MULTIBODY DYNAMICS 2003
Jorge A.C. Ambrósio(Ed.)
IDMEC/IST, Lisbon, Portugal, July 1-4 2003

DYNAMIC MODEL BASED SELECTION CRITERION FOR ROBOT MANIPULATORS

Prasadkumar P. Bhangale, S. K. Saha, V. P. Agrawal

Dept. of Mechanical Engineering
Indian Institute of Technology Delhi,
Hauz Khas, New Delhi – 110016, India.
prasad_bhangale@rediffmail.com, saha@mech.iitd.ernet.in, vagrawal@mech.iitd.ernet.in

Keywords: Manipulator, Architecture selection, DeNOC, GIM, Computational Complexity, Computation time.

Abstract. *The design of robot manipulators, 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, namely, the inverse dynamics algorithm, are used here to evaluate manipulator architecture. The proposed criterion is illustrated with two-link manipulators 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.*

1 INTRODUCTION

A robot is characterized by its degree of freedom, number of joints, type of joints, joint placement, link lengths and shapes, and their orientation which influence its performances, namely, the workspace, manipulability, ease and 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 different computational complexities. Hence, in order to

select a suitable robot, both aspects of kinematics and 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), 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, i.e., 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] (1999) 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 (Saha, 1999; 2003)[17,18], and parallel (Saha and Schiehlen, 2001)[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 they will be used, e.g., in the inverse dynamics algorithm.

This paper is organized as follows: Section 2 provides an outline of the dynamic modeling using the Decoupled Natural Orthogonal Compliment (DeNOC), Section 3 introduces the proposed criteria for robot selection. Section 4 illustrates the selection criteria using 2-link robots. Section 5 considers an example of an arm selection from RTX and Stanford architectures. Finally, conclusions are given in Section 6.

2 DYNAMIC MODELING USING THE DENOC

For an n-degree of freedom open-loop serial-robot, as shown in Fig.1, if m_i is the mass of the i^{th} link and $\mathbf{I_i}$ denotes the 3×3 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

$$\mathbf{M_i}\,\dot{\mathbf{t}}_\mathbf{i} + \mathbf{W_i}\mathbf{M_i}\mathbf{E_i}\mathbf{t_i} = \mathbf{w_i} \tag{1}$$

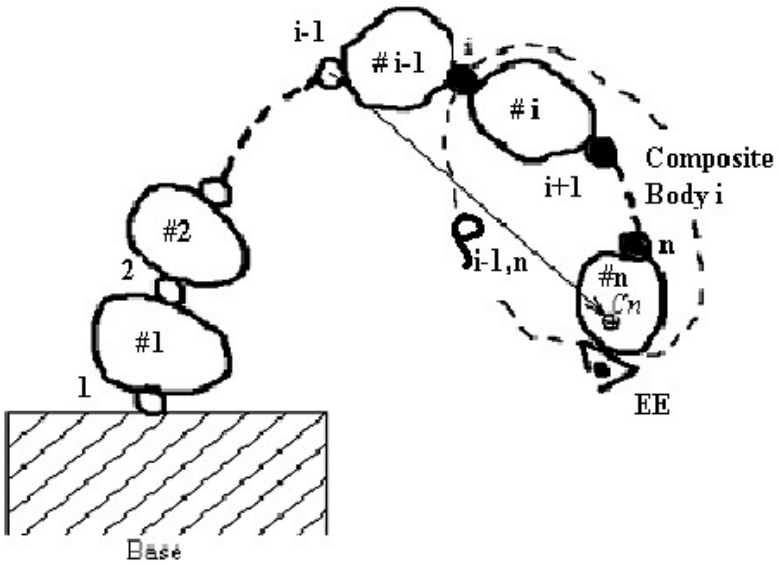
where the 6×6 mass matrix, $\mathbf{M_i}$, the 6×6 angular velocity matrix, $\mathbf{W_i}$, and the 6×6 coupling matrix, $\mathbf{E_i}$, for the i^{th} link are given as (Saha and Schiehlen, 2001)[19]

$$\mathbf{M_i} \equiv \begin{bmatrix} \mathbf{I_i} & \mathbf{m_i d_i} \times \mathbf{1} \\ -\mathbf{m_i d_i} \times \mathbf{1} & \mathbf{m_i 1} \end{bmatrix};\ \mathbf{W_i} \equiv \begin{bmatrix} \boldsymbol{\omega_i} \times \mathbf{1} & \mathbf{O} \\ \mathbf{O} & \boldsymbol{\omega_i} \times \mathbf{1} \end{bmatrix};\ \mathbf{E_i} = \begin{bmatrix} \mathbf{1} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \tag{2}$$

in which $\mathbf{I_i} \equiv \mathbf{I_i^c} - \mathbf{d_i}\times(m_i\mathbf{d_i}\times\mathbf{1}) -- \mathbf{I_i^c}$ being the 3×3 inertia tensor about the mass centre of the i^{th} body, C_i, $\mathbf{d_i}\times\mathbf{1}$ and $\boldsymbol{\omega_i}\times\mathbf{1}$ are the 3×3 cross-product tensors associated with the vector $\mathbf{d_i}$ shown in Fig. 2, and the angular velocity vector, $\boldsymbol{\omega_i}$, respectively. Note that **1** and **O** in eq.(2) are the 3×3 identity and zero matrices, respectively. The twist and wrench vectors, $\mathbf{t_i}$ and $\mathbf{w_i}$, are then defined as

$$\mathbf{t_i} = \begin{bmatrix} \boldsymbol{\omega}_i \\ \mathbf{v}_i \end{bmatrix} \quad \text{and} \quad \mathbf{w_i} = \begin{bmatrix} \mathbf{n}_i \\ \mathbf{f}_i \end{bmatrix} \tag{3}$$

where $\boldsymbol{\omega_i}$ and $\mathbf{v_i}$ are the 3-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



