2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

Generating Optimal Trajectory of Humanoid Arm that Minimizes Torque Variation using Differential Dynamic Programming

In-Won Park, Young-Dae Hong, Bum-Joo Lee and Jong-Hwan Kim

Abstract— **This paper proposes an optimal control method to generate a minimum-torque change trajectory of humanoid arm by using a differential dynamic programming (DDP). Since DDP is a locally optimal feedback controller, the convergence is not guaranteed unless DDP starts with a good reference trajectory for high-dimensional nonlinear dynamical systems. The reference trajectory is generated by using the minimum-jerk trajectory method, and then the corresponding torque profile is obtained by using the computed-torque method. This reference trajectory is not optimal because it is solely based on the kinematics of the system. In the proposed method, the rate of torque change is used as control input in DDP to generate the optimal trajectory, which concurrently minimizes the torque variations and considers the dynamics of the system. The effectiveness of the proposed method is verified by computer simulations for generating the optimal trajectory of a 7 degrees-of-freedom (DOF) MyBot humanoid arm in Webots simulator.**

I. Introduction

Learning optimal controllers for high-dimensional and nonlinear dynamical systems in continuous state, action, and time spaces have received a great deal of attention in recent times. Learning controllers having discretized state and action space confront with the dimensionality of domain and the computational complexity. To cope with the problem, trajectory-based techniques, such as differential dynamic programming (DDP) [1]-[5] and iterative linear quadratic regulator (ILQR) [6], were proposed to compute a locally optimal feedback control law.

DDP is a local method, based on the dynamic programming, and uses a quadratic expansion of cost function. Each iteration comprises of two sweeps: a backward and a forward sweep. The local model of value function is generated in the backward sweep. Then, the open-loop and feedback terms are combined in the forward sweep to generate a new control sequence. DDP has a second-order convergence because it uses the first (Jacobian) and second (Hessian) spatial derivatives of the value function. Due to the local model characteristic of DDP, the convergence is not guaranteed unless it starts with a good reference trajectory for high-dimensional problems.

There were many researches related to DDP by using the trajectory-based local models. Atkeson et al. [3], [4] represented both policies and value functions nonparametrically along the trajectories to speed up the global optimization in dynamic programming. Morimoto et al. [7] and Abbeel et al. [8] applied a DDP-based controller to a robust biped walking robot and autonomous helicopter flight, respectively. Tassa et al. [9] introduced the receding horizon DDP algorithm to construct more stable and robust controllers based on a library of local trajectories. Atkeson et al. [5] combined random sampling of states and DDP to stabilize a standing biped robot against external disturbances.

This paper proposes a novel control method to generate the optimal trajectory, which minimizes the rate of torque change by using DDP. DDP requires a good reference trajectory because it is a locally optimal feedback control method. In the initial step, the reference trajectory is generated by using a minimum-jerk trajectory method, and then the corresponding torque profile is obtained by using a computed-torque method. This trajectory is not optimal because the minimum-jerk trajectory only considers the kinematics of system and neglects the dynamics.

The contribution of this paper is to optimize the reference trajectory in order to generate the optimal trajectory that minimizes the torque variation. The state of DDP consists of joint position, joint velocity, and joint torque, where the control input of DDP consists of the rate of torque change. The cost function is defined in the quadratic form of the posture and its velocity errors, and the rate of torque changes. Another important contribution in this paper is developing an optimal control method, which can be applied to highly nonlinear dynamical system.

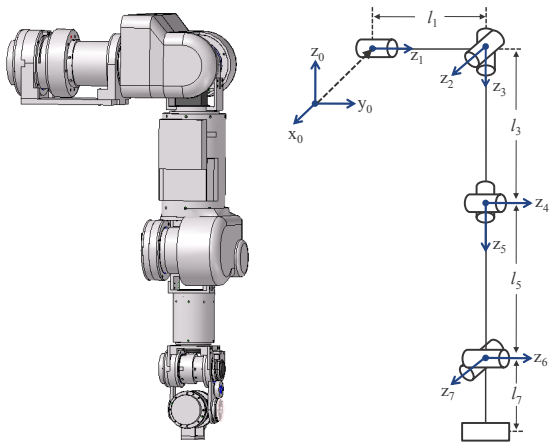
The proposed optimal control method is verified with a 7 DOF MyBot humanoid arm in Webots simulator. Simulation results confirm that the optimized trajectory of MyBot arm requires less torque and results in a shorter path in comparison to the reference trajectory. However, the optimized trajectory is unable to reach the final posture precisely because it is generated based on the dynamics of MyBot arm and the gravitational force. In addition, the precision of end-effector posture is highly dependent of the values of cost-weighting matrices in DDP.

