Proceedings of ISFA2016
2016 International Symposium on Flexible Automation
ISFA2016
1 - 3 August, 2016, Cleveland, Ohio, U.S.A.

Object Position and Orientation Tracking for Manipulators Considering Nonnegligible Sensor Physics

Yongxiang Fan, Hsien-Chung Lin
Department of Mechanical Engineering
University of California
Berkeley, California, 94720
Email: {yongxiang_fan, hclin}@berkeley.edu

Yu Zhao, Chung-Yen Lin
Department of Mechanical Engineering
University of California
Berkeley, California, 94720
Email: {yzhao334, chung_yen}@berkeley.edu

Te Tang, Masayoshi Tomizuka
Department of Mechanical Engineering
University of California
Berkeley, California, 94720
Email: {tetang, tomizuka}@berkeley.edu

Wenjie Chen
FANUC Corporation
Yamanashi Prefecture, 401-0597, Japan
Email: wenjie.chen.zju@gmail.com

ABSTRACT

Real-time object tracking for manipulators has been applied to industrial automation such as assembly and human robot collaboration. During the tracking process, the vision sensor is used as a feedback means to the robot controller. It means that the imperfect sensing and stability issues must be carefully considered. This paper deals with the modeling of sensor physics (e.g. limited sampling rate, irregular sampling time, packet loss, noise, and latency), and realizes a globally asymptotically stable tracking controller. First, an varying rate Kalman filter is applied to estimate the markers' positions and the corresponding error covariances. Then, a maximum likelihood estimation problem is solved to estimate the transformation between the target frame and the world frame. In addition, a dynamic tracking controller is implemented to realize object tracking. The stability of the tracking controller under model uncertainties is also discussed. The proposed tracking algorithm is experimentally verified on an industrial robot.

1 INTRODUCTION

Machine vision techniques have broad applications in industry, including material handling, assembly and human-robot collaboration. Traditionally, a manipulator follows the look-then-move algorithm [1], which involves 1) taking an image and locating the static workpieces, 2) running the motion planning algorithm and generating a feasible trajectory, and 3) executing the trajectory to reach the workpiece. However, with increasingly complex tasks and stringent requirement, the traditional look-then-move method becomes insufficient. On the other hand, real-time visual tracking is able to detect moving workpieces and feedback the sensor signal into the robot controller real time. Therefore, it is desired to explore the possibilities to control a manipulator to execute more complicated tasks with higher accuracy.

It is best to have the vision signal updated in each controller sampling period. This requires high-speed vision sensor, which is not practical due to cost. On the other hand, low-cost vision systems may introduce challenges such as a limited sampling rate, irregular sampling time, packet loss, noise, and latency. These challenges are called sensor physics. Without properly handling these challenges, the tracking accuracy and stability of

the vision feedback loop cannot be guaranteed.

Apart from the sensor physics, there exist other challenges in the design of tracking controller. First, a global parameterization of attitude is desired to avoid singularities and parameterize arbitrary attitude. Secondly, a Cartesian space controller needs to be properly designed to achieve globally asymptotically stable tracking. Thirdly, the closed-loop system should have a desired speed of convergence to track a moving target. Lastly, the system should tolerate a certain amount of model uncertainties.

Some methods have been proposed in order to model the sensor physics and estimate the pose of the target. Considering the irregular sampling time and noise, a varying rate Kalman filter is applied to a kinematic model [2] to estimate target position. To increase the estimation accuracy, the model parameters for Kalman filter is identified by expectation-maximization in an off-line manner [3]. To overcome the slow sampling rate and the latency effects, a kinematic Kalman filter and the sensor fusion are utilized [4] to estimate the pose of the robot end-effector. These methods only partially deal with sensing issues.

With regard to the tracking controller design, a local parameterization of orientation and a PID controller are applied in [5] to track the position and orientation of the object. However, the visual tracking controller would lose the object if the orientation error is larger than a threshold. To realize global parameterization and asymptotically stable tracking, a quaternion based tracking controller from kinematic level is provided in [6]. The stability of a kinematic and a dynamic controller are studied in [7] without considering parameter uncertainties.

In this paper, a visual tracking control framework is introduced in presence of nonnegligible sensor physics. It begins with proposing an varying rate Kalman filter and the maximum likelihood technique to estimate the pose of the target. Then a tracking controller is designed considering both robot kinematics and robot dynamics. Also, the stability analysis under parameter uncertainties is conducted. Experimental results on an industrial robot are presented to demonstrate the proposed approach.

2 TARGET POSE ESTIMATION

Given the positions of feature points measured by the vision sensor, we can calculate the pose of the target by least squares, as shown in [8]. However, it assumes that the data sets are complete and reliable. In reality, the sensor might have aforementioned sensor physics, and the tracking performance will be downgraded if these properties are not taken into consideration. A reasonable solution to reduce the effect of the sensor physics is to adjust the weights of different feature points based on their relative measurement uncertainties. Therefore, in this section, a Kalman filter and a maximum likelihood technique are combined to estimate the target rotation matrix R_t and the translation vector $\mathbf{t}$. First, an varying rate Kalman filter is adopted to estimate positions and uncertainties of feature points of the target, then the maximum likelihood method is employed to estimate R_t and $\mathbf{t}$ of the target, with the weights reflecting the relative uncertainties of different feature points.

