# Modeling and Analysis of Signal Estimation for Stepper Motor Control

Dan Simon
Cleveland State University
October 8, 2003

Cleveland State University

# Outline

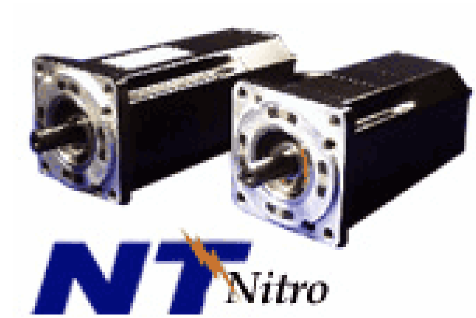
- Problem statement
- Simplorer and Matlab
- Optimal signal estimation
- Postprocessing
- Simulation results
- Conclusion

# Problem statement

- Speed control for PM DC motor
- Aerospace applications
  - Flywheel energy storage
  - Flight control trim surfaces
  - Hydraulics
  - Fans
  - Thrust vector control
  - Fuel pumps

# Problem statement

DC Machine Permanent Excitation




$$\dot{I} = V/L - RI/L - k\omega/L$$

$$\dot{\omega} = kI/J - T_L/J$$

$I$ = armature current

$V$ = armature voltage

$L$ = inductance

$R$ = resistance

$k$ = motor constant

$\omega$ = rotor speed

$\theta$ = rotor angle

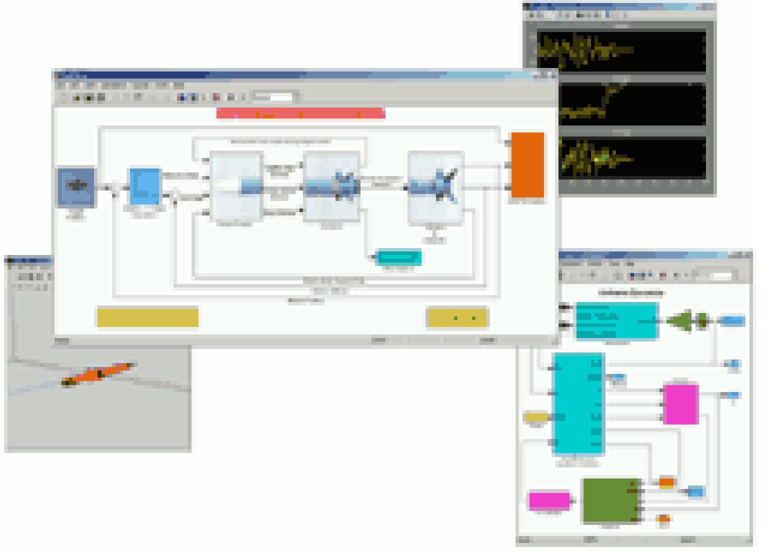
$J$ = moment of inertia

$T_L$ = load torque

# Problem statement

State assignment:

- $x_1 = I$
- $x_2 = \omega$
- $x_3 = \theta$
- $x_4 = T_L / J$



Measurements:

- $y$ = current (and possibly position)

# Problem statement

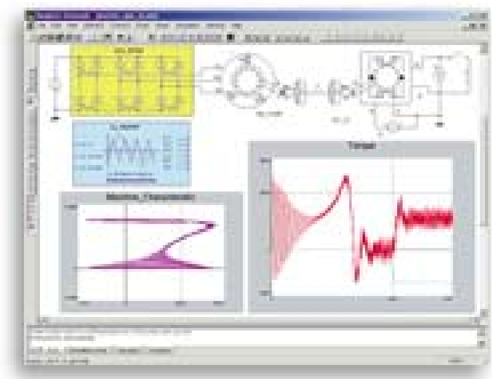
$$\dot{x} = \begin{bmatrix} -R/L & -k/L & 0 & 0 \\ k/J & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 1/L \\ 0 \\ 0 \\ 0 \end{bmatrix} V + \text{noise}$$

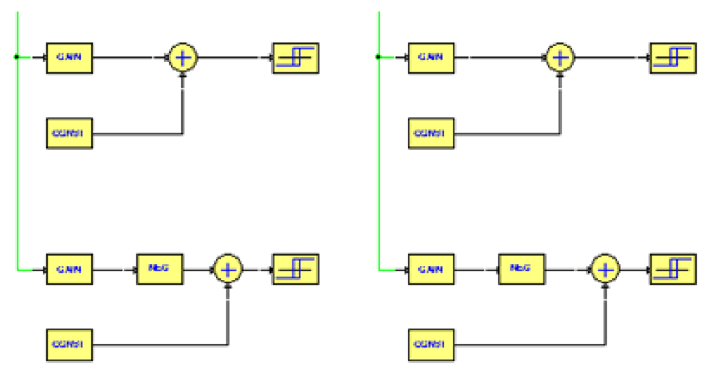
$$\mathrm{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \text{noise}$$

Estimate velocity $x_2$

# Simplorer and Matlab

- Simplorer
  - Circuit element models
  - Electric machine models
  - Data analysis tools
  - Interfaces with Matlab / Simulink

