Prediction Error Method Estimation for Simultaneous Localisation and Mapping

Zoran Sjanic and Martin A. Skoglund

Original Publication:

Zoran Sjanic and Martin A. Skoglund, Prediction Error Method Estimation for Simultaneous Localisation and Mapping, 2016, Proceedings of the 19th International Conference on Information Fusion (FUSION), July 4-8 2016, 927-934.

Prediction Error Method Estimation for Simultaneous Localisation and Mapping

Zoran Sjanic*,**
*Saab AB
Linköping
Sweden
Email: zoran.sjanic@saabgroup.com

Martin A. Skoglund**
**Division of Automatic Control
Department of Electrical Engineering
Linköping University
Linköping, Sweden
Email: ms@isy.liu.se

***Abstract*—This paper presents a batch estimation method for Simultaneous Localization and Mapping (SLAM) using the Prediction Error Method (PEM). The estimation problem considers landmarks as parameter while treating dynamics using state space models. The gradient needed for parameter estimation is computed recursively using an Extended Kalman Filter (EKF). Results using simulations with a monocular camera and inertial sensors are presented and compared to a Nonlinear Least-Squares (NLS) estimator. The presented method produce both lower RMSE's and scale better to the batch length.**

I. Introduction

This paper investigates the use of the Prediction Error Method (PEM), see e.g., [1], as a way to utilize the particular structure of Nonlinear Least-Squares (NLS) problems encountered in Simultaneous Localization And Mapping (SLAM). The aim in SLAM is to estimate a moving platform's position and orientation while mapping the observed environment, [2, 3]. A strong trend in SLAM algorithm research is (incremental) batch optimisation which usually solves some form of NLS problem [4–8]. These often suffers from poor complexity scaling, usually quadratic in batch length as both motion of the platform and landmarks are considered as parameters. The idea behind using PEM is to model the landmarks as (static) parameters included in the dynamic model used for the platform's motion which is modeled as dynamic states. This is a classic system identification setup and PEM is one, rather standard and successful, way of estimating the parameters of the system [1]. The main advantage of this approach is that problem is split into an optimisation part, where landmarks are estimated, and predictor part, where the output of the system (measurements) is predicted. This is done utilizing the time series nature of data through a filter which also output a complete state trajectory estimate. This property allows also for the gradient of the predictor w.r.t. parameters to be calculated recursively (for a particular choice of the predictor) so that no numerical gradients are necessary. This approach is different from both the standard EKF-SLAM approach, [2], where both the platform's motion and the landmarks are considered as states and NLS approaches where everything is considered as parameters. This split into two parts allows for computation complexity reduction, the optimisation problem is smaller (in the number of parameters) than NLS problem, and the prediction problem is smaller than usual EKF-SLAM problem since the number of states is constant. Basically, the optimisation problem will scale with the number of parameters (landmarks) and the predictor part will scale linearly with the batch length (time steps). This complexity reduction is the main motivation behind using PEM as an estimation method for SLAM, and this will be demonstrated with the empirical experiments comparing PEM and NLS complexities. Since both PEM and NLS approaches are optimisation ones while EKF-SLAM is a filtering approach without gradient or Jacobian calculation and no comparison with it is done.

The outline of the paper is the following; In Section II a brief description of PEM and the model structure is given; In Section III present expressions for a recursive gradient computation using EKF; In Section IV the example models that are used for evaluation are presented; In Section V Monte Carlo simulation results are presented and compared to NLS-SLAM, as well as some empirical computation complexity comparison to NLS-SLAM; Section VI ends the paper with conclusions and directions for future work.

II. The Prediction Error Method

Assume that measurements, $\{y_t\}_{t=1}^N$, from a dynamic system are available. Suppose also that a model of this system is parametrised with some (unknown) parameters, θ, and that we want to estimate these using PEM. This is done by minmising the sum of the squared prediction errors

$$V(\theta) = \frac{1}{2N}\sum_{t=1}^{N} \|y_t - \hat{y}_{t|t-1}(\theta)\|_2^2 \tag{1}$$

as

$$\hat{\theta} = \arg\min_{\theta} V(\theta) \tag{2}$$

where $\hat{y}_{t|t-1}(\theta)$ are the (parameter dependent) predicted measurements of the system at time t given all the information until time $t-1$. The exact choice of the predictor to produce these is usually a matter of the nature of the system that is

considered. The formulation above is a standard nonlinear least-squares problem and any NLS method, such as the Levenberg-Marquardt method [9, 10], can be used to solve it. In this context it is quite advantageous if the predictor is an analytical function of the parameters and if its gradient w.r.t. to parameters is available, because that both simplifies and speeds up the iterative optimisation procedure, compared to using numerical methods for calculating the gradient.

A quite general system description is a discrete-time nonlinear state space model of the form

$$x_{t+1} = f_t(x_t, u_t, w_t, \theta) \tag{3a}$$
$$y_t = h_t(x_t, \theta) + e_t \tag{3b}$$

where the state dynamics is modeled with the function $f_t(\,\cdot\,)$, u_t is the known input, w_t is an unknown system noise, the measurement to state relation is represented with function $h_t(\,\cdot\,)$, and e_t is the measurement noise. In general, it is assumed that both the dynamics and measurement equation are dependent on parameters θ, which is explicitly expressed in the equations. Given this model, any nonlinear filter can be used to produce the predicted measurements that is used in the optimisation procedure. In this particular case we will use EKF in a similar manner to e.g., [11, 12]. EKF as a predictor has an advantage because it is simple to implement and has a possibility to explicitly calculate the gradient of the loss function.

III. EKF with Recursive Gradient Calculation

In the following section we will give expressions for the time and measurement updates in the EKF together with the recursive calculations needed to obtain the gradient of the loss function in (1). The time update from the EKF gives a predicted state estimate at time t given all the measurements up to time $t-1$, $\hat{x}_{t|t-1}$. This prediction of the state together with the measurement model in (3b) can be used to obtain the predicted measurement needed for the PEM loss function, namely $\hat{y}_{t|t-1}(\theta) = h_t(\hat{x}_{t|t-1}(\theta), \theta)$. This relation will be used in the continuation to derive all the necessary expressions that are used. Notice also that prediction of the state depends on the parameters, which is a consequence of the EKF estimation procedure.

