

Improved Noise Covariance Estimation in Visual Servoing Using an Autocovariance Least-squares Approach

Jasper Brown^{*} Daobilige Su^{*} He Kong^{*} Salah Sukkarieh^{*}
Eric Kerrigan^{**}

^{*} Australian Centre for Field Robotics, The University of Sydney,
Australia (e-mail: [j.brown, d.su, h.kong, salah]@acfr.usyd.edu.au)

^{**} Department of Electrical & Electronic Engineering and Department
of Aeronautics, Imperial College London, UK (e-mail:
e.kerrigan@imperial.ac.uk)

Abstract: For pose estimation in visual servoing, by assuming the relative motion over one sample period to be constant, many existing works adopt a linear time invariant (LTI) dynamic model. Since the standard feature point transformation is nonlinear, extended Kalman filtering (EKF) has become popular due to its simplicity. Thus, the problem at hand becomes filtering of an LTI system with a time-varying output matrix. To obtain satisfactory performance, accurate knowledge of the noise covariances is essential. Various methods have been proposed on how to adaptively update their values to improve performance. However, these techniques cannot guarantee the positive semidefiniteness (PSD) of the covariance estimates. In this paper, we propose to apply the autocovariance least-squares (ALS) approach to covariance identification in pose estimation. The ALS approach can provide reliable estimates of the covariance matrices while maintaining their PSD and imposing desired structural constraints. Our tests show that using the covariance estimates from the ALS method in EKF can reduce the average pose estimation error by more than 30% in simulation, and the average position estimation error by about 30% using experimental data, respectively, compared to a hand-tuned EKF.

Keywords: visual servoing, pose estimation, Kalman filtering, noise covariance identification

1. INTRODUCTION

Visual servoing (VS) is a mature robotic technique having wide applications such as target/feature tracking, autonomous vehicle navigation (Lupton and Sukkarieh, 2012; Liang et al., 2018; Atanasov et al., 2014). Existing VS schemes can be classified as image-based VS (IBVS), position-based VS (PBVS), and hybrid approaches (Chaumette et al., 2016). In IBVS and PBVS, the tracking error is defined in the image and the Cartesian space respectively, while in hybrid approaches, the error is defined in both spaces. Of particular interest in this paper is pose estimation in PBVS using Kalman filtering-based methods. Although there are many existing estimation frameworks for systems with noises and/or unmodeled dynamics (see Gustafsson (2000), Kong and Sukkarieh (2019), and the references therein), here we are especially interested in EKF-based pose estimation in VS and accurately identifying the noise covariances for its implementation.

In computer vision and robotics, visual pose estimation refers to estimating the relative position and orientation of a camera with respect to (w.r.t.) an object using visual information, or vice versa. Given the presence of noisy effects in VS, filtering methods for pose estimation have received much attention (Wilson et al., 1996; Lippiello et. al, 2007; Janabi-Sharifi and Marey, 2010; Assa and Janabi-Sharifi, 2015). In these works, based on the assumption that the

relative motion over one sample period remains constant, an LTI dynamic model is proposed whose state is a 12 dimensional vector comprised of the 6 pose parameters and their rates of change. The measurement outputs are the image plane coordinates of the feature points obtained from a standard pin-hole camera model. Gaussian noises are then used to capture the dynamic modelling and measurement errors. Since feature point transformation is nonlinear in the system state, EKF is adopted in the above-mentioned works, in which the output equations are linearized around the current state estimates. While the noise covariances are taken to be constant in Wilson et al. (1996), more recent research in Lippiello et. al (2007), Janabi-Sharifi and Marey (2010), Assa and Janabi-Sharifi (2015) has shown how to update them adaptively to potentially improve estimation performance, although the PSD of the updated noise covariances is not guaranteed. In Janabi-Sharifi and Marey (2010), the above issue is dealt with by resetting the diagonal elements of covariance estimates to their absolute values. From a physical point of view, however, both non-PSD of the covariance estimates and the former way of handling that, are hard to interpret.

Noise covariance estimation is a parameter identification problem often encountered in model-based and adaptive control/estimation (Gustafsson, 2000). Existing noise covariance estimation methods can be classified as Bayesian, maximum likelihood, covariance matching, and correlation

techniques (Dunik et al., 2017). The ALS approach, an appealing correlation-based method, was proposed for LTI systems in Odelson et al. (2006), Rajamani and Rawlings (2009). The core concept of the ALS approach is to design a stable suboptimal filter with gain L , derive the L -innovations model and the corresponding steady-state conditions. Since the filter is not necessarily optimal, the sequence of the L -innovations, unlike the one associated with the optimal KF (Anderson and Moore, 2012, chp. 9), is *not* white. Based on the steady-state conditions and the ergodicity of the process, the question of noise covariance estimation can be transformed to a least-squares optimization problem, where the error between steady-state analysis and its approximation is minimized (Odelson et al., 2006). Compared to traditional techniques, the ALS method not only offers existence and uniqueness conditions for the covariance estimates but also leads to smaller estimation error. Extension of the ALS method to time-varying and nonlinear systems has been discussed in Rajamani and Rawlings (2007), Ge and Kerrigan (2017a).

