https://ntrs.nasa.gov/search.jsp?R=20170000373 2019-08-29T15:51:16+00:00Z Comparison of Factorization-based Filtering for Landing Navigation

James S. McCabe 1 Aaron J. Brown 2 Kyle J. DeMars 1 John M. Carson III 2

¹Missouri University of Science and Technology

²NASA Johnson Space Center





Modeling 0000 Results Co

Conclusions









Modeling 0000 Results C

Conclusions



Motivation





Modeling 0000 Results C

Conclusions



Motivation



McCabe, et al.

Comparison of Factorization-based Filtering for Landing Navigation

2 / 27



g Modeling 0000 Results Co

Conclusions



Motivation



Introd	luction
000	

d Filtering Modeling 0000 Results 0000000000

Conclusions



Objective

- Blend data from sensors
 - inertial measurement unit
 - star camera
 - \circ altimeter
 - \circ velocimeter
 - terrain camera

Introd	luction
000	

iltering Modeling

Results C

Conclusions



Objective

- Blend data from sensors
 - inertial measurement unit
 - star camera
 - \circ altimeter
 - \circ velocimeter
 - terrain camera
- Produce estimates of position, velocity, and attitude
 - accurate
 - \circ precise
 - \circ consistent
 - \circ robust

Introd	luction
000	

Filtering Modeling 0000 Results C

Conclusions



Objective

- Blend data from sensors
 - inertial measurement unit
 - star camera
 - \circ altimeter
 - \circ velocimeter
 - terrain camera
- Produce estimates of position, velocity, and attitude
 - accurate
 - \circ precise
 - \circ consistent
 - \circ robust
- Work within minimum variance estimation framework
 - Kalman filter
 - $\circ~$ extended Kalman filter
 - $\circ~$ unscented Kalman filter

Introduction

Factorization-based Filtering

Modeling 0000

Results Conclusions



- Blend data from sensors
 - inertial measurement unit
 - star camera
 - altimeter
 - \circ velocimeter
 - terrain camera
- Produce estimates of position, velocity, and attitude
 - accurate
 - \circ precise
 - \circ consistent
 - \circ robust
- Work within minimum variance estimation framework
 - Kalman filter
 - $\circ~$ extended Kalman filter
 - unscented Kalman filter

Introduction Fa

Factorization-based Filtering

Results 000000000 Conclusions



Scope of this Work

- Blend data from sensors
 - \checkmark inertial measurement unit
 - 🗸 star camera
 - ✓ altimeter
 - \circ velocimeter
 - terrain camera
- Produce estimates of position, velocity, and attitude

Modeling

- accurate
- \circ precise
- \circ consistent
- \circ robust
- Work within minimum variance estimation framework
 - Kalman filter
 - extended Kalman filter
 - unscented Kalman filter

Introduction F

Factorization-based Filtering

Modeling Results

Conclusions



- Blend data from sensors
 - \checkmark inertial measurement unit
 - 🗸 star camera
 - ✓ altimeter
 - \circ velocimeter
 - terrain camera
- Produce estimates of position, velocity, and attitude
 - accurate
 - \circ precise
 - ✓ consistent
 - 🗸 robust
- Work within minimum variance estimation framework
 - Kalman filter
 - extended Kalman filter
 - unscented Kalman filter

Introduction F

Factorization-based Filtering

Modeling Results

Conclusions



- Blend data from sensors
 - \checkmark inertial measurement unit
 - 🗸 star camera
 - ✓ altimeter
 - \circ velocimeter
 - terrain camera
- Produce estimates of position, velocity, and attitude
 - accurate
 - \circ precise
 - ✓ consistent
 - 🗸 robust
- Work within minimum variance estimation framework
 - Kalman filter
 - extended Kalman filter
 - $\circ~$ unscented Kalman filter

Introduction F

Factorization-based Filtering

Modeling Results

s Conclusions



- Blend data from sensors
 - ✓ inertial measurement unit
 - 🗸 star camera
 - ✓ altimeter
 - \circ velocimeter
 - terrain camera
- Produce estimates of position, velocity, and attitude
 - accurate
 - \circ precise
 - ✓ consistent
 - 🗸 robust
- Work within minimum variance estimation framework
 - Kalman filter
 - extended Kalman filter
 - unscented Kalman filter

Introduction

Factorization-based Filtering

Modeling F

Results Con

Conclusions



Scope of this Work

- Blend data from sensors
 - ✓ inertial measurement unit
 - 🗸 star camera
 - ✓ altimeter
 - \circ velocimeter
 - terrain camera
- Produce estimates of position, velocity, and attitude
 - accurate
 - o precise
 - ✓ consistent
 - 🗸 robust
- Work within minimum variance estimation framework
 - Kalman filter
 - extended Kalman filter
 multiplicative extended Kalman filter
 - unscented Kalman filter

multiplicative extended Kalman filter multiplicative unscented Kalman filter Introduction

