FUZZY CONTROLLER DESIGN FOR A THREE JOINT ROBOT LEG IN PROTRACTION PHASE*

An optimal behavior inspired fuzzy controller design

Mustafa Suphi Erden, Kemal Leblebicioğlu

Department of Electrical & Electronics Engineering, Computer Vision and Intelligent Systems Research Laboratory,
Middle East Technical University, 06531 Ankara, Turkey

Keywords: Robot leg, fuzzy controller, optimal control.

Abstract: A fuzzy controller design is performed for a three joint robot leg in protraction phase. The aim is to develop a controller to carry the tip point to any given destination. The design is based on the inspirations derived from optimal behaviors of the leg. The optimal trajectories are obtained by using optimization methods utilizing "numerical gradient" and "optimal control" successively. Separate fuzzy controllers are designed for each actuator. In writing the rules each actuator is considered to be an independent agent of the leg system. The protraction motion is divided into two epochs. For each epoch different controller systems are designed to switch from one to the other in between.

1 INTRODUCTION

The aim in this study is to design a fuzzy controller system, which will carry a three joint robot leg from any initial position to any commanded position in a protraction phase. In Fig.1 a schematic representation of a three joint robot leg is given. A three joint robot leg can be considered as a three joint robot manipulator. Eq.1 relates the tip point position to the three joint angles. In this equation $\overline{P}_e^{(b)}(\overline{\theta})$, represents the position of the tip point with respect to the body frame, when the joint angles take the values in the $\overline{\theta}$ vector. The variables in the form of a_i represent the length of the *ith* link. The variables in the form of θ_{ij} mean the sum of the *ith* and *jth* joint angles $(\theta_{ij}=\theta_i+\theta_j)$. The variables in the form of c_i are not shown in the figure but exist in the equation. $(-c_i,0,0)$ represents the position of the center of gravity of the *ith* link with respect to the *ith* coordinate frame. Namely, the term (a_i-c_i) designates the distance of the center of gravity of the *ith* link from the *ith* joint.

In order to calculate the energy dissipated in actuators during a protraction phase, one needs to calculate the joint torques throughout the movement. If the motion is slow, the inertial related forces due to acceleration, Coriolis and centrifugal effects can be neglected and the torque can be calculated as resulting only by the gravitational forces. Namely, if the motion is slow, the result of a static analysis of torques can be substituted for the dynamic analysis. This means that the torques are equal in amount and direction to compensate the gravitational force on the leg. In Eq.2, the torque vector (Q_1, Q_2, Q_3) for a position represented by the $\overline{\theta}$ vector is given. The m_1, m_2, m_3 values in this equation correspond to the link masses, and g is the gravitational acceleration.

$$\overline{P}_e^{(b)}(\overline{\theta}) = \begin{bmatrix} \cos\Psi(a_1\sin\theta_1 + a_2\sin\theta_1\cos\theta_2 + a_3\sin\theta_1\cos\theta_{23}) + \sin\Psi(a_2\sin\theta_2 + a_3\sin\theta_{23}) \\ -(a_1\cos\theta_1 + a_2\cos\theta_1\cos\theta_2 + a_3\cos\theta_1\cos\theta_{23}) \\ -\sin\Psi(a_1\sin\theta_1 + a_2\sin\theta_1\cos\theta_2 + a_3\sin\theta_1\cos\theta_{23}) + \cos\Psi(a_2\sin\theta_2 + a_3\sin\theta_{23}) \end{bmatrix}$$

$$\overline{Q}(\overline{\theta}) = \begin{bmatrix} Q_1(\overline{\theta}) \\ Q_2(\overline{\theta}) \\ Q_3(\overline{\theta}) \end{bmatrix} = \begin{bmatrix} \left\{ \begin{matrix} -m_1 g\sin\Psi(a_1-c_1)\cos\theta_1 - m_2 g\sin\Psi\left[a_1\cos\theta_1 + (a_2-c_2)\cos\theta_1\cos\theta_2\right] \\ -m_3 g\sin\Psi\left[a_1\cos\theta_1 + a_2\cos\theta_1\cos\theta_2 + (a_3-c_3)\cos\theta_1\cos\theta_{23}\right] \end{matrix} \right\} \\ \left\{ \begin{matrix} m_2 g\left[\sin\Psi(a_2-c_2)\sin\theta_1\sin\theta_2 + \cos\Psi(a_2-c_2)\cos\theta_2\right] \\ m_3 g\left[\begin{matrix} \sin\Psi a_2\sin\theta_1\sin\theta_2 + \sin\Psi(a_3-c_3)\sin\theta_1\sin\theta_{23}) \\ +\cos\Psi a_2\cos\theta_2 + \cos\Psi(a_3-c_3)\cos\theta_{23} \end{matrix} \right] \end{matrix} \right\} \\ \left\{ m_3 g\left[\sin\Psi(a_3-c_3)\sin\theta_1\sin\theta_{23} + \cos\Psi(a_3-c_3)\cos\theta_{23}\right] \right\} \end{bmatrix} \quad (2)$$