1, 2, …,n : Joints; #1, #2, …#n : Links

Fig. 1: An n-link manipulator

$(i-1)^{st}$ body, respectively. Moreover, $\mathbf{n_i}$ and $\mathbf{f_i}$ are the 3-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 eq.(1) is obviously the time rate of the twist vector or twist-rate vector. Eq.(1) when written for all the n links, i.e., i = 1,…, n, it can be expressed in a compact form as

$$\mathbf{M}\,\dot{\mathbf{t}} + \mathbf{WMEt} = \mathbf{w} \tag{4}$$

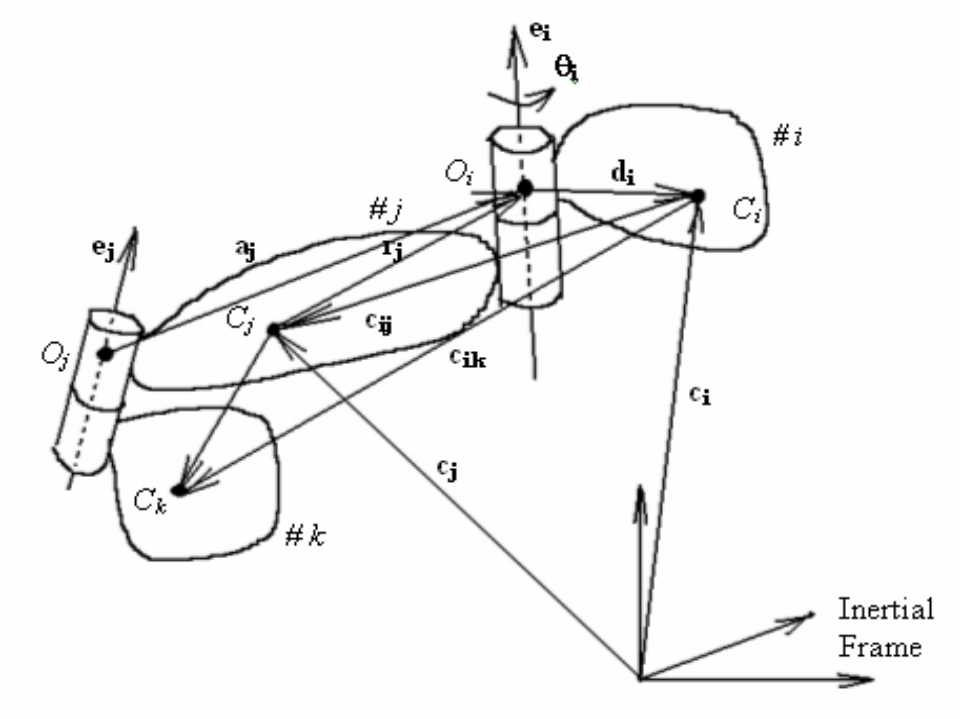


Fig. 2: A coupled system of three bodies

where **M** and **W** are the 6n×6n generalized mass and the generalized matrix of the angular velocities, respectively, whereas **E** is the 6n×6n generalized coupling matrix. They are defined as follows:

$$\mathbf{M} \equiv \begin{bmatrix} \mathbf{M_1} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \ddots & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{M_n} \end{bmatrix}; \mathbf{W} \equiv \begin{bmatrix} \mathbf{W_1} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \ddots & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{W_n} \end{bmatrix}; \mathbf{E} \equiv \begin{bmatrix} \mathbf{E_1} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \ddots & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{E_n} \end{bmatrix} \tag{5}$$

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

$$\mathbf{t} \equiv [\mathbf{t_1}^T, \mathbf{t_2}^T, \ldots, \mathbf{t_n}^T]^T; \mathbf{w} \equiv [\mathbf{w_1}^T, \mathbf{w_2}^T, \ldots, \mathbf{w_n}^T]^T \tag{6}$$

Note now that the generalized twist, **t**, can be expressed as a linear transformation of the n independent joint rates, $\dot{\boldsymbol{\theta}} \equiv [\dot{\theta}_1, \ldots, \dot{\theta}_n]^T$ -- $\boldsymbol{\theta_i}$ being the i^{th} joint displacement shown in Fig. 2 --, i.e.,

$$\mathbf{t} = \mathbf{T}\,\dot{\boldsymbol{\theta}}, \quad \text{where } \mathbf{T} \equiv \mathbf{T_I}\,\mathbf{T_d} \tag{7}$$

T being the 6n×n Natural Orthogonal Complement (NOC) matrix (Angeles and Lee, 1988)[7], whereas the 6n×6n and 6n×n matrices, $\mathbf{T_I}$ and $\mathbf{T_d}$, respectively, are the Decoupled NOC (DeNOC) matrices (Saha, 1999)[18].