2.1 Feature Points Estimation by Kalman Filter

Without any prior knowledge of target acceleration information, we assume that the target kinematics is a noise driven constant acceleration model [9]:

$$\begin{aligned} \dot{x}_k(t) &= \underbrace{\begin{bmatrix} 0 & I_3 & 0 \\ 0 & 0 & I_3 \\ 0 & 0 & 0 \end{bmatrix}}_{A^c} x_k(t) + \underbrace{\begin{bmatrix} 0 \\ 0 \\ I_3 \end{bmatrix}}_{B^c} w^c(t) \\ y_k(t) &= \underbrace{\begin{bmatrix} I_3 & 0 & 0 \end{bmatrix}}_{C^c} x_k(t) + v^c(t) \end{aligned} \tag{1}$$

where $x_k(t) \in \mathbb{R}^9$ is the state vector for 3D position, velocity and acceleration of the k-th point at time t, $w^c(t)$ is the process noise to the model, and $v^c(t)$ represents the measurement noise.

By considering the time interval from t_{i-1}^k to t_i^k, where t_i^k is the i-th sample receiving time for the k-th point, Eq. (1) can be discretized as:

$$\begin{aligned} x_k(t_i^k) &= e^{A^c(t_i^k - t_{i-1}^k)} x_k(t_{i-1}^k) + \int_{t_{i-1}^k}^{t_i^k} e^{A^c(t_i^k - \tau)} B^c w^c(\tau) d\tau \\ y_k(t_i^k) &= C^c x_k(t_i^k - t_L) + \frac{1}{T_{i-1}^k} \int_{t_{i-1}^k - t_L}^{t_i^k - t_L} v^c(\tau) d\tau \end{aligned} \tag{2}$$

where $T_{i-1}^k = t_i^k - t_{i-1}^k$ is the sampling time at step $i-1$, and t_L is the time delay and is assumed to be a constant. Without prior knowledge, w^c and v^c are set to be white noises, i.e. $w^c \sim \mathcal{N}(0, W^c)$, and $v^c \sim \mathcal{N}(0, V^c)$. Consider the latency in the measurement system, we have:

$$x_k(t_i^k) = e^{A^c t_L} x_k(t_i^k - t_L) + \int_{t_i^k - t_L}^{t_i^k} e^{A^c(t_i^k - \tau)} B^c w^c(\tau) d\tau \tag{3}$$

By pre-multiplying $C^c e^{-A^c t_L}$ to Eq. (3), and combining Eq. (3)

with Eq. (2), we obtain a discrete-time model of the target:

$$
\begin{aligned}
&x_k(t_i^k) = A(T_{i-1}^k)x_k(t_{i-1}^k) + w(t_{i-1}^k, t_i^k)\\
&y_k(t_i^k) = Cx_k(t_i^k) + v(t_{i-1}^k, t_i^k)\\
&\text{with:}\\
&A(T_{i-1}^k) = e^{A^c(t_i^k - t_{i-1}^k)}\\
&C = C^c e^{-A^c t_L}\\
&w(t_{i-1}^k, t_i^k) = \int_{t_{i-1}^k}^{t_i^k} e^{A^c(t_i^k-\tau)} B^c w^c(\tau) d\tau\\
&v(t_{i-1}^k, t_i^k) = -C^c \int_{t_i^k - t_L}^{t_i^k} e^{A^c(t_i^k - t_L - \tau)} B^c w^c(\tau) d\tau +\\
&\qquad \frac{1}{T_{i-1}^k} \int_{t_{i-1}^k - t_L}^{t_i^k - t_L} v^c(\tau) d\tau
\end{aligned}
\tag{4}
$$

$w(t_{i-1}^k, t_i^k)$ and $v(t_{i-1}^k, t_i^k)$ are zero mean, and the time-variant covariances $W(T_{i-1}^k)$ and $V(T_{i-1}^k)$ of $w(t_{i-1}^k, t_i^k)$ and $v(t_{i-1}^k, t_i^k)$ in Eq. (4) can be calculated explicitly. Once receiving measurement $y_k(t_i^k)$ at time t_i^k, we can estimate $\hat{x}_k(t_i^k|t_i^k)$ and $\Sigma_k(t_i^k|t_i^k)$ based on $\hat{x}_k(t_{i-1}^k|t_{i-1}^k)$ and $\Sigma_k(t_{i-1}^k|t_{i-1}^k)$ by the following form:

$$
\begin{aligned}
&\hat{x}_k(t_i^k|t_{i-1}^k) = A(T_{i-1}^k)\hat{x}_k(t_{i-1}^k|t_{i-1}^k)\\
&\Sigma_k(t_i^k|t_{i-1}^k) = A(T_{i-1}^k)\Sigma_k(t_{i-1}^k|t_{i-1}^k)A^T(T_{i-1}^k) + W(T_{i-1}^k)\\
&\hat{x}_k(t_i^k|t_i^k) = \hat{x}_k(t_i^k|t_{i-1}^k) + \Sigma_k(t_i^k|t_{i-1}^k)C^T \times\\
&\quad \left(C\Sigma_k(t_i^k|t_{i-1}^k)C^T + V(T_{i-1}^k)\right)^{-1} \left(y_k(t_i^k) - C\hat{x}_k(t_i^k|t_{i-1}^k)\right)\\
&\Sigma_k(t_i^k|t_i^k) = \Sigma_k(t_i^k|t_{i-1}^k) - \Sigma_k(t_i^k|t_{i-1}^k)C^T \times\\
&\quad \left(C\Sigma_k(t_i^k|t_{i-1}^k)C^T + V(T_{i-1}^k)\right)^{-1} C\Sigma_k(t_i^k|t_{i-1}^k)
\end{aligned}
\tag{5}
$$

where the initial state vector is assumed to have a distribution $\hat{x}_k(t_0^k|t_{-1}^k) \sim \mathcal{N}(x_{k0}, \Sigma_{k0})$.

