Proceedings of the World Congress on Engineering 2007 Vol I
WCE 2007, July 2 - 4, 2007, London, U.K.

An Adaptive Cross-EntropyTuning of the PID Control for Robot Manipulators

Mehmet Bodur, *Member, IEEE*

***Abstract*— This paper proposes a population based adaptive tuning for dynamic position control of robot manipulators. The dynamic behavior of a robot manipulator is highly nonlinear, and the positional control is conventionally achieved by inverse dynamics feedforward and PID feedback controllers. The proposed method tunes the PID controller parameters using cross-entropy optimization to minimize the error in tracking a repeated desired trajectory in real-time. The stability of the system is granted by switching the inappropriate settings to a stable default using a real-time cost evaluation function.**

The proposed tuning method is tested on a two-joint planar manipulator, and on a planar inverted pendulum. The test results indicated that the proposed method improves the settling time and reduces the position error over the repeated paths.

***Index Terms*— Cross entropy optimization, dynamic control, PID tuning, population based optimization.**

I. INTRODUCTION

Robot manipulators are mainly positioning devices with multiple degrees-of-freedom (DoF). Dynamic position control of robot manipulators is a well known problem in control engineering due to the multi-input-multi-output nonlinear behaviour of their equation of motion. There are a large number of excellent survey books and articles on the control of the manipulators [1]-[4]. In a typical position control problem, the plant is characterized by the dynamic equation of motion of the manipulator, which describes the motion of the joint displacements corresponding to the applied joint torques. The controller is designed to generate appropriate joint torques to track a desired task space trajectory, or equivalently to track the corresponding desired joint space trajectory. The conversion of trajectories from the task space to the joint space requires kinematic modeling of the robot manipulator. The equation of motion is obtained by one of the methods such as Newton-Euler, Lagrangian, and their backward and forward recursive applications.

The decentralized PD and PID control is the standard method for position control of the manipulator joint displacement. PD control can achieve global asymptotic stability in tracking a trajectory in the absence of gravity. The integral action of the PID control further compansates any tracking offset due to gravitational forces [4]. Several methods were proposed in literature to obtain suitable propotional, integral and derivative controller gain settings: K_p, K_i and K_d. In absence of friction, the gains may be selected using a Lyapunov function candidate and LaSalle's Invariance Principle [4]. Adaptive PID settings were also proposed for obtaining proper local gain settings based on plant identification and pole placement techniques [5].

Cross-Entropy (CE) Method is an information theoretic method of inference about an unknown probability density from a prior estimate of the density and new expected values [7]-[8]. CE Method was further developed as a combinatorial optimization method to obtain the optimal solutions for rare-event problems and optimization of the scalar functions [8]-[10]. Similar to Monte-Carlo and simulated annealing methods, it is one of the optimization methods with the proven convergence to the optimal parameters.

This paper is organized as follows: Section 2 presents the kinematics, dynamic modeling and control of a robot manipulator. Section 3 introduces the cross-entropy method for optimization of the controller gains. Section 4 explains the objective and algorithm of the optimization, Section 5 and 6 contains the experimental details and the results of the proposed parameter switching on a two-link planar manipulator, and on an inverted pendulum system. Section 7 concludes the paper.

II. KINEMATICS DYNAMIC AND CONTROL

The forward kinematics relation maps the joint space position q to the end-effector pose

$$p = f_0(q) \tag{1}$$

using homogenous link-frame coordinate transformation matrices. Denavit-Hartenberg (D-H) convention provides the D-H transformation matrix A_i that maps a coordinate at i-th frame to the (i−1)-st frame [11].

The Lagrangian method is a systematic method to obtain the equation of motion of a multi-degree-of-freedom open chain mechanism. The *Lagrangian* function L is defined by $L = K - P$, where K is the kinetic energy, and P is the potential energy of the analyzed system. The partial derivatives of L provide a simple means of calculation for the joint force and torques. The derivatives of the D-H transformation matrices A_i can be expressed by

$$\frac{\partial A_i}{\partial q_i} = Q_i\, A_i \text{ , where } Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ for revolute joints.} \tag{2}$$

Manuscript received March 22, 2007.

Dr. M. Bodur is with the EMU- Computer Engineering Department, Mersin -10 Turkey. Phone: +90(533)876 4317; fax: +90(392)3650711; Submitted to ICCIS-07 for Evolutionary Systems topic. (e-mail: mehmet @ ieee . org).

This property provides a convenient form for the derivatives of the link transformation matrices ${}^0T_i$ with respect to the j-th generalized-joint displacement variable q_j .

$$U_{ij} = \frac{\partial\, {}^0T_i}{\partial q_j} = \frac{\partial(A_1A_2...A_j...A_i)}{\partial q_j} = A_1A_2 \ldots Q_jA_j \ldots A_i \;;\; \mathrm{j} \le \mathrm{i} \,. \quad (3)$$

Linear or revolute, the kinetic energy of a mass dm_i located at the position ${}^ir_{dm}$ and moving with the i-th coordinate frame is

$$d\,k_i = \tfrac{1}{2}\, d\, m_i\, {}^0v_i^{\mathrm{T}}\; {}^0v_i \,, \quad (4)$$

where ${}^0v_i$ is the velocity of mass m_i.

