

## A Dynamic Model Based Robot Arm Selection Criterion

PRASADKUMAR P. BHANGALE, S.K. SAHA and V.P. AGRAWAL

*Department of Mechanical Engineering, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India; E-mail: prasad\_bhangale@rediffmail.com*

(Received: 7 July 2003; accepted in revised form: 26 January 2004)

**Abstract.** The selection of robot manipulator architecture, i.e. the determination of link lengths, their relative orientations, types of joints, e.g. revolute or prismatic, etc., has been largely done so far by experience, intuition and at most based on the kinematic considerations like workspace, manipulability, etc. Dynamics is generally ignored at this stage even though it is widely used for control and simulation. This paper attempts to introduce a criterion based on the dynamics of a robot manipulator, namely, simplicity of the associated generalized inertia matrix (GIM). Since the GIM influences both the control and simulation algorithms significantly, its fast computation and / or making its shape diagonal will certainly enhance the speed, precision, and stability of the robots. Two measures of simplicity, the computation complexity of the GIM in terms of floating point operations and the computer CPU time of an algorithm where the GIM appears, e.g., the inverse dynamics algorithm, are used here to evaluate a robot arm. The proposed criterion is illustrated with two -link robot arms with revolute and prismatic joints and compared with the two commonly used criteria, namely, the workspace, and manipulability. Finally, an example is taken to select an arm from the two spatial robot architectures, RTX and Stanford.

**Keywords:** manipulator, architecture selection, DeNOC, GIM, computational complexity, computation time

### 1. Introduction

A robot is characterized by its degree of freedom, number of joints, type of joints, joint placement and their orientation, link lengths and shapes, which influence its performances, namely, the workspace, manipulability, speed of operation, etc. The speed of operation significantly depends on the complexities of the kinematic and dynamic equations and their computations. The robots with different architectures will have different kinematic and dynamic equations with varying computational complexities. Hence, in order to select a suitable robot, both aspects i.e. kinematics as well as dynamics should be looked into. Generally, kinematic characteristics like workspace [1–3], manipulability [4–6], etc. are considered for the selection of a robot. Dynamics is generally neglected at this stage, even though it is widely used for control [7–12] and simulation of robots [12–17]. In this paper, simplicity of dynamic model and its computations are emphasized, particularly, with respect to the generalized inertia matrix (GIM), which has been proposed as a criterion for robot architecture selection. The simplicity of the GIM is emphasized here because,

for example, if it is diagonal, control becomes decoupled, i.e. each joint motion servo control is not influenced by the servo gains of the other joints. This improves overall robot performance, namely, speed and precision. Besides, the simulation, in which the inversion of the GIM is required, becomes straight forward, as the inversion of a diagonal matrix is just another diagonal matrix whose non-zero elements are the reciprocals of the corresponding non-zero elements of the original matrix.

In order to decide whether the GIM of robot architecture is simple or not, it is essential to have the expressions of each elements of GIM as descriptive as possible. Thus, the concept of the decoupled natural orthogonal compliment (DeNOC) introduced in Saha [18] is used here to derive the dynamic equations of motion. The DeNOC concept allows one not only to write the elements of the GIM in explicit analytical forms, but also for the elements of the matrix of convective inertia (MCI). These forms led to the development of recursive dynamics algorithms for both the inverse and forward dynamics of serial [17, 18], and parallel [19] robotic systems. Using the analytical expressions, one can estimate the computational complexities of the GIM and MCI in terms of floating point operations, which will have bearing on the overall computer CPU time of an algorithm where GIM and MCI appear, e.g., in the inverse dynamics algorithm.

This paper is organized as follows: Section 2 provides an outline of the dynamic modelling using the decoupled natural orthogonal compliment (DeNOC); Section 3 introduces the proposed criteria for robot selection; Section 4 illustrates the selection criterion using 2-link arms; Section 5 considers an example of an arm selection from RTX and Stanford architectures. Finally, conclusions are given in Section 6.

## 2. Dynamic Modelling Using the DeNOC

For an  $n$ -degree of freedom open-loop serial-robot, as shown in Figure 1, if  $m_i$  is the mass of the  $i$ th link and  $\mathbf{I}_i$  denotes the  $3 \times 3$  inertia tensor of the  $i$ th link about its origin point,  $O_i$ , then the uncoupled Newton–Euler equations (NE) governing the motion of the  $i$ th link can be written as

$$\mathbf{M}_i \mathbf{t}_i + \mathbf{W}_i \mathbf{M}_i \mathbf{E}_i \mathbf{t}_i = \mathbf{w}_i \quad (1)$$

where the  $6 \times 6$  mass matrix,  $\mathbf{M}_i$ , the  $6 \times 6$  angular velocity matrix,  $\mathbf{W}_i$ , and the  $6 \times 6$  coupling matrix,  $\mathbf{E}_i$ , for the  $i$ th link are given as [19]

$$\begin{aligned} \mathbf{M}_i &\equiv \begin{bmatrix} \mathbf{I}_i & \mathbf{m}_i \mathbf{d}_i \times \mathbf{1} \\ -\mathbf{m}_i \mathbf{d}_i \times \mathbf{1} & \mathbf{m}_i \mathbf{1} \end{bmatrix} & \mathbf{W}_i &\equiv \begin{bmatrix} \boldsymbol{\omega}_i \times \mathbf{1} & \mathbf{O} \\ \mathbf{O} & \boldsymbol{\omega}_i \times \mathbf{1} \end{bmatrix} \\ \mathbf{E}_i &\equiv \begin{bmatrix} \mathbf{1} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \end{aligned} \quad (2)$$

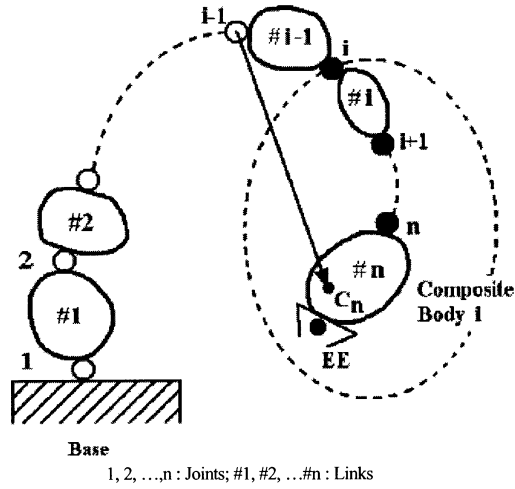


Figure 1. An  $n$ -link manipulator.

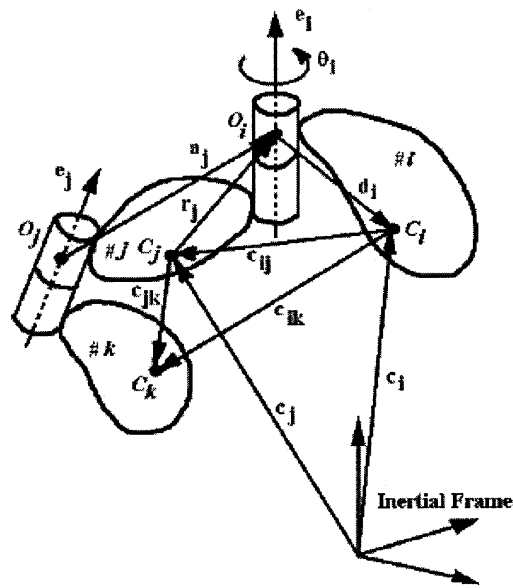


Figure 2. A coupled system of three bodies.