Premultiplying eq.(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,

$$\mathbf{T}^T (\mathbf{M}\,\dot{\mathbf{t}} + \mathbf{WMEt}) = \mathbf{T}^T (\mathbf{w}^E + \mathbf{w}^C) \tag{8}$$

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 eq.(8) vanishes, as the constraint wrench produces no work. Substitution of the expression of $\mathbf{T} \equiv \mathbf{T_l T_d}$ from eq.(7) and its time derivative, $\dot{\mathbf{T}} = \mathbf{T_l}\dot{\mathbf{T}}_\mathbf{d} + \dot{\mathbf{T}}_\mathbf{l}\mathbf{T_d}$, into eq.(8), results in the following form of the dynamic equations of motion:

$$\mathbf{I}\,\ddot{\theta} + \mathbf{C}\,\dot{\theta} = \tau \tag{9}$$

which is nothing but the Euler-Lagrange equations of motion (Angeles and Lee, 1988)[7]. In eq.(9),

$\mathbf{I} \equiv \mathbf{T}^T\mathbf{MT} \equiv \mathbf{T}_\mathbf{d}^T\widetilde{\mathbf{M}}\mathbf{T_d}$: the n×n generalized inertia matrix (GIM);

$\mathbf{C} \equiv \mathbf{T}^T(\mathbf{M}\dot{\mathbf{T}} + \mathbf{WMET}) \equiv \mathbf{T}_\mathbf{d}^T(\mathbf{T}_\mathbf{l}^T\mathbf{M}\dot{\mathbf{T}}_\mathbf{l} + \widetilde{\mathbf{M}}\mathbf{W} + \dot{\widetilde{\mathbf{M}}})\mathbf{T_d}$: the n×n Matrix of Convective Inertia (MCI) terms;

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

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

$$\widetilde{\mathbf{M}} \equiv \mathbf{T}_\mathbf{l}^T\mathbf{MT_l};\ \dot{\widetilde{\mathbf{M}}} \equiv \mathbf{T}_\mathbf{l}^T\mathbf{WMET_l};\ \text{and}\ \ \widetilde{\mathbf{w}}^E \equiv \mathbf{T}_\mathbf{l}^T\mathbf{w}^E$$

3 DYNAMICS SIMPLICITY BASED SELECTION CRITERIA

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

3.1 Explicit Expressions for the GIM and MCI Elements

Using the expressions for the n×n GIM, **I** and, MCI, **C**, as appearing after eq.(9), each element of **I**, i_{ij}, and **C**, c_{ij}, are written as

$$i_{ij} \equiv \mathbf{p}_\mathbf{i}^T\widetilde{\mathbf{M}}_\mathbf{i}\mathbf{A_{ij}}\mathbf{p_j} \tag{10}$$

$$c_{ij} \equiv \mathbf{p}_\mathbf{i}^T(\mathbf{A}_\mathbf{ji}^T\widetilde{\mathbf{M}}_\mathbf{j}\mathbf{W_j} + \mathbf{A}_\mathbf{j+1,i}^T\widetilde{\mathbf{H}}_\mathbf{j+1,j} + \mathbf{A}_\mathbf{ji}^T\dot{\widetilde{\mathbf{M}}}_\mathbf{j})\mathbf{p_j} \qquad \text{if} \quad i \le j$$

$$c_{ij} \equiv \mathbf{p}_\mathbf{i}^T(\widetilde{\mathbf{M}}_\mathbf{i}\mathbf{A_{ij}}\mathbf{W_j} + \widetilde{\mathbf{H}}_\mathbf{ij} + \dot{\widetilde{\mathbf{M}}}_\mathbf{i}\mathbf{A_{ij}})\mathbf{p_j} \qquad \text{otherwise}$$

where, the 6×6 matrix $\mathbf{A_{ij}}$ and 6-dimensional vector $\mathbf{p_i}$ are the block elements of the DeNOC matrices, $\mathbf{T_l}$ and $\mathbf{T_d}$, respectively, i.e.,

$$\mathbf{T_l} \equiv \begin{bmatrix} \mathbf{1} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{A_{21}} & \mathbf{1} & \cdots & \mathbf{O} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A_{n,1}} & \mathbf{A_{n,2}} & \cdots & \mathbf{1} \end{bmatrix} \quad \text{and} \quad \mathbf{T_d} \equiv \begin{bmatrix} \mathbf{p_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{p_2} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{p_n} \end{bmatrix} \tag{11}$$

In eq. (11), $\mathbf{O}$ and $\mathbf{0}$ are the 6×6 zero matrix and 6-dimensional zero vector, whereas the matrix, $\widetilde{\mathbf{H}}_{\mathbf{ij}}$, in the expression of c_{ij}, is given as

$$\widetilde{\mathbf{H}}_{\mathbf{ij}} \equiv \widetilde{\mathbf{M}}_{\mathbf{i}} \dot{\mathbf{A}}_{\mathbf{ij}} + \mathbf{A}^{\mathbf{T}}_{\mathbf{i+1,i}} \widetilde{\mathbf{H}}_{\mathbf{i+1,i}} \mathbf{A}_{\mathbf{ij}} \tag{12}$$

in which $\dot{\mathbf{A}}_{\mathbf{ij}}$ is the time derivative of the 6×6 matrix, $\mathbf{A_{ij}}$. Note that the 6×6 matrix, $\widetilde{\mathbf{M}}_{\mathbf{i}}$, has the following features:

1) Matrix $\widetilde{\mathbf{M}}_{\mathbf{i}}$ can be recursively computed as

$$\widetilde{\mathbf{M}}_{\mathbf{i}} \equiv \mathbf{M}_{\mathbf{i}} + \mathbf{A}^{\mathbf{T}}_{\mathbf{i+1,i}} \widetilde{\mathbf{M}}_{\mathbf{i+1}} \mathbf{A}_{\mathbf{i+1,i}} \qquad \text{for i = n, n-1, …, 1} \tag{13}$$

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

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

2) Matrix $\widetilde{\mathbf{M}}_{\mathbf{i}}$ has a physical interpretation. It represents the mass matrix of the "composite body", i, formed by rigidly joining the bodies, i,…,n, as indicated in Fig. 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 inertia element, $i_{i+1,i} \equiv \mathbf{p_{i+1}}^{\mathbf{T}} \widetilde{\mathbf{M}}_{\mathbf{i+1}} \mathbf{A_{i+1,i}} \mathbf{p_i}$ 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,…, n,

1. *Constant* Denavit-Hartenberg (DH) parameters[20] of the system under study, i.e., α_i, a_i, and b_i (for revolute joints) or θ_i (for prismatic joints). They are also defined in Appendix A.
2. Time history of *variable* DH parameters, i.e., θ_i, for a revolute pair, and b_i, for a prismatic joint, and their first and second time derivatives.

