Technical Note: Computational Cost of the CRT Estimator

Paul F. Roysdon†

Jay A. Farrell‡

I. Introduction

In this Technical Note we consider the computational cost (number of floating-point operations (FLOPs)) for each update of the CRT estimator. We assume that the implementation uses a sparse matrix library, and only compute and store the upper-triangular elements of all symmetric positive-definite matrices used in the CRT estimator. First we define the computational cost of the Kalman Filter (KF) [1], then evaluate the most expensive operation, the computation of $\boldsymbol{\delta}\mathbf{X}$ in the CRT estimator [2].

Consider the general matrices/vectors, $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{v} \in \mathbb{R}^{n\times 1}$, upper-triangular $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$, and $\mathbf{B} \in \mathbb{R}^{m\times n}$, with $m > n$. The computational cost for basic linear algebra operations on dense matrices [3] is shown in the first-half of Table I. If $\mathbf{A}$ and $\mathbf{B}$ are sparse, a lower-bound can be determined using sparse matrix operations (see [4]), the results are shown in the second-half of Table I.

The computational cost for three matrix decompositions is provided in Table II.

The computational cost for three Least-Square (LS) algorithms is provided in Table III.

TABLE I

Computational cost for dense and sparse matrices.

Operation	Computational Cost
Dense matrices	
$\mathbf{A}^{-1}$	$\approx n^3$
$\mathbf{AA}$	n^3
$\mathbf{Av}$	n^2
$\mathbf{AB}^\intercal$	mn^2
$\mathbf{B}^\intercal\mathbf{B}$	mn^2
$\mathbf{Bv}$	mn
$\mathbf{\Sigma v}$	$\frac{n^2-n}{2}+n$
$\mathbf{\Sigma A}$	$\frac{n^3-n^2}{2}+n^2$
$\mathbf{\Sigma\Sigma}$	$\frac{n^3}{3}$
Sparse matrices	
$\mathbf{A}^{-1}$	n
$\mathbf{AA}$	n
$\mathbf{Av}$	n
$\mathbf{AB}^\intercal$	n
$\mathbf{B}^\intercal\mathbf{B}$	n
$\mathbf{Bv}$	n

II. Computational cost for KF

For the KF state estimate $\hat{\boldsymbol{x}} \in \mathbb{R}^n$, and measurement $\mathbf{y} \in \mathbb{R}^m$ with $m > n$, the computational cost with dense matrices is presented in Table IV.

†Ph.D. graduate, ‡Professor at the Dept. of Electrical & Computer Engineering, UC Riverside. {proysdon, farrell}@ece.ucr.edu.

TABLE II

Computational cost for matrix decompositions.

Decomposition	Computational Cost
Cholesky (Section 4.2.1 [3])	$\frac{n^3}{3}$
Householder QR (Section 5.2.1 [3])	$4\left(m^2n - mn^2 + \frac{n^3}{3}\right)$
Givens QR (Section 5.2.3 [3])	$3n^2\left(m - \frac{n}{3}\right)$

TABLE III

Computational cost for LS algorithms (Section 5.5.9 [3]).

LS Algorithm	Computational Cost
Normal Equations	$mn^2 + \frac{n^3}{3}$
Householder QR Orthogonalization	$2mn^2 + \frac{2n^3}{3}$
Givens QR Orthogonalization	$3mn^2 - n^3$

TABLE IV

Computational cost for the KF (Section 3.3.1 [5]).

Operation	Computational Cost
System Propagation	
$\hat{\boldsymbol{x}}_k^- = \mathbf{\Phi}_{k-1}\hat{\boldsymbol{x}}_{k-1}^+$	n^2
$\mathbf{P}_k^- = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}^-\mathbf{\Phi}_{k-1}^\intercal + \mathbf{Qd}_{k-1}$	$2n^3$
Measurement update	
$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^\intercal(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^\intercal + \mathbf{R}_k)^{-1}$	$2mn^2 + 2mn$
$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\hat{\boldsymbol{x}}_k^-)$	$2mn$
$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-$	$2mn^2$

III. Computational cost for CRT using Normal Equations

This section is split into two primary parts: first compute $\boldsymbol{\delta}\mathbf{X}$ by dense matrices, then compute $\boldsymbol{\delta}\mathbf{X}$ by sparse matrices. In both cases we only consider one iteration of the optimization. Because the calculation of $\boldsymbol{\delta}\mathbf{X}$ is the most expensive operation in the CRT estimator, assume the state transition matrix $\mathbf{\Phi}$, measurement matrix $\mathbf{H}$, measurement prediction $\hat{\mathbf{y}}$, integrated state vector $\boldsymbol{x}_i$, and optimized state vector $\boldsymbol{x}^*$ are each given (refer to Sections 2 & 3 of [2]) and are therefore not included in the cost to compute $\boldsymbol{\delta}\mathbf{X}$.

