CHAPTER V

Efficient Simulation of Multilegged Vehicles

5.1 Introduction

In the previous three chapters, the simulation of multiple manipulator systems categorized as Type 2 systems [13] has been discussed. Efficient algorithms, parallelization, and the problems of singularities have all been addressed. In this chapter, dynamic simulation of Type 1 systems is examined. This category of simple multiple chain mechanisms is characterized by a single central body, also called a reference member, which acts as the mobile base for one or more chains. Legged vehicles, such as the Adaptive Suspension Vehicle [12], are the typical example of Type 1 systems. Whereas, Type 2 systems can vary the topology of the system by making and breaking contact between the chains and the reference member, Type 1 systems maintain this connection and make and break contact with the ground at the opposite end of the chains. The purpose of this chapter is to develop an efficient algorithm for the computation of the dynamics for such systems.

One method to accomplish this task was presented by Oh and Orin [4]. It is based on a method presented in [14] for computing the joint-space inertia matrix of a chain. It is somewhat less efficient than Walker and Orin's Composite Rigid Body (CRB) method (also in [14]) and yet has the same $O(N^3)$ complexity for a serial chain. Oh and Orin used this approach to develop a method based on equation augmentation using the constraint force equations arising from the closed kinematic loops. Since the size of this system grows very large and the algorithm has a complexity of $O(m^3N^3)$, it is computationally costly and the approach was abandoned in favor of more efficient methods.

With legged vehicles, however, contacts with the ground may be modeled differently from the hard constraints imposed in a multiple manipulator system. It is more realistic to model ground compliance with spring/damper systems as Shih, Frank, and Ravani did in [5] thus imposing soft constraints. Differences in terrain conditions could then be easily modeled by differences in spring and damper constants or even simple nonlinear equations. Shih, Frank and Ravani's work actually goes one step further by modeling the compliant constraints between adjacent links

so that forces between bodies is a function of the state only. While this simplifies the dynamics by decoupling every rigid body in the system, this approach requires a larger number of system states and can cause numerical problems if joints are to be modeled very stiffly in the constrained directions.

Freeman and Orin develop an improved modeling technique which is a compromise between hard constraints and a completely decoupled system with soft constraints by introducing compliance at the ground contacts only [7]. As a result, the kinematic loops are broken leaving a tree-structured, open chain system. An advantage of this approach is that singularities do not affect open chain computations and are no longer an issue in the simulation of this system. With this Decoupled Tree-Structure (DTS) approach, Freeman and Orin go on to describe a very efficient $O(mN)$ simulation algorithm for legged vehicles that is based on Featherstone's $O(N)$ Articulated Body (AB) algorithm [15].

The results in Chapter 2 showed that an $O(mN^3)$ algorithm using the efficient, open chain algorithms from Appendix B was more efficient than corresponding $O(mN)$ algorithms when $N \leq 21$ for the simulation of multiple, closed chain manipulators. In this chapter, these algorithms are also used as the basis for a new approach to legged vehicle simulation. The resulting algorithm is called the Composite Rigid Body/Decoupled Tree-Structure (CRB/DTS) Method and also has a computational complexity of $O(mN^3)$.

In the next section, the formulation of the dynamic equations of motion for this system are developed. This requires that multiple chain/mobile base algorithms for the computation of the joint space inertia matrix, bias forces (using the inverse dynamics computation), and accelerations be developed. Section 5.3 presents the details of these three parts. Also included in this discussion is the efficient kinematic modeling and extensions of the efficient algorithms from Appendix B for the tree-structure topology of legged vehicles. Then, Section 5.4 details the computational requirements of the resulting $O(mN^3)$ algorithm. For the typical legged vehicle which has three degrees of freedom in each leg, the CRB/DTS Method requires less computation than any of the previous simulation algorithms developed for legged vehicles including the more popular $O(mN)$ approaches using the AB methods.

5.2 Dynamics Formulation

In this section, dynamic equations of motion for legged vehicles are developed based on the CRB approach, and Figure 16 illustrates the model that is used. It contains m serial chains that correspond to the legs of the vehicle which make and break contact with the ground. When in contact spring/damper systems can be used to compute the state-dependent force, $\mathbf{f}_k$, on the last link of each leg. At the other

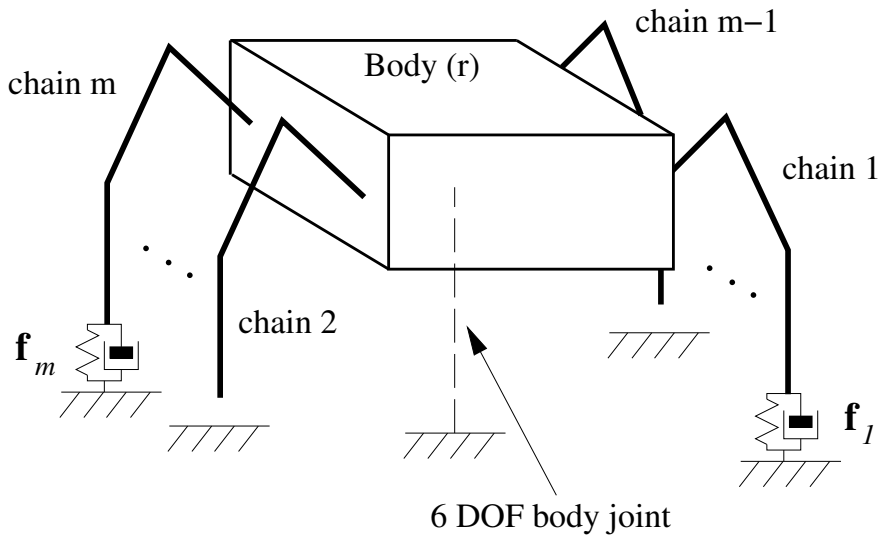

Figure 16: DTS model for a legged vehicle with a body joint.

end of each chain, the first joint of the chain connects the leg to the body of the vehicle which is also referred to as the reference member, r, of the system. Finally, a fictitious (virtual) six degree-of-freedom joint, called the *body joint* connects the reference member to the inertial coordinate frame. If an external control force can be applied to the reference member, not by the legs or gravity but through thrusters attached to the body for example, this force can be thought of as the input force of the virtual body joint. In legged vehicles on land, however, this joint is generally unpowered.