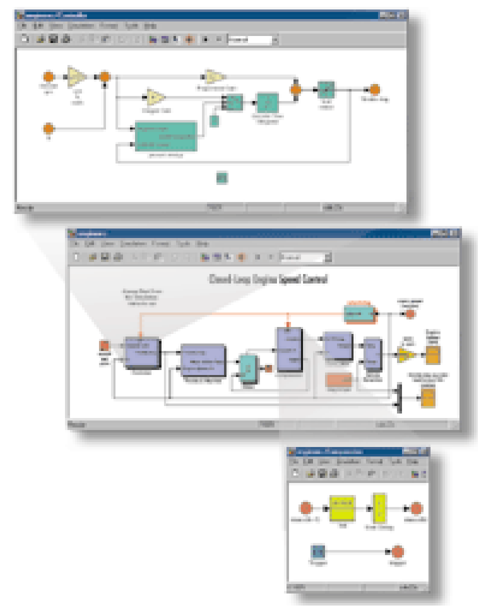




Block diagram

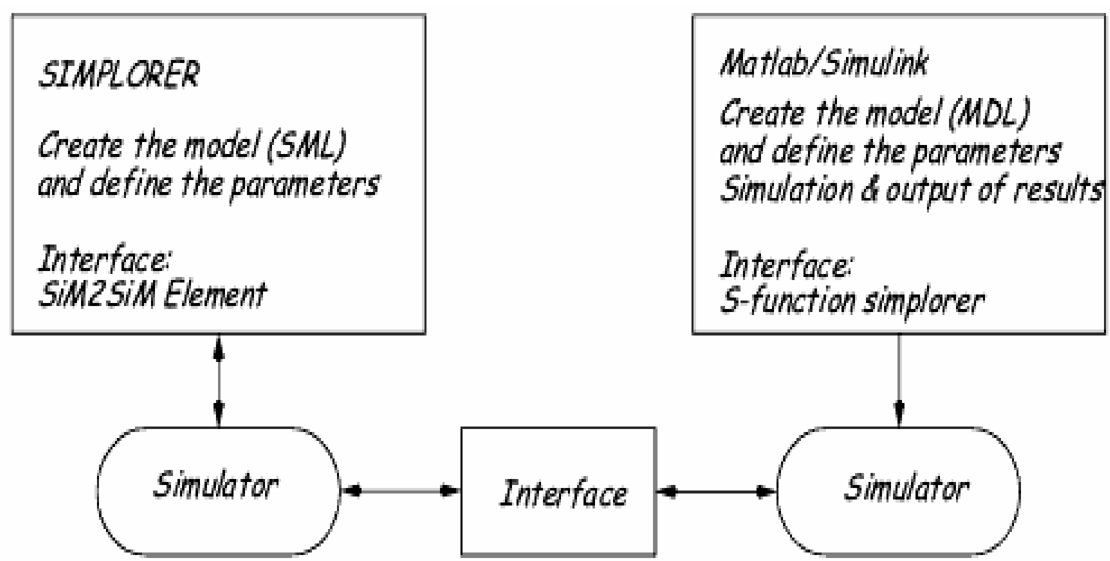
# Simplorer and Matlab

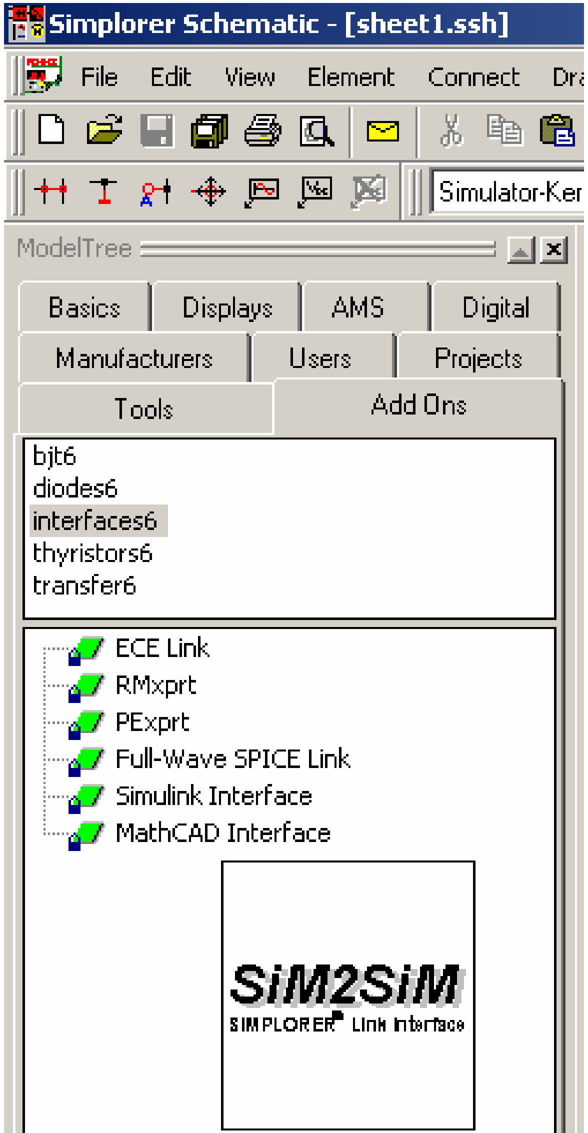
- Matlab
  - Powerful math and matrix capabilities
- Co-Simulation
  - Link Simplorer and Matlab
  - Plot and analyze data in either environment



# Simplorer and Matlab

- Use the SiM2SiM tool in Add Ons / interfaces6
- Begin the simulation in Simulink

