and the machine state number. Intuitively, our result shows that the value of optimal hedging points should increase as the system gets closer to the complete failure (zero capacity) state. This kind of structural properties of the optimal control can reduce the control space over which we want to search for the optimal control, and greatly facilitate the process of finding near-optimal controls.

Three possible avenues for extending our work are:

1) Establishing a result similar to that of Theorem 2 with respect to the age of the machine (how long the machine has been in operation). As we already mentioned in Section IV, the failure stages of a machine are merely mathematical states and are often not observable. Therefore, it would be more interesting if we could show that the hedging point increases as the age of the machine increases. So far, we have only proven this for the deterministic failure time. We believe this should, however, hold for far more general cases (see footnote 2).

2) Extending the results of this paper to more complicated systems, such as systems with multiple part-types. Conceivably, it will be much harder for the multiple part-types systems since we need to know not only relative positions among hedging points at different states but also relative positions among production switching surfaces.

3) Applying the structure properties derived here to find optimal or near-optimal controls. One obvious way is to limit ourselves to those controls satisfying these properties while searching for an optimal or near-optimal control. It is also possible to establish some equivalent relationships between the systems under controls with these special structures and some stochastic processes with which we are familiar (e.g., some classic queueing systems), e.g., see [8].

ACKNOWLEDGMENT

The authors thank Prof. Michael Caramanis for his many helpful comments and suggestions on an earlier draft of this paper. They also thank the referee for his/her comments and suggestions.

REFERENCES

[1] R. Akella and P. R. Kumar, "Optimal control of production rate in a failure prone manufacturing system," *IEEE Trans. Automat. Contr.*, vol. AC-31, no. 2, pp. 116–126, Feb. 1986.
[2] P. H. Algoet, "Flow balance for the steady-state distribution of a flexible manufacturing system," *IEEE Trans. Automat. Contr.*, vol. AC-34, no. 8, pp. 917–921, Aug. 1989.
[3] T. Bielecki and P. R, Kumar, "Optimality of zero-inventory policies for unreliable manufacturing systems," *Operations Research*, vol. 36, no. 4, pp. 532–541, Jul.–Aug. 1988.
[4] E. K. Boukas and A. Haurie, "Manufacturing flow control and preventive maintenance: A stochastic control approach," *IEEE Trans. Automat. Contr.*, vol. 35, no. 9, Sep., 1990.
[5] M. Caramanis and G. Liberopoulos, "Perturbation analysis for the design of flexible manufacturing systems flow controllers," to appear in *Operations Research*, vol. 40, no. 6, pp. 1107–1125, 1992.
[6] M. Caramanis and A. Sharifnia, "Design of near optimal flow controllers for flexible manufacturing systems," in *Proc. 3th ORSA/TIMS Conf. FMS.*
[7] S. B. Gershwin, R. Akella, and Y. F. Choong, "Short-term production scheduling of an automated manufacturing facility," *IBM J. Research and Development*, vol. 29, pp. 392–400, July 1985.
[8] J. Q. Hu and D. Xiang, "A queueing equivalent to optimal control of a manufacturing system with failures," to appear in *IEEE Trans. Automat. Contr.*, vol. 38, no. 3, pp. 499–502, 1993.
[9] ——, "Optimal Production Control for Failure Prone Manufacturing Systems with Deterministic Machine Up Times," submitted for publication, 1992.
[10] ——, "Monotonicity of optimal flow controls for failure prone production systems," submitted for publication, 1993.
[11] J. G. Kimemia and S. B. Gershwin, "An algorithm for the computer control of production in flexible manufacturing systems," *IIE Trans.*, vol. 15, pp. 353–362, Dec. 1983.
[12] G. Liberopoulos, "Design of manufacturing flow controllers: Theory and applications to the determination of near-optimal policies," Ph.D. dissertation, Boston University, 1993.
[13] G. J. Olsder and R. Suri, "The optimal control of parts routing in a manufacturing system with failure prone machines," in *Proc. 19th IEEE Conf. Decision and Contr.*, 1980.
[14] R. Rishel, "Dynamic programming and minimum principles for systems with jump markov distributions," *SIAM J. Contr.*, vol. 29, no. 2, Feb. 1975.
[15] A. Sharifnia, "Optimal production control of a manufacturing system with machine failures," *IEEE Trans. Automat. Contr.*, vol. AC-33, no. 7, pp. 620–625, Jul. 1988.
[16] S. Sethi, H. M. Soner, Q. Zhang, and J. Jiang, "Turnpike sets in stochastic production planning problems," in *Proc. Contr. Decision Conf.*, pp. 590–595, 1990.
[17] S. Sethi, H. M. Soner, Q. Zhang, and J. Jiang, "Turnpike sets and their analysis in stochastic production planning problems," to appear in *Mathematics of Operations Research.*
[18] J. N. Tsitsiklis, "Convexity and characterization of optimal policies in a dynamic routing problem," *J. Optimiz. Theory and Applications*, vol. 44, no. 1, Sep. 1984.

