
Matrix Theory for Data Association in PVS

Haley R. King¹, J. Tanner Slagel²,

¹Department of Mathematics
Texas State University
hrk9@txstate.edu

² Safety Critical Avionics Systems Branch
NASA Langley
j.tanner.slagel@nasa.gov

Aug 2021 @ NASA Langley
Intern exit presentation

Data Association

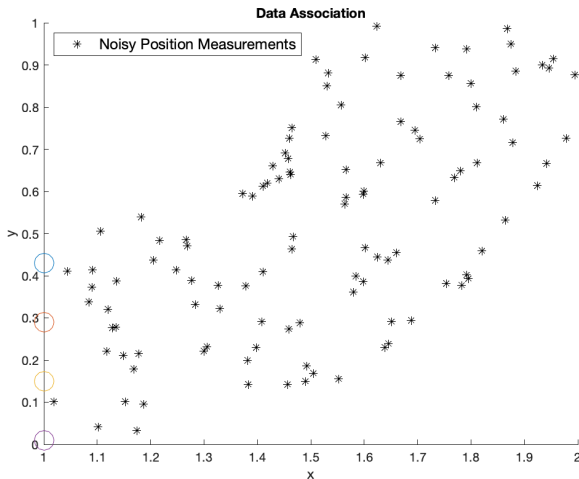
Given sets of data

$$D_k = \{d_{k,i} \mid 1 \leq i \leq N_k\}$$

associate each $d_{k,i}$ with a vehicle "track"

Example:

- Four vehicles, traveling over time
- Noisy position/velocity measurements
- Which measurements come from which vehicle?



Data Association

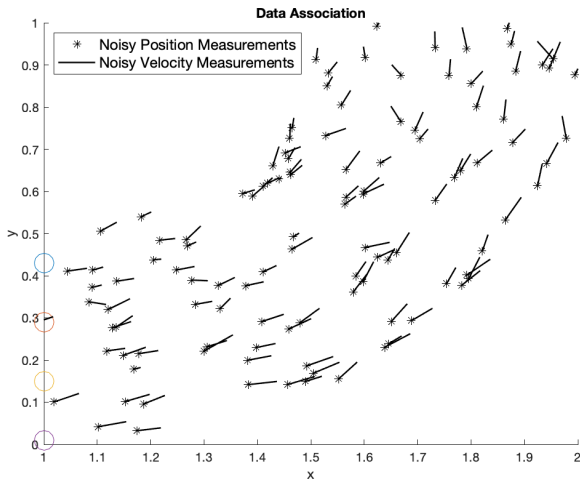
Given sets of data

$$D_k = \{d_{k,i} \mid 1 \leq i \leq N_k\}$$

associate each $d_{k,i}$ with a vehicle "track"

Example:

- Four vehicles, traveling over time
- Noisy position/velocity measurements
- Which measurements come from which vehicle?



Data Association

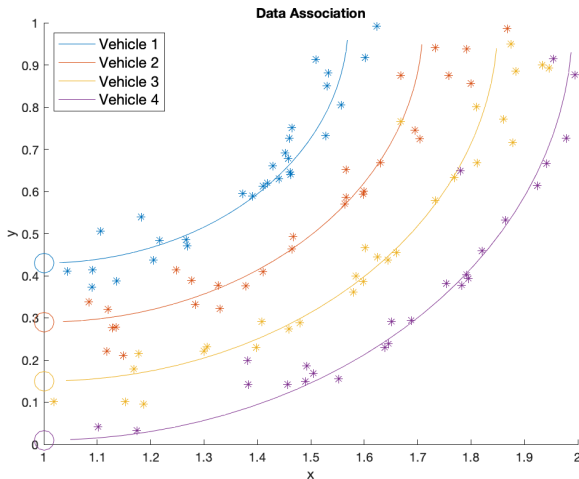
Given sets of data

$$D_k = \{d_{k,i} \mid 1 \leq i \leq N_k\}$$

associate each $d_{k,i}$ with a vehicle "track"

Example:

- Four vehicles, traveling over time
- Noisy position/velocity measurements
- Which measurements come from which vehicle?



