A. Preintegration Using a Schur Complement

The method of preintegration presented in section 3.1 is mathematically equivalent to marginalizing out the unwanted states from the full Bayesian posterior. Marginalization can also be performed using a Schur complement. We will use the Schur complement to efficiently marginalize out unwanted states from our continuous-time formulation. By exploiting sparsity, this can be performed in $O(K)$ time, which is the same time complexity as the classic approach presented in section 3.1.

We consider the factor graph shown in Figure 1, which could potentially be a result of our continuous-time state estimation with binary motion prior factors, unary measurement factors, and a unary prior factor on the initial state $\mathbf{x}_0$. Equivalently, the Gauss-Newton system of equations associated with Figure 1 can be written in the following form:

$$\underbrace{\left(\mathbf{A}^{-T}\mathbf{Q}^{-1}\mathbf{A}+\mathbf{C}^{T}\mathbf{R}^{-1}\mathbf{C}\right)}_{\mathbf{L}}\hat{\mathbf{x}} = \underbrace{\mathbf{A}^{-T}\mathbf{Q}^{-1}\check{\mathbf{x}}+\mathbf{C}^{T}\mathbf{R}^{-1}\mathbf{y}}_{\mathbf{r}}, \tag{A1}$$

where $\mathbf{L}$ is block-tridiagonal,

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{0,0} & \mathbf{L}_{0,1:3} & & & \\ \mathbf{L}_{0,1:3}^{T} & \mathbf{L}_{1:3,1:3} & \mathbf{L}_{1:3,4} & & \\ & \mathbf{L}_{1:3,4}^{T} & \mathbf{L}_{4,4} & \mathbf{L}_{4,5:7} & \\ & & \mathbf{L}_{4,5:7}^{T} & \mathbf{L}_{5:7,5:7} & \mathbf{L}_{5:7,8} \\ & & & \mathbf{L}_{5:7,8}^{T} & \mathbf{L}_{8,8} \end{bmatrix} = \left[\begin{array}{c|ccc|c|ccc|c} * & * & & & & & & & \\ \hline * & * & * & & & & & & \\ & * & * & * & & & & & \\ & & * & * & * & & & & \\ \hline & & & * & * & * & & & \\ \hline & & & & * & * & * & & \\ & & & & & * & * & * & \\ & & & & & & * & * & * \\ \hline & & & & & & & * & * \end{array}\right]. \tag{A2}$$

Here, we consider the case where we would like to marginalize the full posterior such that we only retain states $\{\mathbf{x}_0, \mathbf{x}_4, \mathbf{x}_8\}$. After marginalizing out the unwanted states, our system becomes

$$\mathbf{L}_{\text{small}}\hat{\mathbf{x}}_{\text{small}} = \mathbf{r}_{\text{small}}, \tag{A3}$$

where by the Schur complement,

$$\begin{aligned} \mathbf{L}_{\text{small}} &= \begin{bmatrix} \mathbf{L}_{0,0} & & \\ & \mathbf{L}_{4,4} & \\ & & \mathbf{L}_{8,8} \end{bmatrix} - \begin{bmatrix} \mathbf{L}_{0,1:3} & \\ \mathbf{L}_{1:3,4}^{T} & \mathbf{L}_{4,5:7} \\ & \mathbf{L}_{5:7,8}^{T} \end{bmatrix} \begin{bmatrix} \mathbf{L}_{1:3,1:3} & \\ & \mathbf{L}_{5:7,5:7} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{L}_{0,1:3}^{T} & \mathbf{L}_{1:3,4} & \\ & \mathbf{L}_{4,5:7}^{T} & \mathbf{L}_{5:7,8} \end{bmatrix} \\ &= \left[\begin{array}{c|c|c} * & & \\ \hline & * & \\ \hline & & * \end{array}\right] - \left[\begin{array}{ccc|ccc} * & & & & & \\ \hline & & * & * & & \\ \hline & & & & & * \end{array}\right] \times \left[\begin{array}{ccc|ccc} * & * & & & & \\ * & * & * & & & \\ & * & * & & & \\ \hline & & & * & * & \\ & & & * & * & * \\ & & & & * & * \end{array}\right]^{-1} \left[\begin{array}{c|c|c} * & & \\ & & \\ & * & \\ \hline & * & \\ & & \\ & & * \end{array}\right]. \end{aligned}$$