This paper is organized as follows: Section II describes the MyBot arm including the formulation of kinematics, dynamics, and Jacobian matrix. Section III explains the process of generating a reference trajectory by using the damped least-squares method, the minimum-jerk trajectory method, and the computed-torque method. Section IV presents the proposed optimal controller and quadratic expansion of the cost function. Section V presents the simulation results and concluding remarks follow in Section VI.

I.-W. Park, Y.-D. Hong and J.-H. Kim are with the Department of Electrical Engineering, KAIST, 335 Gwahangno (373-1 Guseong-dong), Yuseong-gu, Daejeon, 305-701, Republic of Korea, e-mail: ({iwpark, ydhong, johkim}@rit.kaist.ac.kr).

B.-J. Lee is with the Department of Electrical Engineering, Myongji University, San 38-2 Nam-dong, Cheoin-gu, Youngin, Gyeonggi-do, 449-728, Republic of Korea, e-mail: (leebumjoo@gmail.com).

978-1-4673-1405-3/12/$31.00 ©2012 IEEE



(a) Snapshot of MyBot arm at the initial posture. (b) Configuration and frame assignments.

Fig. 1. 7 DOF MyBot arm.

II. Humanoid Arm Manipulator

The simulation model of MyBot arm having 7 DOF is used in the present work, which consists of seven DC motors with harmonic drives to provide control accuracy, gear reduction, and sufficient power. It is mechanically designed such that four strain gauges will be attached on the surface of each shaft in order to measure accurate values of applied torque by sensing the actual shaft deflection caused by a twisting force. The snapshot of MyBot arm at the initial posture and the frame assignments are shown in Fig. 1. The base frame $\{0\}$ is located at the first joint of MyBot arm.

SD/FAST is used to obtain both kinematics and dynamics equation of MyBot arm [10]. SD/FAST is a software developed by Symbolic Dynamics Inc., which uses Kane's formulation and symbolic manipulation to generate the nonlinear equations of motion of a multibody mechanical system. The nominal values of link mass, length, and inertia of MyBot arm used in SD/FAST are summarized in Table I. 'SH', 'EL', and 'WR' correspond to shoulder, elbow, and wrist, respectively, and the postfix represents the rotation axis with respect to the base frame. The values in the bracket represent the distance from the center of mass to joint.

Jacobian matrix, $\mathbf{J}$, which relates the joint velocities, $\dot{\mathbf{q}}$, to the end-effector velocities in Cartesian space, $\dot{\mathbf{p}}$, can be expressed as a 6×7 matrix as follows:

$$\dot{\mathbf{p}} = \begin{bmatrix} \upsilon_x \\ \upsilon_y \\ \upsilon_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \mathbf{J}(\mathbf{q}) \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_7 \end{bmatrix} \tag{1}$$

where υ and ω represent the linear and angular velocities of end-effector of MyBot arm, respectively. $\mathbf{q}$ denotes the joint position vector having a size of 7×1. Note that velocity propagation method is used to find $\mathbf{J}$, which calculates the velocities of each link in order starting from the base frame.

TABLE I
Nominal values of MyBot arm.

Joint	Name	Length [m]	Mass [kg]	Inertia [kg · m^2]
1	SHY	0.1 (0.05)	1.0	$\begin{bmatrix} 0.1 & 0.01 & 0.01 \\ 0.01 & 0.05 & 0.01 \\ 0.01 & 0.01 & 0.1 \end{bmatrix}$
2	SHX	0.0	0.8	$\begin{bmatrix} 0.03 & 0.005 & 0.005 \\ 0.005 & 0.08 & 0.005 \\ 0.005 & 0.005 & 0.08 \end{bmatrix}$
3	SHZ	0.2 (0.1)	1.0	$\begin{bmatrix} 0.15 & 0.01 & 0.01 \\ 0.01 & 0.15 & 0.01 \\ 0.01 & 0.01 & 0.05 \end{bmatrix}$
4	ELY	0.0	0.5	$\begin{bmatrix} 0.05 & 0.005 & 0.005 \\ 0.005 & 0.01 & 0.005 \\ 0.005 & 0.005 & 0.05 \end{bmatrix}$
5	WRZ	0.2 (0.1)	0.5	$\begin{bmatrix} 0.04 & 0.004 & 0.004 \\ 0.004 & 0.04 & 0.004 \\ 0.004 & 0.004 & 0.008 \end{bmatrix}$
6	WRY	0.0	0.3	$\begin{bmatrix} 0.03 & 0.003 & 0.003 \\ 0.003 & 0.006 & 0.003 \\ 0.003 & 0.003 & 0.03 \end{bmatrix}$
7	WRX	0.05 (0.05)	0.2	$\begin{bmatrix} 0.006 & 0.003 & 0.003 \\ 0.003 & 0.03 & 0.003 \\ 0.003 & 0.003 & 0.03 \end{bmatrix}$

