AUTOMATIC SENSOR FRAME IDENTIFICATION IN INDUSTRIAL ROBOTS WITH JOINT ELASTICITY*

Chung-Yen Lin
Dept. of Mechanical Engineering
University of California
Berkeley, California 94720
Email: chung_yen@berkeley.edu

Wenjie Chen
Dept. of Mechanical Engineering
University of California
Berkeley, California 94720
Email: wjchen@me.berkeley.edu

Masayoshi Tomizuka
Dept. of Mechanical Engineering
University of California
Berkeley, California 94720
Email: tomizuka@me.berkeley.edu

ABSTRACT

For robots with joint elasticity, discrepancies exist between the motor side information and the load side (i.e., end-effector) information. Therefore, high tracking performance at the load side can hardly be achieved when the estimate of load side information is inaccurate. To minimize such inaccuracies, it is desired to calibrate the load side sensor (in particular, the exact sensor location). In practice, the optimal placement of the load side sensor often varies due to the task variation necessitating frequent sensor calibrations. This frequent calibration need requires significant effort and hence is not preferable for industries which have relatively short product cycles. To solve this problem, this paper presents a sensor frame identification algorithm to automate this calibration process for the load side sensor, in particular the accelerometer. We formulate the calibration problem as a nonlinear estimation problem with unknown parameters. The Expectation-Maximization algorithm is utilized to decouple the state estimation and the parameter estimation into two separated optimization problems. An overall dual-phase learning structure associated with the proposed approach is also studied. Experiments are designed to validate the effectiveness of the proposed algorithm.

NOMENCLATURE

q_l	Load side position $\in \mathbb{R}^n$
$\dot{q}_m$	Motor side position $\in \mathbb{R}^n$
$\dot{q}_l$	Load side velocity $\in \mathbb{R}^n$
$\dot{q}_m$	Motor side velocity $\in \mathbb{R}^n$
$\ddot{q}_l$	Load side acceleration $\in \mathbb{R}^n$
$\ddot{q}_m$	Motor side acceleration $\in \mathbb{R}^n$
$M_l(q_l)$	Load side inertia matrix $\in \mathbb{R}^{n\times n}$
M_m	Motor side inertia matrix (diagonal) $\in \mathbb{R}^{n\times n}$
M_n	Nominal Load side inertia matrix (diagonal) $\in \mathbb{R}^{n\times n}$
$C(q_l,\dot{q}_l)$	Coriolis and centrifugal force matrix $\in \mathbb{R}^{n\times n}$
N	Gear ratio matrix (diagonal) $\in \mathbb{R}^{n\times n}$
$G(q_l)$	Gravity vector $\in \mathbb{R}^n$
D_l	Load side damping matrix (diagonal) $\in \mathbb{R}^{n\times n}$
D_m	Motor side damping matrix (diagonal) $\in \mathbb{R}^{n\times n}$
D_J	Joint damping matrix (diagonal) $\in \mathbb{R}^{n\times n}$
K_J	Joint stiffness matrix (diagonal) $\in \mathbb{R}^{n\times n}$
τ_m	Motor torque input $\in \mathbb{R}^n$
$f_m(\dot{q}_m)$	Friction effect on motor side $\in \mathbb{R}^n$
$f_l(\dot{q}_l)$	Friction effect on load side $\in \mathbb{R}^n$
f_{ext}	External force acting on end-effector $\in \mathbb{R}^6$
d_m	Motor side fictitious disturbance torque $\in \mathbb{R}^n$
d_l	Load side fictitious disturbance torque $\in \mathbb{R}^n$
a_l	Translational acceleration $\in \mathbb{R}^3$
$a_{l,s}$	Measurement of translational acceleration $\in \mathbb{R}^3$
$q_{m,s}$	Measurement of motor side position $\in \mathbb{R}^n$
R_i^{i-1}	Rotation matrix describing the frame $\{i\}$ relative to the frame $\{i-1\} \in \mathbb{R}^{3\times 3}$
$\bar{J}_s(q_l;\xi_s)$	The first three rows of the Jacobian matrix mapping from the load side joint space to the sensor frame Cartesian space $\in \mathbb{R}^{3\times n}$
$\dot{\bar{J}}_s(q_l;\xi_s)$	Derivative of the first three rows of the Jacobian matrix $J_s(q_l;\xi_s) \in \mathbb{R}^{3\times n}$
$J(q_l)$	Jacobian matrix mapping from the load side joint space to the end-effector Cartesian space $\in \mathbb{R}^{6\times n}$

*THIS WORK WAS SUPPORTED BY FANUC CORPORATION, JAPAN. REAL-TIME CONTROL HARDWARE AND SOFTWARE WERE PROVIDED BY NATIONAL INSTRUMENTS, INC.

n_m	Measurement noise of motor-side encoder $\in \mathbb{R}^n$
n_a	Measurement noise of accelerometer $\in \mathbb{R}^3$
n_w	Processes noise $\in \mathbb{R}^n$
$q_{l,s}$	Measurement of load side position $\in \mathbb{R}^n$

INTRODUCTION

For industrial robots with indirect drive mechanisms (e.g., gear transmissions), the load side encoder is often not available due to the cost and assembly issues. Using motor side encoder signals for feedback control, however, may not guarantee satisfactory control performance at the load side due to the robot joint dynamics. To overcome this problem, it has been suggested to attach a low-cost MEMS accelerometer at the robot end-effector to measure the translational acceleration [1–3]. Then, the accelerometer measurements are incorporated with the motor encoder signals to obtain an estimate of the end-effector position and/or velocity. Although these methods have successfully improved the load side tracking performance [4], there still remain various practical issues when applying these techniques to industrial robot systems. For example, the accelerometer location must be precisely known, but manually calibrating the sensor location requires a long period of time. Thus, the sensor based estimation scheme is difficult for industrial applications which involve frequent changes of sensor location due to task variations. More specifically, the load side accelerometer is manually mounted at the location which requires the most attention for performance improvement. Such locations may change from one task to another due to task variations. The accelerometer may be mounted magnetically to the robot end-effector so that the accelerometer location can be changed easily. In such situation, the sensor frame location needs to be identified every time when the sensor moves to a new location. A standard sensor frame calibration method [5] is to command the robot to move such that the designated sensor location touches a reference point with different postures. Then, the sensor frame location can be computed using the robot posture information and the corresponding accelerometer measurements. This calibration process requires significant effort. Therefore, developing an algorithm that can automatically detect the accelerometer's mounting position and orientation becomes an important issue.