Factorization-based Filtering

Modeling Results

Conclusions

MISSOURI SET

Scope of this Work

- Blend data from sensors
 - ✓ inertial measurement unit
 - 🗸 star camera
 - ✓ altimeter
 - \circ velocimeter
 - terrain camera
- Produce estimates of position, velocity, and attitude
 - accurate
 - \circ precise
 - ✓ consistent
 - 🗸 robust
- Work within minimum variance estimation framework
 - Kalman filter
 - ✓ extended Kalman filter multiplicative extended Kalman filter
 - unscented Kalman filter

multiplicative extended Kalman filter multiplicative unscented Kalman filter

Introduction	Fac
000	

Modeling Results

Conclusions



The Linear Problem

• Consider the linear state-space model

$$egin{aligned} oldsymbol{x}_k &= oldsymbol{F}_{k-1}oldsymbol{x}_{k-1} + oldsymbol{w}_{k-1} \ oldsymbol{z}_k &= oldsymbol{H}_koldsymbol{x}_k + oldsymbol{v}_k \end{aligned}$$

Introduction	Factorization-ba
000	00000

sed Filtering Modeling

Results Conclusions



The Linear Problem

• Consider the linear state-space model

$$egin{aligned} oldsymbol{x}_k &= oldsymbol{F}_{k-1}oldsymbol{x}_{k-1} + oldsymbol{w}_{k-1} \ oldsymbol{z}_k &= oldsymbol{H}_koldsymbol{x}_k + oldsymbol{v}_k \end{aligned}$$

• The well-known Kalman filter produces the conditional mean and covariance through a two-stage recursion:

Initial Cond.	$m_{k-1}^+=m_0$
	$\boldsymbol{P}_{k-1}^+=\boldsymbol{P}_0$
Mean Prop.	$m{m}_k^- \ = m{F}_{k-1} m{m}_{k-1}^+$
Cov. Prop.	$oldsymbol{P}_k^- = oldsymbol{F}_{k-1}oldsymbol{P}_{k-1}^+oldsymbol{F}_{k-1}^T + oldsymbol{Q}_{k-1}$
Kalman Gain	$oldsymbol{K}_k = oldsymbol{P}_k^-oldsymbol{H}_k^T [oldsymbol{H}_k oldsymbol{P}_k^-oldsymbol{H}_k^T + oldsymbol{R}_k]^{-1}$
Mean Update	$m{m}_k^+ \;= m{m}_k^- + m{K}_k (m{z}_k - m{H}_k m{m}_k^-)$
Cov. Update	$oldsymbol{P}_k^+ = oldsymbol{P}_k^ oldsymbol{K}_koldsymbol{H}_koldsymbol{P}_k^-$

Introduction	Factorization-bas
000	00000

ed Filtering Modeling

Results Co

Conclusions



The Linear Problem

• Consider the linear state-space model

$$egin{aligned} oldsymbol{x}_k &= oldsymbol{F}_{k-1}oldsymbol{x}_{k-1} + oldsymbol{w}_{k-1} \ oldsymbol{z}_k &= oldsymbol{H}_koldsymbol{x}_k + oldsymbol{v}_k \end{aligned}$$

• The well-known Kalman filter produces the conditional mean and covariance through a two-stage recursion:

Initial Cond.	$m_{k-1}^+=m_0$
	$\boldsymbol{P}_{k-1}^+=\boldsymbol{P}_0$
Mean Prop.	$m_k^- \ = {\it F}_{k-1} m_{k-1}^+$
Cov. Prop.	$oldsymbol{P}_k^- = oldsymbol{F}_{k-1} oldsymbol{P}_{k-1}^+ oldsymbol{F}_{k-1}^T + oldsymbol{Q}_{k-1}$
Kalman Gain	$oldsymbol{K}_k =oldsymbol{P}_k^-oldsymbol{H}_k^T[oldsymbol{H}_koldsymbol{P}_k^-oldsymbol{H}_k^T+oldsymbol{R}_k]^{-1}$
Mean Update	$m{m}_k^+ \;= m{m}_k^- + m{K}_k (m{z}_k - m{H}_k m{m}_k^-)$
Cov. Update	$oldsymbol{P}_k^+ = oldsymbol{P}_k^ oldsymbol{K}_koldsymbol{H}_koldsymbol{P}_k^-$

Introduction	Factorizat
000	000000

tion-based Filtering

Modeling Results

Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - \circ symmetric
 - \circ positive definite
- A proper filtering recursion should always maintain these properties.
- For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

Introduction	Factoriza
000	00000

ntion-based Filtering Modeling

Results 0000000000

Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - \circ symmetric
 - \circ positive definite
- A proper filtering recursion should always maintain these properties.
- For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

Symmetry: Propagation

Given that
$$oldsymbol{P}^+_{k-1} = (oldsymbol{P}^+_{k-1})^T$$
, it is clear from

$$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1}$$

that the propagated covariance matrix is algebraically symmetric.