Computational Aspects of the Product-of-Exponentials Formula for Robot Kinematics

Frank C. Park

***Abstract*— In this article we investigate the modeling and computational aspects of the product-of-exponentials (POE) formula for robot kinematics. While its connections with Lie groups and Lie algebras give the POE equations mathematical appeal, little is known regarding its usefulness for control and other applications. We show that the POE formula admits a simple global interpretation of an open kinematic chain and possesses several useful device-independent features absent in the Denavit–Hartenberg (DH) representations. Methods for efficiently computing the forward kinematics and Jacobian using these equations are presented. In particular, the computational requirements for evaluating the Jacobian from the POE formula are compared to those of the recursive methods surveyed in Orin and Schrader [5].**

I. INTRODUCTION

Robotic control systems that depend on sensory information from the environment are often naturally implemented in terms of end-effector coordinates. In such cases the transformation between end-effector and joint coordinates is necessary, since the control inputs are in the form of torques applied to the joints. In [2] Brockett shows that the equations for an open kinematic chain containing either revolute or prismatic joints can be expressed as a product of matrix exponentials. Because of its compact representation and its connection with Lie groups and Lie algebras, the product-of-exponentials (POE) formula has proven to be a useful tool in kinematic theory [6], [8]. While this formula clearly has theoretical appeal, its effectiveness as a modeling and computational tool for

Manuscript received November 22, 1991; revised July 1, 1992 and June 11, 1993.
The author is with the Department of Mechanical and Aerospace Engineering, University of California, Irvine, CA 92717.
IEEE Log Number 9216416.

0018–9286/94$04.00 © 1994 IEEE

practical applications has, with only a few exceptions [11], received little attention.

In this article we investigate certain engineering aspects of the POE formula. We argue that the POE formula enjoys several modeling advantages over representations based on the Denavit–Hartenberg (DH) parameters and, by representing screw motions as matrix exponentials, admits an elegant global interpretation of the joint axes. Closed-form expressions for the matrix exponential and logarithm on the rigid-body motions are also derived, from which the POE formula can be evaluated directly.

The matrix exponential plays a fundamental role in control theory, and most control engineers have no difficulty in understanding or manipulating such quantities. On the other hand, while kinematic representations based on screws offer a number of advantages over those based on DH parameters—part of the appeal of the dual quaternion kinematic representation in fact stems from its transparent connections with screw theory (see McCarthy [3])—the unconventional notation and definitions used and their complex rules for manipulation make such representations difficult to grasp. The POE formula embodies all the advantages of screw-based kinematics in a form that is quite familiar to control engineers, namely as matrix exponentials. Unfortunately its general use has largely been avoided because of the belief that matrix exponentials are computationally prohibitive. Our goal is to demonstrate that the POE formula not only has intuitive simplicity and theoretical appeal, but can also be a computationally effective tool.

The article is organized as follows. In Section II we review the geometry of the rigid-body motions, hereafter denoted $SE(3)$, and derive explicit formulas for the logarithm and exponential mapping on $SE(3)$. In Sections III and IV we review the POE formula and investigate its computational aspects, comparing these results to a number of different schemes for computing the Jacobian considered by Orin and Schrader [5].

II. The Geometry of Rigid-Body Motions

For our purposes it is sufficient to think of $SE(3)$, the Euclidean group of rigid-body motions (or the Special Euclidean Group, also known in the robotics literature as the homogeneous transformations) as consisting of matrices of the form $\left[\begin{smallmatrix}\Theta & b\\ 0 & 1\end{smallmatrix}\right]$ where $\Theta \in SO(3)$ and $b \in \mathcal{R}^3$; here $SO(3)$ denotes the group of 3×3 proper rotation matrices. Both $SO(3)$ and $SE(3)$ have the structure of a differentiable manifold and an algebraic group and are examples of Lie groups.[1]

