TWO METHODS FOR COMPUTING REDUCED-FORM DYNAMICS OF CLOSED – CHAIN MECHANISM

A. Chennakesava Reddy
Associate Professor
Department of Mechanical Engineering
JNTU College of Engineering
Kukatpally
Hyderabad –500 072

P. Ram Reddy
Registrar
Jawaharlal Nehru Technological University
Kukatpally
Hyderabad – 500 072

B. Kotiveerachari
Professor
Department of Mechanical Engineering
National Institute of Technology
Warangal

Abstract: This paper covers two methods for calculating the reduced-form dynamics of closed-chain mechanism. The two methods convert an equivalent unconstrained tree-structured mechanism to a reduced, closed-chain form. They are ideal for calculating the dynamics in symbolic form with the generalized coordinates chosen to be those of the actuated (revolute or prismatic).

1. INTRODUCTION

Solving for the dynamics of closed-chain mechanisms generally involves converting a constrained system into an open tree structure by virtually cutting an unactuated joint of the closed chain. The dynamics of the open tree structure are found using well-known methods [1]. Constraints are then reapplied at the virtually cut joints allowing a representation of the closed form dynamics,

$$T_A^c = M\ddot{q} + V(\dot{q}, q) + G(q) \qquad \ldots(1)$$

Let T_A^c represents the actuated torques of the closed chain. The subscript P will be used rather than A in reference to the passive joints of the closed chain. The superscript o will be used rather than c in reference to the open tree structure.

For the open tree structure of the robot, let n denote the total number of joints and let m denote the number of passive joints. For nonredundant mechanisms, there will be n-m joints requiring actuation (giving n-m degrees of freedom). These two variables will often be included as subscripts indicating matrix dimension for clarity and quick reference.

It is convenient to represent the forces and joint variables including their respective derivatives as a group of actuated followed by a passive grouping,

$$T^0 = \begin{bmatrix} T^0_{An-mx1} \\ T^0_{Pmx1} \end{bmatrix} \quad \text{and}$$
$$q^0 = \begin{bmatrix} q^0_{An-mx1} \\ q^0_{Pmx1} \end{bmatrix} \in R^{nx1} \qquad \ldots(2)$$

The primary result of most research is the formulation of a constraint matrix, W, which converts an equivalent open tree structured mechanism back to its closed-chain form expressing the dynamics in the actuated coordinates (3). The remainder of this paper will focus on the derivation of this matrix.

$$T^{c}_{An-mx1} = W^{T} T^{0}_{nx1} \qquad \ldots(3)$$

2. CONSTRAINT MATRIX FORMULATIONS

There have been two main approaches for finding the constraint matrix, W. Both methods were originally inspired from work found in [2, 3]. The two methods are equivalent as shown in [4] and [5] but differ greatly in approach.

2.1 METHOD - I

The method first introduced by Luh and Zheng [6] in 1985 was derived from d'Alembert's principle. A set of m constraints applied at the cut joints are defined which satisfy,

$$c_{\mathrm{mx1}} = 0 \ldots(4)$$

These constraints are easily derived positional and/or orientation constraints at the cut joints. Each closed chain contributes as many constraints as there are passive joints (excluding cut joint) in the loop.

With the constraints known, the matrix has the form,

$$W = \begin{bmatrix} I_{n-m} \\ -\dfrac{\partial c^{-1}}{\partial q^{0}_{Pmxm}} \bullet \dfrac{\partial c}{\partial q^{0}_{Amxn-m}} \end{bmatrix} \in R^{nxn-m} \ldots(5)$$

Note that the ordering of the elements in c does not matter.

2.2 METHOD - II

The second method of computing the constraint matrix, W, was introduced in [4]. This research uses the fact that by definition, the W matrix is a Jacobian matrix of the total joints with respect to the actuated, i.e.,

$$W = \frac{\partial q^{0}}{\partial q^{0}_{A}} \in R^{nxn-m} \qquad \ldots(6)$$

Which is easily found assuming that the function $q^{0}_{P}\left(q^{0}_{A}\right)$ is explicitly known.

