The 2012 IEEE/ASME International Conference on
Advanced Intelligent Mechatronics
July 11-14, 2012, Kaohsiung, Taiwan

Load Side State Estimation in Robot with Joint Elasticity

Wenjie Chen and Masayoshi Tomizuka

Abstract— **For robots with joint elasticity, discrepancies exist between the motor side and the load side. Thus the load side (end-effector) performance can hardly be guaranteed with motor side measurements alone. In this paper, a computationally easy load side state estimation scheme is proposed for the robots with joint elasticity, which is equipped with motor encoders and a low-cost end-effector MEMS sensor such as 3-axial accelerometer. An optimization based inverse differential kinematics algorithm is developed to obtain the load side joint acceleration estimate. Then the joint position and velocity estimation problem is decoupled into simple 2-order kinematic Kalman filters for each joint. Maximum likelihood principle is utilized to estimate the fictitious noise covariances. Both off-line and online solutions are derived. The extension to other sensor configurations is discussed as well. The effectiveness of the developed method is validated through simulation and experimental study on a 6-DOF industrial robot.**

I. Introduction

In robot applications, discrepancies between the available and 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 side information alone. This 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, load side joint state estimation from the end-effector sensing is of particular interest for this purpose.

In [1], a Kalman filter scheme using either dynamic model or kinematic model was investigated for a single-joint robot with joint elasticity. The scheme fused the motor encoder measurements with the load side inertia sensor signals to estimate the load side position. A multi-dimensional kinematic Kalman filter (KKF) was proposed in [2] for multi-joint robot end-effector sensing. Several end-effector sensors (i.e., camera, gyroscope, and accelerometer) were required, while motor encoders were not made use of and joint space estimation was not directly achieved. In [3], [4], 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 [5], [6], 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. The computation load, however, was nontrivial due to the complex dynamic model and the EKF/PF algorithms. Thus, the methods were only intended for the applications where off-line computing was feasible, such as the iterative learning control.

In this paper, a sensor fusion scheme, which is computationally easy and suitable for various applications, is proposed for the robots with joint elasticity and an end-effector accelerometer. 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 kinematic Kalman filter for each joint to estimate the load side joint position and velocity. Expectation maximization (EM) ([7], [8], [9]) is utilized to off-line determine the unknown parameters and the online solution is also proposed. The computation load and extensions to other sensor configurations are discussed in Section IV. Section V presents the simulation and experimental study on a 6-DOF robot. The conclusion is given in Section VI at last.

II. Robot inverse differential kinematics

A. Basic Differential Kinematics

Consider a 6-DOF robot manipulator with n elastic joints ($n \geq 6$). Let $v_e = \begin{bmatrix} \dot{p}_e & \omega_e \end{bmatrix} \in \mathbb{R}^6$ denotes the end-effector Cartesian velocity vector composing of the translational velocity $\dot{p}_e$ and the angular velocity ω_e at the accelerometer mounting point. The kinematic relation between the joint space and the Cartesian space can be described as

$$v_e = J(q_\ell)\dot{q}_\ell \tag{1}$$

where $J(q_\ell) \in \mathbb{R}^{6\times n}$ is the Jacobian matrix mapping from the load side joint velocity $\dot{q}_\ell \in \mathbb{R}^n$ to the end-effector velocity v_e. Take the time derivative of both sides of (1), which gives

$$\dot{v}_e = J(q_\ell)\ddot{q}_\ell + \dot{J}(q_\ell,\dot{q}_\ell)\dot{q}_\ell \tag{2}$$

Note that the acceleration measured by the end-effector accelerometer is only three-dimensional 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 (2) can be rewritten as

$$\ddot{p}_e = \bar{J}(q_\ell)\ddot{q}_\ell + \bar{\dot{J}}(q_\ell,\dot{q}_\ell)\dot{q}_\ell \tag{3}$$

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

This work was supported by FANUC Ltd., Japan.