This work leverages concepts from the ALS method and examines the technique for identifying the noise covariances in PBVS. The LTI dynamic modelling approach in Wilson et al. (1996), Lippiello et. al (2007), Janabi-Sharifi and Marey (2010), Assa and Janabi-Sharifi (2015) will be adopted. As mentioned above, the linearized output matrix is time-varying due to the nonlinearity in feature point transformation. Similarly with Wilson et al. (1996) and different from Lippiello et. al (2007), Janabi-Sharifi and Marey (2010), Assa and Janabi-Sharifi (2015), we assume the true noise covariances to be constant and use the ALS method to identify them. Firstly, we will implement EKF for state estimation based on some initial guessed value for the covariances. The suboptimal filter gains L_k can be recursively obtained from the KF equations. Secondly, based on the calculated innovations associated with the filter gains L_k and their correlations, the problem of covariance estimating with PSD guarantees is formulated as a standard least-squares problem, which can be transformed to a standard semidefinite programming (SDP) problem (Boyd and Vandenberghe, 2004, chp. 4) and solved efficiently by existing software packages such as CVX Grant and Boyd (2013). Compared to the methods of Wilson et al. (1996), Lippiello et. al (2007), Janabi-Sharifi and Marey (2010), Assa and Janabi-Sharifi (2015), a major advantage of the ALS framework is that PSD and any other desired structure of the covariance estimates can be enforced in the optimization straightforwardly. As it will be shown via simulation and experimental results later, using the noise covariance estimates from the ALS method in EKF for pose estimation can considerably reduce the pose estimation error, compared to EKF with noise covariances that are selected by trial and error. **Notation:** Same notation as that of Rajamani and Rawlings (2007), Ge and Kerrigan (2017a) is used.

2. PRELIMINARIES

We use the classic pinhole camera model and define the camera, object, and the image frames by their axes (X^C, Y^C, Z^C) , (X^O, Y^O, Z^O) , (u^C, v^C) , respectively. Without considering distortion of camera lens, the projection of a 3D point $P^O = [P_x^O \ P_y^O \ P_z^O]^T$ in the object

frame to a 2D point $P^C = [P_u^C \ P_v^C]^T$ in the camera image frame can then be described as

$$\begin{bmatrix} P^C \\ 1 \end{bmatrix} = \mathbf{M}_C [\mathbf{R}_{OC} \ \mathbf{t}_{OC}] \begin{bmatrix} P^O \\ 1 \end{bmatrix}, \quad (1)$$

where, $\mathbf{R}_{OC}$ and $\mathbf{t}_{OC}$ are the rotation and translation matrix, $\mathbf{M}_C$ is the intrinsic calibration matrix of the camera defined by $\mathbf{M}_C = \begin{bmatrix} f_x & c & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$, within which f_x and

f_y are horizontal and vertical focal lengths in pixels, c is the skewness of the axes in the image frame and u_0 and v_0 are the coordinates of the principal point on the image. When the camera lens' distortion is considered, radial and tangential distortions apply to the camera pixel coordinates. When the camera is fully calibrated Zhang (2000), its intrinsic parameters (f_x, f_y, u_0, v_0, c) and distortion coefficients are known. In (1), coordinates of the object feature points in the object frame $[P_x^O \ P_y^O \ P_z^O]^T$ can be obtained from the CAD model of the object, and the camera observation $[P_u^C, P_v^C]^T$ can be obtained using standard computer vision techniques. Therefore, we have two nonlinear equations with six unknown parameters corresponding to rotation $\mathbf{R}_{OC}$ and translation $\mathbf{t}_{OC}$ between the camera and object frames. Thus, one needs at least four non-collinear or coplanar feature points on the object to obtain a unique solution. Many VS applications utilize 4 to 6 feature points (Janabi-Sharifi and Marey, 2010). For pose estimation, the state vector, comprised of pose parameters and their derivatives, is defined as

$$x = [X \ \dot{X} \ Y \ \dot{Y} \ Z \ \dot{Z} \ \phi \ \dot{\phi} \ \alpha \ \dot{\alpha} \ \psi \ \dot{\psi}]^T. \quad (2)$$