in which  $I_i \equiv I_i^c - \mathbf{d}_i \times (\mathbf{m}_i \mathbf{d}_i \times \mathbf{1}) - I_i^c$  being the  $3 \times 3$  inertia tensor about the mass centre of the  $i$ th body,  $C_i$ ,  $\mathbf{d}_i \times 1$  and  $\omega_i \times 1$  are the  $3 \times 3$  cross-product tensors associated with the vector  $\mathbf{d}_i$  shown in Figure 2, and the angular velocity vector,  $\omega_i$  respectively. Note that  $\mathbf{1}$  and  $\mathbf{0}$  in Equation (2) are the  $3 \times 3$  identity and zero matrices, respectively. The twist and wrench vectors,  $\mathbf{t}_i$  and  $\mathbf{w}_i$ , are then defined as

$$\mathbf{t}_i \equiv \begin{bmatrix} \omega_i \\ \mathbf{v}_i \end{bmatrix} \quad \text{and} \quad \mathbf{w}_i \equiv \begin{bmatrix} \mathbf{n}_i \\ \mathbf{f}_i \end{bmatrix} \quad (3)$$

where  $\omega_i$  and  $v_i$  are the three-dimensional vectors of angular velocity and the linear velocity of the origin point  $O_i$ , of the  $i$ th body, where it is coupled with its previous body in the chain, i.e., the  $j$ th body, respectively, Figure 2. Moreover,  $\mathbf{n}_i$  and  $\mathbf{f}_i$  are the three-dimensional vectors denoting the resultant moment about  $O_i$ , and the resultant forces acting at  $O_i$ , respectively. Note that  $\dot{\mathbf{t}}_i$  of Equation (1) is the time derivative of the twist vector or twist-rate vector. Equation (1) when written for all ' $n$ ' links, i.e.,  $i = 1, \dots, n$ , it can be expressed in a compact form as

$$\mathbf{M}\dot{\mathbf{t}} + \mathbf{W}\mathbf{M}\mathbf{E}\mathbf{t} = \mathbf{w} \quad (4)$$

where  $\mathbf{M}$  and  $\mathbf{W}$  are the  $6n \times 6n$  generalized mass matrix and the generalized matrix of the angular velocities, respectively, whereas  $\mathbf{E}$  is the  $6n \times 6n$  generalized coupling matrix. They are defined as follows:

$$\mathbf{M} \equiv \begin{bmatrix} \mathbf{M}_1 & \mathbf{O} & \dots & \mathbf{O} \\ \mathbf{O} & \mathbf{M}_2 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \dots & \dots & \mathbf{M}_n \end{bmatrix}; \mathbf{W} \equiv \begin{bmatrix} \mathbf{W}_1 & \mathbf{O} & \dots & \mathbf{O} \\ \mathbf{O} & \mathbf{W}_2 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \dots & \dots & \mathbf{W}_n \end{bmatrix};$$

$$\mathbf{E} \equiv \begin{bmatrix} \mathbf{E}_1 & \mathbf{O} & \dots & \mathbf{O} \\ \mathbf{O} & \mathbf{E}_2 & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \dots & \dots & \mathbf{E}_n \end{bmatrix} \quad (5)$$

Moreover, the  $6n$ -dimensional vectors of generalized twist and wrench are defined as

$$\mathbf{t} \equiv [\mathbf{t}_1^T, \mathbf{t}_2^T, \dots, \mathbf{t}_n^T]^T \quad \mathbf{w} \equiv [\mathbf{w}_1^T, \mathbf{w}_2^T, \dots, \mathbf{w}_n^T]^T \quad (6)$$

It is pointed out here that the generalized twist,  $\mathbf{t}$ , can be expressed as a linear transformation of the  $n$  independent joint rates,  $\dot{\boldsymbol{\theta}} \equiv [\dot{\theta}_1, \dots, \dot{\theta}_n]^T$ — $\theta_i$  being the  $i$ th joint displacement shown in Figure 2, i.e.,

$$\mathbf{t} = \mathbf{T}\dot{\boldsymbol{\theta}}, \quad \text{where } \mathbf{T} \equiv \mathbf{T}_1\mathbf{T}_d \quad (7)$$

$\mathbf{T}$  being the  $6n \times n$  natural orthogonal complement (NOC) matrix [7], whereas the  $6n \times 6n$  and  $6n \times n$  matrices,  $\mathbf{T}_1$  and  $\mathbf{T}_d$ , respectively, are the decoupled NOC (DeNOC) matrices [18].

Premultiplying Equation (4) with the transpose of the NOC matrix, i.e.,  $\mathbf{T}^T$ , one gets  $n$  independent dynamic equations of motion of the coupled system, namely,

$$\mathbf{T}^T(\mathbf{M}\dot{\mathbf{t}} + \mathbf{W}\mathbf{M}\mathbf{E}\mathbf{t}) = \mathbf{T}^T(\mathbf{w}^E + \mathbf{w}^C) \quad (8)$$

where  $\mathbf{w}$  is substituted by  $\mathbf{w} \equiv \mathbf{w}^E + \mathbf{w}^C$ ,  $\mathbf{w}^E$  and  $\mathbf{w}^C$  being the  $6n$ -dimensional vectors of external and constraint wrenches, respectively. The term  $\mathbf{T}^T \mathbf{w}^C$  in Equation (8) vanishes, as the constraint wrench produces no work. Substitution of the expression of  $\mathbf{T} \equiv \mathbf{T}_1 \mathbf{T}_d$  from Equation (7) and its time derivative,  $\dot{\mathbf{T}} = \dot{\mathbf{T}}_1 \mathbf{T}_d + \mathbf{T}_1 \dot{\mathbf{T}}_d$ , into Equation (8), results in the following form of the dynamic equations of motion:

$$\mathbf{I} \ddot{\boldsymbol{\theta}} + \mathbf{C} \dot{\boldsymbol{\theta}} = \boldsymbol{\tau} \quad (9)$$

which is nothing but the Euler–Lagrange equations of motion [7]. In Equation (9),

$\mathbf{I} \equiv \mathbf{T}^T \mathbf{M} \mathbf{T} \equiv \mathbf{T}_d^T \tilde{\mathbf{M}} \mathbf{T}_d$ : the  $n \times n$  generalized inertia matrix (GIM);

$\mathbf{C} \equiv \mathbf{T}^T (\mathbf{M} \dot{\mathbf{T}} + \mathbf{W} \mathbf{M} \mathbf{E} \mathbf{T}) \equiv \mathbf{T}_d^T (\mathbf{T}_1^T \mathbf{M} \dot{\mathbf{T}}_1 + \tilde{\mathbf{M}} \mathbf{W} + \tilde{\mathbf{M}}) \mathbf{T}_d$ : the  $n \times n$  matrix of convective inertia (MCI);

$\boldsymbol{\tau} \equiv \mathbf{T}^T \mathbf{w}^E \equiv \mathbf{T}_d^T \tilde{\mathbf{w}}^E$ : the  $n$ -dimensional vector of generalized forces due to driving forces / torques and those resulting from the gravity and dissipation.

The  $6n \times 6n$  matrices,  $\tilde{\mathbf{M}}$ ,  $\tilde{\mathbf{M}}$  and the  $6n$ -dimensional vector  $\tilde{\mathbf{w}}^E$  are given by

$$\tilde{\mathbf{M}} \equiv \mathbf{T}_1^T \mathbf{M} \mathbf{T}_1 \quad \tilde{\mathbf{M}} \equiv \mathbf{T}_1^T \mathbf{W} \mathbf{M} \mathbf{E} \mathbf{T}_1 \quad \text{and} \quad \tilde{\mathbf{w}}^E \equiv \mathbf{T}_1^T \mathbf{w}^E.$$

### 3. Dynamics Simplicity Based Selection Criterion

In this section, explicit expressions for the elements of the GIM and MCI, i.e.,  $\mathbf{I}$  and  $\mathbf{C}$  of Equation (9), respectively, are derived first. Next, their simplicity is proposed to be measured in terms of their computational complexity and the computer CPU time of the algorithm where they appear.