The following illustrates the equivalency of the two approaches,

$$W = \frac{\partial q^{0}}{\partial q^{0}_{A}} = \begin{bmatrix} I \\ -\dfrac{\partial c^{-1}}{\partial q^{0}_{P}} \bullet \dfrac{\partial c}{\partial q^{0}_{A}} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial q^{0}_{A}}{\partial q^{0}_{A}} \\ \dfrac{\partial q^{0}_{P}}{\partial q^{0}_{A}} \end{bmatrix} \ldots(7)$$

3. REDUCED-FORM REPRESENTATIONS

From (6), the following relations hold:

$$q^{0} = q^{0}\left(q^{0}_{A}\right) = \begin{bmatrix} q^{0}_{A} \\ q^{0}_{P}\left(q^{0}_{A}\right) \end{bmatrix}$$

$$\dot{q}^{0} = W \dot{q}^{0}_{A} \qquad \ldots(8)$$

$$\ddot{q}^{0} = W \ddot{q}^{0}_{A} + \dot{W} \dot{q}^{0}_{A}$$

From (3) and (8), the elements of (1) can be computed individually from the open tree structure dynamics using the following equations.

$$M = W^{T} M^{0} W$$

$$V = W^{T} V^{0} + W^{T} M^{0} \dot{W} \dot{q}_{A} \qquad \ldots(9)$$

$$G = W^{T} G^{0}$$

It may occasionally be desirable to express the reduced system with a set of generalized coordinates other than the actuated [8]. Let q_G denote the desired generalized coordinates. A Jacobian, W_G, can be formulated similar to (6), relating the measured coordinates to those of the open tree structure.

$$W_{G} = \frac{\partial q^{0}}{\partial q_{g}} \in R^{nxn-m} \qquad \ldots(10)$$

Similar to equations (8) and (9), the dynamics of the open tree structure can then be reduced by,

$$\dot{q}^{0} = W_{G} \dot{q}_{G}$$

$$\ddot{q}^{0} = W_{G} \ddot{q}_{G} + \dot{W}_{G} \dot{q}_{G}$$

$$M = W^{T} M^{0} W_{G} \qquad \ldots(11)$$

$$V = W^{T} V^{0} + W^{T} M^{0} \dot{W}_{G} \dot{q}_{G}$$

$$G = W^{T} G^{0}$$

An alternative to the formulation found in this paper defines W as $\frac{\partial q^0}{\partial q_G}$ where q_G denotes a set of generalized coordinates. The resulting set of generalized forces (similar to (3)), T_G, are mapped into those of the actuated by solving, $\left(\frac{\partial q_A}{\partial q_G}\right)^T T_A = T_G$. Notice that the solution is straightforward if the mechanism is nonredundantly actuated, i.e., dim (q_A) = dim (q_G). This formulation also handles systems in a much more general manner and is suggested for any future study.

4. A CASE STUDY

With Matlab software package, the constraint matrix is derived using the two-presented methods for the popularized closed-chain system.

Matlab Program:

```
% Derivation of constraint matrix using Method 1
% Define system structure and variables
n=4;
m=2; % |Actuated|Passive|
syms th1 th2 th3 th4; q = [ th1 th2 th3 th4].';
% The constraints, c, can be the difference between
the x&y branched forward
% kinematics from the base to the cut joint.
syms L1 L2
c = [
L2*cos(th2)+L1*cos(th2+th3)+L2*cos(th2+th3+th
4)-L1*cos(th1) ;
L2*sin(th2)+L1*sin(th2+th3)+L2*sin(th2+th3+th4
)-L1*sin(th1) ];
% Find constraint matrix
W = [ eye(n-m) ;
-inv(jacobian(c(1:m),q(n-
m+1:end)))*jacobian(c(1:m),q(1:n-m)) ];
% Simplify and substitute the functional
relationship between q_P(q_A)
W = simple( subs(W,{th3,th4},{th1-th2,-pi-
th1+th2}) )
% Giving ...
% W = [ 1, 0 ]
% [ 0, 1 ]
% [ 1, -1 ]
% [ -1, 1 ]
% Derivation of constraint matrix using Method 2
% Redefine q with q_P(q_A) already substituted
(assumed to be known, but this is
% generally not the case).
q = [th1 th2 th1-th2 -pi-th1+th2].';
W = jacobian(q,q(1:n-m))
% Which gives the same result for W as above.
% The reduced-form dynamic equations can then
be formulated with knowledge of the tree-
% structured equations of motion which can be
found using the toolbox on this website.
% M11 = I1+m1*g1^2+I3+m3*g3^2+m4*L1^2
V1 = (m4*g4*L1-m3*g3*L2)*sin(th2-th1)*thd2^2
% M12 = (m3*g3*L2-m4*g4*L1)*cos(th2-th1) V2
= (m3*g3*L2-m4*g4*L1)*sin(th2-th1)*thd1^2
% M22 = m2*g2^2+m4*g4^2+I4+m3*L2^2+I2
G1 = (m1*g1+m3*g3+m4*L1)*cos(th1)*g
% G2 = (m2*g2+m3*L2-m4*g4)*cos(th2)*g
```

3. CONCLUSIONS

Notice that direct application of Method II requires explicit knowledge of q_P (q_A) and is therefore only useful for symbolic calculation. Recent numerical algorithms similar to Method I have been published [9] and give a computationally efficient means for handling constraints of type (4).