For the functions that are dependent on both states and parameters, there will be two different gradients/Jacobians, one dependent on the states and one dependent on the parameters. For the ones dependent on the states we will suppress the dependence notation, while it will not be suppressed for the parameters dependence or when it is important to emphasize the dependence. For example, for function $h(x, \theta)$, the Jacobian w.r.t. x will only be called H, while the Jacobian w.r.t. θ will be called H_θ.

The gradient of (1) w.r.t. a parameter θ is

$$\begin{aligned}\frac{\partial}{\partial\theta}V(\theta) &= \frac{1}{2N}\sum_{t=1}^{N}\frac{\partial}{\partial\theta}\|y_t - h_t(\hat{x}_{t|t-1}(\theta),\theta)\|_2^2 \\ &= \frac{1}{N}\sum_{t=1}^{N} J_t^T r_t\end{aligned} \tag{4}$$

where

$$r_t = y_t - h_t(\hat{x}_{t|t-1}(\theta), \theta) \tag{5a}$$
$$J_t = \frac{\partial r_t}{\partial\theta} = -\frac{\partial h_t(x,\theta)}{\partial x}\frac{\partial x}{\partial\theta} - \left.\frac{\partial h_t(x,\theta)}{\partial\theta}\right|_{x=\hat{x}_{t|t-1}(\theta)}. \tag{5b}$$

In the continuation we will denote the Jacobian of the measurement function w.r.t. the states as

$$\left.\frac{\partial h_t(x,\theta)}{\partial x_t}\right|_{x=\hat{x}_{t|t-1}(\theta)} = H_t \tag{6}$$

and the Jacobian of the measurement function w.r.t. the parameters as

$$\left.\frac{\partial h_t(x,\theta)}{\partial\theta}\right|_{x=\hat{x}_{t|t-1}(\theta)} = H_{t,\theta}. \tag{7}$$

The Jacobian of the state vector w.r.t. the parameters is given by the term

$$\left.\frac{\partial x}{\partial\theta}\right|_{x=\hat{x}_{t|t-1}(\theta)} = X_{t,\theta}. \tag{8}$$

This notation gives that (5b) can be written as

$$J_t = -H_t X_{t,\theta} - H_{t,\theta}. \tag{9}$$

The only term above that does not have an explicit expression is $X_{t,\theta}$. However, due to the recursive nature of the EKF, this term can be calculated in a recursive fashion in the time update. In the next subsections, the necessary equations in order to calculate all the needed terms are given.

A. Time Update

The time update step in EKF is

$$\hat{x}_{t|t-1}(\theta) = f(\hat{x}_{t-1|t-1}(\theta), u_t, \theta) \tag{10a}$$
$$P_{t|t-1}(\theta) = F_{t-1}(\theta)P_{t-1|t-1}(\theta)F_{t-1}(\theta)^T + Q_{t-1}(\theta) \tag{10b}$$
$$F_{t-1}(\theta) = \left.\frac{\partial f(x,u,\theta)}{\partial x}\right|_{u=u_t,\ x=\hat{x}_{t-1|t-1}(\theta)} \tag{10c}$$

where we have explicitly given parameter θ dependence. Given these equations, the Jacobians are

$$\begin{aligned}
X_{t,\theta} =& \frac{\partial f(x,u,\theta)}{\partial x}\frac{\partial x}{\partial \theta} + \frac{\partial f(x,u,\theta)}{\partial \theta}\Big|_{u=u_t,\ x=\hat{x}_{t-1|t-1}(\theta)} \\
=& F_{t-1}(\theta)\widetilde{X}_{t-1,\theta} + F_{t-1,\theta} && \text{(11a)} \\
P_{t|t-1,\theta} =& F_{t-1,x\theta}P_{t-1|t-1}F_{t-1}^T + F_{t-1}P_{t-1|t-1,\theta}F_{t-1}^T \\
&+ F_{t-1}P_{t-1|t-1}F_{t-1,x\theta}^T + Q_{t-1,\theta} && \text{(11b)} \\
Q_{t-1,\theta} =& \frac{\partial Q_{t-1}}{\partial x}\frac{\partial x}{\partial \theta}\Big|_{x=\hat{x}_{t-1|t-1}(\theta)} = Q_{t-1,x}\widetilde{X}_{t-1,\theta} && \text{(11c)} \\
F_{t-1,x\theta} =& \frac{\partial F_{t-1}}{\partial x}\frac{\partial x}{\partial \theta}\Big|_{x=\hat{x}_{t-1|t-1}(\theta)} = F_{t-1,x}\widetilde{X}_{t-1,\theta} && \text{(11d)}
\end{aligned}$$

where we have defined $\widetilde{X}_{t-1,\theta}$ as the Jacobian of the measurement updated state estimate $\hat{x}_{t-1|t-1}$. These expressions give a recursive update of the time updated state Jacobian, $X_{t,\theta}$, and its covariance matrix, $P_{t|t-1,\theta}$, where measurement updated quantities from the previous timestep are used.

