Direct Joint Space State Estimation in Robots With Multiple Elastic Joints

Wenjie Chen, *Member, IEEE*, and Masayoshi Tomizuka, *Fellow, IEEE*

***Abstract*—For robots with joint elasticity, discrepancies exist between the motor side and the load side (e.g., the link of the robotic joint). Thus, the load side (end-effector) performance can hardly be guaranteed with motor side measurements alone. In this paper, a computationally efficient load side state estimation scheme is proposed for the multi-joint robot with joint elasticity, which is equipped with motor encoders and a low-cost end-effector MEMS sensor such as a three-axial accelerometer. An optimization-based inverse differential kinematics algorithm is developed to obtain the load side joint state rough estimates. With these rough estimates, the estimation problem is decoupled into simple second-order kinematic Kalman filter for each joint to refine the joint position and velocity estimates. Maximum likelihood principle is utilized to estimate the fictitious noise covariances used in the Kalman filter. Both offline and online solutions are derived. The extensions to other sensor configurations are discussed as well. The effectiveness of the developed method is validated through the simulation and the experimental study on a 6-DOF industrial robot.**

***Index Terms*—Elastic joint, expectation maximization (EM), Kalman filter, state estimation.**

I. Introduction

In robot applications, discrepancies between the available and the desired measurements make it difficult to achieve good control performance. These discrepancies are caused by both sensor and robot dynamics. Particularly, in robots with complex joint dynamics (e.g., flexibilities, friction, etc.), end-effector performance can hardly be guaranteed with motor encoder information alone [1], [2]. This mismatched sensing problem is a critical issue for most practical robot applications where only motor side information is available.

This problem can be tackled by adopting a low-cost MEMS sensor such as accelerometer for robot end-effector sensing. Decoupled joint space position/velocity control, however, is usually preferred in industrial robot control configurations. Thus, the load side joint state estimation from the end-effector sensing would be of the particular interest for this purpose.

In [3], a Kalman filter scheme using either dynamic model or kinematic model was investigated for a single-joint robot with joint elasticity. The scheme fused motor encoder measurements with the load side inertia sensor signals to estimate the load side position. A multi-dimensional kinematic Kalman filter (MD-KKF) was proposed in [4] for multi-joint robot end-effector sensing with application to a 2-DOF planar robot. This scheme was later applied in [5] on a 6-DOF industrial robot, where a position sensitive detector-based camera was developed for robot end-effector position sensing. The MD-KKF scheme utilized several end-effector sensors (i.e., camera, gyroscope, and accelerometer) rather than motor encoders for the estimation of the end-effector state information (especially the velocity state). An adaptive real-time method for end-effector information estimation was also developed in [6] using end-effector position measurement. However, joint space estimation was not directly achieved by these schemes [4]–[6] and the computation/implementation complexity was an issue. In [7] and [8], joint angle estimation was achieved utilizing an accelerometer (and a gyroscope) for each joint without the use of motor encoders. The achieved accuracy was only good for service robots where millimeter-order errors are acceptable. In [9]–[11], the load side state estimation problem was handled with extended Kalman filter (EKF) or particle filter (PF) utilizing both motor encoders and end-effector accelerometer. These schemes were applied to a simplified 2-DOF robot. The computation load, however, was nontrivial due to the complex dynamic/kinematic model and the EKF/PF algorithms. Thus, the methods were only intended for applications where offline computing was feasible, such as the iterative learning control.

In our previous work [12], a sensor fusion scheme, which is computationally efficient and suitable for various applications, was proposed for the multi-joint robots with joint elasticity. This scheme achieved direct load side joint state estimation with limited low-cost sensing configuration (i.e., motor encoders and an end-effector accelerometer). In this paper, this scheme is revisited with more complete details. The theoretical derivation of the parameter estimation scheme for the Kalman filter will be provided. Also more extensive simulation and experimental study will be presented to validate the effectiveness of the proposed scheme. The proposed scheme is also tested for its sensitivity to the model uncertainty.

This paper is organized as follows. In Section II, an optimization-based inverse differential kinematics approach is designed to obtain the joint acceleration estimates. Section III continues with the estimation scheme as decoupled KKF for each joint to refine the estimates for the load side joint position and velocity. Expectation maximization (EM) ([13]–[15]) is utilized to determine the unknown parameters offline and the online solution is also proposed. The computation load and extensions to other sensor configurations are discussed in

Manuscript received May 15, 2012; revised September 16, 2012 and December 7, 2012; accepted February 21, 2013. Date of publication April 16, 2013; date of current version February 20, 2014. Recommended by Technical Editor R. Oboe. This work was supported by FANUC Corporation, Japan.

The authors are with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720 USA (e-mail: wjchen@berkeley.edu; tomizuka@me.berkeley.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2013.2255308

1083-4435 © 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Section IV. Section V presents the simulation and experimental study on a 6-DOF robot. The conclusion is given in Section VI.

II. Robot Inverse Differential Kinematics

A. Dynamic Model for Rough Estimates

Consider a 6-DOF robot manipulator with n elastic joints ($n \geq 6$). The dynamics of this robot with the joint compliance can be expressed as [16]

$$\begin{aligned} M_\ell(q_\ell)\ddot{q}_\ell + C(q_\ell,\dot{q}_\ell)\dot{q}_\ell + G(q_\ell) + D_\ell\dot{q}_\ell + F_{\ell c}\mathrm{sgn}(\dot{q}_\ell) \\ + J^{\mathrm{T}}(q_\ell)f_{\mathrm{ext}} = K_J\left(N^{-1}q_m - q_\ell\right) \\ + D_J\left(N^{-1}\dot{q}_m - \dot{q}_\ell\right) \end{aligned} \quad (1)$$

$$\begin{aligned} M_m\ddot{q}_m + D_m\dot{q}_m + F_{mc}\mathrm{sgn}(\dot{q}_m) = \tau_m \\ - N^{-1}[K_J\left(N^{-1}q_m - q_\ell\right) \\ + D_J\left(N^{-1}\dot{q}_m - \dot{q}_\ell\right)] \end{aligned} \quad (2)$$

where $q_\ell, q_m \in \mathbb{R}^n$ are the load side and the motor side position vectors, respectively. $\tau_m \in \mathbb{R}^n$ is the motor torque vector. $M_\ell(q_\ell) \in \mathbb{R}^{n\times n}$ is the load side inertia matrix, $C(q_\ell, \dot{q}_\ell) \in \mathbb{R}^{n\times n}$ is the Coriolis and centrifugal matrix, and $G(q_\ell) \in \mathbb{R}^n$ is the gravity vector. M_m, K_J, D_J, D_ℓ, D_m, $F_{\ell c}$, F_{mc}, and $N \in \mathbb{R}^{n\times n}$ are all diagonal matrices. The (i,i)-th elements of these matrices represent the motor inertia, reducer stiffness, reducer damping, load side damping, motor side damping, load side Coulomb friction, motor side Coulomb friction, and gear ratio of the i-th joint, respectively. $f_{\mathrm{ext}} \in \mathbb{R}^6$ denotes the external force acting on the robot due to contact with the environment. $J(q_\ell) \in \mathbb{R}^{6\times n}$ is the Jacobian matrix mapping from the load side joint space to the end-effector Cartesian space.

Due to the joint compliance dynamics, discrepancies exist between the motor side joint position q_m and load side joint position q_ℓ (i.e., $q_m \neq q_\ell$ in general). Normally, q_m can be measured by motor side encoders, while q_ℓ, which is of the ultimate interest, is not measurable due to the lack of position sensors at the load side.

