2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

Robust Trajectory Design for Object Throwing based on Sensitivity for Model Uncertainties

Masafumi Okada[*1*2], Alexander Pekarovskiy[*2] and Martin Buss[*2]

***Abstract*— Throwing an object by a powered robot system is of great importance in unmanned environments. In this paper, we consider the problem of throwing a point-mass object to minimize uncertainty in the object's landing position, given uncertainty in (1) the robot's initial configuration and (2) friction at the joints. Our analysis assumes that the robot's throw is executed using open-loop torque commands, and it relies on the linearized sensitivities of (a) landing location with respect to release state, (b) release state with respect to initial robot configuration and (c) joint friction. Moreover, the effectiveness of the proposed method is evaluated by Monte-Carlo simulations.**

I. Introduction

Throwing an object by a robot system is of great importance in unmanned environments. For example, as shown in figure 1-(a), a robot system will leave a sensor system to gather information in a disaster-stricken area by throwing. In a mountainous region, as shown in figure 1-(b), throwing by a powered manipulator will be utilized for wiring. Tsukakoshi et al.[1] developed a casting device for search and rescue operation in disaster site. They emphasized the effectiveness of throwing an object to gather information in unmanned area. Fagiolini et al.[2] developed a casting robot that throws its end-effector to catch objects which is far from the robot.

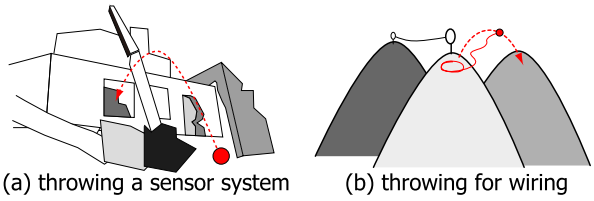

Fig. 1. Throwing an object by a powered robot system

So far, many researches have been reported on the throwing. Mason and Lynch defined dynamic manipulation[3] as a manipulation with not only kinematics and statics but also dynamic forces, and developed throwing motion controlling underactuated objects[4]. Frank et al.[5] suggested to use throwing for industrial applications to speed up transportation of parts. The throwing motion was designed by dynamical considerations, and the end-effector is controlled based on vision system. Tabata et al.[6] presented a passing manipulation. The trajectory of the 1-DOF manipulator was derived by Newton's method with top height of the parabolic orbit.

In the most of the conventional approaches, to obtain high accuracy of the landing point, (i) trajectory design of the throwing based on dynamical consideration, and (ii) robust controller design that realizes high accuracy for tracking of the manipulator along the designed trajectory, have been discussed. However, a robust controller sometimes does not work well because of too large/small estimation of the uncertainties or disturbances.

On the other hand, variability of the landing point caused by uncertainties will be changed by the trajectory of throwing. Figure 2 shows an example of throwing with a perturbation of releasing position. Supposing that A is a nominal

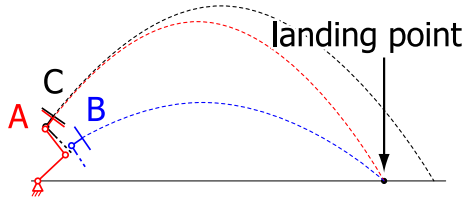

Fig. 2. Free flying locus that passes through a landing point

releasing position and velocity, the larger perturbation of B is allowable because the object passes through the landing point. However C is not allowed even though a small perturbation. Based on this concept, we have proposed control goal manifolds of the manipulator based on a sensitivity of the landing point [7]. However, it has to be discussed which kind of uncertainties yield these perturbations from dynamical point of view. An optimal trajectory that has smallest sensitivity for uncertainties will be a robust trajectory.

In this paper, we introduce a sensitivity analysis of the landing point with respect to model uncertainties (uncertainty of inital position and joint Coulomb friction), and design the optimal throwing trajectory that minimizes the sensitivity. Our analysis assumes that the robot's throw is executed using open-loop torque commands, and it relies on the linearized sensitivities of (a) landing location with respect to release state, (b) release state with respect to initial robot configuration and (c) joint friction. Moreover, the effectiveness of the

M. Okada is with [*1]Dept. of Mechanical Sciences and Engineering, Tokyo Institute of Technology, 2-12-1 Ookayama Meguro-ku, Tokyo, Japan, 152-8552. He was a visiting researcher in the [*2]Inst. of Automatic Control Engineering, Technische Universitat Munchen. okada@mep.titech.ac.jp

A. Pekarovskiy and M. Buss are with the [*2]Inst. of Automatic Control Engineering, Technische Universitat Munchen, Munich, Germany, 80290 {a.pekarovskiy,mb}@tum.de

proposed method is evaluated by Monte-Carlo simulations.

