = [\mathbf{a}_2 \breve{\mathbf{a}}_2^{\,T} + \mathbf{a}_3 \breve{\mathbf{a}}_3^{\,T}]^j$ and $\mathbf{A}^j = (\mathbf{S}^j - \mathbf{1})$ for $k = 1, 2$ and 3 and $j = I, II$ and III. Further, matrix $\bar{\mathbf{B}}$ is block-diagonal, namely,

$$\bar{\mathbf{B}} = \text{diag}\left(\mathbf{B}_{P1}^I \mathbf{p}_1^I, \mathbf{B}_{P1}^{II} \mathbf{p}_1^{II}, \mathbf{B}_{P1}^{III} \mathbf{p}_1^{III} \right)$$
$$\text{diag}(\bar{\mathbf{b}}^I, \bar{\mathbf{b}}^{II}, \bar{\mathbf{b}}^{III}) \tag{60}$$

Substituting $\bar{\mathbf{B}}$ into Equation (58) we obtain

$$\begin{bmatrix} (\bar{\mathbf{b}}^I)^T \mathbf{L}_{11} \bar{\mathbf{b}}^I & & \text{sym} \\ (\bar{\mathbf{b}}^{II})^T \mathbf{L}_{21} \bar{\mathbf{b}}^I & (\bar{\mathbf{b}}^{II})^T \mathbf{L}_{22} \bar{\mathbf{b}}^{II} & \\ (\bar{\mathbf{b}}^{III})^T \mathbf{L}_{31} \bar{\mathbf{b}}^I & (\bar{\mathbf{b}}^{III})^T \mathbf{L}_{32} \bar{\mathbf{b}}^{II} & (\bar{\mathbf{b}}^{III})^T \mathbf{L}_{33} \bar{\mathbf{b}}^{III} \end{bmatrix} \ddot{\boldsymbol{\theta}}_A = \begin{bmatrix} \bar{\tau}_1 \\ \bar{\tau}_2 \\ \bar{\tau}_3 \end{bmatrix} = \bar{\boldsymbol{\tau}} \tag{61}$$

where $\mathbf{L}_{(k,l)} = \mathbf{L}_{(k,l)}^I + \mathbf{L}_{(k,l)}^{II} + \mathbf{L}_{(k,l)}^{III}$. Now we solve the above system for $\ddot{\boldsymbol{\theta}}_A$ using the reverse Gaussian elimination technique, as suggested by [38] and [40], to obtain

$$\ddot{\theta}_1^I = \frac{\hat{\bar{\tau}}_1}{\alpha_1} \tag{62}$$

$$\ddot{\theta}_1^{II} = \frac{\hat{\tau}_2 - (\mathbf{b}^{II})^T \hat{\mathbf{L}}_{21} \mathbf{b}^I \ddot{\theta}_1^T}{\alpha_2} \tag{63}$$

$$\ddot{\theta}_1^{III} = \frac{\bar{\tau}_3 - [(\mathbf{b}^{III})^T \mathbf{L}_{31} \mathbf{b}^I \ddot{\theta}_1^I + (\mathbf{b}^{III})^T \mathbf{L}_{32} \mathbf{b}^{II} \ddot{\theta}_1^{II}]}{\alpha_3} \tag{64}$$

where

$$\alpha_3 = (\mathbf{b}^{III})^T \mathbf{L}_{33} \mathbf{b}^{III}$$

$$\hat{\mathbf{L}}_{11} = \mathbf{L}_{11} - \frac{\mathbf{L}_{31}^T \mathbf{b}^{III}}{\alpha_3} (\mathbf{b}^{III})^T \mathbf{L}_{31}$$

$$\hat{\mathbf{L}}_{21} = \mathbf{L}_{21} - \frac{\mathbf{L}_{32}^T \mathbf{b}^{III}}{\alpha_3} (\mathbf{b}^{III})^T \mathbf{L}_{31}$$

$$\hat{\mathbf{L}}_{22} = \mathbf{L}_{22} - \frac{\mathbf{L}_{32}^T \mathbf{b}^{III}}{\alpha_3} (\mathbf{b}^{III})^T \mathbf{L}_{32}$$

$$\hat{\tau}_1 = \bar{\tau}_1 - (\mathbf{b}^I)^T \frac{\mathbf{L}_{31}^T \mathbf{b}^{III}}{\alpha_3} \bar{\tau}_3$$

$$\hat{\tau}_2 = \bar{\tau}_2 - (\mathbf{b}^{II})^T \frac{\mathbf{L}_{32}^T \mathbf{b}^{III}}{\alpha_3} \bar{\tau}_3$$

$$\alpha_2 = (\mathbf{b}^{II})^T \hat{\mathbf{L}}_{22} \mathbf{b}^{II}$$

$$\hat{\hat{\tau}}_1 = \hat{\tau}_1 - (\mathbf{b}^I)^T \frac{\hat{\mathbf{L}}_{21}^T \mathbf{b}^{II}}{\alpha_2} \hat{\tau}_2$$

$$\alpha_1 = (\mathbf{b}^I)^T \hat{\mathbf{L}}_{11} \mathbf{b}^I - (\mathbf{b}^I)^T \frac{\hat{\mathbf{L}}_{21}^T \mathbf{b}^{II}}{\alpha_2} (\mathbf{b}^{II})^T$$

thereby completing the analysis.