With (2), the load side joint position q_ℓ can be roughly estimated as

$$\begin{aligned} \hat{q}_\ell^o = (\hat{D}_J s + \hat{K}_J)^{-1}\Big[\hat{K}_J N^{-1}q_m + \hat{D}_J N^{-1}\dot{q}_m \\ -N\left(\tau_m - \hat{M}_m\hat{\ddot{q}}_m - \hat{D}_m\dot{q}_m - \hat{F}_{mc}\mathrm{sgn}(\dot{q}_m)\right)\Big] \end{aligned} \quad (3)$$

where q_m and $\dot{q}_m$ are obtained from motor encoder measurements, and τ_m can be either motor torque command or measured by motor current. $\hat{\bullet}$ denotes the nominal value of the dynamic parameter $\bullet$. The desired trajectory $\ddot{q}_{md}$ can be used instead of $\hat{\ddot{q}}_m$ in (3) as approximation. Equation (3) is implemented in discrete time by applying Euler forward method. Furthermore, with Euler differentiation of $\hat{q}_\ell^o$, the rough estimate of the load side joint velocity $\hat{\dot{q}}_\ell^o$ is obtained.

Note that the rough estimate $\hat{\dot{q}}_\ell^o$ could be acceptable but noisy. So the load side joint acceleration estimate $\hat{\ddot{q}}_\ell$ cannot be obtained by direct differentiation of $\hat{\dot{q}}_\ell^o$. Also, this rough estimate by (3) is subject to the model uncertainty. Thus, it is necessary to adopt an end-effector sensor such as accelerometer to supplement the lacking information.

B. Basic Differential Kinematics

Let $v_e = [\,\dot{p}_e^{\mathrm{T}} \quad \omega_e^{\mathrm{T}}\,]^{\mathrm{T}} \in \mathbb{R}^6$ denote the end-effector Cartesian velocity vector composed of the translational velocity $\dot{p}_e$ and the angular velocity ω_e at the accelerometer measurement point. The kinematic relation between the joint space and the Cartesian space can be described as

$$v_e = J(q_\ell)\dot{q}_\ell. \quad (4)$$

And the acceleration relationship can be obtained by taking the time derivative of both sides of (4), which gives

$$\dot{v}_e = J(q_\ell)\ddot{q}_\ell + \dot{J}(q_\ell,\dot{q}_\ell)\dot{q}_\ell. \quad (5)$$

Note that the acceleration measured by the end-effector accelerometer is only 3-D translational acceleration. Let $\bar{J}(q_\ell) \in \mathbb{R}^{3\times n}$ and $\bar{\dot{J}}(q_\ell,\dot{q}_\ell) \in \mathbb{R}^{3\times n}$ denote the first three rows of the Jacobian matrix $J(q_\ell)$ and its time derivative $\dot{J}(q_\ell,\dot{q}_\ell)$ respectively. Then, (5) is rewritten as

$$\ddot{p}_e = \bar{J}(q_\ell)\ddot{q}_\ell + \bar{\dot{J}}(q_\ell,\dot{q}_\ell)\dot{q}_\ell. \quad (6)$$

This provides the base to fully retrieve the load side joint acceleration information from the limited-dimensional end-effector measurements.

C. Optimization-Based Inverse Differential Kinematics

Define the pseudoinverse of $\bar{J}(q_\ell)$ as

$$J^\dagger(q_\ell) = \bar{J}(q_\ell)^{\mathrm{T}}\left[\bar{J}(q_\ell)\bar{J}(q_\ell)^{\mathrm{T}}\right]^{-1}. \quad (7)$$

Then, from (6), the load side joint acceleration estimate can be obtained as the following general solutions:

$$\hat{\ddot{q}}_\ell = J^\dagger(q_\ell)\left[\ddot{p}_e - \bar{\dot{J}}(q_\ell,\dot{q}_\ell)\dot{q}_\ell\right] + \left[I - J^\dagger(q_\ell)\bar{J}(q_\ell)\right]\varphi \quad (8)$$

where I is an $n\times n$ identity matrix and $\varphi \in \mathbb{R}^n$ is an arbitrary vector. The term $J^\dagger(q_\ell)\left[\ddot{p}_e - \bar{\dot{J}}(q_\ell,\dot{q}_\ell)\dot{q}_\ell\right] \in \mathrm{Null}^\perp\left(\bar{J}(q_\ell)\right) \equiv \mathrm{Row}\left(\bar{J}(q_\ell)^{\mathrm{T}}\right)$ is the particular solution which minimizes the Euclidean norm of the solution $\|\hat{\ddot{q}}_\ell\|_2$. The term $\left[I - J^\dagger(q_\ell)\bar{J}(q_\ell)\right]\varphi$ is the projection of φ into $\mathrm{Null}\left(\bar{J}(q_\ell)\right)$ and is termed homogeneous solution.

The choice of φ is thus important for selecting an appropriate estimate for the load side joint acceleration. The redundancy of these infinite solutions makes it possible to enforce some practical constraints.

Rewrite (6) as

$$\bar{J}(q_\ell)\ddot{q}_\ell = \ddot{p}_e - \bar{\dot{J}}(q_\ell,\dot{q}_\ell)\dot{q}_\ell \quad \Rightarrow \quad \bar{A}\hat{\ddot{q}}_\ell = \bar{b} \quad (9)$$

which becomes a constraint for the satisfactory load side acceleration estimate $\hat{\ddot{q}}_\ell$. Therefore, the inverse differential kinematics problem can be reformulated as the following standard optimization problem:

$$\min_{\hat{\ddot{q}}_\ell} f(\hat{\ddot{q}}_\ell) \quad \text{s.t. } \bar{A}\hat{\ddot{q}}_\ell = \bar{b} \quad (10)$$

where the imposed physical constraint is expressed as to minimize $f(\hat{\ddot{q}}_\ell)$. This optimization problem and the function f will be detailed in the next section along with the solution.

D. Final Optimization Solution

Note that the Euler differentiation of $\hat{q}_\ell^o$ in (3) gives the velocity rough estimate $\hat{\dot{q}}_\ell^o$. Denote the acceleration rough estimate as $\hat{\ddot{q}}_\ell^o$. This leads to $\hat{\dot{q}}_\ell^o \triangleq \int \hat{\ddot{q}}_\ell^o dt$, which basically means the velocity rough estimate should be consistent with the integration of the acceleration rough estimate. It is understood that this acceleration rough estimate cannot be computed directly by double differentiation of (3) due to the noisy signals. Therefore, instead of using $\hat{\ddot{q}}_\ell^o$ directly, the load side acceleration final estimate $\hat{\ddot{q}}_\ell$ is computed using velocity rough estimate $\hat{\dot{q}}_\ell^o$. This can be done through an optimization problem to minimize the difference between the integration of the acceleration final estimate $\hat{\ddot{q}}_\ell$ and the integration of the rough estimate $\hat{\ddot{q}}_\ell^o$. Such a least-squares optimization problem can be formulated as

$$\min_{\hat{\ddot{q}}_\ell} f(\hat{\ddot{q}}_\ell) = \frac{1}{2}\left\|\hat{\dot{q}}_\ell - \hat{\dot{q}}_\ell^o\right\|_2^2 = \frac{1}{2}\left\|\int \hat{\ddot{q}}_\ell dt - \int \hat{\ddot{q}}_\ell^o dt\right\|_2^2$$
$$\text{s.t. } \bar{A}\hat{\ddot{q}}_\ell = \bar{b} \tag{11}$$

where the equality constraint is to ensure that the resulting estimate will match with the end-effector measurement. This optimization problem over the whole time series would be impractical to solve especially within the real-time environment. Instead, a pointwise optimization can be performed for each time step to generate the suboptimal solution. For each time step t_k, let

$$\hat{\dot{q}}_{\ell,k}^o = \int_0^{t_k} \hat{\ddot{q}}_\ell^o(t)dt, \quad \hat{\dot{q}}_{\ell,k} = \sum_{i=0}^{k} \hat{\ddot{q}}_{\ell,i}\Delta t \tag{12}$$

where the subscript k denotes the time index and Δt is the sampling time. Then by denoting $\Delta\hat{\ddot{q}}_{\ell,k} \triangleq \frac{\hat{\dot{q}}_{\ell,k}^o - \hat{\dot{q}}_{\ell,k-1}}{\Delta t}$, (11) can be relaxed to a convex optimization problem for each time step t_k as

$$\min_{\hat{\ddot{q}}_{\ell,k}} f(\hat{\ddot{q}}_{\ell,k}) = \frac{1}{2}\left\|\hat{\ddot{q}}_{\ell,k} - \Delta\hat{\ddot{q}}_{\ell,k}\right\|_2^2$$
$$\text{s.t. } A_k\hat{\ddot{q}}_{\ell,k} = b_k \tag{13}$$

where

$$A_k = \bar{J}(q_{\ell,k}), \quad b_k = \ddot{p}_{e,k} - \bar{\dot{J}}(q_{\ell,k}, \dot{q}_{\ell,k})\dot{q}_{\ell,k}. \tag{14}$$