The total kinetic energy of the link requires the integration of $d\,k_i$ over the complete mass of the i-th link.

$$k_i = \int_{m_i} dk_i = \tfrac{1}{2}\mathrm{Trace}\left[\left(\Sigma_{p=1}^{i}\, \Sigma_{r=1}^{i} U_{ip}\, J_{pi}\;\; U_{ir}^{\mathrm{T}}\, \dot{q}_r\, \dot{q}_p\right)\right], \quad (5)$$

where $J_{pi}=\int_{m_i} {}^ir_{dm}\; {}^ir_{dm}^{\mathrm{T}} dm_i$ is the pseudo-inertia matrix of the i-th link. The potential energy of i-th link due to the gravitational field is expressed using the mass m_i, the center of mass ${}^ic_m$ and the gravitational acceleration vector g_a

$$p_i = -\, m_i\, g_a^{\mathrm{T}}\;\, {}^0T_i\;\, {}^ic_m \,. \quad (6)$$

Finally, the Lagrangian of the complete system is obtained

$$L = K - P = \tfrac{1}{2}\, \Sigma_{i=1}^{n}\, \Sigma_{k=1}^{i}\, \Sigma_{r=1}^{i}\, \mathrm{Trace}\left(U_{ik}\, J_{pi}\;\; U_{ir}^{\mathrm{T}}\right) \dot{q}_r\, \dot{q}_p - \Sigma_{i=1}^{n}\, m_i\, g_a^{\mathrm{T}}\;\, {}^0T_i\, {}^ic_m \quad (7)$$

and its derivatives gives the generalized joint-force at the joint-j.

$$\tau_j = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j} \,, \quad (8)$$

$$\tau_j = \Sigma_{i=j}^{n}\, \Sigma_{k=1}^{n}\, \mathrm{Trace}\left(U_{ik}\, J_{pi}\, U_{ij}^{\mathrm{T}}\right) \ddot{q}_k + \Sigma_{k=1}^{n}\, \Sigma_{r=1}^{n}\, \mathrm{Trace}\left(U_{ikr}\, J_{pi}\, U_{ij}^{\mathrm{T}}\right) \dot{q}_k\, \dot{q}_r - \Sigma_{i=j}^{n}\, m_i\, g_a^{\mathrm{T}}\, U_{ij}\, {}^ic_m \,. \quad (9)$$

Further, (8) and (9) can be organized to Uicker-Kahn form.

$$\tau_i = \Sigma_{i=j}^{n}\, D_{ij}\, \ddot{q}_j + \Sigma_{i=j}^{n}\, \Sigma_{k=1}^{n}\, C_{ijk}\, \dot{q}_j\, \dot{q}_k + g_i. \quad (10)$$

where $D_{ij} = \Sigma_{i=j}^{n}\mathrm{Trace}\left(U_{ik}\, J_{pi}\, U_{ij}^{\mathrm{T}}\right)$; $C_{ijk} = \Sigma_{r=1}^{n}\mathrm{Trace}\left(U_{ikr}\, J_{pi}\, U_{ij}^{\mathrm{T}}\right)$, and $g_j = -\, \Sigma_{i=j}^{n}\, m_i\, g_a^{\mathrm{T}}\, U_{ij}\, {}^ic_m$, are the inertial, coriolis, and gravitational terms of the equation of motion. The equation is also written in a matrix form [4]

$$\tau = D(q)\, \ddot{q} + C(q, \dot{q}\,)\, \dot{q} + g(q) \,. \quad (11)$$

A constant field dc-motor converts the total control action by an almost linear relation of rotor current i_{rotor} to the joint torque $\tau = k_i \times i_{rotor}$. Accordingly, without loss of generality, the torque coefficient k_i is assumed k_i=1 so that the joint torques are determined directly by the control action.

A fully actuated rigid manipulator has an independent control input for each degree-of-freedom, which simplifies its decentralized control. Robots with flexible links or joints have control problems and which may require singular perturbation and two-time-scale control techniques [4], [5].

The equation of motion is highly nonlinear to apply independent PID control directly to each joint motion. The usual method is to apply the expected joint torques for the desired trajectory by a feedforward control, and correct any deviation from the trajectory by the feedback loop of the PID control. The feedforward control action τ_c is obtained by the sum of the anticipated gravitational and inertial terms, so that all coriolis forces remains as a disturbance to the feedback loop

$$\tau_c = D_c(q_d)\, \ddot{q}_d + g_c(q_d) \,, \quad (12)$$

where D_c and g_c are the inertial coefficients and gravitational torques of the anticipated equation of motion; q_d is the desired trajectory in joint space. The discrepancy of the anticipated equation of motion from the actual dynamics of the manipulator is expected to contribute to the disturbance torques, and thus a feedback control loop is inevitable for the stability of the control action.

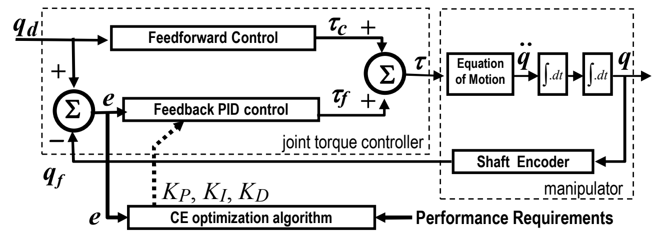


Fig. 1. Position Control System.

