CHESS—Calibrating the Hand-Eye Matrix With Screw Constraints and Synchronization

Krittin Pachtrachai, Francisco Vasconcelos, George Dwyer, Vijay Pawar, Stephen Hailes, and Danail Stoyanov

***Abstract*—Hand–eye calibration is a classic problem in robotics that aims to find the transformation between two rigidly attached reference frames, usually a camera and a robot end-effector or a motion tracker. Most hand–eye calibration techniques require two data streams, one containing the eye (camera) motion and the other containing the hand (robot/tracker) motion, and the classic hand–eye formulation assumes that both data streams are fully synchronized. However, different motion capturing devices and cameras often have variable capture rates and timestamps that cannot always be easily triggered in sync. Although probabilistic approaches have been proposed to solve for nonsynchronized data streams, they are not able to deal with different capture rates. We propose a new approach for unsynchronized hand–eye calibration that is able to deal with different capture rates and time delays. Our method interpolates and resamples the signal with the lowest capture rate in a way that is consistent with the twist motion constraints of the hand–eye problem. Cross-correlation techniques are then used to produce two fully synchronized data streams that can be used to solve the hand–eye problem with classic methods. In our experimental section, we show promising validation results on simulation data and also on real data obtained from a robotic arm holding a camera.**

***Index Terms*—Calibration and identification, kinematics.**

I. Introduction

SENSORS and vision systems are often integrated into robotic platforms to provide extra information of the surroundings which helps in localisation of the workspace [1]. An example in robotics-assisted minimally invasive surgery (RMIS) is stereo-laparoscopic cameras or GPS in mobile vehicles for various types of simultaneous localization and mapping (SLAM) applications [2]. For RMIS, accurate and real-time localisation of an operative site with a laparoscope is an important component towards developing new capabilities in robotic surgery such as autonomy with visual servoing or surgical instrument guidance with augmented reality (AR) [3], [4]. Similarly, in other SLAM applications, robots have to map the surroundings in case interaction is needed or in order to avoid any obstacles in navigating a path [5], [6]. In order to correctly transform positional and orientation data from a sensors' coordinate frame to the robot coordinate frame, the hand-eye transformation that connects these two coordinate systems has to be estimated. When the hand-eye transformation is calibrated and known, it is possible to realise advanced methods for path planning, motion compensation or AR visualisation that rely both on robot kinematics and on sensor measurements [7]. The challenge is that synchronisation of the data streams from different sensors may have variable specifications (sampling rate, data loss, activation time) which can result in temporal signal misalignment as shown in Fig. 1, where asynchronicity between the data streams renders spatially well-posed hand-eye calibration approaches invalid.

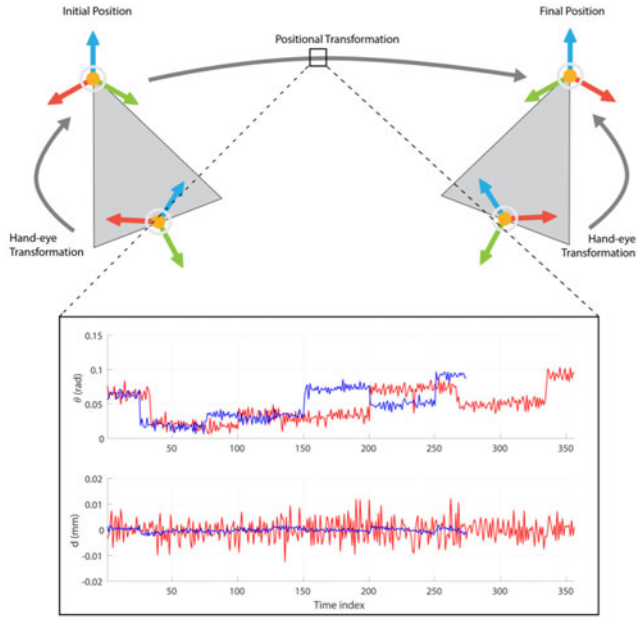

Fig. 1. An example of data capture with different type of sensors. One is the camera and the other is a set of markers detected by a motion tracking system rigidly attached to the camera. Cameras used in robotic systems usually have capture rate around 15–60 fps, while many tracking systems can publish data at the rate higher than 100 fps. The two data streams will be severely asynchronous and some data will be unusable since the rate on the other end is not fast enough, e.g. the red signal and the blue signal are similar but they are asynchronous which creates differences in their shape.

Manuscript received September 10, 2017; accepted January 4, 2018. Date of publication January 31, 2018; date of current version March 21, 2018. This letter was recommended for publication by Associate Editor M. Liu and Editor D. Song upon evaluation of the reviewers' comments. This work was supported by the EPSRC (EP/N013220/1, EP/N022750/1, EP/N027078/1, and NS/A000027/1). *(Corresponding author: Krittin Pachtrachai.)*

K. Pachtrachai, F. Vasconcelos, G. Dwyer, and D. Stoyanov are with the Wellcome/EPSRC Centre for Interventional and Surgical Sciences and the Department of Computer Science, University College London, London WC1E 6BT, U.K. (e-mail: krittin.pachtrachai.13@ucl.ac.uk; f.vasconcelos@ucl.ac.uk; george.dwyer.14@ucl.ac.uk; danail.stoyanov@ucl.ac.uk).

V. Pawar and S. Hailes are with the Department of Computer Science, University College London, London WC1E 6BT, U.K. (e-mail: v.pawar@cs.ucl.ac.uk; s.hailes@cs.ucl.ac.uk).

Digital Object Identifier 10.1109/LRA.2018.2800088

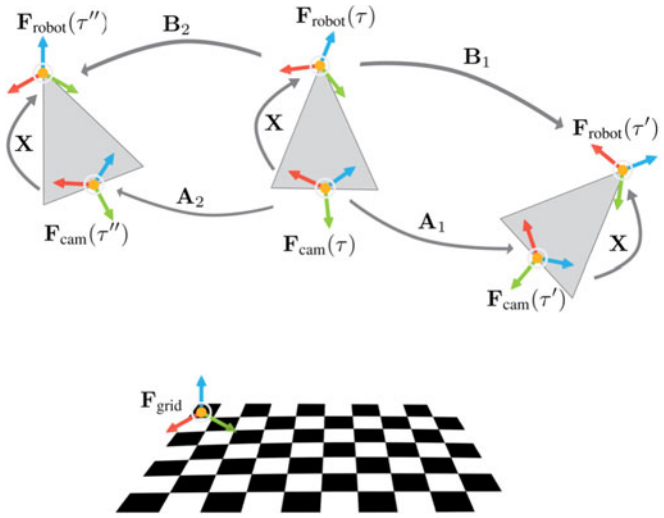

Fig. 2. The schematic shows the transformation loop for hand-eye calibration as shown in (1). The camera is rigidly mounted onto the robot and moved to different poses to capture images of the calibration object corresponding to each robot pose.