Note that $\Delta\hat{\ddot{q}}_{\ell,k}$ also includes the accumulated acceleration estimation error not compensated from previous steps. The resulting problem has the global optimal closed form solution (i.e., load side joint acceleration estimate) as

$$\hat{\ddot{q}}_{\ell,k} = A_k^{\mathrm{T}}\left(A_kA_k^{\mathrm{T}}\right)^{-1}b_k + \left[I - A_k^{\mathrm{T}}\left(A_kA_k^{\mathrm{T}}\right)^{-1}A_k\right]\Delta\hat{\ddot{q}}_{\ell,k} \tag{15}$$

which is in the form of the general solution (8).

E. Practical Implementation Issues

In practice, the acceleration measurement f_a provided by the end-effector accelerometer is the translational acceleration with additional gravity effect expressed in the accelerometer coordinate frame. Thus, the end-effector translational acceleration $\ddot{p}_e$ in the world coordinates can be obtained as

$$\ddot{p}_e = R_a(q_\ell)f_a - g_s \tag{16}$$

where $R_a(q_\ell)$ is the rotation matrix of the accelerometer coordinate frame with respect to the world coordinate frame and $g_s = [0 \quad 0 \quad 9.8]^{\mathrm{T}}\,\mathrm{m/s}^2$ is the measured gravity vector expressed in the world coordinate frame.

Furthermore, since the measurements of q_ℓ and $\dot{q}_\ell$ are generally not available, the rough estimates $\hat{q}_\ell^o$ and $\hat{\dot{q}}_\ell^o$ are used instead in (14) and (16) to calculate $\bar{J}(q_\ell)$, $\bar{\dot{J}}(q_\ell, \dot{q}_\ell)\dot{q}_\ell$, and $R_a(q_\ell)$. These adjustments for the calculation are reasonable under the fact that the tiny discrepancies between the actual motion and the rough estimates normally do not make much difference in the Jacobian matrices and the orientation matrix. Thus, the resulting acceleration $\ddot{p}_e$ is still fairly accurate even in the presence of gravity effect.

III. Kinematic Kalman Filter

A. Decoupled KKF

As discussed earlier, the load side rough approximations have been obtained as $\hat{q}_\ell^o$ in (3) and $\hat{\ddot{q}}_\ell$ in (15) for each joint. Thus, the estimation problem for the whole robot can be decoupled into n KKF running in parallel, which are computationally simple, to better estimate the load side joint position and velocity. The discrete time kinematic model for the Kalman filter is written as

$$\underbrace{\begin{bmatrix} q_{\ell,k+1} \\ \dot{q}_{\ell,k+1} \end{bmatrix}}_{x_{k+1}} = \underbrace{\begin{bmatrix} I & \Delta tI \\ \mathbf{0} & I \end{bmatrix}}_{A}\underbrace{\begin{bmatrix} q_{\ell,k} \\ \dot{q}_{\ell,k} \end{bmatrix}}_{x_k} + \underbrace{\begin{bmatrix} \frac{1}{2}\Delta t^2 I \\ \Delta tI \end{bmatrix}}_{B}\underbrace{\hat{\ddot{q}}_{\ell,k}}_{u_k} + w_k \tag{17a}$$

$$\underbrace{\hat{q}_{\ell,k}^o}_{y_k} = \underbrace{[I \quad \mathbf{0}]}_{C}\underbrace{\begin{bmatrix} q_{\ell,k} \\ \dot{q}_{\ell,k} \end{bmatrix}}_{x_k} + v_k \tag{17b}$$

which is in the following standard form:

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{18a}$$
$$y_k = Cx_k + v_k \tag{18b}$$

with the assumption that $1 \le k \le T$, $x_1 \sim X_1 = \mathcal{N}(\hat{x}_1, P_1)$, $w_k \sim W_k = \mathcal{N}(\mathbf{0}, Q)$, and $v_k \sim V_k = \mathcal{N}(\mathbf{0}, R)$, where w_k and v_k are fictitious noises.

In the offline case, the Kalman smoother using the following forward recursion and backward recursion procedures [14] can be applied to estimate the state.

1) Forward recursion

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_{k-1} \tag{19a}$$
$$P_{k|k-1} = AP_{k-1|k-1}A^{\mathrm{T}} + \hat{Q} \tag{19b}$$
$$K_k = P_{k|k-1}C^{\mathrm{T}}(CP_{k|k-1}C^{\mathrm{T}} + \hat{R})^{-1} \tag{19c}$$

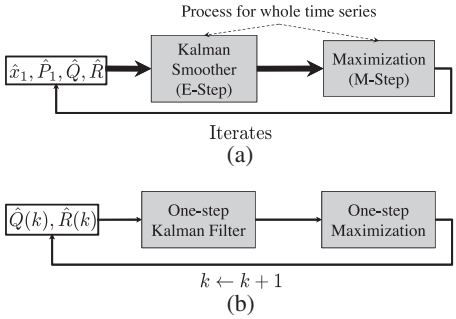

Fig. 1. Adaptive kinematic Kalman filter process. (a) Ofline estimation structure. (b) Online estimation structure.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1}) \quad (19d)$$

$$P_{k|k} = P_{k|k-1} - K_k C P_{k|k-1} \quad (19e)$$

$$P_{(k,k-1)|k} = (I - K_k C)AP_{k-1|k-1}. \quad (19f)$$

2) Backward recursion

$$L_{k-1} = P_{k-1|k-1}A^{\mathrm{T}}P_{k|k-1}^{-1} \quad (20a)$$

$$\hat{x}_{k-1|T} = \hat{x}_{k-1|k-1} + L_{k-1}(\hat{x}_{k|T} - \hat{x}_{k|k-1}) \quad (20b)$$

$$P_{k|T} = P_{k|k} + L_k(P_{k+1|T} - P_{k+1|k})L_k^{\mathrm{T}} \quad (20c)$$

$$P_{(k,k-1)|T} = P_{k|T}P_{k|k}^{-1}P_{(k,k-1)|k} \quad (20d)$$

where $\hat{\bullet}_{k|j}$ represents the conditional expectation of $\bullet_k$ given the information up to the j-th time step. $\hat{Q}$ and $\hat{R}$ are the estimates of Q and R. The online version of the Kalman filter is the causal forward recursion part (19) only. Note that, $P_{k|j}$ denotes the error covariance of $\hat{x}_{k|j}$, while $P_{(k,k-1)|j}$ denotes the cross error covariance of $\hat{x}_{k|j}$ and $\hat{x}_{k-1|j}$, i.e.,

$$P_{k|j} = E\left[\left(x_k - \hat{x}_{k|j}\right)\left(x_k - \hat{x}_{k|j}\right)^{\mathrm{T}}\right] \quad (21)$$

$$P_{(k,k-1)|j} = E\left[\left(x_k - \hat{x}_{k|j}\right)\left(x_{k-1} - \hat{x}_{k-1|j}\right)^{\mathrm{T}}\right]. \quad (22)$$

Recall that $\hat{q}^o_{\ell,k}$ and $\hat{\dot{q}}_{\ell,k}$ are only approximations instead of direct measurements. Thus, to implement this KKF, it is critical to determine the appropriate covariances for the fictitious noises w_k and v_k. This means $\hat{x}_1, P_1, Q$, and R are the unknowns to be estimated first, which is detailed in the following section.