In this work, we aim to design an algorithm which deals with the load side sensor frame identification problem using only accelerometer and motor encoder measurements over a designed trajectory. The main difficulty of this problem is that the unknown system parameters (i.e., the sensor frame location) have complex dependencies on the immeasurable system states (i.e., the load side information). Such problem is usually referred to as the "parameter estimation using incomplete data" problem, which has been studied by a number of researchers [6–8]. Among the algorithms presented in this area, the Expectation-Maximization (EM) algorithm is computationally simple and compatible with the state space model. While the algorithms presented in [7] only works for linear state space model, we propose a modified algorithm that can deal with the specific non-linear problem in this paper.

This paper begins with the introduction of a standard robot dynamic model and parameterizing the sensor frame location using the Denavit-Hartenberg parameters [9]. Then, a stochastic model is designed to parameterize the model uncertainties in terms of the statistics in a Gaussian distribution. Once the model is established, the EM algorithm is applied to accomplish the state estimation and parameter estimation simultaneously. The extended Kalman filter, extended Kalman smoother, and Monte Carlo integration [10] techniques are used to obtain the information we needed in the EM algorithm. This paper also introduces a dual-phase learning structure in which the unknown parameters can be estimated systematically. The effectiveness of the proposed load side sensor frame identification algorithm is validated by experimentation on a single joint robot test-bed.

SYSTEM

Multi-joint Indirect Drive Model

Consider an n-joint manipulator with elastic joints described by [4]:

$$\begin{aligned} &M_l(q_l)\ddot{q}_l + C(q_l,\dot{q}_l)\dot{q}_l + G(q_l) + D_l\dot{q}_l + f_l(\dot{q}_l) \\ &\quad = K_J\left(N^{-1}q_m - q_l\right) + D_J\left(N^{-1}\dot{q}_m - \dot{q}_l\right) - J^T(q_l)f_{\text{ext}} \\ &M_m\ddot{q}_m + D_m\dot{q}_m + f_m(\dot{q}_m) \\ &\quad = \tau_m - N^{-1}\left(K_J\left(N^{-1}q_m - q_l\right) + D_J\left(N^{-1}\dot{q}_m - \dot{q}_l\right)\right) \end{aligned}$$

where $f_m(\dot{q}_m)$ and $f_l(\dot{q}_l)$ account for the motor side and load side frictions respectively. The definitions of all the variables and quantities are listed in the nomenclature.

We then introduce the decoupled model by defining the nominal load side inertia matrix as follows:

$$M_n = \text{diag}\left(M_{l,11}(q_{l0}),\ldots,M_{l,nn}(q_{l0})\right)$$

where q_{l0} is the robot home position and $M_{l,ii}$ is the (i,i)-th element of the load side inertia matrix M_l. By separating the load side inertia matrix M_l into the nominal inertia matrix M_n and the coupling terms, we obtain:

$$\begin{aligned} &M_n\ddot{q}_l + D_l\dot{q}_l - d_l(q_l,q_m,\dot{q}_l,\dot{q}_m) \\ &\quad = K_J\left(N^{-1}q_m - q_l\right) + D_J\left(N^{-1}\dot{q}_m - \dot{q}_l\right) \qquad (1) \\ &M_m\ddot{q}_m + D_m\dot{q}_m - d_m(\dot{q}_m) \\ &\quad = \tau_m - N^{-1}\left(K_J\left(N^{-1}q_m - q_l\right) + D_J\left(N^{-1}\dot{q}_m - \dot{q}_l\right)\right) \qquad (2) \end{aligned}$$

where all the variable are diagonal matrices or vectors. In (1)-(2), we introduced fictitious disturbance torque vectors d_m and d_l, which account for the coupling and nonlinear dynamics (e.g.,

off-diagonal terms in the load side inertia matrix, Coriolis and centrifugal force, gravity, friction, etc.). More specifically, these fictitious disturbances can be formulated as:

$$\begin{aligned} d_l = &\left[M_n M_l^{-1}(q_l) - I_n\right]\left[K_J\left(N^{-1}q_m - q_l\right) + D_J\left(N^{-1}\dot{q}_m - \dot{q}_l\right)\right. \\ &-D_l\dot{q}_l] - M_n M_l^{-1}(q_l)\left[C(q_l,\dot{q}_l)\dot{q}_l + G(q_l)\right. \\ &\left.+f_l(\dot{q}_l) + J^T(q_l) f_{\text{ext}}\right] \\ d_m = &-f_m(\dot{q}_m) \end{aligned}$$

where I_n is an $n \times n$ identity matrix.

Measurement Model

Consider a 3-axial accelerometer mounted on the n-th link of a robot. We parameterize the sensor frame by using the Denavit-Hartenberg (DH) parameters $\xi_s = \{r_s, \alpha_s, d_s, \theta_s\}$ (see [9] for more details). We thus can express the corresponding three-dimensional translational acceleration in the sensor frame $\{s\}$ in terms of DH parameters:

$$\begin{aligned} a_l =& R_0^s(q_l;\xi_s)\left(\bar{J}_s(q_l;\xi_s)\ddot{q}_l + \bar{\dot{J}}_s(q_l,\dot{q}_l;\xi_s)\dot{q}_l + g\right) \\ \triangleq& h_a(q_l,\dot{q}_l;\xi_s) \end{aligned} \tag{3}$$

where $g = [0,0,-9.8]^T$ m/sec^2 is the gravity vector and $R_0^s(q_l;\xi_s)$ is the rotation matrix describing the robot base frame $\{0\}$ relative to the sensor frame $\{s\}$. Note that since the accelerometer captures only the translational acceleration, the Jacobian matrix $\bar{J}_s(q_l;\xi_s) \in \mathbb{R}^{3\times n}$ and the differential Jacobian matrix $\bar{\dot{J}}_s(q_l,\dot{q}_l;\xi_s) \in \mathbb{R}^{3\times n}$ refer to the first three rows of the standard Jacobian matrices respectively.