III. Generating the Reference Trajectory

Generally, both initial and final postures of end-effector represented with respect to the base frame are given for a path generation as follows:

$$\begin{aligned} \mathbf{p}^0 &= \begin{bmatrix} x^0, & y^0, & z^0, & \psi^0, & \theta^0, & \phi^0 \end{bmatrix}^T \\ \mathbf{p}^N &= \begin{bmatrix} x^N, & y^N, & z^N, & \psi^N, & \theta^N, & \phi^N \end{bmatrix}^T \end{aligned} \tag{2}$$

where $[x, y, z]$ and $[\psi, \theta, \phi]$ represent the position and orientation information, respectively, and N represents the total number of samples. The convergence of DDP is not guaranteed unless it starts with a good reference trajectory because DDP is a locally optimal feedback control method. Thus, the reference trajectory is generated by using the minimum-jerk trajectory method, and then the corresponding torque profile is obtained by using the computed-torque method.

Prior to generating the reference trajectory, the joint position of initial and final postures have to be determined by using the inverse kinematics. Damped least-squares method, also called as Levenberg Marquardt method, is used for the inverse kinematics because it provides a numerically stable value of $\Delta\mathbf{q}$ [11], [12]. It updates the values of joint position vector as follows:

$${}^{j+1}\mathbf{q} = {}^{j}\mathbf{q} + \Delta^j\mathbf{q} \tag{3}$$

with

$$\Delta^j\mathbf{q} = \mathbf{J}^T \left(\mathbf{J}\mathbf{J}^T + \lambda^2\mathbf{I}\right)^{-1} \Delta^j\mathbf{p}$$

where j is the iteration number, λ is the damping factor, $\mathbf{I}$ is the 6×6 identity matrix, and $\mathbf{J}$ is obtained from (1). This process is repeated until the norm of posture errors in Cartesian space is minimized. The value of λ should be chosen carefully because there is a tradeoff between the accuracy and the feasibility depending on this value. As λ

gets smaller, (3) returns accurate solutions but low robustness to singular or near-singular postures. In contrast, this problem can be solved by setting λ large enough, but the convergence rate becomes too slow. Note that LDLT decomposition is used to speed up the inverse calculation of a matrix in (3).

Once the joint positions of initial and final postures are calculated, the minimum-jerk trajectory is applied to generate a smooth path from the initial position to final position in a desired time. Joint position errors increase as the jerk increases, where the jerk is the rate of change of acceleration. Thus, the minimum-jerk trajectory minimizes the sum of the squared jerk along its trajectory to limit vibration and to control humanoid arm trajectory similar to human arm trajectory [13]. The joint position, velocity, and acceleration of minimum-jerk trajectory are calculated as follows:

$$\begin{aligned}\mathbf{q}_d^i &= \mathbf{q}^0 + \left(\mathbf{q}^N - \mathbf{q}^0\right)(10k_3 - 15k_4 + 6k_5)\\ \dot{\mathbf{q}}_d^i &= \frac{\left(\mathbf{q}^N - \mathbf{q}^0\right)}{N}(30k_2 - 60k_3 + 30k_4)\\ \ddot{\mathbf{q}}_d^i &= \frac{\left(\mathbf{q}^N - \mathbf{q}^0\right)}{N^2}(60k_1 - 180k_2 + 120k_3)\end{aligned} \tag{4}$$

with

$$k_1 = \frac{i}{N},\ k_2 = k_1^2,\ k_3 = k_1^3,\ k_4 = k_1^4,\ k_5 = k_1^5$$

where $i = \{0, 1, \ldots, N\}$ represents the time index. Note that the minimum-jerk trajectory only considers the kinematics and independent of the dynamics of the nonlinear dynamical system.

Computed-torque method is used to obtain torque profiles of each joint to follow the minimum-jerk trajectory. It is robust to small changes in nonlinear dynamic equations and assures low steady-state errors or loss of stability. The equation of computed-torque method is given as follows:

$$\tau^* = \mathbf{M}(\mathbf{q})\left[\ddot{\mathbf{q}}_d + K_v\dot{\tilde{\mathbf{q}}} + K_p\tilde{\mathbf{q}}\right] + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \tag{5}$$

where τ^* is the torque vector of reference trajectory, $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is the centrifugal and Coriolis matrix, and $\mathbf{g}$ is the gravity vector. The values of $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{g}$ are obtained from SD/FAST software. K_v and K_p are the positive definite gains, and $\dot{\tilde{\mathbf{q}}} = \dot{\mathbf{q}}_d - \dot{\mathbf{q}}$ and $\tilde{\mathbf{q}} = \mathbf{q}_d - \mathbf{q}$ represent the joint velocity error and joint position error, respectively. The desired joint position $\mathbf{q}_d$, desired joint velocity $\dot{\mathbf{q}}_d$, and desired joint acceleration $\ddot{\mathbf{q}}_d$ are obtained from (4).