B. Parameter Estimation

These parameters can be adapted based on the maximum likelihood principle [13], [14], [17]. The derivation of the following estimation solutions for this specific problem is detailed in the Appendix. The basic structure of the adaptive KKF procedure is illustrated in Fig. 1.

1) Offline Estimation [see Fig. 1(a)]: If offline processing is available, which is applicable in iterative applications, the whole time series data can be accessed. In this case, EM algorithm ([13]–[15]) can be applied as follows.

1) E-*step:* run Kalman smoother (19) and (20) with current best estimates of $\hat{x}_1, P_1, Q$, and R.

2) M-*step:* update $\hat{x}_1, P_1, Q$, and R as in (23) using the acausal estimates from Kalman smoother

$$\hat{x}_1 = \hat{x}_{1|T} \qquad \hat{P}_1 = P_{1|T} \quad (23a)$$

$$\hat{Q} = \frac{1}{T-1}\sum_{k=2}^{T}\Big[\left(\hat{x}_{k|T} - A\hat{x}_{k-1|T} - Bu_{k-1}\right) \cdot \left(\hat{x}_{k|T} - A\hat{x}_{k-1|T} - Bu_{k-1}\right)^{\mathrm{T}} + P_{k|T} - AP^{\mathrm{T}}_{(k,k-1)|T} - P_{(k,k-1)|T}A^{\mathrm{T}} + AP_{k-1|T}A^{\mathrm{T}}\Big] \quad (23b)$$

$$\hat{R} = \frac{1}{T}\sum_{k=1}^{T}[\left(y_k - C\hat{x}_{k|T}\right)\left(y_k - C\hat{x}_{k|T}\right)^{\mathrm{T}} + CP_{k|T}C^{\mathrm{T}}]. \quad (23c)$$

3) Iterate from E-*step* until the increment of the expected likelihood is within chosen threshold.

Remark 1: The initial estimates for $\hat{x}_1, P_1, Q$, and R can be computed by robot home position with zero velocity and covariances of $\hat{q}^o_{\ell,k}$ and $\hat{\dot{q}}_{\ell,k}$ during the initial static period. Note that the covariance of the fictitious noises adapted by EM may be different depending on the workspace as well as the trajectory characteristics (e.g., velocity and acceleration). However, the major components of the fictitious noises normally come from the characteristics of the signals (e.g., accelerometer and encoder signals). Also, mostly one typical motion trajectory does not span the entire workspace, and thus the covariances along this trajectory would not vary significantly. Therefore, it is reasonable to offline optimize the covariances for the whole trajectory. But it should be understood that the optimized covariances are only "optimal" for this particular trajectory rather than the entire workspace.

2) Online Estimation [see Fig. 1(b)]: If real-time computing is desired, only causal estimation from forward recursion (19) (i.e., Kalman filter) can be used. Furthermore, instead of estimating using the whole time series as in (23), here the covariances Q and R are adapted for each time step as

$$\hat{Q}^o_{k+1} = \left(\hat{x}_{k|k} - A\hat{x}_{k-1|k-1} - Bu_{k-1}\right) \cdot \left(\hat{x}_{k|k} - A\hat{x}_{k-1|k-1} - Bu_{k-1}\right)^{\mathrm{T}} + P_{k|k} - AP^{\mathrm{T}}_{(k,k-1)|k} - P_{(k,k-1)|k}A^{\mathrm{T}} + AP_{k-1|k-1}A^{\mathrm{T}} \quad (24a)$$

$$\hat{R}^o_{k+1} = \left(y_k - C\hat{x}_{k|k}\right)\left(y_k - C\hat{x}_{k|k}\right)^{\mathrm{T}} + CP_{k|k}C^{\mathrm{T}}. \quad (24b)$$

In practice, to avoid drastic change to the covariances, exponential moving average could be applied to control the adaptive rate for smooth estimation. This is done as

$$\hat{Q}_{k+1} = \left(1 - \frac{1}{N_Q}\right)\hat{Q}_k + \frac{1}{N_Q}\hat{Q}^o_{k+1} \quad (25a)$$

$$\hat{R}_{k+1} = \left(1 - \frac{1}{N_R}\right)\hat{R}_k + \frac{1}{N_R}\hat{R}^o_{k+1} \quad (25b)$$

where N_Q and N_R are the window sizes for the moving average filters. $\hat{Q}_k$ and $\hat{R}_k$ are the estimated covariance matrices actually

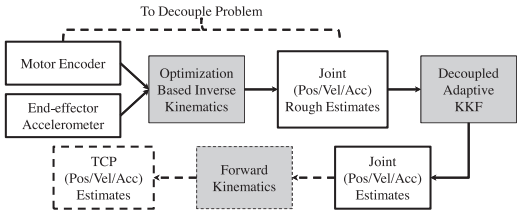

Fig. 2. The structure of load side state estimation approach.

utilized in the online Kalman filter. Also, note that, the initial conditions $\hat{x}_1$ and P_1 cannot be adapted in this real-time case.

IV. Discussion of the Approach

A. Computation Load

One of the great advantages of the proposed method over others is its light computation load. As shown in Fig. 2, this approach mainly consists of two stages, optimization-based inverse differential kinematics and KKF with parameter estimation. The forward kinematics stage is additional if the tool center point (TCP) information is desired. Due to the decoupling feature of this method, the computation requirement[1] is significantly low compared to the EKF scheme proposed in [9] and [11], which can only achieve about 200 Hz sampling rate (it was also reported that 4.9 s was required to estimate 1s of data) for a simplified 2-DOF robot and will drastically increase the computation demand for a higher DOF robot. The PF scheme was even much more computationally demanding as reported in [11] due to the large set of samples required for probability approximation. In our experimental setup (e.g., MATLAB environment on a regular PC with a 2.53 GHz dual-core CPU[2]), however, our real-time scheme can easily achieve a 1 kHz sampling rate for a 6-DOF industrial robot, which actually requires only 1.787 s computation time to estimate for a 5.512 s trajectory. Therefore, the overall computation load for the proposed approach is sufficiently light for both online computing and offline processing with limited onboard industrial computation power.

B. Extensions to Other Sensor Configurations

The developed approach is designed for the case where motor encoders and the end-effector accelerometer are available. It should be noted, however, that the extensions to other sensor configurations can also be easily derived.

[1]At the inverse differential kinematics stage, the main computation lies in (3), which is diagonal matrix calculation due to the property of motor side model, and (15), where a 3×3 matrix inversion requires the most effort. The calculations of (14) and (16) are basically 3×1 vector operations and 3×3 matrix multiplications. After this stage, the problem becomes decoupled Kalman filter (or smoother) for each joint with the kinematic model of only 2 states, 1 input, and 1 output. The matrix inversion becomes scalar inversion for Kalman filter case and 2×2 matrix inversion for smoother case, both of which are computationally simple. In the parameter estimation, only 2×2 matrix multiplications and scalar operations are required. The optional forward kinematics stage, which consists of n 4×4 matrix multiplications, can also be efficiently processed.

[2]The computation time may be even less if we further customize the algorithm and exploit the parallel computing power from the dual-core CPU.