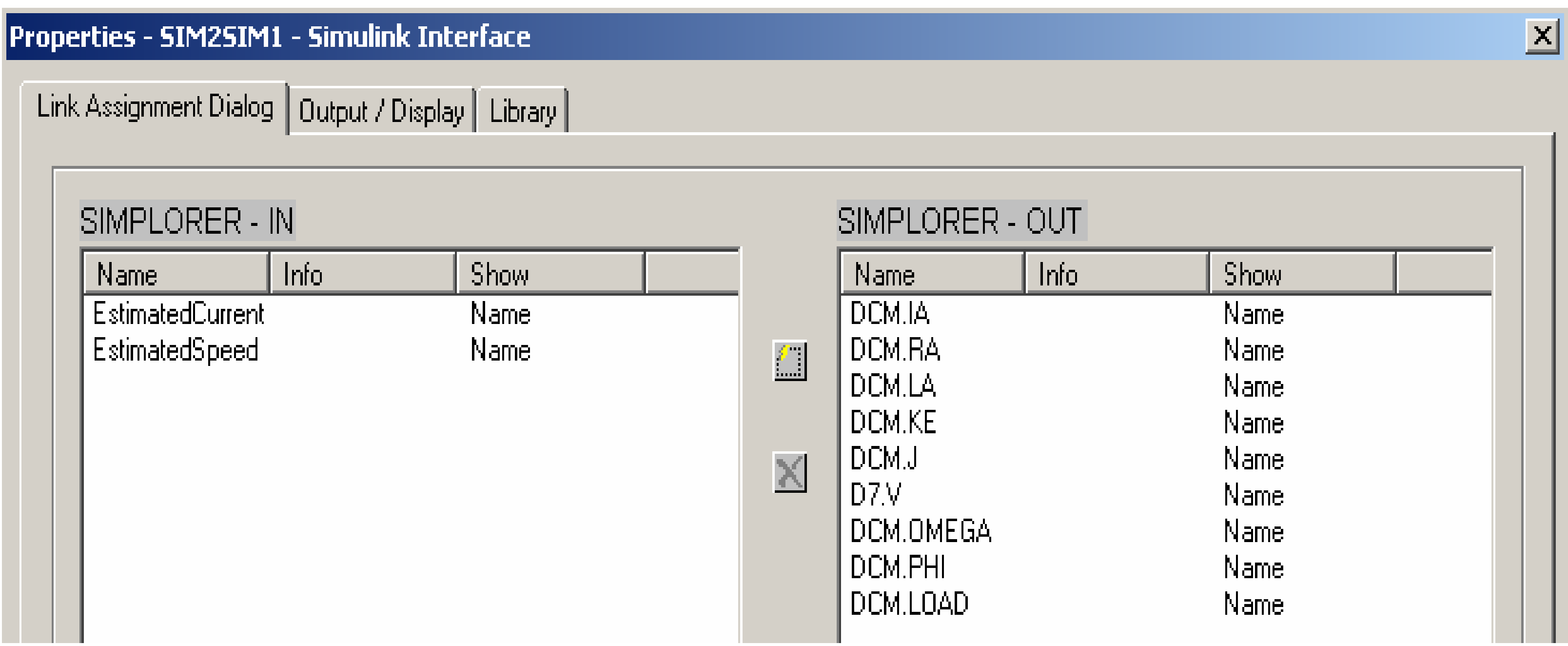


# Simplorer and Matlab

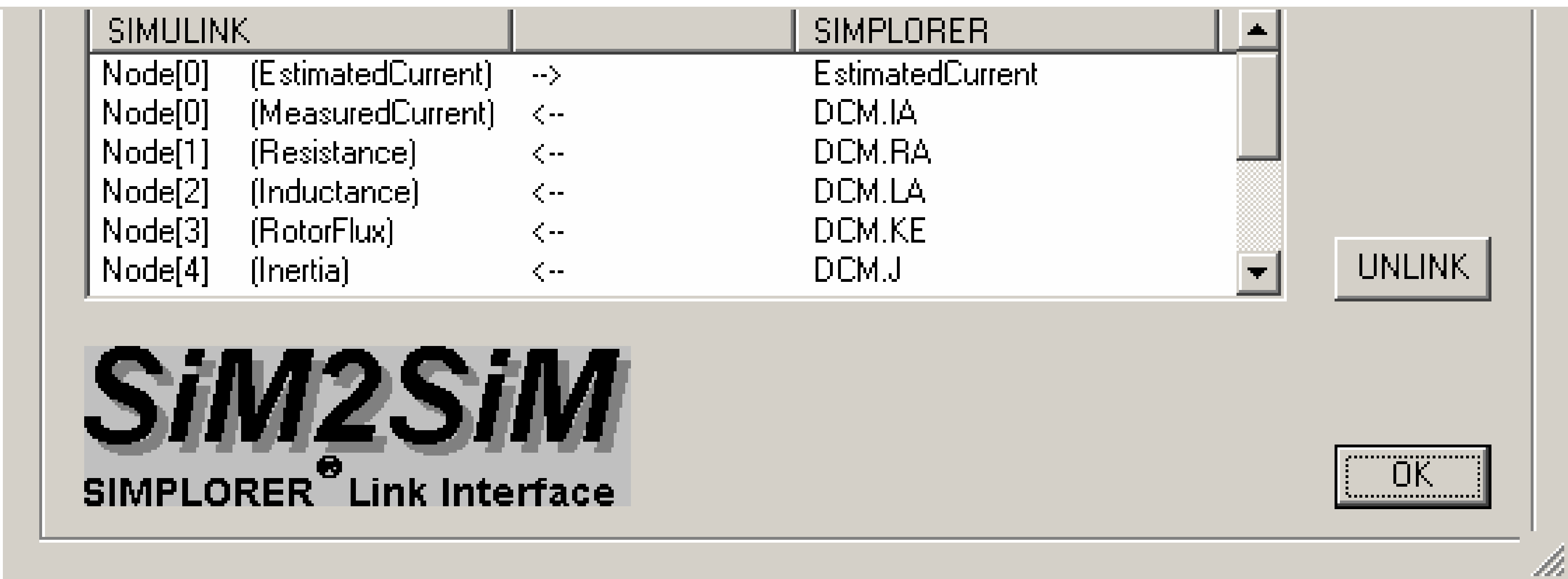
Define Simplorer inputs and outputs in the property dialog of the SiM2SiM component

Simplorer ↔ Simulink



# Simplorer and Matlab

- Use the S-function property dialog in Matlab to link Simplorer / Matlab signals



# Simplorer and Matlab

- Begin the simulation in Matlab
- Couple Simplorer's and Matlab's strengths
  - Simplorer: power electronics, electromechanics, data analysis, state diagrams
  - Matlab: matrix algebra, toolboxes

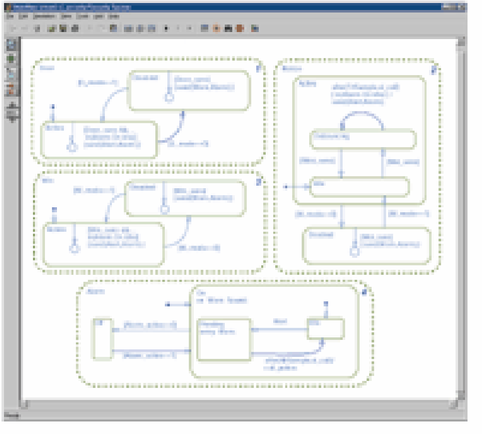
- Data analysis / viewing can be done in either Simplorer or Matlab

# Optimal signal estimation

Given a linear system: $\dot{x} = Ax + B_u u + B_w w$

$$y = Cx + Dv$$

- $x$ = state
- $y$ = measurement
- $u$ = control input
- $w$, $v$ = noise



Find the best estimate for the state $x$

# Optimal signal estimation

Suppose *w ~ N(0, Q)* and *v ~ N(0, R)*.

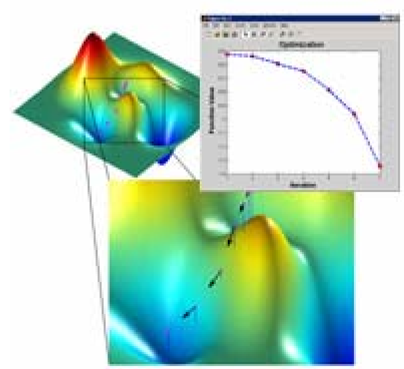
The Kalman filter solves the problem

$$\min E\left\{\int (x-\hat{x})^T (x-\hat{x})\, dt\right\}$$

$$\dot{P} = AP + PA^T + B_w Q B_w^T - KCP$$

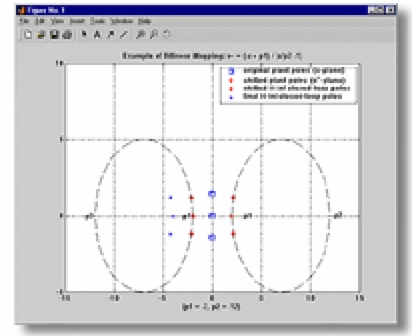
$$K = PC^T D^{-T} R^{-1} D^{-1}$$

$$\dot{\hat{x}} = A\hat{x} + K(y - C\hat{x})$$



# Optimal signal estimation

The H∞ filter solves the problem



$$\dot{x} = Ax + B_u u + B_w w$$

$$y = Cx + Dv$$

$$\frac{\int \left\| x - \hat{x} \right\|_S^2 \, dt}{\left\| x(0) - \hat{x}(0) \right\|_{P_0^{-1}}^2 + \int \left\| w \right\|_{Q^{-1}}^2 \, dt + \int \left\| v \right\|_{R^{-1}}^2 \, dt} < \frac{1}{\theta}$$

This is a game theory approach.

Nature tries to maximize the estimation error.

The engineer tries to minimize the error.

# Optimal signal estimation

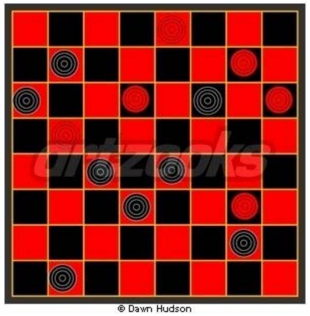
Rewrite the previous equation:

$$-\frac{1}{\theta}\left\|x(0)-\hat{x}(0)\right\|^2_{P_0^{-1}} + \int \left\|x-\hat{x}\right\|^2_S dt - \frac{1}{\theta}\int \left\|w\right\|^2_{Q^{-1}} + \left\|v\right\|^2_{R^{-1}} dt < 0$$

$$J < 0$$

Game theory: nature tries to maximize *J* and the engineer tries to minimize *J*

$$\min_{\hat{x}} \max_{w,v,x(0)} J$$



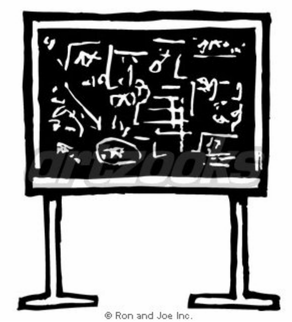
# Optimal signal estimation

The H∞ filter is given as follows:

$$\dot{P} = AP + PA^T + B_w Q B_w^T - KCP + \theta PSP$$

$$K = PC^T D^{-T} R^{-1} D^{-1}$$

$$\dot{\hat{x}} = A\hat{x} + K(y - C\hat{x})$$




Note this is identical to the Kalman filter except for an extra term in the Riccati equation.

