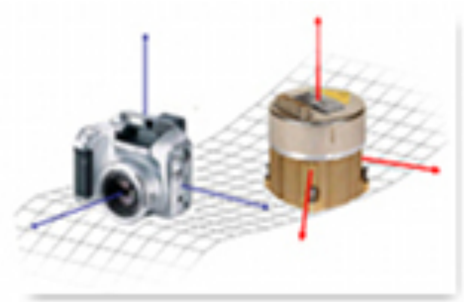


Sensor Orientation Extended Kalman Filter

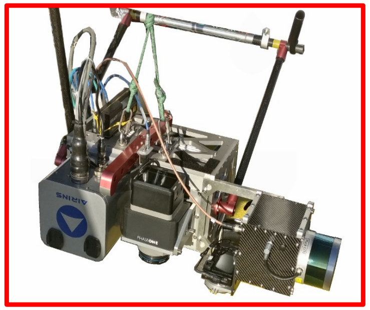
Jan SKALOUD

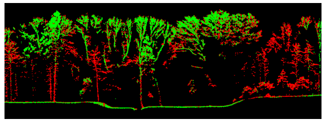
EPFL, spring semester 2024

École polytechnique fédérale de Lausanne

Sensor orientation – main topics

This translates into three rough big areas





1. Fundamentals
   - How to characterize sensor noise
   - How to transform from the sensed signals to navigation frame?

2. Position, velocity, attitude (navigation)
   - How to formulate navigation equation in different frames?
   - How to resolve them numerically?

3. Sensor fusion
   - How to formulate models for sensor fusion?
   - How to implement it in optimization and use it for mapping?

You need the frames

You need the navigation quantities and the noise properties

Cockpit view of SO course's topics

How to reach *integrated* sensor orientation?

- Sensor Fusion
  - Sensors (inertial + GPS)
    - Noise characterization
    - Stochastic Processes
  - GPS – Polycopié, read Ch.8
  - <u>Extended</u> Kalman Filter
    - Strapdown Inertial Navigation
      - Navigation Frames
    - Estimation principle - RLS

Sensor fusion – agenda

Kalman filter – base (Week 9)

- Intuitive approach
- Discrete KF – components, steps, implementation (Lab 5)

Kalman filter – extension (Week 10)

- **Computation of transition and process noise matrices $\mathbf{\Phi}_k,\ \mathbf{Q}_k$**
- Linearized & Extended Kalman filter
- Some other 'motion model' examples

INS/GPS integration (Week 11)

- Theory of a differential filter
- Practice – derivation & implementation (Lab 6)

Sensor orientation (Week 12)

- Direct & integrated orientation of optical sensors

Numerical evaluation of $\Phi_{t_0,t}$ $\mathbf{Q}_{t_0,t}$

Preamble - only for information (can be “safely” skipped) :

- Derivation & proof of :

1) Why?

$$\boxed{\mathbf{F}(t) = \frac{\partial \mathbf{\Phi}(t_0,t)}{\partial t}}$$

2) Why?

$$\boxed{\Phi_{t_0,t} = e^{\mathbf{F}(t-t_0)}}$$

(1) Relation $\Phi(t_0, t) \longleftrightarrow \mathbf{F}(t)$

Process model is expressed as a linear system of homogenous differential equation (continuous form): (1) $\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t)$

Its general solution is: (2) $\mathbf{x}(t) = \mathbf{\Phi}(t_0, t)\mathbf{x}(t_0)$

Relation:

I. Take the derivative of the solution: $$\frac{\partial \mathbf{x}(t)}{\partial t} = \frac{\partial \mathbf{\Phi}(t_0,t)}{\partial t}\mathbf{x}(t_0)$$

II. Substitute it to Eq. (1) on its left-side, & on the right-side for $\mathbf{x}(t) \rightarrow$ Eq.(2):

$$\underbrace{\frac{\partial \mathbf{\Phi}(t_0, t)}{\partial t}\mathbf{x}(t_0)}_{\dot{\mathbf{x}}(t)} = \mathbf{F}(t)\,\underbrace{\mathbf{\Phi}(t_0, t)\mathbf{x}(t_0)}_{\mathbf{x}(t)}$$

$$: \Longrightarrow \boxed{\mathbf{F}(t) = \frac{\partial \mathbf{\Phi}(t_0,t)}{\partial t}}$$

Notation: $\mathbf{\Phi}(t_0, t) = \mathbf{\Phi}_{t_0,t}$ Note: in a stationary system this matrix is time invariant!

(2) Proof $\Phi_{t_0,t} = e^{\mathbf{F}(t-t_0)}$

Taylor expansion for $\mathbf{x}(t)$:

$$\mathbf{x}(t) = \mathbf{x}(t_0) + \dot{\mathbf{x}}(t_0)(t-t_0) + \ddot{\mathbf{x}}(t_0)\frac{(t-t_0)^2}{2!} + \dddot{\mathbf{x}}(t_0)\frac{(t-t_0)^3}{3!} + \ldots$$

Considering that:

$$\dot{\mathbf{x}}(t_0) = \mathbf{F}\mathbf{x}(t_0)$$
$$\ddot{\mathbf{x}}(t_0) = \mathbf{F}\dot{\mathbf{x}}(t_0) = \mathbf{F}\mathbf{F}\mathbf{x}(t_0) = \mathbf{F}^2\mathbf{x}(t_0)$$
$$\dddot{\mathbf{x}}(t_0) = \mathbf{F}\ddot{\mathbf{x}}(t_0) = \mathbf{F}^3\mathbf{x}(t_0)$$

Substituting for derivatives in the Taylor expansion above:

$$\begin{aligned}\mathbf{x}(t) &= \mathbf{x}(t_0) + \mathbf{F}\mathbf{x}(t_0)(t-t_0) + \mathbf{F}^2\mathbf{x}(t_0)\frac{(t-t_0)^2}{2!} + \mathbf{F}^3\mathbf{x}(t_0)\frac{(t-t_0)^3}{3!} + \ldots \\ &= \mathbf{x}(t_0)\left(\mathbf{I} + \mathbf{F}(t-t_0) + \mathbf{F}^2\frac{(t-t_0)^2}{2!} + \mathbf{F}^3\frac{(t-t_0)^3}{3!} + \ldots\right)\end{aligned}$$

$$: \Longrightarrow \boxed{\mathbf{x}(t) = \underbrace{e^{\mathbf{F}(t-t_0)}}_{\Phi_{t_0,t}}\mathbf{x}(t_0)}$$

Note (for a squared matrix) :

$$\left[e^{\mathbf{A}} = \mathbf{I} + \mathbf{A} + \tfrac{1}{2!}\mathbf{A}^2 + \tfrac{1}{3!}\mathbf{A}^3 + \ldots\right]$$

EPFL

Numerical evaluation of Φ_k $\mathbf{Q}_k$

Step 1: form and auxiliary matrix **A**

$$\mathbf{A} = \begin{bmatrix} -\mathbf{F} & \mathbf{G}\mathbf{W}\mathbf{G}^T \\ \mathbf{0} & \mathbf{F}^T \end{bmatrix} \cdot (t_k - t_{k-1})$$

Note 1: on the diagonal of **W** are either zeros or variances of the process (white) noise ***Q***

Step 2: using Matlab / Python form $e^{\mathbf{A}}$, call it **B**

$$\mathbf{B} = \mathrm{expm}(\mathbf{A}) = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix} = \begin{bmatrix} \dots & \boldsymbol{\Phi}_k^{-1}\mathbf{Q}_k \\ \mathbf{0} & \boldsymbol{\Phi}_{\mathbf{k}}{}^T \end{bmatrix}$$

Step 3: Obtain $\boldsymbol{\Phi}_k$, $\mathbf{Q}_k$ from the components of **B** :

$$\boxed{\begin{aligned} \boldsymbol{\Phi}_k &= (\mathbf{B}_{22})^T \\ \mathbf{Q}_k &= \boldsymbol{\Phi}_k \cdot \left(\boldsymbol{\Phi}_k^{-1}\mathbf{Q}_k\right) = \boldsymbol{\Phi}_k \cdot \mathbf{B}_{12} \end{aligned}}$$

Note 2: for const. time interval and invariant F, this operation is needed only once!