We denote $t_{\text{end}}^k = \max_i\{t_i^k | t_i^k \le t\}$ as the latest sample receiving time before the current time for the k-th point. Based on Eq. (5), the state $\hat{x}_k(t|t_{\text{end}}^k)$ and covariance $\Sigma_k(t|t_{\text{end}}^k)$ estimation at current time t for the k-th point is:

$$
\begin{aligned}
&\hat{x}_k(t|t_{\text{end}}^k) = A(\Delta t_{\text{end}}^k)\hat{x}_k(t_{\text{end}}^k|t_{\text{end}}^k)\\
&\Sigma_k(t|t_{\text{end}}^k) = A(\Delta t_{\text{end}}^k)\Sigma_k(t_{\text{end}}^k|t_{\text{end}}^k)A^T(\Delta t_{\text{end}}^k) + W(\Delta t_{\text{end}}^k)
\end{aligned}
\tag{6}
$$

where $\Delta t_{\text{end}}^k = t - t_{\text{end}}^k$, and $A(\Delta t_{\text{end}}^k)$, $W(\Delta t_{\text{end}}^k)$ are calculated according to Eq. (4).

The covariance $\Sigma_k(t|t_{\text{end}}^k)$ in Eq. (6) indicates the uncertainty of estimation. It can be seen that the larger the Δt_{end}^k, the more uncertainties in the estimation.

The rotation matrix R_t and the translation vector $\mathbf{t}$ of target can be estimated with the state $\{\hat{x}_k\left(t|t_{\text{end}}^k\right)\}_{k=1}^n$ and covariance $\{\Sigma_k(t|t_{\text{end}}^k)\}_{k=1}^n$ of n points on that target given by Eq. (6). For those measurements provided by 2D camera, refer [10] for further details.

2.2 Maximum Likelihood Transformation Estimation

Three frames will be used in this section. The first one is the world frame $\mathscr{F}_w$. The second one is the robot end-effector frame $\mathscr{F}_r$, which is fixed on the TCP of robot end-effector. The third one is the target frame $\mathscr{F}_t$. The feature points are assumed to have given coordinates $\{s_k\}_{k=1}^n$ in the target frame, where n is the number of the feature points.

Given the state estimates $\{\hat{x}_k\left(t|t_{\text{end}}^k\right)\}_{k=1}^n$ in $\mathscr{F}_w$ and covariances $\{\Sigma_k(t|t_{\text{end}}^k)\}_{k=1}^n$, we extract their position part as $\{z_k\}_{k=1}^n$ and $\{\Sigma_z^k\}_{k=1}^n$, and estimate the rotation matrix R_t and the translation vector $\mathbf{t}$ at current time t.

Given the distribution $z_k \sim \mathcal{N}\left(R_t s_k + \mathbf{t}, \Sigma_z^k\right)$, the likelihood function of the parameters R_t and $\mathbf{t}$ is:

$$
p(Z|R_t, \mathbf{t}) = \prod_{k=1}^n \frac{|\Sigma_z^k|^{-\frac{1}{2}}}{(2\pi)^{\frac{3}{2}}} e^{-\frac{1}{2}(z_k - R_t s_k - \mathbf{t})^T (\Sigma_z^k)^{-1}(z_k - R_t s_k - \mathbf{t})}
$$

Then the maximum likelihood estimation problem can be formulated as:

$$
\begin{aligned}
e(R_t, \mathbf{t}) = \min_{R_t \in SO(3), \mathbf{t}} \sum_{k=1}^n &(z_k - R_t s_k - \mathbf{t})^T (\Sigma_z^k)^{-1} \times\\
&(z_k - R_t s_k - \mathbf{t})
\end{aligned}
\tag{7}
$$

where R_t is in the special orthogonal group $SO(3)$.

Take the derivative of Eq. (7) with respect to $\mathbf{t}$, and set it equals to zero, we get:

$$
\mathbf{t}^* = \left(\sum_{k=1}^n (\Sigma_z^k)^{-1}\right)^{-1} \left(\sum_{k=1}^n (\Sigma_z^k)^{-1}(z_k - R_t^* s_k)\right) \tag{8}
$$

where R_t^* denotes the optimal R_t. However, the analytical solution of R_t^* cannot be found by simply substituting Eq. (8) into $e(R_t, \mathbf{t})$ and solving the first order optimal condition. A numerical approach for this weighted least squares problem can be found in [11]. An alternative solution is to approximate $(\Sigma_z^k)^{-1}$ with $\alpha_k I$ by minimizing the induced 2-norm $||(\Sigma_z^k)^{-1} - \alpha_k I||_2$.

The optimal α_k can be represented as:

$$\begin{aligned}\alpha_k &= \underset{\alpha_k}{\operatorname{argmin}} ||(\Sigma_z^k)^{-1} - \alpha_k I||_2 \\ &= \underset{\alpha_k}{\operatorname{argmin}} \sigma_{\max}\left((\Sigma_z^k)^{-1} - \alpha_k I\right) \\ &= \underset{\alpha_k}{\operatorname{argmin}} \max\{|\lambda_{1,k} - \alpha_k|, |\lambda_{2,k} - \alpha_k|, |\lambda_{3,k} - \alpha_k|\} \\ &= (\lambda_{\max,k} + \lambda_{\min,k})/2\end{aligned} \tag{9}$$

where $\sigma_{\max}(\bullet)$ denotes the maximum singular value, and $\lambda_{j,k}$ denotes the j-th eigenvalue of $(\Sigma_z^k)^{-1}$.

With this approximation, $\mathbf{t}^*$ can be computed by:

$$\mathbf{t}^* \approx \frac{1}{\sum_{k=1}^n \alpha_k}\left(\sum_{k=1}^n \alpha_k (z_k - R_t^* s_k)\right) \approx \bar{z}_w - R_t^* \bar{s}_w \tag{10}$$

where $\bar{z}_w$ and $\bar{s}_w$ denote the weighted mean of $\{z_k\}_{k=1}^n$ and $\{s_k\}_{k=1}^n$.