Associated with every Lie group is its Lie algebra; on $SO(3)$ its Lie algebra, denoted $so(3)$, is the set of 3×3 skew-symmetric matrices of the form

$$[\omega] \triangleq \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

where $\omega \in \mathcal{R}^3$. The Lie algebra of $SE(3)$, denoted $se(3)$, is the set of all 4×4 matrices of the form $\left[\begin{smallmatrix}[w] & v\\ 0 & 0\end{smallmatrix}\right]$ where $[\omega] \in so(3)$ and $v \in \mathcal{R}^3$. Elements of $SE(3)$ and $se(3)$ will alternatively be denoted by the pairs (Θ, b) and (ω, v), respectively.

For our purposes the primary connection between a matrix Lie group and its Lie algebra is the exponential mapping. Given some element A of a Lie algebra its exponential $\exp A$ is an element of the corresponding Lie group defined by the usual matrix exponential, i.e., $\exp A = I + A + \frac{A^2}{2!} + \cdots$. The following explicit formulas for the exponential map on $so(3)$ and $se(3)$ are derived in Park [9].

[1] See, e.g., Boothby [1] for an introduction to Lie groups and Riemannian geometry.

Lemma 1: Let $[\hat{\omega}] \in so(3)$ have Euclidean norm 1, that is, $\hat{\omega}_1^2 + \hat{\omega}_2^2 + \hat{\omega}_3^2 = 1$. Then for any $\phi \in \mathcal{R}$

$$\exp([\hat{\omega}]\phi) = I + \sin\phi[\hat{\omega}] + (1 - \cos\phi)[\hat{\omega}]^2$$

is an element of $SO(3)$.

Lemma 2: Let $[\hat{\omega}] \in so(3)$ be of Euclidean norm 1 as before, and $v \in \mathcal{R}^3$. Then for any $\phi \in \mathcal{R}$,

$$\exp\left(\begin{bmatrix} [\hat{\omega}] & v \\ 0 & 0 \end{bmatrix}\phi\right) = \begin{bmatrix} \exp([\hat{\omega}]\phi) & b \\ 0 & 1 \end{bmatrix}$$

is an element of $SE(3)$, where

$$b = (\phi I + (1 - \cos\phi)[\hat{\omega}] + (\phi - \sin\phi)[\hat{\omega}]^2)v.$$

On $SO(3)$ and $SE(3)$ the exponential map is onto but not one-to-one. The inverse of the exponential map, or logarithm, will have multiple values characterized by the following lemmas.

Lemma 3: Suppose $\Theta \in SO(3)$ such that $Tr(\Theta) \neq -1$, and let ϕ satisfy $1 + 2\cos\phi = Tr(\Theta)$. Then

$$\log\Theta = \frac{\phi}{2\sin\phi}(\Theta - \Theta^T).$$

Furthermore, $\|\log\Theta\|^2 = \phi^2$.

Remark 1: If $Tr(\Theta) = -1$, it can be shown that $\log\Theta = (2k+1)\pi[v]$, where k is any integer and v is the unit eigenvector of Θ corresponding to the eigenvalue 1.

Remark 2: The set of matrices $\Theta \in SO(3)$ such that $Tr(\Theta) \neq -1$ forms an open subset of $SO(3)$. A one-to-one logarithm mapping can be defined over this subset, by restricting ϕ to be between 0 and 2π.

Lemma 4: Suppose $\Theta \in SO(3)$ such that $Tr(\Theta) \neq -1$, and let $b \in \mathcal{R}^3$. Then

$$\log\begin{bmatrix} \Theta & b \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} [\omega] & A^{-1}b \\ 0 & 0 \end{bmatrix}$$

where $\omega = \log\Theta$ as given by Lemma 3, and

$$A^{-1} = I - \frac{1}{2}\cdot[\omega] + \frac{2\sin\|\omega\| - \|\omega\|(1 + \cos\|\omega\|)}{2\|\omega\|^2\sin\|\omega\|}\cdot[\omega]^2.$$