$\mathbf{L}_{\text{small}}$ can be computed efficiently by exploiting the primary and secondary sparsity. Note that

$$\begin{bmatrix} \mathbf{L}_{1:3,1:3} & \\ & \mathbf{L}_{5:7,5:7} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{L}_{0,1:3}^{T} & \mathbf{L}_{1:3,4} & \\ & \mathbf{L}_{4,5:7}^{T} & \mathbf{L}_{5:7,8} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{1:3,1:3} \backslash \mathbf{L}_{0,1:3}^{T} & \mathbf{L}_{1:3,1:3} \backslash \mathbf{L}_{1:3,4} & \\ & \mathbf{L}_{5:7,5:7} \backslash \mathbf{L}_{4,5:7}^{T} & \mathbf{L}_{5:7,5:7} \backslash \mathbf{L}_{5:7,8} \end{bmatrix}, \tag{A4}$$

where each entry similar to $\mathbf{L}_{1:3,1:3} \backslash \mathbf{L}_{1:3,4}$ is shorthand for solving $\mathbf{L}_{1:3,1:3}\boldsymbol{\ell} = \mathbf{L}_{1:3,4}$ for $\boldsymbol{\ell}$. Each of these terms can be solved in linear time thanks to $\mathbf{L}_{1:3,1:3}$ and $\mathbf{L}_{5:7,5:7}$ being block-tridiagonal. Thus, $\mathbf{L}_{\text{small}}$ can be constructed in linear time, the same time complexity as the classic approach. Furthermore, it can be shown that the resulting matrix $\mathbf{L}_{\text{small}}$ is block-tridiagonal. In summary, Schur complement preintegration is a generalization of classic preintegration that can handle both binary motion prior factors and unary measurement factors while retaining the same linear time complexity as classic preintegration.

B. Analytical Gradients for Training the Singer Prior

Following the method presented by Wong et al. [30] for training the parameters of the Singer prior, we found it necessary to derive the analytical gradients of our objective with respect to the desired parameters. Since these gradients were not provided in [30], we provide them here instead for the convenience of the reader. Starting with the objective from (31), the discrete-time covariance $\mathbf{Q}_k$ of the Singer prior can be written as the product of two factors where

$$\mathbf{Q}_k = \underbrace{\begin{bmatrix} \sigma^2 & & \\ & \sigma^2 & \\ & & \sigma^2 \end{bmatrix}}_{\mathbf{Q}_{\sigma^2}} \mathbf{Q}(\Delta t_k, \alpha). \tag{B1}$$

The components of $\mathbf{Q}(\Delta t_k, \alpha)$ are provided by Wong et al. [30] and are repeated here,

$$\mathbf{Q}(\Delta t_k, \alpha) = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} & \mathbf{Q}_{13} \\ \mathbf{Q}_{12}^T & \mathbf{Q}_{22} & \mathbf{Q}_{23} \\ \mathbf{Q}_{13}^T & \mathbf{Q}_{23}^T & \mathbf{Q}_{33} \end{bmatrix}, \tag{B2}$$

where

$$\mathbf{Q}_{11} = \frac{1}{2}\alpha^{-5}\Big(\mathbf{1} - e^{-2\alpha\Delta t_k} + 2\alpha\Delta t_k + \frac{2}{3}\alpha^3\Delta t_k^3 - 2\alpha^2\Delta t_k^2 - 4\alpha\Delta t_k e^{-\alpha\Delta t_k}\Big), \tag{B3a}$$

$$\mathbf{Q}_{12} = \frac{1}{2}\alpha^{-4}\Big(e^{-2\alpha\Delta t_k} + \mathbf{1} - 2e^{-\alpha\Delta t_k} + 2\alpha\Delta t_k e^{-\alpha\Delta t_k} - 2\alpha\Delta t_k + \alpha^2\Delta t_k^2\Big), \tag{B3b}$$

