A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation

Ao Li

Anastasios I. Mourikis and Stergios I. Roumeliotis

2007 IEEE International Conference on Robotics and Automation
Roma, Italy, 10-14 April 2007

Content

I. INTRODUCTION

- Advances in the manufacturing of MEMS-based inertial sensors have made it possible to build small, inexpensive, and very accurate Inertial Measurement Units (IMUs).

- An important advantage of visual sensing is that images are high-dimensional measurements, with rich information content.

I. INTRODUCTION

- Our approach is motivated by the observation that, when a static feature is viewed from several camera poses, it is possible to define geometric constraints involving all these poses.

- The primary contribution of our work is a measurement model that expresses these constraints without including the 3D feature position in the filter state vector, resulting in computational complexity only linear in the number of features.

feature = landmark

II. RELATED WORK

- ① One family of algorithms for fusing inertial measurements with visual feature observations follows the Simultaneous Localization and Mapping (SLAM) paradigm.

- In these methods, the current IMU pose, as well as the 3D positions of all visual landmarks are jointly estimated with the difference that IMU measurements, instead of a statistical motion model, are used for state propagation.

MonoSLAM + IMU

II. RELATED WORK

- The fundamental advantage of SLAM based algorithms is that they account for the correlations that exist between the pose of the camera and the 3D positions of the observed features.

- On the other hand, the main limitation of SLAM is its high computational complexity.

II. RELATED WORK

- ② Several algorithms exist that, contrary to SLAM, estimate the pose of the camera only(i.e., do not jointly estimate the feature positions), with the aim of achieving real-time operation.

- Utilize the feature measurements to derive constraints between pairs of images.

- Our algorithm can express constraints between multiple camera poses, and can thus attain higher estimation accuracy.

* Introduction to Kalman Filter

- Discrete Kalman filter

State vector

Control vector

System Model $\boxed{x_k} = Ax_{k-1} + B\boxed{u_{k-1}} + w_{k-1}$

Measure Model $\boxed{z_k} = Hx_k + v_k$

Measurement

* Introduction to Kalman Filter

- Discrete Kalman filter — predict equations.

Predict state $$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

Predict covariance $$P_k^- = AP_{k-1}A^T + Q$$

* Introduction to Kalman Filter

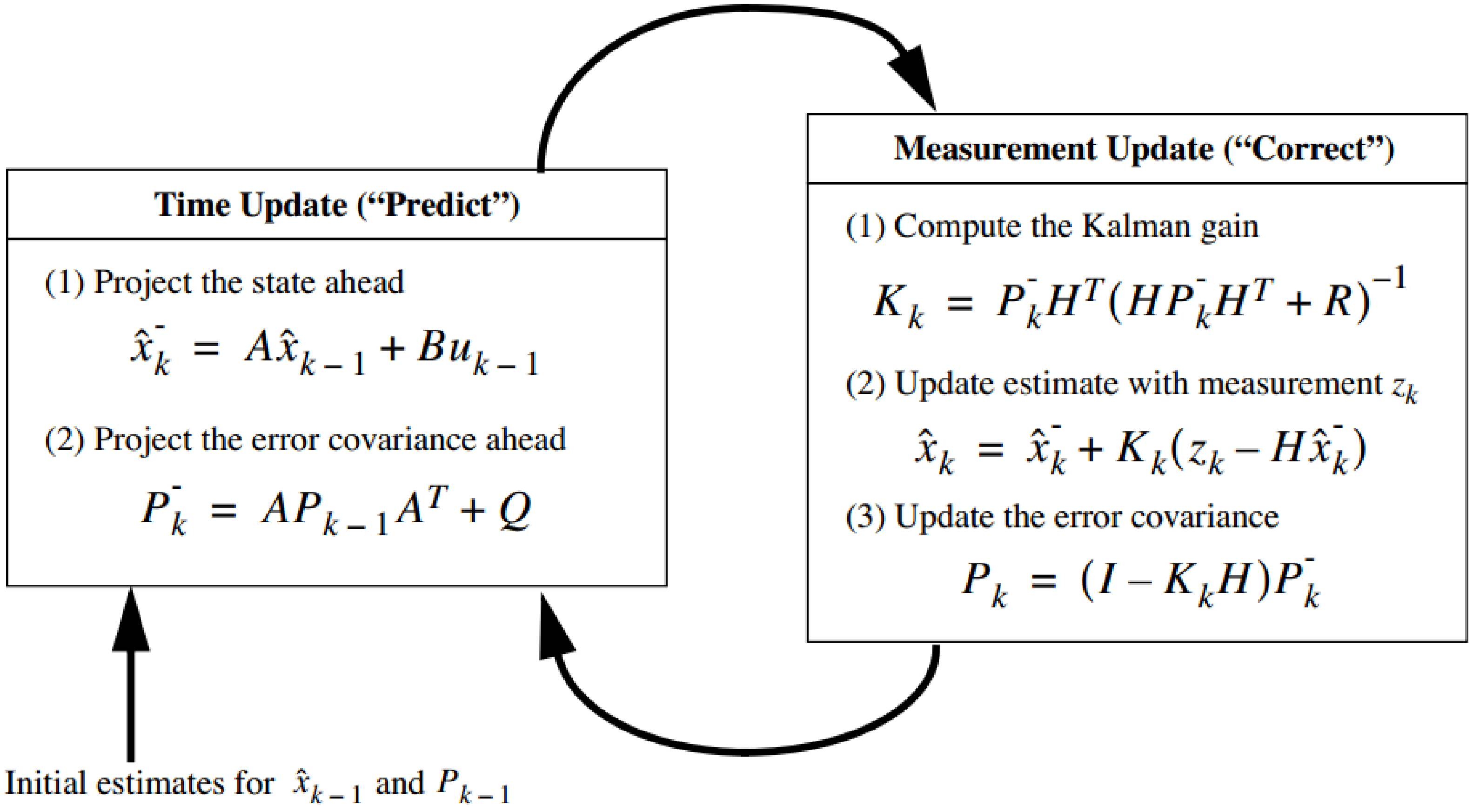
- Discrete Kalman filter — update equations.