Lemma 5: Let $x(t)$ be a curve in $se(3)$. Then $X(t) = \exp x(t)$ is a curve in $SE(3)$, and $\dot{X}(t)X^{-1}(t)$ and $X^{-1}(t)\dot{X}(t)$ are curves in $se(3)$ given by

$$\dot{X}(t)X^{-1}(t) = \int_0^1 e^{x(t)s}\dot{x}(t)e^{-x(t)s}\,ds$$

and

$$X^{-1}(t)\dot{X}(t) = \int_0^1 e^{-x(t)s}\dot{x}(t)e^{x(t)s}\,ds.$$

III. The Product-of-Exponentials Formula

The Forward Kinematics

The standard convention for describing the kinematics of open chains is to relate a reference frame attached to each link (typically with the z axis directed along the joint axis) to the reference frame attached to the previous link. The Euclidean transformation which describes the position and orientation of the i^{th} frame in terms of the $(i-1)^{st}$ frame is $f_{i-1,i} = e^{P_i x_i}M_i$, where $M \in SE(3)$, $P_i \in se(3)$, and $x_i \in \mathcal{R}$ is the joint variable, $i = 1, 2, \cdots, n$. In terms of the Denavit–Hartenberg parameters the matrices M_i and P_i for a revolute joint are, following the notation of Paul [10],

$$M_i = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad P_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

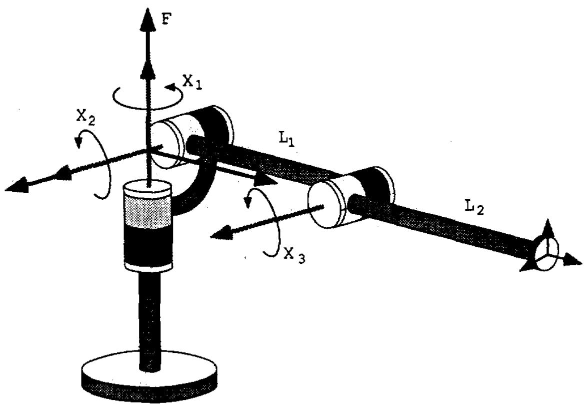

Fig. 1. A $3R$ spatial open chain.

where $c\alpha_i = \cos\alpha_i$ and $s\alpha_i = \sin\alpha_i$. If x_i is a prismatic joint then

$$M_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & 0 & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad P_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

From the matrix identity $M^{-1}e^{P}M = e^{M^{-1}PM}$, each $e^{P_i x_i}M_i$ can be written as $M_i e^{S_i x_i}$, where $S_i = M_i^{-1}P_i M_i$. The frame fixed at the tip is related to that at the base by

$$\begin{aligned} f(x_1, \cdots, x_n) &= e^{P_1 x_1} M_1 e^{P_2 x_2} M_2 \cdots e^{P_n x_n} M_n \\ &= M_1 e^{S_1 x_1} M_2 e^{S_2 x_2} M_2 \cdots M_n e^{S_n x_n}. \end{aligned}$$

By further application of the matrix identity f can be simplified to

$$f(x_1, x_2, \cdots, x_n) = e^{A_1 x_1} e^{A_2 x_2} \cdots e^{A_n x_n} M \tag{1}$$

$$= M e^{B_1 x_1} e^{B_2 x_2} \cdots e^{B_n x_n} \tag{2}$$

where $A_1 = P_1$, $A_2 = M_1 P_2 M_1^{-1}$, $A_3 = (M_1 M_2) P_3 (M_1 M_2)^{-1}$, etc., and $B_i = M^{-1} A_i M$, $i = 1, \cdots, n$.

The POE formula can alternatively be derived independently of the Denavit–Hartenberg parameters, by appealing to the interpretation of the A_i (B_i) as the screw parameters for joint i's motion (for details see [6]).[2] Specifically, denote the rotational and translational components of A_i (B_i) by ω_i and v_i, respectively. Then ω_i is a unit vector in the direction of joint axis i, expressed in inertial (tip) frame coordinates; v_i is a vector, also described in inertial (tip) frame coordinates, such that the pitch of the screw motion generated by joint i is $\omega_i^T v_i$. Note that for revolute joints the pitch is zero; for prismatic joints $\omega_i = 0$ and v_i is in the direction of movement. In both cases $\omega_i \times v_i$ is required to be a point lying on the joint axis.

Example 1: Let the forward kinematics for the $3R$ open chain of Fig. 1 be of the form $f(x_1, x_2, x_3) = e^{A_1 x_1} e^{A_2 x_2} e^{A_3 x_3} M$. Setting the joint axes to zero, the tip frame $M = (\Theta, b)$ relative to the base frame is given by $\Theta = I$, $b = (L_1 + L_2, 0, 0)$. Joint 1 has a screw axis in the direction $\omega_1 = (0, 0, 1)$; since $\omega_1 \times v_1$ is a point lying on the joint axis, with v_1 normal to ω_1, it follows that $v_1 = (0, 0, 0)$. Similarly, for joints 2 and 3 we have $\omega_2 = (0, -1, 0)$, $v_2 = (0, 0, 0)$, and $\omega_3 = (0, -1, 0)$, $v_3 = (0, 0, -L_1)$.