Besides the accelerometer, the motor side encoder for each joint is also assumed to be available. Thus, the measurement model becomes:

$$\begin{bmatrix} q_{m,s} \\ a_{l,s} \end{bmatrix} = \begin{bmatrix} q_m \\ h_a(q_l,\dot{q}_l;\xi_s) \end{bmatrix} + \begin{bmatrix} n_m \\ n_a \end{bmatrix}$$

where $\left[n_m^T, n_a^T\right]^T$ are introduced to denote the measurement noises in the sensor system.

State Space Representation

By selecting the state vector as $x_i = [q_{mi}, \dot{q}_{mi}, q_{li}, \dot{q}_{li}]^T$, the system dynamics for the i-th joint (1)-(2) is formulated as:

$$\dot{x}_i(t) = A_{ci}x_i(t) + B_{uci}\tau_{mi}(t) + B_{dci}\begin{bmatrix} d_{mi}(t) \\ d_{li}(t) \end{bmatrix} \tag{4}$$

where $A_{ci} \in \mathbb{R}^{4\times4}$, $B_{uci} \in \mathbb{R}^{4\times1}$, and $B_{dci} \in \mathbb{R}^{4\times2}$ are given by:

$$\begin{aligned} A_{ci} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{K_{Ji}}{N^2M_{mi}} & -\frac{D_{mi}+D_{ji}/N^2}{M_{mi}} & \frac{K_{Ji}}{NM_{mi}} & \frac{D_{ji}}{NM_{mi}} \\ 0 & 0 & 0 & 1 \\ \frac{K_{Ji}}{NM_{li}} & \frac{D_{Ji}}{NM_l} & -\frac{K_{Ji}}{M_{mi}} & \frac{D_{Ji}+D_{li}}{M_{li}} \end{bmatrix} \\ B_{uci} &= \begin{bmatrix} 0 & \frac{1}{M_{mi}} & 0 & 0 \end{bmatrix}^T \\ B_{dci} &= \begin{bmatrix} 0 & \frac{1}{M_{mi}} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{M_{li}} \end{bmatrix}^T \end{aligned}$$

where $\bullet_i$ denotes the i-th (or the (i,i)-th) element of the corresponding vector (matrix) $\bullet$.

Then, by combining the dynamics for all joints, we obtain the overall state vector $x = \left[x_1^T, \cdots, x_n^T\right]^T \in \mathbb{R}^{4n}$, the input vector $u = [\tau_{m1}, \cdots, \tau_{mn}]^T \in \mathbb{R}^n$, and the disturbance vector $d = [d_{m1}, d_{l1}, \cdots, d_{mn}, d_{ln}]^T \in \mathbb{R}^{2n}$. The corresponding system matrices are given as:

$$A_c = \begin{bmatrix} A_{c1} & 0 & \cdots & 0 \\ 0 & A_{c2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{cn} \end{bmatrix}$$

$$B_{uc} = \begin{bmatrix} B_{uc1} & 0 & \cdots & 0 \\ 0 & B_{uc2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{ucn} \end{bmatrix}$$

$$B_{dc} = \begin{bmatrix} B_{dc1} & 0 & \cdots & 0 \\ 0 & B_{dc2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{dcn} \end{bmatrix}$$

With the system matrices $\{A_c, B_{uc}, B_{dc}\}$, the zero-order-hold (ZOH) equivalent model can be obtained as [11]:

$$x(k+1) = A_d x(k) + B_{ud}u(k) + B_{dd}d(k) + n_w(k) \tag{5}$$

where $\{A_d, B_{ud}, B_{dd}\}$ are the transition matrix, the input matrix, and the disturbance matrix of the discrete-time model respectively. The state vector $x(k)$ is considered to be random. We assume that the initial state vector has a Gaussian distribution $\mathcal{N}(\mu,\Sigma)$, while the process noise n_w has independent and identically distributed (i.i.d.) Gaussian distribution $\mathcal{N}(0,W_d)$ at each time step k.

On the other hand, the discrete-time measurement model can

be obtained by:

$$y(k) = \begin{bmatrix} q_m \\ h_a(q_l(k), \dot{q}_l(k); \xi_s) \end{bmatrix} + \begin{bmatrix} n_m(k) \\ n_a(k) \end{bmatrix} \triangleq h(x(k); \xi_s) + \begin{bmatrix} n_m(k) \\ n_a(k) \end{bmatrix} \quad (6)$$

where $y = \left[q_{m,s}^T, a_{l,s}^T\right]^T \in \mathbb{R}^{n+3}$ is the output vector; $\left[n_m^T, n_a^T\right]^T$ is the measurement noise vector that is assumed to have i.i.d. zero-mean Gaussian distribution with covariance V_d.

ALGORITHM

The measurement model in (6) shows that the translational acceleration is a function of DH parameters and load side information. Therefore, if we can measure all quantities in the model, i.e., $q_{l,s}, \dot{q}_{l,s}$, the sensor frame identification problem can be solved by:

$$\min_{\xi_s} \sum_{k=1}^{L} \left\| a_l(k) - h_a(q_{l,s}(k), \dot{q}_{l,s}(k); \xi_s) \right\|_2^2 \quad (7)$$

where L is the number of data samples.

However, in the case of robots with indirect drive mechanisms, the load side joint space position and velocity are usually not measured directly. Thus, the standard system identification problem in (7) breaks down. Another algorithm is needed to address this problem in the case of incomplete data.

Sensor Frame Identification

Given a batch of observations $Y(L) = \{y(1), \ldots, y(L)\}$, we aim to maximize the likelihood of $Y(L)$ in the stochastic model (5)-(6). Such optimization problem is intractable since the state vectors are random. In this case, we can apply the EM algorithm to iteratively achieve the optimum.

