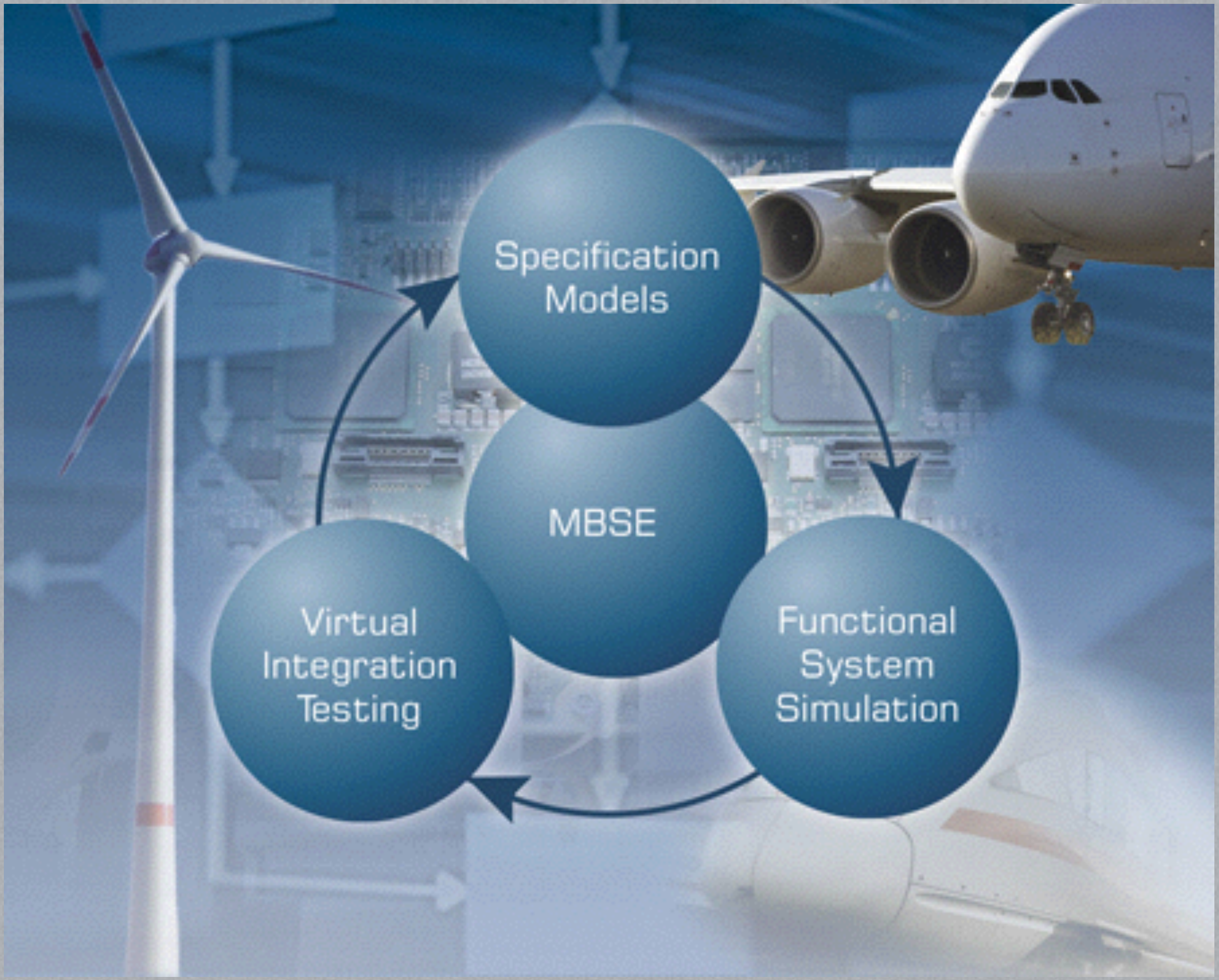


Model-based Control, Design and Resilience of Complex Systems

Dr. Abhishek Dutta at UConn

AEROSPACE ENGINEERING
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

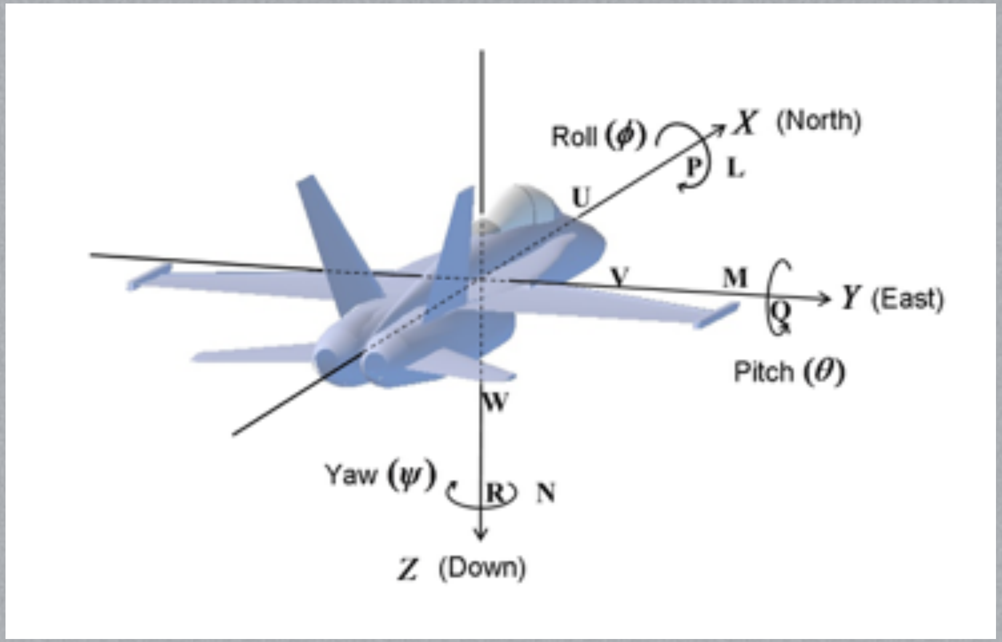
Complex Dynamical Systems

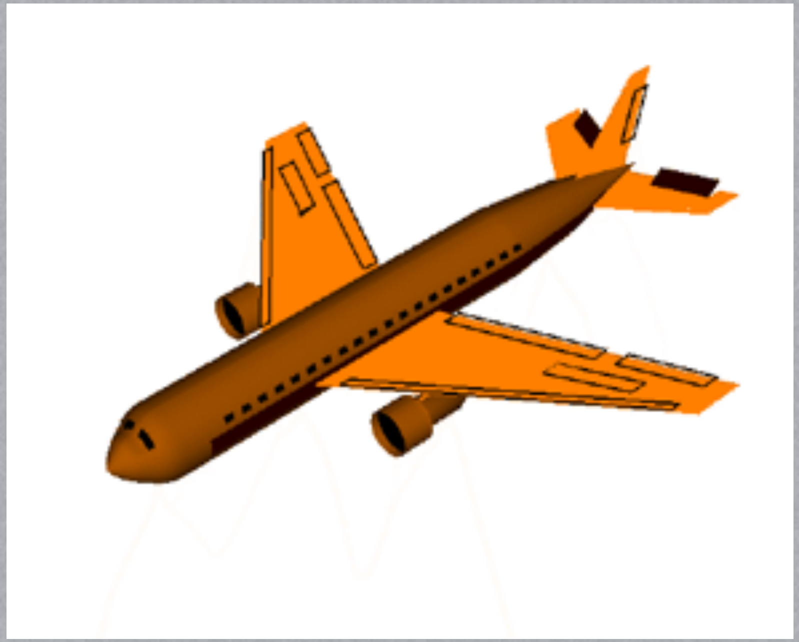
Constrained, **D**istributed, **F**ast, **T**ime-varying, **V**ulnerable

Key: **M**odel-based design, control, resilience

Modeling Flight Dynamics

Force and Moment equations





Time-varying (nonlinear), **D**istributed (longitudinal+lateral)

$$\frac{1}{m}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \dot{U}+QW-RV \\ \dot{V}+RU-PW \\ \dot{W}+PV-QU \end{bmatrix}, \quad \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx}\dot{P}+I_{xz}\dot{R}+QR(I_{zz}-I_{yy})+PQI_{xz} \\ I_{yy}\dot{Q}+PR(I_{xx}-I_{zz})+(R^2-P^2)I_{xz} \\ I_{zz}\dot{R}+I_{xz}\dot{P}+PQ(I_{yy}-I_{xx})-QRI_{xz} \end{bmatrix}$$

Linearize at level-flight, Decouple longitudinal mode

$$\begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} Z_w/m & U_0 & 0 & 0 \\ \frac{(M_w+M_{\dot{w}}Z_w/m)}{I_{yy}} & \frac{(M_q+M_{\dot{w}}U_0)}{I_{yy}} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & U_0 & 0 \end{bmatrix} \begin{bmatrix} w \\ q \\ \theta \\ h \end{bmatrix} + \begin{bmatrix} Z_{\delta e}/m \\ \frac{(M_{\delta e}+M_{\dot{w}}Z_{\delta e}/m)}{I_{yy}} \\ 0 \\ 0 \end{bmatrix} \delta_e + \begin{bmatrix} -Z_w/m \\ -\frac{(M_w+M_{\dot{w}}Z_w/m)}{I_{yy}} \\ 0 \\ 0 \end{bmatrix} w_g$$

$\dot{x}$ = A x + B u + E w

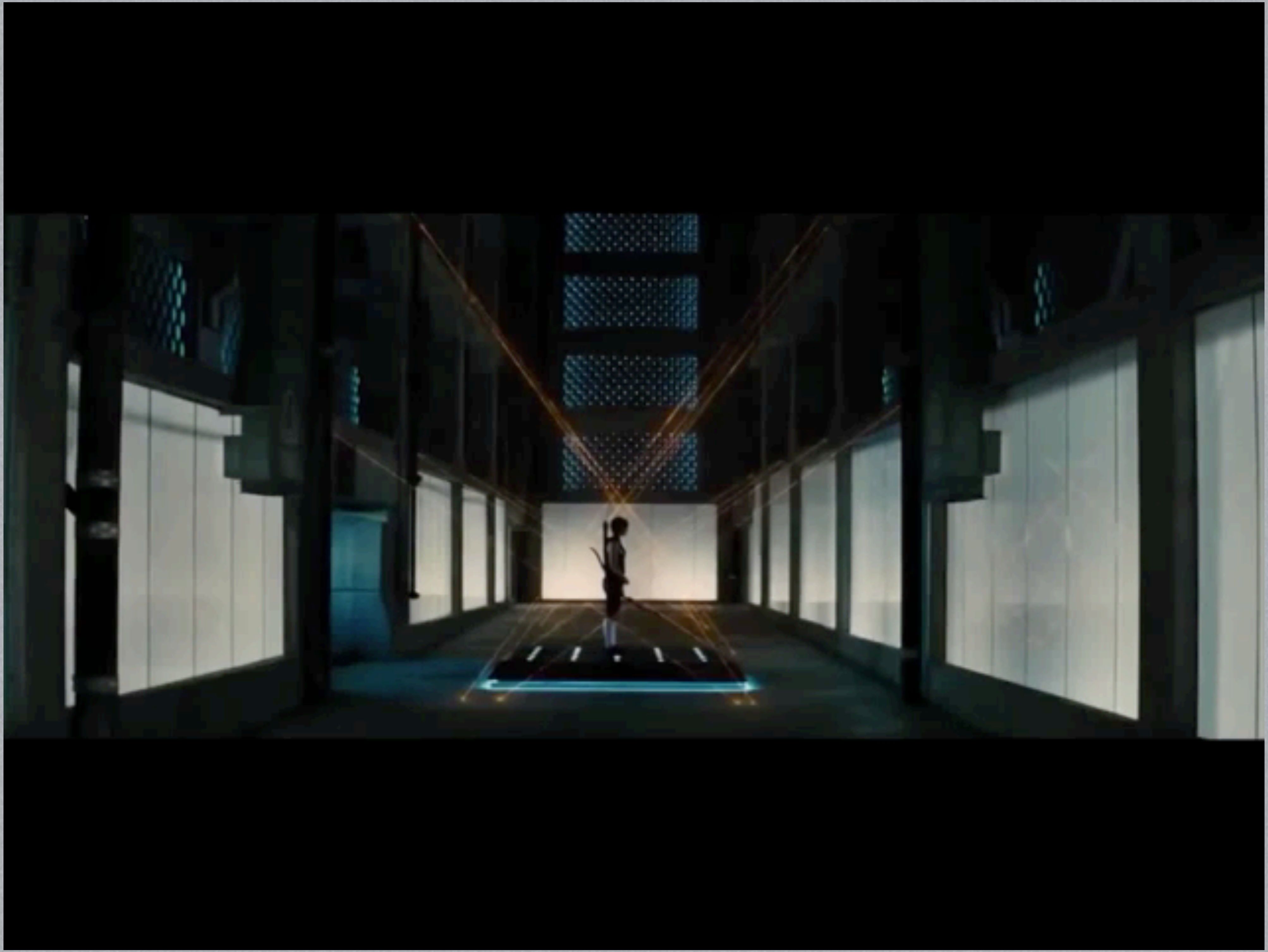
$$h = [0\ 0\ 0\ 1][w\ q\ \theta\ h]^T \iff y = Cx$$