As the above example illustrates, the POE formula is a global description of an open kinematic chain that can be obtained independently of the DH parameters. Once inertial and tool frames have been chosen, and a zero position selected for each of the joints, there exists a unique set of constant matrices $A_1, A_2, \cdots, A_n \in se(3)$ and $M \in SE(3)$ that uniquely characterizes the forward kinematics of the mechanism. Unlike the DH representations, it is not necessary to attach reference frames to each link. Second, the POE formula treats both revolute and prismatic joints in a uniform way; recall that using the DH parameters the joint variable can be either θ_i or d_i depending on whether the joint is revolute or prismatic. Third, it is well known (see [4]) that the DH parameters are extremely sensitive to small kinematic variations when neighboring joint axes are nearly parallel. In particular, this sensitivity makes kinematic calibration routines based on DH parameters unnecessarily complicated. On the other hand the POE formula is a continuous parametric model, in the sense that the A_i's are known to vary smoothly with variations in the joint axes.

Finally, if the forward kinematic map f is transformed to $f' = PfQ$, where $P, Q \in SE(3)$ denote transformations of the base and tool frames, respectively, the new map f' becomes

$$\begin{aligned} f'(x_1, \cdots, x_n) &= e^{A_1' x_1} \cdots e^{A_n' x_n} M' \\ &= M' e^{B_1' x_1} \cdots e^{B_n' x_n} \end{aligned}$$

where $A_i' = PA_iP^{-1}$, $B_i' = Q^{-1}B_iQ$, and $M' = PMQ$.

The Jacobian

The Jacobian of a mechanism is the linear transformation relating joint rates to end-effector rates. If f is the forward kinematic map given by the POE formula, and $x(t)$ is an n-dimensional joint trajectory, then $f(x(t)) = (\Theta(t), b(t))$ is a curve in $SE(3)$ describing the motion of the tool frame relative to the inertial reference frame. It is then easily verified that both $\dot{f}f^{-1} = (\dot{\Theta}\Theta^T, \dot{b} - \dot{\Theta}\Theta^T b)$ and $f^{-1}\dot{f} = (\Theta^T\dot{\Theta}, \Theta^T\dot{b})$ are elements of $se(3)$. (Observe that both $\dot{\Theta}\Theta^T$ and $\Theta^T\dot{\Theta}$ are skew symmetric and, therefore, elements of $so(3)$.) The latter is referred to as the body-fixed velocity representation of $\dot{f}$, since $\Theta^T\dot{\Theta}$ and $\Theta^T\dot{b}$ are the angular and translational velocities of the tool frame expressed in tool frame coordinates, respectively. By a similar argument we call $\dot{f}f^{-1}$ the inertial velocity representation of $\dot{f}$. One subtle and important difference in the interpretation of the inertial velocity representation, however, is that while $\dot{\Theta}\Theta^T$ is indeed the angular velocity of the tool frame relative to the inertial frame, the translational velocity relative to the inertial frame is not $\dot{b} - \dot{\Theta}\Theta^T b$, but simply $\dot{b}$.

By a repeated application of the matrix identity $Me^{P}M^{-1} = e^{MPM^{-1}}$, it can be shown that

$$\dot{f}f^{-1} = A_1\dot{x}_1 + e^{A_1x_1}A_2e^{-A_1x_1}\dot{x}_2 + \cdots$$

and

$$f^{-1}\dot{f} = B_n\dot{x}_n + E^{-B_nx_n}B_{n-1}e^{B_nx_n}\dot{x}_{n-1} + \cdots.$$

Rearranging both $f^{-1}\dot{f}$ and $\dot{f}f^{-1}$ as six-dimensional generalized velocity vectors and expressing their corresponding right-hand sides as the multiplication of a $6 \times n$ matrix $J(x)$ with an n-dimensional joint velocity vector $(\dot{x}_1(t), \cdots, \dot{x}_n(t))$, the two equations above can be more easily identified with the conventional matrix representations for the Jacobian.

