APPENDICES

Appendix A

Efficient Transformations with Modified Denavit-Hartenberg Parameters

A.1 Introduction

Transformations of vectors and matrices account for a significant portion of the operations in any of the dynamics algorithms developed in this dissertation. Hence, efficient implementations of these operations is very important for achieving an overall efficient algorithm. All transformations in this dissertation can be represented as one or more successive planar operations in which rotation and possibly translation occur about and around the same axis. It turns out to be most efficient to perform these transformations in multiple stages of planar operations. The procedure and computational requirements are the same regardless of which axis is used so in many of the presentations only one planar operation is discussed in detail and results for more complex operations are extrapolated from these results. This appendix begins with a presentation of the modified Denavit-Hartenberg (MDH) notation used extensively in this dissertation. Then the efficient transformations of Cartesian and spatial vectors and matrices that results with the use of this notation are developed.

A.2 Link Coordinate Systems and MDH Parameters

The most efficient dynamics algorithms for articulated mechanisms result when coordinate systems are assigned to each link in the system. For a vast majority of systems, the links in the serial chains are attached to one another with single degree-of-freedom revolute or prismatic joints. In these cases, it is especially efficient to assign these coordinate systems according to a set of rules similar to those first described by Denavit and Hartenberg in [70]. The resulting modified Denavit-Hartenberg (MDH) notation has also been presented by Craig in [26].

Using this notation, the links are numbered, in succession, from link 1, attached to the base, to the last link at the tip or end-effector, link N. As shown in Figure 38, the joint between links $i-1$ and i is labeled joint i, and the origin of coordinate

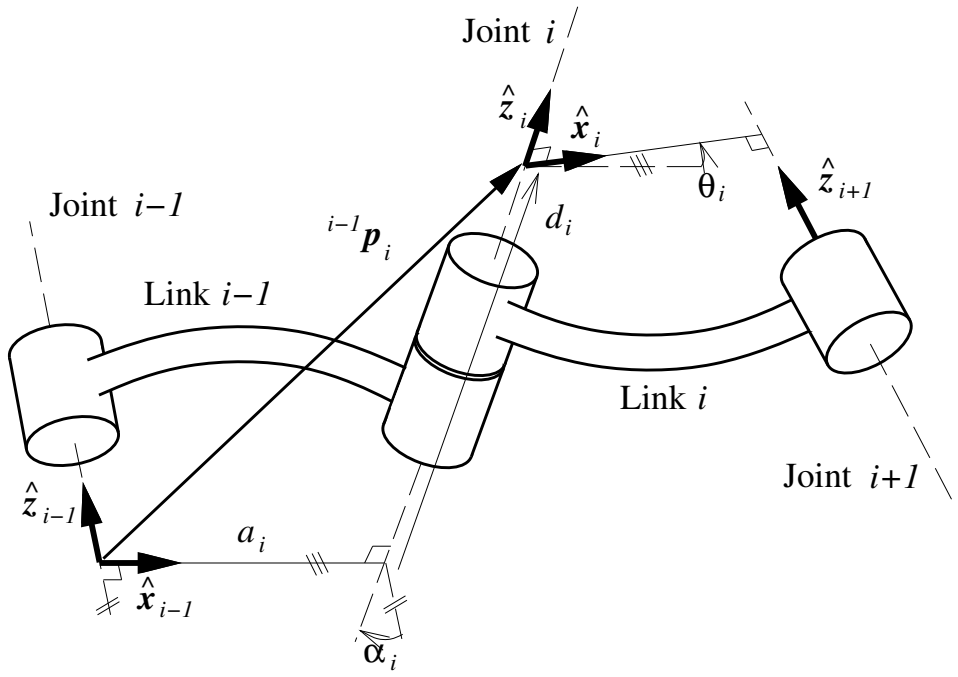

Figure 38: Link coordinate system assignment and MDH parameters.

frame i lies on the axis of this joint's motion. Further, this coordinate frame is fixed to link i at the end "nearest" the base.

The axes of each coordinate system are aligned so that the z-axis, $\hat{\boldsymbol{z}}$ lies along the axis of motion of the joint. This is the axis about which the joint rotates for revolute joints, and along which the joint translates for prismatic joints. The x-axis, $\hat{\boldsymbol{x}}$, lies along the common normal between this joint axis and the next one in the chain as shown in Figure 38. With this arrangement, only four parameters are needed to describe the relative position and orientation between adjacent coordinate systems. To move from link $i-1$'s coordinate system to link i, these parameters are defined as follows:

a_i = the perpendicular distance along $\hat{\boldsymbol{x}}_{i-1}$ from $\hat{\boldsymbol{z}}_{i-1}$ to $\hat{\boldsymbol{z}}_i$,
α_i = the angle about $\hat{\boldsymbol{x}}_{i-1}$ from $\hat{\boldsymbol{z}}_{i-1}$ to $\hat{\boldsymbol{z}}_i$,
d_i = the perpendicular distance along $\hat{\boldsymbol{z}}_i$ from $\hat{\boldsymbol{x}}_{i-1}$ to $\hat{\boldsymbol{x}}_i$, and
θ_i = the angle about $\hat{\boldsymbol{z}}_i$ from $\hat{\boldsymbol{x}}_{i-1}$ to $\hat{\boldsymbol{x}}_i$,

where d_i is variable if joint i is prismatic, or θ_i is variable if it is revolute. Note that when the base of the chain is fixed with respect to an inertial frame, it is convenient to place the origin of coordinate frame 0 coincident with coordinate system 1 with

its z-axis also along joint 1. As a result, the only non-zero parameter is the joint variable, θ_1 or d_1.