Fast dynamics (discretize) $\quad x_{k+1} = Ax_k + Bu_k + Cw_k, \quad y_k = Cx_k$

Constraints $\quad |q| \leq 0.2rad/s, |\delta_e| \leq 0.3rad, |w_g| \leq 1m/s \iff x_k \in \mathbb{X}, u_k \in \mathbb{U}, w_k \in \mathbb{W}, \forall k$

UNIVERSITEIT
GENT

WOLFSON COLLEGE
CAMBRIDGE



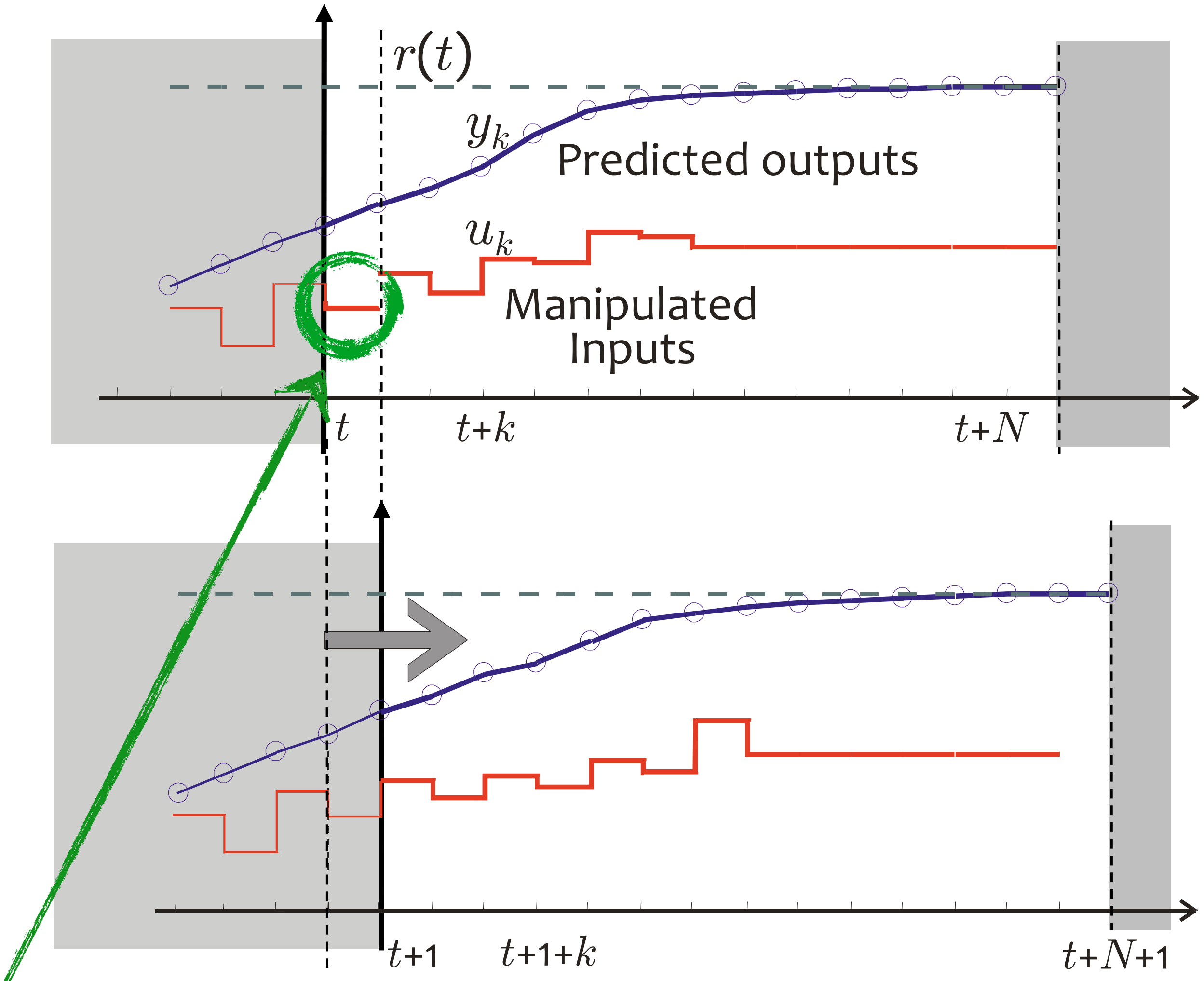
MPC algorithm

- **At time** t: solve an **optimal control** problem over a future horizon of N steps

penalty on tracking error

penalty on actuation effort

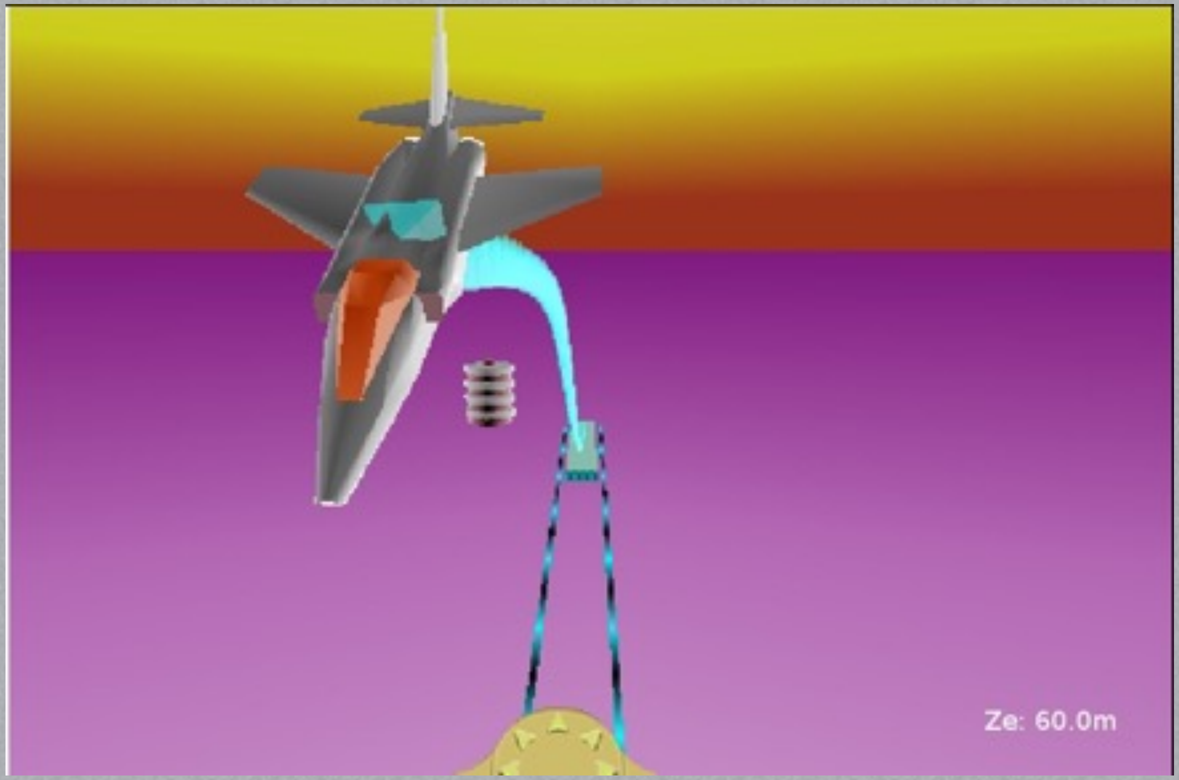
$$\begin{aligned}
\min \quad & \sum_{k=0}^{N-1} \ell(y_k - r(t+k), u_k) \\
\text{s.t.} \quad & x_{k+1} = f(x_k, u_k) \\
& y_k = g(x_k, u_k) \\
& \text{constraints on } u_k, x_k, y_k \\
& x_0 = x(t)
\end{aligned}$$



- Apply only the first optimal move $u^*(t)$, throw the rest of the sequence away
- **At time** $t+1$: Get new measurements, repeat the optimization. And so on …

MPC transforms open-loop optimal control into **feedback** control

Nonlinear **M**odel Predictive Control (hard **C**onstraints)



Taylor series

$$V(U_{base}) = \bar{Y}^T \cdot \bar{Y}$$

Levenberg-Marquardt

$$V(U_{base} + \delta U) \approx (\bar{Y} + G \cdot \delta U)^T \cdot (\bar{Y} + G \cdot \delta U) + \boxed{\delta U^T \cdot \Lambda \cdot \delta U}$$

Steepest descent

$$\delta U = -(G^T \cdot G + \Lambda)^{-1} G^T \bar{Y}$$

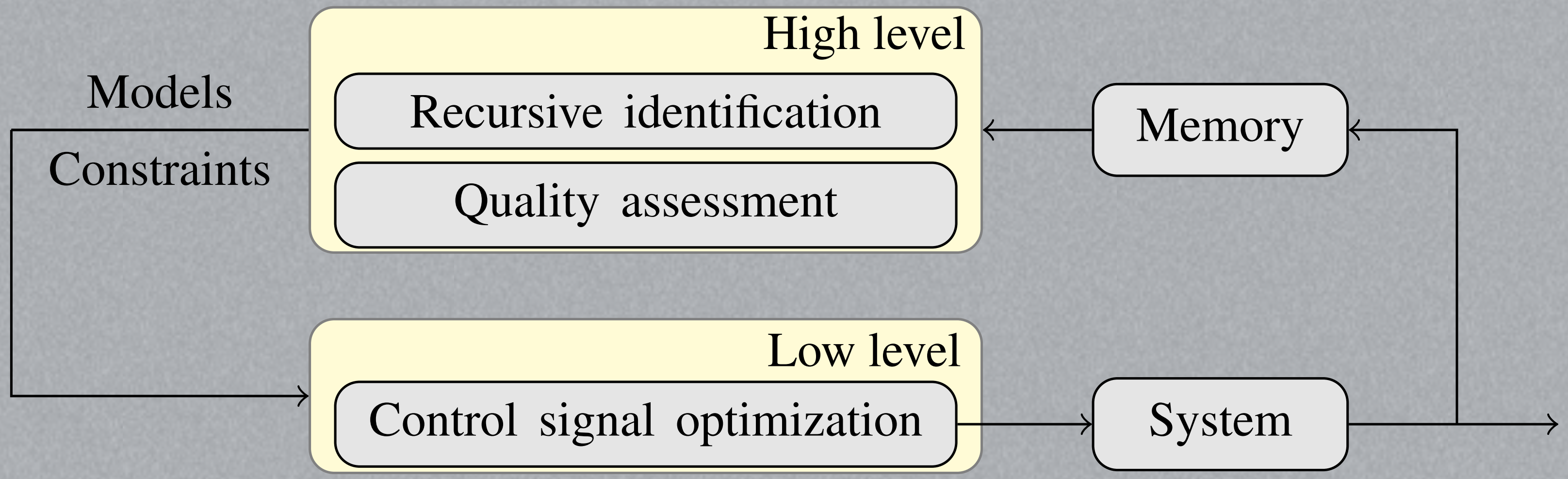
Repeat

$$U_{base} = U_{base} + \delta U$$

Guarantees descent: **F**ast

High-level Learning + Low-level MPC

Challenges: (i) Reference trajectories are **T**ime-varying, seldom known.
(ii) **F**ast and **D**istributed (switching) dynamics.