II. Problem Formulation

Consider a planar 3 degrees-of-freedom (3-DOF) manipulator as shown in figure 3. The lengths of each link are ℓ_1,

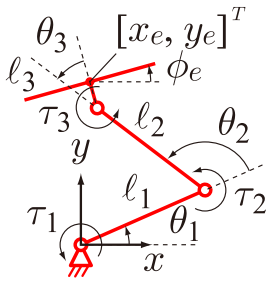

Fig. 3. Model of planar 3-DOF manipulator

ℓ_2 and ℓ_3. The joint angles are defined by θ_1, θ_2 and θ_3, and

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T \tag{1}$$

is defined. The torques of each joint are defined by τ_1, τ_2 and τ_3, and

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 \end{bmatrix}^T \tag{2}$$

is also defined. The position and orientation of the end-effector in absolute coordinates are defined by

$$\boldsymbol{x}_e = \begin{bmatrix} x_e & y_e \end{bmatrix}^T \tag{3}$$

and ϕ_e.

The robot throws an object as shown in figure 4. Position,

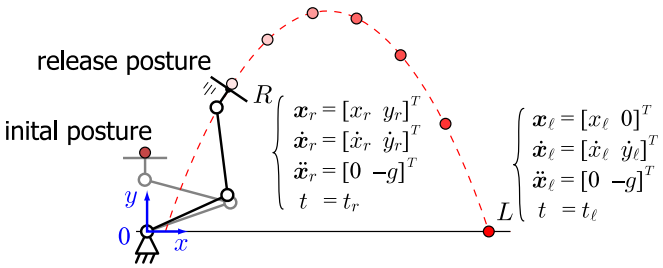

Fig. 4. Planar 3-DOF manipulator that throws an object

velocity and acceleration of the object are defined by

$$\boldsymbol{x}_r = \begin{bmatrix} x_r, & y_r \end{bmatrix}^T, \quad \boldsymbol{x}_\ell = \begin{bmatrix} x_\ell, & 0 \end{bmatrix}^T \tag{4}$$

$$\dot{\boldsymbol{x}}_r = \begin{bmatrix} \dot{x}_r, & \dot{y}_r \end{bmatrix}^T, \quad \dot{\boldsymbol{x}}_\ell = \begin{bmatrix} \dot{x}_\ell, & \dot{y}_t \end{bmatrix}^T \tag{5}$$

$$\ddot{\boldsymbol{x}}_r = \begin{bmatrix} 0, & -g \end{bmatrix}^T, \quad \ddot{\boldsymbol{x}}_\ell = \begin{bmatrix} 0, & -g \end{bmatrix}^T \tag{6}$$

in point of release R and landing point L at Cartesian coordinates $\boldsymbol{x}_r$ and $\boldsymbol{x}_\ell$, where g represents gravity. A time of release and landing time are set at t_r and t_ℓ respectively. By giving $\boldsymbol{x}_\ell$ and $\dot{\boldsymbol{x}}_\ell$, a flying locus of the object is uniquely defined by a parabolic line as

$$y = ax^2 + bx + c \tag{7}$$

neglecting air resistance, where the coefficients a, b and c satisfy

$$\boldsymbol{a} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} x_\ell^2 & x_\ell & 1 \\ 2x_\ell\dot{x}_\ell & \dot{x}_\ell & 0 \\ 2\dot{x}_\ell^2 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \dot{y}_\ell \\ -g \end{bmatrix} \tag{8}$$

Because the purpose of the throwing is to make an object land at $\boldsymbol{x}_\ell$, x_r is arbitrary selected by a designer and $\dot{x}_r$, t_r, y_r and $\dot{y}_r$ have the following constraints.

$$\dot{x}_r = \dot{x}_\ell, \qquad t_r = t_\ell - \frac{x_\ell - x_r}{\dot{x}_r} \tag{9}$$

$$y_r = ax_r^2 + bx_r + c, \quad \dot{y}_r = (2ax_r + b)\dot{x}_r \tag{10}$$

By setting x_ℓ, $\dot{\boldsymbol{x}}_\ell$, t_ℓ and x_r, all parameters are uniquely defined. The purpose of this paper is to calculate a sensitivity of x_ℓ with respect to model uncertainties, and to find the optimal x_r that minimizes the sensitivity.

III. Motion Design and Analysis

A. Throwing motion design

For a sensitivity analysis, the throwing motion of the manipulator is designed in this section. The position and orientation of the end effector are defined by a time function of 5th order polynomial as;

$$\boldsymbol{x}_e(t) = \sum_{i=0}^{5} \boldsymbol{c}_i^x t^i, \quad \phi_e(t) = \sum_{i=0}^{5} c_i^\phi t^i \tag{11}$$

