'89 韓國自動制御學術會議論文集 1989. 10. 27~28

TIME-OPTIMAL MULTISTAGE CONTROLLERS FROM THE THEORY OF DYNAMICAL CELL-TO-CELL MAPPINGS

Joong Sun Yoon

Robotics & Automation R/D Div., Information System Business
Samsung Electronics
Gumi, Kyung Buk, Korea

This work deals with fast-to-compute global control laws for time-optimal motion of strongly nonlinear dynamic systems like revolute robots. The theory of cell-to-cell mappings for dynamical systems offer the possibility of doing the vast majority of the control law computation offline in case of time optimization with constrained inputs. These cells result from a coarse discretization of likely swaths of state space into a set of nonuniform, contiguous volumes of relatively simple shapes. Once the cells have been designed, the bang-bang schedules for the inputs are determined for all likely starting cells and terminating cells. The resulting control law is an open-loop optimal control law with feedback monitoring and correction.

1. INTRODUCTION

The problem of robot control is very complex because the equations of motion of a typical industrial robot constitute a set of highly nonlinear and coupled, second-order ordinary differential equations. In general, inverse solutions of the complete equations of motion for use in feedback control laws consume too much computer time to be used for real-time regulation.

Several investigators have based computationally fast control laws on radically simplified equations of motion. Paul [Paul 72] neglected all Coriolis and centrifugal forces in the exact rigid-body model which greatly simplifies the equations of motion. Luh, Walker and Paul [Luh 80] point out that this simplification may be justified during low-speed approach to the desired terminal state, but the resulting control law still may be too slow for general on-line use. Moreover, this is an extreme simplification, undesirable in general and inapplicable to rapid, global motions. Unacceptable errors are introduced by ignoring Coriolis and centrifugal terms in the example of Luh [Luh 80]. Bejczy [Bejczy 74] went further and diagonalized the coefficient matrices to obtain an approximate model that allows rapid computation. This approximation is insufficiently accurate to be generally useful and velocity and position feedback are required to stabilize the system and compensate for model inaccuracies.

Luh, Walker and Paul have an algorithm for computing the generalized inputs required to produce specified motion of a typical robot that may be fast enough to be used for control if a computer comparable to their PDP 11/45 with floating-point hardware is programmed in assembly language. As their paper shows, that is a remarkable improvement but still falls short of permitting use of less powerful computers programmed in a high-level language. Even so, it only does half of the regulation task. It can be used to compute the accelerations, velocities and displacements from generalized force inputs or required force inputs from specified motions, but says nothing about what the intermediate positions should be for efficient movement between specified end points.

The theory of cell-to-cell mappings for dynamical systems [Hsu 80a, 80b] offers the possibility of doing the vast majority of the control law computation off line, leaving only a small amount of integer computation to be performed on line in real time. The resulting control law is an open-loop optimal control law with feedback monitoring and correction. It is suitable for all rapid, large-scale changes in state of nonlinear dynamic systems with constrained inputs. The table lookup burden is expected to be less than that of Raibert and Horn [Raibert 78] and only integer data need to be stored.

2. TIME-OPTIMAL CONTROL PROBLEM

For an increased level of productivity, it is important that the end-point of the robot manipulator move from an initial location to a specified final position in the minimum time subject to the available gmaximum actuator torque/force at each joint. Kahn [Kahn 69] presented a sub-optimal control, based on a linearized and uncoupled model with gravity and averaged angular velocity compensations. The problem of path tracking is studied by parametrization of the given path which provides solutions along the path [Bobrow 82].