There is extensive literature on different hand-eye calibration approaches as the geometric problem dates back to the 1980s with a solution proposed by Shiu and Ahmad [8]. Mathematically, the problem is formulated into the equation $\mathbf{AX} = \mathbf{XB}$ where $\mathbf{A}$ and $\mathbf{B}$ are the relative transformations with respect to different coordinate frames. Several formulations on different domain representations can be developed to tackle the problem e.g. Euclidean group [7], [9], [10], quaternion [11]–[13] and dual quaternion [14]–[16]. Constraints for the calibration optimisation can be developed, such as using epipolar constraints [17], using pure rotational movements [18], performing hand-eye calibration for fewer unknown parameters for SCARA robot [19] or using global optimisation methods [20]. Such numerous approached solve the problem with different numerical stability, but critically they all rely on full synchronisation between the two data streams. But synchronisation is not always possible or easy in practice because sensors or robot operational frequencies or triggers may vary.

Probabilistic approaches have recently been proposed to address the synchronisation problem [21]–[24]. These do not require the priori knowledge of correspondence between two data streams and view data as shifted Dirac delta functions on the special Euclidean group $SE(3)$ [21]. Using the property of this function, the problem can be reformulated into the relation equating the data stream $\mathbf{A}$ and $\mathbf{B}$ convolved with the delta function $\mathbf{X}$. However, this does not solve the problem of different sampling rate because at least one corresponding transformation $\mathbf{A}$ in the pool for any $\mathbf{B}$ is still needed. For example, if we have large sets of $\{\mathbf{A}_i\}$ and $\{\mathbf{B}_j\}$, with random orders, it is necessary to have at least one corresponding $\mathbf{A}_i$ for a $\mathbf{B}_j$ such that the hand-eye equation holds. This is not possible when different sensors whose sampling rates are not the same, which creates more than one transformation on one data stream to have unknown corresponding transformations on the other data stream. Li proposed a solution [24] to synchronise hand-eye data streams using cross-correlation in the frequency domain together with the probabilistic solution to solve hand-eye and robot-world calibration but the approach assumes that the two data have the same length which is not always true in practice where we have time delay and different sampling rate.

In this letter, we introduce a new approach to calibrate the hand-eye matrix when the two data streams have different sampling rates and activation times. Our method can both synchronise the data streams for hand-eye calibration using cross-correlation and recover the missing transformations for lower-sampling rates. We develop our approach using screw constraints leading to a number of advantages:

- Increased calibration accuracy by including a higher number of motions [7] which are recovered by our algorithm based on screw motion;
- Recovery of respective hand-hand transformation for given eye-eye pairs and vice versa;
- Screw motion constraints can handle ambiguity in parallel rotation axes [25];
- Constraints on both rotation and translation components are taken into account in time-delay estimation;
- Compensation for both time-delay and different sampling rates with recovered transformations conforming to the screw constraints.

Our method's formulation consists of data synchronisation using cross-correlation with normalisation and resampling, followed by data recovery using screw constraints which outperforms linear interpolation, followed by the hand-eye solution. Experimentally we demonstrate the performance of our algorithm through extensive experiments with both simulated and real data. We provide comparison with state-of-the-art algorithms in [23]. By solving the hand-eye equation with synchronised relative transformations our method achieved more accurate calibration result than previous efforts.

Notation: Vectors are represented by a lower-case letter with an arrow $\vec{a}$ and skew symmetric matrices of a vector $\vec{a}$ are represented by $[\vec{a}]_\times$, i.e., $\vec{a} \times b$ is equivalent to $[\vec{a}]_\times b$, where $\times$ denotes the cross product between two vectors. Matrices are represented by a bold capital letter, e.g. $\mathbf{K}$. Bold lower-case letters represent quaternions, e. g. $\mathbf{q}$. For more information about quaternions, interested readers can refer to [26]. The rigid transformation between frame i and frame j is denoted by the 4×4 matrix ${}^j\mathbf{T}_i$.

II. Background and Problem Formulation

The hand-eye calibration problem is formulated as

$$\mathbf{A}_i\mathbf{X} = \mathbf{X}\mathbf{B}_i \tag{1}$$

where $\mathbf{A}_i$ is the relative motion of the camera with respect to the coordinate frame locating at the calibration object and $\mathbf{B}_i$ is the relative motion of the robot arm with respect to the coordinate frame locating at the robot base. The relative transformation can be written in terms of the multiplication between two rigid

transformations, e.g.

$$\begin{aligned}\mathbf{A}_1 &= {}^{\text{cam}}\mathbf{T}_{\text{grid}}(\tau)({}^{\text{cam}}\mathbf{T}_{\text{grid}}(\tau'))^{-1}\\ \mathbf{B}_1 &= {}^{\text{robot}}\mathbf{T}_{\text{base}}(\tau)({}^{\text{robot}}\mathbf{T}_{\text{base}}(\tau'))^{-1}\\ \mathbf{A}_2 &= {}^{\text{cam}}\mathbf{T}_{\text{grid}}(\tau)({}^{\text{cam}}\mathbf{T}_{\text{grid}}(\tau''))^{-1}\\ \mathbf{B}_2 &= {}^{\text{robot}}\mathbf{T}_{\text{base}}(\tau)({}^{\text{robot}}\mathbf{T}_{\text{base}}(\tau''))^{-1}\end{aligned} \tag{2}$$

where τ, τ' and τ'' indicate different time instances where two robot configurations are not not same, as shown in Fig. 2. Hand-eye calibration determines the unknown rigid transformation $\mathbf{X} = {}^{\text{cam}}\mathbf{T}_{\text{robot}}$ that links between the two relative motions. If the two relative motions are not synchronised correctly, (1) does not hold.

