Part I

Multiple Chain Dynamic Simulation on Land

CHAPTER II

Dynamics of Multiple Manipulator Systems: An Operational Space Approach

2.1 Introduction

In this and the following two chapters, the problem of simulating multiple manipulators that cooperate to lift large loads is addressed. Each manipulator rigidly grasps a common object, resulting in closed kinematic chains which requires special consideration in the development of the dynamic equations for this system. Since real-time and even super-real-time simulation is desirable, computationally efficient dynamics algorithms are a high priority. This chapter addresses this facet of the problem specifically.

While the problem of efficient dynamics algorithms for single, open-chain mechanisms have been extensively studied [14, 15, 16, 17, 28, 29], much less has been published on simulation algorithms for systems with closed chains. In [4], Oh and Orin extended an $O(N^3)$ approach in [14] to develop a dynamic simulation algorithm based on equation augmentation using the constraint forces applied to the reference member resulting from the closed kinematic loops. However, this results in a large system of equations to be solved, with an $O(m^3N^3)$ complexity where m is the number of chains in the system, and N is the number of degrees of freedom in each chain.

The $O(N)$ approach to dynamic simulation of single chains developed by Featherstone [15] has also been extended to the simulation of multiple chain mechanisms. Rodriguez, Jain, and Kreutz-Delgado [30] develop mathematical tools called linear operators to derive simulation algorithms for general robotic systems. This was used in [31, 32] to develop the dynamic equations for multiple manipulator systems grasping a common object on fixed and mobile bases respectively. The resulting algorithms have a computational complexity of $O(mN)$ when the manipulators are in nonsingular configurations.

None of this work, however, has addressed the issue of detailed computational requirements which is important in obtaining efficient algorithms. To this end, Lilly

and Orin have studied this issue for single and multiple closed-chain mechanisms [24, 33]. The efficient algorithms developed in their work have computational complexities $O(mN^3)$ and $O(mN)$, respectively, but the former requires fewer operations for $N < 21$. The algorithm developed in this chapter builds on these results and presents a new and more efficient algorithm for the simulation of these systems. This also provides a background for key results in later chapters of this Part. Parallelization of this algorithm is examined in Chapter 3 in order to achieve higher, possibly real-time, computational rates. Problems arising from singular manipulators are studied in Chapter 4. Finally in Chapter 5, efficient multiple chain algorithms are also developed specifically for multilegged vehicles that also have a computational complexity of $O(mN^3)$.

In the following section, the dynamic equations for a general multiple manipulator system are developed with arbitrarily constrained grasps on the reference member. The operational space [34] formulation developed by Lilly and Orin in [24] is used which simplifies the conceptual framework for the development of the algorithm. In Section 2.3, it is refined assuming rigid grasps on the reference member. The result is a very efficient algorithm for the simulation of multiple manipulator systems under consideration. Finally, the efficient implementation and resulting computational requirements for this algorithm are presented in Section 2.4. The system used for this analysis has four PUMA 560 manipulators each with six revolute joints.

2.2 Multiple Closed-Chain Dynamics

In this section, the equations for efficient dynamic simulation of a multiple, closed-chain manipulator system are developed. Figure 2 illustrates the system that is used as the model in this chapter. It contains m manipulators, each with N degrees of freedom, that are grasping a common load, called the reference member. The number of degrees of freedom in each chain is assumed to be the same for all chains for simplicity and without loss of generality. In this development, grasps with arbitrary motion constraints are assumed and can be modeled by an augmented chain that has an additional unpowered joint at the point of contact, called the General Contact Joint, and an extra link fixed to the reference member. This joint can have more than one degree of freedom, and can be simulated efficiently using the general joint model developed in [24, 35]. The extra link at the end of the augmented chain is fictitious as shown by the dotted lines in the figure.

To begin, each manipulator is regarded as a simple, closed chain where the motion of its base is completely known. Numbering the manipulators in the system 1 through m, each is governed by the dynamic equations of motion which can be