Sensor fusion – agenda

Kalman filter – base (Week 9)

- Intuitive approach
- Discrete KF – components, steps, implementation (Lab 5)

Kalman filter – extension (Week 10)

- Computation of transition and process noise matrix
- **Linearized & Extended Kalman filter**
- Some other ‘motion model’ examples

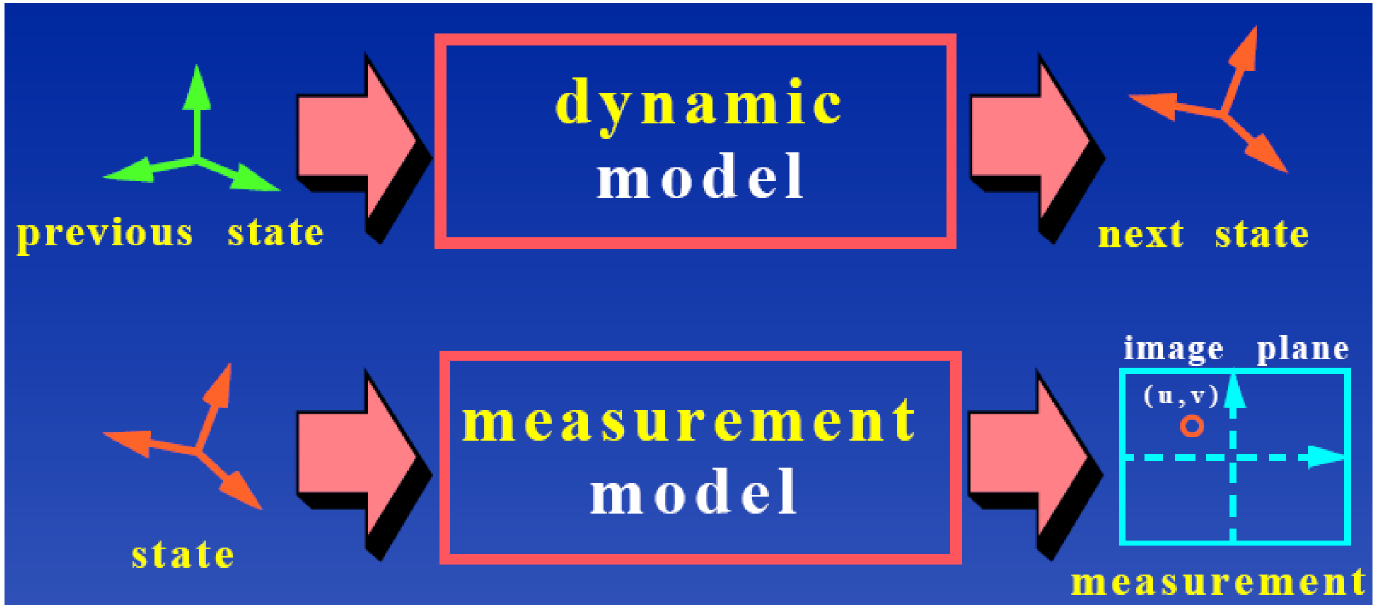
INS/GPS integration (Week 11)

- Theory of a differential filter
- Practice – derivation & implementation (Lab 6)

Sensor orientation (Week 12)

- Direct & integrated orientation of optical sensors

Discrete Kalman Filter
models



Linearized Kalman Filter (1)

"Raison d'être"

- Either the process model and/or measurement model are non-linear functions

	Linear (last time)	Non-linear (general)
Process	$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{w}$	$\dot{\mathbf{x}} = f\left(\mathbf{x},\, t,\, \underbrace{\mathbf{u}_d}_{\text{deterministic forcing input}}\right) + \underbrace{\mathbf{u}(t)}_{\text{random noise}}$
Measurement	$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v}$	$\mathbf{z} = h\left(\mathbf{x},\, t\right) + \mathbf{v}(t)$

Linearized Kalman Filter (2)

Approach

- With an approximate* knowledge of states, we can re-formulate the problem as a "linear in a difference"

$$\mathbf{x}(t) = \mathbf{x}^*(t) + \Delta\mathbf{x}(t)$$

* approximate (nominal) state

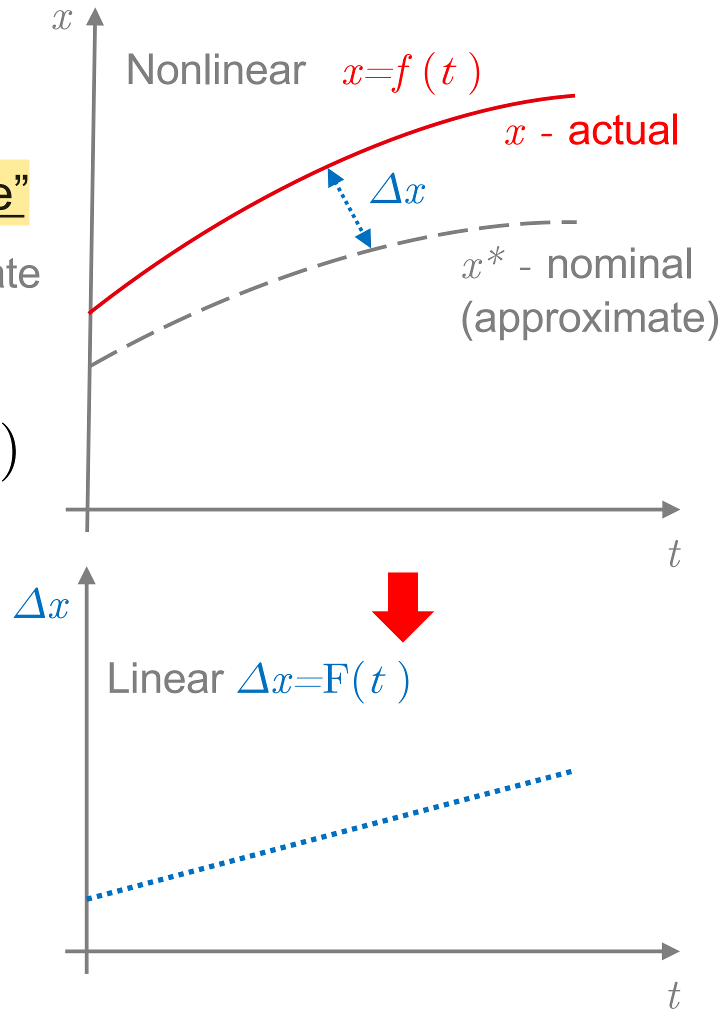
state correction

$$\dot{\mathbf{x}}^*(t) + \Delta\dot{\mathbf{x}}(t) = f\left(\mathbf{x}^* + \Delta\mathbf{x},\, t,\, \mathbf{u}_d\right) + \mathbf{u}(t)$$

$$\mathbf{z} = h\left(\mathbf{x}^* + \Delta\mathbf{x},\, t\right) + \mathbf{v}(t)$$

Assumption

- Avalibility of $\mathbf{x}^*$ - nominal state (e.g. trajectory), "close enough" in a sense that the correction is linear !



Linearized Kalman Filter (3)

Non-linear process model

$$\dot{\mathbf{x}}^*(t) + \Delta\dot{\mathbf{x}}(t) = f\left(\mathbf{x}^* + \Delta\mathbf{x},\, t,\, \mathbf{u}_d\right) + \mathbf{u}(t)$$

$$\dot{\mathbf{x}}^* + \Delta\dot{\mathbf{x}} \approx f\left(\mathbf{x}^*,\, t,\, \mathbf{u}_d\right) + \left[\frac{\partial f()}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*} \Delta\mathbf{x} + \mathbf{u}(t)$$

Needed - existence of nominal states

- e.g. trajectory “close enough”, for example INS:

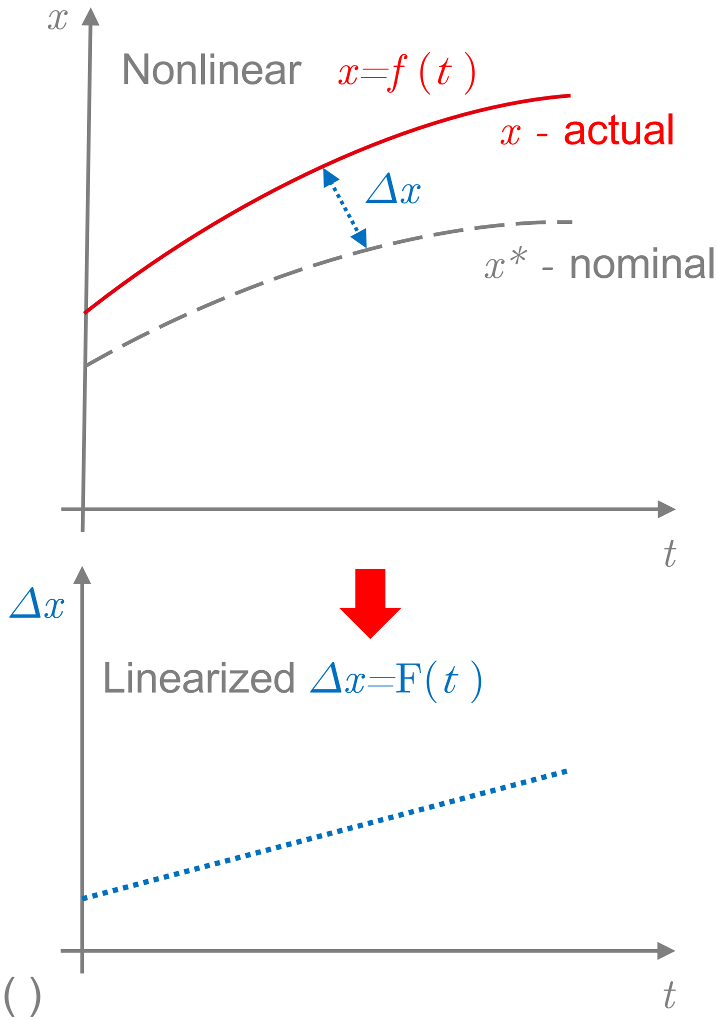
$$\dot{\mathbf{x}}^* = f\left(\mathbf{x}^*,\, t,\, \mathbf{u}_d\right) \Longrightarrow \mathbf{x}^*$$