# Optimal signal estimation

- The Kalman filter is a least-mean-squares estimator
- The H∞ filter is a worst-case estimator
- The Kalman filter is often made more robust by artificially increasing *P*
- The H∞ filter shows exactly how to increase *P* in order to *add robustness*

$$\dot{P} = AP + PA^T + B_w Q B_w^T - KCP + \theta PSP$$

# Optimal signal estimation

- Steady state:

$$\dot{P} = AP + PA^T + B_w Q B_w^T - KCP + \theta PSP$$
$$= 0$$

$$K = PC^T D^{-T} R^{-1} D^{-1}$$

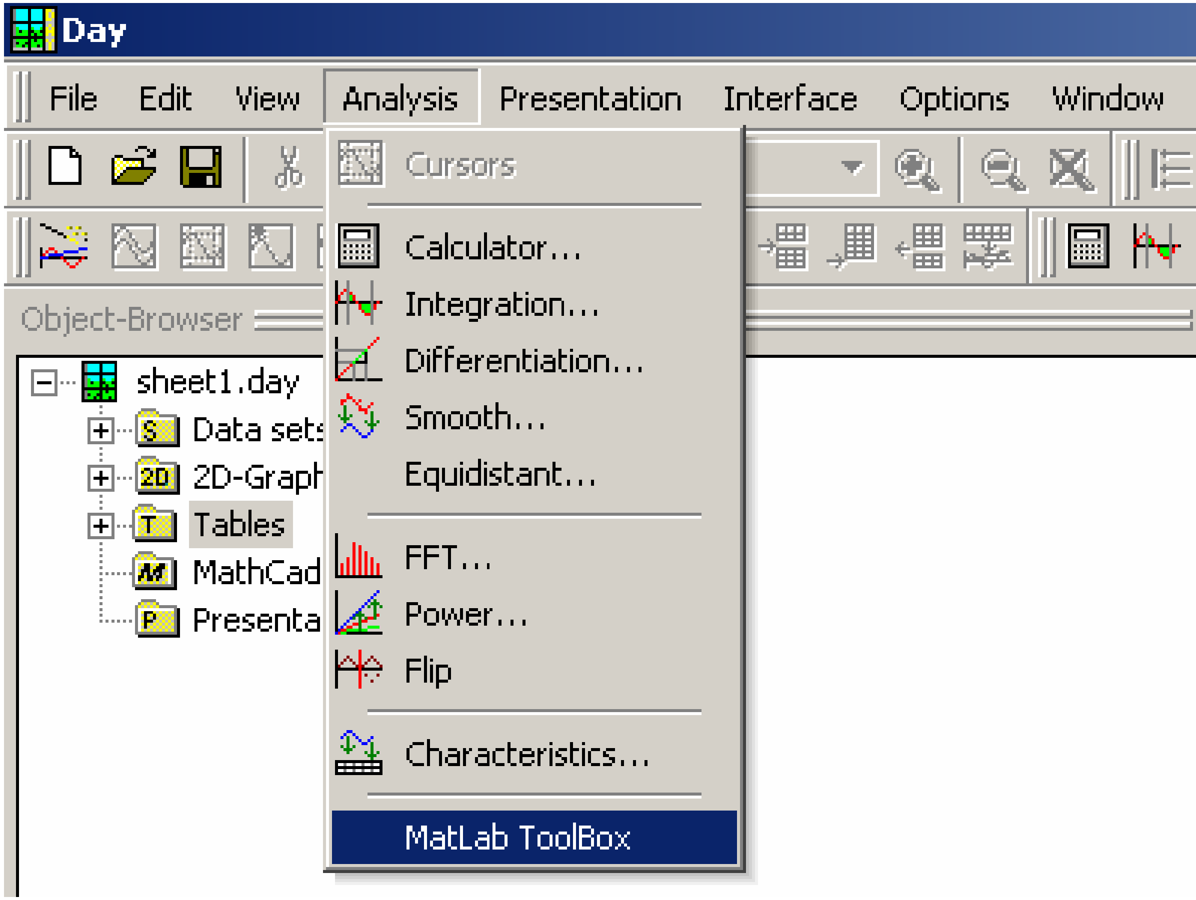
$$\dot{\hat{x}} = A\hat{x} + K(y - C\hat{x})$$