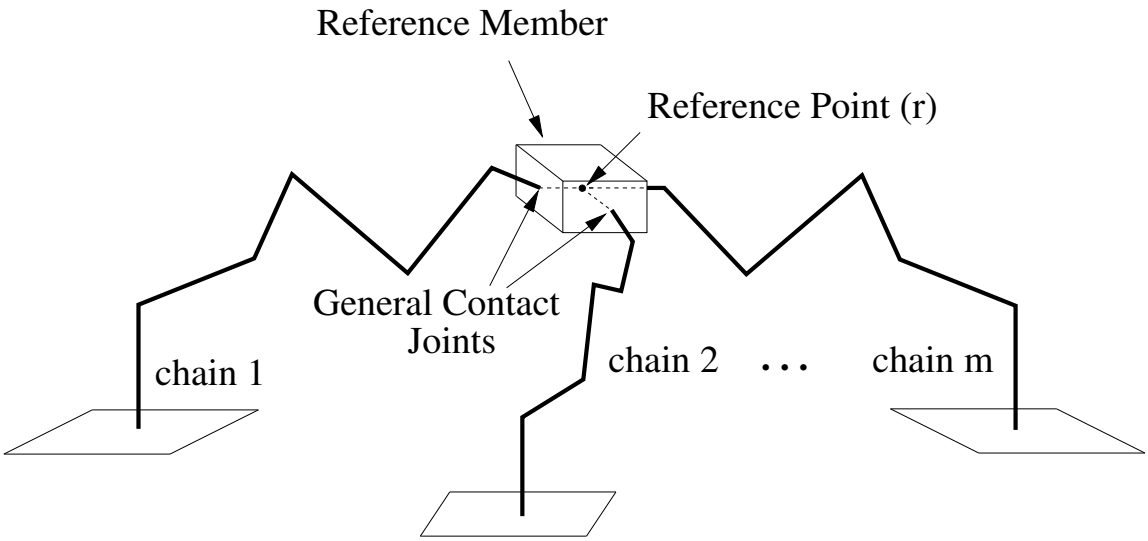

Figure 2: The multiple manipulator system grasping a common object.

written as follows:

$$\boldsymbol{\tau}_k = \mathbf{H}_k(\mathbf{q}_k)\ddot{\mathbf{q}}_k + \mathbf{C}_k(\mathbf{q}_k, \dot{\mathbf{q}}_k) + \mathbf{G}_k(\mathbf{q}_k) + \mathbf{J}_k^T(\mathbf{q}_k)\mathbf{f}_k, \tag{2.1}$$

where the k index refers to the manipulator number,

$\boldsymbol{\tau}_k$ $N \times 1$ vector of torques/forces of the joint actuators,

$\mathbf{q}_k, \dot{\mathbf{q}}_k, \ddot{\mathbf{q}}_k$ $N \times 1$ vectors of joint positions, rates, and accelerations,

$\mathbf{H}_k$ $N \times N$ positive definite joint space inertia matrix,

$\mathbf{C}_k$ $N \times 1$ vector specifying centrifugal and coriolis effects,

$\mathbf{G}_k$ $N \times 1$ vector specifying the effects due to gravity,

$\mathbf{J}_k$ $6 \times N$ Jacobian matrix, and

$\mathbf{f}_k$ 6×1 force exerted by tip of chain k on the reference member.

Since we are interested in the solution to the Forward Dynamics or simulation problem, the state of the system, consisting of joint positions and rates, and the input joint torques/forces are given. The joint accelerations and the tip forces for each chain must then be computed.

By grouping some terms from Eq. (2.1) and solving for the joint accelerations, we obtain the following equation for each chain [24]:

$$\ddot{\mathbf{q}}_k = \ddot{\mathbf{q}}_{k_{open}} - \mathbf{H}_k^{-1}\mathbf{J}_k^T\mathbf{f}_k, \tag{2.2}$$

where $\ddot{\mathbf{q}}_{k_{open}}$ is the vector of joint accelerations for chain k if it were not contacting the reference member resulting in a zero tip force. This is called its open-chain solution which was implemented in [14, 36]. Note that this quantity, as well as $\mathbf{H}_k$ and $\mathbf{J}_k$, depend only on the system state and input joint torques/forces and may be computed from known quantities.

Because the last link of each augmented chain is rigidly attached to the reference member, a point, called the reference point in Figure 2, can be chosen on the reference member that is also fixed with respect to each end-effector. It then becomes convenient to use an operational space approach [34] in the development of the dynamic equations. Hence, this point will serve as the location at which all of the chains' spatial quantities will be resolved. Analogous to Eq. (2.2), the operational space formulation for the dynamics of a closed-chain manipulator can be found by introducing the following well known relationship, using the manipulator Jacobian, between joint velocities and spatial (Cartesian) velocity of the end-effector of a manipulator at the reference point, $\dot{\mathbf{x}}_k$:

$$\dot{\mathbf{x}}_k = \mathbf{J}_k\dot{\mathbf{q}}_k. \tag{2.3}$$

Taking the time derivative of both sides yields the equation for the closed-chain acceleration of the tip:

$$\ddot{\mathbf{x}}_k = \dot{\mathbf{J}}_k\dot{\mathbf{q}}_k + \mathbf{J}_k\ddot{\mathbf{q}}_k, \tag{2.4}$$

and substituting $\ddot{\mathbf{q}}_k$ from Eq. (2.2) yields the following for $k = 1, \ldots, m$:

$$\ddot{\mathbf{x}}_k = \left(\dot{\mathbf{J}}_k\dot{\mathbf{q}}_k + \mathbf{J}_k\ddot{\mathbf{q}}_{k_{open}}\right) - \left(\mathbf{J}_k\mathbf{H}_k^{-1}\mathbf{J}_k^T\right)\mathbf{f}_k, \tag{2.5}$$

$$= \ddot{\mathbf{x}}_{k_{open}} - \mathbf{\Lambda}_k^{-1}\mathbf{f}_k, \tag{2.6}$$

where $\ddot{\mathbf{x}}_{k_{open}}$ is the acceleration of the tip of chain k (at the reference point on the end-effector) if it were in an open, unconstrained configuration, and the matrix, $\mathbf{\Lambda}_k^{-1}$, is the inverse of its operational space inertia matrix [34].

The physical interpretation of $\mathbf{\Lambda}_k$ is an inertial quantity which combines the dynamic properties of the links of the chain and projects them to its tip. It is the physical resistance that is "felt" when a force is exerted on the tip, and it defines the relationship between the force that is applied *onto* the tip, $-\mathbf{f}_k$, and the resulting acceleration of the tip of the chain, $\ddot{\mathbf{x}}_k$, as illustrated in Figure 3. The advantage to using these operational space quantities is that the matrix sizes are constant

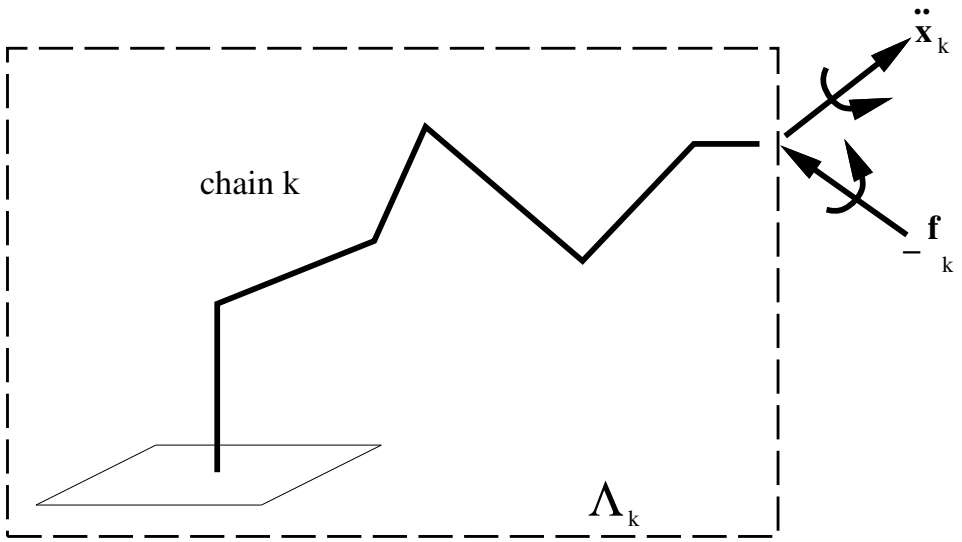

Figure 3: Operational space model for a single chain.

regardless of the number of degrees of freedom in the chain. Given the current state of the system and the input, both $\ddot{\mathbf{x}}_{k_{open}}$ and $\mathbf{\Lambda}_k$ can be computed with efficient methods described in [33, 37].

To complete the development, the dynamic equations for the reference member must be defined. Since the end-effector acceleration of each chain is computed at the reference point which is also fixed on the reference member, it is convenient to define the acceleration of the reference member, $\mathbf{a}_r$, at this point as well. As a result, the relationship, $\ddot{\mathbf{x}}_k = \mathbf{a}_r$, holds for all chains and Eq. (2.6) can be written as follows:

$$\mathbf{a}_r \;=\; \ddot{\mathbf{x}}_{k_{open}} - \mathbf{\Lambda}_k^{-1}\mathbf{f}_k. \tag{2.7}$$

Using a spatial force balance equation, the sum of the spatial forces exerted by the end-effector of each chain and any other external forces (including gravity) is related to the acceleration of the reference member through its spatial inertia, $\mathbf{I}_r$. This can be written in equation form for an arbitrary reference point as follows:

$$\sum_{k=1}^{m} \mathbf{f}_k + \mathbf{f}_r^g \;=\; \mathbf{I}_r\mathbf{a}_r + \bar{\mathbf{f}}_r. \tag{2.8}$$

The spatial force due to gravity exerted on the reference member is specified by $\mathbf{f}_r^g$, and $\bar{\mathbf{f}}_r$, is a spatial vector specifying the velocity-dependent bias forces of the

reference member:

$$\bar{\mathbf{f}}_r = \begin{bmatrix} \boldsymbol{\omega}_r \times \bar{\boldsymbol{I}}_r \boldsymbol{\omega}_r \\ \boldsymbol{\omega}_r \times (\boldsymbol{\omega}_r \times \boldsymbol{h}_r) \end{bmatrix}, \tag{2.9}$$

where $\boldsymbol{\omega}_r$ is the angular velocity, $\bar{\boldsymbol{I}}_r$ is the moment of inertia, and $\boldsymbol{h}_r$ is the first moment of mass of the reference member as defined in Chapter 1.