These parameters are used to define the rotational transformation matrix and position vector from one coordinate frame to the next. The rotational transformation from coordinate frame $i-1$ to frame i is defined as follows:

$$ {}^{i}\boldsymbol{R}_{i-1} = \begin{bmatrix} c\theta_i & s\theta_i c\alpha_i & s\theta_i s\alpha_i \\ -s\theta_i & c\theta_i c\alpha_i & c\theta_i s\alpha_i \\ 0 & -s\alpha_i & c\alpha_i \end{bmatrix}, \tag{A.1} $$

where $s\alpha_i$, $c\alpha_i$, $s\theta_i$, and $c\theta_i$ denote $\sin\alpha_i$, $\cos\alpha_i$, $\sin\theta_i$, and $\cos\theta_i$, respectively. This can also be expressed as two planar rotations:

$$ {}^{i}\boldsymbol{R}_{i-1} = \boldsymbol{R}(z,\theta_i)\,\boldsymbol{R}(x,\alpha_i), \tag{A.2} $$

$$ = \begin{bmatrix} \mathrm{c}\theta_i & \mathrm{s}\theta_i & 0 \\ -\mathrm{s}\theta_i & \mathrm{c}\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{c}\alpha_i & \mathrm{s}\alpha_i \\ 0 & -\mathrm{s}\alpha_i & \mathrm{c}\alpha_i \end{bmatrix}. \tag{A.3} $$

The 3×1 position vector from the origin of frame $i-1$ to the origin of frame i (with components expressed in frame $i-1$) shown in Figure 38 is defined as follows:

$$ {}^{i-1}\boldsymbol{p}_i = \begin{bmatrix} a_i \\ -d_i\, s\alpha_i \\ d_i\, c\alpha_i \end{bmatrix}, \tag{A.4} $$

which is constant when joint i is revolute.

These two quantities are also used to define the 6×6 spatial transformation matrix, ${}^{i}\mathbf{X}_{i-1}$, which is used to transform spatial vectors between adjacent coordinate systems. The spatial transformation between joints $i-1$ and i is expressed as follows:

$$ {}^{i}\mathbf{X}_{i-1} = \begin{bmatrix} {}^{i}\boldsymbol{R}_{i-1} & \mathbf{0} \\ {}^{i}\boldsymbol{R}_{i-1}\,{}^{i-1}\tilde{\boldsymbol{p}}_i{}^{T} & {}^{i}\boldsymbol{R}_{i-1} \end{bmatrix}, \tag{A.5} $$

where the tilde above the position vector signifies that its components should be combined in a skew symmetric matrix such that $\tilde{\boldsymbol{b}}\boldsymbol{a} = \boldsymbol{b} \times \boldsymbol{a}$. This matrix can also be expressed as successive planar screw transformations about the x and z axes as follows:

$$ {}^{i}\mathbf{X}_{i-1} = \mathbf{X}(z,d_i,\theta_i)\,\mathbf{X}(x,a_i,\alpha_i), \tag{A.6} $$

$$ = \begin{bmatrix} \boldsymbol{R}(z,\theta_i) & \mathbf{0} \\ \boldsymbol{R}(z,\theta_i)\,\tilde{\boldsymbol{p}}_z^{T} & \boldsymbol{R}(z,\theta_i) \end{bmatrix} \begin{bmatrix} \boldsymbol{R}(x,\alpha_i) & \mathbf{0} \\ \boldsymbol{R}(x,\alpha_i)\,\tilde{\boldsymbol{p}}_x^{T} & \boldsymbol{R}(x,\alpha_i) \end{bmatrix}, \tag{A.7} $$

where $\boldsymbol{p}_x = [a_i\ 0\ 0]^T$ and $\boldsymbol{p}_z = [0\ 0\ d_i]^T$.

A.3 Vector Transformations

In this section, transformations of Cartesian and spatial vectors are presented. Examples using only the z-axis planar transformation are given which were also reported by Featherstone ([27], p. 146). The computational cost of successive planar transformations on the result is the same so that the cost of transforming vectors using the MDH parameters from the previous section can be obtained from this result directly.

A.3.1 Cartesian Vectors

The rotation of a vector about the z-axis through an angle θ is accomplished by the following multiplication:

$$\boldsymbol{R}(z,\theta)\,\boldsymbol{p} \;=\; \boldsymbol{p}' \tag{A.8}$$

$$\begin{bmatrix} \mathrm{c}\theta & \mathrm{s}\theta & 0 \\ -\mathrm{s}\theta & \mathrm{c}\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \;=\; \begin{bmatrix} \mathrm{c}\theta p_x + \mathrm{s}\theta p_y \\ \mathrm{c}\theta p_y - \mathrm{s}\theta p_x \\ p_z \end{bmatrix}. \tag{A.9}$$

The computational cost of this step is only [4M, 2A], which is the same if the rotation is carried out about either of the other two axes. The rotation of a vector from link $(i-1)$ to link i's coordinate system using MDH parameters requires a rotation of α_i about the x-axis and then θ_i about the z-axis as follows:

$$\boldsymbol{p}_i \;=\; \boldsymbol{R}(z,\theta_i)\;\boldsymbol{R}(x,\alpha_i)\;\boldsymbol{p}_{i-1}. \tag{A.10}$$

Each planar rotation requires [4M, 2A] for a total cost of [8M, 4A]. This is much more efficient than computing ${}^{i}\boldsymbol{R}_{i-1}$ from Eq. (A.1), requiring [4M, 0A], and performing the matrix-vector multiply requiring [9M, 6A]. It should also be noted that rotations in the opposite direction, which require ${}^{i}\boldsymbol{R}_{i-1}^{T}$, require [8M, 4A] as well.