As in Wilson et al. (1996), Janabi-Sharifi and Marey (2010), we assume a constant velocity model for each time step and have the state transition model as

$$x_{k+1} = Ax_k + w_k, \quad (3)$$

where $A \in \mathbb{R}^{n \times n}$ is a block diagonal matrix with $n = 12$, and each block diagonal entry as $\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$, T is the length of the sampling interval. From the projection equation given by (1), the nonlinear measurement model defining the image-feature locations in terms of x_k is

$$y_k = h(x_k) + v_k, \quad (4)$$

where, $y_k \in \mathbb{R}^{q \times n}$ with $q = 2p$ and p denoting the number of feature points. For models (3)-(4), we assume the initial state of (3) is $x_0 \sim \mathcal{N}(\bar{x}_0, P_{1|0})$, i.e., x_0 is normally distributed with mean $\bar{x}_0$ and covariance $P_{1|0}$; w_k and v_k are uncorrelated zero-mean Gaussian noise with covariance matrices $Q \succeq 0$ and $R \succeq 0$, respectively, and statistically independent of x_0 . Since the true initial state and covariances Q and R are unknown, some guessed values of the initial state $\hat{x}_{1|0}$ with its error covariance P_g , and noise covariances Q_g and R_g have to be provided. By linearizing the nonlinear function around the current estimate $\hat{x}_k$, i.e., $C_k = \left. \frac{\partial h(\cdot)}{\partial x_k} \right|_{x_k = \hat{x}_k} \in \mathbb{R}^{q \times n}$, we get the discrete-time LTV model:

$$x_{k+1} = Ax_k + w_k, \ y_k \approx C_k x_k + v_k, \quad (5)$$

as an approximation of (3)-(4). Based on the guessed covariances, we can recursively derive the sub-optimal EKF filter gains L_k^{so} :

$$\begin{aligned} P_{k|k-1} &= AP_{k-1}A^\top + Q_g, \\ L_k^{so} &= (P_{k|k-1}C_k^\top) (C_kP_{k|k-1}C_k^\top + R_g)^{-1}, \\ P_k &= (I - L_k^{so}C_k)P_{k|k-1}. \end{aligned} \quad (6)$$

The subsequent analysis follows from our previous work in Ge and Kerrigan (2017a). The estimated state sequence of (5) can be obtained

$$z_k = y_k - \hat{y}_{k|k-1}, \quad \hat{x}_k = \hat{x}_{k|k} = \hat{x}_{k|k-1} + L_k^{so}z_k,$$

where, z_k are the L_k^{so} -innovations, and $\hat{x}_{k|k-1}$ and $\hat{y}_{k|k-1}$ are the predicted state and output given by

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1}, \quad \hat{y}_{k|k-1} = C_k\hat{x}_{k|k-1}.$$

Define the state estimation error as $\varepsilon_k = x_k - \hat{x}_{k|k-1}$, for $k = 1, \dots, M$. We then have

$$\hat{x}_{k+1|k} = A(\hat{x}_{k|k-1} + L_k^{so}(y_k - \hat{y}_{k|k-1})).$$

Denote $\bar{A}_k = A - AL_k^{so}C_k$, $\bar{G}_k = [I_n - AL_k^{so}]$, $\bar{w}_k = [w_k^\top v_k^\top]^\top$. The state-space model for L_k^{so} -innovations becomes:

$$\varepsilon_{k+1} = \bar{A}_k\varepsilon_k + \bar{G}_k\bar{w}_k, \quad z_k = C_k\varepsilon_k + v_k. \quad (7)$$

Denote M as the length of the data points. We assume in the remainder of the paper that (A, C_k) in (5) is uniformly detectable and $\bar{A}_k$ is exponentially stable over the time horizon $k = 1, \dots, M$ such that $\mathbb{E}[\varepsilon_k] \rightarrow 0$ as $k \rightarrow \infty$.

3. THE ALS METHOD FOR NOISE COVARIANCE ESTIMATION IN PBVS

A necessary and sufficient condition for the optimality of the KF is that the innovation sequence be white Gaussian noise (Anderson and Moore, 2012, chp. 9). However, for a sub-optimal filter, $z_1, z_2, \dots, z_M$ in (7) are correlated, thus we can calculate the auto-covariance of z_k Odelson et al. (2006). For any $k \in \{1, \dots, M\}$, the auto-covariance of z_k with j time-lags is defined as

$\mathcal{C}_j(z_k) = \mathbb{E}[(z_{k+j} - \mu_{k+j})(z_k - \mu_k)^\top] = \mathbb{E}[z_{k+j}z_k^\top] - \mu_{k+j}\mu_k^\top$, for $j = 0, 1, \dots, N-1$, where N is the maximum number of time lags and $\mu_{k+j} = \mathbb{E}[z_{k+j}]$. Given the output measurements $(y_k)_{k=1}^{\bar{k}}$, $\bar{k} \leq M$, assume that there exists a smoothed initial state $\hat{x}_{1|\bar{k}}$, where $\hat{x}_{1|\bar{k}} = \mathbb{E}[x_1|(y_k)_{k=1}^{\bar{k}}]$, such that if we let $\hat{x}_{1|0} = \hat{x}_{1|\bar{k}}$, the expectation of initial state error term ε_1 can be taken to be zero. Based on the above assumption, and because for any k , the state error term ε_{k+1} is a function of ε_1 and $(\bar{w}_k)_{k=1}^M$, auto-covariance of vector z_k with j time-lags can be simplified as

$$\forall k, j : \mu_{k+j} = 0 \implies \mathcal{C}_j(z_k) = \mathbb{E}[z_{k+j}z_k^\top].$$