#### 3.1. EXPLICIT EXPRESSIONS FOR THE ELEMENTS OF THE GIM AND MCI

Using the expressions for the  $n \times n$  GIM,  $\mathbf{I}$  and, MCI,  $\mathbf{C}$ , as appearing after Equation (9), each element of  $\mathbf{I}$ ,  $i_{ij}$ , and  $\mathbf{C}$ ,  $c_{ij}$ , are written as

$$\begin{aligned} i_{ij} &\equiv \mathbf{p}_i^T \tilde{\mathbf{M}}_i \mathbf{A}_{ij} \mathbf{p}_j \\ c_{ij} &\equiv \mathbf{p}_i^T (\mathbf{A}_{ji}^T \tilde{\mathbf{M}}_j \mathbf{W}_j + \mathbf{A}_{j+1,i}^T \tilde{\mathbf{H}}_{j+1,j} + \mathbf{A}_{ji}^T \tilde{\mathbf{M}}_j) \mathbf{p}_j \quad \text{if } i \leq j \\ c_{ij} &\equiv \mathbf{p}_i^T (\tilde{\mathbf{M}}_i \mathbf{A}_{ij} \mathbf{W}_j + \tilde{\mathbf{H}}_{ij} + \tilde{\mathbf{M}}_i \mathbf{A}_{ij}) \mathbf{p}_j \quad \text{otherwise} \end{aligned} \quad (10)$$

where, the  $6 \times 6$  matrix  $\mathbf{A}_{ij}$  and the six-dimensional vector,  $\mathbf{p}_i$ , are the block elements of the DeNOC matrices,  $\mathbf{T}_1$  and  $\mathbf{T}_d$ , respectively, i.e.,

$$\mathbf{T}_1 \equiv \begin{bmatrix} \mathbf{1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{1} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{n,1} & \mathbf{A}_{n,2} & \dots & \mathbf{1} \end{bmatrix} \quad \text{and} \quad \mathbf{T}_d \equiv \begin{bmatrix} \mathbf{p}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{p}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{p}_n \end{bmatrix} \quad (11)$$

In Equation (11),  $\mathbf{O}$  and  $\mathbf{0}$  are the  $6 \times 6$  zero matrix and six-dimensional zero vector, whereas the matrix,  $\tilde{\mathbf{H}}_{ij}$ , in the expression of  $c_{ij}$ , is given by

$$\tilde{\mathbf{H}}_{ij} \equiv \tilde{\mathbf{M}}_i \dot{\mathbf{A}}_{ij} + \mathbf{A}_{i+1,i}^T \tilde{\mathbf{H}}_{i+1,i} \mathbf{A}_{ij} \quad (12)$$

in which  $\dot{\mathbf{A}}_{ij}$  is the time derivative of the  $6 \times 6$  matrix,  $\mathbf{A}_{ij}$ . Note that the  $6 \times 6$  matrix,  $\tilde{\mathbf{M}}_i$ , has the following features:

- i. Matrix  $\tilde{\mathbf{M}}_i$  can be recursively computed as

$$\tilde{\mathbf{M}}_i \equiv \mathbf{M}_i + \mathbf{A}_{i+1,i}^T \tilde{\mathbf{M}}_{i+1} \mathbf{A}_{i+1,i} \quad \text{for } i = n, n-1, \dots, 1 \quad (13)$$

where  $\tilde{\mathbf{M}}_n \equiv \mathbf{M}_n$  since there is no  $(n+1)$ st link, i.e.,  $\tilde{\mathbf{M}}_{n+1} \equiv \mathbf{O}$ , and

$$\tilde{\mathbf{M}}_{n-1} \equiv \mathbf{M}_{n-1} + \mathbf{A}_{n,n-1}^T \tilde{\mathbf{M}}_n \mathbf{A}_{n,n-1}$$

- ii. Matrix  $\tilde{\mathbf{M}}_i$  has a physical interpretation. It represents the mass matrix of the “composite body  $i$ ,” formed by rigidly joining the bodies,  $i, \dots, n$ , as indicated in Figure 1.

### 3.2. COMPUTATIONAL COMPLEXITY OF GIM AND MCI

One of the ways to simplify the GIM and MCI is to make some of the elements vanish, or constant with suitable choice of the link masses and geometries. For example, if any two successive, say,  $i$ th and  $(i+1)$ st joints are prismatic, orthogonal and intersecting, the corresponding off-diagonal inertia element, namely,  $i_{i+1,i} \equiv \mathbf{P}_{i+1}^T \tilde{\mathbf{M}}_{i+1} \mathbf{A}_{i+1,i} \mathbf{p}_i$ , can be proven to be zero. In case the elements do not vanish, their computational complexity in terms of the floating point operations, namely, the number of multiplications/divisions ( $M$ ) and additions/subtractions ( $A$ ) are counted. For the GIM and MCI, the complexities are obtained for the following input: For  $i = 1, \dots, n$ ,

- i. *Constant* Denavit–Hartenberg (D-H) parameters [20] of the system under study, i.e.,  $\alpha_i$ ,  $a_i$ , and  $b_i$  (for revolute joints) or  $\theta_i$  (for prismatic joints). They are also defined in Appendix A.
- ii. Time history of *variable* D–H parameter, i.e.,  $\theta_i$ , for a revolute pair, and  $b_i$ , for a prismatic joint, and their first and second time derivatives.
- iii. Mass of each body,  $m_i$ .
- iv. Vector denoting the distance of the  $(i+1)$ st joint from the  $i$ th mass centre,  $C_i$ , in the  $(i+1)$ st frame, i.e.,  $[\mathbf{r}_1]_{i+1}$ , as shown in Figure 2.
- v. Inertia tensor of the  $i$ th link about its mass centre,  $C_i$ , in the  $(i+1)$ st frame.

Note here that  $[-]_{i+1}$  will represent the argument “-” in the  $(i + 1)^{\text{st}}$  coordinate frame which is rigidly attached to the  $i$ th link. Thus, the quantities,  $[\mathbf{I}_i]_{i+1}$  and  $[\mathbf{r}_i]_{i+1}$  are constant and supplied as input. Moreover, the complexity count is given inside  $\{ \}$  at the right of each step and substep, which is based on the assumption that all joints of the robot are revolute with arbitrary orientation.

1. Calculate  $[\mathbf{a}_i]_i$  (Angeles, 1997)<sup>21</sup>:  $\{2M(n)\}$

For  $i = 1, \dots, n$ ,

$$[\mathbf{a}_i]_i = [\mathbf{a}_i \cos \theta_i \ a_i \sin \theta_i \ b_i]^T \quad :2M$$

Note that, for each  $[\mathbf{a}_i]_i$ ,  $2M$  computations are required. Hence, for all  $n$  links,  $2M(n)$  computations will be required to find  $[\mathbf{a}_i]_i$ , for  $i = 1, \dots, n$ , that appears above within the braces,  $\{2M(n)\}$ .

2. Calculate  $[\mathbf{r}_i]_i$ :  $\{8M \ 4A(n)\}$

For  $i = 1, \dots, n$ ,

$$[\mathbf{r}_i]_i = \mathbf{Q}_i[\mathbf{r}_i]_{i+1} \quad :8M4A$$

where  $\mathbf{Q}_i$  is given as

$$\mathbf{Q}_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{bmatrix}$$

Note above that the multiplication of the  $3 \times 3$  matrix,  $\mathbf{Q}_i$  or  $\mathbf{Q}_i^T$  with a three-dimensional vector is obtained efficiently with only  $8M \ 4A$ , instead of  $9M \ 6A$  required to multiply a  $3 \times 3$  matrix with a three-dimensional vector, as shown in Appendix B.1.

3. Calculate  $[\mathbf{d}_i]_i$ :  $\{3A(n)\}$

For  $i = 1, \dots, n$ ,

$$[\mathbf{d}_i]_i = [\mathbf{a}_i]_i - [\mathbf{r}_i]_i \quad :3A$$

4. Calculate  $[\mathbf{m}_i \mathbf{d}_i]_i$ :  $\{3M(n)\}$

For  $i = 1, \dots, n$ ,

$$[\mathbf{m}_i \mathbf{d}_i]_i = m_i[\mathbf{d}_i]_i \quad :3M$$

5. a) Calculate  $[\mathbf{M}_i]_i$ , for  $i = n$ :  $\{22M \ 26A(n)\}$

$$[\mathbf{M}_i]_i = \begin{bmatrix} \mathbf{I}_i^c - \mathbf{d}_i \times (\mathbf{m}_i \mathbf{d}_i \times \mathbf{1}) & \mathbf{m}_i \mathbf{d}_i \times \mathbf{1} \\ -\mathbf{m}_i \mathbf{d}_i \times \mathbf{1} & \mathbf{m}_i \mathbf{1} \end{bmatrix}$$

$$\begin{aligned}
[\mathbf{I}_i^c]_i &= \mathbf{Q}_i[\mathbf{I}_i^c]_{i+1}\mathbf{Q}_i^T & :16M17A \\
[\mathbf{d}_i \times (\mathbf{m}_i\mathbf{d}_i \times \mathbf{1})]_i & & :6M3A \\
[m_i\mathbf{d}_i \times \mathbf{1}]_i & & :nil \\
[\mathbf{I}_i^c - \mathbf{d}_i \times (\mathbf{m}_i\mathbf{d}_i \times \mathbf{1})]_i & & :6A
\end{aligned}$$