Redundant mechanisms were not considered in this paper because the choice of actuation forces from a set of generalized (a smaller set) involves some form of optimization [4]; although, these types of systems do not present a similar problem for the forward dynamics.

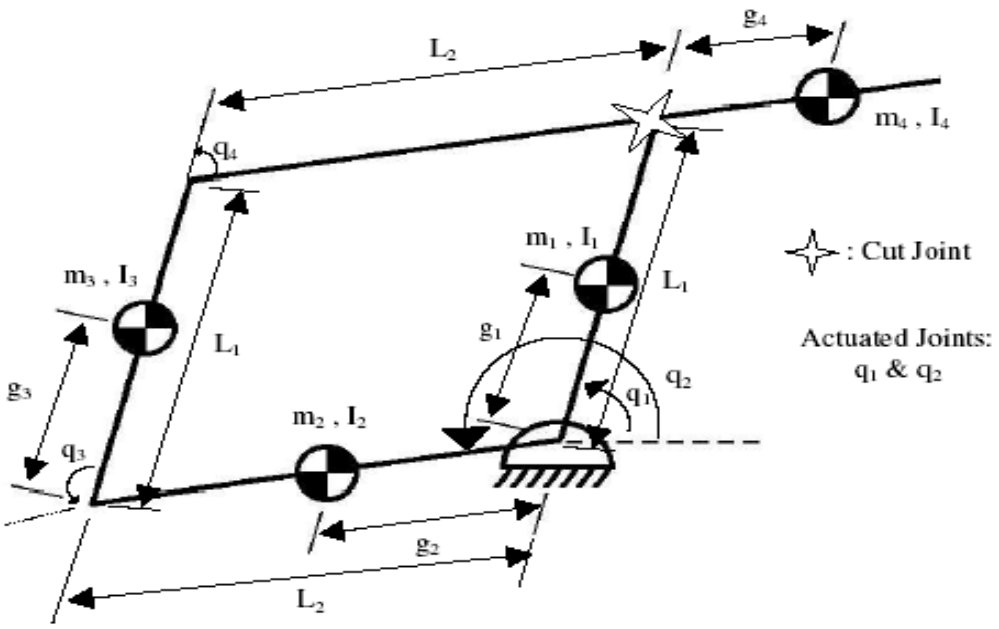

Fig-1. DD arm

REFERENCES:

[1]. R. Featherstone. "Robot Dynamics Algorithms", Kluwer Academic Publishers, Boston, Dordrecht, Lancaster, (1987).

[2]. J. Witenburg. "Dynamics of Systems of Rigid Bodies", Stuttgrt, (1977).

[3]. A. Chennakesava Reddy, B. Kotiveerachari and P. Ram Reddy, "Kinematic model of a flexible link using local curvatures as the deformation coordinates," National Conference on Advances in Manufacturing Technology, Palghat, 15-16th February, 2003.

[4] Y. Nakamura and M. Ghodoussi, "Dynamics computation of closed-link robot mechanisms with nonredundant and redundant actuators", IEEE Trans. on Robotic & Automation, (1989).

[5] Z.H. Xiong Z.X. Li G.F. Liu Y.K. Yiu, H. Cheng, "On the dynamics of parallel manipulators. International Conference on Robotics and Automation", (2001), 3766-3771.

[6] J. Y. S. Luh and Y. Zheng, "Computation of input generalized forces for robots with closed kinematic chain mechanisms", IEEE J. Robotics Automat, (1985), RA-1: 95-103.

[7] J.J. Murray and G.H. Lovell, "Dynamic modeling of closed-chain robotic manipulators and implications for trajectory control", IEEE Transactions on Robotics and Automation, Vol.5, (1989), 522-528.

[8] H. Asada and K. Y. Toumi, "Direct Drive Robots", MIT Press, Cambridge, MA, (1987).

[9] Y. Nakamura and K. Yamane, "Dynamics computation of structure-varying kinematic chains and its application to human figures", IEEE Trans. on Robotic & Automation, Vol.16, (2000), 124-134.