By substituting Eq. (10) into Eq. (7), we get R_t^* as:

$$\begin{aligned}R_t^* &= \underset{R_t \in SO(3)}{\operatorname{argmin}} \sum_{k=1}^n \alpha_k (\tilde{z}_k - R_t \tilde{s}_k)^T (\tilde{z}_k - R_t \tilde{s}_k) \\ &= \underset{R_t \in SO(3)}{\operatorname{argmin}} \sum_{k=1}^n -2\alpha_k \tilde{z}_k^T R_t \tilde{s}_k + \tilde{s}_k^T \underbrace{R_t^T R_t}_{I} \tilde{s}_k \\ &= \underset{R_t \in SO(3)}{\operatorname{argmax}} \sum_{k=1}^n \alpha_k \tilde{z}_k^T R_t \tilde{s}_k \\ &= \underset{R_t \in SO(3)}{\operatorname{argmax}} \operatorname{trace}(R_t \tilde{S}_\lambda \tilde{Z}_\lambda^T)\end{aligned} \tag{11}$$

where

$$\begin{aligned}&\tilde{z}_k = z_k - \bar{z}_w \qquad \tilde{s}_k = s_k - \bar{s}_w \\ &\tilde{Z}_\lambda = [\sqrt{\alpha_1}\tilde{z}_1, \sqrt{\alpha_2}\tilde{z}_2, \cdots, \sqrt{\alpha_n}\tilde{z}_n] \\ &\tilde{S}_\lambda = [\sqrt{\alpha_1}\tilde{s}_1, \sqrt{\alpha_2}\tilde{s}_2, \cdots, \sqrt{\alpha_n}\tilde{s}_n]\end{aligned}$$

If we define $\Sigma_{sz} = \tilde{S}_\lambda \tilde{Z}_\lambda^T$, then the optimal R_t for Eq. (11) is:

$$R_t^* = U_l \operatorname{diag}\left(1, 1, \cdots, 1, \det(U_l U_r^T)\right) U_r^T \tag{12}$$

where U_l and U_r are unitary matrices from SVD of Σ_{sz}, i.e. $\Sigma_{sz} = U_l D U_r^T$.

From Eq. (10), one can find that the k-th feature point is weighted by: $w_k = \frac{\alpha_k}{\sum_{j=1}^n \alpha_j}$, which means the weight of a particular feature point is influenced by the eigenvalues of the corresponding covariance inverse, and the larger the eigenvalues of the covariance inverse for the k-th point, the larger the weight w_k. Furthermore, based on Eq. (6), the covariance of the k-th point is effected by uncertainties such as the packet loss, noise, and irregular sampling time. Moreover, the more severe the sensor physics, the smaller the eigenvalues of the covariance inverse. Therefore, the proposed algorithm can reduce the influence of sensor physics by decreasing those weights with larger uncertainties.

3 VISUAL TRACKING ALGORITHM

3.1 Tracking Controller Design

The objective of the tracking controller is to drive the translational translation error e_p and the orientation error e_o to zero, where $e_p = \mathbf{t} - p_r$ and $e_o = \int \omega_t dt - \int \omega_r dt$, with $\mathbf{t}$ and p_r denoting the translation of $\mathscr{F}_t$ and $\mathscr{F}_r$ in $\mathscr{F}_w$, and ω_t, ω_r denoting the angular velocity of $\mathscr{F}_t$ and $\mathscr{F}_r$ in $\mathscr{F}_w$.

In [12], cross-product of axes vectors of $\mathscr{F}_r$ and $\mathscr{F}_t$ is employed to represent the orientation difference between them. This method requires several assumptions, which result in low tracking bandwidth. On the other hand, a unit quaternion $\xi = \left[\eta, \varepsilon^T\right]^T \in \mathbb{R}^4$ with $||\xi||^2 = \eta^2 + \varepsilon^T\varepsilon = 1$ can represent rotation between two frames. The transformation between the quaternion and the rotation matrix is described in [13].

The quaternion error from $\mathscr{F}_r$ to $\mathscr{F}_t$ can be represented by corresponding unit quaternions $\xi_r = \left[\eta_r, \varepsilon_r^T\right]^T$ and $\xi_t = \left[\eta_t, \varepsilon_t^T\right]^T$ as:

$$\begin{aligned}\tilde{\xi} = \xi_t \otimes \xi_r^* &= \begin{bmatrix}\eta_t \\ \varepsilon_t\end{bmatrix} \otimes \begin{bmatrix}\eta_r \\ -\varepsilon_r\end{bmatrix} \\ &= \begin{bmatrix}\tilde{\eta} \\ \tilde{\varepsilon}\end{bmatrix} = \begin{bmatrix}\eta_r\eta_t + \varepsilon_r^T\varepsilon_t \\ \eta_r\varepsilon_t - \eta_t\varepsilon_r + S(\varepsilon_r)\varepsilon_t\end{bmatrix}\end{aligned} \tag{13}$$

where $\otimes$ denotes the quaternion multiplication, $\xi_r^* = \left[\eta_r, -\varepsilon_r^T\right]^T$ denotes the conjugate of ξ_r, and $S(\varepsilon_r)$ is cross-product operator of ε_r.