This development begins with the joint space equations of motion for a serial chain with a fixed base and is repeated here for reference:

$$\boldsymbol{\tau}_k = \mathbf{H}_k(\mathbf{q}_k)\ddot{\mathbf{q}}_k + \mathbf{C}_k(\mathbf{q}_k, \dot{\mathbf{q}}_k) + \mathbf{G}_k(\mathbf{q}_k) + \mathbf{J}_k^T(\mathbf{q}_k)\mathbf{f}_k, \tag{5.1}$$

where the various terms have been defined in Chapter 2. This would correspond to the equations of motion for leg k provided the body of the vehicle, acting as the base of the chain, is motionless. When working with legged vehicles, however, the body of the vehicle acts as a mobile base whose motion is unknown. As a result, augmented chains with fixed bases could be formed which contain the links of a leg and an additional link (the body) with a six degree-of-freedom joint (the body joint) connecting it to the inertial coordinate frame.

To dynamically model each one of these augmented chains, additional terms accounting for the base's position (linear and angular), velocity and acceleration

must be added to the basic equation above. First an extra term relating its spatial acceleration, $\mathbf{a}_r$, to leg k's joint torques must be included. For a given state, this relationship is linear and is defined by an $N \times 6$ *coupling inertia* matrix, $\mathbf{P}_k$, which is a function of the joint positions in the leg. Second, the foot contact force, if present, along with coriolis, centrifugal, and gravitational forces in the leg are grouped into a bias force vector, $\mathbf{b}_k$. Unlike, the derivation in Chapter 2, however, this vector is now a function of the body's state (position and velocity) as well as the chain's joint positions and velocities. This leads to the following equations of motion:

$$\boldsymbol{\tau}_k \quad = \quad \mathbf{H}_k \ddot{\mathbf{q}}_k + \mathbf{P}_k \mathbf{a}_r + \mathbf{b}_k. \tag{5.2}$$

Repeated for all m legs in the system, a system of mN equations is obtained containing the $(mN + 6)$ unknown accelerations ($\ddot{\mathbf{q}}_1, \ldots, \ddot{\mathbf{q}}_m$, and $\mathbf{a}_r$).

As in Eq. (2.8) of Chapter 2, a force balance equation for the reference member is required to complete the system of equations. This equates the forces exerted *onto* the body to its acceleration. Written in the body fixed coordinate system, this equation can be expressed as follows:

$$\mathbf{f}_r + \mathbf{f}_r^g - \sum_{k=1}^{m} {}^r\mathbf{f}_k \quad = \quad \mathbf{I}_r \mathbf{a}_r + \bar{\mathbf{f}}_r, \tag{5.3}$$

where $\mathbf{f}_r$ is an external control force (if present) exerted onto the body, and ${}^r\mathbf{f}_k$ is the force exerted *by* the body onto the first link of each leg k. The remaining terms are the same as those used in Eq. (2.8).

Expressions for the body-to-chain forces, ${}^r\mathbf{f}_k$, need to be derived as a function of the unknown accelerations in the system. These forces are linearly related to the accelerations in the system. Therefore, the forces for each leg are decomposed into terms related to the accelerations of its own joints and that of the vehicle body, and an additional force, ${}^r\mathbf{f}'_k$, containing gravity, foot contact force, and chain velocity-dependent terms. This body-to-chain force can be written as follows:

$${}^r\mathbf{f}_k \quad = \quad \mathbf{Q}_k \ddot{\mathbf{q}}_k + {}^r\mathbf{K}_k \mathbf{a}_r + {}^r\mathbf{f}'_k, \tag{5.4}$$

where $\mathbf{Q}_k$ is the transpose of the inertia coupling matrix, $\mathbf{P}_k$ from Eq. (5.2) (verified below), and ${}^r\mathbf{K}_k$ is the spatial CRB inertia for all of the links in leg k and transformed to the body's coordinate system. Note that ${}^r\mathbf{f}_k$ is not directly a function of the motion in other chains. Instead, the effects of other chains are coupled through the reference member acceleration and Eq. (5.3).

Substituting Eq. (5.4) into Eq. (5.3) and rearranging the terms to put it into a form similar to Eq. (5.2) leads to the following equation for the body of the vehicle:

$$\mathbf{f}_r \quad = \quad \sum_{k=1}^{m} \mathbf{Q}_k \ddot{\mathbf{q}}_k + \left(\mathbf{I}_r + \sum_{k=1}^{m} {}^r\mathbf{K}_k \right) \mathbf{a}_r + \left(\mathbf{f}'_r + \sum_{k=1}^{m} {}^r\mathbf{f}'_k \right), \tag{5.5}$$

where $\mathbf{f}'_r$ combines gravity and bias forces associated with the reference member as follows:

$$\mathbf{f}'_r \;=\; \bar{\mathbf{f}}_r - \mathbf{f}^g_r. \tag{5.6}$$

Note that in comparison with Eq. (5.2), the spatial force, $\mathbf{f}_r$, corresponds to $\boldsymbol{\tau}_r$ which can be thought of as the joint forces and moments exerted by the body joint on the reference member. This term could correspond to thruster forces exerted on the body. Since most land-based legged vehicles have no mechanisms to apply forces directly to the body, the "body joint" is said to be unpowered and this term is zero.

Combining Eqs. (5.2) and (5.5) results in the following system of equations describing the motion in the system:

$$\begin{bmatrix} \boldsymbol{\tau}_1 \\ \boldsymbol{\tau}_2 \\ \vdots \\ \boldsymbol{\tau}_m \\ \mathbf{f}_r \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{P}_1 \\ \mathbf{0} & \mathbf{H}_2 & & \mathbf{0} & \mathbf{P}_2 \\ \vdots & & \ddots & & \vdots \\ \mathbf{0} & \mathbf{0} & & \mathbf{H}_m & \mathbf{P}_m \\ \mathbf{Q}_1 & \mathbf{Q}_2 & \cdots & \mathbf{Q}_m & \mathbf{K}_r \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_1 \\ \ddot{\mathbf{q}}_2 \\ \vdots \\ \ddot{\mathbf{q}}_m \\ \mathbf{a}_r \end{bmatrix} + \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_m \\ \mathbf{f}'_r + \sum_{k=1}^{m} {}^r\mathbf{f}'_k \end{bmatrix}, \tag{5.7}$$

where

$$\mathbf{K}_r \;=\; \mathbf{I}_r + \sum_{k=1}^{m} {}^r\mathbf{K}_k,$$

which is the CRB inertia of the entire system defined with respect to the body-fixed coordinate system. The $(mN+6)\times(mN+6)$ block matrix in Eq. (5.7) corresponds to the joint-space inertia matrix for the entire legged vehicle including the fictitious "body joint". Just as the joint space inertia matrices for each leg, $\mathbf{H}_k$, are symmetric and positive definite, so is the entire matrix. This verifies that $\mathbf{Q}_k \;=\; \mathbf{P}_k^T$ and the same inertia coupling matrix is used in both Eqs. (5.2) and (5.4).