The last process in the reference trajectory generation is to update the values of joint position and velocity based on the torque profiles obtained in (5). The forward dynamics can be computed as follows:

$$\ddot{\mathbf{q}}^* = \mathbf{M}^{-1}(\mathbf{q})\left[\tau - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\,\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q})\right]. \tag{6}$$

Then the fixed-step Runge-Kutta fourth-order integrator built in SD/FAST is used to update the values of joint position and velocity. The resulting trajectory and torque profiles become the reference trajectory for MyBot arm motion, which is optimized by the DDP described in Section IV.

IV. Optimizing the Reference Trajectory

Consider the discrete-time nonlinear dynamical system, $\mathbf{x}^{i+1} = F(\mathbf{x}^i, \mathbf{u}^i)$, with states $\mathbf{x} = [q_1, \ldots, q_7, \dot{q}_1, \ldots, \dot{q}_7, \tau_1, \ldots, \tau_7]^T$ and controls $\mathbf{u} = [\dot{\tau}_1, \ldots, \dot{\tau}_7]^T$. In the proposed optimal controller, the rate of torque change is used as control input in order to generate the optimal trajectory that minimizes the torque variations. In this way, controlling the torque variation can reduce the amount of energy consumption and the wear on dynamical system by recognizing the abnormal signals.

The cost function is defined in the quadratic form of the end-effector position and orientation errors, the linear and angular velocity errors, and the rate of joint torque changes as follows:

$$\begin{aligned}L(\mathbf{x}, \mathbf{u}) &= \frac{1}{2}(\mathbf{p}^N - \mathbf{p}^{*^N})^T\mathbf{W}_1(\mathbf{p}^N - \mathbf{p}^{*^N})T_s\\ &+ \frac{1}{2}(\dot{\mathbf{p}}^N - \dot{\mathbf{p}}^{*^N})^T\mathbf{W}_2(\dot{\mathbf{p}}^N - \dot{\mathbf{p}}^{*^N})T_s\\ &+ \frac{1}{2}\sum_{i=0}^{N-1}[(\mathbf{p}^i - \mathbf{p}^{*^i})^T\mathbf{W}_1(\mathbf{p}^i - \mathbf{p}^{*^i})T_s\\ &+ (\dot{\mathbf{p}}^i - \dot{\mathbf{p}}^{*^i})^T\mathbf{W}_2(\dot{\mathbf{p}}^i - \dot{\mathbf{p}}^{*^i})T_s + (\mathbf{u}^i)^T\mathbf{W}_3(\mathbf{u}^i)T_s]\end{aligned} \tag{7}$$

where $\mathbf{p}^N$ and $\mathbf{p}^{*^N}$ represent the final posture and the reference posture of end-effector, respectively. T_s is the sampling time and $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_3$ are the cost-weighting matrices, where all these matrices are symmetric and positive definite. The control objective is to find the optimal trajectory of states by using DDP, which minimizes the cost function over a time step, N.

DDP is an iterative scheme, which calculates the sequence of optimal input by solving the value function as follows:

$$V(\mathbf{x}) = \min_u[L(\mathbf{x}, \mathbf{u}) + V(F(\mathbf{x}, \mathbf{u}))]. \tag{8}$$

DDP is a locally optimal feedback controller, which is based on the dynamic programming [1]-[5]. DDP uses the first (Jacobian) and second (Hessian) spatial derivatives of the value function. Thus, DDP has a second-order convergence. The first step of DDP is to initialize $\mathbf{x}^i$ and $\mathbf{u}^i$ of DDP as follows:

$$\begin{aligned}\mathbf{x}^i &= \left[q_1^{*^i}, \ldots, q_7^{*^i}, \dot{q}_1^{*^i}, \ldots, \dot{q}_7^{*^i}, \tau_1^{*^i}, \ldots, \tau_7^{*^i}\right]^T\\ \mathbf{u}^i &= \left[\frac{\left(\tau_1^{*^{i+1}} - \tau_1^{*^i}\right)}{T_s}, \ldots, \frac{\left(\tau_7^{*^{i+1}} - \tau_7^{*^i}\right)}{T_s}\right]^T\end{aligned} \tag{9}$$

where all initial values are obtained from the reference trajectory and $\mathbf{u}^N = [0, \ldots, 0]^T$. Every iteration in DDP is comprised of two sweeps of trajectory: a backward and a forward sweep.