Wenjie Chen and Masayoshi Tomizuka are with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720, USA {wjchen,tomizuka}@me.berkeley.edu

978-1-4673-2576-9/12/$31.00 ©2012 IEEE

B. Optimization Based Inverse Differential Kinematics

Define the pseudo-inverse of $\bar{J}(q_\ell)$ as

$$J^\dagger(q_\ell) = \bar{J}(q_\ell)^{\mathrm{T}}\left[\bar{J}(q_\ell)\bar{J}(q_\ell)^{\mathrm{T}}\right]^{-1} \tag{4}$$

Then from (3), the load side joint acceleration estimate can be solved as the following infinite solutions

$$\hat{\ddot{q}}_\ell = J^\dagger(q_\ell)\left[\ddot{p}_e - \dot{\bar{J}}(q_\ell,\dot{q}_\ell)\dot{q}_\ell\right] + \left[I - J^\dagger(q_\ell)\bar{J}(q_\ell)\right]\varphi \tag{5}$$

where $\hat{\bullet}$ denotes the inverse differential kinematics estimate of $\bullet$, I is the $n\times n$ identity matrix, and $\varphi\in\mathbb{R}^n$ is an arbitrary vector. The term $J^\dagger(q_\ell)\left[\ddot{p}_e - \dot{\bar{J}}(q_\ell,\dot{q}_\ell)\dot{q}_\ell\right] \in \mathrm{Null}^\perp\left(\bar{J}(q_\ell)\right) \equiv \mathrm{Row}\left(\bar{J}(q_\ell)^{\mathrm{T}}\right)$ is the particular solution which minimizes the norm of the solution $\parallel \hat{\ddot{q}}_\ell \parallel$. 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 (3) as

$$\bar{J}(q_\ell)\ddot{q}_\ell = \ddot{p}_e - \dot{\bar{J}}(q_\ell,\dot{q}_\ell)\dot{q}_\ell \;\Rightarrow\; \bar{A}\hat{\ddot{q}}_\ell = \bar{b} \tag{6}$$

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

$$\min_{\hat{\ddot{q}}_\ell} f(\hat{\ddot{q}}_\ell) \quad \text{s.t. } \bar{A}\hat{\ddot{q}}_\ell = \bar{b} \tag{7}$$

where the imposed physical constraint is expressed as to minimize $f(\hat{\ddot{q}}_\ell)$.

If the a-priori estimate of $\ddot{q}_\ell$ is available as $\hat{\ddot{q}}_\ell^o$, the problem can be simply solved with least squares method as

$$\min_{\hat{\ddot{q}}_\ell} f(\hat{\ddot{q}}_\ell) = \frac{1}{2}\left\|\hat{\ddot{q}}_\ell - \hat{\ddot{q}}_\ell^o\right\|_2^2 \qquad \text{s.t. } \bar{A}\hat{\ddot{q}}_\ell = \bar{b} \tag{8}$$

which has the global optimal closed form solution as

$$\hat{\ddot{q}}_\ell = \bar{A}^{\mathrm{T}}\left(\bar{A}\bar{A}^{\mathrm{T}}\right)^{-1}\bar{b} + \left[I - \bar{A}^{\mathrm{T}}\left(\bar{A}\bar{A}^{\mathrm{T}}\right)^{-1}\bar{A}\right]\hat{\ddot{q}}_\ell^o \tag{9}$$

where the calculations of $\bar{A}$, $\bar{b}$, and $\hat{\ddot{q}}_\ell^o$ are addressed in the following sections.

C. Model Utilization for Rough Estimates

The dynamics of an n-joint robot with the gear compliance can be expressed as

$$\begin{aligned}
&M_\ell(q_\ell)\ddot{q}_\ell + C(q_\ell,\dot{q}_\ell)\dot{q}_\ell + G(q_\ell) + D_\ell\dot{q}_\ell + F_{\ell c}\mathrm{sgn}(\dot{q}_\ell) \quad (10)\\
&\quad + J^{\mathrm{T}}(q_\ell)f_{ext} = K_J\left(N^{-1}q_m - q_\ell\right) + D_J\left(N^{-1}\dot{q}_m - \dot{q}_\ell\right)\\
&M_m\ddot{q}_m + D_m\dot{q}_m + F_{mc}\mathrm{sgn}(\dot{q}_m) = \tau_m \quad (11)\\
&\quad - N^{-1}\left[K_J\left(N^{-1}q_m - q_\ell\right) + D_J\left(N^{-1}\dot{q}_m - \dot{q}_\ell\right)\right]
\end{aligned}$$

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. $f_{ext}\in\mathbb{R}^6$ denotes the external force acting on the robot due to the contact with the environment.