4.3.1. *Summary of Forward Dynamics*

The steps required for the distribution of the forward dynamics computations are summarized below

1. Conduct a displacement analysis and obtain all joint angles for the actuated joint angles $\boldsymbol{\theta}_A$ available.
2. Perform the inverse dynamics for $\ddot{\boldsymbol{\theta}}_A = \mathbf{0}$ of the platform, as discussed above, to obtain $\bar{\boldsymbol{\tau}}$ for *each open chain*.
3. Calculate $\mathbf{L}^j = [\mathbf{T}^T \mathbf{I} \mathbf{T}]^j$ for $j = I,\ II$ and III.
4. Calculate the actuated joint accelerations $\ddot{\boldsymbol{\theta}}_A$.
5. Integrate the actuated joint accelerations, to obtain the actuated joint rates and velocities.

However, conducting a displacement analysis at each iteration is not advisable. The alternative, as mentioned previously, is to integrate all joint rates, which is done as described below

1. Conduct a velocity analysis and obtain the unactuated joint rates for the actuated joint rates $\dot{\boldsymbol{\theta}}_A$ available.

2. Perform the inverse dynamics for $\ddot{\boldsymbol{\theta}}_A = \mathbf{0}$ of the platform, as discussed above to obtain $\bar{\tau}$ for *each open chain*.
3. Calculate $\mathbf{L}^j = [\mathbf{T}^T \mathbf{IT}]^j$, for $j = I,\ II$ and III.
4. Calculate the actuated joint accelerations $\ddot{\boldsymbol{\theta}}_A$.
5. Integrate the actuated joint accelerations and *all* joint rates in corresponding chain nodes, to obtain the actuated joint rates and *all* joint angles.

5. A Numerical Example

5.1. PARAMETERS AND INITIAL CONDITIONS

We use the same parameters for the 3 RRR planar parallel manipulator shown in Figure 3 as in [28]. The end effector, labeled 7, has the shape of an equilateral triangle, with sides of length l_7, links 1,2 and 3 have a length l_1, links 4, 5 and 6 have length l_4, and the three fixed revolute joints form an equilateral triangle with sides of length l_0. The prescribed motion drivers given by [28] are

$$\theta_1(t) = \frac{1}{3}\pi + \frac{1}{6}\left(\frac{2\pi t}{T} - \sin\frac{2\pi t}{T}\right)$$

$$\theta_2(t) = \frac{4}{3}\pi - \frac{1}{6}\left(\frac{2\pi t}{T} - \sin\frac{2\pi t}{T}\right)$$

$$\theta_3(t) = \frac{11}{6}\pi + \frac{1}{12}\left(\frac{2\pi t}{T} - \sin\frac{2\pi t}{T}\right)$$

where $T = 3$ s. However, since the initial values of the foregoing joint angles are not sufficient to define a unique initial posture of the manipulator, we use the initial
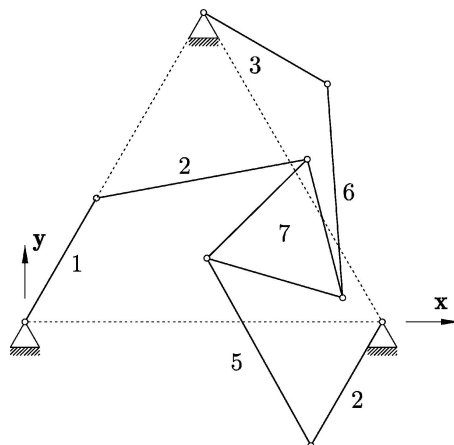

