

# Research Notes

Mohsin Dalvi – DT17MEC050

April 28, 2021

Hello World! [1]

## 1 Summary of Jeroen Hol

Pose Estimation and Calibration Algorithms for Vision and Inertial Sensors  
20088  
Division of Automatic Control  
Department of Electrical Engineering  
Linköping University, Sweden

Augmented reality (AR) involves overlaying a real scene with computer generated graphics in real-time by showing the virtual objects on see-through head-mounted displays or superimposing them on live video imagery. In order to have realistic augmentation, i.e., positioning and aligning the virtual objects correctly with the real world, it is essential to know the position and orientation (pose) of the camera with high accuracy and low latency. A ?vision only approach? gives good absolute accuracy, but is difficult to run at high frame rate and is not robust during fast motions. An inertial measurement unit (IMU) is accurate for a short period and is drift-prone for longer time scales.

In this thesis, pose estimation is approached using a combination of a camera and an IMU. Using a 3D scene model containing natural landmarks removes the need for costly and time consuming procedure of preparing the environment, and allows for AR applications outside dedicated studios. The IMU unit provides rapid measurements of acceleration and angular velocity. The computer vision system generates correspondences between the camera image and a 3D scene model that contains positions of various natural markers and is generated offline using images and/or drawings of the scene. The camera pose is estimated by combining inertial and visual measurements in a sensor fusion model that solves a non-linear state estimation problem in real-time.

Some aspects have been previously published in

F. Gustafsson, T. B. Schön, and J. D. Hol. Sensor fusion for augmented reality. In Proceedings of 17th International Federation of Automatic Control World Congress, Seoul, South Korea, July 2008. Accepted for publication.

J. D. Hol, T. B. Schön, H. Luinge, P. J. Slycke, and F. Gustafsson. Robust real-time tracking by fusing measurements from inertial and vision sensors. Journal of Real-Time Image Processing, 2(2):149?160, Nov. 2007. doi:10.1007/s11554-007-0040-2.

J. D. Hol, T. B. Schön, F. Gustafsson, and P. J. Slycke. Sensor fusion for augmented reality. In Proceedings of 9th International Conference on Information Fusion, Florence, Italy, July 2006b.

doi:10.1109/ICIF.2006.301604.

J. D. Hol, T. B. Schön, and F. Gustafsson. On resampling algorithms for particle filters. In Proceedings of Nonlinear Statistical Signal Processing Workshop, Cambridge, UK, Sept. 2006a. doi:10.1109/NSSPW.2006.4378824.

G. Hendeby, J. D. Hol, R. Karlsson, and F. Gustafsson. A graphics processing unit implementation of the particle filter. In Proceedings of European Signal Processing Conference, Poznań, Poland, Sept. 2007.

## 1.1 System overview

The pose of a camera is estimated in real-time by fusing measurements from inertial sensors (accelerometers and rate gyroscopes) and a camera. The non-linear state estimation is based on a multi-rate extended Kalman filter using a dynamic model with 22 states, where 100 Hz inertial measurements and 12.5 Hz correspondences from vision are processed. The vision measurements are based on natural landmarks, which are detected guided by pose predictions. The measurements from the sensors are used directly rather than being processed to a vision based pose or an inertial based pose. Accurate short time pose estimates are available using the information from the IMU, reducing the need for fast vision updates, which would otherwise require high frame rates and compute.

Coordinate systems used:

- Earth  $e$ : It is fixed to earth and generally vertically aligned. The camera pose is estimated w.r.t. it and features of the scene are modeled in it.
- Camera  $c$ : It is attached to the moving camera. Its origin is located in the optical center of the camera, with the  $z$ -axis pointing along the optical axis. The camera acquires its images in the image plane ( $i$ ). This plane is perpendicular to the optical axis.
- Body  $b$ : It is the coordinate system of the IMU attached by a constant translation and rotation w.r.t. the camera.

These coordinate systems are used to denote geometric quantities, for instance,  $\mathbf{c}^e$  is the position of the camera coordinate system expressed in the earth system and  $R^{cb}$  is the rotation matrix from the body system to the camera system.

