State observers based on detailed multibody models applied to an automobile

Emilio Sanjurjo

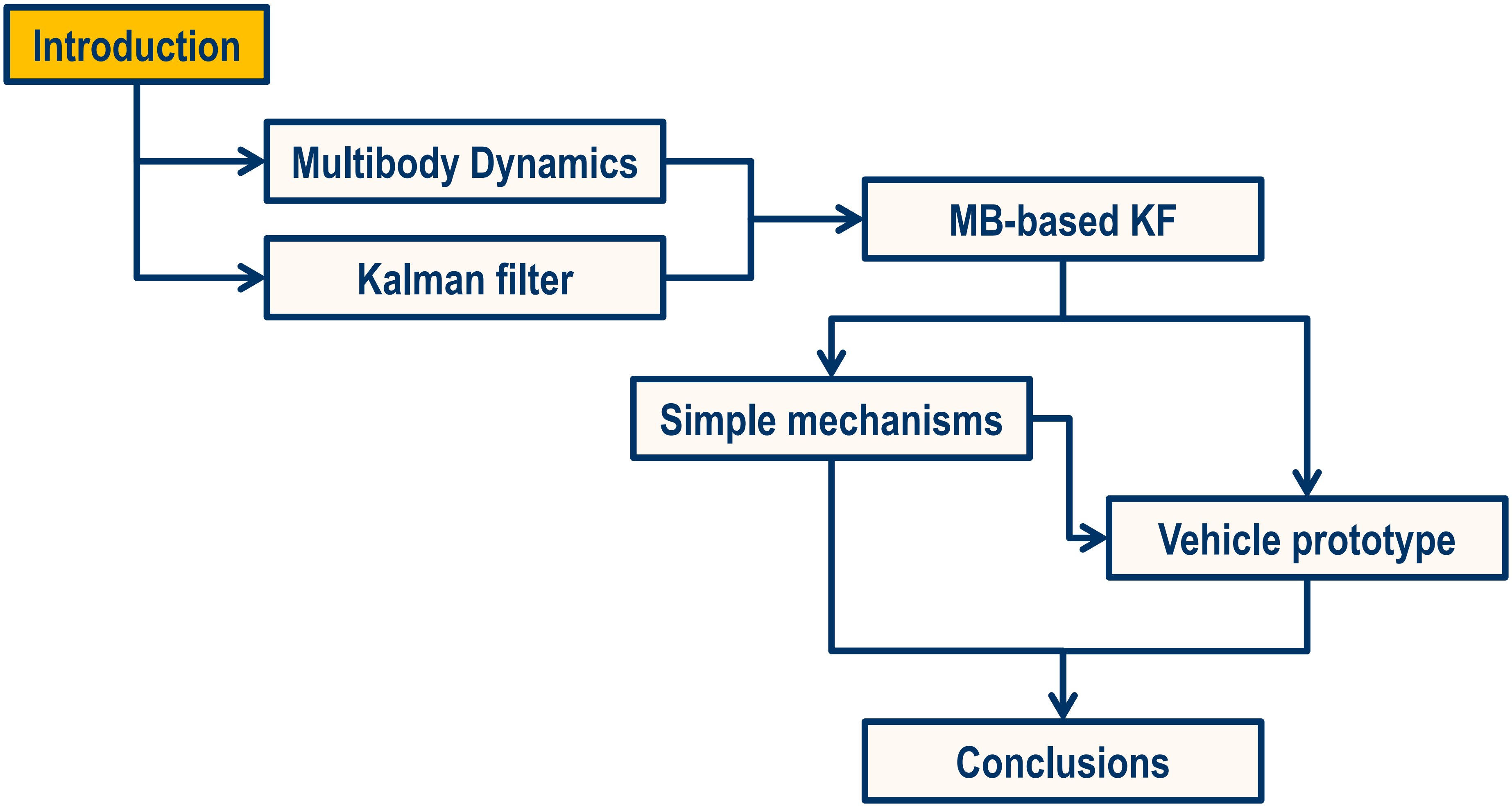
Laboratorio de Ingeniería Mecánica,

University of A Coruña

Advisors:

- Miguel Ángel Naya Villaverde
- Javier Cuadrado Aranda

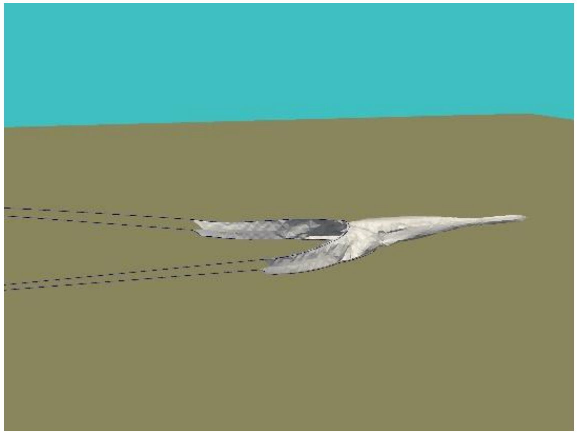
Outline

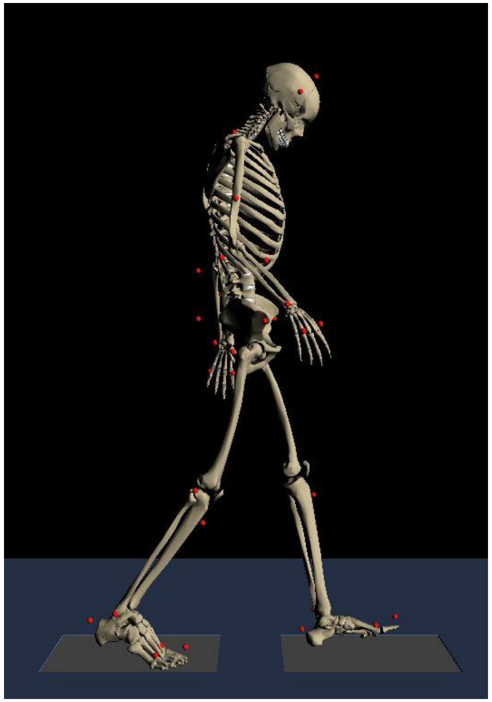


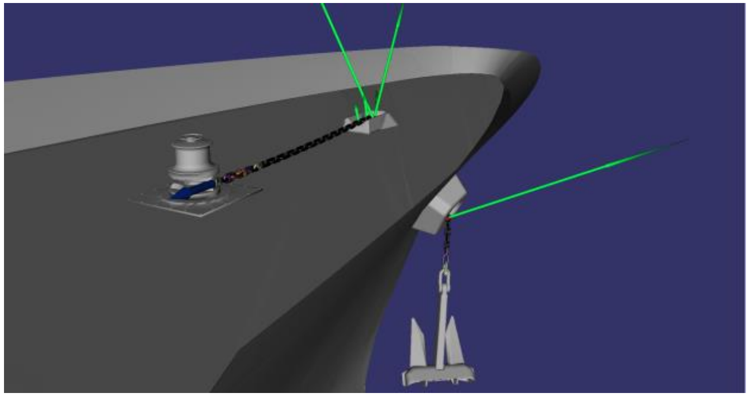
Introduction
Multibody Dynamics
Kalman filter
MB-based KF
Simple mechanisms
Vehicle prototype
Conclusions

Introduction: Research lines at the LIM

- **Efficient multibody simulations**
  - Fishing nets
  - Machinery
  - Biomechanics
  - Anchor lifting

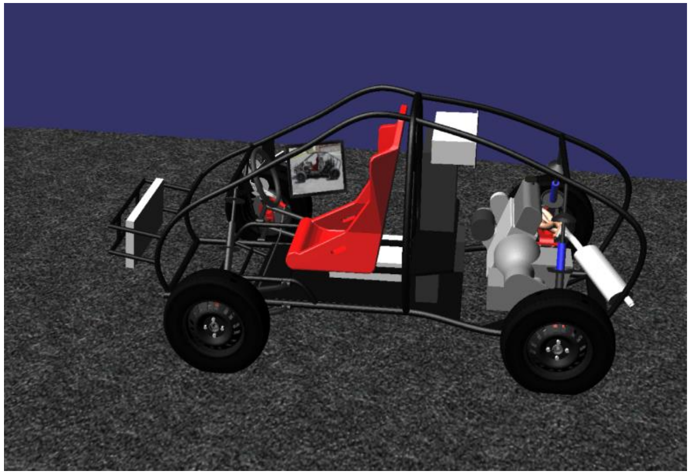




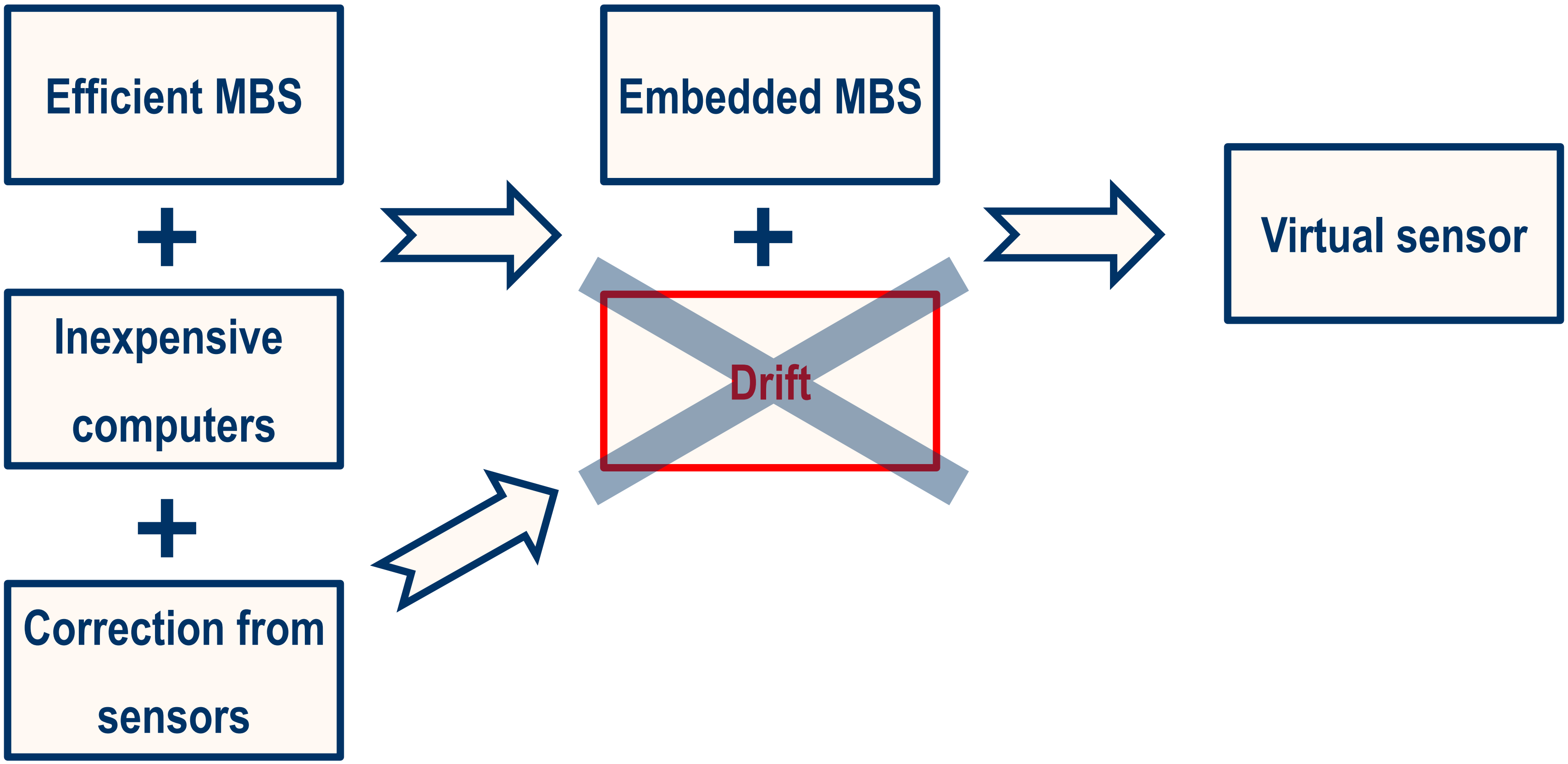


Introduction: real-time multibody simulations

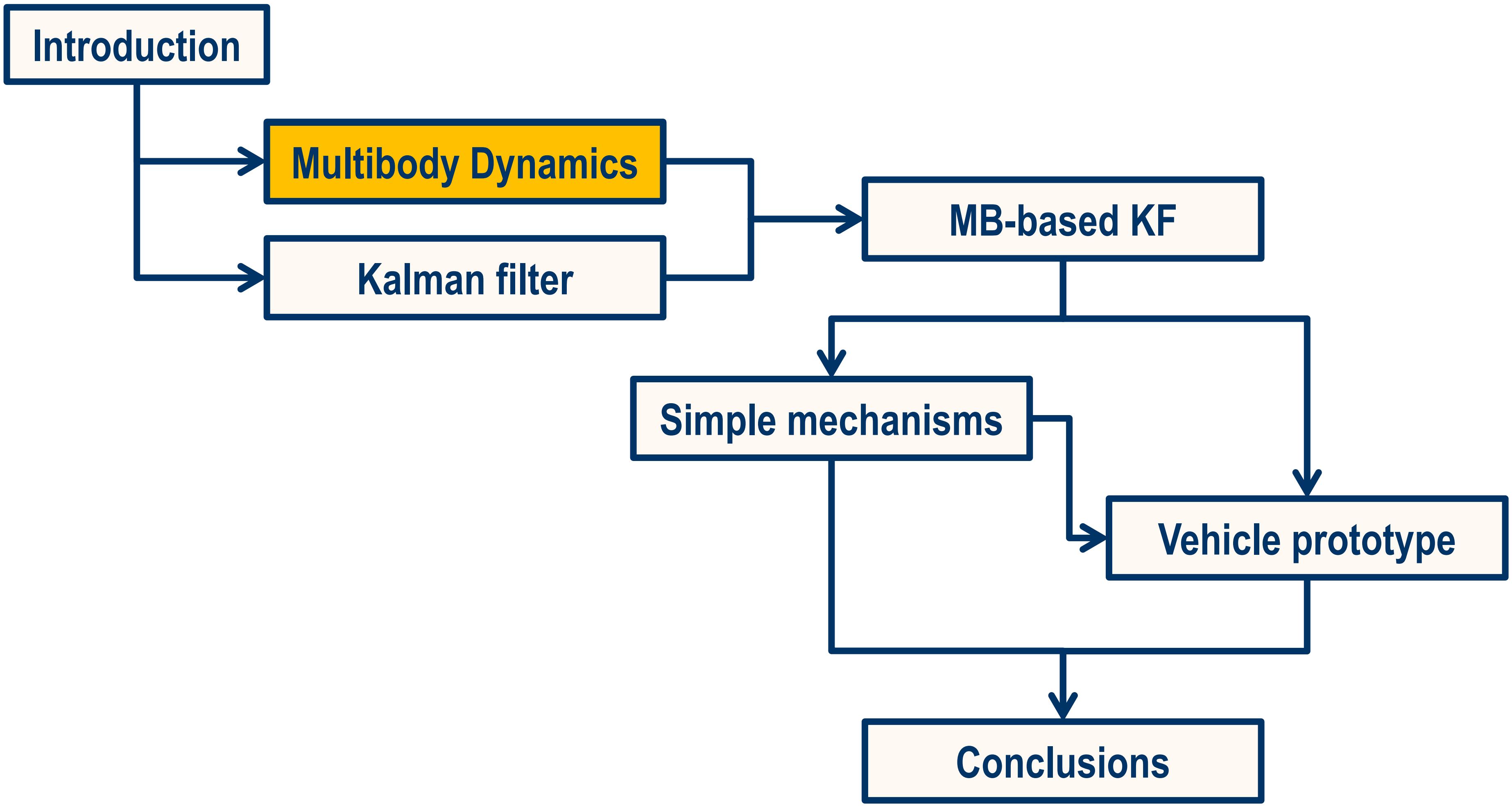
- **Hardware in the loop**
- **Human in the loop**
  - Excavator
  - Automotive simulators
- **Virtual sensors (state observers)**
  - J. Cuadrado, D. Dopico et al. (2009 to 2012), in collaboration with the CTAG (Automotive Technology Centre of Galicia)
  - R. Pastorino's Doctoral thesis (2012) and its derived works (2013 and 2014)
  - More efficiency is needed