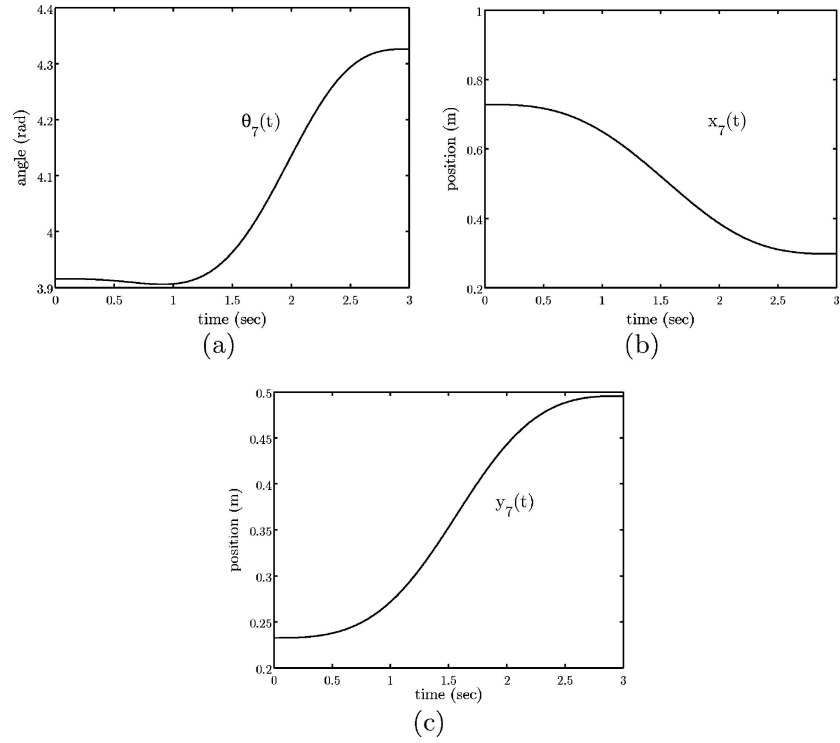*Figure 3.* The three-dof planar parallel manipulator used in the example in its initial posture.

Figure 4. End effector trajectory: (a) Orientation; (b) *x*-position; and (c) *y*-position.

configuration given by [17]:

$$
\begin{array}{lll}
\theta_1(0) = \dfrac{1}{3}\pi & \theta_4(0) = -0.865 \text{ rad} & x_7(0) = 0.728 \text{ m} \\[2mm]
\theta_2(0) = \dfrac{4}{3}\pi & \theta_5(0) = -2.102 \text{ rad} & y_7(0) = 0.233 \text{ m} \\[2mm]
\theta_3(0) = \dfrac{11}{6}\pi & \theta_6(0) = -0.976 \text{ rad} & \theta_7(0) = 3.916 \text{ rad}
\end{array}
$$

The end effector trajectory for the prescribed motion drivers and the chosen initial configuration was first computed using the method outlined in [28]. Figure 4 shows the resulting end effector trajectory. The parameters of the manipulator are given in Table I, with gravity in the $-Y$ direction.

Table I. Dimensions and inertia properties of the example manipulator

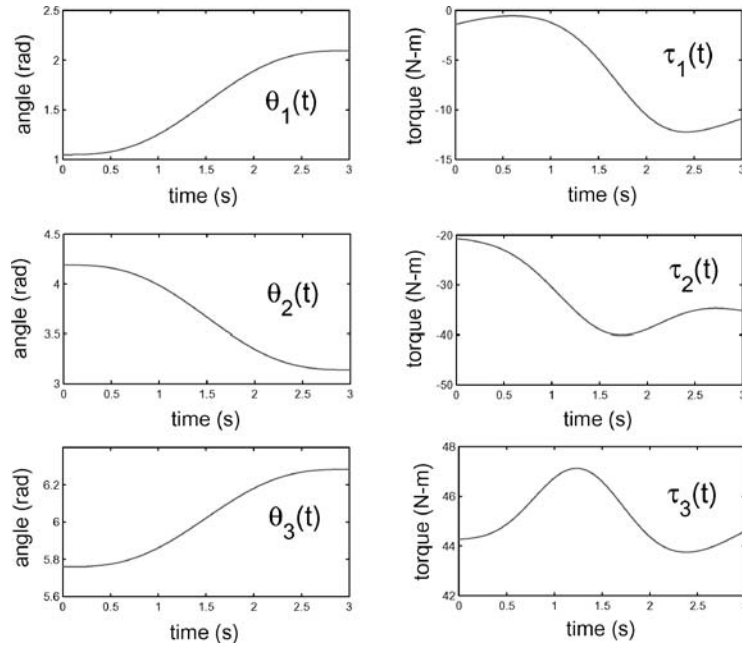| Link i	L_i (m)	m_i (kg)	I_i (kg m^2)
1, 2, 3	0.4	3.0	0.04
4, 5, 6	0.6	4.0	0.12
7	0.4	8.0	0.0817

Figure 5. Desired trajectory and required driving torques.

5.2. RESULTS

We first perform the inverse dynamics in order to compute a time history of actuation forces that would realize the prescribed motions. Using the above parameters, the resulting torques for the actuated joints 1, 2 and 3 were evaluated using the inverse dynamics model discussed in this paper. The resulting set of torques which realize the prescribed motions is shown in Figure 5, which tally with those given by [28].