A. Dense Jacobian

Given the number of error states n_e, the number of measurements n_m, and CRT window length L, the components of the normalized Jacobian matrix $\mathbf{J}$ and residual vector $\mathbf{r}$, are defined below with a mapping to the number of computations required for dense matrices. Both $\mathbf{J}$ and $\mathbf{r}$

are normalized using Cholesky factorization[1], which has computational complexity $\frac{n^3}{3}$. Define $\boldsymbol{\delta}\mathbf{X}_O$, $\mathbf{r}_O$, and $\mathbf{J}_O$ as the number of operations to compute $\boldsymbol{\delta}\mathbf{X}$, $\mathbf{r}$, and $\mathbf{J}$, respectively.

The components of the Jacobian are,

$$\begin{aligned}
\mathbf{J}_{\mathbf{P}_0} &\triangleq \boldsymbol{\Sigma}_{\mathbf{P}_0}[\mathbf{I}, \mathbf{0}] \mapsto \left(\frac{n_e^3}{3} + n_e^3\right) \\
\mathbf{J}_{\mathbf{Qd}} &\triangleq \boldsymbol{\Sigma}_{\mathbf{Qd}}[\boldsymbol{\Phi}, -\mathbf{I}] \mapsto \left(\frac{n_e^3}{3} + 2n_e^3\right) L \\
\mathbf{J}_{\mathbf{R}} &\triangleq \boldsymbol{\Sigma}_{\mathbf{R}}\mathbf{H} \mapsto \left(\frac{n_m^3}{3} + n_m n_e^2\right) L \\
\mathbf{J} &\triangleq \begin{bmatrix} \mathbf{J}_{\mathbf{P}_0} \\ \mathbf{J}_{\mathbf{Qd}} \\ \mathbf{J}_{\mathbf{R}} \end{bmatrix} \mapsto \begin{bmatrix} \frac{n_e^3}{3} + n_e^3 \\ \left(\frac{n_e^3}{3} + 2n_e^3\right) L \\ \left(\frac{n_m^3}{3} + n_m n_e^2\right) L \end{bmatrix},
\end{aligned}$$

where, $\mathbf{J}_{\mathbf{P}_0} \in \mathbb{R}^{n_e \times n_e}$, $\mathbf{J}_{\mathbf{Qd}} \in \mathbb{R}^{n_e L \times n_e L}$, $\mathbf{J}_{\mathbf{R}} \in \mathbb{R}^{n_m L \times n_e L}$, and $\mathbf{J} \in \mathbb{R}^{(n_e(L+1)+n_m L) \times (n_e(L+1))}$. The number of operations to compute $\mathbf{J}$ is

$$\mathbf{J}_O = \frac{n_e^3}{3} + n_e^3 + \left(\frac{n_e^3}{3} + 2n_e^3 + \frac{n_m^3}{3} + n_m n_e^2\right) L.$$

The components of the residual vector are,

$$\begin{aligned}
\mathbf{r}_{\mathbf{P}_0} &\triangleq \boldsymbol{\Sigma}_{\mathbf{P}_0}[\boldsymbol{\mu} - \boldsymbol{x}^*] \mapsto n_e^2 \\
\mathbf{r}_{\mathbf{Qd}} &\triangleq \boldsymbol{\Sigma}_{\mathbf{Qd}}[\boldsymbol{x}^* - \boldsymbol{x}_i] \mapsto n_e^2 L \\
\mathbf{r}_{\mathbf{R}} &\triangleq \boldsymbol{\Sigma}_{\mathbf{R}}\boldsymbol{\delta}\mathbf{y} \mapsto n_m n_e L \\
\mathbf{r} &\triangleq \begin{bmatrix} \mathbf{r}_{\mathbf{P}_0} \\ \mathbf{r}_{\mathbf{Qd}} \\ \mathbf{r}_{\mathbf{R}} \end{bmatrix} \mapsto \begin{bmatrix} n_e^2 \\ n_e^2 L \\ n_m n_e L \end{bmatrix},
\end{aligned}$$