Introduction	Factoriza
000	00000

ation-based Filtering O

Modeling Results

Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - \circ symmetric 🗸
 - \circ positive definite
- A proper filtering recursion should always maintain these properties.
- For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

Symmetry: Propagation

Given that
$$oldsymbol{P}^+_{k-1} = (oldsymbol{P}^+_{k-1})^T$$
, it is clear from

$$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1}$$

that the propagated covariance matrix is algebraically symmetric.

Introduction	Factorizat
000	00000

ation-based Filtering

Modeling Results 0000 00000000 Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - \circ symmetric 🗸
 - \circ positive definite
- A proper filtering recursion should always maintain these properties.
- For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

Symmetry: Update

Given that $oldsymbol{P}_k^- = (oldsymbol{P}_k^-)^T$, the update is given by

$$egin{aligned} &oldsymbol{P}_k^+ = oldsymbol{P}_k^- - oldsymbol{K}_koldsymbol{H}_koldsymbol{P}_k^- \ &= oldsymbol{P}_k^- - oldsymbol{P}_k^-oldsymbol{H}_koldsymbol{P}_k^-oldsymbol{H}_k^T + oldsymbol{R}_k]^{-1}oldsymbol{H}_koldsymbol{P}_k^- \ &= oldsymbol{P}_k^- - oldsymbol{P}_k^-oldsymbol{H}_koldsymbol{P}_k^- oldsymbol{H}_koldsymbol{P}_k^- \ &= oldsymbol{P}_k^- - oldsymbol{P}_k^-oldsymbol{H}_koldsymbol{P}_k^- oldsymbol{H}_koldsymbol{P}_k^- \ &= oldsymbol{P}_k^- - oldsymbol{P}_k^-oldsymbol{H}_koldsymbol{P}_k^- oldsymbol{H}_koldsymbol{P}_k^- oldsymbol{H}_koldsymbol{P}_k^- \ &= oldsymbol{P}_k^- - oldsymbol{P}_k^-oldsymbol{H}_koldsymbol{P}_k^- oldsymbol{H}_k^- oldsymbol$$

Therefore, the updated covariance matrix is algebraically symmetric.

Introduction	Factoria
000	00000

zation-based Filtering

Modeling Results

Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - $\circ\,$ symmetric $\checkmark\checkmark$
 - \circ positive definite
- A proper filtering recursion should always maintain these properties.
- For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

Symmetry: Update

Given that $oldsymbol{P}_k^- = (oldsymbol{P}_k^-)^T$, the update is given by

$$egin{aligned} &m{P}_k^+ = m{P}_k^- - m{K}_km{H}_km{P}_k^- \ &= m{P}_k^- - m{P}_k^-m{H}_k^T[m{H}_km{P}_k^-m{H}_k^T + m{R}_k]^{-1}m{H}_km{P}_k^- \end{aligned}$$

Therefore, the updated covariance matrix is algebraically symmetric.

Introduction	Factoriza
000	00000

ation-based Filtering >

Modeling Results 0000 0000000 Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - $\circ\,$ symmetric $\checkmark\checkmark$
 - \circ positive definite
- A proper filtering recursion should always maintain these properties.
- For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

Symmetry: General Comment

In the worst case, brute-force symmetrization can be used:

$$\boldsymbol{P}_k = \frac{1}{2}(\boldsymbol{P}_k + \boldsymbol{P}_k^T)$$

Introduction	Factorizat
000	00000

tion-based Filtering

Modeling Results

Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - $\circ\,$ symmetric $\checkmark\checkmark$
 - \circ positive definite
- A proper filtering recursion should always maintain these properties.
- For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

Positive Definiteness: Propagation

Given that $P_{k-1}^+ > 0$ and that F_{k-1} is full rank, the noise-free propagation of covariance is guaranteed to be positive definite; therefore,

Introduction	Factoriz
000	0000

ation-based Filtering 0 Results 000000000 Conclusions



Covariance Constraints

• By definition, the covariance matrix **must** be

Modeling

- $\circ\,$ symmetric $\checkmark\checkmark$
- \circ positive definite
- A proper filtering recursion should always maintain these properties.
- For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

Positive Definiteness: Propagation

Given that $P_{k-1}^+ > 0$ and that F_{k-1} is full rank, the noise-free propagation of covariance is guaranteed to be positive definite; therefore,

$$\underbrace{ F_{k-1} P_{k-1}^+ F_{k-1}^T }_{> \mathbf{0}}$$

Introduction	Factori
000	00000

zation-based Filtering

Modeling Results

Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - $\circ\,$ symmetric $\checkmark\checkmark$
 - \circ positive definite
- A proper filtering recursion should always maintain these properties.
- For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

Positive Definiteness: Propagation

Given that $P_{k-1}^+ > 0$ and that F_{k-1} is full rank, the noise-free propagation of covariance is guaranteed to be positive definite; therefore,

$$P_k^- = \underbrace{F_{k-1}P_{k-1}^+F_{k-1}^T}_{> oldsymbol{0}} + \underbrace{Q_{k-1}}_{\geq oldsymbol{0}}$$

Algebraically, the propagated covariance is positive definite.

McCabe, et al.