Observe that if the kinematic map f is transformed according to $f' = PfQ$, then $\dot{f}'f'^{-1} = P(\dot{f}f^{-1})P^{-1}$, i.e., it is the original Jacobian transformed by $P(\cdot)P^{-1}$. Note that Q has no effect on $\dot{f}'f'^{-1}$; the dual expression for $f'^{-1}\dot{f}'$ will similarly be unaffected by P.

[2] For a discussion of screw motions see, e.g., McCarthy [3].

IV. Computational Aspects

In this section we investigate methods of efficiently computing the Jacobian from the POE formula, our goal being to demonstrate that the POE equations can be an effective computational tool.

In [5], Orin and Schrader survey a class of recursive methods for computing the Jacobian relative to different choices of reference and tool frames. Without restating these methods, the structure and computational requirements of these recursive algorithms are seen to vary widely depending on the choice of reference and tool frame. Moreover, each of these recursive algorithms require advance knowledge of the joint type. One of the principal advantages of the POE equations, and we elaborate on this below, is the regular structure of the computation and the uniform treatment of joint types. We illustrate two computational POE-based algorithms for evaluating the Jacobian equation

$$f^{-1}\dot{f} = B_n\dot{x}_n + e^{-B_n x_n}B_{n-1}e^{B_n x_n}\dot{x}_n + \cdots$$

where $f^{-1}\dot{f}$ is the tool frame's generalized velocity in tool frame coordinates, and the $\dot{x}_i$ are the joint velocities.

Direct Method

The special structure of the matrices in the Jacobian equation considerably reduces the number of operations when compared to multiplying arbitrary matrices. First, each term is an element of $se(3)$ and, with the exception of the $\dot{x}_n$ term, consists of a multiplication of the form

$$\begin{bmatrix}\Theta & b\\ 0 & 1\end{bmatrix}\begin{bmatrix}[\omega] & v\\ 0 & 0\end{bmatrix}\begin{bmatrix}\Theta^T & -\Theta^T b\\ 0 & 1\end{bmatrix} = \begin{bmatrix}[\Theta\omega] & \Theta v - [\Theta\omega]b\\ 0 & 0\end{bmatrix} \quad (3)$$

where $\Theta \in SO(3)$ and $b,\ \omega,\ v \in \mathcal{R}^3$. Observe also that much of the computation for each term, in particular the matrix exponential products, can be reused for subsequent terms. Taking these factors into account, we see that for an n-link open chain, $n-2$ multiplications of $SE(3)$ matrices and $n-1$ manipulations of the form above are required to compute the Jacobian.

The matrix exponentials appearing in the Jacobian equation can be computed fairly easily from the formula given in Lemma 2. For each $e^{A_i x_i}$, where $A_i = (\omega_i, v_i)$, once the quantities $\sin x_i$ and $\cos x_i$ have been determined, the remaining matrix exponential involves taking simple sums and products of entries of ω_i and v_i, which are themselves fixed for any given kinematic chain. Alternatively, by generating a table of values for $e^{A_i x_i}$ at regularly-spaced intervals of the x_i, the matrix exponentials can be obtained by table lookup.

Recursive Method

If the forward kinematics are given in "local" form—that is, with reference frames attached to each link—as

$$f(x_1, \cdots, x_n) = M_1 e^{S_1 x_1} M_2 e^{S_2 x_2} \cdots M_n e^{S_n x_n}$$

then a recursive method of evaluating the Jacobian $f^{-1}\dot{f}$, similar to the methods studied in Orin and Schrader, can also be derived. Specifically, define $f_{i-1,\,i} = M_i e^{S_i x_i}$ and $f_i = f_{i-1} f_{i-1,\,i}$, with $f_0 = I$; the forward kinematic map is then $f = f_n$. Further define $V_i = f_{i-1}^{-1}\dot{f}_{i-1}$, with $V_0 = 0$. The recursive algorithm is then

$$V_i = f_{i-1,\,i}^{-1} V_{i-1} f_{i-1,\,i} + S_i \dot{x}_i, \qquad i = 1, 2, \cdots, n$$

with $\dot{f}f^{-1} = V_n$. Note that the S_i will be constant and of special form depending on whether joint i is prismatic or revolute. In particular, if the link reference frames are attached with the z—axis aligned along the joint axis, tthen $S_i = (w_i, v_i)$ will be of the form (0, 0, 1, 0, 0, 0) for revolute joints and (0, 0, 0, 0, 0, 1) for prismatic joints. An explicit formula for each $f_{i-1,\,i} = M_i e^{S_i x_i}$ can be obtained easily from our earlier lemmas; alternatively a table of values for $M_i e^{S_i x_i}$ can be generated in advance to employ a table lookup strategy. Note also that the first term in the expression for V_i is of the special form described in (3), so that it can be determined in a computationally efficient manner.