5.3 The CRB/DTS Algorithm

In this section, the efficient algorithm to compute the desired accelerations is presented. The approach is the same as the CRB method in [14] where elements of a bias force vector and inertia matrix are computed, and then the resulting system of equations is solved for the accelerations. This approach used serial, open chain algorithms such as those found in Appendix B. Some modifications to these routines must be made in order to compute the needed terms in the simulation of a legged vehicle. First, efficient kinematic modeling for the tree-structured system is presented in Section 5.3.1. Based on this new modeling, the inverse dynamics and

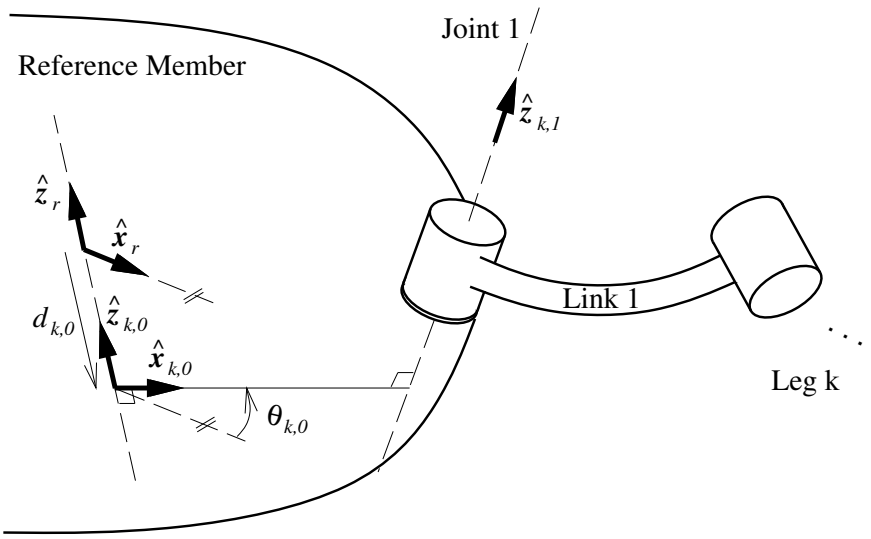

Figure 17: Efficient coordinate system assignments and transformations between the reference member and each chain.

inertia matrix algorithms are then modified in Sections 5.3.2 and 5.3.3, respectively. These routines compute the elements of the $(mN + 6)$-vector of bias forces and the non-zero blocks of the matrix in Eq. (5.7). Finally, an efficient method to solve the resulting system of equations for the accelerations is given in Section 5.3.4.

5.3.1 Kinematic Modeling of Legged Vehicles

The serial chain algorithms in Appendix B assume that the base of the serial chain is fixed, and that its coordinate system can be arbitrarily placed. With this freedom, modified Denavit-Hartenberg (MDH) parameters can be specified between adjacent coordinate systems including the base and the first link. The computational requirements would be lower as a result. With a legged vehicle, the body can be considered the mobile base for each serial chain (leg) in the system. In general, MDH parameters cannot be specified from this single coordinate system to the one attached to the first link of each chain.

The answer to this problem is to specify an additional coordinate system for each chain that is attached to the reference member and acts as its base coordinate system. This is illustrated in Figure 17 for one such leg k. The reference member coordinate system is specified with the r subscript and the base coordinate system for chain k is specified with the $k, 0$ subscripts. However, the placement of the base

coordinate system is not arbitrary, and follows guidelines for kinematic modeling of tree-structured mechanisms that were developed by Khalil and Kleinfinger in [44].

First, the $\hat{\boldsymbol{x}}_{k,0}$ axis is placed along the common normal between $\hat{\boldsymbol{z}}_r$ and $\hat{\boldsymbol{z}}_{k,1}$, and $\hat{\boldsymbol{z}}_{k,0}$ is placed collinear with $\hat{\boldsymbol{z}}_r$ and pointing in the same direction. This is repeated for all of the legs in the system, and results in a simplified transformation between the reference member coordinate system and all of the base coordinate systems. These transformations are planar screws along/about the z-axis and can be specified by two parameters which are the same as the second two used in the MDH notation presented in Appendix A:

$d_{k,0}$ = the perpendicular distance along $\hat{\boldsymbol{z}}_{k,0}$ from $\hat{\boldsymbol{x}}_r$ to $\hat{\boldsymbol{x}}_{k,0}$,

$\theta_{k,0}$ = the angle about $\hat{\boldsymbol{z}}_{k,0}$ from $\hat{\boldsymbol{x}}_r$ to $\hat{\boldsymbol{x}}_{k,0}$.

A full set of MDH parameters are then specified between the base and the first link of each chain.

5.3.2 Inverse Dynamics

To compute the $(mN+6)$ elements of the bias vector, an inverse dynamics algorithm for the tree-structured topology of legged vehicles is executed with the system's accelerations set to zero. The resulting joint torques/forces and reference member force that are computed are the desired elements [see Eq. (5.7)]. The inverse dynamics algorithm for a single chain mechanism from Appendix B is used as the basis for the new algorithm developed in this section.

To begin, the Forward Kinematics recursion is examined. Its purpose is to compute each rigid body's angular velocity, spatial acceleration, and bias force. In the appendix, the angular velocity of the base is set to zero because it is fixed. In this development the base is the mobile reference member, and therefore the angular velocity is specified by the state of the reference member. Due to the addition of separate reference member and chain base coordinates in the previous discussion, the angular velocity of the base is then obtained as follows:

$$\boldsymbol{\omega}_{k,0} = {}^0\boldsymbol{R}_{k,r}\,\boldsymbol{\omega}_r, \tag{5.8}$$

where ${}^0\boldsymbol{R}_{k,r}$ is the matrix representing a z-planar rotation about the angle, $\theta_{k,0}$.

