Robotics and Computer-Integrated Manufacturing 31 (2015) 91–100

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Online robot calibration based on hybrid sensors using Kalman Filters

Guanglong Du, Ping Zhang*, Di Li

South China University of Technology, PR China

ARTICLE INFO

Article history:
Received 17 November 2013
Received in revised form
22 July 2014
Accepted 10 August 2014
Available online 15 September 2014

Keywords:
Online robot calibration
Kinematic identification
Inertial measurement unit
Kalman Filter
Factored Quaternion Algorithm

ABSTRACT

This paper presents an online robot self-calibration method based on an inertial measurement unit (IMU) and a position sensor. In this method, a position marker and an IMU are required to rigidly attach to the robot tool, which makes it possible to obtain the position of the manipulator from the position sensor and the orientation from the IMU in real time. An efficient approach which incorporates Kalman Filters (KFs) to estimate the position and the orientation of the manipulator is proposed in this paper. Using these pose (orientation and position) estimation methods will result in improving the reliability and accuracy of pose measurements. Finally, an Extended Kalman Filter (EKF) is used to estimate kinematic parameter errors. Compared with the existing self-calibration methods, the greatest advantage of this method is that it does not need any complex steps, such as camera calibration, corner detection and laser tracking, which makes the proposed robot calibration procedure more autonomy in a dynamic manufacturing environment. What's more, reduction in complex steps leads to improving the accuracy of calibration. Experimental studies on a GOOGOL GRB3016 robot show that this proposed method has better accuracy, convenience and effectiveness.

1. Introduction

Because of the manufacturing and assembly tolerance, the actual kinematic parameters of a robot deviate from their nominal values, which is referred to as kinematic errors. The kinematic errors would result in the robot tool's errors if the nominal kinematics were used to estimate the poses of the robot. With the restriction of cost, the kinematic calibration is an effective way to improve the absolute accuracy of robots. Nowadays, calibration tasks use a lot of measurement techniques like coordinate measuring machines, laser tracking interferometer systems, and inexpensive customized fixtures [1]. These systems are not only very expensive, but also not friendly to use or with small working volume. A system which is used in a dynamic environment is expected to perform calibration without any external expensive calibration apparatus and elaborate setups, which means self-calibration.

Self-calibration techniques can be classified into two kinds: (1) redundant sensor approach and (2) motion constraint approach.

There is a self-calibration method for parallel mechanism with a case study on the Stewart platform which is proposed by Zhuang in [2]. He used the forward and inverse kinematics with six rotary encoders for three objective functions of parameter identification. Khalil and Besnard [3] installed two orthogonally allocated inclinometers to the tool to calibrate the Stewart platform except the redundant sensors which are mentioned above. However, there are some limitations of these methods. One of them is that some kinematic parameters orthogonally are not independent of the error models and the position and/or orientation of the tool on the platform cannot be calibrated.

For the other approach, that is the motion constraint approach, the mobility of the resultant system will be lower than its inherent degrees-of-sensing by fixing one or more passive joints or constraining partial DOF of the manipulator so that the calibration algorithm can be performed [4]. Park et al. [5] lowered the mobility of the tool of a serial manipulator and performed self-calibration by using only the inherent joint sensors in the manipulator. And this idea was used and extended to calibrate a robot system with a hand-mounted instrumented stereo camera [6]. However, the position and/or orientation of the tool on the platform cannot be calibrated, and some parameter errors related to the locked passive joints may become unobservable in the calibration algorithm because of the mobility constraints.

To solve these limitations, advances in robot calibration allow the researchers to use a hand-mounted camera to calibrate a robot instead of using measurements from passive joints or imposing mechanical constraints. Compared to those mechanical measuring devices, this camera system costs less and it is easier to use and more accurate. The traditional vision-based methods [7] to calibrate a robot require the precise 3D fixtures measured in a

* Corresponding author.
E-mail address: pzhang@scut.edu.cn (P. Zhang).

http://dx.doi.org/10.1016/j.rcim.2014.08.002
0736-5845/

reference coordinate system and the procedure is inconvenient, time consuming and it may not be feasible for some applications. The self-calibration methods [8,9] assume that the camera is rigidly attached to the robot tool. Closed-loop method "Virtual closed kinematic chain", which is proposed in [10], uses the joint angle measurements already in the robot can be considered self-calibrating. A method uses laser to capture robot position data to model the stiffness of the manipulator [11] and predict kinematic parameters [12]. O'Brien et al. [13] used a magnetic motion to capture robot data to estimate the kinematic parameters. Du et al. [14] used two pose sensors to calibrate the parameters. However, particle filter is inefficiency. Rauf et al. [15] used a vision-based measuring device and a pose measurement device for kinematic calibration, respectively. Du et al. [16] employed a continuous data capture method by using a camera to estimate the kinematic parameters. However, these approaches have a limitation, i.e., the calibration is completed off-line. The optimization technique was based on the measured positions of the tool. The parameter error was minimized in the measured positions, but the error increased in positions that are very different. Moreover, the parameter error increased while the robot withstood different loads. When the robot is used in high-temperature or high-pressure environment, such as deep sea or outer space, the shape of the robot links are easy to change. Therefore, online calibration is an indispensable method to rectify the kinematic parameters.

In this paper, we propose an original approach of online robot calibration by using an IMU and a position sensor to measure the robot poses. In our method, an IMU and a position marker are required to rigidly attach to the robot tool (Fig. 1) to measure the robot poses in real time. Fig. 2 shows the outline of the proposed method. The key steps of kinematic parameter calibration are kinematic modeling, pose measurement, parameter identification. Standard Denavit–Hartenberg (D–H) [17] is used as kinematic modeling because it has a minimal representation for the common normal between two revolute links [4]. For measuring the pose, an IMU and a position marker are attached to the robot tool to measure the orientation and the position of the tool. By using a set of measured poses, KFs are used to estimate the accurate poses. A Jacobian matrix, which represents how each kinematic parameter error influences the variance deviation between the theoretical and measured pose is formulated for parameter identification. EKF is iterated to estimate parameter errors by using the estimated poses. Then, parameter errors are used to correct the kinematic parameters. Unlike some existing self-calibration methods, the described method does not require special complex steps such as camera calibration or laser tracking. Moreover, this method does not require the robot to make the motion for obtaining the measurements, which makes our method more efficient.

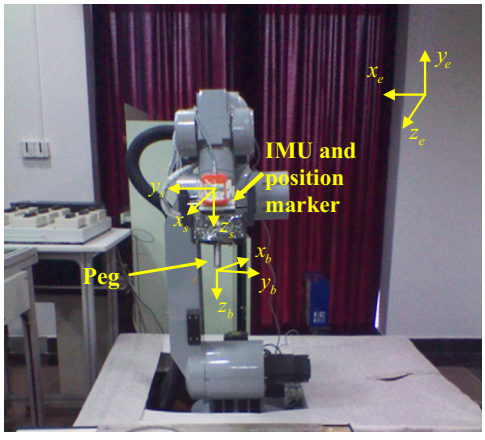

Fig. 1. Structure of the system.

The remainder of the paper is organized as follows. Section 2 provides kinematic modeling for the serial robot. In Section 3, a method of the pose measurement is detailed. Parameters identification algorithm based on EKF is proposed in Section 4. Finally, the experimental results are shown in Section 5 and we conclude the paper in Section 6.

2. Kinematic modeling

A robot kinematic model relates the robot joint coordinate with the pose of the robot tool. A robot kinematic model should meet the following rules for the kinematic-parameter identification [18].