The equation of motion for an n-degree-of-freedom manipulator is [Yoon 89]

$$D(g)\,\ddot{g} + C(g,\dot{g}) + G(g) = \tau \,. \qquad (1)$$

Each joint of the manipulator is driven by an actuator which has limited driving torque/force. The

control constraint set becomes

$$|\tau_i(t)| \leq \tau_{i_{max}} \quad \text{for all } t \text{ and } i = 1, 2, \ldots, n \quad (2)$$

Introducing 2n state variables $\underline{x} = [\ \underline{q}^T, \dot{\underline{q}}^T\ ]^T$, the time-optimal control problem can be stated as follows : Given a continuous dynamical system with specified initial state $\underline{x}_0$, specified terminal state $\underline{x}_1$ and constrained control set U; find the admissible controls, $\underline{\tau}(t)$ belonging to U, which transfer the system from $\underline{x}_0$ to $\underline{x}_1$ in minimum time

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), \underline{\tau}(t), t) \quad (3)$$

$$\underline{x}(t_0) = \underline{x}_0 \quad (4)$$

$$\underline{x}(t_1) = \underline{x}_1 \quad (5)$$

$$\underline{\tau}(t) \in U \text{ for all } t \quad (6)$$

where $\underline{x}$ is a state vector, $\underline{x} = (x_1, x_2, \cdots, x_{2n})^T$, and $\underline{\tau}$ is a control input vector, $\underline{\tau} = (\tau_1, \tau_2, \cdots, \tau_n)^T$; and the cost functional J is given as

$$J = \int_{t_0}^{t_1} f_0(\underline{x}(t), \underline{\tau}(t))\, dt \ . \quad (7)$$

The search for an optimal solution is guided by Pontryagin's Maximum Principle [Pontryagin 64]. The Hamiltonian is

$$H(\underline{x}, \underline{\psi}, \underline{\tau}, t) = \sum_{i=0}^{2n} \psi_i f_i(\underline{x}, \underline{\tau}) \quad (8)$$

where $\psi_0 = -1$, $f_0 = 1$ and the covariant and state equations form a canonical system

$$\dot{\psi}_i = -\frac{\partial H}{\partial x_i}, \qquad i = 0, 1, 2, \cdots, 2n \ , \quad (9)$$

$$\dot{x}_i = \frac{\partial H}{\partial \psi_i}, \qquad i = 1, 2, \cdots, 2n \ . \quad (10)$$

The optimal control is the control which extremizes the Hamiltonian throughout the time required to move from $\underline{x}_0$ to $\underline{x}_1$. To validate an optimal control strategy, it is necessary to solve the coupled state equations and covariant equations. This is a set of 4n nonlinear, first-order ordinary differential equations in variables $\underline{x}$ and $\underline{\psi}$, subject to boundary conditions on $\underline{x}$ at times t_0 and t_1, i.e., a two-point boundary-value problem. The computational difficulties are such that obtaining accurate solutions for two-point boundary-value problems involving more than a few equations is not a trivial exercise.

Recently optimal control problems especially with bounded inputs, based on the Maximum Principle, have been of interest in the design of manipulator controls [Kahn 69, Geering 86]. A general numerical scheme, which is based on Davidenko's method, has been developed for a fixed-end-points, free-terminal-time, optimal-control problem [Yoon 88a, 88b].

3. TIME-OPTIMAL MULTISTAGE CONTROLLERS FOR NONLINEAR CONTINUOUS PROCESSES

In a 1977 paper [Hsu 77], Hsu described a technique for approximating the state-space regions of asymptotic stability surrounding the stable equilibria of nonlinear difference equation systems. The backward evolution, point-to-point mapping of a small closed curve encircling the stable singularities was said to determine regions of asymptotic stability and these regions can be systematically enlarged so as to approach the total regions of asymptotic stability for those singularities.

In 1980, Hsu described a cell-to-cell approach [Hsu 80a, 80b], replacing the point-to-point mappings, which dramatically lowered the computational burden of determining global regions of asymptotic stability if one would accept the uncertainty resulting from cellular discretization of the state space. Since the "unravelling algorithm of global analysis" turned out to be inexpensive to compute precision could be regained by use of a great many small cells, typically 100 cell widths per prominent feature width along each state-space axis in early examples.