Kalman Gain $$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

Update state $$\hat{x}_k = \hat{x}_k^- + K_k \underbrace{\boxed{(z_k - H\hat{x}_k^-)}}_{\text{Residual}}$$

Update Covariance $$P_k = (I - K_k H) P_k^-$$

* Introduction to Kalman Filter



* Introduction to Kalman Filter

- Extended Kalman Filter (Discrete)

System Model $$x_k = f(x_{k-1}, u_{k-1}, w_{k-1})$$

Measure Model $$z_k = h(x_k, v_k)$$

* Introduction to Kalman Filter

- Predict state

Predict state $$\tilde{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0)$$

* Introduction to Kalman Filter

- EKF:

How to predict covariance matrix ?

linearization

$$x_k \approx \tilde{x}_k + \boxed{A}(x_{k-1} - \hat{x}_{k-1}) + \boxed{W} w_{k-1}$$

$$z_k \approx \tilde{z}_k + \boxed{H}(x_k - \tilde{x}_k) + \boxed{V} v_k .$$

* Introduction to Kalman Filter

- Two important equations

Error State

$$\tilde{e}_{x_k} \approx A\tilde{e}_{x_{k-1}} + \varepsilon_k$$

How the error propagates

$$\tilde{e}_{z_k} \approx H\tilde{e}_{x_k} + \eta_k$$

How the error is measured

Residual

* Introduction to Kalman Filter

- EKF predict equations.

Predict state $$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0)$$

Predict covariance $$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T$$

* Introduction to Kalman Filter

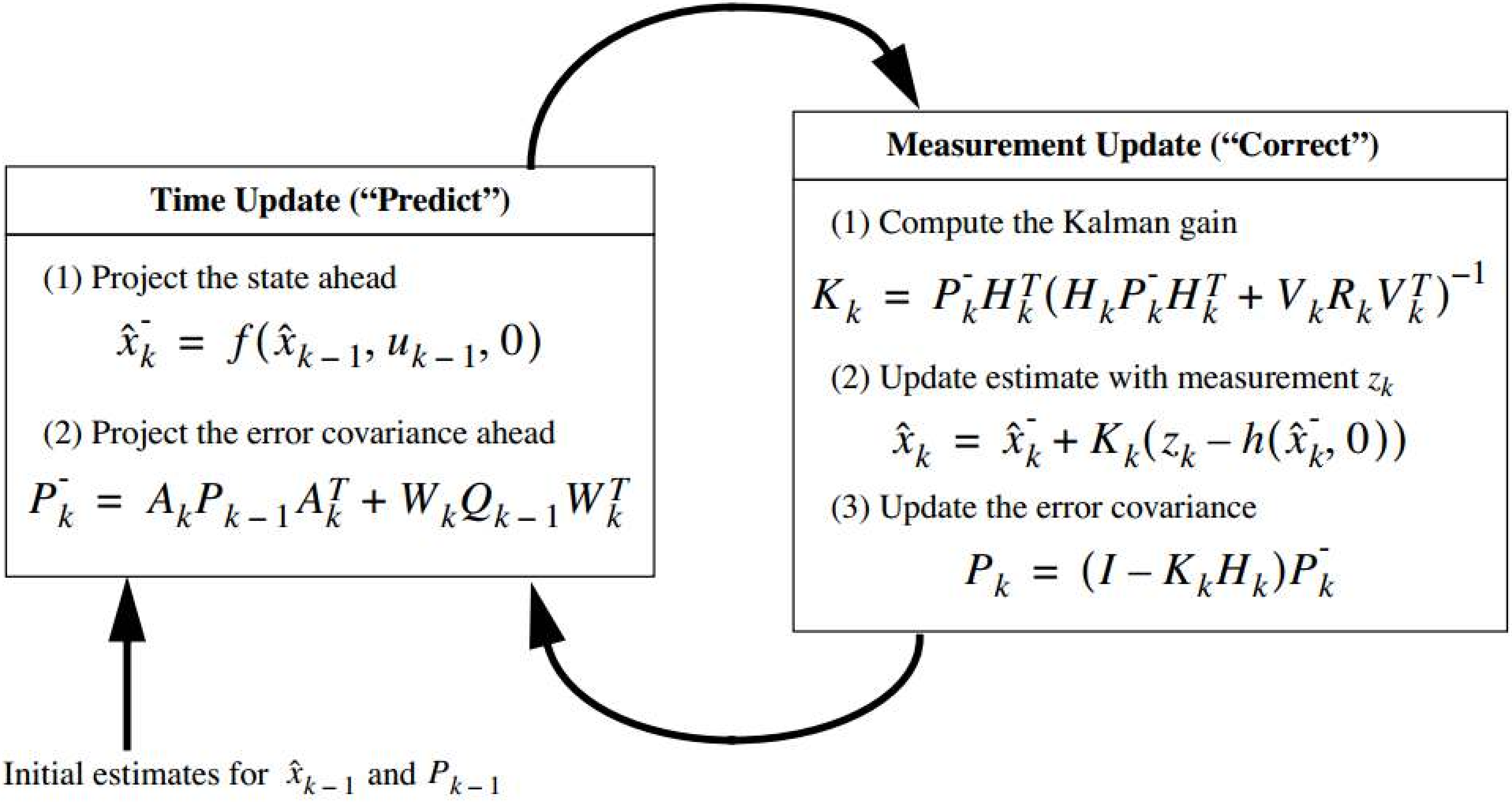
- EKF update equations.