Like in item 2 above, two multiplications of an orthogonal matrix,  $\mathbf{Q}_i$ , with a symmetric matrix,  $[\mathbf{I}_i^c]_{i+1}$  and another orthogonal matrix, namely,  $\mathbf{Q}_i^T$ , can be computed with  $16M 17A$  instead of two matrix multiplications requiring  $54M 36A$ , as explained in Appendix B.2.

b)  $[\tilde{\mathbf{M}}_i]_i$  for  $i = n - 1, \dots, 1$   $\{18M18A(n - 1)\}$

$$\tilde{\mathbf{M}}_i \equiv \mathbf{M}_i + \mathbf{A}_{i+1,i}^T \tilde{\mathbf{M}}_{i+1} \mathbf{A}_{i+1,i} \quad :18M18A$$

6. a) Calculate  $i_{ii}$  :

For  $i = n, \dots, 1$   $\{nil\}$

$$i_{ii} \equiv \mathbf{p}_i^T \tilde{\mathbf{M}}_i \mathbf{p}_i \quad :nil$$

For a revolute joint,

$$i_{ii} = [\mathbf{e}_i^T \quad \mathbf{0}^T]_i [\tilde{\mathbf{M}}_i]_i \begin{bmatrix} \mathbf{e}_i \\ \mathbf{0} \end{bmatrix}_i$$

Since  $[\mathbf{e}_i]_i = [001]^T$ , the calculation of  $i_{ii}$  is simply the 3,3 element of the (1,1) block matrix of  $[\tilde{\mathbf{M}}_i]_i$ , which do not require any computation.

b) Calculate  $i_{ij}$  :  $\{22M14A(n^2 - n)/2\}$

For  $i = n, \dots, 1; j = i - 1, \dots, 1$

$$i_{ij} = \mathbf{p}_i^T \tilde{\mathbf{M}}_i \mathbf{A}_{ij} \mathbf{p}_j \quad :22M14A$$

So the total number of multiplications/divisions and additions/subtractions are computed by adding the values inside the symbols, { and }, as shown in Table 1.

The computation complexity to find the GIM is

$$(11n^2 + 42n - 18)M(7n^2 + 44n - 18)A. \quad (14)$$

For six revolute jointed robot ( $n = 6$ ), the complexity for the GIM is  $630M 498 A$ . Note that the complexity of Equation (14) is less than that reported earlier in [14], which is

$$(12n^2 + 56n - 27)M(7n^2 + 67n - 53)A. \quad (15)$$



Table I. The GIM computational complexity for spatial robots.

Steps	Variables	Multiplications	Additions
1	$[\mathbf{a}_i]_i, i = 1, \dots, n$	$2n$	–
2	$[\mathbf{r}_i]_i, i = 1, \dots, n$	$8n$	$4n$
3	$[\mathbf{d}_i]_i, i = 1, \dots, n$	–	$3n$
4	$[\mathbf{m}_i \mathbf{d}_i]_i, i = 1, \dots, n$	$3n$	–
5(a)	$[\mathbf{M}_i]_i, i = n, \dots, 1$	$22n$	$26n$
5(b)	$[\tilde{\mathbf{M}}_i]_i, \text{ for } i = n - 1, \dots, 1$	$18(n - 1)$	$18(n - 1)$
6(a)	$i_{ii}, i = n, \dots, 1$	–	–
6(b)	$i_{ij}, i = n, \dots, 1; j = i - 1, \dots, 1$	$22(n^2 - n)/2$	$14(n^2 - n)/2$
	Total	$11n^2 + 42n - 18$	$7n^2 + 44n - 18$

For  $n = 6$ , Equation (15) gives  $741M \ 601A$ . Thus an efficient algorithm for the GIM calculations is proposed in this paper. Similarly, the computational complexity for the matrix of convective inertia (MCI) terms is also found out, which is

$$(14n^2 + 22n + 4)M(13.5n^2 + 55.5n - 65.5)A. \tag{16}$$

Again, for  $n = 6$  with all revolute joints, the MCI complexity is  $640M \ 753.5A$ , which could not be compared due to non-availability of such results.

### 3.2.1. Planar GIM and MCI Complexity

For planar robots, for example, as shown in Figure 3, the computational complexity counts in Equations (14) and (15) resulting in very high values as many zero components of the two-dimensional vector and matrix elements will be processed / counted. Hence, an alternative simplified algorithm is proposed.

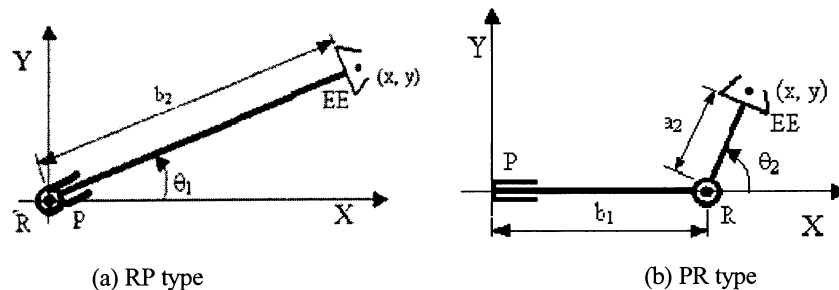


Figure 3. Two link planar arms.

1. Calculate 2-dimensional  $[\mathbf{a}_i]_i$ :  $\{2M(n)\}$   
For  $i = 1, \dots, n$ ,

$$[\mathbf{a}_i]_i = [a_i \cos \theta_i \quad a_i \sin \theta_i]^T \quad : 2M$$

2. Calculate 2-dimensional  $[\mathbf{r}_i]_i$ :  $\{4M \ 2A(n)\}$   
For  $i = 1, \dots, n$ ,

$$[\mathbf{r}_i]_i = \mathbf{Q}_i [\mathbf{r}_i]_{i+1} \quad : 4M2A$$

where  $\mathbf{Q}_i$  is given as

$$\mathbf{Q}_i \equiv \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix}$$

3. Calculate 2-dimensional  $[\mathbf{d}_i]_i$ :  $\{2A(n)\}$   
For  $i = 1, \dots, n$ ,

$$[\mathbf{d}_i]_i = [\mathbf{a}_i]_i - [\mathbf{r}_i]_i \quad : 2A$$

4. Calculate 2-dimensional  $[m_i \mathbf{d}_i]_i$ :  $\{2M(n)\}$   
For  $i = 1, \dots, n$ ,

$$[m_i \mathbf{d}_i]_i = m_i [\mathbf{d}_i]_i \quad : 2M$$

5. a) Calculate  $3 \times 3$  matrix,  $[\mathbf{M}_i]_i$ , for  $i = n$  :  $\{\text{nil}(n)\}$

$$[\mathbf{M}_i]_i \equiv \begin{bmatrix} \mathbf{i}_{zz}^c & m_i \mathbf{d}_i^T \mathbf{R} \\ m_i \mathbf{R} \mathbf{d}_i & m_i \mathbf{1} \end{bmatrix}$$

where the scalar,  $\mathbf{i}_{zz}^c$ , is the (3,3) element of  $\mathbf{I}_i^c$  that only contribute to the planar motion, which do not require any computation as it is one of the inputs item (v) of Section 3.2. Moreover, for planar motion  $m_i d_i \times \mathbf{1}$  can be shown to be equivalent to  $m_i \mathbf{d}_i^T \mathbf{R}$  [22]. The  $2 \times 2$  matrix  $\mathbf{R}$  is defined as  $\mathbf{R} \equiv \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ . Note,  $m_i \mathbf{d}_i^T \mathbf{R}$  is known from the observations. Hence, no computation will be required for  $[\mathbf{M}_i]_i$ .

- b) The  $3 \times 3$  matrix  $[\tilde{\mathbf{M}}_i]_i$  for  $i = n - 1, \dots, 1$  :  $\{7M7A(n - 1)\}$

$$\tilde{\mathbf{M}}_i \equiv \mathbf{M}_i + \mathbf{A}_{i+1,i}^T \tilde{\mathbf{M}}_{i+1} \mathbf{A}_{i+1,i} \quad : 7M7A$$

6. a) Calculate  $i_{ii}$  :  $\{\text{nil}\}$   
For  $i = 1, \dots, n$ ,

$$i_{ii} \equiv \mathbf{p}_i^T \tilde{\mathbf{M}}_i \mathbf{p}_i \quad : \text{nil}$$

Table II. The GIM computational complexity in planar robots.

