Advanced Robotics

ISSN: 0169-1864 (Print) 1568-5535 (Online) Journal homepage: https://www.tandfonline.com/loi/tadr20

Executing optimized throwing motion on robot arm with free joint

Chyon Hae Kim & Shigeki Sugano

To cite this article: Chyon Hae Kim & Shigeki Sugano (2016) Executing optimized throwing motion on robot arm with free joint, Advanced Robotics, 30:24, 1571-1578, DOI: 10.1080/01691864.2016.1255566

To link to this article: https://doi.org/10.1080/01691864.2016.1255566

ADVANCED ROBOTICS, 2016
VOL. 30, NO. 24, 1571–1578
http://dx.doi.org/10.1080/01691864.2016.1255566

FULL PAPER

Executing optimized throwing motion on robot arm with free joint

Chyon Hae Kim[a] and Shigeki Sugano[b]

[a]Faculty of Engineering, Department of Electrical Engineering and Computer Science, Iwate University, Morioka-shi, Japan; [b]Graduate School of Creative Science and Engineering, Waseda University, Tokyo, Japan

ABSTRACT
We address the throwing motion optimization for robot. In order to pursue the best throwing motion, we may need heuristics/intuition free methods. We propose a throwing method that is composed of rapid semi-optimal motion-planning and output zeroing method. So as to execute the optimized trajectories in real rigid body systems, we need some compensations for the noises around the optimized trajectories. We introduce a compensation method for the optimized throwing motions of a robot arm with a free joint. To validate the effectiveness of the proposed method, we conducted a throwing experiment using a two-link arm. As a result of the experiment, the robot arm threw a ball with 63.7 km/h, which was the best record through the past experiments of this arm.

ARTICLE HISTORY
Received 6 May 2016
Revised 22 August 2016
Accepted 20 October 2016

KEYWORDS
Throwing; dynamics; motion planning; optimization; passive joint

1. Introduction

In this paper, we address the throwing motion optimization for a robot. In recent years, throwing is getting attention both from the scientific study on human skills and the actual use (engineering) on robotic solutions. From the scientific interest, Feltner and Dapena analyzed fast throwing motions. In human's throwing, motion-dependent torque is mainly generated around shoulder joint and not around elbow joint.[1] Sakurai suggested the effective energy transferring from trunk to forearm in the acceleration of a ball using whip-like motion.[2] From the engineering, many applications are proposed in the field of sport,[3] casting,[4,5] traveling a robot in the air,[6] and so on. The optimal throwing motion is a common interesting topic in both fields, science and engineering.

On the fundamental research on ball throwing, Cheng et al. [7] proposed a mobile robot that throws a ball to a target using visual information. Nagayama et al. [8] developed 2 degrees of freedom (DOF) throwing manipulator using artificial muscles. These robots successfully throw balls. However, their throwing motions are hand-corded by the heuristics and intuition of operators.

Some researches indicate the improvement direction through systematic approach. Okada et al. [9] proposed a method to decrease the sensitivity for model uncertainty in the throwing dynamics. Cheng et al. [3] programmed a fuzzy logic that controls throwing power. Shoji et al. [10] optimized the trajectories of throwing motions using output zeroing method and manual optimization for the pathway of the end-effector. These robots successfully performed throwing according to their approaches. However, the required amount of the heuristics and intuition is not sufficiently small. In [9], the trajectory of the manipulator is predefined by an operator. In [3], the fuzzy logic requires much hand tuning. In [10], candidates of the end-effector's pathway must be pre-defined by an operator. Heuristics and inituition result not only in large developmental cost but also in less quality of throwing in many cases.

In order to pursuit the best throwing motion, we may need heuristics/intuition free methods. Yedeg et al. [11] proposed a gradient-based torque profile optimization algorithm. Miyazaki et al. [12] optimized not only the trajectory of a robot arm, but also the structure of the arm using a gradient method. Gui et al. [13] proposed an evolutionally computing method for the optimization of throwing motion. These methods allow a robot to optimize its motion without serious heuristics and intuition such as the above-mentioned works. But, gradient methods are easily trapped by the local optima when the objective function includes multiple local solutions. In [13], strict limitation in torque input pattern (a trigonometric function) is inevitable in order to assure the performance of the optimization.

Against these problems, we propose a throwing method that is composed of rapid semi-optimal motion-planning (RASMO) and output zeroing method under the assumption that the throwing robot is a rigid body system. The proposed method automatically generates

CONTACT Chyon Hae Kim tenkai@iwate-u.ac.jp

optimized trajectory without the tunings for the pathway, velocity, and control rules. Also, the proposed method is free from the risk of local optima and is applicable without strict limitation in torque input pattern. We applied the proposed method to a pendubot-typed robot arm, which has free joint at the elbow. As a result, a whip-like motion [2] that remarked the best throwing speed of this arm was generated.