Comparison

The following table compares the number of operations required to compute the Jacobian of a general n-link open kinematic chain:

	mult.	add./sub.
POE recursive	$42n-78$	$30n-57$
POE direct	$60n-96$	$45n-72$
DH recursive	$42n$	$30n$

The DH recursive method listed in the table is the algorithm proposed by Orin and Schrader [5], which they conclude is the most efficient for computing the Jacobian $f^{-1}\dot{f}$ (${}^E J_E$ using their notation). In the POE recursive method, it is assumed that the link reference frames are attached with the z-axis aligned along the joint axis, which considerably simplifies the computation. Alternatively, if the reference link frames are arbitrarily attached, then the number of computations for the POE direct and recursive methods are identical.

While the POE recursive method is computationally the most efficient, it should be emphasized that with recent advances in computing, evaluating the kinematic equations is no longer the computational bottleneck that it once was. Rather, what seems more desirable is a kinematic representation that is elegant, easy to identify and manipulate. In the POE recursive method the forward kinematics are no longer expressed strictly as a product of exponentials, but instead have M_i matrices embedded, i.e., $f(x_1, \cdots, x_n) = M_1 e^{S_1 x_1} \cdots M_n e^{S_n x_n}$. On the other hand the original POE formula $f(x_1, \cdots, x_n) = e^{A_1 x_1} \cdots e^{A_n x_n} M$ is clearly the easiest to factor and manipulate and moreover does not require local frames attached to each link. In either case it is clear that the POE formula can be used to compute in an efficient manner the forward kinematics and Jacobian of open chains.

V. Conclusion

In this paper, we have illustrated some important advantages that the POE formula enjoys over the DH-based kinematic representations, specifically its compact form, Lie theoretic foundations, easy geometric visualization, and device-independent features. We have also demonstrated, through a number of algorithms, that the POE formula can indeed be a computationally effective tool. The formulas provided in this paper should also be of independent interest, with potential applications to trajectory planning, inverse kinematics, and kinematic calibration. We believe these results will make the POE formula more accessible and useful in a wide range of robotic applications.

References

[1] W. M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry.* New York: Academic, 1975.

[2] R. W. Brockett, "Robotic manipulators and the product of exponentials formula," in *Mathematical Theory of Networks and Systems,* P. A. Fuhrman, Ed. New York: Springer-Verlag, 1984.

[3] J. M. McCarthy, *Introduction to Theoretical Kinematics.* Cambridge: MIT Press, 1990.

[4] S. Hayati, "Robot arm geometric parameter estimation," in *Proc. 22nd IEEE Int. Conf. Decision and Control,* 1983.

[5] D. Orin and W. Schrader, "Efficient computation of the Jacobian for robot manipulators," *Int. J. Robotics Research*, vol. 3, no. 4, 1984.

[6] B. Paden and S. S. Sastry, "Optimal kinematic design of 6R manipulators," *Int. J. Robotics Research*, vol. 7, no. 2, 1988.

[7] B. Paden, "Lie groups of spatial kinematics," private communication, 1992.

[8] F. Park and D. Pack, "Motion control using the product-of-exponentials kinematic equations," in *Proc. IEEE Conf. Robotics & Automation*, Sacramento, 1991.

[9] F. Park, "The optimal kinematic design of mechanisms," Ph.D. dissertation, Harvard Univ., 1991.

[10] R. P. Paul, *Robot Manipulators: Mathematics, Programming and Control.* Cambridge: MIT Press, 1982.

[11] F. Salam and C. Yoon, "On the computational aspect of the matrix exponentials and their use in robot kinematics," *IEEE Trans. Automat. Contr.*, vol. 3, no. 4, 1986.

Force/Position Regulation of Compliant Robot Manipulators

Stefano Chiaverini, Bruno Siciliano, and Luigi Villani

***Abstract*— Stable force/position regulation of robot manipulators in contact with an elastically compliant surface is discussed in this work. The controller consists of a PD action on the position loop, a PI action on the force loop, together with gravity compensation and desired contact force feedforward. Asymptotic stability of the system in the neighborhood of the equilibrium state is proven via the classical Lyapunov method with LaSalle invariant set theorem. A modification of the Lyapunov function leads to deriving an exponential stability result. Numerical case studies are developed for an industrial manipulator.**

I. Introduction

For typical robotic tasks that require interaction with the environment, contact forces must properly be handled by the robot controller [1]. In such cases, a pure motion controller usually gives poor performance and can even cause instability.