1) Completeness: the robot kinematic model should have enough parameters to define any possible deviation from the nominal values.
2) Continuity: any small changes in the structure of the robot must correspond to small changes in kinematic parameters [18].
3) Minimality: the kinematic model must include only a minimal number of parameters [2].

Many researchers have found suitable kinematic models for robot since 1980s, such as the Hayati et al. models [19], the Veitschegger and Wu's model [20], the Stone and Sanderson's S-model [21], and the Zhuang et al. model [22]. Standard Denavit–Hartenberg (D–H) convention is the most widely used to describe the robot kinematics. The error models of D–H are not continuous for robots that possess parallel joint axes (Fig. 2). To avoid the singularity of D–H convention, D–H modeling or Hayati modeling convention were used, respectively. The singularity-free calibration model prevents the use of a single minimal modeling convention which can be used to identify all possible robot parameters.

The robot tool position and the orientation are defined according to the controller conventions. Through consecutive homogeneous transformations from the base coordinate to the robot tool coordinate, the kinematic equation can be defined as

$$T_N^0 = T_N^0(\boldsymbol{v}) = T_1^0 T_2^1 \ldots T_N^{N-1} = \prod_{i=1}^{N} T_i^{i-1} \tag{1}$$

where T_i^{i-1} is the translation matrix from $i-1$ coordinate to i coordinate, N is the number of joints (Fig. 3). $\boldsymbol{v} = [v_1^{\mathrm{T}} v_2^{\mathrm{T}} \ldots v_N^{\mathrm{T}}]^{\mathrm{T}}$ is the parameter vector for the robot, and v_i^{T} is the link parameter

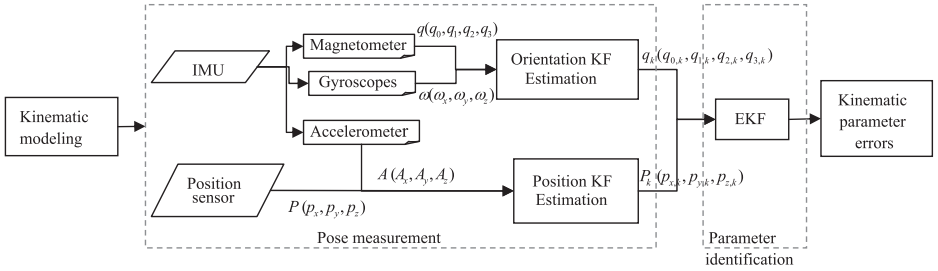

Fig. 2. Outline of the proposed method.

vector for the joint i, which includes the joint errors

$$v_i = v_i' + \Delta v_i \tag{2}$$

v_i' is the nominal value vector for the joint i, $\Delta v = [\Delta v_1^{\mathrm{T}} \Delta v_2^{\mathrm{T}} \ldots \Delta v_N^{\mathrm{T}}]^{\mathrm{T}}$ is the link parameter error vector for the joint i. The exact kinematic equation is

$$K_N^0 = K_N^0(\boldsymbol{v}) = K_1^0 K_2^1 \ldots K_N^{N-1} = \prod_{i=1}^{N} K_i^{i-1} \tag{3}$$

and

$$K_i^{i-1} = T_i^{i-1}(v_i' + \Delta v_i) = T_i^i + \Delta T_i \tag{4}$$

Taking the joint variables into consideration, thus

$$K_N^0 = T_N^0 + \Delta T, \quad \Delta T = \Delta T(\boldsymbol{u},\ \Delta v) \tag{5}$$

where $\boldsymbol{u} = \left[\theta_1^{\mathrm{T}} \theta_2^{\mathrm{T}} \ldots \theta_N^{\mathrm{T}}\right]^{\mathrm{T}}$ is a vector of joint variables. Thus K_N^0: $\Re^n \times \Re^N$ is a function of $\boldsymbol{u}$ and Δv.

3. Pose measurement

3.1. Factored Quaternion Algorithm (FQA)

The FQA, which is based on Earth gravity and magnetic field measurements, is used to estimate the orientation of a rigid body [12]. But this algorithm is applied for static or slow-moving rigid body only. To make it applicable for relatively large linear accelerations, KF fusion algorithm is used together with angular rate information to estimate the orientation of dynamic body (either slow-moving or fast-moving) in the next section [23].

In our application, a sensor module, which uses an inertial measurement unit, is employed to determine the orientation (roll, pitch and yaw) of the robot End-Effector (EE). The IMU sensor consists of one three-axis accelerometer, two two-axis gyroscopes and one three-axis magnetometer.

Three frames are defined as follows: body frame $x_b y_b z_b$, sensor frame $x_s y_s z_s$ and Earth-fixed frame $x_e y_e z_e$. The sensor frame $x_s y_s z_s$ corresponds to the axes of three orthogonally mounted accelerometers/magnetometers. Because the sensor is rigidly attached to the robot EE, the body frame $x_b y_b z_b$ is assumed to coincide with the sensor frame $x_s y_s z_s$. The Earth-fixed frame $x_e y_e z_e$ follows the North-East-Down (NED) convention, where x_e points north, y_e points east and z_e points down. The IMU measures its own orientation (roll, pitch and yaw). Define the rotation ϕ about the x_e axis as the roll, the rotation θ about y_e axis as the pitch and the rotation ψ about z_e axis as the yaw. According to Euler's theorem [24] of finite rotations, the conversion from Euler angle to quaternion is

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos(\phi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \sin(\phi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\sin(\theta/2)\cos(\psi/2) + \sin(\phi/2)\cos(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\phi/2)\sin(\theta/2)\cos(\psi/2) \end{bmatrix} \tag{6}$$

And the four Euler parameters are constrained as [25]

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \tag{7}$$

where q_0 is a scalar and (q_1, q_2, and q_3) is a vector. So the direction cosine matrix M_s^e from the sensor frame to the Earth-fixed frame is represented as

$$M_s^e = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_0 q_2 + q_1 q_3) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_0 q_1 + q_2 q_3) & q_0^2 - q_1^2 - q_2^2 - q_3^2 \end{bmatrix} \tag{8}$$

3.2. Kalman Filter

The state is estimated in KFs using measurements, system model and measurement model in two steps, which are prediction and update [26]. For a time variant system and measurement model, a state space dynamic equation should be considered as

$$x_k = f_k(x_{k-1}, u_{k-1}, b_{k-1}) \tag{9}$$

$$z_k = h_k(x_k, v_k) \tag{10}$$

where t_k is the iteration time represented by the subscript k, x_k is the state from time t_{k-1} to time t_k, f_k is the state transition function, u_{k-1} is the deterministic input, b_{k-1} is the process noise is, z_k is the measurement, h_k is the measurement function, and v_k is the measurement noise. The measurement and process noises can be assumed to be zero-mean Gaussian when the system is linear so that a KF is used. Then, the rewritten form of Eqs. (9) and (10) becomes

$$x_k = \Phi_k \cdot x_{k-1} + \Gamma_k \cdot u_{k-1} + b_{k-1} \tag{11}$$

$$z_k = H_k \cdot x_k + v_k \tag{12}$$

where Φ_k is the system transition matrix for time t_{k-1} to time t_k, Γ_k is the input matrix at time t_k, and H_k is the measurement matrix at time t_k. The posterior density function (PDF) with a mean and covariance can be estimated by the KF when the system can be expressed in the forms of Eqs. (11) and (12). Therefore, the prediction and update steps for KF are as follows:

Prediction:

Predicted state:

$$\hat{x}_k^- = \Phi_k \cdot \hat{x}_{k-1}^+ + \Gamma_k \cdot u_{k-1} \tag{13}$$

Prediction covariance:

$$P_k^- = \Phi_k \cdot P_{k-1}^+ \cdot \Phi_k^{\mathrm{T}} + Q_{k-1} \tag{14}$$

Update:

Kalman gain:

$$K_k = P_k^- \cdot H_k^{\mathrm{T}} \cdot \left[H_k \cdot P_k^- + H_k^{\mathrm{T}} + R_k\right]^{-1} \tag{15}$$

Estimated covariance:

$$\hat{P}_k = [1 - K_k \cdot H_k]^{-1} \cdot P_k^- \tag{16}$$

Estimated state:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \cdot (z_k - H_k \cdot \bar{x}_k) \tag{17}$$

3.3. Orientation KF

Because both gyroscope and magnetometer have white noise and random walk, we use Kalman Filter to estimate the state x of IMU from a set of noisy and incomplete measurements [22]. The differential equation of the quaternion q with respect to time t is

$$\begin{bmatrix} \partial q_0/\partial t \\ \partial q_1/\partial t \\ \partial q_2/\partial t \\ \partial q_3/\partial t \end{bmatrix} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ w_x/2 \\ w_y/2 \\ w_z/2 \end{bmatrix} \tag{18}$$

where w_x, w_y, and w_z are the angular velocity components of IMU in x_s, y_s, and z_s axes. Since the state x_k includes the quaternion states and the angular velocities, x_k has the following form:

$$x_k = \begin{bmatrix} q_{0,k} & q_{1,k} & q_{2,k} & q_{3,k} & w_{x,k} & w_{y,k} & w_{z,k} \end{bmatrix} \tag{19}$$

where $q_{0,k}$, $q_{1,k}$, $q_{2,k}$, $q_{3,k}$, $w_{x,k}$, $w_{y,k}$ and $w_{z,k}$ are the quaternion states and the angular velocities at time k. From Eq. (18), the state-transition

matrix is [27,28]

$$\Phi_{ori}=\begin{bmatrix}1&0&0&0&-q_{1,k}\cdot\Delta t/2&-q_{2,k}\cdot\Delta t/2&-q_{3,k}\cdot\Delta t/2\\0&1&0&0&q_{0,k}\cdot\Delta t/2&q_{3,k}\cdot\Delta t/2&q_{2,k}\cdot\Delta t/2\\0&0&1&0&q_{3,k}\cdot\Delta t/2&q_{0,k}\cdot\Delta t/2&-q_{1,k}\cdot\Delta t/2\\0&0&0&1&-q_{2,k}\cdot\Delta t/2&q_{1,k}\cdot\Delta t/2&q_{0,k}\cdot\Delta t/2\\0&0&0&0&1&0&0\\0&0&0&0&0&1&0\\0&0&0&0&0&0&1\end{bmatrix} \tag{20}$$

where Δt is sampling time. Let Γ_{ori} be the zero matrix because there is no control inputs. We use angular velocities to estimate the quaternion state, so the process noise vector is

$$w_k=\begin{bmatrix}0&0&0&0&w_x&w_y&w_z\end{bmatrix}^{\mathrm{T}} \tag{21}$$

where w_x, w_y and w_z are the process noises of the angular velocity. Because we use calibrated gyroscopes to measure angular velocities, the observation matrix H is

$$H_{ori}=\begin{bmatrix}0^{n\times p}&I^{n\times n}\end{bmatrix} \tag{22}$$

where n is the number of angular velocities vector and p is the number of the quaternion. In order to satisfy (7), the determined quaternion q_k at time k should be normalized by

$$q_k=\begin{bmatrix}q_{0,k}/M&q_{1,k}/M&q_{2,k}/M&q_{3,k}/M\end{bmatrix}$$
$$M=\sqrt{q_{0,k}^2+q_{1,k}^2+q_{2,k}^2+q_{3,k}^2} \tag{23}$$

3.4. Position KF

KF is used in the position estimating process to estimate P which is the state of the position from a set of noisy measurements. There are six measurements are available in this estimation: three acceleration components in the tool frame from the IMU and three position components in the local frame., The direction cosine matrix M_{H2S} from the tool frame to the local frame, can be presented as

$$M_{H2S}=\begin{bmatrix}m_{X_x}&m_{Y_x}&m_{Z_x}\\m_{X_y}&m_{Y_y}&m_{Z_y}\\m_{X_z}&m_{Y_z}&m_{Z_z}\end{bmatrix} \tag{24}$$

The acceleration of the tool in the local frame can be expressed as

$$\dot{V}_x=m_{X_x}\cdot A_x+m_{Y_x}\cdot A_y+m_{Z_x}\cdot A_z$$
$$\dot{V}_y=m_{X_y}\cdot A_x+m_{Y_y}\cdot A_y+m_{Z_y}\cdot A_z$$
$$\dot{V}_z=m_{X_z}\cdot A_x+m_{Y_z}\cdot A_y+m_{Z_z}\cdot A_z-|g_l| \tag{25}$$

where $|g_l|$ is the magnitude of the local gravity vector and (A_x, A_y, A_z) is the acceleration measurement component in each axis in the tool frame. The velocity components (V_x, V_y, V_z) in each axis in the shoulder frame can be defined as

$$V_x=\dot{p}_x$$
$$V_y=\dot{p}_y$$
$$V_z=\dot{p}_z \tag{26}$$

From Eqs. (25) and (26), the state x_k' of the position KF is expressed as

$$x_k'=[p_{x,k},V_{x,k},A_{x,k},p_{y,k},V_{y,k},A_{y,k},p_{z,k},V_{z,k},A_{z,k}] \tag{27}$$

It involves values of variables at the time k. According to Eqs. (25) and (26), the state-transition matrix A_k' is defined as [27]

$$\Phi_{pos}=\begin{bmatrix}1&t&m_{X_x}\cdot t^2/2&0&0&m_{Y_x}\cdot t^2/2&0&0&m_{Z_x}\cdot t^2/2\\0&1&m_{X_x}\cdot t&0&0&m_{Y_x}\cdot t&0&0&m_{Z_x}\cdot t\\0&0&1&0&0&0&0&0&0\\0&0&m_{X_y}\cdot t^2/2&1&t&m_{Y_y}\cdot t^2/2&0&0&m_{Z_y}\cdot t^2/2\\0&0&m_{X_y}\cdot t&0&1&m_{Y_y}\cdot t&0&0&m_{Z_y}\cdot t\\0&0&0&0&0&1&0&0&0\\0&0&m_{X_z}\cdot t^2/2&0&t&m_{Y_z}\cdot t^2/2&1&t&m_{Z_z}\cdot t^2/2\\0&0&m_{X_z}\cdot t&0&0&m_{Y_z}\cdot t&0&1&m_{Z_z}\cdot t\\0&0&0&0&0&0&0&0&1\end{bmatrix} \tag{28}$$

Since the acceleration measurements are affected by gravitational force and the Z-axis is parallel to the gravity vector, the system input matrix is

$$\Gamma_{pos}\cdot u_{k-1}'=[0,0,0,0,0,0,-|g_l|\cdot t^2/2,-|g_l|\cdot t,0]^{\mathrm{T}} \tag{29}$$

We use the acceleration to estimate the position state, so the process noise vector is

$$w_k'=\begin{bmatrix}0,0,w_x',0,0,w_y',0,0,w_z\end{bmatrix}^{\mathrm{T}} \tag{30}$$

where (w_x', w_y', w_z) is the process noise of the tool acceleration.

Since the 3D camera and the IMU are calibrated and initialized, the observation matrix H' for the position estimation is

$$H_{pos}=\begin{bmatrix}1&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0\\0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&1\end{bmatrix} \tag{31}$$

So the nine states are observable.