The remainder of this paper is organized as follows: Section 2 describes the proposed method. In Section 3, we confirm its applicability and precision through throwing experiments using a pendubot-typed robot arm. Section 4 presents the conclusion.

2. Proposed method

As in [10], pendubot provides close mechanical model to humans' throwing dynamics. We proposed the method that is composed of RASMO and output zeroing method assuming the application on pendubot.

Let the trajectory optimization problem of the pendubot be defined as

$$u^*_{[0,t]} := \underset{u_{[0,t]}}{\operatorname{argmin}} J(\tau_{[0,t]}; x_I) \tag{1}$$

where $\tau_{[0,t]}$ is input vector set from time 0 to t, $J(u_{[0,t]}; x_I)$ is cost function of the input vector set. J requires the information of initial state x_I for its calculation.

The pendubot that is grabbing a ball in its hand is considered as a rigid body system. Thus, we need to solve the optimization problem under the constraint of the motion equation of rigid body system:

$$A(q)\ddot{q} + B(q, \dot{q}) = \tau \tag{2}$$

where $A(q)$ is inertia matrix, $B(q, \dot{q})$ is bias force vector that includes Coriolis force, centrifugal force, gravity force and so on, τ is joint torque vector of the robot arm.

For this optimization problem of throwing motion, path planning like [14] or velocity planning like [15] is not sufficient. We need to optimize state and torque profiles that satisfy some limitations. Hand cording of conditional branches [16–18] is not applicable to assure the resulting quality of the trajectory. Swing up motion planning [19] is also not applicable to throwing motion planning.

For this optimization problem, we developed RASMO.[20–22] RASMO is a tree-based motion optimization algorithm that searches nearly optimal motion using search tree. RASMO is not trapped by local optima because it does not require gradient method. RASMO has better efficiency than evolutionally computing algorithms or whole search algorithms like,[23] because it can reduce the search space by eliminating redundant branches before calculation is performed. Compared to reinforcement learning,[24,25] RASMO requires smaller number of trials, because RASMO is a model-based algorithm. We enhanced the capability of RASMO to online optimization [26,27] and improved the precision.[28] However, RASMO is not sufficient to execute throwing motion on a robot, because it does not take into account the disturbance around an actual robot.

In the execution of planned motion, state[1] feedback is used in order to keep a rigid body system on a planned trajectory, because only small amount of input compensation is required when actual state is close to planned one.

When position data were given as a time sequence $(q(0), q(1), \ldots, q(t))$, position feedback and state feedback are nearly equal, because when a rigid body system follows the position and time completely, its velocity sequence is also that of planned. Thus, in the proposed method, we use optimized position sequences generated by RASMO as a reference.

If noise does not affect a targeted rigid body system, the system follows the optimized position sequence with complete timing. However, the noise and chaotic effect enlarge the gap between the optimized sequence and that of the system. So, we need to compensate the gap with some additional torque input.

However, frequently RASMO does not allow the additional torque input, because it calculates nearly optimal trajectories that have no allowance of torque input. If we apply the trajectories to rigid body systems, we cannot add compensational torque input, because the system is using the maximum torque input already. So as to make compensational torque input, we need to restrict the maximum torque while optimizing the trajectories so as to make allowance at first.

In the proposed method, we applied an allowance of torque input while computing nearly optimal trajectories by RASMO. Also, we employed a position feedback control based on output zeroing method that cancel the noise around the optimized position sequence (reference).

We show the abstract flow of the proposed method in Figure 1. The proposed method identifies a nearly optimal trajectory using RASMO at the first stage. In this stage, the maximum torque inputs are not used in the optimization. Therefore, the torque sequence has an allowance that can be used for noise canceling. This allowance was decided manually. The controller that includes the noise cancelling is made by output zeroing method. This controller feed backs the positions of the rigid body system to those of optimized ones while considering the timing.

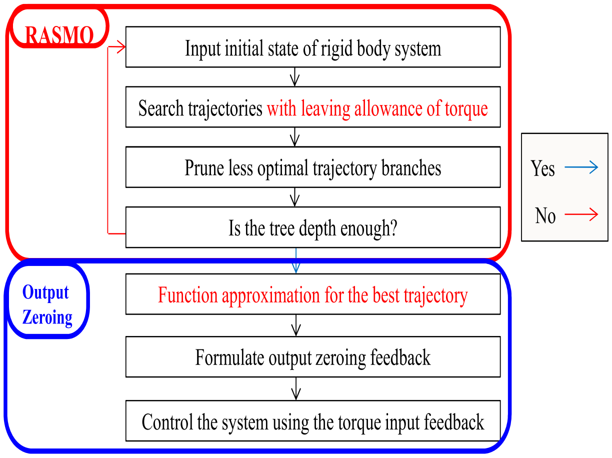

Figure 1. Flow of the proposed method. RASMO searches nearly optimal trajectory according to the initial state and boundary conditions of a rigid body system. Output zeroing method formulates feedback to the optimized trajectory. So as to apply output zeroing method, we need a twice differentiable trajectory. Therefore, we apply a function approximation to the optimized trajectory.

