Addis Ababa University
School of Graduate Studies

COMPUTER-AIDED DYNAMIC FORCE ANALYSIS OF FOUR-BAR PLANAR MECHANISMS

By

Robel Metiku

Advisor

Dr. Alem Bazezew

November 2004

Addis Ababa University

Faculty of Technology

School of Graduate Studies

COMPUTER-AIDED DYNAMIC FORCE ANALYSIS OF FOUR-BAR PLANAR MECHANISMS

A thesis submitted to the School of Graduate Studies of Addis Ababa University in partial fulfillment of the Degree of Masters of Science in Mechanical Engineering (Applied Mechanics stream)

By

Robel Metiku

Advisor
Dr. Alem Bazezew

Addis Ababa
November 2004

Acknowledgement

First, I would like to express my heartfelt appreciation and gratitude to my thesis advisor Dr. Alem Bazezew for his continual encouragement and patient guidance throughout the course of this work. His deep insight into the subject of Mechanics and its many applications, his infectious enthusiasm, his patience in helping me learn the basics, and his inexhaustible supply of interesting ideas and materials, have all greatly enriched my experience as a graduate student.

Many thanks are also extended to all my colleagues and friends especially Ato Yoseph Alemu and Ato Chanyalew Taye in editing the finial thesis. Their helpful comments during the research and writing of this thesis are appreciated.

At last, but not least, I would like to take this opportunity to thank my father and my mother for all their help and understanding. Without my family's encouragement, love and support, this work would never have been accomplished.

Abstract

The four-bar linkage is the most basic chain of pin-connected links that allows relative motion between links. Although a simple mechanism, the four-bar linkage mechanism is very versatile and used in thousands of applications.

The main themes of this thesis are the modeling, computer-aided dynamic force analysis and simulation of four-bar planar mechanisms composed of rigid bodies and massless force and torque producing elements. Motions of the rigid bodies are predicted by numerically integrating Differential-Algebraic Equations (DAEs) developed from principles of mechanics by using the Newton-Euler's approach. The computer program used for solving the equations developed in the analysis problem and that integrates the differential equations is Matlab.

Chapter 1

Literature Survey

During the last quarter century, rigid body dynamics has received considerable attention due to the central role it plays in robot simulation, control, design, and computer animation. A great number and variety of formalisms have been developed for rigid body systems despite the fact that all of them can be derived from a few fundamental principles of mechanics. What is commonly known as the Newton-Euler method includes the constraint forces acting on all bodies of the system, which results in redundant equations with more equations than unknowns.

In other formulations, such as Lagrange's and Kane's method [17][†], the constraint forces are eliminated by using d'Alembert's principle. Efficient simulation algorithms were developed based on these formulations for systems with different structures. Popular algorithms include Articulated Body Method (ABM) developed for typical robot arms with serial chain structure, Composite Rigid Body Methods (CRBM) which can explicitly formulate the joint space inertia matrix for systems with tree structures, Dynamic Analyzer DYANA, ADAMS, DADS and IMP [16]. While there has been a significant amount of research in rigid body dynamics, the need for dynamic analysis of mechanisms has been known for some time but little work has been done in this area.

[†] Numbers in square brackets pertain to references at the end of the thesis.

A.T. Yang [45] in his paper, “Static Force and Torque Analysis of Spherical Four-Bar Mechanisms”, gave a complete static force and torque analysis of spherical four-bar mechanisms of general proportions by using the equations of static equilibrium. The results from the analysis are expressed in concise algebraic forms ready to programmed digital computation. In addition, the numerical results could be applied directly by the designer in determining the dimensions of the links since the forces and moments acting on each member were all expressed in terms of components along the base vectors of the reference frame fixed to the corresponding link.

J. J. Uicker [46] presented a paper “Dynamic Force Analysis of Spatial Linkages”, that used the matrix method of linkage analysis to analyze the bearing forces and torques which result from the inertia of the moving links when a single-loop, single degree of freedom linkage is driven with known input velocity and acceleration. The method is well suited to digital computation and has been tested on several examples of spatial linkages.

M. A. Chace [14] again wrote a paper “Analysis of the Time-Dependence of Multi-Freedom Mechanical Systems in Relative Coordinates”, in which a theoretical model and computational procedure are proposed for the analysis of the time dependence of highly constrained, multi-degree-of-freedom mechanical systems. Problems are formulated as a simultaneous set of Lagrange differential equations in the system relative coordinates, accompanied by second-order constraints. The equations are linear in the second-order terms and are solved by a numerical integration procedure.

Paul, B. et. al. [48] in their paper "Computer Analysis of Machines with planar Motion I-Kinematics; II-Dynamics", presented a uniform procedure for establishing the dynamic equation of motion for machines with single-degree of freedom or multi-degree of freedom. The procedure, which utilizes the independent kinematic loops of machines, is readily programmed for a digital computer and it is capable of treating input forces, internal springs and dampers.

Bagci C. [47] wrote his paper "Static Force and Torque Analysis Using 3x3 Screw Matrix, and Transmission Criteria for Space Mechanisms", where he presented the method of determining the static force and torque distributions in space mechanisms by the use of the 3x3 screw matrix. The dual equilibrium equations for an n-link multi-loop mechanism were written and the explicit expressions for the force and torque components at any pair location derived.

D.A. Smith [49], in his work "Reaction Force Analysis in Generalized Machine Systems", developed a method which reduces the calculation of reaction forces for multi-degree-of-freedom, constrained, mechanical, dynamic systems to a process of accumulating a sum of terms representing inertial forces, external applied forces, and Lagrange multiplier forces. The method resulted in an approach to reaction force calculations which are computationally more efficient than either virtual work or equilibrium when the methods are applied in conventional ways.

R. R. Allen et. al. [50] wrote "Connection Force Analysis of Mechanisms Described by Explicit Equations of Motion in Generalized Coordinates" where the connection forces

acting at the joints of a kinematic mechanism are derived from the generalized variables of the mechanism equation of motion using an equivalent force analysis. The connection force calculation was numerically efficient and independent of the formulation and solution of the equations of motion as illustrated by a numerical example applied to a planar four-bar linkage.

V.K. Gupta [51] in his paper "Dynamic Analysis of Multi-Rigid-Body Systems" presented a method for formulating and solving the Newton-Euler equations of motion of a system of interconnected rigid bodies. The reaction forces and torques resulting from rigid body constraints imposed at the connecting joints were determined. The derivation of kinematic expressions for first and higher derivatives is demonstrated based on direct differentiation of a rotation matrix in the sprit of the classical vector approach.

R. A. Wehage et. al. [8] in their paper, "Generalized coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems", presented a computer based method for formulation and efficient solution of nonlinear, constrained differential equations of motion for mechanical systems. Nonlinear holonomic constraint equations and differential equations of motion are written in terms of a maximal set of Cartesian generalized coordinates, to facilitate the general formulation of constraints and forcing functions. They used Gaussian elimination algorithm with full pivoting to decompose the constraint Jacobian matrix, identify dependent variables, and construct an influence coefficient matrix relating variations in dependent and independent variables and employed this information to numerically construct a reduced system of differential

equations of motion whose solution yields the total system dynamic response.

S. S. Kim et. al. [9] in their paper, "A General and Efficient Method for Dynamic Analysis of Mechanical Systems Using Velocity Transformations", presented a new formulation for the equation of motion of interconnected rigid bodies by initially using Cartesian coordinates to define the position of the system, the kinematic joints between the bodies, and forcing functions on and between bodies. The equations of motion are then derived in terms of relative joint coordinates through the use of velocity transformation matrix which relates relative coordinates to Cartesian coordinates.

Radu Serban et. al. [18] in their paper " A Topology Based Approach for Exploiting Sparsity in Multibody Dynamics in Cartesian Formulation", presented a new method to efficiently compute accelerations and Lagrange multipliers in the equations of multibody dynamics in which simple manipulations bring the original problem of solving a system of (n + m) equations in (n + m) unknowns, to an equivalent problem in which a positive definite system of dimension (m x m) has to be solved for the Lagrange multipliers and then accelerations are efficiently determined.

Joel A. P. Dunlop [52] wrote his paper, "Automated Dynamic Analysis and Simulation of Planar Mechanical Systems, with Applications to Flexible Belt Drive Gantry Robots", in which he developed a dynamic analysis program capable of simulating the behavior of gantry robots actuated by flexible drive-belts.

Erik Spencer [53] in his paper, "Inverse Dynamics of Non-redundant Closed-Chain Mechanisms", presented two methods for calculating the dynamics of closed-chain mechanisms derived from the field of robotics. The two methods reduce the dimension of the equations of motion such that it corresponds with the degree of freedom of the mechanism and they are ideal for calculating the dynamics in symbolic form with the generalized coordinates chosen to be those of the actuated.

Brian Tavis Rundgren [54] wrote a paper, "Optimized Synthesis of Dynamically Based Force Generating Planar Four-bar Mechanism", and showed a technique for designing planar four-bar linkages by coupling optimization, dynamics and kinematics. The design approach presented in this paper calculates the resistance force by using both the static and anticipated dynamic effects of the resistance loading.

Waseem A. Khan [55] wrote "Distributed Dynamics of Systems with Closed Kinematic Chains", in which he examined the formulation of modular and distributed models and evaluated their performance as applied to mechanical systems with closed kinematic chains.

Mekonnen Gebreselassie [4] in his paper "Computer-Aided Synthesis, Kinematic Analysis and Simulation of Planar Mechanisms", presented the kinematic synthesis and analysis of four, five, and six-bar mechanisms by using the complex number approach.

J. García de Jalón et. al. [7] in their paper, "Improved Dynamic Formulations for the Dynamic Simulation of Multibody Systems", reviewed some ways to improve the efficiency of multibody dynamic formulations by considering two global formulations

that speed up the dynamic simulations using dependent Cartesian coordinates; one of them enforces the constraint equations by Lagrange multipliers and the other one by the penalty method. The ideas behind these improvements of global formulations are used to improve the topological formulations when they are applied to closed-loop multibody systems.

A research was done by Erdman A.G. on the historical perspective of computer-aided design of mechanisms. A summery of events, decade by decade, is given in [1].

Chapter 2

Introduction

2.1 Introduction to Planar Mechanisms

A mechanism is a mechanical device that has the purpose of transferring motion and/or force from a source to an output. A linkage consists of links (or bars) generally considered rigid, which are connected by joints, such as pins (or revolute), or prismatic joints, to form open or closed chains (or loops). Such kinematic chains, with at least one link fixed, become mechanisms if at least two other links retain mobility, or structures if no mobility remains. In other words, a mechanism permits relative motion between it rigid links; a structure does not.

Definitions

Joint (kinematic pair) is defined as the connection between two or more links at nodes which allows motion to occur between the links.

A link is a rigid body that possesses at least two nodes, which are the attachment points to other links.

A linkage is a system of links connected at joints with rotary or linear bearings.

A coupler is a link that undergoes complex motion and is not connected to ground.

A crank is a link that makes a complete revolution and is connected to the ground by a pivot and

A follower is a link that follows the motion of the crank.

Joints and Constraints: Kinematic relationships between bodies are defined by joints. In

the context of how a multibody system is described, a joint defines a set of holonomic[1] constraints that limit the geometric relationships that are possible between the bodies. Forces, torques, and speeds are not factors in a holonomic constraint. In addition to the holonomic constraints applied by joints, a system may also be subject to non-holonomic constraints. These are constraints on motion but not position or orientation.

Mechanisms are used in a great variety machines and devices. The simplest closed-loop linkage is the four-bar linkage, which has three moving links, one fixed link (a linkage with one link fixed is a mechanism) and four revolute, pivoted or pin joints (see fig. 2.1). The link that is connected to the power source or prime mover is called the input link or the crank (A_0A). The output link or the follower connects the moving pivot B to the ground pivot B_0. The coupler or floating link connects the two moving pivots, A and B, thereby coupling the input to the output link.

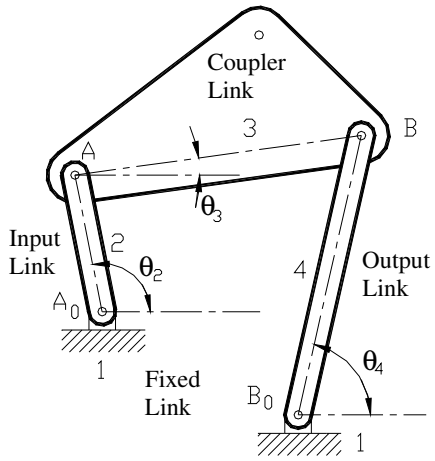


Fig. 2. 1 Four-bar linkage notations

Depending on the relative motion of the rigid bodies, mechanisms can be divided as planar mechanisms and spatial mechanisms. In planar mechanisms, all of the relative

[1] Holonomic constraints are geometric constraints on position or orientation.

motions of the rigid bodies are in one plane or in parallel planes. If there is any relative motion that is not in the same plane or in parallel planes, the mechanism is referred to as spatial mechanism.

Planar mechanisms are simple linkages that can be designed to perform useful and complex tasks and much research has been and is being carried out in developing different approaches for efficient design of planar mechanisms. Traditional methods of analysis, such as graphical and analytical, may be limited when they are applied to complicated problems. Graphical methods, although they provide a good understanding of the kinematics, lack accuracy and tend to be time-consuming. These are the reasons why they are not used for repetitive or three-dimensional analyses. Analytical or closed-form methods can be extremely efficient, although they are application-dependent, and may suffer from an excessive complexity in a multitude of practical problems.

An alternative to overcome these limitations is to resort to numerical analysis and the fast processing of alphanumeric data available in current digital computers that emphasize the use of formulations and computational methods for multibody dynamic simulation. Various general purpose programs for multibody kinematics and dynamics have been given in literatures or made available in the market.

2.2. Problem Definition and Scope of Research

Based on the background (literature review) in chapter 1, it has been deemed desirable to develop a computer-aided dynamic force analysis for planar mechanisms. The focus of the dynamic analysis will be on four-bar linkages. The method applied for the

formulation of the equations of motion is the Newton-Euler's approach.

Therefore, the aim of this research is to develop and demonstrate a computer tool that can help in the dynamic force analysis of single degree of freedom planar mechanisms, particularly the four bar linkage mechanisms. The research includes coded computer modules for the dynamic force analysis of the linkage. In here, dynamic analysis refers to the calculation of required joint forces and torques for the linkage. Hence, dynamic force analysis best defines the problem of this thesis.

2.3. Organization of the Thesis

The other chapters of the thesis deal with different aspects of the research. The first chapter outlined previous research work carried in the field and gave the necessary background to proceed throughout the thesis. The third chapter deals with the basic concepts in the dynamics of multibody systems. In the fourth chapter, time-response analysis of four-bar planar mechanisms is discussed by using the Newton-Euler's approach (the method of Lagrange's multipliers) which leads to a representation of the equations of motion in descriptor form constituting a set of index-3 Differential Algebraic Equations (DAEs). The chapter defines the analysis and simulation problems, and presents the solution procedures to the problem. Also the Baumgarte stabilization technique is used to reduce the index of the DAEs and make the solution tractable by means of standard Ordinary Differential Equation (ODE) solvers. Next, a more efficient and direct stabilization technique called direct constraint violation correction method which has a clear physical meaning, less calculation and obvious correction effects than the Baumgarte's stabilization technique is introduced. Chapter five presents the computation (using Matlab computer language) and discussion of results. Finally, chapter

six gives a conclusion for the thesis, its contribution and possible future research directions. Lastly, a particular numerical example of planar four-bar linkage mechanism is given to demonstrate the application of the developed models, analysis methods and simulation techniques.

2.4. Objectives and Assumptions

In this thesis, we will investigate the dynamic analysis and simulation problems associated with planar four-bar linkages. My principal objective is to create accurate models and practical simulation algorithms that can provide physically sound solutions in describing the gross motion of the rigid body systems. The following are the assumptions and restrictions imposed in this thesis.

Rigidity: We focus our investigations on rigid body systems, which means that macroscopic or global deformations are not allowed when a rigid body is exposed to varying force fields. This is an approximation of the reality.

Point contact: Point contact is assumed to simplify the modeling process. This is implied by the rigidity assumption in the limiting case.

Mass: Mass of each body is assumed to be concentrated at its center of gravity and connection elements like springs, dampers, actuators and joints are assumed to be massless.

Force-acceleration approach: We formulate the system dynamics on the force-acceleration level. Impulse is not allowed.

Friction: All friction effects are neglected in the analysis.

Planar Motion: All links move in parallel planes (two-dimensioinal motion).

Chapter 3

Dynamics of Multibody Systems

3.1 Introduction

The dynamics of multibody systems, such as motion of robotic manipulators, mechanical chains and space crafts, is becoming increasingly important in engineering. A computer simulation of such multibody systems requires a coordinated integration scheme involving several computational aspects. Basic methods for multibody system simulations are provided by the disciplines of dynamics (the multibody formulations), numerical mathematics and computer science.

The job of simulating a multibody mechanical system involves three steps:

1. Creating an idealized model of the system,
2. Formulating equations of motion, and
3. Solving the resulting DAEs numerically.

The first step is the most critical, for it requires the creative application of engineering knowledge and judgment to determine:

a) What characteristics of the system are important,

b) What characteristics should be neglected, and

c) A strategy for modeling the important characteristics using rigid bodies, massless springs, dampers and other idealized elements.

The process of developing a mathematical model for a mechanical system involves describing the dynamic characteristics of its components and the manner in which they interact. By applying the appropriate physical laws and vector mechanics, the dynamic analysis of the system yields a set of governing equations of motion. These equations typically take the form of differential-algebraic equations (DAEs), which can be integrated numerically to simulate the response of the multibody mechanical system.

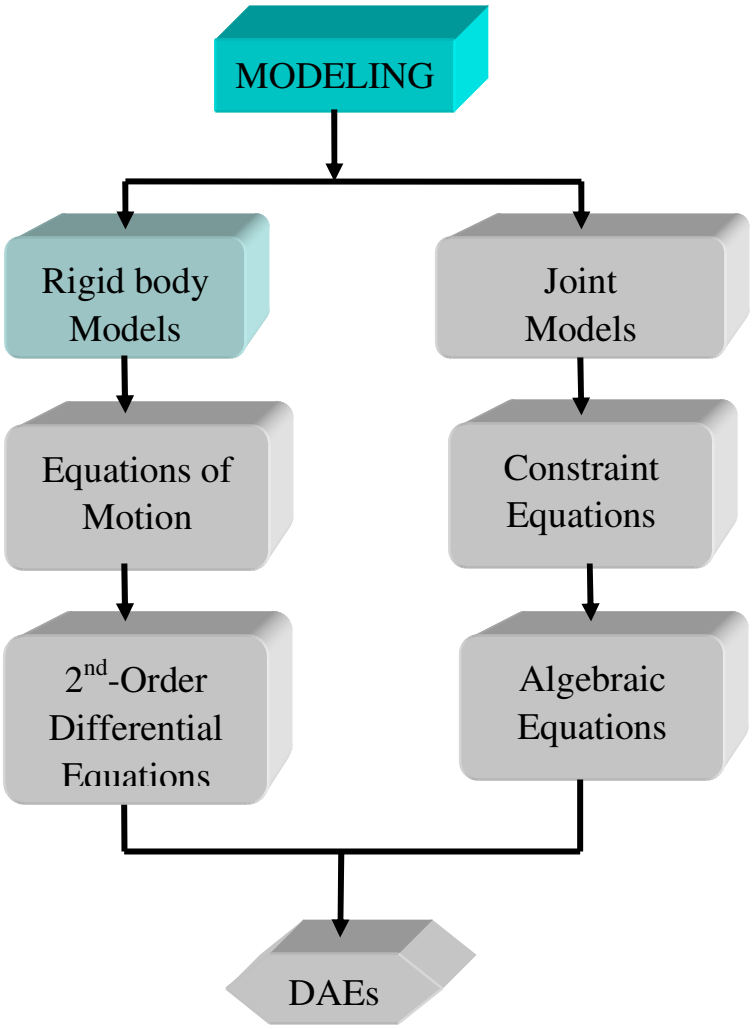


Fig. 3. 1 Mathematical model development procedure for multibody systems

Figure 3.1 shows the steps taken when developing mathematical models (governing equations of motion) for constrained multibody systems.

There are three well-known methods to derive governing equations of motion: Newton-Euler's, Lagrange's [3, 12-14] and Kane's [17] methods. The Newton-Euler's methodology involves introducing a set of Lagrange's multipliers representing reaction forces of the joints which require an additional set of algebraic constraint equations. Consequently, the size of the solution matrix becomes larger, thereby occupying a large memory space for computation. However, this approach is relatively simple to implement. The Lagrange's and Kane's methods select generalized coordinates to eliminate constraint equations from the formulation; therefore, a minimum number of equations are generated. Other approaches like coordinate partitioning [8, 10], velocity transformations [9] transform the equations of motion to a minimum set of coordinates that are directly solvable by ODE methods. For these techniques implementation is not as easy and straight forward as the Newton-Euler's method.

In all cases of formulation, the details of their variations matter little; ultimately, we are faced with a basic choice. Either we model constraints by reducing the number of coordinates needed to describe the system's state or we introduce additional forces into the system to maintain the constraints. The choice of dynamic formulation also determines the subsequent choice of numerical integration schemes.

The configuration of a multibody system is identified by a set of variables called generalized coordinates that completely define the location and orientation of each individual component in the multibody system. Since this thesis considers that the flexibility of formulation is more important than that of having a compact form the

formulation procedure implemented here is based on the Newton-Euler's approach using a maximal set of Cartesian generalized coordinates to facilitate the general formulation of constraints. It involves introducing a set of Lagrange's multipliers representing reaction forces of the joints which require additional set of algebraic constraint equations, i.e., constraints are enforced by introducing constraint forces into the system. The Lagrange multipliers (which we must compute) are a vector of m scalar coordinates that describe the constraint force. As mentioned before, one advantage offered by the Newton-Euler's formulation is the straightforward and efficient manner by which the dynamic equations of motion are assembled numerically. Furthermore, this approach has been established as a reliable and viable algorithm, which has been implemented in several general-purpose programs capable of addressing a broad range of systems such as ADAMS, DADS and IMP [16]. These applications are characterized as so-called "sparse-matrix" methods [6, 12], where a multibody system is broken down into a set of relatively simple components for which the behavior is described by a very large system of DAEs. Additionally, Lagrange multipliers allow us to handle non-holonomic constraints, such as velocity-dependent constraints; reduced-coordinate approaches inherently lack this capability.