If we use a fragment of innovations $(z_{k+1})_{k=1}^{N_z}$, where $N_z = M_e - N + 1$, the auto-covariance with j time-lags becomes

$$\mathcal{C}_j \left((z_{k+1})_{k=1}^{N_z} \right) = [\mathcal{C}_j(z_2) \cdots \mathcal{C}_j(z_{M_e-N+2})],$$

where M_e is the estimation data length with $N \ll M_e \leq M$. Denote the auto-covariance matrix of $(z_{k+1})_{k=1}^{N_z}$ as

$$\begin{aligned} \mathcal{R} &= [\mathcal{C}_0^\top \left((z_{k+1})_{k=1}^{N_z} \right) \cdots \mathcal{C}_{N-1}^\top \left((z_{k+1})_{k=1}^{N_z} \right)] \\ &= [\mathcal{R}_0 \ \mathcal{R}_1 \cdots \mathcal{R}_{M_e-N}], \end{aligned}$$

where, $\mathcal{R}_i = \mathbb{E}[z_{2+i}z_{2+i}^\top \cdots z_{2+i}z_{N+1+i}^\top]^\top$, for $i = 0, \dots, M_e - N$. Note that the auto-covariance matrix $\mathcal{R}$ is a function of Q, R and $P_{1|0}$. The innovation sequence obtained from the LTV model (7) is generally non-stationary.

However, since the process is ergodic, one can use the sample estimates to approximate $\mathcal{R}$:

$$\bar{\mathcal{R}} = \begin{bmatrix} \bar{z}_2\bar{z}_2^\top & \cdots & \bar{z}_{M_e-N+2}\bar{z}_{M_e-N+2}^\top \\ \bar{z}_3\bar{z}_2^\top & \cdots & \bar{z}_{M_e-N+3}\bar{z}_{M_e-N+2}^\top \\ \vdots & \ddots & \vdots \\ \bar{z}_{N+1}\bar{z}_2^\top & \cdots & \bar{z}_{M_e+1}\bar{z}_{M_e-N+2}^\top \end{bmatrix} \quad (8)$$

where $\bar{z}_k$ represents the actual innovation terms calculated from the data using

$$\bar{z}_k = y_k - \hat{y}_{k|k-1}. \quad (9)$$

One can then define an unconstrained least squares optimization problem to estimate the true covariance:

$$(\hat{P}_{1|0}^*, \hat{Q}^*, \hat{R}^*) = \arg \min_{\hat{P}_{1|0}, \hat{Q}, \hat{R}} \left\| \mathcal{R}(\hat{P}_{1|0}, \hat{Q}, \hat{R}) - \bar{\mathcal{R}} \right\|_F^2. \quad (10)$$

It is possible to use the above derivations to show that

$$\begin{aligned} \mathcal{R}(P_{1|0}, Q, R) &= \Gamma(I_{N_z} \otimes P_{1|0})\bar{\Gamma}^\top + \Omega(I_{N_d} \otimes Q)\bar{\Omega}^\top \\ &\quad + \Phi(I_{N_d} \otimes R)\bar{\Phi}^\top + \Psi(I_{N_z} \otimes R)\bar{\Psi}^\top, \end{aligned}$$

where detailed expressions of the matrices are given in Ge and Kerrigan (2017a) and skipped here. To reformulate problem (10) as a standard least-squares problem, matrix $\mathcal{R}$ must be vectorized, which is the column-wise stacking of a matrix into a vector. Similarly, we vectorize $\bar{\mathcal{R}}$ given before (9), which is the approximated value of $\mathcal{R}$. Let $\bar{b} = (\bar{\mathcal{R}})_s$. In the ALS approach, the PSD of the covariance estimates can be enforced very conveniently in the optimization by adding additional constraints $\hat{P}_{1|0}, \hat{Q}, \hat{R} \succeq 0$. Thus, the original optimization problem (10) can be rearranged into a constrained least-squares problem with decision variables $\vartheta = [(\hat{P}_{1|0})_{ss} \ (\hat{Q})_{ss} \ (\hat{R})_{ss}]^\top$:

$$\min_{\vartheta} \|\mathcal{A}\vartheta - \bar{b}\|_2^2 \text{ s.t. } \hat{P}_{1|0}, \hat{Q}, \hat{R} \succeq 0 \quad (11)$$

where, for $i = 0, \dots, M_e - N$, and the expression of $\mathcal{A}$ can be found in Ge and Kerrigan (2017a). The above optimization problem can be reformulated as an SDP problem (Boyd and Vandenberghe, 2004, chp. 4), and then be solved effectively using CVX (Grant and Boyd, 2013).