After selecting the best trajectory from the branches of the tree, the proposed method applies a small modification for the trajectory using a continuous differentiable function and output-zeroing method.

From generalized motion equation, the state equation is derived as follows:

$$\dot{x} = f(x) + g(x)\tau, \tag{3}$$

where

$$f(x) = \begin{bmatrix} \dot{q} \\ -A(q)^{-1}B(q,\dot{q}) \end{bmatrix}, \; g(x) = \begin{bmatrix} 0 \; A(q)^{-1} \end{bmatrix}. \tag{4}$$

When arbitrary variable y of a rigid body system follows a target function $B(z)$, which is the best trajectory of y generated by RASMO, the output function h is as follows:

$$h = y - B(z) \tag{5}$$

To assure that the variable y follows $B(z)$. We need to control the output function h to zero. The first order differential of the output function is as follows:

$$\begin{aligned} \frac{\mathrm{d}h}{\mathrm{d}t} &= \frac{\partial h(z)}{\partial z}\dot{z} \\ &= L_f h(z) + L_g h(z)\tau \end{aligned} \tag{6}$$

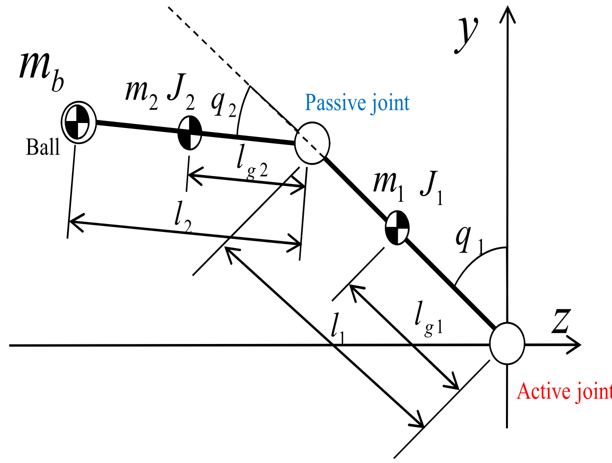

Figure 2. Mechanical model of the two-link arm. This arm was equipped with two links and two joints. One of the joint is a free (passive) joint. The end effector of the arm has an electromagnetic latch that allows this machine to release a ball. This model was used for the optimization process.

If we notice that torque does not affect the first derivative of motion equation (i.e. $\dot{q}$), $L_g h(z)$ must be zero. Thus, the second order derivative is as follows:

$$\begin{aligned} \frac{\mathrm{d}^2 h}{\mathrm{d}t^2} &= \frac{\partial L_f h(z)}{\partial z}\dot{z} \\ &= L_f^2 h(z) + L_g L_f h(z)\tau \end{aligned} \tag{7}$$

In these equations, $L_f h$, $L_g h$, $L_f^2 h$, $L_g L_f h$ are Lie derivatives. A new input u, which satisfies $\ddot{y} = v$ is obtained as follows:

$$u = L_g L_f h^{-1}(v - L_f^2 h) \tag{8}$$

We use this input to realize the feedback. Theoretically, function $B(z)$ must be differentiable at least twice. However, optimized pathway $B_0(z)$ generated by RASMO does not assure this requirement. Therefore, a polynomial equation and a regression method were used to approximate the pathway.

3. Experiment and discussion

We examined the proposed method using a two-link arm with a free joint that was developed at the Sampei Laboratory at the Tokyo Institute of Technology. [10,29,30]

3.1. Settings

The mechanical model is shown in Figure 2 and the parameters are shown in Table 1.

Table 1. Physical parameters.

Object	Parameter	Symbol	Value	
Ball	Mass	m_b	0.057	[kg]
First link	Mass	m_1	1.940	[kg]
	Length	l_1	0.250	[m]
	CoM from the root joint	l_{g1}	0.0659	[m]
	Inertia moment around CoM	J_{g1}	0.0726	[kgm^2]
First joint	Viscous friction coefficient	d_1	1.10	[Nms/rad]
Second link	Mass	m_2	1.395	[kg]
	Length	l_2	0.300	[m]
	CoM from the root joint	l_{g2}	0.00936	[m]
	Inertia moment around CoM	J_{g2}	0.0876	[kgm^2]
Second joint	Viscous friction coefficient	d_2	0.000650	[Nms/rad]