With (11), the load side joint position q_ℓ can be roughly estimated as

$$\begin{aligned}
\hat{q}_\ell^o = (\hat{D}_J s + \hat{K}_J)^{-1}\Big[&\hat{K}_J N^{-1}q_m + \hat{D}_J N^{-1}\dot{q}_m\\
&- N\left(\tau_m - \hat{M}_m\hat{\ddot{q}}_m - \hat{D}_m\dot{q}_m - \hat{F}_{mc}\mathrm{sgn}(\dot{q}_m)\right)\Big]
\end{aligned} \tag{12}$$

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 (12) as approximation. 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.

D. Final Optimization Problem

Note that the rough estimate $\hat{\dot{q}}_\ell^o$ could be acceptable but noisy. So the load side joint acceleration estimate cannot be obtained by direct differentiation of $\hat{\dot{q}}_\ell^o$. With $\hat{\dot{q}}_\ell^o \triangleq \int \hat{\ddot{q}}_\ell^o dt$, the optimization problem (8) is reformulated as

$$\min_{\hat{\ddot{q}}_\ell} f(\hat{\ddot{q}}_\ell) = \frac{1}{2}\left\|\int \hat{\ddot{q}}_\ell dt - \int \hat{\ddot{q}}_\ell^o dt\right\|_2^2 \quad \text{s.t. } \bar{A}\hat{\ddot{q}}_\ell = \bar{b} \tag{13}$$

This optimization problem over the whole time series would be impractical to solve especially within the real-time environment. Instead, a point-wise optimization can be performed for each time step to generate the sub-optimal solution. For each time step t_k, let

$$\hat{\dot{q}}_{\ell,k}^o = \int_0^{t_k}\hat{\ddot{q}}_\ell^o(t)dt, \quad \hat{\dot{q}}_{\ell,k} = \sum_{i=0}^{k}\hat{\ddot{q}}_{\ell,i}\Delta t \tag{14}$$

where the subscript k denotes the time index and Δt is the sampling time. Then (13) can be relaxed into a convex optimization problem for each time step t_k as

$$\begin{aligned}
&\min_{\hat{\ddot{q}}_{\ell,k}} f(\hat{\ddot{q}}_{\ell,k}) = \frac{1}{2}\left\|\hat{\ddot{q}}_{\ell,k} - \Delta\hat{\ddot{q}}_{\ell,k}\right\|_2^2\\
&\text{s.t. } A_k\hat{\ddot{q}}_{\ell,k} = b_k
\end{aligned} \tag{15}$$

where $\Delta\hat{\ddot{q}}_{\ell,k} = \frac{\hat{\dot{q}}_{\ell,k}^o - \hat{\dot{q}}_{\ell,k-1}}{\Delta t}$ also includes the accumulated acceleration estimation error not compensated from previous steps. Similar as (9), the optimal load side joint acceleration estimate is thus obtained as

$$\hat{\ddot{q}}_{\ell,k} = A_k^{\mathrm{T}}\left(A_k A_k^{\mathrm{T}}\right)^{-1}b_k + \left[I - A_k^{\mathrm{T}}\left(A_k A_k^{\mathrm{T}}\right)^{-1}A_k\right]\Delta\hat{\ddot{q}}_{\ell,k} \tag{16}$$

where

$$A_k = \bar{J}(q_{\ell,k}), \quad b_k = \ddot{p}_{e,k} - \dot{\bar{J}}(q_{\ell,k},\dot{q}_{\ell,k})\dot{q}_{\ell,k} \tag{17}$$

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

$$\ddot{p}_e = R_a(q_\ell) f_a + g \tag{18}$$

where $R_a(q_\ell)$ is the rotation matrix of the accelerometer coordinate frame with respect to the world coordinate frame and $g = \begin{bmatrix} 0 & 0 & -9.8 \end{bmatrix}^{\mathrm{T}} \mathrm{m/sec}^2$ is the 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 $\hat{q}^o_\ell$ and $\hat{\dot{q}}^o_\ell$ are used instead in (17) and (18) to calculate $J(q_\ell)$, $\dot{J}(q_\ell, \dot{q}_\ell)\dot{q}_\ell$, 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.