3. Mass of each body, m_i.
4. Vector denoting the distance of the $(i + 1)^{st}$ joint from the i^{th} mass center, C_i, in the $(i + 1)^{st}$ frame, i.e., $[r_i]_{i+1}$, as shown in Fig. 2.
5. Inertia tensor of the i^{th} link about its mass center, C_i, in the $(i + 1)^{st}$ frame

Note here that $[-]_{i+1}$ will represent the argument "-" in the $(i + 1)^{st}$ 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 the first line of each step and sub-step based on the assumption that all joints of the robot are revolute with arbitrary orientation.

1. Calculate $[\mathbf{a}_i]_i$ (Angeles, 1997)[21]: {2*M* (n)}
   For $i = 1, \ldots, n$,
   $[\mathbf{a}_i]_i = [\ a_i\text{Cos}\theta_i \quad a_i\text{Sin}\theta_i \quad b_i]^T$: 2*M*
2. Calculate $[\mathbf{r}_i]_i$: {8*M* 4*A* (*n*)}
   For $i = 1, \ldots, n$,
   $[\mathbf{r}_i]_i = \mathbf{Q}_i\,[\mathbf{r}_i]_{i+1}$: 8*M* 4*A*
   where $\mathbf{Q_i}$ is given as

$$\mathbf{Q_i} = \begin{bmatrix} \text{Cos}\theta_i & -\text{Sin}\theta_i\text{Cos}\alpha_i & \text{Sin}\theta_i\text{Sin}\alpha_i \\ \text{Sin}\theta_i & \text{Cos}\theta_i\text{Cos}\alpha_i & -\text{Cos}\theta_i\text{Sin}\alpha_i \\ 0 & \text{Sin}\alpha_i & \text{Cos}\alpha_i \end{bmatrix}$$

Note that the multiplication of the 3×3 matrix, $\mathbf{Q}_i$ or $\mathbf{Q}_i^T$ with a 3-dimensional vector is obtained efficiently with only 8*M* 4*A*, instead of 9*M* 6*A* required to multiply a 3×3 matrix with a 3-dimensional vector.

3. Calculate $[\mathbf{d}_i]_i$: {3*A* (*n*)}
   For $i = 1, \ldots, n$,
   $[\mathbf{d}_i]_i = [\mathbf{a}_i]_i - [\mathbf{r}_i]_i$: 3*A*
4. Calculate $[m_i\mathbf{d}_i]_i$: {3*M* (*n*)}
   For $i = 1, \ldots, n$,
   $[m_i\mathbf{d}_i]_i = m_i\,[\mathbf{d}_i]_i$: 3*M*
5. a) Calculate $[M_i]_i$, for i = n : {22*M* 26*A*(*n*)}

$$[\mathbf{M_i}]_\mathbf{i} = \begin{bmatrix} \mathbf{I_i^c} - \mathbf{d_i} \times (\mathbf{m_i d_i} \times \mathbf{1}) & \mathbf{m_i d_i} \times \mathbf{1} \\ -\mathbf{m_i d_i} \times \mathbf{1} & \mathbf{m_i 1} \end{bmatrix}$$

$[\mathbf{I}_i^c]_i = \mathbf{Q}_i[\mathbf{I}_i^c]_{i+1}\,\mathbf{Q}_i^T$: 16*M* 17*A*
$[\mathbf{d}_i \times (m_i\mathbf{d}_i \times \mathbf{1})]_i$: 6*M* 3*A*
$[m_i\mathbf{d}_i \times \mathbf{1}]_i$: *nil*

$[\mathbf{I}_i - \mathbf{d}_i \times (m_i \mathbf{d}_i \times \mathbf{1})]_i$: $6A$

b) $[\widetilde{\mathbf{M}}_i]_i$ for i = n-1,…, 1 {$18M\ 18A(n-1)$}

$\widetilde{\mathbf{M}}_i \equiv \mathbf{M}_i + \mathbf{A}^T_{i+1,i} \widetilde{\mathbf{M}}_{i+1} \mathbf{A}_{i+1,i}$: $18M\ 18A$

6. Calculate i_{ii} :

For i = 1, …, n, {*nil*}

$i_{ii} \equiv \mathbf{p}_i^T \widetilde{\mathbf{M}}_i \mathbf{p}_i$: *nil*

For a revolute joint, $i_{ii} = \begin{bmatrix} \mathbf{e}_i^T & \mathbf{0}^T \end{bmatrix}_i [\widetilde{\mathbf{M}}_i]_i \begin{bmatrix} \mathbf{e}_i \\ \mathbf{0} \end{bmatrix}_i$. Since $[\mathbf{e}_i]_i = [0\ 0\ 1]^T$, and the calculation of i_{ii} is simply the 3,3 element of the (1,1) block matrix of $[\widetilde{\mathbf{M}}_i]_i$.

7. Calculate i_{ij} : {$22M\ 14A\ (n^2-n)/2$}

For i = n,…1; j = i-1,…,1

$i_{ij} = \mathbf{p}_i^T \widetilde{\mathbf{M}}_i \mathbf{A}_{ij} \mathbf{p}_j$: $22M\ 14A$

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

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

For six revolute jointed robot (n = 6), the complexity for the GIM is $630M\ 498A$. Note that the complexity in eq.(14) is less than that reported earlier in Walker and Orin[14] (1982), which is

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

For n = 6, eq. (15) gives $741M\ 601A$. thus an efficient algorithm for the GIM calculations is proposed in this paper.