The motion equation is as follows:

$$\begin{bmatrix} m_{11} & * \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix} \tag{9}$$

$$m_{11} = a_1 + a_3 + 2a_2 \cos\theta_2 \tag{10}$$

$$m_{21} = a_3 + a_2 \cos\theta_2 \tag{11}$$

$$m_{22} = a_3 \tag{12}$$

$$c_1 = (a_4 \cos\theta_1 + a_5 \cos(\theta_1 + \theta_2))g - c_2\dot{\theta}_2(2\dot{\theta}_1 + \dot{\theta}_2)\sin\theta_2 + d_1\dot{\theta}_1 \tag{13}$$

$$c_2 = c_5 \cos(\theta_1 + \theta_2)g + c_2\dot{\theta}_1^2 \sin\theta_2 + d_2\dot{\theta}_2 \tag{14}$$

$$a_1 = J_{g1} + l_{g1}^2 m_1 + l_1^2(m_2 + m_b) \tag{15}$$

$$a_2 = l_1(l_{g2}m_2 + l_2 m_b) \tag{16}$$

$$a_3 = J_{g2} + l_{g2}^2 m_2 + l_2^2 m_b \tag{17}$$

$$a_4 = l_{g1}m_1 + l_1(m_2 + m_b) \tag{18}$$

$$a_5 = l_{g2}m_2 + l_2 m_b \tag{19}$$

where

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} q_1 - \frac{\pi}{2} \\ q_2 \end{bmatrix} \tag{20}$$

In order to calculate motion equations, we may be able to use the algorithms like,[31] which has lower calculation order. But, in case of low DOF, we cannot take its benefit. Therefore, we directly calculated the above equations in this research.

Using this two-link arm, nearly optimal throwing motion generated by RASMO was executed. As in the previous research,[29,30] the trajectory was divided into four phases.

(1) Constraint phase (CP)
(2) Non-constraint phase (NCP)
(3) Releasing phase (RP)
(4) Follow through phase (FTP)

Table 2. Partition parameters.

Parameter			Value	
Torque	τ_1	Max	79.0	[Nm]
		Min	−79.0	[Nm]
		Resolution	1.0	[Nm]
First joint position	q_1	Max	$\pi/2$	[rad]
		Min	$-\pi/2$	[rad]
		Resolution	$\pi/50$	[rad]
First joint velocity	$\dot{q}_1$	Max	8π	[rad/s]
		Min	0	[rad/s]
		Resolution	$4\pi/25$	[rad/s]
Second joint position	q_2	Max	$\pi/4$	[rad]
		Min	$-\pi/2$	[rad]
		Resolution	$3\pi/200$	[rad]
Second joint velocity	$\dot{q}_2$	Max	20π	[rad/s]
		Min	0	[rad/s]
		Resolution	$\pi/5$	[rad/s]

A position limitation constrains the second joint at $q_2 = -\pi/2$ in CP. NCP begins from the moment when the second link works off from the limitation. The dynamics of the machine automatically works off the second link. RP is a short interval while the machine releases a ball. During FTP, the machine safely slows down both links.

We applied the proposed method during NCP. For the cost function of RASMO, we used time. When we use time, RASMO calculates the fastest trajectory that reaches each state. As a result, the planned trajectory becomes quick motion. To find the best trajectory from the tree of RASMO, we prepared two evaluation functions EF1 $R = -\dot{E}_z$ (for Experiment 1) and EF2 $R = -\dot{E}_z - 10|\dot{E}_y|$ (for Experiment 2), where $\dot{E}_z$ and $\dot{E}_y$ are the horizontal and vertical velocities of the end effector. Each evaluation function gives each best trajectory. RASMO finds a trajectory that includes a state which maximizes R. For the RASMO partition, we used the partition parameters in Table 2. These resolutions were much smaller than those of the experiments in [22], in which nearly optimal trajectories were obtained. For the search, the minimum time step was 0.2 ms. Torque was switched at a time resolution 4 ms. The initial state of NCP was fixed to the parameters $(q_1, \dot{q}_1, q_2, \dot{q}_2)$ = (−0.3283, 18.6139, −1.4931, 3.4907) obtained from a preliminary experiment. The calculation of RASMO was finished within 1 h.

The output-zeroing method was applied to position y of the end effector trajectory. To approximate the planned trajectory, a nine-order polynomial function was used.