For the purposes of this and the next chapter, we assume that $N \geq 6$ and the manipulators are not at singular configurations. Consequently, all of the $\boldsymbol{\Lambda}_k^{-1}$ matrices are nonsingular (see Chapter 4 for modifications to this procedure that account for singular configurations). In this case, Eq. (2.7) may be solved for the contact force, $\mathbf{f}_k$, of each chain and substituted into Eq. (2.8). After collecting terms, the following equation results:

$$\left[\mathbf{I}_r + \sum_{k=1}^{m} \boldsymbol{\Lambda}_k\right] \mathbf{a}_r = \left[\mathbf{f}_r^g - \bar{\mathbf{f}}_r + \sum_{k=1}^{m} \boldsymbol{\Lambda}_k \ddot{\mathbf{x}}_{k_{open}}\right]. \tag{2.10}$$

This equation states that the sum of all the inertias of the system multiplied by the reference member acceleration is equal to the sum of all the forces that are exerted on the reference member. This acceleration may now be determined from this linear system of equations. Finally $\mathbf{a}_r$ is substituted back into Eq. (2.7) to determine the tip forces, $\mathbf{f}_k$:

$$\mathbf{f}_k = \boldsymbol{\Lambda}_k \left(\ddot{\mathbf{x}}_{k_{open}} - \mathbf{a}_r\right). \tag{2.11}$$

The tip forces may then be used in Eq. (2.2) to determine the joint accelerations.

2.3 An Efficient Algorithm for Rigid Grasps

In the application of interest, it is assumed that four PUMA manipulators are rigidly grasping the reference member. Using this assumption, the algorithm can be modified to increase its efficiency. Instead of using a single reference point, r, with respect to which all chains' operational space quantities ($\ddot{\mathbf{x}}_{k_{open}}$, $\mathbf{f}_k$, and $\boldsymbol{\Lambda}_k$) are computed, each chain computes its quantities in its last link's coordinate system. This will be referred to as the kth coordinate system, subsequently. One advantage of this approach is that most efficient single chain robotics algorithms have already been developed by others to compute these quantities and can then be directly applied to this problem. In addition, these coordinate systems are fixed with respect to the reference point, and the cost of transforming between them is much less than simulating the augmented chains with the extra link each.

Because the operational space quantities of each chain and the reference member are now computed with respect to different coordinate system, some of the equations

in the previous derivation require modification. First, Eq. (2.7) must be rewritten to express the reference member acceleration at the kth coordinate system:

$$
\begin{aligned}
{}^{k}\mathbf{a}_r &= \ddot{\mathbf{x}}_{k_{open}} - \mathbf{\Lambda}_k^{-1}\mathbf{f}_k \qquad \text{for } k = 1, \ldots, m. \qquad (2.12)
\end{aligned}
$$

where the standard spatial transformation for accelerations at different points on a rigid body is used to compute this acceleration:

$$
\begin{aligned}
{}^{k}\mathbf{a}_r &= {}^{k}\mathbf{X}_r\mathbf{a}_r + \boldsymbol{\zeta}_k, \qquad (2.13)
\end{aligned}
$$

and $\boldsymbol{\zeta}_k$ contains the velocity-dependent acceleration term as follows:

$$
\begin{aligned}
\boldsymbol{\zeta}_k &= \begin{bmatrix} \mathbf{0} \\ {}^{k}\boldsymbol{R}_r\left[\boldsymbol{\omega}_r \times (\boldsymbol{\omega}_r \times {}^{r}\boldsymbol{p}_k)\right] \end{bmatrix}, \qquad (2.14)
\end{aligned}
$$

where $\boldsymbol{\omega}_r$ is the angular velocity of the reference member, and ${}^{k}\boldsymbol{R}_r$ and ${}^{r}\boldsymbol{p}_k$ define the orientation and position between the two coordinate systems. Note that a variable without a leading superscript implies that components are expressed in the coordinate system indicated by the trailing subscript. Therefore $\mathbf{a}_r$ is expressed in the reference member coordinate system while ${}^{k}\mathbf{a}_r$ is expressed in the kth coordinate system.