Similarly, computational complexity for the Matrix of the Convective Inertia (MCI) terms is also carried out, which provided the complexity as

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

Again, for n = 6 with all revolute joints, the MCI complexity is $640M\ 753.5A$ which could not be compared with other due to non-availability of such results. For the planar case, the above complexity can be computed from the simplified expressions of twist and mass matrix that are 3-dimensional and 3×3 matrix, respectively, instead of 6-dimensional and 6×6 matrix for spatial motion.

The computational complexities are:

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

and

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

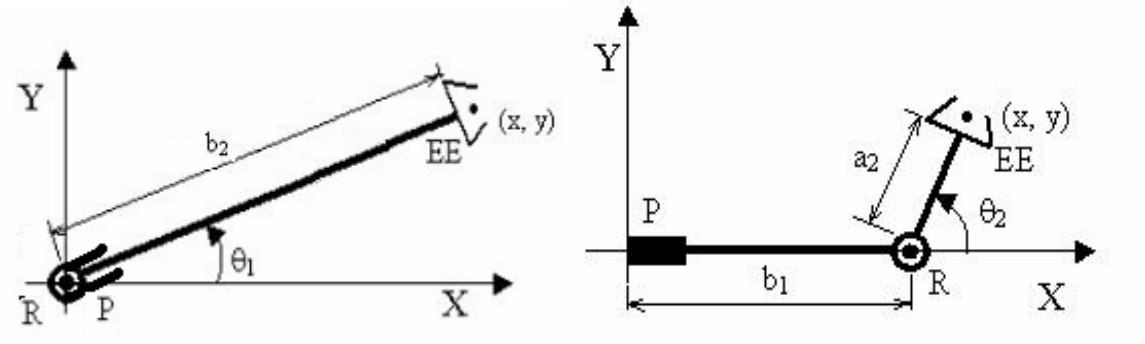
For n = 3 with all revolute joints, $59M\ 38A$ computations are required for the GIM, and $106M\ 77A$ are required for the MCI.

3.3 Computational Time for Inverse Dynamics

If the computational complexities of the GIM and MCI are less, 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. The CPU time is basically to compute the controlling joint torques/forces. Less computation time implies that the maximum speed of the manipulator can be enhanced. Thus the manipulator with least computation speed is preferable over the others.

4 ILLUSTRATION WITH PLANAR 2-LINK ROBOTS

A planar two-degree of freedom (DOF) robot suitable for planar positioning purposes only has two links and two joints. There are three options, 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 2×2 diagonal matrix with constant elements, whereas its MCI elements are zeros. Even though this architecture is simplest from 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., 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 can choose the first joint as revolute and second joint as prismatic (RP), Fig. 3(a), or vice-versa, i.e., PR type, Fig. 3(b).



(a) RP type (b) PR type

Fig. 3: Two link planar manipulators

4.1 GIM and MCI Complexity

Based on the planar computational complexities, eqs. (17) and (18), a planar robot with both revolute joints would require 88*M,* 63*A* (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 1.

4.2 Computational Time for Inverse Dynamics

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

The joint trajectories and other parameters for inverse dynamics are taken as

For RP type: 1st link length a_1= 0; 1st link mass, m_1= 12.25kg; 2nd link mass, m_2= 2.45kg;

1st joint trajectory $\theta_1 = (\pi t/T) - 0.5\text{Sin}(2\pi t/T)$ rad;

2nd joint trajectory $b_2 = (\pi t/T) - 0.5\text{sin}(2\pi t/T)$ m (19)

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

For RP type: 1st link mass, m_1= 12.25kg; 2nd link length, a_2=0.8584 m; 2nd link mass, m_2= 2.45kg;

1st joint trajectory $b_1 = (\pi t/T) - 0.5\text{Sin}(2\pi t/T)$ m;

2nd joint trajectory $\theta_2 = (\pi t/T) - 0.5\text{sin}(2\pi t/T)$ rad (20)

Computation times to obtain the necessary joint torques and forces using eq.(9) are reported in Table 1, where no gravity is considered as for the motion on a horizontal plane gravity does not have any effect.

4.4 Workspace and Manipulability

The set of points representing the maximum and minimum extent or reach of the manipulator in all directions is known as workspace or work envelope. Workspace can be calculated by visualizing the possible motion of the robot in all direction and finding the area, in the case of planar, and volume, in the case of spatial[3,11]. The workspace area for the PR can be obtained from the motion range of the robot, as shown in Fig. 4. if no limit on the joints are assumed the workspace area of PR robot for Fig. 4 can be calculated as $A = \pi a_2^2 + 2a_2b_1$. Similarly, for RP type the workspace area is $A = \pi (2a_1b_2 + b_2^2)$. The values are tabulated in Table.1 and compared while $a_i = b_i = a$ 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 (1998)[4] is called manipulability measure, w, and defined as