Furthermore, we employ the quaternion rate $\dot{\xi}_t$ to calculate the angular velocity of $\mathscr{F}_t$ with respect to $\mathscr{F}_w$, that is:

$$\omega_t = 2\begin{bmatrix}-\varepsilon_t^T \\ \eta_t I - S(\varepsilon_t)\end{bmatrix}^T \dot{\xi}_t \tag{14}$$

Based on [6], the necessary and sufficient condition for $\mathscr{F}_r$ and $\mathscr{F}_t$ coinciding is $\tilde{\varepsilon} = 0$, therefore, the orientation error e_o can be replaced by $\tilde{\varepsilon}$ and the joint velocity command can be calculated by:

$$\dot{q}_{\text{cmd}} = J^\dagger(q_r)\begin{bmatrix}\dot{\mathbf{t}} + K_p e_p \\ \omega_t + K_o\tilde{\varepsilon}\end{bmatrix} \tag{15}$$

where $J^{\dagger}(q_r)$ denotes the pseudo inverse of robot Jacobian matrix, and $K_p \succ 0$ and $K_o \succ 0$ are the translational gain and the orientation gain, respectively.

The global asymptotic stability of Eq. (15) can be proved from kinematic perspective by assuming $\dot{q}_r = \dot{q}_{\text{cmd}}$, which infers that the joint velocity command $\dot{q}_{\text{cmd}}$ can be instantly followed by low level servo controller. The detailed derivation is presented in [6].

In reality, the instantaneous following assumption can not be satisfied because of the robot dynamics:

$$M(q_r)\ddot{q}_r + C(q_r,\dot{q}_r)\dot{q}_r + N(q_r,\dot{q}_r) = \tau_r \tag{16}$$

where $\tau_r \in \mathbb{R}^6$ is the vector of actuator torques, $M(q_r) \in \mathbb{R}^{6\times 6}$ is the inertia matrix, $C(q_r,\dot{q}_r) \in \mathbb{R}^{6\times 6}$ is the centripetal-Coriolis matrix and $N(q_r,\dot{q}_r) \in \mathbb{R}^6$ represents the gravity and other external forces applying on the robot.

With the consideration of dynamics Eq. (16), we define the term $\upsilon = \dot{q}_{\text{cmd}} - \dot{q}_r$, and design a dynamic controller based on Eq. (15) and robot dynamics:

$$\begin{aligned}\tau_{\text{cmd}} =& M(q_r)\ddot{q}_{\text{cmd}} + C(q_r,\dot{q}_r)\dot{q}_{\text{cmd}} + N(q_r,\dot{q}_r) + \\ & K_v\upsilon + J^T(q_r)\begin{bmatrix} e_p \\ \tilde{\varepsilon}\end{bmatrix}\end{aligned} \tag{17}$$

where $K_v \succ 0$ is another gain matrix. The derivation and the global asymptotic stability proof of Eq. (17) can be accomplished by following a similar approach as [14].

3.2 Parameter Uncertainties

With parameter uncertainties, the system could be unstable if the gain matrices K_p, K_o, and K_v are improperly tuned. In this section, the system stability with dynamics parameter uncertainties is analyzed and gain matrices design is discussed.

The torque command with nominal parameters is denoted as:

$$\begin{aligned}\hat{\tau}_{\text{cmd}} =& \hat{M}(q_r)\ddot{q}_{\text{cmd}} + \hat{C}(q_r,\dot{q}_r)\dot{q}_{\text{cmd}} + \hat{N}(q_r,\dot{q}_r) + \\ & K_v\upsilon + J^T(q_r)\begin{bmatrix} e_p \\ \tilde{\varepsilon}\end{bmatrix}\end{aligned} \tag{18}$$

where $\hat{M}(q_r)$, $\hat{C}(q_r,\dot{q}_r)$ and $\hat{N}(q_r,\dot{q}_r)$ represent nominal parameters.

Therefore, the torque mismatch $\tilde{\tau} = \tau_{\text{cmd}} - \hat{\tau}_{\text{cmd}}$ can be represented as:

$$\tilde{\tau} = \tilde{M}(q_r)\ddot{q}_{\text{cmd}} + \tilde{C}(q_r,\dot{q}_r)\dot{q}_{\text{cmd}} + \tilde{N}(q_r,\dot{q}_r)$$

where $\tilde{\tau}$ is bounded with assumptions of bounded motion command and parameter mismatch, i.e.

$$\begin{aligned}||\tilde{\tau}|| \le& ||\tilde{M}(q_r)|| \cdot ||\ddot{q}_{\text{cmd}}|| + ||\tilde{C}(q_r,\dot{q}_r)|| \cdot ||\dot{q}_{\text{cmd}}|| + \\ & ||\tilde{N}(q_r,\dot{q}_r)|| =: B_\tau\end{aligned} \tag{19}$$

By taking the time derivative of the Laypunov function candidate:

$$V = \frac{1}{2}e_p^T e_p + (1-\tilde{\eta})^2 + \tilde{\varepsilon}^T\tilde{\varepsilon} + \frac{1}{2}\upsilon^T M(q_r)\upsilon$$

and combining with Eq. (14) and the time derivative of Eq. (13), we have:

$$\dot{V} = -e_p^T K_p e_p - \tilde{\varepsilon}^T K_o \tilde{\varepsilon} - \upsilon^T K_v \upsilon + \upsilon^T \tilde{\tau} \tag{20}$$

Rewrite Eq. (20) by defining the error vector $\zeta = \left[e_p^T, \tilde{\varepsilon}^T, \upsilon^T\right]^T$ as:

$$\dot{V} = -\zeta^T Q \zeta + \zeta^T \tilde{\tau}_A \le -\lambda_{\min}(Q)||\zeta||^2 + B_\tau||\zeta||$$

where the augmented gain matrix is defined by $Q = \text{diag}(K_p, K_o, K_v)$, $\lambda_{\min}(Q)$ is the minimum eigenvalue of Q, and $\tilde{\tau}_A = \left[0^T, 0^T, \tilde{\tau}^T\right]^T$.