A.3.2 Spatial Vectors

Successive planar screw transformation operations, which require rotation and translation about the same axis, can be used to transform spatial vectors from one coordinate system to another. For example rotating a spatial velocity vector, $\mathbf{v}$, about the z-axis through an angle θ and translating along it through a distance, d, can be accomplished with the following matrix operation:

$$\mathbf{X}(z,d,\theta)\,\mathbf{v} \;=\; \mathbf{v}', \tag{A.11}$$

$$\begin{bmatrix} \mathrm{c}\theta & \mathrm{s}\theta & 0 & 0 & 0 & 0 \\ -\mathrm{s}\theta & \mathrm{c}\theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -\mathrm{s}\theta d & \mathrm{c}\theta d & 0 & \mathrm{c}\theta & \mathrm{s}\theta & 0 \\ -\mathrm{c}\theta d & -\mathrm{s}\theta d & 0 & -\mathrm{s}\theta & \mathrm{c}\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} \mathrm{c}\theta\omega_x + \mathrm{s}\theta\omega_y \\ \mathrm{c}\theta\omega_y - \mathrm{s}\theta\omega_x \\ \omega_z \\ \mathrm{c}\theta\left(v_x + \omega_y d\right) + \mathrm{s}\theta\left(v_y - \omega_x d\right) \\ \mathrm{c}\theta\left(v_y - \omega_x d\right) - \mathrm{s}\theta\left(v_x + \omega_y d\right) \\ v_z \end{bmatrix}. \quad \text{(A.12)}$$

This operation requires [10M, 6A]. The transformation of a spatial vector between links using all of the MDH parameters involves an x-planar screw as well:

$$\mathbf{v}_i \;=\; \mathbf{X}\left(z, d, \theta\right)\; \mathbf{X}\left(x, a, \alpha\right)\; \mathbf{v}_{i-1}. \quad \text{(A.13)}$$

Each screw transformation requires [10M, 6A] for a total of [20M, 12A]. Spatial transformations of forces in the other direction requires the transpose of these matrices. Nevertheless, the computational cost remains the same for this operation as well.

A.4 Matrix Congruence Transformations

In order to transform inertia matrices from one coordinate system to the next, a congruence transformation operation is required. In this section, transformations for symmetric and general 3×3 matrices are examined in detail. The results represent the most efficient that have been reported in the robotics literature. These operations form the basis for the more complex transformations involving symmetric 6×6 matrices such as the operational space inertias used in Part I and Articulated Body (AB) inertias used in Part II of this dissertation. Finally, the transformation of specialized 6×6 matrices such as the reflected AB inertia and the spatial inertia, which has the same form as a CRB inertia matrix, are covered.

A.4.1 Symmetric 3×3 Matrices

First, the efficient procedure for performing the congruence transformation of a symmetric 3×3 matrix, $\bar{\boldsymbol{K}}_i$, from link i to link $(i-1)$'s coordinate system is presented. The transformation matrix is ${}^{i}\boldsymbol{R}_{i-1}$, and has already been shown to consist of two planar rotations. Therefore, this computation can be divided into two steps: a z-axis planar congruence transform through the angle θ_i and an x-axis planar congruence transform through the angle α_i.

The first transformation results in the following matrix:

$$\bar{\boldsymbol{K}}^{z} \;=\; \boldsymbol{R}^{T}(z, \theta)\; \bar{\boldsymbol{K}}_i \boldsymbol{R}(z, \theta) \quad \text{(A.14)}$$

$$= \begin{bmatrix} \mathrm{c}\theta & -\mathrm{s}\theta & 0 \\ \mathrm{s}\theta & \mathrm{c}\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{12} & k_{22} & k_{23} \\ k_{13} & k_{23} & k_{33} \end{bmatrix} \begin{bmatrix} \mathrm{c}\theta & \mathrm{s}\theta & 0 \\ -\mathrm{s}\theta & \mathrm{c}\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.15}$$

$$= \begin{bmatrix} k_{11}+a & b & k_{13}\mathrm{c}\theta - k_{23}\mathrm{s}\theta \\ b & k_{22}-a & k_{13}\mathrm{s}\theta + k_{23}\mathrm{c}\theta \\ k_{13}\mathrm{c}\theta - k_{23}\mathrm{s}\theta & k_{13}\mathrm{s}\theta + k_{23}\mathrm{c}\theta & k_{33} \end{bmatrix}, \tag{A.16}$$

where

$$a = (k_{22}-k_{11})\mathrm{s}^2\theta - k_{12}(2\mathrm{s}\theta\mathrm{c}\theta) \tag{A.17}$$

$$b = k_{12}(\mathrm{c}^2\theta - \mathrm{s}^2\theta) - (k_{22}-k_{11})(\mathrm{s}\theta\mathrm{c}\theta). \tag{A.18}$$

The number of operations required to compute this matrix are given in the following table:

Computation	$\times$	$+$
$\mathrm{s}^2\theta = \sin\theta\sin\theta$	1	0
$\mathrm{c}^2\theta - \mathrm{s}^2\theta = 1 - \mathrm{s}^2\theta - \mathrm{s}^2\theta$	0	2
$\mathrm{s}\theta\mathrm{c}\theta = \sin\theta\cos\theta$	1	0
$2\mathrm{s}\theta\mathrm{c}\theta = \mathrm{s}\theta\mathrm{c}\theta + \mathrm{s}\theta\mathrm{c}\theta$	0	1
$k_{22} - k_{11}$	0	1
$(k_{22}-k_{11})(\mathrm{s}\theta\mathrm{c}\theta),\ (k_{22}-k_{11})\mathrm{s}^2\theta$	2	0
$k_{12}(\mathrm{c}^2\theta - \mathrm{s}^2\theta),\ k_{12}(2\mathrm{s}\theta\mathrm{c}\theta)$	2	0
$a,\ b$	0	2
Elements $k^z_{33} = k_{33},\ k^z_{12} = b$	0	0
Elements $k^z_{11},\ k^z_{22}$	0	2
Elements $k^z_{13},\ k^z_{23}$	4	2

Note that the resulting matrix is also symmetric so only the upper triangular elements need to be computed resulting in [10M, 10A] (only [8M, 7A] when θ is constant).