B. Measurement Update

The EKF measurement update is given by

$$\begin{aligned}
\hat{x}_{t|t}(\theta) &= \hat{x}_{t|t-1}(\theta) + K_t(\theta)r_t(\theta) && \text{(12a)} \\
P_{t|t}(\theta) &= P_{t|t-1}(\theta) - K_t(\theta)C_t(\theta)^T && \text{(12b)} \\
r_t(\theta) &= y_t - h_t(\hat{x}_{t|t-1}(\theta),\theta) && \text{(12c)} \\
K_t(\theta) &= P_{t|t-1}(\theta)H_t(\theta)^T(H_t(\theta)P_{t|t-1}(\theta)H_t(\theta)^T + R_t)^{-1} \\
&= C_t(\theta)S_t(\theta)^{-1} && \text{(12d)} \\
C_t(\theta) &= P_{t|t-1}(\theta)H_t(\theta)^T && \text{(12e)} \\
S_t(\theta) &= H_t(\theta)P_{t|t-1}(\theta)H_t(\theta)^T + R_t. && \text{(12f)}
\end{aligned}$$

Following the same principle as for time update, the Jacobians of the measurement updated state and its covariance are

$$\begin{aligned}
\widetilde{X}_{t,\theta} &= \frac{\partial \hat{x}_{t|t-1}}{\partial \theta} + \frac{\partial K_t}{\partial \theta}r_t + K_t\frac{\partial r_t}{\partial \theta} \\
&= X_{t,\theta} + K_{t,\theta}r_t + K_tJ_t && \text{(13a)} \\
P_{t|t,\theta} &= P_{t|t-1,\theta} - K_{t,\theta}C_t^T - K_tC_{t,\theta}^T && \text{(13b)}
\end{aligned}$$

and $X_{t,\theta}$ and $P_{t|t-1,\theta}$ are available from the previous time update iteration. The Jacobian of the loss function, J_t, is calculated according to (9).

In the above calculations the terms $K_{t,\theta}$ and $C_{t,\theta}^T$ are needed, and by using their definitions the following expressions are obtained

$$\begin{aligned}
K_{t,\theta} &= C_{t,\theta}S_t^{-1} - C_tS_t^{-1}S_{t,\theta}S_t^{-1} && \text{(14a)} \\
C_{t,\theta}^T &= H_tP_{t|t-1,\theta} + H_{t,x\theta}P_{t|t-1} && \text{(14b)}
\end{aligned}$$

as well as

$$\begin{aligned}
C_{t,\theta} &= P_{t|t-1,\theta}H_t^T + P_{t|t-1}H_{t,x\theta}^T && \text{(15a)} \\
S_{t,\theta} &= H_{t,x\theta}C_t + H_tC_{t,\theta}. && \text{(15b)}
\end{aligned}$$

Algorithm 1 Time Update step with Jacobian calculation.

Require: $\hat{\theta}$, $\hat{x}_{t-1|t-1}$, $P_{t-1|t-1}$, $\widetilde{X}_{t-1,\theta}$, $P_{t-1|t-1,\theta}$, u_t
Ensure: $\hat{x}_{t|t-1}$, $P_{t|t-1}$, $X_{t,\theta}$, $P_{t|t-1,\theta}$

$$\begin{aligned}
&\hat{x}_{t|t-1} = f(\hat{x}_{t-1|t-1}, u_t, \hat{\theta}) \\
&P_{t|t-1} = F_{t-1}P_{t-1|t-1}F_{t-1}^T + Q_{t-1} \\
&\\
&X_{t,\theta} = F_{t-1}\widetilde{X}_{t-1,\theta} + F_{t-1,\theta} \\
&Q_{t-1,\theta} = Q_{t-1,x}\widetilde{X}_{t-1,\theta} \\
&F_{t-1,x\theta} = F_{t-1,x}\widetilde{X}_{t-1,\theta} \\
&P_{t|t-1,\theta} = F_{t-1,x\theta}P_{t-1|t-1}F_{t-1}^T + F_{t-1}P_{t-1|t-1,\theta}F_{t-1}^T \\
&\qquad + F_{t-1}P_{t-1|t-1}F_{t-1,x\theta}^T + Q_{t-1,\theta}
\end{aligned}$$

C. Multirate treatment

In many cases, the time update step is performed in a faster rate than the measurement update, where it is even possible to have different measurements with different rates. For example if the available sensors are the inertial measurement unit (IMU) and camera, the data from IMU is usually available in a higher rate than camera images. The same updates are performed even in that case, but if the measurements are not available at a certain time, only the time update is performed at that time and the state estimate, covariance and their Jacobians are simply copied from predicted to measurement update according to

$$\begin{aligned}
\hat{x}_{t|t} &= \hat{x}_{t|t-1} && \text{(16a)} \\
P_{t|t} &= P_{t|t-1} && \text{(16b)} \\
\widetilde{X}_{t,\theta} &= X_{t,\theta} && \text{(16c)} \\
P_{t|t,\theta} &= P_{t|t-1,\theta}. && \text{(16d)}
\end{aligned}$$

Basically, these boil down to a recursive state, covariance and their Jacobians update according to Algorithms 1 and 2. These calculations give the residual and Jacobian of the loss function that can be used in any NLS solver.

IV. Models

In this section the models that are used for estimation of the states and parameters in the SLAM problem will be specified. The sensors of interest are monocular camera and 6-DOF inertial sensors, i.e., gyroscopes and accelerometers. To reduce the size of state space the inertial sensors are considered as inputs to state dynamics. A minimal 3D point landmark parametrisation is used and its measurement function is given by the pinhole projection model.

A. State Dynamics

The gyroscope and accelerometer signals are considered to be inputs to the process model. The gyroscope signals are denoted $u^\omega = [u_x^\omega, u_y^\omega, u_z^\omega]^T$ where the subscript refers to each axis of the body frame. Similarly the accelerometer signals are denoted $u^\mathsf{a} = [u_x^\mathsf{a}, u_y^\mathsf{a}, u_z^\mathsf{a}]^T$. Both of these are measured in the

Algorithm 2 Measurement Update step with Jacobian calculation.