Steps	Variables	Multiplications	Additions
1	$[\mathbf{a}_i]_i, i = 1, \dots, n$	$2n$	–
2	$[\mathbf{r}_i]_i, i = 1, \dots, n$	$4n$	$2n$
3	$[\mathbf{d}_i]_i, i = 1, \dots, n$	–	$2n$
4	$[\mathbf{m}_i \mathbf{d}_i]_i, i = 1, \dots, n$	$2n$	–
5(a)	$[\mathbf{M}_i]_i, i = n, \dots, 1$	–	–
5(b)	$[\tilde{\mathbf{M}}_i]_i, i = n - 1, \dots, 1$	$7(n - 1)$	$7(n - 1)$
6(a)	$\mathbf{i}_{ii}, i = 1, \dots, n$	–	–
6(b)	$\mathbf{i}_{ij}, i = n, \dots, 1; j = i - 1, \dots, 1$	$7(n^2 - n)/2$	$4(n^2 - n)/2$
	Total	$3.5n^2 + 11.5n - 7$	$2n^2 + 9n - 7$

7. Calculate  $i_{ij}$ :  $\{7M4A(n^2 - n)/2\}$   
 For  $i = n, \dots, 1; j = i - 1, \dots, 1$

$$i_{ij} \equiv \mathbf{p}_i^T \tilde{\mathbf{M}}_i \mathbf{A}_{ij} \mathbf{p}_j \quad : 7M4A$$

The total computational complexities in the planar case can be obtained by adding the above values inside and, as given in Table II.

The computation complexity to find GIM for planar robot is

$$\text{GIM} : (3.5n^2 + 11.5n - 7)M(2n^2 + 9n - 7)A \quad (17)$$

whereas the complexity for MCI can be obtained as

$$\text{MCI} : (7n^2 + 13n + 4)M(4n^2 + 13n + 2)A. \quad (18)$$

For  $n = 3$  with all revolute joints,  $59M 38A$  computations are required for the GIM, and  $106M 77A$  are required for the MCI, whereas if the spatial algorithm of Equations (14) and (16) are used, for  $n = 3$ , then  $207M 177A$ , and  $196M 222.5A$  are required for the GIM and MCI, respectively, which are high and not recommended for planar robots.

### 3.3. COMPUTATIONAL TIME FOR INVERSE DYNAMICS

The computational time is the time taken by the computer of the robot controller to evaluate or solve the equations of motions, which depends on the computational complexity of the associated equations. If the computational complexities of the GIM and MCI vary for different robots, they should reflect in the computational times of the algorithm where they are used. As a test, inverse dynamics algorithms based on the GIM and MCI elements obtained in Section 3.1 are developed for

different architectures. Computer CPU times are recorded along with the computational complexities of GIM and MCI. Here, the CPU time is basically the time taken to compute the controlling joint torques/forces from the inverse dynamics algorithms. The computer CPU will take comparatively less time to solve the inverse dynamics equations if they are simpler. Less computation time implies that the maximum speed of a robot arm can be high. Thus the manipulator with least computation time is preferable over others.

Though the CPU time difference for various architectures will look miniscule here, there will be significant difference when the algorithms run on robot controllers with processing speeds 30 to 60 times slower than the computer (Pentium III, 1Ghz with 256 Mb memory) used here.

#### 4. Illustration with Planar 2 -Link Robots

A planar two-degree of freedom (DOF) robot which has two links and two joints, is suitable for planar positioning purposes. There are three options for such robot arm, namely, two links are connected by 1) two prismatic joints; 2) two revolute joints; and 3) one revolute and the other one prismatic joint. On the basis of proposed criterion, i.e., simplicity, option 1 is best as its GIM can be easily shown to be the  $2 \times 2$  diagonal matrix with constant elements, whereas its MCI elements are zeros. Even though this architecture is simplest from the control point of view, it is not preferred from the physical realization angle. Prismatic joints can be realized using nut and lead screw, which are expensive for higher accuracy and difficult to maintain as dust can accumulate if proper bellows, etc. are not fitted. To overcome the above drawbacks, revolute joints are used where the rotary actuators can be coupled directly to the links without any nut, ball screw and bellows. This is option 2 which is economic from physical realization point of view but computationally, i.e., in terms of the GIM complexity etc., it is most expensive. What left is option 3, which is a kind of tradeoff between options 1 and 2. The criterion proposed in this paper is effective for such situation when one has to choose the first joint as revolute and the second one as prismatic (RP), Figure 3(a), or vice-versa, i.e., PR type, Figure 3(b).

##### 4.1. GIM AND MCI COMPLEXITY

On the basis of the planar computational complexities, Equations (17) and (18), a planar robot with both revolute joints would require  $88M$ ,  $63A$  (with  $n = 2$ ). However, if one of the joint is prismatic further simplification is expected. The degree of the simplification would depend on the location of prismatic joint. The results are tabulated in Table III.

##### 4.2. COMPUTATIONAL TIME FOR INVERSE DYNAMICS

As explained in Section 3.3, inverse dynamics algorithms for RP and PR architectures shown in Figures 3(a), and 3(b), respectively using pre-determined joint

Table III. Comparison of 2-link robot arms.

Type	RP Robot [Figure 3(b)]	PR Robot [Figure 3(b)]
GIM	$12M \ 6A$	$21M \ 6A$
CC	$26M \ 14A$	$28M \ 18A$
CT* (sec)	$0.651$	$0.761$
Workspace	$\pi(2a_1b_2 + b_2^2)$	$\pi a_2^2 + 2a_2b_1$
area 'A'	$a_i = b_i = a$ $4\pi a^2$	$(\pi + 2)a^2$
Manipulability	$a_1 + b_2$	$a_2 \cos \theta_2 $
'w'	$a_i = b_i = a$ Max. a	Max. a

\*All computations are done on a Pentium-III PC with processor speed of 1GHz with memory size 256Mb.

*M*: Multiplication/division; *A*: Addition/subtraction. CC: Computational complexity; CT: Computation time; \_\_\_\_: Better value of the two.

trajectory values, i.e., joint position, velocity, and acceleration, are implemented in MATLAB. The time taken to compute the joint torques/forces are obtained using 'tic' and 'toc' commands of MATLAB at the beginning and end of an algorithm, respectively. The joint trajectories and other parameters for inverse dynamics are taken as:

For RP type:

$$\begin{aligned}
 & \text{1st link length, } a_1 = 0; \text{ 1st link mass, } m_1 = 12.25 \text{ kg;} \\
 & \text{2nd link mass, } m_2 = 2.45 \text{ kg;} \\
 & \text{1st joint trajectory, } \theta_1 = (\pi t/T) - 0.5 \sin(2\pi t/T) \text{ rad;} \\
 & \text{2nd joint trajectory, } b_2 = (\pi t/T) - 0.5 \sin(2\pi t/T) \text{ m}
 \end{aligned} \tag{19}$$

where  $T$  and  $t$  are the total time of traverse and the time instant at which the results are desired.

For PR type:

$$\begin{aligned}
 & \text{1st link mass, } m_1 = 12.25 \text{ kg; 2nd link length, } a_2 = 0.8584 \text{ m;} \\
 & \text{2nd link mass, } m_2 = 2.45 \text{ kg;} \\
 & \text{1st joint trajectory, } b_1 = (\pi t/T) - 0.5 \sin(2\pi t/T) \text{ m;} \\
 & \text{2nd joint trajectory, } \theta_2 = (\pi t/T) - 0.5 \sin(2\pi t/T) \text{ rad}
 \end{aligned} \tag{20}$$

Computation times to obtain the necessary joint torques and forces using Equation (9) are reported in Table I, where no gravity is considered as for the motion on a horizontal plane gravity does not have any effect.

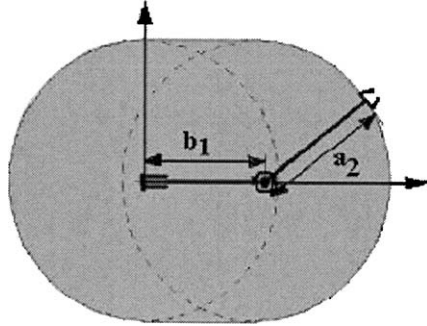


Figure 4. The workspace of 2-link PR-robot arm.