where, $\mathbf{r}_{\mathbf{P}_0} \in \mathbb{R}^{n_e \times 1}$, $\mathbf{r}_{\mathbf{Qd}} \in \mathbb{R}^{n_e L \times 1}$, $\mathbf{r}_{\mathbf{R}} \in \mathbb{R}^{n_m L \times 1}$, and $\mathbf{r} \in \mathbb{R}^{(n_e(L+1)+n_m L) \times 1}$. Note, the $\frac{n_e^3}{3}$ and $\frac{n_m^3}{3}$ are not included in the components $\mathbf{r}$, because the cost to compute the Cholesky decomposition is included in the components of $\mathbf{J}$. The number of operations to compute $\mathbf{r}$ is

$$\mathbf{r}_O = n_e^2 + (n_e^2 + n_m n_e)L.$$

Given the residual vector $\mathbf{r} \in \mathbb{R}^{p \times 1}$ and (dense) Jacobian $\mathbf{J} \in \mathbb{R}^{p \times q}$ for $p > q$, where $p = (n_e \times (L+1)) + (n_m \times L)$ and $q = (n_e \times (L+1))$, the number of operations to compute $\boldsymbol{\delta}\mathbf{X}$ by the Normal Equations is

$$\boldsymbol{\delta}\mathbf{X} = (\mathbf{J}^\intercal\mathbf{J})^{-1}\mathbf{J}^\intercal\mathbf{r} \mapsto pq^2 + \frac{q^3}{3} = \boldsymbol{\delta}\mathbf{X}_O.$$

Expanding p and q:

$$\begin{aligned}
\boldsymbol{\delta}\mathbf{X}_O = n_e^3 &\left(\frac{4}{3}L^3 + 4L^2 + 4L + \frac{4}{3}\right) \\
&+ n_e^2 n_m (L^3 + 2L^2 + L).
\end{aligned}$$

[1] Householder QR, or Givens QR factorization may be used instead (see Table II).

Define $\boldsymbol{\delta}\mathbf{X}_{TO}$ as the total number of operations to compute $\boldsymbol{\delta}\mathbf{X}$ by the Normal Equations.

$$\begin{aligned}
\boldsymbol{\delta}\mathbf{X}_{TO} &= \boldsymbol{\delta}\mathbf{X}_O + \mathbf{r}_O + \mathbf{J}_O \\
&= n_e^3 \left(\frac{4}{3}L^3 + 4L^2 + 4L + \frac{4}{3}\right) \\
&\quad + n_e^2 n_m (L^3 + 2L^2 + L). \\
&\quad + n_e^2 + \left(n_e^2 + n_m n_e\right) L \\
&\quad + \frac{n_e^3}{3} + n_e^3 + \left(\frac{n_e^3}{3} + n_e^2 + \frac{n_m^3}{3} + n_m n_e^2\right) L \\
&= n_e^3 \left(\frac{4}{3}L^3 + 4L^2 + \frac{19}{3}L + \frac{8}{3}\right) \\
&\quad + n_e^2 (n_m L^3 + 2n_m L^2 + 2n_m L + L + 1) \\
&\quad + n_e n_m L + \frac{n_m^3}{3}L.
\end{aligned}$$

Therefore the computation for a single iteration of the dense $\boldsymbol{\delta}\mathbf{X}$ by the Normal Equations is $\mathcal{O}(n_e^3 L^3)$.

B. Sparse Jacobian

Presently, common sparse libraries do not support LS by the Normal Equations because the operations are not efficient (see [3] and [4]). By inspection, the operation $\mathbf{J}^\intercal\mathbf{J}$ produces a dense matrix, and the inversion of $\mathbf{J}^\intercal\mathbf{J}$ is $\mathcal{O}(n_e^3 L^3)$.

IV. Computational cost for CRT using QR Factorization

This section first introduces the theory for solving systems of linear equations using QR factorization, then follows an outline similar to Section III. The same assumptions of Section III apply here.

A. Theory

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$ and $\mathbf{b} \in \mathbb{R}^m$. Suppose that the orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{m \times m}$ is computed such that

$$\mathbf{Q}^\intercal\mathbf{A} = \mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix}$$

is upper-triangular, with $\mathbf{R}_1 \in \mathbb{R}^{n \times n}$ and $\mathbf{0} \in \mathbb{R}^{(m-n) \times n}$. If

$$\mathbf{Q}^\intercal\mathbf{b} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}$$

where $\mathbf{c} \in \mathbb{R}^{n \times 1}$ and $\mathbf{d} \in \mathbb{R}^{(m-n) \times 1}$, then

$$\begin{aligned}
\|\mathbf{A}\boldsymbol{x} - \mathbf{b}\|_2^2 &= \|\mathbf{Q}^\intercal\mathbf{A}\boldsymbol{x} - \mathbf{Q}^\intercal\mathbf{b}\|_2^2 \\
&= \|\mathbf{R}_1\boldsymbol{x} - \mathbf{c}\|_2^2 + \|\mathbf{d}\|_2^2 \qquad (1)
\end{aligned}$$