4. RESULTS AND DISCUSSIONS

4.1 Simulation Setup and Results

We first set up a simulation scenario as shown in Fig. 1 using the V-REP software package (Rohmer et al., 2013). A Microsoft Kinect V1 camera is installed at the end effector of a Universal Robotics UR5, a six-DOF robotic arm. The target object, with five colored markers at known locations, is placed under the Kinect camera. The coordinate frames of the Kinect camera and target object are shown by their X,Y,Z axes in Fig. 1. When the simulation starts, the UR5 robot arm follows a predefined trajectory and the Kinect continuously observes the target from different poses. The ground truth poses of the robot arm end effector, the Kinect camera, the target object and positions of all five markers at all time instance can be obtained from the V-REP simulator. The RGB image frames recorded by the Kinect camera are also stored. Target marker positions in the image frames are estimated by thresholding in the hue-saturation-value color space and taking the centroids of resulting regions. The kinect camera is assumed to have no lens distortion to simplify

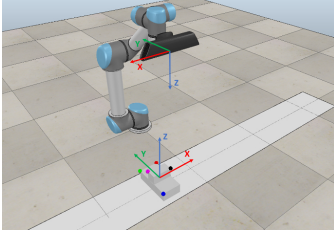


Fig. 1. VRep simulation of a visual servoing scenario

the simulation. The details about the key parameters used in the simulation are skipped here due to limited space. In Simulation Test 1, a linear trajectory is used such that the constant velocity model in (3) is accurate. See Fig. 2 for the first 5 seconds of the full trajectory. In Simulation Test 2, a nonlinear trajectory with rapid accelerations is used. See Fig. 3 for the first third of Simulation Test 2, this pattern repeats twice more for the full 33 second trajectory. For both tests, a trial and error process over several EKF's is run to estimate the camera states in (2) with 7 equally spaced values for Q_g (ranging from $10^{-7} \times I_n$ to $0.1 \times I_n$) and 5 equally spaced values for R_g (ranging from $1 \times I_n$ to $5 \times I_n$). The EKF with the initial guessed noise covariances $Q_g = 0.01 \times I_{12}$ and $R_g = 3 \times I_{10}$ was observed to give the lowest mean error. The units for the diagonal entries of Q_{gs} are in m^2 , $(m/sec)^2$, rad^2 and $(rad/sec)^2$ and the unit for R_{gs} is $pixels^2$. As the camera model and marker location extraction are nearly ideal, we add observation noise drawn from $\mathcal{N}(0, 1)$ to the observation vector. For both tests, the hand-tuned EKF with the above covariance guesses and $P = 1 \times I_{12}$ is firstly run to generate data. In calculating the approximated autocovariance $\hat{R}$ in (8), we select $M_e = 70$, $N = 20$. The first 50 time steps of innovations are discarded while the filter converges, the ALS problem (11) is solved repeatedly, using a moving window of 50 images to cover the entire $M = 1450$ frame simulation with 28 segments, resulting in 28 estimates of $\hat{Q}$ and $\hat{R}$. The mean of these estimates is used to run a second filter, called the ALS-EKF. The average of the ALS estimates for $\hat{Q}$ and $\hat{R}$ are:

$$\hat{Q} = 10^{-6} \times \text{diag} (0.021, 0.054, 0.026, 0.040, 0.592, 0.409, 3.418, 3.760, 30.736, 24.353, 13.042, 23.685),$$

$$\hat{R} = \text{diag} (0.921, 0.888, 1.087, 0.909, 0.962, 0.976, 0.829, 0.865, 0.878, 0.865).$$

In the above simulations, both $\hat{Q}$ and $\hat{R}$ are enforced to being diagonal when solving the ALS problem (11). Other desired structural constraints on the covariances can also be easily enforced within the CVX optimization framework. The estimated states using the hand tuned EKF and the ALS-EKF for the first 5 seconds of the Simulation Test 1 are shown in Fig. 2. The error comparisons of the two EKF's for Simulation Test 1 are given in Table 1. Using the ALS estimates for $\hat{Q}$ and $\hat{R}$ has reduced the RMS position and angle error by 33.1% and 50.5%, respectively. The ALS estimates for $\hat{Q}$ and $\hat{R}$ in Simulation Test 2 are

$$\hat{Q} = 10^{-4} \times \text{diag} (0.001, 1.681, 0.000, 0.483, 0.004, 1.053, 0.033, 6.812, 0.191, 21.506, 0.219, 0.018),$$

$$\hat{R} = \text{diag} (0.664, 1.058, 0.684, 0.794, 1.348, 0.945, 0.777, 1.063, 0.610, 0.780).$$

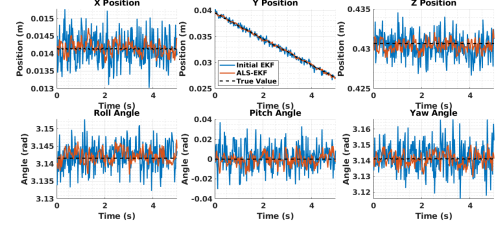


Fig. 2. The estimated states and ground truth for the first 5 seconds of Simulation Test 1