This result is used in the x-planar congruence transform to complete the operation as follows:

$${}^{i-1}\bar{\boldsymbol{K}}_i = \boldsymbol{R}^T(x,\alpha)\,\bar{\boldsymbol{K}}^z\boldsymbol{R}(x,\alpha). \tag{A.19}$$

The procedure for breaking down this computation into the fewest number of operations is the same as that found for the z-planar transformation. Note, however, that the link twist angle, α, is constant and the trigonometry functions involving this angle are constant and can be computed off-line. Therefore, this step requires [8M, 7A], and a total of [18M, 17A] is needed for the complete congruence transformation ([16M, 14A] when θ is constant).

A.4.2 General 3×3 Matrices

To find an efficient method for transforming a general 3×3 matrix, a procedure similar to the one used above is followed. The z-planar transformation results in the following matrix:

$$\begin{aligned}
\bar{\boldsymbol{N}}^z &= \boldsymbol{R}^T(z,\theta)\, \bar{\boldsymbol{N}}_i \boldsymbol{R}(z,\theta) && \text{(A.20)}\\
&= \begin{bmatrix} \mathrm{c}\theta & -\mathrm{s}\theta & 0 \\ \mathrm{s}\theta & \mathrm{c}\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_{11} & n_{12} & n_{13} \\ n_{21} & n_{22} & n_{23} \\ n_{31} & n_{32} & n_{33} \end{bmatrix} \begin{bmatrix} \mathrm{c}\theta & \mathrm{s}\theta & 0 \\ -\mathrm{s}\theta & \mathrm{c}\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} && \text{(A.21)}\\
&= \begin{bmatrix} n_{11}+a & n_{12}-b & n_{13}\mathrm{c}\theta - n_{23}\mathrm{s}\theta \\ n_{21}-b & n_{22}-a & n_{13}\mathrm{s}\theta + n_{23}\mathrm{c}\theta \\ n_{31}\mathrm{c}\theta - n_{32}\mathrm{s}\theta & n_{31}\mathrm{s}\theta + n_{32}\mathrm{c}\theta & n_{33} \end{bmatrix}, && \text{(A.22)}
\end{aligned}$$

where

$$\begin{aligned}
a &= (n_{22}-n_{11})\mathrm{s}^2\theta - (n_{12}+n_{21})\mathrm{s}\theta\mathrm{c}\theta && \text{(A.23)}\\
b &= (n_{22}-n_{11})\mathrm{s}\theta\mathrm{c}\theta + (n_{12}+n_{21})\mathrm{s}^2\theta. && \text{(A.24)}
\end{aligned}$$

The number of operations required to compute this matrix are given in the following table:

Computation	$\times$	$+$
$\mathrm{s}^2\theta, \mathrm{s}\theta\mathrm{c}\theta$	2	0
$(n_{22}-n_{11})\mathrm{s}^2\theta,\ (n_{22}-n_{11})(\mathrm{s}\theta\mathrm{c}\theta)$	2	1
$(n_{12}+n_{21})\mathrm{s}^2\theta,\ (n_{12}+n_{21})(\mathrm{s}\theta\mathrm{c}\theta)$	2	1
$a,\ b$	0	2
Elements $n^z_{11},\ n^z_{12},\ n^z_{21},\ n^z_{22}$	0	4
Elements $n^z_{13},\ n^z_{31},\ n^z_{32},\ n^z_{23}$	8	4
Element $n^z_{33} = n_{33}$	0	0

The computational cost of this step is [14M, 12A]. To complete the transformation, the x-planar transformation proceeds in a similar fashion but only requires [12M, 12A] because α is constant when using MDH parameters and the trigonometric computations can be computed off-line.

A.4.3 Symmetric 6×6 Matrices

This section presents the details required to efficiently transform both operational space inertia matrices that are used in Chapter 2 and AB inertia matrices that are used in Chapter 7. Efficient computation of spatial congruence transformations

Table 21: Computational cost of a 6×6 planar congruence transformation.

Step	Computation	$\times$	$+$
(A)	$s^2\theta$, $c^2\theta$ - $s^2\theta$, $s\theta c\theta$, $2s\theta c\theta$	2	3
(B)	$\boldsymbol{R}^T \boldsymbol{\Lambda}_{11} \boldsymbol{R}$	8	7
(C)	$[\mathbf{X}^T \boldsymbol{\Lambda} \mathbf{X}]_{22} = \boldsymbol{R}^T \boldsymbol{\Lambda}_{22} \boldsymbol{R}$	8	7
(D)	$\boldsymbol{R}^T \boldsymbol{\Lambda}_{12} \boldsymbol{R}$	12	12
(E)	$\tilde{\boldsymbol{p}}(\boldsymbol{R}^T \boldsymbol{\Lambda}_{22} \boldsymbol{R})$	5	0
(F)	$(\tilde{\boldsymbol{p}} \boldsymbol{R}^T \boldsymbol{\Lambda}_{22} \boldsymbol{R}) \tilde{\boldsymbol{p}}^T$	3	0
(G)	$(\boldsymbol{R}^T \boldsymbol{\Lambda}_{12} \boldsymbol{R}) \tilde{\boldsymbol{p}}^T$	6	0
(H)	$[\mathbf{X}^T \boldsymbol{\Lambda} \mathbf{X}]_{11}$ = (B)+(F)+(G)+(G)^T	0	11
(I)	$[\mathbf{X}^T \boldsymbol{\Lambda} \mathbf{X}]_{12} = [\mathbf{X}^T \boldsymbol{\Lambda} \mathbf{X}]_{21}^T$ = (D)+(E)	0	6
	Total:	44	46