The motion of the manipulator, using these driving torques, was simulated and the results are reported below. Briefly, we note that MATLAB^{TM} ODE suite offers two groups of integration schemes: (a) fixed time-stepping schemes, where the user can specify the size of the time step; and (b) adaptive time-stepping, where an estimate of the integration error is made, and the time step is adapted to keep this error below a specific tolerance level. In the latter case, the error in each state is estimated to ensure the $\mathbf{e}(i) \leq \max(\text{RelTol} * \text{abs}(\mathbf{x}(i)), \text{AbsTol}(i))$ condition for the ith component of the state vector $\mathbf{x}$, where $\mathbf{e}$ is the error vector. The two quantities under user-control here, in addition to the actual selection of the algorithm, are the value for the relative tolerance (RelTol) and the absolute tolerance (AbsTol). Below, we report on the results from the forward dynamics simulations with *both* fixed time-stepping and adaptive time stepping algorithms.

Adaptive-Time-Stepping Case

In this case, the relative tolerance was pre-specified and an adaptive time-stepping scheme was used for the simulation. The two primary metrics of performance evaluation for this case were: (a) extent of the constraint error and (b) number of iterations. Four different relative tolerances, varying in orders of magnitude from 10^{-3} to 10^{-6}, were examined in this case. The ode45 (Dormand-Prince) integration scheme from MATLAB^{TM}'s ODE suite was used for the for adaptive time-stepping simulation. Testing the models with adaptive time-stepping methods can give insight into the overall characteristics of a formulation, including: (a) formulation stiffness and computational complexity of implementation, as measured by the number of iterations or the total time taken to simulate a fixed simulation time.

Fixed Time-Stepping Case

In this case, the principal parameter that can be selected by the user, in addition to the actual algorithm from the ODE suite, is the step size of fixed time step. This selection has critical implications in that an order of magnitude reduction in step-size increases the number of iterations by the same order of magnitude. The principal metric for evaluating the effectiveness of reducing the time step, and thereby slowing the computation by increasing the number of iterations, will be the actuated joint-angle errors between the prescribed and the simulated joint-trajectories.

The results of the DeNOC model now follow: Figure 6 shows the resulting error between desired and simulated motion trajectories for adaptive time stepping with relative tolerances of 10^{-3} and 10^{-6}, respectively. A relatively small number of
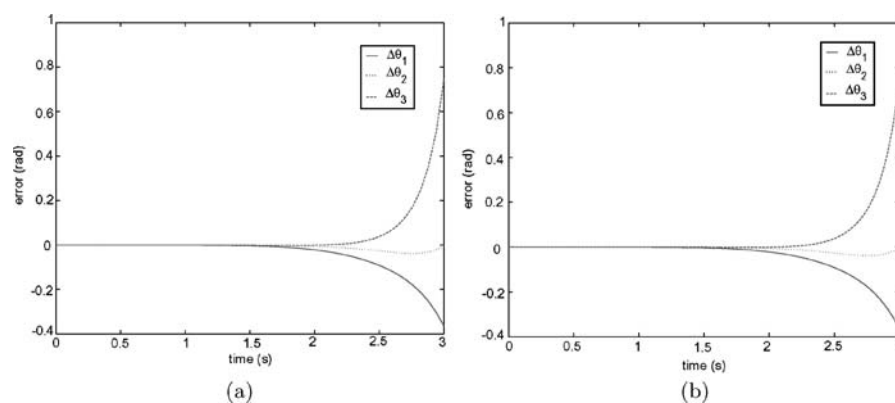

Figure 6. Joint-angle errors between desired and actual joint trajectories for the decoupled natural orthogonal complement with adaptive time stepping for: (a) relative tolerance of 10^{-3}; and (b) relative tolerance of 10^{-6}.
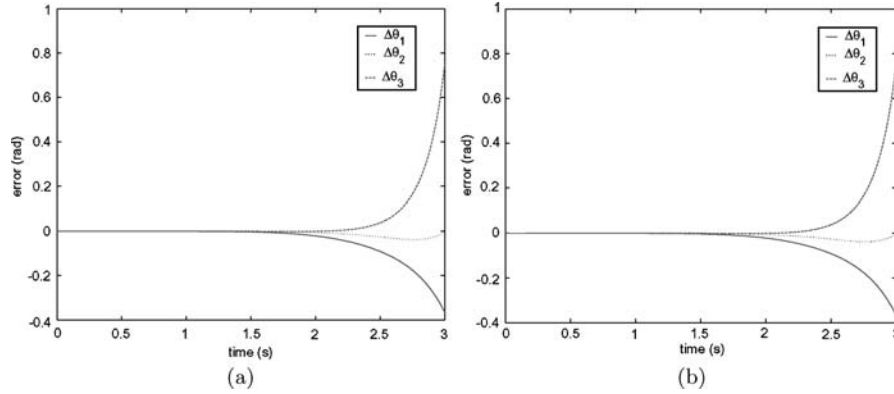
Figure 7. Simulation results for the DeNOC with fixed time-step. Joint-angle errors between desired and actual joint trajectories for time steps of: (a) 0.01 s; and (b) 0.001 s.

time-steps was required, namely, 301, when simulated with a relative tolerance of 10^{-3}, and 319 when simulated with a tolerance of 10^{-6}.