III. Hand-Eye Calibration Using Unsynchronised Data

The algorithm we propose in this letter consists of synchronisation using cross-correlation to estimate the time-delay between two data streams and then data recovery using a method based on screw constraints for the lower-sampling rate data. The resynchronised data streams are then fed into the hand-eye calibration algorithm developed in [13] to finally determine the hand-eye transformation.

A. Data Synchronisation Using Cross-Correlation

In a synchronised hand-eye problem, A and B motions have two constraints that are independent from the calibration parameters

$$\begin{aligned}\theta_A &= \theta_B\\ d_A &= d_B\end{aligned} \tag{3}$$

where θ_A, θ_B denote the angles of rotation in both frames, and d_A, d_B the amount of translation along the principal axes of rotation. The parameters can be calculated through the following equations:

$$\begin{aligned}\theta &= \arccos\left(\frac{r_{11}+r_{22}+r_{33}-1}{2}\right)\\ d &= \vec{t}^{\,T}\vec{l}\end{aligned} \tag{4}$$

where r_{ij} is an element in a rotation matrix, $\vec{t}$ represents translation vector in a transformation and $\vec{l}$ is a unit vector denoting the principal axis of rotation [27]. This geometric relationship is called screw motion and it is invariant to any rigid transformation, and is thus the same in any reference frames. The complete proof of this relationship is explained by [25].

These constraints are still valid even in the case of insufficient rotational motion, i.e. the obtained motion has a small range of rotational motion. The screw constraint on translation component $d = \vec{t}^{\,T}\vec{l}$ does not consider the angle of rotation, i.e. $\vec{l}$ in (4) that defines the axis of rotation is a unit vector.

Cross-correlation on this information can then be applied to find a point where the similarity between two signals is highest to synchronise the data streams for hand-eye calibration. However, due to different capture rates as shown in the upper graph in Fig. 3, the data requires normalisation and resampling in order to recover the shape of the signal.

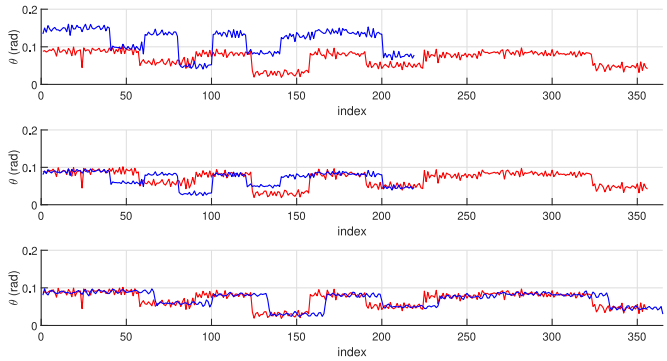

Fig. 3. Degree of rotation that occurs in each motion. The three graphs show the screw information θ_B (blue) in comparison with its corresponding signal θ_A (red) after normalising with sampling frequency and upsampling. The top graph shows the signal before normalisation and upsampling. The middle graph shows the signal after normalisation and the bottom graph shows the signal after upsampling which is the real input for the cross-correlation function.

1) Data Normalisation and Resampling: The hand-eye transformation parameters are derived from the difference between two rigid transformations. Differences are not going to be the same but they will be proportional to each other when the two data streams are not captured at the same sampling rate. Normalisation of the data should be applied before resampling and correlation so the two data streams have the same weight. Normalisation can be performed by:

$$\begin{aligned}\widehat{\theta_B} &= \frac{f_B}{f_A}\theta_B\\ \widehat{d_B} &= \frac{f_B}{f_A}d_B\end{aligned} \tag{5}$$

where f_A and f_B are the capture rate of the data streams $\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ respectively. The result after normalising is shown in the middle graph in Fig. 3. The next step is to resample the data which consists of three procedures: upsample, apply low-pass filter and downsample. The details of the resampling method can be found in [28], [29]. Firstly, the data θ_B in the middle graph is upsampled by a factor of q. This can be done by adding 0's into the signal such that:

$$\widehat{\theta_B}[n] = \begin{cases}\widehat{\theta_B}[\frac{n}{q}], & \frac{n}{q} \in \mathbb{I}^+\\ 0, & \text{otherwise}\end{cases} \tag{6}$$

Then, an anti-aliasing finite impulse response (FIR) filter is applied to the upsampled data which later will be downsampled by the factor of p. The downsampling method discards the data when the index is not a multiple of p. The example of pre-processing data before cross-correlation is shown in the bottom graph.

2) Generalised Cross-Correlation With FFT Pruning and Hilbert Transform: The generalised cross-correlation method (GCC) is a well known approach to estimate time delay between two signals which works well for both complex-valued data and real data. The method is fully described in [30]. In order to

use both screw parameters in the cross-correlation method, the complex-valued data is created with the real part θ and imaginary part d. Therefore, we use GCC to estimate time-delay between the two data streams and we also apply Fourier Transform pruning (FFT pruning) together with Hilbert transform to interpolate the correlation result in case the time-delay is not a factor of a sampling rate. The detail about FFT pruning and Hilbert transform for cross correlation can be found here [31].

B. Data Recovery Using Screw Constraints

The method described in Section III-A produces two synchronised data streams that are adequate inputs for hand-eye calibration. However, the stream with the highest capture rate will still produce motion data that has no correspondence with the other stream, and hence cannot be readily used. Given that calibration accuracy improves with increasing number of input motions [7], we propose a method to interpolate the missing motion correspondences of the stream with the lowest capture rate. Although linear interpolation could be used to recover the missing frames, this does not guarantee that the recovered data conforms to the hand-eye screw constraints (3).