requires the same approach as before by computing planar screw transformations in succession. Therefore the congruence transformation is written as follows:

$$\mathbf{X}^T \boldsymbol{\Lambda} \mathbf{X} \;=\; \mathbf{X}^T(x,a,\alpha)\, \mathbf{X}^T(z,d,\theta)\, \boldsymbol{\Lambda}\, \mathbf{X}(z,d,\theta)\, \mathbf{X}(x,a,\alpha). \tag{A.25}$$

Each planar congruence transform is performed separately using the z planar screw transformation first, and then performing the x planar screw transformation on this result.

First the z planar screw transformation step is written in terms of its Cartesian components as follows:

$$\boldsymbol{\Lambda}^z = \mathbf{X}^T(z,d,\theta)\, \boldsymbol{\Lambda}\, \mathbf{X}(z,d,\theta) \tag{A.26}$$

$$= \begin{bmatrix} \boldsymbol{R}^T & \tilde{\boldsymbol{p}} \boldsymbol{R}^T \\ \mathbf{0} & \boldsymbol{R}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \mathbf{0} \\ \boldsymbol{R} \tilde{\boldsymbol{p}}^T & \boldsymbol{R} \end{bmatrix} \tag{A.27}$$

$$= \left[\begin{array}{l|l} \boldsymbol{R}^T \boldsymbol{\Lambda}_{11} \boldsymbol{R} + \tilde{\boldsymbol{p}} \boldsymbol{R}^T \boldsymbol{\Lambda}_{22} \boldsymbol{R} \tilde{\boldsymbol{p}}^T + & \boldsymbol{R}^T \boldsymbol{\Lambda}_{12} \boldsymbol{R} + \tilde{\boldsymbol{p}} \boldsymbol{R}^T \boldsymbol{\Lambda}_{22} \boldsymbol{R} \\ \quad \boldsymbol{R}^T \boldsymbol{\Lambda}_{12} \boldsymbol{R} \tilde{\boldsymbol{p}}^T + \left(\boldsymbol{R}^T \boldsymbol{\Lambda}_{12} \boldsymbol{R} \tilde{\boldsymbol{p}}^T\right)^T & \\ \hline \left(\boldsymbol{R}^T \boldsymbol{\Lambda}_{12} \boldsymbol{R} + \tilde{\boldsymbol{p}} \boldsymbol{R}^T \boldsymbol{\Lambda}_{22} \boldsymbol{R}\right)^T & \boldsymbol{R}^T \boldsymbol{\Lambda}_{22} \boldsymbol{R} \end{array}\right], \tag{A.28}$$

where $\boldsymbol{R}$ and $\boldsymbol{p}$ are $\boldsymbol{R}(z,d,\theta)$ and $\boldsymbol{p}_z$, respectively, and defined earlier in this appendix. Nine steps are identified in this computation and listed in Table 21.

Note that the numbers reported in this table rely heavily on the previous results of this section for transformation of 3×3 matrices. The trigonometric functions

needed for 3×3 symmetric and general matrices overlap; therefore, they are listed separately in Step (A). The remainder of the computation for the symmetric and general 3×3 congruence transformations are listed in Steps (B), (C), and (D).

The multiplication of a 3×3 matrix by $\tilde{\boldsymbol{p}}$ in Steps (E), (F), and (G) are equivalent to performing three cross products with a vector whose third element is the only non-zero element. In Step (E), the matrix is symmetric so only [5M, 0A] are required and the third row of the result is zero. The result of Step (F) is symmetric and has only four non-zero (three in the upper triangular portion) elements which requires [3M, 0A] to compute. Unlike Step (E), Step (G) begins with a general matrix so that [6M, 0A] are required to produce the result which also has a zero row.

Step (H) involves the addition of four 3×3 matrices whose result is symmetric so that only six elements need to be determined. Since the matrices in Steps (F) and (G) have zero elements, the amount of computation for this step is only [0M, 11A]. The same happens in Step (I) with the result of Step (E). The result is a general 3×3 matrix requiring [0M, 6A].

To complete the 6×6 congruence transformation when using MDH parameters, the steps in Table 21 are repeated for the x-planar screw. Since α is always constant, the computations for Step (A) may be completed off-line. The rest of the steps have the same computational cost as listed in the table for a total of [42M, 43A], and the complete transformation requires a total of [86M, 89A]. In Chapter 2, a full set of constant MDH parameters is used to transform the operational space inertia matrix, $\boldsymbol{\Lambda}_k$ to the reference member's coordinate system. In this case both θ and α are constant so the computational cost of this step is [84M, 86A]. In Chapter 7, a constant z planar screw transformation is needed to transform the reflected AB inertia matrix, ${}^0\mathbf{N}_{k,1}$, from a chain's base coordinate system to the reference member's coordinate system. This results in a computational cost of [42M, 43A].