Table 1. Estimation error comparison of the two EKF's for Simulation Test 1

Filter Type	x (mm)	y (mm)	z (mm)	ϕ (mrad)	α (mrad)	ψ (mrad)
Mean Absolute Error						
Initial EKF	0.4026	0.3529	1.9380	3.6648	12.3844	8.9688
ALS-EKF	0.1919	0.1983	1.4119	1.9667	6.1530	4.2575
Max Error						
Initial EKF	1.3984	1.3797	6.5636	12.8461	43.0329	30.5378
ALS-EKF	0.7485	0.6775	3.4719	7.3749	27.3873	15.8724
Error Std Dev						
Initial EKF	0.4027	0.3513	1.5518	3.6476	12.3847	8.9711
ALS-EKF	0.1920	0.1935	0.7679	1.9531	6.1232	4.2535

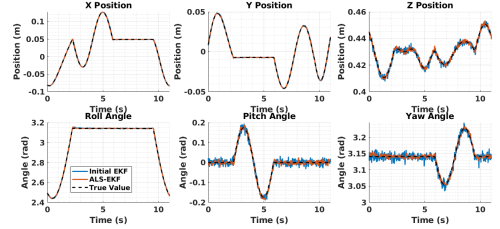


Fig. 3. The filter estimated states and ground truth for Simulation Test 2

Table 2. Estimation error comparison of the two EKF for Simulation Test 2

Filter Type	x (mm)	y (mm)	z (mm)	ϕ (mrad)	α (mrad)	ψ (mrad)
Mean Absolute Error						
Initial EKF	1.3226	0.7634	1.9533	5.2877	11.9752	8.8000
ALS-EKF	0.5519	0.3701	1.5612	3.5728	8.2918	5.2000
Max Error						
Initial EKF	4.6820	1.9997	6.3692	21.3880	41.8449	32.1000
ALS-EKF	4.1088	1.7287	4.9078	19.2321	32.8545	19.3000
Error Std Dev						
Initial EKF	1.3230	0.7620	1.5826	5.2715	11.9418	8.8000
ALS-EKF	0.5520	0.3688	1.1690	3.5659	8.2676	5.3000

As it can be seen, the ALS technique correctly estimates higher noise values for the velocity components, as the constant velocity model is violated for these state elements in Simulation Test 2. The error comparison for Simulation Test 2 is in Table 2. Using the ALS estimates for $\hat{Q}$ and $\hat{R}$ has reduced the RMS position and angle estimation error by 38.5% and 34.5%, respectively. As stated previously, a necessary and sufficient condition for the optimality of the KF is that the innovation sequence be white Gaussian noise (Anderson and Moore, 2012, chp. 9). To assess the "whiteness" of the innovations associated with both EKF, we apply the Fourier transform to the magnitude of the pixel space innovations sequence for the first marker and show this in Fig. 4. For the hand-tuned EKF, all markers showed a pattern with a significant low frequency peak, which is not present in the ALS-EKF data.

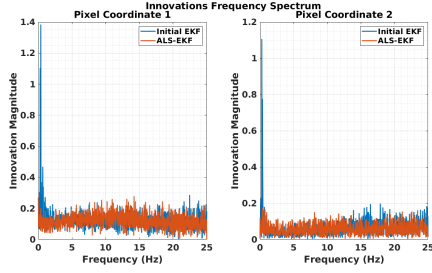


Fig. 4. The frequency spectrum for innovation sequence in Simulation Test 2. Left and Right are innovations of image pixel coordinate 1 & 2, respectively.

4.2 Physical Experiment Setup and Results

For the physical experiment, a FLIR Grasshopper3 camera is mounted to the end of a UR5 arm, shown in Figure 5. The arm follows a complex 60 second trajectory above a checkerboard target. The details about the key parameters used in the experiment are skipped here due to limited space. For each image, captured at 60Hz, 54 checkerboard



Fig. 5. The hardware setup for physical experiments

corner points are extracted and the camera pose which minimises the reprojection error over all the checkerboard corner points is calculated. A two point moving average filter is applied to form the ground truth for camera pose. Moreover, 4 corner points and 1 centre point are also extracted from each frame which form the observations vector for the EKF discussed next. To better capture the changes in the trajectory and prevent near-zero estimates of $\hat{Q}$ values, we have found that it is necessary to down sample the image data by using only one in n consecutive images. Overly aggressive down sampling using high values for n , however, results in large innovation magnitudes that can compromise the validity of the ALS method. Ideally, to reduce memory requirement in running the ALS, the smallest value of M_e length windows which both covers the full image dataset and generates sufficiently small innovations should be used. Under the above guidelines, an ALS window size of $M_e = 110$ and a down sampling size of $n = 6$ is adopted here when using the data from physical experiments. In other words, we sequentially choose the first image from 6 consecutive images. We do this repeatedly to form 5 data segments, and collect 110 images for each segment to run ALS. Note that a small data window size might lead to ALS estimates for Q which may result in an unstable EKF. To prevent this an additional constraint is added to the CVX problem,