The determined position $P_k(p_{x,k}, p_{y,k}, p_{z,k})$ at time k is the optimal value of the position of robot tool.

4. Parameter identification

A. Differential kinematics

Kinematic identification is to identify the kinematic model parameters of a robot manipulator by giving a set of pose measurements and the corresponding joint angles readings. The objective of a kinematic identification algorithm is to minimize the difference between the computed and the measured poses.

Assume that the number of measure poses is m, it can be stated that

$$\hat{K}=\hat{K}_N^0=(\hat{K}(\boldsymbol{u}_1,\boldsymbol{v}),\hat{K}(\boldsymbol{u}_2,\boldsymbol{v}),\ldots,\hat{K}(\boldsymbol{u}_m,\boldsymbol{v}))^{\mathrm{T}} \tag{32}$$

and

$$\Delta\hat{T}=\Delta\hat{T}_N^0=(\Delta\hat{T}(\boldsymbol{u}_1,\boldsymbol{v}),\Delta\hat{T}(\boldsymbol{u}_2,\boldsymbol{v}),\ldots,\Delta\hat{T}(\boldsymbol{u}_m,\boldsymbol{v}))^{\mathrm{T}} \tag{33}$$

where $\boldsymbol{u}_i(i=1, 2\ldots m)$ is the vector of joint variables for the ith measure pose.

The objective of the kinematic identification is the computation for the parameter vector $v^*=v'+\triangle v$, which is to minimize the discrepancy between the computed and the measured poses

$$A(v^*,\boldsymbol{u})=B(\boldsymbol{u}) \tag{34}$$

A is the function of pose of $\hat{T}$ and $B(\boldsymbol{u})=(B(u_1),\ B(u_2),\ldots B(u_m))^{\mathrm{T}}$ is the measured function of joint variables $\boldsymbol{u}$.

For each measurement pose $B(u_i)$, it concludes orientation measurement R_i and position measurement P_i, and

$$B(u_i)=\begin{bmatrix} R_i & P_i \\ \mathbf{0} & 1 \end{bmatrix} \tag{35}$$

If the measurement system can provide orientation measurement and position measurement, each pose can formulate six measurement equations. From Eq. (5)

$$A(v^*,\boldsymbol{u})=B(\boldsymbol{u})=A(v,\boldsymbol{u})+C(\Delta v,\boldsymbol{u}) \tag{36}$$

where C is the discrepancy function of the pose components of $\Delta\hat{T}$. Introducing the Jacobian matrix,

$$C(\Delta v,\boldsymbol{u})=J\cdot\Delta v \tag{37}$$

and then

$$C(\Delta v,\boldsymbol{u})=B(\boldsymbol{u})-A(v,\boldsymbol{u}) \tag{38}$$

when using

$$b=B(\boldsymbol{u})-A(v,\boldsymbol{u}) \tag{39}$$

and

$$x=\Delta v \tag{40}$$

Eq. (37)can be rewritten

$$J\cdot x=b \tag{41}$$

To solve the non-linear Eq. (41), the least-squares estimation (LSE) method [29] is a fast and computationally efficient identification algorithm. But LSE is sensitive to the noise. In this paper, EKF, which is detailed in Section 6, is used to fit data to the non-linear model. Most particularly and commonly method in the case that m is much larger than n is the non-linear least-squares [30].

4.1. Estimating parameter errors using Extended Kalman Filter

Initially, the poses of the tool are measured from the IMU and the position sensor. Since uncertainty exists in the measurement, Extended Kalman Filter (EKF) is used as an optimization algorithm and the Jacobian matrices are used to estimate the kinematic errors of D–H parameters from the measured poses values [4].

Since there are four parameters for N revolute joints and four parameters for the transformation from the sensors to the tool, the number of total parameters to be considered is $4(N+1)$. So the predicted state $\hat{x}$ is $4(N+1)$ of the D–H parameters in the prediction step of the EKF. The covariance matrix of the predicted state P is

$$\hat{x}_{k+1|k}=\hat{x}_{k|k}$$

$$P_{k+1|k}=P_{k|k}+Q_k \tag{42}$$

where Q_k is the covariance matrix of the system noise at the kth iteration.

In the observation step of the EKF, Jacobian matrix J, measurement residual $\tilde{y}$, and residual covariance S are calculated as follows:

$$\begin{aligned} J_{k+1}&=\left.\frac{\partial T(x)}{\partial x}\right|_{\hat{x}_{k+1|k}} \\ \tilde{y}_{k+1}&=m_{k+1}-T(\tilde{x}_{k+1|k}) \\ S_{k+1}&=J_{k+1}P_{k+1}J_{k+1}^{\mathrm{T}}+R_{k+1} \end{aligned} \tag{43}$$

where m_k and R_k are the measured pose value and the covariance matrix of measurement noise at the kth iteration. $k+1|k$ means a prior estimate, and $k+1|k+1$ means a posteriori estimate.

In the update step, the state covariance matrix is updated by an optimal Kalman gain K

$$\begin{aligned} K_{k+1}&=P_{k+1|k}J_{k+1}^{\mathrm{T}}S_{k+1}^{-1} \\ \hat{x}_{k+1|k+1}&=\hat{x}_{k+1|k}+K_{k+1}\tilde{y}_{k+1} \\ P_{k+1|k+1}&=(I-K_{k+1}J_{k+1})P_{k+1|k} \end{aligned} \tag{44}$$

where I is the identity matrix. The norm value of the state vector is calculated for every iteration once the updating procedure is completed. Note that if Q and R are set to zero, then EKF simply reduces to the Newton–Raphson method (Fig. 3).

5. Experiment

5.1. Experiments environment

To verify the above method, A GOOGOL GRB3016 robot was used in this experiment. In the experiment, the robot could self-calibrate online in the working status. There were four steps in the self-calibration procedure:

1) Data collection: the poses of the tool and the corresponding joint angles were captured with the robot tool moving in different positions.
2) Manipulator pose estimation: the poses of the manipulator were estimated via the KFs from the obtained data.
3) Kinematic parameters identification: the manipulator kinematic parameters were identified from the estimated poses and the joint angles.
4) Calibration accuracy assessment: 3D pose errors were used to verify the calibration accuracy.

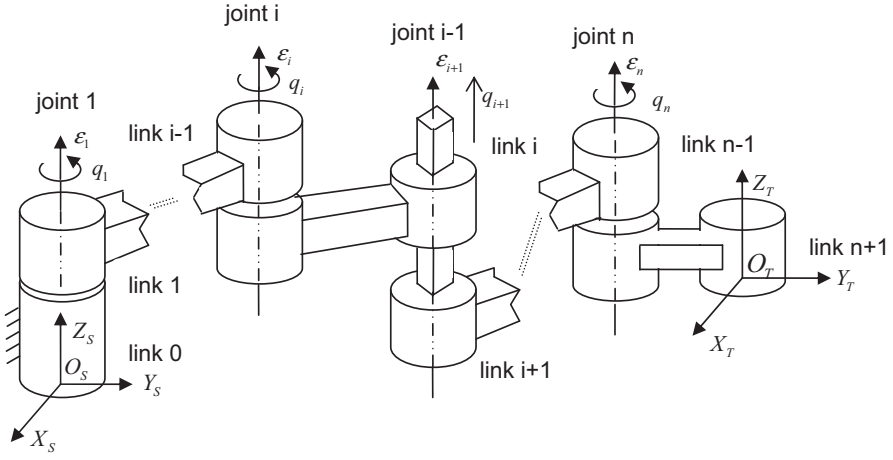

Fig. 3. Forward kinematics of a N-DOF robot.