$$\mathbf{Q}_{13} = \frac{1}{2}\alpha^{-3}\left(\mathbf{1} - e^{-2\alpha\Delta t_k} - 2\alpha\Delta t_k e^{-\alpha\Delta t_k}\right), \tag{B3c}$$

$$\mathbf{Q}_{22} = \frac{1}{2}\alpha^{-3}\left(4e^{-\alpha\Delta t_k} - 3\cdot\mathbf{1} - e^{-2\alpha\Delta t_k} + 2\alpha\Delta t_k\right), \tag{B3d}$$

$$\mathbf{Q}_{23} = \frac{1}{2}\alpha^{-2}\left(e^{-2\alpha\Delta t_k} + \mathbf{1} - 2e^{-\alpha\Delta t_k}\right), \tag{B3e}$$

$$\mathbf{Q}_{33} = \frac{1}{2}\alpha^{-1}\left(\mathbf{1} - e^{-2\alpha\Delta t_k}\right). \tag{B3f}$$

The motion error is given by

$$\mathbf{e}_k = \mathbf{x}_k - \boldsymbol{\Phi}(t_k, t_{k-1})\mathbf{x}_{k-1}, \tag{B4}$$

where

$$\boldsymbol{\Phi}(t_k, t_{k-1}) = \begin{bmatrix} \mathbf{1} & \Delta t_k \mathbf{1} & (\alpha\Delta t_k - \mathbf{1} + \exp(-\boldsymbol{\alpha}\Delta t_k))\boldsymbol{\alpha}^{-2} \\ \mathbf{0} & \mathbf{1} & (\mathbf{1} - \exp(-\boldsymbol{\alpha}\Delta t_k))\boldsymbol{\alpha}^{-1} \\ \mathbf{0} & \mathbf{0} & \exp(-\boldsymbol{\alpha}\Delta t_k) \end{bmatrix} \tag{B5}$$

is the state transition function. $\boldsymbol{\sigma}^2$ and $\boldsymbol{\alpha}$ are both diagonal matrices, whose size depends on the dimension of the state. For example, for a 6D state, $\boldsymbol{\sigma}^2 = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2, \sigma_5^2, \sigma_6^2)$. The gradients of the objective with respect to the components of $\boldsymbol{\sigma}^2$ and $\boldsymbol{\alpha}$ are then

$$\frac{\partial J_t}{\partial \alpha_i} = \frac{1}{2}\sum_k \left\{ 2\mathbf{e}_k^T \mathbf{Q}_k^{-1} \frac{\partial \mathbf{e}_k}{\partial \alpha_i} - \mathbf{e}_k^T \mathbf{Q}_k^{-1} \frac{\partial \mathbf{Q}_k}{\partial \alpha_i} \mathbf{Q}_k^{-1} \mathbf{e}_k + \mathrm{tr}\left(\mathbf{Q}_k^{-1} \frac{\partial \mathbf{Q}_k}{\partial \alpha_i} \right) \right\}, \tag{B6a}$$

$$\frac{\partial J_t}{\partial \sigma_i^2} = \frac{3K}{2\sigma_i^2} - \frac{1}{2}\sum_k \frac{1}{\sigma_i^4} \mathbf{e}_k^T \mathbf{Q}(\Delta t_k, \boldsymbol{\alpha})^{-1} \frac{\partial \mathbf{Q}_{\sigma^2}}{\partial \sigma_i^2} \mathbf{e}_k, \tag{B6b}$$

for each α_i and σ_i^2, respectively, where $J = \sum_{t=1}^{T} J_t$. The partial derivatives of $\mathbf{Q}(\Delta t_k, \boldsymbol{\alpha})$ and $\mathbf{e}_k$ with respect to α_i are then given by