3.2. Results

3.2.1. Experiment 1

In Experiment 1, we used EF1. Figure 3 shows the trajectory which was optimized by RASMO using EF1 and the actual trajectory of the machine. The pathway of the end effector was controlled at precisely the same

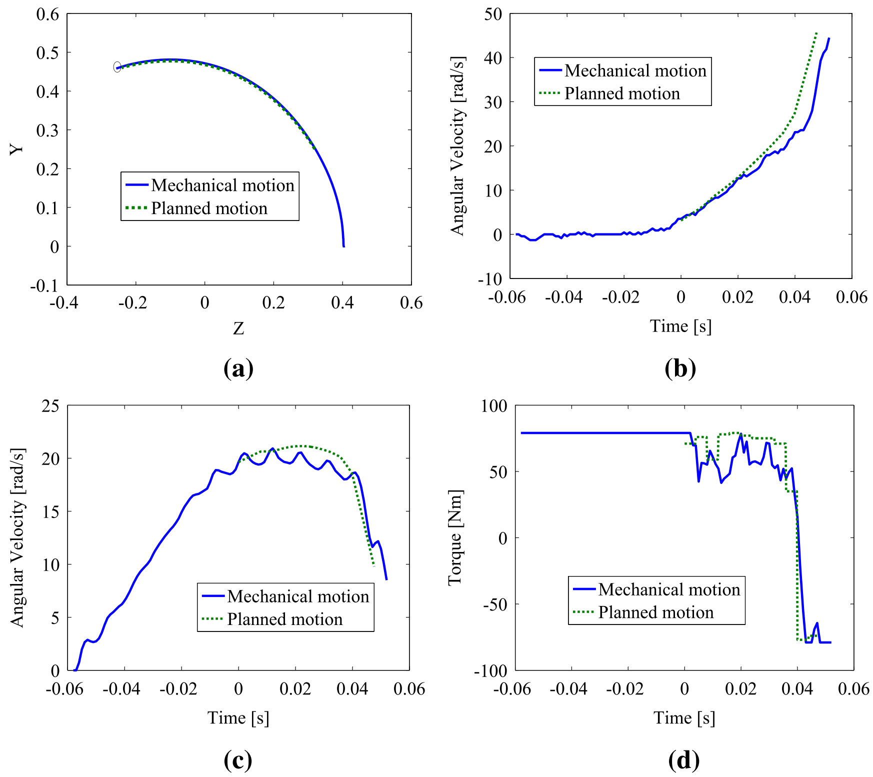

Figure 3. Result of EF1. (a) Pathway of end effector (throwing direction is $-z$, the releasing point is depicted with a circle); (b) Angular velocity of second joint; (c) Angular velocity of first joint; (d) Torque input.

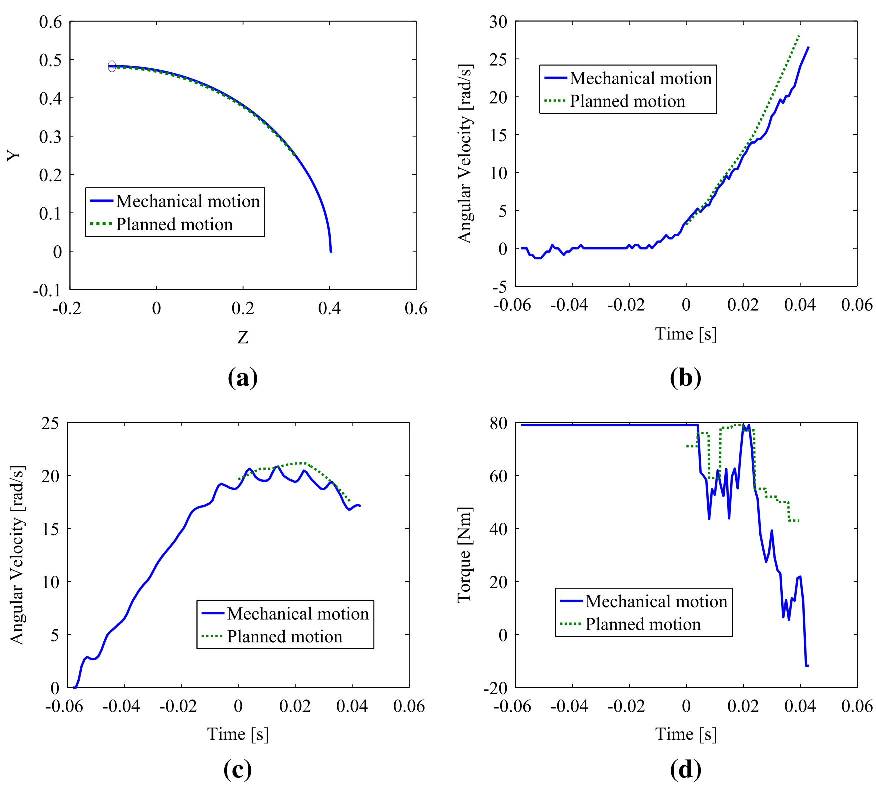

Figure 4. Result of EF2. (a) Pathway of end effector (throwing direction is $-z$, the releasing point is depicted with a circle); (c) Angular velocity of second joint; (c) Angular velocity of first joint; (d) Torque input.

pathway as planned. Angular velocity of the second joint remained nearly the same as planned. However, the angular velocity of the first joint vibrated around the planned value. Consequently, the torque sequence also fluctuated. The throwing speed at RP was the best record for this machine, at 63.7 km/h. This result is also almost equal to the planned throwing speed of 67.0 km/h calculated by RASMO.