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

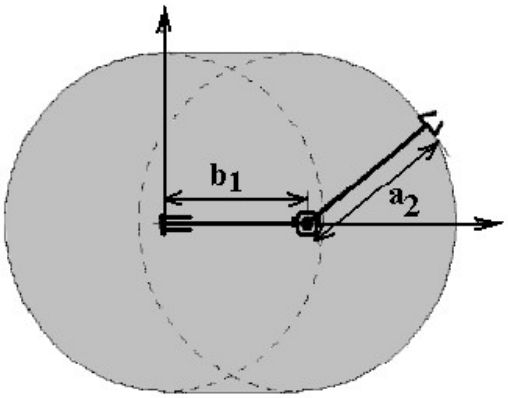


Fig. 4: The workspace of 2-link PR-manipulators

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

Type		**RP Robot** (Fig. 3(b))	**PR Robot** (Fig. 3(b))
CC	**GIM**	$\underline{12M\ 6A}$	$21M\ 16A$
	MCI	$\underline{26M\ 14A}$	$28M\ 18A$
CT* (sec)		<u>0.651</u>	0.761
Workspace area 'A'		$\pi\ (2a_1b_2 + b_2^2)$	$2\ b_1a_2 + \pi\ a_2^2$
	$\mathbf{a_i = b_i = a}$	$\underline{4\pi a^2}$	$(\pi + 2)a^2$
Manipulability '*w*'		$a_1 + b_2$	$a_2 \mid \mathrm{Cos}\theta_2 \mid$
	Max.	<u>Max. 2a</u>	Max. a

***** All computations are done on a 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 parameter

Table 1: Comparison of 2-link structures

Manipulability measure '*w*' for the RP and PR calculated, which are tabulated in Table.1. As an illustration, '*w*' for the PR robot is computed as $w = a_2|\mathrm{Cos}\theta_2|$ where the $\mathbf{J}(\theta)$ of PR robot is derived as

$$\mathbf{J}(\theta) = \begin{bmatrix} 1 & -a_2\mathrm{Sin}\theta_2 \\ 0 & a_2\mathrm{Cos}\theta_2 \end{bmatrix} \tag{22}$$

This shows that *w* varies with θ_2 and will change value through 0 to a_2.

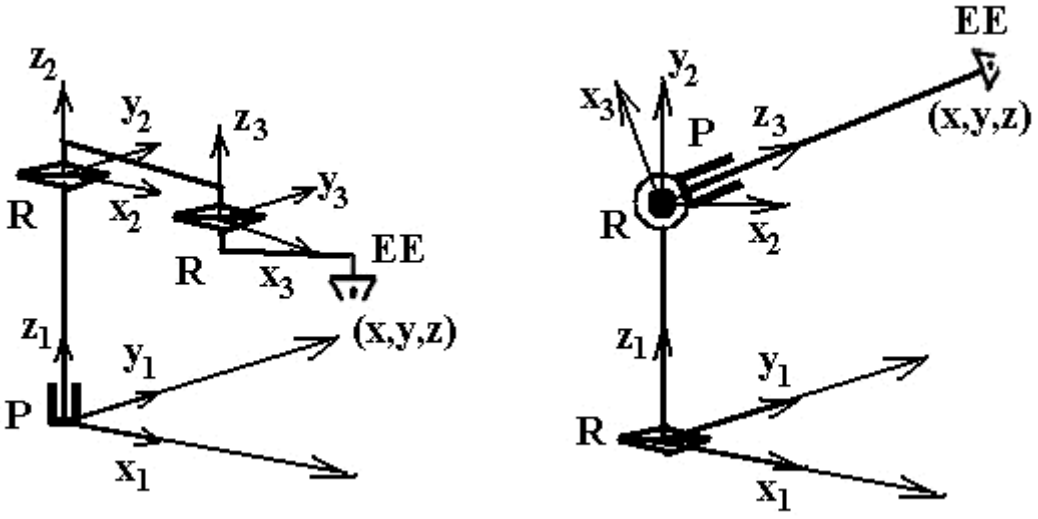
Based on the proposed criterion, i.e., Computation Complexity (CC), Computation Time (CT) and RP is found better, which is in line with kinematic based workspace and manipulability criteria also.

5 EXAMPLE: SELECTION BETWEEN RTX AND STANFORD ARM

In this section, assume a situation where selection has to be made between two standard robot architectures, namely, RTX (Fig. 5(a)) and Stanford (Fig. 5(b)) arm, whose DH parameters along with the mass and inertia parameters are shown in Tables 3(a) and (b) respectively.

The following joint motions are considered for the inverse dynamics calculations, i.e., the joint torques and forces values.