Kalman Gain $$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}$$

Update state $$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0))$$

Update Covariance $$P_k = (I - K_k H_k) P_k^-$$

* Introduction to Kalman Filter



III. ESTIMATOR DESCRIPTION

- A. Structure of the EKF state vector

EKF state vector comprises

(i) the evolving IMU state

(ii) a history of up to $N_{\max}$ past poses of the camera

III. ESTIMATOR DESCRIPTION

- EKF state vector — IMU State

$$\mathbf{X}_{\mathrm{IMU}} = \begin{bmatrix} {}^I_G\bar{q}^T & \mathbf{b}_g{}^T & {}^G\mathbf{v}_I{}^T & \mathbf{b}_a{}^T & {}^G\mathbf{p}_I^T \end{bmatrix}^T$$

${}^I_G\bar{q}$: the unit quaternion describing the rotation from {G} to {I}

$\mathbf{b}_g$: the bias affecting gyroscope measurements

${}^G\mathbf{v}_I$: IMU velocity with respect to {G}

$\mathbf{b}_a$: the bias affecting accelerometer measurements

${}^G\mathbf{p}_I$: IMU position with respect to {G}

III. ESTIMATOR DESCRIPTION

- IMU error-state

$$\widetilde{\mathbf{X}}_{\mathrm{IMU}} = \begin{bmatrix} \boldsymbol{\delta\theta}_I^T & \widetilde{\mathbf{b}}_g^T & {}^G\widetilde{\mathbf{v}}_I^T & \widetilde{\mathbf{b}}_a^T & {}^G\widetilde{\mathbf{p}}_I^T \end{bmatrix}^T$$

Error definition:

$$\widetilde{x} = x - \hat{x}$$

For quaternion $\bar{q} = \delta\bar{q} \otimes \hat{\bar{q}}$

III. ESTIMATOR DESCRIPTION

- EKF state vector

Assuming that N camera poses are included in the EKF state vector at time-step k, this vector has the following form

$$\hat{\mathbf{X}}_k = \begin{bmatrix} \hat{\mathbf{X}}_{\mathrm{IMU}_k}^T & \boxed{{}^{C_1}_{G}\hat{\bar{q}}^T \quad {}^{G}\hat{\mathbf{p}}_{C_1}^T} & \dots & {}^{C_N}_{G}\hat{\bar{q}}^T & {}^{G}\hat{\mathbf{p}}_{C_N}^T \end{bmatrix}^T$$

R & T of camera pose

${}^{C_i}_{G}\hat{\bar{q}}$ and ${}^{G}\hat{\mathbf{p}}_{C_i}$, $i = 1 \dots N$ are the estimates of the camera attitude and position, respectively.

III. ESTIMATOR DESCRIPTION

- EKF error state vector is defined accordingly:

$$\widetilde{\mathbf{X}}_k = \begin{bmatrix} \widetilde{\mathbf{X}}_{\mathrm{IMU}_k}^T & \boldsymbol{\delta\theta}_{C_1}^T & {}^G\widetilde{\mathbf{p}}_{C_1}^T & \ldots & \boldsymbol{\delta\theta}_{C_N}^T & {}^G\widetilde{\mathbf{p}}_{C_N}^T \end{bmatrix}^T$$

III. ESTIMATOR DESCRIPTION

- B. Propagation (Predict)

The filter propagation equations are derived by discretization of the continuous-time IMU system model

1) Continuous-time system modeling:

The time evolution of the IMU state is described by

$$ {}^{I}_{G}\dot{\bar{q}}(t) = \frac{1}{2}\boldsymbol{\Omega}\left(\boxed{\boldsymbol{\omega}(t)}\right){}^{I}_{G}\bar{q}(t), \quad \dot{\mathbf{b}}_g(t) = \boxed{\mathbf{n}_{wg}(t)} $$

$$ {}^{G}\dot{\mathbf{v}}_I(t) = \boxed{{}^{G}\mathbf{a}(t)}, \quad \dot{\mathbf{b}}_a(t) = \boxed{\mathbf{n}_{wa}(t)}, \quad {}^{G}\dot{\mathbf{p}}_I(t) = {}^{G}\mathbf{v}_I(t) $$

III. ESTIMATOR DESCRIPTION

- Applying the expectation operator in the state propagation equations we obtain the equations for propagating the estimates of the evolving IMU state.