for any $\boldsymbol{x} \in \mathbb{R}^n$. If $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{R}_1) = n$, then the Least-Square estimate $\hat{\boldsymbol{x}}$ is defined by the upper-triangular system $\mathbf{R}_1\hat{\boldsymbol{x}} = \mathbf{c}$. Given $\mathbf{R}_1$, solving this system requires $2n^2\left(m - \frac{n}{3}\right)$ FLOPs. While $\mathcal{O}(mn)$ are required to update $\mathbf{c}$, and $\mathcal{O}(n^2)$ are required for back-substitution (see Section 5.3.2 of [3]), the most expensive operation is the QR factorization of $\mathbf{A}$, which requires at least $3n^2\left(m - \frac{n}{3}\right)$ FLOPs (assuming Givens QR).

Modifying eqn. (1) for the CRT estimator, define $\mathbf{J} \in \mathbb{R}^{p\times q}$, $\mathbf{r} \in \mathbb{R}^{p\times 1}$, $\boldsymbol{\delta}\mathbf{X} \in \mathbb{R}^{q\times 1}$, $\mathbf{R}_1 \in \mathbb{R}^{q\times q}$, $\mathbf{c} \in \mathbb{R}^{q\times 1}$ and $\mathbf{d} \in \mathbb{R}^{(p-q)\times 1}$. For

$$\mathbf{J}\boldsymbol{\delta}\mathbf{X} = \mathbf{r}$$

the Least-Square estimate $\boldsymbol{\delta}\hat{\mathbf{X}}$ of $\boldsymbol{\delta}\mathbf{X}$ is found by

$$\|\mathbf{J}\boldsymbol{\delta}\mathbf{X} - \mathbf{r}\|_2^2 = \|\mathbf{R}_1\boldsymbol{\delta}\mathbf{X} - \mathbf{c}\|_2^2 + \|\mathbf{d}\|_2^2$$

where $\boldsymbol{\delta}\hat{\mathbf{X}}$ is defined by the upper-triangular system $\mathbf{R}_1\boldsymbol{\delta}\hat{\mathbf{X}} = \mathbf{c}$.

B. Dense Jacobian

Let $\mathbf{J}_{QR,d}$ denote the number of operations to compute the QR factorization of (dense) $\mathbf{J}$. Using the previously defined values for p and q, and the values in Table II,

$$\begin{aligned}\mathbf{J}_{QR,d} &= 3q^2\left(p - \frac{q}{3}\right)\\ &= 2n_e^3(L^3 + 3L^2 + 3L + 1)\\ &\quad + 3n_e^2 n_m(L^3 + 2L^2 + L).\end{aligned}$$

Let $\boldsymbol{\delta}\hat{\mathbf{X}}_{QR,d}$ denote the number of operations to compute the Least-Square estimate $\boldsymbol{\delta}\hat{\mathbf{X}}$ with Givens QR of (dense) $\mathbf{J}$ (see Table III). Then

$$\begin{aligned}\boldsymbol{\delta}\hat{\mathbf{X}}_{QR,d} &= 2q^2\left(p - \frac{q}{3}\right)\\ &= 2n_e^3(L^3 + 3L^2 + 3L + 1)\\ &\quad + 3n_e^2 n_m(L^3 + 2L^2 + L).\end{aligned}$$

Therefore the computation for a single iteration of the dense $\boldsymbol{\delta}\mathbf{X}$ using Givens QR is $\mathcal{O}(n_e^3 L^3)$.

C. Sparse Jacobian

Let $\mathbf{J}_{QR,s}$, $\mathbf{r}_{QR,s}$, and $\boldsymbol{\delta}\mathbf{X}_{QR,s}$, denote the number of operations to compute the QR factorization of (sparse) $\mathbf{J}$, $\mathbf{r}$, and $\boldsymbol{\delta}\mathbf{X}$, respectively. The components of the sparse Jacobian are,

$$\begin{aligned}\mathbf{J}_{\mathbf{P}_0} &\mapsto \left(\frac{n_e^3}{3} + \frac{n_e^2 - n_e}{2} + n_e\right)\\ \mathbf{J}_{\mathbf{Qd}} &\mapsto \left(\frac{n_e^3}{3} + \frac{n_e^3 - n_e^2}{2} + n_e^2 + \frac{n_e^2 - n_e}{2} + n_e\right)L\\ \mathbf{J}_{\mathbf{R}} &\mapsto \left(\frac{n_m^3}{3} + 3n_m\right)L.\end{aligned}$$