Some of the numerical performance improvement is no doubt lost when the procedure is applied directly to nonlinear dynamic systems governed by ordinary differential equations. In such cases cell-to-cell transitions may be determined by integrating the equations of motion,

$$\frac{dx}{dt} = f(x) \quad (11)$$

over a small time interval, Δt, from initial states, x(0), at the geometric centers of cells

$$x(\Delta t) = x(0) + \int_0^{\Delta t} f(x)\, dt. \quad (12)$$

The cells which contain $x(\Delta t)$ are defined to be the nominal images of the cells which contained x(0). Then the same unravelling algorithm allows delineation of global regions of like dynamic behavior.

In 1981 [Hsu 81], Hsu addressed the problem of representing the fine structure of dynamic system behavior without resorting to extremely large numbers of very small cells. In the Generalized Theory of Cell-to-Cell Mapping, Hsu abandoned the deterministic assumption that the mapping is unique in favor of a probabilistic process in which the complete image of any cell can span multiple cells. Then the probability that a point-to-point transition originating in a given cell will end in a specific other cell is a function of the fraction of the volume of the origination cell which maps into the specific termination cell. By this process Hsu takes full account of the inexactness of the mapping of right parallelopiped cells by arbitrary continuous dynamical systems.

3.1 DYNAMIC SYSTEM REPRESENTATION

Departing from Hsu's work, perhaps the liklihood of unique mappings can be increased in order to eliminate the need to resort to multiple image cells and probabilistic transitions by tailoring the cell sizes and shapes to specific systems [Johnson 86]. Consider a swath of state space filled with cells with longitudinal boundaries which conform to the stream hypersurfaces of a particular dynamic system phase fluid motion and transverse boundaries which are equal elapsedtime hypersurfaces. In the absence of singularities and discontinuities, exact conformity would assure deterministic cell-to-cell transitions despite the use of comparatively large volume cells. The benefit would be the ability to represent with confidence the dynamic behavior of a system as a sequence of cell residences determined solely by the initial cell and the particular inputs. This would permit computationally inexpensive representation of the dynamical behavior of strongly nonlinear systems. In particular, it could mean that optimal on-line computer control would become practical. This would be achieved by exchanging an amount of on-line computation, which is too great to be performed in present computers seeking to control fast nonlinear dynamical systems using existing methods of control law design, for the considerably larger amount of off-line computing needed to define cells and cell-to-cell transitions.

As a simple example consider the pendulum shown in Figure 1. If this pendulum is thought of as a simple, single-degree-of-freedom robot arm, it would be a robot with a joint actuator of considerable power for the swath simulated and one capable of developing full torque immediately. The resulting trajectories and contours of equal elapsed time define a set of regularly shaped cells of similar volumes which impose uniform requirements on actuator and feedback transducer accuracies.

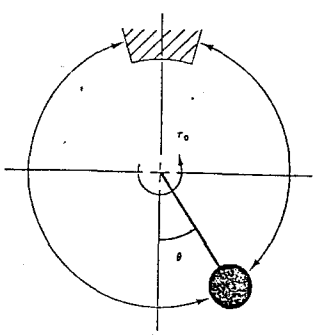

Figure 1 A Driven Pendulum. The equation of motion is $(d\theta^2/dt^2)=\tau/ml^2-(g/l)\sin\theta$.

Practical control systems require continual, or at least frequent, verification of anticipated system behavior and the possibility of correcting errant behavior. If the computational benefits of cellular discretization are to be maintained, the computational burden of converting measured state vectors into cell identities must be low. Simplest to identify from state measurements would be Hsu's uniform rectangles. A compromise strategy could be the use of system specific, nonuniform rectangles. Potentially, rectangular cells offer easy identification with system-specific tailoring to reduce the extent to which the cell mappings are nonunique.

If the swath of interest is assumed to be made up of those trajectories which represent motion from initial positions to the left with zero initial velocities to final positions to the right of the axis of symmetry with zero final velocities then the time optimal torque schedule should be full positive torque followed by full negative torque. Therefore the picture of the swath can be completed by plotting negative-time trajectories back from the positive abscissa with full negative torque. See Figure 2.