If motor encoders and the end-effector gyroscope are available, the optimization problem (11) in the inverse differential kinematics stage can be modified to obtain the load side joint velocity estimate as

$$\min_{\dot{\hat{q}}_\ell} f(\dot{\hat{q}}_\ell) = \frac{1}{2}\left\|\dot{\hat{q}}_\ell - \dot{\hat{q}}_\ell^o\right\|_2^2 \qquad \text{s.t. } \underline{J}(\hat{q}_\ell^o)\dot{\hat{q}}_\ell = \omega_e \tag{26}$$

where $\underline{J}(\hat{q}_\ell^o) \in \mathbb{R}^{3\times n}$ denotes the last three rows of $J(\hat{q}_\ell^o)$. The Kalman filter (smoother) reduces to the one with a first-order kinematic model

$$q_{\ell,k+1} = q_{\ell,k} + \Delta t \dot{\hat{q}}_{\ell,k} + w_k, \quad \hat{q}_{\ell,k}^o = q_{\ell,k} + v_k. \tag{27}$$

Note that only q_ℓ and $\dot{q}_\ell$ can be estimated due to the lack of accelerometer. However, if the rotational vibration is the motion of interest to observe, this approach could be suitable.

Similarly, if the robot is only equipped with motor encoders and the end-effector position sensor such as camera, the inverse differential kinematics stage (15) becomes the case to obtain the load side joint position estimate as

$$\begin{aligned} &\min_{\hat{q}_\ell} f(\hat{q}_\ell) = \frac{1}{2}\left\|\hat{q}_\ell - \hat{q}_\ell^o\right\|_2^2 \\ &\text{s.t. } p_e - p_o = \bar{J}(\hat{q}_\ell^o)\left(\hat{q}_\ell - \hat{q}_\ell^o\right) \end{aligned} \tag{28}$$

where p_e is the measured TCP position. $p_o \triangleq \text{fk}(\hat{q}_\ell^o)$ and $\text{fk}(\hat{q}_\ell^o)$ represents the forward kinematic function to compute the TCP position corresponding to the load side joint angles $\hat{q}_\ell^o$. The particular solution to minimize the norm $\|\hat{q}_\ell - \hat{q}_\ell^o\|_2$ is thus derived as

$$\begin{aligned} &\hat{q}_\ell - \hat{q}_\ell^o = J^\dagger(\hat{q}_\ell^o)(p_e - p_o) \\ \Rightarrow \quad &\hat{q}_\ell = J^\dagger(\hat{q}_\ell^o)(p_e - \text{fk}(\hat{q}_\ell^o)) + \hat{q}_\ell^o. \end{aligned} \tag{29}$$

Note that this problem (28) is formulated based on the assumption that the initial rough estimate $\hat{q}_\ell^o$ is close to the desired estimate $\hat{q}_\ell$. Thus, this stage needs to be iterated with newly updated $\hat{q}_\ell^o \leftarrow \hat{q}_\ell$ until the solution converges. The Kalman filter (smoother) stage becomes unnecessary and the final approach is only suitable for estimating the load side joint position q_ℓ.

Remark 2: It is important to note that the formulated optimization problems [(11), (26), and (28)] are not only to minimize the differences from the rough estimates (by least-squares cost functions), but also to match with the end-effector measurements (by enforcing the equality constraints). In other words, these estimation schemes have taken into consideration both the motor encoder measurements and the model knowledge (for the rough estimates) as well as the end-effector measurements (for the final estimates). Therefore, the resulting load side state estimates should be reasonably satisfactory considering the limited measurements and model knowledge available.

V. Simulation and Experimental Study

A. Test Setup

The proposed methods are implemented on a six-joint industrial robot, FANUC M-16*i*B/20 [18], as shown in Fig. 3. The robot is equipped with built-in motor encoders for each joint. An inertia sensor (Analog Devices, ADIS16400) consisting of

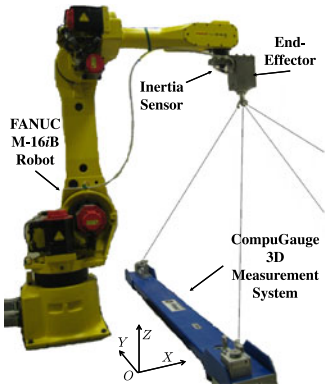

Fig. 3. FANUC M-16*i*B robot system.

a three-axial accelerometer and a three-axial gyroscope is attached to the end-effector. The three-dimensional position measurement system, CompuGauge 3D (accuracy of ± 0.15 mm, resolution of 0.01 mm), is utilized to measure the end-effector TCP position as the ground truth for performance validation. In the following simulation and experimental study, only motor encoders and the end-effector accelerometer are utilized for the load side state estimation (i.e., the gyroscope and the CompuGauge 3D are not utilized for sensor fusion). The sampling rates of all the sensor signals as well as the real-time controller implemented through MATLAB xPC Target are set to be 1 kHz.

Note that, since the proposed scheme is aimed for direct load side joint state estimation, the performance validation in the joint space is desired. Due to the unavailability of load side sensors at the joints, this joint space validation is not possible with the experimental setup. Thus, the simulation study using a high-fidelity robot simulator, which can provide simulated load side true quantities, is adopted here for joint space validation. This high-fidelity robot simulator is designed based on MATLAB Simulink and SimMechanics Toolbox (i.e., by multi-physical-body modeling instead of mathematical model formulation) using robot dynamic and kinematic parameters as well as the calibrated sensor parameters. On the other hand, for experimental study, we utilize the CompuGauge 3D and the accelerometer measurements for the Cartesian space performance validation of the forward kinematics results of the load side state estimates.

B. Algorithm Settings

The testing TCP trajectory (see Fig. 4) is a 10 cm $\times$ 10 cm square path on the *Y–Z* plane with fixed orientation and maximum speed of 1 m/s. For this motion, all joints except Joint 4 need to be moved. The estimation algorithm settings for comparisons are listed as follows.

1) *KKF-Offline*: Offline estimation with EM and Kalman smoother. A 30 Hz zero-phase low-pass filter is applied to the raw accelerometer measurements f_a and the rough estimate $\hat{\ddot{q}}_\ell^o$.
2) *KKF-Online*: Online estimation with Kalman filter forward recursion only and online covariance updating using window sizes $N_Q = N_R = 500$. A 100 Hz causal low-pass filter is applied to the raw signals. Note that this low-pass filter will introduce some phase delay to the signals and thus the bandwidth is chosen to be 100 Hz as a tradeoff between the phase delay effect and high-frequency noise filtering.
3) *InvKine*: Using the rough estimates (i.e., $\hat{q}_\ell^o$, $\hat{\dot{q}}_\ell^o$, and $\hat{\ddot{q}}_\ell$) from the inverse differential kinematics stage (see Section II) and their forward kinematics results.
4) *CG3D*: Real measurement and its differentiation from CompuGauge 3-D system.
5) *Motor*: Using motor side information (i.e., simulated motor side position/velocity/acceleration in simulation, or motor encoder signals in experiment) directly as load side information and calculating the TCP information by forward kinematics using motor side information.

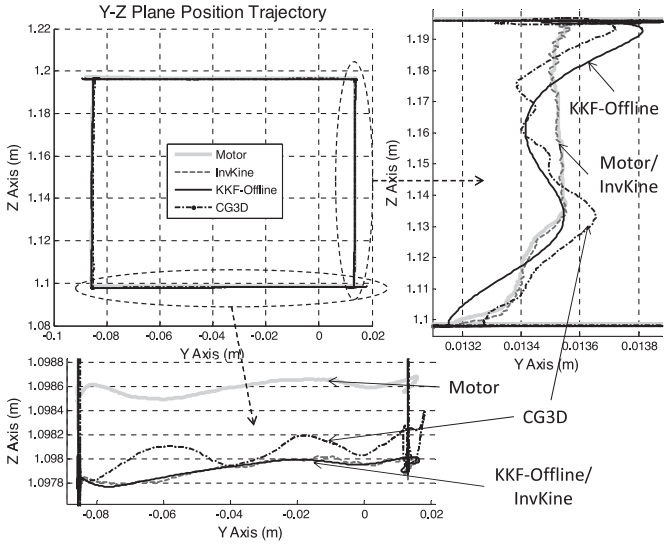

Fig. 4. *Y–Z* plane TCP position estimation (experiment). *KKF-Online* results are similar to those of *KKF-Offline* and thus not shown here for simplicity.