By setting the boundary conditions as

$$\left.\begin{array}{lll} \boldsymbol{x}_e(0) = \boldsymbol{x}_0, & \dot{\boldsymbol{x}}_e(0) = 0, & \ddot{\boldsymbol{x}}_e(0) = 0 \\ \phi_e(0) = 0, & \dot{\phi}_e(0) = 0, & \ddot{\phi}_e(0) = 0 \\ \boldsymbol{x}_e(t_r) = \boldsymbol{x}_r, & \dot{\boldsymbol{x}}_e(t_r) = \dot{\boldsymbol{x}}_r, & \ddot{\boldsymbol{x}}_e(t_r) = \ddot{\boldsymbol{x}}_r \\ \phi_e(t_r) = \phi_r, & \dot{\phi}_e(t_r) = 0, & \ddot{\phi}_e(t_r) = 0 \end{array}\right\} \tag{12}$$

the coefficients in equation (11) are uniquely specified, where $\boldsymbol{x}_0$ is an initial position of the end-effector. To obtain the solutions in equation (12), a specified x_r is selected. Other parameters are obtained from equations (9), (10) and

$$\phi_r = -\tan^{-1}\frac{\dot{x}_r}{\dot{y}_r} \tag{13}$$

From equation (11), the throwing motion of the manipulator (joint angles, angular velocities and accelerations) are obtained as a time function from inverse kinematics computation. Moreover, from inverse dynamics analysis, joint torque is obtained through the motion. They are assumed to be represented by

$$\Theta = \left\{ \begin{array}{cccc} \boldsymbol{\theta}_1 & \boldsymbol{\theta}_2 & \cdots & \boldsymbol{\theta}_N \end{array} \right\} \tag{14}$$

$$T = \left\{ \begin{array}{cccc} \boldsymbol{\tau}_1 & \boldsymbol{\tau}_2 & \cdots & \boldsymbol{\tau}_N \end{array} \right\} \tag{15}$$

and $\dot{\Theta}$, $\ddot{\Theta}$ in discrete time domain at equal time intervals, where N means a data number at $t = t_r$. $\boldsymbol{\theta}_N$ and $\boldsymbol{\tau}_N$ are represented by;

$$\boldsymbol{\theta}_N = \boldsymbol{\theta}(t_r), \quad \boldsymbol{\tau}_N = \boldsymbol{\tau}(t_r) \tag{16}$$

B. Forward dynamics solution

The dynamic equation of this manipulator is given by

$$M(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + C(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + K(\boldsymbol{\theta}) = \boldsymbol{\tau}, \tag{17}$$

where M is an inertia matrix, C represents Coriolis and centrifugal forces, K represents gravity force. From equation (17), the following state-space formula is obtained.

$$\dot{\boldsymbol{q}} = \boldsymbol{f}(\boldsymbol{q}) + \boldsymbol{g}(\boldsymbol{q})\boldsymbol{\tau} \tag{18}$$

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{\theta}^T & \dot{\boldsymbol{\theta}}^T \end{bmatrix}^T \tag{19}$$

In equations (14) and (15), we have already obtained the trajectory of $\boldsymbol{q}$ and $\boldsymbol{\tau}$. Based on these time sequence data, equation (18) is linearly approximated around $\boldsymbol{q}_i$ and $\boldsymbol{\tau}_i$, and discretized to obtain solution for forward dynamics. The linear approximation of equation (18) around $\boldsymbol{q}_i$ and $\boldsymbol{\tau}_i$ is written as

$$\dot{\boldsymbol{q}}_i = A_i^c \boldsymbol{q}_i + B_i^c \boldsymbol{\tau}_i + C_i^c \tag{20}$$

$$A_i^c = \frac{\partial \boldsymbol{f}(\boldsymbol{q}_i)}{\partial \boldsymbol{q}} + \begin{bmatrix} \frac{\partial \boldsymbol{g}(\boldsymbol{q}_i)}{\partial q_1}\boldsymbol{\tau}_i & \cdots & \frac{\partial \boldsymbol{g}(\boldsymbol{q}_i)}{\partial q_6}\boldsymbol{\tau}_i \end{bmatrix} \tag{21}$$

$$B_i^c = \boldsymbol{g}(\boldsymbol{q}_i) \tag{22}$$

$$C_i^c = \boldsymbol{f}(\boldsymbol{q}_i) - A_i^c \boldsymbol{q}_i \tag{23}$$

By considering a trapezoidal integration (Tustin transformation),

$$\boldsymbol{q}_{i+1} = \boldsymbol{q}_i + \frac{\Delta T}{2}(\dot{\boldsymbol{q}}_{i+1} + \dot{\boldsymbol{q}}_i) \tag{24}$$

is satisfied, where ΔT means a sampling time. By substituting equation (20) into (24), the following discretized system is obtained.