<u>Linearized filter</u>

- Estimates only the *corrections* of states :

$$\Delta\dot{\mathbf{x}} = \underbrace{\left[\frac{\partial f()}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*}}_{\mathbf{F}} \Delta\mathbf{x} + \mathbf{u}(t)$$

$\mathbf{F}$ - perturbation of nominal differential eq. $f\,(\,)$



Linearized Kalman Filter (4)

Non-linear measurement model

$$\mathbf{z} = h\left(\mathbf{x},\ t\right) + \mathbf{v}(t)$$

$$\mathbf{z} \approx h\left(\mathbf{x}^*,\ t\right) + \left[\frac{\partial h()}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*} \Delta\mathbf{x} + \mathbf{v}(t)$$

From the real measurement subtract the "predicted" :

- To obtain a differential measurement

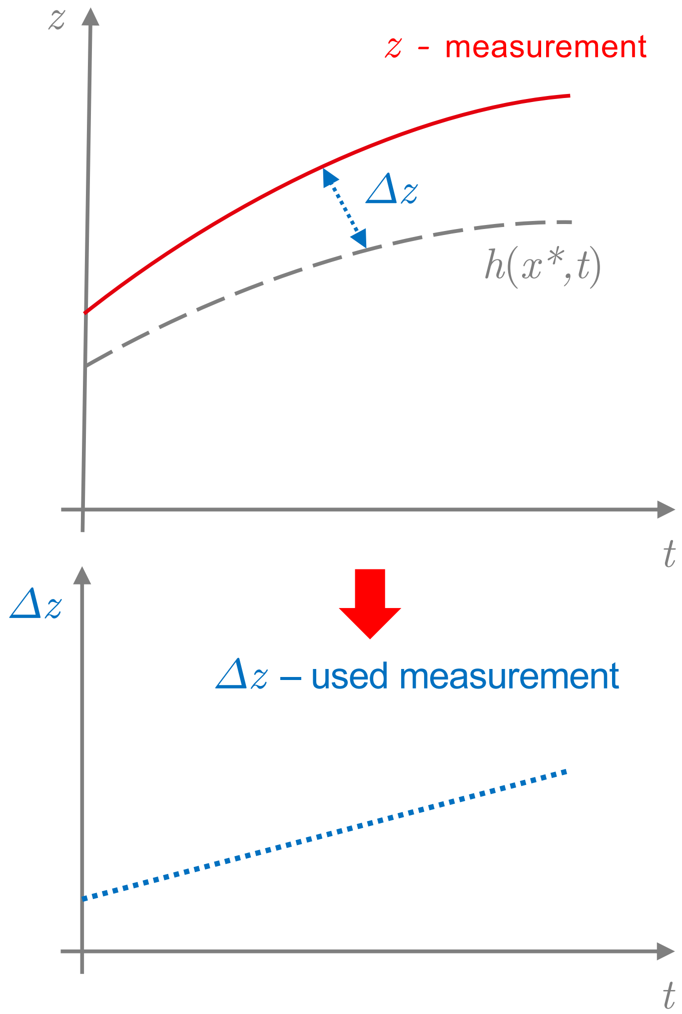
$$\underset{\text{used instead of } \mathbf{z} \text{ in the KF}}{\Delta\mathbf{z}} = \mathbf{z} - h\left(\mathbf{x}^*, t\right) = \underbrace{\left[\frac{\partial f()}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*}}_{\mathbf{H}-\text{ Jacobian}} \Delta\mathbf{x} + \mathbf{v}(t)$$

→ Use as **H** in the KF

In case of non-linear process model, we need

- "close enough param.", for example via INS:

$$\dot{\mathbf{x}}^* = f\left(\mathbf{x}^*,\ t,\ \mathbf{u}_d\right) \Longrightarrow \mathbf{x}^* \approx \mathbf{x}$$



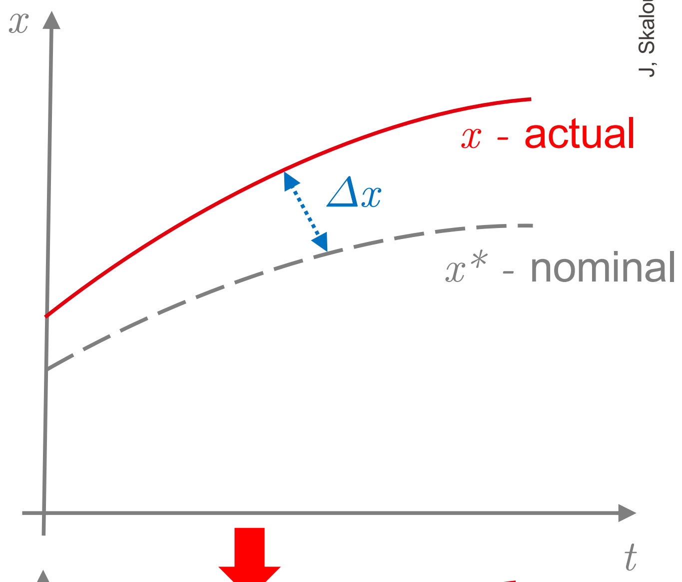
Extended Kalman Filter

Mathematical “acrobacy” in engineering

Idea

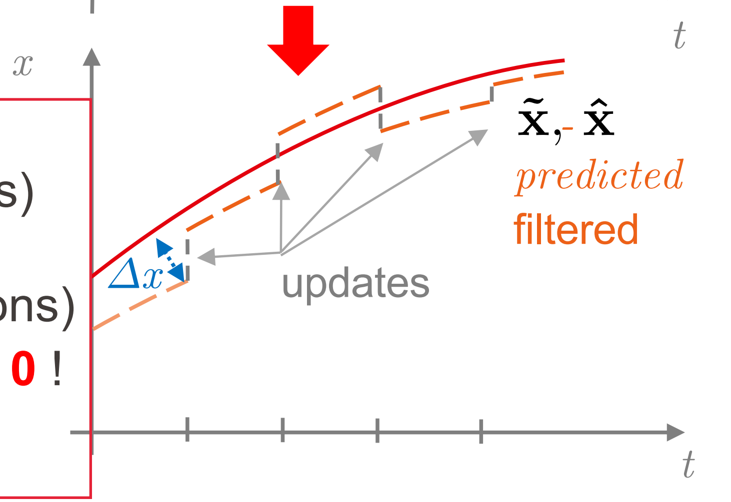
- In the approximation replace the nominal state with the predicted/filtered state:

$$\mathbf{x}^*(t) \longrightarrow \tilde{\mathbf{x}}(t)/\hat{\mathbf{x}}(t)$$



Implications

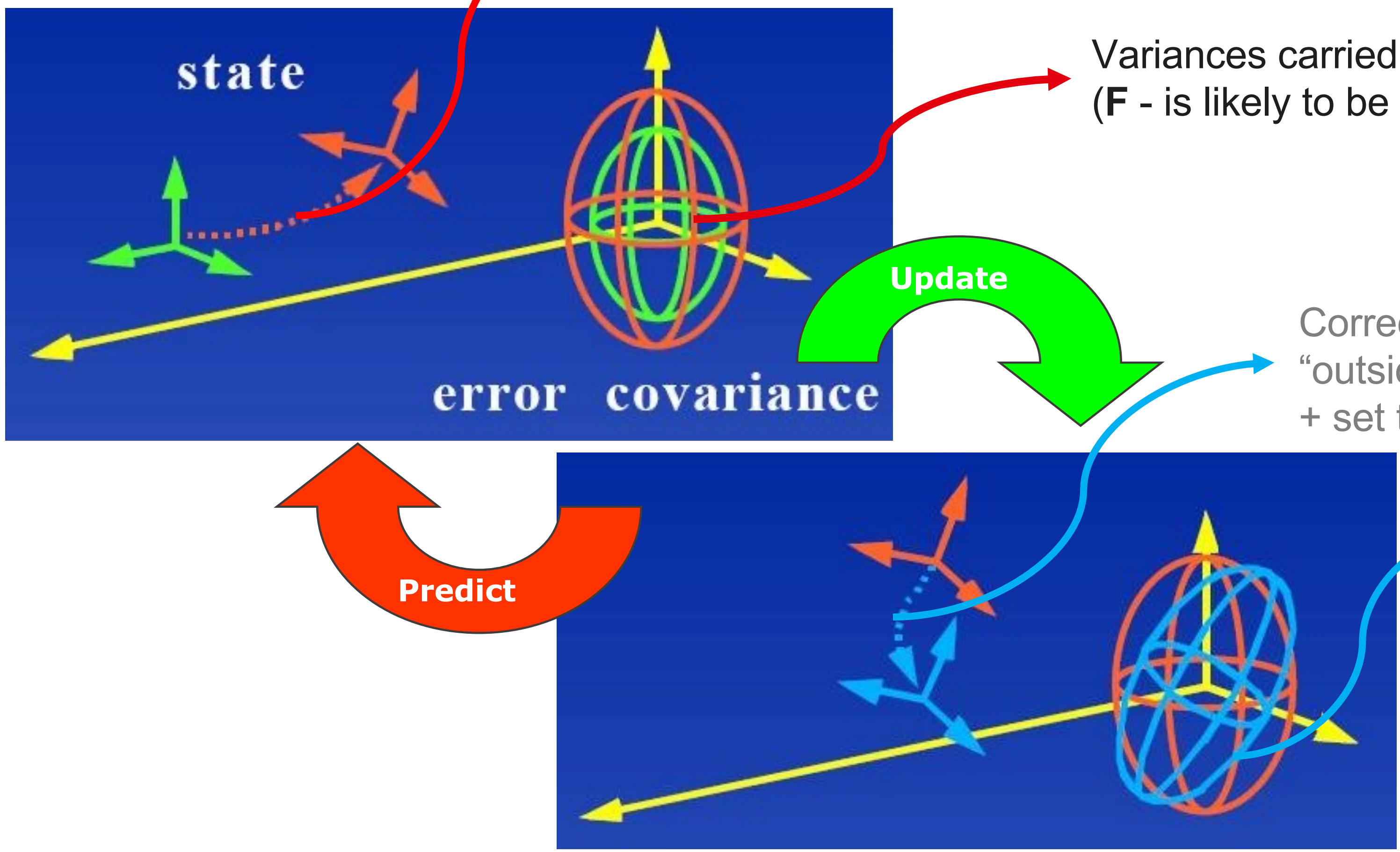
1. Nominal state is predicted via a non-linear equation
2. The filter estimates only differential quantities (errors)
3. After measurement update the nominal state (1) is corrected with the estimated values (errors/corrections)
4. After (3), the differential states in the filter are set to **0** ! [corrections are considered in prediction via (1)+(3)]



Extended Kalman Filter
operation

Main state “outside” the filter (by solving non-linear eq.)
(inside only errors / differential filter)

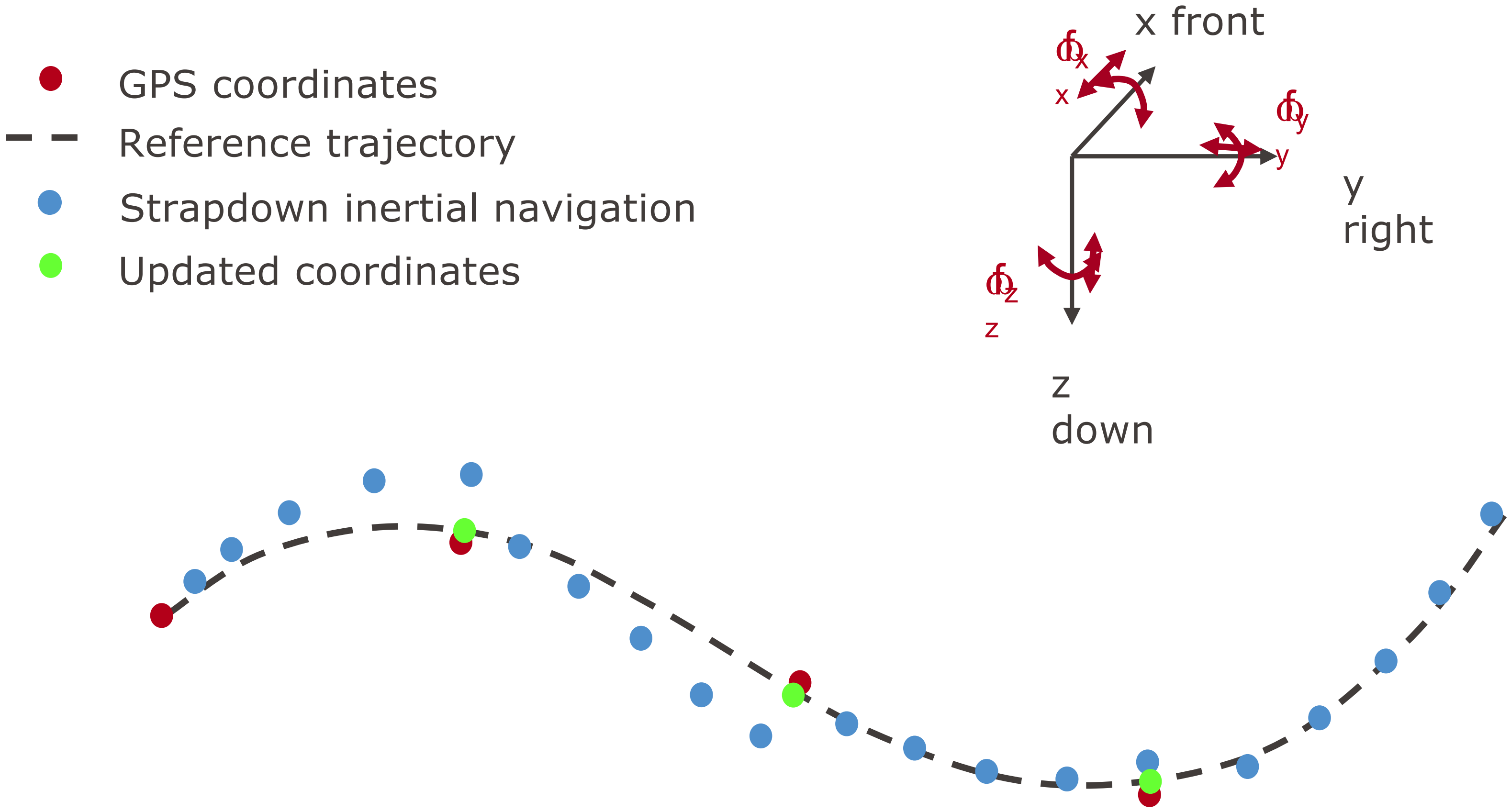
Variances carried “inside” the filter
(**F** - is likely to be time-vayring)



Corrections are applied
“outside” the filter
+ set to zero in the filter

Appplied
”inside”
of filter

EKF in INS/GPS(GNSS) integration



EPFL

Linear vs Linearized (Extended) Kalman filter

Gain (weight)

possibly via linearization

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k \mathbf{H}_k^T \left(\mathbf{H}_k \tilde{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1}$$

Only if non-linear
→ replaced by $\Delta\mathbf{z}$!