Data Association

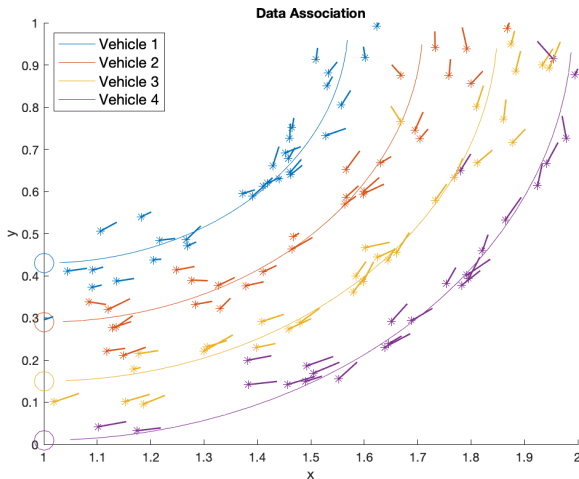
Given sets of data

$$D_k = \{d_{k,i} \mid 1 \leq i \leq N_k\}$$

associate each $d_{k,i}$ with a vehicle "track"

Example:

- Four vehicles, traveling over time
- Noisy position/velocity measurements
- Which measurements come from which vehicle?



Data Association

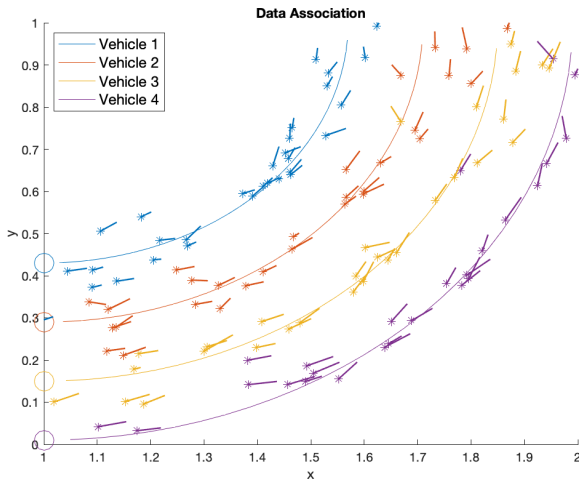
Given sets of data

$$D_k = \{d_{k,i} \mid 1 \leq i \leq N_k\}$$

associate each $d_{k,i}$ with a vehicle "track"

Example:

- Four vehicles, traveling over time
- Noisy position/velocity measurements
- Which measurements come from which vehicle?



Goal: Formal verification of data association algorithm and related theory in PVS

Formalization of Data Association

One approach:

1. For $x_k \in \mathbb{R}^n$, $P_k \in \mathbb{R}^{n \times n}$, state and covariance approximation, estimate next state:

$$\hat{x}_{k+1} = F * x_k + B_k * u_k$$

$$\hat{P}_{k+1} = F * P_k F^\top + Q_k$$

2. Find "closest" data point to $\hat{x}_{k+1}$ using Mahalanobis distance

$$d_{k+1} = \arg \min_i ||\hat{x}_k - d_{i,k+1}||_{\hat{P}_{k+1}^{-1}}$$

3. Use $\hat{x}_{k+1}$ and d_{k+1} to get x_{k+1} using Kalman filter

Formalization of Data Association

One approach:

1. For $x_k \in \mathbb{R}^n$, $P_k \in \mathbb{R}^{n \times n}$, state and covariance approximation, estimate next state:

$$\hat{x}_{k+1} = F * x_k + B_k * u_k$$

$$\hat{P}_{k+1} = F * P_k F^\top + Q_k$$

2. Find "closest" data point to $\hat{x}_{k+1}$ using Mahalanobis distance

$$d_{k+1} = \arg \min_i \|\hat{x}_k - d_{i,k+1}\|_{\hat{P}_{k+1}^{-1}}$$

3. Use $\hat{x}_{k+1}$ and d_{k+1} to get x_{k+1} using Kalman filter

Formalization requires:

- Kalman filters
 - Executable implementation
 - Properties of Matrix/Vector Arithmetic
 - Matrix Inverse formulas
 - Sherman-Woodbury
 - Sherman-Morrison
 - Statistical theory (derivation of Kalman Filters)
 - Outer product of vectors
- Mahalanobis distance
 - Covariance/Positive definite matrices
 - Symmetric Matrices
 - Matrix weighted inner-products

Formalization of Data Association

One approach:

1. For $x_k \in \mathbb{R}^n$, $P_k \in \mathbb{R}^{n \times n}$, state and covariance approximation, estimate next state:

$$\hat{x}_{k+1} = F * x_k + B_k * u_k$$

$$\hat{P}_{k+1} = F * P_k F^\top + Q_k$$

2. Find "closest" data point to $\hat{x}_{k+1}$ using Mahalanobis distance

$$d_{k+1} = \arg \min_i \|\hat{x}_k - d_{i,k+1}\|_{\hat{P}_{k+1}^{-1}}$$

3. Use $\hat{x}_{k+1}$ and d_{k+1} to get x_{k+1} using Kalman filter

Formalization requires:

- Kalman filters
 - Executable implementation
 - Properties of Matrix/Vector Arithmetic
 - Matrix Inverse formulas
 - Sherman-Woodbury
 - Sherman-Morrison
 - Statistical theory (derivation of Kalman Filters)
 - Outer product of vectors
- Mahalanobis distance
 - Covariance/Positive definite matrices
 - Symmetric Matrices
 - Matrix weighted inner-products

Outline

- Introduction to Formalization to Data Association
 - Mahalanobis Distance
 - Kalman Filters
 - Numerical Validation of Kalman Filter
 - Conclusions and Future Work
-

Mahalanobis Distance

The Mahalanobis distance of an observation $\vec{x} \in \mathbb{R}^n$ from a set observations with mean $\vec{\mu}$ and covariance matrix $S \in \mathbb{R}^{n \times n}$ is defined as

$$\begin{aligned} D_{S,\mu}(\vec{x}) &= \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})} \\ &= \|\vec{x} - \vec{\mu}\|_{S^{-1}} \end{aligned}$$

- In the directions that have larger variance, the distance is weighted less heavily. In the directions that have smaller variance the distances are weighted more heavily.

Mahalanobis Distance Example

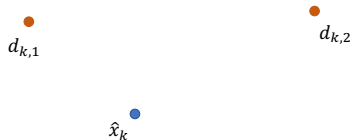
$\hat{x}_k \in \mathbb{R}^2$: Model prediction of location of vehicle x at time k

$\hat{C}_k \in \mathbb{R}^{2 \times 2}$: Model prediction of covariance of $\hat{x}_k$ at time

$d_{k,1}, d_{k,2} \in \mathbb{R}^2$: Sensor location data, one comes from x , one doesn't

Idea #1, use Euclidian distance:

$$\operatorname{argmin}_i \|\hat{x}_k - d_{k,i}\|_2 = d_{k,1}$$



Mahalanobis Distance Example

$\hat{x}_k \in \mathbb{R}^2$: Model prediction of location of vehicle x at time k

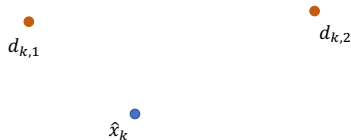
$\hat{C}_k \in \mathbb{R}^{2 \times 2}$: Model prediction of covariance of $\hat{x}_k$ at time

$d_{k,1}, d_{k,2} \in \mathbb{R}^2$: Sensor location data, one comes from x , one doesn't

Idea #1, use Euclidian distance:

$$\operatorname{argmin}_i \|\hat{x}_k - d_{k,i}\|_2 = d_{k,1}$$

Does not take into account uncertainty in predicted $\hat{x}_k$.