Introduction: multibody-based state observers



Multibody Dynamics



Multibody dynamics

- Multibody dynamics: efficient methods to simulate machines and mechanisms
- Systems represented by rigid (or flexible) solids and their joints
- Geometry of bodies and joints is preserved
- General methodology
- In general, redundant coordinates are employed
- System of DAEs

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi_q}^{\top}\boldsymbol{\lambda} & = & \mathbf{Q} \\ \mathbf{\Phi} & = & \mathbf{0} \end{cases}$$

Multibody dynamics: Matrix R

- Method in independent coordinates

$$\begin{cases} \dot{\mathbf{z}} & = & \mathbf{D}\dot{\mathbf{q}} \\ \mathbf{\Phi_q}\dot{\mathbf{q}} & = & -\mathbf{\Phi}_t \end{cases} \Rightarrow \begin{bmatrix} \mathbf{\Phi_q} \\ \mathbf{D} \end{bmatrix} \dot{\mathbf{q}} = \begin{bmatrix} -\mathbf{\Phi}_t \\ \dot{\mathbf{z}} \end{bmatrix}$$

$$\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{\Phi_q} \\ \mathbf{D} \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{\Phi}_t \\ \dot{\mathbf{z}} \end{bmatrix} \equiv \begin{bmatrix} \mathbf{S} & \mathbf{R} \end{bmatrix} \begin{bmatrix} -\mathbf{\Phi}_t \\ \dot{\mathbf{z}} \end{bmatrix} = -\mathbf{S}\mathbf{\Phi}_t + \mathbf{R}\dot{\mathbf{z}}$$

- EOM in independent coordinates

$$\mathbf{R}^\top \mathbf{M}\mathbf{R}\ddot{\mathbf{z}} = \mathbf{R}^\top \left[\mathbf{Q} + \mathbf{M}\mathbf{S} \left(\dot{\mathbf{\Phi}}_t + \dot{\mathbf{\Phi}}_\mathbf{q}\dot{\mathbf{q}} \right) \right] \Rightarrow \bar{\mathbf{M}}\ddot{\mathbf{z}} = \bar{\mathbf{Q}}$$

- Any integration scheme can be used
- Position, velocity, and acceleration problems solved every iteration

Multibody dynamics: ALI3-P

- Method in dependent coordinates

$$\left\{ \begin{array}{rcl} \mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}{}^{\top}\boldsymbol{\lambda} & = & \mathbf{Q} \\ \boldsymbol{\Phi} & = & \mathbf{0} \end{array} \right. \Rightarrow \mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}{}^{\top}\alpha\boldsymbol{\Phi} + \boldsymbol{\Phi}_{\mathbf{q}}{}^{\top}\boldsymbol{\lambda}^{*} = \mathbf{Q}$$

$$\boldsymbol{\lambda}^{*}_{k,i+1} = \boldsymbol{\lambda}^{*}_{k,i} + \alpha\boldsymbol{\Phi}_{k,i+1}$$

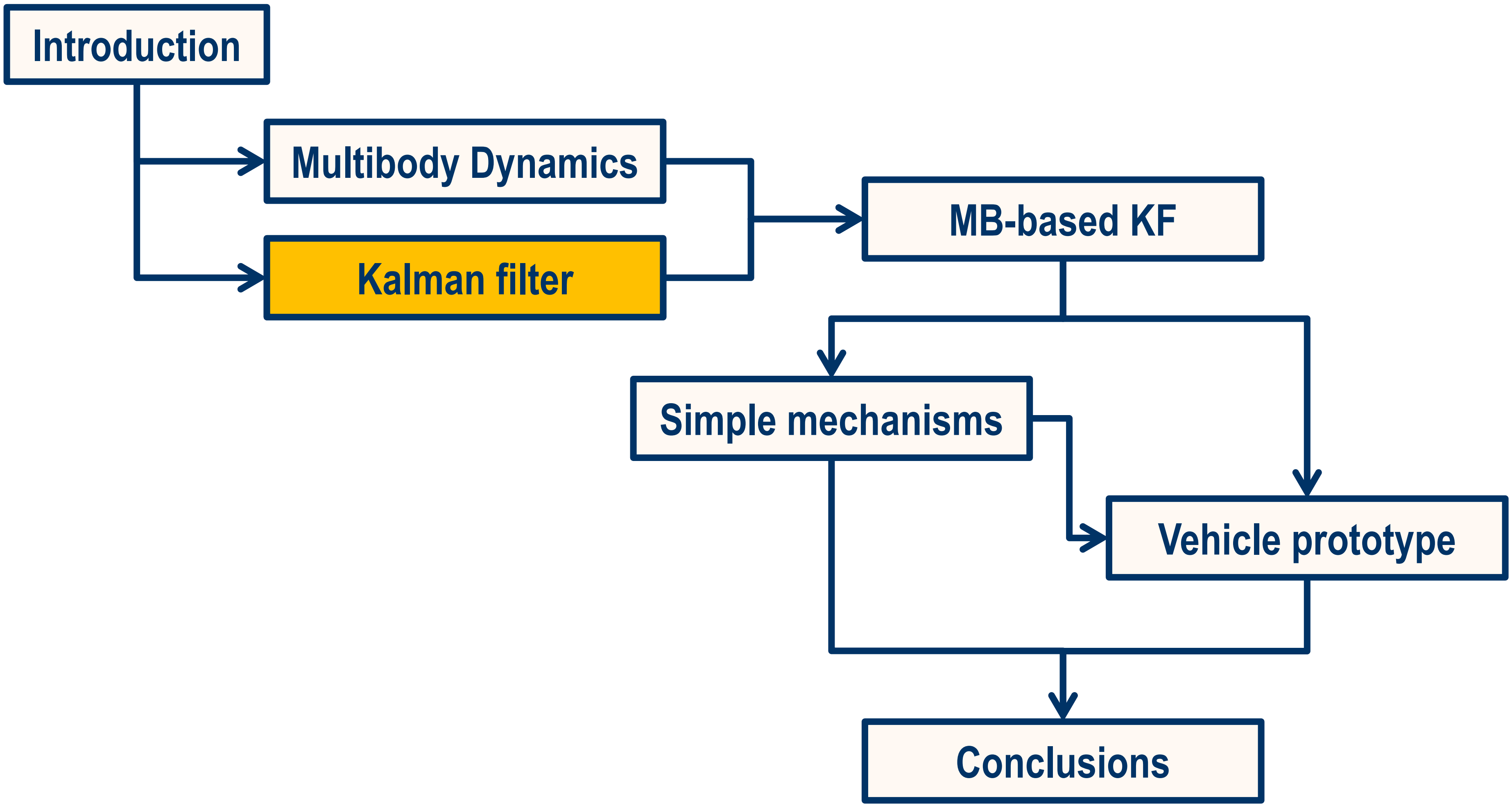
- Particularly convenient with implicit integrators

$$\dot{\mathbf{q}}_{k+1} = \frac{2}{\Delta t}\mathbf{q}_{k+1} + \dot{\mathbf{q}}^{*}_{k} \text{ , where } \dot{\mathbf{q}}^{*}_{k} = -\left(\frac{2}{\Delta t}\mathbf{q}_{k} + \dot{\mathbf{q}}_{k}\right)$$

$$\ddot{\mathbf{q}}_{k+1} = \frac{4}{\Delta t^{2}}\mathbf{q}_{k+1} + \ddot{\mathbf{q}}^{*}_{k} \text{ , where } \ddot{\mathbf{q}}^{*}_{k} = -\left(\frac{4}{\Delta t^{2}}\mathbf{q}_{k} + \frac{4}{\Delta t}\dot{\mathbf{q}}_{k} + \ddot{\mathbf{q}}_{k}\right)$$

- Velocity and acceleration projections to impose constraints at velocity and acceleration level

The Kalman filter



Discrete-time Kalman filter

- First presented in 1960
- First practical application during the Project Apollo (NASA)

$$\text{System}\left\{\begin{array}{rcl} \mathbf{x}_k & = & \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}\mathbf{u}_{k-1} + \mathbf{w}_k \\ \mathbf{y}_k & = & \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \\ \mathbf{w} & \sim & N(0, \mathbf{\Sigma}^P) \\ \mathbf{v} & \sim & N(0, \mathbf{\Sigma}^S) \end{array}\right.$$

$$\text{Prediction}\left\{\begin{array}{rcl} \hat{\mathbf{x}}_k^- & = & \mathbf{F}\hat{\mathbf{x}}_{k-1}^+ + \mathbf{G}\mathbf{u}_{k-1} \\ \mathbf{P}_k^- & = & \mathbf{F}\mathbf{P}_{k-1}^+\mathbf{F}^\top + \mathbf{\Sigma}^P \end{array}\right.$$

$$\text{Correction}\left\{\begin{array}{rcl} \tilde{\mathbf{y}}_k & = & \mathbf{o}_k - \mathbf{H}\hat{\mathbf{x}}_k^- \\ \mathbf{\Sigma}_k & = & \mathbf{H}\mathbf{P}_k^-\mathbf{H}^\top + \mathbf{\Sigma}_k^S \\ \mathbf{K}_k & = & \mathbf{P}_k^-\mathbf{H}^\top\mathbf{\Sigma}_k^{-1} \\ \hat{\mathbf{x}}_k^+ & = & \hat{\mathbf{x}}_k^- + \mathbf{K}_k\tilde{\mathbf{y}}_k \\ \mathbf{P}_k^+ & = & (\mathbf{I}_g - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^- \end{array}\right.$$

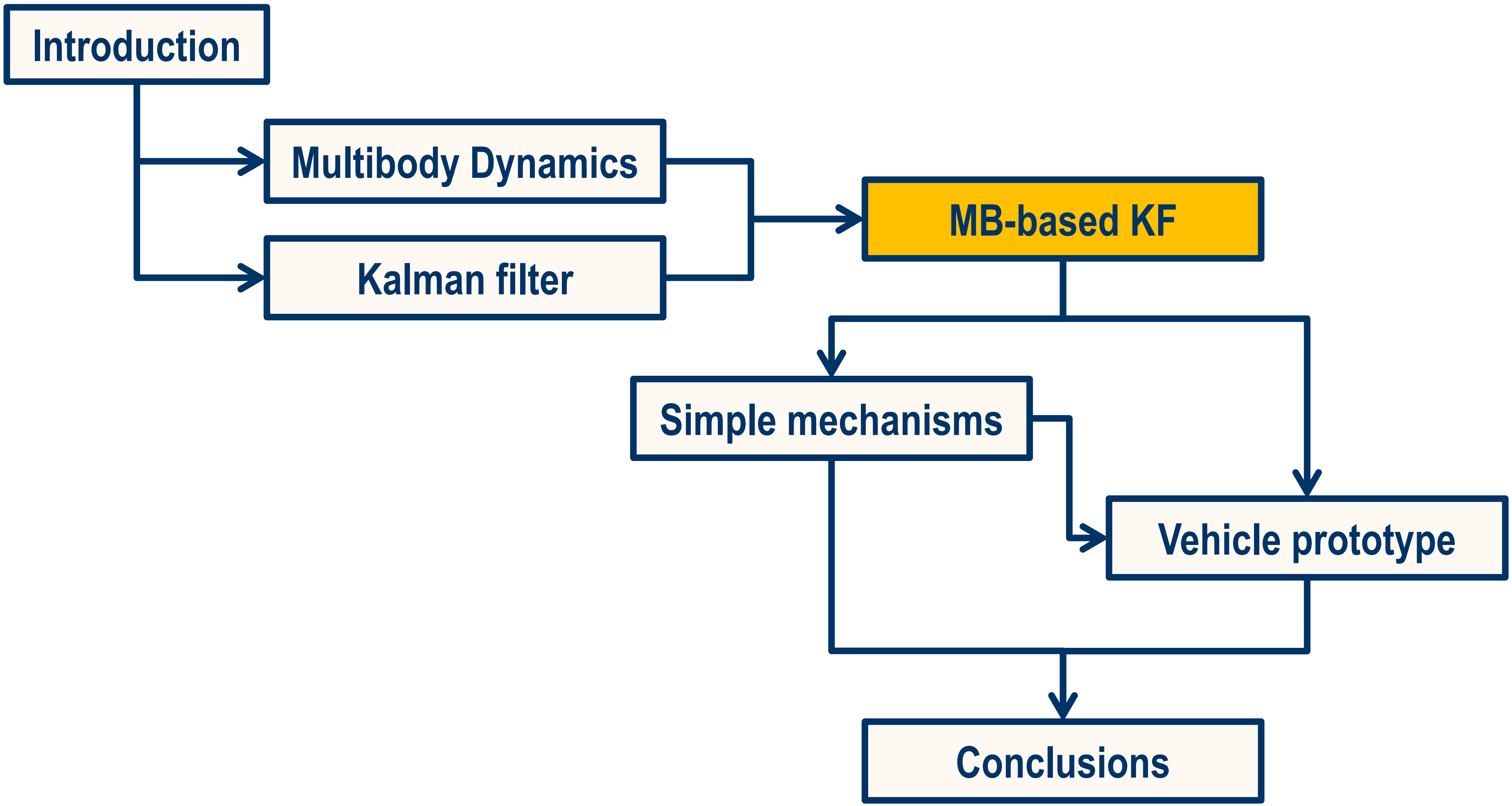
Continuous-time Kalman filter

- Presented in 1961 by Kalman and Bucy

$$\text{System}\begin{cases}\dot{\mathbf{x}} &= \mathbf{Ax}+\mathbf{Bu}+\mathbf{w}\\ \mathbf{y} &= \mathbf{Hx}+\mathbf{v}\\ \mathbf{w} &\sim N(0,\boldsymbol{\Sigma}_c^P)\\ \mathbf{v} &\sim N(0,\boldsymbol{\Sigma}^S)\end{cases}$$

$$\begin{aligned}\mathbf{K} &= \mathbf{PH}^\top\left[\boldsymbol{\Sigma}^S\right]^{-1}\\ \dot{\hat{\mathbf{x}}} &= \mathbf{A}\hat{\mathbf{x}}+\mathbf{Bu}+\mathbf{K}\left(\mathbf{y}-\mathbf{H}\hat{\mathbf{x}}\right)\\ \dot{\mathbf{P}} &= \mathbf{AP}+\mathbf{PA}^\top-\mathbf{PH}\left[\boldsymbol{\Sigma}^S\right]^{-1}\mathbf{HP}+\boldsymbol{\Sigma}_c^P\end{aligned}$$