#### 4.3. WORKSPACE AND MANIPULABILITY

Workspace and manipulability are two kinematics based criteria often used for robot architecture selection. Here, they are included to show whether an architecture would differ, if proposed dynamics model based selection criterion is used. Workspace or work envelope is defined as the set of points representing the maximum and minimum extent or reach of the manipulator in all directions. It can be calculated by visualizing the possible motion of the robot in all directions and finding the area, in the case of planar, and volume, in the case of spatial [3, 11]. The workspace area for the PR type robot arm can be obtained from the motion range of the robot, as shown in Figure 4. If no limit on the joints are assumed the workspace area of PR arm in Figure 4 can be calculated as,  $A = \pi a_2^2 + 2a_2b_1$ . Similarly, for the RP type the workspace area is,  $A = \pi(2a_1b_2 + b_2^2)$ . The values are tabulated in Table I and compared considering  $a_i = b_i = a$  for  $i = 1, 2$ .

Manipulability is defined as the ease with which a manipulator operates in its workspace. There are many ways to quantify the manipulability. For example, condition number of the velocity Jacobian matrix, minimum or maximum singular value of the Jacobian, etc. The one proposed by Yoshikawa [4] is used here. This is called manipulability measure,  $w$ , and defined as

$$w = |[\det\{\mathbf{J}(\boldsymbol{\theta})\mathbf{J}^T(\boldsymbol{\theta})\}]^{1/2}| \quad (21)$$

where  $\mathbf{J}(\boldsymbol{\theta})$  is the Jacobian matrix, which is a function of joint positions,  $\boldsymbol{\theta}$ . For any robot,  $w \geq 0$ . However, the value of  $w = 0$  is undesirable as it implies that the robot arm is in the singular configuration, i.e., the arm cannot reach the instructed position. When an arm is in singular configuration, it loses one or more degree of freedom and becomes difficult to control.

Manipulability measure ' $w$ ' for the RP and PR arms are calculated, which are tabulated in Table III. As an illustration, ' $w$ ' for the PR robot is computed as,

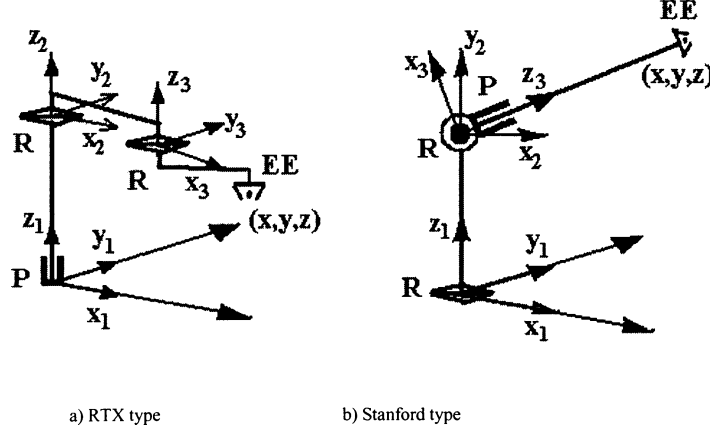


Figure 5. Three link spatial robot arms.

$w = a_2 |\cos \theta_2|$  where the  $\mathbf{J}(\theta)$  of the PR arm is given below:

$$\mathbf{J}(\theta) = \begin{bmatrix} 1 & -a_2 \sin \theta_2 \\ 0 & a_2 \cos \theta_2 \end{bmatrix} \quad (22)$$

As  $\theta_2$  varies,  $w$  changes from 0 to  $a_2$ .

On the basis of the proposed criterion, i.e., the simplicity of dynamics model that has been evaluated in terms of computation complexity (CC) and computation time (CT), the RP type is found better. The choice which is in line with kinematics based workspace criterion when  $a_i = b_i = a$ , as in Table III. However, there may be cases when the workspace of PR may be larger, for example, if  $a_i = 2b; b_i = b$ , then the workspaces of RP and PR robots are,  $5\pi b^2$  and  $4(\pi + 1)b^2$ , respectively. In such situations, proposed dynamic based choice will provide a robot selection that will allow faster robot motion, as explained in Section 3.3.

### 5. Example: Selection Between RTX and Stanford Arms

In this section, assume a situation where selection has to be made between two standard robot architectures, namely, RTX, Figure 5(a), and Stanford, (Figure 5(b)) arms, whose DH parameters along with the mass and inertia parameters are shown in Tables IV(a) and (b) respectively.

The following joint motions are considered for the inverse dynamics calculations:

$$\begin{aligned} \theta_i &= \pi t/T - 0.5 \sin(2\pi t/T) && \text{for revolute joints, and} \\ b_i &= \pi t/T - 0.5 \sin(2\pi t/T) && \text{for prismatic joints} \end{aligned} \quad (23)$$

Table IV. D-H, mass and Inertia parameters of spatial robot arms.

Link	$a_i$ (m)	$b_i$ (m)	$\alpha_i$ (rad)	$\theta_i$ (rad)	$m_i$ (kg)	$r_x$	$r_y$ (m)	$r_z$	$I_{xx}$	$I_{yy}$ (kg-m <sup>2</sup> )	$I_{zz}$
(a) D-H Parameters for RTX robot arm											
1	0	$b_1$	0	0	9	0	0	0.054	0.1	0.02	0.1
2	0.432	0.01	0	$\theta_2$	6	0.292	0	0	0.5	0.6	0.01
3	0.2	0.01	0	$\theta_3$	4	-0.02	0	0	0.4	0.4	0.1
(b) D-H Parameters for Stanford robot arm											
1	0	0.1	$-\pi/2$	$\theta_1$	9	-0.1	0	0.1	0.1	0.02	0.1
2	0	0.1	$-\pi/2$	$\theta_2$	6	0	0	0.5	0.5	0.6	0.01
3	0	$b_3$	0	0	4	0	0	0.4	0.4	0.4	0.1

Table V. Comparison of 3-link spatial structures.

Type		RTX Robot	Stanford Robot
CC	GIM	<u>142M; 138A</u>	170M; 154A
	MCI	<u>389M; 317A</u>	522M; 352A
	CT* (sec)	<u>9.443</u>	11.166
Workspace		$4\pi b_1 a_2 a_3$	$4\pi a_3^3/3$
Volume 'V'	$a_i = b_i = a$	<u><math>4\pi a^3</math></u>	$4\pi a^3/3$
Manipulability		$a_2 a_3  \sin \theta_3 $	$b_3^2  \cos \theta_2 $
'w'	$a_i = b_i = a$	<u><math>a^2</math></u>	$a^2$

\* All computations are done on a Pentium-III PC with processor speed of 1GHz with memory size 256Mb.

M: Multiplication/division; A: Addition/subtraction. CC: Computational complexity; CT: Computation time; \_\_\_\_: Better value of the two.

where  $\theta_i(0) = 0$  and  $b_i(0) = 0$ , while  $T = 10$  sec with step size,  $t = 0.01$  sec. Note here that unlike the planar robot arm motion in horizontal plane, Figures 3(a)–(b), gravity plays an important role, which are taken into account.

For the above spatial arm configurations, proposed 'dynamics model simplicity' based criterion has been evaluated, along with the kinematics criteria. The results are tabulated in Table V. Based on the results, RTX robot is found suitable and, hence, should be selected. The better values are highlighted with underlines in the table.

## 6. Conclusions

A robot arm selection methodology based on the simplicity of dynamics model of a robot arm is proposed. The concept is illustrated with the help of 2-link planar robots. Then a selection is made from two 3-link spatial robot arms. The proposed criterion can be used when kinematic criteria are not sufficient or contradict, as explained



with respect to the 2-link example, after Equation (22). Also, in Table V, both RTX and Stanford arms have same maximum manipulability, which is not a clear indication for the selection if the manipulability is the main criterion. The present criterion is then also provides a way. The proposed criterion also allows to choose the best physical parameters for good dynamic performances. For example, one can make the generalized inertia matrix (GIM), Equations (9) and (10), well-conditioned, as reported in Bhangale et al. [23], by suitably choosing the link masses and lengths.