In the backward sweep, unknown sequences of $\mathbf{V}_x$ and $\mathbf{V}_{xx}$ are calculated to generate the local model of value function as follows:

$$\begin{aligned}\mathbf{V}_x^i &= \mathbf{Z}_x^{i+1} - \mathbf{Z}_u^i\left(\mathbf{Z}_{uu}^i\right)^{-1}\mathbf{Z}_{ux}^i\\ \mathbf{V}_{xx}^i &= \mathbf{Z}_{xx}^{i+1} - \mathbf{Z}_{xu}^i\left(\mathbf{Z}_{uu}^i\right)^{-1}\mathbf{Z}_{ux}^i\end{aligned} \tag{10}$$

where

$$\begin{aligned}
\mathbf{Z}_x^i &= \mathbf{L}_x^i + \mathbf{V}_x^i \mathbf{F}_x^i \\
\mathbf{Z}_u^i &= \mathbf{L}_u^i + \mathbf{V}_x^i \mathbf{F}_u^i \\
\mathbf{Z}_{xx}^i &= \mathbf{L}_{xx}^i + (\mathbf{F}_x^i)^T \mathbf{V}_{xx}^i \mathbf{F}_x^i + \mathbf{V}_x^i \mathbf{F}_{xx}^i \\
\mathbf{Z}_{xu}^i &= \mathbf{L}_{xu}^i + (\mathbf{F}_x^i)^T \mathbf{V}_{xx}^i \mathbf{F}_u^i + \mathbf{V}_x^i \mathbf{F}_{xu}^i \\
\mathbf{Z}_{ux}^i &= \mathbf{L}_{ux}^i + (\mathbf{F}_u^i)^T \mathbf{V}_{xx}^i \mathbf{F}_x^i + \mathbf{V}_x^i \mathbf{F}_{ux}^i \\
\mathbf{Z}_{uu}^i &= \mathbf{L}_{uu}^i + (\mathbf{F}_u^i)^T \mathbf{V}_{xx}^i \mathbf{F}_u^i + \mathbf{V}_x^i \mathbf{F}_{uu}^i
\end{aligned} \tag{11}$$

with the boundary conditions,

$$\begin{aligned}
\mathbf{V}_x^N &= \mathbf{L}_x^N \\
\mathbf{V}_{xx}^N &= \mathbf{L}_{xx}^N.
\end{aligned} \tag{12}$$

The subscripts represent the derivatives, where $\mathbf{F}_x^i$ and $\mathbf{F}_u^i$ are evaluated Jacobians of dynamic system along $\mathbf{x}^i$ and $\mathbf{u}^i$ with respect to $\mathbf{x}$ and $\mathbf{u}$, respectively. Due to the dynamic system complexity, it is difficult to calculate the precise values of $\mathbf{F}_x^i$ and $\mathbf{F}_u^i$. Thus, these values are approximated as follows:

$$\begin{aligned}
\mathbf{F}_x^i &= \frac{F\left(\mathbf{x}^i + \alpha, \mathbf{u}^i\right) - F\left(\mathbf{x}^i - \alpha, \mathbf{u}^i\right)}{2\alpha} \\
\mathbf{F}_u^i &= \frac{F\left(\mathbf{x}^i, \mathbf{u}^i + \alpha\right) - F\left(\mathbf{x}^i, \mathbf{u}^i - \alpha\right)}{2\alpha}.
\end{aligned} \tag{13}$$

Note that Hessians $(\mathbf{F}_{xx}, \mathbf{F}_{xu}, \mathbf{F}_{ux}, \mathbf{F}_{uu})$ of dynamic system, and Jacobians $(\mathbf{L}_x, \mathbf{L}_u)$ and Hessians $(\mathbf{L}_{xx}, \mathbf{L}_{xu}, \mathbf{L}_{ux}, \mathbf{L}_{uu})$ of cost function are calculated in the similar manner.

In the forward sweep of DDP iteration, the optimal control improvement is calculated as follows:

$$\mathbf{u}_{new}^i = \mathbf{u}^i - \epsilon \left(\mathbf{Z}_{uu}^i\right)^{-1} \mathbf{Z}_u^i - \left(\mathbf{Z}_{uu}^i\right)^{-1} \mathbf{Z}_{ux}^i \left(\mathbf{x}_{new}^i - \mathbf{x}^i\right) \tag{14}$$

where $\epsilon \in (0, 1)$. A new nominal trajectory can be calculated by the forward integration of dynamic system as follows:

$$\ddot{\mathbf{q}}_{new}^{i+1} = \mathbf{M}^{-1}(\mathbf{q}_{new}^i) \left[\tau_{new}^i - \mathbf{C}\left(\mathbf{q}_{new}^i, \dot{\mathbf{q}}_{new}^i\right) \dot{\mathbf{q}} - \mathbf{g}\left(\mathbf{q}_{new}^i\right)\right] \tag{15}$$

where the values of $\mathbf{q}_{new}^i$, $\dot{\mathbf{q}}_{new}^i$, and τ_{new}^i are obtained from $\mathbf{x}_{new}^i$. Then, the nominal value of state, $\mathbf{x}_{new}^{i+1}$, is updated as follows:

$$\mathbf{x}_{new}^{i+1} = \begin{bmatrix}
\mathbf{q}_{new}^{i+1}[1] \\
\vdots \\
\mathbf{q}_{new}^{i+1}[7] \\
\dot{\mathbf{q}}_{new}^{i+1}[1] \\
\vdots \\
\dot{\mathbf{q}}_{new}^{i+1}[7] \\
\tau_{new}^i[1] + \mathbf{u}_{new}^i[1]\, T_s \\
\vdots \\
\tau_{new}^i[7] + \mathbf{u}_{new}^i[7]\, T_s
\end{bmatrix} \tag{16}$$

where the values of joint position and velocity can be found by integrating (15) in the Runge-Kutta fourth-order integrator built in SD/FAST. Norm value of state is calculated for every iteration, and the backward and forward sweeps are iterated until the norm value converges to a certain desired value, which is defined by the user.

Fig. 2. Simulation model of MyBot arm in Webots simulator.

V. Simulation Results

A. Simulation Environment

To demonstrate the effectiveness of the proposed optimal control method, the reference and optimized trajectory of MyBot arm were compared in the simulation model. Both trajectories were applied to MyBot arm modeled by Webots as shown in Fig. 2, which is a 3D robotics simulation software and enables users to conduct a physical and dynamic simulation [14]. The value of q_4 was always set as a negative value because 'ELY' joint cannot bend backward from its initial configuration.

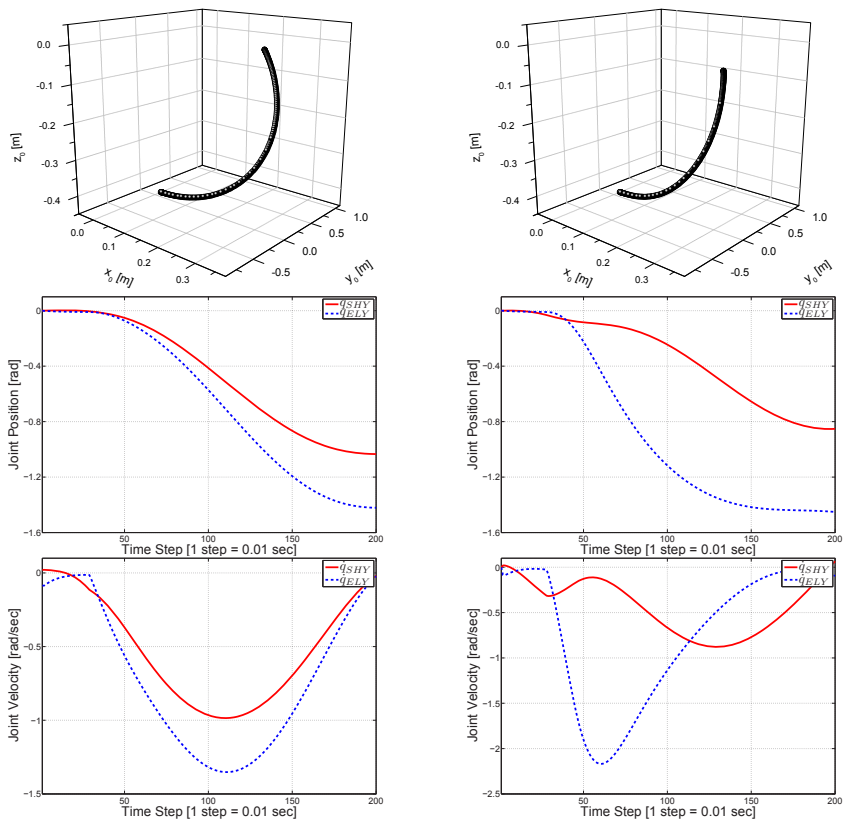
Parameter used in the reference trajectory generation and optimization in DDP are given in Table II. For the damped least-squares method, the number of maximum iteration was set as $10,000$, but the iteration terminated when the norm of posture error was less than 1.0×10^{-6}. The values of cost-weighting matrices in DDP were given as follows:

$$\begin{aligned}
\mathbf{W}_1 &= \text{diag}(10.0, 10.0, 10.0, 0.1, 0.1, 0.1) \\
\mathbf{W}_2 &= \text{diag}(1.0, 1.0, 1.0, 0.1, 0.1, 0.1) \\
\mathbf{W}_3 &= \text{diag}(0.001, 0.001, 0.001, 0.001, 0.001, 0.001, 0.001)
\end{aligned} \tag{17}$$

where 'diag' represents the diagonal matrix. The value of cost function in (7) was maximized when the posture error of end-effector of MyBot arm was minimized under a quadratic torque variation penalty.