Derived by measurement of gyro

$$ {}^{I}_{G}\dot{\hat{\bar{q}}} = \frac{1}{2}\boldsymbol{\Omega}(\boxed{\hat{\omega}})\,{}^{I}_{G}\hat{\bar{q}}, \quad \dot{\hat{\mathbf{b}}}_g = \mathbf{0}_{3\times 1}, $$

$$ {}^{G}\dot{\hat{\mathbf{v}}}_I = \mathbf{C}_{\hat{q}}^{T}\boxed{\hat{\mathbf{a}}} - 2\lfloor \boldsymbol{\omega}_G \times \rfloor\, {}^{G}\hat{\mathbf{v}}_I - \lfloor \boldsymbol{\omega}_G \times \rfloor^2\, {}^{G}\hat{\mathbf{p}}_I + {}^{G}\mathbf{g} $$

Derived by measurement of accelerometer

$$ \dot{\hat{\mathbf{b}}}_a = \mathbf{0}_{3\times 1}, \quad {}^{G}\dot{\hat{\mathbf{p}}}_I = {}^{G}\hat{\mathbf{v}}_I $$

III. ESTIMATOR DESCRIPTION

- 2) Discrete-time implementation
- IMU state estimate

Every time a new IMU measurement is received, the IMU state estimate is propagated using 5th order Runge-Kutta numerical integration of Eqs. (9)

$$\boxed{\begin{gathered} {}^I_G\dot{\hat{\bar{q}}} = \tfrac{1}{2}\boldsymbol{\Omega}(\hat{\omega})\,{}^I_G\hat{\bar{q}}, \quad \dot{\hat{\mathbf{b}}}_g = \mathbf{0}_{3\times 1}, \\ {}^G\dot{\hat{\mathbf{v}}}_I = \mathbf{C}_{\hat{q}}^T\hat{\mathbf{a}} - 2\lfloor\boldsymbol{\omega}_G\times\rfloor\,{}^G\hat{\mathbf{v}}_I - \lfloor\boldsymbol{\omega}_G\times\rfloor^2\,{}^G\hat{\mathbf{p}}_I + {}^G\mathbf{g} \\ \dot{\hat{\mathbf{b}}}_a = \mathbf{0}_{3\times 1}, \quad {}^G\dot{\hat{\mathbf{p}}}_I = {}^G\hat{\mathbf{v}}_I \end{gathered}}$$

III. ESTIMATOR DESCRIPTION

- What about the covariance? we need this

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{\Phi}(t_k + T, t_k)\tilde{\mathbf{x}}_k$$

III. ESTIMATOR DESCRIPTION

- The linearized continuous time model for the IMU error-state is:

$$\dot{\widetilde{\mathbf{X}}}_{\text{IMU}} = \mathbf{F}\widetilde{\mathbf{X}}_{\text{IMU}} + \mathbf{G}\mathbf{n}_{\text{IMU}}$$

$$\mathbf{F} = \begin{bmatrix} -\lfloor\hat{\boldsymbol{\omega}}\times\rfloor & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -\mathbf{C}_{\hat{q}}^T\lfloor\hat{\mathbf{a}}\times\rfloor & \mathbf{0}_{3\times3} & -2\lfloor\boldsymbol{\omega}_G\times\rfloor & -\mathbf{C}_{\hat{q}}^T & -\lfloor\boldsymbol{\omega}_G\times\rfloor^2 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \qquad \mathbf{G} = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{C}_{\hat{q}}^T & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix}$$

$$\mathbf{n}_{\text{IMU}} = \begin{bmatrix} \mathbf{n}_g^T & \mathbf{n}_{wg}^T & \mathbf{n}_a^T & \mathbf{n}_{wa}^T \end{bmatrix}^T$$

III. ESTIMATOR DESCRIPTION

- The error state transition matrix $\mathbf{\Phi}(t_k + T, t_k)$ is similarly computed by numerical integration of the differential equation

$$\dot{\mathbf{\Phi}}(t_k + \tau, t_k) = \mathbf{F}\mathbf{\Phi}(t_k + \tau, t_k), \qquad \tau \in [0, T]$$

with initial condition $\mathbf{\Phi}(t_k, t_k) = \mathbf{I}_{15}$.

- A simple approximation $\mathbf{\Phi} \simeq \mathbf{I} + \mathbf{F}\Delta t$

$$\boxed{\tilde{\mathbf{X}}_{k+1} = \mathbf{\Phi}(t_k + T, t_k)\tilde{\mathbf{X}}_k}$$

III. ESTIMATOR DESCRIPTION

- Two important equations of EKF

Error State

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{\Phi}_k \tilde{\mathbf{x}}_k + \mathbf{w}_{d_k}$$

$$\tilde{\mathbf{z}}_k = \mathbf{H}_k \tilde{\mathbf{x}}_k + \mathbf{n}_k$$

Residual

III. ESTIMATOR DESCRIPTION

- EKF covariance estimate

Partition:

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{II_{k|k}} & \mathbf{P}_{IC_{k|k}} \\ \mathbf{P}_{IC_{k|k}}^T & \mathbf{P}_{CC_{k|k}} \end{bmatrix} \quad (11)$$

$\mathbf{P}_{II_{k|k}}$ The 15 × 15 covariance matrix of the evolving IMU state

$\mathbf{P}_{CC_{k|k}}$ The 6N × 6N covariance matrix of the camera pose estimates

III. ESTIMATOR DESCRIPTION

- EKF covariance

covariance matrix of the state estimate is propagated as:

$$\mathbf{P}_{k+1|k} = \begin{bmatrix} \mathbf{P}_{II_{k+1|k}} & \mathbf{\Phi}(t_k + T, t_k)\mathbf{P}_{IC_{k|k}} \\ \mathbf{P}_{IC_{k|k}}^T \mathbf{\Phi}(t_k + T, t_k)^T & \mathbf{P}_{CC_{k|k}} \end{bmatrix}$$