Comparison of Factorization-based Filtering for Landing Navigation

Introd	luction
000	

Modeling 0000 Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - $\circ\,$ symmetric $\checkmark\checkmark$
 - $\circ\,$ positive definite $\checkmark\,$
- A proper filtering recursion should always maintain these properties.

Results

• For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

Positive Definiteness: Propagation

Given that $P_{k-1}^+ > 0$ and that F_{k-1} is full rank, the noise-free propagation of covariance is guaranteed to be positive definite; therefore,

$$P_k^- = \underbrace{F_{k-1}P_{k-1}^+F_{k-1}^T}_{> oldsymbol{0}} + \underbrace{Q_{k-1}}_{\geq oldsymbol{0}}$$

Algebraically, the propagated covariance is positive definite.

McCabe, et al.

Comparison of Factorization-based Filtering for Landing Navigation

Introduction	F
000	0

Modeling Results

Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - $\circ\,$ symmetric $\checkmark\checkmark$
 - $\circ\,$ positive definite $\checkmark\,$
- A proper filtering recursion should always maintain these properties.
- For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

Positive Definiteness: Update

Consider the measurement update that results from

$$oldsymbol{H}_k = egin{bmatrix} 1 & 1 & 1 \ 1 & 1 & + \delta \end{bmatrix}, \quad oldsymbol{P}_k^- = oldsymbol{I}_3, \quad ext{and} \quad oldsymbol{R}_k = \delta^2 oldsymbol{I}_2$$

where $\delta^2 < \epsilon_{\text{roundoff}}$ but $\delta > \epsilon_{\text{roundoff}}$.

Introd	luction
000	

Modeling 0000 Results Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - $\circ\,$ symmetric $\checkmark\checkmark$
 - \circ positive definite \checkmark
- A proper filtering recursion should always maintain these properties.
- For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

Positive Definiteness: Update

In this case, although $oldsymbol{H}_k$ clearly has a rank of 2,

$$oldsymbol{H}_k oldsymbol{P}_k^- oldsymbol{H}_k^T + oldsymbol{R}_k = egin{bmatrix} 3 & 3+\delta \ 3+\delta & 3+2\delta \end{bmatrix}$$

with roundoff, which is a singular matrix.

Introd	luction
000	

Modeling 0000 Results Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - $\circ\,$ symmetric $\checkmark\checkmark$
 - \circ positive definite \checkmark
- A proper filtering recursion should always maintain these properties.
- For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

Positive Definiteness: Update

In this case, although $oldsymbol{H}_k$ clearly has a rank of 2,

$$oldsymbol{H}_k oldsymbol{P}_k^- oldsymbol{H}_k^T + oldsymbol{R}_k = egin{bmatrix} 3 & 3+\delta \ 3+\delta & 3+2\delta \end{bmatrix}$$

with roundoff, which is a singular matrix.

Introduction	
000	

Modeling 0000 Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - $\circ\,$ symmetric $\checkmark\checkmark$
 - \circ positive definite \checkmark
- A proper filtering recursion should always maintain these properties.

Results

• For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

- The update can fail because of numerical issues.
 - also true in propagation

Introduction	Fa
000	0

Modeling Results

Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - $\circ\,$ symmetric $\checkmark\checkmark$
 - positive definite XX
- A proper filtering recursion should always maintain these properties.
- For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

- $\circ~$ The update can fail because of numerical issues.
 - also true in propagation

Introduction	Fa
000	0

Modeling Results

Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - $\circ\,$ symmetric $\checkmark\checkmark$
 - positive definite XX
- A proper filtering recursion should always maintain these properties.
- For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

- The update can fail because of numerical issues.
 - also true in propagation
- Enforcing positive definiteness is very challenging.

Introduction	Fa
000	0

Modeling Results

Conclusions



Covariance Constraints

- By definition, the covariance matrix **must** be
 - $\circ\,$ symmetric $\checkmark\checkmark$
 - positive definite XX
- A proper filtering recursion should always maintain these properties.
- For the linear case, where the Kalman filter is theoretically exact, do these properties hold?

- $\circ~$ The update can fail because of numerical issues.
 - also true in propagation
- $\circ~$ Enforcing positive definiteness is very challenging.
- $\circ~$ Can be mitigated with factorization-based filtering methods.