Modification of the acceleration computation is more involved. When the inverse dynamics algorithm is used to compute bias terms, the joint accelerations are set to zero. Correspondingly, the acceleration of the reference member is set to zero. At the same time, the base acceleration is biased by negative gravitational acceleration. This implies that the starting condition for the spatial acceleration recursion in the new algorithm is given as follows:

$$\mathbf{a}_r = \begin{bmatrix} \mathbf{0} \\ -g\ {}^r\boldsymbol{R}_e\,\hat{\mathbf{z}}_e \end{bmatrix}, \tag{5.9}$$

where g is the magnitude of the gravitational acceleration (e.g., 9.81 m/s), ${}^{r}\boldsymbol{R}_{e}$ is the rotation matrix between the inertial and reference member coordinate systems, and $\hat{\mathbf{z}}_{e}$ is the inertial coordinate system's z-axis assumed to be pointing "down." To obtain, the resulting acceleration of the base coordinate system of each chain, the transformation for accelerations shown in Eq. (2.13) is used:

$$\mathbf{a}_{k,0} \;=\; {}^{0}\mathbf{X}_{k,r}\,\mathbf{a}_{r} + \begin{bmatrix} \mathbf{0} \\ {}^{0}\boldsymbol{R}_{k,r}\left[\boldsymbol{\omega}_{r} \times \left(\boldsymbol{\omega}_{r} \times {}^{r}\boldsymbol{p}_{k,0}\right)\right] \end{bmatrix}, \tag{5.10}$$

where ${}^{0}\mathbf{X}_{k,r}$ is the spatial screw transformation defined by $d_{k,0}$ and $\theta_{k,0}$, and ${}^{r}\boldsymbol{p}_{k,0}$ is the position of the base coordinate system with respect to the reference member coordinate system and is given by $[0\ 0\ d_{k,0}]^{T}$.

The final modification to the Forward Kinematics recursion, is the addition of the computation of the gravity and bias forces for the reference member, $\mathbf{f}'_{r}$. It is the same as the computation for a link in the serial chain algorithm and is written as follows:

$$\mathbf{f}'_{r} \;=\; \mathbf{I}_{r}\mathbf{a}_{r} + \begin{bmatrix} \boldsymbol{\omega}_{r} \times \bar{\boldsymbol{I}}_{r}\boldsymbol{\omega}_{r} \\ \boldsymbol{\omega}_{r} \times (\boldsymbol{\omega}_{r} \times \boldsymbol{h}_{r}) \end{bmatrix}, \tag{5.11}$$

where $\bar{\boldsymbol{I}}_{r}$ and $\boldsymbol{h}_{r}$ are the moment of inertia and first moment of mass for the reference member as defined in Chapter 1. Since $\mathbf{a}_{r}$ has been biased by a gravitational acceleration term, the first term on the right hand side of this equation computes the desired gravitational force.

The purpose of the Backward Dynamics recursion, is to compute the spatial force exerted on each link by its inboard neighbor in the chain. The first iteration requires the force exerted by the foot onto the ground, ${}^{N}\mathbf{f}_{k,N+1}$, which can be any function of the state of the system (positions and velocities) as specified by the DTS approach. An example of a compliant foot force computation is given in [7]. These forces are projected onto the joint axis to determine the joint torque. When this algorithm is used as a bias computation, accelerations have been set to zero, the force on the link consists of only state-dependent terms, and the joint torque computed for link i of leg k is $[\mathbf{b}_{k}]_{i}$, the ith element of $\mathbf{b}_{k}$. This computation remains largely unchanged from the appendix for the links of each leg. To compute the desired bias forces exerted by the reference member onto each chain, ${}^{r}\mathbf{f}'_{k}$, the force exerted onto the first link must be transformed to the reference member coordinate system. This involves two spatial transformations as follows:

$${}^{r}\mathbf{f}'_{k} \;=\; {}^{0}\mathbf{X}^{T}_{k,r}\,{}^{1}\mathbf{X}^{T}_{k,0}\mathbf{f}_{k,1}. \tag{5.12}$$

The Bias algorithm can now be written as shown in Table 10. Note that the transformations in the Backward Dynamics recursion have been slightly rearranged.

Table 10: Bias algorithm for the CRB/DTS Method.

Given: ${}^{r}\boldsymbol{R}_{e},\ \boldsymbol{\omega}_{r},\ \dot{\boldsymbol{\omega}}_{r}$.

$$\begin{aligned}
\mathbf{a}_{r} &= \begin{bmatrix} \mathbf{0} \\ -g\ {}^{r}\boldsymbol{R}_{e}\,\hat{\mathbf{z}}_{e} \end{bmatrix} \\
\mathbf{f}'_{r} &= \mathbf{I}_{r}\mathbf{a}_{r} + \begin{bmatrix} \boldsymbol{\omega}_{r} \times \bar{\boldsymbol{I}}_{r}\boldsymbol{\omega}_{r} \\ \boldsymbol{\omega}_{r} \times (\boldsymbol{\omega}_{r} \times \boldsymbol{h}_{r}) \end{bmatrix}
\end{aligned}$$

for all $k = 1, \ldots, m$. Given: ${}^{0}\boldsymbol{R}_{k,r}$.

$$\begin{aligned}
\boldsymbol{\omega}_{k,0} &= {}^{0}\boldsymbol{R}_{k,r}\,\boldsymbol{\omega}_{r} \\
\mathbf{a}_{k,0} &= {}^{0}\mathbf{X}_{k,r}\,\mathbf{a}_{r} + \begin{bmatrix} \mathbf{0} \\ {}^{0}\boldsymbol{R}_{k,r}\left[\boldsymbol{\omega}_{r} \times \left(\boldsymbol{\omega}_{r} \times {}^{r}\boldsymbol{p}_{k,0}\right)\right] \end{bmatrix}
\end{aligned}$$

for $i = 1, \ldots, N$. Given: ${}^{i}\boldsymbol{R}_{k,i-1}$, ${}^{i-1}\boldsymbol{p}_{k,i}$, $\dot{q}_{k,i}$.