*Jacopo Riccati*
*1676-1754*

- This is an *Algebraic Riccati Equation*
- Real time computational savings

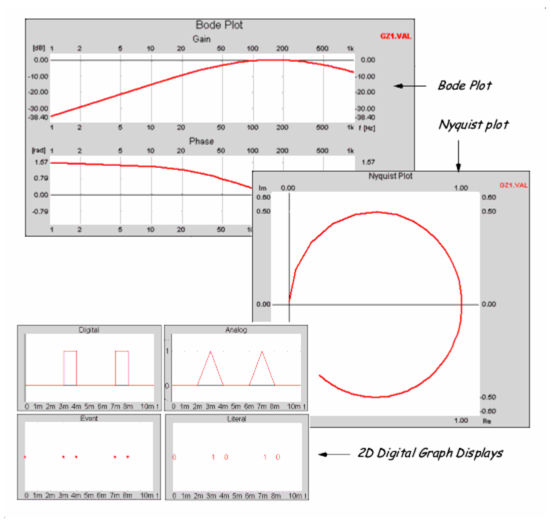
# Postprocessing

Transfer Matlab data to Simplorer for plotting and analysis



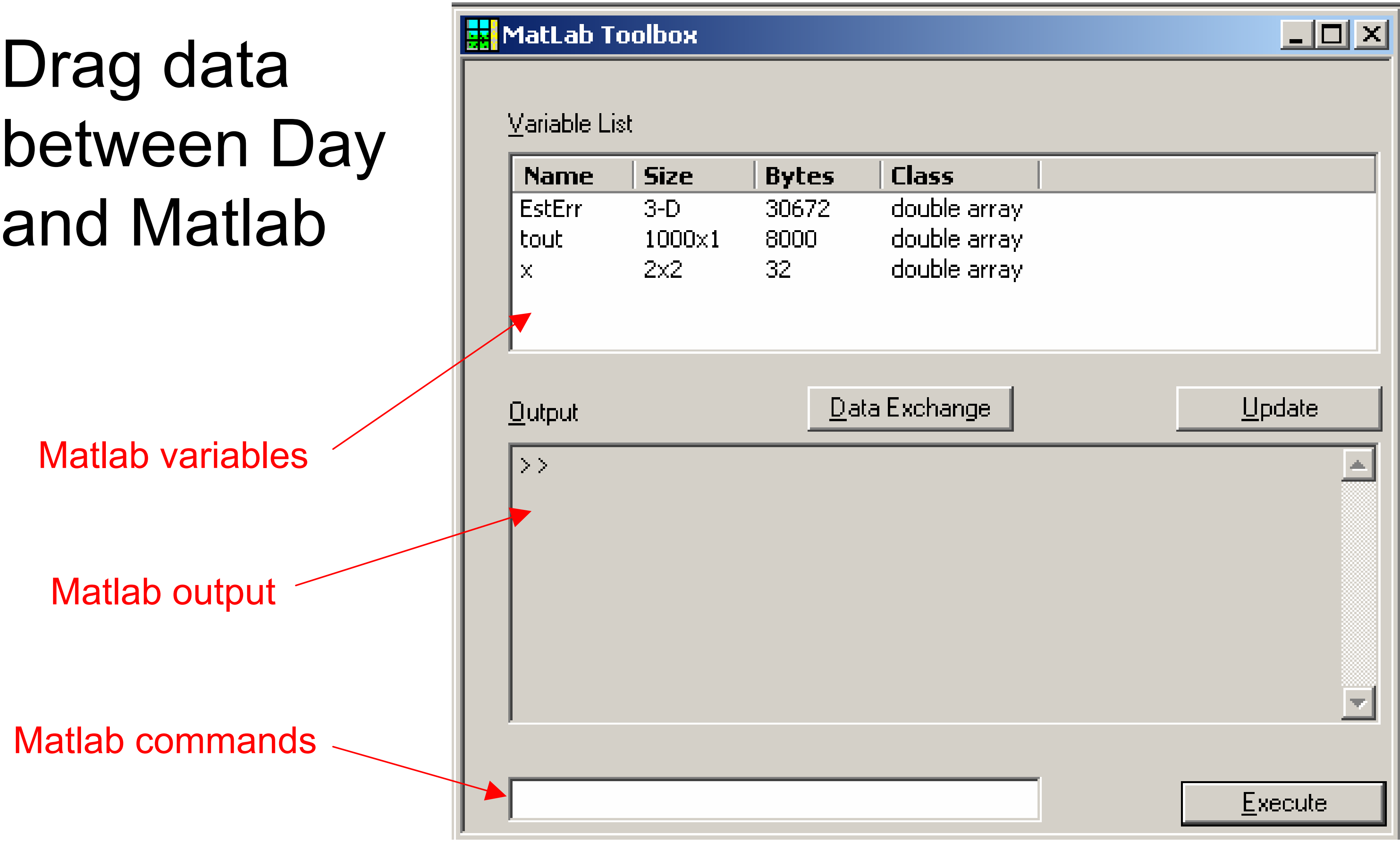
# Postprocessing

- Start the Matlab postprocessor interface *before* starting the co-simulation
- After running the co-simulation, the Day postprocessor can exchange data with Matlab



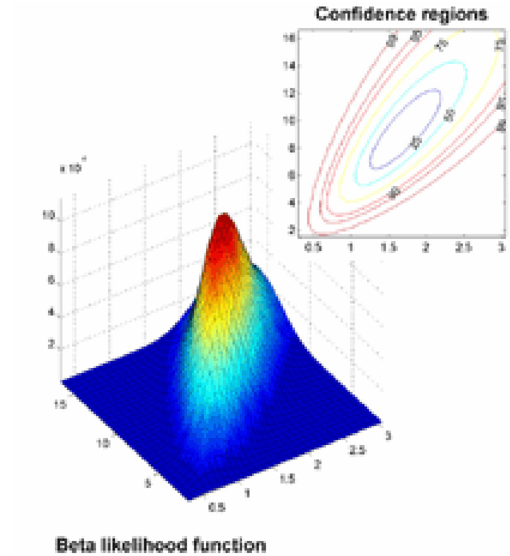
# Postprocessing

Drag data between Day and Matlab

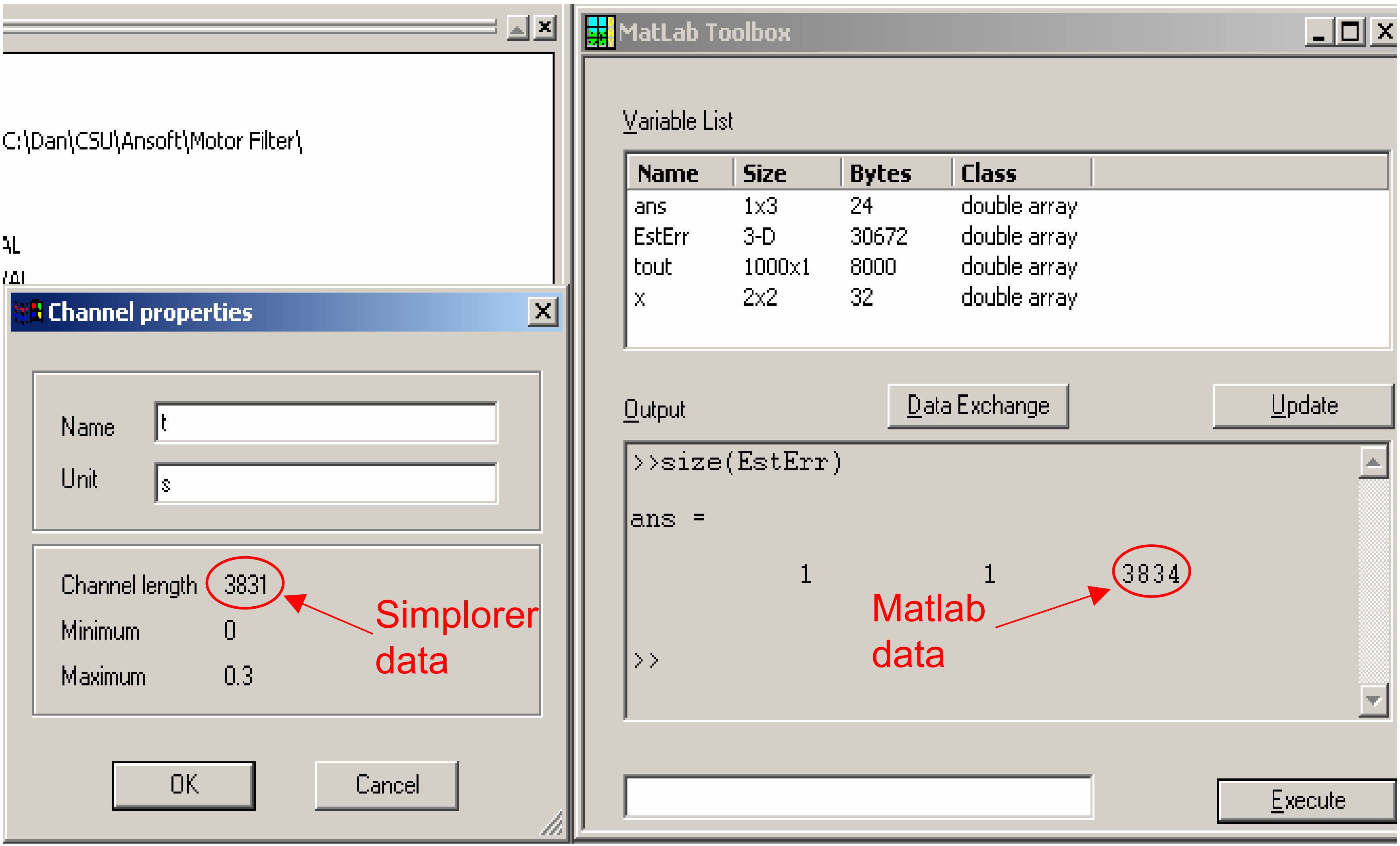


# Postprocessing

- Day cannot handle arrays with more than two dimensions – use Matlab's "squeeze" command
- Make sure Matlab data is not longer than Simplorer's time array