The difference of the desired position $q_d(t)$ and the measured position $q_f(t)$ is called the displacement error $e(t)$ of the control system. A proportional gain provides main correction of the error, an integral gain provides correction of the offset, and a derivative gain provides faster response of the controlled system.

$$\tau_f = K_P\, e(t) + K_I \int_{t_0}^{t} e(t)\, dt + K_D\, \dot{e}(t) \,. \quad (13)$$

Assuming that the anticipated feedforward control law can be predicted by the mechanical properties of the manipulator, the PID parameters remain to be determined for the optimum tracking of the specified joint space trajectory. The following section contains the cross entropy method, which is employed to find the optimum PID gains.

III. Cross Entropy Method

In the optimization of the PID gains, The closed loop control system is considered as a stochastic system with the controller parameters. Cross Entropy method is a population based optimization algorithm that utilize the scores of the trial runs optimal in the information rhetoric meaning.

Let X be the set of real valued states, and let the scores S be a real function on X. CE-method targets to find the minimum of S over X, and the corresponding states x^* satisfying the minimum

$$\gamma^* = S(x^*) = \min_{x \in X} S(x), \quad (14)$$

by employing importance sampling and minimizing the cross entropy between the samples of a family of succeeding probability mass function $f(\text{-}, \upsilon_k)$. A naive random search can find an expected value for x^* and determine γ^* with probability 1, if some of the scores $S(x)$ for the random states x can satisfy the minimum. However, methods like Crude Monte-Carlo requires considerable computational effort because it uses homogeneously distributed random states in searching x^*. CE method provides a methodology for creating a sequence of vectors x_0, x_1 ... and levels γ_0, γ_1, ... such that γ_0, γ_1, ... converges to γ^* and x_0, x_1 , ... converges to x^*.

Define a collection of functions $\{ H(- ; \gamma) \}$ on X, via

$$H(x ; \gamma) = I_{\{S(x^{(i)}) > \gamma\}} = \begin{cases} 1 & \text{if } S(x) \le \gamma \\ 0 & \text{if } S(x) > \gamma \end{cases} . \quad (15)$$

for each $x \in X$, and threshold $\gamma \in R$. Let $f(-; \upsilon)$ be a family of probability mass function (pmf) on X, parameterized by a real valued vector $\upsilon \in R$. Consider the probability measure under which the state x satisfies the threshold γ

$$\ell_\upsilon (\gamma) = P_\upsilon (S(X) \ge \gamma)$$
$$= \Sigma_x H(x; \gamma) f(-; \upsilon) = E_\upsilon H(X; \gamma) , \quad (16)$$

where E_υ denotes the corresponding expectation operator. It converts the optimization problem to an associated stochastic rare event problem. Using the Importance Sampling (IS) simulation, the unbiased estimator of ℓ with the random sets of states $x^{(1)}, \ldots$ taken from different independent pmf $f(x, \upsilon)$ and $g(x)$ is

$$\hat{\ell} = \frac{1}{N} \Sigma_{i=1}^{N} I_{\{S(x^{(i)}) > \gamma\}} W (x^{(i)}) , \quad (17)$$

where $W(x) = f(x, \upsilon)/g(x)$ is the likelihood ratio. Searching the optimal importance sampling density $g^*(x)$ is problematic, since determination of $g^*(x)$ requires ℓ to be known. Instead, the optimum tilting parameter υ^* of a pmf $f(\text{x}, \upsilon)$ reduces the problem to scalar case.

The tilting parameter υ can be estimated by minimizing Kullback-Leibler "distance" (also called cross entropy) between the two densities $f(x)$ and $g(x)$

$$CE(f, g)= \int f(x) \ln\frac{f(x)}{g(x)} dx . \quad (17)$$

After reducing the problem to tilt parameter υ, the cross-entropy between $f (x, \upsilon)$ and the optimal distribution $f(x, \upsilon^*)$ is described by

$$CE(\upsilon, \upsilon^*) := E_a\left[\frac{I_{\{S(X)\ge\gamma\}}}{c} \ln \frac{I_{\{S(X)\ge\gamma\}} f(x,\upsilon^*)}{c\ f(x,\upsilon)}\right], \quad (18)$$

where $c = \int I_{\{S(X)\ge\gamma\}} f(x)\, dx$. The optimal solution υ^* is obtained by solving v* from $\min_{\upsilon} CE_N(\upsilon)$, where

$$CE_N(\upsilon) = E_{\upsilon 1} = \frac{1}{N} \Sigma_{i=1}^{N} I_{\{S(x^{(i)}) > \gamma\}} W (x^{(i)}, \upsilon, \upsilon^*) , \quad (19)$$

For example, if all γ is equal to γ^*, in that case $\ell_\upsilon (\gamma) = f(x^*; \gamma)$, which typically would be a very small number. $\ell_\upsilon (\gamma^*)$ is estimated by Importance Sampling with $\upsilon_k = \upsilon$

$$\upsilon^* = \underset{\bar{\upsilon}}{\operatorname{argmax}} \ \Sigma_{i=1}^{N} E_\upsilon H(X ; \gamma) \ln f (X ; \bar{\upsilon}) , \quad (20)$$

and using $X^{(i)}$, which are generated from pmf $f (X ; \bar{\upsilon})$ by

$$\underset{\bar{\upsilon}}{\operatorname{argmax}} \ \Sigma_{i=1}^{N} H(X^{(i)} ; \gamma) \ln f (X^{(i)} ; \bar{\upsilon}) . \quad (21)$$

However, the estimator of υ^* is only valid when $H(X^{(i)} ; \gamma) = 1$ for enough samples.