Measurement

$\mathbf{z}_k, \mathbf{R}_k$

= 0 in EKF
(after update applied)

F → obtained
by linearization

Prediction (~) or (-) symbol

$$\tilde{\mathbf{x}}_k = \Phi_k \hat{\mathbf{x}}_{k-1}$$

$$\tilde{\mathbf{P}}_k = \Phi_{k-1,k} \hat{\mathbf{P}}_{k-1} \Phi_{k-1,k}^T + \mathbf{Q}_k$$

State update (hat) or (+) symbol

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \tilde{\mathbf{x}}_k)$$

start

$\hat{\mathbf{x}}_0, \mathbf{P}_0$

A priory info

$\Phi, \mathbf{G}, Q, \mathbf{R}$

$k+1 \rightarrow k$

Covariance update

$$\hat{\mathbf{P}}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \tilde{\mathbf{P}}_k$$

$\hat{\mathbf{x}}_k, \mathbf{P}_k$

output

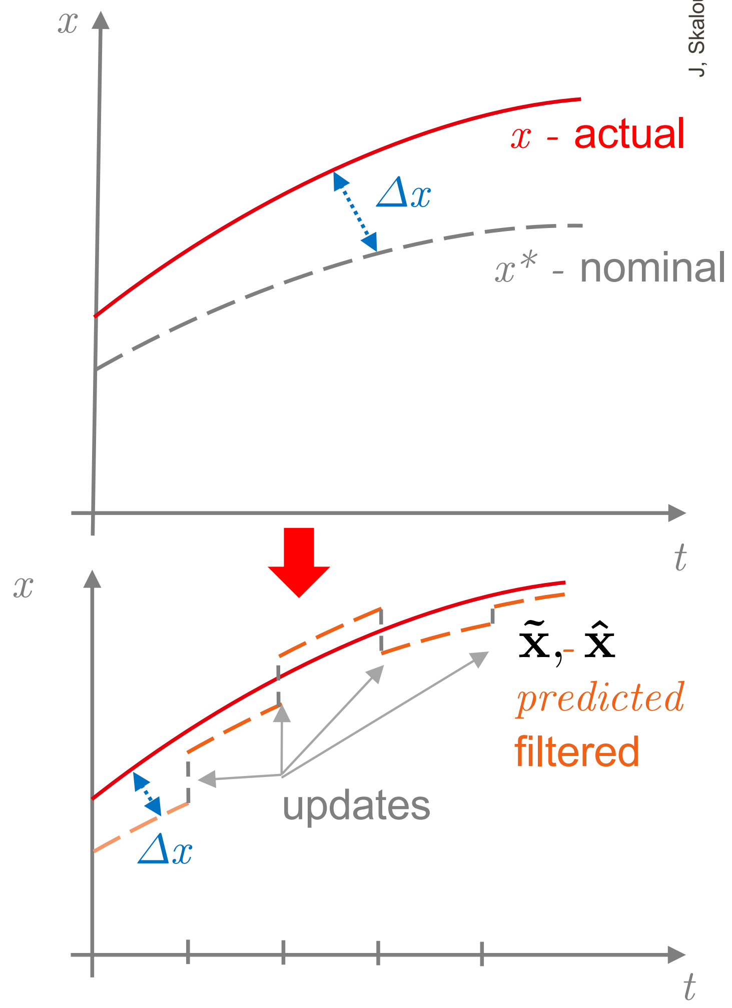
More on Extended Kalman Filter

Important details are in the implementation of (last) Lab 6

Preparation – at home (before next lecture!)

- read 4 pages in Lab 6 help (8-11):
  Lab 6(10) - help (filter setup)
  on Moodle in Week 13-19 May!

$$\mathbf{x}^*(t) \longrightarrow \tilde{\mathbf{x}}(t)/\hat{\mathbf{x}}(t)$$



Sensor fusion – agenda

Kalman filter – base (Week 9)

- Intuitive approach
- Discrete KF – components, steps, implementation (Lab 5)

Kalman filter – extension (Week 10)

- Computation of transition and process noise matrix
- Linearized & Extended Kalman filter
- **Some other 'motion model' examples**

INS/GPS integration (Week 11)

- Theory of a differential filter
- Practice – derivation & implementation (Lab 6)

Sensor orientation (Week 12)

- Direct & integrated orientation of optical sensors

Discrete Kalman Filter

Process. model – const. velocity (Lab 5)

Transition model, Φ

1.

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \mathbf{v}_k \Delta t$$

$$\mathbf{v}_k = \mathbf{v}_{k-1} + [\mathbf{w}_v]$$

Dynamic model, $\mathbf{F}$

2.

$$\dot{p}_n = v_n$$
$$\dot{p}_e = v_e$$
$$\dot{v}_n = 0 + [w_{\dot{v}_n}]$$
$$\dot{v}_e = 0 + [w_{\dot{v}_e}]$$

zero mean white noise

4. $$\mathbf{\Phi} = e^{\mathbf{F}\Delta t} = \mathbf{I} + \mathbf{F}\Delta t + \cancel{\mathbf{F}^2 \frac{\Delta t^2}{2!}} + \ldots$$

$$\begin{bmatrix} p_n \\ p_e \\ v_n \\ v_e \end{bmatrix}_k = \begin{bmatrix} 1 & \cdot & \Delta t & \cdot \\ \cdot & 1 & \cdot & \Delta t \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & 1 \end{bmatrix} \begin{bmatrix} p_n \\ p_e \\ v_n \\ v_e \end{bmatrix}_{k-1}$$

3. Form $\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{w}$

$$\begin{bmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{v}_n \\ \dot{v}_e \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} p_n \\ p_e \\ v_n \\ v_e \end{bmatrix} + \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ 1 & \cdot \\ \cdot & 1 \end{bmatrix} \begin{bmatrix} w_{\dot{v}_n} \\ w_{\dot{v}_e} \end{bmatrix}$$

Lab 5 / KF
Process. model – const. velocity demo

Discrete Kalman Filter

Process. model – const. acceleration

Transition Φ

Dynamic $\mathbf{F}$

1.

2.

4. $\boldsymbol{\Phi} = e^{\mathbf{F}\Delta t} = \mathbf{I} + \mathbf{F}\Delta t + \mathbf{F}^2\frac{\Delta t^2}{2!} + \dots$

3. Form $\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{w}$

Discrete Kalman Filter
Process. model – const. acceleration

Transition Φ

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0 & \Delta t & 0 & \frac{1}{2}\Delta t^2 & 0 \\ \cdot & 1 & 0 & \Delta t & 0 & \frac{1}{2}\Delta t^2 \\ \cdot & \cdot & 1 & 0 & \Delta t & 0 \\ \cdot & \cdot & \cdot & 1 & 0 & \Delta t \\ \cdot & \cdot & \cdot & \cdot & 1 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 \end{bmatrix}$$

Noise shaping G

$$\mathbf{G} = \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & \cdot \\ \cdot & 1 \end{bmatrix}$$

Continuous noise $Q(t)$

$$Q(t) = \begin{bmatrix} q_{\dot{a}}^2 & \cdot \\ \cdot & q_{\dot{a}}^2 \end{bmatrix}$$

where $q_{\dot{a}}$ is noise PSD

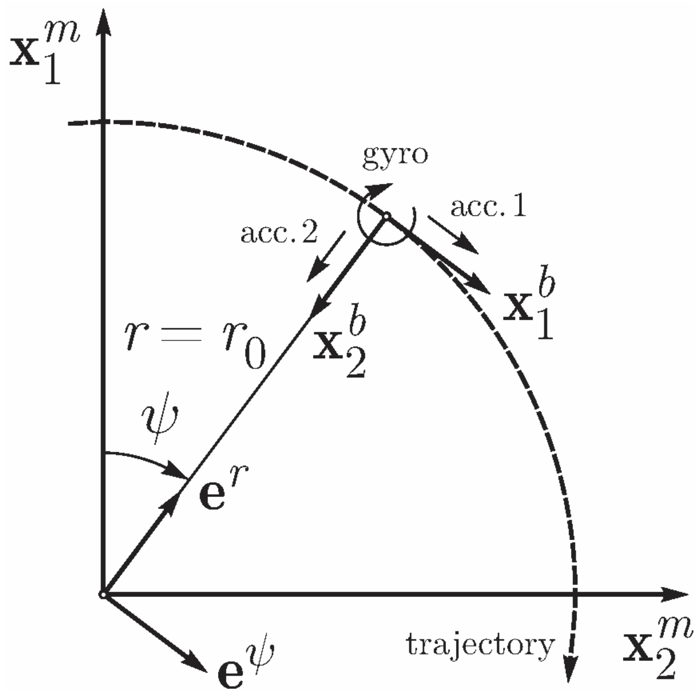
Discrete noise $\mathbf{Q}_k$

$$\mathbf{Q}_k = \int \mathbf{\Phi}\mathbf{G}Q(t)\mathbf{G}^T\mathbf{\Phi}^T d\tau$$

Evaluated numerically, see slide # 8

Discrete Kalman Filter
Process. model – case Lab 5

Which model is better?



Const. velocity?

Const. acceleration?