The major drawback in the Cartesian approach is that the dimension of the problem increases dramatically as the number of bodies increases when compared to the alternative provided by the reduced formulation. This is a key aspect that justifies the use of the joint formulation for certain mechanical system models.

The accelerations and Lagrange multipliers are the solutions of a system of linear

equations whose coefficient matrix has the special structure with a large number of zero entries. Advantage is taken of both the special structure and the sparsity of the coefficient matrix. Simple manipulations bring the original problem of solving a system of (n + m) equations in (n + m) unknowns to an equivalent problem in which a positive definite system of dimension (n x n) has to be solved for the accelerations. Lagrange multipliers of dimension (m x m) are then efficiently determined.

3.2 Component Models

The modeling of a mechanical system (linkages) by means of a multibody system is characterized by a composition of rigid bodies, interconnected by joints, springs, dampers. Force elements like springs, dampers and actuators acting in discrete attachment points result in applied forces and torques on the rigid bodies. Joints with different kinematic properties constrain the motion of the bodies of the system, reduce the degrees of freedom of the multibody system, and result in constraint forces and torques.

In general, mechanical systems are comprised of two basic types of components: body components and constraint components. Body components represent the most basic building blocks of mechanical systems. Their functionality is dictated by their physical properties and geometry. It should be noted that in the context of this thesis, all bodies are assumed to be rigid. Constraint elements are responsible for prescribing the motion and actuation of the bodies in the system. As such, the determination of the overall response of any mechanical system hinges upon the evaluation of constraining elements. In the following sub-sections, some commonly encountered planar mechanical constraints are formulated.

3.2.1. Kinematic Constraints

Kinematic constraints, or passive constraints, impose restrictions on the relative position and orientation of two or more bodies in the system. The surface to surface contact of the physical joints used to interconnect bodies constrains the relative motion between the connected bodies. Rigid bodies in planar motion each have three degrees of freedom, which correspond to the set of generalized Cartesian coordinates $[x, y, \phi]^T$ associated with each body. The kinematic joints in the system can be described by algebraic constraint equations, denoted as $\mathbf{\Phi}$, each of which eliminates one degree of freedom from the system. The two principal types of kinematic constraints found in planar mechanical systems are revolute joints and prismatic or translational joints.

3.2.1.1. Revolute Joints

A revolute or “pin" joint permits only the relative rotation of two bodies, i and j, about a common point P, as illustrated in Fig. 3.2. The constraint equations for a revolute joint specify that the relative position of the pivot point P remains fixed for both bodies. Thus, a revolute joint eliminates two degrees of freedom from the pair of bodies it connects. The constraint equations for the revolute joint, using vector array notation are given by:

$$\mathbf{\Phi} = \mathbf{r}_j + \mathbf{r}_{P_j} - \left(\mathbf{r}_i + \mathbf{r}_{P_i}\right) = \mathbf{r}_j + \mathbf{A}_j \mathbf{r}'_{P_j} - \mathbf{r}_i - \mathbf{A}_i \mathbf{r}'_{P_i} = \mathbf{0} \qquad \text{..... (3.1)}$$

The $\mathbf{r}_{P_i}$ vector represents the location of point P relative to the mass center of body i as expressed in the body fixed (x'_i, y'_i) frame and the $\mathbf{A}_i$ matrix is the rotational transformation matrix relating the i^{th} and global reference frames.

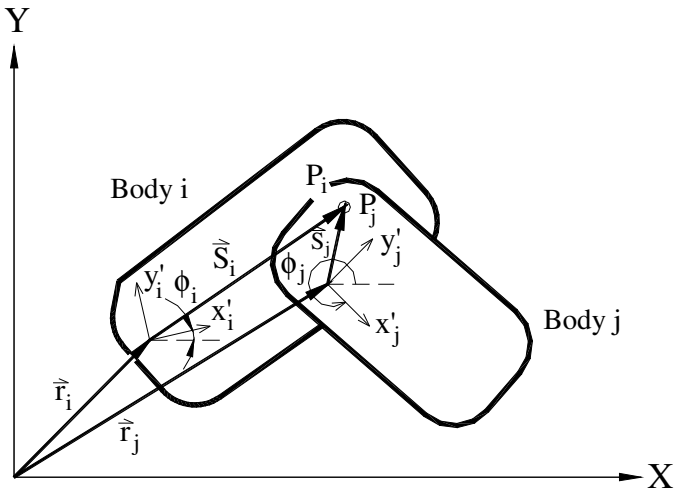


Fig. 3. 2 Revolute Joint

Note that equation (3.1) represents two constraint equations, which can be shown explicitly in expanded matrix form as:

$$\begin{Bmatrix} \mathbf{\Phi}_{r,x} \\ \mathbf{\Phi}_{r,y} \end{Bmatrix} \equiv \begin{bmatrix} x_i + x'_{Pi}\cos\phi_i - y'_{Pi}\sin\phi_i - x_j - x'_{Pj}\cos\phi_j + y'_{Pj}\sin\phi_j \\ y_i + x'_{Pi}\sin\phi_i + y'_{Pi}\cos\phi_i - y_j - x'_{Pj}\sin\phi_j - y'_{Pj}\cos\phi_j \end{bmatrix} = \mathbf{0} \quad \text{..... (3.2)}$$

For a four-bar linkage we have four revolute joints at A, B, O_2 and O_4 (see Fig. 3.3).

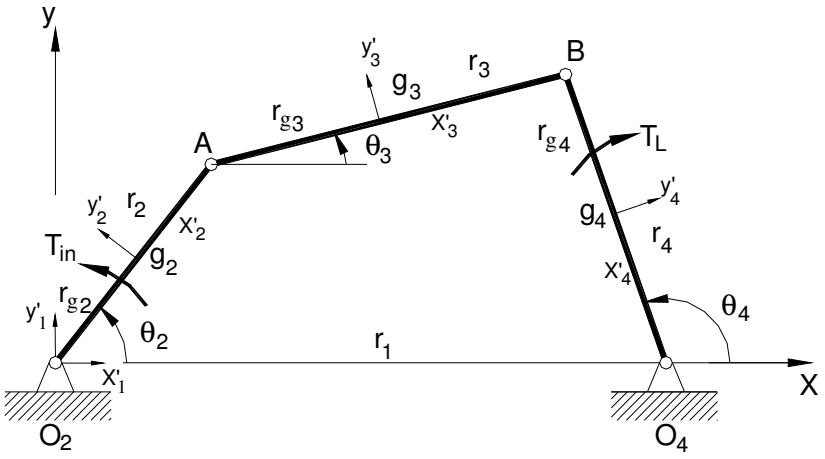


Fig. 3. 3 Four-bar linkage mechanism joint and coordinate properties

Two of these joints at O_2 and O_4 connect link 2 (crank) and link 4 (follower) to the fixed-ground "link". We can choose a reference point on each link in which the body-fixed

frame is attached. The most convenient reference point is the center of mass of each body. Therefore, the local coordinates coincide with the center of mass for each moving link (links 2, 3, and 4) and with the global coordinate system XY for the fixed-ground "link". In addition, the x'-axes of the local coordinates lie along the link lengths so that the y'-axes are perpendicular to the links. This type of local coordinate choice avoids all the y'_{Pi} and y'_{Pj} terms by making them equal to zero. After using this simplifications and applying equation (3.2) for each revolute joint, the structure of the constraint equations will be:

$$
\begin{aligned}
\mathbf{\Phi}_1 &= x_2 - r_{g2}\cos\theta_2 = 0 \\
\mathbf{\Phi}_2 &= y_2 - r_{g2}\sin\theta_2 = 0 \\
\mathbf{\Phi}_3 &= x_2 + \left(r_2 - r_{g2}\right)\cos\theta_2 - \left(x_3 - r_{g3}\cos\theta_3\right) = 0 \\
\mathbf{\Phi}_4 &= y_2 + \left(r_2 - r_{g2}\right)\sin\theta_2 - \left(y_3 - r_{g3}\sin\theta_3\right) = 0 \\
\mathbf{\Phi}_5 &= x_3 + \left(r_3 - r_{g3}\right)\cos\theta_3 - \left(x_4 + r_{g4}\cos\theta_4\right) = 0 \\
\mathbf{\Phi}_6 &= y_3 + \left(r_3 - r_{g3}\right)\sin\theta_3 - \left(y_4 + r_{g4}\sin\theta_4\right) = 0 \\
\mathbf{\Phi}_7 &= x_4 - \left(r_4 - r_{g4}\right)\cos\theta_4 - r_1 = 0 \\
\mathbf{\Phi}_8 &= y_4 - \left(r_4 - r_{g4}\right)\sin\theta_4 = 0
\end{aligned}
\qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.3)
$$

3.2.1.2. Prismatic (Translational) Joints

A prismatic or "slider" joint permits only the relative translation of two bodies, i and j along a direction specified by the geometry of the joint, which is defined as the axis of sliding s. A typical slider joint arrangement is depicted in Fig. 3.4, where the point P on each body identifies the points of attachment to the slider joint between the two bodies.

The vector d_{ij} connecting points P_{i1} and P_j defines relative displacement of body j with respect to body i along the axis of sliding. A third point P_{i2} is placed on body i to

uniquely determine the axis of sliding as $u' = \begin{Bmatrix} x'_{P_{i2}} - x'_{P_{i1}} \\ y_{P_{i2}} - y_{P_{i1}} \end{Bmatrix}$.

The constraint equations associated with a prismatic joint eliminate the relative translation perpendicular to the defined axis of sliding and maintain the relative orientation of bodies i and j. The in-plane vector normal to the axis of sliding is denoted as $u^{\perp}$.

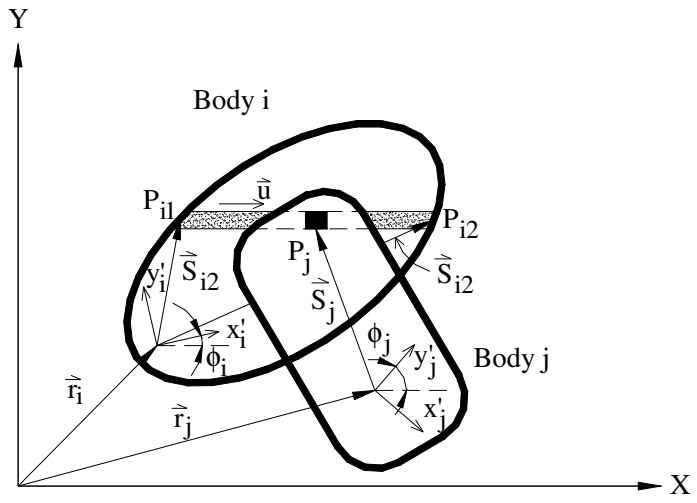


Fig. 3. 4 Prismatic Joint

The constraint equations for the translational joint are given by:

$$\begin{aligned} \mathbf{\Phi}_{s,u^{\perp}} &= u^{\perp} \cdot d_{ij} = 0 \\ \mathbf{\Phi}_{s,\phi} &= \left(\phi_j - \phi_{j0}\right) - \left(\phi_i - \phi_{i0}\right) = \mathbf{0} \end{aligned} \quad \text{..... (3.4)}$$

and in expanded scalar matrix form the constraint equations are:

$$\begin{Bmatrix} \mathbf{\Phi}_{s,u^{\perp}} \\ \mathbf{\Phi}_{s,\phi} \end{Bmatrix} = \begin{bmatrix} \left(x'_{P_{i2}} - x'_{P_{i1}}\right)\left(y'_{P_j} - y'_{P_{i1}}\right) - \left(y'_{P_{i2}} - y'_{P_{i1}}\right)\left(x'_{P_j} - x'_{P_{i1}}\right) \\ \phi_j - \phi_i - \left(\phi_{j0} - \phi_{i0}\right) \end{bmatrix} = \mathbf{0} \text{ ..... (3.5)}$$

3.2.2. Compliant Constraints

Design components of this type describe a compliant connection between a pair of

bodies. Compliant constraints do not directly restrict or prescribe the relative motion of bodies in the system. They do, however, to a large extent determine the force system that is applied to a mechanism or machine. Springs, dampers, force actuators, and torque actuators are examples of some compliant constraint components commonly encountered in mechanical systems.

3.2.2.1. Translational Springs

The most commonly encountered compliant constraint or force element in mechanical systems is the translational or `point to point' spring. Fig. 3.5 illustrates a spring placement between the attachment points P_i and P_j, which remain fixed on bodies i and j respectively. The force developed in a spring is a function of its stiffness and the magnitude of deformation. Typically, a linear stiffness characteristic is assumed for determining the spring force magnitude as:

$$F_{spring} = k(l - l_{uns}) \qquad \text{..... (3.6)}$$

where k is the spring stiffness coefficient, l is the current or deformed length and l_{uns} is the un-stretched length of the spring. A vector expression for the deformed spring length l along the orientation unit vector $\vec{u}$ has the form:

$$\begin{Bmatrix} l_x \vec{\mathbf{i}} \\ l_y \vec{\mathbf{j}} \end{Bmatrix} = \mathbf{r}_j + \mathbf{A}_j \mathbf{s}'_j - \mathbf{r}_i - \mathbf{A}_i \mathbf{s}'_i = 0 \qquad \text{..... (3.7)}$$

The $\mathbf{s}'_i$ vector represents the location of point P_i relative to the mass center of body i as expressed in the body fixed (x'_i, y'_i) frame and the $\mathbf{A}_i$ matrix is the rotational transformation matrix relating the i^{th} and global reference frames. Using the scalar components of equation (3.7), the deformed spring length required to evaluate equation

(3.6) is given by $l = \sqrt{l_x{}^2 + l_y{}^2}$.

The unit vector $\vec{\mathbf{u}}$ defines the orientation of the spring force vector, which determines the line of action for the spring component, as:

$$\vec{\mathbf{u}} = \frac{1}{l}\begin{Bmatrix} l_x\vec{i} \\ l_y\vec{j} \end{Bmatrix} \qquad \text{..... (3.8)}$$

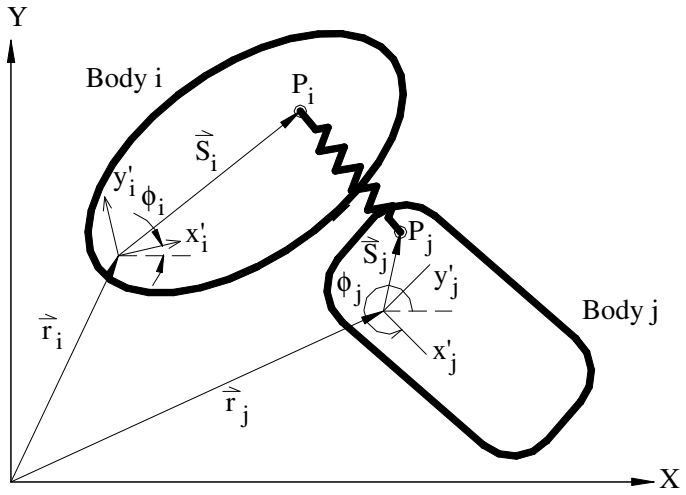


Fig. 3. 5 Translational Spring

Equation (3.6) implies a sign convention whereby tensile forces are positive and compressive forces are negative in magnitude. As a result, the spring force acting on body i is determined by the magnitude of the spring force F_{spring} and the orientation unit vector $\vec{\mathbf{u}}$ (from P_i to P_j), as:

$$\mathbf{F}_i = F_{spring}\vec{\mathbf{u}} \qquad \text{..... (3.9)}$$

In addition to imparting force on body i, the spring element may, in general, apply a moment to the body. A moment results when the center of mass of the body does not lie along the line of action of the spring. The applied moment calculation is based on the vector cross product of the attachment vector $\vec{\mathbf{s}}_i$ and the spring force acting on body i, or

$$\vec{\mathbf{n}}_i = \vec{\mathbf{s}}_i \times \vec{\mathbf{F}}_i \quad \text{..... (3.10)}$$

It is important to note that the vectors appearing in the above equation must be expressed in a consistent reference frame, usually the global (XY) reference frame. Furthermore, following the principals of Newton's third law, the spring force acting on body j must be equal in magnitude and opposite in direction, or

$$\mathbf{F}_j = -F_{spring}\vec{\mathbf{u}} \quad \text{..... (3.11)}$$

Similarly, the moment applied to body j is given by:

$$\vec{\mathbf{n}}_j = \vec{\mathbf{s}}_j \times \vec{\mathbf{F}}_j \quad \text{..... (3.12)}$$

3.2.2.2. Translational Dampers

Translational damping components are often used in parallel with translational springs to dissipate energy and curb the oscillatory response of mechanical systems. Fig. 3.6 illustrates an isolated translational damper placed between the attachment points P_i and P_j, which remain fixed on bodies i and j respectively. The force developed in a viscous damper is a function of its damping coefficient and the rate of change in the component length. Once again, a linear characteristic is typically assumed for planar mechanisms to determine the damping force magnitude, as:

$$F_{damp} = d\dot{l} \quad \text{..... (3.13)}$$

where d is the damping coefficient, l is the current rate of change in the damper component length.

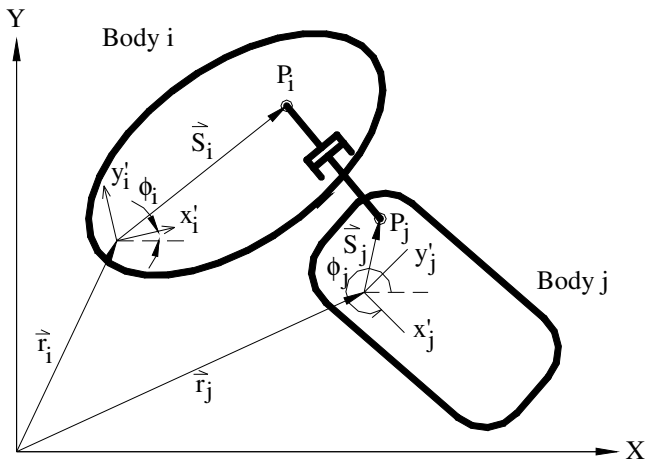


Fig. 3. 6 Translational Damper

A vector expression for the rate of change of length $\dot{l}$ along the orientation unit vector $\vec{\mathbf{u}}$ has the form:

$$\begin{Bmatrix} \dot{l}_x \vec{\mathbf{i}} \\ \dot{l}_y \vec{\mathbf{j}} \end{Bmatrix} = \dot{\mathbf{r}}_j + \mathbf{B}_j \mathbf{s}'_j \dot{\phi}_j - \dot{\mathbf{r}}_i - \mathbf{B}_i \mathbf{s}'_i \dot{\phi}_i = \mathbf{0} \qquad \text{..... (3.14)}$$

where $\dot{\phi}_i$ is the rotational velocity of body i and the $\mathbf{B}_i$ matrix represents the time derivative of the rotational transformation matrix $\mathbf{A}_i$. Using the vector dot product of the component length vector given by equation (3.7) normalized using the scalar length with the vector expression for the component elongation velocity presented above, the scalar rate of change in length required to evaluate equation (3.13) is given by:

$$\dot{l} = \frac{\vec{l} \cdot \dot{\vec{l}}}{l} = \frac{l_x \dot{l}_y + l_y \dot{l}_x}{l} \qquad \text{..... (3.15)}$$

The forces acting on body i and j are determined from the magnitude of the damping force (F_{damp}) and the orientation unit vector $\vec{\mathbf{u}}$, as:

$$\mathbf{F}_i = F_{damp} \vec{\mathbf{u}} \qquad \text{..... (3.16)}$$

and

$$\mathbf{F}_j = -F_{damp}\vec{\mathbf{u}} \qquad \text{..... (3.17)}$$

respectively. Finally, the resultant moments applied to body i and body j are calculated using equations (3.10) and (3.12) respectively.