Due to the stochastic property of the dynamic model, we cannot compute the likelihood for the unknown parameter directly. Therefore, we first introduce the expected complete log likelihood $\mathcal{L}(x, Y(L); \xi_s, \eta) \triangleq E\{l_c(x, Y(L); \xi_s, \eta | Y(L))\}$, where the parameter η denotes the collection of the statistics $\{\mu, \Sigma, W_d\}$ in the stochastic model. The expected complete log likelihood corresponding to the proposed model (5)-(6) can be obtained in the following form:

$$\begin{aligned} &\mathcal{L}(x, Y(L); \xi_s, \eta) \\ &= -\frac{1}{2} \int (x(1) - \mu)^T \Sigma^{-1} (x(1) - \mu) p(x|Y(L)) dx(1) \\ &\quad - \frac{1}{2} \sum_{k=1}^{L-1} \iint \tilde{x}^T(k+1) W_d^{-1} \tilde{x}(k+1) p(x|Y(L)) dx(k+1) dx(k) \\ &\quad - \frac{1}{2} \sum_{t=1}^{L} \int \tilde{y}^T(k) V_d^{-1} \tilde{y}(k) p(x|Y(L)) dx(k) \\ &\quad - \frac{1}{2} \log\det(\Sigma) - \frac{L-1}{2} \log\det(W_d) - \frac{L}{2} \log\det(V_d) + c \end{aligned} \quad (8)$$

where $\tilde{x}(k+1) = x(k+1) - A_d x(k) - B_{ud} u(k) - B_{dd} d(k)$, $\tilde{y}(k) = y(k) - h(x(k); \xi_s)$, and c is a constant that accounts for the constant coefficients in Gaussian distribution.

As shown in [6], maximizing the expected complete log likelihood with respect to the parameter is equivalent to maximizing the lower bound of the log likelihood with respect to the parameter. We thus deal with the optimization problem for the expected likelihood:

$$\xi_s \leftarrow \arg\max_{\xi_s} \mathcal{L}(x, Y(L); \xi_s, \eta)$$

where $\leftarrow$ denotes the assignment operator that resets the parameter in the left hand side by the value computed from the right hand side. Note that the statistics η are assumed to be known at this point. Since (8) is a function with deterministic quantities, we can apply gradient ascent method to yield the optimum. The parameter can be updated as follows:

$$\xi_s \leftarrow \xi_s + \lambda \nabla_{\xi_s} \mathcal{L}(x, Y(L); \xi_s, \eta) \quad (9)$$

where the i-th component of the gradient $\nabla_{\xi_s} \mathcal{L}(x, Y(L); \xi_s, \eta)$can be obtained in the following form:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \xi_{si}} &= \frac{\partial}{\partial \xi_{si}} \left\{ -\frac{1}{2} \sum_{k=1}^{L} E\left\{ \tilde{y}^T(k) V_d^{-1} \tilde{y}(k) \middle| Y(L) \right\} \right\} \\ &= -\frac{1}{2} \sum_{k=1}^{L} E\left\{ \mathrm{tr}\left(2V_d^{-1} \tilde{y}(k) \frac{\partial \tilde{y}^T(k)}{\partial \xi_{si}} \right) \middle| Y(L) \right\} \end{aligned} \quad (10)$$

Usually, it is hard to find the closed form for the expected value of the nonlinear function in (10). The corresponding deterministic function (i.e., without computing the expectation), however, can often be obtained. We thus draw samples from the stochastic model and approximate the expected value of $\tilde{y}(k)$ by Monte Carlo integration. Applying Monte Carlo integration to (10)

yields:

$$\frac{\partial \mathcal{L}}{\partial \xi_{si}} = -\sum_{k=1}^{L} \mathrm{tr}\left(V_d^{-1} E\left\{ \tilde{y}(k) \frac{\partial \tilde{y}^T(k)}{\partial \xi_{si}} \middle| Y(L) \right\} \right)$$
$$\approx \frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{L} \mathrm{tr}\left(V_d^{-1} \left(y(k) - h(x^i(k);\xi_s) \right) \frac{\partial h^T(x^i(k);\xi_s)}{\partial \xi_{si}} \right) \tag{11}$$

where M is the number of samples for each random vector $x(k)$, $x^i(k)$ is the i-th sample drawn from i.i.d Gaussian $\mathcal{N}\left(\hat{x}^s(k), S(k)\right)$, $\hat{x}^s(k) = E\left\{x(k)|Y(L)\right\}$ is the state estimate, and $S(k) = E\left\{ \left(x(k) - \hat{x}^s(k)\right)\left(x(k) - \hat{x}^s(k)\right)^T | Y(L) \right\}$ is the covariance of state estimation error. The computation of $\frac{\partial h^T(x^i(k);\xi_s)}{\partial \xi_{si}}$ can be completed by a closed form derivation. The computation of $\hat{x}^s(k)$ and $S(k)$ will be introduced in the following section.

Load Side State Estimation

To complete the EM iteration, we need to estimate the state vector and its corresponding error covariance. More specifically, each time after we update the parameter by (9), the state vector and its distribution need to be re-estimated for computing the Monte Carlo integration in (11) for the next step. It is important to note that the expectation of the system states are evaluated with respect to the distribution $p(x|Y(L))$ with the current estimated parameter ξ_s.