The IMU consists of 1200 deg/s gyroscopes and 5g accelerometers. The signals from the inertial components of the IMU are synchronously measured at 100 Hz using a 16 bit A/D converter. A temperature sensor is added to compensate for the temperature dependency of the different sensing components. The assembly containing the gyroscopes and accelerometers is subjected to a calibration procedure to calibrate for the exact physical alignment of each component, the gains, the offsets, and the temperature relations of the gains and offsets. With these, a 3D angular velocity vector and a 3D accelerometer vector, both resolved in the body coordinate system, are computed.

The calibrated gyroscope signal  $\mathbf{y}_{\omega,t}$  contains measurements of the angular velocity  $\boldsymbol{\omega}_{eb,t}^b$  from body to earth ( $eb$ ) expressed in the body coordinate system ( $^b$ ):

$$\mathbf{y}_{\omega,t} = \boldsymbol{\omega}_{eb,t}^b + \boldsymbol{\delta}_{\omega,t}^b + \mathbf{e}_{\omega,t}^b \quad (1)$$

Here,  $\boldsymbol{\delta}_{\omega,t}^b$  is the low-frequency offset fluctuations due to the unmodeled acceleration dependency and  $\mathbf{e}_{\omega,t}^b$  is zero mean white noise. A change in orientation is obtained by integration of the gyroscope signal without

relying on any infrastructure. However, accuracy in orientation deteriorates for periods longer than a few seconds.

The calibrated accelerometer signal  $\mathbf{y}_{a,t}$  contains measurements of the combination of the body acceleration vector  $\ddot{\mathbf{b}}_t$  and the gravity vector  $\mathbf{g}$  both expressed in the body coordinate system  ${}^b$ :

$$\mathbf{y}_{a,t} = \ddot{\mathbf{b}}_t^b - \mathbf{g}_t^b + \boldsymbol{\delta}_{a,t}^b + \mathbf{e}_{a,t}^b \quad (2)$$

Here,  $\boldsymbol{\delta}_{a,t}^b$  is the low-frequency offset and  $\mathbf{e}_{a,t}^b$  is zero mean white noise. Gravity expressed in body coordinates depends on the orientation of the sensor unit. Once the orientation is known, the accelerometer signal is used to estimate the acceleration, or alternatively, once the acceleration is known, the direction of the vertical is estimated. Acceleration is integrated twice to obtain change in position, as long as an accurate orientation estimate is available from the gyroscopes. However, accuracy of position change will deteriorate quickly as a result of double integration and orientation errors.

The camera has a perspective lens with focal length of 3.2 mm and captures images with a resolution of 320×240 pixels at a frame-rate of 12.5 Hz. The camera is triggered by the IMU clock allowing for synchronized measurements. Extracting camera position and orientation from images is a known and well studied problem in computer vision. The key ingredient is to find correspondences, i.e., relations between features in the image which correspond to an element in the scene model. Point correspondences  $\mathbf{z}^c \leftrightarrow \mathbf{z}^i$  are the relation between 3D points  $\mathbf{z}^c$  and 2D image points  $\mathbf{z}^i$ . For a perspective lens and a pinhole camera, the correspondence relation is

$$\mathbf{z}^i = \begin{bmatrix} f z_c^x / z_c^z \\ f z_c^y / z_c^z \end{bmatrix} + \mathbf{e}^i \quad (3a)$$

or equivalently,

$$\mathbf{0} \approx \begin{bmatrix} -f \mathbf{I}_2 & \mathbf{z}_t^i \end{bmatrix} \mathbf{z}_t^c = \begin{bmatrix} -f \mathbf{I}_2 & \mathbf{z}_t^i \end{bmatrix} \mathbf{R}_t^{cc} (\mathbf{z}^c - \mathbf{z}_t^c) \quad (3b)$$

where  $f$  is the focal length,  $\mathbf{I}_2$  is  $2 \times 2$  identity matrix,  $\mathbf{e}_t^i$  is zero-mean white noise. The camera pose depends on the rotation matrix  $\mathbf{R}^{cc}$  and the position  $\mathbf{c}^c$ . Hence, given sufficient correspondences and a calibrated camera, the camera pose can be found.