Figure 7a shows the errors between the desired trajectory and the trajectory obtained from simulation, for a fixed time step of 0.01s. The ode5(Dormand-Prince) integration scheme in MATLAB^{TM}was used for the for fixed time stepping simulation. Figure 7b shows the same for fixed time stepping of 0.001s. The deviation is due to the corresponding zero eigenvalue instability, which is overcome using feedback control.

In order to have a better picture of the accuracy of the model, we created a feedback compensation control scheme to force the simulated actuated joint trajectories to match their prescribed counterparts. Figure 8a shows the error between the
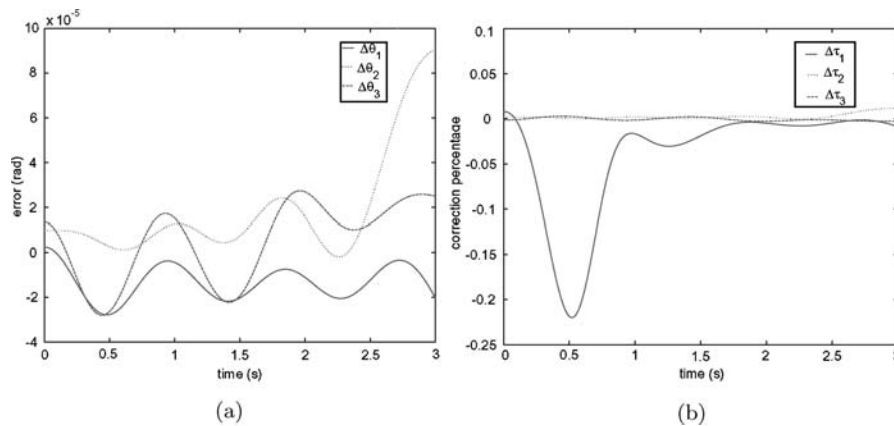

Figure 8. Simulation results for the DeNOC with feedback control for a fixed time step of 0.001s: (a) joint angle errors between the desired and actual joint trajectories; and (b) driving-torque correction time history.

desired trajectory and the trajectory obtained from simulation with feedback control. Figure 8b shows the percentage of torque correction required to obtain the desired trajectory where we would like to highlight the extremely small correction required. Based on the above results we can conclude that:

1. The number of iterations required for adaptive time stepping for relative tolerances of 10^{-3} and 10^{-6} is low.
2. The accuracy of this method, for our experiment, is quite high, of around 99%.

6. Conclusion

A dynamic modeling methodology for parallel multibody mechanical systems – based on the *Newton-Euler* EOM and the *decoupled natural orthogonal complement* (DeNOC) matrices associated with the loop-closure kinematic constraints – was presented for parallel architecture manipulators.

The modeling methodology takes advantage of the spatial parallelism of the supporting legs inherent in such parallel manipulators systems to develop a modular composition of the overall system dynamics in the extended configuration space with non-minimal generalized coordinates. The resulting equations are projected onto the space of feasible motions described in terms of a minimal, actuated set of joint-configuration coordinates using the decoupled natural orthogonal complement matrices of the loop-closure kinematic constraints. Traditional approaches to deriving closed-loop dynamics equations tend to sacrifice recursivity at this stage – however, the recursive computation of DeNOC matrices facilitates a recursive formulation of both inverse and forward dynamics algorithms in this paper. Further, inverse and forward dynamics of both exactly actuated and redundant manipulators can be easily incorporated within this framework.

These concepts were applied to an example of a 3 R̲RR planar parallel manipulator and the results showed good numerical behavior both in terms of the accuracy of the results and perhaps more importantly the absence of formulation stiffness.

Appendix A: Hybrid Twist Representation

Consider a body in motion with an instantaneous screw axis $\mathcal{L}$. In any given reference frame $\mathcal{F}_i$, the associated unit screw may be defined in terms of the axis unit vector $\mathbf{u}$, the position vector $\mathbf{p}$ of a point P on $\mathcal{L}$ and the pitch h as

$$\$ = \begin{bmatrix} \mathbf{u} \\ (\mathbf{p} \times \mathbf{u}) + h\mathbf{u} \end{bmatrix} \tag{65}$$

For a revolute joint R the underlying unit screw $\$_R$ has zero pitch, and may be written as

$$\$_R = \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \times \mathbf{u} \end{bmatrix} \tag{66}$$