Table 1 lists the nominal robot link parameters of the robot, which was chosen as the initial condition for the above kinematic identification algorithm. Fig. 4 shows the kinematic diagram of the GOOGOL GRB3016 robot.

Note that a GOOGOL GRB3016 robot with 6-DOF needs 24 geometric parameters to be modeled. From Eq. (36), each 3D robot pose provides 6 model equations. So a unique of computation of the 24 parameters needs 4 pose measurements at least. And more pose measurements will decrease the calibration errors. But limited by the measurement accuracy, the calibration errors intend to be stable as the increase of the pose measurements. The matrices R and Q for KF can be determined by the adaptive method described in [34]. The matrices R and Q for EKF can be calculated through the method of [35].

Two experiments were designed to verify the proposed method. To verify the effectiveness of the proposed algorithm, 100 poses were measured in experiment one. The former 50 poses were used as the calibration set and the latter 50 poses were used as the testing set. The calibration set contained the poses covering the large range of workspace as much as possible.

In order to verify the advantage of the proposed robot calibration method, we have performed a number of Peg-into-hole tests in experiment two. In experiment two, three calibration methods were used to carry out the experiments of peg-into-hole. Our method was initially compared with vision-based robot calibration [16] and laser-based robot calibration [5]. There were 16 holes in the steel plate (Fig. 5) and 16 tests of peg-into-hole were carried out with two methods. The peg was the cylinder with 7.5 mm in radius and 150 mm in long. The radius of the hole was 8 mm. The size of the steel plate was 300 mm × 500 mm. In the step of our method, one IMU (Xsens MTi-100 IMU) and one position sensor were rigidly attached to the robot tool flange to measure the poses of the robot tool. The noise density of gyroscopes was 0.01 °/s/$\sqrt{\text{Hz}}$ and that of the magnetometer was 200 μ$G\sqrt{\text{Hz}}$. Both position sensor and the IMU can be sampled at 100 Hz. Due to the IMU and position marker being light weight and small, they are used to determine the poses of the tool related to the local fixed frame. Fig. 1 shows the position of the tool, tool frame, and local fixed frame. Either a roughly or factory calibrated IMU can be used in this method. There is a temperature drift feature in many commercial IMUs that have been calibrated at the factory. Despite this, this type of IMU is still regarded as just roughly calibrated as it can still suffer from a small calibration error. If the IMU has not been factory-calibrated, they need to be calibrated at room temperature in the factory. As long as the factory's temperature does not change a lot, the gains and biases will not change a large amount in the accelerometers and gyros.

Table 1
The nominal link parameters in D–H model for the GOOGOL GRB3016 robot.

Joints	D–H			
	a	α	d	θ
1	150	$-\pi/2$	250	0
2	570	$-\pi$	0	$-\pi/2$
3	150	$\pi/2$	0	0
4	0	$-\pi/2$	645	0
5	0	$-\pi/2$	0	$-\pi/2$
6	0	0	−200	0

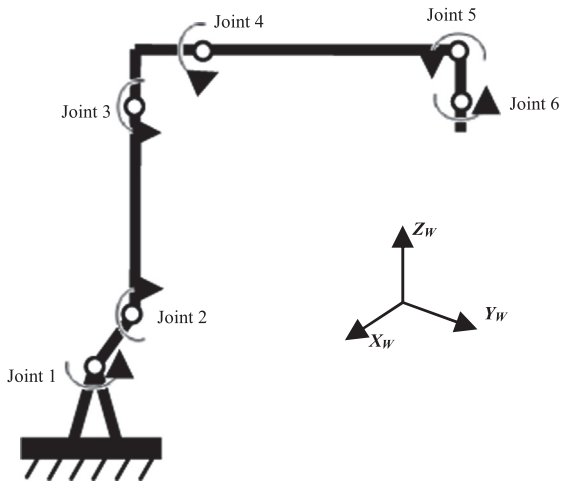

Fig. 4. Skeleton of the GOOGOL GRB3016 robot with coordinate frames in the zero position.

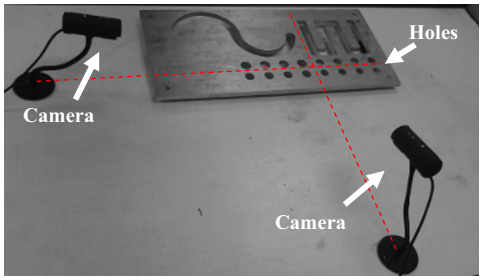

Fig. 5. Steel plate and evaluation system.

A structured laser module is attached to the robot tool following the method of [4]. A stationary camera is utilized to determine the accurate position where the laser came out. The picture elements of the cameras were 1280(H) × 960(V), and the frequency was 22 frame/s. The diffraction length of the laser beams was approximately 4.0 mm. In the step of method [16], a camera, with 1280(H) × 960(V) picture elements and 22 frame/s frequency, was mounted on the robot tool to capture the RGB images of the chessboard. A chessboard pattern, the position of which was unknown in the reference frame, was placed on the platform. The distance, which measured by a caliper, between two adjacent corners of the square were known.

Since the robot will be stopped after it executes a command, in order to improve the accuracy, the system collects the static data of sensors after the robot stops. The coordinates of the center of the hole with respect to the base coordinate system were known. Note that the robot has different kinematic parameter errors in different positions. In order to improve the accuracy, the system made a robot calibration before inserting the peg into each hole.

The robot inverse kinematics method of [31] was used to control the robot. From Eq. (5), every angle is achieved by robot inverse kinematics through the given pose M of end-effector:

$$T_6^0 = M \tag{45}$$

To solve the non-linear Eq. (45), Levenberg–Marquardt (LM algorithm) [30]is used to fit $(\theta_1, \theta_2, \ldots, \theta_6)$ to the non-linear model.

To evaluate the calibration accuracy, 3D position errors between the peg and the hole were proposed. two calibrated camera with 1280 (H) × 960(V) picture elements were used to measure the 3D errors (Fig. 6). In the initiation of the system, the camera measured the center and the axis line of the hole with respect to the camera coordinate system. After the peg was inserted into a hole, two cameras can measure the depth and the direction of insertion by detecting the edge of the peg. And then the system could calculate the center and the axis line of the peg with respect to the camera coordinate system. Let (x_o, y_o, z_o) and (x_t, y_t, z_t) be the centers of the peg and the hole,

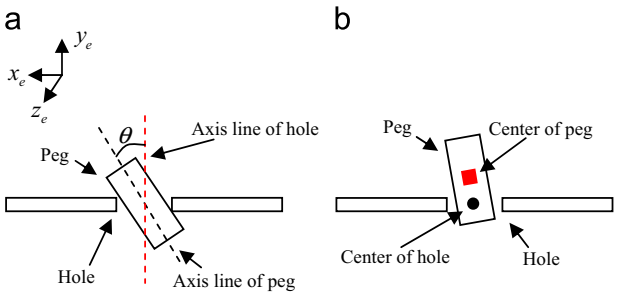

Fig. 6. Definition of 3D errors.

respectively (Fig. 6), the 3D position error e_{Pos} can be written as

$$e_{Pos} = \sqrt{(x_o - x_t)^2 + (y_o - y_t)^2 + (z_o - z_t)^2} \quad (46)$$

Two cameras can measure the inclined angle in the x and z directions, so the orientation error can be defined as

$$e_{ori} = \sqrt{\theta_x^2 + \theta_z^2} \quad (47)$$

where θ_x and θ_z are the inclined angles between the axis line of the hole and the axis line of the peg in the x and z directions.