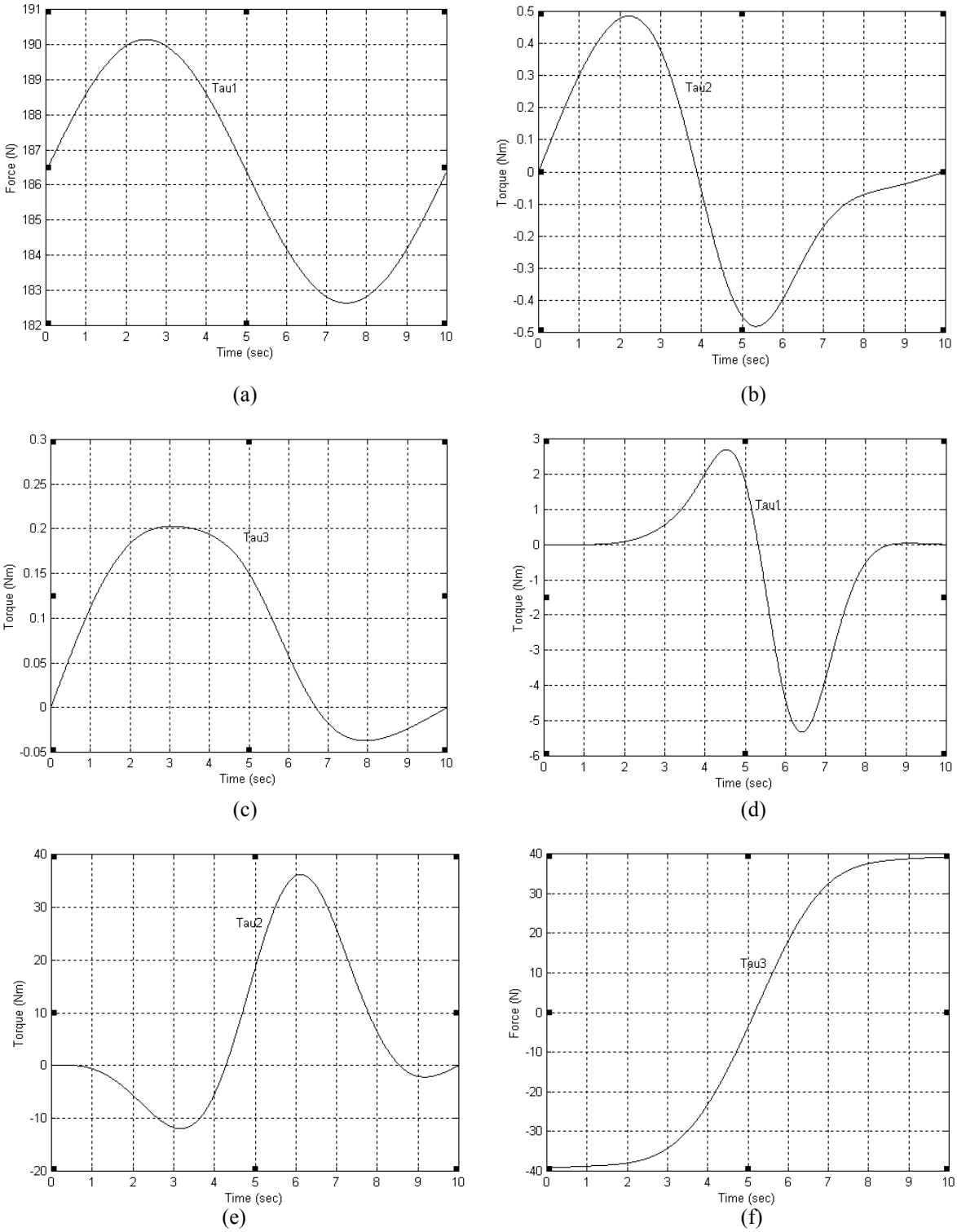
$$\theta_i = \pi t/T - 0.5\mathrm{Sin}(2\pi t/T) \quad \text{for revolute joints, and}$$
$$b_i = \pi t/T - 0.5\mathrm{Sin}(2\pi t/T) \quad \text{for prismatic joints} \tag{23}$$



a) RTX Robot b) Stanford Robot

Fig. 5: Two link planar manipulators

where $\theta_i(0) = 0$ and $b_i(0) = 0$, while T= 10 sec with step size t = 0.01. The joint torques and forces are plotted in Figs.6(a)-(f), which are verified using an in-house developed RIDIM[18] software in C++. Note here that unlike motion in horizontal plane gravity plays an important role, which are taken into account.



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

Fig. 6 Required torques at joints of RTX and Stanford Robot.

Link	a_i	b_i	α_i	θ_i	m_i	r_x	r_y	r_z	I_{xx}	I_{yy}	I_{zz}
	(m)	(m)	(rad)	(rad)	(kg)	(m)			(kg-m^2)		
1	0	b_1	0	0	9	0	0	0.054	0.1	0.02	0.1
2	0.432	0.01	0	θ_2	6	0.292	0	0	0.5	0.6	0.01
3	0.2	0.01	0	θ_3	4	-0.02	0	0	0.4	0.4	0.1

a) DH Parameters for RTX robot

Link	a_i	b_i	α_i	θ_i	m_i	r_x	r_y	r_z	I_{xx}	I_{yy}	I_{zz}
	(m)	(m)	(rad)	(rad)	(kg)	(m)			(kg-m^2)		
1	0	0.1	$-\pi/2$	θ_1	9	-0.1	0	0.1	0.1	0.02	0.1
2	0	0.1	$-\pi/2$	θ_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

b) DH Parameters for Stanford robot

Table 3: DH, mass and Inertia parameters of spatial robot arms

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

***** All computations are done on a 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 parameter

Table 4: Comparison of 3-link spatial structures

For the above spatial arm configurations proposed criterion is evaluated, along with the kinematics criteria, that are tabulated in Table 4. Based on the results, RTX robot is selected. The better values are highlighted with underlines in the table.

6 CONCLUSIONS

A robot selection methodology based on the dynamics of the manipulator, arising from the dynamic equations of motion of the manipulator at hand, is proposed. The concepts are illustrated with the help of 2-link planar robots. Then a choice is made from two 3-link spatial robot arms. The proposed criterion can be used when kinematic criteria are not sufficient or contradict. For example in Table 4 both RTX and Stanford arm have same maximum manipulability, which is not clear indication for the selection. The present methodology provides a way to choose the physical parameters for the best dynamic performances. For example, one can make the I-matrix well-conditioned as reported in Bhangale et. al.,(2001)[22], by suitably choosing the masses and the link lengths, which makes the manipulator well behaved from dynamic stability point of view.

Though the computation time difference in various architectures look miniscule here, one can imagine the difference when the inverse dynamics program will run on generally used robot controllers whose processing speed 30 to 60 times slower than the computer used here. The contributions of this paper are highlighted as

1. Introduction of a new selection criteria, namely, the computation complexity, based on the dynamic modeling of a manipulator and associated CPU time.
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 eq.(14), where MCI calculation are never reported earlier.
3. Inverse dynamics algorithms development frommodular approach, i.e., GIM and MCI, are computed separately and the joint torques and forces are computed using eq.(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 architecture.