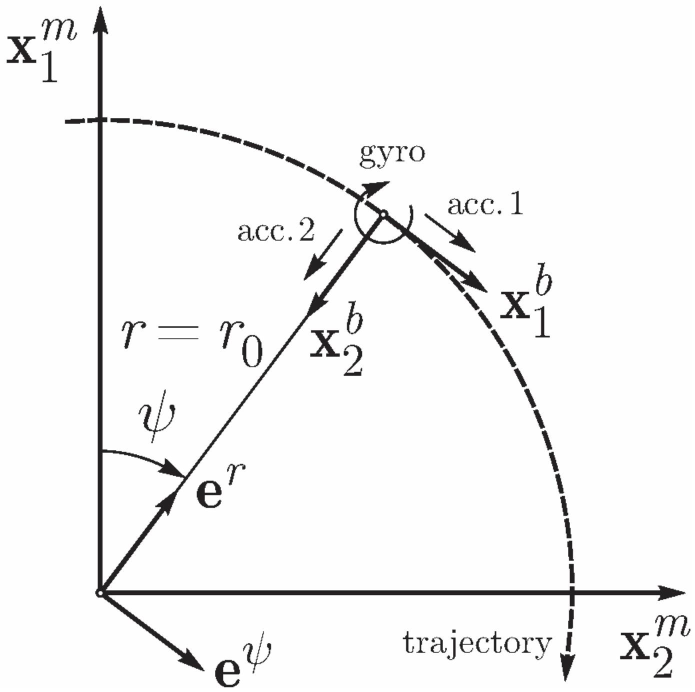
Approaches
- Try and see?

Lab 5 / KF
Process. model – const. acceleration demo

Discrete Kalman Filter

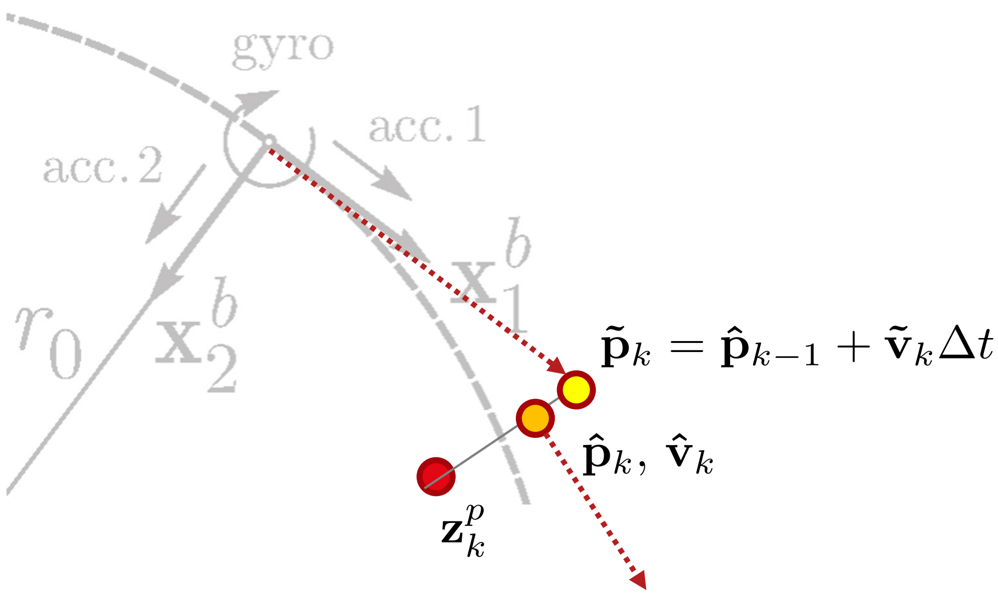
Process. model – case Lab 5

Which model is better?

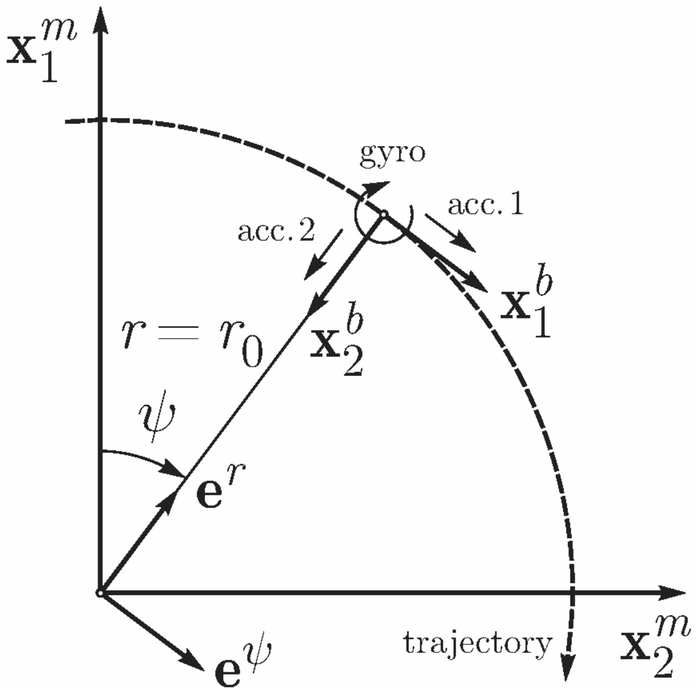


Approaches

- Try and see? - not conclusive
- Reason? – may not be optimal …
- Why?



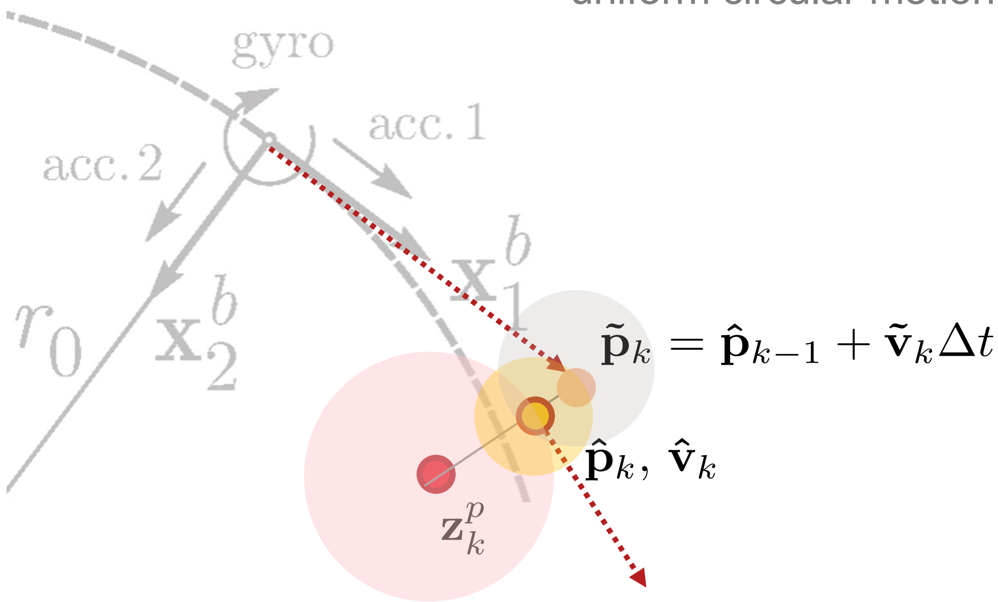
Discrete Kalman Filter
Process. model – case Lab 5



Approaches

- Try and see? – not conclusive
- Reason? – certainly not optimal !
- How to improve? – model the reality!

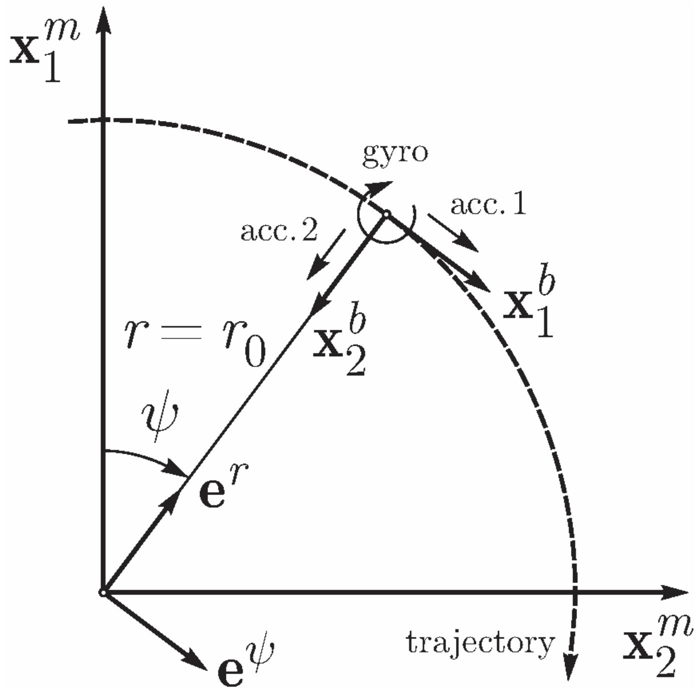
uniform circular motion



Discrete Kalman Filter
Process. model – case Lab 5

How to model

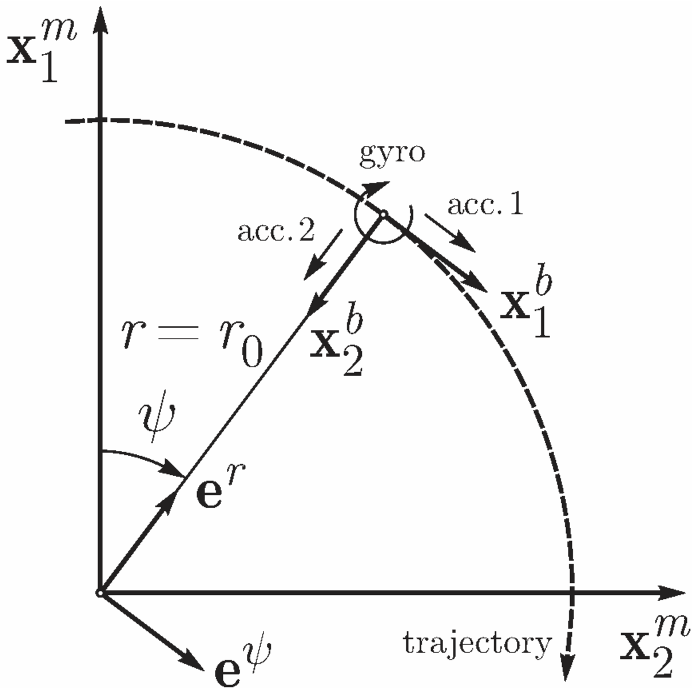
uniform circular motion?