3.3. Formulation of Equations of Motion

In this thesis, $\mathbf{q} = [\mathbf{q}_1, \mathbf{q}_2 ...., \mathbf{q}_n]^T$ denotes a set of n dependent vector of generalized coordinates that define the state of a multibody system and m is the total number of independent constraint equations (geometric and kinematic), and f = n-m is the number of dynamic degrees of freedom. The generalized coordinates are Cartesian position coordinates and orientation parameters of body centroidal reference frames. Joints connecting the bodies of a mechanical system restrict their relative motion and impose constraints on the generalized coordinates. In most cases, constraints are expressed as algebraic expressions involving generalized coordinates; i.e., expressions of the form:

$$\mathbf{\Phi}(\mathbf{q}, \mathrm{t}) = [\mathbf{\Phi}_1(\mathbf{q}, \mathrm{t}), \mathbf{\Phi}_2(\mathbf{q}, \mathrm{t}), ..., \mathbf{\Phi}_{\mathrm{m}}(\mathbf{q}, \mathrm{t})]^T = \mathbf{0} \qquad \text{..... (3.18)}$$

where $\mathbf{\Phi}(\mathbf{q}, \mathrm{t}) = \mathbf{0}$ are the non-linear loop-closure constraint equations expressed in an implicit form. Each constraint equation is one component of the vector function $\mathbf{\Phi}$. $\mathbf{\Phi}$ is a function of the coordinates, **q**, and time, t, and the constraints are satisfied when $\mathbf{\Phi}$ is zero. In other words, we can regard $\mathbf{\Phi}$ as a measure of the violation of the constraints for a given configuration, **q**; if the violation is zero, the mechanism is assembled. Emphasis should be given to the fact that the constraints are composed of building-blocks from each joint; we shall refer to the block-entries associated with each joint, e.g. $\mathbf{\Phi}^{(\mathrm{jntk},\mathrm{i_k}\,j_k)}$ for the joint *k*. Such block-entries can be formulated once and for all for a given type of joint. The explicit dependence of t implies that driver functions may enter $\mathbf{\Phi}$, whereas

constraints imposed by the joints will not have this dependence of t. Indeed, there must be drivers present for a system to be kinematic determinate, in fact as many as the system has degrees of freedom; thereby we have the same number of constraints and unknown coordinates, which is the necessity for the system to be solvable. It can be very useful to set-up the constraint quantities in an early phase of the system analysis. This provides an overview of quantities, which can be helpful while actually setting up mathematical expressions or in order to understand what is going on when using a software package. Thus, if we generally assume that the system has *n* bodies, *m* joints, and *l* drivers we can write the kinematic constraints in the form:

$$\boldsymbol{\Phi}(\mathbf{q},\mathrm{t}) = \begin{bmatrix} \boldsymbol{\Phi}^{(\mathrm{jnt}_1, i_1 j_1)}\left(\mathbf{q}_{i1}, \mathbf{q}_{j1}\right) \\ \vdots \\ \boldsymbol{\Phi}^{(\mathrm{jnt}_m, i_m j_m)}\left(\mathbf{q}_{im}, \mathbf{q}_{jm}\right) \\ \boldsymbol{\Phi}^{(\mathrm{drv}_1, i_1 j_1)}\left(\mathbf{q}_{i1}, \mathbf{q}_{j1}, t\right) \\ \vdots \\ \boldsymbol{\Phi}^{(\mathrm{drv}_m, i_l j_l)}\left(\mathbf{q}_{il}, \mathbf{q}_{jl}, t\right) \end{bmatrix} = \mathbf{0} \qquad \text{(3.19)}$$

where $\mathbf{q} = \left\lfloor \mathbf{q}_1{}^T \ldots \mathbf{q}_\mathrm{m}{}^T \right\rfloor$.

Let $\mathrm{T}(\mathbf{q},\dot{\mathbf{q}})$ be the kinetic energy of the system, $\mathrm{V}(\mathbf{q})$ the potential energy and $\mathbf{Q}_{ex}(\mathbf{q})$ the vector of generalized external forces acting along the dependent coordinates $\mathbf{q}$ of an unconstrained mechanical system. The Lagrange's equations of such systems have been derived in Chapter 4 of [13] in the form:

$$\frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\partial \mathbf{L}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathbf{L}}{\partial \mathbf{q}} = \mathbf{Q}_{ex} \qquad \text{(3.20)}$$

where $\mathbf{L} = \mathbf{T} - \mathbf{V}$ is the lagrangian function.

After performing the operations indicated in Lagrange's equations (see Appendix C), the terms may be arranged in a form so familiar from robotics literatures as:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{Q}_{ex} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.21)$$

where $\mathbf{M}$ is the inertia matrix, $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})$ consists of centripetal, Coriolis, and gravity terms, and $\mathbf{Q}_{ex}$ contains the generalized input forces and torques, i.e., it is a force vector composed of the external loads and moments acting on the system. Therefore, equation (3.21) leads to a set of n equilibrium equations in which the internal constraint forces are missing. These forces should appear in the equilibrium equations.

Constrained mechanical systems incorporate kinematic joints and drivers to limit and prescribe the permissible motion of a mechanism. Consequently, in order to impose the motion restrictions between bodies, kinematic joints impart reaction forces on the bodies they connect. These forces are purely reactive in nature, and for passive (revolute and translational) joints they always act perpendicularly to the direction of admissible motion. Thus, passive joints do no work in the system. This property of kinematic joints is exploited to derive the equations of motion for kinematically constrained systems of rigid bodies. In order to account for the effect of kinematic constraints on the dynamics of the system a set of Lagrange multipliers are introduced into the formulation. Lagrange multipliers are employed in classical mechanics to reduce a variational or “virtual work" form of the equations of motion into a mixed system of differential-algebraic equations. Using the principle of virtual work the constraint forces can be expressed in terms of the Jacobian matrix, $\mathbf{\Phi}_q$, of the constraint equations and a vector of Lagrange multipliers, as (see Appendix A) :

$$\mathbf{F}_{(c)} = \mathbf{\Phi}_q{}^T \boldsymbol{\lambda} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.22)$$

We need to add this set of forces in the direction of the constraint violation where the columns of $\mathbf{\Phi}_q^T$ (rows of $\mathbf{\Phi}_q$) give the direction of the constraint forces and $\boldsymbol{\lambda}$ is the vector of their unknown magnitudes. Therefore, the method of Lagrange multipliers entails adjoining the constraints to the n differential equations, resulting in a set of n equations in (n + m) unknowns. Adjoining means appending the Jacobian of the constraint matrix, $\mathbf{\Phi}_q$ with a pre-multiplying vector of m undetermined coefficients (Lagrange multipliers) $\boldsymbol{\lambda}$ to the n Lagrange equations.

Adding the reaction force vector $\mathbf{F}_{(c)}$ to the generalized force vector $\mathbf{Q}_{ex}$ of Equation (3.21) and arranging the known terms to the right hand side yields the Lagrange multiplier form of the constrained dynamic equations of motion for the system:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}^{T}(\mathbf{q}, \mathrm{t})\boldsymbol{\lambda} = \mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, \mathrm{t}) \qquad \text{..... (3.23)}$$

where $\mathbf{q}$, $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ are vectors of dimension n representing the generalized position, velocity, and acceleration of the system, $\mathbf{M}$ is the (n x n) generalized mass matrix, $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers, and $\mathbf{Q}(\mathbf{q}, \dot{\mathbf{q}}, \mathrm{t})$ is the vector of generalized external forces plus all the velocity dependent inertia terms (centrifugal and coriolis).

Differentiating equation (3.18) with respect to time yields the kinematic velocity equation:

$$\dot{\mathbf{\Phi}}(\mathbf{q}, \mathrm{t}) = \mathbf{\Phi}_q \dot{\mathbf{q}} + \mathbf{\Phi}_t = \mathbf{0} \qquad \text{..... (3.24)}$$

where subscript denotes partial differentiation; i.e., $\mathbf{\Phi}_{\mathrm{q}} = \left[\dfrac{\partial \mathbf{\Phi}_{\mathrm{i}}}{\partial \mathbf{q}_{\mathrm{j}}}\right]$, and an over dot denotes differentiation with respect to time. Also the term $\mathbf{\Phi}_{\mathrm{t}}$ is given as:

$$\boldsymbol{\Phi}_t(\mathbf{q},t)=\begin{bmatrix}\mathbf{0}\\ \vdots\\ \mathbf{0}\\ \boldsymbol{\Phi}_t^{(\mathrm{drv}_1,i_1j_1)}(\mathbf{q}_{i1},\mathbf{q}_{j1},t)\\ \vdots\\ \boldsymbol{\Phi}_t^{(\mathrm{drv}_m,i_lj_l)}(\mathbf{q}_{il},\mathbf{q}_{jl},t)\end{bmatrix}=\mathbf{0} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.25)$$

Notice the systematic form of $\boldsymbol{\Phi}$ and the derived quantities such as $\boldsymbol{\Phi}_t$ in equation (3.19) and (3.25). For instance $\boldsymbol{\Phi}^{(\mathrm{jnt}_k,i_kj_k)}$ is the constraint violation vector function for joint *k* that connects body i_k and j_k.

It is difficult to illustrate the Jacobian in the same manner as in equation (3.19) or (3.25) because the bodies connected by a given joint are not specified before we have an actual system. However, it is obvious that the constraint Jacobian also will have a very systematic appearance. It will indeed be a sparse matrix composed of block-entries where non-zero blocks only appear in the intersection between the rows of a joint and the columns of the two bodies connected by it as given by equation (3.26) below.

Differentiating equation (3.24) with respect to time yields the acceleration equation:

$$\ddot{\boldsymbol{\Phi}}(\mathbf{q},t)=\boldsymbol{\Phi}_q\ddot{\mathbf{q}}+\dot{\boldsymbol{\Phi}}_q\dot{\mathbf{q}}+\dot{\boldsymbol{\Phi}}_t\equiv\mathbf{0} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.26)$$

Equations (3.18, 3.23, 3.24, 3.26) characterize the admissible motion of the mechanical system. Finally, $\boldsymbol{\Phi}_q(\mathbf{q},t)=\dfrac{\partial\boldsymbol{\Phi}(\mathbf{q},t)}{\partial\mathbf{q}}$ is the (m x n) constraint Jacobian matrix, where m < n. The Jacobian matrix is a matrix of the partial derivatives of the constraints, $\boldsymbol{\Phi}$, with respect to the coordinates, $\mathbf{q}$ and it turns out to be very useful in a number of contexts. The elements of $\boldsymbol{\Phi}_q$ can be explained as:

$$\boldsymbol{\Phi}_{\mathrm{q}} = \begin{bmatrix} & j^{th}\ columns & \\ & \vdots & \\ \cdots & \dfrac{\partial \boldsymbol{\Phi}_i}{\partial \mathbf{q}_j} & \cdots \\ & \vdots & \end{bmatrix} i^{th}\ row \qquad \text{..... (3.27)}$$

which indicates that the i^{th} row and the j^{th} column of $\boldsymbol{\Phi}_{\mathrm{q}}$ contains the partial derivative of the i^{th} constraint with respect to the j^{th} coordinates in $\mathbf{q}$.

In a similar manner $\boldsymbol{\Phi}_{\mathrm{t}}$ is a vector of the partial derivatives of $\boldsymbol{\Phi}$ with respect to t. Notice that $\boldsymbol{\Phi}_{\mathrm{t}}$ only contains non-zero elements for driver equations. The elements of $\boldsymbol{\Phi}_{\mathrm{t}}$ can be explained as:

$$\boldsymbol{\Phi}_{\mathrm{t}} = \begin{bmatrix} \vdots \\ \dfrac{\partial \boldsymbol{\Phi}_i}{\partial t} \\ \vdots \end{bmatrix} i^{th}\ row \qquad \text{..... (3.28)}$$

The kinematic constraints $\boldsymbol{\Phi}(\mathbf{q}, \mathrm{t})$ are assumed to be independent, so the Jacobian matrix has full row rank. The differentiation of Equation (3.26) is not trivial to carry by means of the matrix form. Therefore, we shall collapse the equations into a somewhat less confusing form:

$$\boldsymbol{\Phi}_{\mathrm{q}} \ddot{\mathbf{q}} = \boldsymbol{\gamma} \qquad \text{..... (3.29)}$$

where

$$\boldsymbol{\gamma} = -\left(\boldsymbol{\Phi}_{\mathrm{q}} \dot{\mathbf{q}}\right)_{\mathrm{q}} \dot{\mathbf{q}} - 2\boldsymbol{\Phi}_{\mathrm{qt}} \dot{\mathbf{q}} - \boldsymbol{\Phi}_{\mathrm{tt}} \qquad \text{..... (3.30)}$$

When dealing with a particular set of equations, $\boldsymbol{\gamma}$ is typically easier to find directly from the differentiation with respect to time, than from the matrix form in Equation (3.30).

$\boldsymbol{\gamma}$,$\boldsymbol{\Phi}_q$	Revolute		*Prismatic*	
	$\boldsymbol{\Phi}_{r,x}$	$\boldsymbol{\Phi}_{r,y}$	$\boldsymbol{\Phi}_{s,u^\perp}$	$\boldsymbol{\Phi}_{s,\phi}$
$\boldsymbol{\gamma}$	$\left(x'_{P_i}\cos\phi_i - y'_{P_i}\sin\phi_i\right)\dot{\phi}_i^{\,2}$ $-\left(x'_{P_j}\cos\phi_j - y'_{P_j}\sin\phi_j\right)\dot{\phi}_j^{\,2}$	$\left(x'_{P_i}\sin\phi_i - y'_{P_i}\cos\phi_i\right)\dot{\phi}_i^{\,2}$ $-\left(x'_{P_j}\sin\phi_j - y'_{P_j}\cos\phi_j\right)\dot{\phi}_j^{\,2}$	$\left[\Delta x'_{P_i}\left(y_i - y_j\right) - \Delta y'_{P_i}\left(x_i - x_j\right)\right]\dot{\phi}_i^{\,2} +$ $2\left[\Delta x'_{P_i}\left(\dot{x}_i - \dot{x}_j\right) - \Delta y'_{P_i}\left(\dot{y}_i - \dot{y}_j\right)\right]\dot{\phi}_i$	*0*
$\frac{\partial\boldsymbol{\Phi}}{\partial x_i}$	1	0	$\Delta x'_{P_i}\sin\phi_i + \Delta y'_{P_i}\cos\phi_i$	*0*
$\frac{\partial\boldsymbol{\Phi}}{\partial y_i}$	0	1	$-\Delta x'_{P_i}\cos\phi_i + \Delta y'_{P_i}\sin\phi_i$	*0*
$\frac{\partial\boldsymbol{\Phi}}{\partial \phi_i}$	$-\left(x'_{P_i}\sin\phi_i + y'_{P_i}\cos\phi_i\right)$	$\left(x'_{P_i}\cos\phi_i - y'_{P_i}\sin\phi_i\right)$	$-\left(x_{P_j} - x_i\right)\Delta x'_{P_i} - \left(y_{P_j} - y_i\right)\Delta y'_{P_i}$	*1*
$\frac{\partial\boldsymbol{\Phi}}{\partial x_j}$	-1	0	$-\Delta x'_{P_i}\sin\phi_i - \Delta y'_{P_i}\cos\phi_i$	*0*
$\frac{\partial\boldsymbol{\Phi}}{\partial y_j}$	0	-1	$\Delta x'_{P_i}\cos\phi_i - \Delta y'_{P_i}\sin\phi_i$	*0*
$\frac{\partial\boldsymbol{\Phi}}{\partial \phi_j}$	$-\left(x'_{P_j}\sin\phi_j + y'_{P_j}\cos\phi_j\right)$	$-\left(x'_{P_j}\cos\phi_j - y'_{P_j}\sin\phi_j\right)$	$\left(x'_{P_j}\cos\phi_j - y'_{P_j}\sin\phi_j\right)\Delta x'_{P_i} +$ $\left(x'_{P_j}\sin\phi_j + y'_{P_j}\cos\phi_j\right)\Delta y'_{P_i}$	*-1*

Note: $\Delta x'_{P_i} = x'_{P_{i2}} - x'_{P_{i1}}$ *and* $\Delta y'_{P_i} = y'_{P_{i2}} - y'_{P_{i1}}$.

Table 3. 1 Jacobian $\boldsymbol{\Phi}_q$ entries and the acceleration constraint right side vectors $\boldsymbol{\gamma}$ for revolute and prismatic joints.

Summarized in Table 3.1 are the Jacobian $\mathbf{\Phi}_q$ entries and the acceleration constraint right side vectors γ for revolute and prismatic joints.

For the four-bar linkage mechanism studied in the next chapter, modeled as a planar mechanical system, the number of generalized coordinates is 9; i.e., three coordinates (two positions and one orientation angle) for each of the three moving links. The number m of constraint equations is 8, two for each of the four revolute joints. The mechanism thus has (9-8=1) one degree of freedom. The situation is more drastic if the mechanism is modeled as a spatial mechanical system. In this case, each body in the model is described by seven generalized coordinates (three positions and four orientation Euler parameters). It can be seen that the dimension of the augmented matrix builds up fast, especially when spatial mechanical systems with few degrees of freedom are considered.

3.4. Solving Differential-Algebraic Equations

In dynamics of mechanical sytems we seek to formulate mathematical descriptions that describe, in some approximation, the motion of that system subject to external forces. The most familiar mathematical descriptions are coupled sets of ordinary differential equations (ODEs). A more general description for dynamic systems, differential-algebraic equations (DAEs), has recently begun to receive attention, as it lends itself well to computer generated dynamic descriptions and results naturally from application of Lagrange's formulation of equations of motion whenever a non-minimal coordinate set is chosen. The constraints on the system are included in the formulation via Lagrange multipliers. DAEs are the focus of this thesis when dealing with the equations of motion.

By definition, a differential-algebraic equation system is a system that consists of differential equations together with algebraic equations as in equations (3.18) and (3.23). Equation (3.18) is a set of algebraic equations resulting from constraint conditions and equation (3.23) is a set of differential equations for differentiation variable $\mathbf{q}(t)$. Equation (3.18) must be solved together with equation (3.23). For nonlinear systems there are many numerical methods available to solve Ordinary Differential Equations (ODEs).

This is not the case for (DAEs). Establishing solvers for DAEs is still a very active research area. Numerical integration of DAEs [23, 27] is often characterized by their index. Roughly speaking, the index equals the number of times the constraints must be differentiated to arrive at a set of ordinary differential equations. The index can be viewed upon as a measure of how far a DAE is from being an ODE. Constrained mechanical systems often have an index equal to three. It is well known that index three systems can not be solved directly by standard ODE solvers and the task of obtaining a numerical solution from a DAE system is more difficult than that of solving an ODE system. Analytical solutions of constrained systems satisfy their constraints, but this is not necessarily true for numerical solutions. In this context several strategies are available for the numerical solution of the DAE of multibody dynamics with stabilization [28, 49].

Many methods have been proposed for the stabilization of higher index DAEs. The methods often involve constraint stabilization, thus obtaining a stabilized index reduction. The basic reason for replacing the original problem by one with lower index is that the reformulated problem is presumably easier, or more convenient to solve numerically. In

the case of multibody systems with holonomic constraints (with closed loops), the original method of integrating DAEs is to differentiate the constraint equations to reduce the DAEs to ODEs so that the integration methods for ODEs can be utilized. However, the state variables, i.e., generalized coordinates and generalized velocities, no longer satisfy the position and velocity constraint equations due to numerical errors indicative of violations in constraints. In general, it has long been recognized that a direct constraint differentiation, especially when it is repeated more than once, leads to (mild) instabilities for long-time numerical integrations. The effect is often measured by the "drift"- the error in the original constraint grows. Hence some stabilization is required. A popular stabilization technique is Baumgarte's Stabilization method which is described in detail in section 4.3.1. Another and more efficient and correct stabilization method is presented in this thesis and it is applied by correcting the constraint violation at each time step in the numerical integration process. It is called a direct violation correction method. This second method is discussed in detail in section 4.3.2.

Chapter 4

Dynamic Simulation of Four Bar Planar Mechanisms

4.1. Introduction

Kinematics and dynamic simulations of multibody systems allow the accurate prediction of the behavior of heavy machinery, spacecraft, automobile suspensions and steering systems, graphic arts and textile machinery, robots, packaging machinery, machine tools, etc. Dynamic simulation means integration of the resulting linear or nonlinear ODEs or DAEs from the formulations to obtain the motion trajectories of the rigid bodies given the inertial properties and the external loads (forces and moments). In this context, computing the joint forces and the accelerations becomes the central goal in the dynamic simulation problem.

The first issue to consider in the simulation process is the selection of a set of parameters or coordinates that will allow to unequivocally define at all times the position, velocity and acceleration of the system. The most useful kinds of coordinates currently used to define the motion of multibody systems are Cartesian coordinates and relative coordinates. These coordinates, when combined with the principles of dynamics, lead to the final form of the equations of motion.

When using explicit numerical integration in solving the equations of multibody dynamics, accelerations are needed to progress the simulation to the next time step, while the Lagrange multipliers are used to determine constraint forces. Numerical simulation of the conventional formulation of a constrained mechanical system exhibits severe stability

problems because of constraint violations in position and velocity. As a result several computational procedures have been proposed to overcome the stability problems.

In engineering practice, the constraint violation stabilization technique presented by Baumgarte [3, 21], borrowed from the feed back control theory, is often applied to solve DAEs because it is conceptually simple and it is easy to implement. Additional terms are added to the acceleration constraint equations to stabilize the violations in position and velocity constraint equations. All generalized coordinates are then treated as independent ones and integrated by using standard numerical integration methods. The drawback of this method is that the selection of correction parameters has no certain rules to follow, it must depend on experience.

Yoon et al. [34] presented a direct correction method to eliminate the constraint violations in numerical simulations. The method corrected the values of state variables directly so that they could fit the constraint equations as well. But when applying this method to correct the values of generalized velocity, the energy function of the system must be calculated. For complex multibody systems, this is difficult and sometimes even impractical in numerical simulation. Q. Yu and I. M. Chen [22] showed a new direct violation correction method which corrected the position and velocity constraint violations at each time step of integration. This method is applied in this thesis.

4.2. Solution Method (Method of Lagrange Multipliers)

The solution of a system of index-3 DAEs by direct finite difference discretization is not possible using explicit discretization methods. The principal approach adopted for the forward dynamics simulation of such systems is the converted ODE approach wherein all the algebraic position and velocity level constraints are differentiated and represented at

the acceleration level to obtain an index-1 DAE (in terms of both the unknown accelerations and the unknown Lagrange multipliers).