To recover the data, we have to consider (1) and (2) transformations in a dual quaternion form [26]. For any rigid transformation $\mathbf{T} \in SE(3)$, there is a dual quaternion $\mathbf{q} = \mathbf{q}_r + \epsilon \mathbf{q}_t$, such that

$$\mathbf{q}_r = \begin{bmatrix} \sin\frac{\theta}{2}\vec{l} \\ \cos\frac{\theta}{2} \end{bmatrix} \quad \mathbf{q}_t = \begin{bmatrix} \frac{1}{2}(\cos\frac{\theta}{2}\vec{t} + \sin\frac{\theta}{2}(\vec{t}\times\vec{l})) \\ -\frac{1}{2}d\sin\frac{\theta}{2} \end{bmatrix} \tag{7}$$

where θ, d are the screw constraints defined in (3), $\vec{t}$ is a translation component of the transformation and $\vec{l}$ is a unit vector representing the principal axis of rotation of a transformation. Let us define the time instances $\tau_1, \tau_2, \ldots$ as the instances that $\{\mathbf{A_i}\}$ and $\{\mathbf{B_i}\}$ are synchronised and τ_b as the recovered instance for the lower sampling rate data $\{\mathbf{B_i}\}$, we will have

$$\mathbf{q}_{A,i} = \mathbf{q}_a(\tau_i)\mathbf{q}_a(\tau_b)^{-1} \tag{8}$$

$$\mathbf{q}_{B,i} = \mathbf{q}_b(\tau_i)\mathbf{q}_b(\tau_b)^{-1} \tag{9}$$

The only unknown on the right hand side of (8) and (9) is $\mathbf{q}_b(\tau_b)$. We can equate the square of scalar parts of the rotation and translation quaternions of $\mathbf{q}_{A,i}$ and $\mathbf{q}_{B,i}$ together and simplify the problem using matrix multiplication form for a quaternion multiplication [32]. The scalar has to be squared because the value $\cos\frac{\theta}{2}$ can be either negative or positive while still maintaining the equivalent rotation by simply changing the direction of the rotation axis $\vec{l}$. Let the rotation quaternions of $\mathbf{q}_b(\tau_i)$ be $[b_{xi}, b_{yi}, b_{zi}, b_{wi}]$ and the scalar part of the rotation quaternion of $\mathbf{q}_{A,i}$ be a_{ri}, the rotation quaternion of $\mathbf{q}_b(\tau_b)$ can be optimised by

$$\begin{aligned} \phi_r(\mathbf{q}) = \sum_{i=1}^{N} ||(b_{xi}q_x + b_{yi}q_y + b_{zi}q_z + b_{wi}q_w)^2 \\ - (a_{ri})^2|| \\ \text{subject to } \sqrt{q_x^2 + q_y^2 + q_z^2 + q_w^2} = 1 \end{aligned} \tag{10}$$

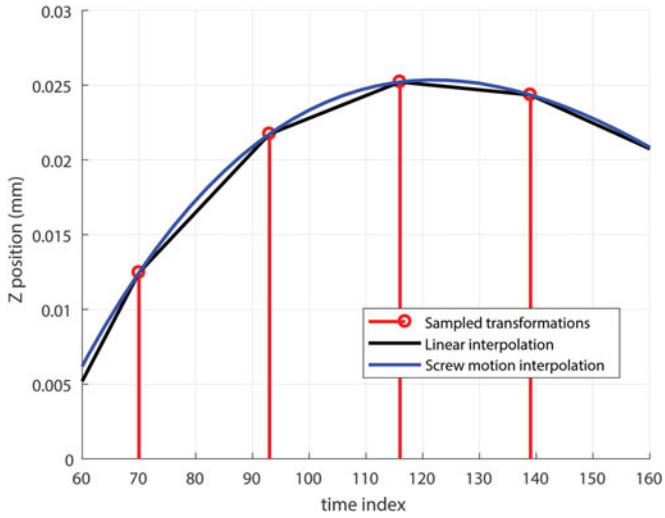

Fig. 4. An example of recovered transformations in between samples. The trajectory recovered using (10) and (12) is clearly different from the linear interpolation that assumes the trajectory between any two samples to be linear which is not always the case.

where i runs through every synchronised time index. For the translation part, optimisation can be run on the translation vector directly. The scalar part of the translation quaternion in (9) can be represented as,

$$q_{bi,t} = \frac{1}{2}(\vec{t_b} - \vec{t_i})^T(\cos\frac{\theta_b}{2}\sin\frac{\theta_i}{2}\vec{l_i} - \sin\frac{\theta_b}{2}\cos\frac{\theta_2}{2}\vec{l_b}) \tag{11}$$

and the optimisation equation is

$$\phi_t(\vec{t}) = \sum_{i=1}^{N} ||q_{bi,t}^2 - a_{ti}^2|| \tag{12}$$

where $\vec{t_b}, \theta_b, \vec{l_b}$ are the translation component, degree of rotation and principal axis of rotation of the recovered transformation, $\vec{t_i}, \theta_i, \vec{l_i}$ are those of the synchronised transformations and a_{ti} is the scalar part of the translation quaternion in (7).

After the data recovery method is applied, the interpolated transformations will have different shape to the one recovered by the linear interpolation method as shown in Fig. 4 which illustrates data recovered by the two approaches.

C. Hand-Eye Calibration

For the hand-eye problem itself, we adopt the solution from [13] since it allows us to minimise the errors due to interpolation. This method estimates the rotation component first, and then the translation is determined using the rotation motions of either one of the data streams. This means we can discard the motions from one of the data streams when estimating the translation. Given that we expect all interpolated motions to have a higher error than the measured ones, we always use the rotation motions from the non-interpolated data stream.

Algorithm 1: Synchronising and Calibrating Hand-eye.

procedure CHESS
  $\mathbf{A} \leftarrow$ construct camera's relative motions
  $\mathbf{B} \leftarrow$ construct robot's relative motions
  $\theta_A, \theta_B, d_A, d_B \leftarrow$ compute screw constraints
  Estimating time-delay parameter τ and synchronise the data streams
  Recover the lost data using (10) and (12).
  $\mathbf{R}_0 \leftarrow$ solve (14)
  repeat
    $\vec{t}_{\text{refine}} \leftarrow$ solve (15) using R_0
    $\vec{t}_0 \leftarrow \vec{t}_{\text{refine}}$
    $R_{\text{refine}} \leftarrow$ solving (14), (16) using $\vec{t}_0$
    $\mathbf{R}_0 \leftarrow \mathbf{R}_{\text{refine}}$
  until the solution converges
  $\mathbf{X} \leftarrow [\mathbf{R}_0, \vec{t}_0; 0\ 0\ 0\ 1]$
  return X ▷ Hand to eye transformation
end repeat

The main equation for solving the rotation component is,

$$\begin{bmatrix} a_0 - b_0 & -(\vec{a} - \vec{b})^T \\ \vec{a} - \vec{b} & [\vec{a} + \vec{b}]_\times + (a_0 - b_0)\mathbf{I}_3 \end{bmatrix} \mathbf{q} = \mathbf{0} \tag{13}$$

$$\mathbf{K}(\mathbf{a}, \mathbf{b})\mathbf{q} = \mathbf{0} \tag{14}$$

where $\mathbf{I}_3$ defines an identity matrix and $\mathbf{K}$ is a matrix of size $4N \times 4$ containing the same element as shown in (13). The translation is solved by using the initialised rotation component from the previous equation.

$$[\vec{\omega}_a]_\times \vec{t} = \mathbf{R}\vec{v}_b - \vec{v}_a \tag{15}$$

The term $\vec{\omega}_a$ can be replaced by $\mathbf{R}\vec{\omega}_b$ if the data stream $\{\mathbf{A}\}$ is the one with slower capture rate. Similarly, this applies to the equation for optimising rotation component as well.

$$\vec{v}_a = [\vec{t}]_\times \vec{\omega}_a + \mathbf{R}\vec{v}_b \tag{16}$$

The whole process to achieve is summarised in Algorithm 1.