Thus the spatial twist $\mathbf{t}^S$ [21] of the body $\mathcal{B}$ associated with rotation about this unit screw axis $\$_R$ can be written as

$$\mathbf{t}^S = \omega \$_R = \begin{bmatrix} \boldsymbol{\omega}^S \\ \mathbf{v}_O^S \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}^S \\ -\boldsymbol{\omega}^S \times \mathbf{p} \end{bmatrix} \tag{67}$$

where $\boldsymbol{\omega}^S$ is the angular velocity of $\mathcal{B}$ expressed in the inertial frame $\mathcal{F}_i$; and $\mathbf{v}_O^S$ is the velocity of a (potentially imaginary) point on $\mathcal{B}$ that is instantaneously coincident with origin O of $\mathcal{F}_i$. Given the spatial twist $\mathbf{t}^S$, the velocity $\dot{\mathbf{c}}$ of a given point C on the body may be written as

$$\dot{\mathbf{c}} = \mathbf{v}_C^S = \mathbf{v}_O^S + \boldsymbol{\omega}^S \times \mathbf{c} \tag{68}$$

leading to

$$\mathbf{t}_C^S = \begin{bmatrix} \boldsymbol{\omega}^S \\ \dot{\mathbf{c}} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \mathbf{O} \\ -[\mathbf{c}\times] & \mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}^S \\ \mathbf{v}_O^S \end{bmatrix}$$

where $\mathbf{t}_C^S$ is the "hybrid representation" [32] of the twist of body $\mathcal{B}$ in a newly selected inertial frame $\mathcal{F}_i^C$ whose origin is instantaneously coincident with the point C. In this light it is also evident that the so-called "twist transfer formula", introduced in Equation (9), is nothing other than the adjoint transformation between two inertial frames. In this paper, we will preferentially employ this hybrid representation of the twist.
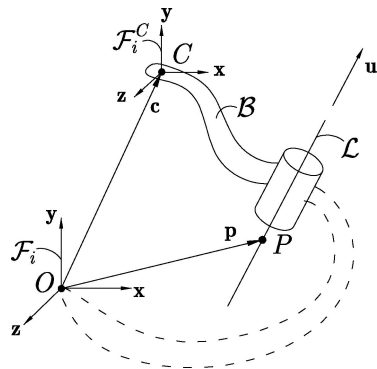

Figure 9. A body undergoing a twist motion about an axis.

Appendix B: Twist Annihilators

Both twists and wrenches may be denoted in terms of a unit screw scaled by a magnitude. Thus, while both admit representation as 6-tuples, neither is a member of a bi-invariant metric vector space. Therefore the notion of orthogonality of two twists (or correspondingly two wrenches) is not well-defined since it is not possible to define an inner product. However, a *reciprocity relationship* may be defined by considering twists to be members of a vector space and wrenches to be members of the corresponding *dual vector space*. This reciprocity principle, also termed "natural pairing" is no more than a restatement of the Principle of Virtual Work in the mechanics context.

Thus, noting the existence of such a reciprocal wrench for any given twist, the twist annihilator [1], may be defined as a 6×6 singular mapping of a twist into a six dimensional zero vector. The annihilator $\mathbf{\Phi}$ of a unit twist $\$_R = [\mathbf{u}, (\mathbf{p} - \mathbf{c}) \times \mathbf{u}]^T$ may be explicitly constructed geometrically as

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{1} - \mathbf{u}\mathbf{u}^T & \mathbf{O} \\ -\hat{\mathbf{r}} & \mathbf{1} \end{bmatrix} \tag{69}$$

where $\hat{\mathbf{r}}$ is the cross product matrix of $\mathbf{p} - \mathbf{c}$ such that $\mathbf{\Phi}\$_R = \mathbf{0}$. Twist annihilators for other types of joints are presented in detail in [1].

As we shall see in the paper, this annihilator matrix plays a critical role in eliminating the unactuated joint twists from the kinematics equations without destroying the recursive nature of the relationships. It is noteworthy that this definition is possible solely in geometric terms i.e. in terms of the underlying unit screw of the twist. We also note that such a formal construction of $\mathbf{\Phi}$ does not lead to any dimensional inconsistencies.

However, we would also like to note a useful relation which will be exploited in analysis in the paper. Given a recursive velocity expression relating twists between two links of the form $\mathbf{t}_B = \mathbf{t}_A + \$_R\dot{\theta}$ where $\mathbf{t}_A$ and $\mathbf{t}_B$ are n-dimensional twists, $\$_R$ is a unit twist and $\dot{\theta}$ is a scalar, we may obtain explicit expressions for $\mathbf{\Phi}$ and $\dot{\theta}$ as

$$\begin{aligned} \mathbf{\Phi} &= \left[\mathbf{1} - \frac{\$_R\$_R^T}{\delta} \right] \\ \dot{\theta} &= \frac{\$_R^T}{\delta}(\mathbf{t}_B - \mathbf{t}_A) \end{aligned} \tag{70}$$

where $\delta = \$_R{}^T\$_R$ and $\mathbf{1}$ is the 6×6 identity matrix. These expressions offer a convenient method to compute expressions for the annihilator eliminating the need for decomposing the unit twist $\$_R$ into its geometric components and then constructing the annihilator as shown in Equation (69).