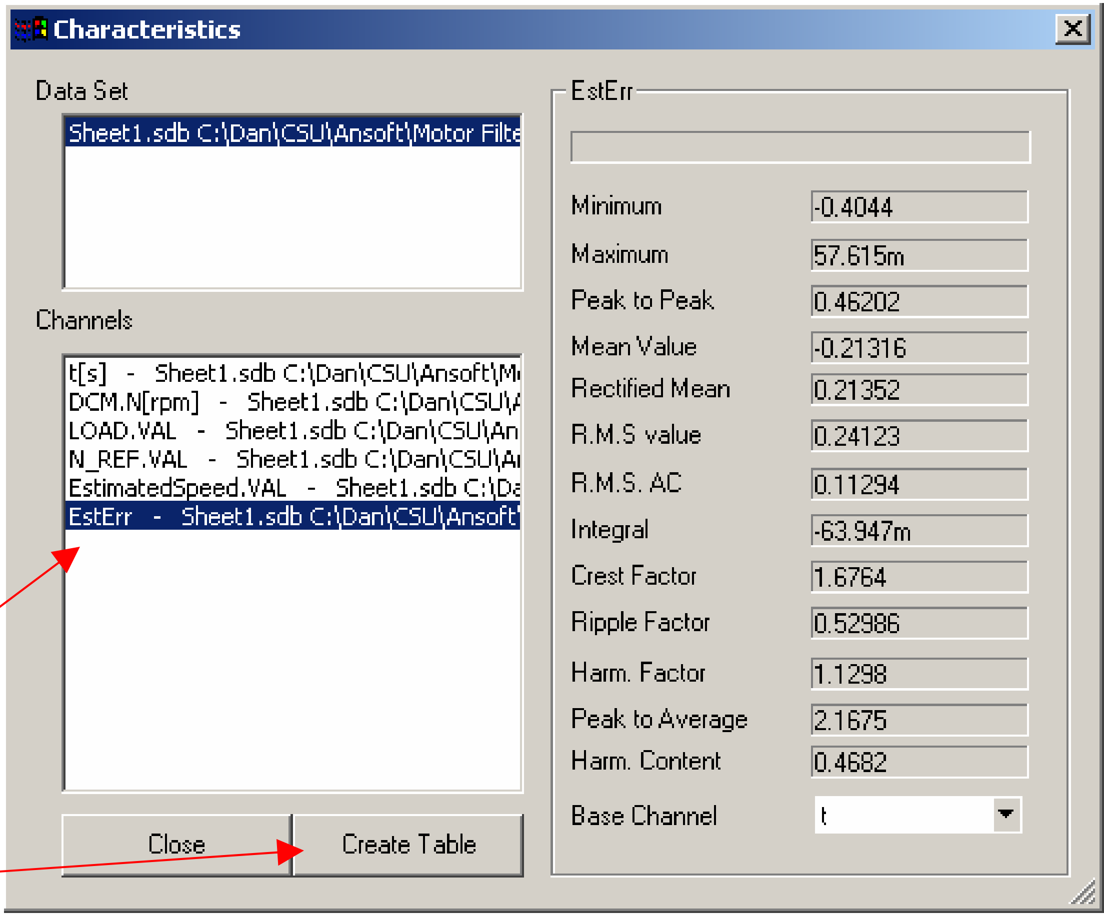


# Postprocessing



# Postprocessing

*Analysis → Characteristics* to view statistical information




Select the desired output variable

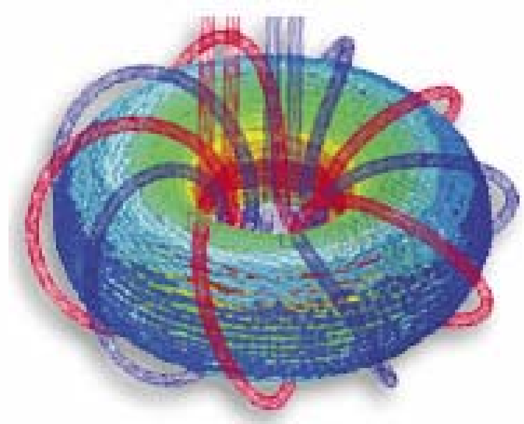
Export to table

# Simulation results

SIM2SIM1

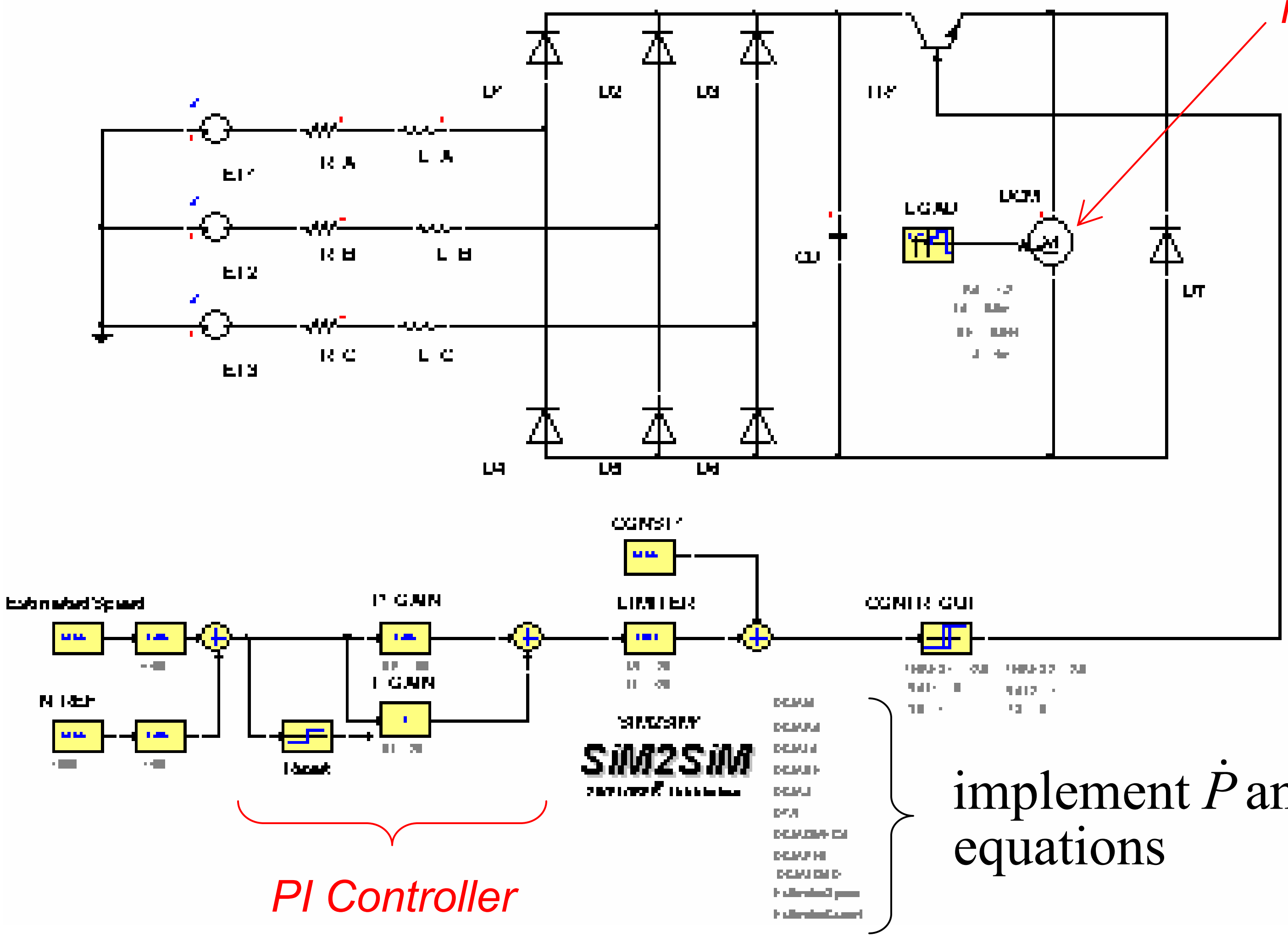
SiM2SiM

SIMPLORER® Link Interface



- DCM.RA, DCM.LA, DCM.KE, DCM.J — *motor parameters*
- D7.V — *control input*
- DCM.IA, DCM.PHI — *measurements*
- EstimatedSpeed, EstimatedCurrent — *Ouput from Matlab*