$\mathbf{P}_{II_{k+1|k}}$ the covariance matrix of the propagated IMU state

$\mathbf{\Phi}(t_k + T, t_k)$ the error-state transition matrix for the IMU

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{\Phi}(t_k + T, t_k)\tilde{\mathbf{X}}_k$$

III. ESTIMATOR DESCRIPTION

- $\mathbf{P}_{II_{k+1|k}}$

A simple Version:

$$\mathbf{P}_{k+1|k} = \begin{bmatrix} \boxed{\mathbf{\Phi}_{I_k}\mathbf{P}_{II_{k|k}}\mathbf{\Phi}_{I_k}^T + \mathbf{Q}_d} & \mathbf{\Phi}_{I_k}\mathbf{P}_{IC_{k|k}} \\ \mathbf{P}_{IC_{k|k}}^T\mathbf{\Phi}_{I_k}^T & \mathbf{P}_{CC_{k|k}} \end{bmatrix}$$

III. ESTIMATOR DESCRIPTION

- C. State Augmentation

Upon recording a new image, the camera pose estimate is computed from the IMU pose estimate as:

$$ {}^{C}_{G}\hat{\bar{q}} = \boxed{{}^{C}_{I}\bar{q}} \otimes {}^{I}_{G}\hat{\bar{q}}, \quad \text{and} \quad {}^{G}\hat{\mathbf{p}}_{C} = {}^{G}\hat{\mathbf{p}}_{I} + \mathbf{C}^{T}_{\hat{q}} \boxed{{}^{I}\mathbf{p}_{C}} $$

III. ESTIMATOR DESCRIPTION

- When a camera pose estimate is appended to the state vector, the covariance matrix of the EKF is augmented accordingly

$$\mathbf{P}_{k|k} \leftarrow \begin{bmatrix} \mathbf{I}_{6N+15} \\ \mathbf{J} \end{bmatrix} \mathbf{P}_{k|k} \begin{bmatrix} \mathbf{I}_{6N+15} \\ \mathbf{J} \end{bmatrix}^T$$

$$\mathbf{J} = \begin{bmatrix} \mathbf{C}\left({}^{C}_{I}\bar{q}\right) & \mathbf{0}_{3\times 9} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 6N} \\ \lfloor \mathbf{C}_{\hat{q}}^{T}\,{}^{I}\mathbf{p}_{C} \times \rfloor & \mathbf{0}_{3\times 9} & \mathbf{I}_{3} & \mathbf{0}_{3\times 6N} \end{bmatrix}$$

III. ESTIMATOR DESCRIPTION

- D. Measurement Model

We now present the measurement model employed for updating the state estimates, which is the primary contribution of this paper.

General form of measurement model in EKF：

$$\mathbf{r} = \mathbf{H}\widetilde{\mathbf{X}} + \text{noise}$$

$\mathbf{H}$ is the measurement Jacobian matrix

III. ESTIMATOR DESCRIPTION

- Measurement model

jth feature , ith pose

$$\mathbf{z}_i^{(j)} = \frac{1}{{}^{C_i}Z_j} \begin{bmatrix} {}^{C_i}X_j \\ {}^{C_i}Y_j \end{bmatrix} + \mathbf{n}_i^{(j)}, \qquad i \in \mathcal{S}_j$$

$${}^{C_i}\mathbf{p}_{f_j} = \begin{bmatrix} {}^{C_i}X_j \\ {}^{C_i}Y_j \\ {}^{C_i}Z_j \end{bmatrix} = \mathbf{C}({}^{C_i}_{G}\bar{q})(\boxed{{}^{G}\mathbf{p}_{f_j}} - {}^{G}\mathbf{p}_{C_i})$$

Where ${}^{G}\mathbf{p}_{f_j}$ is the 3D feature position in the global frame. Since this is unknown, in the first step of our algorithm we employ least-squares minimization to obtain an estimation.

III. ESTIMATOR DESCRIPTION

- the measurement residual

$$\mathbf{r}_i^{(j)} = \mathbf{z}_i^{(j)} - \hat{\mathbf{z}}_i^{(j)}$$

Estimated position of feature

$$\hat{\mathbf{z}}_i^{(j)} = \frac{1}{{}^{C_i}\hat{Z}_j} \begin{bmatrix} {}^{C_i}\hat{X}_j \\ {}^{C_i}\hat{Y}_j \end{bmatrix} , \quad \begin{bmatrix} {}^{C_i}\hat{X}_j \\ {}^{C_i}\hat{Y}_j \\ {}^{C_i}\hat{Z}_j \end{bmatrix} = \boxed{\mathbf{C}({}^{C_i}_{G}\hat{\bar{q}})}(\boxed{{}^{G}\hat{\mathbf{p}}_{f_j}} - \boxed{{}^{G}\hat{\mathbf{p}}_{C_i}})$$

- Linearization $\quad \mathbf{r}_i^{(j)} \simeq \mathbf{H}_{\mathbf{X}_i}^{(j)} \widetilde{\mathbf{X}} + \mathbf{H}_{f_i}^{(j)} {}^{G}\widetilde{\mathbf{p}}_{f_j} + \mathbf{n}_i^{(j)}$

III. ESTIMATOR DESCRIPTION

- By stacking the residuals of all M_j measurements of this feature, we obtain:

$$\mathbf{r}^{(j)} \simeq \mathbf{H}_{\mathbf{X}}^{(j)}\widetilde{\mathbf{X}} + \boxed{\mathbf{H}_{f}^{(j)}\,{}^{G}\widetilde{\mathbf{p}}_{f_j}} + \mathbf{n}^{(j)}$$

$$\cancel{\mathbf{r} = \mathbf{H}\widetilde{\mathbf{X}} + \text{noise}}$$

- Thus, the residual cannot be directly applied for measurement updates in the EKF.