III. Kinematic Kalman Filter

A. Decoupled Kinematic Kalman Filter

As discussed above, the load side rough approximations have been obtained as $\hat{q}^o_\ell$ in (12) and $\hat{\ddot{q}}_\ell$ in (16) for each joint. Thus, the estimation problem for the whole robot can be decoupled into n kinematic Kalman filters (KKF) running in parallel, which are computationally easy, to better estimate the load side joint position and velocity. The discrete time kinematic model for the Kalman filter is written as

$$\underbrace{\begin{bmatrix} q_{\ell,k+1} \\ \dot{q}_{\ell,k+1} \end{bmatrix}}_{x_{k+1}} = \underbrace{\begin{bmatrix} I & \Delta t I \\ \mathbf{0} & I \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} q_{\ell,k} \\ \dot{q}_{\ell,k} \end{bmatrix}}_{x_k} + \underbrace{\begin{bmatrix} \frac{1}{2}\Delta t^2 I \\ \Delta t I \end{bmatrix}}_{B} \underbrace{\hat{\ddot{q}}_{\ell,k}}_{u_k} + w_k \tag{19a}$$

$$\underbrace{\hat{q}^o_{\ell,k}}_{y_k} = \underbrace{\begin{bmatrix} I & \mathbf{0} \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} q_{\ell,k} \\ \dot{q}_{\ell,k} \end{bmatrix}}_{x_k} + v_k \tag{19b}$$

which is in the following standard form

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{20a}$$

$$y_k = Cx_k + v_k \tag{20b}$$

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

Note that $\hat{q}^o_{\ell,k}$ and $\hat{\ddot{q}}_{\ell,k}$ 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.

B. Covariance Estimation

These parameters can be adapted based on the maximum likelihood principle [7], [8], [9]. The derivation of the following estimation solutions for this specific problem is detailed in the Appendix.

1) Off-line Estimation: If off-line processing is available, which is applicable in iterative applications, the whole time series data can be accessed. In this case, expectation maximization (EM) algorithm can be applied as follows

- E-step: run Kalman smoother with current best estimates of $\hat{x}_1, P_1, Q$, and R.
- M-step: update $\hat{x}_1, P_1, Q$, and R as in (21) using the acausal estimations from Kalman smoother.

$$\begin{aligned}
\hat{x}_1 &= \hat{x}_{1|T} \qquad\qquad \hat{P}_1 = P_{1|T} \\
\hat{Q} &= \frac{1}{T-1} \sum_{k=2}^{T} \Big[\left(\hat{x}_{k|T} - A\hat{x}_{k-1|T} - Bu_{k-1} \right) \\
&\quad \cdot \left(\hat{x}_{k|T} - A\hat{x}_{k-1|T} - Bu_{k-1} \right)^{\mathrm{T}} \\
&\quad + P_{k|T} - AP^{\mathrm{T}}_{(k,k-1)|T} - P_{(k,k-1)|T}A^{\mathrm{T}} + AP_{k-1|T}A^{\mathrm{T}} \Big] \\
\hat{R} &= \frac{1}{T} \sum_{k=1}^{T} \Big[\left(y_k - C\hat{x}_{k|T} \right) \left(y_k - C\hat{x}_{k|T} \right)^{\mathrm{T}} + CP_{k|T}C^{\mathrm{T}} \Big]
\end{aligned} \tag{21}$$

  where $\hat{\bullet}_{k|j}$ represents the conditional expectation of $\bullet_k$ given the information up to the j-th time step.
- Iterate from E-step until the increment of the expected likelihood is within chosen threshold.