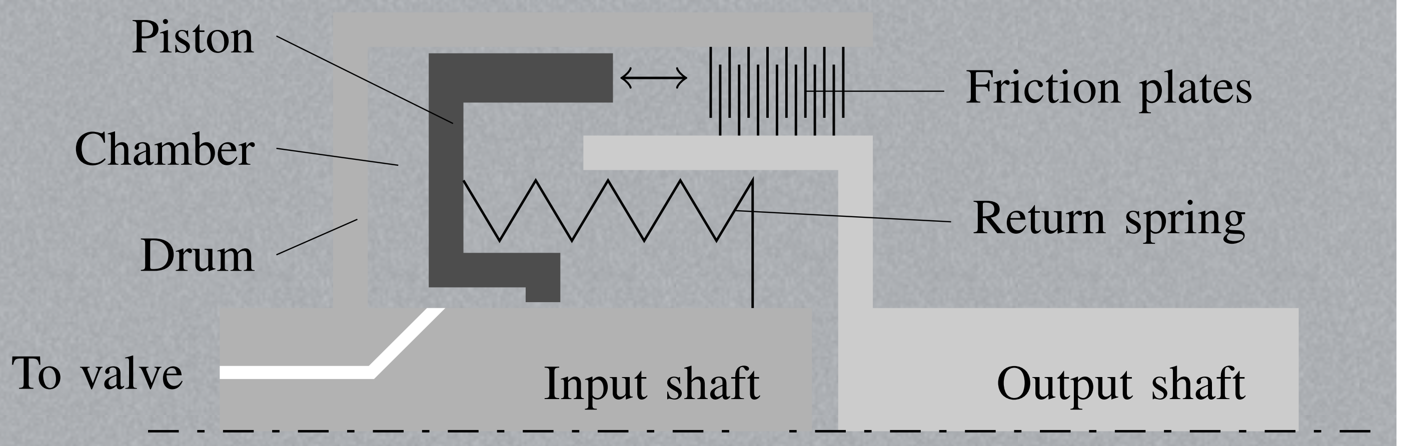


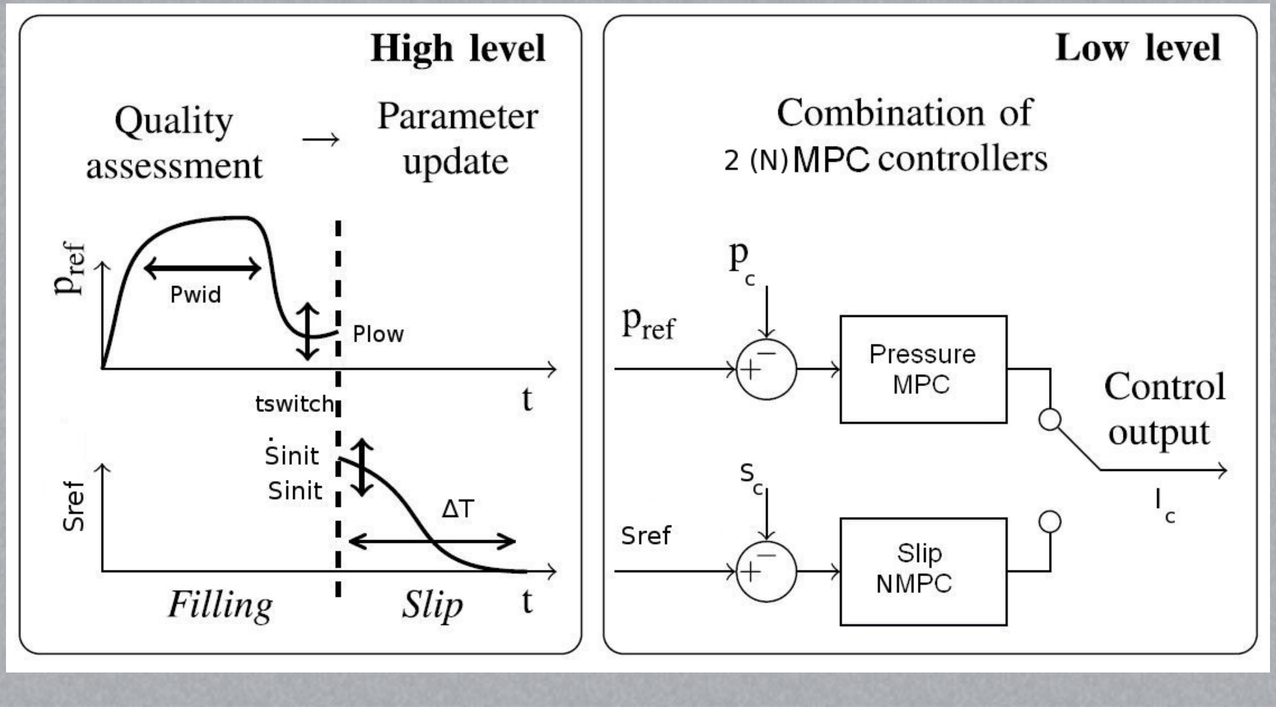
Two-level control: (i) High-level: learns parameterized references for time-varying conditions.
(ii) Low-level: fast tracking MPC over low-fidelity models.

Model-free techniques can also learn control but are inferior.

Method/Property	2l-NMPC	2l-ILC	IO	GA	RL
Modeling requirement	$\frown$	$\smile$	$\smile$	$\smile\smile$	$\smile\smile$
Learning rate	$\smile\smile$	$\smile$	$\smile\smile$	$\frown\frown$	$\frown$
Stability	$\smile$	$\smile$	$-\,-$	$-\,-$	$-\,-$
Learning transient/Safety	$\smile$	$\smile$	$\smile$	$\frown\frown$	$\frown\frown$
Multi-objective	$\smile$	$\smile$	$\smile$	$\smile\smile$	$\smile$