The complete set of equations of motion for a kinematically constrained mechanical system can be given as:

$$\mathbf{\Phi} \equiv \mathbf{\Phi}(\mathbf{q}, t) = \mathbf{0} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.1)$$

$$\dot{\mathbf{\Phi}} \equiv \mathbf{\Phi}_q \dot{\mathbf{q}} + \mathbf{\Phi}_t = \mathbf{0} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.2)$$

$$\ddot{\mathbf{\Phi}} \equiv \mathbf{\Phi}_q \ddot{\mathbf{q}} - \boldsymbol{\gamma} = \mathbf{0} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.3)$$

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_q^{\,T} \boldsymbol{\lambda} = \mathbf{Q} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.4)$$

where $\boldsymbol{\gamma} = -\left(\mathbf{\Phi}_q \dot{\mathbf{q}}\right)_q \dot{\mathbf{q}} - 2\mathbf{\Phi}_{qt}\dot{\mathbf{q}} - \mathbf{\Phi}_{tt}$ as given in equation (3.30)

4.2.1. Alternative 1

The following derivation will eliminate the Lagrange multipliers and solve for the generalized acceleration in terms of the known quantities. First using equation (4.4) one can solve for:

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}\left(-\mathbf{\Phi}_q^{\,T}\boldsymbol{\lambda} + \mathbf{Q}\right) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.5)$$

which can only be used if the mass matrix is non-singular, as it will be in most of the cases. By substituting equation (4.5) in equation (4.3), the Lagrange multiplier could be expressed as:

$$\boldsymbol{\lambda} = \left(\mathbf{\Phi}_q \mathbf{M}^{-1} \mathbf{\Phi}_q^{\,T}\right)^{-1}\left(\mathbf{\Phi}_q \mathbf{M}^{-1}\mathbf{Q} - \boldsymbol{\gamma}\right) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.6)$$

Substituting equation (4.6) back in equation (4.4) to eliminate the Lagrange multiplier, the ODE for the closed-loop system will have the form:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}^{*} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.7)$$

where $\mathbf{Q}^{*} = \mathbf{Q} - \boldsymbol{\Phi}_q^{T}\left(\boldsymbol{\Phi}_q \mathbf{M}^{-1} \boldsymbol{\Phi}_q^{T}\right)^{-1}\left(\boldsymbol{\Phi}_q \mathbf{M}^{-1} \mathbf{Q} - \boldsymbol{\gamma}\right)$

Once the generalized acceleration has been solved, one can apply equation (4.6) to obtain the Lagrange multipliers, $\boldsymbol{\lambda}$, and then the reaction forces at the joints can be calculated.

Many of the numerical solution methods for constrained multibody dynamics have been based upon numerical treatment of equation (4.7). Since equation (4.7) consists of n second-order ODEs, excluding the m Lagrange multipliers, the governing equation can be integrated by using standard ODE solvers. However, the solution obtained from equation (4.7) may cause violation in the position and velocity constraint equations of equations (4.1) and (4.2). It is necessary to correct the values of the state variables in equation (4.7) to eliminate the constraint violation (see section 4.3).

4.2.2. Alternative 2

Generally, equations (4.1) and (4.4) represent a set of differential- algebraic equations DAEs of index-3 [27, 30, 31]. In order to reduce the equation to index-1 DAEs, one can use the acceleration kinematic equations (4.3) which are obtained by differentiating the constraint equations (4.1) twice with respect to time. By writing expressions (4.3) and (4.4) jointly, one obtains:

$$\begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_{\mathrm{q}}^{T} \\ \boldsymbol{\Phi}_{\mathrm{q}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ -\boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{Q} \\ \boldsymbol{\gamma} \end{Bmatrix} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.8)$$

which is a system of (n + m) equations with (n + m) unknowns. We assume for simplicity that the mass matrix is symmetric positive definite and that the Jacobian matrix has a full row rank for all **q** encountered and also very sparse in many practical cases.

During numerical simulation of the system dynamics, the second-order differential-algebraic equations of motion must be integrated at each time step. Since no derivatives of the Lagrange multipliers $\boldsymbol{\lambda}$ appear in equation (4.8), however, the set of mixed differential-algebraic equations of motion must first be solved as a set of linear algebraic equations. The resulting set of generalized accelerations $\ddot{\mathbf{q}}$ must then be numerically integrated to determine the generalized coordinates and velocities used to evaluate the system at the following time step.

A number of comments could be accompanied with this form of the equations of motion:

- The system of equations is linear in $\ddot{\mathbf{q}}$, $\boldsymbol{\lambda}$ and can be solved by standard methods.
- The coefficient matrix is symmetric due symmetry of $\mathbf{M}$.
- The coefficient matrix is very sparse, because $\mathbf{M}$ is block diagonal and also the constraint Jacobian, $\boldsymbol{\Phi}_q$, is sparse in itself.

The main advantage of the dynamic formulation in dependent coordinates using Lagrange multipliers, besides the conceptual simplicity of the method, is permitting the calculation of forces associated with the constraints (which depend on the Lagrange multipliers) with a minimum additional effort. A simple but crude method for obtaining the dynamic response of a system represented by equations (4.1) to (4.4) is to employ the following direct integration algorithm.

4.2.3 Algorithm DI

a) Input data (link masses, moments of inertia, lengths, centroid distances, input and load torques, and acceleration due to gravity).

b) Specify initial conditions $\mathbf{q}^0$ and $\dot{\mathbf{q}}^0$ for position and velocity.

c) Evaluate the sub-matrices $\mathbf{M}$, $\mathbf{\Phi_q}$, $\boldsymbol{\gamma}$ *and* $\mathbf{Q}$ to construct the descriptor matrix.

d) Solve the descriptor matrix (using Equation (4.8)) for $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$.

e) Numerically integrate $\ddot{\mathbf{q}}$ to get $\mathbf{q}^1$ and $\dot{\mathbf{q}}^1$ for the next time step.

f) Return to step (b).

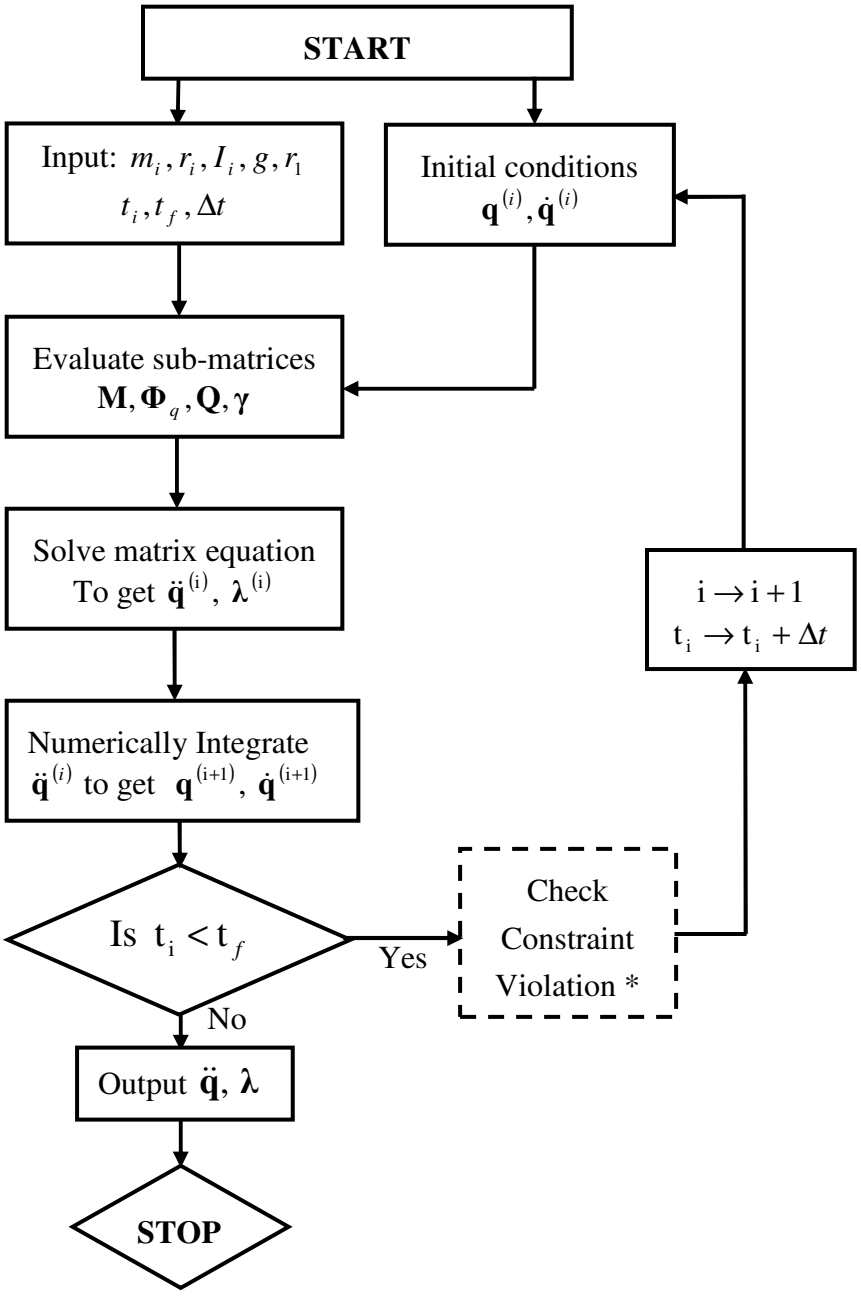


Fig. 4. 1 Algorithm for Direct Integration (without constraint violation stabilization)

* Constraint violation stabilization is added at this position (see Fig. 4.3).

The idea is illustrated using the simplest possible method as follows. First from the solution of the equations of motion, equation (4.8), the accelerations, $\ddot{\mathbf{q}}$, are known. In order to set up and solve these equations, we must know the time, *t*, the positions, $\mathbf{q}$, and the velocities, $\dot{\mathbf{q}}$, because of the dependency of the coefficients and right-hand side of the equations in equation (4.8). At the beginning of the analysis, the values of these quantities are referred to as the initial conditions ($\mathbf{q}\big|_{t=t_0} = \mathbf{q}^0$ and $\dot{\mathbf{q}}\big|_{t=t_0} = \dot{\mathbf{q}}^0$) and they must be specified for the analysis to be possible. To apply a simple time-integration scheme, we assume that the found accelerations are constant in some interval of time, Δt. This assumption allows us to find the new velocities and positions after this time step Δt. We can write time-stepping expressions like:

$$\dot{\mathbf{q}}^{(i+1)} = \dot{\mathbf{q}}^{(i)} + \ddot{\mathbf{q}}^{(i)}\Delta t$$
$$\mathbf{q}^{(i+1)} = \mathbf{q}^{(i)} + \dot{\mathbf{q}}^{(i)}\Delta t + \frac{1}{2}\ddot{\mathbf{q}}^{(i)}\Delta t^2 \qquad \text{..... (4.9)}$$

where i indicates values known at time t_i and $i+1$ indicates values at time $t_{i+1} = t_i + \Delta t$. The initial conditions for time t_0 are denoted $\mathbf{q}^{(0)}$ and $\dot{\mathbf{q}}^{(0)}$. After each time step using equation (4.9) and prior to the next one, the new values of $\mathbf{q}^{(i+1)}$ and $\dot{\mathbf{q}}^{(i+1)}$ must be used to find new values of $\ddot{\mathbf{q}}^{(i+1)}$ from the equations of motion (4.8).

This numerical time-integration method in equation (4.9) is known as the Cauchy-Euler method. It is a general and very simple approach for integrating second-order differential equations, but it is not a choice to be recommended for multibody system analysis due to stability problem when the time step size is increased. The method is given here just to

illustrate the idea of numerical integration in the dynamics of multibody systems.

In order to enable usage of standard methods for numerical time-integration like ode45 of Matlab, the governing equations should be formulated as ordinary differential equation (See Appendix C). This implies that we must turn the equations into first-order differential equations having the general state-space form $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathrm{t})$ where $\mathbf{y}$ is the state vector, i.e., $\mathbf{y} = [\mathbf{q}, \dot{\mathbf{q}}]^T$ and $\mathbf{f}$ is a vector function that supplies the derivatives of the state.

The initial conditions are the value of state vector, denoted $\mathbf{y}_0 = [\mathbf{q}_0, \dot{\mathbf{q}}_0]^T$, at the starting time, t_0. If these are known, $\mathbf{f}$ can be evaluated at this time, $\dot{\mathbf{y}}_0 = \mathbf{f}(\mathbf{y}_0, \mathrm{t}_0)$. The derivatives, $\dot{\mathbf{y}}_0$, can be used for "integrating" or stepping forward in time providing new estimates of the state $\mathbf{y}_1$ at a new time t_1. This can be done iteratively until the ending time (or ending condition) of the analysis has been reached. This as a whole is the simplified principle of time-integration; however, efficient and reliable implementation of the idea is a large and complicated issue.

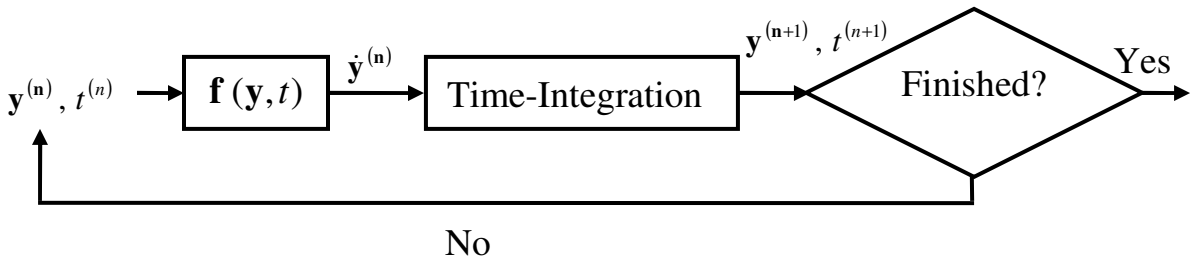


Fig. 4. 2 Time-integration

The second-order differential equations of motion can be transformed into first-order state-space form by doubling the dimension of the state compared to coordinate vector. We assemble the state vector of positions and velocities:

$$\mathbf{y} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix}, \qquad \dot{\mathbf{y}} = \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} \qquad \text{..... (4.10)}$$

The function $\mathbf{f}$ must compute the accelerations $\ddot{\mathbf{q}}$ from the equations of motion.

Notice that the definition of the state containing both positions and velocities fits very well with the formulation of the equations of motion , equation (4.8): all entries in coefficient matrix and on right-hand side are assumed to be functions of namely position, velocities, and time, i.e., exactly what is supplied to the function $\mathbf{f}$ in the arguments $\mathbf{y}$ and *t*. The initial conditions $\mathbf{q}$ and $\dot{\mathbf{q}}$ must satisfy equations (4.3) and (4.4). However, on account of the numerical integration error, these equations may be violated. In the preceding section some methods for circumventing this problem is presented.

Note: Matlab has several numerical integration algorithms implemented as .m files. To numerically integrate $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ to get $\mathbf{q}$ and $\dot{\mathbf{q}}$ for the next time step of the integration process I have used the most commonly used of these algorithms, ode45, which implements the standard Runge-Kutta algorithm.

The systems of differential equations (4.8) suffer from the constraint stabilization problem. As only the acceleration constraint equations have been imposed, the positions and velocities provided by the integrator suffer from the "drift" phenomenon. Here it must be noticed and strongly emphasized that the equations of motion formed with kinematic constraints are in fact not ordinary differential equations (ODEs). We have turned the equations into the form of ODEs by two-times differentiation of the constraints, but in fact they are not. They are indeed so-called differential and algebraic

equations or in short DAEs. Cartesian coordinates provides us with such DAEs except for unconstrained systems and so do any other non-minimal set of coordinates; only a minimal (reduced) set of coordinates provides us with real ODEs. The implication of turning DAEs into ODEs like I have done is that there is no guaranty that the constraints, i.e. the algebraic equations, are fulfilled. The acceleration constraints, i.e. the second derivatives, will be fulfilled, but the position and velocity constraints may actually be violated. Such constraint violation will typically arise from the numerical integration due to round-off and truncation errors and accumulation of the errors will eventually make them significant over longer simulation time intervals.

4.3. Stabilization of Constraint Violations

Constraint violation stabilization is an old problem in numerically simulating the motion of multibody systems and much has been done on it during the last 30 years. The descriptor form of constrained multibody systems and any general formulation of such systems with closed-loops yields index-3 differential-algebraic equations (DAEs) as in equation (4.8). Generally, some index reduction techniques have to be used before numerical discretization since a direct differentiation of constraints introduces mild instability. There are two basic ways to reduce constraint violations in numerical simulation. One is by improving the stabilization conditions of the dynamic equations, of which one popular method is the Baumgarte's technique [21]. The other is by correcting the violation at each time step in the numerical integration process, which is called direct violation correction [27, 34]. Both methods are discussed in detail below. Other methods for constraint stabilization are given in [22, 24, 25, 30, 31, 33, 35].

4.3.1. Baumgarte's Stabilization Method

Baumgarte's constraint violation stabilization method adopted concepts from feedback control theory to construct a modified differential equation, which implicitly accounted for the violation in constraint equation. It is widely used and very effective in controlling the violations.

We let $\mathbf{q}$ denote the real positions, which we ideally want to determine. The numerically computed positions will differ from these due to integration errors. We denote these $\mathbf{q}^*$ and the error will be $\Delta\mathbf{q} = \mathbf{q}^* - \mathbf{q}$. In the integration, we satisfy that $\Delta\ddot{\mathbf{q}} = \mathbf{q}^* - \ddot{\mathbf{q}}$; however, numerical errors are added so that $\Delta\mathbf{q}$ and $\Delta\dot{\mathbf{q}}$ may differ from zero. In order to stabilize these errors close to zero, we can introduce a feedback control:

$$\Delta\ddot{\mathbf{q}} + 2\alpha\Delta\dot{\mathbf{q}} + \beta^2\Delta\mathbf{q} = \mathbf{0} \qquad \text{..... (4.11)}$$

where α and β are control constants. The control theory has shown that by setting $\alpha = \beta$ we can achieve critical damping.

Pre-multiplying the constraint Jacobian matrix to equation (4.11), we obtain:

$$\mathbf{\Phi}_q\Delta\ddot{q} + 2\alpha\mathbf{\Phi}_q\Delta\dot{\mathbf{q}} + \beta^2\mathbf{\Phi}_q\Delta\mathbf{q} = \mathbf{0} \qquad \text{..... (4.12)}$$

The terms $\mathbf{\Phi}_q\Delta\mathbf{q}$ and $\mathbf{\Phi}_q\Delta\dot{\mathbf{q}}$ can be considered as linearized versions of the position and velocity constraints, respectively, i.e.,

$$\mathbf{\Phi}(\mathbf{q}^*) \approx \mathbf{\Phi}_q\Delta\mathbf{q}, \qquad \dot{\mathbf{\Phi}}(\dot{\mathbf{q}}^*) \approx \mathbf{\Phi}_q\Delta\dot{\mathbf{q}}$$

We can insert these relationships and that $\Delta\ddot{\mathbf{q}} = \mathbf{q}^* - \ddot{\mathbf{q}}$ in equation (4.12) and obtain:

$$\mathbf{\Phi}_q(\ddot{\mathbf{q}}^* - \ddot{\mathbf{q}}) + 2\alpha\dot{\mathbf{\Phi}} + \beta^2\mathbf{\Phi} = \mathbf{0} \qquad \text{..... (4.13)}$$

Rearranging equation (4.13) by introducing $\boldsymbol{\Phi}_q \ddot{\mathbf{q}} = \boldsymbol{\gamma}$, we obtain:

$$\boldsymbol{\Phi}_q \ddot{\mathbf{q}}^* = \boldsymbol{\gamma} - 2\alpha\dot{\boldsymbol{\Phi}} - \beta^2\boldsymbol{\Phi} = \boldsymbol{\gamma}^* \quad \text{..... (4.14)}$$

Appending equation (4.14) and equation (4.4) yields the stabilized form of equation (4.8)

$$\begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_q^T \\ \boldsymbol{\Phi}_q & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ -\boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{Q} \\ \boldsymbol{\gamma} - 2\alpha\dot{\boldsymbol{\Phi}} - \beta^2\boldsymbol{\Phi} \end{Bmatrix} \quad \text{..... (4.15)}$$

where $\ddot{\mathbf{q}}$ represents the computed accelerations. When there is no violation in the constraints, equation (4.15) becomes identical to equation (4.8).

In equation (4.14), we have obtained the acceleration constraints including the stabilizing feedback control. We find that the control terms can be assembled with $\boldsymbol{\gamma}$ on the right-hand side yielding equations in $\ddot{\mathbf{q}}^*$ that are similar to the original acceleration constraints; the only difference is the added terms on right-hand side. We shall denote this right-hand side $\boldsymbol{\gamma}^*$ and by using this instead of the ordinary $\boldsymbol{\gamma}$ in the equations of motion, we have introduced the constraint violation stabilization. We can think of the additional term in $\boldsymbol{\gamma}^*$ as penalties on the violation of the constraints that are not handled exactly by the numerical integration.

The introduced control has a certain response time and it may lead to oscillation of the response. The speed of the controller, and amplitude and frequency of oscillations are determined by the values of α and β. In other words, these values determine the efficiency of the constraint violations suppression. However, no matter the values we must also expect violation, though hopefully insignificant when the values have been chosen properly. Nikravesh [3] suggests values of the constants between 1 and 10 and critical damping, i.e. $\alpha = \beta$, which can serve as directions for initial trial values.

However, it is advisable to assess the effect of varying the values for a given simulation in order to obtain the best performance. The above constraint violation stabilization method described by Baumgarte's stabilization technique is one of the most familiar and commonly used methods, because of its simplicity. However, there are some drawbacks in this method:

1. Different types of violations in the position and velocity constraint equations are not considered because the same correction parameters are used. Hence some violations will be eliminated well while others can not be.
2. This is an indirect correction method. Instead of correcting the values of state variables directly to satisfy the constraint equations, it adds feedback control terms in dynamic equations to control the constraint violations. Therefore, it can not control the violations within a specified error tolerance effectively.
3. The selection of correction parameters α and β has no certain rules to follow, it must depend on experience.

If the values of α and β are too large, the dynamic equations can be influenced seriously, and the obtained state variables are not acceptable to describe the behavior of the system; In contrast, if the values are too small, the violations can not be controlled effectively.