To do this, we apply the extended Kalman filtering (EKF) and extended Kalman smoothing (EKS) techniques using the stochastic model. The extended Kalman filter is given by:

$$\hat{x}^o(k) = A_d \hat{x}(k-1) - B_{ud} u(k-1) - B_{dd} \hat{d}(k-1) \tag{12}$$
$$M(k) = A_d Z(k-1) A_d^T + W_d \tag{13}$$
$$K(k) = M(k) C_d^T(k) \left(C_d(k) M(k) C_d^T(k) + V_d \right)^{-1} \tag{14}$$
$$Z(k) = \left(I - K(k) C_d(k)\right) M(k) \tag{15}$$
$$\hat{x}(k) = \hat{x}^o(k) + K(k)\left(y(k) - h(\hat{x}^o(k);\xi_s)\right) \tag{16}$$

where $\hat{x}^o(k) = E\left\{x(k)|Y(k-1)\right\}$ and $\hat{x}(k) = E\left\{x(k)|Y(k)\right\}$ are the *a-priori* and the *a-posteriori* state estimate respectively; $M(k) = E\left\{ \left(x(k) - \hat{x}^o(k)\right)\left(x(k) - \hat{x}^o(k)\right)^T | Y(k-1) \right\}$ and $Z(k) = E\left\{ \left(x(k) - \hat{x}(k)\right)\left(x(k) - \hat{x}(k)\right)^T | Y(k) \right\}$ are the covariance of the *a-priori* and the *a-posteriori* state estimation error respectively; $\hat{d}(k)$ is the disturbance term approximated by $\hat{x}(k)$; $C_d(k) = \nabla h(x;\xi_s)|_{x=\hat{x}^o(k)}$ is the linear approximation of $h(x(k);\xi_s)$ defined in the following way:

$$h(x(k);\xi_s) \approx h(\hat{x}^o(k);\xi_s) + \nabla h(x;\xi_s)|_{x=\hat{x}^o(k)} \left(x(k) - \hat{x}^o(k)\right)$$
$$\triangleq h(\hat{x}^o(k);\xi_s) + C_d(k)\left(x(k) - \hat{x}^o(k)\right)$$

where $C_d(k)$ can be computed by a closed form derivation for the simple case or be approximated by the first order numerical differentiation.

Then, applying the extended Kalman smoother yields

$$J(k) = Z(k) A_d^T(k) M^{-1}(k) \tag{17}$$
$$S(k) = Z(k) + J(k)\left(S(k+1) - M(k+1)\right) J^T(k) \tag{18}$$
$$\hat{x}^s(k) = \hat{x}(k) + J(k)\left(\hat{x}^s(k+1) - \hat{x}^o(k+1)\right) \tag{19}$$

where $\hat{x}^s(k) = E\left\{x(k)|Y(L)\right\}$ is the state estimate after smoothing; $S(k) = E\left\{ \left(x(k) - \hat{x}^s(k)\right)\left(x(k) - \hat{x}^s(k)\right)^T | Y(L) \right\}$ is the covariance of state estimation error after smoothing.

By collecting the estimation result in (18)-(19) for $k = 1, \ldots, L$, we can continue to update the model parameters ξ_s using the update law in (9) and the Monte Carlo integration in (11). The algorithm is running iteratively until the model parameters converge.

Furthermore, it is seen that (12)-(16) is identical to the standard extended Kalman filter, which can be implemented as a state observer for real-time feedback control. Therefore, the state estimation problem and the system identification problem are solved simultaneously by the proposed algorithm.

Tuning of the Statistics in Stochastic Model

It is important to note that the effectiveness of the EKF/EKS techniques relies on the a-priori knowledge of the stochastic model. In practice, the measurement noise covariance matrix V_d can often be obtained experimentally. The process noise W_d, however, is often not available since it accounts for the nonlinear dynamic model uncertainty. We thus apply a similar parameter learning technique (i.e., the EM algorithm) to estimate these noise covariances.

In this case, we assume that the sensor frame parameter ξ_s is known and we try to deal with the optimization problem:

$$\eta \leftarrow \arg\max_{\eta} \mathcal{L}\left(x, Y(L); \xi_s, \eta\right) \tag{20}$$

The argument of the maximization problem can be found by:

$$\mu \leftarrow \hat{x}^s(1) \tag{21}$$
$$\Sigma \leftarrow S(1) \tag{22}$$
$$W_d \leftarrow \frac{1}{L-1} \sum_{k=1}^{L-1} E\left\{ \tilde{x}(k+1)\tilde{x}^T(k+1) \middle| Y(L) \right\} \tag{23}$$

where $\tilde{x}(k)$ is defined in (8).

Again, since the fictitious disturbance term $d(k)$ in $\tilde{x}(k)$ is a nonlinear function of $x(k)$, we use Monte Carlo integration to approximate the expected value. As a result, the parameter update

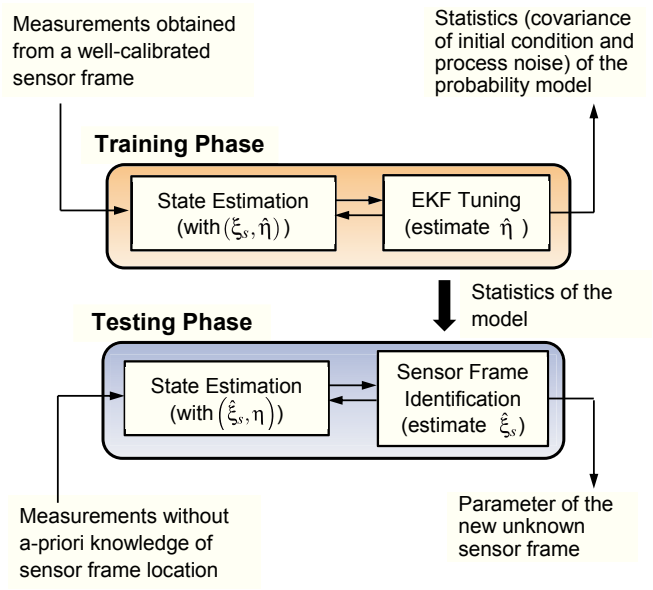

Figure 1: DUAL PHASE LEARNING STRUCTURE

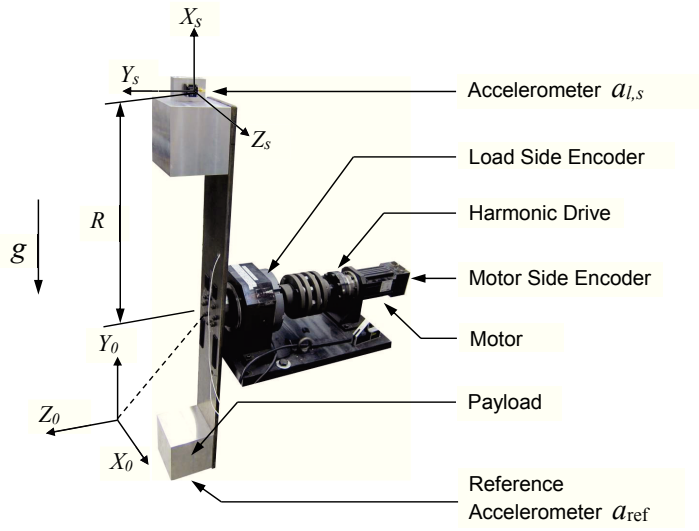

Figure 2: SINGLE JOINT ROBOT TEST-BED.