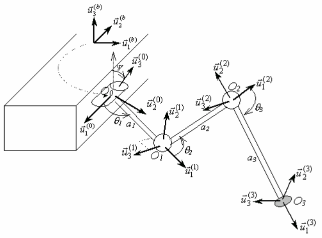

Figure 1: A three joint leg diagram with coordinate frames attached to the links.

* This research is supported by the research fund of Middle East Technical University as a scientific research project: BAP – 2002 – 03 – 01 – 06.

Suphi Erden M. and Leblebicioğlu K. (2004).
FUZZY CONTROLLER DESIGN FOR A THREE JOINT ROBOT LEG IN PROTRACTION PHASE - An optimal behavior inspired fuzzy controller design.
In *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, pages 302-306
DOI: 10.5220/0001132503020306

In robotic applications it is common to use the sum of squares of joint torques as a criteria of dissipated energy (Bobrow et al., 2001; Liu et al., 2000). Following this approach, Eq.3 will be utilized as a criterion of dissipated energy in this study. The input of the three joint leg system for a protraction movement is the trace of joint velocities throughout the protraction period. This period is taken to be *5 seconds* in this work. The input vector at a time instant can be represented as in Eq.4.

$$\int_0^{t_f} \bar{Q}(t)\bar{Q}'(t)dt \quad (3) \quad ; \quad \bar{u}(t) = \dot{\bar{\theta}}(t) = \begin{bmatrix} \dot{\theta}_1(t) \\ \dot{\theta}_2(t) \\ \dot{\theta}_3(t) \end{bmatrix} \quad (4)$$

2 TRAJECTORY OPTIMIZATION

The energy optimization problem for the protraction movement of three joint leg can be stated as to find the optimal $\bar{u}(t)$ trajectory to minimize Eq.3 from the initial to the final time with given initial and final tip point positions. This problem is a typical "optimal control" problem, which is trivial to formulate and solve (see, e.g. Kirk, 1970). However, the solution of this optimal control problem is very much dependent on the initial trajectory used at the start. If the initial trajectory is not feasible, it is very probable that the resultant trajectory will also not be feasible. Therefore, the optimal control technique needs a feasible initial trajectory. In order to generate this initial trajectory a method using "numerical gradient", in which feasibility is imposed by penalty functions, is used. The output trajectories of this method were pretty good. The optimal control method then is used to tune the trajectories slightly around the trajectories found by the initial method.

In the **method based on the numerical gradient** the conventional method of "steepest descent" is utilized after the problem is discretized. The total duration is divided into *50* equal durations, and the starts of durations are signified as the *50* time instants (denoted by *n*). The actuators are assigned a velocity at each instant and this velocity is held constant in the sub-period starting with that instant. In this way, the movement of an actuator is accomplished by *50* velocity values throughout the protraction period of *5 seconds*. Since there are three actuators, the input vector which makes the system to accomplish a whole protraction is a *50×3* long vector $[\bar{\bar{u}}]_{j,n}$). The initial and final tip point positions are constant. Therefore the required joint angles for initial and final positions are given. Initial input, namely initial trajectory of joint velocities, is taken as the average velocity that would take the tip point from the initial position to the final. The cost function is constructed by summing up the following four penalty functions. Eq.5 is the penalty related with the sum of torque squares. Eq.6 is the penalty to avoid the tip point of the leg to go under the ground (ground level is –5). Eq.7 is the penalty function related with the joint angle limits. The last penalty in Eq.8 is related with the required final condition of angles. If the final angles are not equal to the required values this term becomes positive.