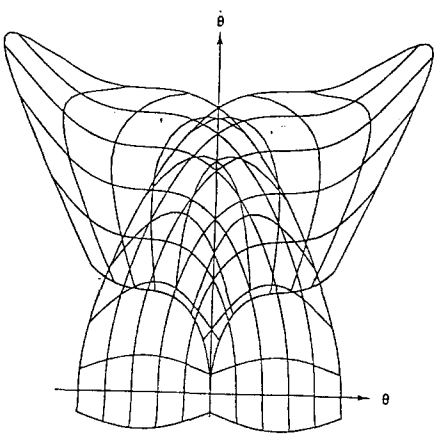

Figure 2 Intersecting Exact Cells

This considerably complicates the design of system-specific cells with acceptably low incidence of multiple imaging, since, depending on which particular initial and final states are specified, a given centrally located cell could be an approximation of the positive torque phase-fluid motion or the negative torque phase-fluid motion. In fact, all of the cells in the region of positive torque trajectories/negative torque trajectories overlap could be approximations of either. Depending upon the particular task, the switch from positive to negative torque could take place anywhere in the region of overlap. Figure 3 shows a set of cells designed to represent the complete swath. Most cells in the overlap region serve as starting points for both positive torque trajectories and negative torque trajectories. Table 1 lists the one-Δt transitions for positive and negative torques for all of the cells in Figure 3 which are required by the tasks represented by the swath in Figure 2.

Table 1 Nominal Cell-to-Cell Transitions for One Δt.

CELL	+τ	−τ	CELL	+τ	−τ
1	2(.72)	–	21	–	27(.52)
2	6(.48)	–	22	–	33(.77)
3	7(.83)	5(.34)	23	24(.47)	–
4	5(.72)	–	24	25(.33)	23(.87)
5	10(.65)	9(.50)	25	30(.58)	24(.58)
6	11(.29)	10(.70)	26	31(.62)	30(.60)
7	16(.29)	11(.39)	27	33(.44)	31(.57)
8	9(.70)	–	28	–	–
9	14(.75)	13(.47)	29	–	28(.73)
10	15(.58)	14(.91)	30	36(.40)	29(.32)
11	21(.37)	15(.37)	31	38(.98)	36(.71)
12	13(.73)	–	32	–	37(.88)
13	19(.75)	–	33	–	38(.76)
14	20(.68)	19(.89)	34	–	–
15	27(.64)	20(.48)	35	–	34(.45)
16	–	27(.78)	36	41(.50)	35(.44)
17	18(.75)	–	37	–	40(.56)
18	25(.48)	17(.42)	38	–	41(.68)
19	26(.84)	25(.51)	39	–	–
20	27(.61)	26(.84)	40	–	39(.41)
			41	–	40(.50)

3.2 OPTIMAL INPUT SCHEDULES

From Table 1, it is possible to determine the optimal input schedules for all tasks originating in cells 1, 4, 8, 12 and 17 and terminating in cells 23, 28, 34, 39 and 35, the latter cell representing the inability of the system to accomplish all intended swath tasks while switching the torque only once.

Table 2 Optimal Torque Schedules

SCHEDULE Δt	1t	2t	3t	4t	5t	6t	7t	8t	9t
1 23+−−−−−	2	6	10	14	19	25	24	23	
1 28+++−−−−	2	6	11	15	20	26	30	29	28
1 34++++−−−	2	6	11	21	27	31	36	35	34
1 39++++−−−	2	6	11	21	27	31	36	35	*
4 23+−−−−	5	10	14	19	25	24	23		
4 28+++−−−	5	10	15	20	26	30	29	28	
4 34++++−−	5	10	15	27	31	36	35	34	
4 39++++−−	5	10	15	27	31	36	35	*	
8 23+−−−	9	14	19	25	24	23			
8 28+++−−	9	14	20	26	30	29	28		
8 34++++−−	9	14	20	27	31	36	35	34	
8 39++++−−	9	14	20	27	31	36	35	*	
12 23++−−	13	19	25	24	23				
12 28+++−−	13	19	26	30	29	28			
12 34++++−−	13	19	26	31	36	35	34		
12 39+++++−−	13	19	26	31	38	41	40	39	
17 23++−	18	25	24	23					
17 28+++−	18	25	30	29	28				
17 34++++−	18	25	30	36	35	34			
17 39++++−	18	25	30	36	35	*			

For this simple, single-degree-of-freedom system, Table 2 can be constructed from Table 1 by inspection. In higher degree-of-freedom systems the development of optimal torque schedules will have to be automated.