$$\boldsymbol{q}_{i+1} = A_i \boldsymbol{q}_i + B_i^1 \boldsymbol{\tau}_i + B_i^2 \boldsymbol{\tau}_{i+1} + C_i \tag{25}$$

$$A_i = \left(I - \frac{\Delta T}{2} A_{i+1}^c\right)^{-1} \left(I + \frac{\Delta T}{2} A_i^c\right) \tag{26}$$

$$B_i^1 = \left(I - \frac{\Delta T}{2} A_{i+1}^c\right)^{-1} \frac{\Delta T}{2} B_i^c \tag{27}$$

$$B_i^2 = \left(I - \frac{\Delta T}{2} A_{i+1}^c\right)^{-1} \frac{\Delta T}{2} B_{i+1}^c \tag{28}$$

$$C_i = \left(I - \frac{\Delta T}{2} A_{i+1}^c\right)^{-1} \frac{\Delta T}{2} (C_i^c + C_{i+1}^c) \tag{29}$$

where I is an identity matrix. Moreover, by considering a multi-step ahead of $\boldsymbol{q}_i$ recursively, the joint angle and angular velocity of each joints at $t = t_r$ is given by

$$\boldsymbol{q}_N = \mathcal{A}\boldsymbol{q}_1 + \mathcal{B}\begin{bmatrix} \boldsymbol{\tau}_1 \\ \vdots \\ \boldsymbol{\tau}_N \end{bmatrix} + \mathcal{C} \tag{30}$$

$$\mathcal{A} = \prod_{i=1}^{N-1} A_i \tag{31}$$

$$\mathcal{B} = \begin{bmatrix} \prod_{i=2}^{N-1} A_i B_1^1 & \left(\prod_{i=2}^{N-1} A_i B_1^2 + \prod_{i=3}^{N-1} A_i B_2^1\right) \\ \cdots & \left(A_{N-2}B_{N-2}^2 + B_{N-1}^1\right) \quad B_{N-1}^2 \end{bmatrix} \tag{32}$$

$$\mathcal{C} = \prod_{i=2}^{N-1} A_i C_1 + \prod_{i=3}^{N-1} A_i C_2 + \cdots + C_{N-1} \tag{33}$$

In general, the solution of forward dynamics contains time integral of a nonlinear dynamics. On the other hand, equation (30) means that the solution is represented by an initial value and input sequence by considering linear approximation around a specified trajectory.

IV. Sensitivity analysis

A. Sensitivity of x_ℓ with respect to perturbation of $\boldsymbol{q}_N$

1) From $\Delta\boldsymbol{q}_N$ to $\Delta\boldsymbol{x}_r$: $\boldsymbol{x}_r$ and $\dot{\boldsymbol{x}}_r$ are calculated from the state-variable $\boldsymbol{q}_N$ by kinematic analysis. Suppose that the kinematic relationship of the manipulator is resented by

$$\boldsymbol{x}_r = \boldsymbol{F}(\boldsymbol{\theta}_N) \quad \dot{\boldsymbol{x}}_r = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\theta}}\dot{\boldsymbol{\theta}}_N \tag{34}$$

and the sensitivity of $\boldsymbol{x}_r$ and $\dot{\boldsymbol{x}}_r$ with respect to $\boldsymbol{q}_N$ are obtained by

$$\frac{\partial \boldsymbol{x}_r}{\partial \boldsymbol{q}_N} = \begin{bmatrix} \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\theta}} & 0 \end{bmatrix}, \quad \frac{\partial \dot{\boldsymbol{x}}_r}{\partial \boldsymbol{q}_N} = \begin{bmatrix} \frac{\partial^2 \boldsymbol{F}}{\partial \boldsymbol{\theta}^2}\dot{\boldsymbol{\theta}}_N & \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\theta}} \end{bmatrix} \tag{35}$$

where $\partial^2 \boldsymbol{F}/\partial \boldsymbol{\theta}^2 \ \dot{\boldsymbol{\theta}}_N$ is defined as

$$\frac{\partial^2 \boldsymbol{F}}{\partial \boldsymbol{\theta}^2}\dot{\boldsymbol{\theta}}_N = \begin{bmatrix} \frac{\partial^2 \boldsymbol{F}}{\partial \theta_1 \partial \boldsymbol{\theta}}\dot{\boldsymbol{\theta}}_N & \frac{\partial^2 \boldsymbol{F}}{\partial \theta_2 \partial \boldsymbol{\theta}}\dot{\boldsymbol{\theta}}_N & \frac{\partial^2 \boldsymbol{F}}{\partial \theta_3 \partial \boldsymbol{\theta}}\dot{\boldsymbol{\theta}}_N \end{bmatrix} \tag{36}$$

2) From $\Delta\boldsymbol{x}_r$ and $\Delta\dot{\boldsymbol{x}}_r$ to Δx_ℓ: Based on $\boldsymbol{x}_r$ and $\dot{\boldsymbol{x}}_r$, the flying locus of the object in equation (7) is decided by

$$\boldsymbol{a} = X^{-1}(x_r, \dot{x}_r) Y(y_r, \dot{y}_r) \tag{37}$$

$$X(x_r, \dot{x}_r) = \begin{bmatrix} x_r^2 & x_r & 1 \\ 2x_r\dot{x}_r & \dot{x}_r & 0 \\ 2\dot{x}_r^2 & 0 & 0 \end{bmatrix} \tag{38}$$

$$Y(y_r, \dot{y}_r) = \begin{bmatrix} y_r & \dot{y}_r & -g \end{bmatrix}^T \tag{39}$$