The computer vision implementation is based on a sum of absolute difference (SAD) block matcher in combination with a planar patch or free-form surface model of the scene. Both pixel data and 3D positions are stored for each feature. While tracking, search templates are generated by warping the patches in the model according to homographies calculated from the latest prediction of the camera pose. These templates are then matched with the current calibrated camera image using the block matcher. In this way correspondences are generated.

## 1.2 Sensor fusion

Vision in combination with the map gives accurate absolute pose information at a low rate, but experiences problems during moderately fast motions. The IMU provides high rate relative pose information regardless of the motion speed, but becomes inaccurate after a short period of time. By fusing information from both sources it is possible to obtain robust camera pose estimates.

One way is to extend vision-based methods by using pose predictions from the IMU. These pose predictions are used to determine where in the image the features are to be expected. Once detected, the features are used to calculate the pose and this pose is then used as a starting point for the next pose prediction by

the IMU. Another way is to consider IMU to be the main sensor, which is quite common in the navigation industry. Then vision is used for error correction, similar to how radio beacons or global positioning system (GPS) are used to correct the drift in an inertial navigation system (INS).

The objective in filtering is to extract as much information as possible from the measurements. More specifically, this amounts to finding the best possible estimate of the filtering probability density function (pdf)  $p(x_t|y_{1:t})$  where  $y_{1:t} \triangleq \{y_1, \dots, y_t\}$

The objective in sensor fusion is to recursively in time estimate the state in the dynamic model,

$$x_{t+1} = f_t(x_t, u_t, v_t) \quad (4a)$$

$$y_t = h_t(x_t, u_t, e_t) \quad (4b)$$

where  $x_t \in \mathbb{R}^{n_x}$  denotes state estimate,  $y_t \in \mathbb{R}^{n_y}$  denotes measurements from sensors,  $v_t$  and  $e_t$  denote the stochastic process and measurement noise respectively,  $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_x}$  is process model that describes evolution of state over time, and  $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_e} \rightarrow \mathbb{R}^{n_y}$  is measurement model that describes how measurements from IMU and camera relate to the state.

The goal is to infer information from measurements  $y_t$  onto the state  $x_t$  by computing the filtering pdf  $p(x_t|y_{1:t})$ . The filtering pdf contains everything there is to know about the state at time  $t$ , given the information in all the past measurements  $y_{1:t}$ . The approximation of  $p(x_t|y_{1:t})$  is used to form different (point) estimates, including maximum likelihood estimates, confidence intervals and expectation estimate  $\hat{x}_t = E(x_t|y_{1:t})$ . The key element in solving the nonlinear state estimation problem in real-time is the propagation of  $p(x_t|y_{1:t})$  over time.

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int p(y_t|x_t)p(x_t|y_{1:t-1}) dx_t} \quad (5a)$$

$$p(x_{t+1}|y_{1:t}) = \int p(x_{t+1}|x_t)p(x_t|y_{1:t}) dx_t \quad (5b)$$

Eqs. 5a and 5b are called measurement update and time update, respectively. The sensor fusion problem has now been reduced to propagating eq. 5 over time as new measurements arrive. When eq. 4 is linear and noise is Gaussian, then involved densities will be Gaussian and the updating recursions are given by the Kalman Filter (KF). It is often sufficient to assume that the noise enters additively, according to

$$x_{t+1} = f_t(x_t) + v_t \quad (6a)$$

$$y_t = h_t(x_t) + e_t \quad (6b)$$

Additive noise modifies 5 to

$$p(x_{t+1}|x_t) = p_{v_t}(x_{t+1} - f_t(x_t)) \quad (7a)$$

$$p(y_t|x_t) = p_{e_t}(y_t - h_t(x_t)) \quad (7b)$$

where  $p_{v_t}(\cdot)$  and  $p_{e_t}(\cdot)$  denote the pdf's for the noise  $v_t$  and  $e_t$ , respectively.