$$\begin{aligned}
\boldsymbol{\omega}_{k,i} &= {}^{i}\boldsymbol{R}_{k,i-1}\boldsymbol{\omega}_{k,i-1} + \sigma_{k,i}\dot{q}_{k,i}\hat{\mathbf{z}}_{k,i} \\
\mathbf{a}_{k,i} &= {}^{i}\mathbf{X}_{k,i-1}\mathbf{a}_{k,i-1} + \begin{bmatrix} \mathbf{0} \\ {}^{i}\boldsymbol{R}_{k,i-1}\left[\boldsymbol{\omega}_{k,i-1} \times \left(\boldsymbol{\omega}_{k,i-1} \times {}^{i-1}\boldsymbol{p}_{k,i}\right)\right] \end{bmatrix} \\
&\qquad + \begin{bmatrix} \sigma_{k,i}\left(\boldsymbol{\omega}_{k,i} \times \dot{q}_{k,i}\hat{\mathbf{z}}_{k,i}\right) \\ \bar{\sigma}_{k,i}\left(2\boldsymbol{\omega}_{k,i} \times \dot{q}_{k,i}\hat{\mathbf{z}}_{k,i}\right) \end{bmatrix} \\
\mathbf{f}'_{k,i} &= \mathbf{I}_{k,i}\mathbf{a}_{k,i} + \begin{bmatrix} \boldsymbol{\omega}_{k,i} \times \bar{\boldsymbol{I}}_{k,i}\boldsymbol{\omega}_{k,i} \\ \boldsymbol{\omega}_{k,i} \times (\boldsymbol{\omega}_{k,i} \times \boldsymbol{h}_{k,i}) \end{bmatrix}
\end{aligned}$$

end for i.

for $i = N, \ldots, 1$. Given: ${}^{N}\mathbf{f}_{k,N+1}$ (external tip force).

$$\begin{aligned}
\mathbf{f}_{k,i} &= \mathbf{f}'_{k,i} + {}^{i}\mathbf{f}_{k,i+1} \\
\left[\mathbf{b}_{k}\right]_{i} &= \boldsymbol{\phi}_{k,i}^{T}\mathbf{f}_{k,i} \\
{}^{i-1}\mathbf{f}_{k,i} &= {}^{i}\mathbf{X}_{k,i-1}^{T}\mathbf{f}_{k,i}
\end{aligned}$$

end for i.

$$ {}^{r}\mathbf{f}'_{k} = {}^{0}\mathbf{X}_{k,r}^{T}\,{}^{0}\mathbf{f}_{k,1} $$

end for all k.

5.3.3 Inertia Matrix

In this section, the algorithm to compute the non-zero blocks of the inertia matrix in Eq. (5.7) is developed. The single chain algorithm in Table 27 of Appendix B is used as the basis for this development. To begin, a physical interpretation is given to the algorithm in Table 27. Then, appropriate extensions to accommodate the topology of legged vehicles are added to the algorithm.

The first equation of Table 27 computes the ith CRB inertia, $\mathbf{K}_i$, which is the spatial inertia of links i through N assuming the joints are locked in the position specified by the current state. A unit acceleration is applied to joint i and the resulting force on link i is given by:

$$\mathbf{f}_i = \mathbf{K}_i \boldsymbol{\phi}_i. \tag{5.13}$$

Projecting this force onto joint i yields element (i,i) of the joint space inertia matrix for the chain:

$$[\mathbf{H}]_{i,i} = \boldsymbol{\phi}_i^T \mathbf{f}_i. \tag{5.14}$$

This spatial force is transformed to the inboard links and projected to their joint axes to obtain the off-diagonal elements of $\mathbf{H}$ as follows:

$$^j\mathbf{f}_i = {}^{j+1}\mathbf{X}_j^T \, {}^{j+1}\mathbf{f}_i \tag{5.15}$$

$$[\mathbf{H}]_{i,j} = \boldsymbol{\phi}_j^{Tj}\mathbf{f}_i. \tag{5.16}$$

This step ends at $j = 1$ for a single serial chain with a fixed base. This computation is repeated for each serial chain in the legged vehicle, and is sufficient to produce the elements of the desired $\mathbf{H}_k$ matrices.

To obtain the ith column of the inertia coupling matrix, $\mathbf{Q}_k$, the force on link 1 of leg k as computed above ($^1\mathbf{f}_{k,i}$ for leg k), must also be projected onto the body joint. This is accomplished with two spatial force transformations to the reference member coordinate system:

$$^r\mathbf{f}_{k,i} = {}^0\mathbf{X}_{k,r}^T \, {}^1\mathbf{X}_{k,0}^T \, {}^1\mathbf{f}_{k,i}, \tag{5.17}$$

and a projection onto the body joint as follows:

$$[\mathbf{Q}_k]_i = \boldsymbol{\phi}_r \, {}^r\mathbf{f}_{k,i}. \tag{5.18}$$

Since the body joint has six degrees of freedom, $\boldsymbol{\phi}_r$ is actually the 6×6 identity matrix; therefore, the ith column of $\mathbf{Q}_k$ is $^r\mathbf{f}_{k,i}$.

The last matrix to be computed is the CRB inertia of the entire system, $\mathbf{K}_r$. The serial chain algorithm computes, $\mathbf{K}_{k,1}$, the CRB inertia of links 1 through N for leg k. First, these must be expressed with respect to a common coordinate system, the

reference member coordinate system. Two successive congruence transformations are required as follows:

$$ {}^{r}\mathbf{K}_k \;=\; {}^{0}\mathbf{X}_{k,r}^{T}\left({}^{1}\mathbf{X}_{k,0}^{T}\mathbf{K}_{k,1}\,{}^{1}\mathbf{X}_{k,0}\right){}^{0}\mathbf{X}_{k,r}. \tag{5.19} $$

The CRB inertia for the system, $\mathbf{K}_r$, is the sum of these and the spatial inertia of the reference member, $\mathbf{I}_r$. Adding these steps to the algorithm in Table 27 and rearranging some of the transformation operations, the desired algorithm is obtained as shown in Table 11.

5.3.4 Computation of Accelerations

With the results from the previous two sections and the simulation's input, joint torques and forces, Eq. (5.7) becomes an $(mN+6)\times(mN+6)$ linear system of equations with joint and reference member accelerations as unknown quantities. The accelerations could be determined by employing a standard linear system solver for Eq. (5.7) which results in an $\mathrm{O}(m^3N^3)$ algorithm as was done in [4]. In this section, the structure of the system matrix is taken into account to develop an alternative method and significantly reduce the complexity and the computational requirements of this step.