Require: $\hat{\theta}$, $\hat{x}_{t|t-1}$, $P_{t|t-1}$, $X_{t,\theta}$, $P_{t|t-1,\theta}$
Ensure: $\hat{x}_{t|t}$, $P_{t|t}$, $\widetilde{X}_{t,\theta}$, $P_{t|t,\theta}$

if Measurements y_t available **then**
Note: The calculations below are done only for the observed landmarks
Calculate the Jacobian of the loss function
$J_t = -H_t X_{t,\theta} - H_{t,\theta}$

Perform the Measurement Update
$r_t = y_t - h_t(\hat{x}_{t|t-1}, \hat{\theta})$
$C_t = P_{t|t-1} H_t^T$
$S_t = H_t C_t + R_t$
$K_t = C_t S_t^{-1}$
$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t r_t$
$P_{t|t} = P_{t|t-1} - K_t C_t^T$

$C_{t,\theta} = P_{t|t-1,\theta} H_t^T + P_{t|t-1} H_{t,x\theta}^T$
$C_{t,\theta}^T = H_t P_{t|t-1,\theta} + H_{t,x\theta} P_{t|t-1}$
$S_{t,\theta} = H_{t,x\theta} C_t + H_t C_{t,\theta}$
$K_{t,\theta} = C_{t,\theta} S_t^{-1} - C_t S_t^{-1} S_{t,\theta} S_t^{-1}$
$\widetilde{X}_{t,\theta} = X_{t,\theta} + K_{t,\theta} r_t + K_t J_t$
$P_{t|t,\theta} = P_{t|t-1,\theta} - K_{t,\theta} C_t^T - K_t C_{t,\theta}^T$
else
No measurements available, only copy the predicted quantities
$\hat{x}_{t|t} = \hat{x}_{t|t-1}$
$P_{t|t} = P_{t|t-1}$
$\widetilde{X}_{t,\theta} = X_{t,\theta}$
$P_{t|t,\theta} = P_{t|t-1,\theta}$
end if

sensor body frame. A discretised process model for the three-dimensional position, velocity and rotation, $[p_t^T, v_t^T, q_t^T]^T$, in the local navigation frame is then

$$p_{t+1} = p_t + T v_t + \frac{T^2}{2} \mathsf{R}^T(q_t) \left(u_t^{\mathsf{a}} + g^b + w_t^{\mathsf{a}}\right) \tag{17a}$$

$$v_{t+1} = v_t + T \mathsf{R}^T(q_t) \left(u_t^{\mathsf{a}} + g^b + w_t^{\mathsf{a}}\right) \tag{17b}$$

$$q_{t+1} = \exp\left(\frac{T}{2} S_\omega(u_t^\omega + w_t^\omega)\right) q_t \tag{17c}$$

where T denotes the sampling interval, $\mathsf{R}(q_t)$ is a rotation matrix parametrisation of the unit quaternion $q_t = [q_t^0, q_t^1, q_t^2, q_t^3]^T$ which describes the rotation from navigation to body frame, $g^b = \mathsf{R}(q_t) g^n$, is the gravity expressed in the body frame, $g^n = [0, 0, -g]$ is the local gravity vector expressed in the navigation frame where $g \approx 9.82$ and $\exp(\cdot)$ is here considered as the matrix exponential. The noise terms are assumed Gaussian and independent $[(w_t^{\mathsf{a}})^T, (w_t^\omega)^T]^T = w_t \sim \mathcal{N}(0, diag(Q_{\mathsf{a}}, Q_\omega)) = \mathcal{N}(0, Q)$. The skew-symmetric matrix

$$S_\omega(u) = \begin{bmatrix} 0 & -u_x & -u_y & -u_z \\ u_x & 0 & u_z & -u_y \\ u_y & -u_z & 0 & u_x \\ u_z & u_y & -u_x & 0 \end{bmatrix} \tag{18}$$

parametrises the quaternion dynamics. This parametrisation is very similar to reduced-dimension observers in [13]. It is worth noticing that in this case, the dynamics of the system is independent of the landmarks (parameters θ) which simplifies the recursive calculation of the Jacobian a bit. In this case, the term $\partial f / \partial \theta = F_\theta$ is omitted from (11a).

B. Camera Measurements

The monocular camera is modeled as a standard pinhole camera, see cf. [14]. The camera calibration matrix and lens distortion were estimated prior to usage. Since the calibration and distortion are known the undistorted pixels can be pre-multiplied with the inverse of the camera matrix, thus the camera then works as a projective map in Euclidean space, $\mathsf{P} : \mathbb{R}^3 \to \mathbb{R}^2$. The projection is defined as $\mathsf{P}([X, Y, Z]) = [X/Z, Y/Z]$ and the Z coordinate is assumed positive and non-zero since otherwise the point would be behind the camera. Then a normalised camera measurement, $y_t = [u_t, v_t]^T$, of a landmark, θ, at time t is

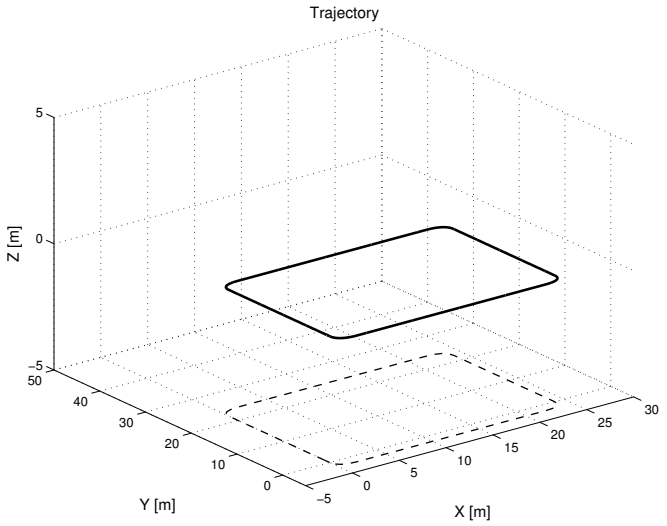
$$y_t = \mathsf{P}\left(\mathsf{R}(q_t)(\theta - p_t)\right) + e_t \tag{19}$$

which relates the pose (position and orientation) of the camera to the 3D location of the point. The measurement noise is assumed i.i.d. Gaussian, $e_t = [e_t^u, e_t^v]^T \sim \mathcal{N}(0, R)$. The correspondence variables at time t, $\mathsf{C}_t = \{c_t^i\} \subseteq \{1, \ldots, M\}$, encode the measurement-landmark assignment, $y_t^i \leftrightarrow \theta^{c_t^i}$. This gives that all observed landmarks at time t are defined as a set $\mathsf{M}_t = \left\{ \theta^{c_t^i} \right\}$ where $c_t^i = j$ if measurement i corresponds to landmark j. At time t the stacked measurement equation, for all observed landmarks at that time, is then