IV. Experiments and Results

A. Experiments With Simulated Data

Synthetic data is generated by creating a loop of transformations denoting camera motions, robot motions, the hand-eye transformation and the robot-world transformation. The camera transformations and robot transformations are corrupted by Gaussian noise before synchronisation, data recovery and calibration as shown in the following equation.

$$\tilde{\mathbf{T}} = \mathbf{T} \begin{bmatrix} \text{rodrigues}(\sigma_r v_r) & \sigma_t v_t \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{17}$$

where $\mathbf{T}$ is an arbitrary transformation, σ_r, σ_t are standard deviations of a zero-mean Gaussian noise for rotation and translation and v_r, v_t are the 3×1 rotation and translation vectors. The Rodrigues function transforms a vector of angle-axis representation to a rotation matrix.

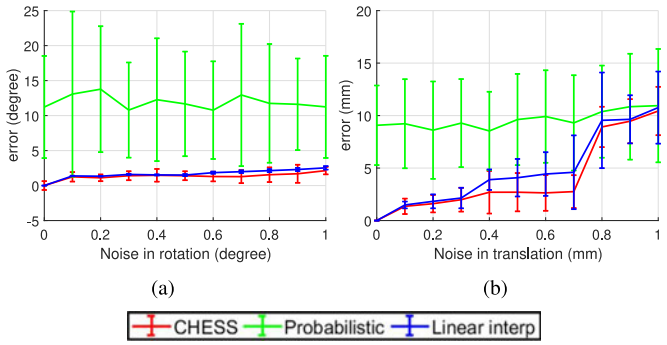

Fig. 5. Performance of the two algorithms when they are tested with increasing Gaussian noise in the transformations. (a) Error in the rotation estimation. (b) Error in the translation estimation.

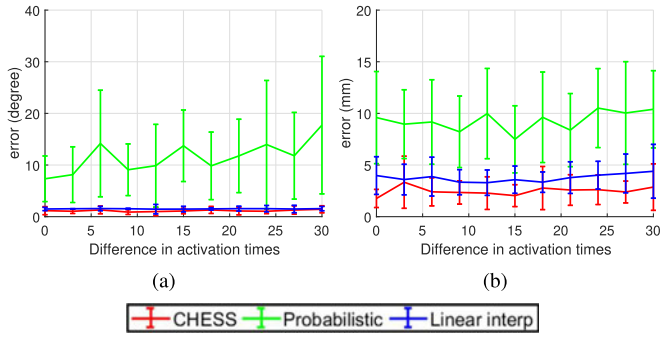

Fig. 6. Performance of the two algorithms when they are tested with increasing time-delay imposing on one data stream. This simulates the situation when the two tracking equipments are not activated at the same time. (a) Error in the rotation estimation. (b) Error in the translation estimation.

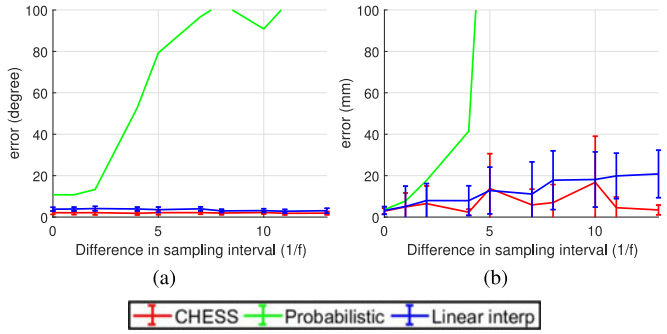

Fig. 7. Performance of the two algorithms when they are tested with different sampling interval. This simulates the situation when the two data streams do not have the same capture rates. (a) Error in the rotation estimation. (b) Error in the translation estimation.

We compare the performance of our algorithm with the linear interpolation method and probabilistic method described in [23] in different set-ups by varying simulation parameters, standard deviation of Gaussian noise, time-delay and the difference in sampling interval. For each set of parameters, the experiment is run for 50 times.

1) Increasing Gaussian Noise in Transformations: In this experiment, the noise in translation is increased from 0 mm to 1 mm and the noise in rotation is varied from 0 degree to 1 degrees, while the time-delay is kept constant as 30 samples.

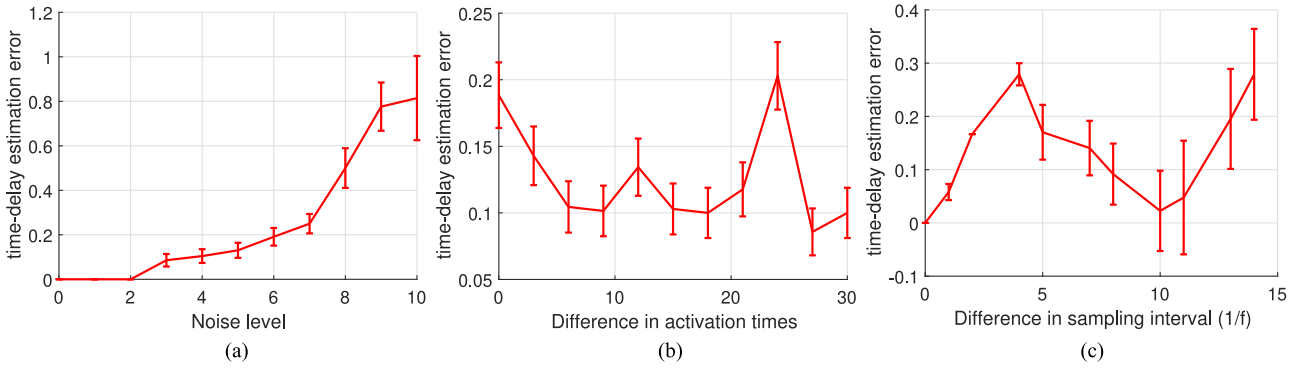

Fig. 8. Performance of the time-delay estimation algorithm when it is tested with different experimental parameters. (a) Increasing noise in transformations. (b) Increasing in time-delay. (c) Increasing in the difference between sampling rate.