2) Online Estimation: If real-time computing is desired, only causal estimation from forward recursion (i.e., Kalman filter) can be used. Furthermore, instead of estimating using the whole time series as in (21), here the covariances Q and R can be adapted for each time step as follows

$$\begin{aligned}
\hat{Q}^o_{k+1} = &\left(\hat{x}_{k|k} - A\hat{x}_{k-1|k-1} - Bu_{k-1} \right) \\
&\cdot \left(\hat{x}_{k|k} - A\hat{x}_{k-1|k-1} - Bu_{k-1} \right)^{\mathrm{T}} \\
&+ P_{k|k} - AP^{\mathrm{T}}_{(k,k-1)|k} - P_{(k,k-1)|k}A^{\mathrm{T}} + AP_{k-1|k-1}A^{\mathrm{T}}
\end{aligned} \tag{22a}$$

$$\hat{R}^o_{k+1} = \left(y_k - C\hat{x}_{k|k} \right) \left(y_k - C\hat{x}_{k|k} \right)^{\mathrm{T}} + CP_{k|k}C^{\mathrm{T}} \tag{22b}$$

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

$$\hat{Q}_{k+1} = \left(1 - \frac{1}{N_Q} \right) \hat{Q}_k + \frac{1}{N_Q} \hat{Q}^o_{k+1} \tag{23a}$$

$$\hat{R}_{k+1} = \left(1 - \frac{1}{N_R} \right) \hat{R}_k + \frac{1}{N_R} \hat{R}^o_{k+1} \tag{23b}$$

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 utilized in the online Kalman filter.

IV. Discussion of the Approach

A. Computation Load

It should be noted that one of the great advantages of the proposed method over others is its light computation load. This approach mainly consists of 2 stages, optimization based inverse differential kinematics, and KKF with covariance estimation. At the inverse differential kinematics stage, the main computation lies in (12), which is diagonal calculation due to the property of motor side model, and (16), where a 3 × 3 matrix inversion requires the most effort.

The calculations of (17) and (18) are basically 3×1 vector operations and 3×3 matrix multiplications. After this stage, the problem becomes decoupled Kalman filter (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 easy. In the covariance estimation, only 2×2 matrix multiplications and scalar operations are required. Therefore, the overall computation load for the proposed approach is sufficiently light for both online computing and off-line 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 be easily derived.

If motor encoders and the end-effector gyroscope are available, the optimization problem (8) in the inverse differential kinematics stage can be modified to obtain the load side joint velocity estimate as

$$\min_{\hat{\dot{q}}_\ell} f(\hat{\dot{q}}_\ell) = \frac{1}{2}\left\|\hat{\dot{q}}_\ell - \hat{\dot{q}}_\ell^o\right\|_2^2 \qquad \text{s.t. } \underline{J}(\hat{q}_\ell^o)\hat{\dot{q}}_\ell = \omega_e \tag{24}$$

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) is then simplified to be the one with a 1-order kinematic model

$$q_{\ell,k+1} = q_{\ell,k} + \Delta t\hat{\dot{q}}_{\ell,k} + w_k, \quad \hat{q}_{\ell,k}^o = q_{\ell,k} + v_k \tag{25}$$

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 (8) becomes the case to obtain the load side joint position estimate as

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

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 q_ℓ.

V. Simulation & Experimental Study

A. Test Setup

The proposed methods are implemented on a 6-joint industrial robot, FANUC M-16*i*B/20, as shown in Fig. 1. The robot is equipped with built-in motor encoders for each joint. An inertia sensor (Analog Devices, ADIS16400) consisting of a 3-axial accelerometer and a 3-axial gyroscope is attached to the end-effector. The three-dimensional position measurement system, CompuGauge 3D (accuracy of 0.15mm, resolution of 0.01mm), is utilized to measure the end-effector tool center point (TCP) position as a ground truth for performance validation. The sampling rates of all the sensor signals as well as the real-time controller implemented through MATLAB xPC Target are set to be 1kHz.

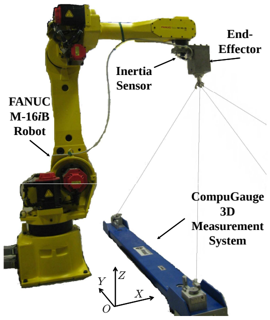

Fig. 1. FANUC M-16*i*B Robot System

Furthermore, a robot simulator has been designed using MATLAB Simulink & SimMechanics Toolbox based on robot dynamic and kinematic parameters as well as the calibrated sensor parameters. Therefore, robot dynamic simulation is available with the access of load side joint information for performance evaluation. The experimental validation, however, can be only conducted in Cartesian space to compare with CompuGauge 3D measurements.