Multibody-based Kalman filters



Multibody-based Kalman filters

- **Multibody system: second order, constrained, nonlinear equations**

$$\left\{\begin{array}{rcl} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi_q}^{\top}\boldsymbol{\lambda} & = & \mathbf{Q} \\ \mathbf{\Phi} & = & \mathbf{0} \end{array}\right.$$

- **Kalman filter: first order, unconstrained, linear equations**

$$\text{Prediction}\left\{\begin{array}{rcl} \hat{\mathbf{x}}_k^- & = & \mathbf{F}\hat{\mathbf{x}}_{k-1}^+ + \mathbf{G}\mathbf{u}_{k-1} \\ \mathbf{P}_k^- & = & \mathbf{F}\mathbf{P}_{k-1}^+\mathbf{F}^\top + \mathbf{\Sigma}^P \end{array}\right.$$

$$\text{Correction}\left\{\begin{array}{rcl} \tilde{\mathbf{y}}_k & = & \mathbf{o}_k - \mathbf{H}\hat{\mathbf{x}}_k^- \\ \mathbf{\Sigma}_k & = & \mathbf{H}\mathbf{P}_k^-\mathbf{H}^\top + \mathbf{\Sigma}_k^S \\ \mathbf{K}_k & = & \mathbf{P}_k^-\mathbf{H}^\top\mathbf{\Sigma}_k^{-1} \\ \hat{\mathbf{x}}_k^+ & = & \hat{\mathbf{x}}_k^- + \mathbf{K}_k\tilde{\mathbf{y}}_k \\ \mathbf{P}_k^+ & = & (\mathbf{I}_g - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^- \end{array}\right.$$

Adaptations of MB and KF equations

- **Variable duplication**

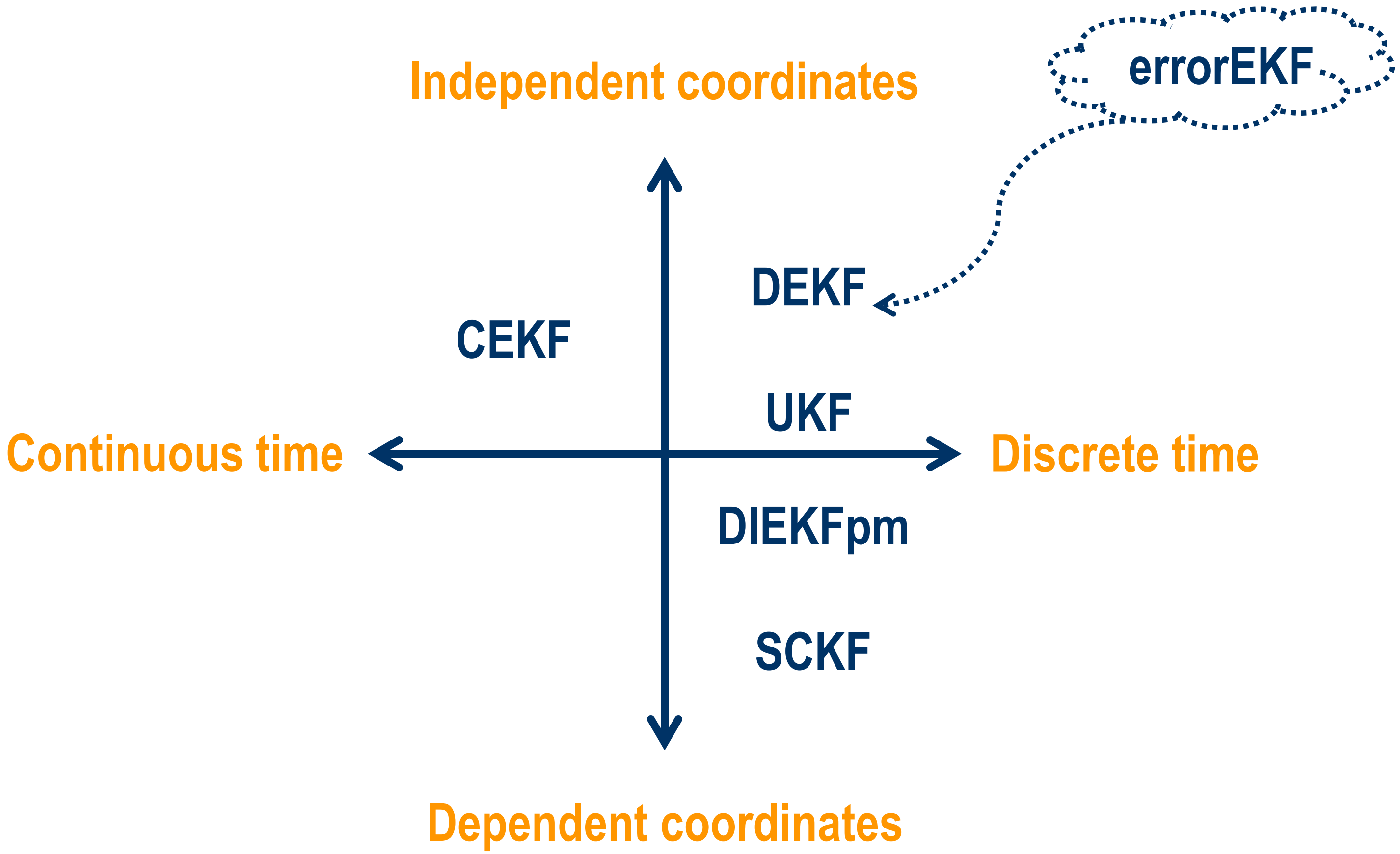
$$\mathbf{x} = \begin{bmatrix} \mathbf{z} \\ \dot{\mathbf{z}} \end{bmatrix}$$

- **Extended Kalman filter: linearization of plant model and measurement model at the estimated point**

$$\begin{cases} \hat{\mathbf{x}}_k^- & = & \mathbf{F}\hat{\mathbf{x}}_{k-1}^+ + \mathbf{G}\mathbf{u}_{k-1} \\ \mathbf{P}_k^- & = & \mathbf{F}\mathbf{P}_{k-1}^+\mathbf{F}^\top + \mathbf{\Sigma}^P \end{cases} \Rightarrow \begin{cases} \hat{\mathbf{x}}_k^- & = & \mathbf{f}\left(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}\right) \\ \mathbf{P}_k^- & = & \mathbf{f_x}\mathbf{P}_{k-1}^+\mathbf{f_x}^\top + \mathbf{\Sigma}^P \end{cases}$$

$$\begin{cases} \tilde{\mathbf{y}}_k & = & \mathbf{o}_k - \mathbf{H}\hat{\mathbf{x}}_k^- \\ \mathbf{\Sigma}_k & = & \mathbf{H}\mathbf{P}_k^-\mathbf{H}^\top + \mathbf{\Sigma}_k^S \\ \mathbf{K}_k & = & \mathbf{P}_k^-\mathbf{H}^\top\mathbf{\Sigma}_k^{-1} \\ \hat{\mathbf{x}}_k^+ & = & \hat{\mathbf{x}}_k^- + \mathbf{K}_k\tilde{\mathbf{y}}_k \\ \mathbf{P}_k^+ & = & (\mathbf{I}_g - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^- \end{cases} \Rightarrow \begin{cases} \tilde{\mathbf{y}}_k & = & \mathbf{o}_k - \mathbf{h}\left(\hat{\mathbf{x}}_k^-\right) \\ \mathbf{\Sigma}_k & = & \mathbf{h_x}\mathbf{P}_k^-\mathbf{h_x}^\top + \mathbf{\Sigma}_k^S \\ \mathbf{K}_k & = & \mathbf{P}_k^-\mathbf{h_x}^\top\mathbf{\Sigma}_k^{-1} \\ \hat{\mathbf{x}}_k^+ & = & \hat{\mathbf{x}}_k^- + \mathbf{K}_k\tilde{\mathbf{y}}_k \\ \mathbf{P}_k^+ & = & (\mathbf{I}_g - \mathbf{K}_k\mathbf{h_x})\mathbf{P}_k^- \end{cases}$$

Overview of the methods



CEKF: Continuous Extended Kalman Filter

- MB model in independent coordinates (Matrix R method)

$$\mathbf{x} = \left[\mathbf{z}^\top, \dot{\mathbf{z}}^\top\right]^\top$$

$$\begin{bmatrix} \dot{\mathbf{z}} \\ \ddot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \hat{\dot{\mathbf{z}}} \\ \bar{\mathbf{M}}^{-1}\bar{\mathbf{Q}} \end{bmatrix} \quad \Rightarrow \quad \dot{\mathbf{x}} = \mathfrak{f}\left(\hat{\mathbf{x}}, \mathbf{u}\right)$$

$$\mathfrak{F} = \frac{\partial \mathfrak{f}}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \frac{\partial\left(\bar{\mathbf{M}}^{-1}\bar{\mathbf{Q}}\right)}{\partial \mathbf{z}} & \frac{\partial\left(\bar{\mathbf{M}}^{-1}\bar{\mathbf{Q}}\right)}{\partial \dot{\mathbf{z}}} \end{bmatrix}$$

$$\begin{cases} \dot{\mathbf{z}} - \hat{\dot{\mathbf{z}}} + \mathbf{K}^{\mathbf{z}}\left(\mathbf{h}(\mathbf{x}) - \mathbf{o}\right) & = & \mathbf{0} \\ \bar{\mathbf{M}}\ddot{\mathbf{z}} - \bar{\mathbf{Q}} + \bar{\mathbf{M}}\mathbf{K}^{\dot{\mathbf{z}}}\left(\mathbf{h}(\mathbf{x}) - \mathbf{o}\right) & = & \mathbf{0} \end{cases}$$

$$\begin{cases} \hat{\dot{\mathbf{z}}}_{n+1} & = & \frac{2}{\Delta t}\hat{\mathbf{z}}_{n+1} - \left(\frac{2}{\Delta t}\hat{\mathbf{z}}_n + \hat{\dot{\mathbf{z}}}_n\right) \\ \hat{\ddot{\mathbf{z}}}_{n+1} & = & \frac{2}{\Delta t}\hat{\dot{\mathbf{z}}}_{n+1} - \left(\frac{2}{\Delta t}\hat{\dot{\mathbf{z}}}_n + \hat{\ddot{\mathbf{z}}}_n\right) \end{cases}$$

- MB, implicit integrator, and corrections solved at once

DEKF: Discrete Extended Kalman Filter

- MB model in independent coordinates
- Forward Euler integrator as the transition model
- Nonlinear model: extended Kalman filter

$$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_{k-1}) \quad \Rightarrow \begin{bmatrix} \hat{\mathbf{z}}_k \\ \hat{\dot{\mathbf{z}}}_k \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{z}}_{k-1} + \Delta t \hat{\dot{\mathbf{z}}}_{k-1} \\ \hat{\dot{\mathbf{z}}}_{k-1} + \Delta t \underbrace{\ddot{\mathbf{z}}_{k-1}}_{\mathrm{MB}} \end{bmatrix}$$

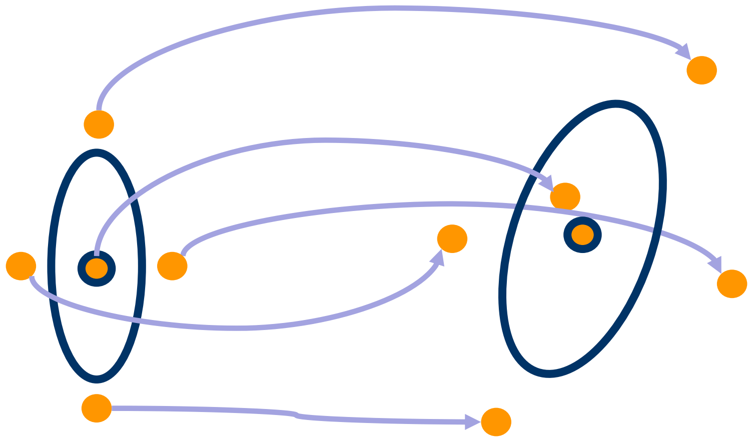
$$\mathbf{f}_{\mathbf{x}} \equiv \frac{\partial \mathbf{f}}{\partial \hat{\mathbf{x}}} = \begin{bmatrix} \mathbf{I} & \Delta t \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

$$\mathbf{P}_k^- = \mathbf{f}_{\mathbf{x}k-1} \mathbf{P}_{k-1}^+ \mathbf{f}_{\mathbf{x}k-1}^\top + \mathbf{\Sigma}_{k-1}^P$$

- Position and velocity problem have to be solved every time step

UKF: Unscented Kalman Filter

- Propagation of uncertainties through a set of deterministically chosen samples (Unscented transform)



- Any MB formulation and integrator can be used to propagate the state (TR and FE tested here)
- Better management of nonlinearities
- The Jacobian matrices are not needed

DIEKFpm: Iterated DEKF with perfect measurements

- MB in dependent coordinates: the MB constraints are imposed as sensor measurements

$$\hat{\mathbf{x}}^\top = \left\{ \hat{\mathbf{q}}^\top, \hat{\dot{\mathbf{q}}}^\top \right\}$$

$$\tilde{\mathbf{y}}_{k,i} = \begin{bmatrix} \mathbf{o}_k \\ \mathbf{0}_{2n\times 1} \end{bmatrix} - \mathbf{h}'(\hat{\mathbf{x}}_{k,i}) = \begin{bmatrix} \mathbf{o}_k - \mathbf{h}(\hat{\mathbf{x}}_{k,i}) \\ \mathbf{0}_{n\times 1} - \mathbf{\Phi}(\hat{\mathbf{x}}_{k,i}) \\ \mathbf{0}_{n\times 1} - \dot{\mathbf{\Phi}}(\hat{\mathbf{x}}_{k,i}) \end{bmatrix}$$

- The constraints are nonlinear, so the algorithm must iterate until fulfilling the constraints to the desired level
- All the measurements are applied only once!