It causes same $\boldsymbol{a}$ as equation (8) if there are no uncertainties. Because x_ℓ is given by

$$x_\ell = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (a < 0) \tag{40}$$

as a solution of $y_\ell = 0$, the sensitivity of x_ℓ with respect to $\boldsymbol{x}_r$ and $\dot{\boldsymbol{x}}_r$ are given by

$$\frac{\partial x_\ell}{\partial \boldsymbol{x}_r} = \frac{\partial x_\ell}{\partial \boldsymbol{a}}\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{x}_r}, \quad \frac{\partial x_\ell}{\partial \dot{\boldsymbol{x}}_r} = \frac{\partial x_\ell}{\partial \boldsymbol{a}}\frac{\partial \boldsymbol{a}}{\partial \dot{\boldsymbol{x}}_r} \tag{41}$$

$$\frac{\partial x_\ell}{\partial \boldsymbol{a}} = \begin{bmatrix} \frac{b^2 - 2ac + bD}{2a^2 D} & -\frac{b + D}{2aD} & \frac{1}{D} \end{bmatrix} \tag{42}$$

$$D = \sqrt{b^2 - 4ac} \tag{43}$$

$$\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{x}_r} = \begin{bmatrix} \frac{\partial X^{-1}}{\partial x_r}Y & \frac{\partial X^{-1}}{\partial y_r}Y \end{bmatrix} + X^{-1}\frac{\partial Y}{\partial \boldsymbol{x}_r} \tag{44}$$

$$\frac{\partial \boldsymbol{a}}{\partial \dot{\boldsymbol{x}}_r} = \begin{bmatrix} \frac{\partial X^{-1}}{\partial \dot{x}_r}Y & \frac{\partial X^{-1}}{\partial \dot{y}_r}Y \end{bmatrix} + X^{-1}\frac{\partial Y}{\partial \dot{\boldsymbol{x}}_r} \tag{45}$$

where the relation of

$$\frac{\partial A^{-1}}{\partial x} = -A^{-1}\frac{\partial A}{\partial x}A^{-1} \tag{46}$$

will be utilized.

From equations (35) and (41), the sensitivity of x_ℓ with respect to a perturbation of joint angle and angular velocity $\boldsymbol{q}_N$ at the throwing point is obtained by

$$\frac{\partial x_\ell}{\partial \boldsymbol{q}_N} = \frac{\partial x_\ell}{\partial \boldsymbol{x}_r}\frac{\partial \boldsymbol{x}_r}{\partial \boldsymbol{q}_N} + \frac{\partial x_\ell}{\partial \dot{\boldsymbol{x}}_r}\frac{\partial \dot{\boldsymbol{x}}_r}{\partial \boldsymbol{q}_N} \tag{47}$$

B. Sensitivity of x_ℓ with respect to model uncertainty

1) Perturbation of initial position: There are two types of uncertainties on initialization. One is initial positioning of the end-effector, the other is zero adjustment of joint angles. The former will be overcome by position control, however, because many robots use incremental position sensors (encoders), the latter cannot be. A positioning jig or positioning sensor will be utilized for zero adjustment, however, because of elasticity of the jig or reaction region of the sensor, the joint angles would have uncertainty, which causes bias term of the joint angles. The throwing manipulator also has uncertainties on $\boldsymbol{\theta}(0)$. From equation (30), the sensitivity of $\boldsymbol{q}_N$ with respect to $\boldsymbol{\theta}(0)$ is obtained by

$$\frac{\partial \boldsymbol{q}_N}{\partial \boldsymbol{\theta}(0)} = \mathcal{A}\begin{bmatrix} I_3 \\ 0 \end{bmatrix} \tag{48}$$

where I_3 is $\mathbb{R}^{3\times 3}$ identity, and we assume $\dot{\boldsymbol{\theta}}(0)$ does not have any uncertainties (because it is strictly zero). The sensitivity of x_ℓ with respect to the perturbation of initial position is obtained by

$$\frac{\partial x_\ell}{\partial \boldsymbol{\theta}(0)} = \frac{\partial x_\ell}{\partial \boldsymbol{q}_N}\frac{\partial \boldsymbol{q}_N}{\partial \boldsymbol{\theta}(0)} \tag{49}$$

which is calculated using equation (47) and (48).

2) Disturbance by Coulomb friction: Joint friction is sometimes modeled by static, Coulomb and viscous frictions. Viscous friction will be well modeled. Static friction yields hysteresis but occurs only when the angular velocity is zero, therefore it is neglectable. At the same time, the modeling of Coulomb friction torque f_{Co} requires a signum function as

$$f_{Co} = -\mu_c \mathrm{sign}(\dot{\theta}) \tag{50}$$

where μ_c is a coefficient of Coulomb friction. Because it is a discontinuous function, it yields oscillation by friction compensation and Coulomb friction is sometimes regarded as a disturbance. In this section, the sensitivity of x_ℓ with respect to Coulomb friction is calculated.