A.4.4 Reflected AB Inertia Matrices

In the algorithm developed in Part II, the congruence transformation of the reflected inertia is another computationally expensive step. This step is very similar to the transformation of a symmetric 6×6 matrix that was described in the previous section. The structure of reflected AB inertia matrices contain extra zeros which lead to a more efficient implementation than in the previous section. If joint i is revolute, where $\boldsymbol{\phi}_i = [0\ 0\ 1\ 0\ 0\ 0]^T$, the third row and column of the ith reflected inertia matrix, $\mathbf{N}_i$, are zero as follows:

$$\mathbf{N}_i = \left[\begin{array}{c|c} \mathbf{N}_{11} & \mathbf{N}_{12} \\ \hline \mathbf{N}_{21} & \mathbf{N}_{22} \end{array}\right] \tag{A.29}$$

Table 22: Computational cost for a z-planar congruence transformation of a reflected AB inertia.

		Revolute		Prismatic	
Step	Computation	$\times$	$+$	$\times$	$+$
(A)	$s^2\theta$, $c^2\theta$ - $s^2\theta$, $s\theta c\theta$, $2s\theta c\theta$	2	3	0	0
(B)	$\boldsymbol{R}^T\mathbf{N}_{11}\boldsymbol{R}$	4	5	8	7
(C)	$[\mathbf{X}^T\mathbf{N}\mathbf{X}]_{22} = \boldsymbol{R}^T\mathbf{N}_{22}\boldsymbol{R}$	8	7	4	5
(D)	$\boldsymbol{R}^T\mathbf{N}_{12}\boldsymbol{R}$	8	10	8	10
(E)	$\tilde{\boldsymbol{p}}(\boldsymbol{R}^T\mathbf{N}_{22}\boldsymbol{R})$	5	0	3	0
(F)	$(\tilde{\boldsymbol{p}}\boldsymbol{R}^T\mathbf{N}_{22}\boldsymbol{R})\tilde{\boldsymbol{p}}^T$	3	0	3	0
(G)	$(\boldsymbol{R}^T\mathbf{N}_{12}\boldsymbol{R})\tilde{\boldsymbol{p}}^T$	4	0	6	0
(H)	$[\mathbf{X}^T\mathbf{N}\mathbf{X}]_{11}$ = (B)+(F)+(G)+(G)^T	0	9	0	11
(I)	$[\mathbf{X}^T\mathbf{N}\mathbf{X}]_{12} = [\mathbf{X}^T\mathbf{N}\mathbf{X}]_{21}^T$ = (D)+(E)	0	6	0	4
	Total:	34	40	32	37

$$= \left[\begin{array}{ccc|ccc} n_{11} & n_{12} & 0 & n_{14} & n_{15} & n_{16} \\ n_{12} & n_{22} & 0 & n_{24} & n_{25} & n_{26} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline n_{14} & n_{24} & 0 & n_{44} & n_{45} & n_{46} \\ n_{15} & n_{25} & 0 & n_{45} & n_{55} & n_{56} \\ n_{16} & n_{26} & 0 & n_{46} & n_{56} & n_{66} \end{array}\right]. \tag{A.30}$$

For a prismatic joint, $\boldsymbol{\phi}_i = [0\ 0\ 0\ 0\ 0\ 1]^T$ and so the sixth column and row are zero:

$$\mathbf{N}_i = \left[\begin{array}{ccc|ccc} n_{11} & n_{12} & n_{13} & n_{14} & n_{15} & 0 \\ n_{12} & n_{22} & n_{23} & n_{24} & n_{25} & 0 \\ n_{13} & n_{23} & n_{33} & n_{34} & n_{35} & 0 \\ \hline n_{14} & n_{24} & n_{34} & n_{44} & n_{45} & 0 \\ n_{15} & n_{25} & n_{35} & n_{45} & n_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{A.31}$$

Due to the difference in the structure for revolute and prismatic joints, both cases are covered in this section.

The z-planar congruence transform is performed first, and the same nine steps that were identified in Table 21 of the previous section apply to this computation. These are listed in Table 22 for both types of joints. Because of the extra zeros, Steps (B), (D), (G), and (H) require less computation than the previous section

Table 23: Computational cost for a x-planar congruence transformation of a reflected AB inertia.

		Revolute		Prismatic	
Step	Computation	$\times$	$+$	$\times$	$+$
(A)	$s^2\alpha$, $c^2\alpha$ - $s^2\alpha$, $s\alpha c\alpha$, $2s\alpha c\alpha$	0	0	0	0
(B)	$\boldsymbol{R}^T\mathbf{N}_{11}\boldsymbol{R}$	4	1	8	7
(C)	$[\mathbf{X}^T\mathbf{N}\mathbf{X}]_{22} = \boldsymbol{R}^T\mathbf{N}_{22}\boldsymbol{R}$	8	7	4	1
(D)	$\boldsymbol{R}^T\mathbf{N}_{12}\boldsymbol{R}$	10	6	10	6
(E)	$\tilde{\boldsymbol{p}}(\boldsymbol{R}^T\mathbf{N}_{22}\boldsymbol{R})$	5	0	5	0
(F)	$(\tilde{\boldsymbol{p}}\boldsymbol{R}^T\mathbf{N}_{22}\boldsymbol{R})\tilde{\boldsymbol{p}}^T$	3	0	3	0
(G)	$(\boldsymbol{R}^T\mathbf{N}_{12}\boldsymbol{R})\tilde{\boldsymbol{p}}^T$	6	0	6	0
(H)	$[\mathbf{X}^T\mathbf{N}\mathbf{X}]_{11}$ = (B)+(F)+(G)+(G)^T	0	11	0	11
(I)	$[\mathbf{X}^T\mathbf{N}\mathbf{X}]_{12} = [\mathbf{X}^T\mathbf{N}\mathbf{X}]_{21}^T$ = (D)+(E)	0	6	0	6
	Total:	36	31	36	31

for a revolute joint. For a prismatic joint, Steps (C), (D), (E), and (I) require less computation. The computation requires [34M, 40A] for either type of joint, and more importantly, the zero rows and columns that are shown in Eqs. (A.30) and (A.31) are present in the final result of this step as well.