First Eq. (5.7) is rewritten noting that $\mathbf{P}_k = \mathbf{Q}_k^T$:

$$ \begin{bmatrix} \mathbf{H}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{Q}_1^T \\ \mathbf{0} & \mathbf{H}_2 & & \mathbf{0} & \mathbf{Q}_2^T \\ \vdots & & \ddots & & \vdots \\ \mathbf{0} & \mathbf{0} & & \mathbf{H}_m & \mathbf{Q}_m^T \\ \mathbf{Q}_1 & \mathbf{Q}_2 & \cdots & \mathbf{Q}_m & \mathbf{K}_r \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_1 \\ \ddot{\mathbf{q}}_2 \\ \vdots \\ \ddot{\mathbf{q}}_m \\ \mathbf{a}_r \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_1 - \mathbf{b}_1 \\ \boldsymbol{\tau}_2 - \mathbf{b}_2 \\ \vdots \\ \boldsymbol{\tau}_m - \mathbf{b}_m \\ \mathbf{f}_r - \mathbf{f}'_r - \sum_{k=1}^{m} {}^{r}\mathbf{f}'_k \end{bmatrix}. \tag{5.20} $$

Upon examination of this equation, the sparse structure of the matrix is found to be the same as the one that resulted from modeling the dynamics for multiple manipulator systems using an operational space formulation in Eq. (4.3) of the previous chapter. In that problem, block elimination techniques were developed to achieve more efficient and robust methods for solving the system of equations. The same technique can be applied to this problem to achieve substantial improvements over a standard linear system solution method.

The first step in this technique is to invert the joint space inertia matrices of the chains, $\mathbf{H}_k$, and use them to block-eliminate the coupling inertia matrices $\mathbf{Q}_k$ along the bottom row. This is accomplished by subtracting from the bottom row, $\mathbf{Q}_k\mathbf{H}_k^{-1}$ multiplied times row k, for all $k = 1,\ldots,m$. The resulting system of equations

Table 11: Inertia matrix algorithm for the CRB/DTS Method.

for all $k = 1, \ldots, m$. Given: ${}^{N}\mathbf{K}_{k,N+1} = \mathbf{0}$.

for $i = N, \ldots, 1$.

$$
\begin{aligned}
\mathbf{K}_{k,i} &= \mathbf{I}_{k,i} + {}^{i}\mathbf{K}_{k,i+1} \\
\mathbf{f}_{k,i} &= \mathbf{K}_{k,i}\boldsymbol{\phi}_{k,i} \\
[\mathbf{H}_k]_{i,i} &= \boldsymbol{\phi}_{k,i}^{T}\mathbf{f}_{k,i} \\
{}^{i-1}\mathbf{K}_{k,i} &= {}^{i}\mathbf{X}_{k,i-1}^{T}\mathbf{K}_{k,i}\,{}^{i}\mathbf{X}_{k,i-1} \\
{}^{i-1}\mathbf{f}_{k,i} &= {}^{i}\mathbf{X}_{k,i-1}^{T}\mathbf{f}_{k,i}
\end{aligned}
$$

for $j = i - 1, \ldots, 1$.

$$
\begin{aligned}
[\mathbf{H}_k]_{i,j} &= \boldsymbol{\phi}_{k,j}^{T}\,{}^{j}\mathbf{f}_{k,i} \\
{}^{j-1}\mathbf{f}_{k,i} &= {}^{j}\mathbf{X}_{k,j-1}^{T}\,{}^{j}\mathbf{f}_{k,i}
\end{aligned}
$$

end for j.

$$
[\mathbf{Q}_k]_i = {}^{0}\mathbf{X}_{k,r}^{T}\,{}^{0}\mathbf{f}_{k,i}
$$

end for i.

$$
{}^{r}\mathbf{K}_k = {}^{0}\mathbf{X}_{k,r}^{T}\,{}^{0}\mathbf{K}_{k,1}\,{}^{0}\mathbf{X}_{k,r}
$$

end for all k.

$$
\mathbf{K}_r = \mathbf{I}_r + \sum_{k=1}^{m} {}^{r}\mathbf{K}_k
$$

appears as follows:

$$\begin{bmatrix} \mathbf{H}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{Q}_1^T \\ \mathbf{0} & \mathbf{H}_2 & & \mathbf{0} & \mathbf{Q}_2^T \\ \vdots & & \ddots & & \vdots \\ \mathbf{0} & \mathbf{0} & & \mathbf{H}_m & \mathbf{Q}_m^T \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{K}_r^* \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_1 \\ \ddot{\mathbf{q}}_2 \\ \vdots \\ \ddot{\mathbf{q}}_m \\ \mathbf{a}_r \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_1 - \mathbf{b}_1 \\ \boldsymbol{\tau}_2 - \mathbf{b}_2 \\ \vdots \\ \boldsymbol{\tau}_m - \mathbf{b}_m \\ \mathbf{f}_r^* \end{bmatrix}, \tag{5.21}$$

where

$$\mathbf{K}_r^* = \mathbf{K}_r - \sum_{k=1}^{m} \mathbf{Q}_k \mathbf{H}_k^{-1} \mathbf{Q}_k^T, \tag{5.22}$$

$$\mathbf{f}_r^* = \mathbf{f}_r - \mathbf{f}_r' - \sum_{k=1}^{m} \left[{}^r\mathbf{f}_k' + \mathbf{Q}_k \mathbf{H}_k^{-1} \left(\boldsymbol{\tau}_k - \mathbf{b}_k \right) \right]. \tag{5.23}$$

In Chapter 4, this step involved the inversion of matrices that could become singular. In this development, however, the joint space inertia matrices, $\mathbf{H}_k$, are always nonsingular so the inverses always exist.

The accelerations in Eq. (5.21) can now be determined. The acceleration of the reference member can be computed as follows:

$$\mathbf{a}_r = \left[\mathbf{K}_r^* \right]^{-1} \mathbf{f}_r^*, \tag{5.24}$$

where $\mathbf{K}_r^*$ is invertible because the original matrix in Eq. (5.7) is positive definite which implies that the matrices along the diagonal of the reduced system in Eq. (5.21) are nonsingular. With $\mathbf{a}_r$ known, back substitution in Eq. (5.21) decouples the rest of the system of unknowns, and the chain's joint accelerations are computed as follows:

$$\ddot{\mathbf{q}}_k = \mathbf{H}_k^{-1} \left[\left(\boldsymbol{\tau}_k - \mathbf{b}_k \right) - \mathbf{Q}_k^T \mathbf{a}_r \right], \qquad \text{for all } k = 1, \ldots, m, \tag{5.25}$$

which completes the computation of the dynamics. Again, a numerical integration routine is needed in the simulation of the system in order to update the system's state based on these new accelerations.