Table 3. Estimation error comparison of the two EKF's using real experiment data

Filter Type	x (mm)	y (mm)	z (mm)	ϕ (mrad)	α (mrad)	ψ (mrad)
Mean Absolute Error						
Initial EKF	1.200	1.080	1.103	5.895	5.560	2.523
ALS-EKF	0.829	0.721	0.841	5.463	5.569	2.391
Max Error						
Initial EKF	7.225	3.581	4.010	17.345	15.676	4.767
ALS-EKF	4.385	1.867	2.267	12.877	17.062	2.985
Error Std Dev						
Initial EKF	1.199	1.081	1.075	4.857	4.992	2.496
ALS-EKF	0.829	0.720	0.791	4.517	5.007	2.382

enforcing that $\hat{Q}_{1,1} = \hat{Q}_{3,3} = \hat{Q}_{5,5}$, $\hat{Q}_{2,2} = \hat{Q}_{4,4} = \hat{Q}_{6,6}$, $\hat{Q}_{7,7} = \hat{Q}_{9,9} = \hat{Q}_{11,11}$, $\hat{Q}_{8,8} = \hat{Q}_{10,10} = \hat{Q}_{12,12}$. Following a similar procedure to that in the previous subsection, we select an EKF with initial guessed $Q_g = 10^{-6} \times I_n$. The mean reprojection error from the camera calibration step is used to generate R_g as $R_g = 0.34 \times I_n$. This EKF is run over all the data points and the ALS problem (11) is solved repeatedly for five times, each using a window of 110 image data. This results in 5 estimates for $\hat{Q}$ and $\hat{R}$ which are averaged to give an estimate of the process and measurement noise covariance matrices:

$$\hat{Q} = 10^{-5} \times \text{diag} (0.044, 4.059, 0.044, 4.059, 0.044, 4.059, 1.368, 71.036, 1.368, 71.036, 1.368, 71.036),$$

$$\hat{R} = \text{diag} (0.098, 0.346, 0.206, 0.665, 0.072, 0.408, 0.063, 1.82 \times 10^{-8}, 0.051, 0.404).$$

It turns out that $\hat{R}$ obtained above might not be as accurate as R_g found during calibration. Hence, we use the ALS estimate of $\hat{Q}$ along with R_g to run the ALS-EKF, and compare with the initial EKF. Figure 6 shows the results for the initial EKF and ALS-EKF. As shown in Table 3, the ALS-EKF provides considerably improved accuracy than the initial EKF by around 30% in position estimation, although the former slightly underperforms the latter in estimating the pitch angle.

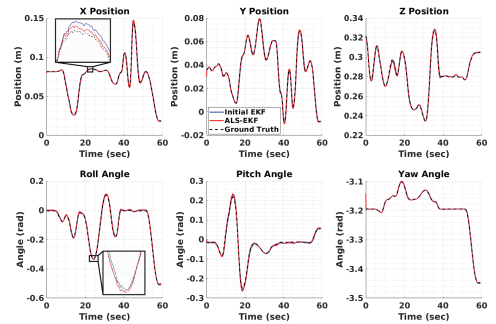


Fig. 6. The filter estimated states and ground truth using real experiment data

4.3 Discussions

For both simulation tests and real experiments, the obtained results indicated that the ALS EKF gives considerably improved performance over the initial hand-tuned EKF. In practice, better noise covariance estimates can be obtained by choosing $M_e \gg N$, ideally $M_e = M$. Doing so will increase the memory requirement in computing

matrix $\mathcal{A}$ in (11). Overall, the ALS method should be considered as an offline method. However, it can be used online if the computational architecture and algorithms get improved in the future. The current implementation of the ALS method using $M_e = 70$ and $M_e = 110$ respectively required 14 GB and 25 GB of computer memory to run. Memory limitations can be resolved in a number of different ways. In our previous work, Ge and Kerrigan (2017a), two memory efficient implementation of the ALS method have been proposed, resulting in a few hundreds of MB computer memory requirement. In this paper, we have utilised an alternative approach by taking the mean of ALS estimates generated with a sliding window of data. The ALS method can also offer estimates of the initial state error covariance $\hat{P}$, as indicated in the optimisation problem (11). One implication of implementing the ALS method on a sliding window is that estimates of $\hat{P}$ may be meaningless. However, for applications with periodic trajectories such as on-demand factory processes, the window size M_e can be matched to the periodic trajectory for accurate estimates of the initial state covariance. The ALS estimate of $\hat{R}$ was less accurate than using the mean reprojection error from the camera calibration step, where possible, estimates of R_g from calibration data should be used in conjunction with ALS estimates for Q and P . Alternatively, one could fix $R = R_g$, and use ALS to estimate Q and P , which we have not done so due to limited space. Note that we assume the noise covariances to be constant. For most VS applications this might be untrue, especially for process noise covariance Q . Employing a Q which varies with each data window is possible but requires a periodic trajectory and could lead to an unstable EKF if the window to trajectory alignment shifts, such as due to changing the motion control algorithm for the task.