Zhenkuan Pan et al. (1996) developed a method which can select α and β automatically. Expanding $\mathbf{\Phi}_{i+1}$ with Taylor formula at $t = t_i$ $(\mathbf{\Phi}_i \neq 0)$, one can obtain:

$$\mathbf{\Phi}_{i+1} = \mathbf{\Phi}_i + h\dot{\mathbf{\Phi}}_i + \frac{h^2}{2}\ddot{\mathbf{\Phi}}_i + O(h^3) \qquad \text{..... (4.16)}$$

Here h is the step-length. If state variables satisfy $\ddot{\mathbf{\Phi}}_i + \frac{2}{h}\dot{\mathbf{\Phi}}_i + \frac{2}{h^2}\mathbf{\Phi}_i = \mathbf{0}$ there exists $\mathbf{\Phi}_{i+1} = O(h^3)$. Compared with equation (4.13), α and β can be selected as:

$$\alpha = \frac{1}{h}, \qquad \beta = \frac{\sqrt{2}}{h} \qquad \text{..... (4.17)}$$

Based on equation (4.17), the correction parameters can be selected automatically for Baumgarte's constraint violation stabilization method. However, when applying variable step-length integration programs, it is difficult to obtain the current step length. So equation (4.17) becomes impractical.

Nikravesh has studied comparatively the numerical integration of the equations of motion with dependent coordinates without stabilization, with Baumgarte stabilization, and integrating mixed systems of differential and algebraic equations. His conclusions indicate that the Baumgarte stabilization is twice as efficient as the integration of the mixed systems, even though not all the problems examined were satisfactorily solved with the said stabilization method. On the other hand, the direct integration of equations (4.8) produced unacceptable results. The Baumgarte stabilization is general, simple, and numerically efficient. Its computational cost is a small fraction of the total required.

4.3.1.1 Algorithm BCS

An algorithm BCS for the Baumgarte constraint stabilization method can be stated by a slight modification to the algorithm DI:

a) Input data m_k, I_k, r_k, r_{gk} $(k = 2,3,4)$; r_1 (ground/fixed "link"), Q, R (motor characteristics for input torque), TL (load torque), g.

b) Specify initial conditions $\mathbf{q}^0$ and $\dot{\mathbf{q}}^0$ for $\mathbf{q}$ and $\dot{\mathbf{q}}$.

c) Evaluate $\mathbf{M}$, $\mathbf{\Phi}_q$, $\mathbf{Q}$ *and* $\boldsymbol{\gamma}^*$ (values are assigned to α and β).

d) Solve equation (4.24) for $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$.

e) Numerically integrate $\dot{\mathbf{q}}^0$ and $\ddot{\mathbf{q}}$ to get $\mathbf{q}^1$ and $\dot{\mathbf{q}}^1$ for the next time step.

f) Return to step (b).

4.3.2. Direct Violation Correction Method

In this section a direct violation correction method for constrained multibody systems is presented. Using the generalized inverse matrix theorem [43], the violation correction formulation is established without any modification to the dynamic equations of motion. By correcting the values the state variables of the system directly at every time-step, the violation of the constraint equations are efficiently controlled within any specified error tolerance, hence it is suitable for any integration method. Compared to the conventional indirect methods, especially Baumgarte's constraint violation stabilization method, this method has clear physical meaning, less calculation and obvious correction effects. Again, compared to the Yoon's method the proposed method corrects the values of generalized coordinates in a clear way and has a distinct physical meaning. Instead of calculating the energy of the system, this method uses the velocity constraint equations directly to correct the values of the generalized velocities so that the calculation is minimized to the largest extent. In addition, the method does little influence on the original dynamic equations of the system. The detailed algorithm for this method is presented and an example is used to demonstrate the effectiveness of the method.

4.3.2.1. Generalized Coordinate Correction

In this method, the values of generalized coordinates are corrected after each integration time-step in order to satisfy the position constraint equations. No additional items are involved in the equations of motion.

Assuming that the integration is at the i[th] step $t = t_i$ and the generalized coordinates of the system found by numerical integration is $\mathbf{q}_i^*$, when the violation in position constraint equations exceed a given error tolerance, which implies $\mathbf{\Phi}_i = \mathbf{\Phi}(\mathbf{q}_i^*, t_i) \neq \mathbf{0}$, a correction term $\delta\mathbf{q}_i$ should be added to $\mathbf{q}_i^*$ to satisfy $\mathbf{\Phi}(\mathbf{q}_i, t_i) = \mathbf{0}$. The corrected generalized coordinate $\mathbf{q}_i$ could be represented as:

$$\mathbf{q}_i = \mathbf{q}_i^* + \delta\mathbf{q}_i \qquad \text{..... (4.18)}$$

such that

$$\mathbf{\Phi}_i = \mathbf{\Phi}(\mathbf{q}_i, t_i) = \mathbf{\Phi}(\mathbf{q}_i^*, t_i) + \delta\mathbf{\Phi}_i = \mathbf{0} \qquad \text{..... (4.19)}$$

and this leads to:

$$\delta\mathbf{\Phi}_i = -\mathbf{\Phi}(\mathbf{q}_i^*, t_i) \qquad \text{..... (4.20)}$$

Here $\mathbf{\Phi}(\mathbf{q}_i^*, t_i)$ is assumed to be small enough so that it could be approximated by the first variation. The first variation of $\mathbf{\Phi}$ in the position constraint equation (equation (4.3)) can be written as:

$$\delta\mathbf{\Phi} = \frac{\partial\mathbf{\Phi}}{\partial\mathbf{q}}\delta\mathbf{q} = \mathbf{\Phi}_\mathbf{q}\delta\mathbf{q} \qquad \text{..... (4.21)}$$

Here the time t is fixed during the variation, since the correction of $\mathbf{q}_i^*$ is made after each integration time-step with t fixed. By substituting equation (4.21) into equation (4.20), the following equation could be derived:

$$(\mathbf{\Phi}_\mathbf{q})_k \delta\mathbf{q}_i = -\mathbf{\Phi}(\mathbf{q}_i^*, t_i) \qquad \text{..... (4.22)}$$

If the constraint Jacobian matrix $\mathbf{\Phi}_\mathbf{q}$ is a square matrix, the process of correcting $\mathbf{q}_i^*$ is just a conventional Newton-Raphson iteration scheme. However, the number of

generalized coordinates is usually larger than that of the independent coordinates, which means that equation (4.22) has infinite number of solutions and the Jacobian matrix $\mathbf{\Phi_q}$ is not a square matrix. Because the constraints are independent, and based on the theorem of generalized inverse of matrices, the inverse of $\mathbf{\Phi_q}\mathbf{\Phi_q}^T$ exists.

The Moore-Penrose generalized inverse $\mathbf{\Phi_q}^+$ of $\mathbf{\Phi_q}$ is (see Appendix B or [43, 56]):

$$\mathbf{\Phi_q}^+ = \mathbf{\Phi_q}^T \left(\mathbf{\Phi_q}\mathbf{\Phi_q}^T\right)^{-1} \quad \text{..... (4.23)}$$

The Minimum Norm solution of equation (4.22) could be written as:

$$\delta\mathbf{q}_i = -\mathbf{\Phi_q}^+ \mathbf{\Phi}_i = -\left(\mathbf{\Phi_q}\right)_i^T \left[\left(\mathbf{\Phi_q}\right)_i \left(\mathbf{\Phi_q}\right)_i^T\right]^{-1} \mathbf{\Phi}\left(\mathbf{q}_i^*, t_i\right) \quad \text{..... (4.24)}$$

Substituting $\delta\mathbf{q}_i$ in to equation (4.18), the value of generalized coordinate $\mathbf{q}_i^*$ could be corrected to satisfy the position constraint equations assuming that the higher-order terms in equation (4.19) are negligible.

The minimum norm solution defined in equation (4.24) has clear physical meaning, that means not only can this method correct the generalized coordinates to satisfy the constraints, but it also makes the least influence to the equations of motion because $\delta\mathbf{q}_i$ has minimum norm. This is a very important characteristic for the stability of the numerical simulation.

Under the assumption that the constraints are independent, the Moore-Pensor generalized inverse of the constraint Jacobian matrix exists. In this situation, the unit of the generalized coordinates has little influence on the result of violation correction. One need not care about the unit of the generalized coordinates of the displacement and rotation, and can obtain a unique solution. If the constraint equations are not independent, which means the constraint Jacobian is ill-conditioned, a prior procedure should be applied to

eliminate the dependent constraints and ensure that the pseudo-inverse of the constraint Jacobian matrix always exists. This pre-procedure, usually known as Singular Value Decomposition [19, 20, 56], needs an error tolerance to determine the dependent constraints. However, the selection of this error tolerance has no general rule and the unit of displacement coordinates does affect the outcome of the SVD procedure. According to experiences, the error tolerance is selected between 10^{-5} and 10^{-8} and it can achieve the desired numerical result regardless of the unit of displacement coordinates.

4.3.2.2. Generalized Velocity Correction

When the violation in velocity constraint equation is beyond a specified error tolerance, i.e., $\dot{\mathbf{\Phi}}_i = \dot{\mathbf{\Phi}}\left(\mathbf{q}_i, \dot{\mathbf{q}}_i^*, t_i\right) \neq \mathbf{0}$, whereby $\dot{\mathbf{q}}_i^*$ is obtained by the numerical integration and the correction term $\delta\dot{\mathbf{q}}_i$ could be added to the generalized velocity $\dot{\mathbf{q}}_i^*$ to satisfy $\dot{\mathbf{\Phi}}_i = \dot{\mathbf{\Phi}}\left(\mathbf{q}_i, \dot{\mathbf{q}}_i, t_i\right) = \mathbf{0}$. In other words, the generalized velocity could be corrected as:

$$\dot{\mathbf{q}}_i = \dot{\mathbf{q}}_i^* + \delta\dot{\mathbf{q}}_i \qquad \text{..... (4.25)}$$

to satisfy

$$\dot{\mathbf{\Phi}}_i = \dot{\mathbf{\Phi}}\left(\mathbf{q}_i, \dot{\mathbf{q}}_i, t_i\right) = \dot{\mathbf{\Phi}}\left(\mathbf{q}_i, \dot{\mathbf{q}}_i^*, t_i\right) + \delta\dot{\mathbf{\Phi}}_i = \mathbf{0} \qquad \text{..... (4.26)}$$

The first variation of $\dot{\mathbf{\Phi}}$ in the velocity constraint equation (equation (4.4)) could be written as:

$$\delta\dot{\mathbf{\Phi}} = \frac{\partial\dot{\mathbf{\Phi}}}{\partial\dot{\mathbf{q}}}\delta\dot{\mathbf{q}} \qquad \text{..... (4.27)}$$

Here $\dot{\mathbf{\Phi}}\left(\mathbf{q}_i, \dot{\mathbf{q}}_i^*, t_i\right)$ was also assumed to be very small so that it could be approximated by the first variation. However, the variation including $\delta\mathbf{q}$ was not considered in equation (4.27), because the generalized coordinate $\mathbf{q}$ has already been corrected in

previous steps discussed in section 4.3.2.1 to make $\mathbf{\Phi}(\mathbf{q}_i,\ t_i) = \mathbf{0}$. Its variation is equal to zero. Equation (4.27) can be written as:

$$\delta\dot{\mathbf{\Phi}} = \mathbf{\Phi}_{\mathbf{q}}\delta\dot{\mathbf{q}} \qquad \text{..... (4.28)}$$

By substituting equation (4.28) in to equation (4.26), the following equation could be deduced:

$$\left(\mathbf{\Phi}_{\mathbf{q}}\right)_k \delta\dot{\mathbf{q}}_i = -\dot{\mathbf{\Phi}}\left(\mathbf{q}_i,\ \dot{\mathbf{q}}_i^{\ *},\ t_i\right) \qquad \text{..... (4.29)}$$

The minimum norm solution of equation (4.29) is:

$$\delta\dot{\mathbf{q}}_i = -\left(\mathbf{\Phi}_{\mathbf{q}}\right)_i^T \left[\left(\mathbf{\Phi}_{\mathbf{q}}\right)_i \left(\mathbf{\Phi}_{\mathbf{q}}\right)_i^T\right]^{-1} \dot{\mathbf{\Phi}}\left(\mathbf{q}_i,\ \mathbf{q}_i^{\ *},\ t_i\right) \qquad \text{..... (4.30)}$$

Thus the violation in velocity constraint of Equation (4.4) could be eliminated by adding $\delta\dot{\mathbf{q}}_i$ to $\dot{\mathbf{q}}_i^{\ *}$ after each integration time step.

4.3.2.3. Algorithm DVC

The detailed algorithm of the method proposed in section 4.3.2 is as follows:

Step 1. At any time step $t = t_i$, use the values of the generalized coordinates from the numerical integration to calculate $\mathbf{\Phi}\left(\mathbf{q}_i^{\ *}, t_i\right)$.

Step 2. If the violations in $\mathbf{\Phi}\left(\mathbf{q}_i^{\ *}, t_i\right)$ are within a given error tolerance, accept $\mathbf{q}_i^{\ *}$ as $\mathbf{q}_i$ and then go to step 3. Otherwise calculate $\delta\mathbf{q}_i$ in equation (4.24), and then add $\delta\mathbf{q}_i$ as in equation (4.18) to correct $\mathbf{q}_i^{\ *}$. This process should be repeated until the violations in $\mathbf{\Phi}\left(\mathbf{q}_i^{\ *}, t_i\right)$ are within the given error tolerance.

Step 3. Use $\mathbf{q}_i$ and the generalized velocity $\dot{\mathbf{q}}_i^{\ *}$ from the numerical integration to calculate $\dot{\mathbf{\Phi}}\left(\mathbf{q}_i,\ \dot{\mathbf{q}}_i^{\ *},\ t_i\right)$.

Step 4. If the violations in $\dot{\mathbf{\Phi}}\left(\mathbf{q}_i, \dot{\mathbf{q}}_i^*, t_i\right)$ are within the given error tolerance, accept $\dot{\mathbf{q}}_i^*$ as $\dot{\mathbf{q}}_i$ and then go to step 5. Otherwise calculate $\delta\ddot{\mathbf{q}}_i$ in equation (4.30), and then add $\delta\ddot{\mathbf{q}}_i$ as in equation (4.25) to correct $\dot{\mathbf{q}}_i^*$. This process should be repeated until the violations in $\dot{\mathbf{\Phi}}\left(\mathbf{q}_i, \dot{\mathbf{q}}_i^*, t_i\right)$ are within the given error tolerance.

Step 5. Substitute the corrected generalized coordinates $\mathbf{q}_i$ and the generalized velocity $\dot{\mathbf{q}}_i$ into the governing equations of motion to continue the integration.

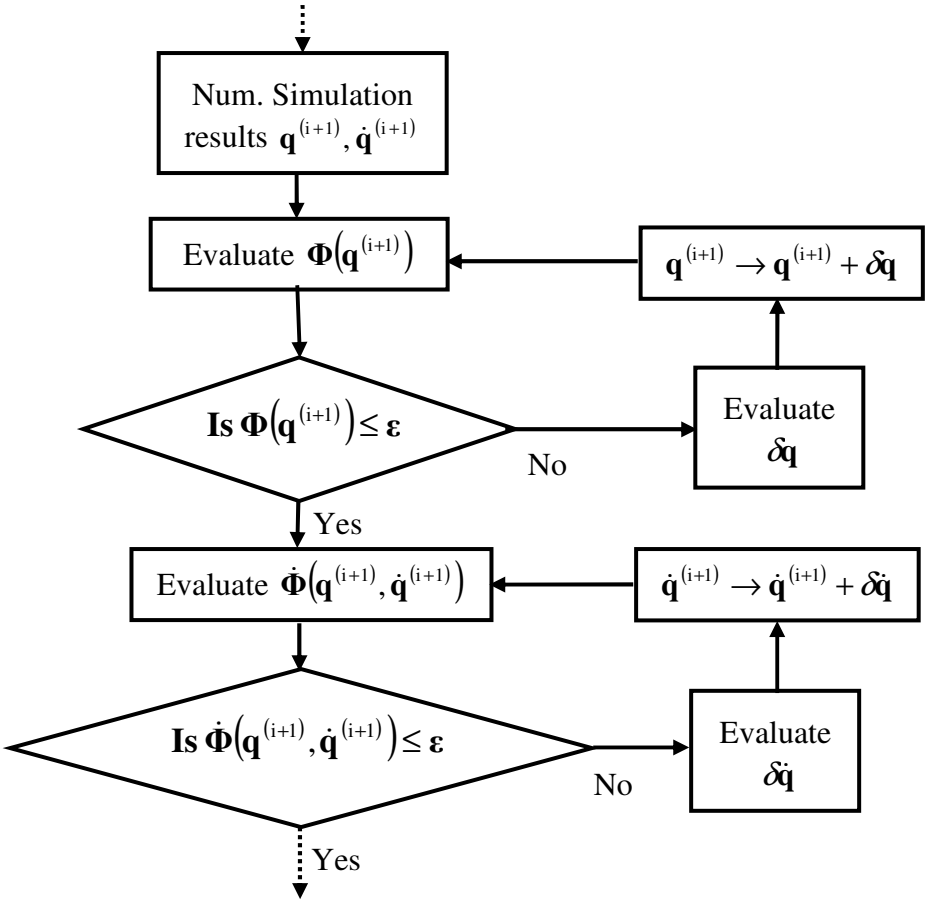


Fig. 4. 3 Algorithm for direct violation correction (combined with Fig. 4.1)

Note: One can use the Jacobian matrix $\mathbf{\Phi}_\mathbf{q}$ and its generalized inverse $\mathbf{\Phi}_\mathbf{q}^+$ directly when correcting $\dot{\mathbf{q}}_i^*$ since it has already been calculated while correcting $\mathbf{q}_i^*$. Also it is a

function of $\mathbf{q}_i^*$ only and is independent of $\mathbf{q}_i$.

4.4. Body of Matrix Equations

In this section the dynamic analysis formulation procedure outlined above and the component models presented in Section 3.2 are brought together in order to assemble the equations of motion for four-bar mechanisms.

Consider the four-bar linkage mechanism depicted in Figure 4.4. The system is composed of three moving rigid bodies of mass m_2, m_3 and m_4, respectively. The configuration can be parameterized in terms of the Cartesian coordinates of the mass center of each link, together with the angle that each link makes with the horizontal. Each body is constrained to rotate about two revolute joints at each of their ends. The weight of the rigid bodies must appear in the force system.

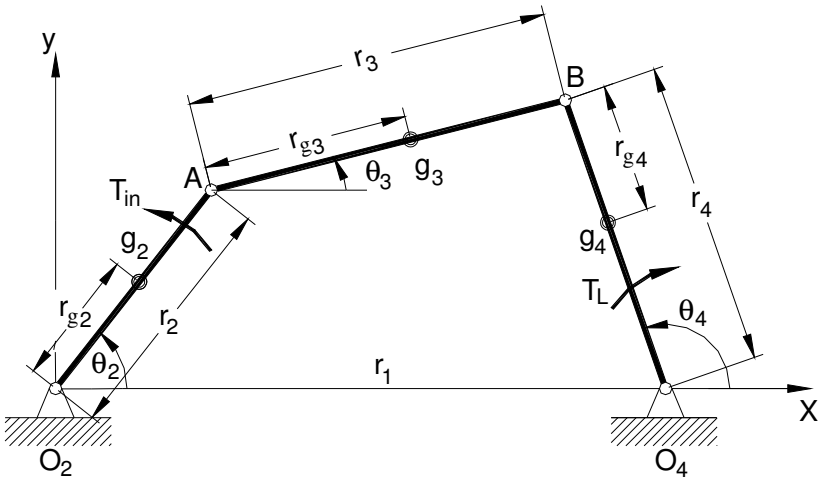


Fig. 4. 4 Parameters for four-bar linkage mechanism

The Cartesian coordinates employ a body-fixed coordinate system for each body. The body-fixed systems are described by a set of coordinates $\mathbf{q}_i$.

4.4.1 Mass Matrix

The mass matrix for the total system is given as

$$\mathbf{M} = \mathbf{diag}\left[\mathbf{M}_2, \mathbf{M}_3, \mathbf{M}_4\right] \quad \text{..... (4.31)}$$

where $\mathbf{M}_k = \begin{bmatrix} m_k & 0 & 0 \\ 0 & m_k & 0 \\ 0 & 0 & I_k \end{bmatrix}$ $k = 2,3,4,$ and I_k = mass moment of inertia of link k.

4.4.2 External Force Vector

To begin, the set of external forces $\mathbf{Q}$ that act on the rigid bodies of the system are evaluated. Contributions to the force system are made from the weight of the three rigid bodies plus spring and damper components, if there are any. The number of bodies in the system determines the dimension of the external force array. The array contains the X and Y components of the external force as well as the resultant moment applied to each body. Therefore, in this case, nine entries are required:

$$\mathbf{Q} = \left[\mathbf{F}_{2,x} \quad \mathbf{F}_{2,y} \quad \mathbf{n}_2 \quad \mathbf{F}_{3,x} \quad \mathbf{F}_{3,y} \quad \mathbf{n}_3 \quad \mathbf{F}_{4,x} \quad \mathbf{F}_{4,y} \quad \mathbf{n}_4\right]^{\mathrm{T}} \quad \text{..... (4.32)}$$

Taking the orientation of the acceleration due to gravity along the negative Y-axis, the weights of body 2, 3, and 4 are expressed simply as:

$$\begin{aligned} \mathbf{F}_{2g,y} &= -m_2 g \\ \mathbf{F}_{3g,y} &= -m_3 g \\ \mathbf{F}_{4g,y} &= -m_4 g \end{aligned} \quad \text{..... (4.33)}$$

If there are any spring, damper and/or actuator forces and moments, they should be included in the generalized external load matrix according to their locations and orientations as derived in section 3.2. Finally, the overall external force vector $\mathbf{Q}$ of equation (4.32) is obtained by assembling the force and moment contributions from

equation (4.33) and any contribution from springs, dampers and/or actuators.