law is found as follows:

$$W_d \approx \frac{1}{M(L-1)} \sum_{i=1}^{M} \sum_{k=1}^{L-1} \tilde{x}^i(k+1)\tilde{x}^{iT}(k+1) \qquad (24)$$

where M is the number of sample and $\tilde{x}^i(k)$ is the i-th sample of the random vector $\tilde{x}(k)$ drawn from i.i.d Gaussian distribution of $x(k) \sim \mathcal{N}(\hat{x}^s(k), S(k))$ based on the observations of the whole time series.

Dual-Phase Learning Structure

Figure 1 shows the dual-phase learning structure for the automatic sensor frame identification problem, where the sensor frame parameters ξ_s and the statistics of the stochastic model η are both unknown. This problem is formulated into two parts. Namely, we first identify the statistics in the stochastic model using the measurement from a well-calibrated sensor frame (i.e., the training phase), and then we estimate the parameter of the new un-calibrated sensor frame (i.e., the testing phase).

The first part is referred to as the training phase, in which we try to find a stochastic model that best fits the measurements and the system dynamics. In this phase, the data is obtained by a well-calibrated sensor (i.e., the parameter of the sensor frame is known) so that the assumption in (20) is satisfied. Thus, (21)-(24) can be computed and the estimate of η is obtained.

We then utilize the stochastic model obtained from the training phase to estimate the unknown sensor frame. Since the statistics of the stochastic model has been computed, we can run the parameter update law (9)-(11) considering η as known quantities.

Note that, the training phase is normally required to perform only once and the resulting estimate of η should be good for the testing phase for any unknown sensor frame. This learning structure, however, may not be appropriate in the case where the calibrated sensor frame in the training phase is far away from the unknown sensor frame. Since the stochastic model is trained at a specific sensor frame, it may not be able to represent the system dynamics globally. In the following experimental study, however, we will show that the proposed algorithm is robust against the unknown sensor frame at various locations which are not close to the calibrated sensor frame in the training phase.

EXPERIMENTAL STUDY

System Setup

The proposed algorithm is implemented to identify an accelerometer frame on a single-joint robot test-bed as shown in Fig. 2. The system is equipped with a motor side 20,000 counts/revolution encoder, a load side 144,000 counts/revolution encoder, and two accelerometers (Kistler, Type: 8330A3) which are mounted at the ends of the payload symmetrically. Among these sensors, the motor side encoder and one of the accelerometers (i.e., $a_{l,s}$ in Fig. 2) are assumed to be available for the entire algorithm, where the other accelerometer (i.e., a_{ref} in Fig. 2) is utilized for the training phase only. The load side encoder is utilized for performance validation. The DH parameters of the accelerometers' locations are listed in Tab.1. A basic controller used in this single-joint system is a PID controller with state feedback. The control law is implemented in a LabVIEW

Table 1: DENAVIT-HARTENBERG PARAMETERS OF THE SENSOR FRAME IN THE SINGLE JOINT SYSTEM.

	r_s (mm)	α_s (rad)	d_s (mm)	θ_s (rad)
$a_{l,s}$	$R = 600$	$\pi/2$	0	q_l
a_{ref}	$R = 600$	$\pi/2$	0	$q_l + \pi$

real-time target installed with LabVIEW Real-Time and FPGA modules. The sampling rate of data acquisition and controller is 1 kHz.

Sensor Frame Identification

In this experiment, the single-joint system is designed to track a load side trajectory $q_{l,r} = \sin(0.5\pi t + 0.5\pi)$ rad. The tracking performance, however, is not of interest since the main objective is to test the capability of the algorithm for identifying an unknown sensor frame. Note that the DH parameter d_s (i.e., the offset between $\{s\}$ and $\{o\}$ along Z_o) cannot be identified in this setup, since the measurements of the accelerometer do not change when we move the sensor frame $\{s\}$ from one location to another along Z_o direction. In this work, we assume that r_s is the only unknown parameter since the orientation (i.e., α_s and θ_s) can be computed directly using the gravity effect.

Figure 3 shows the monotonic convergence of the estimated sensor frame location using the proposed algorithm. The red dash line represents the true value of the DH parameter r_s whereas the solid blue line represents the estimate of the DH parameter $\hat{r}_s$ in each iteration. In general, the accuracy and the variance of the estimation result depend significantly on the number of the sample data M in Monte Carlo integration. Also, since the Monte-Carlo integration is a randomized algorithm (i.e., the result can be changed from one trial to another), we repeat the above estimation process several times to show the robustness of the algorithm for different random Monte-Carlo samples. Table 2 shows the average error and the standard deviation of the parameter estimation results for 15 trials using different values of M. It shows that the standard deviation of the parameter estimation is small. Also, it is clearly seen that the execution time increases proportionally with M, which implies that the Monte Carlo integration dominates the computation of this algorithm.