In equation (14), we have already obtained the trajectory of the joint. Based on these data, the joint torque is changed as follows

$$\tau_i \leftarrow \begin{cases} \tau_i - \mu_c & (\dot{\theta} > 0) \\ \tau_i + \mu_c & (\dot{\theta} < 0) \end{cases} \tag{51}$$

and equation (30) is modified as follows

$$\boldsymbol{q}_N = \mathcal{A}\boldsymbol{q}_1 + \mathcal{B}\left(\begin{bmatrix} \boldsymbol{\tau}_1 \\ \vdots \\ \boldsymbol{\tau}_N \end{bmatrix} + \begin{bmatrix} \boldsymbol{\delta}_1 \\ \vdots \\ \boldsymbol{\delta}_N \end{bmatrix}\mu_c\right) + \mathcal{C} \tag{52}$$

where $\boldsymbol{\delta}_i$ is $\mathbb{R}^{3\times 1}$ vector with its elements -1 or 0 or 1. From equation (52), the sensitivity of $\boldsymbol{q}_N$ with respect to Coulomb friction is obtained by

$$\frac{\partial \boldsymbol{q}_N}{\partial \mu_c} = \mathcal{B}\begin{bmatrix} \boldsymbol{\delta}_1 \\ \vdots \\ \boldsymbol{\delta}_N \end{bmatrix} \tag{53}$$

and similarly to equation (49),

$$\frac{\partial x_\ell}{\partial \mu_c} = \frac{\partial x_\ell}{\partial \boldsymbol{q}_N}\frac{\partial \boldsymbol{q}_N}{\partial \mu_c} \tag{54}$$

is obtained, which is a sensitivity of landing distance with respect to Coulomb friction.

V. Trajectory Design Based on Sensitivity

A. Configuration of the manipulator

To obtain a low sensitive trajectory, a 3-DOF robot is established. The specifications of the physical parameters are shown in table I. By setting

TABLE I

PHYSICAL PARAMETERS OF THE MANIPULATOR

	length [m]	mass [kg]	coefficient of viscosity [Nms]
Link 1	ℓ_1 =1.0	m_1 =0.5	1.0
Link 2	ℓ_2 =1.0	m_2 =0.5	1.0
Link 3	ℓ_3 =0.2	m_3 =0.4	1.0

$$\boldsymbol{x}_\ell = \begin{bmatrix} 5 & 0 \end{bmatrix}^T, \quad \dot{\boldsymbol{x}}_\ell = \begin{bmatrix} 2 & -12 \end{bmatrix}^T \tag{55}$$

the locus of the flying object is defined. And by setting the initial position of the end effector $\boldsymbol{x}_e(0)$ as

$$\boldsymbol{x}_e(0) = \begin{bmatrix} 0 & 0.5 \end{bmatrix}^T \tag{56}$$

The landing time is set at $t_\ell = 2.5$ sec and the throwing trajectories of the manipulator are obtained by changing x_r. Figure 5 shows two examples of the throwing motion in

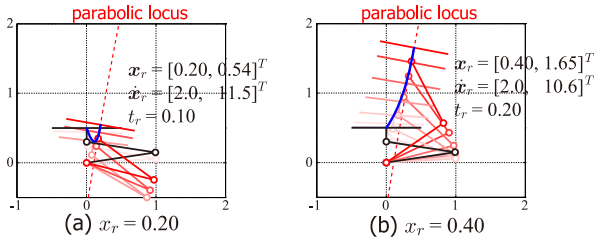

Fig. 5. Throwing trajectory of the manipulator

case of $x_r = 0.20$ and 0.40. The black line represents the initial posture and the red line shows the final position. Its transitions are also shown. The blue line shows the trajectory of the end-effector. The red dashed line shows the parabolic locus of the object which passes through $\boldsymbol{x}_\ell$.

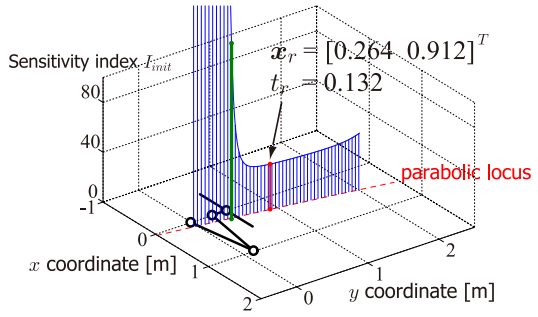

Fig. 6. Sensitivity of x_ℓ with respect to a perturbation of the initial position

B. Sensitivity for perturbation of initial position

Based on the proposed method in section IV, the sensitivity of x_ℓ with respect to a perturbation of the initial position is calculated. The result is shown in figure 6. Because sensitivity is obtained by $\mathbb{R}^{1\times 3}$ vector in equation (49), the sensitivity index is defined by

$$I_{init} = \left\| \frac{\partial x_\ell}{\partial \boldsymbol{\theta}(0)} \right\| \tag{57}$$

The initial position of the manipulator (nominal position) is shown by a black line. The red dashed line represents a parabolic locus of the object, and a release point R is set on this line by changing x_r from 0.104 to 0.47, which is inside of its workspace. I_{init} on each x_r are plotted on vertical axis. From this result, by setting $x_r = 0.264$, I_{init} is minimized, which means even though a perturbation of the initial value of joint angles, the object will come near the nominal landing point L.