5.2. Result analysis

Table 2 shows the pose errors of the robot tool between the measured poses and the theoretical poses, which includes the mean error and the maximum error. Before the calibration procedure, the mean errors for the calibration set and the testing set were more than 3.26 mm with the maximum error value of 9.11 mm.

Table 2 presents the comparison between the proposed EKF and measured data. Without calibration, the mean errors of the calibration set and the testing set were 3.26 mm and 4.24 mm in position, 3.28° and 3.69° in orientation. The mean error value for EKF was less than 0.1 mm in position and 0.2° in orientation. The results of the calibration set and the testing set confirmed that EKF had better performance. In addition, the pose errors of the calibration were smaller than that of the testing set. So the calibration set should include the poses that covered the whole workspace as much as possible.

Table 3. shows the estimated error values of the kinematic parameters by using EKF for 200 iterations. Fig. 7. shows that EKF quickly converged to the stable error values after 20 iterations.

Following the initialization step, the system performed a series of tests for peg-into-hole. The KFs processed the measurements and concurrently estimated the state vector. For a 6-DOF robot, the total parameters to be considered is $4\times6+5$ for the method of [5], $4\times(6+1)$ for method [16] and $4\times(6+1)$ for our method. Since each measurement can formulate one measurement equation in method [5].To complete a calibration, method [5] needs at least 29 measurements. Method [16] and our method can formulate six measurement equations from each pose measurement, so method [16] and our method need 5 measurements at least. For the convenience of accuracy comparison, the experiment collected 50 pose measurements for each algorithm in each test. Fig. 8 shows the measured data and the estimation by using KF in our method. The state-estimate errors with 50 pose measurements are shown in Fig. 9. As evident from Fig. 9, even with the noise error for the IMU measurements, the algorithm is still able to attain very accurate estimates of the calibration parameters. The position errors of method [5], method [16] and our method are 0.083 mm, 0.123 mm, and 0.063 mm. The orientation errors of method [5], method [16] and our method are 0.14°, 0.22° and 0.12°. Our method estimated the calibration parameters more accurately.

Table 2
Pose errors before the calibration procedure and after the calibration procedure.

	Position/mm		Orientation/b	
	Mean	Max	Mean	Max
Calibration set				
Measured data	3.26	8.13	3.28	5.67
EKF	0.07	0.12	0.11	0.18
Testing set				
Measured data	4.24	9.11	3.69	5.88
EKF	0.09	0.15	0.15	0.22

Table 3
Estimated parameter errors of 6 DOF robot. Δa (mm), $\Delta\alpha$ (deg), Δd (mm), and $\Delta\theta$ (deg).

	Joints					
	1	2	3	4	5	6
EKF						
Δa	3.596	3.945	2.545	2.957	2.425	2.636
$\Delta\alpha$	−0.589	−0.234	−1.296	−0.385	−0.424	0.654
Δd	3.157	3.321	3.598	3.475	3.235	2.474
$\Delta\theta$	2.385	−0.838	0.454	0.168	−0.325	−0.368

In method [16] and our method, a unique of computation of the 28 kinematic parameters needs 5 pose measurements. In method [5], it needs 29 pose measurements at least. We compared three methods from 5 pose measurements to 50 pose measurements. The results presented in Fig. 10 show that with more pose measurements, the parameter error decreases gradually.

The estimation errors trend slows down after two times of the minimum number of pose measurements.

Compare with method [16], the greatest advantage of our method is that the system does not need to make a more motion to take the photo and complex calculation steps. After the robot executed a command, the robot would stop and the system concurrently obtained the static measurement data from the IMU and the position sensor. Fig. 11 shows the execution time of three methods with 50 pose measurements. The average testing time of method [5], method [16], and our method are 9.145 s, 12.150 s and 4.235 s, respectively. Method [5] requires the robot to do 50 different movements and the static camera will shoot after every movement to calculate the length of two laser beams. It takes about 0.1 s for robot to complete one command and 0.08 s for the camera to take one picture and calculate the length of laser beam. Compared with method [5], method [16] has to spend about 0.1 s to extract corners of the pictures. Therefore, it takes about $(0.24\times50+0.1)$ s for method [16] to accomplish one test. Our algorithm only requires the robot to do different movements. Therefore, it takes about $(0.1\times50+0.1)$ s for our algorithm to accomplish one test. Since our algorithm requires no specific movement (e.g. directing at chessboard or laser tracking) to collect data, the system can collect the measurements when the robot moving towards the target holes. Moreover, the proposed method can estimate the parameter precisely from twelve pose measurements. In fact, the proposed method can accomplish one peg-into-hole test less than 1 s.

6. Discussion

When the working conditions of a manipulator is relatively simple, such as constant temperature, constant load, calibration can be used to identify the kinematic parameters off-line. But

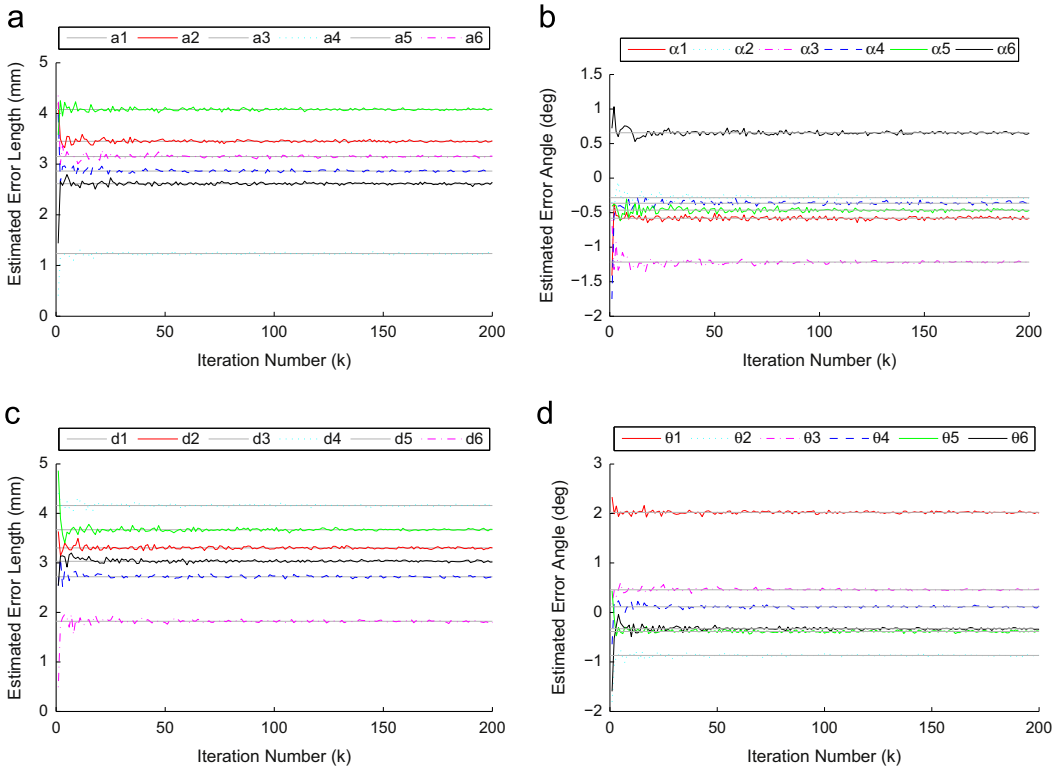

Fig. 7. Estimated D–H parameter errors using EKF.