Second, it is also desirable to include a gravitational acceleration term on both sides of the resulting equation:

$$
\begin{aligned}
{}^{k}\mathbf{X}_r\left(\mathbf{a}_r - {}^{r}\mathbf{a}_g\right) + \boldsymbol{\zeta}_k &= \ddot{\mathbf{x}}_{k_{open}} - {}^{k}\mathbf{a}_g - \mathbf{\Lambda}_k^{-1}\mathbf{f}_k, \qquad (2.15)
\end{aligned}
$$

where ${}^{r}\mathbf{a}_g$ and ${}^{k}\mathbf{a}_g$ are the same acceleration expressed at the reference and kth coordinate systems, respectively. Finally, Eq. (2.15) is solved for the tip forces as follows:

$$
\begin{aligned}
\mathbf{f}_k &= \mathbf{\Lambda}_k\left(\ddot{\mathbf{x}}'_{k_{open}} - {}^{k}\mathbf{X}_r\mathbf{a}'_r - \boldsymbol{\zeta}_k\right), \qquad (2.16)
\end{aligned}
$$

where

$$
\begin{aligned}
\ddot{\mathbf{x}}'_{k_{open}} &= \ddot{\mathbf{x}}_{k_{open}} - {}^{k}\mathbf{a}_g, \quad \text{and} \qquad (2.17)\\
\mathbf{a}'_r &= \mathbf{a}_r - {}^{r}\mathbf{a}_g. \qquad (2.18)
\end{aligned}
$$

The reason for including these gravitational acceleration terms is a consequence of the method by which the open-chain accelerations will be computed, and is explained in the next section.

Finally, the force balance on the reference member in Eq. (2.8) is modified. The tip forces must be transformed to the reference point, and the gravitational force

vector, $\mathbf{f}_r^g$, is replaced with the equivalent expression in terms of the gravitational acceleration. The resulting equation is written as follows:

$$\sum_{k=1}^{m} {}^k\mathbf{X}_r^T \mathbf{f}_k = \mathbf{I}_r \left(\mathbf{a}_r - {}^r\mathbf{a}_g\right) + \bar{\mathbf{f}}_r, \tag{2.19}$$

$$= \mathbf{I}_r \mathbf{a}'_r + \bar{\mathbf{f}}_r. \tag{2.20}$$

After substituting from Eq. (2.16) and collecting terms, the following system of equations results:

$$\left[\mathbf{I}_r + \sum_{k=1}^{m} \left({}^k\mathbf{X}_r^T \mathbf{\Lambda}_k {}^k\mathbf{X}_r\right)\right] \mathbf{a}'_r = \left[-\bar{\mathbf{f}}_r + \sum_{k=1}^{m} {}^k\mathbf{X}_r^T \mathbf{\Lambda}_k \left(\ddot{\mathbf{x}}'_{k_{open}} - \boldsymbol{\zeta}_k\right)\right], \tag{2.21}$$

which can be solved to obtain $\mathbf{a}'_r$ and from that the reference member acceleration. Using this derivation, the dynamics algorithm is summarized in Table 1.

This approach is an extension of the efficient way that gravitational forces are incorporated into the inverse dynamics computation. By biasing the acceleration with negative gravitational acceleration, the corresponding force is computed along with the bias forces when the biased acceleration is multiplied by the inertia. Physically, this is equivalent to the statement that the force exerted on a stationary body in a gravitational field is equivalent to the one felt if the body is accelerating with the negative of gravitational acceleration in zero-g. This is carried over into the calculation of the reference member acceleration, in Eq. (2.21), where the biased reference member acceleration times its inertia provides the missing gravitational force from the force balance equation. Due to more complex equations, however, unwanted terms are generated by the multiplication with the chains operational space inertias. These are negated by corresponding terms generated by using a biased open chain tip acceleration on the right hand side.

2.4 Computational Requirements

In this section, the computational requirements for an efficient implementation of each step of the algorithm in Table 1 is presented. To simplify the presentation, multiplication and division operations are combined and labeled as multiplies ($\times$), and subtractions are combined with additions ($+$). The computational requirements of the resulting implementation are summarized in Table 2 at the end of this section.

The Open Chain Dynamics (OCD) step is the most computationally expensive step. Details of its implementation have thus far been ignored in this chapter since efficient single, open-chain algorithms to complete this step have been developed by others. Lilly and Orin developed the Modified Composite-Rigid-Body Method (Method III) in [25] which computes $\mathbf{H}_k$ and $\mathbf{J}_k$ simultaneously with a computational

Table 1: Multiple closed-chain dynamics algorithm.