The sampling interval of the camera motion is 3 samples and the robot motion is recorded every 4 samples. The results in Fig. 5(a) and (b) show that our algorithm outperforms both methods when noise is present in the systems. This also shows that calibrating the hand-eye matrix using synchronised data followed by conventional methods yields more accurate result than working directly with unsynchronised data using a probabilistic approach.

2) Increasing Time-Delay Between Two Data Streams: The noise is kept constant at 0.5 mm in translation and 0.5 degrees in rotation in this experiment. The sampling intervals of the two transformations also remain the same as the previous experiment. Figs. 6(a) and 5(b) show that CHESS can provide better calibration accuracy than both methods. It also shows that our algorithm can maintain the level of error regardless of increasing time-delay between the two data streams which mean it can estimate the time-delay based on screw constraints correctly despite the noise added into the systems.

3) Increasing Difference in Sampling Interval: While the noise is kept constant and the time-delay is set as 30 samples, we vary the sampling interval of one data stream from 3 to 16 (the difference to the other sampling interval is 0 to 13). As shown in Fig. 7(a) and (b), the proposed algorithm can recover the transformation on the lower capture rate data stream and perform hand-eye calibration more accurately than the linear interpolation and the probabilistic algorithms. The error from the probabilistic method increases rapidly because the condition of having at least one correspondence in the transformations pool cannot be satisfied when the capture rates of the two data streams are not the same.

4) Error in Time-Delay Estimation: In this experiment, we evaluate the performance of CHESS only the time-delay estimation part with the same experimental set-up as the previous experiments. Fig. 8(a)–(c) show that our proposed method can still estimate time-delay accurately with the varying activation times and sampling intervals, although the expected increasing trend in the error is shown in the experiment with increasing noise intensity.

B. Experiment With Real Data

In the experiment with real data, we work with KUKA LBA iiwa 7 R800 and we attach endoscope to the flange of the robot.

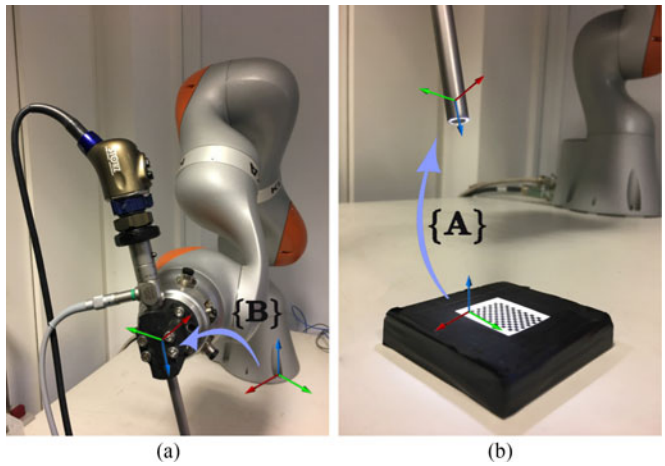

Fig. 9. The set-up for the experiment with the KUKA robot. (a) Endoscope attached to the flange of the robot. (b) Endoscope capturing a video of the calibration grid. The data streams $\{\mathbf{A}\}$ and $\{\mathbf{B}\}$ define the relative motions of the camera and the robot arm, respectively.

The set-up of the experiment is displayed in Fig. 9. The capture rate of the robot and the endoscope are 700 and 30 frames per second, respectively. The data streams defining the eye motions $\{\mathbf{A}\}$ and the hand motions $\{\mathbf{B}\}$ are computed by using (2). For the grid detector, we use the MATLAB Camera Calibration Toolbox[1] to detect the grid in every frame of the data stream with examples shown in Fig. 10.

The experiment is run on a 2.5 GHz Intel Core i5-7200U laptop and the synchronisation/recovery step takes the processor around 975 seconds to complete the whole dataset of 7,000 transformations. Our algorithm tends to have high computational cost with long motion. While the computational cost is one of the main evaluation criteria in a real-time application, it is not the main priority as the synchronisation/recovery step is typically an offline procedure.

Since we do not know the ground truth of the hand-eye matrix in the experiment on real data, we adopt the camera extrinsic parameters prediction as the performance evaluation method [16]. This validation method is widely used in evaluating calibration

[1]http://www.vision.caltech.edu/bouguetj/calib_doc

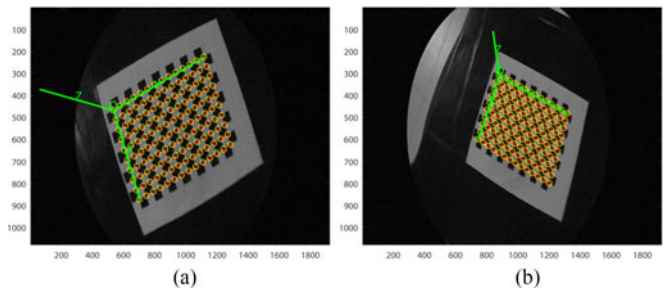

Fig. 10. Examples of how the images captured from the camera shown in Fig. 9 look like with X-Y axes showing the location in pixels. Every frame is fed into grid detector function and the extrinsic parameters (grid orientation with respect to the camera) are then computed.