REFERENCES

1. E. I. Rivin, *Mechanical Design of Robots*, McGraw Hill Book Company, New York (1988).
2. R.C. Dorf and S.Y. Nof (Ed.), *International Encyclopedia of Robotics, Applications and Automation*, John Wiley & Sons, Inc., New York (1988).
3. M. Ceccarelli and A. Vinciguerra, "On the workspace of general 4R manipulators," *International J. of Robotics Research,* **14**, 152-160 (1995).
4. T. Yoshikawa, *Foundations of Robotics Analysis and Control*, Prentice-Hall of India Pvt. Ltd., New Delhi (1998).

5. I. A. Gravagne and I. D. Walker, “Manipulability, force and compliance analysis for planar continuun manipualtors,” *IEEE Transactions of Robotics and Automation,* **18**, 263-273 (2002).
6. J. T-Y. Wen and L. S. Wilfinger, "Kinematic manipulability of general constrained rigid multibody systems," *IEEE Trans. on Robotics and Automation*, **15**, 558-567 (1999).
7. J. Angeles and S. K. Lee, "The formulation of dynamical equations of holonomic mechanical systems using a natural orthogonal compliment," *ASME J. of Appl. Mech*., **55**, 243-244 (1988).
8. W. W. Armstrong, "Recursive solution to the equations of motion of an n-link manipulator," *Proc. of 5th World Cong. on Theory of Machines and Mechanisms (ASME)*, Montreal, Canada, **2**, 1343-1346 (1979).
9. H. Asada, and K. Youcef-Toumi, *Direct drive robots- Theory and practice*, MIT Press, Cambridge, USA (1987).
10. W. O. Schiehlen, “Multibody System Dynamics: Roots and Perspectives,” *Multibody System Dynamics*, **1**, 149-188 (1997).
11. K. S. Fu, R. C. Gonzales, and C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill, Inc., NewYork (1987).
12. R. Featherstone, *Robot Dynamics Algorithms*, Kluwer Academic Publishers, Massachusetts, USA (1987).
13. J. Angeles and O. Ma, "Dynamic simulation of n-axis serial robotic manipulators using a natural orthogonal complement," *The Int. J. of Robotics Research*, **7**, 32-47 (1988).
14. M. W. Walker and D. E. Orin, "Efficient dynamic computer simulation of robotic mechanisms," *ASME J. of Dyn. Sys., Measurement and Control*, **104**, 205-211 (1982).
15. L.T. Wang and B. Ravani, “Recursive computations of kinematic and dynamic equations for mechanical manipulators,” *IEEE Journal of Robotics and Automation,* **RA-1**, 124-131 (1985).
16. A. Stokes and R. Brockett, “Dynamics of kinematic chains,” *International Jornal of Robotics Research,* **15**, 393-405 (1996).
17. S. K. Saha, “Simulation of Industrial manipulators based on UDU^T decomposition of inertia matrix,” *Multibody System Dynamics*, **9**, 63-851 (2003).
18. S. K. Saha, "Dynamics of serial multibody systems using the Decoupled Natural Orthogonal Complement matrices," *ASME J of Appl. Mech.*, **66**, 986-996 (1999).
19. S. K. Saha and W. O. Schiehlen, “Recursive kinematics and dynamics for closed loop multibody systems,” *Int. J. of Mechanics of Structures and Machines*, **29**, 143-175 (2001).
20. J. Denavit and R. S. Hartenberg, “A kinematic notation for lower-pair mechanisms based on matrices,” *ASME J. of Appl. Mech.*, **77**, 215-221 (1955).
21. J. Angeles, *Fundamentals of Robotic Mechanical Systems*, Springer-Verlag, Berlin (1997).
22. P. P. Bhangale, S. K. Saha and V. P. Agrawal, “Concept of Decoupled Natural Orthogonal Complement (DeNOC) Matrices for Robot Architecture Selection,” *Proc. of National Conference on Machines and Mechanisms (NACOMM),* IIT Kharagpur, India, Dec 21-22, 177-184 (2001).

APPENDIX A

DH-parameters 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. DH-parameters can be assigned differently to the same system, the Fig. 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.

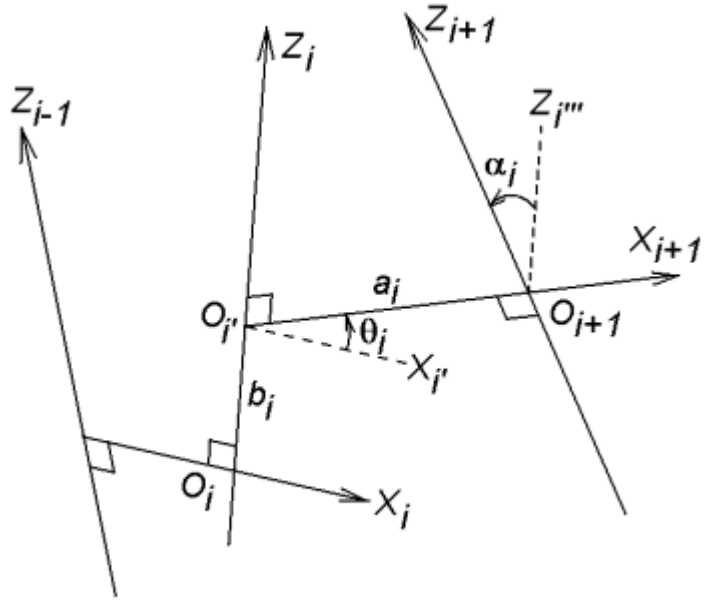


Fig. 7: Transformation from frame i to i+1.