The state vector consists of position of the IMU (the body coordinate system) expressed in the earth system  $b^e$ , its velocity  $\dot{b}^e$  and acceleration  $\ddot{b}^e$ , the orientation (quaternion) of the body wrt earth  $q^{be}$ , its

angular velocity  $\omega_{eb}^b$ , the gyroscope bias  $\delta_\omega^b$  and the accelerometer bias  $\delta_a^b$ . The state vector has dimension 22 and is given by

$$x_t = \begin{bmatrix} \mathbf{b}_t^e & \dot{\mathbf{b}}_t^e & \ddot{\mathbf{b}}_t^e & q_t^{be} & \boldsymbol{\omega}_{eb,t}^b & \boldsymbol{\delta}_{\omega,t}^b & \boldsymbol{\delta}_{a,t}^b \end{bmatrix}^T \quad (8)$$

The process model is given by

$$\mathbf{b}_{t+1}^e = \mathbf{b}_t^e + T\dot{\mathbf{b}}_t^e + \frac{T^2}{2}\ddot{\mathbf{b}}_t^e \quad (9a)$$

$$\dot{\mathbf{b}}_{t+1}^e = \dot{\mathbf{b}}_t^e + T\ddot{\mathbf{b}}_t^e \quad (9b)$$

$$\ddot{\mathbf{b}}_{t+1}^e = \ddot{\mathbf{b}}_t^e + \mathbf{v}_{b,t}^e \quad (9c)$$

$$q_{t+1}^{be} = \exp\left(-\frac{T}{2}\boldsymbol{\omega}_{eb,t}^b\right) \odot q_t^{be} \quad (9d)$$

$$\boldsymbol{\omega}_{eb,t+1}^b = \boldsymbol{\omega}_{eb,t}^b + \mathbf{v}_{\omega,t}^b \quad (9e)$$

$$\boldsymbol{\delta}_{\omega,t+1}^b = \boldsymbol{\delta}_{\omega,t}^b + \mathbf{v}_{\delta_\omega,t}^b \quad (9f)$$

$$\boldsymbol{\delta}_{a,t+1}^b = \boldsymbol{\delta}_{a,t}^b + \mathbf{v}_{\delta_a,t}^b \quad (9g)$$

where the quaternion multiplication and exponential are defined according to

$$\begin{bmatrix} p_0 \\ \mathbf{p} \end{bmatrix} \odot \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix} \triangleq \begin{bmatrix} q_0 p_0 - \mathbf{p} \cdot \mathbf{q} \\ q_0 \mathbf{q} + q_0 \mathbf{p} + \mathbf{p} \times \mathbf{q} \end{bmatrix} \quad (10a)$$

$$\exp(\mathbf{v}) \triangleq \begin{bmatrix} \cos \|\mathbf{v}\| \\ \frac{\mathbf{v}}{\|\mathbf{v}\|} \sin \|\mathbf{v}\| \end{bmatrix} \quad (10b)$$

The measurement models are given by

$$\mathbf{y}_{a,t} = R_t^{be}(\ddot{\mathbf{b}}_t^e - \mathbf{g}^e) + \boldsymbol{\delta}_{a,t}^b + \mathbf{e}_{a,t}^b \quad (11a)$$

$$\mathbf{y}_{\omega,t} = \boldsymbol{\omega}_{eb,t}^b + \boldsymbol{\delta}_{\omega,t}^b + \mathbf{e}_{\omega,t}^b \quad (11b)$$

$$\mathbf{y}_{c,t} = \begin{bmatrix} -f\mathbf{I}_2 & \mathbf{z}_t^i \end{bmatrix} \mathbf{R}^{cb}(\mathbf{R}_t^{be}(\mathbf{z}_t^e - \mathbf{b}_t^e) - \mathbf{c}^b) + \mathbf{e}_{a,t}^b \quad (11c)$$

The rotation matrix  $\mathbf{R}_t^{be}$  is constructed from  $q_t^{be}$ . The transformation from body to camera coordinate system is included in vision measurement.