# Simulation results




*motor*

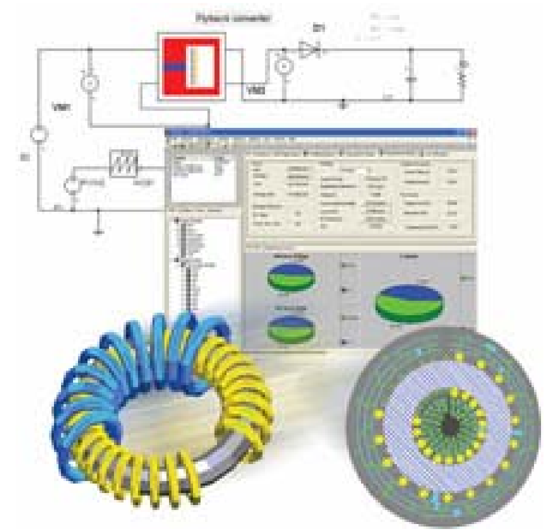
*PI Controller*

implement $\dot{P}$ and $\dot{\hat{x}}$ equations

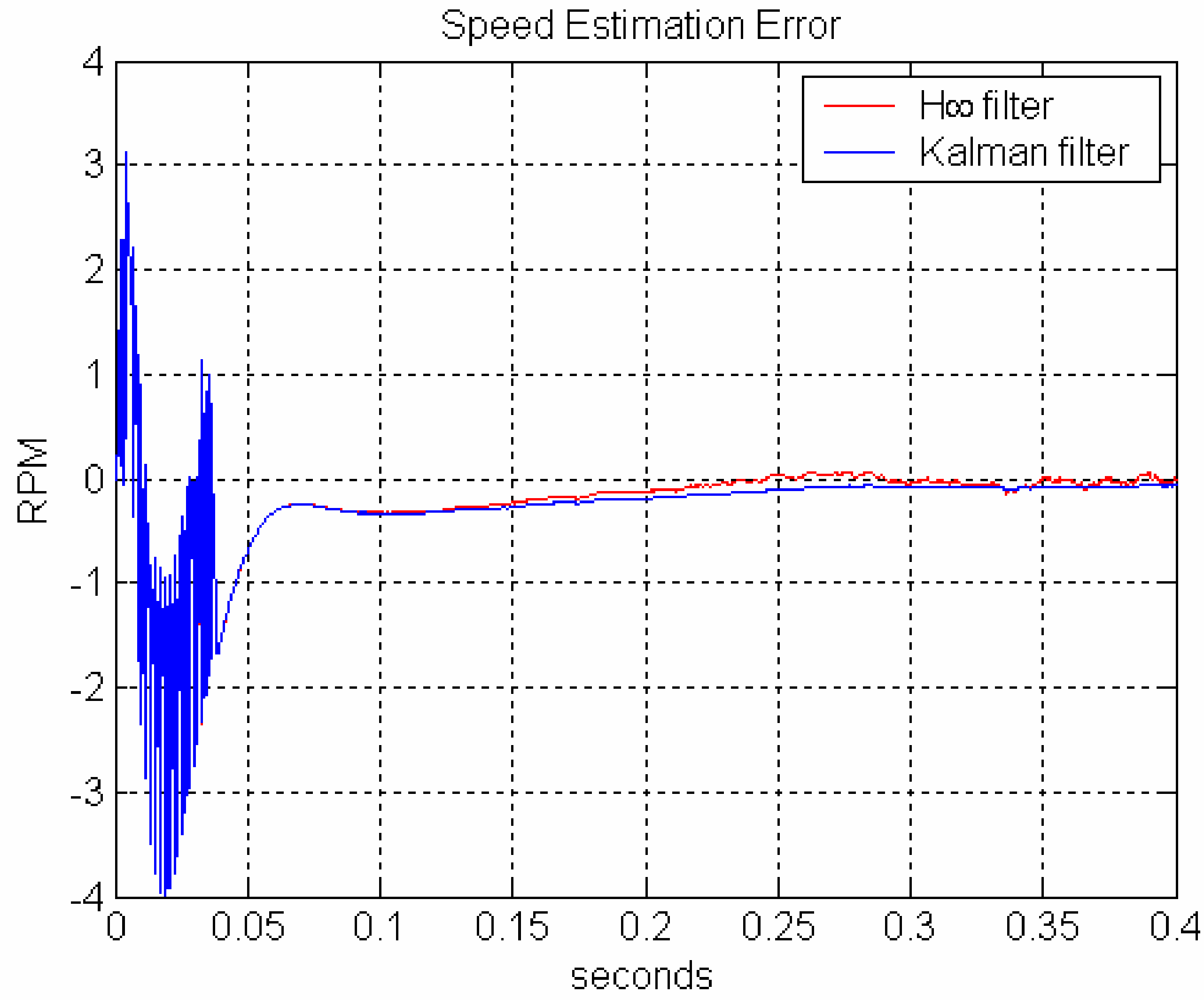
# Simulation results

Simulation parameters:

- 1.2 ohms, 9.5 mH, 0.544 Vs, 0.004 kg·m$^2$
- Initial speed = 0
  cmd speed = 1000 RPM
- External load torque changes from 0 to 0.1
- Measurement errors
  0.1 A, 0.1 rad (1 $\sigma$)