4.4.3 Jacobian Matrix

Matrix $\mathbf{\Phi}_q$ is the Jacobian matrix is the matrix of partial derivatives of the constraint equations with respect to the dependent coordinates. When the Jacobian matrix entries are evaluated, the number of constraint equations determines the number of rows in the Jacobian matrix and the number of columns is dependent upon the number of generalized coordinates. Since this formulation specifies the use of Cartesian coordinates, three columns are required for each body in the system. This matrix takes the following form:

$$\mathbf{\Phi}_q = \begin{bmatrix} \frac{\partial \mathbf{\Phi}_1}{\partial q_1} & \frac{\partial \mathbf{\Phi}_1}{\partial q_2} & ..... & \frac{\partial \mathbf{\Phi}_1}{\partial q_n} \\ \frac{\partial \mathbf{\Phi}_2}{\partial q_1} & \frac{\partial \mathbf{\Phi}_2}{\partial q_2} & ..... & \frac{\partial \mathbf{\Phi}_2}{\partial q_n} \\ .... & .... & ..... & .... \\ \frac{\partial \mathbf{\Phi}_m}{\partial q_1} & \frac{\partial \mathbf{\Phi}_m}{\partial q_2} & ..... & \frac{\partial \mathbf{\Phi}_m}{\partial q_n} \end{bmatrix} \qquad \text{..... (4.34)}$$

In equation (4.34), *m* is the number of constraint equations and *n* the number of dependent coordinates. The constraint Jacobian matrix $\mathbf{\Phi}_q$ has a dual use. In addition to relating the velocities to the rate of change of the constraint function $\mathbf{\Phi}$, the rows of $\mathbf{\Phi}_q$ act as basis vectors for constraint forces. Thus, when we solve for the constraint forces, we actually just need to solve for the coefficient vector $\boldsymbol{\lambda}$ (whose components are the Lagrange multipliers) that contains the magnitudes of the forces that correspond to each of these basis vectors. The total force acting on the system is the sum of the external forces and the constraint forces**.** The generalized Jacobian entries for the revolute joints are calculated (see Table 3.1) and Jacobian matrix will be:

$$\boldsymbol{\Phi}_q = \begin{bmatrix} 1 & 0 & r_{g2}\sin\theta_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -r_{g2}\cos\theta_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -(r_2 - r_{g2})\sin\theta_2 & -1 & 0 & -r_{g3}\sin\theta_3 & 0 & 0 & 0 \\ 0 & 1 & (r_2 - r_{g2})\cos\theta_2 & 0 & -1 & r_{g3}\cos\theta_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -(r_3 - r_{g3})\sin\theta_3 & -1 & 0 & r_{g4}\sin\theta_4 \\ 0 & 0 & 0 & 0 & 1 & (r_3 - r_{g3})\cos\theta_3 & 0 & -1 & -r_{g4}\cos\theta_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & (r_4 - r_{g4})\sin\theta_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -(r_4 - r_{g4})\cos\theta_4 \end{bmatrix}$$

4.4.4 Constraint Acceleration Vector

Lastly, the elements of the constraint accelerations right-hand side vector must be evaluated. The same number of entries is required as the number of constraint equations in the system. Again making use of Table 3.1, the $\boldsymbol{\gamma}$ array becomes:

$$\boldsymbol{\gamma} = \begin{bmatrix} r_{g2}\dot{\theta}_2^{\,2}\cos\theta_2 \\ r_{g2}\dot{\theta}_2^{\,2}\sin\theta_2 \\ (r_2 - r_{g2})\dot{\theta}_2^{\,2}\cos\theta_2 + r_{g3}\dot{\theta}_3^{\,2}\cos\theta_3 \\ (r_2 - r_{g2})\dot{\theta}_2^{\,2}\sin\theta_2 + r_{g3}\dot{\theta}_3^{\,2}\sin\theta_3 \\ (r_3 - r_{g3})\dot{\theta}_3^{\,2}\cos\theta_3 + r_{g4}\dot{\theta}_4^{\,2}\cos\theta_4 \\ (r_3 - r_{g3})\dot{\theta}_3^{\,2}\sin\theta_3 + r_{g4}\dot{\theta}_4^{\,2}\sin\theta_4 \\ (r_4 - r_{g4})\dot{\theta}_4^{\,2}\cos\theta_4 \\ (r_4 - r_{g4})\dot{\theta}_4^{\,2}\sin\theta_4 \end{bmatrix} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.35)$$

Finally, having derived and evaluated the key sub-matrix and vectors required by the dynamic analysis formulation, the governing equations of motion can be assembled according to Equation (4.8).

Chapter 5

Numerical Example and Discussion of Results

5.1. Numerical Example

In this section, an example, found from [3], is studied to evaluate the efficiency and the stability of the proposed method. Three methods are used here to perform numerical simulation: (Method 1) direct integration method without constraint violation control, (Method 2) Baumgarte's constraint violation stabilization method with $\alpha = \beta = 10$ and (Method 3) the direct constraint violation correction method proposed.

The planar four-bar mechanism (see Fig. 5.1) is moving in the gravity field. It consists of three movable links (crank, coupler and follower) of lengths r_2, r_3 and r_4, respectively whose orientations with respect to the horizontal are denoted by angles θ_2, θ_3 and θ_4.

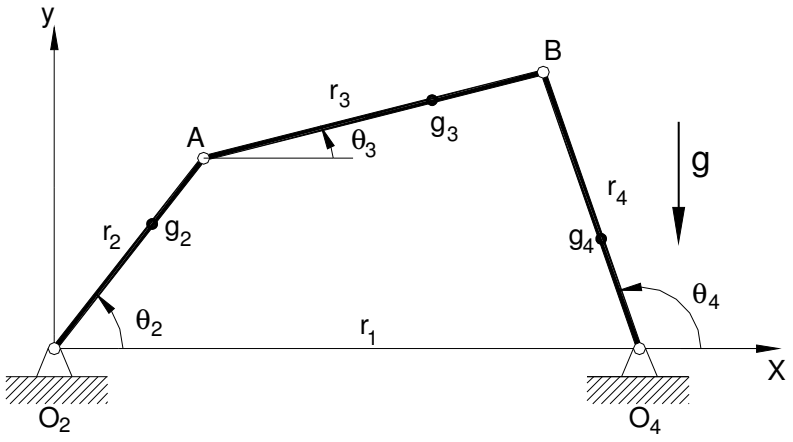


Fig. 5. 1 Example for a planar four-bar mechanism

The mass of each moving link of length r_i is m_i, and the moments of inertia of each moving link about the axis through the center of its mass and perpendicular to the plane of its motion is I_i, where $i = 2,3,4$. The mass center of each link is situated at a distance r_{gi} from the proximal joint of each link. The fixed supports are r_1 apart.

Link	Crank	Coupler	Follower	Fixed(r_1)
Mass (Kg)	1	2.25	2	-
Moments of Inertia (Kg m^2)	0.3	2	1.35	-
Length (m)	2	4	4	2.5
CM Distance- r_{gi} (m)	1	2	2	-

Table 5. 1 Input data

All joints in the system are revolute joints and $\theta_2(0) = \left(\pi/3\right) rad$ and $\dot{\theta}_2(0) = 0 \left(rad/s\right)$.

The input torque is calculated by considering the inertia loads on the links and the load (working) torque. It is found to be T_{in} = 39.75 N.m.

As a time integration method, Runge-Kutta formulas in their Matlab built-in states are used. Simulations are carried out with time step size $\Delta t = 0.025 s$. The analysis is carried out from t = 0 to 5 s. The constraint violation tolerance used for methods 2 and 3 is 10^{-7}.

5.2. Discussion of Results

Figs. 5.2 ~ Fig. 5.5 are output plots of position, velocity, acceleration of each moving link and Lagrange multipliers for each joint. As is evident from the figure 5.3, it takes a very small amount time before the mechanism operates in a steady state. We can see from the figure that the speed of the mechanism is fluctuating about an average angular speed.

These speed fluctuations can generally degrade machine's dynamic performance. In order to reduce the speed fluctuation, a flywheel is usually employed as an energy storage device to absorb the kinetic energy during acceleration and to return the absorbed energy during other parts of the cycle. From the acceleration graph of link 2 (figure 5.4), we can deduce that the input link has a steady fluctuating acceleration value of zero (because of constant velocity).

To evaluate the characteristics of the analysis and the improvements made due to the stabilization techniques, the constraint errors in position and velocity for each method is calculated and a comparison is given in graphical form in Fig.5.6.

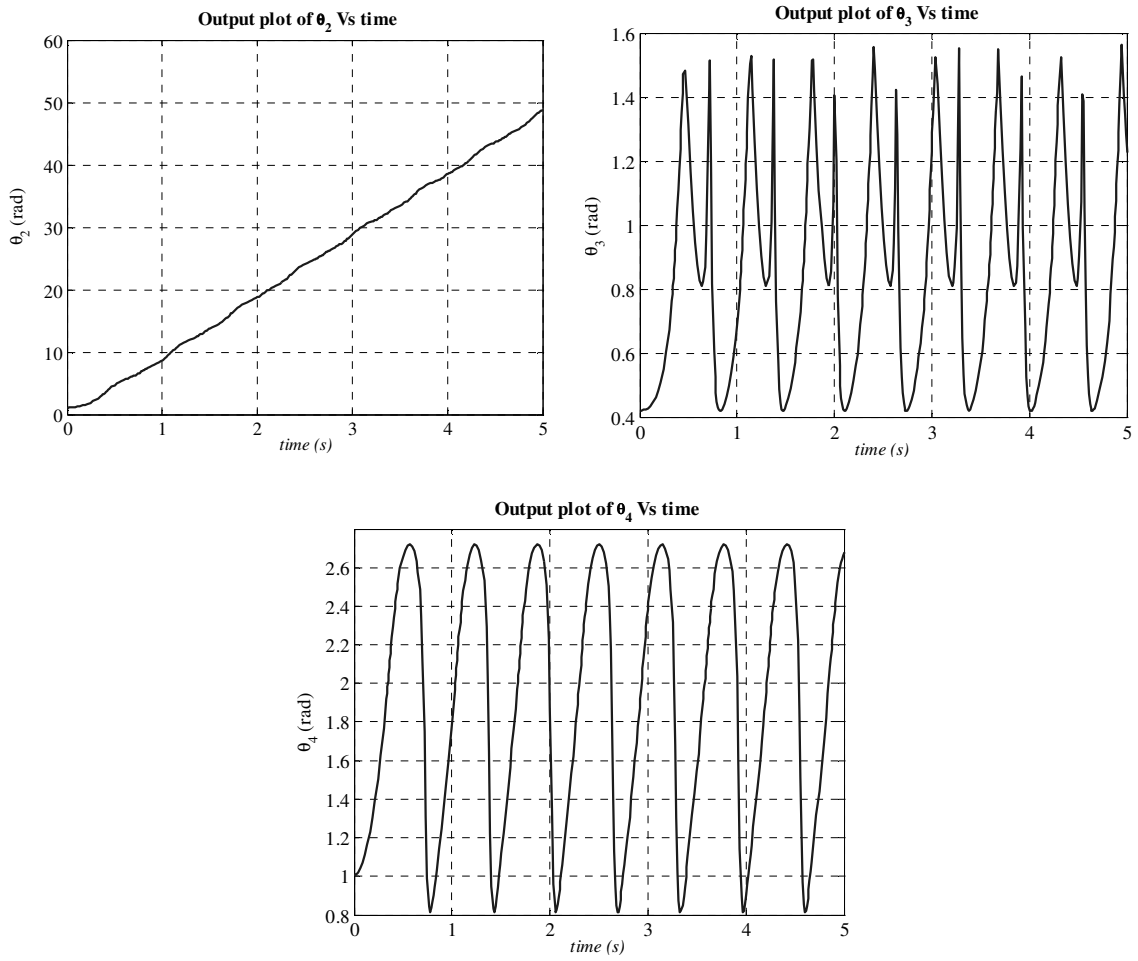


Fig. 5. 2 Output plot of position of links versus time

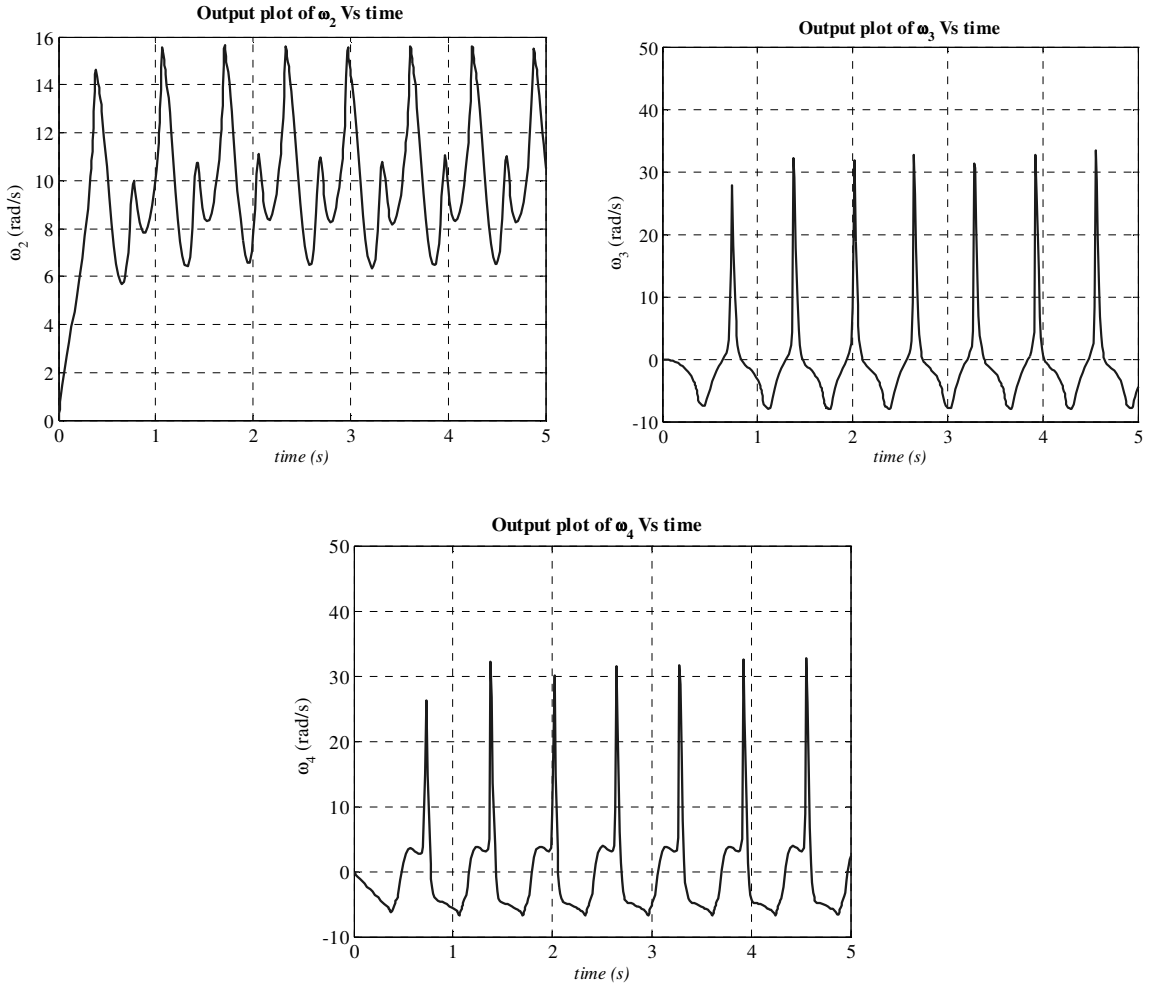
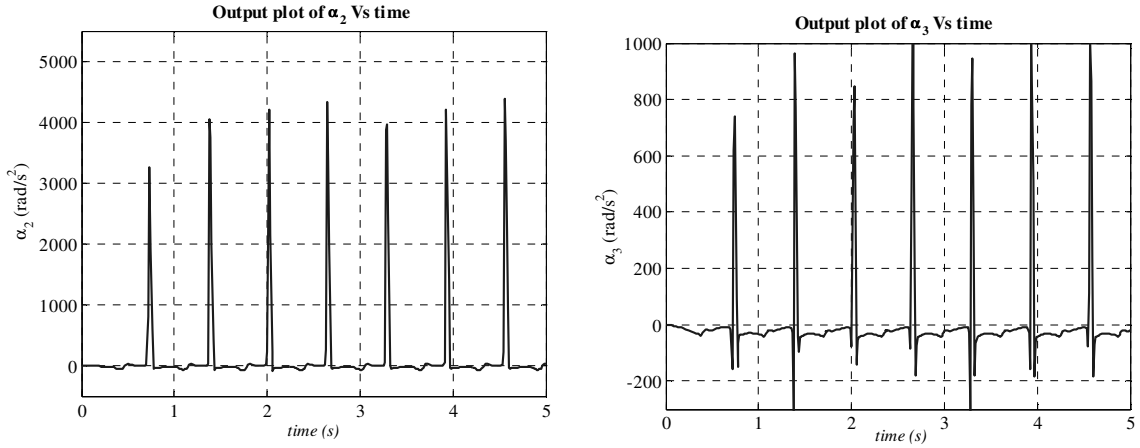


Fig. 5. 3 Output plot of velocity of links versus time



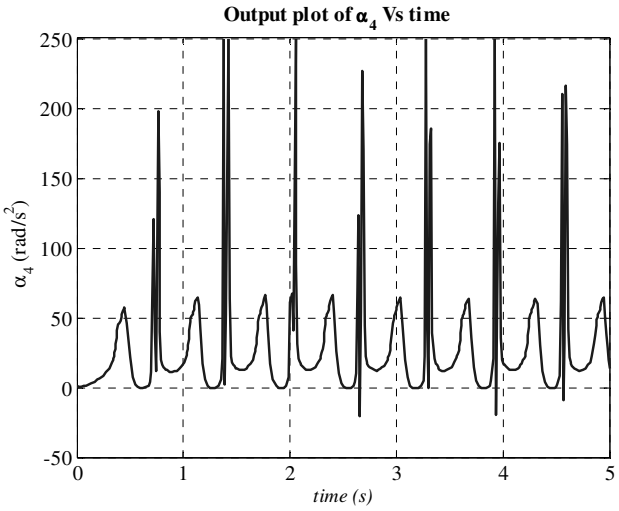


Fig. 5. 4 Output plot of acceleration of links versus time

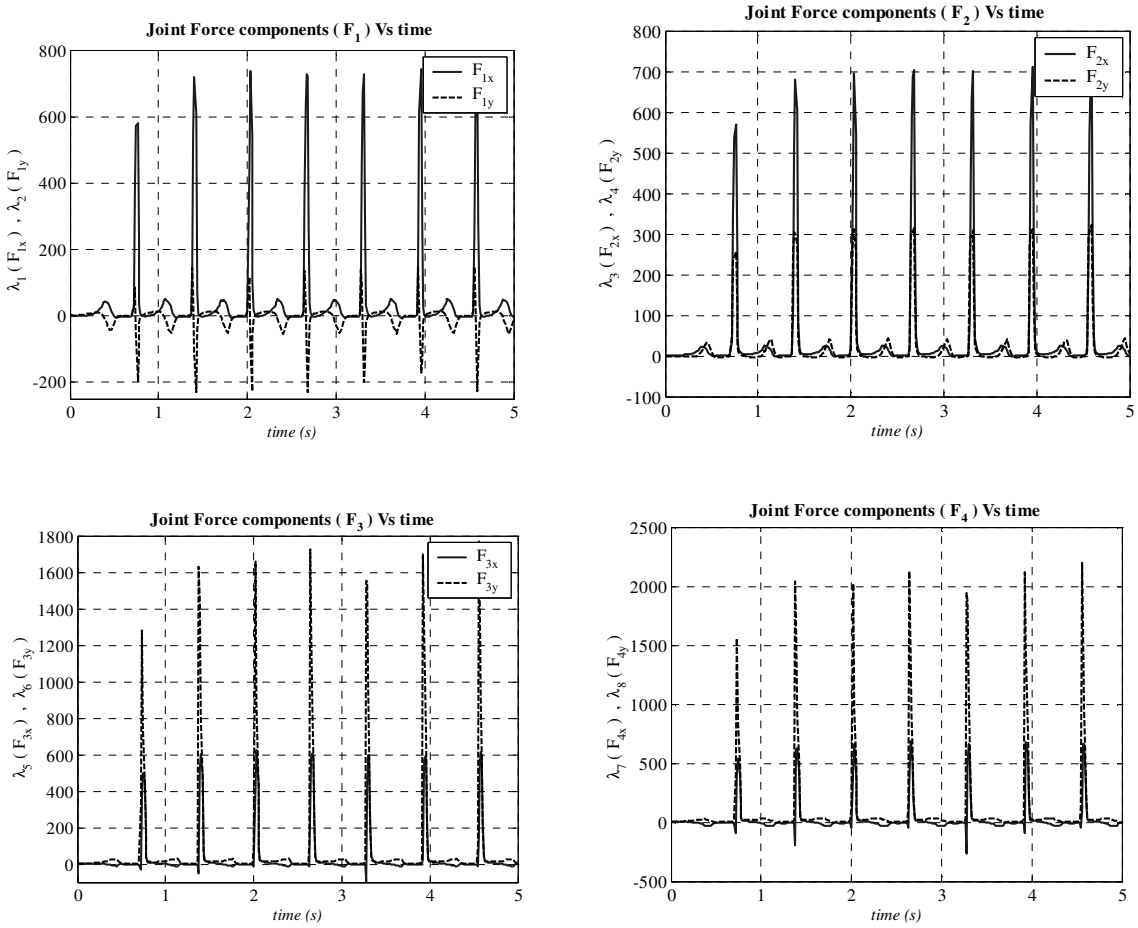


Fig. 5. 5 Output plot of Lagrange multipliers versus time

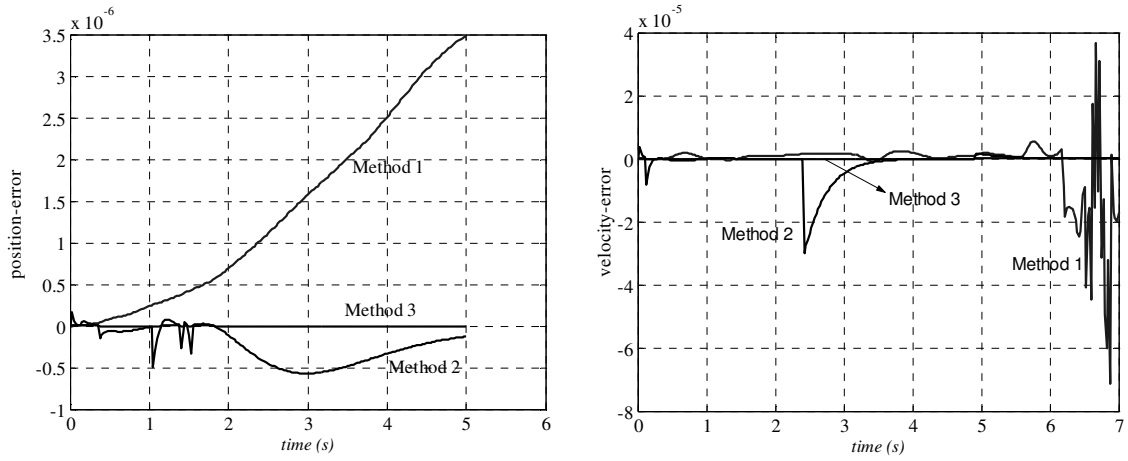


Fig. 5.6 Violation of position and velocity constraint equations versus time

From Fig. 5.6 it is apparent that the constraint violation is serious and may be unlimited when applying the direct integration method without violation control method. That is why we see drastic increase of the constraint errors in position and velocity with the progress of time when no stabilization technique is applied (they both diverge as seen in Fig. 5.6, Method 1). Baumgartes constraint violation stabilization method is a better one, but it can not control the violation within the given error tolerance effectively. We can clearly see that the direct constraint violation method proposed in this thesis has the best violation control effect and the errors in position and velocity are almost completely diminished depending on the error tolerance chosen. The smaller the error tolerance, the more CPU time required for the constraint violation correction and thus for the whole simulation process. Baumgarte's method adds modified terms to the dynamic equation of the system. Therefore, it would affect the dynamic equation inevitably. However, the influence is less than that with the direct integration method. The proposed direct violation correction method would change the dynamic behavior of the system slightly so that values of the state variables could satisfy the constraint equations.