C. Sensitivity for Coulomb friction

Same as a previous result, the sensitivity of x_ℓ with respect to Coulomb friction is calculated. Because equation (54) has a scalar value, the sensitivity index I_{Co} is defined by

$$I_{Co} = \left| \frac{\partial x_\ell}{\partial \mu_c} \right| \tag{58}$$

The results are shown in figure 7. From this result, by setting

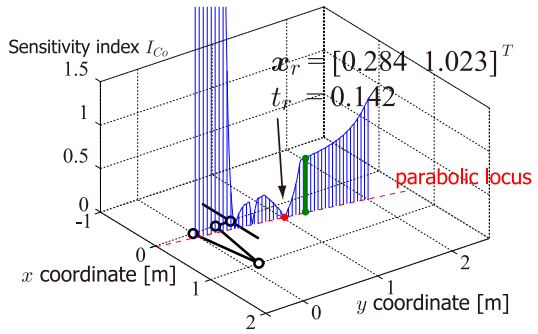

Fig. 7. Sensitivity of x_ℓ with respect to Coulomb friction

$x_r = 0.284$, I_{Co} is minimized, which means even though an existence of Coulomb friction, the object will come near the nominal landing point L.

VI. Throwing simulation

A. Throwing with a perturbation of initial joint angles

To show the effectiveness of the sensitivity based optimization of the throwing motion, throwing simulations are executed. By changing the initial position of the manipulator, the distribution of x_ℓ are obtained.

The initial joint angles are set by

$$\boldsymbol{\theta}(0) = \boldsymbol{\theta}_0 + N(0, \sigma^2) \tag{59}$$

where $\boldsymbol{\theta}_0$ is a nominal value and $N(0, \sigma^2)$ means Gaussian perturbation with zero mean and σ^2 variance. The robot is torque controlled (only feed forward controlled) and Monte-Carlo simulations are executed 2000 times. The optimal motion has $\boldsymbol{x}_r = [0.264 \;\; 0.912]^T$ with its throwing time $t_r = 0.132$. For comparison, a non-optimal throwing is selected with $\boldsymbol{x}_r = [0.180 \;\; 0.431]^T$, $t_r = 0.090$, which is indicated by green line in figure 6. Their sensitivity indexes are shown in table II. In these simulations, Runge-Kutta

TABLE II

Specification of motions (perturbation in initial joint angle)

	$\boldsymbol{x}_r$[m]	t_r[sec]	I_{init}
optimal	$[0.264,\ 0.912]^T$	0.132	36.1
non-optimal	$[0.180,\ 0.431]^T$	0.090	139.2

integral is utilized for forward dynamics computation.

The variance of the position of the landing point is shown in figure 8, which is a histogram of the positions. The red

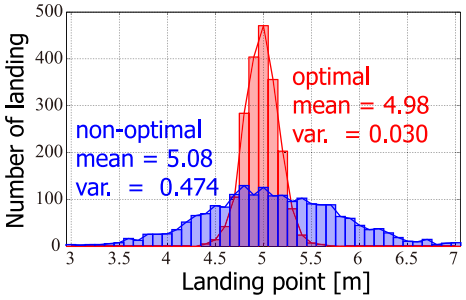

Fig. 8. Histogram of the landing point with respect to a perturbation of the initial joint angle

line represents the results of optimal throwing and the blue shows non-optimal. The means and variances of each result are shown in the figure. By using the optimal motion, the smaller variance of the landing point is realized. Moreover the ratio of I_{init} and a square root of variance (standard deviation) in each motion are almost the same

$$\frac{36.1}{139.2} = 0.259 \simeq \frac{\sqrt{0.030}}{\sqrt{0.474}} = 0.252 \tag{60}$$

This result shows that the sensitivity calculation by the proposed method has high accuracy. Figure 9 shows the loci of the object. The first 100 trials are shown with the same perturbation in both cases. The attatched movie shows the results of these simulations.

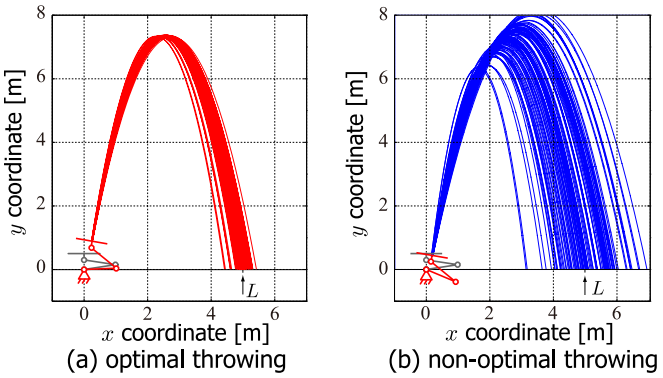

Fig. 9. Loci of flying object with perturbations of the initial joint angle

B. Throwing with Coulomb friction

Same as the previous simulation, the throwing simulations with Coulomb friction are executed. $x_r = 0.284$ is the optimal throwing which has smallest sensitivity for Coulomb friction. For comparison, $x_r = 0.330$ is selected which is shown by green line in figure7. Same as table II, the specification of two points of release are shown in table III. Once you prototype a 3-DOF manipulator, a coefficient of