Mahalanobis Distance Example

$\hat{x}_k \in \mathbb{R}^2$: Model prediction of location of vehicle x at time k

$\hat{C}_k \in \mathbb{R}^{2 \times 2}$: Model prediction of covariance of $\hat{x}_k$ at time

$d_{k,1}, d_{k,2} \in \mathbb{R}^2$: Sensor location data, one comes from x , one doesn't

Idea #1, use Euclidian distance:

$$\operatorname{argmin}_i \|\hat{x}_k - d_{k,i}\|_2 = d_{k,1}$$

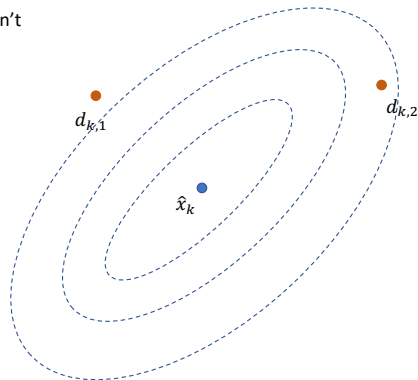
Does not take into account uncertainty in predicted $\hat{x}_k$.

Idea #2, use Mahalanobis distance:

$$\operatorname{argmin}_i \|\hat{x}_k - d_{k,i}\|_{\hat{C}_k^{-1}} = d_{k,2}, \text{ where:}$$

$$\hat{C}_k = \begin{bmatrix} v_1 & v_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} v_1 & v_2 \end{bmatrix}^\top$$

Considering uncertainty in $\hat{x}_k$, $d_{k,2}$ is the more likely data point



Mahalanobis Distance Example

$\hat{x}_k \in \mathbb{R}^2$: Model prediction of location of vehicle x at time k

$\hat{C}_k \in \mathbb{R}^{2 \times 2}$: Model prediction of covariance of $\hat{x}_k$ at time

$d_{k,1}, d_{k,2} \in \mathbb{R}^2$: Sensor location data, one comes from x , one doesn't

Idea #1, use Euclidian distance:

$$\operatorname{argmin}_i \|\hat{x}_k - d_{k,i}\|_2 = d_{k,1}$$

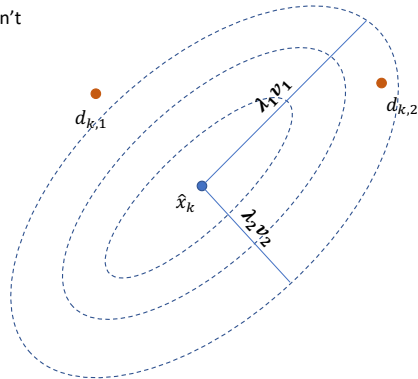
Does not take into account uncertainty in predicted $\hat{x}_k$.

Idea #2, use Mahalanobis distance:

$$\operatorname{argmin}_i \|\hat{x}_k - d_{k,i}\|_{\hat{C}_k^{-1}} = d_{k,2}, \text{ where:}$$

$$\hat{C}_k = \begin{bmatrix} v_1 & v_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} v_1 & v_2 \end{bmatrix}^\top$$

Considering uncertainty in $\hat{x}_k$, $d_{k,2}$ is the more likely data point



Mahalanobis Distance in PVS

```
57 % Find closest point in data set "D"
58 % Using Mahalanobis distance with mu,invCov
59 % m is number of points, n is dimension of points
60 closest_point(m:posnat,n:nat,invCov:(pos_defn?(n)),mu:VectorN(n))
61 (D:MatrixMN(m,n)): VectorN(n) =
62 argmin(n)(D, Mah_dist_sq(n,invCov, mu))
63
64 %Example
    prove | status-proofchain | show-prooflite
65 ex_closest_point: LEMMA
66 LET
67 %Most noise is in [1,1] direction
68 invCov = inverse((: (0.0550, 0.0450 :), (:0.0450,0.0550 :) :)),
69 mu = (: 1, 1 :),
70 D = [(: (:2,2:), (: 2, 0 :) :)]
71 IN
72 closest_point(2,2,invCov,mu)(D) = (: 2,2 :)
73
```