Chapter 6

Conclusion and Recommendation

6.1 Conclusion of the Thesis

The primary focus of this thesis has been to develop and demonstrate a commuter program for the dynamic force analysis of planar four-bar linkages. The formulation of the equations of motion for the dynamic analysis is done based on the Newton-Euler's (Lagrange Multipliers) approach while the analysis of the motion and joint force characteristics is carried out by solving a set of DAEs derived from the formulation. The Runge-Kutta numerical integration version of Matlab's ode45 has been used to integrate the accelerations to go forward in the numerical integration process. It is shown that constraint violation problems exist when the numerical integration progresses with the time step. The thesis presents a direct violation correction method which turned out to be much more effective than the other conventional method, Baumgarte's constraint violation stabilization method. This method could control the violations within any given accuracy by correcting the values of the state variables to satisfy the constraint equations. Unlike Baumgarte's method, the direct constraint violation correction method did not modify the dynamic equation of the system and kept the change of the dynamic behavior to the minimum. It could improve the simulation accuracy of multibody systems significantly. A computer program is written using Matlab to implement the solutions of the analysis problem.

A numerical example has been worked out to compare the validity of the solutions resulting from the computer program against results found from already done literatures [3, 14]. The results found from the computer program match reasonably to the results obtained from the literatures. Hence, to conclude, the program developed in this work can be used for design analysis of the four-bar mechanism. This is so because when designing four-bar linkages or performing stress analysis, we need critical loads applied on the linkages and these critical loads are found from the maximum values of the Lagrange multipliers, λs, which in effect are the components of the joint (bearing) forces. Selection of bearings during design is also affected by the magnitude of these critical loads. This work can also be useful as a benchmark for further research in this area.

6.2 Recommendation for Future Works

As mentioned before, the initial goal of this dissertation research was the creation of analytical tools for determining the dynamics of planar four-bar linkage mechanisms described by differential-algebraic equations. In chapter 2, the assumptions and limitations of this thesis are given. First, the linkages are assumed to be rigid. But in reality linkages of mechanisms deform due to application of forces and they should be modeled as elastic bodies. Secondly, the thesis considers only dynamics of four-bar mechanisms. However, it is possible to include mechanisms with other than four-bar linkages for dynamic force analysis and also mechanisms in general plane (three-dimensional) motion. Friction could also be included in the dynamic force analysis.

References

1. A. G. Erdman, and G.N. Sandor, "*Mechanism Design: Analysis and Synthesis," Vol. I*, 3rd ed., Prentice-Hall International, Inc., 1984.
2. A. G. Erdman, and Sridhar Kota, "*Mechanism Design: Analysis and Synthesis," Vol. I*, 4th ed., Prentice-Hall International, Inc., 2001.
3. Nikravesh, Parviz E., "*Computer-Aided Analysis of Mechanical Systems*," , Prentice-Hall, Inc., New Jersey,1988.
4. Mekonnen G.Selassie, Thesis of Master of Science on "*Computer-Aided Synthesis and Analysis of Planar Mechanisms*" Mechanical Engineering Dept., AAU, 2002.
5. Brian.T. Rundgren, "*Optimized Synthesis of a dynamically based force generating planar Four Bar Mechanism*", thesis of Master of Science, Blacksburg, Virginia, 2001.
6. Orlandea, N., Chace, M. A., and Calahan, D. A., "A Sparrsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems, Part I and II," ASME, *Journal of Engineering for Industry*, Vol. 99, pp. 773-784, 1977.
7. J. Garcia de Jalon, E. Alvarez, F. A. de Ribera, "*Improved Dynamic Formulations for the Dynamic Simulation of Multibody Systems*", 2000.
8. R. Wehage and E. J. Haug, "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Mechanical Systems", *ASME Journal of Mechanical Design*, Vol. 104, pp. 247-255, 1982.
9. S.-S. Kim and M. J. Vanderploeg, "A General and Efficient Method for

Dynamic Analysis of Mechanical Systems Using Velocity Transformations", *ASME Journal of Mechanisms, Transmissions and Automation in Design*, Vol. 108, pp. 176-182, 1986.

10. Nikravesh, P.E. and Haug, E.J. "Generalized Coordinate Partitioning for Analysis of Mechanical Systems with Nonholonomic Constraints." *ASME Journal of Mechanisms, Transmissions, and Automation in Design,* Vol. 105, pp. 379-384, 1983.
11. Barzel R., Barr A,"A Modeling System Based On Dynamic Constraints" in *Computer Graphics Proceedings SIGGRAPH*, pp. 179-187, August 1988.
12. David Baraff, "Linear-Time Dynamics Using Lagrange Multipliers", in *Computer Graphics Proceedings SIGGRAPH*, pp. 137-146, August 1988.
13. J. García de Jalón and E. Bayo, "*Kinematic and Dynamic Simulation of Multi-Body Systems-the real-Time Challenge",* Springer-Verlag, New York, 1994.
14. Chace, M. A., "Analysis of the Time-Dependence of Multi-Freedom Mechanical Systems in Relative Coordinates", *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 89, No. 1, 1967, pp. 119-125.
15. R. Brent Gillespie and J. Edward Colgate," A Survey of Multibody Dynamics for Virtual Environments", in *Proceedings of the ASME Dynamic Systems and Control*, 1997.
16. P. N. Sheth and J. J. Uricker Jr.," IMP (Integrated Mechanisms Program), A Computer-Aided Design Analysis System for Mechanisms and Linkages", ASME, *Journal of Engineering for Industry*, Vol.94: pp. 454-464, 1972.
17. R. Brent Gillespie," Kane's Equations for Haptic Display of Multibody

Systems", University of Michigan, Dept. of Mech. Eng'g, *Haptics-e*, Vol.3. No.2., 2003 http://www.haptics-e.org.

18. Serban, R., Negrut, D., Potra, F. A., Haug, E. J., "A Topology Based Approach for Exploiting Sparsity in Multibody Dynamics in Cartesian Formulation," *Mechanics of Structures and Machines*, Vol. 25, pp 379-39, 1997.
19. Singh, R.P. and Likins, P.W., "Singular Value Decomposition for Constrained Dynamic Systems", *ASME Journal of Applied Mechanics*, Vol. 52, pp. 943-948, (1985).
20. Mani, N.K., Haug, E.J., and Atkinson, K.E., "Application of Singular Value Decomposition for Analysis of Mechanical System Dynamics", *ASME Journal on Mechanisms, Transmissions and Automation in Design*, Vol. 107, pp. 82-87, (1985).
21. J. Baumgarte, "A New Method of Stabilization for Holonomic Constraints", *Journal of Applied Mechanics*, Vol. 50, pp. 869-870, 1983.
22. Q. Yu, I.M. Chen, "A Direct Violation Correction Method in Numerical Simulation of Constrained Multibody Systems", *Computational Mechanics* 26, 52-57, (2000), Springer-Verlag.
23. E. Pennestri, L. Vita," *Strategies for the Numerical Integration of DAE Systems in Multibody Dynamics*", University of Rome, Italy, 2003.
24. Zdravko Terze, Joris Naudet and Dirk Lefeber, "Constraint Gradient Projective Methods for Stabilized Dynamic Simulation of Constrained Multibody Systems", *Proceedings of DETC'03, ASME 2003*.

25. Hong Sheng Chin, "*Stabilization Methods for Simulations of Constrained Multibody Dynamics*", Doctor of Philosophy, University of British Colombia, 1995.
26. http://www.ime.auc.dk/people/employees/md/LectureNotes.htm. "*Dynamics of Multibody Systems*".
27. Claus Bendtsen, Per Grove Thomsen, "*Numerical Solution of Differential Algebraic Equations*", Technical University of Denmark, Lyngby – Denmark, May 1999.
28. Shinya Karasawa, Kiyoshi Sogabe, "Evaluation of Multibody Dynamics", Dept. of Mech. Eng'g, Sophia University, *Proceedings of ACMD 2004*.
29. J. Cardinal, J. Cuadrado, P. Morer and E. Bayo, "A multi-index Variable Time Step Method for the Dynamic Simulation of Multibody Systems", *International Journal of Numerical Methods in Engineering*, Vol. 44, pp. 1579-1598, 1999.
30. Uri M. Ascher, Hongsheng Chin, Linda R. Petzold, Sebastian Reich, "*Stabilization of Constrained Mechanical Systems with DAEs and Invariant Manifolds*", 1994.
31. Edda Eich, Michael Hanke, "*Regularization Methods for Constrained Mechanical Multibody Systems*", 1994.
32. Waseem Ahmed Khan, Venkat Krovi, "Comparison of Two Alternative Methods for Distributed Forward Dynamic Simulation of a Four-bar Linkage", *Proceedings of the WORKSHOP on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*,

Quebec, Canada, 2002.

33. T. Park, E. Haug, "A Hybrid Constraint Stabilization Generalized Coordinate Partitioning Method for Machine Dynamic Simulation", *Journal of Mechanisms, Transmission and Automation Design*, Vol. 108, No.2, pp. 211-216, 1986.
34. Yoon S, Howe RM, and Greenwood DT, "Geometric Elimination of Constraint Violations in Numerical Simulation of Lagrangian Equations", *Journal of Mechanical Design*, Vol. 116, pp1058-1065, 1994.
35. Takashi Nagata, "Variable-Gain Constraint Stabilization for General Multibody Systems with Applications", *Journal of Vibration and Control*, Vol.10, pp. 1335–1357, 2004, Sage Publications.
36. Michael B. Cline and Dinesh K. Pai, "Post-Stabilization for Rigid Body Simulation with Contact and Constraints", *Proc. IEEE International Conference on Robotics & Automation*, 2003.
37. Fisette P, Vaneghem B, "Numerical Integration of Multibody System Dynamic Equations Using Coordinate Partitioning Method in an Implicit Newmark Scheme", *Computer Methods in Applied Mechanics and Engineering*, Vol. 135, pp. 85-105, 1996.
38. Blajer W, "A Projection Method Approach to Constrained Dynamic Analysis", *Journal of Applied Mechanics*, Vol. 59, pp. 643-649, 1992.
39. Blajer W, "An Orthonormal Tangent Space Method for Constrained Multibody Sysstems", *Computer Methods in Applied Mechanics and Engineering*, Vol. 121, pp. 45-57, 1995.

40. Yen J, Chou CC, "Automatic Generation and Numerical Integration of Differential-Algebraic Equations of Multibody Dynamics", *Computer Methods in Applied Mechanics and Engineering*, Vol. 104, pp. 317-331, 1993.
41. Park KC, Chiou JC, "Stabilization of Computational Procedures for Constrained Dynamical Systems", *Journal of Guidance and Control*, Vol. 11, No. 4, pp. 365-370, 1988.
42. Jonathan B. Alberts, "*Contributions to Stability Analysis and Numerical Solution of Differential-Algebraic Systems*", Doctor of Philosophy, University of Washington, 1999.
43. James Allen Fill, Donniell E. Fishkind, "*The Moore- Penrose Generalized Inverse for Sums of Matrices*", Johns Hopkins University, Maryland, 1998.
44. Hajrudin Pasic, Robert L. Williams II, Chunwu Hui, "A Numerical Algorithm for Solving Manipulator Forward Dynamics", *Mechanism and Machine Theory*, Vol. 34, pp. 843-855, 1999.
45. Yang A.T., "Static Force and Torque Analysis of Spherical Four-Bar Mechanisms", *Journal of Engineering for Industry*, Trans. ASME, Series B, Vol. 87, pp. 221-222, 1965.
46. Uicker J.J., Jr., "Dynamic Force Analysis of Spatial Linkages", *Journal of Applied Mechanics*, Trans. ASME, Vol. 89, No. 2, pp. 418-424, 1967.
47. Bagci C., "Static Force and Torque Analysis Using 3x3 Screw Matrix, and Transmission Criteria for Space Mechanisms", *Journal of Engineering for Industry*, Trans. ASME, pp. 90-101, 1971.
48. Paul, B., and D. Krajcinovic, D., "Computer Analysis of Machines with planar

Motion I-Kinematics; II-Dynamics", *Journal of Applied Mechanics*, 1970.

49. Smith, D. A., "Reaction Force Analysis in General Machine Systems", *Journal of Engineering for Industry*, Trans. ASME, Vol. 95, No. 2, pp. 617-623, 1973.
50. Allen, R. R. and Harrell, J. P., " Connection Force Analysis of Mechanisms Described by Explicit Equations of Motion in Generalized Coordinates", *Journal of Mechanical Design*, Trans. ASME, Vol. 104, pp. 168-174, 1982.
51. Gupta, V.K., "Dynamic Analysis of Multi-Rigid-Body Systems", *Journal of Engineering for Industry*, Trans. ASME, pp. 886-891, 1974.
52. Joel A. P. Dunlop, "*Automated Dynamic Analysis and Simulation of Planar Mechanical Systems, With Applications to Flexible Belt Drive Gantry Robots*", thesis on Masters of Applied Science, University of Waterloo, Canada, 1999.
53. Erik Spencer, "*Inverse Dynamics of Non-redundant Closed-Chain Mechanisms*", 2001. url = http://citeseer.ist.psu.edu/522295.html.
54. Brian Tavis Rundgren, "*Optimized Synthesis of Dynamically Based Force Generating Planar Four-bar Mechanism*", thesis on Masters of Science, Virginia Polytechnic Institute and State University, Virginia, 2001.
55. Waseem A. Khan, "*Distributed Dynamics of Systems with Closed Kinematic Chains*", 2002.
56. Les S. Gennings, "Computational Linear Algebra", University of Western Australia, 1999. http://www.maths.uwa.edu.eu/"les/

Appendix A

Constraint Reaction Forces (Principle of Virtual Work)

It is possible to obtain a relationship between the constraint reaction forces, $\mathbf{F}^{(c)}$, and the constraint equations, $\mathbf{\Phi} = \mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$, if the constraint reaction forces can be transformed to a coordinate system consistent with $\mathbf{q}$. This is done as follows. If the joints are assumed to be frictionless, the work done by the constraint forces in a virtual (infinitesimal) displacement $\delta\mathbf{q}$ is zero; i.e.,

$$\mathbf{F}^{(c)T}\delta\mathbf{q} = 0 \qquad \text{..... (A1)}$$

Equation (A1) is applied due to the fact that the work done by a force $\mathbf{f}$ acting on a system and causing a displacement $\mathbf{q}$ is defined as $w = \mathbf{f}^T\mathbf{q}$. Also a virtual displacement of a system is defined as an infinitesimal change in the coordinates of the system consistent with the constraints and forces imposed on the system at time t. The displacement is called a virtual one to distinguish it from an actual displacement of the system occurring in a time interval dt, during which the constraints and forces may change.

The Taylor series expansion of the constraint equation $\mathbf{\Phi} = \mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$ about $\mathbf{q}$ is:

$$\mathbf{\Phi}(\mathbf{q} + \delta\mathbf{q}) = \mathbf{\Phi}(\mathbf{q}) + \mathbf{\Phi}_q\delta\mathbf{q} + higher - order\ terms$$

A displacement $\delta\mathbf{q}$ consistent with this constraints yields $\mathbf{\Phi}(\mathbf{q} + \delta\mathbf{q}) = \mathbf{0}$. Using $\mathbf{\Phi}(\mathbf{q}) = \mathbf{0}$ and eliminating the higher order terms for infinitesimal $\delta\mathbf{q}$, we find that

$$\mathbf{\Phi}_q\,\delta\mathbf{q} = \mathbf{0} \qquad \text{..... (A2)}$$

The vector of n coordinates **q** may be partitioned into a set of m dependent coordinates **u**, and a set of (n – m) independent coordinates ν, as $\mathbf{q} = \left[\mathbf{u}^T, \mathbf{v}^T\right]^T$. This yields a partitioned vector of virtual displacements $\delta\mathbf{q} = \left[\delta\mathbf{u}^T, \delta\mathbf{v}^T\right]^T$ and a partitioned Jacobian matrix $\mathbf{\Phi}_q = \left[\mathbf{\Phi}_u, \mathbf{\Phi}_\nu\right]$. The matrix $\mathbf{\Phi}_u$ is (m x m) and nonsingular, since the constraint equations are assumed to be independent.

If vector $\mathbf{F}^{(c)}$ is also partitioned as $\mathbf{F}^{(c)} = \left[\mathbf{F}_{(u)}{}^{(c)T}, \mathbf{F}_{(\nu)}{}^{(c)T}\right]$, then Equation (A1) can be written as:

$$\mathbf{F}_{(u)}{}^{(c)T}\delta\mathbf{u} + \mathbf{F}_{(\nu)}{}^{(c)T}\delta\mathbf{v} = 0$$

$$or \qquad \text{..... (A3)}$$

$$\mathbf{F}_{(u)}{}^{(c)T}\delta\mathbf{u} = -\mathbf{F}_{(\nu)}{}^{(c)T}\delta\mathbf{v}$$

Similarly, Equation (A2) yields

$$\mathbf{\Phi}_u\delta\mathbf{u} = -\mathbf{\Phi}_\nu\delta\mathbf{v} \qquad \text{..... (A4)}$$

If Equation (A3) is combined with the system of equations represented by Equation (A4), the result can be written as:

$$\begin{bmatrix} \mathbf{F}_{(u)}{}^{(c)T} \\ \mathbf{\Phi}_u \end{bmatrix}\delta\mathbf{u} = -\begin{bmatrix} \mathbf{F}_{(\nu)}{}^{(c)T} \\ \mathbf{\Phi}_\nu \end{bmatrix}\delta\mathbf{v} \qquad \text{..... (A5)}$$

The matrix to the left in Equation (A5) is an ((m+1) x m) matrix. Since $\mathbf{\Phi}_u$ is an (m x m) nonsingular matrix, the first row of the ((m+1) x m) matrix, i.e., $\mathbf{F}_{(u)}{}^{(c)T}$, can be expressed as a linear combination of the other rows of the matrix:

$$\mathbf{F}_{(u)}{}^{(c)} = \mathbf{\Phi}_u{}^T\boldsymbol{\lambda} \qquad \text{..... (A6)}$$

where $\boldsymbol{\lambda}$ is an m-vector of multipliers known as Lagrange multipliers. Substitution of Equation (A6) into Equation (A3) yields:

$$\boldsymbol{\lambda}^T \boldsymbol{\Phi}_u \delta\mathbf{u} = -\mathbf{F}_{(\nu)}{}^{(c)T} \delta\mathbf{v}$$

or ……. (A7)

$$\boldsymbol{\lambda}^T \boldsymbol{\Phi}_\nu \delta\mathbf{v} = -\mathbf{F}_{(\nu)}{}^{(c)T} \delta\mathbf{v}$$

where Equation (A4) has been employed.

Vector $\delta\mathbf{v}$ is an arbitrary (independent) vector. The consistency of the constraints for virtual displacements $\delta\mathbf{q}$ is guaranteed by solving Equation (A4) for $\delta\mathbf{u}$ .Since Equation (A7) must hold for any arbitrary $\delta\mathbf{v}$, then:

$$\boldsymbol{\lambda}^T \boldsymbol{\Phi}_\nu = \mathbf{F}_{(\nu)}{}^{(c)T}$$

or ……. (A8)

$$\mathbf{F}_{(\nu)}{}^{(c)} = \boldsymbol{\Phi}_\nu{}^T \boldsymbol{\lambda}$$

Combining equation (A6) with equation (A8) yields:

$$\mathbf{F}^{(c)} = \boldsymbol{\Phi}_q{}^T \boldsymbol{\lambda}$$ ……. (A9)

Equation (A9) expresses the joint reaction forces in terms of the Jacobian matrix $\boldsymbol{\Phi}_q$ of the constraint equations and a vector of Lagrange multipliers $\boldsymbol{\lambda}$.