The CE-algorithm consist of the two phases: 1) Generate random samples using $f(-;\upsilon_k)$, and calculate the estimate of the objective function; 2) Update $f(-;\upsilon_k)$ on the basis of the data collected in the first phase via the CE method. The main CE Algorithm is

1) Initialize k=0, and choose initial parameter vector $\upsilon_0 = \upsilon$.

2) Generate a sample of states $x^{(1)}, \ldots, x^{(M)}$ according to the pmf $f(-;\upsilon_k)$.

3) Calculate the scores $S(x^{(i)})$ for all i, and order them from the biggest to the smallest,

$$s_1, \ge \ldots \ge s_N .$$

Define $\gamma_k = s_{[\rho N]}$, where $[\rho N]$ is the integer part of ρN, so that $\gamma_k > \gamma$, set $\gamma_k = \gamma$, this yields a reliable estimate by ensuring the target event is temporarily made less rare for the next step if the target is not reached by at least a fraction of the samples.

4) For j=1, ..., 5, let

$$\upsilon_{k+1,j} = \frac{\Sigma_{i=1}^{M} I_{\{S(x^{(i)}) > \gamma_k\}} W (x^{(i)} ; \nu , \upsilon_k) x_j^{(i)}}{\Sigma_{i=1}^{M} I_{\{S(x^{(i)}) > \gamma_k\}} W (x^{(i)} ; \nu , \upsilon_k)} . \quad (22)$$

5) Increment k and repeat steps 2, to 5, until the parameter vector has converged. Let υ^* denote the final parameter vector.

6) Generate a sample $x^{(1)}, \ldots, x^{(N)}$ according to the pmf $f(-; \upsilon_k)$ and estimate ℓ via the IS estimate

$$\hat{\ell} = \frac{1}{N} \Sigma_{i=1}^{N} H (x^{(i)}) W (x^{(i)} ; \upsilon , \upsilon^*) . \quad (23)$$

Step (6) of the CE algorithm is not necessary in optimization of the scalar cost functions, since the goal is to minimize the cost rather than calculation of the probability of a rare event.

IV. Optimization Criteria and Test Method

The optimization of motion control parameters of a manipulator using reinforcement, genetic, or other population methods comes across important problems in testing the generated parameter sets. These methods are mostly applied on simulations, because the unrestricted population may contain unstable controller settings, which may cause misoperation, and even damage of the robot system. Lin solved the problem by testing stability properties of the parameters before applying them to the dynamic control system [12]. His method is based on restricting the search space to a narrow region that guarantees Liapunov stability of the system.

This paper proposes a simpler, and a more flexible method for the application of population based optimization methods. The proposed method assumes that, starting with a stable control parameter set, the system stability is guaranteed along the trajectory for a tolerable deviation from the path. The population based algorithm generates a list of controller parameter sets. The evaluation module switches the control parameters of the PID controllers, and calculates the cost function of the parameter set along a test period. If the cost function does not exceed the specified critical level along the complete test period, a cost score is calculated from the cost criteria, which may include any absolute, derivative, integral and quadratic terms of tracking error. Whenever the cost function exceeds a prespecified critical value the control parameters of the PID controllers are switched back to the stable control parameter set, which reduces the tracking errors, and prepares the system for the next test. In this case, a high cost score is returned for the tested unsuccessful parameter set. The following algorithm is a demonstration of the proposed method on CE optimization algorithm.

Algorithm:

1) At initialization, specify a stable controller parameter set w_s, a real-time cost function $J(t, p(t), p_d(.))$, a cost threshold J^*, and a cost score function $S(w, p_d(.))$. Initialize k=0, and j=1, and choose initial $\upsilon_k = w_s$. Specify initial γ_k and 0<ρ<1, and so that "the ratio of stable parameter vectors to all parameter vectors" is higher than ρ. This ensures reliable estimates of the target. Select the coefficients $0 < \alpha \le 1$ and $0 < \beta \le 1$ to update υ_{k+1} and γ_{k+1} smoothly.

2) Generate a population of states $w^{(1)}, ..., w^{(M)}$ according to the pmf $f(\gamma_k;\upsilon_k)$.

3) For each $w^{(i)}$, at the beginning of a trajectory period $p_d(.)$ switch the controller settings to $w^{(i)}$, and during the period, evaluate $J(t, p, p_d)$. Whenever $J(t)$ exceeds J^* switch the controller settings to w_s , so that system stays stable, and the trajectory deviation remains within tolerable limits. At the end of the trajectory period, calculate the cost score of the sample, $s^{(i)}= S(w^{(i)}, p_d)$.

4) Order the scores from the biggest to the smallest, i.e.,

$$s_1, \ge ... \ge s_N .$$

Use $\gamma_k = s_{[\rho N]}$, where $[\rho N]$ is the integer part of ρN, to select the elite subset of population. Estimate w_{k+1} , and γ_{k+1} from the mean and standard deviation of the parameters in the elite subset. Update

$$w_{k+1} = \alpha\, w_{k+1} + (1-\alpha) w_k,$$

$$\beta_m = \beta - \beta\,(1-\tfrac{1}{k})\, q_s , \text{ and}$$

$$\gamma_{k+1} = \beta_m\, \gamma_{k+1,j} + (1-\beta_m)\,\gamma_k .$$

If υ_{k+1} converges to υ_k , increment j=j+1; else reset j=0.