---

**Algorithm 1:** Orientation estimation using EKF with orientation deviation states [2]

---

**Input:** Inertial data  $\{\mathbf{y}_{a,t}, \mathbf{y}_{\omega,t}\}_{t=1}^N$ , magnetometer data  $\{\mathbf{y}_{m,t}\}_{t=1}^N$

**Output:** Orientation estimate and covariance  $\mathbf{P}_{t|t}$  for  $t = 1, \dots, N$

- 1: Compute  $\tilde{q}_1^{nb}$  and  $\Sigma_i$ .
  - 2: Set  $\tilde{q}_{1|1}^{nb} = \tilde{q}_1^{nb}$  and  $\mathbf{P}_{1|1} = \Sigma_i$ .
  - 3: **for**  $t = 2$  **to**  $N$  **do**
  - 4:   **Time update:**
  - 5:    $\mathbf{p}_t = \mathbf{p}_{t-1} + T \mathbf{v}_{t-1}^n + \frac{T^2}{2} (\mathbf{R}_{t-1}^{nb} \mathbf{y}_{a,t-1} + \mathbf{g}^n)$
  - 6:    $\mathbf{v}_t = \mathbf{v}_{t-1} + T (\mathbf{R}_{t-1}^{nb} \mathbf{y}_{a,t-1} + \mathbf{g}^n)$
  - 7:    $\tilde{q}_{t|t-1}^{nb} = \tilde{q}_{t-1|t-1}^{nb} \odot \exp_q \left( \frac{T}{2} \mathbf{y}_{\omega,t-1} \right)$
  - 8:    $\mathbf{F}_{t-1} = \begin{bmatrix} \mathbf{I}_3 & T\mathbf{I}_3 & -\frac{T^2}{2} [\tilde{\mathbf{R}}_{t|t}^{nb} \mathbf{y}_{a,t} \times] \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & -T [\tilde{\mathbf{R}}_{t|t}^{nb} \mathbf{y}_{a,t} \times] \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \end{bmatrix}$
  - 9:    $\mathbf{G}_{t-1} = \begin{bmatrix} \mathbf{I}_6 & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & T\tilde{\mathbf{R}}_{t|t-1} \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \Sigma_a & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \Sigma_a & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \Sigma_\omega \end{bmatrix}$
  - 10:    $\mathbf{P}_{t|t-1} = \mathbf{F}_{t-1} \mathbf{P}_{t-1|t-1} \mathbf{F}_{t-1}^\top + \mathbf{G}_{t-1} \mathbf{Q} \mathbf{G}_{t-1}^\top$  {approximation of covariance}
  - 11:   **Measurement update:**
  - 12:    $\boldsymbol{\varepsilon}_t = \mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1} = \begin{bmatrix} \mathbf{y}_{a,t} \\ \mathbf{y}_{m,t} \end{bmatrix} - \begin{bmatrix} -\tilde{\mathbf{R}}_{t|t-1}^{bn} \mathbf{g}^n \\ \tilde{\mathbf{R}}_{t|t-1}^{bn} \mathbf{m}^n \end{bmatrix}$
  - 13:    $\mathbf{H}_t = \mathbf{I}_3, \quad \mathbf{R} = \Sigma_p$  ;
  - 14:    $\mathbf{S}_t = \mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^\top + \mathbf{R} \quad \mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}_t \mathbf{S}_t^{-1}$
  - 15:    $\hat{\boldsymbol{\eta}}_t^n = \mathbf{K}_t \boldsymbol{\varepsilon}_t$  {State variable update}
  - 16:    $\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{S}_t \mathbf{K}_t^\top$  {Covariance update}
  - 17:   **Linearization point update:**
  - 18:    $\tilde{q}_{t|t}^{nb} = \exp_q \left( \frac{\hat{\boldsymbol{\eta}}_t^n}{2} \right) \odot \tilde{q}_{t|t-1}^{nb}$
  - 19: **end for**
-

---

**Algorithm 2:** Orientation estimation using EKF with orientation deviation states [2]

---

**Input** : Inertial data  $\{\mathbf{y}_{a,t}, \mathbf{y}_{\omega,t}\}_{t=1}^N$ , magnetometer data  $\{\mathbf{y}_{m,t}\}_{t=1}^N$