Kalman Filter

Given

- $F_k \in \mathbb{R}^{n \times n}$: Forward Model
- $B_k \in \mathbb{R}^{n \times n}$, $u_k \in \mathbb{R}^n$: Control model/input
- $Q_k \in \mathbb{R}^{n \times n}$: Covariance of model
- $H_k \in \mathbb{R}^{p \times n}$ state to observation space model
- $z_k \in \mathbb{R}^p$ sensor measurement
- $R_k \in \mathbb{R}^{p \times p}$: Covariance of sensor noise

Predict next state:

- $\hat{x}_k = F_k x_{k-1} + B_k U_k$
- $\hat{P}_k = F_k P_{k-1} F_k^T + Q_k$

Update prediction: Using data z_k , arrive at 'optimal' state approximation:

$$x_k = \hat{x}_k + K_k \tilde{y}_k$$

$$P_k = (I_n - K_k H_k) \hat{P}_{k-1}$$

where

- $K_k = \hat{P}_k H_k^T S_k^{-1}$
- $S_k = H_k \hat{P}_k H_k^T + R_k$

Kalman Filter

Given

- $F_k \in \mathbb{R}^{n \times n}$: Forward Model
- $B_k \in \mathbb{R}^{n \times n}$, $u_k \in \mathbb{R}^n$: Control model/input
- $Q_k \in \mathbb{R}^{n \times n}$: Covariance of model
- $H_k \in \mathbb{R}^{p \times n}$ state to observation space model
- $z_k \in \mathbb{R}^p$ sensor measurement
- $R_k \in \mathbb{R}^{p \times p}$: Covariance of sensor noise

Predict next state:

- $\hat{x}_k = F_k x_{k-1} + B_k U_k$
- $\hat{P}_k = F_k P_{k-1} F_k^T + Q_k$

► Kalman filters in PVS

Update prediction: Using data z_k , arrive at 'optimal' state approximation:

$$x_k = \hat{x}_k + K_k \tilde{y}_k$$

$$P_k = (I_n - K_k H_k) \hat{P}_{k-1}$$

where

- $K_k = \hat{P}_k H_k^T S_k^{-1}$
- $S_k = H_k \hat{P}_k H_k^T + R_k$

Kalman Filter Example

- Want positions and velocities

$$x_k = \begin{bmatrix} p_x \\ p_y \\ v_x \\ v_y \end{bmatrix} \in \mathbb{R}^4$$

for vehicle with true path in figure

- Forward model

$$F = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Covariance of model

$$Q = .1I_4,$$

- No control inputs $B_k = \mathbf{0}_{4 \times 4}$
- State space is observation space, i.e. $H_k = I_4$
- Covariance of noise

$$R = W \begin{bmatrix} .1 & 0 & 0 & 0 \\ 0 & .01 & 0 & 0 \\ 0 & 0 & .01 & 0 \\ 0 & 0 & 0 & .01 \end{bmatrix} W^\top$$