SCKF: Smoothly constrained Kalman filter

- MB in dependent coordinates
- Position and velocity vectors are the states, propagated as if they were independent
- Application of actual measurements
- Iterative application of constraints as “weakened” measurements (several times)

$$\boldsymbol{\xi}_0 = \tilde{\alpha} \begin{bmatrix} \boldsymbol{\Phi}_{\hat{\mathbf{x}}} \\ \dot{\boldsymbol{\Phi}}_{\hat{\mathbf{x}}} \end{bmatrix} \mathbf{P}^{+}_{k,0} \begin{bmatrix} \boldsymbol{\Phi}_{\hat{\mathbf{x}}} \\ \dot{\boldsymbol{\Phi}}_{\hat{\mathbf{x}}} \end{bmatrix}^{\top}$$

$$\boldsymbol{\xi}_{i+1} = \boldsymbol{\xi}_i e^{-\tilde{\beta}}$$

errorEKF: Error-state EKF

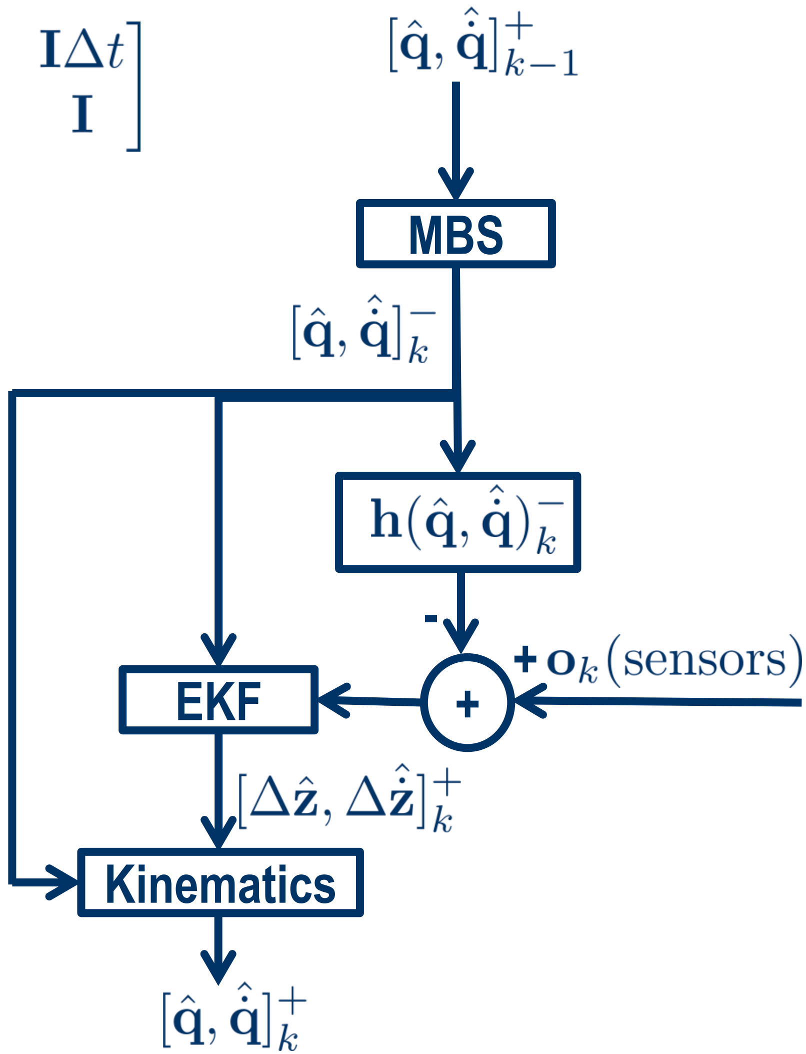
$$\mathbf{x} = \begin{bmatrix} \Delta\mathbf{z}^\top & \Delta\dot{\mathbf{z}}^\top \end{bmatrix}^\top$$

$$\begin{cases} \Delta\mathbf{z}_k & = & \Delta\mathbf{z}_{k-1} + \Delta\dot{\mathbf{z}}_{k-1}\Delta t \\ \Delta\dot{\mathbf{z}}_k & = & \Delta\dot{\mathbf{z}}_{k-1} \end{cases} \Rightarrow \mathbf{f_x} = \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

$$\text{Pred.} \begin{cases} \hat{\mathbf{x}}_k^- & = & \mathbf{0} \\ \mathbf{P}_k^- & = & \mathbf{f_x}\mathbf{P}_{k-1}^+ \mathbf{f_x}^\top + \mathbf{\Sigma}_{k-1}^P \end{cases}$$

$$\text{Corr. (KF)} \begin{cases} \tilde{\mathbf{y}}_k & = & \mathbf{o}_k - \mathbf{h}(\hat{\mathbf{q}}_k^-, \hat{\dot{\mathbf{q}}}_k^-) \\ \mathbf{\Sigma}_k & = & \mathbf{h_x}_k \mathbf{P}_k^- \mathbf{h_x}_k^\top + \mathbf{\Sigma}_k^S \\ \mathbf{K}_k & = & \mathbf{P}_k^- \mathbf{h_x}_k^\top \mathbf{\Sigma}_k^{-1} \\ \hat{\mathbf{x}}_k^+ & = & \mathbf{0} + \mathbf{K}_k \tilde{\mathbf{y}}_k \\ \mathbf{P}_k^+ & = & (\mathbf{I}_g - \mathbf{K}_k \mathbf{h_x}_k)\mathbf{P}_k^- \end{cases}$$

$$\text{Corr. (MB)} \begin{cases} [\Delta\hat{\mathbf{z}}, \Delta\hat{\dot{\mathbf{z}}}]_k^+ & \rightarrow [\Delta\hat{\mathbf{q}}, \Delta\hat{\dot{\mathbf{q}}}]_k^+ \\ \hat{\mathbf{q}}_k^+ & = \hat{\mathbf{q}}_k^- + \Delta\hat{\mathbf{q}}_k^+ \\ \hat{\dot{\mathbf{q}}}_k^+ & = \hat{\dot{\mathbf{q}}}_k^- + \Delta\hat{\dot{\mathbf{q}}}_k^+ \\ \hat{\mathbf{x}}_k^+ & = \mathbf{0} \end{cases}$$



Sensor models

- MB provides a general framework to develop sensor models
- Methods in dependent coordinates: $\mathbf{h_q}$,$\mathbf{h_{\dot{q}}}$
- Methods in independent coordinates: $\mathbf{h_z}$,$\mathbf{h_{\dot{z}}}$

$$\mathbf{h_z} \equiv \frac{\partial \mathbf{h}(\mathbf{q},\dot{\mathbf{q}})}{\partial \mathbf{z}} = \mathbf{h_q}\frac{\partial \mathbf{q}}{\partial \mathbf{z}} + \mathbf{h_{\dot{q}}}\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{z}}$$

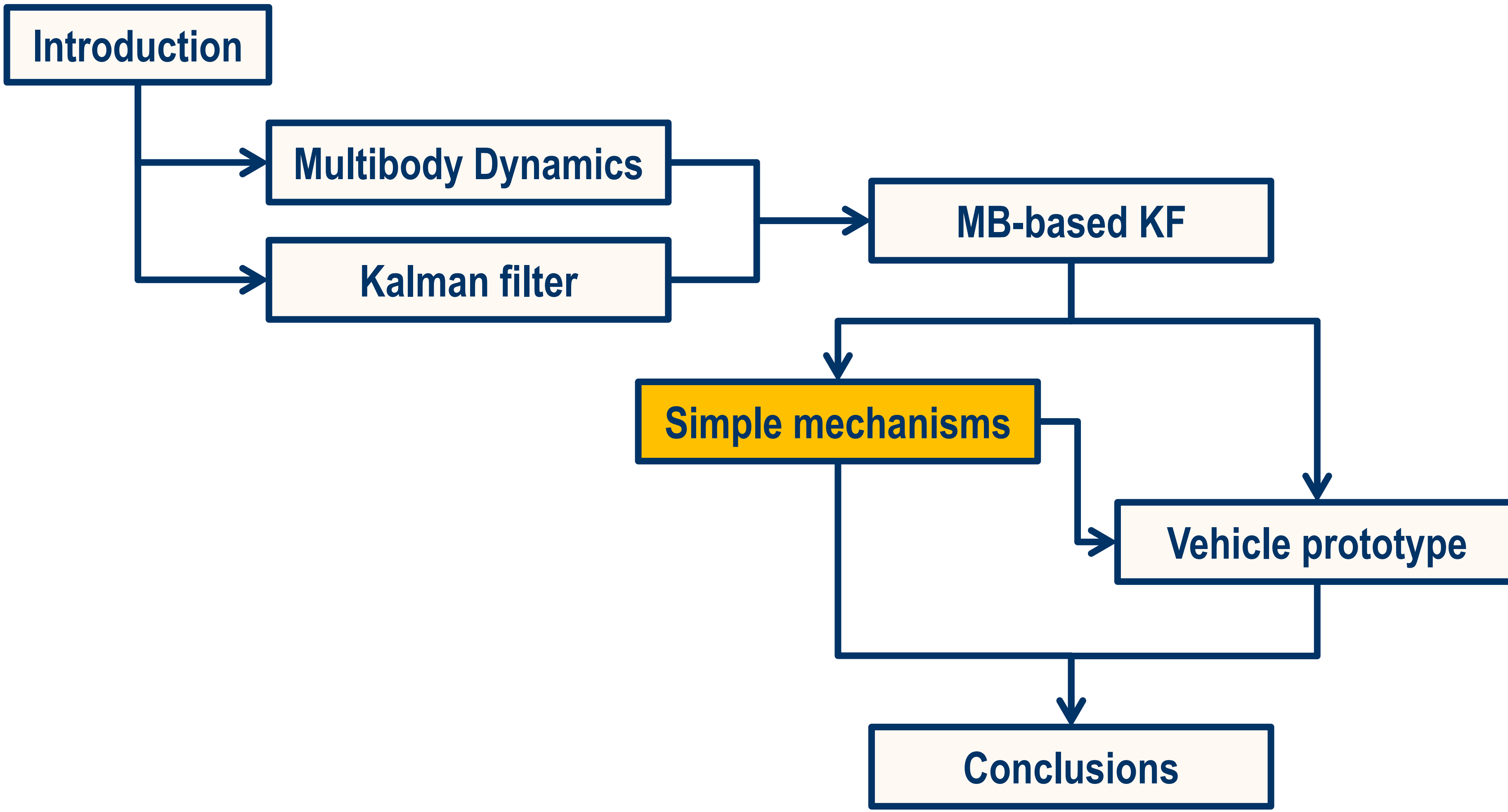
$$\mathbf{h_{\dot{z}}} \equiv \frac{\partial \mathbf{h}(\mathbf{q},\dot{\mathbf{q}})}{\partial \dot{\mathbf{z}}} = \mathbf{h_q}\underbrace{\frac{\partial \mathbf{q}}{\partial \dot{\mathbf{z}}}}_{\to\, 0} + \mathbf{h_{\dot{q}}}\frac{\partial \dot{\mathbf{q}}}{\partial \dot{\mathbf{z}}}$$

$$\Leftarrow \begin{cases} \dfrac{\partial \mathbf{q}}{\partial \mathbf{z}} = \dfrac{\partial \dot{\mathbf{q}}}{\partial \dot{\mathbf{z}}} = \mathbf{R} \\ \dfrac{\partial \dot{\mathbf{q}}}{\partial \mathbf{z}} = \dfrac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}}\dfrac{\partial \mathbf{q}}{\partial \mathbf{z}} = -\mathbf{S}\dot{\mathbf{\Phi}}_{\mathbf{q}}\mathbf{R} \end{cases}$$

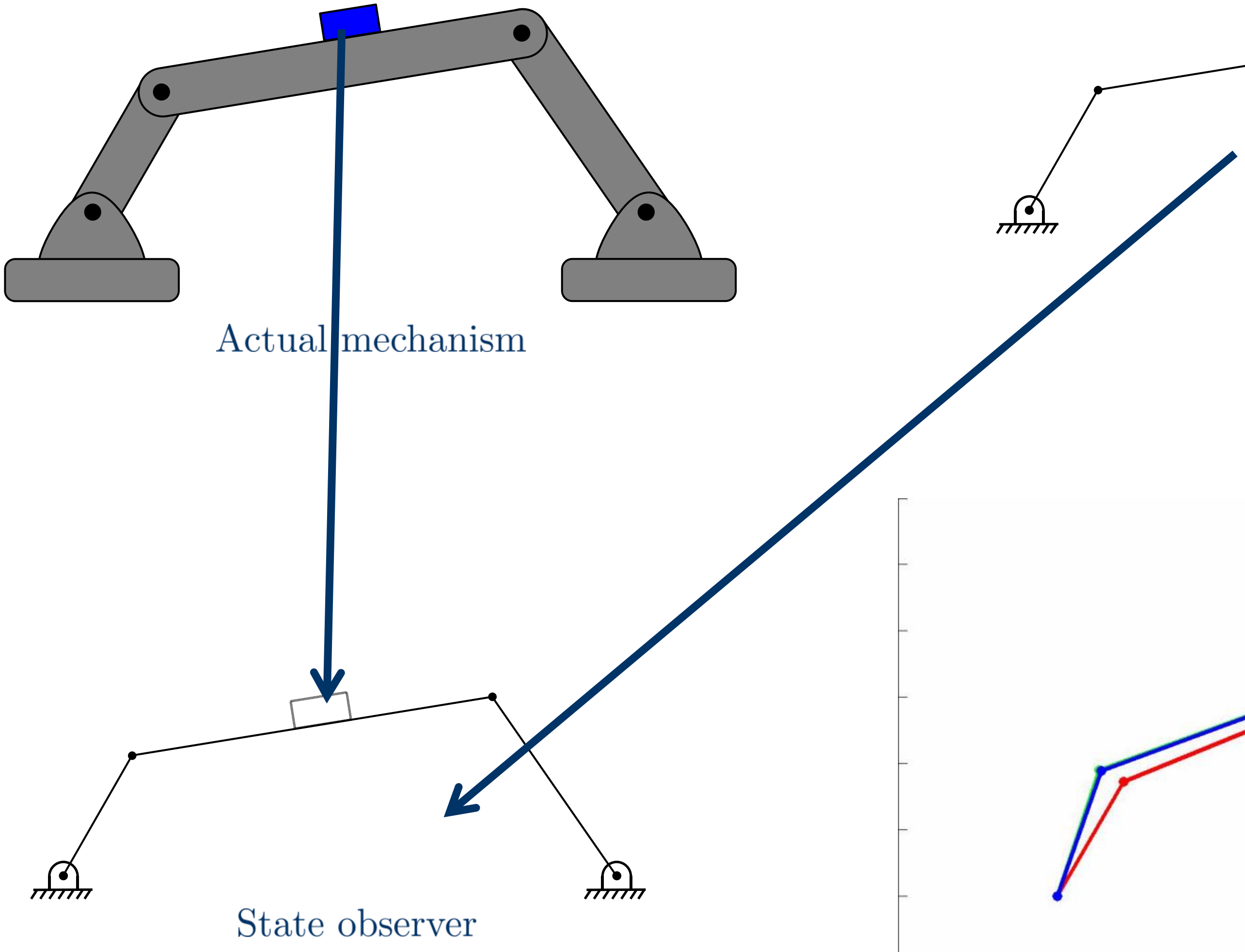
$$\dot{\mathbf{z}} = \mathbf{D}\dot{\mathbf{q}}$$