We also want to demonstrate the performance robustness of the algorithm against the uncertainty induced by different sensor frame locations. Note that two accelerometers are mounted symmetrically and thus the load side angular acceleration can be computed by $\ddot{q}_{l,s} = \frac{a_{l,s}+a_{\text{ref}}}{2R}$, where R is the true value of r_s. We thus can generate a signal as follows that mimics the measurement of a fictitious accelerometer mounted at a location with DH

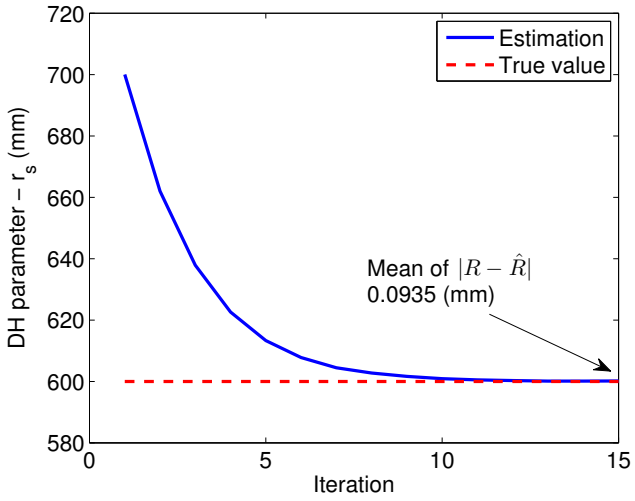

Figure 3: CONVERGING OF THE PARAMETER WHEN $M = 15$.

Table 2: PARAMETER ESTIMATION (THE LAST ITERATION) RESULT FOR DIFFERENT NUMBERS OF SAMPLE POINTS IN MONTE CARLO INTEGRATION.

	Mean of $\lvert R-\hat{R}\rvert$	Standard Deviation $\sigma(\hat{R})$	Execution Time
$M = 3$	0.1667 mm	0.1834 mm	2.61 sec
$M = 6$	0.1460 mm	0.2099 mm	4.30 sec
$M = 15$	0.0935 mm	0.1237 mm	9.39 sec
$M = 30$	0.0803 mm	0.0797 mm	18.01 sec

parameter $r_s = R + \Delta R$ (for an arbitrary ΔR):

$$a_{l,s}(r_s = R+\Delta R) = (R+\Delta R)\ddot{q}_{l,s} + g\cos(q_{l,s}) \quad (25)$$

$$= (R+\Delta R)\frac{a_{l,s}+a_{\text{ref}}}{2R} + g\cos(q_{l,s}) \quad (26)$$

Here, since the gravity effect is nonlinear, the parameter estimation error (Fig. 4) is expected to increase as ΔR becomes larger.

Applying the algorithm to these fictitious signals with $M = 15$ for 15 iterations, we can obtain the parameter estimation error for different sensor frame locations as shown in Fig. 4. It is seen that the algorithm achieves less than 1mm parameter estimation error (i.e., mean of $|R-\hat{R}|$ for the last iteration) when we applying it to a sensor frame within $\Delta R = \pm 75$mm. This demonstrates the performance robustness of the proposed algorithm when the

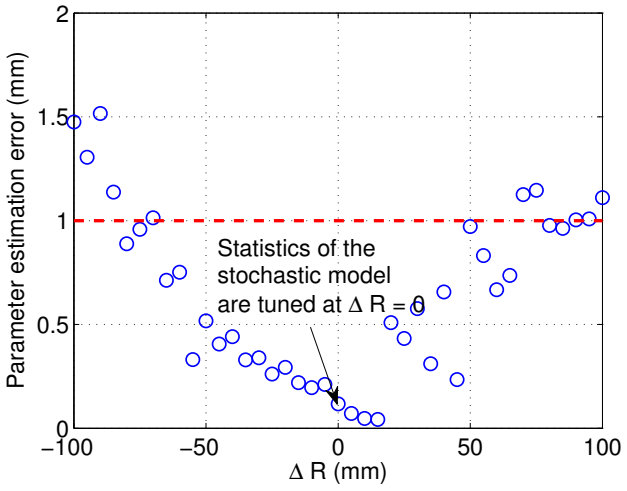

Figure 4: PARAMETER ESTIMATION ERROR FOR DIFFERENT SENSOR FRAME LOCATIONS

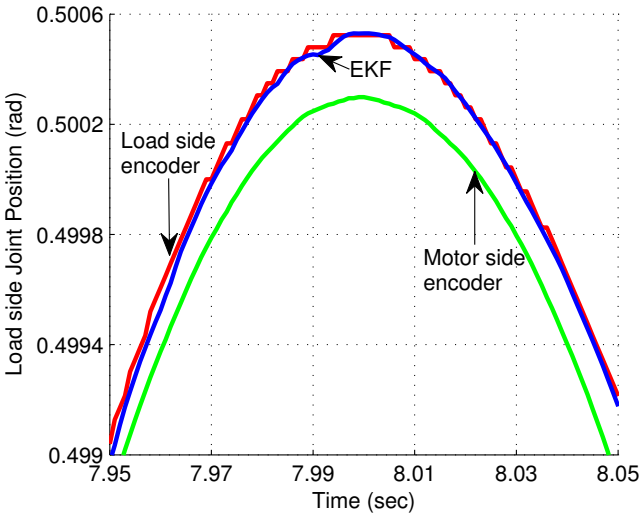

Figure 5: LOAD SIDE POSITION ESTIMATION.

actual sensor frame deviates from the calibrated sensor frame in the training phase.

Load Side State Estimation

As mentioned, the proposed algorithm computes the state estimates simultaneously with the system identification. Figure 5 shows that the estimated states using EKF are much better than the estimated states using the motor side information (i.e., we assume the joint is rigid and thus use $\frac{q_{m,s}}{N}$ as the load side position estimate) only. Here, we evaluate the performance using the true value from the load side encoder measurements. In fact, the EKF estimate is even smoother than the load side encoder measurements, which implies the velocity estimate by EKF would be also much better than the direct differentiation of the load side encoder measurements. The corresponding estimation errors are shown in Fig. 6. It is seen that the root-mean-square (RMS) estimation error of the proposed method is less than half of the one using motor encoder.