$$\frac{\partial \mathbf{Q}_{11}}{\partial \alpha_i} = \left[-\frac{2\Delta t_k^3}{3\alpha_i^3} + \frac{\Delta t_k^2 (2e^{-\alpha_i \Delta t_k} + 3)}{\alpha_i^4} + \frac{5(e^{-2\alpha_i \Delta t_k} - 1)}{2\alpha_i^6} + \frac{\Delta t_k (e^{-2\alpha_i \Delta t_k} + 8e^{-\alpha_i \Delta t_k} - 4)}{\alpha_i^5} \right] \boldsymbol{\delta}_{ii}, \tag{B7a}$$

$$\frac{\partial \mathbf{Q}_{12}}{\partial \alpha_i} = \left[-\frac{\Delta t_k^2 (e^{-\alpha_i \Delta t_k} + 1)}{\alpha_i^3} + \frac{\Delta t_k (3 - e^{-2\alpha_i \Delta t_k} - 2e^{-\alpha_i \Delta t_k})}{\alpha_i^4} + \frac{4e^{-\alpha_i \Delta t_k} - 2e^{-2\alpha_i \Delta t_k} - 2}{\alpha_i^5} \right] \boldsymbol{\delta}_{ii}, \tag{B7b}$$

$$\frac{\partial \mathbf{Q}_{13}}{\partial \alpha_i} = \left[\frac{\Delta t_k^2 e^{-\alpha_i \Delta t_k}}{\alpha_i^2} + \frac{3(e^{-2\alpha_i \Delta t_k} - 1)}{2\alpha_i^4} + \frac{\Delta t_k (e^{-2\alpha_i \Delta t_k} + 2e^{-\alpha_i \Delta t_k})}{\alpha_i^3} \right] \boldsymbol{\delta}_{ii},$$

$$\frac{\partial \mathbf{Q}_{22}}{\partial \alpha_i} = \left[\frac{3e^{-2\alpha_i \Delta t_k} - 12e^{-\alpha_i \Delta t_k} + 9}{2\alpha_i^4} + \frac{\Delta t_k (e^{-2\alpha_i \Delta t_k} - 2e^{-\alpha_i \Delta t_k} - 2)}{\alpha_i^3} \right] \boldsymbol{\delta}_{ii}, \tag{B7c}$$

$$\frac{\partial \mathbf{Q}_{23}}{\partial \alpha_i} = \left[\frac{2e^{-\alpha_i \Delta t_k} - e^{-2\alpha_i \Delta t_k} - 1}{\alpha_i^3} + \frac{\Delta t_k (e^{-\alpha_i \Delta t_k} - e^{-2\alpha_i \Delta t_k})}{\alpha_i^2} \right] \boldsymbol{\delta}_{ii}, \tag{B7d}$$

$$\frac{\partial \mathbf{Q}_{33}}{\partial \alpha_i} = \left[\frac{e^{-2\alpha_i \Delta t_k} - 1}{2\alpha_i^2} + \frac{\Delta t_k e^{-2\alpha_i \Delta t_k}}{\alpha_i} \right] \boldsymbol{\delta}_{ii}, \tag{B7e}$$

$$\frac{\partial \mathbf{e}_k}{\partial \alpha_i} = -\begin{bmatrix} \left(\frac{2(1 - e^{-\alpha_i \Delta t_k})}{\alpha_i^3} - \frac{\Delta t_k (e^{-\alpha_i \Delta t_k} + 1)}{\alpha_i^2} \right) \boldsymbol{\delta}_{ii} \\ \left(\frac{e^{-\alpha_i \Delta t_k} - 1}{\alpha_i^2} + \frac{\Delta t_k e^{-\alpha_i \Delta t_k}}{\alpha_i} \right) \boldsymbol{\delta}_{ii} \\ \left(-\Delta t_k e^{-\alpha_i \Delta t_k} \right) \boldsymbol{\delta}_{ii} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \mathbf{x}_k, \tag{B7f}$$

where $\boldsymbol{\delta}_{ii}$ is the Kronecker delta. We can use these gradients to learn the parameters of the Singer prior using gradient descent. In order to speed up training, we can solve for the optimal value of σ_i^2 at each iteration of gradient descent:

$$\sigma_i^{2\star} = \frac{1}{3KT} \sum_t \sum_k \mathbf{e}_{k,t}^T \mathbf{Q}(\Delta t_{k,t}, \boldsymbol{\alpha})^{-1} \frac{\partial \mathbf{Q}_{\sigma^2}}{\partial \sigma_i^2} \mathbf{e}_{k,t}. \tag{B8}$$