**Output** : Orientation estimate and covariance  $\mathbf{P}_{t|t}$  for  $t = 1, \dots, N$

---

```

1  Compute  $\tilde{q}_1^{nb}$  and  $\Sigma_i$ .
2  Set  $\tilde{q}_{1|1}^{nb} = \tilde{q}_1^{nb}$  and  $\mathbf{P}_{1|1} = \Sigma_i$ .
3  for  $t = 2$  to  $N$  do
    /* Time update: */
4     $\mathbf{p}_t = \mathbf{p}_{t-1} + T\mathbf{v}_{t-1}^n + \frac{T^2}{2} (\mathbf{R}_{t-1}^{nb} \mathbf{y}_{a,t-1} + \mathbf{g}^n)$ 
5     $\mathbf{v}_t = \mathbf{v}_{t-1} + T (\mathbf{R}_{t-1}^{nb} \mathbf{y}_{a,t-1} + \mathbf{g}^n)$ 
6     $\tilde{q}_{t|t-1}^{nb} = \tilde{q}_{t-1|t-1}^{nb} \odot \exp_q \left( \frac{T}{2} \mathbf{y}_{\omega,t-1} \right)$ 
7     $\mathbf{F}_{t-1} = \begin{bmatrix} \mathbf{I}_3 & T\mathbf{I}_3 & -\frac{T^2}{2} [\tilde{\mathbf{R}}_{t|t}^{nb} \mathbf{y}_{a,t} \times] \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & -T [\tilde{\mathbf{R}}_{t|t}^{nb} \mathbf{y}_{a,t} \times] \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \end{bmatrix}$ 
8     $\mathbf{G}_{t-1} = \begin{bmatrix} \mathbf{I}_6 & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & T\tilde{\mathbf{R}}_{t|t-1} \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \Sigma_a & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \Sigma_a & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \Sigma_\omega \end{bmatrix}$ 
9     $\mathbf{P}_{t|t-1} = \mathbf{F}_{t-1} \mathbf{P}_{t-1|t-1} \mathbf{F}_{t-1}^\top + \mathbf{G}_{t-1} \mathbf{Q} \mathbf{G}_{t-1}^\top$  // approximation of covariance
    /* Measurement update: */
10    $\boldsymbol{\varepsilon}_t = \mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1} = \begin{bmatrix} \mathbf{y}_{a,t} \\ \mathbf{y}_{m,t} \end{bmatrix} - \begin{bmatrix} -\tilde{\mathbf{R}}_{t|t-1}^{bn} \mathbf{g}^n \\ \tilde{\mathbf{R}}_{t|t-1}^{bn} \mathbf{m}^n \end{bmatrix}$ 
11    $\mathbf{H}_t = \mathbf{I}_3, \quad \mathbf{R} = \Sigma_p$ 
12    $\mathbf{S}_t = \mathbf{H}_t \mathbf{P}_{t|t} \mathbf{H}_t^\top + \mathbf{R} \quad \mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}_t \mathbf{S}_t^{-1}$ 
13    $\hat{\boldsymbol{\eta}}_t^n = \mathbf{K}_t \boldsymbol{\varepsilon}_t$  // State variable update
14    $\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t \mathbf{S}_t \mathbf{K}_t^\top$  // Covariance update
    /* Linearization point update: */
15    $\tilde{q}_{t|t}^{nb} = \exp_q \left( \frac{\hat{\boldsymbol{\eta}}_t^n}{2} \right) \odot \tilde{q}_{t|t-1}^{nb}$ 
16 end for

```

---

## 2 Reviews

ID, Title, Journal	Author,	Research areas, Tools	Objectives, Discussion	Methodology,	Conclusions, My inference, Gaps
<ul style="list-style-type: none"> <li>• authyear</li> <li>• paper title</li> <li>• firstname1 surname1 and firstname2 surname2</li> <li>• journal</li> </ul>		<ul style="list-style-type: none"> <li>• keyword1</li> <li>• keyword2</li> </ul>	<ul style="list-style-type: none"> <li>• objec</li> <li>• method</li> <li>• concl</li> </ul>		<ul style="list-style-type: none"> <li>• gap</li> </ul>