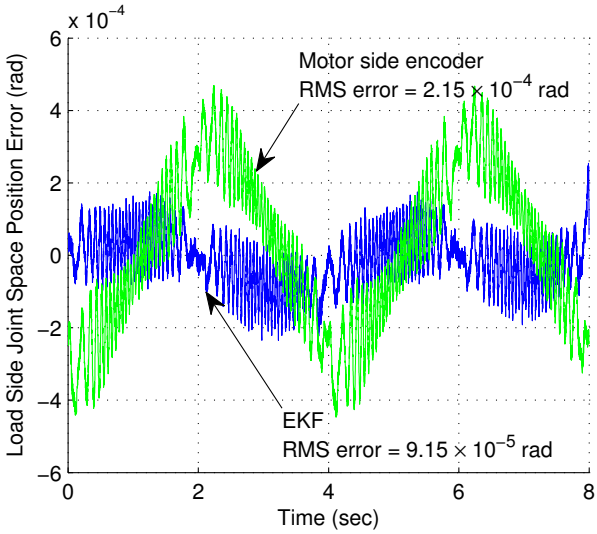

Figure 6: COMPARISON OF LOAD SIDE POSITION ESTIMATION ERROR.

Generalization to Multi-joint Robot

One way of generalization for multi-joint robot is to directly follow the multi-joint robot dynamic model and algorithm derived in this paper. However, the computational complexity is greatly increased due to the model nonlinearities. To simplify the problem, an ad-hoc way can be employed to decouple the problem. In fact, although the experimental validation is done in the single-joint robot test-bed, this result may be representative for a more general case of multi-joint robot. By comparing Fig. 7 to Fig. 2, we can consider a multi-joint robot as a single-joint robot by properly designing the trajectory and the robot posture. To be precise, we select one of the robot joints to be actuated while assuming the remaining fixed robot joints are rigid (e.g., by activating the brakes for these joints). Then, each DH parameter can be identified separately by repeating the same experiment with different end-effector postures.

CONCLUSION

This paper proposed an algorithm that automatically detects an accelerometer's sensor frame (position and orientation) using the measurements from this accelerometer and the motor encoder. A stochastic model was formulated based on the knowl-

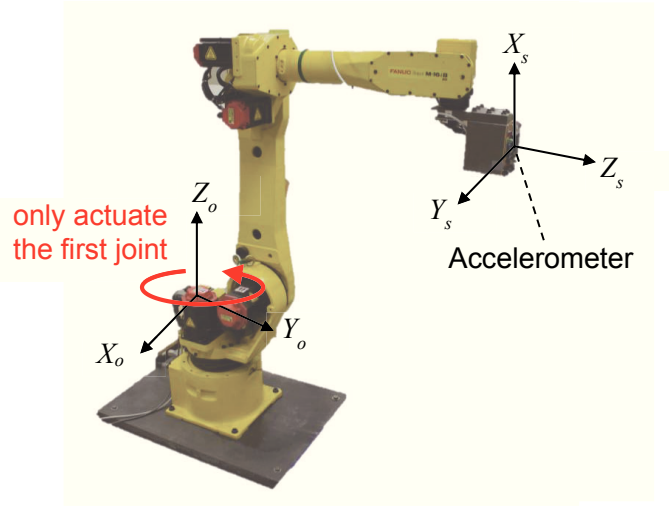

Figure 7: A ROBOT POSTURE FOR SENSOR FRAME IDENTIFICATION IN MULTI-JOINT ROBOT.

edge of the robot dynamics and the measurement model. The closed form of the parameter update law was derived using the Expectation Maximization algorithm with the modification for the nonlinear model. Furthermore, the Monte Carlo integration was utilized to approximate the parameter update law we obtained in the Expectation Maximization algorithm. The effectiveness of the algorithm was experimentally verified. It has been shown that the proposed method is able to yield a good estimate of the unknown sensor frame location even when the actual sensor frame deviates from the calibrated sensor frame in the training phase. Although the experimental validation was done in a single-joint robot test-bed only, the proposed algorithm is formulated for a more general case of multi-joint robot. The experimental validation on a multi-joint robot will be the immediate future work.

REFERENCES

[1] Jeon, S., and Tomizuka, M., 2007. "Benefits of acceleration measurement in velocity estimation and motion control". *Control Engineering Practice,* **15**(3), pp. 325–332.

[2] Henriksson, R., Norrlof, M., Moberg, S., Wernholt, E., and Schon, T., 2009. "Experimental comparison of observers for tool position estimation of industrial robots". In Decision and Control, Proceedings of the 48th IEEE Conference on, IEEE, pp. 8065–8070.

[3] Chen, W., and Tomizuka, M., 2010. "Estimation of load side position in indirect drive robots by sensor fusion and Kalman filtering". In the 2010 American Control Conference (ACC), IEEE, pp. 6852–6857.

[4] Chen, W., and Tomizuka, M., 2012. "Iterative learning control with sensor fusion for robots with mismatched dynamics and mismatched sensing". In Proceedings of the 2012 ASME Dynamic Systems and Control Conference (DSCC), pp. 1480–1488.

[5] Quigley, M., Brewer, R., Soundararaj, S. P., V. Pradeep, Q. L., and Ng, A. Y., 2010. "Low-cost accelerometers for robotic manipulator perception". In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6168–6174.

[6] Dempster, A. P., Laird, N. M., and Rubinr, D. B., 1977. "Maximum likelihood from incomplete data via the EM algorithm". *Journal of the Royal Statistical Society. Series B (Statistical Methodological),* **39**(1), pp. 1–38.

[7] Shumway, R. H., and Stoffer, D. S., 1982. "An approach to time series smoothing and forecasting using the EM algorithm". *Journal of Time Series Analysis,* **3**, pp. 253–264.

[8] Bilmes, J. A., et al., 1998. "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models". *International Computer Science Institute,* **4**(510), p. 126.

[9] Craig, J. J., 2004. *Introduction to Robotics: Mechanics and Control (3rd Edition)*, 3 ed. Prentice Hall.

[10] Robert, C. P., and Casella, G., 2005. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[11] Franklin, G. F., Powell, D. J., and Workman, M. L., 1997. *Digital Control of Dynamic Systems (3rd Edition)*. Prentice Hall, Dec.