If force sensor information is not available for control purposes, one can assign a suitable dynamic behavior between position and force variables at the contact (e.g., impedance control) [2], [3]. On the other hand, several schemes can be devised which attempt to control both end-effector position and contact force by embedding force measurements in the controller. Hybrid control is perhaps the most widely adopted strategy to force/position control of robot manipulators [4]–[7]. The basic idea is the possibility to choose whether to control position or force along each task space direction through the use of proper selection matrices. Stability of hybrid control was addressed in [8]. The problem of force/position control with force sensory feedback was also treated in [9], [10] for the general case of constrained motion tasks.

A conceptually different approach to force/position control of robot manipulators is the parallel control strategy [11]. As opposed to the hybrid control strategy, both force and position variables are used along the same task space direction without any selection mechanism. The effectiveness of the scheme is ensured by the dominance of the force control loop over the position control loop along the constrained task directions where interaction occurs. This makes the scheme suitable to manage contacts with unstructured environment and unplanned collisions, which are known to represent a drawback for hybrid controllers. Extensive description of the parallel approach and performance analysis of a control scheme with full dynamic compensation in the case of contact with an elastically compliant frictionless surface can be found in [12], [13].

In view of real-time implementation, a new parallel control scheme was recently proposed which is based on simple position PD control+ gravity compensation + desired force feedforward + force PI control [14]. A preliminary analysis, inspired by the work in [15], showed asymptotic stability of the system around an equilibrium state. In detail, for given force and position set points, the force error is driven to zero at the expense of a position error at steady state. The proof in [14], however, leads to restrictive conditions on the feedback gains.

This work presents an improved proof of local asymptotic stability based on the Lyapunov direct method with use of LaSalle invariant set theorem [16]. A different Lyapunov function is chosen which results in relaxed conditions on the feedback gains; in particular, the position proportional gain does not directly affect stability of the contact.

The Lyapunov function is further modified to prove local exponential stability of the scheme yielding a new set of conditions to be satisfied for the feedback gains.

The proposed control scheme is tested in simulation on the industrial robot COMAU SMART 6.10R; only the first three joints are considered. The numerical case study confirms the results anticipated in theory.

II. Modeling

The class of robot manipulators considered in this work is that of open kinematic chains of rigid links connected by actuated joints. If the manipulator interacts with the environment, it is convenient to describe its dynamics in an m-dimensional operational space [5] that is the space where manipulation tasks are naturally specified. The equations of motion can be written in the form

$$\boldsymbol{B}(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{C}(\boldsymbol{x}, \dot{\boldsymbol{x}})\dot{\boldsymbol{x}} + \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{u} - \boldsymbol{f} \tag{1}$$

where $\boldsymbol{x}$ is the $(m \times 1)$ vector of operational variables (usually end-effector location), $\boldsymbol{B}$ is the $(m \times m)$ symmetric and positive definite inertia matrix, $\boldsymbol{C}\dot{\boldsymbol{x}}$ is the $(m \times 1)$ vector of Coriolis and centrifugal generalized forces, $\boldsymbol{g}$ is the $(m \times 1)$ vector of gravitational generalized forces, $\boldsymbol{u}$ is the $(m \times 1)$ vector of driving generalized forces, and $\boldsymbol{f}$ is the $(m \times 1)$ vector of contact generalizes forces exerted by the manipulator on the environment; all operational space quantities are expressed in a common reference frame.

Notice that the model (1) describes an ideal robot system where the effects of joint friction, backlash and elasticity, actuator dynamics, etc. are neglected. This is a common assumption which is reasonable in a suitable operational range.

The $(n \times 1)$ vector $\boldsymbol{\tau}$ of joint actuating generalized forces is computed as

$$\boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{u}, \tag{2}$$

where $\boldsymbol{J}$ is the $(m \times n)$ manipulator Jacobian matrix.

Manuscript received March 10, 1992; revised November 27, 1992 and March 19, 1993. This work was supported by Consiglio Nazionale delle Ricerche under contract 93.00905.PF67.

The authors are with Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Napoli, Italy.

IEEE Log Number 9216418.