Introduction 000 Factorization-based Filtering 000000

Modeling Results

Conclusions



Loss of Positive Definiteness

• Positive definiteness can be lost during filtering
Introduction Factoriza

Factorization-based Filtering 000000

Modeling Results

Conclusions



Loss of Positive Definiteness

- Positive definiteness can be lost during filtering
 - $\circ~$ Large prior uncertainty +~ precise measurements

 $\circ~$ Condition number of the covariance matrix

Factorization-based Filtering 000000

Modeling Results

Conclusions



- Positive definiteness can be lost during filtering
 - $\circ~$ Large prior uncertainty + precise measurements
 - commonly encountered in landing navigation
 - uncertainties "grow" unabated for long periods of time
 - precise data, such as altimetry, becomes available
 - $\circ~$ Condition number of the covariance matrix

Factorization-based Filtering 000000

Modeling Results

Conclusions



- Positive definiteness can be lost during filtering
 - $\circ~$ Large prior uncertainty + precise measurements
 - commonly encountered in landing navigation
 - $-\,$ uncertainties "grow" unabated for long periods of time
 - precise data, such as altimetry, becomes available
 - $\circ~$ Condition number of the covariance matrix
 - commonly encountered in large-state filters
 - estimate position, velocity, attitude, biases, etc.
 - $-\,$ units of states become important, but want to be agnostic to this

Factorization-based Filtering

Modeling Results

ts Conclusions



- Positive definiteness can be lost during filtering
 - $\circ~$ Large prior uncertainty +~ precise measurements
 - commonly encountered in landing navigation
 - $-\,$ uncertainties "grow" unabated for long periods of time
 - precise data, such as altimetry, becomes available
 - $\circ~$ Condition number of the covariance matrix
 - $-\,$ commonly encountered in large-state filters
 - estimate position, velocity, attitude, biases, etc.
 - $-\,$ units of states become important, but want to be agnostic to this
- Factorization-based filtering mitigates loss of positive definiteness
 - $\circ~$ Avoid working with covariance
 - $\circ~$ Work with factors of covariance
 - $\circ~$ Establish propagation/update equations for the factors

Factorization-based Filtering

Modeling Results

Conclusions



- Positive definiteness can be lost during filtering
 - $\circ~$ Large prior uncertainty +~ precise measurements
 - commonly encountered in landing navigation
 - $-\,$ uncertainties "grow" unabated for long periods of time
 - precise data, such as altimetry, becomes available
 - $\circ~$ Condition number of the covariance matrix
 - $-\,$ commonly encountered in large-state filters
 - estimate position, velocity, attitude, biases, etc.
 - $-\,$ units of states become important, but want to be agnostic to this
- Factorization-based filtering mitigates loss of positive definiteness
 - $\circ~$ Avoid working with covariance
 - $\circ~$ Work with factors of covariance
 - $\circ~$ Establish propagation/update equations for the factors
 - Examples:
 - UDU
 - Cholesky

Factorization-based Filtering

Modeling R 0000 C

Results Conclusions



- Positive definiteness can be lost during filtering
 - $\circ~$ Large prior uncertainty +~ precise measurements
 - commonly encountered in landing navigation
 - $-\,$ uncertainties "grow" unabated for long periods of time
 - precise data, such as altimetry, becomes available
 - $\circ~$ Condition number of the covariance matrix
 - $-\,$ commonly encountered in large-state filters
 - estimate position, velocity, attitude, biases, etc.
 - $-\,$ units of states become important, but want to be agnostic to this
- Factorization-based filtering mitigates loss of positive definiteness
 - $\circ~$ Avoid working with covariance
 - $\circ~$ Work with factors of covariance
 - $\circ~$ Establish propagation/update equations for the factors
 - Examples:
 - UDU (more details in paper)
 - ✓ Cholesky

Introduction	Factorizat
000	000000

ation-based Filtering

Modeling Results

Conclusions



- Originated with Potter's idea of the square-root filter
 - Replace covariance with Cholesky factor
 - $\circ~$ Propagate and update Cholesky factor
 - $\circ~$ No process noise + scalar measurements

Introduction	Factoriz
000	000000

ation-based Filtering O Modeling Results

Conclusions



- Originated with Potter's idea of the square-root filter
 - Replace covariance with Cholesky factor
 - $\circ~$ Propagate and update Cholesky factor
 - $\circ~$ No process noise +~ scalar measurements
- UDU Factorization
 - \circ Factor $oldsymbol{P}$ as $oldsymbol{U} oldsymbol{D} oldsymbol{U}^T$
 - $\circ~oldsymbol{U}$ is upper diagonal with ones on the diagonal
 - $\circ~oldsymbol{D}$ is diagonal
 - $\circ\,$ Propagate and update U and D

Introduction	Factoriza
000	000000

ation-based Filtering

Modeling Results

Conclusions



- Originated with Potter's idea of the square-root filter
 - Replace covariance with Cholesky factor
 - $\circ~$ Propagate and update Cholesky factor
 - $\circ~$ No process noise +~ scalar measurements
- UDU Factorization
 - \circ Factor $oldsymbol{P}$ as $oldsymbol{U} oldsymbol{D} oldsymbol{U}^T$
 - $\circ~{\boldsymbol{U}}$ is upper diagonal with ones on the diagonal
 - $\circ~oldsymbol{D}$ is diagonal
 - $\circ\,$ Propagate and update U and D
 - Modified Weighted Gram-Schmidt orthogonalization
 - Carlson rank-1 updates