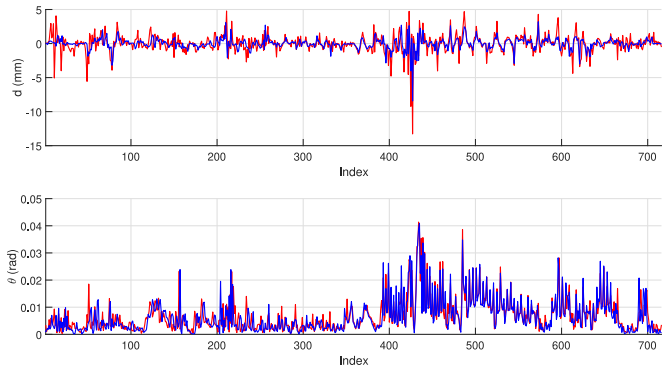

Fig. 11. Synchronisation of the two signals after resampling and cross-correlation. The upper graph shows the screw constraint on the translation component and the bottom graph show the constraint on the rotation component.

performance of the hand-eye problem. The concept is to calculate the error of the predicted parameters using the hand-eye matrix obtained from the calibration and compare with the pose calculated by vision using the equation:

$$^{\text{cam}}\mathbf{T}_{\text{grid}}(\tau'_{\text{sync, predicted}}) = \mathbf{X}\mathbf{B}\mathbf{X}^{-1}\ {}^{\text{cam}}\mathbf{T}_{\text{grid}}(\tau) \qquad (18)$$

where τ is the time instance we recover using our algorithm, τ'_{sync} is the time instance that the two data streams are already synchronised and $\mathbf{B}$ is the transformation defining the robot motion from the time instance τ and τ'.

Fig. 11 shows the screw constraints on rotation and translation after the two data streams are synchronised. Although the pattern in both d and θ are both corrupted heavily by noise and the patterns are not clearly visible especially the constraint on the translation component, the algorithm can still find the correlation and synchronise them accordingly. The synchronisation is evident at the final half of the signal. Data recovery procedure based on screw constraints is applied afterwards, a segment of the whole motion is shown in Fig. 12. The ripple and uncertainty shown in the plot is the influence of noise. Since the distance recovered by the algorithm is of small magnitude, the noise can become more significant and corrupt the whole optimisation process. However, the recovered data can be used in the calibration and yield accurate result as shown in the next plot.

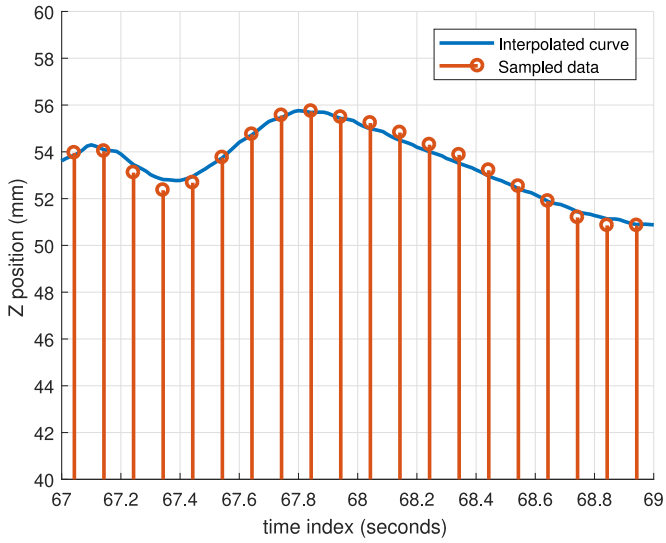

Fig. 12. The segment of the whole motion starting from $t = 67$ seconds to $t = 69$ seconds. The blue curve is the interpolated version of the lower sampling rate data. It interpolates between the two points that synchronise with the higher sampling rate data.

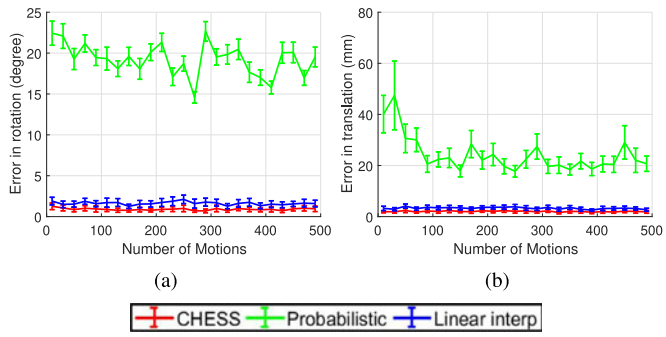

Fig. 13. Performance of the two algorithms when they are tested with the real data captured from the KUKA robot. (a) Error in the rotation estimation. (b) Error in the translation estimation.

The comparison between the calibration performance of our algorithm, the linear interpolation method and the probabilistic method is shown in Fig. 13. Despite the convergence of the error at 1.5–2 mm and 1 degree, the proposed method clearly outperforms the probabilistic method in both rotation and translation estimations. The calibration error evaluated from probabilistic method is extremely high because the capture rate of the camera and the KUKA API are very different, i.e. there exist a substantial amount of data on KUKA's end that do not have correspondences and this does not satisfy the condition for the probabilistic method. On the other hand, with the synchronised data, we can use the whole data streams for the calibration and achieve high accuracy.

While the calibration results from the linear interpolation and CHESS are comparable, our method still outperforms the linear interpolation algorithm. The only difference in these two algorithms are the interpolation methods where the linear method assumes the linear motion between two consecutive poses. Although the motions between any two consecutive poses are small

in magnitude due to high capture rate, the motion still cannot be assumed linear because it must conform to the hand-eye screw constraints. Therefore, the hand-eye equation does not hold for linearly interpolated pose. Our proposed method, on the contrary, interpolates the missing poses using screw constraints which yield the valid hand-eye data for the calibration.

V. Conclusion

In this letter, we proposed a data synchronisation and recovery method of the data streams used for hand-eye calibration using screw constraints. Our method applies cross-correlation and data resampling to synchronise streams and then exploits the fact that due to the rigid hand-eye connection, motion must conform to screw constraints. The algorithm performs data fitting as a preprocessing step before performing hand-eye calibration because in practice data collected from different sources has variable capture rates and activation times which lead to asynchronicity and data loss. Our data recovery method can also predict a good estimate of the lower capture rate data when the rate of the other source is available. This results in fully integrated tracking when the two systems are used at the same time. In the experiments on simulated and real data captured from a KUKA robot, we show that our proposed algorithm outperforms the linear interpolation and the state-of-the-art probabilistic approach in both rotation and translation estimations. We also show that it is better to synchronise the data before performing hand-eye calibration. In future work, we aim to investigate probabilistic methods and to develop hand-eye calibration specifically for the designed synchronisation algorithm to enhance the calibration accuracy so that any tracking equipment can be used with a robotic system.

References

[1] R. Negenborn, "Robot localization and Kalman filters on finding your position in a noisy world," Ph.D. dissertation, Utrecht Univ., Utrecht, The Netherlands, 2003.