The contributions of this paper are highlighted as

1. Introduction of a new selection criterion, namely, the simplicity of the dynamics model of a robot arm, which are evaluated in terms of computation complexity of the associated matrices, i.e., GIM and the matrix of convective inertia (MCI) terms of Equation (10), and the CPU time, of an algorithm, say, the inverse dynamics algorithm that is essential for control purposes.
2. Computation complexity analysis of GIM and MCI for the all revolute  $n$ -degree of freedom robot. For the GIM, the proposed algorithm is efficient, as indicated after Equation (14), whereas MCI calculation is never reported earlier.
3. Modular inverse dynamics algorithms development, i.e., GIM and MCI, are computed separately and the joint torques and forces are computed using Equation (9). This way, any robot architecture can be tested by just replacing the blocks of GIM and MCI calculations.
4. Evaluation of total four criteria of two planar and two spatial robot arms for comprehensive understanding and selection of robot arm architecture.

### Appendix A: Denavit–Hartenberg (D-H) Parameters

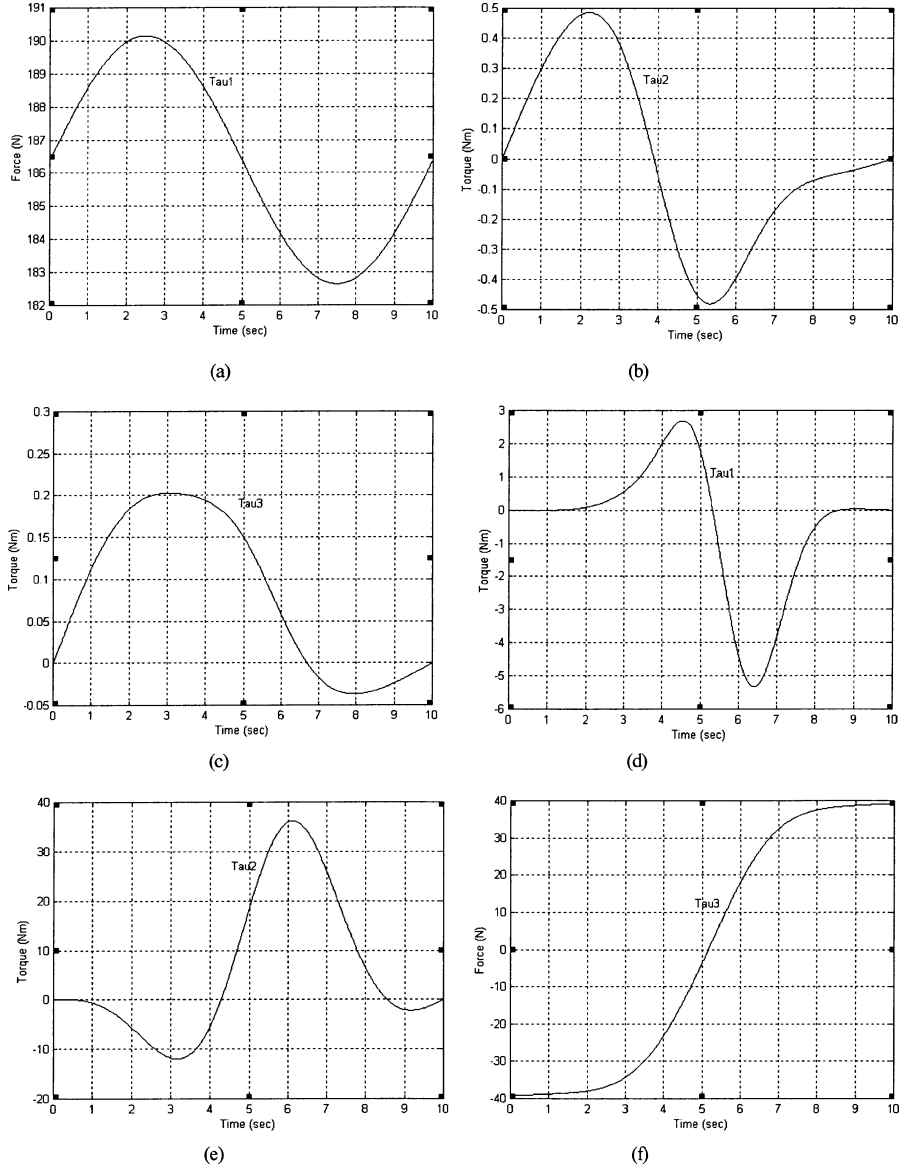
D-H-parameter is a systematic, general method to define the relative position and orientation of the consecutive link. This definition helps in computing the coordinate transformation between them. D-H-parameters can be assigned differently to the same system, the Figure 7 shows one of the popular method to define the parameters, where  $Z_i$  denotes the axis of joints and  $O_i$  and  $X_i$  forms the coordinate system.

### Appendix B: Computational Complexities

#### B.1 TO FIND $\mathbf{Q}_i[\mathbf{r}_i]_{i+1}$

The mass centre location vector  $\mathbf{r}_i$  is known in  $(i + 1)$ st frame and to convert it into  $i$ th frame, we need to pre-multiply it with the rotation matrix,  $\mathbf{Q}_i$ , i.e.,

$$[r_i]_i = \mathbf{Q}_i[\mathbf{r}_i]_{i+1}$$



(a)-(c) Joint torques for RTX and (d)-(f) for Stanford;

Figure 6. Required torques and force at joints of RTX and Stanford arms.

where  $\mathbf{Q}_i$  is given as

$$\mathbf{Q}_i \equiv \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{bmatrix} \text{ and } [\mathbf{r}_i]_{i+1} \equiv \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \quad (\text{A.1})$$

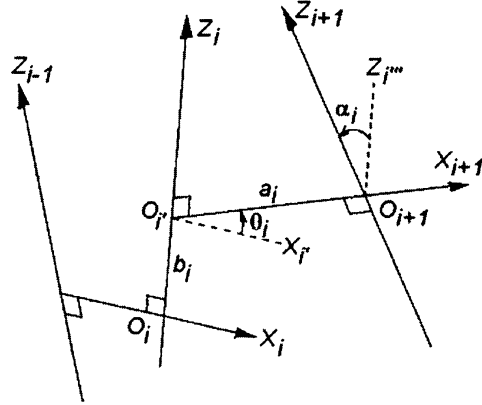


Figure 7. Transformation from frame  $i$  to  $ix + 1$ .

Hence

$$[\mathbf{r}_i]_i \equiv \begin{bmatrix} r_x \cos \theta_i - r_c \sin \theta_i \\ r_x \sin \theta_i + r_c \cos \theta_i \\ r_y \sin \alpha_i - r_z \cos \alpha_i \end{bmatrix} : 6M3A$$

where  $r_c$  is given as

$$r_c \equiv r_y \cos \alpha_i - r_z \sin \alpha_i : 2M1A$$

Thus, the total computations required to find the multiplication of a  $3 \times 3$  rotation matrix with a 3-dimensional vector are  $8M4A$  (i.e. 8 multiplications and 4 additions/subtractions).

## B.2 TO FIND $[\mathbf{M}_i]_i$

The mass matrix  $\mathbf{M}_i$  is required to be found in the  $i$ th frame, i.e.,  $[\mathbf{M}_i]_i$  from the known elements of  $[\mathbf{I}_i^c]_{i+1}$ ,  $\mathbf{Q}_i$ ,  $[\mathbf{d}_i]_i$ ,  $[m_i \mathbf{d}_i]_i$ , whose computational steps are explained below:

$$[\mathbf{M}_i]_i \equiv \begin{bmatrix} \mathbf{I}_i^c - \mathbf{d}_i \times (m_i \mathbf{d}_i \times 1) & m_i \mathbf{d}_i \times 1 \\ -m_i \mathbf{d}_i \times 1 & m_i 1 \end{bmatrix}$$