Two-level control: Automatic Transmission application

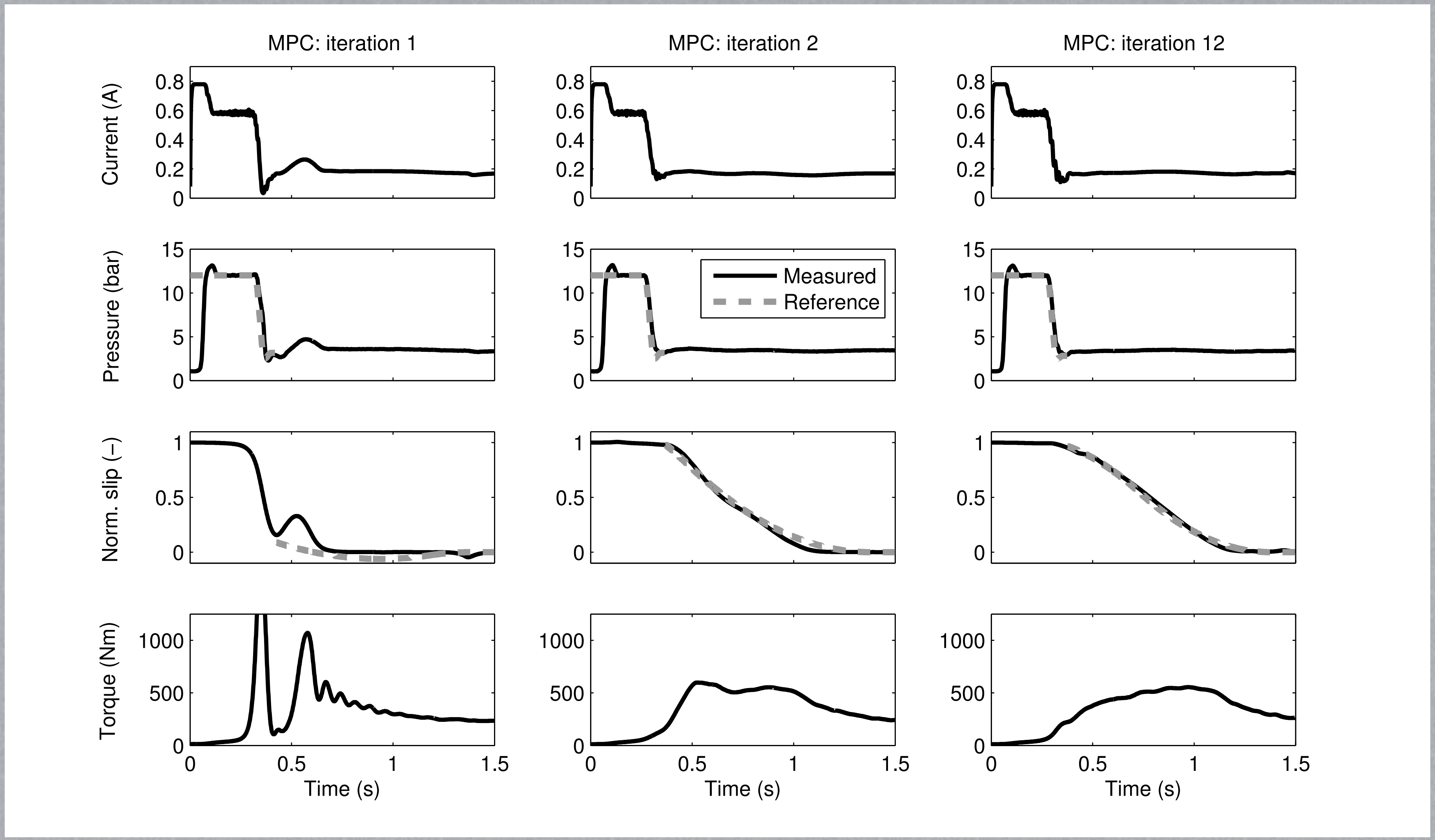




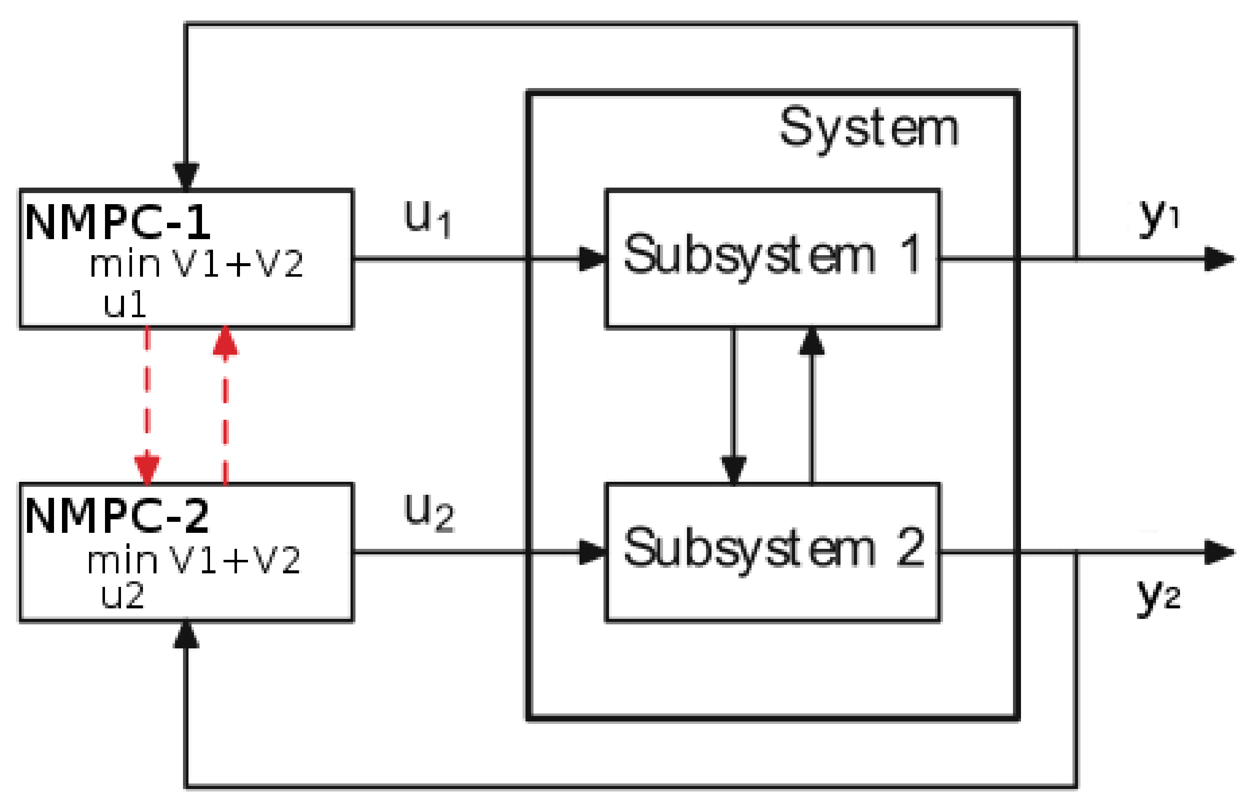
Objective:
Fast engagement
with minimum jerk.

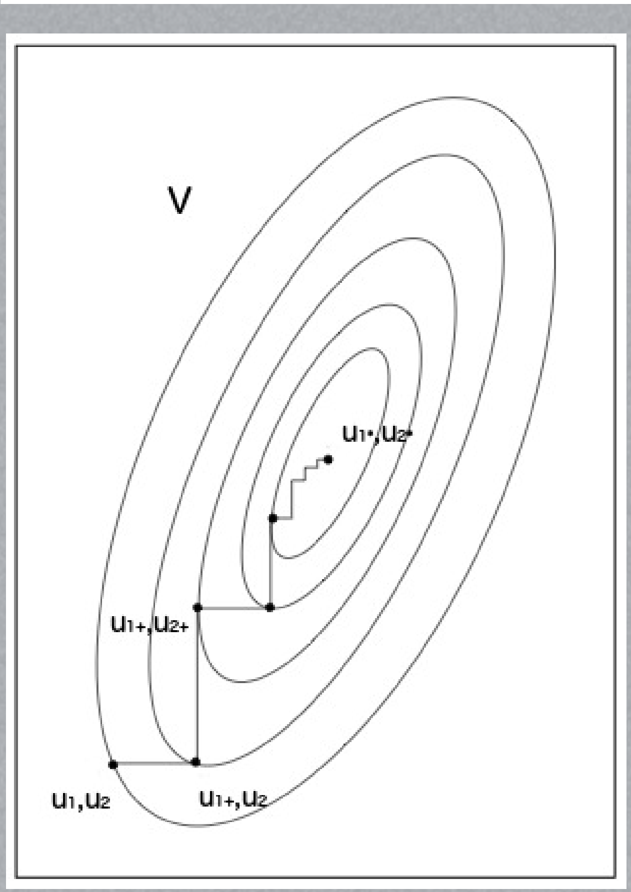
2L-MPC:
Learns the profiles
for optimal performance.

Real-time DSpace
implementation.



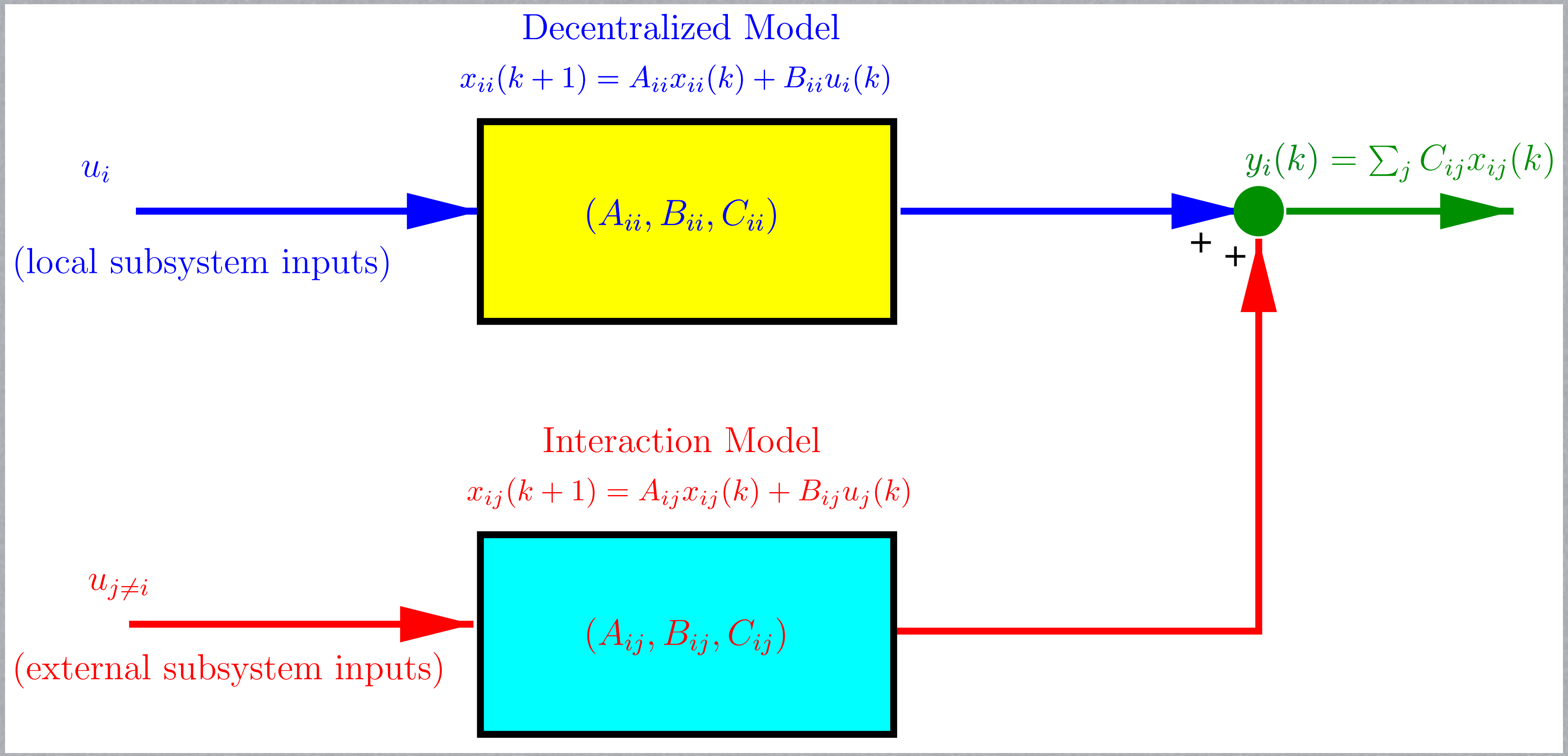
Distributed NMPC with Minimal Communication





Fast: Guaranteed coordinated descent

Distributed Modeling & Learning Time-varying Dynamics

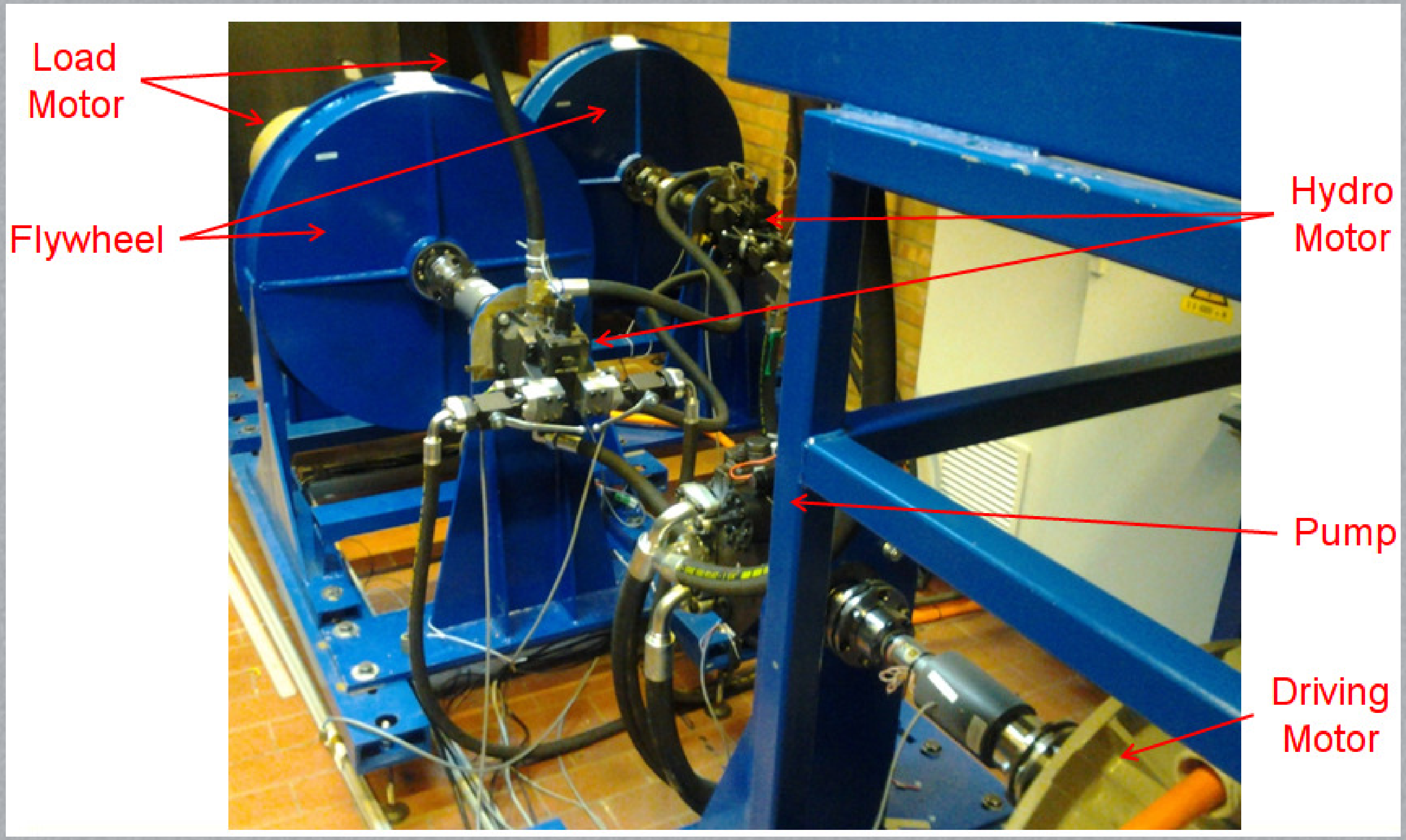


Model of ith subsystem: $y_i(t) = \phi_i^T(t)\cdot\theta_i(t) + \phi_{i-}^T(t)\cdot\theta_{i-}(t) + \nu_i(t)$

Learning Algorithm:

1. Initialize $\theta_1(t-1)$ and $\theta_2(t-1)$;
2. RLS-1. Compute $\theta_1(t) = \theta_1(t-1) + K_1\cdot(y_1(t) - \phi_1^T(t)\cdot\theta_1(t-1) - \phi_2^T(t)\cdot\theta_2(t-1))$ in the least squares sense and communicate to RLS-2.
3. RLS-2. Compute $\theta_2(t) = \theta_2(t-1) + K_2\cdot(y_2(t) - \phi_2^T(t)\cdot\theta_2(t-1) - \phi_1^T(t)\cdot\theta_1(t))$ in the least squares sense and communicate to RLS-1.
4. Go to step 2 at the next sampling period.