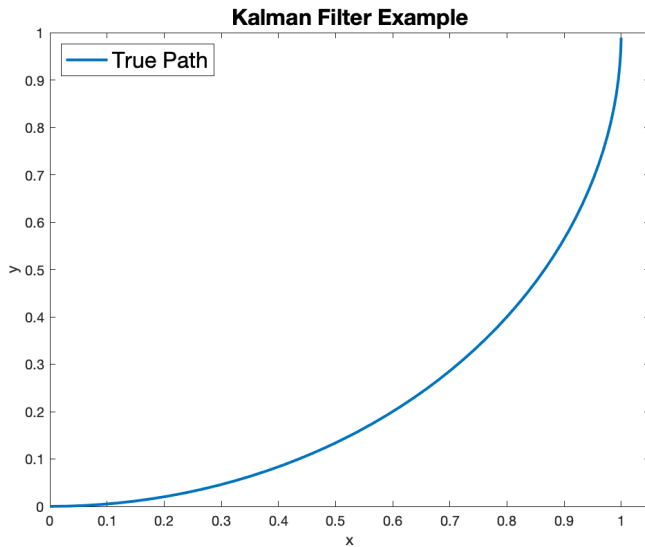
where

$$W = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

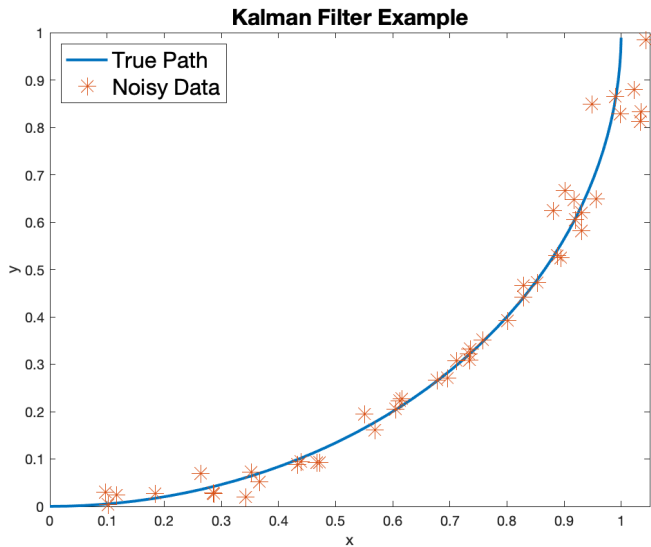
Note: Most noise uncertainty is in the

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \end{bmatrix}^\top \text{ direction}$$