Appendix B

The Moor-Penrose Generalized Inverse

An inverse of a matrix which is not necessarily square or is square, but nevertheless singular, is applied to solve ill-conditioned problems such as large sized matrix computations. Such an inverse is referred to as a generalized inverse. The most frequently used one is a Moore-Penrose type inverse. Symbolic computation of the generalized inverse is one of the most interesting application areas of Computer Algebra.

The generalized inverse $\mathbf{G}$ of a matrix $\mathbf{A}$ is defined as follows:

$$\begin{cases} \mathbf{A}\,\mathbf{G}\,\mathbf{A} = \mathbf{A} \\ \mathbf{G}\,\mathbf{A}\,\mathbf{G} = \mathbf{G} \\ (\mathbf{A}\,\mathbf{G})^T = \mathbf{A}\,\mathbf{G} \\ (\mathbf{G}\,\mathbf{A})^T = \mathbf{G}\,\mathbf{A} \end{cases} \qquad \text{..... (B1)}$$

where $\mathbf{A}$ and $\mathbf{G}$ are (m x n) and (n x m) matrices with entries in **R**, respectively. If a matrix $\mathbf{G}$ satisfies all four relations, it is called the Moore-Penrose type generalized inverse and is denoted as $\mathbf{A}^+$. The generalized inverse $\mathbf{A}^+$ also satisfies the following relations:

1. $\mathbf{A}^+$ is uniquely determined for $\mathbf{A}$
2. if $\mathbf{A}$ is regular, $\mathbf{A}^+ = \mathbf{A}^{-1}$
3. if $\mathbf{A} = \mathbf{0}$, $\mathbf{A}^+ = \mathbf{0}$
4. if $\mathbf{A}$ is an (m x n) matrix with rank m, then $\mathbf{A}^+ = \mathbf{A}^T \left(\mathbf{A}\,\mathbf{A}^T\right)^{-1}$

We call a Moore-Penrose type generalized inverse simply a "generalized inverse". The generalized inverse is usually computed numerically by a SVD (Singular Value Decomposition) algorithm. However, in this thesis, we only want to apply the 4th relation. Symbolic methods for computing a generalized inverse have been proposed by many authors. Greville's algorithm is one of the best algorithms. For an input matrix $\mathbf{A}$, Greville's algorithm is described as follows:

1. Decompose input (n x m) matrix $\mathbf{A}$ into row vectors $\mathbf{a}_i$

$$\mathbf{A} = \left(\mathbf{a}_1{}^T, \mathbf{a}_2{}^T, ..., \mathbf{a}_n{}^T\right)^T \qquad \text{..... (B2)}$$

2. Let (i x n) matrices $\mathbf{A}_i$ be

$$\mathbf{A}_1 = \mathbf{a}_1, \qquad \mathbf{A}_i = \begin{pmatrix} \mathbf{A}_{i-1} \\ \mathbf{a}_i \end{pmatrix} \qquad \text{..... (B3)}$$

3. For i = 1, 2, …, compute (n x i) matrices $\mathbf{A}_i{}^+$ as

$$\mathbf{A}_i{}^+ = \left(\mathbf{A}_{i-1}{}^+ - \mathbf{b}_i{}^T \mathbf{d}_i \middle| \mathbf{b}_i{}^T\right) \qquad \text{..... (B4)}$$

where

$$\mathbf{d}_i = \mathbf{a}_i \mathbf{A}_{i-1}{}^+$$
$$\mathbf{c}_i = \mathbf{a}_i - \mathbf{d}_i \mathbf{A}_{i-1}$$
$$\mathbf{b}_i = \begin{cases} \dfrac{\mathbf{c}_i}{\mathbf{c}_i \mathbf{c}_i{}^T} & (\mathbf{c}_i \neq \mathbf{0}) \\ \dfrac{\mathbf{d}_i \left(\mathbf{A}_{i-1}{}^+\right)^T}{1 + \mathbf{d}_i \mathbf{d}_i{}^T} & (\mathbf{c}_i = \mathbf{0}) \end{cases}$$
$$\mathbf{A}_1{}^+ = \begin{cases} \dfrac{\mathbf{a}_1{}^T}{\mathbf{a}_1 \mathbf{a}_1{}^T} & (\mathbf{a}_1 \neq \mathbf{0}) \\ \mathbf{a}_1{}^T & (\mathbf{a}_1 = \mathbf{0}) \end{cases}$$

4. After m repetitions $\mathbf{A}_m{}^+$ gives the Moore-Penrose generalized inverse of $\mathbf{A}^+$.

Appendix C

Differential Equations of Motion for Numerical Integration

Lagrange's equations are considered to be a very efficient way to develop the dynamic differential equations of the mechanisms. The compact form of Lagrange's equation is:

$$\frac{d}{dt}\left(\frac{\partial \mathrm{T}}{\partial \dot{\mathbf{q}}_j}\right)-\frac{\partial \mathrm{T}}{\partial \mathbf{q}_j}=\mathbf{Q}_j \qquad (j=1,2,...,k) \qquad \text{..... (C1)}$$

where T is the total kinetic energy of the mechanism; $\mathbf{q}_j$ and $\dot{\mathbf{q}}_j$ are the generalized coordinates and velocities; $\mathbf{Q}_j$ is the generalized force corresponding to the coordinate $\mathbf{q}_j$; and k is the number of the independent generalized coordinates of the system. The unique standard form of the Lagrange's equation holds no matter how complicated the system is. There are as many equations of motion as there are degrees of freedom in the system. In a simple case of a single-degree-of-freedom four-bar mechanism, one can choose the rotation of the driver link as the generalized coordinate.

For a general mechanical system consisting of n rigid bodies, the Lagrange's equations can be written in a more explicit form. The kinetic energy, T, of a mechanism composed of rigid bodies is:

$$T=\sum_{i=1}^{n}\frac{1}{2}m_i\left(\dot{x}_{gi}{}^2+\dot{y}_{gi}{}^2\right)+\sum_{i=1}^{n}\frac{1}{2}I_{gi}\dot{\theta}_i{}^2 \qquad \text{..... (C2)}$$

where $\left(\dot{x}_{gi},\dot{y}_{gi}\right)$ are the coordinates of the center of mass of link i in the mechanism; and n is the number of the moving links. Inserting equation (C2) into equation (C1), we have:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}_j}\right)-\frac{\partial T}{\partial \mathbf{q}_j} = \frac{d}{dt}\sum_{i=1}^{n}\left\{m_i\left(\dot{x}_{gi}\frac{\partial \dot{x}_{gi}}{\partial \dot{\mathbf{q}}_j}+\dot{y}_{gi}\frac{\partial \dot{y}_{gi}}{\partial \dot{\mathbf{q}}_j}\right)+I_{gi}\dot{\theta}_i\frac{\partial \dot{\theta}_i}{\partial \dot{\mathbf{q}}_j}\right\}$$
$$-\sum_{i=1}^{n}\left\{m_i\left(\dot{x}_{gi}\frac{\partial \dot{x}_{gi}}{\partial \mathbf{q}_j}+\dot{y}_{gi}\frac{\partial \dot{y}_{gi}}{\partial \mathbf{q}_j}\right)+I_{gi}\dot{\theta}_i\frac{\partial \dot{\theta}_i}{\partial \mathbf{q}_j}\right\} \quad \text{..... (C3)}$$
$$=\mathbf{Q}_j\ (j=1,2,...,k)$$

Taking advantage of the two basic relations in Lagrangian mechanics,

$$\frac{\partial \dot{\mathbf{r}}_i}{\partial \dot{\mathbf{q}}_j}=\frac{\partial \mathbf{r}_i}{\partial \mathbf{q}_j}$$
$$\frac{d}{dt}\left(\frac{\partial \dot{\mathbf{r}}_i}{\partial \dot{\mathbf{q}}_j}\right)=\frac{d}{dt}\left(\frac{\partial \mathbf{r}_i}{\partial \mathbf{q}_j}\right)=\frac{\partial \dot{\mathbf{r}}_i}{\partial \mathbf{q}_j}$$

where $\mathbf{r}_i$ is an arbitrary function of generalized coordinates, the $(d)/(dt)((\partial T)/(\partial \dot{\mathbf{q}}_j))$ term can be expanded as:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}_j}\right)=\sum_{i=1}^{n}\left\{m_i\left(\ddot{x}_{gi}\frac{\partial \dot{x}_{gi}}{\partial \dot{\mathbf{q}}_j}+\ddot{y}_{gi}\frac{\partial \dot{y}_{gi}}{\partial \dot{\mathbf{q}}_j}\right)+I_{gi}\ddot{\theta}_i\frac{\partial \dot{\theta}_i}{\partial \dot{\mathbf{q}}_j}\right\}$$
$$+\sum_{i=1}^{n}\left\{m_i\left(\dot{x}_{gi}\frac{d}{dt}\left(\frac{\partial \dot{x}_{gi}}{\partial \dot{\mathbf{q}}_j}\right)+\dot{y}_{gi}\frac{d}{dt}\left(\frac{\partial \dot{y}_{gi}}{\partial \dot{\mathbf{q}}_j}\right)\right)+I_{gi}\dot{\theta}_i\frac{d}{dt}\left(\frac{\partial \dot{\theta}_i}{\partial \dot{\mathbf{q}}_j}\right)\right\} \quad \text{..... (C4)}$$
$$=\sum_{i=1}^{n}\left\{m_i\left(\ddot{x}_{gi}\frac{\partial \dot{x}_{gi}}{\partial \dot{\mathbf{q}}_j}+\ddot{y}_{gi}\frac{\partial \dot{y}_{gi}}{\partial \dot{\mathbf{q}}_j}\right)+I_{gi}\ddot{\theta}_i\frac{\partial \dot{\theta}_i}{\partial \dot{\mathbf{q}}_j}\right\}$$
$$+\sum_{i=1}^{n}\left\{m_i\left(\dot{x}_{gi}\frac{\partial \dot{x}_{gi}}{\partial \mathbf{q}_j}+\dot{y}_{gi}\frac{\partial \dot{y}_{gi}}{\partial \mathbf{q}_j}\right)+I_{gi}\dot{\theta}_i\frac{\partial \dot{\theta}_i}{\partial \mathbf{q}_j}\right\}$$

Substituting equation (C4) into (C3), we have the simplified form of Lagrange's equation

$$\sum_{i=1}^{n}\left\{m_i\left(\ddot{x}_{gi}\frac{\partial \dot{x}_{gi}}{\partial \dot{\mathbf{q}}_j}+\ddot{y}_{gi}\frac{\partial \dot{y}_{gi}}{\partial \dot{\mathbf{q}}_j}\right)+I_{gi}\ddot{\theta}_i\frac{\partial \dot{\theta}_i}{\partial \dot{\mathbf{q}}_j}\right\}=\mathbf{Q}_j\ (j=1,2,...,k) \quad \text{..... (C5)}$$

The generalized force $\mathbf{Q}_j$ can be determined by:

$$\mathbf{Q}_j=\sum_{i=1}^{n}\mathbf{F}_i\frac{\partial \mathbf{r}_i}{\partial \mathbf{q}_j} \quad \text{..... (C6)}$$

where $\mathbf{F}_i$ is the external force acting on link i and $\mathbf{r}_i$ is the radius vector of $\mathbf{F}_i$ measured from the origin of the global coordinate system. The generalized external force $\mathbf{Q}_j$ consists of potential forces and non-potential forces, and, therefore, the generalized force can be further expressed as:

$$\mathbf{Q}_j = -\frac{\partial V}{\partial q_j} + \sum_{i=1}^{n} \mathbf{F}_i' \frac{\partial \mathbf{r}_i}{\partial \mathbf{q}_j} \qquad \text{..... (C7)}$$

where V is the potential energy and $\mathbf{F}_i'$ are the non-potential forces. Examples of V are the potential energy of gravity and springs. Examples of $\mathbf{F}_i'$ are driving torque, external load, and friction forces in the mechanical system.

Application of Lagrange's Equation

The application of Lagrange's equations for predicting the dynamic performance of four-bar linkage mechanisms will be illustrated below. The mechanism has a single-degree-of-freedom θ_2 and it is chosen as the independent generalized coordinate.

In order to use equation (C5) for developing the differential equation of motion of the mechanism, we need to express angular position, velocity, and acceleration of links 3 and 4 as a function of the independent quantities $\dot{\theta}_2$ and $\ddot{\theta}_2$. Similar expressions are for mass centers of all moving links. The kinematic analysis equations given in [4] are used here (in a different form) to compute the required positions, velocities, and accelerations.

The angular positions of link 3 and 4 are given by:

$$\begin{Bmatrix} \theta_3 \\ \theta_4 \end{Bmatrix} = \begin{Bmatrix} \sin^{-1}\left[(r_4 \sin\theta_4 - B)/r_3\right] \\ \cos^{-1}\left(C/\sqrt{A^2+B^2}\right) - \tan^{-1}(B/A) \end{Bmatrix} \qquad \text{..... (C8)}$$

where $A = r_1 - r_2\cos\theta_2$, $B = r_2\sin\theta_2$, $C = \dfrac{A^2 + B^2 + r_4 - r_3}{-2r_4}$

The angular velocities of link 3 and 4 are given by:

$$\begin{Bmatrix}\dot{\theta}_3 \\ \dot{\theta}_4\end{Bmatrix} = r_2\dot{\theta}_2\begin{Bmatrix}-\sin(\theta_4-\theta_2)/r_3\sin(\theta_4-\theta_3) \\ \sin(\theta_3-\theta_2)/r_4\sin(\theta_3-\theta_4)\end{Bmatrix} \quad \text{..... (C9)}$$

The angular accelerations of link 3 and 4 are given by:

$$\begin{Bmatrix}\ddot{\theta}_3 \\ \ddot{\theta}_4\end{Bmatrix} = \begin{Bmatrix}\dfrac{-r_2\ddot{\theta}_2\sin(\theta_4-\theta_2)+r_2\dot{\theta}_2^{\,2}\cos(\theta_4-\theta_2)+r_3\dot{\theta}_3^{\,2}\cos(\theta_4-\theta_3)-r_4\dot{\theta}_4^{\,2}}{r_3\sin(\theta_4-\theta_3)} \\ \dfrac{r_2\ddot{\theta}_2\sin(\theta_3-\theta_2)-r_2\dot{\theta}_2^{\,2}\cos(\theta_3-\theta_2)+r_4\dot{\theta}_4^{\,2}\cos(\theta_3-\theta_4)-r_3\dot{\theta}_3^{\,2}}{r_4\sin(\theta_3-\theta_4)}\end{Bmatrix} \quad \text{..... (C10)}$$

The mass center accelerations are given by:

$$\begin{Bmatrix}\ddot{x}_{g2} \\ \ddot{y}_{g2}\end{Bmatrix} = r_{g2}\begin{Bmatrix}-\ddot{\theta}_2\sin\theta_2-\dot{\theta}_2^{\,2}\cos\theta_2 \\ \ddot{\theta}_2\cos\theta_2-\dot{\theta}_2^{\,2}\sin\theta_2\end{Bmatrix} \quad \text{..... (C11)}$$

$$\begin{Bmatrix}\ddot{x}_{g3} \\ \ddot{y}_{g3}\end{Bmatrix} = r_2\begin{Bmatrix}-\ddot{\theta}_2\sin\theta_2-\dot{\theta}_2^{\,2}\cos\theta_2 \\ \ddot{\theta}_2\cos\theta_2-\dot{\theta}_2^{\,2}\sin\theta_2\end{Bmatrix} + r_{g3}\begin{Bmatrix}-\ddot{\theta}_3\sin\theta_3-\dot{\theta}_3^{\,2}\cos\theta_3 \\ \ddot{\theta}_3\cos\theta_3-\dot{\theta}_3^{\,2}\sin\theta_3\end{Bmatrix} \quad \text{..... (C12)}$$

$$\begin{Bmatrix}\ddot{x}_{g4} \\ \ddot{y}_{g4}\end{Bmatrix} = r_{g4}\begin{Bmatrix}-\ddot{\theta}_4\sin\theta_4-\dot{\theta}_4^{\,2}\cos\theta_4 \\ \ddot{\theta}_4\cos\theta_4-\dot{\theta}_4^{\,2}\sin\theta_4\end{Bmatrix} \quad \text{..... (C13)}$$

The partial derivative terms are:

$$\begin{Bmatrix}\dfrac{\partial\dot{\theta}_3}{\partial\dot{\theta}_2} \\ \dfrac{\partial\dot{\theta}_4}{\partial\dot{\theta}_2}\end{Bmatrix} = r_2\begin{Bmatrix}-\sin(\theta_4-\theta_2)/r_3\sin(\theta_4-\theta_3) \\ \sin(\theta_3-\theta_2)/r_4\sin(\theta_3-\theta_4)\end{Bmatrix} \quad \text{..... (C14)}$$

$$\begin{Bmatrix}\dfrac{\partial\dot{x}_{g2}}{\partial\dot{\theta}_2} \\ \dfrac{\partial\dot{y}_{g2}}{\partial\dot{\theta}_2}\end{Bmatrix} = r_{g2}\begin{Bmatrix}-\sin\theta_2 \\ \cos\theta_2\end{Bmatrix} \quad \text{..... (C15)}$$

$$\begin{Bmatrix} \dfrac{\partial \dot{x}_{g3}}{\partial \dot{\theta}_2} \\ \dfrac{\partial \dot{y}_{g3}}{\partial \dot{\theta}_2} \end{Bmatrix} = r_2 \begin{Bmatrix} -\sin\theta_2 \\ \cos\theta_2 \end{Bmatrix} + r_{g3} \frac{\partial \dot{\theta}_3}{\partial \dot{\theta}_2} \begin{Bmatrix} -\sin\theta_3 \\ \cos\theta_3 \end{Bmatrix} \quad \text{..... (C16)}$$

$$\begin{Bmatrix} \dfrac{\partial \dot{x}_{g4}}{\partial \dot{\theta}_2} \\ \dfrac{\partial \dot{y}_{g4}}{\partial \dot{\theta}_2} \end{Bmatrix} = r_{g4} \frac{\partial \dot{\theta}_4}{\partial \dot{\theta}_2} \begin{Bmatrix} -\sin\theta_4 \\ \cos\theta_4 \end{Bmatrix} \quad \text{..... (C17)}$$

Based on equation (C5), the Lagrange's equation of motion of the four-bar linkage mechanism, where θ_2 has been chosen as the generalized coordinate is:

$$\sum_{i=2}^{4} \left\{ m_i \left(\ddot{x}_{gi} \frac{\partial \dot{x}_{gi}}{\partial \dot{\theta}_2} + \ddot{y}_{gi} \frac{\partial \dot{y}_{gi}}{\partial \dot{\theta}_2} \right) + I_{gi} \ddot{\theta}_i \frac{\partial \dot{\theta}_i}{\partial \dot{\theta}_2} \right\} = \mathbf{Q}_2 \quad \text{..... (C18)}$$

where the generalized force Q_2 corresponding to θ_2 is:

$$\mathbf{Q}_2 = -g \sum_{i=2}^{4} m_i \frac{\partial \dot{y}_{gi}}{\partial \dot{\theta}_2} + T_{in} - T_L \frac{\partial \dot{\theta}_4}{\partial \dot{\theta}_2} \quad \text{..... (C19)}$$

where g is the gravitational acceleration.

By substituting equations (C8) – (C17) and equation (C19) in to equation (C18), the dynamic differential equation of motion of the four-bar mechanism is formed. Algebraic manipulation is needed to express the differential equation as an explicit function of generalized coordinate and generalized velocity so that a numerical integration method can be applied. That is, we need to arrange the differential equation in the form:

$$\ddot{\theta}_2 = f\left(\theta_2, \dot{\theta}_2\right) \quad \text{..... (C20)}$$

To do this, first we can write $\ddot{\theta}_2$ as a function of all the position and velocities of the moving links. After some algebraic manipulation, one can write $\ddot{\theta}_2$ as follows:

$$\ddot{\theta}_2 = f\left(\theta_2, \dot{\theta}_2, \theta_3, \dot{\theta}_3, \theta_4, \dot{\theta}_4\right) = \frac{numerator}{deno\min ator} \qquad \text{..... (C21)}$$

$$\begin{aligned} numerator = & -g\left[m_2 r_{g2}\cos\theta_2 + m_3\left(r_2\cos\theta_2 + r_{g3}k\cos\theta_3\right) + m_4\left(r_{g4}p\cos\theta_4\right)\right] \\ & + T_{in} - pT_L - km\left(m_3 {r_{g3}}^2 + I_{g3}\right) - pn\left(m_4 {r_{g4}}^2 + I_{g4}\right) \\ & - r_2 r_{g3}\left[m_3 m\cos(\theta_2 - \theta_3) + m_2\left(\dot{\theta}_3^{\,2} - k\dot{\theta}_2^{\,2}\right)\sin(\theta_2 - \theta_3)\right] \end{aligned}$$

$$\begin{aligned} deno\min ator = & \left\{\left(m_2 {r_{g2}}^2 + I_{g2}\right) + k^2\left(m_3 {r_{g3}}^2 + I_{g3}\right) + m_3\left[{r_2}^2 + 2r_2 r_{g3}k\cos(\theta_2 - \theta_3)\right]\right. \\ & \left. + p^2\left(m_4 {r_{g4}}^2 + I_{g4}\right)\right\} \end{aligned}$$

where $k = -\dfrac{r_2\sin(\theta_4 - \theta_2)}{r_3\sin(\theta_4 - \theta_3)}$, $\quad p = \dfrac{r_2\sin(\theta_3 - \theta_2)}{r_4\sin(\theta_3 - \theta_4)}$

By substituting equations (C8) and (C9) into equation (C21), we can arrive at a form of equation similar to that of equation (C20). Equation (C20) is a second-order non-linear ordinary differential equation. Numerical technique, such as the Runge-Kutta method, is commonly used since a closed-form exact solution does not generally exist.