TABLE II
Parameter Settings of Reference Trajectory Generation and Optimization.

Algorithm	Parameter	Value
Damped least-squares	Damping factor, λ	0.001
	Maximum number of iterations	10,000
Computed -torque	K_v	10
	K_p	100
DDP	Number of iterations	10
	Total number of samples, N	200
	Sampling time, T_s	0.01 sec
	α	0.00001
	ϵ	0.7



(a) Reference trajectory. (b) Optimized trajectory using DDP.

Fig. 3. Trajectory of end-effector position, joint position, and joint velocity of MyBot arm.

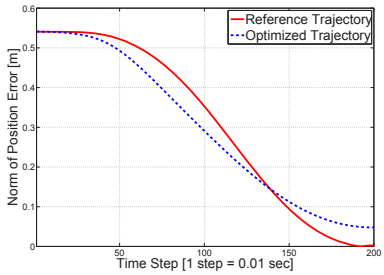


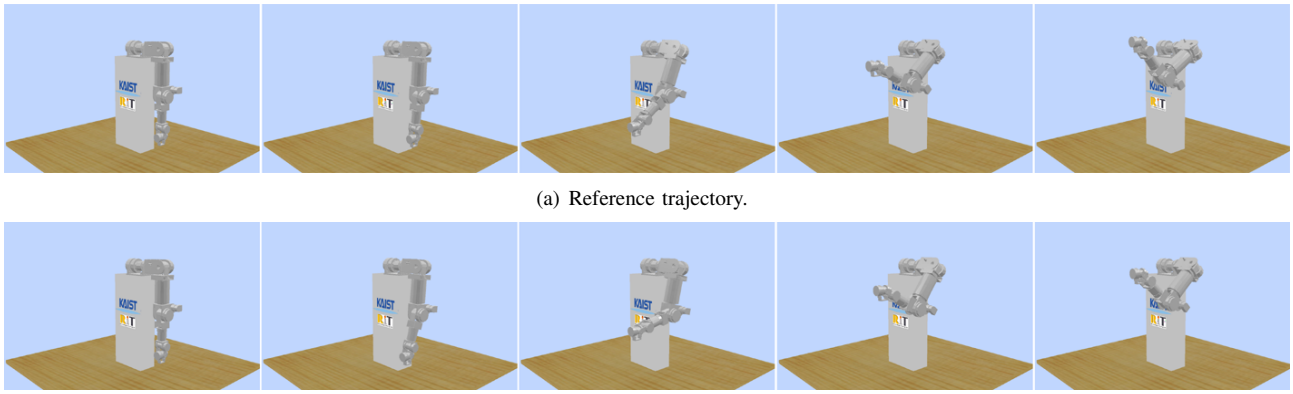
Fig. 4. Norm of position error of end-effector.

B. Results

Fig. 3 shows the reference and optimized trajectory of end-effector position, joint position, and joint velocity of MyBot arm when the initial posture $\mathbf{p}^0 = [0.0, 0.1, -0.45, 0.0, 0.0, 0.0]^T$ and the final posture $\mathbf{p}^N = [0.3, 0.1, 0.0, 0.0, 0.0, 0.0]^T$. As shown in Fig. 3, the end-effector trajectories represented in 3D spaces with respect to the base frame were a vertical upward movement features a curved arm path, but the reference and optimized paths were relatively different. The curvature of reference trajectory reached at a maximum and then decreased as the end-effector reached to the final posture. This was the typical characteristic of minimum-jerk trajectory because it did not considered the effectiveness of gravity, which was verified in the bell-shaped velocity profiles of shoulder and elbow joints. In contrast, the shape of optimized trajectory was a convex curve, which generated a shorter path compared to the reference trajectory.

Fig. 4 shows the norm of position error of end-effector for the reference and optimized trajectory. Since the minimum-jerk trajectory was only dependent of the kinematics of MyBot arm, the norm of position error in reference trajectory at the final time step was zero as shown as a solid line in Fig. 4. However, this value was approximately 0.05 for the optimized trajectory because the path minimizing the rate of torque change was determined by the dynamics of MyBot arm such as link length, mass, inertia, joint torques etc., and the gravitational force, which were constantly changed according to the region of workspace. The norm of position error in optimized trajectory decreased faster than the other because it was a shorter path compared to the reference trajectory as mentioned above.