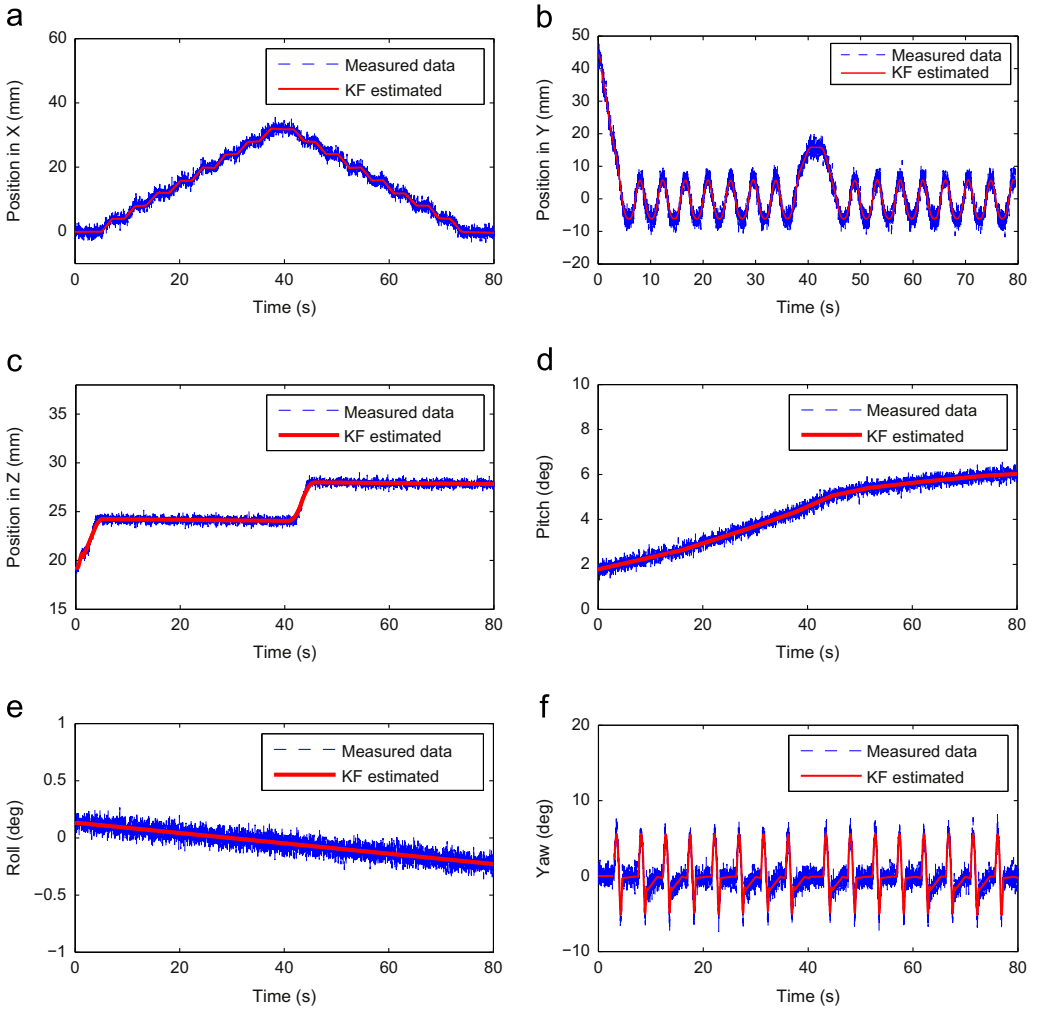

Fig. 8. Measured data and KF estimated data: (a) Position in *X*. (b) Position in *Y*. (c) Position in *Z*. (d) Orientation in pitch. (e) Orientation in roll. (f) Orientation in yaw.

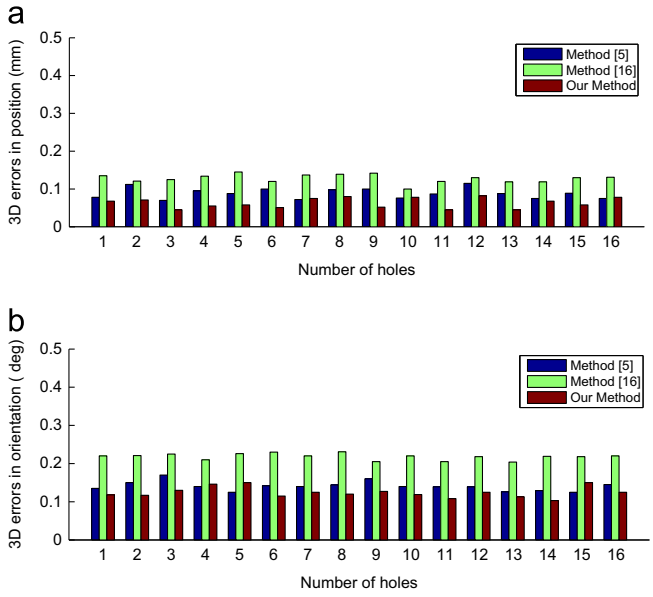

Fig. 9. 3D errors for three methods: (a) Position errors. (b) Orientation errors.

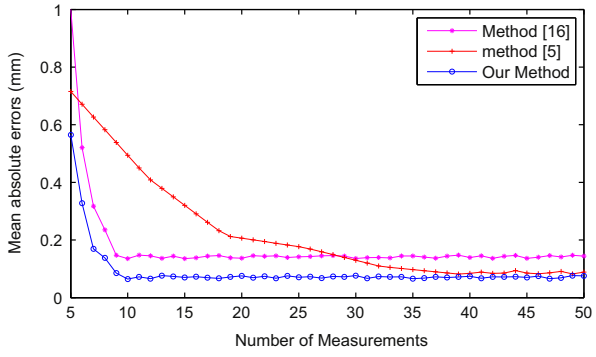

Fig. 10. Mean absolute errors in different number of measurements for three methods.

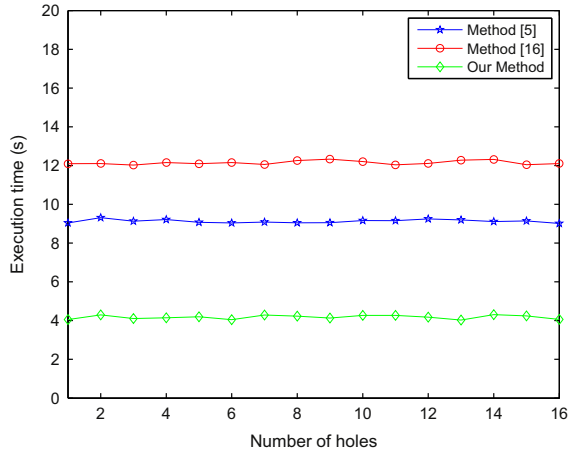

Fig. 11. The comparison in execution time.

calibration works badly in the complex conditions because the kinematic parameters change at any time. Closed-loop control and online calibration can work well in the complex conditions (such as LaserLAB: http://www.wiest-ag.com). Although closed-loop control can tolerance for parameter errors [32], these methods [33] need to repeat adjustments of the robot joints to reach the target point. So closed-loop control methods consume a lot of time for the adjustments and the accuracy of the manipulator depends on the feedback errors. Online calibration can find the position and orientation rapidly after calibrating the kinematic parameters of the robot. Online calibration can be seen as a tradeoff between calibration and closed loop control. If online calibration is used before the manipulation, online calibration simply reduces to calibration. When online calibration is used at any time to adjust the kinematic parameters for precise positioning, online calibration rises to close-loop control. So online calibration is used for the task whose time requirements is prior to accuracy requirements, otherwise, close-loop control is used.