Introduction	Factoriza
000	000000

ation-based Filtering > Modeling Results

Conclusions



- Originated with Potter's idea of the square-root filter
 - Replace covariance with Cholesky factor
 - $\circ~$ Propagate and update Cholesky factor
 - $\circ~$ No process noise +~ scalar measurements
- UDU Factorization
 - \circ Factor $oldsymbol{P}$ as $oldsymbol{U} oldsymbol{D} oldsymbol{U}^T$
 - $\circ~{oldsymbol U}$ is upper diagonal with ones on the diagonal
 - $\circ~oldsymbol{D}$ is diagonal
 - $\circ\,$ Propagate and update U and D
 - Modified Weighted Gram-Schmidt orthogonalization
 - Carlson rank-1 updates
- Cholesky Factorization
 - \circ Factor $oldsymbol{P}$ as $oldsymbol{S}oldsymbol{S}^T$
 - $\circ~{\boldsymbol{S}}$ is lower triangular
 - $\circ\,$ Propagate and update $oldsymbol{S}$

Introduction	Factoriza
000	000000

zation-based Filtering

Modeling Results

Conclusions



- Originated with Potter's idea of the square-root filter
 - Replace covariance with Cholesky factor
 - $\circ~$ Propagate and update Cholesky factor
 - $\circ~$ No process noise +~ scalar measurements
- UDU Factorization
 - \circ Factor $oldsymbol{P}$ as $oldsymbol{U} oldsymbol{D} oldsymbol{U}^T$
 - $\circ~{oldsymbol U}$ is upper diagonal with ones on the diagonal
 - $\circ~oldsymbol{D}$ is diagonal
 - $\circ\,$ Propagate and update U and D
 - Modified Weighted Gram-Schmidt orthogonalization
 - Carlson rank-1 updates
- Cholesky Factorization
 - \circ Factor $oldsymbol{P}$ as $oldsymbol{S}oldsymbol{S}^T$
 - $\circ \ oldsymbol{S}$ is lower triangular
 - $\circ\,$ Propagate and update S
 - QR decomposition
 - Cholesky rank-m downdate

Introduction	Factorizatio
000	000000

ation-based Filtering

Modeling Results

Conclusions



The Cholesky Square-Root Filter

• For the nonlinear state-space model

$$egin{aligned} oldsymbol{x}_k &= oldsymbol{f}(oldsymbol{x}_{k-1}) + oldsymbol{w}_{k-1} \ oldsymbol{z}_k &= oldsymbol{h}(oldsymbol{x}_k) + oldsymbol{v}_k \end{aligned}$$

Introduction	Factor
000	0000

rization-based Filtering ●○ Modeling Results

Conclusions



The Cholesky Square-Root Filter

• For the nonlinear state-space model

$$egin{aligned} oldsymbol{x}_k &= oldsymbol{f}(oldsymbol{x}_{k-1}) + oldsymbol{w}_{k-1} \ oldsymbol{z}_k &= oldsymbol{h}(oldsymbol{x}_k) + oldsymbol{v}_k \end{aligned}$$

• The Cholesky square-root filter is given by the recursion:

4

Mean Prop.	$\boldsymbol{m}_k^- = \boldsymbol{f}(\boldsymbol{m}_{k-1}^+)$
SRF Prop.	$oldsymbol{S}_k^- = \mathrm{qr}\{[oldsymbol{F}_{k-1}oldsymbol{S}_{k-1}^+ \mid oldsymbol{T}_{k-1}]^T\}^T$
Innov. SRF	$oldsymbol{Y}_k = \mathrm{qr}\{[oldsymbol{H}_koldsymbol{S}_k^-\midoldsymbol{L}_k]^T\}^T$
Cross Cov.	$oldsymbol{C}_k = oldsymbol{S}_k^{-}ig[oldsymbol{H}_koldsymbol{S}_k^{-}ig]^T$
Update Factors	$oldsymbol{U}_k = oldsymbol{C}_k (oldsymbol{Y}_k^-)^T$
Kalman Gain	$\boldsymbol{K}_k = \boldsymbol{U}_k \boldsymbol{Y}_k^{-1}$
Mean Update	$oldsymbol{m}_k^+=oldsymbol{m}_k^-+oldsymbol{K}_k(oldsymbol{z}_k-oldsymbol{h}(oldsymbol{m}_k^-))$
SRF Update	$\boldsymbol{S}_{k}^{+} = \text{cholupdate}\{(\boldsymbol{S}_{k}^{-})^{T}, \boldsymbol{U}_{k}, -1\}^{T}$

Introduction Factorization-based Filtering 00000

Filtering Modeling

Results Conclusions

MISSOURI SET

Comments on the Methods

• Cholesky Factorization

- $\circ~$ guarantees symmetry of the covariance matrix
- can guarantee positive definiteness
- requires square root operations
- $\circ~$ quite simple, structurally

Introduction Factorization-based Filtering

ng Modeling 0000 Results Conclusions



Comments on the Methods

Cholesky Factorization

- $\circ~$ guarantees symmetry of the covariance matrix
- can guarantee positive definiteness
- requires square root operations
- quite simple, structurally