5.4 Computational Requirements

The final task of this chapter is to determine the computational requirements of the algorithm described in the previous sections. For discussions of the inverse dynamics and inertia matrix algorithms, the additions to the algorithms from Appendix B are discussed in detail while minor changes made to the steps in the algorithms taken from the appendix are summarized. Also, the computational requirements will focus on legs with revolute joints. A similar analysis may be made for other types of joints.

Inverse Dynamics

This discussion begins with the inverse dynamics routine. Assuming the orientation of the reference member is specified with Z-Y-X Euler angles, the computation of the gravitational acceleration for $\mathbf{a}_r$ requires [4M, 0A] as was found in Chapter 2. The computational requirements for $\mathbf{f}'_r$ depend on where the reference member's coordinate system is placed. In this case, it is placed coincident with its principal axes, the inertia matrices, $\bar{\boldsymbol{I}}_r$ and $\mathbf{I}_r$, are diagonal and the first moment of mass, $\boldsymbol{h}_r$ is zero. Since $\dot{\boldsymbol{\omega}}_r$ is also zero, this step requires [12M, 3A].

To transform kinematic quantities from the reference member to the base coordinate system for each leg, translation along and rotation about the z-axis is required. With $\boldsymbol{\omega}_r$ given, the z-planar rotation to obtain $\boldsymbol{\omega}_{k,0}$ costs [4M, 2A] per leg as discussed in Appendix A. Computation of $\mathbf{a}_{k,0}$ can be reduced to the computation of the linear acceleration portion since angular acceleration is zero. This computation simplifies to:

$$\boldsymbol{a}_{k,0} \;=\; {}^{0}\boldsymbol{R}_{k,r}\left[\boldsymbol{a}_r + \boldsymbol{\omega}_r \times \left(\boldsymbol{\omega}_r \times {}^{r}\boldsymbol{p}_{k,0}\right)\right]. \tag{5.26}$$

Since only the third element of ${}^{r}\boldsymbol{p}_{k,0}$ is non-zero, this step requires [10M, 6A] per leg. In this step, $\boldsymbol{\Omega}_{k,0}$ as defined in the appendix, must also be computed. Since the angular velocity used in this computation is zero, only $\tilde{\boldsymbol{\omega}}_{k,0}\tilde{\boldsymbol{\omega}}_{k,0}$ needs to be computed which requires an additional [6M, 3A] per leg.

Once these terms have been computed, the bias computation for each leg in the system can be performed. This involves the equations in the nested loops with the i index, and are much the same as the ones presented in Appendix B but with the joint accelerations set to zero. In this case with a moving base, many of the vectors that were zero in the appendix are now generally non-zero, with the exception of the angular acceleration at the base coordinate system of each leg. This causes the computational requirements to jump from $[(94N - 114)\text{M}, (81N - 105)\text{A}]$ to $[(94N - 10)\text{M}, (81N - 10)\text{A}]$ for a leg with N revolute joints.

Finally, the force on the first link of the leg must be transformed to the reference member's coordinate system. The spatial transformation to the leg's base coordinate system, involving a full set of MDH parameters, is included in the inverse dynamics requirements above. So only the z-planar screw is necessary from the base to the reference member system which requires [10M, 6A] per leg. A summary of these computational requirements is given in the first part of Table 12 at the end of this section.

Inertia Matrix

As with the inverse dynamics computation above, a significant amount of the computational requirements for the inertia matrix computation can be ascertained from

the algorithm described in Appendix B. In this case, many of the reductions in the number of operations due to unneeded elements of the quantities for the first link cannot be applied. Additional transformations of $\mathbf{K}_{k,1}$ and ${}^{j}\mathbf{f}_{k,1}$ to the leg's base coordinate system have been added to the algorithm, which results in a computational requirement of $[(10N^2 + 39N)\text{M}, (6N^2 + 51N - 9)\text{A}]$ per leg.

The additional computation to find each column of $\mathbf{Q}_k$ requires a z-planar screw transformation requiring [10M, 6A]. This constant planar congruence transformation of the spatial CRB inertia matrix, ${}^{0}\mathbf{K}_{k,1}$ can be shown to require only [15M, 18A] per leg by transforming the moment of inertia matrix and first moment of mass vector separately and using some results from Appendix A. The summation of $m+1$ spatial inertia matrices to obtain $\mathbf{K}_r$ requires only $[0\text{M}, (9m-9)\text{A}]$. This addition involves six unique elements from the upper triangular portions of the 3×3 moment of inertia matrices and the three elements of each first moment of mass vector for each chain and the reference member. Since the reference member coordinate system is placed at the principal axes, $\bar{\boldsymbol{I}}_r$ is diagonal and $\boldsymbol{h}_r$ is zero, nine fewer additions are needed for these quantities. The composite mass of the entire system remains constant and can be computed off-line as was done in Appendix B. A summary of the overall cost of the computations to obtain the inertia matrix are listed in the second part of Table 12.

Accelerations

The most efficient method of computing $\mathbf{K}_r^*$ and $\mathbf{f}_r^*$ does not require the inverse of $\mathbf{H}_k$. As discussed in the previous chapter, a root-free Cholesky decomposition of the matrix is computed requiring $[(\frac{1}{6}N^3 + \frac{1}{2}N^2 - \frac{2}{3}N)\text{M}, (\frac{1}{6}N^3 - \frac{1}{6}N)\text{A}]$. The six columns of $\mathbf{H}_k^{-1}\mathbf{Q}_k^T$ can then be determined from back substitutions with each of the six columns of $\mathbf{Q}_k^T$, requiring $[N^2\text{M}, (N^2-N)\text{A}]$ per column [37]. The result is used in matrix-matrix multiplications to compute $\mathbf{K}_r^*$. Its transpose is equal to $\mathbf{Q}_k\mathbf{H}_k^{-1}$ and is also used in matrix-vector multiplications to compute $\mathbf{f}_r^*$.