Step 1: Open Chain Dynamics.

for all $k = 1, \ldots, m$:

Compute $\mathbf{H}_k^{-1}\mathbf{J}_k^T$, $\mathbf{\Lambda}_k$, $\ddot{\mathbf{q}}_{k_{open}}$, $\ddot{\mathbf{x}}'_{k_{open}}$, ${}^k\mathbf{X}_r^T\mathbf{\Lambda}_k{}^k\mathbf{X}_r$, and ${}^k\mathbf{X}_r^T\mathbf{\Lambda}_k\left(\ddot{\mathbf{x}}'_{k_{open}} - \boldsymbol{\zeta}_k\right)$.

Step 2: Reference Member Acceleration.

Solve the following for $\mathbf{a}'_r$:

$$\left[\mathbf{I}_r + \sum_{k=1}^{m}\left({}^k\mathbf{X}_r^T\mathbf{\Lambda}_k{}^k\mathbf{X}_r\right)\right]\mathbf{a}'_r = \left[-\bar{\mathbf{f}}_r + \sum_{k=1}^{m}{}^k\mathbf{X}_r^T\mathbf{\Lambda}_k\left(\ddot{\mathbf{x}}'_{k_{open}} - \boldsymbol{\zeta}_k\right)\right],$$

$$\mathbf{a}_r = \mathbf{a}'_r - {}^r\mathbf{a}_g.$$

Step 3: Closed Chain Dynamics.

for all $k = 1, \ldots, m$:

$$\mathbf{f}_k = \mathbf{\Lambda}_k\left[\left(\ddot{\mathbf{x}}'_{k_{open}} - \boldsymbol{\zeta}_k\right) - {}^k\mathbf{X}_r\mathbf{a}'_r\right],$$
$$\ddot{\mathbf{q}}_k = \ddot{\mathbf{q}}_{k_{open}} - \left(\mathbf{H}_k^{-1}\mathbf{J}_k^T\right)\mathbf{f}_k.$$

---

cost of $[(12\frac{1}{2}N^2 + 24\frac{1}{2}N - 37)\text{M}, (8N^2 + 40N - 48)\text{A}]^*$ per chain. This algorithm has been improved by replacing its procedure for transforming composite rigid body inertias with the one in Appendix A. For a chain with revolute joints only the computational cost drops to $[(12\frac{1}{2}N^2 + 5\frac{1}{2}N - 18)\text{M}, (8N^2 + 30N - 38)\text{A}]$. Then, $\mathbf{H}_k^{-1}\mathbf{J}_k^T$ (also called $\mathbf{\Omega}_k$ in [25]), $\mathbf{\Lambda}_k^{-1}$, and $\mathbf{\Lambda}_k$ can be determined using the procedure described in Lilly and Orin's Unit Force Method (Method II) of [37].

The inverse dynamics algorithm from Appendix B.2 is used in the computation of $\ddot{\mathbf{q}}_{k_{open}}$ and $\ddot{\mathbf{x}}_{k_{open}}$. By setting the joint accelerations and tip forces to zero, the output of the inverse dynamics computation, also called the bias computation [14] under these conditions, consists of the following terms from Eq. (2.1):

$$\mathbf{b}_k = \mathbf{C}_k(\mathbf{q}_k, \dot{\mathbf{q}}_k) + \mathbf{G}_k(\mathbf{q}_k). \tag{2.22}$$

*$[(x)\text{M}, (y)\text{A}] = x$ multiplications, y additions.

The computational cost is $[(94N-114)\text{M}, (81N-105)\text{A}]$ per chain. With the input joint torques known, the open-chain joint accelerations are computed as follows:

$$\ddot{\mathbf{q}}_{k_{open}} = \mathbf{H}_k^{-1}\left(\boldsymbol{\tau}_k - \mathbf{b}_k\right). \tag{2.23}$$

Since the Cholesky decomposition of $\mathbf{H}_k$ has already been accomplished in the previous step, only a vector subtraction and back-substitution is required to complete this step at a cost of $[(N^2)\text{M}, (N^2)\text{A}]$ per chain.

A by-product of the inverse dynamics computation is the computation of the acceleration of each link. With the joint accelerations set to zero and the fixed base's acceleration set to negative gravitational acceleration, $-\mathbf{a}_g$, it can be shown that the acceleration computed for the last link of chain k is:

$$\mathbf{a}_{k,N} = \dot{\mathbf{J}}_k\dot{\mathbf{q}}_k - {}^k\mathbf{a}_g. \tag{2.24}$$

The desired open-chain quantity, $\ddot{\mathbf{x}}'_{k_{open}}$, can then be computed [from Eqs. (2.4) and (2.17)] as follows:

$$\ddot{\mathbf{x}}'_{k_{open}} = \mathbf{J}_k\ddot{\mathbf{q}}_{k_{open}} + \mathbf{a}_{k,N}, \tag{2.25}$$

at a cost of $[(6N)\text{M}, (6N)\text{A}]$ per chain.