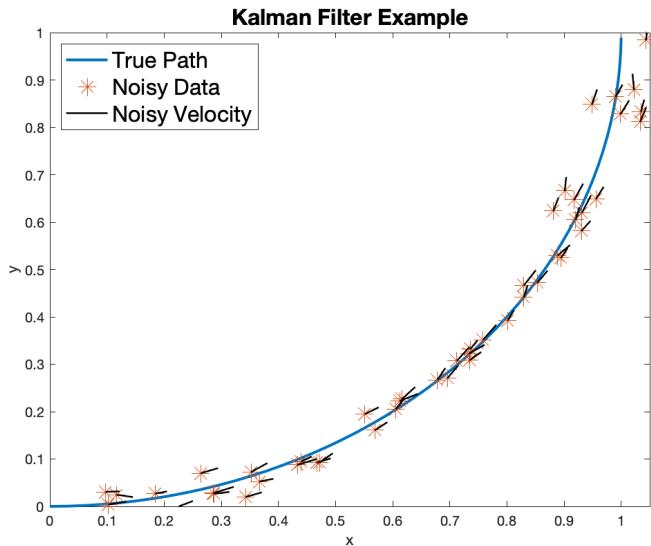
Kalman Filter Example



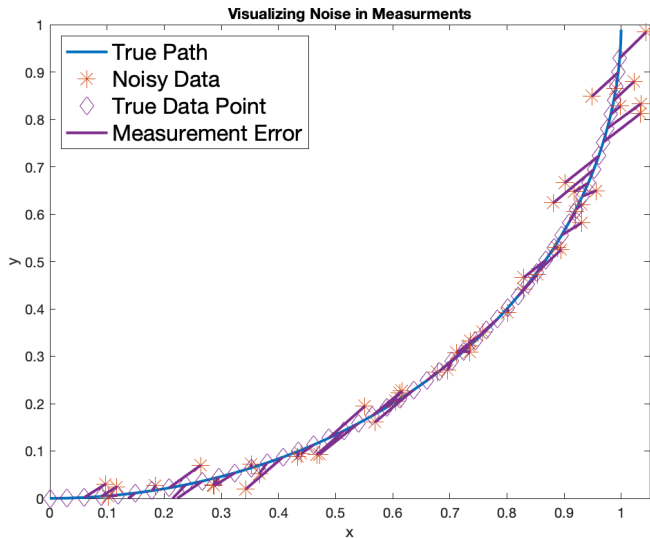
Kalman Filter Example



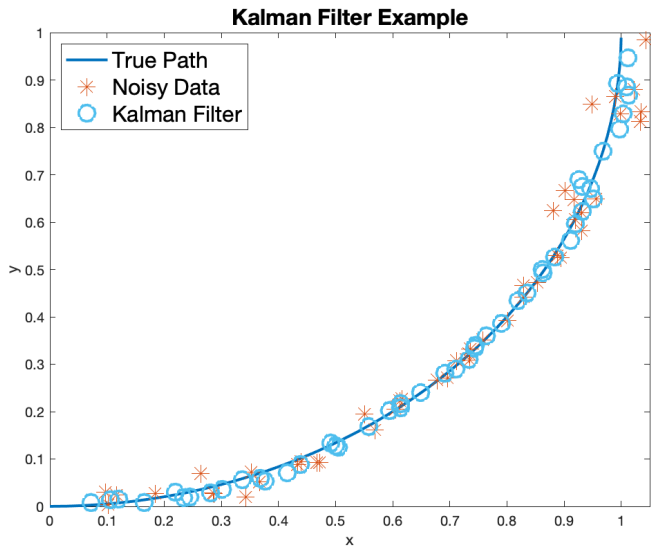
Kalman Filter Example



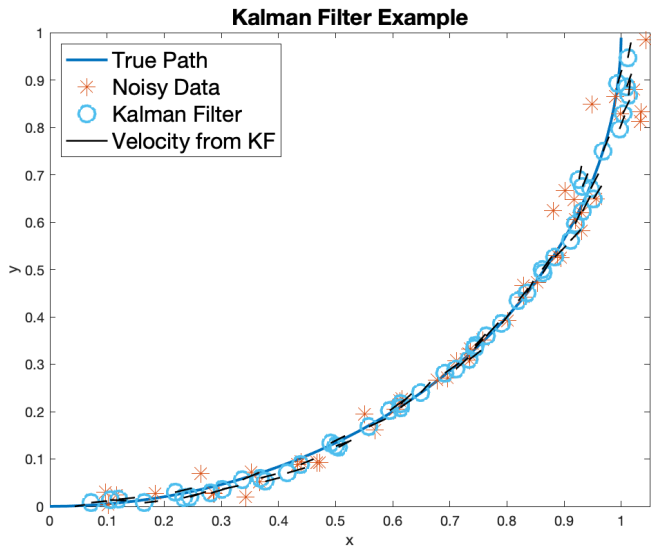
Kalman Filter Example



Kalman Filter Example



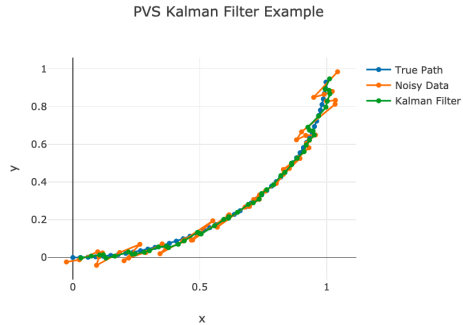
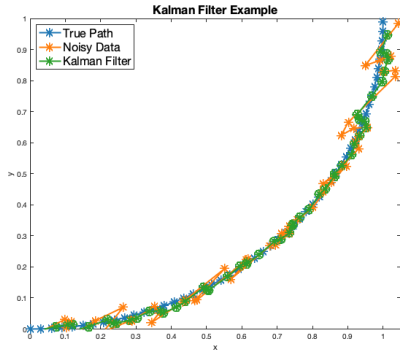
Kalman Filter Example