Hints

1. How circle is defined?
2. How uniform motion is defined?
3. How to express these conditions in “dynamic” (differential) form?

Discrete Kalman Filter
Process. model – uniform circular motion



How many states?

- something for the position
- something for the velocity
- not less or more than is needed

$\mathbf{x} =$

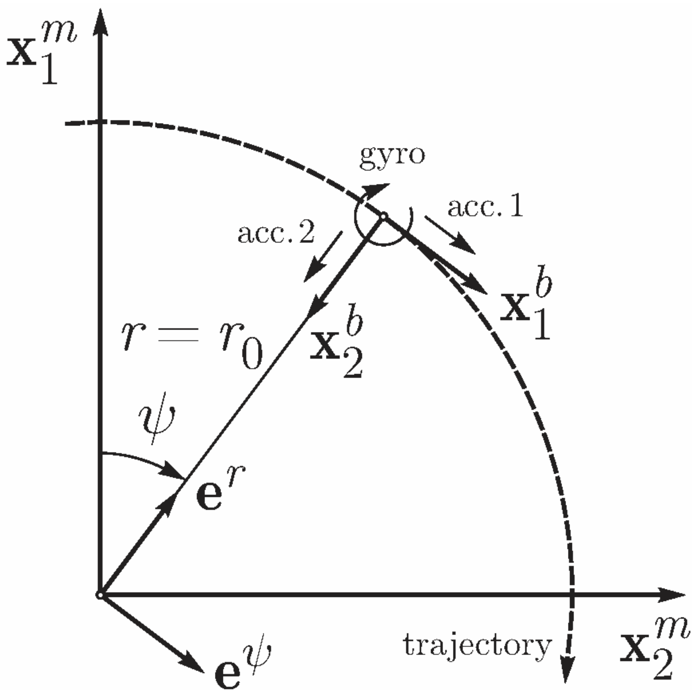
Where to add uncertainity?

Discrete Kalman Filter

Process. model – uniform circular motion

Grouping it all

Uniform circular motion



Process model in a form $\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{w}$

$$\begin{bmatrix} \dot{r} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} r \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 1 & \cdot \\ \cdot & \cdot \\ \cdot & 1 \end{bmatrix} \begin{bmatrix} w_{\dot{r}} \\ w_{\ddot{\psi}} \end{bmatrix}$$

$$\mathbf{\Phi} = e^{\mathbf{F}\Delta t} = \mathbf{I} + \mathbf{F}\Delta t + \cancel{\mathbf{F}^2 \frac{\Delta t^2}{2!}} + \dots$$

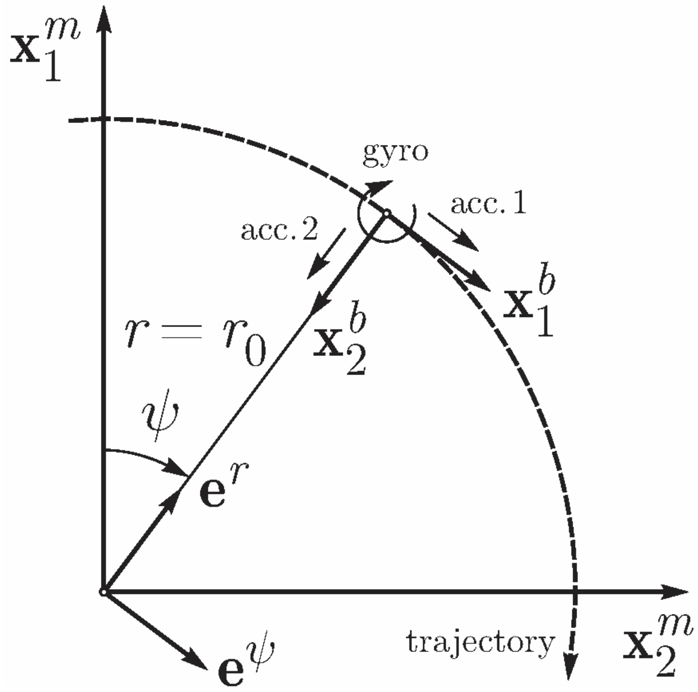
$$\mathbf{\Phi} = \begin{bmatrix} 1 & \cdot & \cdot \\ \cdot & 1 & \Delta t \\ \cdot & \cdot & 1 \end{bmatrix} \qquad Q(t) = \begin{bmatrix} q_{\dot{r}}^2 & \cdot \\ \cdot & q_{\ddot{\psi}}^2 \end{bmatrix}$$

Lab 5 / KF

Process. model – circular motion demo

Discrete Kalman Filter
Process. model – uniform circular motion

Implementation challenges

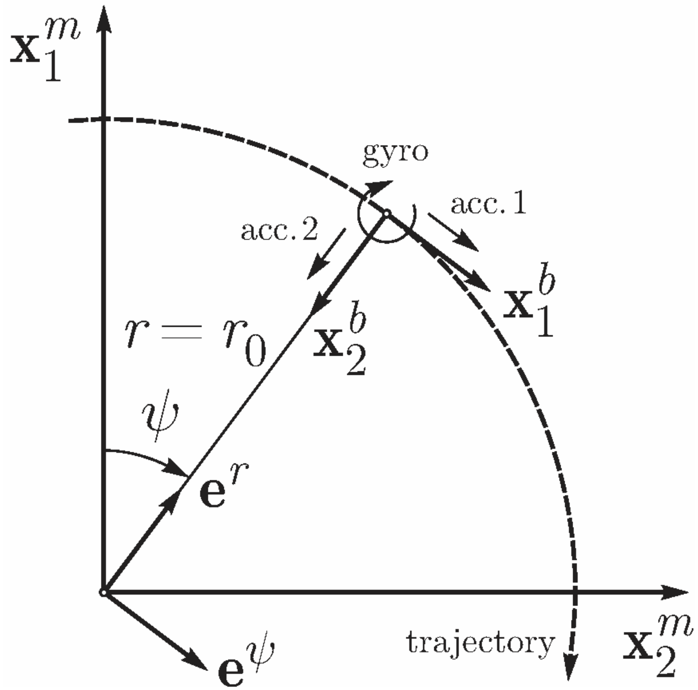


I. GPS position updates are in **cartesian coordinates** !

What to do?