Note that $\mathbf{Q}_k$ is numerically unstable for $\alpha < 1.0$. In this case, we use a Taylor series expansion about $\alpha = 0$ as an approximation. The Jacobian $\frac{\partial \mathbf{Q}_k}{\partial \alpha_i}$ is also numerically unstable for $\alpha < 4.0$. In this case, we can approximate the components of this matrix with either a Laurent series or Taylor series as $\alpha \to 0$.

The previous gradients work well for learning the parameters of a Gaussian process in simulation where the ground truth measurements of the state are noiseless. In reality, our source of ground truth will have some measurement covariance that may be estimated or taken from the datasheet of the sensor being used. In this case, computing the gradients of the objective with respect to the components of $\boldsymbol{\sigma}^2$ and $\boldsymbol{\alpha}$ is slightly more involved. We follow the approach presented by Wong et al. [30]. Now, our objective function looks at the entire trajectory at once,

$$J = -\ln p(\mathbf{y}|\boldsymbol{\sigma}, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{e}^T\mathbf{Q}^{-1}\mathbf{e} + \frac{1}{2}\ln|\mathbf{Q}| + \frac{n}{2}\ln 2\pi, \tag{B9}$$

where $\mathbf{e}$ is a stacked version of all the individual error terms from each timestep $\mathbf{e}_k$, and

$$\mathbf{Q} = \begin{bmatrix} \boldsymbol{\Sigma}_{0,0} & \boldsymbol{\Sigma}_{0,1} & & & \\ \boldsymbol{\Sigma}_{1,0}^T & \boldsymbol{\Sigma}_{1,1} & \boldsymbol{\Sigma}_{1,2} & & \\ & \boldsymbol{\Sigma}_{1,2}^T & \ddots & \ddots & \\ & & \ddots & \boldsymbol{\Sigma}_{K,K} \end{bmatrix}, \tag{B10}$$

where

$$\boldsymbol{\Sigma}_{k,k} \approx \mathbf{R}_k + \boldsymbol{\Phi}(t_k, t_{k-1})\mathbf{R}_{k-1}\boldsymbol{\Phi}(t_k, t_{k-1})^T + \mathbf{Q}_k, \tag{B11a}$$

$$\boldsymbol{\Sigma}_{k,k+1} \approx -\mathbf{R}_k\boldsymbol{\Phi}(t_{k+1}, t_k)^T, \tag{B11b}$$

and $\mathbf{R}_k$ is the measurement covariance associated with local variable $\mathbf{x}_k$. The gradient of the objective function J with respect to GP parameter θ is

$$\frac{\partial J}{\partial \theta} = -\frac{1}{2}\mathbf{e}^T\mathbf{Q}^{-1}\frac{\partial \mathbf{Q}}{\partial \theta}\mathbf{Q}^{-1}\mathbf{e} + \mathbf{e}^T\mathbf{Q}^{-1}\frac{\partial \mathbf{e}}{\partial \theta} + \frac{1}{2}\mathrm{tr}\left(\mathbf{Q}^{-1}\frac{\partial \mathbf{Q}}{\partial \theta}\right) \tag{B12}$$

where each of the Jacobians is evaluated using the current value of θ. We can compute the trace in $O(K)$ time by first computing only the block-tridiagonal components of $\mathbf{Q}^{-1}$. Since $\mathbf{Q}$ is itself block-tridiagonal, we can compute the blocks of the inverse that we need in $O(K)$ time [29]. Then, we can compute the trace of the matrix product in $O(K)$ time by only computing the elements along the diagonal of the matrix product. $\mathbf{Q}^{-1}\mathbf{e}$ can also be evaluated in $O(K)$ time by solving $\mathbf{Qx} = \mathbf{e}$ for $\mathbf{x}$ using a sparse Cholesky solver. Using the gradient in (B12) for each parameter, we can learn the parameters of the Gaussian process by minimizing the negative log likelihood using gradient descent.