References

1. Angeles, J., *Computer-Aided Analysis of Rigid and Flexible Mechanical Systems*, Chapt. on Twist and Wrench Generators and Annihilators. Dordrecht-Boston-London: Kluwer Academic Publishers, 1994.
2. Angeles, J., *Fundamentals of Robotic Mechanical Systems*. New York: Springer-Verlag, 2002.
3. Angeles, J. and Lee, S., 'The formulation of dynamical equations of holonomic mechanical systems using a natural orthogonal complement', *ASME Journal of Applied Mechanics* **55**, 1988, 243–244.
4. Angeles, J. and Ma, O., 'Dynamic simulation of n-axis serial robotic manipulators using a natural orthogonal complement', *The International Journal of Robotics Research* **7**(5), 1988, 32–47.
5. Armstrong, W., 'Recursive solution to the equations of motions of an n-link manipulator', in *Proc. 5th World Congress on Theory of Machines and Mechanisms*, Montreal, 1979, pp. 1343–1346.
6. Ascher, U., Pai, D. and Cloutier, B., 'Forward dynamics, elimination methods, and formulation stiffness in robot simulation', *The International Journal of Robotics Research* **16**(6), 1997, 749–758.
7. Ascher, U. and Petzold, L., *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Philadelphia: SIAM, 1998.
8. Bae, D. and Han, J., 'A generalized recursive formulation for constrained mechanical system dynamics', *Mechanics of Structures and Machines, An International Journal* **27**(3), 1999, 293–315.
9. Bae, D. and Haug, E., 'A recursive formulation for constrained mechanical system dynamics: Part 2. closed loop systems', *Mechanics of Structures and Machines, An International Journal* **15**(4), 1987, 481–506.
10. Balafoutis, C., Patel, R. and Cloutier, B., 'Efficient modelling and computation of manipulator dynamics using orthogonal cartesian tensors', *IEEE Journal of Robotics and Automation* **4**, 1988, 665–676.
11. Blajer, W., 'A geometrical interpretation and uniform matrix formulation of multibody system dynamics', *Zeitschrift fr Angewandte Mathematik und Mechanik* **81**(4), 2001, 247–259.
12. Brandl, H., Johanni, R. and Otter, M., 'A very efficient algorithm for the simulation of robots and similar multibody systems without inversion of the mass matrix', In *Proc. IFAC/IFIP/IMACS International Symposium on Theory of Robots*, Vienna, 1986.
13. Featherstone, R., 'The calculation of robot dynamics using articulated-body inertias', *The International Journal of Robotics Research* **2**(1), 1983, 13–30.
14. Featherstone, R., *Robot Dynamics Algorithms*. Boston-Dordrecht-Lancaster: Kluwer Academic Publishers, 1987.
15. Featherstone, R., 'A divide-and-conquer articulated-body algorithm for parallel $O(\log(n))$ calculation of rigid-body dynamics. Part 2: Trees, loops and accuracy', *The International Journal of Robotics Research* **18**(9), 1999, 876–892.
16. García de Jalón, J. and Bayo, E., *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge*, New York: Springer-Verlag, 1994.
17. Geike, T. and McPhee, J., 'On the automatic generation of inverse dynamic solutions for parallel manipulators', in *Proc. Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*. Quebec City, 2002, pp. 348–358.
18. Goldenberg, A. and He, X., 'An algorithm for efficient computation of dynamics of robotic manipulators'. in *Proc. Fourth International Conference on Advanced Robotics*, Columbus, OH, 1989, pp. 175–188.