Because of these zero rows, the computations for the x-planar congruence transformation are listed in Table 23. In this case, the computation for Step (A) can be accomplished off-line as before because α is a constant. Steps (B) and (D) require less computation in the case of a revolute joint, and Steps (C) and (D) require less computation in the case of a prismatic joint. The final result in either case is a full symmetric 6×6 matrix, and the total computational cost is [70M, 71A] for revolute joints and [68M, 68A] for prismatic joints.

A.4.5 Spatial Inertia Matrices

The spatial inertia matrix, which was described in Chapter 1, has a very specific structure which is the same as the Composite Rigid Body (CRB) inertias, $\mathbf{K}_i$, that are used in the computation of the joint space inertia matrices in Chapters 2 and 5. The structure is repeated here for reference:

$$\mathbf{K}_i = \begin{bmatrix} \bar{\boldsymbol{K}}_i^c & \tilde{\boldsymbol{h}}_i^c \\ \left(\tilde{\boldsymbol{h}}_i^c\right)^T & m_i^c \mathbf{1}_3 \end{bmatrix}, \tag{A.32}$$

where $\bar{\boldsymbol{K}}_i^c$ is the moment of inertia (3×3, symmetric matrix), $\boldsymbol{h}_i^c$ is the first moment of mass (3-vector), and m_i^c is the mass (scalar) of the ith composite rigid body. Note that only ten elements are needed to completely specify the corresponding spatial inertia matrix, and this will be used to significantly reduce the cost of a congruence transformation involving it.

First the z planar screw transformation step is written in terms of its Cartesian components as follows:

$$\mathbf{K}^z = \mathbf{X}^T(z,d,\theta)\,\mathbf{K}\,\mathbf{X}(z,d,\theta), \tag{A.33}$$

$$\begin{bmatrix} \bar{\boldsymbol{K}}^z & \tilde{\boldsymbol{h}}^z \\ \left(\tilde{\boldsymbol{h}}^z\right)^T & m^z\mathbf{1}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}^T & \tilde{\boldsymbol{p}}\boldsymbol{R}^T \\ \mathbf{0} & \boldsymbol{R}^T \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{K}}^c & \tilde{\boldsymbol{h}}^c \\ \left(\tilde{\boldsymbol{h}}^c\right)^T & m^c\mathbf{1}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \mathbf{0} \\ \boldsymbol{R}\tilde{\boldsymbol{p}}^T & \boldsymbol{R} \end{bmatrix}, \tag{A.34}$$

$$= \left[\begin{array}{l|l} \boldsymbol{R}^T\bar{\boldsymbol{K}}^c\boldsymbol{R} + \tilde{\boldsymbol{p}}\boldsymbol{R}^T m^c\mathbf{1}_3\boldsymbol{R}\tilde{\boldsymbol{p}}^T + & \boldsymbol{R}^T\tilde{\boldsymbol{h}}^c\boldsymbol{R} + \tilde{\boldsymbol{p}}\boldsymbol{R}^T m^c\mathbf{1}_3\boldsymbol{R} \\ \quad \boldsymbol{R}^T\tilde{\boldsymbol{h}}^c\boldsymbol{R}\tilde{\boldsymbol{p}}^T + \left(\boldsymbol{R}^T\tilde{\boldsymbol{h}}^c\boldsymbol{R}\tilde{\boldsymbol{p}}^T\right)^T & \\ \hline \left(\boldsymbol{R}^T\tilde{\boldsymbol{h}}^c\boldsymbol{R} + \tilde{\boldsymbol{p}}\boldsymbol{R}^T m^c\mathbf{1}_3\boldsymbol{R}\right)^T & \boldsymbol{R}^T m^c\mathbf{1}_3\boldsymbol{R} \end{array}\right], \tag{A.35}$$

where $\boldsymbol{R}$ and $\boldsymbol{p}$ are $\boldsymbol{R}(z,d,\theta)$ and $\boldsymbol{p}_z$, respectively, and defined earlier in this appendix. The (1,1), (1,2), and (2,2) blocks correspond to the moment of inertia, first moment of mass, and mass of the composite rigid body expressed at a new coordinate system and can likewise be completely specified with ten quantities. Each of these blocks is examined individually.

The (2,2) block corresponds to the mass of the composite rigid body and can be simplified to a scalar as follows:

$$m^z = m^c, \tag{A.36}$$

which requires no computation.