III. ESTIMATOR DESCRIPTION

- To overcome this problem, we let A denote the unitary matrix whose columns form the basis of the left nullspace of H_f

$$\begin{aligned}\mathbf{r}_o^{(j)} &= \mathbf{A}^T(\mathbf{z}^{(j)} - \hat{\mathbf{z}}^{(j)}) \simeq \mathbf{A}^T\mathbf{H}_{\mathbf{X}}^{(j)}\widetilde{\mathbf{X}} + \mathbf{A}^T\mathbf{n}^{(j)} \\ &= \mathbf{H}_o^{(j)}\widetilde{\mathbf{X}}^{(j)} + \mathbf{n}_o^{(j)}\end{aligned}$$

Row number: $2M_j - 3$

- This equation defines a linearized constraint between all the camera poses from which the feature f_j was observed.

III. ESTIMATOR DESCRIPTION

- Two important equations of EKF

Error State

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{\Phi}_k \tilde{\mathbf{x}}_k + \mathbf{w}_{d_k}$$

$$\tilde{\mathbf{z}}_k = \mathbf{H}_k \tilde{\mathbf{x}}_k + \mathbf{n}_k$$

Residual

III. ESTIMATOR DESCRIPTION

- E. EKF Updates

EKF updates are triggered by one of the following two events:

1) When a feature that has been tracked in a number of images is no longer detected, then all the measurements of this feature are processed using the method presented in Section III-D.

This case occurs most often.

III. ESTIMATOR DESCRIPTION

2) The maximum allowable number of camera poses, N_{max}, has been reached.

In our algorithm, we choose $N_{max}/3$ poses that are evenly spaced in time, starting from the second-oldest pose. These are discarded after carrying out an EKF update using the constraints of features that are common to these poses.

III. ESTIMATOR DESCRIPTION

- Update process in detail

Consider that at a given time step the constraints of L features, selected by the above two criteria, must be processed.

$$\mathbf{r}_o^{(j)},\ j = 1 \ldots L \qquad \mathbf{H}_o^{(j)},\ j = 1 \ldots L$$

Row number: $2M_j - 3$

By stacking all residuals in a single vector, we obtain:

$$\mathbf{r}_o = \mathbf{H}_{\mathbf{X}} \widetilde{\mathbf{X}} + \mathbf{n}_o$$

Row number: $d = \sum_{j=1}^{L} (2M_j - 3)$

III. ESTIMATOR DESCRIPTION

Row number of $\mathbf{r}_o$ $d=\sum_{j=1}^{L}(2M_j-3)$

- One issue that arises in practice is that d can be a quite large number. For example, if 10 features are seen in 10 camera poses each, the dimension of the residual is 170. （$d=\sum_{j=1}^{L}(2M_j-3)$）

$$\mathbf{H_X}=\begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{T}_H\\ \mathbf{0}\end{bmatrix}$$

$$\mathbf{r}_o=\begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{T}_H\\ \mathbf{0}\end{bmatrix}\widetilde{\mathbf{X}}+\mathbf{n}_o \Rightarrow \begin{bmatrix}\mathbf{Q}_1^T\mathbf{r}_o\\ \mathbf{Q}_2^T\mathbf{r}_o\end{bmatrix}=\begin{bmatrix}\mathbf{T}_H\\ \mathbf{0}\end{bmatrix}\widetilde{\mathbf{X}}+\begin{bmatrix}\mathbf{Q}_1^T\mathbf{n}_o\\ \mathbf{Q}_2^T\mathbf{n}_o\end{bmatrix}$$

III. ESTIMATOR DESCRIPTION

- the new measurement residual

$$\mathbf{r}_n = \mathbf{Q}_1^T \mathbf{r}_o = \mathbf{T}_H \widetilde{\mathbf{X}} + \mathbf{n}_n$$

$$\mathbf{n}_n = \mathbf{Q}_1^T \mathbf{n}_o$$

r : the number of columns in $\mathbf{Q}_1$

III. ESTIMATOR DESCRIPTION

- update

Kalman Gain

$$\mathbf{K} = \mathbf{P}\mathbf{T}_H^T \left(\mathbf{T}_H \mathbf{P}\mathbf{T}_H^T + \mathbf{R}_n\right)^{-1}$$

Update state

$$\Delta\mathbf{X} = \mathbf{K}\mathbf{r}_n$$

Update Covariance

$$\mathbf{P}_{k+1|k+1} = \left(\mathbf{I}_\xi - \mathbf{K}\mathbf{T}_H\right)\mathbf{P}_{k+1|k}\left(\mathbf{I}_\xi - \mathbf{K}\mathbf{T}_H\right)^T + \mathbf{K}\mathbf{R}_n\mathbf{K}^T$$

$$\xi = 6N+15$$

III. ESTIMATOR DESCRIPTION

r = column number of $\mathbf{Q}_1$

$d = \sum_{j=1}^{L}(2M_j - 3)$

$\xi = 6N+15$

- Computational complexity

The residual $\mathbf{r}_n$, as well as the matrix $\mathbf{T}_H$, can be computed using Givens rotations in $O(r^2 d)$ operations, without the need to explicitly form $\mathbf{Q}_1$.