# Simulation results

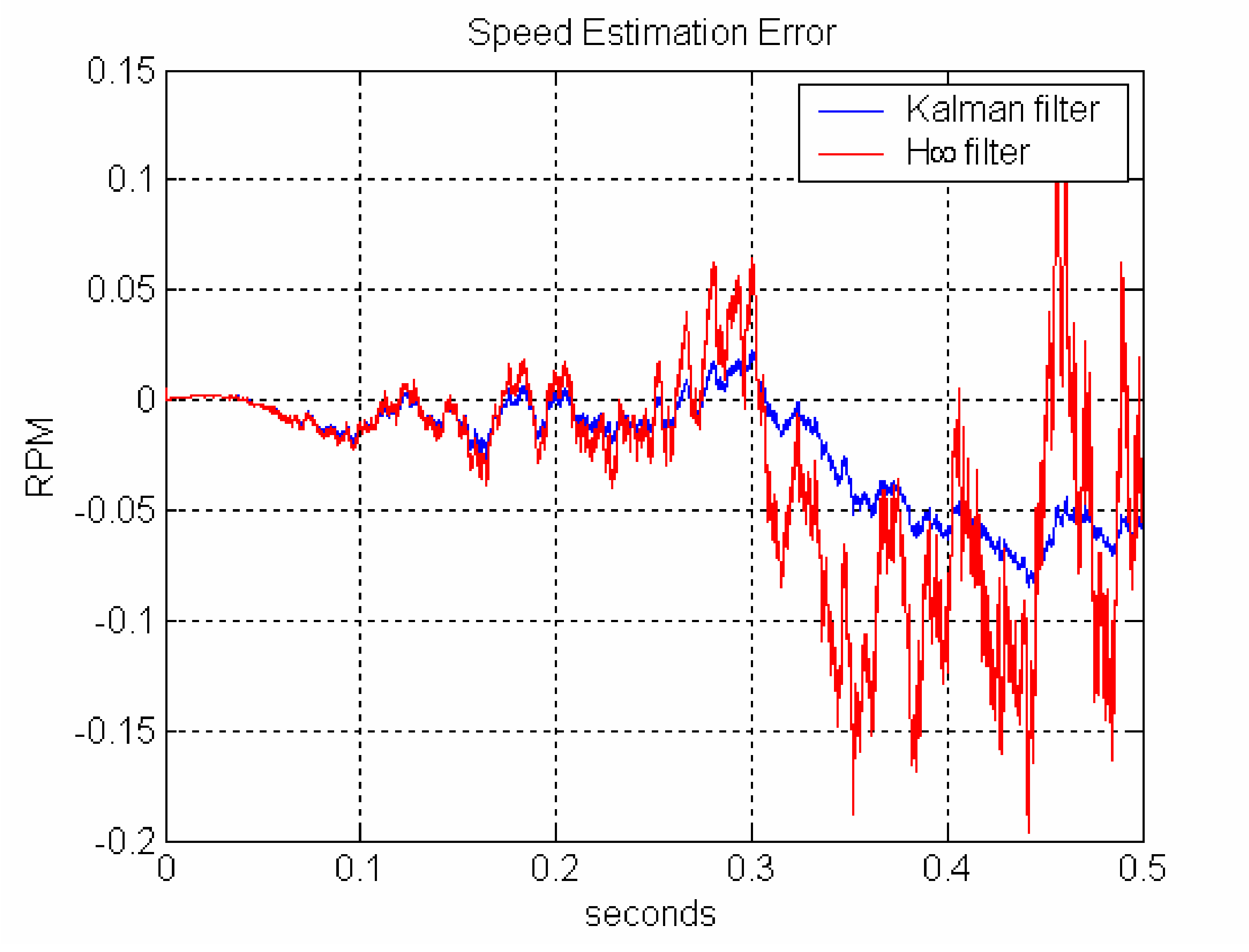


# Simulation results

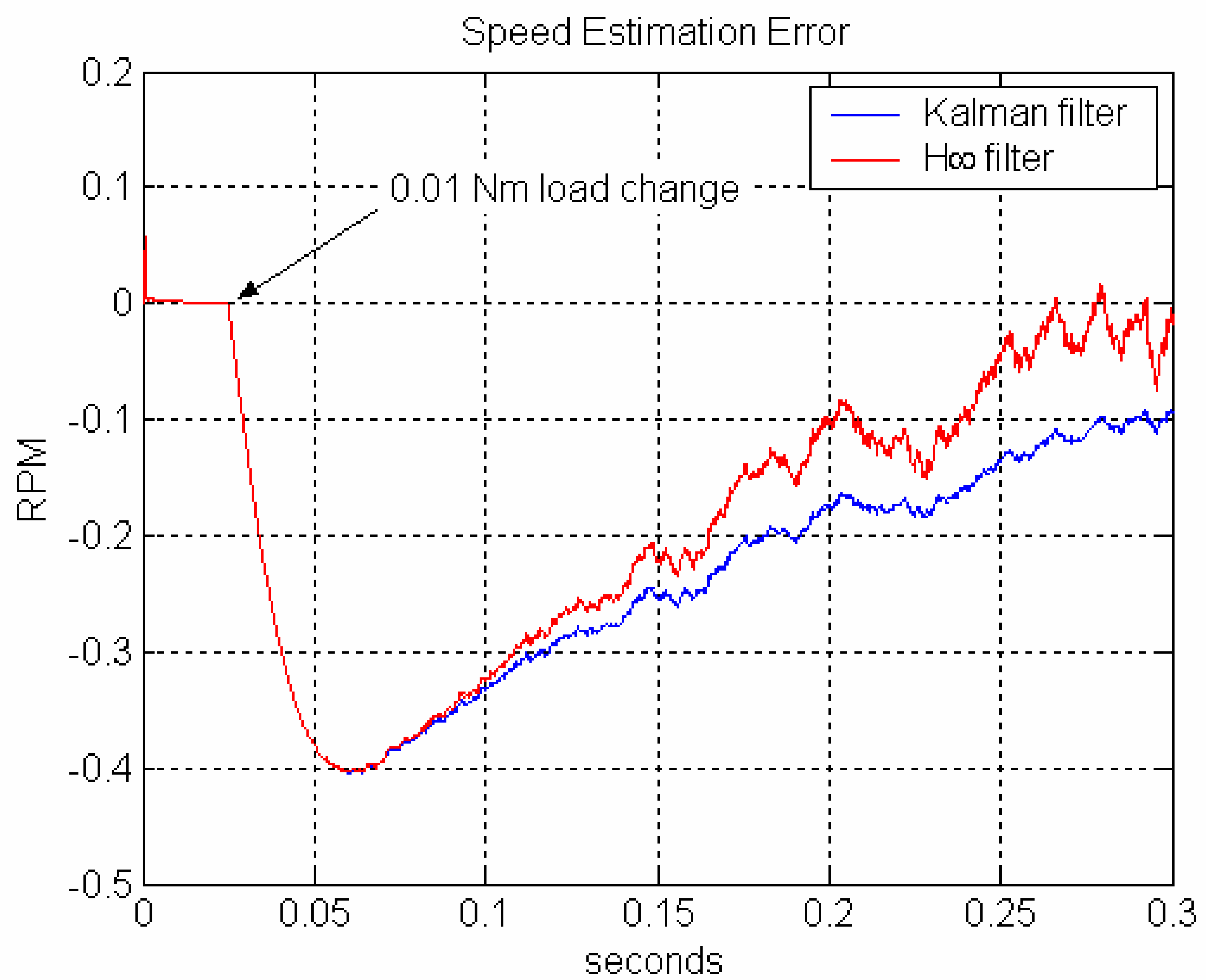
Steady state parameters:

- Initial speed = 1000
  commanded speed = 1000 RPM
- External load torque = 0
- Measurement error = 0.1 A, 0.1 rad (1 $\sigma$)

# Simulation results



# Simulation results



# Simulation results

RMS Estimation Errors (RPM)
current and position measurements

| | **Kalman filter** | **H∞ filter** |
|---|---|---|
| Nominal | 0.033 | 0.057 |
| R = R / 2 | 0.102 | 0.106 |
| L = L / 2 | 0.032 | 0.056 |
| J = 0.6 J | 0.060 | 0.060 |
| k = k / 2 | 834 | 828 |
| R = 2 R | 0.122 | 0.120 |
| L = 2 L | 0.035 | 0.058 |
| J = 2 J | 0.078 | 0.082 |
| k = 2 k | 941 | 938 |

# Simulation results

Now suppose we measure winding current but not rotor position. Can we still get a good estimate of motor velocity?

$$\dot{x} = \begin{bmatrix} -R/L & -k/L & 0 & 0 \\ k/J & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 1/L \\ 0 \\ 0 \\ 0 \end{bmatrix} V + \text{noise}$$

$$\text{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x + \text{noise (0.1 amps, one sigma)}$$

# Simulation results

RMS Estimation Errors (RPM)
current measurement only

| | **Kalman filter** | **H∞ filter** |
|---|---|---|
| Nominal | 0.034 | 0.036 |
| R = R / 2 | 0.102 | 0.102 |
| L = L / 2 | 0.034 | 0.036 |
| J = 0.6 J | 0.060 | 0.060 |
| k = k / 2 | 941 | 939 |
| R = 2 R | 0.154 | 0.154 |
| L = 2 L | 0.036 | 0.038 |
| J = 2 J | 0.081 | 0.081 |
| k = 2 k | 959 | 959 |

# Conclusion

- Motor state estimation is required for motor control
- Kalman filtering and $H_\infty$ filtering can be used for motor state estimation
- Steady state filtering saves time
- Simplorer / Matlab co-simulation
- Estimate motor parameters *R, L, J, k*