$$C_\tau = \sum_{j=1}^{3} \int_0^{t_f} Q_j^2(\bar{\theta}(t))\,dt \cong \sum_{j=1}^{3} \sum_{n=1}^{50} Q_j^2(\bar{\theta}[n]) \quad (5)$$

$$C_u = \sum_{n=1}^{50} \exp(\max(5, -P_{e,z}[n])) \quad (6)$$

$$C_r = \sum_{n=1}^{50} \left\{ \begin{array}{l} [max(0, \theta_{0,1} - \theta_1[n], \theta_1[n] - \theta_{f,1}) \times 100]^2 \\ + \left[\left\{ \begin{array}{c} |\theta_2[n]|, if\ \theta_2[n] is\ out\ of\ range \\ 0, if\ \theta_2[n] is\ in\ the\ range \end{array} \right\} \times 100 \right]^2 \\ + \left[\left\{ \begin{array}{c} |\theta_3[n]|, if\ \theta_3[n] is\ out\ of\ range \\ 0, if\ \theta_3[n] is\ in\ the\ range \end{array} \right\} \times 100 \right]^2 \end{array} \right\} \quad (7)$$

$$C_f = \left|\bar{\theta}(t_f) - \bar{\theta}_f\right|^2 \quad (8)$$

The overall cost function (Eq.9) is a weighted sum of these four penalties. The values of these weights are taken as follows: *T=1; U=200; R=0.01; F=1000*. These values are arranged by trial and error in order to make the four costs comparable and to force the algorithm to generate some feasible solution. According to the steepest descent algorithm, the gradient of the cost function with respect to the input vector is found and the input vector is iterated in the opposite direction of the gradient. The value of α in Eq.10 is determined by a one-dimensional search in each iteration.

$$C = TC_\tau + UC_u + RC_r + FC_f \quad (9) \quad ; \quad u_j[n] = u_j[n] - \alpha\left[\frac{\partial C}{\partial \bar{\bar{u}}}\right]_{j,n} \quad (10)$$

In order to determine the trajectories with the **optimal control technique**, first the Hamiltonian formulation of the problem should be performed, and then the differential equations should be numerically solved. Following the notation in (Kirk, 1970), the Hamiltonian formulation, necessary conditions and boundary conditions for the three joint leg problem are given in Eq.9, 10, and 11, respectively. The first two equations of the necessary conditions make up two differential equations whose initial and final conditions are given respectively by the boundary conditions equations. Starting with an initial $\bar{u}(t)$ trajectory these equations can be solved numerically. Next the $\bar{u}(t)$ trajectory can be updated in the direction to minimize the third equation of necessary conditions. After some iteration the optimal $\bar{u}^*(t)$ trajectory, which makes the third necessary condition as close as possible to *0*, can be achieved. This technique is called "the method of steepest descent for two-point boundary-value problems" (Kirk, 1970). The initial $\bar{u}(t)$ trajectory for this technique is taken from the output of the initial method and it is further improved by the optimal control method.

Hamiltonian formulation (9)

$$\bar{u} = \dot{\bar{\theta}}$$
$$\bar{x} = \bar{\theta}$$
$$\dot{\bar{x}} = \bar{a}(\bar{x}(t), \bar{u}(t), t) = \bar{u}$$
$$J = h(\theta(t_f)) + \int_0^{t_f} g(\bar{x}(t), \bar{u}(t), t)dt$$
$$= (\theta(t_f) - \theta_f)' \mathbf{D_h} (\theta(t_f) - \theta_f) + \int_0^{t_f} \bar{Q}'(t) \mathbf{D_j} \bar{Q}(t) dt$$
$$= (\theta(t_f) - \theta_f)' \mathbf{D_h} (\theta(t_f) - \theta_f) + \int_0^{t_f} \bar{Q}'(\bar{\theta}(t)) \mathbf{D_j} \bar{Q}(\theta(t)) dt$$
$$H(\bar{x}(t), \bar{u}(t), \bar{p}(t), t) = H(\bar{\theta}(t), \bar{u}(t), \bar{p}(t), t)$$
$$= g(\bar{x}(t), \bar{u}(t), t) + \bar{p}'(t)\bar{a}(\bar{x}(t), \bar{u}(t), t)$$
$$= \bar{Q}'(t) \mathbf{D_j} \bar{Q}(t) + \bar{p}'(t)\bar{u}(t)$$