Experimental Results

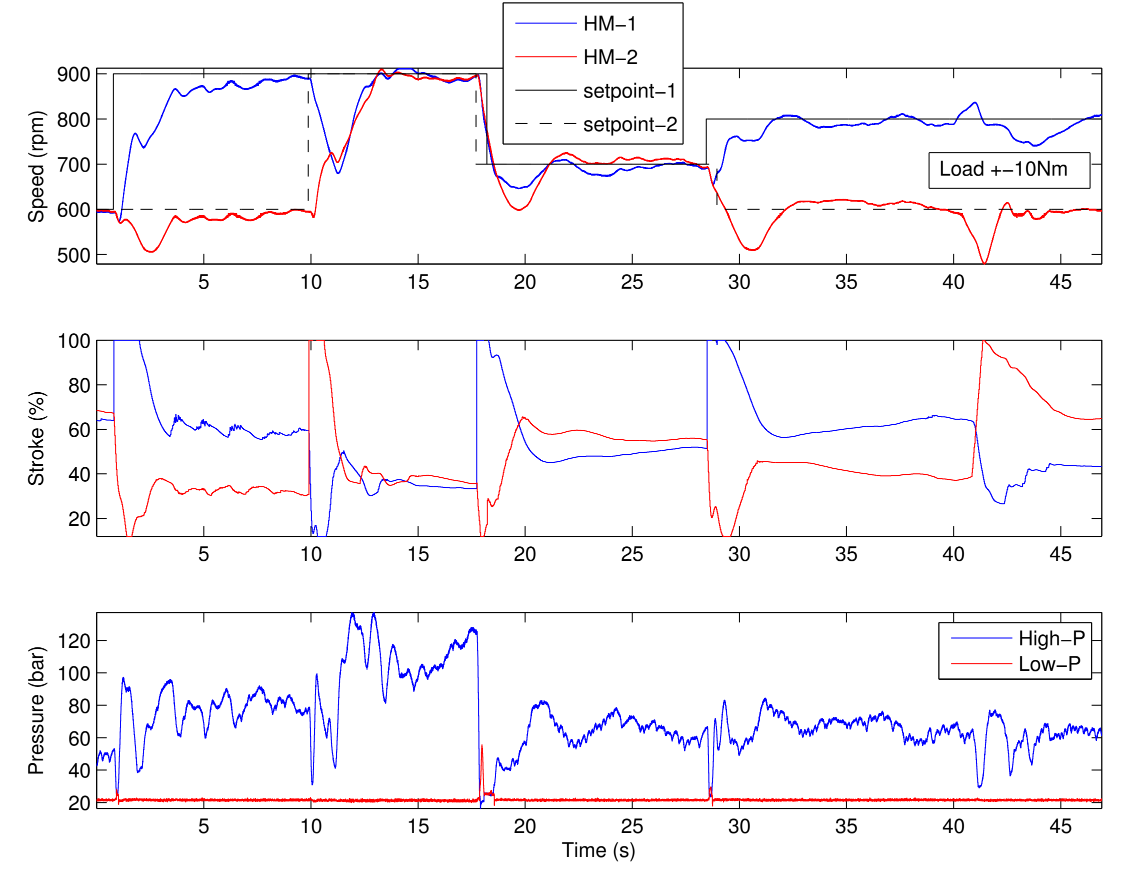


Hydrostatic Drivetrain

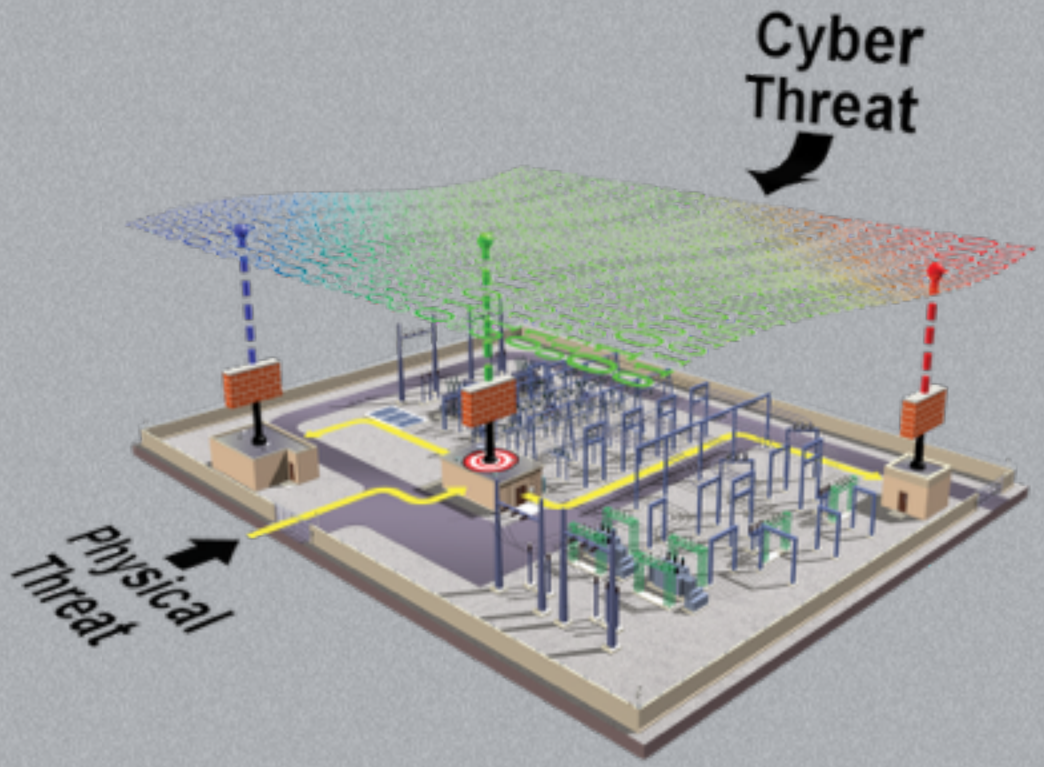
$$S_p\omega_p = S_{m1}\omega_{m1} + S_{m2}\omega_{m2}$$

Learns uncertain model parameters

Real-time implementation



Cyber Physical Systems Security




FAA: Boeing's New 787 May Be Vulnerable to Hacker Attack

WIRED

By Kim Zetter 01.04.08

The Boeing 787 Dreamliner aircraft makes its public d
outside the Boeing assembly plant in Everett, Washing
Photo: Robert Sorbo / Corbis

Boeing's new 787 Dreamliner passenger jet may b
computer networks that could allow passengers t
the U.S. Federal Aviation Administration.

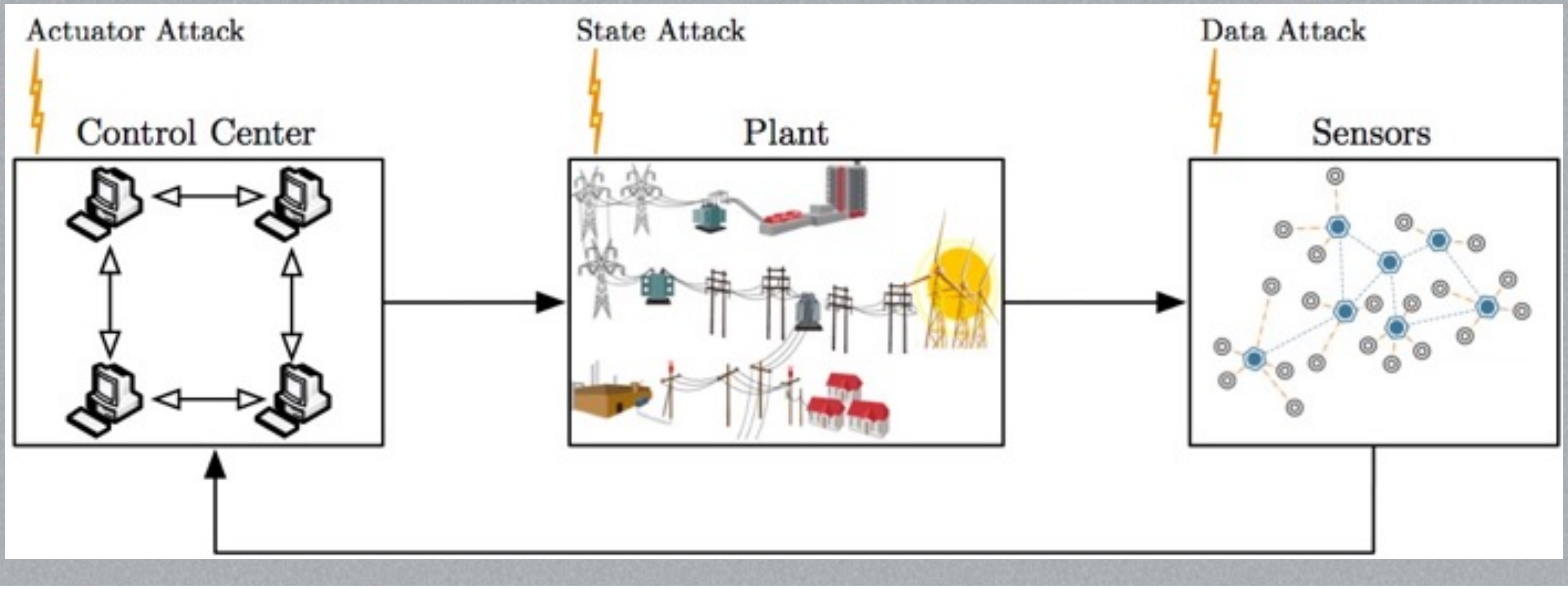
Vulnerabilities give hackers

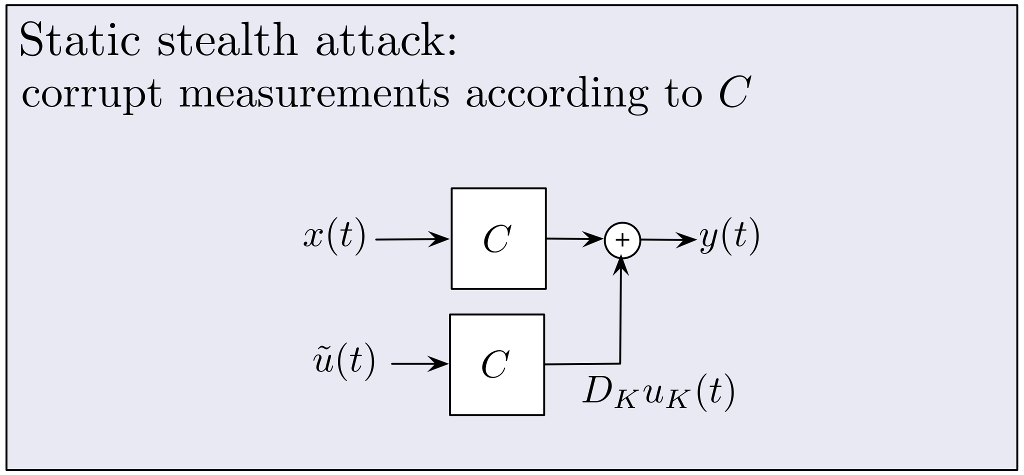
By Sean Gallagher | Published about 21 hours ago