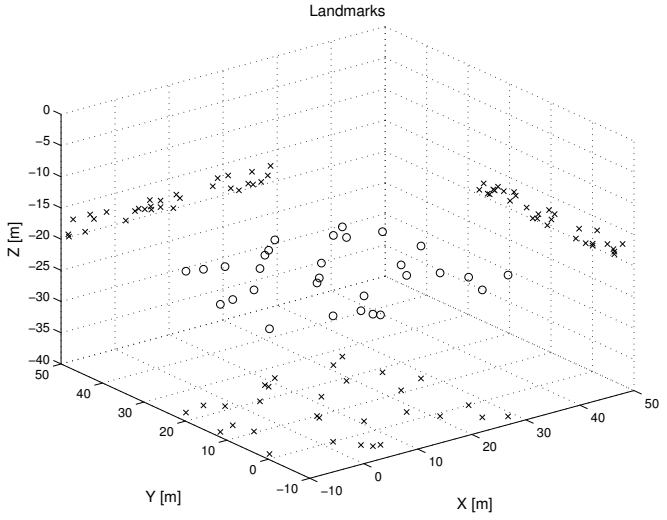
$$\underbrace{\begin{bmatrix} u_t^1 \\ v_t^1 \\ \vdots \\ u_t^{\mathsf{N}_t} \\ v_t^{\mathsf{N}_t} \end{bmatrix}}_{y_t^{\text{cam}}} = \underbrace{\begin{bmatrix} \mathsf{P}\left(\mathsf{R}(q_t)(\theta^{c_t^1} - p_t)\right) \\ \vdots \\ \mathsf{P}\left(\mathsf{R}(q_t)(\theta^{c_t^{\mathsf{N}_t}} - p_t)\right) \end{bmatrix}}_{h_t(x_t, \mathsf{M}_t)} + \underbrace{\begin{bmatrix} e_t^{1u} \\ e_t^{1v} \\ \vdots \\ e_t^{\mathsf{N}_t\, u} \\ e_t^{\mathsf{N}_t\, v} \end{bmatrix}}_{e_t^{\text{cam}}} \tag{20}$$

where N_t is the number of landmark measurements at time t and $e_t^{\text{cam}} \sim \mathcal{N}(0, R_{\text{cam}})$. R_{cam} is a diagonal matrix since all the measurements are assumed to be mutual independent. Note also that first three columns of H_t are equal to $-H_{t,\theta}$ in (9), due to the structure of the measurement function (19).

Solving the correspondence problem in order to find C_t is outside the scope for this work and is here assumed to be correctly solved. However, in practical applications this assumption cannot be guaranteed causing outliers of some kind. These faulty associations will in general bias the PEM-SLAM estimate (just like in any other SLAM method) and

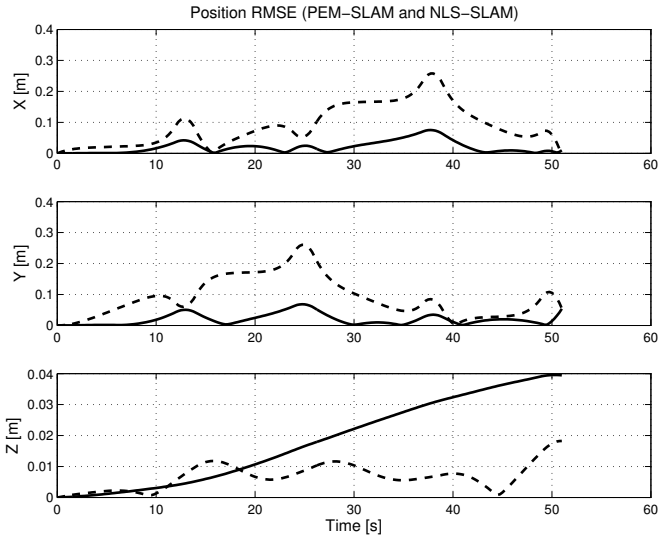


(a) Simulated trajectory used in Monte Carlo simulations with projection on XY-plane for a clearer view (it has constant altitude, i.e., $Z = 0$).

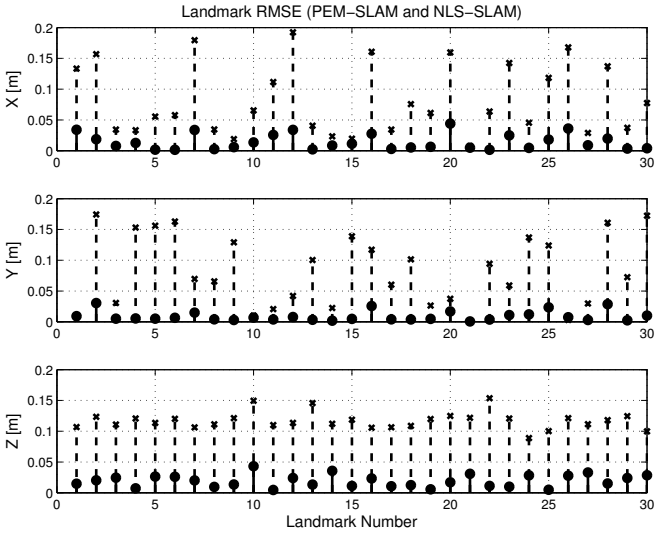


(b) Simulated landmarks used in Monte Carlo simulations with projections on all planes for a clearer view.

Fig. 1: Environment setup used in simulated experiments.



(a) RMS error for the trajectory estimated with PEM-SLAM (solid line) and NLS-SLAM (dashed line) based on 50 MC simulations.



(b) RMS error for all the landmarks estimated with PEM-SLAM (solid line with circles) and NLS-SLAM (dashed line with crosses) for each coordinate based on 50 MC simulations.