As a result, the machine precisely realized the nearly optimal motion of the end effector. However, the throwing direction, shown at the left side most end of the graph in Figure 3(a), was not horizontal.

3.2.2. Experiment 2

In Experiment 2, EF2 was used. Figure 4 shows the motion which was planned by RASMO using EF2 and the actual motion of the machine. These graphs show the same tendency as Experiment 1 except the throwing direction that is horizontal. The throwing speed at RP was 58.0 km/h. This result is also almost close to the planned throwing speed of 59.7 km/h.

3.3. Discussion

3.3.1. Precision

The precise end effector pathway (Figure 3) was a result of the precise motion planning of RASMO and the control capability of the output-zeroing method. The second joint angular velocity exhibited some delay against the semi-optimal velocity. This is attributed to the fact that realization for a nearly optimal trajectory requires so strict conditions for a machine. From the view point of feedback control, output-zeroing method compromised the control by applying delay. As a result, whole trajectory of the links avoided critical error. Also, output-zeroing method compensated the roughness of the velocity of the first joint (Figures 3(c) and 4(c)). As a result, the velocity of the second joint (Figures 3(b) and 4(b)) was smooth. This fact assured the precision of the throwing.

3.3.2. Throwing technique

The planned and executed trajectories showed a throwing technique in its torque profile (Figures 3(d)). Around 0.04 s, the torque was decreased from around 50 Nm to −79 Nm (minimum). This torque shift caused braking in the first joint (Figure 3(c)). However, this first joint braking resulted in acceleration of the second joint (Figure 3(b)). In this instance, some motion energy was transferred from the first joint to the second joint. As a result, the second joint achieved a faster throwing speed. RASMO calculates this type of technique automatically, without advance human knowledge about the mechanical motions. Output-zeroing method realized this technique with high precision. This technique support the hypothesis that whip-like motion is effective in fast throwing.[2]

3.3.3. Horizontal throwing

In Experiment 1, we used the evaluation function $R = -\dot{E}_z$ that resulted in a faster throwing speed in the direction $-z$. However, throwing direction was not taken into account in this calculation. As a result, the realized throwing direction was not horizontal (Figure 3(a)). In Experiment 2, we used $R = -\dot{E}_z - 10|\dot{E}_y|$. This function restricted velocity y of the end effector at the moment of a throwing action. As a result, a horizontal throwing direction was achieved as shown in Figure 4(a).

3.3.4. Optimality of mechanical parameter

RASMO does not plan brake motion in Experiment 2, though that was effective in Experiment 1. This may mean that there is a trade-off between throwing speed and throwing direction (horizontal throwing) inside this machine, and unusability of the throwing technique was better for the setting of Experiment 2. If we select mechanical parameters using calculation results of RASMO before developing machines, the machines may realize faster horizontal throwing using the throwing technique.

3.3.5. The roles of two indexes

In the proposed method, we introduced two indexes J and R. Index R is used in order to decide the terminal state (release point) of the robot arm, which is described like x_G in the last version. Index J is used in order to obtain fast motion that reaches to the terminal state. Thus, these two indexes are assigned to different parameters. In the proposed method, RASMO calculates optimal trajectories that reach different terminal states in order to obtain fast motions using J. The resolution of the terminal states is decided by the resolution of the mesh. After many trajectories were obtained, the best one is selected according to R. In the proposed method, R is more important than J, because the trajectory of a ball is decided by J. J decides only the pathway and velocity profile to reach the best terminal state that is selected by R. But in case that J is not an appropriate function, the robot arm may take too much time before reaching the best terminal state.

4. Conclusion

In order to pursue the best throwing motion, we proposed a method that couples RASMO with an output-zeroing method by considering torque allowance. We conducted two throwing experiments using a two-link arm. As a result of the experiment, the planned nearly optimal trajectory was realized. Using an evaluation function with a restriction term for vertical velocity, horizontal throwing was also achieved. RASMO was effective in realizing high performance of the mechanical System . In future experiments, it will be necessary to validate RASMO in a wider variety of systems.

Note

1. State x is defined by the pair of position and velocity $x = (q, \dot{q})$.

Acknowledgements

We would like to thank Mr. Hikaru Suzuki, Dr. Kazuma Sekiguchi and Dr. Mitsuji Sampei, Tokyo Institute of Technology, Japan for their assistance.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

A part of this work was supported by JSPS KAKENHI [grant number 15K20850].