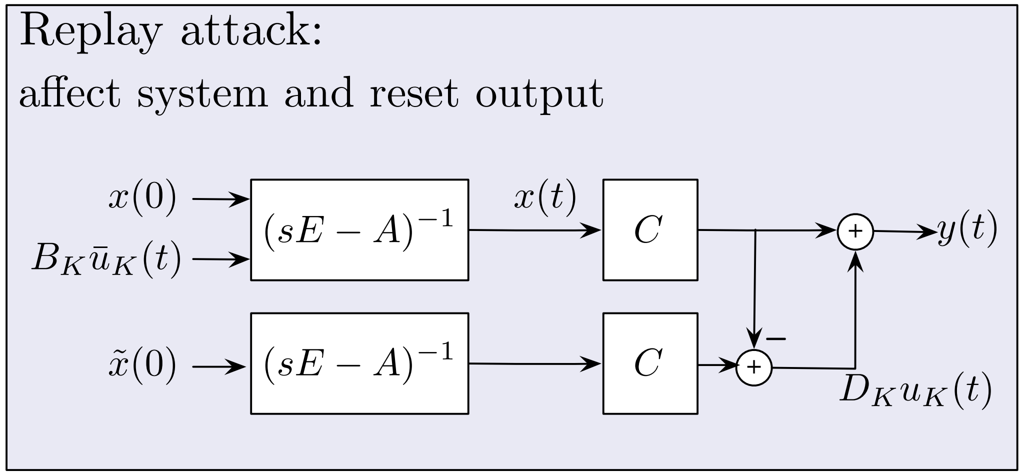
Researchers have demonstrated a vulnerabilit
prisons that could allow an outsider to remote
cell door mechanisms to shutting down intern
John Strauchs, who presented their research
conference in Miami, worked in Newman's ba

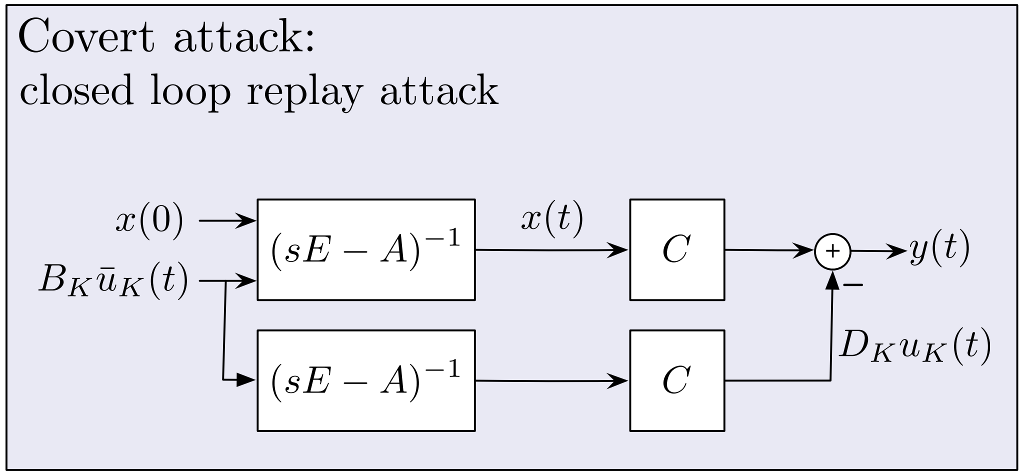
Aerospace Security Scenario: Source FAA

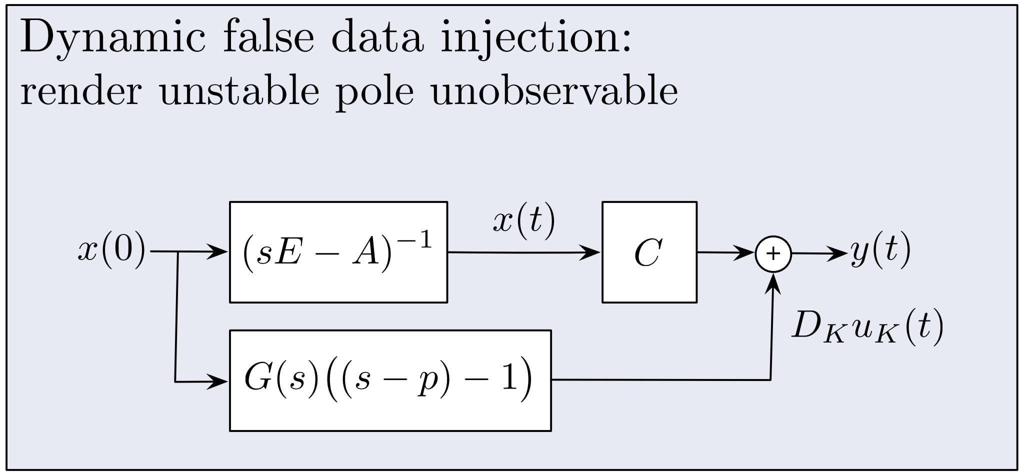
Model-based **V**ulnerability Analysis




Static stealth attack:
corrupt measurements according to C


Replay attack:
affect system and reset output

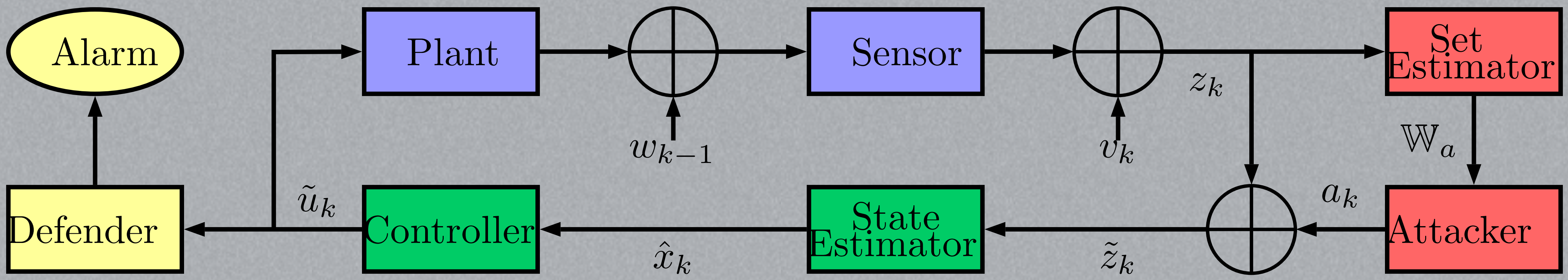

Covert attack:
closed loop replay attack


Dynamic false data injection:
render unstable pole unobservable

CSL:
COORDINATED
SCIENCE LAB

NSF

Control System Under Attack

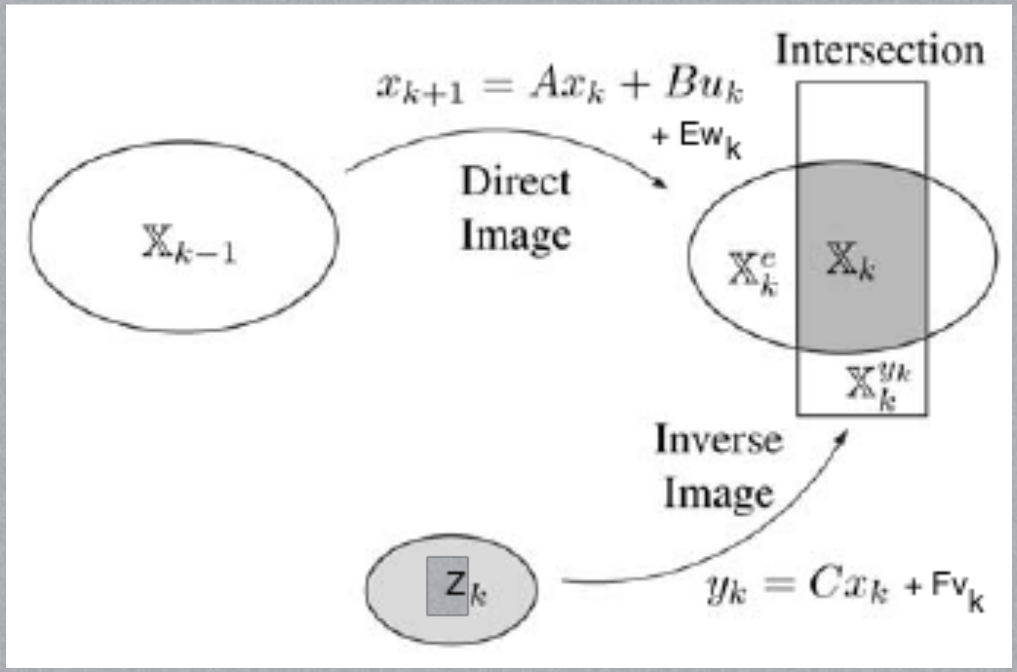


$$
\begin{aligned}
x_{k+1} &= Ax_k + Bu_k + Ew_k \\
z_k &= C_z x_k + F v_k, \\
y_k &= C_y x_k
\end{aligned}
$$

$$\tilde{z}_k = C_z x_k + F v_k + D a_k$$

$$x_k \in \mathbb{X},\ u_k \in \mathbb{U},\ y_k \in \mathbb{Y},\ w_k \in \mathbb{W},\ v_k \in \mathbb{V},\ \forall k$$

Model-based Monitor Design



Stealth attack: Robust Optimization

Guide to attack target

$$\tilde{z}_k^* = \arg\min_{\tilde{z}_k}\max_{w_k} ||y_{k+1|k} - T_a||^2 + ||\tilde{z}_k - \tilde{z}_{k-1}||_\Lambda^2$$