5. CONCLUSION

This paper has presented the ALS approach for identifying the noise covariances for EKF-based pose estimation in VS, using a sliding data window. A major advantage of the proposed method is that the PSD and other structural constraints on the noise covariances can be enforced conveniently in the optimization, which can be solved efficiently using existing software packages. Both simulation and physical experiments were performed to validate the proposed method. Across all tests, using the noise covariance estimates from the ALS method in EKF for pose estimation can considerably reduce the pose estimation error, compared to EKF with hand-tuned noise covariances. Future work will apply moving horizon state estimation methods (Kong and Sukkarieh, 2018a,b; Ge and Kerrigan, 2017b), which could lead to smaller state estimation error and more accurate noise covariance estimates than EKF.

REFERENCES

- Anderson, B.D.O. and Moore, J.B. (2012). *Optimal Filtering*. Courier Corporation.
- Assa, A. and Janabi-Sharifi, F. (2015). A Kalman filter-based framework for enhanced sensor fusion. *IEEE Sensors Journal*, 15(6), 3281–3292.
- Atanasov, N., Zhu, M., Daniilidis, K., and Pappas, G. J. (2014). Semantic localization via the matrix permanent. In *2014 Robotics: Science and Systems*, 4465–4470.
- Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.
- Chaumette, F., Hutchinson, S., and Corke, P. (2016). Visual servoing, Springer Handbook of Robotics, 34, 841–866.
- Dunik, J., Straka, O., Kost, O., and Havlik, J. (2017). Noise covariance matrices in state-space models: a survey and comparison of estimation methods—part I. *Int. J. Adaptive Cont. Signal Proc.*, 31 (11), 1505–1543.
- Gustafsson, F. (2000). Adaptive filtering and change detection, John Wiley & Sons.
- Ge, M. and Kerrigan, E.C. (2017a). Noise covariance identification for time-varying and nonlinear systems. *Int. Journal of Control*, 90(9), 1903–1915.
- Ge, M. and Kerrigan, E.C. (2017b). Noise covariance identification for nonlinear systems using expectation maximization and moving horizon estimation. *Automatica*, 77, 336–343.
- Grant, M. and Boyd, S. (2013). CVX: Matlab software for disciplined convex programming.
- Janabi-Sharifi, F. and Marey, M. (2010). A Kalman-filter-based method for pose estimation in visual servoing. *IEEE Trans. on Robotics*, 26(5), 939–947.
- Kong, H. and Sukkarieh, S. (2018a). Metamorphic moving horizon estimation. *Automatica*, 97, 167–171.
- Kong, H. and Sukkarieh, S. (2018b). Suboptimal receding horizon estimation via noise blocking. *Automatica*, 98, 66–75.
- Kong, H. and Sukkarieh, S. (2019). An internal model approach to estimation of systems with arbitrary unknown inputs. *Automatica*, accepted and to appear.
- Liang, X., Wang, H., Liu, Y. H., Chen, W., Jing, Z. (2018). Image-based position control of mobile robots with a completely unknown fixed camera. *IEEE Trans. on Automatic Control*, 63(9), 3016–3023.
- Lippiello, V., Siciliano, B., Villani, L. (2007). Adaptive extended Kalman filtering for visual motion estimation of 3D objects. *Control Eng. Practice*, 15(1), 123–134.
- Lupton, T., Sukkarieh, S. (2012). Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Trans. on Robotics*, 28(1), 61–76.
- Odelson, B.J., Rajamani, M.R., and Rawlings, J.B. (2006). A new autocovariance least-squares method for estimating noise covariances. *Automatica*, 42(2), 303–308.
- Rajamani, M.R. and Rawlings, J.B. (2007). Application of a new data-based covariance estimation technique to a nonlinear industrial blending drum. Technical Report 2007-03, University of Texas-Wisconsin.
- Rajamani, M.R. and Rawlings, J.B. (2009). Estimation of the disturbance structure from data using semidefinite programming and optimal weighting. *Automatica*, 45(1), 142–148.
- Rohmer, E., Singh, S.P.N., and Freese, M. (2013). V-REP: A versatile and scalable robot simulation framework. In *2013 IEEE IROS*, 1321–1326.
- Wilson, W.J., Hulls, C.C.W., and Bell, G.S. (1996). Relative end-effector control using Cartesian position based visual servoing. *IEEE Trans. on Robotics and Automation*, 12(5), 684–696.
- Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(11), 1330–1334.