The final part of the OCD step involves the spatial transformation of operational space quantities to the reference point. With careful kinematic modeling, the transformation between the last link of each chain and the reference point can be made especially efficient. This is accomplished with the coordinate system assignment shown Figure 4. By placing the x-axis of the last link along the common normal of joint N and the reference member's z-axis, the transformation between these coordinate systems is efficiently specified by another set of modified Denavit-Hartenberg (MDH) parameters. Using these four parameters, the spatial transformation, ${}^r\mathbf{X}_k$, is defined as in Appendix A. Another method of defining this transformation is by two successive planar screws about the x and z axes [27]:

$${}^r\mathbf{X}_k = \mathbf{X}_z(d,\theta)\,\mathbf{X}_x(a,\alpha). \tag{2.26}$$

However, the transformation that is actually needed in the previous development is ${}^k\mathbf{X}_r$ which is the inverse of ${}^r\mathbf{X}_k$. This matrix can be still written as two planar screws as follows:

$${}^k\mathbf{X}_r = \mathbf{X}_x(-a,-\alpha)\,\mathbf{X}_z(-d,-\theta). \tag{2.27}$$

Therefore the efficiency of the transformations using MDH parameters is also achieved with this. With results from Appendix A, the congruence transformation of $\boldsymbol{\Lambda}_k$, a symmetric 6×6 matrix, requires [84M, 86A]. The computation of the spatial vector, $\boldsymbol{\Lambda}_k\left(\ddot{\mathbf{x}}'_{k_{open}} - \boldsymbol{\zeta}_k\right)$ and its spatial transformation requires [76M, 55A].

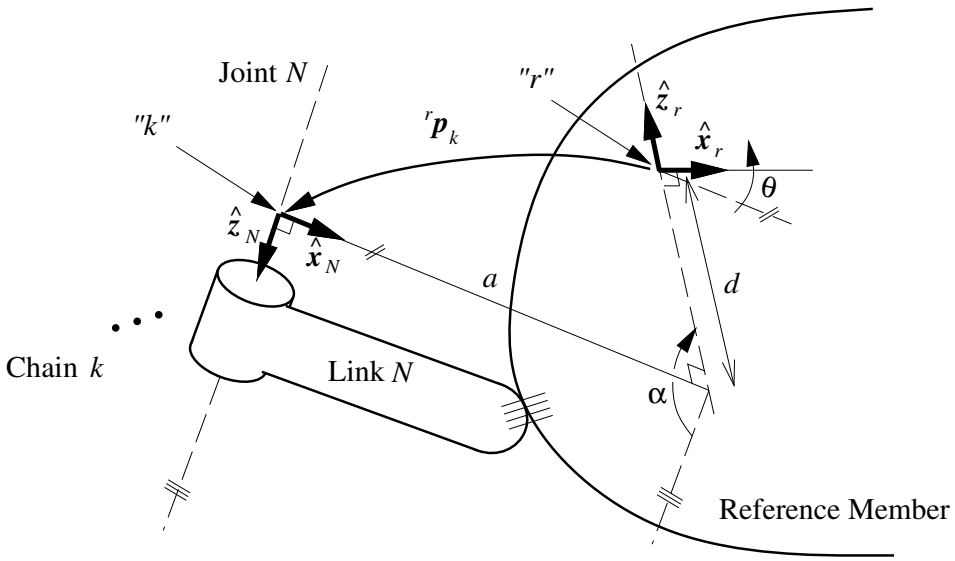

Figure 4: Efficient coordinate system assignment for each chain.

In the Reference Member Acceleration (RMA) step of the computation in Table 1, the computation of $\mathbf{a}'_r$ requires the computation of $\bar{\mathbf{f}}_r$, the summation of seven spatial vectors, the summation of m symmetric 6×6 matrices and a spatial inertia matrix, and the solution of a 6×6 system of equations. By placing the reference member coordinate axes along its principal axes, $\boldsymbol{h}_r$ is zero and $\bar{\boldsymbol{I}}_r$ is diagonal. Therefore, $\bar{\mathbf{f}}_r$ contains three zeros and only requires [9M, 3A] to be computed. The summations then require a total of $27m - 9$ additions. Since the resulting matrix is symmetric and positive definite, root-free Cholesky decomposition and back-substitution is the most efficient procedure to compute $\mathbf{a}'_r$ at a cost of [86M, 65A] [14, 24].