Notes on contributors

Chyon Hae Kim is Associate Professor of Computer, Department of Systems Innovation Engineering, Faculty of Science and Engineering in Iwate University (Japan). He is Invited Researcher (2008–) of Waseda University. He obtained Professor (Engineering) at Waseda University (2008). He was Assistant Researcher of 21st Center of Excellence Project at Waseda University (2005–2008). He was Researcher at Honda Research Institute Japan Co., Ltd. (2008–2013). He was Invited Researcher of RIKEN (Japan) (2008–2013). He was Adjunct Lecturer of Waseda University (2013–2014). He received Student Incentive Award from Information Processing Society of Japan (2007,3), Best Paper Award Nomination at the international conference IEA/AIE (2013,6), Research Incentive Award from Robot Society of Japan(RSJ) on (2013,9), and SICE System Integration branch Best Presentation Award (2014,12). He is Review Board Member of Applied Intelligence (Springer). He is Associate Editor of Advanced Robotics, Aloy Journal of Soft Computing and Applications, and Journal of IPSJ. He is Expert Board Member of Research Expert Committee (Open Intelligence) of RSJ. He is Expert Board Member of SICE Tohoku Branch.

Shigeki Sugano received his BS, MS, and Dr. of Engineering degrees in Mechanical Engineering in 1981, 1983, and 1989 from Waseda University. From 1987 to 1991, he was a research associate at Waseda University. Since 1991, he has been a faculty member in the Department of Mechanical Engineering at Waseda, where he is currently a professor. From 1993 to 1994, he was a visiting scholar in the Mechanical Engineering Department at Stanford University. From 2001 to 2012, he served as the director of the Waseda WABOTHOUSE laboratory. Since 2012, he has been the director of the Institute for Techno-Innovation in Chubu-Area Industries. Since 2000, he has been a member of the Humanoid Robotics Institute of Waseda University. From 2011 to 2014, he served as the associate dean of the School of Creative Science and Engineering, Waseda University. Since 2014, he has served as the dean of the School of Creative Science and Engineering, Waseda University. Since 2013, he has served as the program coordinator of the MEXT Leading Graduate Program: Waseda Embodiment Informatics Program.

His research interests include human-symbiotic anthropomorphic robot design, dexterous and safe manipulator design, and human–robot communication. He received the Technical Innovation Award from the Robotics Society Japan for the development of the Waseda Piano-Playing Robot: WABOT-2 in 1991. He received the JSME Medal for Outstanding Paper from the Japan Society of Mechanical Engineers in 2000, the JSME Fellow Award in 2006, and the IEEE Fellow Award in 2007. He also received IEEE RAS Distinguished Service Award in 2008, the RSJ Fellow Award in 2008, and the SICE Fellow Award in 2011.He received RSJ Distinguished Service Award in 2012.

He served as the secretary of the IEEE Robotics & Automation Society (RAS) in 2006 and 2007. He served as a co-Chair of the IEEE RAS Technical Committee on Humanoid Robotics from 2005 to 2008. He served as the IEEE RAS Conference Board, Meetings Chair from 1997 to 2005. He served as an AdCom member of the IEEE RAS and the associate vice-president of the IEEE RAS Conference Board from 2008 to 2013.

From 2007 to 2012, he served as the editor in chief of the International Journal of Advanced Robotics. He served as the head of the System Integration Division of the Society of Instrument and Control Engineers (SICE) in 2006 and 2007. He serves as a director of SICE in 2008 and 2009. From 2001 to 2010, he served as the president of the Japan Association for Automation Advancement.

He served as the general chair of the 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2003). He was a general co-Chair of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2006) and a program co-Chair of the 2009 IEEE International Conference on Robotics and Automation (ICRA2009). He served as the general chair of the SICE2011 in 2011. He also served as the general co-Chair of the 2012 IEEE International Conference on Robotics and Automation (ICRA2012), the program chair of the 2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2012), and the general chair of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2013) in Tokyo.

References

[1] Feltner M, Dapena J. Dynamics of the shoulder and elbow joints of the throwing arm during a baseball pitch. Int J Sports Biomech. 1986;2:235–259.

[2] Sakurai S. Biomechanics of overhand throwing motion: past, present, and future research trend. Korean J Sport Biomech. 2004;14:183–187.