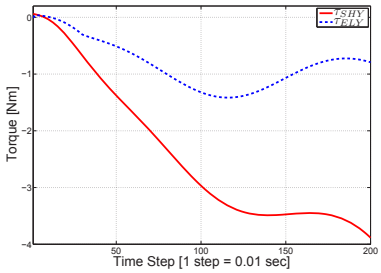
Fig. 5 shows the torque profiles of shoulder and elbow joints for the reference and optimized trajectory. The torque profiles of optimized trajectory minimized the rate of torque changes compared to the reference trajectory. The optimized trajectory decreased the total amount of torque change by 20 percent. The results of optimized trajectory significantly changed depending on the initialization of cost-weighting matrices in DDP. Thus, it is certain that the norm of position



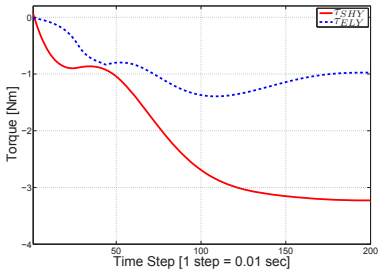
(a) Reference trajectory.

(b) Optimized trajectory using DDP.

Fig. 6. Snapshot of MyBot arm in Webots simulator captured at every 0.5 seconds.



(a) Reference trajectory.



(b) Optimized trajectory using DDP.

Fig. 5. Torque trajectory of MyBot arm.

error in optimized trajectory can be further minimized by optimizing the values of cost-weighting matrices in evolutionary algorithm. Lastly, Fig. 6 demonstrates the snapshots of MyBot arm for the reference and optimized trajectory in Webots simulator.

VI. Conclusions

This paper proposed a method to generate the optimal trajectory of nonlinear dynamical system, which concurrently minimized the rate of torque change in each joints and considered the dynamics of system. The structure of DDP was modified to optimize the reference trajectory, which was generated by using the minimum-jerk trajectory and the computed-torque method. Consequently, simulation results of 7 DOF MyBot arm confirmed that the proposed method could generate the optimal trajectory satisfying less torque and results in a shorter path in comparison to the reference trajectory. Verifying the proposed method in various types of path, such as one having multiple via-points, and in the actual robot are left as a further work.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012-0000321).

References

[1] D. Jacobson and D. Mayne, *Differential Dynamic Programming*. New York, NY: Elsevier, 1970.

[2] P. Dyer and S. McReynolds, *The Computation and Theory of Optimal Control*. New York: Academic Press, 1970.

[3] C. G. Atkeson, "Using local trajectory optimizers to speed up global optimization in dynamic programming," *Advances in Neural Inform. Process. Syst.*, pp. 663–670, 1994.

[4] C. G. Atkeson and J. Morimoto, "Nonparametric representation of policies and value functions: a trajectory-based approach," *Advances in Neural Inform. Process. Syst. 15*, 2003.

[5] C. G. Atkeson and B. J. Stephens, "Random sampling of states in dynamic programming," *IEEE Trans. Syst., Man, Cybern. B*, vol. 38, no. 4, pp. 924–929, Aug. 2008.

[6] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems," in *Proc. of Int. Conf. on Informatics in Control, Autom. Robot.*, Setúbal, Portugal, Aug. 2004, pp. 1–8.

[7] J. Morimoto and C. G. Atkeson, "Minimax differential dynamic programming: an application to robust biped walking," *Advances in Neural Inform. Process. Syst. 14*, 2002.

[8] P. Abbeel, A. Coates, M. Quigley and A. Ng, "An application of reinforcement learning to aerobatic helicopter flight," *Advances in Neural Inform. Process. Syst. 19*, 2007.

[9] Y. Tassa, T. Erez and W. Smart, "Receding horizon differential dynamic programming," *Advances in Neural Inform. Process. Syst. 19*, pp. 1465–1472, 2007.

[10] M. Hollars, D. Rosenthal and M. Sherman. SD/FAST. Symbolic Dynamics, Inc., 1991.

[11] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *ASME J. Dyn. Syst., Meas., Control*, vol. 108, no. 3, pp. 163–171, 1986.

[12] C. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Trans. Syst., Man Cyber.*, vol. 16, no. 1, pp. 93–101, 1986.

[13] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *J. Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.

[14] O. Michel, "Cyberbotics Ltd.WebotsTM: Professional mobile robot simulation," *Int. J. Adv. Robot. Syst.*, vol. 1, no. 1, pp. 39–42, 2004.