On the other hand, covariance update involves multiplication of square matrices of dimension ξ , an $O(\xi^3)$ operation.

Therefore, the cost of the EKF update is $\max(O(r^2 d), O(\xi^3))$

III. ESTIMATOR DESCRIPTION

$$\max(O(r^2 d), O(\xi^3))$$

r = column number of $\mathbf{Q}_1 \ll \xi$

$d = \sum_{j=1}^{L}(2M_j - 3)$

$\xi = 6N+15$

- F. Discussion

- As shown in the previous section, the filter's computational complexity is linear in the number of observed features, and at most cubic in the number of states that are included in the state vector.

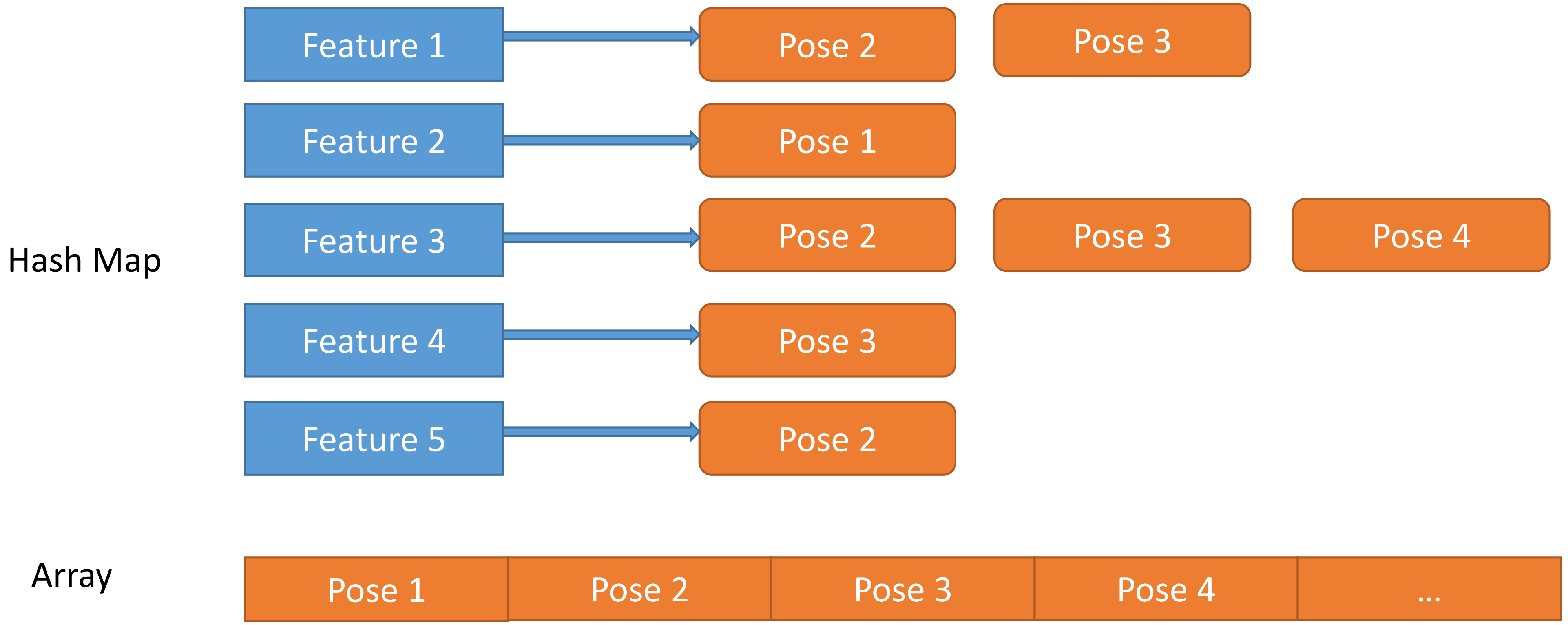
- The $\boxed{\text{number of poses}}$ that are included in the state is the most significant factor in determining the computational cost of the algorithm (tradeoff).

III. ESTIMATOR DESCRIPTION

- If, on the other hand, the residual vector row as employed, without projecting it on the range of $\mathbf{H_X}$, the computational cost of computing the Kalman gain would have been $O(d^3)$. Since typically $d \gg \xi, r$,we see that the use of the residual $\mathbf{r}_n$ results in substantial savings in computation.

III. ESTIMATOR DESCRIPTION

- My guess



III. ESTIMATOR DESCRIPTION

Algorithm 1 Multi-State Constraint Filter

Propagation: For each IMU measurement received, propagate the filter state and covariance (cf. Section III-B).

Image registration: Every time a new image is recorded,

- augment the state and covariance matrix with a copy of the current camera pose estimate (cf. Section III-C).
- image processing module begins operation.

Extract the key points and match

Update: When the feature measurements of a given image become available, perform an EKF update (cf. Sections III-D and III-E). Caused by two criteria

IV. EXPERIMENTAL RESULTS

- The experimental setup：

A camera/IMU system, placed on a car that was moving on the streets of a typical residential area in Minneapolis, MN.