C. Simulation Results

The load side joint estimation errors calculated from the simulated load side information are plotted using absolute values in Figs. 5 and 6. It is shown that the proposed *KKF* schemes outperform the *Motor* setting significantly. Particularly in Fig. 5, for Joints 2, 3, and 5, where gravity effects are evident, the position estimation by *Motor* suffers from noticeable offset error, while the *KKF* schemes successfully account for the gravity effects. The *KKF* schemes also show smaller peak errors for the motion periods as illustrated in Joints 1, 4, and 6. Fig. 6 plots the velocity and the acceleration estimates for the two joints (i.e., Joints 3 and 6) with the largest ranges of motion. It shows that, the online velocity and acceleration estimates from *KKF-Online* are not clean due to the high-bandwidth low-pass filter applied to the raw signals, while *KKF-Offline* provides the best estimates among the all. Even though for some particular instances, *KKF-Online* may perform worse than *Motor*, which uses simulated motor side quantities including velocity and acceleration, it is still beneficial to use *KKF-Online* because in reality motor

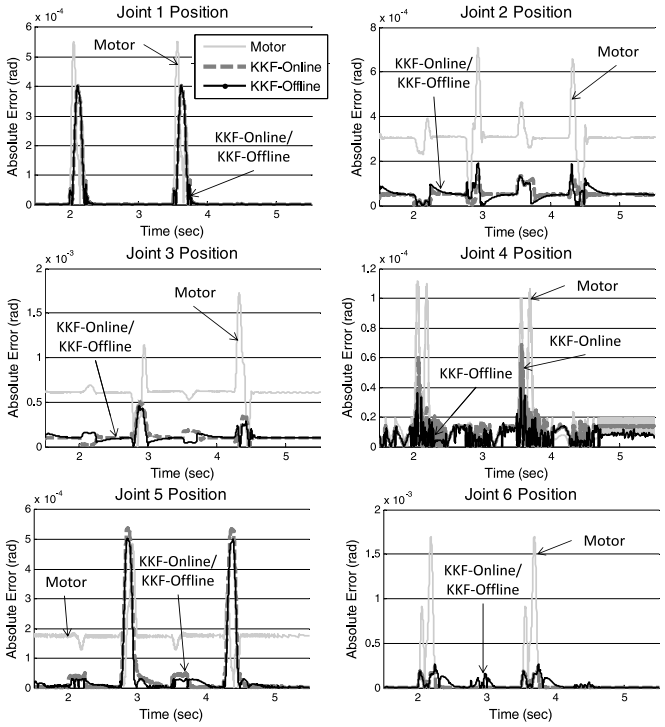

Fig. 5. Load side joint position estimation absolute error (simulation).

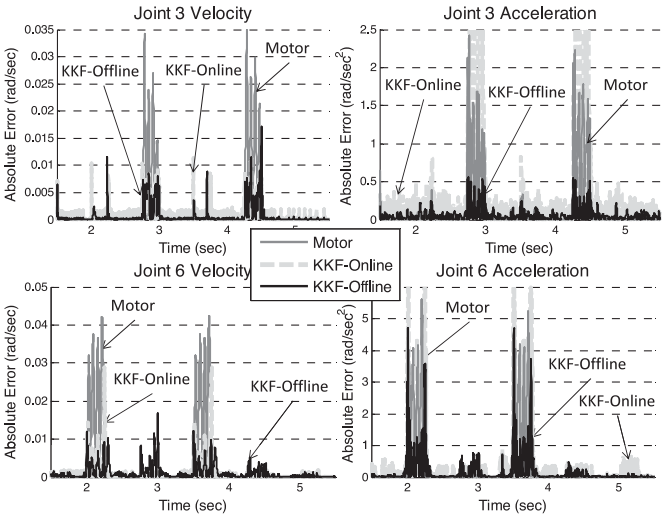

Fig. 6. Load side joint velocity/acceleration estimation absolute error (simulation).

encoders do not provide velocity/acceleration measurements and the direct differentiation of encoder signals would be highly noisy.

D. Experimental Results

The methods are also implemented on the actual experimental setup for Cartesian space comparisons. The load side joint state estimates are used in the forward kinematics to obtain the Cartesian space estimates for comparisons. Fig. 4 shows the estimated position trajectory on the *Y–Z* plane. Again, it is clearly seen that the *KKF-Offline* setting (*KKF-Online* results are similar to *KKF-Offline* except that *KKF-Online* results are a little noisier.

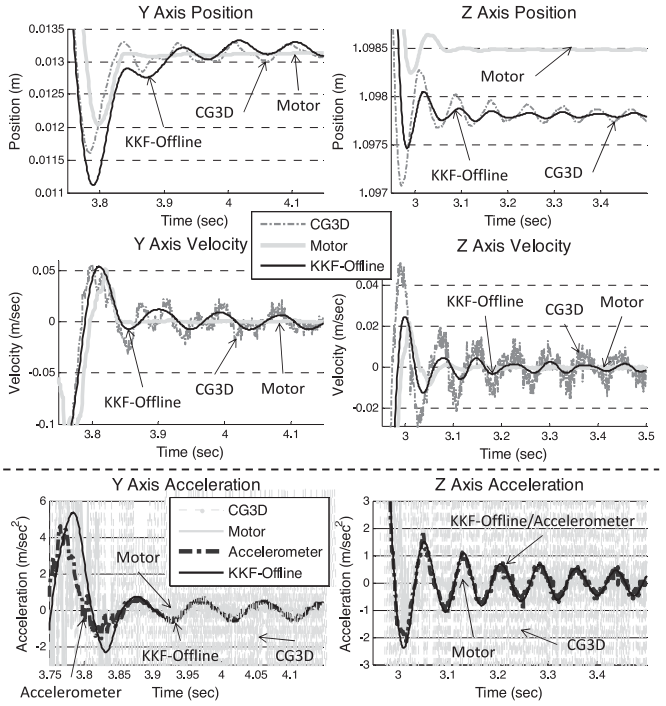

Fig. 7. TCP estimation on *Y*- and *Z*-axes when coming to a stop (experiment). *KKF-Online* results (not shown here for simplicity) are similar to those of *KKF-Offline* except that *KKF-Online* results are a little noisier.

TABLE I
TCP ESTIMATION ERRORS WHEN COMING TO A STOP (EXPERIMENT)

RMS Errors	Pos. (mm)	Vel. (mm/sec)	Acc. (mm/sec^2)
Motor	0.737	56.348	4034.607
KKF-Offline	0.306	25.363	1367.058
KKF-Online	0.409	31.511	2273.706
InvKine	0.410	75.389	2274.250
KKF-Offline-Unc	0.390	26.848	1352.084
KKF-Online-Unc	0.498	32.310	2239.252
InvKine-Unc	0.498	76.860	2239.619

●-*Unc* denotes the case with model parametric uncertainty.

The superior performance of the proposed scheme can be better appreciated for the residual vibration sensing when the robot comes to a stop as shown in Figs. 7 and 8. In general, the *Motor* setting cannot capture the vibratory motion at the

[3]Note that the oscillation on the *Z*-axis not captured by the estimates has the magnitude of about ± 0.1 mm, which is within the CompuGauge accuracy level (± 0.15 mm). This implies the high possibility that this mismatch (within the accuracy level) is due to CompuGauge's own measurement characteristics. Other than this, the estimates by the proposed scheme capture the robot motions (CompuGauge and accelerometer measurements) quite successfully.