C. IMU-as-Input Lidar-Inertial Baseline Jacobians

Perturbations to the state variables are defined as $\mathbf{C}_{iv} = \bar{\mathbf{C}}_{iv}\exp(\delta\boldsymbol{\phi}^\wedge)$, $\mathbf{r}_i^{vi} = \bar{\mathbf{r}}_i^{vi} + \bar{\mathbf{C}}_{iv}\delta\mathbf{r}$, $\mathbf{v}_i^{vi} = \bar{\mathbf{v}}_i^{vi} + \bar{\mathbf{C}}_{iv}\delta\mathbf{v}$, $\mathbf{b} = \bar{\mathbf{b}} + \delta\mathbf{b}$. The Jacobians of the point-to-plane error function (46) with respect to perturbations to the state variables are provided here,

$$\frac{\partial \mathbf{e}_j}{\partial \delta\mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{e}_j}{\partial \mathbf{r}_i^{vi}(\tau_j)} & \frac{\partial \mathbf{e}_j}{\partial \delta\mathbf{C}_{iv}(\tau_j)} \end{bmatrix} \times \begin{bmatrix} \frac{\partial \mathbf{r}_i^{vi}(\tau_j)}{\partial \mathbf{r}_\ell}\frac{\partial \mathbf{r}_\ell}{\partial \delta\mathbf{x}} + \frac{\partial \mathbf{r}_i^{vi}(\tau_j)}{\partial \mathbf{r}_{\ell+1}}\frac{\partial \mathbf{r}_{\ell+1}}{\partial \delta\mathbf{x}} \\ \frac{\partial \delta\mathbf{C}_{iv}(\tau_j)}{\partial \delta\mathbf{C}_\ell}\frac{\partial \delta\mathbf{C}_\ell}{\partial \delta\mathbf{x}} + \frac{\partial \delta\mathbf{C}_{iv}(\tau_j)}{\partial \delta\mathbf{C}_{\ell+1}}\frac{\partial \delta\mathbf{C}_{\ell+1}}{\partial \delta\mathbf{x}} \end{bmatrix}, \tag{C1}$$

where

$$\frac{\partial \mathbf{e}_j}{\partial \mathbf{r}_i^{vi}(\tau_j)} = -\mathbf{n}_j^T, \quad \frac{\partial \mathbf{e}_j}{\partial \delta \mathbf{C}_{iv}(\tau_j)} = \mathbf{n}_j^T \left(\overline{\mathbf{C}}_{iv}(\tau_j)(\mathbf{C}_{vs}\mathbf{q}_j + \mathbf{r}_v^{sv})^\wedge \right), \tag{C2a}$$

$$\frac{\partial \mathbf{r}_i^{vi}(\tau_j)}{\partial \mathbf{r}_\ell} = (1-\alpha)\mathbf{1}, \quad \frac{\partial \mathbf{r}_i^{vi}(\tau_j)}{\partial \mathbf{r}_{\ell+1}} = \alpha \mathbf{1}, \tag{C2b}$$

$$\frac{\partial \delta \mathbf{C}_{iv}(\tau_j)}{\partial \delta \mathbf{C}_\ell} = \mathbf{1} - \mathbf{A}(\alpha, \boldsymbol{\phi}), \quad \frac{\partial \delta \mathbf{C}_{iv}(\tau_j)}{\partial \delta \mathbf{C}_{\ell+1}} = \mathbf{A}(\alpha, \boldsymbol{\phi}), \tag{C2c}$$

where $\mathbf{A}(\alpha, \boldsymbol{\phi}) = \alpha \mathbf{J}_r(\alpha\boldsymbol{\phi})\mathbf{J}_r(\boldsymbol{\phi})^{-1}$, $\boldsymbol{\phi} = \ln(\mathbf{C}_\ell^T \mathbf{C}_{\ell+1})^\vee$, and