Where  $[\mathbf{I}_i^c]_i \equiv \mathbf{Q}_i [\mathbf{I}_i^c]_{i+1} \mathbf{Q}_i^T$ , and  $\mathbf{Q}_i$  is defined as Equation (A.1). Now,  $\mathbf{Q}_i [\mathbf{I}_i^c]_{i+1} \mathbf{Q}_i^T$  can be computed as

$$[\mathbf{I}_i^c]_i \equiv \begin{bmatrix} i_{11} - \beta_5 & \dots & \text{Sym} \\ \beta_2 \sin \alpha_i - \beta_3 \cos \alpha_i & i_{22} - \beta_5 - \beta_8 & \vdots \\ \beta_2 \cos \alpha_i + \beta_3 \sin \alpha_i & \beta_4 - \beta_7 \cos \alpha_i & \beta_8 + i_{33} \end{bmatrix} : 5M7A$$

where  $\mathbf{i}_{ij}$  are the elements of  $[\mathbf{I}_i^c]_{i+1}$  and

$$\begin{aligned}
 \beta_0 &= i_{12} \cos \theta_i && : 1M \\
 \beta_1 &= (i_{11} - i_{12}) \sin \theta_i - (\beta_0 + \beta_0) && : 1M3A \\
 \beta_2 &= i_{13} \cos \theta_i + i_{23} \sin \theta_i && : 2M1A \\
 \beta_3 &= \beta_1 \cos \theta_i + i_{12} && : 1M1A \\
 \beta_4 &= i_{13} \sin \theta_i - i_{23} \cos \theta_i && : 2M1A \\
 \beta_5 &= \beta_1 \sin \theta_i && : 1M \\
 \beta_6 &= \beta_4 \cos \alpha_i && : 1M \\
 \beta_7 &= [(i_{22} - i_{33}) + \beta_5] \sin \alpha_i + (\beta_6 + \beta_6) && : 1M \ 4A \\
 \beta_8 &= \beta_7 \sin \alpha_i && : 1M
 \end{aligned}$$

So the total computations required are  $16M \ 17A$ . Next, note that  $[\mathbf{m}_i \mathbf{d}_i \times \mathbf{1}]_i$  does not require any computation as  $\mathbf{m}_i \mathbf{d}_i$  is already obtained in item 4 of Section 3.2, i.e.,

$$[\mathbf{m}_i \mathbf{d}_i \times \mathbf{1}]_i : \text{nil}$$

Now to compute  $[\mathbf{d}_i \times (\mathbf{m}_i \mathbf{d}_i \times \mathbf{1})]_i$ , following steps are followed: Let us define

$$[\mathbf{d}_i]_i \equiv \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \quad \text{and} \quad [\mathbf{m}_i \mathbf{d}_i]_i \equiv \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \end{bmatrix},$$

where  $[\mathbf{d}_i]_i$  and  $[\mathbf{m}_i \mathbf{d}_i]_i$  are computed in items 3 and 4 of Section 3.2. Then

$$[\mathbf{d}_i \times (\mathbf{m}_i \mathbf{d}_i \times \mathbf{1})]_i = \begin{bmatrix} -\delta_z d_z - \delta_y d_y & & \text{Sym} \\ \delta_x d_y & -\delta_z d_z - \delta_x d_x & \\ \delta_x d_z & \delta_y d_z & \delta_y d_y - \delta_x d_x \end{bmatrix} : 6M \ 3A$$

which will require only  $6M \ 3A$ . Finally the  $3 \times 3$  upper left block matrix of  $[\mathbf{M}_i]_i$ , i.e.,  $[\mathbf{I}_i^c - \mathbf{d}_i \times (\mathbf{m}_i \mathbf{d}_i \times \mathbf{1})]_i$ , can be obtained with additional  $6A$  computations. Thus total computations required to compute  $[\mathbf{M}_i]_i$  is now summarized as:

$$\begin{aligned}
 [\mathbf{I}_i^c]_i &= \mathbf{Q}_i [\mathbf{I}_i^c]_{i+1} \mathbf{Q}_i^T && : 16M \ 17A \\
 [\mathbf{m}_i \mathbf{d}_i \times \mathbf{1}]_i &&& : \text{nil} \\
 [\mathbf{d}_i \times (\mathbf{m}_i \mathbf{d}_i \times \mathbf{1})]_i &&& : 6M \ 3A \\
 [\mathbf{I}_i^c - \mathbf{d}_i \times (\mathbf{m}_i \mathbf{d}_i \times \mathbf{1})]_i &&& : 6A \\
 \text{Total} &&& : 22M \ 26A
 \end{aligned}$$

Thus, computation complexity for  $[\mathbf{M}_i]_i$  is  $22M \ 26A$ , which is used in item 5(a) in Section 3.2.

## References

1. Rivin, E.I., *Mechanical Design of Robots*, McGraw Hill Book Company, New York, 1988.
2. Dorf, R.C. and Nof, S.Y., (ed.), *International Encyclopedia of Robotics, Applications and Automation*, John Wiley & Sons, Inc., New York, 1988.
3. Ceccarelli, M. and Vinciguerra, A., 'On the workspace of general 4R manipulators,' *International Journal of Robotics Research* **14**, 1995, 152–160.
4. Yoshikawa, T., *Foundations of Robotics Analysis and Control*, Prentice-Hall of India Pvt. Ltd., New Delhi, 1998.
5. Gravagne, I.A. and Walker, I.D., 'Manipulability, force and compliance analysis for planar continuum manipulators,' *IEEE Transactions of Robotics and Automation* **18**, 2002, 263–273.
6. Wen, J.T-Y. and Wilfinger, L.S., 'Kinematic manipulability of general constrained rigid multi-body systems,' *IEEE Transactions on Robotics and Automation* **15**, 1999, 558–567.
7. Angeles, J. and Lee, S.K., 'The formulation of dynamical equations of holonomic mechanical systems using a natural orthogonal complement,' *ASME Journal of Applied Mechanics* **55**, 1988, 243–244.
8. Armstrong, W.W., 'Recursive solution to the equations of motion of an  $n$ -link manipulator,' *Proceeding of 5th World Congress on Theory of Machines and Mechanisms (ASME)*, Vol. 9, Montreal, Canada, 1979, 1343–1346.
9. Asada, H. and Youcef-Toumi, K., *Direct Drive Robots—Theory and Practice*, MIT Press, Cambridge, USA, 1987.
10. Schiehlen, W.O., 'Multibody System Dynamics: Roots and Perspectives,' *Multibody System Dynamics* **1**, 1997, 149–188.
11. Fu, K.S. Gonzales, R.C. and Lee, C.S.G., *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill, Inc., New York, 1987.
12. Featherstone, R., *Robot Dynamics Algorithms*, Kluwer Academic Publishers, Massachusetts, USA, 1987.
13. Angeles, J. and Ma, O., 'Dynamic simulation of  $n$ -axis serial robotic manipulators using a natural orthogonal complement,' *The International Journal of Robotics Research* **7**, 1988, 32–47.
14. Walker, M.W. and Orin, D.E., 'Efficient dynamic computer simulation of robotic mechanisms,' *ASME Journal of Dynamical System, Measurement and Control*, **104**, 1982, 205–211.
15. Wang, L.T. and Ravani, B., 'Recursive computations of kinematic and dynamic equations for mechanical manipulators,' *IEEE Journal of Robotics and Automation* **RA-1**, 1985, 124–131.
16. Stokes, A. and Brockett, R., 'Dynamics of kinematic chains,' *International Journal of Robotics Research* **15**, 1996, 393–405.
17. Saha, S.K., 'Simulation of Industrial manipulators based on  $UDU^T$  decomposition of inertia matrix,' *Multibody System Dynamics* **9**, 2003, 63–851.
18. Saha, S.K., 'Dynamics of serial multibody systems using the decoupled natural orthogonal complement matrices,' *ASME Journal of Applied Mechanics* **66**, 1999, 986–996.
19. Saha, S.K. and Schiehlen, W.O., 'Recursive kinematics and dynamics for closed loop multibody systems,' *International Journal of Mechanics of Structures and Machines* **29**, 2001, 143–175.
20. Denavit, J. and Hartenberg, R.S., 'A kinematic notation for lower-pair mechanisms based on matrices,' *ASME Journal of Applied Mechanics* **77**, 1955, 215–221.
21. Angeles, J., *Fundamentals of Robotic Mechanical Systems*, Springer-Verlag, Berlin, 1997.
22. Saha S.K. and Angeles, J., 'The formulation of kinematic constraints in design-oriented machine dynamics,' *Advances in Design Automation ASME*, Vol. 2, 1991, 115–122.
23. Bhangale, P.P., Saha, S.K. and Agrawal, V.P., 'Concept of decoupled natural orthogonal complement (DeNOC) matrices for robot architecture selection,' *In Proceedings of National Conference on Machines and Mechanisms (NACOMM)*, IIT Kharagpur, India, December 21–22, 2001, 177–184.