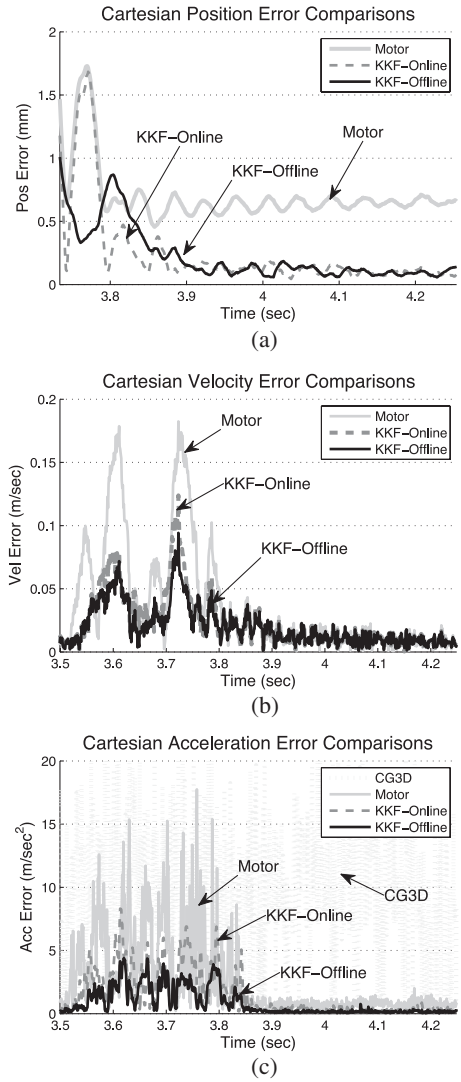

Fig. 8. TCP Estimation error when coming to a stop (experiment). (a) TCP position estimation error . (b) TCP velocity estimation error . (c) TCP acceleration estimation error.

end-effector and the resulting TCP acceleration estimation is very noisy, while the proposed *KKF* schemes are able to do both very well with the fusion of end-effector accelerometer measurements. Recall that, the CompuGauge measurement and its differentiation (*CG3D*) are used as the ground truth for position and velocity estimation performance evaluation. In the acceleration comparison, however, the real accelerometer measurement (*Accelerometer*) is used instead of the noisy *CG3D* data differentiation as the ground truth. It is shown in Fig. 7 that the proposed *KKF* estimates closely follow the *CG3D* and the *Accelerometer* measurements. In particular, during the stopping period (i.e., after 3.8 s for *Y*-axis and after 3 s for *Z*-axis), the *Motor* estimates look static. The actual residual motion, however, is vibratory and can be successfully captured by the *KKF* estimates.

Another benefit of the proposed *KKF* schemes is that the velocity/acceleration estimates are much cleaner than the *CG3D* setting which is obtained by direct differentiation from the position measurement, even though there are also some minor noises present in the *KKF-Online* estimates (see Fig. 7; Table I).

The estimation errors[4] for end-effector TCP position/velocity/acceleration when the robot is coming to a stop are illustrated in Fig. 8. The earlier conclusions are confirmed again in this figure. All the *KKF* settings perform much better than the *Motor* setting by capturing the residual vibratory motion (see Fig. 8). And *KKF-Offline* performs the best due to its acausal processing availability. These conclusions can also be drawn from the rms estimation errors for position/velocity/acceleration as listed in Table I, which shows that the rms estimation errors can be reduced by the proposed *KKF* schemes to about half or even less of that of the *Motor* estimates.

E. Sensitivity to Model Uncertainty

Note that the proposed scheme utilizes the dynamic model as in (3), where the joint stiffness and damping coefficients (i.e., K_J and D_J) are particularly important. The nominal dynamic parameters can be properly identified through system identifications for each individual joint at several different postures. However, in the case that system identifications are not properly conducted or not available for the particular robot, the robustness of the estimation scheme should be examined against the model uncertainty. Table I lists the estimation errors by the settings (denoted as $\bullet$-*Unc*) that utilizes twice the nominal values for the joint stiffness and damping coefficients. It is seen that the position estimation performance by these settings is worsen by the modeling errors compared to the settings using nominal values, but is still significantly better than the one by *Motor* (which can be regarded as the infinite stiffness/damping case). The velocity and the acceleration estimates, however, are robust to these parametric errors and would always be better than *Motor* (no matter how stiffness/damping changes), due to the utilization of the end-effector accelerometer signals and the refinement in the KKF stage.

VI. Conclusion

This paper investigated the direct load side state estimation problem for the robots with mismatched sensing, i.e., robots with joint elasticity. The problem was tackled using the low-cost sensor configuration, i.e., motor encoders and an end-effector accelerometer. The direct joint space estimation was achieved which is suitable for decentralized joint control. With the equipped end-effector accelerometer, the load side joint acceleration estimate was obtained through an optimization-based inverse differential kinematics algorithm. Then the problem was decoupled into n simple KKFs to estimate the load side joint position and velocity. Offline and online solutions were

[4]This Cartesian space error is defined as the Euclidean distance between the estimated position/velocity/acceleration and the actual ones.

presented for the fictitious noise covariance determination. The proposed approach is computationally efficient for both offline applications and online computing. Simulation and experimental study on a 6-DOF industrial robot demonstrated the superior performance of the developed method and the advantage of the end-effector sensing. As the related work, this scheme has been successfully applied to the offline iterative learning control [19], while the online application to use the estimation for real-time feedback control has also been confirmed and the publication is under preparation.

APPENDIX

The later derivation is based on [13]–[15] with the extension to include the input Bu_k in the model (18). Given the input series $\mathbf{U} = \{u_1, \ldots, u_T\}$ and the output series $\mathbf{Y} = \{y_1, \ldots, y_T\}$, the objective is to maximize the conditional joint log likelihood of $\hat{x}_1, P_1, A, B, C, Q$, and R, which can be derived as

$$\begin{aligned} G(\hat{x}_1, P_1, A, B, C, Q, R|\mathbf{U}, \mathbf{Y}) &= \text{Constant} - \frac{T-1}{2}\log|Q| \\ &- \sum_{k=2}^{T}\frac{1}{2}(x_k - Ax_{k-1} - Bu_{k-1})^{\mathrm{T}} Q^{-1}(x_k - Ax_{k-1} - Bu_{k-1}) \\ &- \sum_{k=1}^{T}\frac{1}{2}(y_k - Cx_k)^{\mathrm{T}} R^{-1}(y_k - Cx_k) - \frac{T}{2}\log|R| \\ &- \frac{1}{2}\log|P_1| - \frac{1}{2}(x_1 - \hat{x}_1)^{\mathrm{T}} P_1^{-1}(x_1 - \hat{x}_1). \end{aligned} \tag{30}$$