B. Algorithm Settings

The testing TCP trajectory (Fig. 2) is a 10cm × 10cm square path on the Y-Z plane with fixed orientation and maximum velocity of 1m/sec. 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*: Off-line estimation with EM and Kalman smoother. A 30Hz 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 windows sizes $N_Q = N_R = 500$. A 100Hz 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 as 100Hz as a trade off between the phase delay effect and high frequency noise filtering.
3) *KKF-OnlineFix*: The same as *KKF-Online* except that the covariances Q and R are pre-tuned and fixed.
4) *CG3D*: Real measurements from CompuGauge 3D.
5) *Motor*: Using motor encoder signals directly as load side information to estimate.

C. Simulation Results

The load side joint estimation errors calculated from the simulated load side information are plotted using absolute values in Fig. 3 and Fig. 4. It is shown that the proposed *KKF* schemes outperform the *Motor* setting significantly. Particularly in Fig. 3, for Joint 2, 3, and 5, where gravity effects are evident, the estimation by *Motor* suffers from noticeable position offset, while the *KKF* schemes are fine. The online velocity estimates (Fig. 4) from *KKF-Online* and

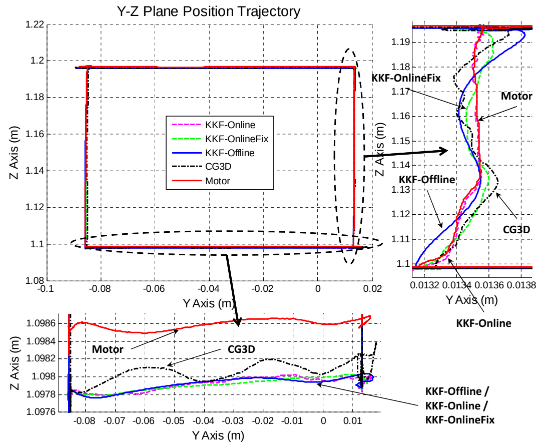

Fig. 2. Y-Z Plane TCP Position Estimation (Experiment)

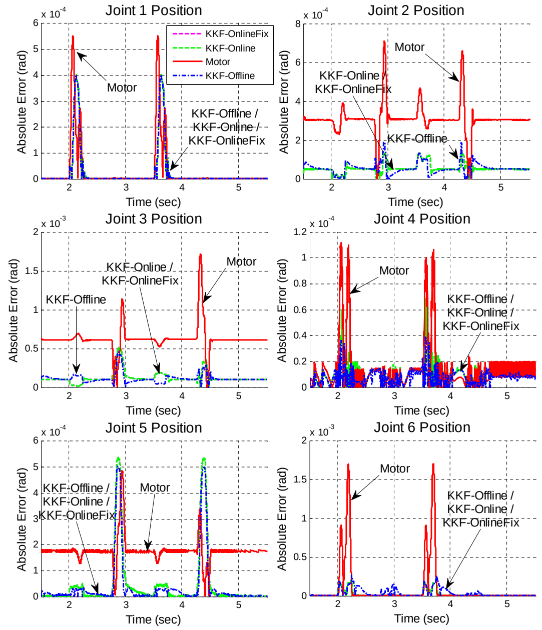

Fig. 3. Load Side Joint Position Estimation Absolute Error (Simulation)

KKF-OnlineFix 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.

D. Experimental Results

The methods are also implemented on the actual experimental setup for Cartesian space comparisons. The joint space estimation results are used in the forward kinematics to obtain the Cartesian space estimates for comparisons. Fig. 2 shows the estimated position trajectory on the Y-Z plane. Again, it is clearly seen that all the *KKF* settings perform much better than the *Motor* setting by capturing closer transient motion on the Y Axis and with much less offset on the Z Axis.