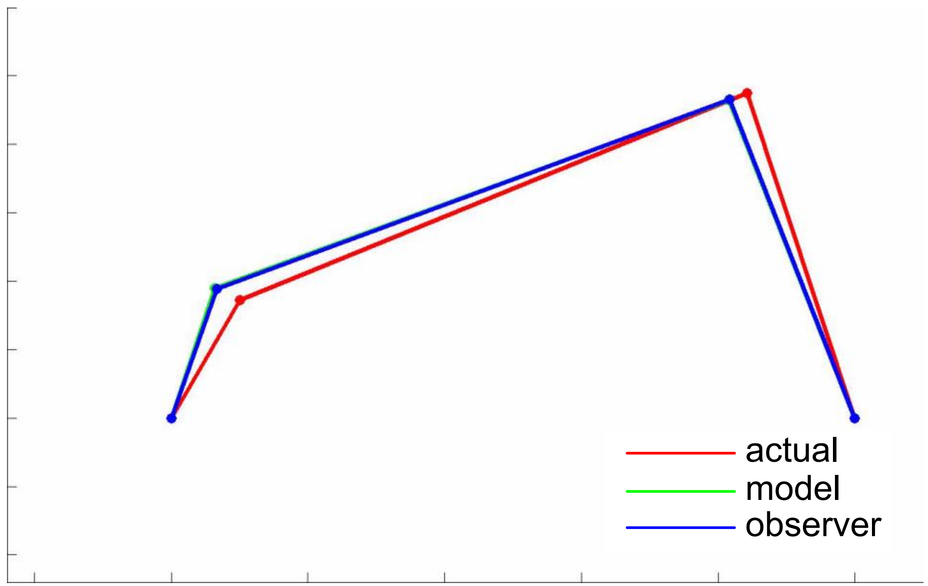
$$\begin{bmatrix} \mathbf{\Phi_q} \\ \mathbf{D} \end{bmatrix}^{-1} \equiv \begin{bmatrix} \mathbf{S} & \mathbf{R} \end{bmatrix}$$

Application to simple mechanisms



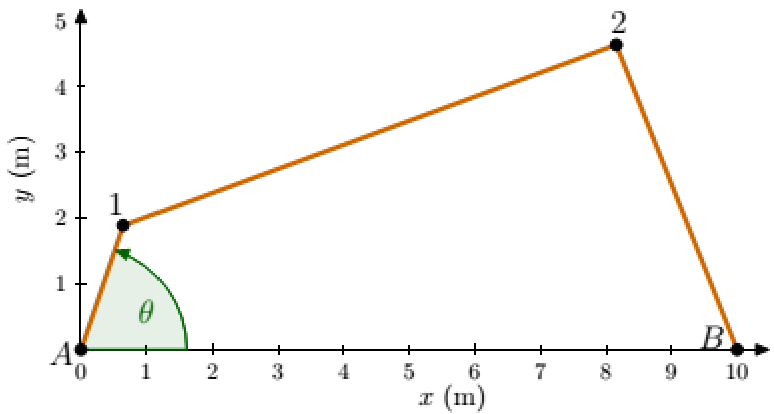
Methodology

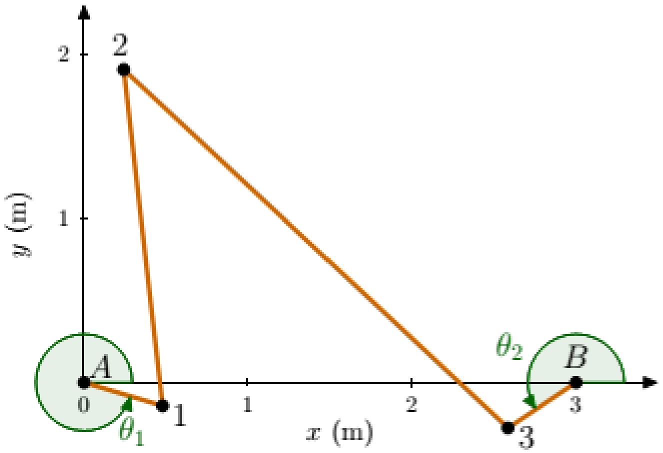




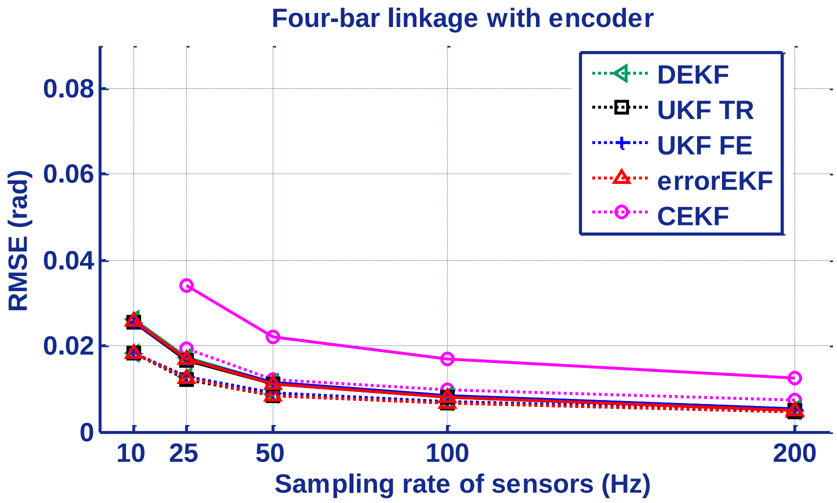
Tests

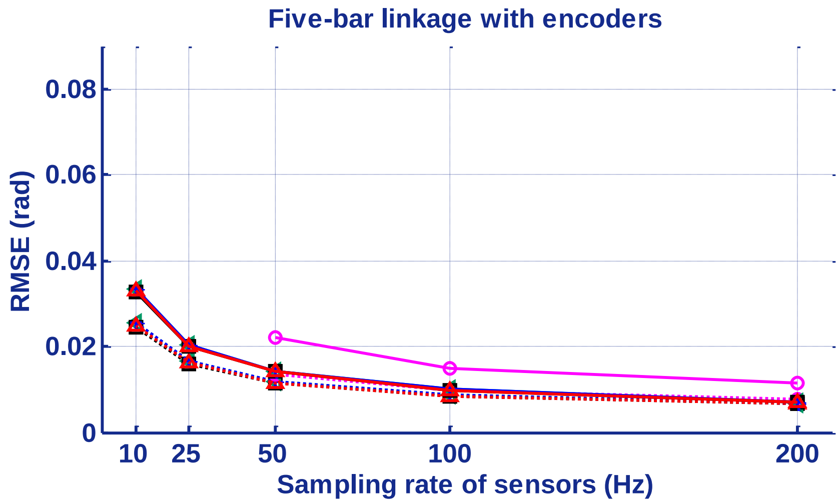
- **Two mechanisms**
  - Four-bar linkage
  - Five-bar linkages
- **Two levels of modeling error**
  - Gravity, Initial position
- **Three sensor configurations**
  - Encoders
  - Gyros on couplers
  - Gyros on cranks
- **Sampling rates from 10 to 200 Hz.**
- **Plant noise adjusted from innovation sequence**





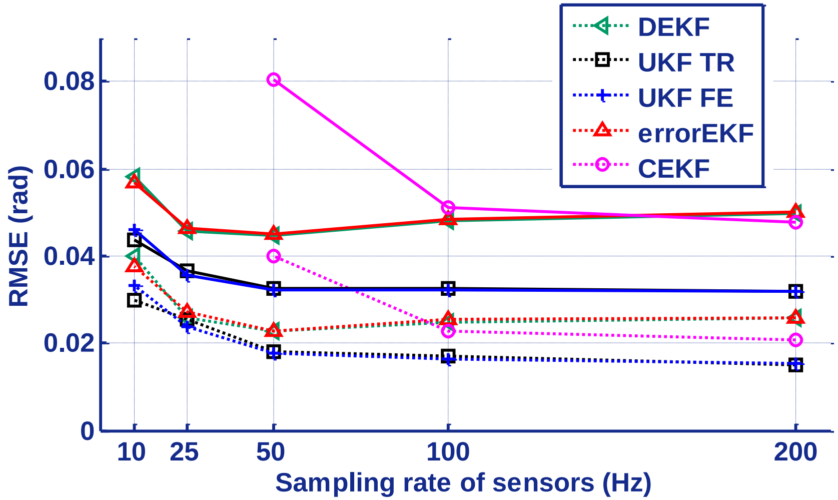
Tests with encoders

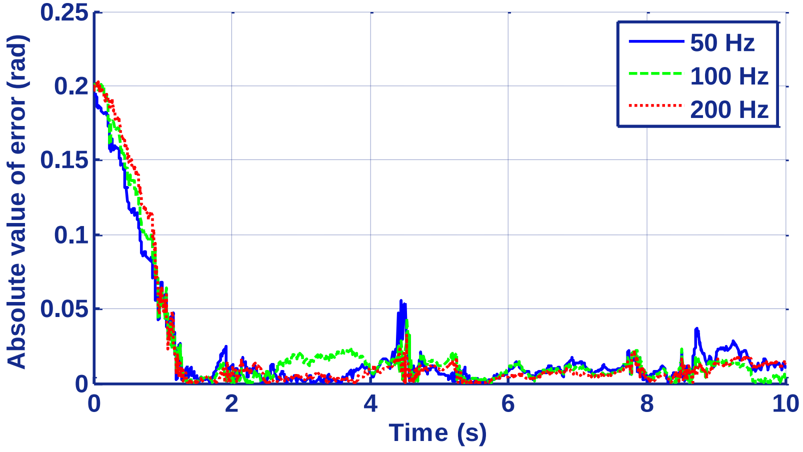


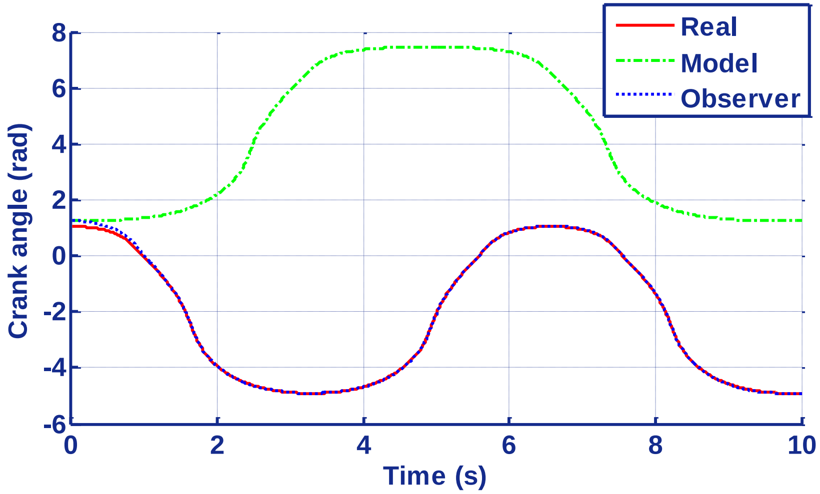


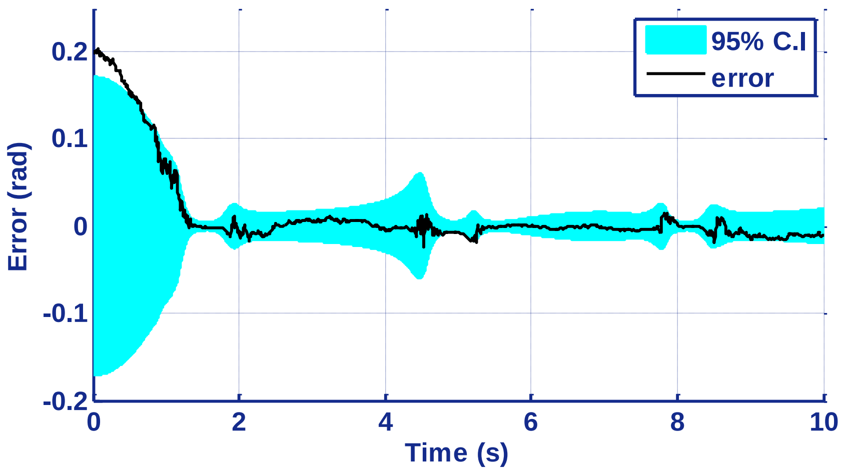
- **The noise of the encoders is π/180 ≈ 0.0175 rad**
- **Errors:**
  - 0.5 m/s^2 gravity acceleration, π/32 rad initial position error
  - 1 m/s^2 gravity acceleration, π/16 rad initial position error