TABLE III

SPECIFICATION OF MOTIONS (COULOMB FRICTION)

	$\boldsymbol{x}_r$ [m]	t_r [sec]	I_{Co}
optimal	$[0.284,\ 1.023]^T$	0.142	3.32×10^{-3}
non-optimal	$[0.330,\ 1.277]^T$	0.165	0.635

Coulomb friction μ_c will be obtained. However, in this paper, μ_c is regarded as uncertainty, and it is changed from 0 to 1 to calculate x_ℓ.

In equation (52), $\boldsymbol{\delta}_i$ is selected by using a nominal trajectory to obtain sensitivity. However, in forward dynamics simulations, the throwing motion is a little bit changed because of the friction term obtaind through dynamics simulations, therefore, in the simulation, $\boldsymbol{\delta}_i$ will be different from equation (52). The flying loci of the object in the existence of Coulomb friction are shown in figure 10. Landing point

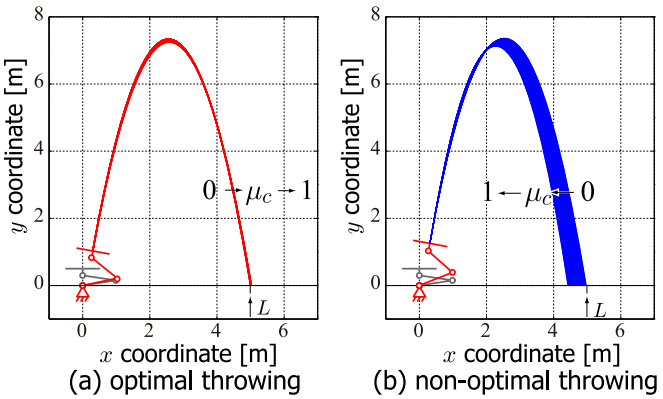

Fig. 10. Loci of flying object with Coulomb friction

L changes from 4.978m to 5.018m (difference is 0.040m) in the optimal motion, and from 4.978m to 4.430m (difference is -0.548m) in the non-optimal motion according to the change of μ_c from 0 to 1. These results show that by using the optimal motion the change of x_ℓ becomes smaller which shows small sensitivity.

VII. CONCLUSIONS

In this paper we focus on a throwing motion by a manipulator and design an optimal trajectory based on a sensitivity of the landing point with respect to model uncertainties. The results are summarized as follows

1) The dynamics of the manipulator are linearly approximated along the throwing trajectory and discretized using trapezoidal integral, which enables to represent the forward dynamics solution by the initial position and torque input sequence.
2) The sensitivity of the landing point with respect to model uncertainties are introduced. The perturbation of the initialization for joint angles and Coulomb friction are considered.
3) Monte-Carlo simulations are executed, and it is shown that the obtained sensitivity represents the variance of the landing point.
4) Simulation with Coulomb friction represents the effectiveness of the optimal trajectory design based on sensitivity.

These results have to be evaluated by experiments. In that case, the positioning uncertainties caused by human error has to be measured, which is our future work.

REFERENCES

[1] H. Tsukagoshi, E. Watari, K. Fuchigami, and A. Kitagawa, "Casting device for search and rescue aiming higher and faster access in disaster site," in *Proc. of the IIEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'12)*, 2012, pp. 4348–4353.

[2] A. Fagiolini, H. Arisumi, and A. Bicchi, "Casting robotic end-effectors to reach faraway moving objects," *Computing Research Repository (CoRR)*, vol. abs/1101.2268, 2011.

[3] M. T. Mason and K. M. Lynch, "Dynamic manipulation," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS1993)*, 1993, pp. 152–159.

[4] K. M. Lynch and M. T. Mason, "Dynamic underacutated nonprehensile manipulation," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS1996)*, 1996, pp. 889–896.

[5] H. Frank, D. Barteit, and F. Kupzog, "Throwing or shooting – a new thchnology for logistic chains within production system," in *Proc. of the IEEE International Conference on Technologies for Practical Robot Applications(TePRA 2008)*, 2008, pp. 62–67.

[6] T. Tabata and Y. Aiyama, "Passing manipulation by 1 degree-of-freedom manipulator - catching manipulation of tossed object without impact," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2003)*, vol. 3, 2003, pp. 2920–2925.

[7] A. Pekarovskiy and M. Buss, "Optimal control goal manifolds for planar nonprehensile throwing," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2013)*, 2013, pp. 4518–4542.