The superior performance of the proposed schemes can be better appreciated for the residual vibration sensing when the robot comes to a stop as shown in Fig. 5 and Fig. 6[1]. In general, the *Motor* setting cannot capture the vibratory motion at the end-effector, while the proposed *KKF* schemes are able to do it with the fusion of end-effector accelerometer measurements. In Fig. 5, it is seen that during the stopping period (i.e., after 3.8sec for Y Axis and after 3sec for Z Axis), the *Motor* estimates look static. The actual residual motion, however, is vibratory and can be successfully captured by the *KKF* estimates.

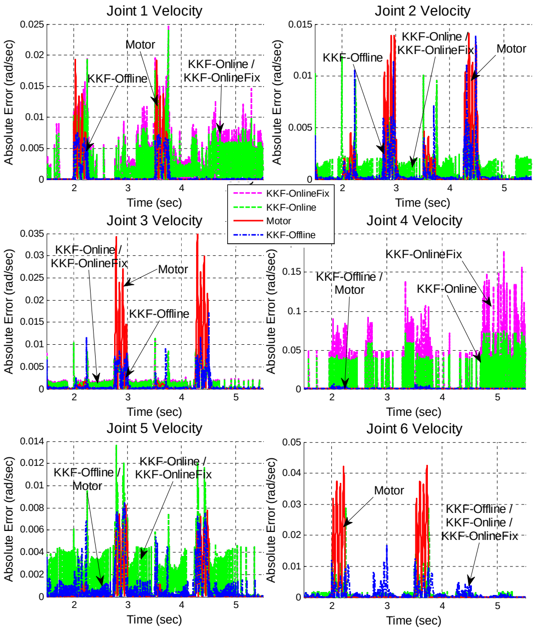

Fig. 4. Load Side Joint Velocity Estimation Absolute Error (Simulation)

TABLE I
TCP Estimation Errors When Coming to a Stop (Experiment)

RMS Errors	Pos. (mm)	Vel. (mm/sec)	Acc. (mm/sec^2)
KKF-Offline	0.262	34.638	1641.400
KKF-Online	0.440	37.832	2729.828
KKF-OnlineFix	0.476	34.517	2729.347
Motor	0.758	67.652	4843.979

Another benefit of the proposed *KKF* schemes is that the velocity estimates are even cleaner than the *CG3D* setting which is obtained by direct differentiation from the position measurement, even though there are also some noises present in the *KKF-Online* and *KKF-OnlineFix* estimates. Among all the *KKF* settings, *KKF-Offline* performs the best due to its acausal processing availability. *KKF-Online* and *KKF-OnlineFix* perform similarly well, which indicates that with properly tuned covariances, the online computing can be further simplified without sacrificing much performance. These conclusions can also be drawn from the position/velocity/acceleration root-mean-square (RMS) errors listed in Table I, which shows the RMS estimation errors can be reduced by the proposed *KKF* schemes to about half or even less of that of the *Motor* estimates.

[1] The plotted Cartesian error is defined as the Euclidean distance between the estimated position and the actual position.

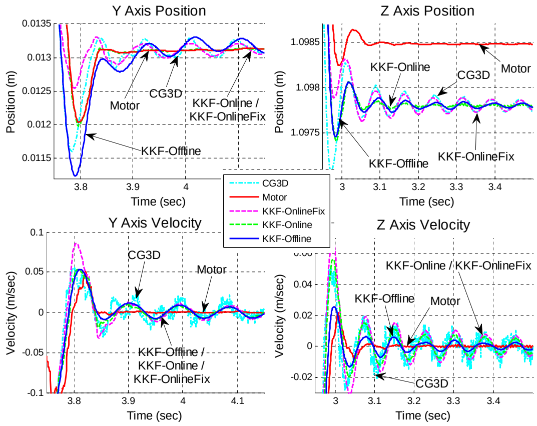

Fig. 5. TCP Position and Velocity Estimation on Y & Z Axes When Coming to a Stop (Experiment)

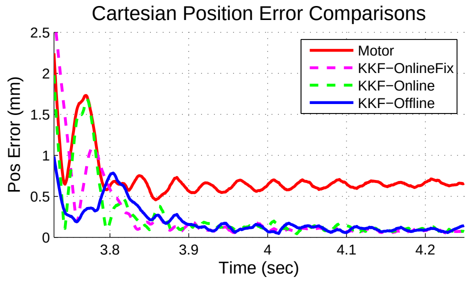

Fig. 6. TCP Estimation Error When Coming to a Stop (Experiment)