Specification of Kalman filters in PVS

```
52  %% One Step of the Kalman Filter
53  Kal_step(p,n:posnat)(Fk,Bk,Qk:SquareMatrix(n),uk:VectorN(n),
54      Hk:MatrixMN(p,n),Rk:SquareMatrix(p),zk:VectorN(p))
55      (xkm1:VectorN(n),Pkm1:SquareMatrix(n)): Kal_Est(n) =
56      %Model Prediction
57      LET hat_xk = predict_state(n)(Fk,Bk,uk)(xkm1),
58      %Cov Prediction
59      hat_Pk = predict_cov(n)(Fk,Qk)(Pkm1),
60      %Residual
61      yk      = residual(p,n)(Hk,zk)(hat_xk),
62      Sk      = prefit_cov(p,n)(Hk,Rk)(hat_Pk),
63      %Kalman Gain
64      Kk      = Kal_gain(p,n)(Hk)(hat_Pk,inv_safe(Sk,Ifun(p)))
65      IN
66      (# xk := hat_xk + Kk*yk, Pk := est_cov(p,n)(Kk,Hk)(hat_Pk) #)
67      %% State Estimation    %% Covariance Estimation
```

Kalman Filter Results Matlab and PVS



Kalman Filter Results Matlab and PVS

Implemented Kalman filter example in PVS:

```
err_Meas =
```

```
0.0657
```

```
err_Kal =
```

```
0.0270
```

PVSio Evaluator

Usage:

- Enter a PVS expression
- or
- Enter a Lisp expression

```
<PVSio> pretty(err_Meas);
```

```
==>
```

```
0.0657
```

```
<PVSio> pretty(err_Kal);
```

```
==>
```

```
0.027
```

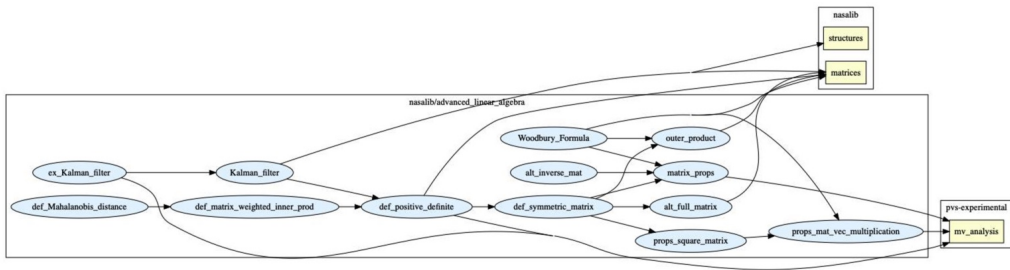
Conclusion and Future work

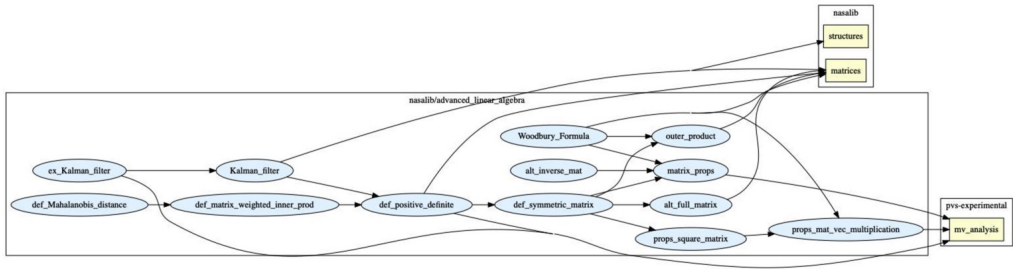
Conclusions

- Started a PVS library to support formal verification and validation of Data Association algorithms
- Specified kalman filters in PVS, compared numerical experiment with MATLAB
- Developed Mahalanobis distance and other topics from linear Algebra

Future work

- Develop theory of Kalman filters in PVS
- Implement extended (nonlinear) Kalman filters
- Implementing Matrix inversion formulas into the Kalman filter process (where appropriate)
- Automatically reading .csv files into PVSio (in progress, Mariano)





Thank you for listening! Questions?

Temporary page!

L^AT_EX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because L^AT_EX now knows how many pages to expect for this document.