For the twenty specific tasks considered, beginning in cells 1, 4, 8, 12 and 17 and going to cells 23, 28, 34 and 39, the schedules for bang-bang application of torque are developed from Table 1. Note that the discretization is so coarse that cell 39 is found to be unreachable from cells 1, 4, 8 and 17 (marked by asterisks), but cell 35 is adjacent and on the safe side of the travel constraint. However if multiple torque switchings are permitted, cell 39 can be reached from cells 1, 4 and 8 according to Table 1.

3.3 TIME-OPTIMAL MULTISTAGE CONTROL LAW

A discrete event controller for gross motion of a continuous plant like a robot would operate as follows:

1. From a higher level in an overall hierarchical control system, the local discrete event controller would receive a target state and convert it to a target cell identity.
2. The local gross motion controller assumes control of the plant and determines the identity of the initial cell.
3. The controller looks up the input schedule for time-optimal control with constrained inputs.
4. The controller looks up the anticipated or nominal sequence of cell residences at integer multiples of the basic time interval.
5. It applies the specified inputs for one time interval and measures the resulting system state.
6. It converts the fedback state measurements into a cell identity.
7. If the current state is within the target cell relinquish control to a local fine-motion controller.
8. If the current state is within the anticipated cell of the nominal sequence, go to step 5.
9. If the current state is within an unanticipated cell, treat the current cell as a new initial cell and return to step 3.

If the computational burden of determining the identity of the cell in which the system state resides from position and velocity measurements is low, the total computational burden of the controller will be low.

3.4 SINGLE-DEGREE-OF-FREEDOM SIMULATION

A local or robot-level discrete event, gross-motion controller was added to the numerical dynamic simulation of the single-degree-of-freedom pendulum. Figure 3 show the connected discrete states resulting from eight typical tasks within the basic swath.

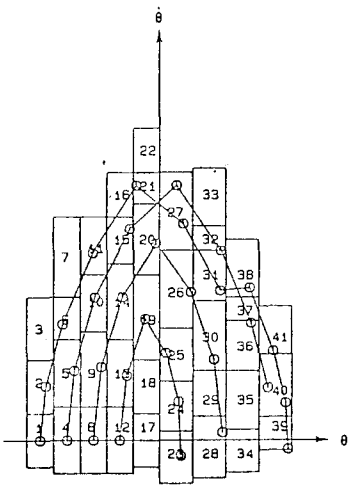

Figure 3 Single Degree of Freedom Simulation

The task beginning in cell 12 and terminating in cell 23 is carried out as Table 2 predicts. After each Δt the state is near the center of the expected cell. The next task, from cell 8 to cell 28, also matches the nominal cell sequence of Table 2 but the states at multiples of Δt are less centrally located in the cells. The task of going from cell 4 to cell 34 is not successfully completed. The optimal torque schedule from Table 2 carries the state along the optimal cell sequence until cell 36 is reached. The optimal torque schedule then drives the state into cell 40 instead of cell 35. The table of optimal torque schedules tells the controller that cell 34 cannot be reached from cell 40. The most complicated aspect of controller design is deciding what to do about cases like these. Usually reaching an adjacent cell will be acceptable and cell 39 can be reached in one time interval from cell 40. The fourth task in Figure 3 begins in cell 1 and goes to cell 39. However, the nominal cell-to-cell transitions predict that cell 39 is unreachable if the torque is to be switched only once. If three switches are permitted the trajectory reaches cell 39 after $10\Delta t$. This is certainly suboptimal. The optimal time-of-flight for a continuous problem with the same initial and final states is $9.48\Delta t$.