VI. Conclusion

This paper investigated a load side state estimation problem for the robots with joint elasticity. 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 kinematic Kalman filters to estimate the load side joint position and velocity. Off-line and online solutions were presented for the fictitious noise covariance determination. The proposed approach is computationally easy, which is suitable for off-line applications as well as online computing. Simulation and experimental study on a 6-DOF industrial robot were conducted to show the superior performance of the developed methods. For the related work, this scheme has been successfully applied in the off-line iterative learning control, while the online application to use the estimation for real-time feedback control is the immediate following work.

References

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

[2] S. Jeon, M. Tomizuka, and T. Katou, "Kinematic Kalman Filter (KKF) for Robot End-Effector Sensing," *Journal of Dynamic Systems, Measurement, and Control*, vol. 131, no. 2, pp. 21010–21018, 2009.

[3] M. Quigley, R. Brewer, S. Soundararaj, V. Pradeep, Q. Le, and A. Ng, "Low-cost accelerometers for robotic manipulator perception," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 6168–6174, IEEE, 2010.

[4] P. Cheng and B. Oelmann, "Joint-Angle Measurement Using Accelerometers and Gyroscopes A Survey," *Instrumentation and Measurement, IEEE Transactions on*, vol. 59, no. 2, pp. 404–414, 2010.

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

[6] P. Axelsson, R. Karlsson, and M. Norrlöf, "Bayesian State Estimation of a Flexible Industrial Robot," *Control Engineering Practice*, 2011.

[7] R. H. Shumway and D. S. Stoffer, "An Approach To Time Series Smoothing and Forecasting Using the Em Algorithm," *Journal of Time Series Analysis*, vol. 3, pp. 253–264, July 1982.

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

[9] Z. Ghahramani and G. Hinton, "Parameter estimation for linear dynamical systems," *University of Toronto technical report CRG-TR-96-2*, vol. 6, 1996.

Appendix

The derivation below is based on [7], [8], [9] with the extension to include the input Bu_k in the model (20). Given the input series u_k and the output series y_k for $1 \le k \le T$, the objective is to maximize the log likelihood of $\hat{x}_1, P_1, A, B, C, Q$, and R, which can be derived as

$$\begin{aligned} G(\hat{x}_1,P_1,A,B,C,Q,R) &= \text{Constant} - \frac{T-1}{2}\log|Q| \\ &- \sum_{k=2}^{T}\frac{1}{2}(x_k - Ax_{k-1} - Bu_{k-1})^{\mathrm{T}}Q^{-1}(x_k - Ax_{k-1} - Bu_{k-1}) \\ &- \sum_{k=1}^{T}\frac{1}{2}(y_k - Cx_k)^{\mathrm{T}}R^{-1}(y_k - Cx_k) - \frac{T}{2}\log|R| \\ &- \frac{1}{2}\log|P_1| - \frac{1}{2}(x_1-\hat{x}_1)^{\mathrm{T}}P_1^{-1}(x_1-\hat{x}_1) \end{aligned} \tag{27}$$

Since the actual state distributions are unknown, the *expected* likelihood[2] $E[G(\hat{x}_1,P_1,A,B,C,Q,R)]$ is used instead to perform maximization. With the first order condition $\frac{\partial E[G(\bullet)]}{\partial(\bullet)} = 0$, the resulting estimates can be derived as

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

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

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

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

$$\begin{aligned} \hat{Q} = \frac{1}{T-1}\Bigg(\sum_{k=2}^{T} E\Big[&(x_k - \hat{A}^o x_{k-1} - \hat{B}^o u_{k-1}) \\ &\cdot (x_k - \hat{A}^o x_{k-1} - \hat{B}^o u_{k-1})^{\mathrm{T}}\Big]\Bigg) \end{aligned}$$

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

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 above expected values can be calculated by applying the standard Kalman smoother/filter.

[2] Hereafter the conditional expectation $E(\bullet|j)$ given the information up to the j-th time step is denoted as $E(\bullet)$ for simplicity.