To complete the RMA step, $\mathbf{a}_r$, must be computed. To accomplish this, the gravitational acceleration must be transformed to the reference coordinate system and subtracted from $\mathbf{a}'_r$. If the z-axis of the inertial coordinate system is placed along the direction of gravitational acceleration, and Euler angles are used to specify the orientation of the reference member with respect to this frame, the transformation requires [4M, 0A] and the subtraction requires [0M, 3A] since the angular acceleration portion of the gravitational acceleration spatial vector is always zero.

The computation requirements for the Closed Chain Dynamics (CCD) step is relatively straightforward. To compute the constraint forces at the tip requires a spatial transformation [20M, 12A], spatial vector addition [0M, 6A], and spatial matrix-vector multiply [36M, 30A]. The joint accelerations of each chain is obtained from a $N \times 6$ matrix-vector multiply [$(6N)$M, $(5N)$A] and an N-vector addition.

Table 2: Computational requirements for the closed-chain dynamic simulation algorithm.

Computation	$\times$	$+$
$\mathbf{H}_k,\ \mathbf{J}_k$	$m\left(12\frac{1}{2}N^2 + 5\frac{1}{2}N - 18\right)$	$m\left(8N^2 + 30N - 38\right)$
$\mathbf{H}_k^{-1}\mathbf{J}_k^T$	$m\left(\frac{1}{6}N^3 + 6\frac{1}{2}N^2 - \frac{2}{3}N\right)$	$m\left(\frac{1}{6}N^3 + 6N^2 - 6\frac{1}{6}N\right)$
$\mathbf{\Lambda}_k^{-1}$	$21mN$	$m(21N - 21)$
$\mathbf{\Lambda}_k$	$141m$	$90m$
$\ddot{\mathbf{q}}_{k_{open}},\ \ddot{\mathbf{x}}'_{k_{open}}$	$m(N^2 + 100N - 114)$	$m(N^2 + 87N - 105)$
${}^k\mathbf{X}_r^T\mathbf{\Lambda}_k{}^k\mathbf{X}_r$	$84m$	$86m$
${}^k\mathbf{X}_r^T\mathbf{\Lambda}_k\left(\ddot{\mathbf{x}}'_{k_{open}} - \boldsymbol{\zeta}_k\right)$	$76m$	$55m$
$\mathbf{a}'_r,\ \mathbf{a}_r$	99	$27m + 62$
$\mathbf{f}_k$	$56m$	$48m$
$\ddot{\mathbf{q}}_k$	$6mN$	$6mN$
Total:	$m(\frac{1}{6}N^3 + 20N^2 + 131\frac{5}{6}N + 225) + 99$	$m(\frac{1}{6}N^3 + 15N^2 + 137\frac{5}{6}N + 142) + 62$
$m = 4, N = 6$	7187	6242

The totals at the bottom of Table 2, show that this algorithm has a computational complexity of $\mathrm{O}(mN^3)$, and for the four PUMA system [7187M, 6242A] are required to compute the desired accelerations.

2.5 Summary and Conclusions

In this chapter, the dynamics of a multiple manipulator system grasping a common object has been presented. For each chain, at least six degrees of freedom, non-singular configurations, and rigid grasps with the reference member were assumed, and an efficient $\mathrm{O}(mN^3)$ algorithm for the computation of the joint accelerations in each chain and the acceleration of the reference member was developed. The most important characteristic of this development is its dependence on the more common serial chain algorithms wherever possible. One reason is that a significant amount

of work has been completed by others in this area. Based on many of their results, the best algorithms have been developed in Appendix B and are heavily relied on in the development in this Chapter. Another reason is that it allows a higher level approach to the development of the algorithm.

For a system consisting of four PUMA 560 manipulators ($m = 4$, $N = 6$), this algorithm requires [7187M, 6242A] to compute the desired accelerations. By comparison, Lilly and Orin's O(mN) algorithm in [24] has a computational cost of [$(656mN - 171m + 128)$M, $(546mN - 208m + 101)$A]. For the four-PUMA system under consideration this method requires [15188M, 12589A]. The new algorithm represents a savings of over 50%, and is more efficient for $N \leq 21$. Although Lilly and Orin's algorithm considers the case of arbitrary grasps on the reference member, most of the savings for the algorithm presented in this chapter is achieved in the computation of the open-chain quantities which are unaffected by the grasp type.

This chapter has also provided the background needed in the next two chapters. While the algorithm developed in this chapter is very efficient, Chapter 3 examines methods of parallelizing this algorithm to further increase the computation rates so that real-time simulation of this system can be achieved. This algorithm also does not address the case when a manipulator becomes singular. This problem will be specifically addressed in Chapter 4.