Fig. 2: RMS errors for the trajectory and landmarks based on 50 MC simulations for both PEM-SLAM and NLS-SLAM.

some kind of outlier rejection technique should be applied. The alternative can also be a robust norm applied in loss function (1) instead of usual 2-norm, see [1].

V. Results

The performance of the method is demonstrated with Monte Carlo (MC) simulations with the synthetic setup according to Figure 1. Trajectory consists of 205 images at 4Hz and 2050 acceleration and angular rate measurements at 40Hz which gives the total trajectory duration time of 51.25s.

The results based on the 50 MC simulations for both PEM-SLAM and NLS-SLAM, where the noise on the accelerations, angular rates and camera measurements is varied, are shown in Figure 2 for the trajectory and the landmarks, respectively. The noise is sampled from the Gaussian distribution with zero mean and standard deviations $\sigma_a = 1 \cdot 10^{-3}$ $[\text{m/s}^2]$, $\sigma_\omega = 1 \cdot 10^{-4}$ $[^o/\text{s}]$ and $\sigma_{\text{cam}} = 1 \cdot 10^{-4}$ $[\text{m}]$. For the comparison of the execution speed between PEM-SLAM and NLS-SLAM an empirical experiment is done and the resulting times are shown in Figure 3. The comparison is done for the calculation of one iteration step for each method, i.e., the time to calculate the loss function value and the Jacobian. The depicted surface is the ratio between the execution time for NLS-SLAM and PEM-SLAM as a function of batch length (basically the number of images) and the number of

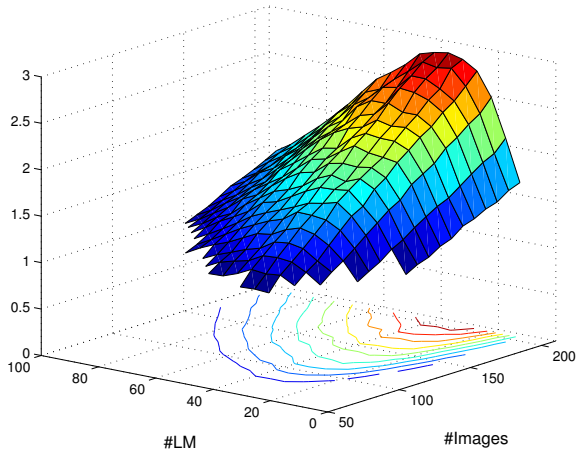


Fig. 3: Ratio of the time used for the residual and Jacobian calculation (for a single iteration) for NLS and PEM as a function of number of images and number of landmarks assuming the worst case (all landmarks are measured the whole time). It is only showing the portion of the surface where the time for NLS is greater than the time for PEM, i.e., when the value is larger than 1.

landmarks. Note that the performance will actually depend on the number of observed landmarks since usually only a portion of the total number is observed in each image (see (20)). In order to get an explicit dependence on the number of landmarks, the simulations are performed in such a way that all the landmarks are observed in every image and this can be seen as a worst case scenario. It can be seen that PEM-SLAM is getting faster when batch length is increasing compared to number of landmarks. This is very interesting result, since the batch length is something that is usually given and fixed and not much can be done about it, while the number of landmarks can be controlled by the user.

VI. Conclusions and Future Work

In this work we have presented how a system identification method, PEM, can be applied to a SLAM problem. This is achieved by considering the map, modeled as discrete landmarks, as parameters in the system, and the motion of the platform as dynamic states. This kind of separation allows for computation complexity reduction compared to other batch methods, like EKF-SLAM or NLS-SLAM. The estimation performance is evaluated with MC simulations on an inertial/visual synthetic and compared to NLS-SLAM, showing the comparable performance.

Furthermore, PEM-SLAM formulation allows for Moving Horizon adaption, opening for online algorithms, which will be the next step of this work.

Acknowledgments

The authors gratefully acknowledge funding from the Vinnova Industry Excellence Center LINK-SIC.

Appendix

The expressions for the Jacobians that are not calculated recursively nor implicitly are given in this Appendix. These Jacobians are F_{t-1}, H_t, $H_{t,\theta}$, $H_{t,x\theta}$ and $Q_{t,x}$. First we define some auxiliary expressions that are used; the rotation matrix $\mathsf{R}(q)$ as a function of the quaternion is (columnwise)

$$\mathsf{R}_{:,1}(q) = \begin{bmatrix} (q^0)^2 + (q^1)^2 - (q^2)^2 - (q^3)^2 \\ 2(q^1q^2 - q^0q^3) \\ 2(q^1q^3 + q^0q^2) \end{bmatrix} \tag{21a}$$

$$\mathsf{R}_{:,2}(q) = \begin{bmatrix} 2(q^1q^2 + q^0q^3) \\ (q^0)^2 - (q^1)^2 + (q^2)^2 - (q^3)^2 \\ 2(q^2q^3 - q^0q^1) \end{bmatrix} \tag{21b}$$

$$\mathsf{R}_{:,3}(q) = \begin{bmatrix} 2(q^1q^3 - q^0q^2) \\ 2(q^2q^3 + q^0q^1) \\ (q^0)^2 - (q^1)^2 - (q^2)^2 + (q^3)^2 \end{bmatrix} \tag{21c}$$

and its Jacobian with respect to q, $\partial\,\mathsf{R}\,/\partial q$, (which is a $3\times3\times4$ tensor), is

$$\frac{\partial\,\mathsf{R}}{\partial q^0} = 2\begin{bmatrix} q^0 & q^3 & -q^2 \\ -q^3 & q^0 & q^1 \\ q^2 & -q^1 & q^0 \end{bmatrix} \tag{22a}$$

$$\frac{\partial\,\mathsf{R}}{\partial q^1} = 2\begin{bmatrix} q^1 & q^2 & q^3 \\ q^2 & -q^1 & q^0 \\ q^3 & -q^0 & -q^1 \end{bmatrix} \tag{22b}$$

$$\frac{\partial\,\mathsf{R}}{\partial q^2} = 2\begin{bmatrix} -q^2 & q^1 & -q^0 \\ q^1 & q^2 & q^3 \\ q^0 & q^3 & -q^2 \end{bmatrix} \tag{22c}$$

$$\frac{\partial\,\mathsf{R}}{\partial q^3} = 2\begin{bmatrix} -q^3 & q^0 & q1 \\ -q^0 & -q^3 & q^2 \\ q^1 & q^2 & q^3 \end{bmatrix}. \tag{22d}$$