5) Repeat the steps (2) – (4) until j=5, (i.e., the last 5 iterations converge to the target w^*.)

The proposed method provides a higher degree of flexibility in specifying the cost function compared to many other controller parameter optimization techniques which are based on plant identification or state estimation techniques. For example, in the following simple demonstration, the cartesian absolute path deviation ($e(t) = |\, p(t) - p_d(.)|$) is used instead of using the conventional trajectory tracking error ($e(t) = q(t) - q_d(t)$) in joint space.

V. Simple 2-Link (2R) Demonstration

Consider the two link manipulator with two revolute joints shown in Figure 1. The length of the links L_1 and L_2 are respectively b_1, and b_2. The masses m_1 and m_2 are homogenously distributed along the links L_1 and L_2. Coordinate frames ${}^0T$, ${}^1T$ and ${}^2T$ are assigned for the base, L_1, and L_2 by applying D-H convention as shown in Fig-1. The symbolic equation of motion for this planar manipulator is derived in symbolic toolbox of MATLAB using Lagrangian formulation

$$\begin{aligned}\tau_1 = &\left(\tfrac{1}{3} m_1 b_1^2 + \tfrac{1}{3} m_2 b_2^2 + m_2\, b_1\, b_2\, C_2 + m_2 b_1^2\right) \ddot{q}_1 \\ &+\left(\tfrac{1}{3} m_1 b_1^2 - m_1 b_1 b_2 - \tfrac{1}{2} m_1 b_1^2\, C_2 + m_1 b_2^2 + m_1 b_1 b_2\, C_2\right) \ddot{q}_2 \\ &- m_2 b_1 b_2\, S_2\, \dot{q}_1 \dot{q}_2 - \tfrac{1}{2} m_2 b_1 b_2\, S_2\, \dot{q}_2^2 \\ &+ \tfrac{1}{2} g \left(b_1 m_1 C_1 + b_2 m_2 C_{12} + 2\, b_1 m_2 C_1\right) , \qquad (24)\end{aligned}$$

$$\begin{aligned}\tau_2 = &+\left(\tfrac{1}{3} m_1 b_1^2 - m_1 b_1 b_2 - \tfrac{1}{2} m_1 b_1^2 C_2 + m_1 b_2^2 + m_1 b_1 b_2 C_2\right) \ddot{q}_1 \\ &+ \tfrac{1}{3} m_2 b_2^2\, \ddot{q}_2 + \tfrac{1}{2} m_2\, b_1 b_2\, S_2\, \dot{q}_1^2 + \tfrac{1}{2} g\, m_2\, b_2 C_{12} , \qquad (25)\end{aligned}$$

where C_i, and S_i, denote $\cos(q_i)$ and $\sin(q_i)$; C_{ij}, and S_{ij}, denote $\cos(q_i + q_j)$ and $\sin(q_i + q_j)$.

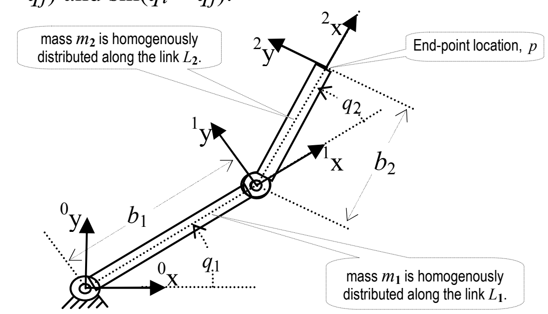


Fig. 2. Parameters of the simulated manipulator with 2-revolute joints.

In the simulation, the equation of motion is integrated for m_1 = 5 kg, m_2 = 3 kg, b_1 = 0.5 m and b_2 = 0.4 m, reducing the simulation model to

$$\begin{aligned}\tau_{s,1} = &\left(\tfrac{199}{150} + \tfrac{3}{5} C_2\right) \ddot{q}_1 + \left(\tfrac{13}{60} + \tfrac{3}{8} C_2\right) \ddot{q}_2 \\ &- \tfrac{3}{5} S_2\, \dot{q}_1 \dot{q}_2 - \tfrac{3}{10} S_2\, \dot{q}_2^2 + \tfrac{1}{2} g \left(\tfrac{11}{2} C_1 + \tfrac{6}{5} C_{12}\right) \\ \tau_{s,2} = &\left(\tfrac{13}{60} + \tfrac{3}{8} C_2\right) \ddot{q}_1 + \tfrac{4}{25} \ddot{q}_2 + \tfrac{3}{10} S_2\, \dot{q}_2^2 + \tfrac{3}{5} g\, C_{12}. \qquad (26)\end{aligned}$$

All joint and actuator friction forces are assumed to be zero to prevent their stabilizing effect on the closed loop stability of the system. The feedforward control force τ_c is calculated only from the inertial and gravitational terms of a similar model ($\tau_c = D_c(q_d)\, \ddot{q}_d + g_c(q_d)$) but with a higher load mass m_2=3.5kg.