Four-bar mechanism with gyro on the coupler

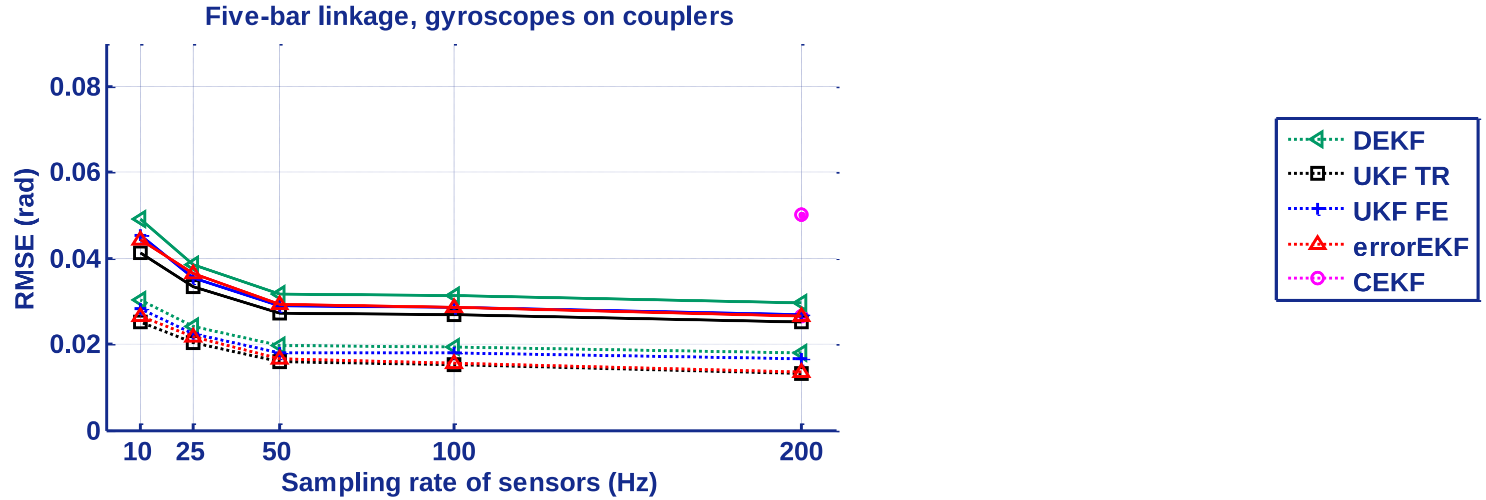


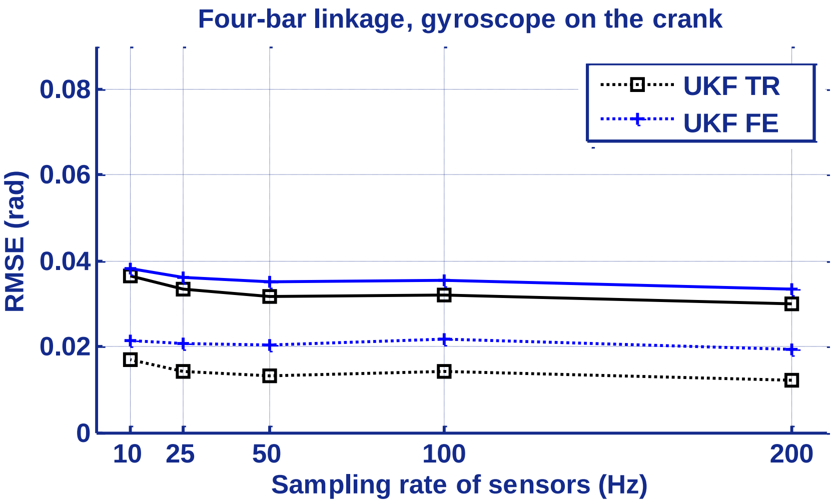


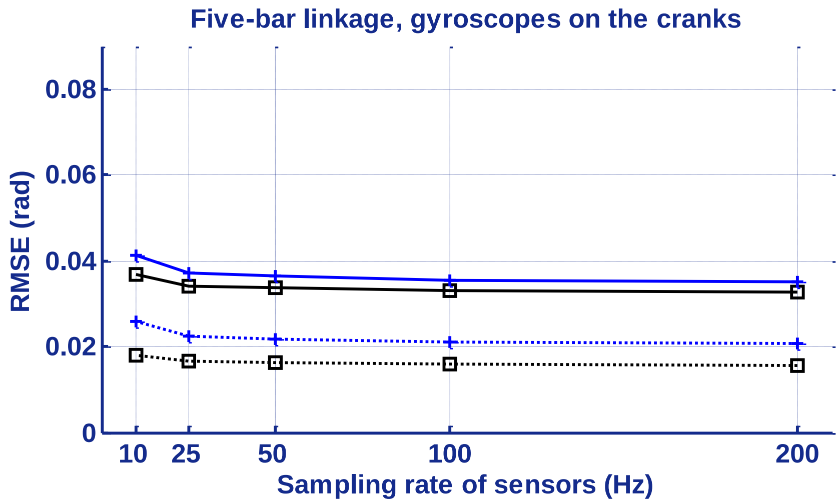




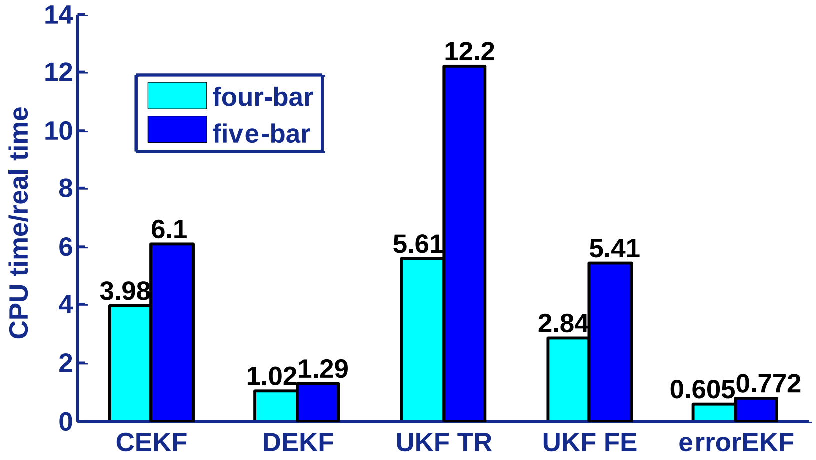
Other tests with gyroscopes





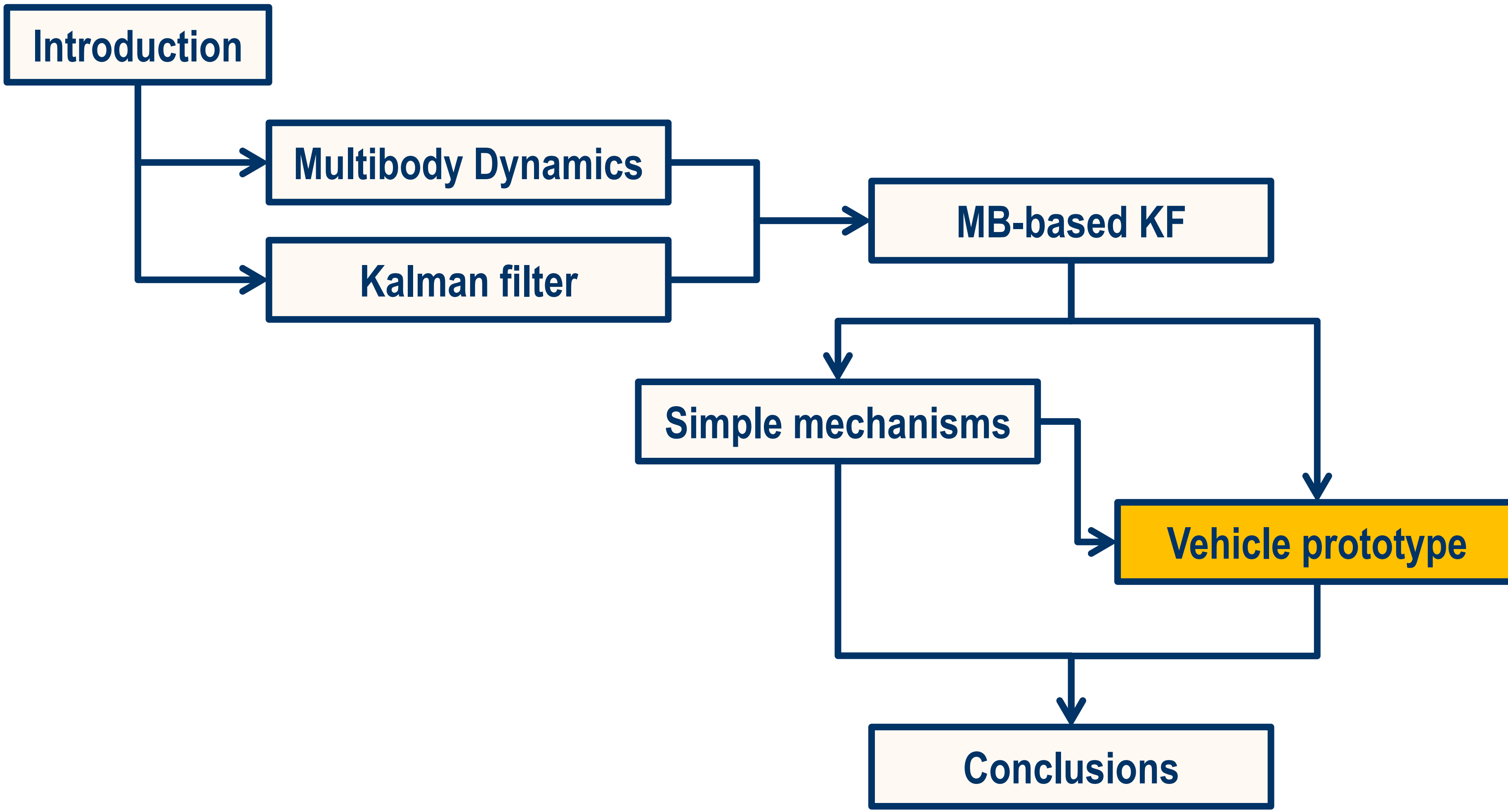


Computational cost

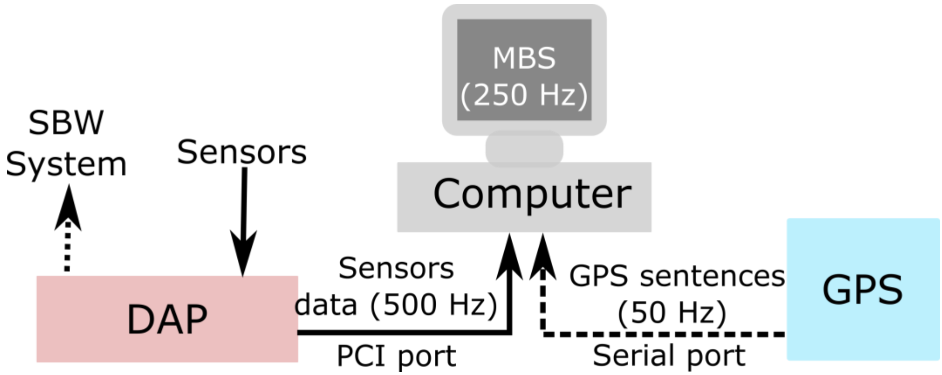


- Benchmark in Matlab®: CPU time only as a rough comparison
- The UKF TR is the slowest
- The errorEKF is the fastest
- UKF FE much more efficient than UKF TR
- Computational cost of UKFs increase faster than that of EKFs

Application to a vehicle prototype



The prototype



Full sized SBW vehicle

- Automatic gearbox
- Rear Wheel Drive
- Open differential
- Disk brakes
- Independent suspensions
- On-board PC (Intel Core 2 Duo @ 3,16 GHz, 2Gb RAM)

Measured Magnitude	Sensor
Vehicle acceleration (X,Y,Z)	Accelerometers
Vehicle angular rates (X,Y,Z)	Gyroscopes
Wheel rotation angles	Hall-effect sensors
Brake line pressure	Pressure sensor
Steering wheel and steer angles	Encoders
Real wheel torque	Wheel torque sensor
Position, speed and course	GPS receiver

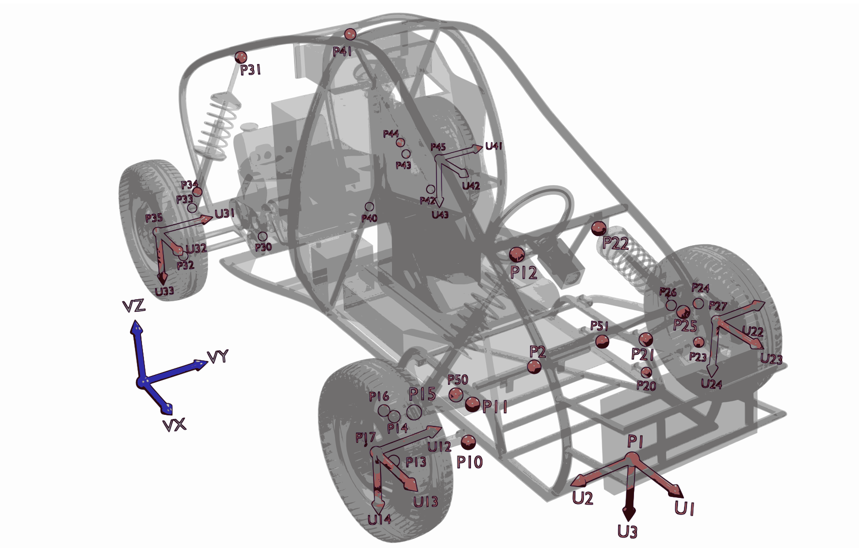
The multibody model

Index-3 augmented Lagrangian formulation with velocity and acceleration projections

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}^{\top}\alpha\mathbf{\Phi} + \mathbf{\Phi}_{\mathbf{q}}^{\top}\boldsymbol{\lambda}^{*} = \mathbf{Q}, \text{ with}$$

$$\boldsymbol{\lambda}_{i+1}^{*} = \boldsymbol{\lambda}_{i}^{*} + \alpha\mathbf{\Phi}_{i+1} \qquad i = 0, 1, 2, \ldots$$

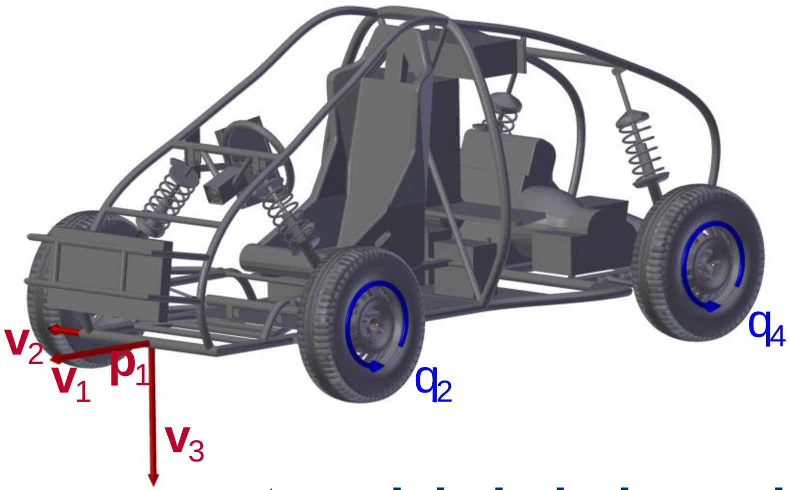
- Mainly modeled with natural coordinates
- Wheels modeled with relative angles
- Trapezoidal rule as the integrator



- 18 rigid bodies
- 169 coordinates
- 14 degrees of freedom
- Kinematically guided steering
- TMeasy tire model
- Inputs:
  - Rear wheel torque
  - Steering position
  - Brake pressure

- States: errors in all the DOFs except the suspensions

$$\mathbf{x} = \begin{bmatrix} \Delta r_{1x} & \Delta r_{1y} & \Delta r_{1z} & \Delta\psi & \Delta v_{1z} & \Delta v_{2z} & \Delta\theta_1 & \Delta\theta_2 & \Delta\theta_3 & \Delta\theta_4 & \dots \\ \Delta\dot{r}_{1x} & \Delta\dot{r}_{1y} & \Delta\dot{r}_{1z} & \Delta\dot{\psi} & \Delta\dot{v}_{1z} & \Delta\dot{v}_{2z} & \Delta\dot{\theta}_1 & \Delta\dot{\theta}_2 & \Delta\dot{\theta}_3 & \Delta\dot{\theta}_4 & \end{bmatrix}^\top$$



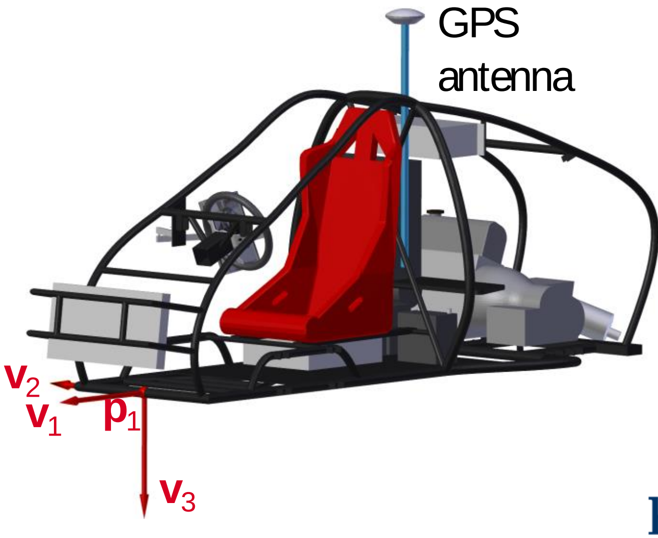
- Jacobian of measurement models in independent coordinates

$$\mathbf{h_z} \equiv \frac{\partial \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{z}} = \mathbf{h_q}\frac{\partial \mathbf{q}}{\partial \mathbf{z}} + \mathbf{h_{\dot{q}}}\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{z}}$$

$$\mathbf{h_{\dot{z}}} \equiv \frac{\partial \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{z}}} = \mathbf{h_q}\underbrace{\frac{\partial \mathbf{q}}{\partial \dot{\mathbf{z}}}}_{0} + \mathbf{h_{\dot{q}}}\frac{\partial \dot{\mathbf{q}}}{\partial \dot{\mathbf{z}}}$$

$$\mathbf{h_x} = \begin{bmatrix} \mathbf{h_z} \\ \mathbf{h_{\dot{z}}} \end{bmatrix}$$