• UDU Factorization

- o guarantees symmetry of the covariance matrix
- $\circ~$ simple check for positive definiteness
- $\circ~$ does not require square root operations
- $\circ~$ more complicated, structurally

Introduction	Factorization-based	Filterin
000	000000	

ring Modeling ●000 Results Co

Conclusions



Trajectory



ntroduction	Factorization-base
000	000000

IMU Model

ed Filtering Modeling 0●00 Results Conclusions



• Inertial Measurement Unit output is given by

$$\Delta oldsymbol{v}_{m,k} = \Delta oldsymbol{v}_k + oldsymbol{b}_v + oldsymbol{w}_{v,k} \ \Delta oldsymbol{ heta}_{m,k} = \Delta oldsymbol{ heta}_k + oldsymbol{b}_ heta + oldsymbol{w}_{ heta,k}$$

where

- $\circ~\Delta \pmb{v}_k$ is the true, integrated, non-gravitational acceleration
- $\circ~\Delta \pmb{\theta}_k$ is the true, integrated angular velocity
- Sensor specifications
 - Accelerometer
 - Bias $(1\sigma) = 300\mu g$
 - \circ Noise (1 σ) = $35 \mu g / \sqrt{\mathrm{Hz}}$
 - \circ Frequency = 40 Hz
 - $\circ~$ Active: always

- $\frac{\text{Gyro}}{\text{Bias}}(1\sigma) = 1^{\circ}/\text{hr}$
- Noise $(1\sigma) = 0.07^{\circ}/\sqrt{\mathrm{hr}}$
- \circ Frequency = 40 Hz
- Active: always

ntroduction	Factorization-based	Fil
000	000000	

Modeling

Conclusions



Altimeter Model

• Spherical Altitude measurement is given by

tering

$$z_k = (\|\boldsymbol{r}_{\text{alt},k}^i\| - r_{\text{sph}}) + b_{\text{alt}} + v_{\text{alt},k}$$

Results

where

$$m{r}_{\mathrm{alt},k}^i = m{r}_{\mathrm{imu},k}^i + m{T}_{c,k}^i m{r}_{\mathrm{alt/imu}}^c$$

- Sensor specifications
 - Bias $(1\sigma) = 0.5 \text{ m}$
 - Noise $(1\sigma) = [500, 5] m$
 - \circ Frequency = 10 Hz
 - $\circ~$ Active: $h \leq 15~{\rm km}$

Star Camera Model

Factorization-based Filtering

• Quaternion Star Camera measurement is given by

Modeling

000

$$ar{oldsymbol{z}}_k = ar{oldsymbol{q}}_{ ext{err},k} \otimes ar{oldsymbol{q}}_c^{ ext{sc}} \otimes ar{oldsymbol{q}}_{i,k}^c$$

Results

Conclusions

where

Introduction

$$\bar{\boldsymbol{q}}_{\text{err},k} = \begin{bmatrix} \sin\left(\frac{1}{2} \|\boldsymbol{\theta}_{\text{err},k}\|\right) \frac{\boldsymbol{\theta}_{\text{err},k}}{\|\boldsymbol{\theta}_{\text{err},k}\|} \\ \cos\left(\frac{1}{2} \|\boldsymbol{\theta}_{\text{err},k}\|\right) \end{bmatrix} \quad \text{and} \quad \boldsymbol{\theta}_{\text{err},k} = \boldsymbol{b}_{sc} + \boldsymbol{v}_{sc,k}$$

- Sensor Specifications
 - Bias $(1\sigma) = 10''$
 - Noise $(1\sigma) = 30''$
 - $\circ \ \ \text{Frequency} = 1 \ Hz$
 - Active: when not thrusting



Factorization-based Filtering

Modeling Results

Conclusions



Monte Carlo Comparison

- Assess statistical consistency
 - $\circ~$ 1000 Monte Carlo trials
 - Resample initial states and noises
 - Compute sample covariance
 - Compare to single run performance
 - $\circ~$ Look at full covariance, UDU factorized, and Cholesky factorized filters

Introduction	Factorization-based	F
000	000000	

ed Filtering Modeling 0000 Results o●oooooooo

Conclusions



Monte Carlo: Position



Introduction Factorization-based Filtering

Filtering Modeling 0000 Results Co

Conclusions



Monte Carlo: Velocity



ntroduction	Factorization-based	Filterir
000	000000	

Modeling 0000

g

Results Conclusions

MISSOURI SET

Monte Carlo: Attitude



Introduction Factorization

Factorization-based Filtering

Modeling Results

Results Conclusions



Monte Carlo: Accel. Bias



Factorization-based Filtering

Modeling Results

Results Conclusions



Monte Carlo Comparison

- Assess statistical consistency
 - \circ 1000 Monte Carlo trials
 - Resample initial states and noises
 - Compute sample covariance
 - Compare to single run performance
 - $\circ~$ Look at full covariance, UDU factorized, and Cholesky factorized filters
- Observations
 - Some full covariance trials failed
 - All UDU and Cholesky factorized trials successful
 - $\circ~$ Translational uncertainty growth before altimeter turns on
 - $\circ~$ Rotational uncertainty growth after star camera turns off
 - errors caused by sampling

Introduction Factorizat

Factorization-based Filtering

Modeling Results

Results Conclusions



- More in-depth analysis during terminal descent
 - $\circ~$ Same simulation, same configuration
 - Enhanced view in terminal descent



Grid Comparison: Position

Modeling

Results

00000000000

Conclusions

Factorization-based Filtering



McCabe, et al.