19. Gosselin, C. and Angeles, J., 'Singularity analysis of closed-loop kinematic chains', *IEEE Transactions on Robotics and Automation* **6**(3), 1990, 281–290.
20. Haug, E., *Computer Aided Kinematics and Dynamics of Mechanical Systems*. Boston: Allyn and Bacon, 1989.
21. Hunt, K. H., *Kinematic Geometry of Mechanisms*. Oxford Science Publications, 1990.
22. Kecskemethy, A., Krupp, T. and Hiller, M., 'Symbolic processing of multi-loop mechanism dynamics using closed form kinematic solutions', *Multibody System Dynamics* **1**(1), 1997, 23–45.
23. Kerr, J. and Roth, B., 'Analysis of multifingered hands', *The International Journal of Robotics Research* **4**(4), 1986, 3–17.
24. Khan, W. A., 'Distributed dynamics of systems with closed kinematic chains', Master's thesis, Mechanical Engineering, McGill University, Montreal, 2002.
25. Koivo, A. J. and Bekey, G. A., 'Report of workshop on coordinated multiple robot manipulators: planning, control, and application', *IEEE Transactions on Robotics and Automation* **4**, 1988, 91–93.
26. Kumar, V. and Waldron, K., 'Force distribution in closed kinematic chains', *IEEE Transactions on Robotics and Automation* **4**(6), 1988, 657–664.
27. Luh, J., Walker, M. and Paul, R., 'On-line computational schemes for mechanical manipulators', *ASME Journal of Dynamic Systems, Measurement and Control* **102**(2), 1980, 69–76.
28. Ma, O. and Angeles, J., 'Direct kinematics and dynamics of a planar 3-dof parallel manipulator', in *Advances in Design Automation*, Vol. 3. Montreal, Quebec, 1989, pp. 313–320.
29. McMillan, S. and Orin, D., 'Efficient computation of articulated-body inertias using successive axial screws', *IEEE Transactions on Robotics and Automation* **11**, 1995, 606–611.
30. McMillan, S., Sadayappan, P. and Orin, D. E., 'Parallel dynamic simulation of multiple manipulator systems: temporal versus spatial methods', *IEEE Transactions on Systems, Man and Cybernetics* **24**(7), 1994, 982–990.
31. Merlet, J.-P., *Parallel Robots*. Dordrecht: Kluwer Academic Publishers, 2000.
32. Murray, R., Li, Z. and Sastry, S., *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, 1994.
33. Orin, D., McGhee, R., Vukobratovic, M. and Hartoch, G., 'Kinematic and kinetic analysis of open-chain linkages utilizing newton-euler methods', *Mathematical Biosciences* **43**, 1979, 107–130.
34. Orin, D. and Walker, M., 'Efficient dynamic computer simulation of robotic mechanisms', *ASME Journal of Dynamic Systems, Measurement and Control* **104**, 1982, 205–211.
35. Rodriguez, G. and Kreutz-Delgado, K., 'Spatial operator factorization and inversion of the manipulator mass matrix', *IEEE Transactions on Robotics and Automation* **8**(1), 1992, 65–76.
36. Saha, S. K., 'A decomposition of the manipulator inertia matrix', *IEEE Transactions on Robotics and Automation* **13**(2), 1997, 301–304.
37. Saha, S. K., 'Analytical expression for the inverted inertia matrix of serial robots', *The International Journal of Robotic Research* **18**(1), 1999, 20–36.
38. Saha, S. K., 'Dynamics of serial multibody systems using the decoupled natural orthogonal complement matrices', *ASME Journal of Applied Mechanics* **66**, 1999, 986–996.
39. Saha, S. K. and Angeles, J., 'Dynamics of nonholonomic mechanical systems using a natural orthogonal complement', *ASME Journal of Applied Mechanics* **58**, 1991, 238–243.
40. Saha, S. K. and Schiehlen, W. O., 'Recursive kinematics and dynamics for parallel structured closed-loop multibody systems', *Mechanics of Structures and Machines, An International Journal* **29**(2), 2001, 143–175.
41. Salisbury, J. K. and Craig, J. J., 'Articulated hands: Force control and kinematic issues', *The International Journal of Robotics Research* **1**(1), 1982, 4–17.

42. Schiehlen, W., 'Multibody systems and robot dynamics', in A. Morecki, G. Bianchi and K. Jaworek (eds.), *Proc. 8thCISM-IFToMM Symposium on Theory and Practice of Robot Manipulators*. Warsaw, Poland, 1990, pp. 14–21.
43. Schiehlen, W., *Multibody Systems Handbook*. Berlin: Springer-Verlag, 1990.
44. Shabana, A. A., *Computational Dynamics*. New York: Wiley, 2001.
45. Song, S. M. and Waldron, K. J., *Machines that Walk*. 2 edn., Cambridge, MA: MIT Press, 1989.
46. Stejskal, V. and Valasek, M., *Kinematics and Dynamics of Machinery*. New York: Marcel Dekker, 1996.
47. Stepanenko, Y. and Vukobratovic, M., 'Dynamics of articulated open-chain active mechanism', *Mathematical Biosciences* **28**, 1976, 137–170.
48. Vereshchagin, A., 'Computer simulation of the dynamics of complicated mechanisms of robot manipulators', *Engineering Cybernetics* **6**, 1974, 65–70.
49. Walker, M. and Orin, D., 'Efficient dynamic computer simulation of robotic mechanisms', *ASME Journal of Dynamic Systems, Measurement and Control* **104**, 1982, 205–211.
50. Wang, J., Gosselin, C. and Cheng, L., 'Dynamic modelling and simulation of parallel mechanisms using virtual spring approach', in *Proc. 2000 ASME Design Engineering Technical Conferences*. Baltimore, Maryland, 2000, pp. 1–10.
51. Yiu, Y., Cheng, H., Xiong, Z., Liu, G. and Li, Z., 'On the dynamics of parallel manipulator', in *Proc. IEEE international Conference on Robotics and Automation*. Seoul, Korea, 2001, pp. 3766–3771.