Sensor models: GPS positioning



$$\mathbf{q} = \left[\dots \mathbf{r}_1^\top \quad \mathbf{v}_1^\top \quad \mathbf{v}_2^\top \quad \mathbf{v}_3^\top \dots \right]^\top$$

$$\dot{\mathbf{q}} = \left[\dots \dot{\mathbf{r}}_1^\top \quad \dot{\mathbf{v}}_1^\top \quad \dot{\mathbf{v}}_2^\top \quad \dot{\mathbf{v}}_3^\top \dots \right]^\top$$

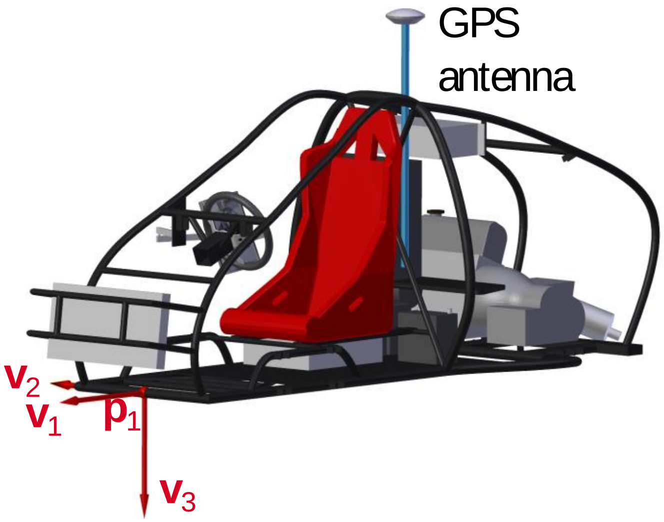
$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{r}_{GPS} = \mathbf{r}_1 + K_1 \mathbf{v}_1 + K_2 \mathbf{v}_2 + K_3 \mathbf{v}_3$$

$$\frac{\partial \mathbf{r}_{GPS}}{\partial \mathbf{q}} = \left[\dots \mathbf{I}_3 \quad \cancel{K_1 \mathbf{I}_3 \quad K_2 \mathbf{I}_3 \quad K_3 \mathbf{I}_3} \dots \right]$$

$$\frac{\partial \mathbf{r}_{GPS}}{\partial \mathbf{q}} = \left[\dots \mathbf{I}_3 \quad \mathbf{0}_{3\times3} \quad \mathbf{0}_{3\times3} \quad \mathbf{0}_{3\times3} \dots \right]$$

$$\frac{\partial \mathbf{r}_{GPS}}{\partial \dot{\mathbf{q}}} = \left[\dots \mathbf{0}_{3\times3} \quad \mathbf{0}_{3\times3} \quad \mathbf{0}_{3\times3} \quad \mathbf{0}_{3\times3} \dots \right]$$

Sensor models: GPS velocity



$$\mathbf{q} = \left[\ldots \mathbf{r}_1^\top \quad \mathbf{v}_1^\top \quad \mathbf{v}_2^\top \quad \mathbf{v}_3^\top \ldots \right]^\top$$

$$\dot{\mathbf{q}} = \left[\ldots \dot{\mathbf{r}}_1^\top \quad \dot{\mathbf{v}}_1^\top \quad \dot{\mathbf{v}}_2^\top \quad \dot{\mathbf{v}}_3^\top \ldots \right]^\top$$

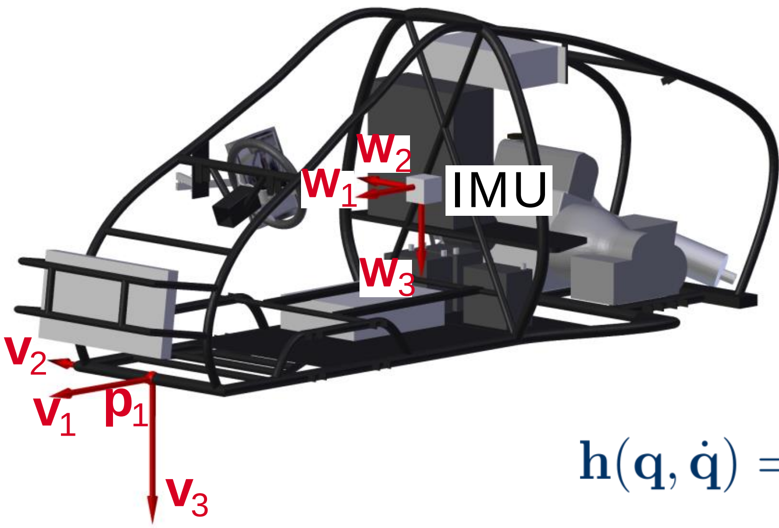
$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{r}}_{GPS} = \begin{bmatrix} \dot{r}_{1x} \\ \dot{r}_{1y} \end{bmatrix} + K_1 \begin{bmatrix} \dot{v}_{1x} \\ \dot{v}_{1y} \end{bmatrix} + K_2 \begin{bmatrix} \dot{v}_{2x} \\ \dot{v}_{2y} \end{bmatrix} + K_3 \begin{bmatrix} \dot{v}_{3x} \\ \dot{v}_{3y} \end{bmatrix}$$

$$\frac{\partial \mathbf{r}_{GPS}}{\partial \mathbf{q}} = \left[\ldots \mathbf{0}_{2\times3} \quad \mathbf{0}_{2\times3} \quad \mathbf{0}_{2\times3} \quad \mathbf{0}_{2\times3} \ldots \right]$$

$$\frac{\partial \mathbf{r}_{GPS}}{\partial \dot{\mathbf{q}}} = \begin{bmatrix} \ldots 1 & 0 & 0 & K_1 & 0 & 0 & K_2 & 0 & 0 & K_3 & 0 & 0 \ldots \\ \ldots 0 & 1 & 0 & 0 & K_1 & 0 & 0 & K_2 & 0 & 0 & K_3 & 0 \ldots \end{bmatrix}$$

GPS provides also the course over ground, which is used as a yaw measurement

Sensor models: IMU, angular rate sensors



$$\mathbf{q} = \begin{bmatrix} \mathbf{r}_1^\top & \mathbf{v}_1^\top & \mathbf{v}_2^\top & \mathbf{v}_3^\top \end{bmatrix}^\top$$

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{\mathbf{r}}_1^\top & \dot{\mathbf{v}}_1^\top & \dot{\mathbf{v}}_2^\top & \dot{\mathbf{v}}_3^\top \end{bmatrix}^\top$$

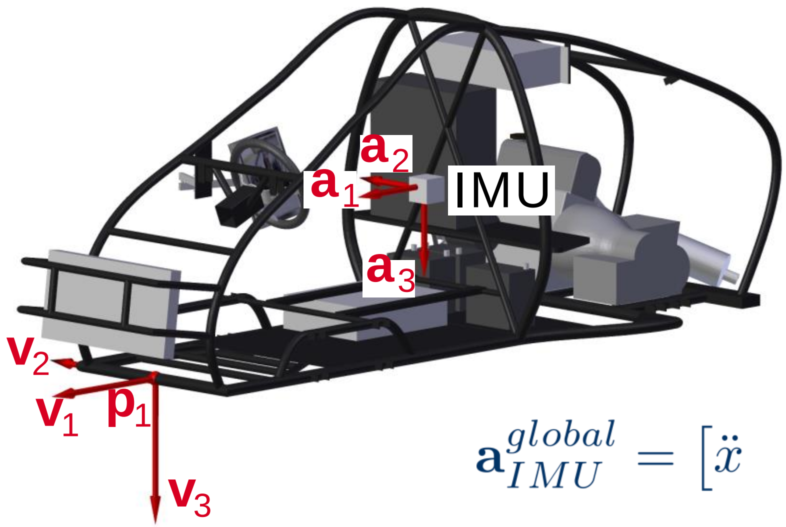
$\mathbf{\Psi}$ contains the axes of the IMU in the chassis frame

$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\omega}_{IMU} = \underbrace{\mathbf{\Psi}^\top}_{\text{rot. matrix}} \underbrace{\begin{bmatrix} \dot{\mathbf{v}}_2^\top \mathbf{v}_3 & \dot{\mathbf{v}}_3^\top \mathbf{v}_1 & \dot{\mathbf{v}}_1^\top \mathbf{v}_2 \end{bmatrix}}_{\text{ang. rate chassis frame}}$$

$$\frac{\partial \boldsymbol{\omega}_{IMU}}{\partial \mathbf{q}} = \mathbf{\Psi}^\top \begin{bmatrix} \ldots 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dot{v}_{2x} & \dot{v}_{2y} & \dot{v}_{2z} \ldots \\ \ldots 0 & 0 & 0 & \dot{v}_{3x} & \dot{v}_{3y} & \dot{v}_{3z} & 0 & 0 & 0 & 0 & 0 & 0 \ldots \\ \ldots 0 & 0 & 0 & 0 & 0 & 0 & \dot{v}_{1x} & \dot{v}_{1y} & \dot{v}_{1z} & 0 & 0 & 0 \ldots \end{bmatrix}$$

$$\frac{\partial \boldsymbol{\omega}_{IMU}}{\partial \dot{\mathbf{q}}} = \mathbf{\Psi}^\top \begin{bmatrix} \ldots 0 & 0 & 0 & 0 & 0 & 0 & v_{3x} & v_{3y} & v_{3z} & 0 & 0 & 0 \ldots \\ \ldots 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_{1x} & v_{1y} & v_{1z} \ldots \\ \ldots 0 & 0 & 0 & v_{2x} & v_{2y} & v_{2z} & 0 & 0 & 0 & 0 & 0 & 0 \ldots \end{bmatrix}$$

Sensor models: IMU, accelerometers



$$\mathbf{q} = \left[\ldots \mathbf{r}_1^\top \quad \mathbf{v}_1^\top \quad \mathbf{v}_2^\top \quad \mathbf{v}_3^\top \ldots \right]^\top$$

$$\dot{\mathbf{q}} = \left[\ldots \dot{\mathbf{r}}_1^\top \quad \dot{\mathbf{v}}_1^\top \quad \dot{\mathbf{v}}_2^\top \quad \dot{\mathbf{v}}_3^\top \ldots \right]^\top$$

$$\ddot{\mathbf{q}} = \left[\ldots \ddot{\mathbf{r}}_1^\top \quad \ddot{\mathbf{v}}_1^\top \quad \ddot{\mathbf{v}}_2^\top \quad \ddot{\mathbf{v}}_3^\top \ldots \right]^\top$$

$$\mathbf{\Psi}_{CH} = \left[\mathbf{v}_1 \quad \mathbf{v}_2 \quad \mathbf{v}_3 \right]$$

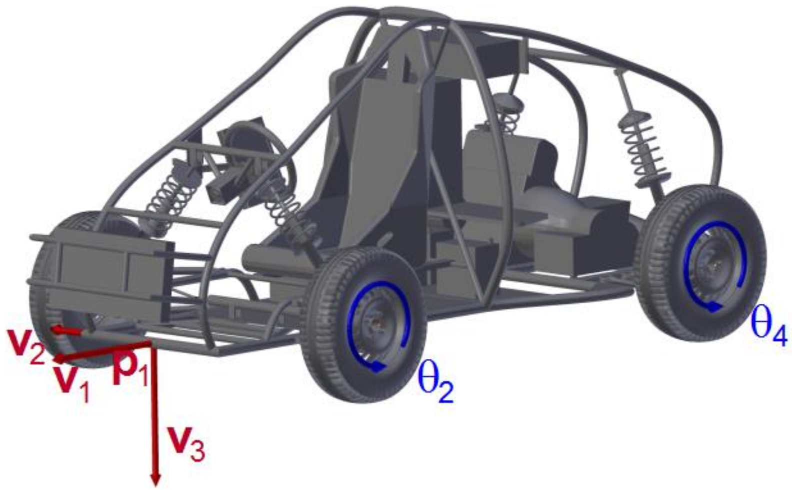
$$\mathbf{a}_{IMU}^{global} = \left[\ddot{x} \quad \ddot{y} \quad \ddot{z} \right]^\top = \ddot{\mathbf{r}}_1 + K_1 \ddot{\mathbf{v}}_1 + K_2 \ddot{\mathbf{v}}_2 + K_3 \ddot{\mathbf{v}}_3 - \mathbf{g}$$

$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{a}_{IMU}^{local} = \mathbf{\Psi}^\top \mathbf{\Psi}_{CH}^\top \mathbf{a}_{IMU}^{global}$$

$$\frac{\partial \mathbf{a}_{IMU}^{local}}{\partial \mathbf{q}} = \mathbf{\Psi}^\top \begin{bmatrix} \ldots 0 & 0 & 0 & \ddot{x} & \ddot{y} & \ddot{z} & 0 & 0 & 0 & 0 & 0 & 0 \ldots \\ \ldots 0 & 0 & 0 & 0 & 0 & 0 & \ddot{x} & \ddot{y} & \ddot{z} & 0 & 0 & 0 \ldots \\ \ldots 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ddot{x} & \ddot{y} & \ddot{z} \ldots \end{bmatrix}$$

$$\frac{\partial \mathbf{a}_{IMU}^{local}}{\partial \dot{\mathbf{q}}} = \left[\ldots \mathbf{0}_{3\times3} \quad \mathbf{0}_{3\times3} \quad \mathbf{0}_{3\times3} \quad \mathbf{0}_{3\times3} \ldots \right]$$

Sensor models: Hall-effect sensors



$$\mathbf{q} = \left[\ldots \theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \ldots \right]^\top$$

$$\dot{\mathbf{q}} = \left[\ldots \dot{\theta}_1 \quad \dot{\theta}_2 \quad \dot{\theta}_3 \quad \dot{\theta}_4 \ldots \right]^\top$$

$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \left[\theta_1 \quad \theta_2 \quad \theta_3 \quad \theta_4 \right]^\top$$