Necessaryconditions (10) *BoundaryConditions (11)*

$$\dot{\bar{x}}^*(t) = \frac{\partial H}{\partial \bar{p}}(\bar{x}^*(t), \bar{u}^*(t), \bar{p}^*(t), t) = \bar{u}^* = \dot{\bar{\theta}}^*$$
$$\dot{\bar{p}}^*(t) = -\frac{\partial H}{\partial \bar{x}}(\bar{x}^*(t), \bar{u}^*(t), \bar{p}^*(t), t)$$
$$= -\frac{\partial(\bar{Q}'(t)\mathbf{D_j}Q(t))}{\partial \theta} = -2\bar{Q}'(t)\mathbf{D_j}\frac{\partial \bar{Q}(t)}{\partial \theta}$$
$$\bar{0} = \frac{\partial H}{\partial \bar{u}} = \frac{\partial g}{\partial \bar{u}} + \bar{p}^*(t) = \bar{p}^*(t)$$

$$\bar{x}^*(t_0) = \bar{\theta}_0$$
$$\bar{p}^*(t_f) = \frac{\partial \bar{h}}{\partial \bar{x}}(\bar{x}^*(t_f)) = \frac{\partial \bar{h}}{\partial \bar{\theta}}(\bar{\theta}(t_f))$$
$$= 2\mathbf{D_h}'(\bar{\theta}(t_f) - \bar{\theta}_f)$$

Optimization Results:

The optimization is performed for nine pairs of initial and final tip point positions. In Fig.2, the first three figures respectively show the three joint angle velocities of the nine results. The fourth figure shows the tip point trajectories for two of those nine results. The fuzzy controller design will be accomplished with the intuitive feeling derived from these four figures.

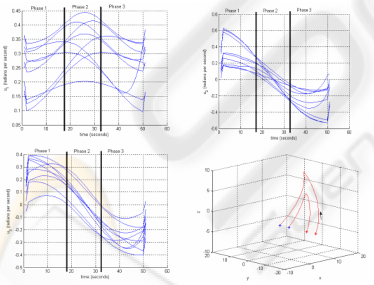

Figure 2: Optimal joint angle velocities for nine results and tip point trajectories for two results

3 FUZZY CONTROLLER

In order to get an intuitive feeling of the behavior of actuators, the optimal velocity trajectories are separated into three phases (Fig.2, first three figures), with straight tick lines. The **first phase** corresponds to the ascending of the tip point, the **second phase** corresponds to the period during which the tip point is at highest levels, and the **third phase** corresponds to the descending of the tip point towards the destination. These three phases are possible to be distinguished by the value of the first actuator angle. Roughly, it can be stated that, *the center of the first phase is the time when the first joint angle is π/4 radians, the center of the second phase is when the first joint angle is π/2 radians, and the center of the third phase is when the first joint angle is 3π/4 radians*. In the first phase the first angle, which is most effective for furthering the tip point in the *y* direction, moves with a rather slow velocity (Fig.2, first figure). It makes its peak in the second phase and stays close to that maximum value. In the third phase the velocity decreases to lower values gradually. Roughly we can state that, *the first angle moves very slow in the first and third phases and moves with a constant high velocity in the second phase*. The behaviors of the second and third angles are very similar (Fig.2, second and third figures). In the first phase they have positive big velocities, in the second phase they have small velocities around zero, and in the third phase they have negative big velocities. Roughly we can state, *the second and third angles move with constant positive and negative high velocities in the first and third phases respectively, and they stay stationary in the second phase*. These intuitive feelings will be useful in writing the rules for the controller of each angle.

The whole motion is controlled with two different controller systems each of which has different controllers for the three joint angles. The whole motion can be roughly divided into two epochs of ascending and descending. The ascending epoch corresponds to the first phase and first half of the second phase. The descending epoch corresponds to the last half of the second phase and the third phase. The boundary between these two epochs is roughly the instant at which the first angle is *π/2* radians. The first controller will make the tip point rise according to the behaviors of the angles in the corresponding phases, and the second controller will bring the tip point to its destination again according to the behaviors in the corresponding phases. Moreover, the second controller has to satisfy the stability around the destination point at the end of the second epoch. In fact this is the crucial point which explains the need of different controller systems for ascending and descending..

In (Erden et al., 2004) we had developed a multi agent perspective based fuzzy controller design paradigm for robot manipulators. The idea there was to consider every actuator as an independent agent and to design a fuzzy controller for every actuator. In designing the fuzzy controller for an agent (actuator), the infinitesimal movement of the tip point resulting from the movement of the actuator is considered. In other words, in constructing the rule for an agent at any instant, all the other agents (actuators) are considered to be stationary. The rule is constructed in a way that the agent will make the best action in order to make the tip point move in the desired direction. The same idea is utilized here in developing the rules for the fuzzy controllers of actuators in both of the two controller systems.