$\dot{V}$ is negative as long as the following condition is satisfied:

$$||\zeta|| > \frac{B_\tau}{\lambda_{\min}(Q)} =: R_\tau$$

We adopt the uniform ultimate boundedness [15], and conclude that the error vector is bounded within R_τ.

Note that the parameter uncertainties in industrial manipulators are usually small (i.e. $||\tilde{M}(q_r)|| \ll 1$, $||\tilde{C}(q_r,\dot{q}_r)|| \ll 1$, $||N(q_r,\dot{q}_r)|| \ll 1$). In addition, by properly increasing the eigenvalues of Q, the dynamic controller given by Eq. (18) will be stable (the tracking error is bounded in a small region). An adaptive controller is also provided in [14] to realize asymptotic tracking.

Based on the precision requirement R_τ and the boundary of parameter uncertainties, one can find a lower bound for gain matrices. In practice, however, the eigenvalues of Q should not be extremely large, since the estimated target pose and the measured robot pose might not be smooth, and the time derivate to these measurements enlarge the non-smoothness. In this case, if the eigenvalues of gain matrices are too large, the joint velocity command in Eq. (15) and torque command in Eq. (17) will be quite noisy, and eventually cause the divergence of the tracking system.

Finally, the proposed visual tracking algorithm is summarized in Algorithm 1 below.

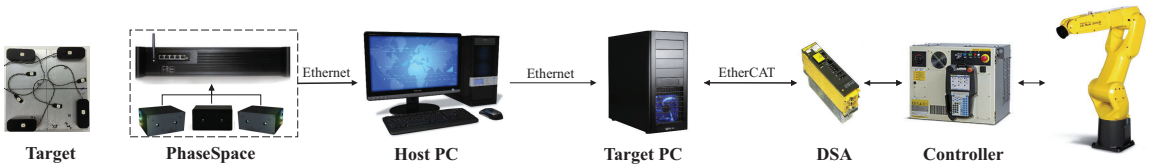


FIGURE 1. OVERALL EXPERIMENTAL SETUP

Algorithm 1 Visual Tracking Algorithm

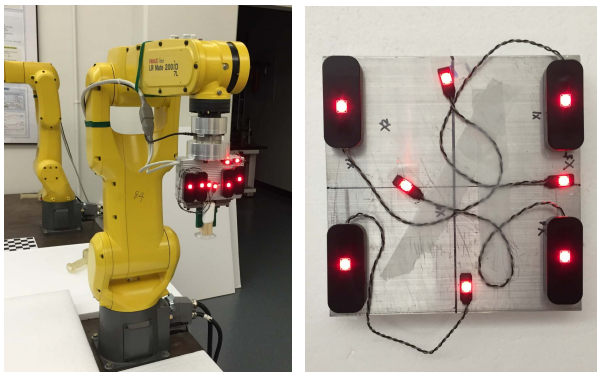
Input: $\{x_{k0}\}_{k=1}^{n}$, $\{\Sigma_{k0}\}_{k=1}^{n}$, W_c, V_c, K_p, K_o, K_v
while $t < t_{\text{Final}}$ **do**
  for k from 1 to n **do**
    Model the k-th feature point by Eq. (4)
    if measurement is updated at t_{end}^k **then**
      Estimate $\hat{x}_k(t_{\text{end}}^k|t_{\text{end}}^k)$, $\hat{\Sigma}_k(t_{\text{end}}^k|t_{\text{end}}^k)$ by Eq. (5)
    end if
    Predict $\hat{x}_k(t|t_{\text{end}}^k)$ and $\hat{\Sigma}_k(t|t_{\text{end}}^k)$ by Eq. (6)
  end for
  Estimate R_t and $\mathbf{t}$ by Eq. (12) and Eq. (10)
  Convert R_t to ξ_t using the method in [13]
  Compute $\tilde{\varepsilon}$ and ω_t by Eq. (13) and Eq. (14)
  Generate $\dot{q}_{\text{cmd}}$ or τ_{cmd} by Eq. (15) or Eq. (17)
end while

4 EXPERIMENTAL RESULTS

4.1 System Overview

The overall experimental setup is shown in Fig. 1. The PhaseSpace motion capture system [16] is used as a vision sensor, and the experiment is implemented on FANUC LRMate 200iD/7L industrial manipulator. For simplification, the feature points are represented by markers, as shown in Fig. 2. To mimic a general low cost commercial vision sensor, the sampling rate of PhaseSpace is down-sampled to 30 Hz, while the controller reference generating rate is 125 Hz, and 30% of the vision data are discarded to mimic packet loss. The vision data is first sent to the host PC, and is further transmitted to target PC. During the transmission, other sensor physics such as latency and irregular sampling time are involved. The pose of the target is calculated by Kalman filter and maximum likelihood method in target PC. Then, the tracking controller computes the trajectory reference and the torque command and send it to the digital servo adapter (DSA). The robot controller connects DSA and robot and controls the robot according to signals from DSA.

The remaining of this section first verifies the pose estimation algorithm by comparing the estimated rotation matrix and translation vector with the true values. Then the pose estimation algorithm and tracking controller are combined to realize real-time visual tracking. The implementation details and results are described in the following sections.