$$\tau_{c,1}(q) = \left(\tfrac{887}{600} + \tfrac{7}{10} C_2\right) \ddot{q}_1 + \left(\tfrac{13}{60} + \tfrac{3}{8} C_2\right) \ddot{q}_{2d} + \tfrac{1}{2} g \left(6 C_1 + \tfrac{7}{5} C_{12}\right)$$

$$\tau_{c,2}(q) = \left(\tfrac{13}{60} + \tfrac{3}{8} C_2\right) \ddot{q}_1 + \tfrac{7}{10} g\, C_{12} . \qquad (27)$$

The test path is selected on a line segment in cartesian workspace starting from the point $({}^0x, {}^0y) = (0.2, 0)$ to the point (0.7, 0.5) through the via-point (0.4, 0.2). The test trajectory is calculated for Δt = 1 ms intervals using parabolic blend with linear segments trajectory generation method with via points [11] under the constraints: maximum linear velocity= 1m/s, and linear acceleration 1 m/s^2. The points of the desired trajectory are shown in Fig. 2 for every 50 ms periods.

Stable control parameters were initialized to be w_s=[K_{1P}, K_{1I}, K_{1D} K_{2P}, K_{2I}, K_{2D}] = [5000, 50, 100, 5000, 50, 100]; and cost function was $J(t, p(t), p_d(t)) = \|p(t), p_d(t)\|$ which is the cartesian distance of end-point position p(t) to the desired trajectory $p_d(t)$; the cost threshold is J^*= 0.002 m; the cost score function was $S(w, p_d(.)) = \max_t J(t)$ over one trajectory period; the standard deviation of the gaussian pdf was $\gamma_k = 0.5\, w_s$; population size was N= 60; elite population ratio was ρ=0.05; smooth update coefficients were $\alpha = 0.9$; $\beta = 0.9$; and q_s=5. The convergence plot of cartesian tracking displacement error is shown in Fig. 5. The CE-optimization provided almost 50% reduction of cartesian displacement error. The oscillatory action of actuator torques may be prevented by assigning an additional cost on the derivative of the controller output.

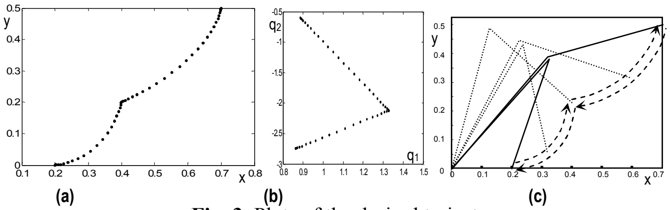


Fig. 3. Plots of the desired trajectory
(a) x vs. y (in meters), (b) q_1 vs. q_2 (in radians),
(c) the movement of the links in x-y plane.

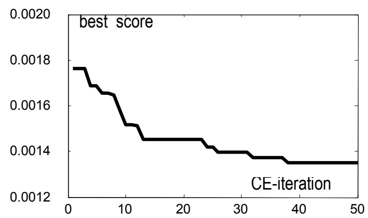


Fig. 4. Convergence plot of the best-score for two-link manipulator.

VI. Control of an Inverted Pendulum

The inverted pendulum is a well analyzed classical control problem [12], [13]. The objective is to track a desired cart trajectory with a small deviation e_x by applying an external force f on the cart along x-axis while keeping the pendulum stable in vertical position. The mechanical parameters of the system are introduced in Fig.6.

The equation of motion of the system is expressed by

$$\begin{bmatrix} m_c+m_p & m_p b\, C_q \\ m_p b\, C_q & J+m_p b^2 \end{bmatrix}\begin{bmatrix}\ddot{x}\\ \ddot{q}\end{bmatrix}+\begin{bmatrix} C_c & -m_p b\, \dot{q}\, S_q \\ 0 & C_p\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{q}\end{bmatrix}+\begin{bmatrix}0\\ m_p b\, g\, S_q\end{bmatrix}=\begin{bmatrix} f\\ 0\end{bmatrix}, \quad (28)$$

where C_q and S_q are cos(q) and sin(q), g is the acceleration due to gravity, m_c is the mass of cart, m_p is the mass of the pole, b is the half length of the pole, C_c is the friction coefficient of cart, C_p is the friction of the pole, J is the mass moment of inertial about the pole, and f is the force applied to the cart [12]. Two PID controllers were cascaded to reduce the tracking errors $e_x = x - x_d$, and $e_q = q - q_d$, asymptotically to zero as shown in Fig 7. Oscillatory behavior was expected since no predictive or inverse dynamic feedforward action employed. In the tests, the desired cart trajectory was specified by a square function switching between –0.01 and 0.01m with a period 25s. Both the cart mass m_c, and pole mass m_p were selected 1 kg., and the half-length of the pole b was taken 2m.

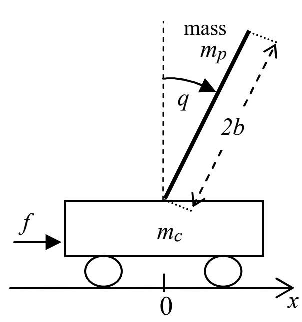


Fig. 6. Inverted Pendulum System.