Note that this is a planned three-switch trajectory made necessary by the fact that the nominal transitions showed cell 39 to be unreachable from cell 1 in one switch and suboptimal performance is to be expected. The torque schedule is ++++−−+−−− and the trajectory follows the nominal cell sequence of 1-2-6-11-21-27-31-38-41-40-39.

4. EXTENSION TO MULTI-DEGREE-OF-FREEDOM SYSTEM

The central off-line computational task is the definition of state-space cell boundaries delineating 6-dimensional volumes for a three-degree-of-freedom robot which represent the approximate state transitions during a fixed increment of time, Δt [Yoon 88a]. This results in a sufficiently nonuniform collection of cells which has sufficiently low cardinality that real-time computations can be made for the motion between the cells. This is achieved by making Δt quite large compared to the time needed by a mini or microcomputer to execute a suitable control law. Ideally a simple definition scheme produces a set of cells which result in high probabilities of correct prediction of cell-to-cell transition, so that the possibility of multiple mapping images can be neglected. Such a scheme may be difficult to find, in which case the generalized theory of cell-to-cell mappings [Hsu 81] must be invoked. The major purpose of the generalized theory is to address the problem of multiple mapping images. In any case, feedback is used to compensate for imperfections in the algorithm. A numerical dynamic simulation of a three-revolute robot like a PUMA is used in the determination of these cell boundaries.

Once the cells have been defined and labelled, the simulation is used to develop all of the transition mappings from each cell under each of six conditions: maximum positive effort and maximum negative effort at each of the three actuators with the intended manipulator load. With the elimination of unnecessary cases and redundant transition matrices and by using a sufficiently large Δt, the storage requirements will be reduced as far as possible. Further dramatic reduction is possible if trajectories are assumed to be confined to a swath across the state space which accomodates all possible motions associated with a particular task or type of task. This is in contrast to filling the entire state space with cells.

The Δt for cell definition is orders of magnitude larger than the sampling interval for a conventional discrete controller or the time step for numerical integration of the equations of motion. It need only be small enough to define cell boundaries sufficiently close to the continuous optimal switching surfaces to be able to deliver the load to a neighborhood of the desired end state without an undue number of suboptimal, feedback-induced corrections. The terminal neighborhood must be small enough to permit completing the process of reaching the target state using a low-speed, terminal-guidance control law without incurring significant extension of the elapsed time.

For an n-degree-of-freedom system, the cells are to be designed from the trajectories in 2n-dimensional state space. It is not easy to visualize trajectories directly in more than a 3-dimensional space. This difficulty can be removed if

trajectories are projected and shown in a series of 2-dimensional views. For the example of the gross-motion of a robot which is usually a three-degree-of-freedom system, the cells occupy volume in 6-dimensional state space.

In an example of PUMA in Figure 4, trajectories for 27 neighboring initial conditions are computed applying positive torques for $1\Delta t$. First, hypercube of trajectories are projected in θ_1- $\dot{\theta}_1$ plane and holding $\dot{\theta}_1$ axis trajectories are projected in $\dot{\theta}_1$- θ_2 plane. These processes are repeated in θ_2- $\dot{\theta}_2$, $\dot{\theta}_2$ - θ_3 and θ_3- $\dot{\theta}_3$ plane. The cells can be designed based on these trajectories in five 2-dimensional plane cells. First horizontal boundary lines which lie approximately halfway of the swath are drawn and horizontal boundary lines are completed by drawing lines simply at the same distance from the horizontal lines to the starting/end trajectories. Vertical boundary lines are drawn in a way that cells can accomodate the trajectories.

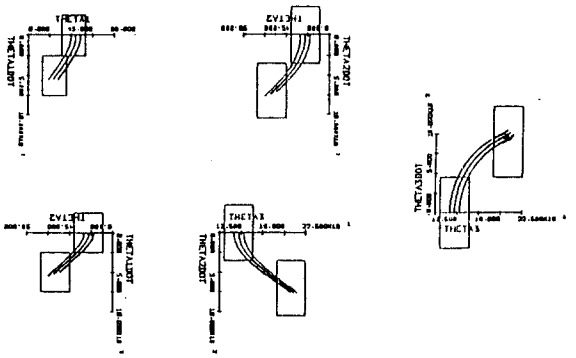

Figure 4 Three-DOF PUMA Cells

The cells in any dimensional space can be designed projecting the hypercube of trajectories forward in 2n-1 2-dimensional space and the cell-to-cell control proposed in this chapter can be applied to any number of degree-of-freedom system.