Since the actual state distributions are unknown, the *conditional expected* likelihood[5] $E[G(\bullet)|\mathbf{U}, \mathbf{Y}]$ is used instead to perform maximization (M-*step*). With first-order condition $\frac{\partial E[G(\bullet)]}{\partial(\bullet)} = 0$ and some matrix trace and calculus manipulations [13], the resulting estimates can be derived as

$$\hat{x}_1 = E[x_1] \qquad \hat{P}_1 = E\left[(x_1 - \hat{x}_1^o)(x_1 - \hat{x}_1^o)^{\mathrm{T}}\right]$$

$$\hat{A} = \left(\sum_{k=2}^{T} E[x_k x_{k-1}^{\mathrm{T}} - \hat{B}^o u_{k-1} x_{k-1}^{\mathrm{T}}]\right)\left(\sum_{k=2}^{T} E\left[x_{k-1}x_{k-1}^{\mathrm{T}}\right]\right)^{-1}$$

$$\hat{B} = \left(\sum_{k=2}^{T} E[x_k u_{k-1}^{\mathrm{T}} - \hat{A}^o x_{k-1} u_{k-1}^{\mathrm{T}}]\right)\left(\sum_{k=2}^{T} E\left[u_{k-1}u_{k-1}^{\mathrm{T}}\right]\right)^{-1}$$

$$\hat{C} = \left(\sum_{k=1}^{T} E\left[y_k x_k^{\mathrm{T}}\right]\right)\left(\sum_{k=1}^{T} E\left[x_k x_k^{\mathrm{T}}\right]\right)^{-1}$$

$$\hat{Q} = \frac{1}{T-1}\left(\sum_{k=2}^{T} E\left[(x_k - \hat{A}^o x_{k-1} - \hat{B}^o u_{k-1}) \cdot (x_k - \hat{A}^o x_{k-1} - \hat{B}^o u_{k-1})^{\mathrm{T}}\right]\right)$$

$$\hat{R} = \frac{1}{T}\left(\sum_{k=1}^{T} E\left[(y_k - \hat{C}^o x_k)(y_k - \hat{C}^o x_k)^{\mathrm{T}}\right]\right)$$

[5] $E[\bullet|\mathbf{U}, \mathbf{Y}]$ is denoted as $E[\bullet]$ afterward for simplicity.

where $\hat{A}^o, \hat{B}^o, \hat{C}^o$, and $\hat{x}_1^o$ are the *a priori* or initial estimates of A, B, C, and $\hat{x}_1$.

The expected values used earlier can be calculated as the E-*step* by applying Kalman smoother (19) and (20) for offline case or applying Kalman filter (19) for online case. The resulting estimation solutions for $\hat{x}_1, P_1, Q$, and R are shown in (23) and (24) as the M-*step* (for offline case, see [13] and [14] for details).

REFERENCES

[1] E. Pereira, S. Aphale, V. Feliu, and S. Moheimani, "Integral resonant control for vibration damping and precise tip-positioning of a single-link flexible manipulator," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 2, pp. 232–240, Apr. 2011.

[2] J. Park, P.-H. Chang, H.-S. Park, and E. Lee, "Design of learning input shaping technique for residual vibration suppression in an industrial robot," *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 1, pp. 55–65, Feb. 2006.

[3] W. Chen and M. Tomizuka, "Estimation of load side position in indirect drive robots by sensor fusion and Kalman filtering," in *Proc. Amer. Control Conf.*, 2010, pp. 6852–6857.

[4] S. Jeon, M. Tomizuka, and T. Katou, "Kinematic Kalman filter (KKF) for robot end-effector sensing," *ASME J. Dynam. Syst., Meas. Control*, vol. 131, no. 2, pp. 21 010–21 018, 2009.

[5] C. Wang, W. Chen, and M. Tomizuka, "Robot end-effector sensing with position sensitive detector and inertial sensors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 5252–5257.

[6] V. Lertpiriyasuwat and M. Berg, "Adaptive real-time estimation of end-effector position and orientation using precise measurements of end-effector position," *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 3, pp. 304–319, Jun. 2006.

[7] M. Quigley, R. Brewer, S. Soundararaj, V. Pradeep, Q. Le, and A. Ng, "Low-cost accelerometers for robotic manipulator perception," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2010, pp. 6168–6174.

[8] P. Cheng and B. Oelmann, "Joint-angle measurement using accelerometers and gyroscopes—A survey," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 2, pp. 404–414, Feb. 2010.

[9] R. Henriksson, M. Norrlöf, S. Moberg, E. Wernholt, and T. Schön, "Experimental comparison of observers for tool position estimation of industrial robots," in *Proc. IEEE 48th Conf. Decision Control*, Dec. 2009, pp. 8065–8070.

[10] J. Wallén, S. Gunnarsson, R. Henriksson, S. Moberg, and M. Norrlöf, "ILC applied to a flexible two-link robot model using sensor-fusion-based estimates," in *Proc. 48th IEEE Conf. Decision Control*, Dec. 2009, no. 1, pp. 458–463.

[11] P. Axelsson, R. Karlsson, and M. Norrlöf, "Bayesian state estimation of a flexible industrial robot," *Control Eng. Practice*, vol. 20, no. 11, pp. 1220–1228, 2012.

[12] W. Chen and M. Tomizuka, "Load side state estimation in robot with joint elasticity," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Jul. 2012, pp. 598–603.

[13] R. H. Shumway and D. S. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," *J. Time Series Anal.*, vol. 3, no. 4, pp. 253–264, Jul. 1982.

[14] V. Digalakis, J. Rohlicek, and M. Ostendorf, "ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 1, no. 4, pp. 431–442, Oct. 1993.

[15] Z. Ghahramani and G. E. Hinton, "Parameter estimation for linear dynamical systems," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. CRG-TR-96-2, 1996.

[16] C. C. de Wit, G. Bastin, and B. Siciliano, *Theory of Robot Control*, 1st ed. New York, NY, USA: Springer-Verlag, Jan. 1996.

[17] L. L. Cam, "Maximum likelihood: An introduction," *Int. Stat. Rev.*, vol. 58, no. 2, pp. 153–171, 1990.

[18] FANUC Corporation, M-16iB/20 Robot Data Sheet. (accessed in 2012). [Online]. Available: http://www.micromech.co.uk/dir_products/pdf/fanuc/m_16ib_20_10l.pdf

[19] W. Chen and M. Tomizuka, "Iterative learning control with sensor fusion for robots with mismatched dynamics and mismatched sensing," in *Proc. ASME Dynam. Syst. Control Conf.*, 2012, pp. 1480–1488.

Wenjie Chen (S'12–M'12) received the B.Eng. degree from Zhejiang University, Zhejiang, China, in 2007, and the M.S. and Ph.D. degrees from the University of California (UC), Berkeley, CA, USA, in 2009 and 2012, respectively, all in mechanical engineering.

He is currently a Postdoctoral Scholar in the Department of Mechanical Engineering, UC Berkeley. His research interests include design and implementation of advanced control and sensing algorithms with applications to robotic/mechatronic systems, such as industrial robots, wearable assistive robotics, and robots for advanced manufacturing.

Dr. Chen was among the Best Student Paper Finalists of the 6th IFAC Symposium on Mechatronic Systems, 2013.

Masayoshi Tomizuka (M'86–SM'95–F'97) received the B.S. and M.S. degrees in mechanical engineering from Keio University, Tokyo, Japan, and the Ph.D. degree in mechanical engineering from Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 1974.

He joined the Department of Mechanical Engineering, University of California, Berkeley, CA, USA, in 1974, where he is currently the Cheryl and John Neerhout, Jr., Distinguished Professor. He teaches courses on dynamic systems and controls and conducts research on optimal and adaptive control, digital control, motion control, and their applications to robotics, manufacturing, information storage devices, and vehicles.

Dr. Tomizuka served as the Program Director of the Dynamic Systems and Control Program of the Civil and Mechanical Systems Division of the National Science Foundation (2002–2004). He was a Technical Editor of the *ASME Journal of Dynamic Systems, Measurement and Control* (1988–1993), and the Editor-in-Chief of the IEEE/ASME TRANSACTIONS ON MECHATRONICS (1997–1999). He was the recipient of the Rudolf Kalman Best Paper Award (ASME, 1995, 2010), the DSCD Outstanding Investigator Award (ASME, 1996), the Charles Russ Richards Memorial Award (ASME, 1997), the Rufus Oldenburger Medal (ASME, 2002), and the John R. Ragazzini Award (American Automatic Control Council, 2006). He is a Fellow of the American Society of Mechanical Engineers (ASME), the International Federation of Automatic Control (IFAC), and the Society of Manufacturing Engineers (SME).