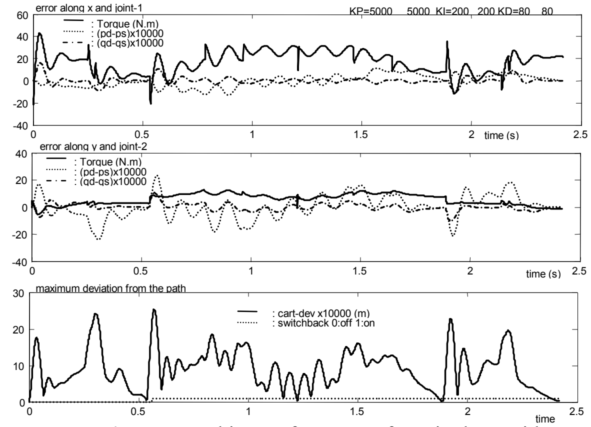


Fig. 5 a. Tracking performance of manipulator with initial stable control parameter set w_S.

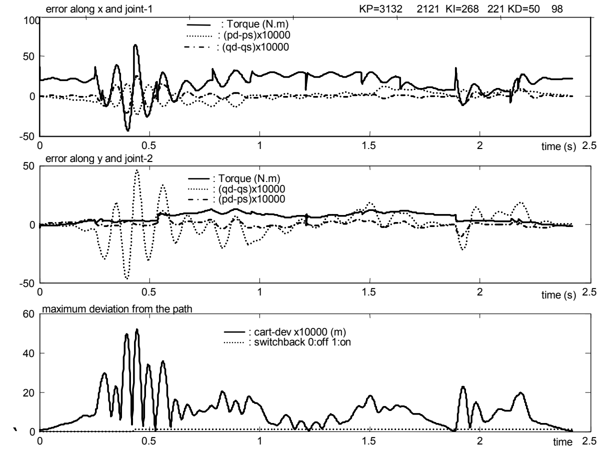


Fig. 5 b. An unstable parameter set causes excessive cartesian error at t=0.42, and control is switched back to w_S.

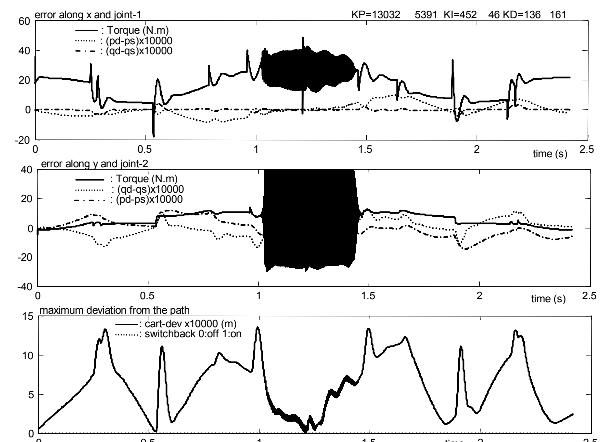


Fig. 5 c. The best parameter set reduced the cartesian error to 50%, but bounded oscillatory action of torques were observed because of the high proportional gain.

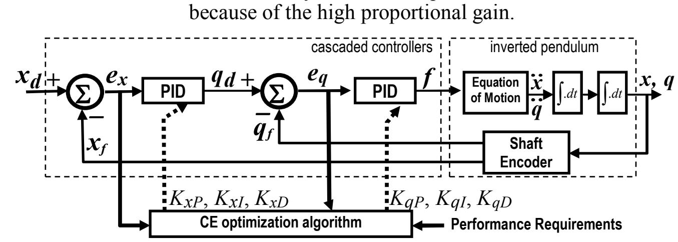


Fig. 7. Block diagram of inverted pendulum system for parameter optimization

ISBN:978-988-98671-5-7

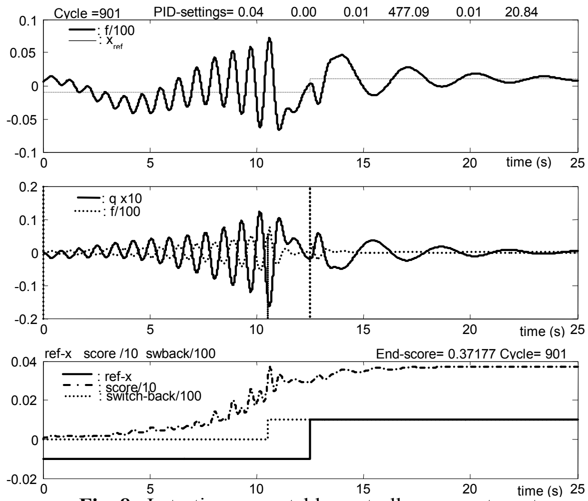
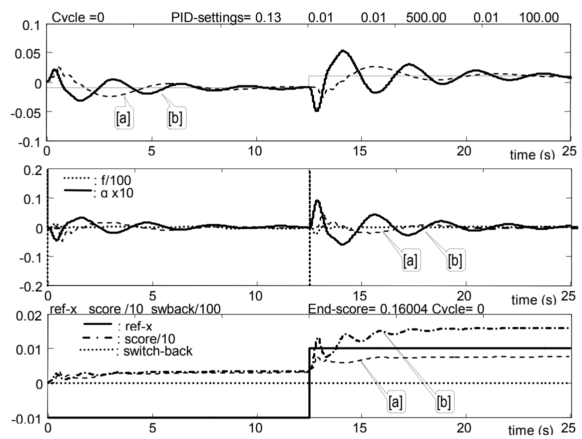


Fig. 8. In testing an unstable controller parameter set, parameters were switched to w_S at t =11s.