The Jacobian of the state dynamics (17) w.r.t. the states is

$$F_{t-1} = \begin{bmatrix} I_3 & TI_3 & \frac{T^2}{2}\frac{\partial\,\mathsf{R}^T}{\partial q_{t-1}}(u^{\mathsf{a}}_{t-1} + g^b) \\ 0 & I_3 & T\frac{\partial\,\mathsf{R}^T}{\partial q_{t-1}}(u^{\mathsf{a}}_{t-1} + g^b) \\ 0 & 0 & \exp\left(\frac{T}{2}S_\omega(u^\omega_{t-1})\right) \end{bmatrix}. \tag{23}$$

In order to calculate the Jacobian of the measurement equation (19) some auxiliary variables are defined as

$$\delta = \theta - p \tag{24a}$$

$$\delta^{\text{cam}} = \mathsf{R}(q)\delta \tag{24b}$$

$$\frac{\partial\delta^{\text{cam}}}{\partial p} = \Delta^{\text{cam}}_p = -\,\mathsf{R}(q) \tag{24c}$$

$$\frac{\partial\delta^{\text{cam}}}{\partial q} = \Delta^{\text{cam}}_q = \frac{\partial\,\mathsf{R}(q)}{\partial q}\delta = \mathsf{R}'\,\delta. \tag{24d}$$

Note that (24d) is a $3 \times 1 \times 4$ tensor. These give now the expression for the Jacobian w.r.t. states for one landmark measurement as

$$H^T = \frac{1}{(\delta_3^{\text{cam}})^2} \begin{bmatrix} -\mathsf{R}_{1,1}\,\delta_3^{\text{cam}} + \delta_1^{\text{cam}}\,\mathsf{R}_{3,1} & -\mathsf{R}_{2,1}\,\delta_3^{\text{cam}} + \delta_2^{\text{cam}}\,\mathsf{R}_{3,1} \\ -\mathsf{R}_{1,2}\,\delta_3^{\text{cam}} + \delta_1^{\text{cam}}\,\mathsf{R}_{3,2} & -\mathsf{R}_{2,2}\,\delta_3^{\text{cam}} + \delta_2^{\text{cam}}\,\mathsf{R}_{3,2} \\ -\mathsf{R}_{1,3}\,\delta_3^{\text{cam}} + \delta_1^{\text{cam}}\,\mathsf{R}_{3,3} & -\mathsf{R}_{2,3}\,\delta_3^{\text{cam}} + \delta_2^{\text{cam}}\,\mathsf{R}_{3,3} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \Delta_{q,1,1}^{\text{cam}}\delta_3^{\text{cam}} - \delta_1^{\text{cam}}\Delta_{q,3,1}^{\text{cam}} & \Delta_{q,2,1}^{\text{cam}}\delta_3^{\text{cam}} - \delta_2^{\text{cam}}\Delta_{q,3,1}^{\text{cam}} \\ \Delta_{q,1,2}^{\text{cam}}\delta_3^{\text{cam}} - \delta_1^{\text{cam}}\Delta_{q,3,2}^{\text{cam}} & \Delta_{q,2,2}^{\text{cam}}\delta_3^{\text{cam}} - \delta_2^{\text{cam}}\Delta_{q,3,2}^{\text{cam}} \\ \Delta_{q,1,3}^{\text{cam}}\delta_3^{\text{cam}} - \delta_1^{\text{cam}}\Delta_{q,3,3}^{\text{cam}} & \Delta_{q,2,3}^{\text{cam}}\delta_3^{\text{cam}} - \delta_2^{\text{cam}}\Delta_{q,3,3}^{\text{cam}} \\ \Delta_{q,1,4}^{\text{cam}}\delta_3^{\text{cam}} - \delta_1^{\text{cam}}\Delta_{q,3,4}^{\text{cam}} & \Delta_{q,2,4}^{\text{cam}}\delta_3^{\text{cam}} - \delta_2^{\text{cam}}\Delta_{q,3,4}^{\text{cam}} \end{bmatrix}. \tag{25}$$

The Jacobian $H_{t,\theta}$, i.e., the one w.r.t. landmark, is simply

$$H_\theta^T = \frac{1}{(\delta_3^{\text{cam}})^2} \begin{bmatrix} \mathsf{R}_{1,1}\,\delta_3^{\text{cam}} - \delta_1^{\text{cam}}\,\mathsf{R}_{3,1} & \mathsf{R}_{2,1}\,\delta_3^{\text{cam}} - \delta_2^{\text{cam}}\,\mathsf{R}_{3,1} \\ \mathsf{R}_{1,2}\,\delta_3^{\text{cam}} - \delta_1^{\text{cam}}\,\mathsf{R}_{3,2} & \mathsf{R}_{2,2}\,\delta_3^{\text{cam}} - \delta_2^{\text{cam}}\,\mathsf{R}_{3,2} \\ \mathsf{R}_{1,3}\,\delta_3^{\text{cam}} - \delta_1^{\text{cam}}\,\mathsf{R}_{3,3} & \mathsf{R}_{2,3}\,\delta_3^{\text{cam}} - \delta_2^{\text{cam}}\,\mathsf{R}_{3,3} \end{bmatrix} \tag{26}$$