The (1,2) block corresponds to the first moment of mass for the composite rigid body. It can be simplified as follows:

$$\tilde{\boldsymbol{h}}^z = \boldsymbol{R}^T(z,\theta)\tilde{\boldsymbol{h}}^c\boldsymbol{R}(z,\theta) + m^c\tilde{\boldsymbol{p}}_z, \tag{A.37}$$

This equation can be expressed in vector form as follows:

$$\boldsymbol{h}^z = \boldsymbol{R}^T(z,\theta)\boldsymbol{h}^c + m^c\boldsymbol{p}_z, \tag{A.38}$$

where the following equivalence theorem was applied to the transformation of a skew symmetric matrix:

$$\tilde{\boldsymbol{a}} = \boldsymbol{R}^T\tilde{\boldsymbol{b}}\boldsymbol{R} \;\rightarrow\; \boldsymbol{a} = \boldsymbol{R}^T\boldsymbol{b}. \tag{A.39}$$

The quantity, $m^c \boldsymbol{p}_z$, has one non-zero element which requires [1M, 0A] to compute, but for revolute joints, it is constant and can be computed off-line. From previous results, [4M, 2A] are required to compute the planar rotation of a Cartesian vector, $\boldsymbol{R}^T(z,\theta)\boldsymbol{h}^c$. The result is obtained by adding these together which only requires [0M, 1A]. For revolute joints, the total cost is [4M, 3A] and for prismatic joints it is [5M, 3A].

The (1,1) block corresponds to the composite rigid body's moment of inertia, and can be simplified to the following:

$$\begin{aligned}\bar{\boldsymbol{K}}^z &= \boldsymbol{R}^T(z,\theta)\bar{\boldsymbol{K}}^c\boldsymbol{R}(z,\theta) + \boldsymbol{R}^T(z,\theta)\tilde{\boldsymbol{h}}^c\boldsymbol{R}(z,\theta)\tilde{\boldsymbol{p}}_z^T + \\ &\qquad \left[\boldsymbol{R}^T(z,\theta)\tilde{\boldsymbol{h}}^c\boldsymbol{R}(z,\theta)\tilde{\boldsymbol{p}}_z^T\right]^T + m^c\tilde{\boldsymbol{p}}_z\tilde{\boldsymbol{p}}_z^T. \qquad \text{(A.40)}\end{aligned}$$

This can be further simplified to the following equation:

$$\begin{aligned}\bar{\boldsymbol{K}}^z &= \boldsymbol{R}^T(z,\theta)\bar{\boldsymbol{K}}^c\boldsymbol{R}(z,\theta) + \left[\boldsymbol{R}^T(\tilde{z},\theta)\boldsymbol{h}^c\right]\tilde{\boldsymbol{p}}_z^T + \\ &\qquad \left\{\left[\boldsymbol{R}^T(\tilde{z},\theta)\boldsymbol{h}^c\right]\tilde{\boldsymbol{p}}_z^T\right\}^T + m^c\tilde{\boldsymbol{p}}_z\tilde{\boldsymbol{p}}_z^T, \qquad \text{(A.41)}\end{aligned}$$

where $\left[\boldsymbol{R}^T(\tilde{z},\theta)\boldsymbol{h}^c\right]$ is the screw symmetric matrix representation of the vector, $\boldsymbol{R}^T(z,\theta)\boldsymbol{h}^c$, which was computed as part of the previous step.

The computational cost of $\boldsymbol{R}^T(z,\theta)\bar{\boldsymbol{K}}^c\boldsymbol{R}(z,\theta)$, the planar congruence transformation of a symmetric 3×3 matrix, is [10M, 10A] when the joint is revolute and θ is variable, or [8M, 7A] when the joint is prismatic. The multiplication of the two skew symmetric matrices, $\left[\boldsymbol{R}^T(\tilde{z},\theta)\boldsymbol{h}^c\right]\tilde{\boldsymbol{p}}_z^T$ only requires [3M, 0A] because only one element of $\tilde{\boldsymbol{p}}_z$ is nonzero. The quantity, $m^c\tilde{\boldsymbol{p}}_z\tilde{\boldsymbol{p}}_z^T$, is constant when the joint is revolute and can be computed off-line; otherwise, it requires [1M, 0A]. Adding these terms together to obtain the final result only requires [0M, 6A] because the intermediate results contain many zero elements. The total computational cost is [13M, 16A] for the case of a revolute joint, and [12M, 13A] for a prismatic joint.

The same steps must be carried out for the x-planar congruence transform as well. For this step, α and $\boldsymbol{p}_x$ are both constant which results in the same amount of computation for either type of joint, and which is less than either of the results for the z-planar congruence transform. The cost for all of the steps is listed in Table 24. For a revolute joint, the transformation of the spatial inertia matrix costs [32M, 35A] and for a prismatic joint it only requires [32M, 32A]

A.5 Summary and Conclusions

In this appendix, the transformation operations used extensively in the multibody dynamics algorithms have been examined. Parameters to specify these transforma-

Table 24: Computation requirements for congruence transformation of a spatial (or CRB) inertia matrix.

	Revolute		Prismatic	
Computation	$\times$	$+$	$\times$	$+$
m^z	—	—	—	—
$\boldsymbol{h}^z$	4	3	5	3
$\bar{\boldsymbol{K}}^z$	13	16	12	13
m^x	—	—	—	—
$\boldsymbol{h}^x$	4	3	4	3
$\bar{\boldsymbol{K}}^x$	11	13	11	13
Total:	32	35	32	32

tions are obtained with the use of modified Denavit-Hartenberg (MDH) notation which is presented first. Use of these parameters are shown in this appendix to yield particularly efficient transformation functions that are used extensively throughout this dissertation. The key to efficient transformations is to decompose them into a series of planar transformations. With this approach, the transformation matrices, $\boldsymbol{R}$ and $\mathbf{X}$, and position vector, $\boldsymbol{p}$, are always represented by a set of scalar parameters and are never explicitly computed. This reduces the amount of computation as well as computer storage requirements. In addition, the set of successive planar operations on the vectors and matrices require fewer operations than the corresponding matrix-vector and matrix-matrix operations if $\boldsymbol{R}$ and $\mathbf{X}$ were computed first.