[a] tracking with CE- optimized w^* ; [b] tracking with initial w_S .
Fig 9. Tracking before and after parameter adaptation on an inverted pendulum control.

The pole inertia J_p was taken 16/3. The friction coefficients on the cart C_c, and on the pole C_p were taken 0.01, and the gravitational acceleration was assumed 9.8 m/s^2. Equation of motion was integrated over Δt=5ms time steps using trapezoidal method. Initially, the inverted pendulum was stabilized by the control parameter set w_s =[K_{xP}, K_{xI}, K_{xD}, K_{qP}, K_{qI}, K_{qD}] = [0.13 0.01 0.01 500 0.01 100]. The cost function was specified using the quadratic displacement errors $e_x{}^2$, $e_q{}^2$, and their integrals:

$$S(w^{(i)}, p_d, t) = 20\, e_x{}^2 + 30 \int e_x{}^2 dt + 50\, e_q{}^2 + 100 \int e_q{}^2\, dt. \quad (29)$$

For each $w^{(i)}$, the parameter set was switched at the beginning of the desired trajectory cycle, for 50s. The value of $S(w^{(i)}, p_d, t)$ at the end of the trajectory cycle was used for the score $s^{(i)}$. Control parameters were switched back to the stable parameter set w_s at the threshold of $S(w^{(i)}, p_d) > 0.3$. For the CE method, the initial standard deviation of the gaussian pdf was selected $\gamma_k = 0.5\ w_s$; population size was N = 60; elite population ratio was $\rho = 0.05$; smooth update coefficients were selected to be $\alpha = 0.9$, $\beta = 0.9$, and $q_s = 5$.

The plot of x and q for the initial stable parameter set, and for the cost optimized parameter set is shown in Fig. 5, and a typical control parameter switching due to unstable character of the tested parameter set $w^{(i)}$ is plotted in Fig. 8. The initial and final performance of the cart and pole system is shown in Fig. 9.

VII. Conclusion

This study proposes a real time evolutionary search method that provides stability by switching the unstable control parameters with a prespecified stable parameter set whenever a tracking error based cost function exceeds a threshold. The proposed method has been tested successfully on two cases, on an open chain manipulator dynamic control, and on the control of an inverted pendulum. The second problem is relatively difficult since without a feedforward or predictive control the system is stable only in a very narrow band of PID parameter settings.

During the tests none of these two systems has been fallen into non controllable states. Even though some of the oscillatory controller settings collected better scores than similar but less oscillatory settings, the mean of the elites always remained preferably stable. In both cases, a considerable reduction of the specified cost function reaching up to 50% is achieved by cross-entropy optimization method during the progress of its repetitive operation.

Acknowledgment

Special thanks go to my colleague Dr. Adnan Acan for discussing several issues on implementation of the proposed method. Thanks to the graduate robotic course students: Maneli Badakshan, Vassilya Abdulova, Dilek Beyaz, Duygu Çelik for their efforts in reproduction of the trajectory planning algorithms; M. Reza Najiminaini, and A. A. Abed for their effort in deriving the symbolic equation of motion.

References

[1] Paul, R.C., "Modeling, Trajectory Calculation, and Servoing of a Computer Controlled Arm" Stanford A.I. Lab, A.I. Memo 177, Stanford, CA, Nov.1972.
[2] Paul, R.P., Robot Manipulators: Mathematics Programming, and Control, MIT Press. Cambridge 1982.
[3] Fu, K.S., Gonzales, R.C., Lee C.S.G., Robotics Control, Sensing, Vision, and Intelligence, Mc.Graw-Hill International Ed. Singapore, 1988.
[4] Mark W. Spong", "Motion Control of Robot Manipulators", online article: url:citeseer.ist.psu.edu/165889.html.
[5] Bodur M., Sezer M. E., Adaptive control of flexible multilink manipulators, Int.J.Control, vol.58, no.3, pp 519-536, 1993.
[6] Rubinstein, R. Y. and Kroese, D.P. (2004). The Cross-Entropy Method:A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning. Springer-Verlag, New York.
[7] Kullback S. Information theory and statistics. NY: Wiley, 1959.
[8] J. Shore and R. Johnson. Properties of cross-entropy minimization. IEEE Trans. Information Theory, 27(4):472-482, 1981
[9] D.P. Kroese and K.-P. Hui: Applications of the Cross-Entropy Method in Reliability, Computational Intelligence in Reliability Engineering (SCI) 40, 37–82 Springer-Verlag Berlin Heidelberg (2007).
[10] P.T. de Boer, D.P. Kroese, S. Mannor, and R.Y. Rubinstein. A tutorial on the cross-entropy method. Annals of Operations Research, 2004
[11] Uicker, J. J., Denavit, J. , Hartenberg, R. S., An iterative method for the displacement analysis of spatial mechanisms. Journal of Applied Mechanics. 26. 309-314, 1964.
[12] S. B. Niku, Introduction to Robotics, Analysis, Systems, Applications. Prentice Hall Inc. pp 153-165, 2001.
[13] Chuan-Kai Lin, Reinforcement learning adaptive fuzzy controller for robots, Fuzzy Sets and Systems, Elsevier Vol.137, pp.339-352, 2002.
[14] Shozo Mori, et al., Control of unstable mechanical system, control of pendulum, Internat. J. Control 23 (5) pp. 673–692, 1976