7. Conclusions

A hybrid-sensors-based online autonomous calibration for serial robot has been proposed in this paper. In this approach, an IMU and a position sensor are rigidly attached to the robot tool to estimate the robot poses automatically during the working time. An efficient approach which incorporates FQA and KFs to estimate the poses of the robot tool is presented in this paper. After the robot poses are estimated, the kinematics identification can be carried out by using EKF. Finally, the robot kinematics parameters can be corrected from the identification results in real time. The whole procedure of the robot calibration is automatic and without any manual intervention. The results of the experiments show the good accuracy, convenience and effectiveness of the presented approach.

Compared with the existing expensive and complex approaches, the proposed method is easier to be used and setuped. Compared with the existing self-calibration methods [5,16], the proposed method can conduct the calibration more accurately and in less execution time. In the future work, we will study the approach which can accurately estimate the robot poses without stopping the robot. With the dynamic pose measurements, the robot calibration will be more efficient.

Acknowledgments

Project funded by "China Postdoctoral Science Foundation (NO:2014M550436)", "National Natural Science Foundation of China (Grant No:61403145)" and "the Fundamental Research Funds for the Central Universities (NO:2014ZM0039)".

References

[1] Zhuang H, Yan J, Masory O. Calibration of Stewart platforms and other parallel manipulators by minimizing inverse kinematic residuals. J Robot Syst 1998;15(7):395–405.
[2] Zhuang H. Self-calibration of parallel mechanisms with a case study on Steward platform. IEEE Trans Robot Autom 1997;13(3):387–97.
[3] Khali W, Besnard S. Self calibration of Stewart–Gough parallel robots without extra sensors. IEEE Trans Robot Autom 1999;15(6):1116–21.
[4] Bennett DJ, Hollerbach JM. Autonomous calibration of single-loop closed kinematic chains formed by manipulators with passive endpoint constraints. IEEE Trans Robot Autom 1991;7(5):597–606.
[5] Park I, Lee B, Cho S, Hong Y. Laser-based kinematic calibration of robot manipulator using differential kinematics. IEEE/ASME Trans Mechatron 2012;17(6):1059–67.
[6] Benneet DJ, Hollerbach JM. Autonomous robot calibration for hand-eye coordination. Int J Robot Res 1991;10(5):550–9.
[7] Yang G, Huat Yeo S, Lin W, Chen M. Self-calibration of a biologically inspired 7 DOF cable-driven robotic arm. IEEE/ASME Trans Mechatron 2008;13(1):66–75.

[8] Renaud Pierre, Andreff Nicolas, Lavest Jean-Marc, D-Home Michel. Simplifying the kinematic calibration of parallel mechanisms using vision-based metrology. IEEE Trans Robot 2006;22(1):12–22.
[9] Guanglong Du, Ping Zhang. IMU-based online kinematic calibration of robot manipulator. Sci World J 2013;2013:1–10.
[10] Zhu Y, Hu C, Hu J, Yang K. Accuracy and simplicity oriented self-calibration approach for two-dimensional precision stages. IEEE Trans Ind Electron 2012;60(6):2264–72.
[11] Alici G, Shirinzadeh B. Enhanced stiffness modeling, identification and characterization for robot manipulators. IEEE Trans Robot 2005;21(4):554–64.
[12] Alici G, Jagielski R, Sekercioglu Y, Shirinzadeh B. Prediction of geometric errors of robot manipulators with particle swarm optimization method. Robot Auton Syst 2006;54(12):956–66.
[13] O'Brien James F, Bodenheimer Robert E, Brostow Gabriel J, Hodgins, Jessica K. Automatic Joint parameter estimation from magnetic motion capture data. In: Proceedings of graphics interface; May 2000. p. 53–60.
[14] Guanglong Du and Ping Zhang, Online serial manipulator calibration based on multi-sensory process via extended Kalman and particle filters, IEEE Transactions on Industrial Electronics, Volume 61, Issue 12, http://dx.doi.org/10.1109/TIE.2014.2314051, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6779595.
[15] Rauf A, Pervez A, Ryu J. Experimental results on kinematic calibration of parallel manipulators using a partial pose measurement device. IEEE Trans Robot 2006;22(2):379–84.
[16] Du Guanglong, Zhang Ping. Online robot calibration based on vision measurement. Robot Comput Integr Manuf 2013;29(6):484–92.
[17] Paul R. Robot manipulators: mathematics, programming, and control. Cambridge, MA: MIT; 1982; 50–5.
[18] He Ruibo, Zhao Yingjun, et al. Kinematic-parameter identification for serial-robot calibration based on POE formula. IEEE Trans Robot 2009;26(3):411–23.
[19] Hayati S, Mirmirani M. Improving the absolute positioning accuracy of robot manipulators. J Robot Syst 1985;2(4):397–413.
[20] Veitschegger W, Wu C. Robot accuracy analysis based on kinematics. IEEE Trans Robot Autom 1986;2(3):171–9.
[21] Stone H, Sanderson A. A prototype arm signature identification system. In: Proceedings of the IEEE conference on robotics and automation; April 1987. p. 175–82.
[22] Zhuang H. Kinematic modeling, identification and compensation of robot manipulators. Florida: Florida Atlantic University; 1989 (Ph.D. dissertation).
[23] Corrales JA, Candelas FA, Torres F. Hybrid tracking of human operators using IMU/UWB data fusion by a Kalman filter. In: Proceeding of the 3rd ACM/IEEE international conference on human robot interaction. Amsterdam; 2008. p. 2167–21.
[24] James Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors, matrix, citeseer; 2006.
[25] Goldstein H, Poole C, Safko J. Classical mechanics. MA: Addison-Wesley; 2002; 33–78.
[26] Won SP, Melek WW, Golnaraghi F. Fastening tool tracking system using a Kalman filter and particle filter combination. Meas Sci Technol 2011;22(12).
[27] Won SP, Golnaraghi F, Melek WW. A fastening tool tracking system using an IMU and a position sensor with Kalman filters and a fuzzy expert system. IEEE Trans Ind Electron 2009;56(5):1782–92.
[28] Won SP, Melek WW, Golnaraghi F. A Kalman/particle filter-based position and orientation estimation method using a position sensor/inertial measurement unit (IMU) hybrid system. IEEE Trans Ind Electron 2010;57(5):1787–98.
[29] Gatti G, Danieli G. A practical approach to compensate for geometric errors in measuring arms: application to a six-degree-of-freedom kinematic structure. Meas Sci Technol 2008;19(1).
[30] Dennis JE, Schnabel RB. Numerical methods for unconstrained optimisation and nonlinear equations. New Jersey: Prentice-Hall; 1983.
[31] Lee CSG, Chang Po Rong. Efficient parallel algorithm for robot inverse dynamics computation. IEEE Trans Syst Man Cybern 1986;16(4):532–42.
[32] Cui Xiang. Closed-loop control for a cable-driven parallel manipulator with joint angle feedback, In: Proceedings of the 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM). Wollongong, Australia; 2013, p. 625–630.
[33] Penning Ryan S. Towards closed loop control of a continuum robotic manipulator for medical applications. In: Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Mech. Eng. Dept., Shanghai; 2011. p. 4822–27.
[34] Weidong Ding Jinling Wang, Rizos Chris. Improving adaptive Kalman estimation in GPS/INS integration. J Navigation 2007;60(3):517–29.
[35] Bavdekar VinayA, Deshpande Anjali P, Patwardhan Sachin C. Identification of process and measurement noise covariance for state and parameter estimation using extended Kalman filter. J Process Control 2011;21(4):585–601.