$$\text{s.t. } w_k \in \mathbb{W}_a$$

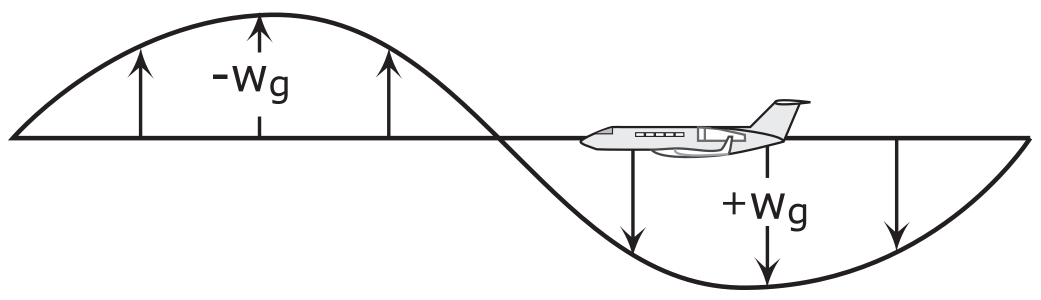
Under worst possible disturbances

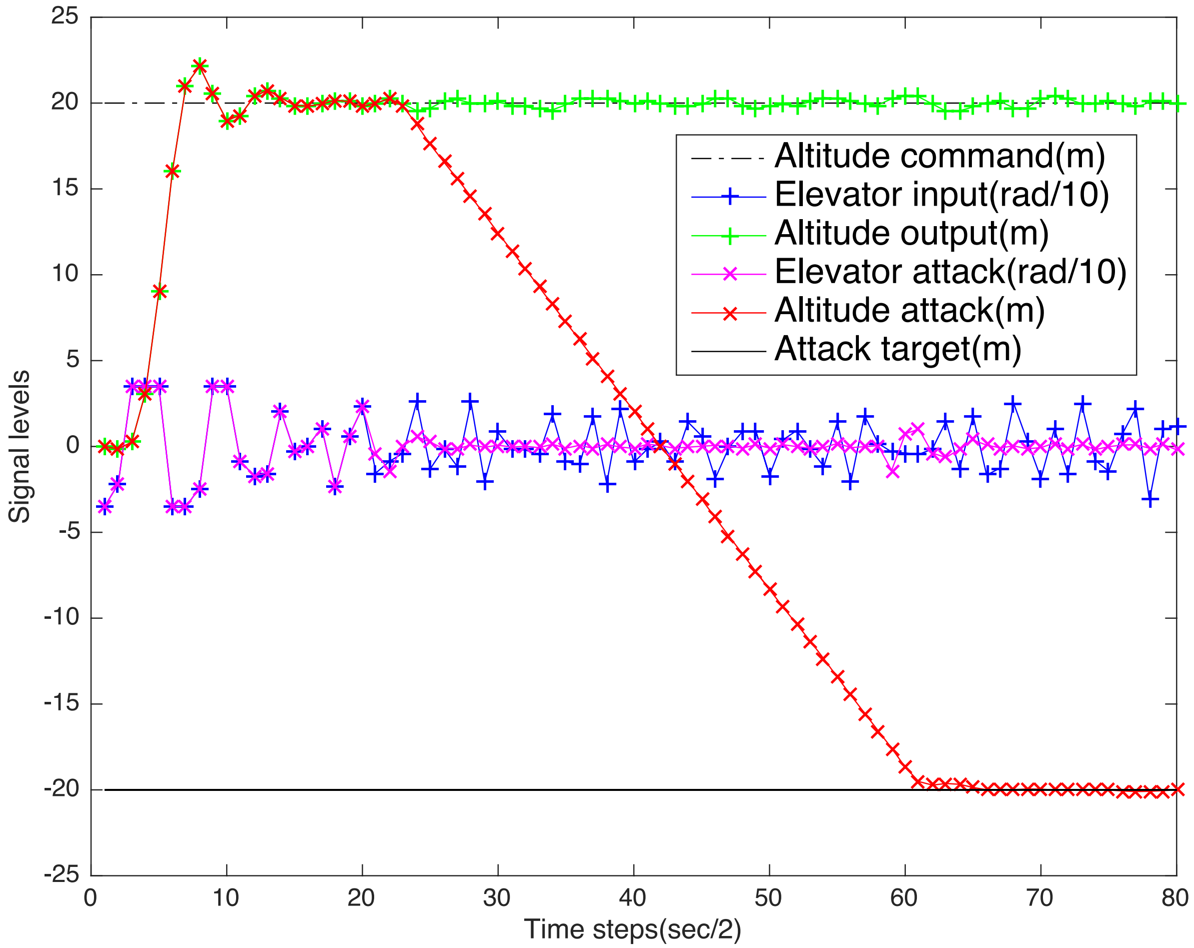
$$\tilde{z}_k \in \cap_{x\in\tilde{\mathbb{X}}_{k-1}} \tilde{R}(x, \mathbb{W}_a, \mathbb{V})$$