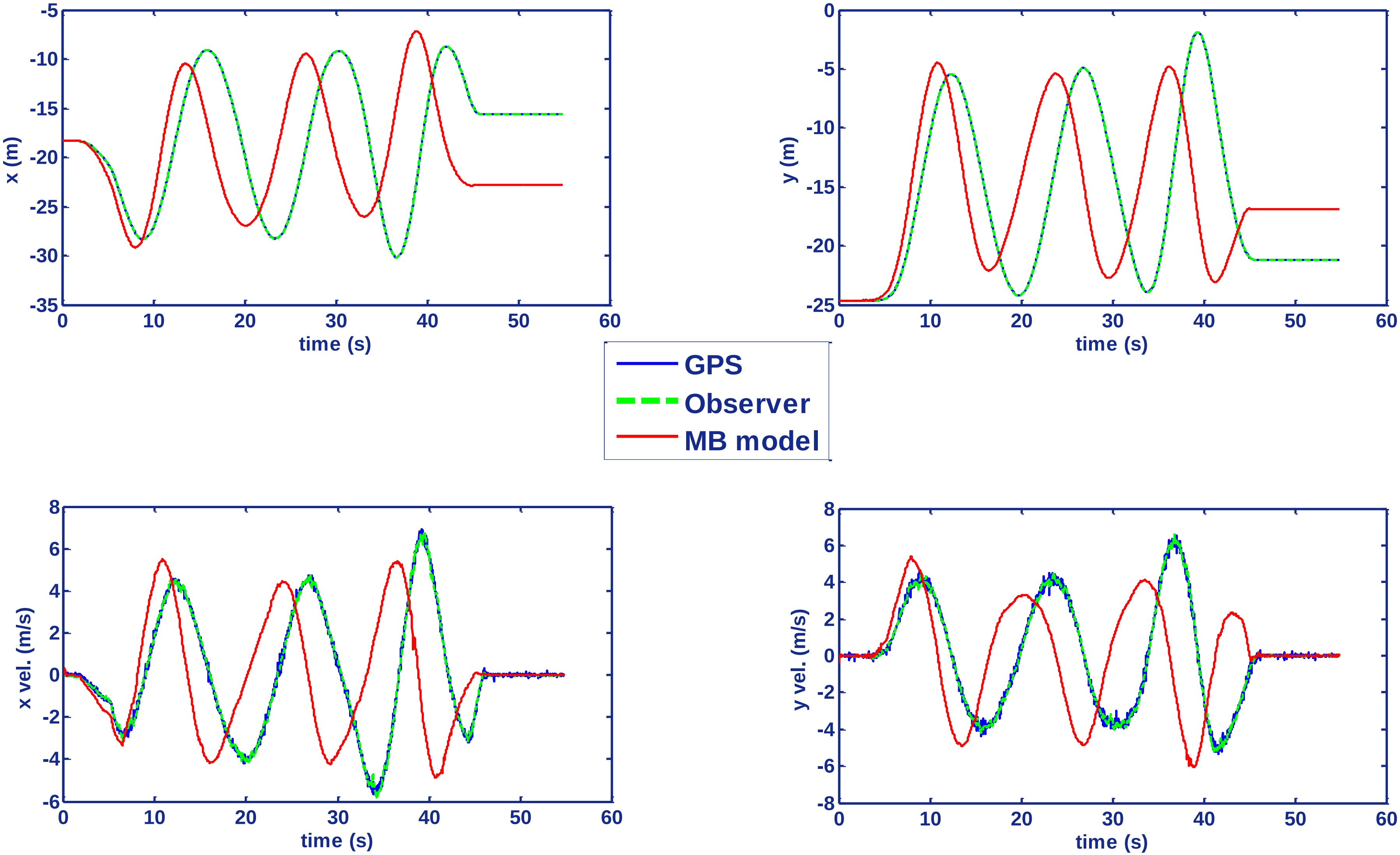
$$\mathbf{h}_\mathbf{q} = \left[\ldots \mathbf{I}_4 \ldots \right]$$

$$\mathbf{h}_{\dot{\mathbf{q}}} = \left[\ldots \mathbf{0}_{4\times 4} \ldots \right]$$

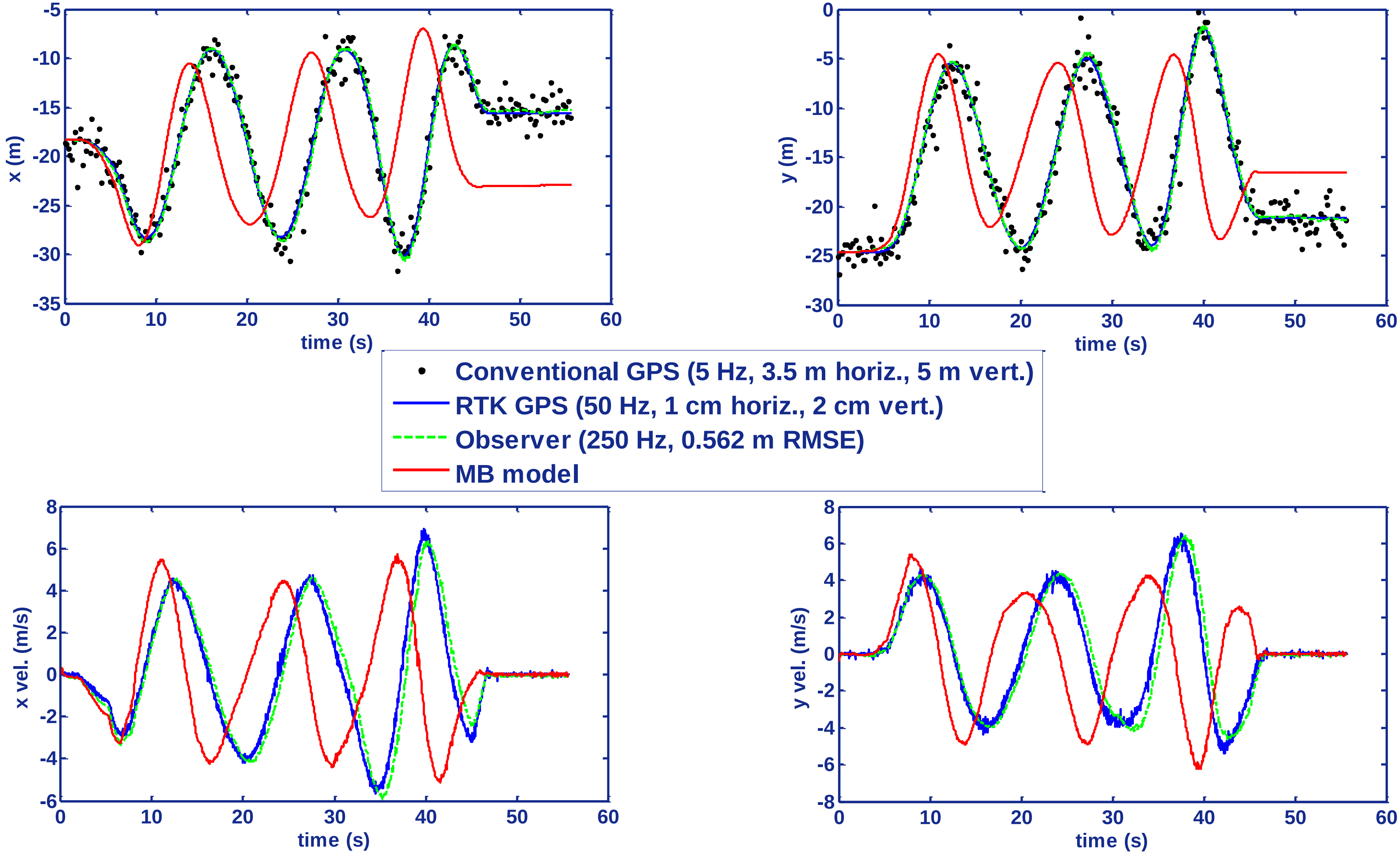
$$\mathbf{h}_\mathbf{x} = \begin{bmatrix} \ldots \mathbf{I}_4 & \mathbf{0}_{4\times 4} \ldots \\ \ldots \mathbf{0}_{4\times 4} & \mathbf{0}_{4\times 4} \ldots \end{bmatrix}$$

Video

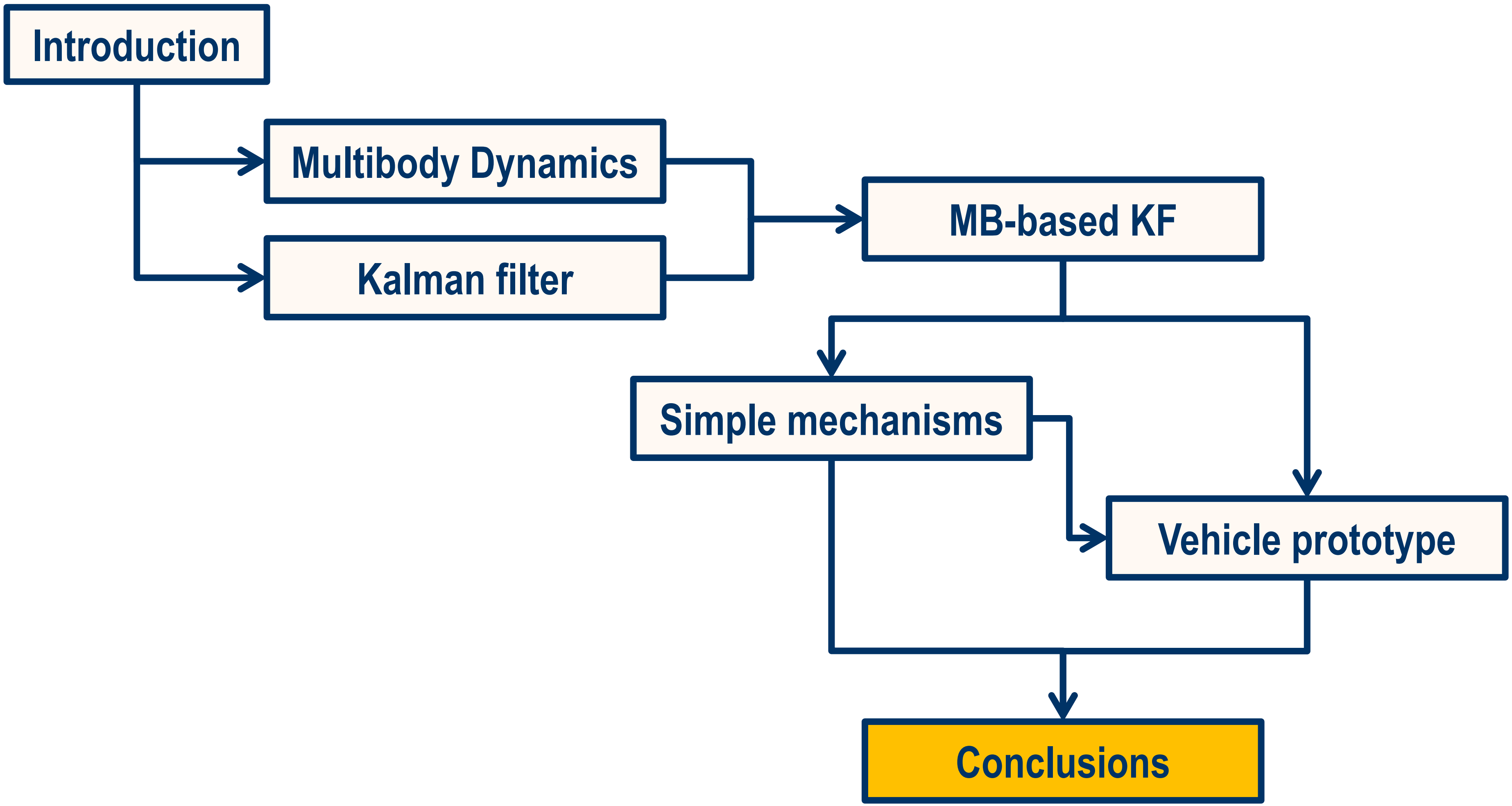
Results (RTK GPS)



Results (Conventional GPS)



Application to a vehicle prototype



Conclusions

- **KF and two MB formulations have been presented**
- **They were used to devise MB-based state observers (plant and sensor models)**
- **Benchmark using simulated simple linkages (four-bar and five-bar linkages)**
  - Two levels of plant modeling error
  - Five sampling rates for sensors
  - Three sensor configurations in each mechanism
  - UKF best accuracy
    - UKF TR: best accuracy
    - UKF FE: less computational cost than the UKF TR
  - errorEKF has the lowest computational cost

Conclusions

- **An error-state Kalman filter has been implemented for a vehicle MB model**
  - Positioning measurement is obtained at a higher rate and with improved accuracy
    - RTK GPS: 50 Hz
    - Conventional GPS: 5 Hz
    - Observer: 250 Hz
  - The implementation runs faster than real time
    - A maneuver of 55.744 s is run in 32.06 s
  - Possibility of dealing with GPS outages (to be tested)
  - More information available: suspensions, slip angle, etc (to be validated)

Future work

- **Precise study of the computational cost**
  - Study iteration limit
  - Increase the integration frequency: formulations, integrators, etc.
- **Study effect of GPS errors and outages**
- **Study the accuracy of magnitudes which are not states of the observer:**
  - Suspensions
  - Slip angles of the wheels
  - Etc.
- **Introduce acceleration errors and force corrections**
  - Acceleration sensors

Works derived from this thesis

Journal papers

- Roland Pastorino, Emilio Sanjurjo, Alberto Luaces, Miguel Á. Naya, Wim Desmet, Javier Cuadrado. Validation of a Real-Time Multibody Model for an X-by-Wire Vehicle Prototype Through Field Testing. Journal of Computational and Nonlinear Dynamics, 10(3):031006, 2015.
- José L. Torres-Moreno, José L. Blanco-Claraco, Antonio Giménez-Fernández, Emilio Sanjurjo, Miguel Á. Naya. Online Kinematic and Dynamic-State Estimation for Constrained Multibody Systems Based on IMUs. Sensors, 16(3):333, 2016.

Submitted journal papers

- Emilio Sanjurjo, Miguel Á. Naya, Javier Cuadrado, Arend Schwab. Roll Angle Estimator Based on Angular Rate Measurement for Single Track Vehicles. Vehicle System Dynamics (under review).
- Emilio Sanjurjo, Miguel Á. Naya, José Luis Blanco-Claraco, José Luis Torres-Moreno, Antonio Giménez-Fernandez. Accuracy and efficiency comparison of various nonlinear Kalman filters applied to multibody models. Nonlinear Dynamics (under review).

Works derived from this thesis

- **International conference communications**
  - Emilio Sanjurjo, Roland Pastorino, Pasquale Gallo, Miguel A. Naya. Implementation Issues of an On Board Real-Time Multibody Model, in 3rd Joint Int. Conference on Multibody System Dynamics (IMSD 2014) and 7th Asian Conference on Multibody Dynamics (ACMD 2014), Busan, Korea, 2014.
  - José L. Torres-Moreno, José L. Blanco-Claraco, Emilio Sanjurjo, Miguel Á. Naya, Antonio Giménez-Fernández. Towards Benchmarking of State Estimators for Multibody Dynamics, in 3rd Joint Int. Conference on Multibody System Dynamics (IMSD 2014) and 7th Asian Conference on Multibody Dynamics (ACMD 2014), Busan, Korea, 2014.
  - Emilio Sanjurjo, José L. Blanco-Claraco, José L. Torres-Moreno, Miguel Á. Naya. Testing the Efficiency and Accuracy of Multibody-Based State Observers, in ECCOMAS Thematic Conference on Multibody Dynamics 2015, Barcelona, Spain, 2015.
  - Emilio Sanjurjo, Edoardo Sinigaglia, Miguel Á. Naya. Multibody-based State Observer for Navigation Applications, in 4th Joint Int. Conference on Multibody System Dynamics (IMSD 2016), Montreal, Canada, 2016.

- **Part of this thesis has been financed by the Spanish Government through the BES-2013-063598 predoctoral fellowship.**

Thank you for your attention

Emilio Sanjurjo

Laboratorio de Ingeniería Mecánica,

University of A Coruña

Advisors:

- Miguel Ángel Naya Villaverde
- Javier Cuadrado Aranda