[2] M. Magnabosco and T. P. Breckon, "Cross-spectral visual simultaneous localization and mapping (SLAM) with sensor handover," *Robot. Auton. Syst.*, vol. 61, no. 2, pp. 195–208, 2013.

[3] M. Hayashibe *et al.*, "Preoperative planning system for surgical robotics setup with kinematics and haptics," *Int. J. Med. Robot. Comput. Assisted Surg.*, vol. 1, no. 2, pp. 76–85, 2005.

[4] D. Stoyanov, A. Darzi, and G.-Z. Yang, "Laparoscope self-calibration for robotic assisted minimally invasive surgery," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2005*, J. S. Duncan and G. Gerig, Eds. Berlin, Germany: Springer, 2005, pp. 114–121.

[5] X. Kuai, K. Yang, S. Fu, R. Zheng, and G. Yang, "Simultaneous localization and mapping (SLAM) for indoor autonomous mobile robot navigation in wireless sensor networks," in *Proc. 2010 Int. Conf. Netw., Sens. Control*, Apr. 2010, pp. 128–132.

[6] I. Wieser, A. V. Ruiz, M. Frassl, M. Angermann, J. Mueller, and M. Lichtenstern, "Autonomous robotic SLAM-based indoor navigation for high resolution sampling with complete coverage," in *Proc. 2014 IEEE/ION Position, Location Navig. Symp.*, May 2014, pp. 945–951.

[7] R. Y. Tsai and R. K. Lenz, "Real time versatile robotics hand/eye calibration using 3D machine vision," in *Proc. 1988 IEEE Int. Conf. Robot. Autom.*, Apr. 1988, vol. 1, pp. 554–561.

[8] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB," *IEEE Trans. Robot. Autom.*, vol. 5, no. 1, pp. 16–29, Feb. 1989.

[9] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 345–358, Jun. 1989.

[10] F. C. Park and B. J. Martin, "Robot sensor calibration: Solving $\mathbf{AX} = \mathbf{XB}$ on the Euclidean group," *IEEE Trans. Robot. Autom.*, vol. 10, no. 5, pp. 717–721, Oct. 1994.

[11] J. C. K. Chou and M. Kamel, "Quaternions approach to solve the kinematic equation of rotation, $\mathbf{A}_a\mathbf{A}_x = \mathbf{A}_x\mathbf{A}_b$, of a sensor-mounted robotic manipulator," in *Proc. 1988 IEEE Int. Conf. Robot. Autom.*, Apr. 1988, vol. 2, pp. 656–662.

[12] J. C. K. Chou and M. Kamel, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *Int. J. Robot. Res.*, vol. 10, no. 3, pp. 240–254, 1991.

[13] K. Pachtrachai, M. Allan, V. Pawar, S. Hailes, and D. Stoyanov, "Hand-eye calibration for robotic assisted minimally invasive surgery without a calibration object," in *Proc. 2016 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2016, pp. 2485–2491.

[14] A. Malti and J. P. Barreto, "Robust hand-eye calibration for computer aided medical endoscopy," in *Proc. 2010 IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 5543–5549.

[15] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE Trans. Robot. Autom.*, vol. 14, no. 4, pp. 617–622, Aug. 1998.

[16] K. Daniilidis, "Hand-eye calibration using dual quaternions," *Int. J. Robot. Res.*, vol. 18, no. 3, pp. 286–298, 1999.

[17] A. Malti, "Hand eye calibration with epipolar constraints: Application to endoscopy," *Robot. Auton. Syst.*, vol. 61, no. 2, pp. 161–169, 2013.

[18] J. Li, G. Endo, and E. F. Fukushima, "Hand-eye calibration using stereo camera through pure rotations -fitting circular arc in 3D space with joint angle constraint-," in *Proc. 2015 IEEE Int. Conf. Adv. Intell. Mechatronics*, Jul. 2015, pp. 1–6.

[19] H. Zhang, "Hand/eye calibration for electronic assembly robots," *IEEE Trans. Robot. Autom.*, vol. 14, no. 4, pp. 612–616, Aug. 1998.

[20] J. Heller, D. Henrion, and T. Pajdla, "Hand-eye and robot-world calibration by global polynomial optimization," in *Proc. 2014 IEEE Int. Conf. Robot. Autom.*, May 2014, pp. 3157–3164.

[21] M. K. Ackerman and G. S. Chirikjian, *A Probabilistic Solution to the $AX = XB$ Problem: Sensor Calibration Without Correspondence*. Berlin, Germany: Springer, 2013, pp. 693–701.

[22] M. K. Ackerman, A. Cheng, and G. Chirikjian, "An information-theoretic approach to the correspondence-free $ax = xb$ sensor calibration problem," in *Proc. 2014 IEEE Int. Conf. Robot. Autom.*, May 2014, pp. 4893–4899.

[23] Q. Ma, H. Li, and G. S. Chirikjian, "New probabilistic approaches to the $ax = xb$ hand-eye calibration without correspondence," in *Proc. 2016 IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 4365–4371.

[24] H. Li, Q. Ma, T. Wang, and G. S. Chirikjian, "Simultaneous hand-eye and robot-world calibration by solving the $ax = yb$ problem without correspondence," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 145–152, Jan. 2016.

[25] H. H. Chen, "A screw motion approach to uniqueness analysis of head-eye geometry," in *Proc. 1991 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1991, pp. 145–151.

[26] J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer With Applications to Orbits, Aerospace, and Virtual Reality*. Princeton, NJ, USA: Princeton Univ. Press, 1999.

[27] S. H. Mark, W. Spong, and M. Vidyasagar, *Robot Modeling and Control*. Hoboken, NJ, USA: Wiley, 2006.

[28] L. R. Rabiner, *Multirate Digital Signal Processing*, 1st ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[29] L. R. Rabiner, "A general program to perform sampling rate conversion of data by rational ratios," 1979. [Online]. Available: http://michaelgellis.tripod.com/dsp/pgm32.html

[30] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-24, no. 4, pp. 320–327, Aug. 1976.

[31] N. S. M. Tamim and F. Ghani, "Hilbert transform of FFT pruned cross correlation function for optimization in time delay estimation," in *Proc. 2009 IEEE 9th Malaysia Int. Conf. Commun.*, Dec. 2009, pp. 809–814.

[32] B. K. Horn, "Some notes on unit quaternions and rotation," 2001.