ID, Title, Author, Journal	Research areas, Tools	Objectives, Methodology, Discussion	Conclusions, My inference, Gaps
<ul style="list-style-type: none"> <li>• <b>aydin2006</b></li> <li>• <i>Quaternion Based Inverse Kinematics for Industrial Robot Manipulators with Euler Wrist</i></li> <li>• Yavuz Aydin and Serdar Kucuk</li> <li>• IEEE 3rd International Conference on Mechatronics (ICM 2006)</li> </ul>	<ul style="list-style-type: none"> <li>• wrist IK</li> <li>• quaternion-vector pair for 6-dof pose</li> </ul>	<ul style="list-style-type: none"> <li>• Double quaternion instead of dual quaternion</li> <li>• Approximate translation magnitue <math>d</math> as rotation by angle <math>\psi = \frac{d}{\bar{R}}</math> about normalised vector <math>d</math>.</li> </ul>	<ul style="list-style-type: none"> <li>• Results obtained by using double quaternions are coordinate frame invariant.</li> </ul>
<ul style="list-style-type: none"> <li>• <b>ge1998</b></li> <li>• <i>Double quaternions for motion interpolation</i></li> <li>• Ge, Q J and Varshney, Amitabh and Menon, Jai P and Chang, Chu-Fei</li> <li>• —</li> </ul>	<ul style="list-style-type: none"> <li>• dual quaternions</li> </ul>	<ul style="list-style-type: none"> <li>• Double quaternion instead of dual quaternion</li> <li>• Approximate translation magnitue <math>d</math> as rotation by angle <math>\psi = \frac{d}{\bar{R}}</math> about normalised vector <math>d</math>.</li> </ul>	<ul style="list-style-type: none"> <li>• Results obtained by using double quaternions are coordinate frame invariant.</li> </ul>
<ul style="list-style-type: none"> <li>• <b>authyear</b></li> <li>• <i>paper title</i></li> <li>• firstname1 surname1 and firstname2 surname2</li> <li>• journal</li> </ul>	<ul style="list-style-type: none"> <li>• keyword1</li> <li>• keyword2</li> </ul>	<ul style="list-style-type: none"> <li>• objec</li> <li>• method</li> <li>• concl</li> </ul>	<ul style="list-style-type: none"> <li>• gap</li> </ul>
<ul style="list-style-type: none"> <li>• <b>authyear</b></li> <li>• <i>paper title</i></li> <li>• firstname1 surname1 and firstname2 surname2</li> <li>• journal</li> </ul>	<ul style="list-style-type: none"> <li>• keyword1</li> <li>• keyword2</li> </ul>	<ul style="list-style-type: none"> <li>• objec</li> <li>• method</li> <li>• concl</li> </ul>	<ul style="list-style-type: none"> <li>• gap</li> </ul>



ID, Title, Author, Journal	Research areas, Tools	Objectives, Methodology, Discussion	Conclusions, My inference, Gaps
<ul style="list-style-type: none"> <li>• <b>ge1998</b></li> <li>• <i>Double quaternions for motion interpolation</i></li> <li>• Ge, Q J and Varshney, Amitabh and Menon, Jai P and Chang, Chu-Fei</li> <li>• —</li> </ul>	<ul style="list-style-type: none"> <li>• dual quaternions</li> </ul>	<ul style="list-style-type: none"> <li>• Double quaternion instead of dual quaternion</li> <li>• Approximate translation magnitue <math>d</math> as rotation by angle <math>\psi = \frac{d}{\bar{R}}</math> about normalised vector <math>d</math>.</li> </ul>	<ul style="list-style-type: none"> <li>• Results obtained by using double quaternions are coordinate frame invariant.</li> </ul>
<ul style="list-style-type: none"> <li>• <b>thomas</b></li> <li>• <i>Approaching Dual Quaternions from Matrix Algebra</i></li> <li>• Frederico Thomas</li> <li>• —</li> </ul>	<ul style="list-style-type: none"> <li>• double quaternion derivation</li> </ul>		
End of Notes			

## References

- [1] Ramkumar Gandhinathan and Lentin Joseph. *ROS Robotics Projects: Build and control robots powered by the Robot Operating System, Machine Learning, and Virtual Reality*. Packt Publishing, 2 edition, 2019.
- [2] Manon Kok, Jeroen D. Hol, and Thomas B. Schön. Using inertial sensors for position and orientation estimation. *Foundations and Trends in Signal Processing*, 11(1-2):1–153, 2017.