(a) Markers configuration for pose estimation algorithm verification (b) The target in visual tracking experiment

FIGURE 2. ENVIRONMENT COLLISION CHECKING

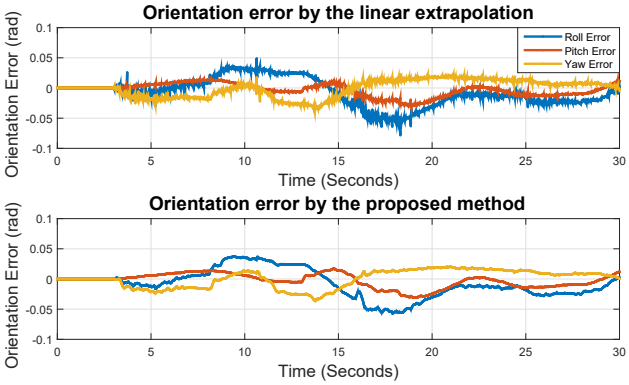


FIGURE 3. POSE ESTIMATION VALIDATION. THE TOP PLOT SHOWS THE RESULT OF THE LINEAR EXTRAPOLATION METHOD (BASELINE), AND THE BOTTOM PLOT SHOWS THE RESULT OF THE PROPOSED METHOD

4.2 Pose Estimation Verification

The pose estimation algorithm is verified in this part of the experiment. First, seven markers are fixed on the robot end-effector, as shown in Fig.2(a). Then, the vision sensor captures the positions of markers when the robot is moving along a predefined trajectory. The robot pose calculated from robot kine-

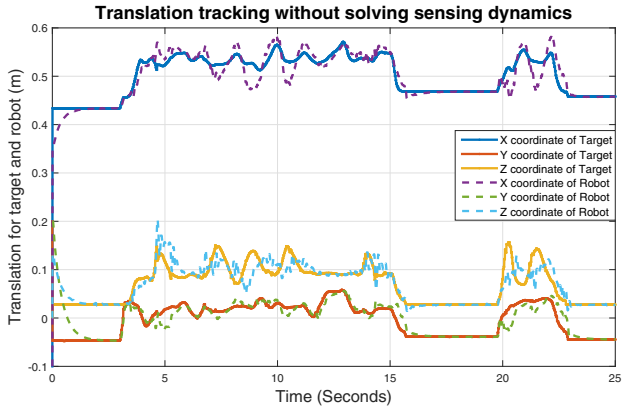


FIGURE 4. TRANSLATION COMPARISON BETWEEN TARGET AND END-EFFECTOR OF ROBOT WITHOUT CONSIDERING SENSOR PHYSICS

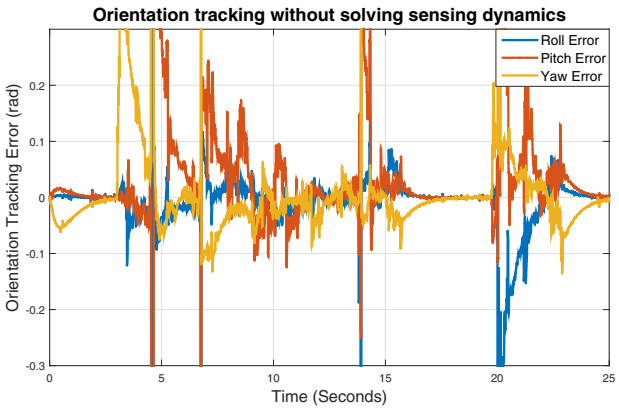


FIGURE 5. ORIENTATION COMPARISON BETWEEN TARGET AND END-EFFECTOR OF ROBOT WITHOUT CONSIDERING SENSOR PHYSICS

matics and encoder feedback is treated as the true values. Orientation errors by 1) linear extrapolation, 2) proposed algorithm are plotted in Fig. 3. The first method estimates positions of markers based on constant velocity extrapolation when losing packets, and then employ least squares method [8] to obtain pose estimation. It is seen that all the Euler angle errors are bounded within 0.05 rad. In addition, the proposed method has smaller noise than the linear extrapolation. This is because 1) the Kalman filter and the maximum likelihood can reduce the noise effect, and 2) the weights of different markers are adjusted dynamically by their relative estimation uncertainties.

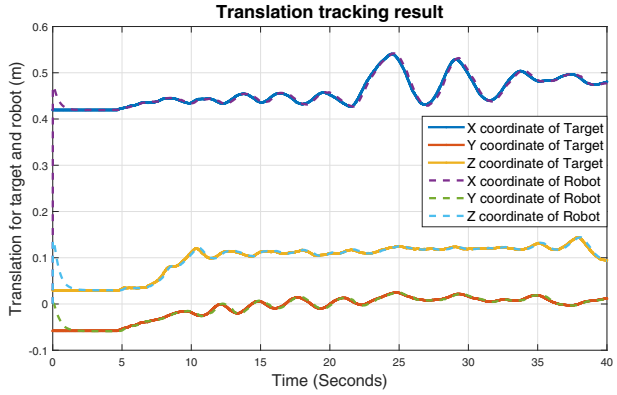


FIGURE 6. TRANSLATION COMPARISON BETWEEN TARGET AND END-EFFECTOR OF ROBOT WITH PROPOSED ALGORITHM

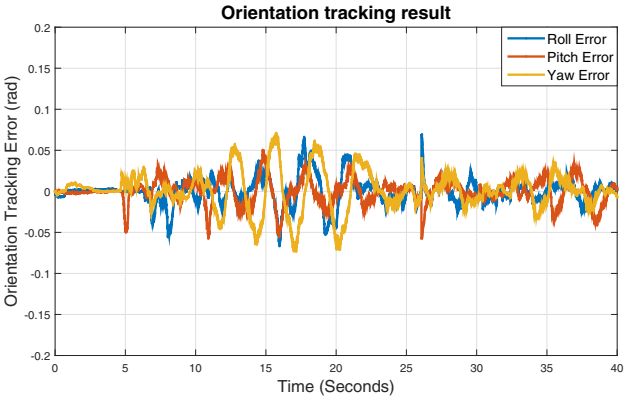


FIGURE 7. ORIENTATION COMPARISON BETWEEN TARGET AND END-EFFECTOR OF ROBOT WITH PROPOSED ALGORITHM

4.3 Visual Tracking Results

This part of the experiment validates the tracking algorithm by combining the proposed pose estimation with the tracking controller. The target in this part is an aluminum board attached with seven markers, which is shown in Fig. 2(b). The trajectory of the target is generated by human operation. The position and orientation are estimated by the proposed algorithm and used by tracking controller to generate command for the robot.