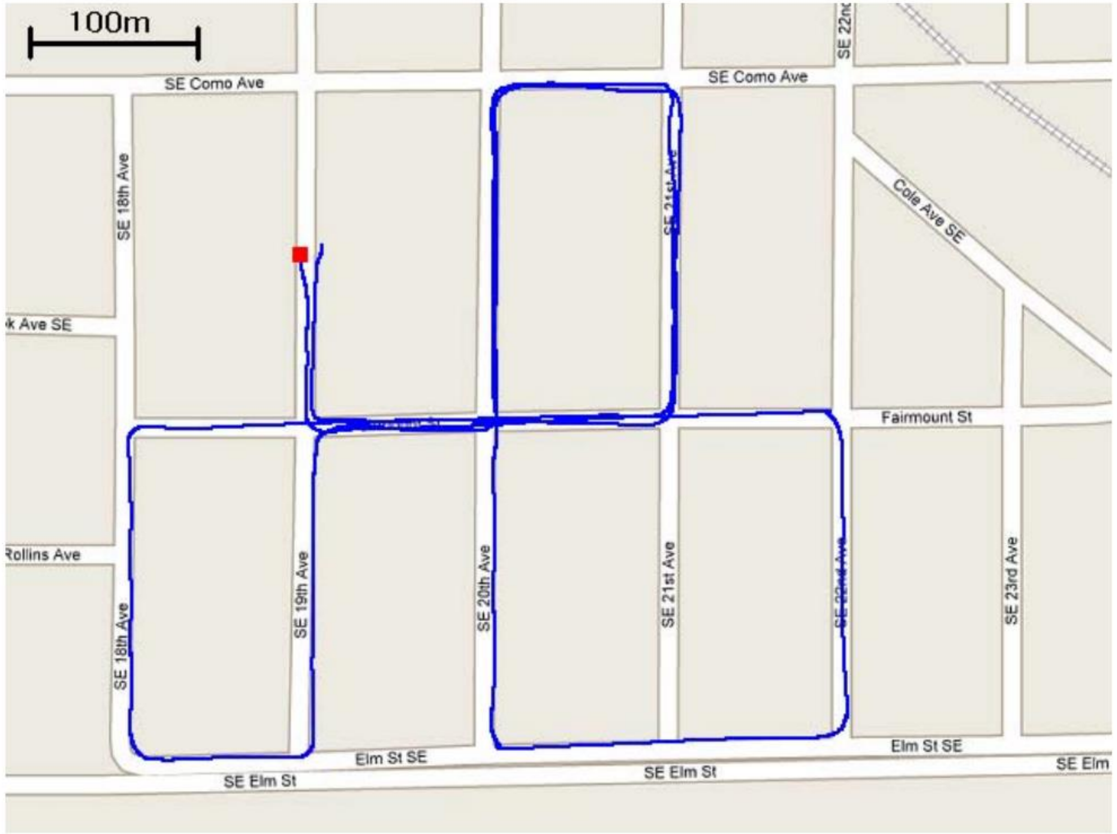
The system comprised a Pointgrey FireFly camera, registering images of resolution 640×480 pixels at $\boxed{\text{3Hz}}$

An Inertial Science ISIS IMU, providing inertial measurements at a rate of $\boxed{\text{100Hz}}$.

IV. EXPERIMENTAL RESULTS

- For the results shown here, feature extraction and matching was performed using the SIFT algorithm
- Even though images were only recorded at 3Hz due to limited hard disk space on the test system, the estimation algorithm is able to process the dataset at 14Hz, on a single core of an Intel T7200 processor (2GHz clock rate).
- During the experiment, a total of 142903 features were successfully tracked and used for EKF updates, along a 3.2km-long trajectory.

IV. EXPERIMENTAL RESULTS



IV. EXPERIMENTAL RESULTS

- Error

$$\hat{\mathbf{X}}_{\mathrm{final}} = [-7.92 \quad 13.14 \quad -0.78]^T$$

$$\mathbf{X}_{\mathrm{final}} = [0 \quad 7 \quad 0]^T$$

Error : 10m

The final position error is approximately 10m in a trajectory of 3.2km, i.e., an error of 0.31% of the travelled distance.

V. CONCLUSIONS

In this paper we have presented an EKF-based estimation algorithm for real-time vision-aided inertial navigation.

The main contribution of this work is the derivation of a measurement model that is able to express the geometric constraints that arise when a static feature is observed from multiple camera poses.

The resulting EKF-based pose estimation algorithm has computational complexity linear in the number of features, and is capable of very accurate pose estimation in large-scale real environments.

Thank you.

III. ESTIMATOR DESCRIPTION

- $\mathbf{P}_{II_{k+1|k}}$ is computed by numerical integration of the Lyapunov equation

$$\dot{\mathbf{P}}_{II} = \mathbf{F}\mathbf{P}_{II} + \mathbf{P}_{II}\mathbf{F}^T + \mathbf{G}\mathbf{Q}_{\mathrm{IMU}}\mathbf{G}^T$$

with initial condition $\mathbf{P}_{II_{k|k}}$ time interval (t_k, t_k+T)

Another Version:

$$\mathbf{P}_{k+1|k} = \begin{bmatrix} \boxed{\mathbf{\Phi}_{I_k}\mathbf{P}_{II_{k|k}}\mathbf{\Phi}_{I_k}^T + \mathbf{Q}_d} & \mathbf{\Phi}_{I_k}\mathbf{P}_{IC_{k|k}} \\ \mathbf{P}_{IC_{k|k}}^T\mathbf{\Phi}_{I_k}^T & \mathbf{P}_{CC_{k|k}} \end{bmatrix}$$