In the **first epoch** there are the first and second phases. These phases will be distinguished by the value of the first actuator angle. The value *S (small)* for the first actuator angle *(θ_1)* will correspond to the first phase and *M (medium)* will correspond the second phase. The aim in this epoch is to rise up the tip point; therefore the second input of the controllers will be the height information, namely the *z* value of the tip point *(dz)*. The exact membership functions and rule tables will not be given here due to the lack of space. In the **second epoch** the aim is to carry the tip point to its final position, and to make the tip point stay there in a stable manner. In order to accomplish these following steps are performed: (1) *The rules that will carry the tip point from any initial position to any final position as if it should follow a straight-line path are written.* (2) *The rules that will stop the tip point at the final position are added.* With these ideas, first a controller that carries the tip point to any given destination is designed. Then the stability is satisfied at the destination point. Without this second step, the tip point would not have a good steady state behavior around the final position; it would be tending to make small circles. With these two steps a controller which carries the tip point to any destination and which stops it there is obtained.

In constructing the rules the following three items are considered:

1.*The first actuator will take care of the position difference in the y direction.*

2.*The second and third actuators will take care of the position difference in x and z directions, and they will behave as if they are in a vertical plane in every instant.*

3.*The first actuator will take care of the behaviors associated with the second and third phases. Namely, the first actuator will be fast in the second phase and slow in the third phase.*

To clarify the idea in the second item here will be given an example. The input to the second and third actuators are the angles of the second and third actuators (state information of the agents) and the position differences in *x* and *z* directions. For the state information of the second agent (for the second joint angle, θ_2) there are three membership functions: *S (small, $\pi/4$ radians), M (medium, $\pi/2$ radians), and B (big, $3\pi/4$ radians)*. For the state of the third agent (θ_3) the three levels take negative values: *NS ($-\pi/4$), NM ($-\pi/2$), and NB ($-3\pi/4$)*. The example here is for the state of *θ_2:M; θ_3:NS*. Fig.3 shows the graphical representation, assuming that the leg is in a vertical plane. Examining Fig.3, the following statements can be derived:

1.When *θ_2:M and θ_3:NS*, a positive change in θ_2 will result in a small negative change in *x* direction.

2. When *θ_2:M and θ_3:NS*, a positive change in θ_2 will result in a medium positive change in *y* direction.

3. When *θ_2:M and θ_3:NS*, a positive change in θ_3 will result in no change in *x* direction.

4. When *θ_2:M and θ_3:NS*, a positive change in θ_3 will result in a big positive change in *y* direction.

The total amount of action to change the *y* position of the tip point will be determined by the position difference in the *y* direction between the tip point and destination. Then this work will be divided between the agents. In order to change the *y* position, agent$_2$ should take a small portion and agent$_3$ should take a bigger portion of workload. The same logic will apply for the *x* direction. In order to change the *x* position, agent$_2$ should take full load of work and agent$_3$ should not have any contribution.

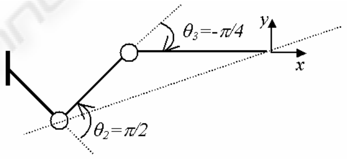

Figure 3: Graphical representation of the situation when *θ_2:M* and *θ_3:S.*

<u>*Simulation Results:*</u>

In Fig.4 the results obtained using the control system are shown. As it is seen in the upper two figures the controller is successful to produce tip point trajectories ‘resembling’ the optimal ones. The three figures below show the three joint velocity trajectories.

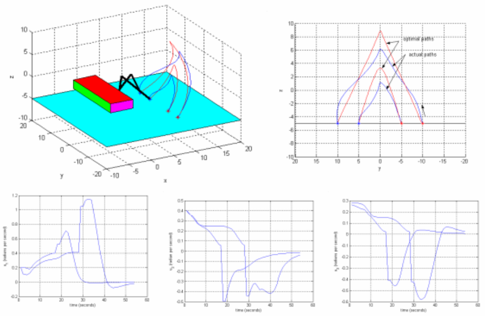

Figure 4: Simulation results.

REFERENCES

Bobrow, J.E., Martin, B., Sohl, G., Wang, E.C., Park, F.C, Kim, K., 2001. Optimal robot motions for physical criteria, *Journal of Robotic Systems*, 18(12):785-792.

Erden, M.S., Leblebicioğlu, K. and Halıcı, U., 2004. Multi-agent system based fuzzy controller design with genetic tuning for a service mobile manipulator robot in the hand-over task, *Journal of Intelligent and Robotic Systems*, 38: 287-306.

Kirk, Donald E, 1970. *Optimal Control Theory – An Introduction*, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

Liu, J.F., Abdel-Malek, K., 2000. Robust control of planar dual-arm cooperative manipulators, *Robotics and Computer-Integrated Manufacturing*, Vol.16, No. 2-3: 109-120.