The number of operations to compute $\mathbf{J}$ is

$$\begin{aligned}\mathbf{J}_{QR,s} &= \frac{n_e^3}{3} + \frac{n_e^2 - n_e}{2} + n_e\\ &\quad + \left(\frac{n_e^3}{3} + \frac{n_e^3 - n_e^2}{2} + n_e^2 + \frac{n_e^2 - n_e}{2} + n_e\right)L\\ &\quad + \left(\frac{n_m^3}{3} + 3n_m\right)L.\end{aligned}$$

The components of the residual vector are,

$$\begin{aligned}\mathbf{r}_{\mathbf{P}_0} &\mapsto \left(\frac{n_e^2 - n_e}{2} + n_e\right)\\ \mathbf{r}_{\mathbf{Qd}} &\mapsto \left(\frac{n_e^2 - n_e}{2} + n_e\right)L\\ \mathbf{r}_{\mathbf{R}} &\mapsto n_m L.\end{aligned}$$

The number of operations to compute $\mathbf{r}$ is

$$\mathbf{r}_{QR,s} = \frac{n_e^2 - n_e}{2} + n_e + \left(\frac{n_e^2 - n_e}{2} + n_e + n_m\right)L$$

Using a sparse library [4], the LS solution via QR factorization is proportional to the number of non-zero elements in the Jacobian (see [3] and [4]). Therefore

$$\boldsymbol{\delta}\mathbf{X}_{QR,s} = n_e(L+1) + n_m L$$

Define $\boldsymbol{\delta}\mathbf{X}_{QR,TO}$ as the total number of operations to compute $\boldsymbol{\delta}\mathbf{X}$ by the sparse QR factorization.

$$\begin{aligned}\boldsymbol{\delta}\mathbf{X}_{QR,TO} &= \boldsymbol{\delta}\mathbf{X}_O + \mathbf{r}_O + \mathbf{J}_O\\ &= n_e^3\left(\frac{1}{3} + \frac{1}{3}L\right)\\ &\quad + n_e^2\left(\frac{n_e - 1}{2}L + 1\right)\\ &\quad + n_e(3L + 3)\\ &\quad + n_m\left(\frac{1}{3}n_m^2 L + 5L\right).\end{aligned}$$

Therefore the computation for a single iteration of the sparse $\boldsymbol{\delta}\mathbf{X}$ using QR factorization is $\mathcal{O}(n_e^3 L)$.

V. Discussion

A. Normal Equations vs. QR

In Section 5.3.8 of [3], the authors identify the challenge with selecting the "right" algorithm to solve a linear system of equations.

- Condition Number: while the Normal Equations effectively square the condition number, QR does not.
- Arithmetic: the Normal Equations use roughly half of the arithmetic as QR when $m \gg n$, while also requiring less storage (memory).
- Applicability: unlike the Normal Equations, QR is more widely applicable because the errors that arise in $\mathbf{A}^\intercal\mathbf{A}$ of the Normal Equations "break down" more quickly than the process on $\mathbf{Q}^\intercal\mathbf{A} = \mathbf{R}$ of LS by QR.

B. Sparse vs. Dense

Considering only LS solutions using QR, Sections IV-B and IV-C clearly define the cost savings using sparse matrices which are $\mathcal{O}(n_e^3 L)$, versus dense matrices which are $\mathcal{O}(n_e^3 L^3)$.

However, on modern computers with dual-precision (64-bit) floating point processors, the primary factor in computational cost is not FLOPs, rather in the storage and memory access of the data [4]. Identifying the computational complexity is important to understand the amount of storage required by a particular algorithm, as sparse representations use less memory and require fewer memory accesses.

C. Estimator Comparison

The computational cost for the calculation of $\boldsymbol{\delta}\mathbf{X}$ in the Iterated Extended KF (IEKF), e.g. CRT with $L = 1$, is $\mathcal{O}(n_e^3)$, whereas the most expensive computation of the KF is the state covariance propagation, $\mathcal{O}(n_e^3)$. In contrast, the CRT estimator, with window length L, scales linearly in L with $\mathcal{O}(n_e^3 L)$ when using sparse matrices.

References

[1] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors*. McGraw Hill, 2008.

[2] P. F. Roysdon and J. A. Farrell, "GPS-INS Outlier Detection and Elimination using a Sliding Window Filter," *American Control Conference, In Presc.*, 2017.

[3] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.

[4] J. R. Gilbert, C. Moler, and R. Schreiber, "Sparse matricies in Matlab: Design and Implementation," *The Mathworks Incorporated*, 1991.

[5] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2013.