$$\frac{\partial \delta \mathbf{C}_\ell}{\partial \delta \mathbf{C}_i} = \Delta \overline{\mathbf{C}}_{i\ell}^T, \text{ where } \quad \Delta \mathbf{C}_{i\ell} = \prod_{k=i}^{\ell-1} \exp\left(\Delta t_k (\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_\omega(t_k))^\wedge\right), \tag{C3a}$$

$$\frac{\partial \delta \mathbf{C}_\ell}{\partial \delta \mathbf{b}_\omega(t_i)} = -\sum_{k=i}^{\ell-1} \Delta \overline{\mathbf{C}}_{k+1,\ell}^T \mathbf{J}_r(\boldsymbol{\phi}_k)\Delta t_k, \text{ where } \quad \boldsymbol{\phi}_k = \Delta t_k (\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_\omega(t_k)), \tag{C3b}$$

$$\frac{\partial \mathbf{v}_\ell}{\partial \delta \mathbf{v}_i} = \mathbf{C}_i, \tag{C3c}$$

$$\frac{\partial \mathbf{v}_\ell}{\partial \delta \mathbf{C}_i} = -\sum_{k=i}^{\ell-1} \mathbf{C}_k(\tilde{\mathbf{a}}_k - \mathbf{b}_a(t_k))^\wedge \Delta \overline{\mathbf{C}}_{ik}^T \Delta t_k, \tag{C3d}$$

$$\frac{\partial \mathbf{v}_\ell}{\partial \delta \mathbf{b}_\omega(t_i)} = -\sum_{k=i}^{\ell-1} \mathbf{C}_k(\tilde{\mathbf{a}}_k - \mathbf{b}_a(t_k))^\wedge \frac{\partial \delta \mathbf{C}_k}{\partial \delta \mathbf{b}_\omega(t_i)} \Delta t_k, \tag{C3e}$$

$$\frac{\partial \mathbf{v}_\ell}{\partial \delta \mathbf{b}_a(t_i)} = -\sum_{k=i}^{\ell-1} \mathbf{C}_k \Delta t_k, \tag{C3f}$$

$$\frac{\partial \mathbf{r}_\ell}{\partial \delta \mathbf{v}_i} = \mathbf{C}_i \Delta t_{ij}, \tag{C3g}$$

$$\frac{\partial \mathbf{r}_\ell}{\partial \delta \mathbf{C}_i} = \sum_{k=i}^{\ell-1} \left[\frac{\partial \mathbf{v}_k}{\partial \delta \mathbf{C}_i} \Delta t_k - \frac{1}{2} \mathbf{C}_k (\tilde{\mathbf{a}}_k - \mathbf{b}_a(t_k))^\wedge \Delta \overline{\mathbf{C}}_{ik}^T \Delta t_k^2 \right], \tag{C3h}$$

$$\frac{\partial \mathbf{r}_\ell}{\partial \delta \mathbf{b}_\omega(t_i)} = \sum_{k=i}^{\ell-1} \left[\frac{\partial \mathbf{v}_k}{\partial \delta \mathbf{b}_\omega(t_i)} \Delta t_k - \frac{1}{2} \mathbf{C}_k (\tilde{\mathbf{a}}_k - \mathbf{b}_a(t_k))^\wedge \frac{\partial \delta \mathbf{C}_k}{\partial \delta \mathbf{b}_\omega(t_i)} \Delta t_k^2 \right], \tag{C3i}$$

$$\frac{\partial \mathbf{r}_\ell}{\partial \delta \mathbf{b}_a(t_i)} = \sum_{k=i}^{\ell-1} \left[\frac{\partial \mathbf{v}_k}{\partial \delta \mathbf{b}_a(t_i)} \Delta t_k - \frac{1}{2} \mathbf{C}_k \Delta t_k^2 \right]. \tag{C3j}$$

(C3k)