Discrete Kalman Filter
Process. model – uniform circular motion

Implementation challenges



I. GPS position updates are in **cartesian coordinates** !

Solution **A** - use the non-linear o. model

$$\mathbf{z} = h(\mathbf{x}) + \mathbf{v}$$

$$\mathbf{z}_k = h(\tilde{\mathbf{x}}_k) + \underbrace{\left[\frac{\partial h}{\partial \mathbf{x}}\right]_{\mathbf{x}=\tilde{\mathbf{x}}_k}}_{\mathbf{H}}$$

$$h(1): \; p_n = r\cos\psi$$

$$h(2): \; p_e = r\sin\psi$$

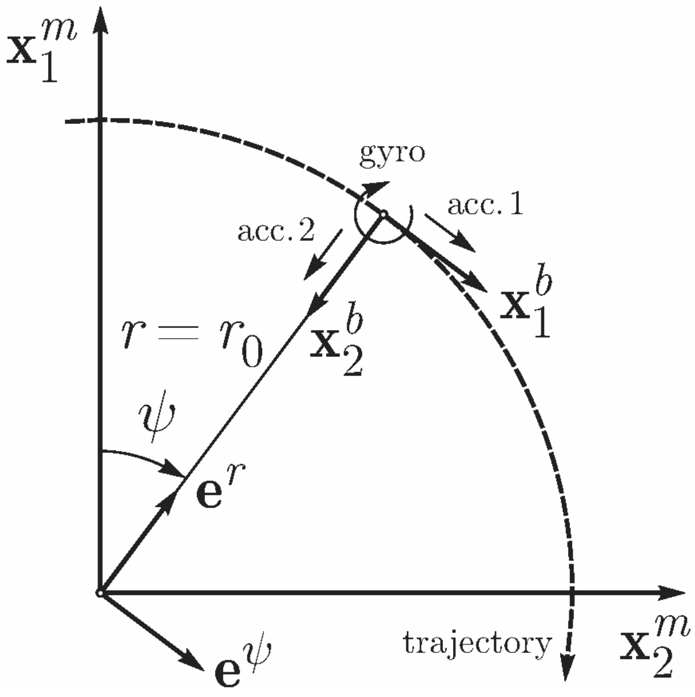
$$\mathbf{H} = \begin{bmatrix} \frac{\partial h(1)}{\partial r} & \frac{\partial h(1)}{\partial \psi} & \frac{\partial h(1)}{\partial \dot{\psi}} \\ \frac{\partial h(2)}{\partial r} & \frac{\partial h(2)}{\partial \psi} & \frac{\partial h(2)}{\partial \dot{\psi}} \end{bmatrix} = \begin{bmatrix} \cos\psi & -r\sin\psi & 0 \\ \sin\psi & r\cos\psi & 0 \end{bmatrix}$$

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}(\mathbf{z}_k - h(\tilde{\mathbf{x}}_k) - \mathbf{H}\underbrace{\cancel{\Delta\mathbf{x}}}_{=0})$$

– after update

Discrete Kalman Filter
Process. model – uniform circular motion

Implementation challenges



I. GPS position updates are in **cartesian coordinates** !

Solution **B** - use "pseudo" observations

$$z_r,\ z_\psi: \qquad r = \sqrt{p_n^2 + p_e^2} \quad \tan\psi = \frac{p_e}{p_n}$$

– linear update:

$$\begin{bmatrix} z_r \\ z_\psi \end{bmatrix} = \begin{bmatrix} 1 & \cdot & \cdot \\ \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} r \\ \psi \\ \dot{\psi} \end{bmatrix}$$

–requires transforming:

$$[\sigma_{p_n}, \sigma_{p_e}, \sigma_{v_n}, \sigma_{v_e}] \longrightarrow [\sigma_r, \sigma_\psi, \sigma_{\dot\psi}]$$

$$\boxed{\begin{bmatrix} \sigma_r^2 & \sigma_{r\psi} \\ \sigma_{r\psi} & \sigma_\psi^2 \end{bmatrix} = \mathbf{M} \begin{bmatrix} \sigma_{p_n}^2 & \cdot \\ \cdot & \sigma_{p_e}^2 \end{bmatrix} \mathbf{M}^T}$$

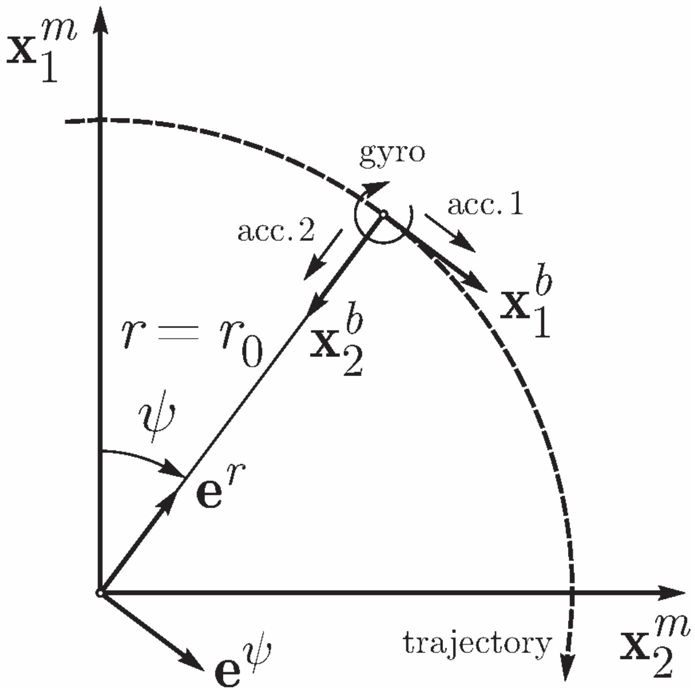
$$\mathbf{M} = \begin{bmatrix} \frac{\partial r()}{\partial p_n} & \frac{\partial r()}{\partial p_e} \\ \frac{\partial \psi()}{\partial p_n} & \frac{\partial \psi()}{\partial p_e} \end{bmatrix}$$

law of covariance propagation !

Discrete Kalman Filter

Process. model – uniform circular motion

Implementation challenges



II. Comparison with previous results

$$\left[r,\, \psi,\, \dot{\psi}\right] \longrightarrow \left[p_n,\, p_e,\, v_n,\, v_e\right]$$

$$\left[\sigma_r,\, \sigma_\psi,\, \sigma_{\dot{\psi}}\right] \longrightarrow \left[\sigma_{p_n},\, \sigma_{p_e},\, \sigma_{v_n},\, \sigma_{v_e}\right]$$

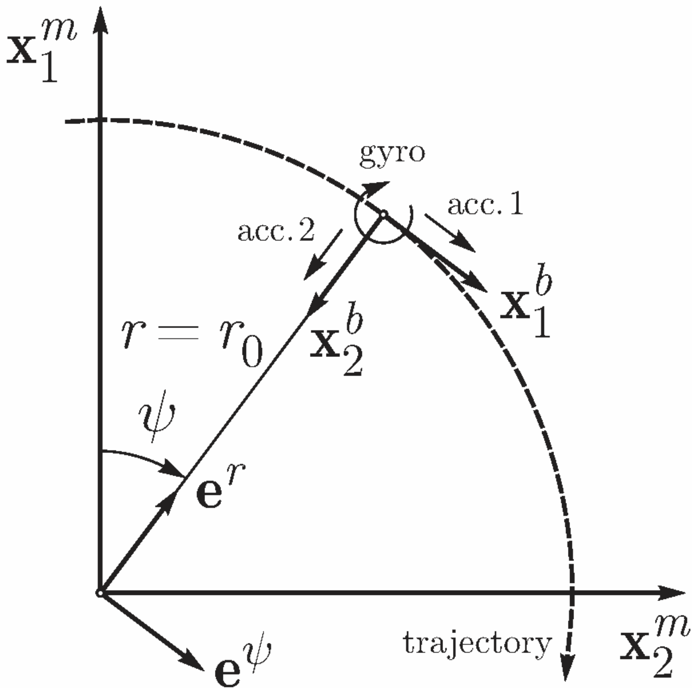
How ?

– Define equations for transformation of north & east positions & velocities

– Apply the law of covariance propagation on these equations

Discrete Kalman Filter
Process. model – uniform circular motion

Implementation challenges



II. Transforming to cartesian position

$$\boxed{\begin{aligned} p_n &= r\cos\psi \\ p_e &= r\sin\psi \end{aligned}}$$

$$\begin{aligned} d_{p_n} &= \cos\psi d_r - r\sin\psi d_\psi \\ d_{p_e} &= \sin\psi d_r + r\cos\psi d_\psi \end{aligned}$$

… and velocity

$$\boxed{\begin{aligned} \dot{p}_n &= \dot{r}\cos\psi - r\dot{\psi}\sin\psi \\ \dot{p}_e &= \dot{r}\sin\psi + r\dot{\psi}\cos\psi \end{aligned}}$$

$$\begin{aligned} d_{\dot{p}_n} &= -\dot{\psi}\sin\psi d_r - r\dot{\psi}\cos\psi d_\psi - r\sin\psi d_{\dot{\psi}} \\ d_{\dot{p}_e} &= +\dot{\psi}\cos\psi d_r - r\dot{\psi}\sin\psi d_\psi + r\cos\psi d_{\dot{\psi}} \end{aligned}$$

– apply the law of covariance propagation with **F**

$$\begin{bmatrix} d_{p_n} \\ d_{p_e} \\ d_{v_n} \\ d_{v_e} \end{bmatrix} = \begin{bmatrix} \cos\psi & -r\sin\psi & \cdot \\ \sin\psi & r\cos\psi & \cdot \\ -\dot{\psi}\sin\psi & -r\dot{\psi}\cos\psi & -r\sin\psi \\ \dot{\psi}\cos\psi & r\dot{\psi}\sin\psi & r\cos\psi \end{bmatrix} \begin{bmatrix} d_r \\ d_\psi \\ d_{\dot{\psi}} \end{bmatrix}$$