Introduction

Comparison of Factorization-based Filtering for Landing Navigation

Factorization-based Filtering

Modeling 0000

Results Conclusions



Grid Comparison: Attitude



Introduction	Factoria
000	00000

zation-based Filtering

Modeling Results

Conclusions



- More in-depth analysis during terminal descent
 - $\circ~$ Same simulation, same configuration
 - Enhanced view in terminal descent
- Observations

Introduction Facto

Factorization-based Filtering 000000

Modeling Results

Conclusions



- More in-depth analysis during terminal descent
 - $\circ~$ Same simulation, same configuration
 - Enhanced view in terminal descent
- Observations
 - Full covariance
 - Conservative in position uncertainty
 - $\$ Overly confident in attitude uncertainty
 - Failures due to loss of positive definiteness

Introduction Factor

Factorization-based Filtering

Modeling Results

Conclusions



- More in-depth analysis during terminal descent
 - $\circ~$ Same simulation, same configuration
 - Enhanced view in terminal descent
- Observations
 - Full covariance
 - Conservative in position uncertainty
 - Overly confident in attitude uncertainty
 - Failures due to loss of positive definiteness
 - UDU factorized
 - Back and forth in position uncertainty
 - Back and forth in attitude uncertainty
 - No failures due to loss of positive definiteness

Introduction Factor

Factorization-based Filtering

Modeling Results

Conclusions



- More in-depth analysis during terminal descent
 - $\circ~$ Same simulation, same configuration
 - Enhanced view in terminal descent
- Observations
 - Full covariance
 - Conservative in position uncertainty
 - Overly confident in attitude uncertainty
 - Failures due to loss of positive definiteness
 - UDU factorized
 - Back and forth in position uncertainty
 - Back and forth in attitude uncertainty
 - No failures due to loss of positive definiteness
 - Cholesky factorized
 - Conservative in position uncertainty
 - Back and forth in attitude uncertainty
 - No failures due to loss of positive definiteness

Introduction	
000	

Modeling 0000 Conclusions



Conclusions

• Comparison of different filtering approaches for descent navigation

Introd	luction
000	

Modeling

Conclusions



Conclusions

• Comparison of different filtering approaches for descent navigation

- Full covariance
 - brute-force symmetrization
 - no guarantee on positive definiteness

Introd	luction
000	

Modeling 0000 Conclusions



Conclusions

• Comparison of different filtering approaches for descent navigation

- Full covariance
 - brute-force symmetrization
 - no guarantee on positive definiteness
- UDU factorization
 - guaranteed symmetry
 - easy check for positive definiteness

Introd	luction
000	

Modeling 0000 Conclusions



Conclusions

• Comparison of different filtering approaches for descent navigation

- Full covariance
 - brute-force symmetrization
 - no guarantee on positive definiteness
- UDU factorization
 - guaranteed symmetry
 - easy check for positive definiteness
- Cholesky factorization
 - guaranteed symmetry
 - can guarantee positive definiteness
| Introd | luction |
|--------|---------|
| 000 | |

Factorization-based Filtering

Modeling 0000 Conclusions



Conclusions

• Comparison of different filtering approaches for descent navigation

Results

- Full covariance
 - brute-force symmetrization
 - no guarantee on positive definiteness
- $\circ~$ UDU factorization
 - guaranteed symmetry
 - easy check for positive definiteness
- Cholesky factorization
 - guaranteed symmetry
 - can guarantee positive definiteness
- When processing IMU, altimeter, and star camera data
 - $\circ~$ observed failures in full covariance filters
 - $\circ~$ similar consistency performance in UDU and Cholesky

Introd	luction
000	

Factorization-based Filtering

Modeling 0000 Conclusions



Conclusions

• Comparison of different filtering approaches for descent navigation

Results

- Full covariance
 - brute-force symmetrization
 - no guarantee on positive definiteness
- $\circ~$ UDU factorization
 - guaranteed symmetry
 - easy check for positive definiteness
- Cholesky factorization
 - guaranteed symmetry
 - can guarantee positive definiteness
- When processing IMU, altimeter, and star camera data
 - $\circ~$ observed failures in full covariance filters
 - $\circ~$ similar consistency performance in UDU and Cholesky
- Which filter should you use?
 - $\circ\;$ vector vs. scalar processing of data
 - computational resources available



This work was partially supported by a NASA Space Technology Research Fellowship and through Grant NNX16AF11A.

The authors would also like to acknowledge the many helpful discussions with Drs. Chris D'Souza and Renato Zanetti of NASA Johnson Space Center.

Questions?