[3] Cheng CT, Wong CC, Hu YY, et al. Fuzzy Basket ball throwing strength control system for vision-based humanoid robot. Intell Rob Syst: Inspiring NEXT Ser Commun Comput Inf Sci. 2013;376:275–285.
[4] Tsukagoshi H, Watari E, Fuchigami K, et al. Casting device for search and rescue aiming higher and faster access in disaster site. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. Portugal; 2012. p. 4348–4353.
[5] Fagiolini A, Arisumi H, Bicchi A. Casting robotic end-effectors to reach faraway moving objects. Comput Res Repository; 2011. abs/1101.2268.
[6] Jung W, Kim Yk, Kim S. Design of a miniature sphere type throwing robot with an axial direction shock absorption mechanism. J Inst Control Rob Syst. 2015;21:361–366. Korean.
[7] Cheng CY, Hsieh CYL, Kuo GH. A vision-based robot arm with ball-throwing capability. In: International Automatic Control Conference. USA; 2013. p. 74–78.
[8] Nagayama T, Ishihara H, Tomori H, et al. Throwing operations by manipulator with a 2-DOF variable viscoelastic joint using pneumatic artificial muscles and a magnetorheological brake. In: IEEE International Conference on Robots Biomimetics. China; 2015. p. 2324–2329.
[9] Okada M, Pekarovskiy A, Buss M. Robust trajectory design for object throwing based on sensitivity for model uncertainties. In: IEEE International Conference on Robotics and Automation. USA; 2015. p. 3089–3094.
[10] Shoji T, Katsumata S, Nakamura S, et al. Throwing motion control of the springed pendubot. IEEE Trans Control Syst Tech. 2013;21:950–957.
[11] Yedeg EL, Wadbro E. State constrained optimal control of a ball pitching robot. Mech Mach Theory. 2013;69:337–349.
[12] Miyazaki T, Kanekiyo A, Tsuchiyama Y, et al. Integrated robot design considering feasible motion conditions and dynamical relations between body parameters. Integration: IEEE/SICE International Symposium on System. Japan; 2014. p. 459–464.
[13] Gai Y, Kobayashi Y, Hoshino Y, et al. Motion control of a ball throwing robot with a flexible robotic arm. Int J Comput Electr Autom Contorl Inf Eng. 2013;7:937–945.
[14] Eele A, Richards A. Rapid updating for path-planning using nonlinear branch-and-bound. In: IEEE International Conference on Robotics and Automation. USA; 2010. p. 3575–3580.
[15] Pfeiffer F, Johanni R. A concept for manipulator trajectory planning. IEEE International Conference on Robotics and Automation. 1986;3:1399–1405.
[16] Block DJ. Mechanical design and control of the pendubot. [thesis] University of Illinois. 1991.
[17] Fantoni I, Lozano R, Spong MW. Energy based control of the pendubot. IEEE Trans Autom Control. 2000;45:725–729.
[18] Murray RM, Hauser J. Nonlinear controllers for non-integrable systems: the acrobot example. Automatic Control Conference. 1990;1:669–671.
[19] Spong MW. The swing up control problem for the acrobot. IEEE Control Syst Mag. 1995;15:45–55.
[20] Kim CH, Tsujino H, Sugano S. Rapid short-time path planning for phase space. J Rob Mechatron. 2011;23:271–280.
[21] Kim CH, Sugano S. A GPU parallel computing method for LPUSS. Adv Rob. 2013;27:1199–1207.
[22] Kim CH, Sugano S. Tree based trajectory optimization based on local linearity of continuous non-linear dynamics. IEEE Trans Autom Control. 2016;61:2610–2617.
[23] Boone G. Minimum time control of the acrobot. In: IEEE International Conference on Robotics and Automation. USA; 1997. p. 3281–3287.
[24] Nishimura M, Yoshimoto J, Ishii S. Acrobot control by learning the switching of multiple controllers. J Artif Life Rob. 2005;9:67–71.
[25] Arie H, Ogata T, Sugano S. Reinforcement learning of a continuous motor sequence with hidden states. Adv Rob. 2007;21:1215–1229.
[26] Kim CH, Tsujino H, Sugano S. Online motion selection for semi-optimal stabilization using reverse-time tree. In: IEEE International Conference on Intelligent Robots and Systems. USA; 2011. p. 3792–3799.
[27] Kim CH, Sugano S. Closed loop trajectory optimization based on reverse time tree. Int J Control Autom Syst. Forthcoming.
[28] Kim CH, Yamazaki S, Sugano S. Motion optimization using state-dispersionbased phase space partitions. Multibody Syst Dyn. 2013;32:159–173.
[29] Katsumata S, Ichinose S, Shoji T, et al. Throwing motion control based on output zeroing utilizing 2-Link Underactuated Arm. In: Proceedings of the American Control Conference. USA; 2009. p. 3057–3064.
[30] Ichinose S, Katsumata S, Nakaura S, et al. Throwing motion control experiment utilizing 2-link arm with passive joint. In: Proceedings of the SICE Annual Conference. Japan; 2008. p. 3256–3261.
[31] Yamane K, Nakamura Y. Efficient parallel dynamics computation of human figures. In: International Conference on Robotics and Automation. USA; 2002. p. 350–357.