5. SUMMARY

A scheme for drastically reducing the computational burden of on-line digital imposition of time-optimal control of strongly nonlinear systems has been presented by example. It appears suitable for gross-motion control of revolute robots. The basis of the scheme is the relaxation of the controller assignment from continuous or point-to-point management to cell-to-cell transition management. The cells, in turn, are based on numerical dynamic simulations of as detailed a model of the plant as necessary and their specification represents the "design step" in the overall process of achieving improved transient performance.

The single-degree-of-freedom system described above performs well under cell-to-cell control. The controller coding is simple and encourages the expectation that an inexpensive microprocessor could provide time-optimal, gross-motion control in real time. The memory requirement is modest but would increase dramatically as the number of degrees of freedom and the dimensionality of the state space increase.

REFERENCES

[1] Bejczy, A.K., *Robot Arm Dynamics and Control*, NASA Tech. Memo. 33-669, JPL, February 1974.

[2] Bobrow, J.E., *Optimal Control of Robotic Manipulators*, PhD dissertation, UCLA, Dec. 1982.

[3] Geering, H.P., Guzzella, L., Hepner, S.A.R., and Onder, C.H., "Time Optimal Motions of Robots in Assembly Tasks", *IEEE Trans. on Automatic Control*, Vol. AC-31, No. 6, June 1986, pp.512-518.

[4] Hsu, C.S., Yee, H.C., and Cheng, W.H., "Determination of Global Regions of Asymtotic Stability for Difference Dynamical Systems", ASME *J. of Appl. Mech.*, March 1977, pp.147-153.

[5] Hsu, C.S., "A Theory of Cell-to-Cell Mapping Dynamical Systems", ASME Paper No.80-WA/APM-26.

[6] Hsu, C.S., and Guttalu, R.S., "An Unravelling Algorithm for Global Analysis of Dynamical Systems: An Application of Cell-to-Cell Mappings", ASME Paper No. 80-WA/APM-27.

[7] Hsu, C.S., "A Generalized Theory of Cell-to-Cell Mapping for Nonlinear Dynamical Systems", ASME Paper No. 81-WA/APM-25.

[8] Johnson, S.H., Harlow, D.G., and Yoon, J.S., "Time-Optimal Multistage Controllers for Nonlinear Continuous Processes",, ASME *J. of DSMC*, September 1986, pp.240-247.

[9] Kahn, M.E., *The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains*, PhD dissertation, Stanford University, 1969.

[10] Luh, J.Y.S., Walker, M.W., and Paul, R.P.C., "On-Line Computational Scheme for Mechanical Manipulators", ASME *J. DSMC*, June 1980, pp.69-76.

[11] Paul, R.P., *Modeling, Trajectory Calculation and Servoing of a Computer-controlled Arm*, Stanford A.I. Lab., A. I. Memo 177, September, 1972.

[12] Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V., and Mishcenko, E.F., *The Mathematical Theory of Optimal Processes*, Macmillan, New York, 1964.

[13] Raibert, M.H., and Horn, B.K., "Manipulator Control Using the Configuration Space Method", *The Industrial Robot*, June 1978, pp.69-73.

[14] Yoon, J.S., *Time-Optimal Control for Robots from the Theory of Dynamical Cell-to-Cell Mappings*, PhD dissertation, Lehigh University, 1988.

[15] Yoon, J.S., and Johnson, S.H., "The Numerical Solution of Multipoint Boundary Value, Time-Optimal Control Problems by Davidenko's Method", submitted to ASME *Journal of DSMC*, Sept. 1988.

[16] Yoon, J.S., "Formulation of Robot Dynamics", KSME 1989.