The Jacobian of H w.r.t. θ, $H_{x\theta}$, is more complicated and here we give a compact expression for its elements. The dimension of the Jacobian is $2 \times 10 \times 3$ and we shall denote the elements of it as $H_{x\theta}(i,j,k)$ where $i = 1:2$, $j = 1:10$ and $k = 1:3$. Elements where $j = 4:6$ are all zeros for all i and k, while for $j = 1:3$ we have

$$H_{x\theta}(i,j,k) = \frac{1}{(\delta_3^{\text{cam}})^2}\left(\mathsf{R}_{i,j}\,\mathsf{R}_{3,k} + \mathsf{R}_{3,j}\left(\mathsf{R}_{i,k} - 2\frac{\delta_i^{\text{cam}}}{\delta_3^{\text{cam}}}\,\mathsf{R}_{3,k}\right)\right) \tag{27}$$

and for $j = 7:10$ we have

$$\begin{aligned} H_{x\theta}(i,j,k) = &\frac{1}{(\delta_3^{\text{cam}})^2}\big(\delta_3^{\text{cam}}\,\mathsf{R}'_{i,k,j-6} - \Delta_{q,i,j-6}^{\text{cam}}\,\mathsf{R}_{3,k} - \\ &- \mathsf{R}_{i,k}\,\Delta_{q,3,j-6}^{\text{cam}} - \delta_i^{\text{cam}}\,\mathsf{R}'_{3,k,j-6} - \\ &- 2\frac{\delta_i^{\text{cam}}}{\delta_3^{\text{cam}}}\Delta_{q,3,j-6}^{\text{cam}}\,\mathsf{R}_{3,k}\big). \end{aligned} \tag{28}$$

The last Jacobian needed is Q_x, and in order to calculate it we will define the state noise covariance as

$$Q = B\bar{Q}B^T \tag{29}$$

where $\bar{Q}$ is a constant 6×6 matrix defined as

$$\bar{Q} = \begin{bmatrix} \sigma_a I_3 & 0 \\ 0 & \sigma_\omega I_3 \end{bmatrix} \tag{30}$$

and the matrix B is

$$B = \begin{bmatrix} \frac{T^2}{2}\,\mathsf{R}(q)^T & 0 \\ T\,\mathsf{R}(q)^T & 0 \\ 0 & \frac{T}{2}\widetilde{S}(q) \end{bmatrix} \tag{31}$$

with the matrix $\widetilde{S}(q)$ defined as

$$\widetilde{S}(q) = \begin{bmatrix} -q^1 & -q^2 & -q^3 \\ q^0 & -q^3 & q^2 \\ q^3 & q^0 & -q^1 \\ -q^2 & q^1 & q^0 \end{bmatrix}. \tag{32}$$

This matrix is used in the B matrix due to the approximation of the quaternion dynamics in (17c) with a first order Taylor expansion and bi-linearity of the resulting expression. Now the Jacobian of the state noise covariance can be expressed as

$$Q_x = B_x\bar{Q}B^T + B\bar{Q}B_x^T. \tag{33}$$

B_x is $10 \times 6 \times 10$ tensor where only elements 7:10 in third dimension are non-zero (since B is function of the quaternions only). This gives that B_x, for these elements, is

$$B_x(:,:,7:10) = \begin{bmatrix} \frac{T^2}{2}\,\mathsf{R}' & 0 \\ T\,\mathsf{R}' & 0 \\ 0 & \frac{T}{2}\widetilde{S}' \end{bmatrix} \tag{34}$$

and $\widetilde{S}' = \partial\widetilde{S}/\partial q$ is simply

$$\frac{\partial\widetilde{S}}{\partial q^0} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{35}$$

$$\frac{\partial\widetilde{S}}{\partial q^1} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \tag{36}$$

$$\frac{\partial\widetilde{S}}{\partial q^2} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \tag{37}$$

$$\frac{\partial\widetilde{S}}{\partial q^3} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{38}$$

REFERENCES

[1] L. Ljung, *System Identification, Theory for the User*, 2nd ed. Prentice Hall PTR, 1999.

[2] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 12, pp. 99–110, June 2006.

[3] T. Bailey and H. Durrant-Whyte, "Simultaneous Localization and Mapping (SLAM): Part II," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, Sep. 2006.

[4] M. Kaess and A. Ranganathan and F. Dellaert, "iSAM: Incremental Smoothing and Mapping," *IEEE Transansactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, December 2008.

[5] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular slam," in *Robotics:*

Science and Systems, Y. Matsuoka, H. F. Durrant-Whyte, and J. Neira, Eds. The MIT Press, 2010.

[6] A. Kim and R. Eustice, "Pose-graph visual slam with geometric model selection for autonomous underwater ship hull inspection," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009, pp. 1559–1565.

[7] S. Williams, V. Indelman, M. Kaess, R. Roberts, J. Leonard, and F. Dellaert, "Concurrent filtering and smoothing: A parallel architecture for real-time navigation and full smoothing," *International Journal of Robotics Research*, vol. 33, pp. 1544–1568, 2014. [Online]. Available: http://ijr.sagepub.com/content/33/12/1544

[8] Z. Sjanic, M. A. Skoglund, T. B. Schön, and F. Gustafsson, "A Nonlinear Least-Squares Approach to the SLAM Problem." in *Proceedings of 18th IFAC World Congress*, Milano, Italy, August/Septemeber 2011.

[9] K. Levenberg, "A method for the solution of certain nonlinear problems in least squares," *Quarterly Journal of Applied Mathmatics*, vol. II, no. 2, pp. 164–168, 1944.

[10] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[11] R. Larsson, Z. Sjanic, M. Enqvist, and L. Ljung, "Direct Prediction-error Identification of Unstable Nonlinear Systems Applied to Flight Test Data," in *Proceedings of the 15th IFAC Symposium on System Identification*, Saint-Malo, France, July 2009.

[12] M. Kok and T. B. Schön, "Maximum likelihood calibration of a magnetometer using inertial sensors," in *Proceedings of the 19th IFAC World Congress*, Cape Town, South Africa, August 2014.

[13] W. J. Rugh, *Linear System Theory*. Prentice Hall, Engelwood Cliffs, NJ, 2nd ed., 1996.

[14] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.