Figure 4 shows the translation tracking result without considering sensor physics. The bold lines represent the actual translation of the target based on the raw measurements of PhaseSpace, while the dash lines represent the translation of robot output. First, the low sampling rate, noise, packet loss, and irregular sampling time increase the uncertainties of feature points and cause the discontinuity of the target reference, which result in robot vibration, overshooting and large derivation. In the meantime, the transmission latency can introduce time delay to the system. It is seen that the translation tracking error can be over

0.05m (at around 8.5 sec), and the robot has large overshooting, latency and vibration. Also, the orientation error in Fig. 5 can be over 0.3 rad.

Figure 6 compares the translation between the robot frame and the target frame considering the sensor physics using the proposed algorithm. The bold lines represent the actual translation of the target based on the raw measurements of PhaseSpace, while the dash lines represent the translation of robot output. The trajectories between $0-2$ sec shows the global asymptotic stability of the closed loop system. It is seen that the translation tracking error is bounded within 0.003 m. Fig. 7 shows the orientation error between the target frame and the robot end-effector frame by the proposed algorithm. It is seen that the orientation errors in all dimensions are bounded within 0.07 rad and the average error is 0.013 rad.

5 CONCLUSIONS

In this paper, a visual tracking control framework considering sensing dynamics is proposed. First, an varying rate Kalman filter was used to estimate the positions and the corresponding error covariances of markers. Then, given the distributions from Kalman prediction, the maximum likelihood technique was employed to estimate the target pose, with the importance of different markers weighted by their uncertainties. Experimental results indicated that the proposed method could reduce the effects of irregular sampling time, packet loss, low sampling rate, noise, and latency. In addition, a dynamic tracking controller was designed to realize stable position and orientation tracking. The controller stability taking into account model uncertainty was discussed. The effectiveness of the proposed tracking scheme was validated on a 6-DOF industrial robot. It was shown that the robot implemented with the proposed algorithm can achieve globally asymptotically stable real-time visual tracking.

In future, the authors would compare the proposed pose estimation method with the method of unscented Kalman filter, where the state in transition equation is composed of the pose of the target. The comparison will include computational efficiency, convergence rate and ease of implementation. Furthermore, the PhaseSpace sensor will be replaced by an affordable RGB camera and the feature extraction procedure will be conducted directly from raw RGB images.

ACKNOWLEDGMENT

This work was supported by FANUC Corporation.

REFERENCES

[1] Corke, P. I., et al., 1996. *Visual Control of Robots: high-performance visual servoing*. Research Studies Press Baldock.

[2] Xue-bo, J., Jing-jing, D., and Jia, B., 2012. "Target tracking of a linear time invariant system under irregular sampling". *International Journal of Advanced Robotic Systems,* ***9***.

[3] Lin, C.-Y., Wang, C., and Tomizuka, M., 2013. "Visual tracking with sensing dynamics compensation using the expectation-maximization algorithm". In American Control Conference (ACC), 2013, IEEE, pp. 6281–6286.

[4] Jeon, S., Tomizuka, M., and Katou, T., 2009. "Kinematic kalman filter (kkf) for robot end-effector sensing". *Journal of dynamic systems, measurement, and control,* ***131***(2), p. 021010.

[5] Luh, J. Y., Walker, M. W., and Paul, R. P., 1980. "Resolved-acceleration control of mechanical manipulators". *Automatic Control, IEEE Transactions on,* ***25***(3), pp. 468–474.

[6] Yuan, J. S., 1988. "Closed-loop manipulator control using quaternion feedback". *Robotics and Automation, IEEE Journal of,* ***4***(4), pp. 434–440.

[7] Campa, R., and Camarillo, K., 2008. *Unit quaternions: A mathematical tool for modeling, path planning and control of robot manipulators*. INTECH Open Access Publisher.

[8] Umeyama, S., 1991. "Least-squares estimation of transformation parameters between two point patterns". *IEEE Transactions on Pattern Analysis & Machine Intelligence*(4), pp. 376–380.

[9] Li, X. R., and Jilkov, V. P., 2000. "Survey of maneuvering target tracking: dynamic models". In AeroSense 2000, International Society for Optics and Photonics, pp. 212–235.

[10] Lin, C.-Y., Wang, C., and Tomizuka, M., 2014. "Pose estimation in industrial machine vision systems under sensing dynamics: A statistical learning approach". In Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE, pp. 4436–4442.

[11] Lin, C.-Y., and Tomizuka, M., 2015. "Estimating rigid transformation with correlated observations". In ASME 2015 Dynamic Systems and Control Conference, American Society of Mechanical Engineers, pp. V002T23A002–V002T23A002.

[12] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G., 2009. *Robotics: modelling, planning and control*. Springer Science & Business Media.

[13] Diebel, J., 2006. "Representing attitude: Euler angles, unit quaternions, and rotation vectors". *Matrix,* ***58***, pp. 15–16.

[14] Xian, B., de Queiroz, M. S., Dawson, D., and Walker, I., 2004. "Task-space tracking control of robot manipulators via quaternion feedback". *Robotics and Automation, IEEE Transactions on,* ***20***(1), pp. 160–167.

[15] Lewis, F., Jagannathan, S., and Yesildirak, A., 1998. *Neural network control of robot manipulators and non-linear systems*. CRC Press.

[16] PhaseSpace Impulse X2. `http://www.phasespace.com/`.