The acceleration of the reference member can be computed as follows:

$$\mathbf{a}_r \;=\; [\mathbf{K}_r^*]^{-1}\,\mathbf{f}_r^*. \tag{5.27}$$

Since the original matrix in Eq. (5.7) is positive definite, the matrices along the diagonal of the reduced system in Eq. (5.21) are also positive definite. Therefore, root-free Cholesky decomposition and a back substitution step is also the most efficient method of determining $\mathbf{a}_r$. Since the matrix is 6×6, the cost is [86M, 65A]. The chain's joint accelerations are computed as follows:

$$\ddot{\mathbf{q}}_k \;=\; \mathbf{H}_k^{-1}\left[(\boldsymbol{\tau}_k - \mathbf{b}_k) - \mathbf{Q}_k^T\mathbf{a}_r\right], \qquad \text{for all } k = 1,\ldots,m. \tag{5.28}$$

Since the Cholesky decomposition of $\mathbf{H}_k$ is determined for the block elimination step, this step requires only matrix-vector multiplies, a vector addition, and a back

substitution step. The cost is $[(N^2+6N)\text{M}, (N^2+5N)\text{A}]$. A summary of these steps are listed in the third part of the table.

The last line of Table 12, lists the cost of computing the dynamics of a system with six chains containing three revolute degrees of freedom each at [4440M, 3860A]. This corresponds to a number of experimental legged vehicles including the OSU Hexapod. The alternative algorithm is the DTS algorithm based on the AB (AB/DTS) method that was developed by Freeman and Orin [7]. Part II of this dissertation presents this algorithm in much more detail for the development of underwater simulation. An efficient implementation for land-based vehicles is shown to cost $[(224mN+59m+99)\text{M}, (205mN+66m+65)\text{A}]$. For the vehicles under consideration, this corresponds to [4485M, 4151A] and higher than the method developed in this chapter. For chains with more than three degrees of freedom, however, the AB method becomes more efficient.

5.5 Summary and Conclusions

In this chapter, the simulation of a multilegged vehicle is discussed. It is different from the multiple manipulator problem discussed in previous chapters primarily because the tips of the chains in the vehicle system contact the environment in a compliant way. With appropriate compliant foot contact modeling as in the Decoupled Tree-Structure (DTS) approach described by Freeman and Orin in [7], closed kinematic loops are removed from the legged vehicle system and the dynamics algorithm is simplified. Most notably the problem of singularities disappears.

The Composite-Rigid-Body (CRB) method is used as the basis for the dynamics algorithm developed in this chapter. The resulting CRB/DTS algorithm has a computational complexity of $\text{O}(mN^3)$. Since legged vehicles typically have few degrees of freedom in each chain, this method proves to be more computationally efficient than Articulated Body (AB)-based methods with a complexity of $\text{O}(mN)$. Specifically, computation of the dynamics for the OSU Hexapod requires [4440M, 3944A] which is somewhat better than the efficient implementation of the AB/DTS method. Another advantage to using the approach in this chapter is the ability to use previous research to parallelize the computation of the chain's joint space inertia matrices [38, 19, 45] to further increase computational rates.

Also note that the structure of the computation is the same as the multiple manipulator algorithm examined in the previous three chapters. The bias quantities, $\mathbf{b}_k$ and ${}^r\mathbf{f}'_k$; inertia matrices, $\mathbf{H}_k$, $\mathbf{Q}_k$ and ${}^r\mathbf{K}_k$; and some of the intermediate computations for the reference member acceleration can all be performed independently for each chain. Then after the reference member acceleration is computed, the computation of the joint accelerations can also be performed independently. This is the

Table 12: Computational requirements for the CRB/DTS Method.

	$\times$	$+$
$\mathbf{a}_r$	4	0
$\mathbf{f}'_r$	12	3
$\boldsymbol{\omega}_{k,0}$	$4m$	$2m$
$\mathbf{a}_{k,0}$	$16m$	$9m$
Inverse dynamics: (bias computation)		
$\boldsymbol{\omega}_{k,i}$	$8mN$	$5mN$
$\mathbf{a}_{k,i}$	$m(33N-10)$	$m(28N-10)$
$\mathbf{f}'_{k,i}$	$33mN$	$30mN$
$\mathbf{f}_{k,i}$	—	$6mN$
$[\mathbf{b}_k]_i$	—	—
${}^{i-1}\mathbf{f}_{k,i}$	$20mN$	$12mN$
${}^r\mathbf{f}'_k$	$10m$	$6m$
Inertia matrix:		
$\mathbf{K}_{k,i}$	—	$m(9N-9)$
$\mathbf{f}_{k,i}$	—	—
$[\mathbf{H}_k]_{i,i}$	—	—
${}^{i-1}\mathbf{K}_{k,i}$	$32mN$	$35mN$
${}^{i-1}\mathbf{f}_{k,i}$	$17mN$	$9mN$
$[\mathbf{H}_k]_{i,j}$	—	—
${}^{j-1}\mathbf{f}_{k,i}$	$m(10N^2-10N)$	$m(6N^2-6N)$
$[\mathbf{Q}_k]_i$	$10mN$	$6mN$
${}^r\mathbf{K}_k$	$15m$	$16m$
$\mathbf{K}_r$	—	$9m-6$
$\mathbf{K}^*_r$	$m(\frac{1}{6}N^3+6\frac{1}{2}N^2+20\frac{1}{3}N)$	$m(\frac{1}{6}N^3+6N^2+14\frac{5}{6}N)-6$
$\mathbf{f}^*_r$	$6mN$	$7mN+6m+6$
$\mathbf{a}_r$	86	65
$\ddot{\mathbf{q}}_k$	$m(N^2+6N)$	$m(N^2+5N)$
Total:	$m(\frac{1}{6}N^3+17\frac{1}{2}N^2+$ $175\frac{1}{3}N+35)+102$	$m(\frac{1}{6}N^3+13N^2+$ $160\frac{5}{6}N+29)+62$
$m=6,\ N=3$	4440	3860

same as the OCD/RMA/CCD structure discussed in Chapter 3. Hence, parallelization of this algorithm can be implemented in much the same way. Parallelization of the inertia matrix computation, as mentioned in the previous paragraph, would introduce a third dimension of parallelism which occurs *within* the spatial level.