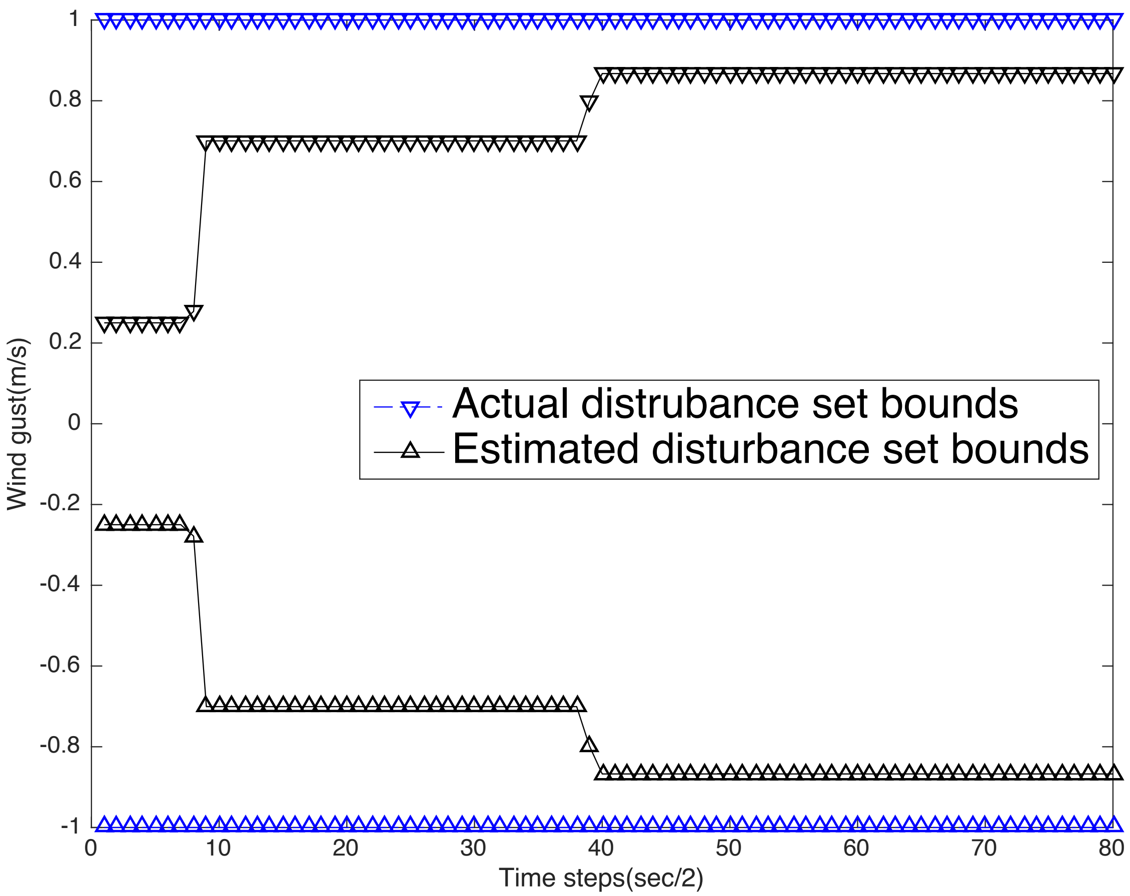
Remain undetected by the monitor

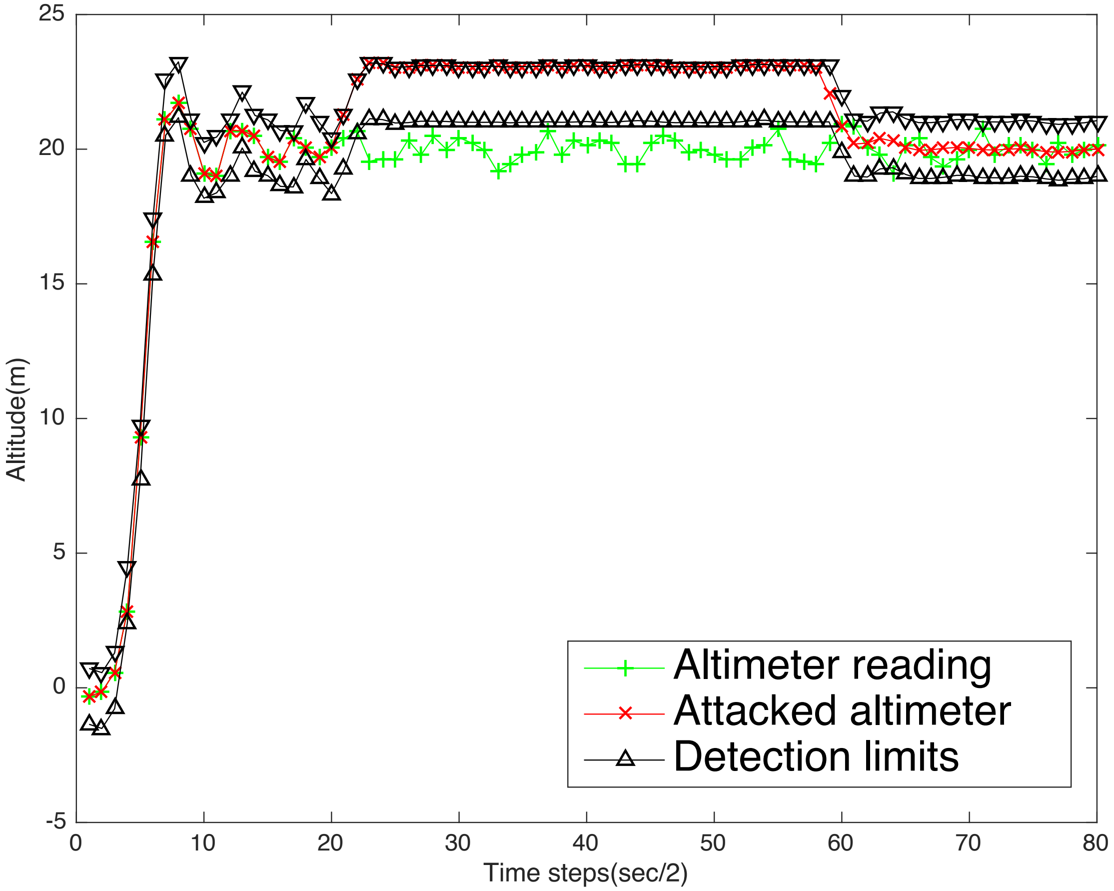
B 747 at 0.8 Mach and 40 Kft Trim

Altitude hold Subject to Wind gust

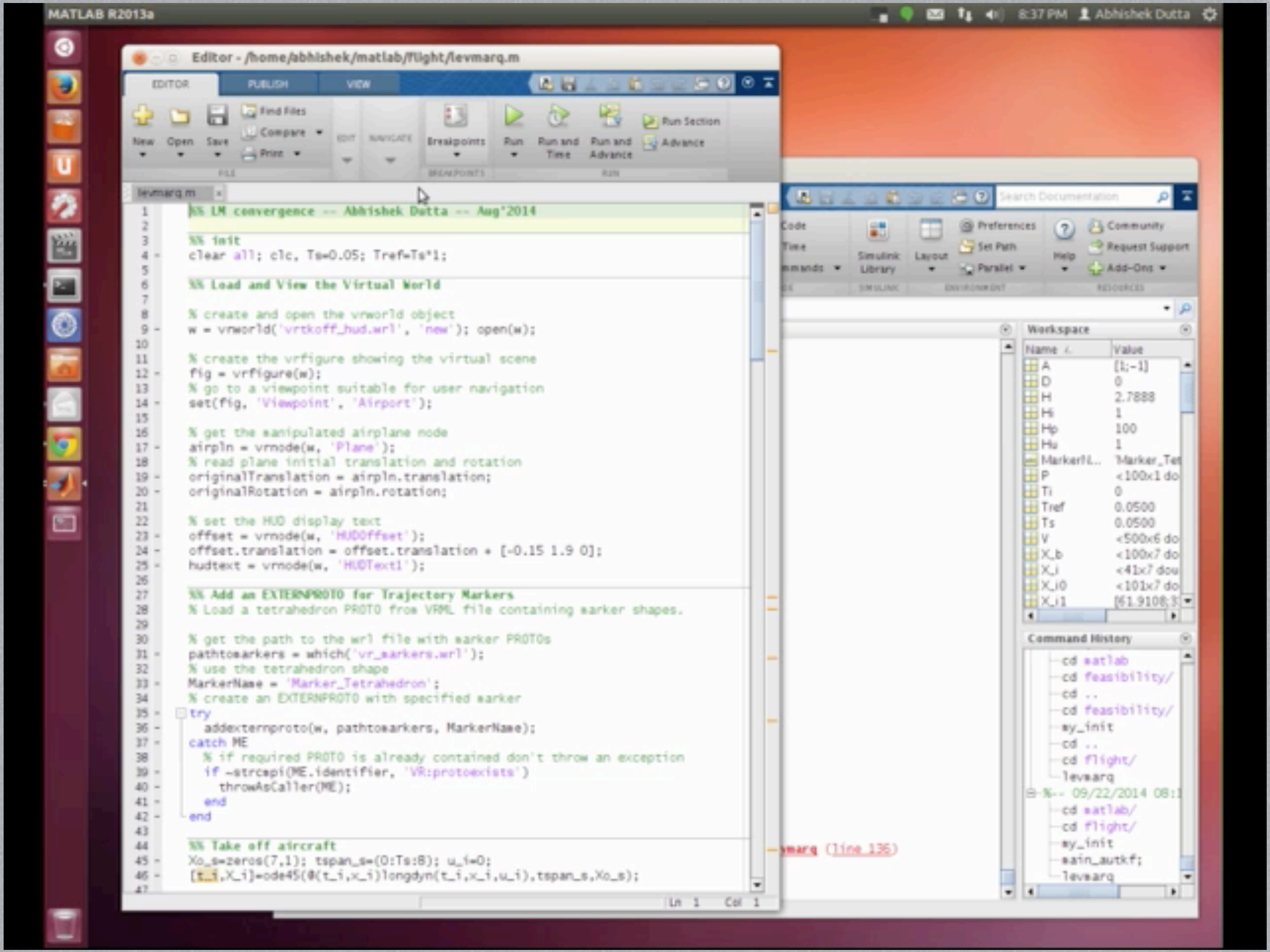








Demonstration of Aircraft Hijacking from 30m (but Limited) to 60m .



```
%% LM convergence -- Abhishek Dutta -- Aug'2014

%% init
clear all; clc, Ts=0.05; Tref=Ts*1;

%% Load and View the Virtual World

% create and open the vrworld object
w = vrworld('vrtkoff_hud.wrl', 'new'); open(w);

% create the vrfigure showing the virtual scene
fig = vrfigure(w);
% go to a viewpoint suitable for user navigation
set(fig, 'Viewpoint', 'Airport');

% get the manipulated airplane node
airpln = vrnode(w, 'Plane');
% read plane initial translation and rotation
originalTranslation = airpln.translation;
originalRotation = airpln.rotation;

% set the HUD display text
offset = vrnode(w, 'HUDOffset');
offset.translation = offset.translation + [-0.15 1.9 0];
hudtext = vrnode(w, 'HUDText1');

%% Add an EXTERNPROTO for Trajectory Markers
% Load a tetrahedron PROTO from VRML file containing marker shapes.

% get the path to the wrl file with marker PROTOs
pathtomarkers = which('vr_markers.wrl');
% use the tetrahedron shape
MarkerName = 'Marker_Tetrahedron';
% create an EXTERNPROTO with specified marker
try
  addexternproto(w, pathtomarkers, MarkerName);
catch ME
  % if required PROTO is already contained don't throw an exception
  if ~strcmpi(ME.identifier, 'VR:protoexists')
    throwAsCaller(ME);
  end
end

%% Take off aircraft
Xo_s=zeros(7,1); tspan_s=(0:Ts:8); u_i=0;
[t_i,X_i]=ode45(@(t_i,x_i)longdyn(t_i,x_i,u_i),tspan_s,Xo_s);
```

```
marq (line 136)
```

Workspace

Name	Value
A	[1;-1]
D	0
H	2.7888
Hi	1
Hp	100
Hu	1
MarkerN...	'Marker_Tet
P	<100x1 do
Ti	0
Tref	0.0500
Ts	0.0500
V	<500x6 do
X_b	<100x7 do
X_i	<41x7 dou
X_i0	<101x7 do
X_i1	[61.9108;3

Command History

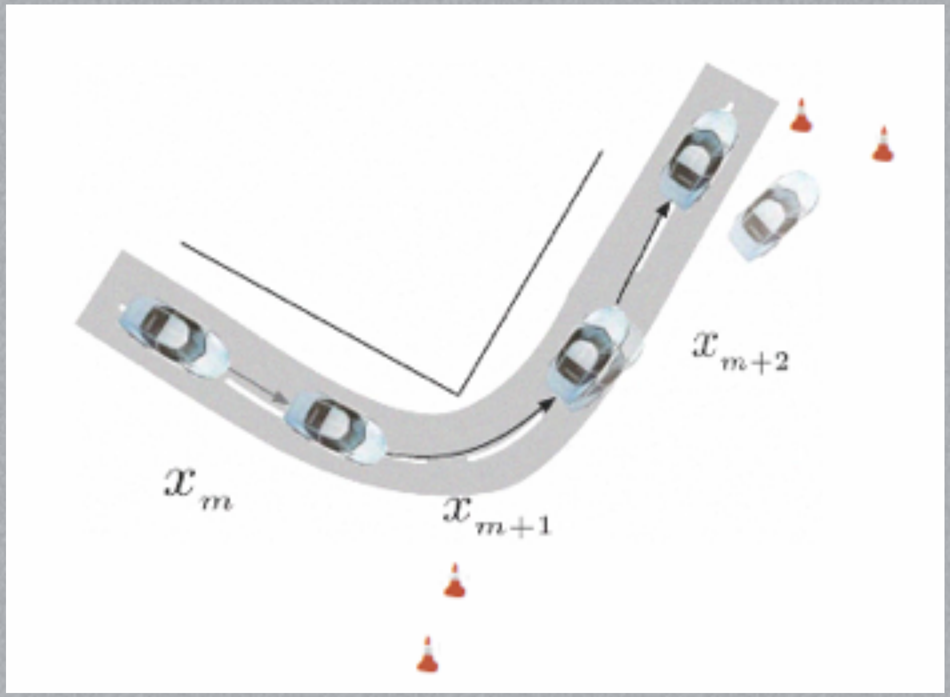
```
cd matlab
cd feasibility/
cd ..
cd feasibility/
my_init
cd ..
cd flight/
levmarq
%-- 09/22/2014 08:1
cd matlab/
cd flight/
my_init
main_autkf;
levmarq
```

Towards resilience - I: Bounded error estimator

$$\hat{x}_k =_k \Phi\Phi_z^\dagger \mathbf{z} + ({}_k\Gamma -_k \Phi\Phi_z^\dagger \Gamma_z)\mathbf{u}$$

$$\mathbb{E} = ({}_k\Xi -_k \Phi\Phi_z^\dagger \Xi_z)(\mathbb{W} \times \ldots \times \mathbb{W}) \oplus -_k\Phi\Phi_z^\dagger F(\mathbb{V} \times \ldots \times \mathbb{V})$$

$$\hat{\mathbb{W}} \triangleq E\mathbb{W} \oplus A\mathbb{E} \oplus (-\mathbb{E})$$



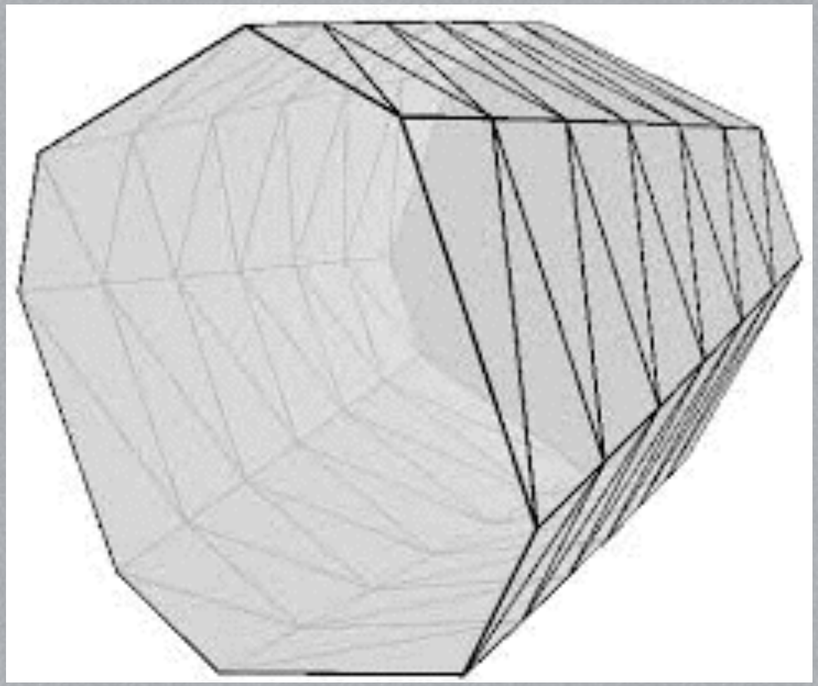
Towards resilience - II: Robust **M**PC

$$\mathbf{u}^* = \arg\min_{u(.|.)} \sum_{i=0}^{P-1} L(\hat{x}_{i+1|k}, \hat{u}_{i|k})$$

$$\text{s.t. } \hat{x}_{k+1} = A\hat{x}_k + B\hat{u}_k$$

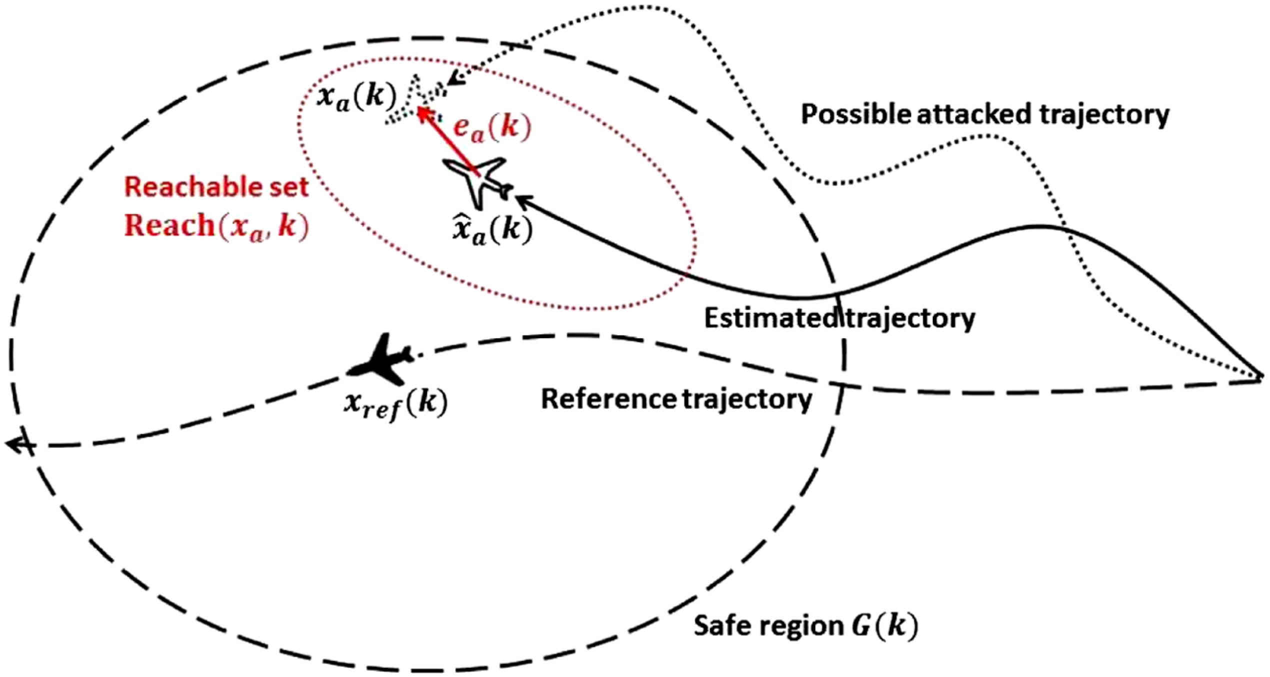
$$\hat{x}_{0|k} = x_k, \quad \hat{x}_{1|k} \in \mathbb{C}_\infty \sim \hat{\mathbb{W}}$$

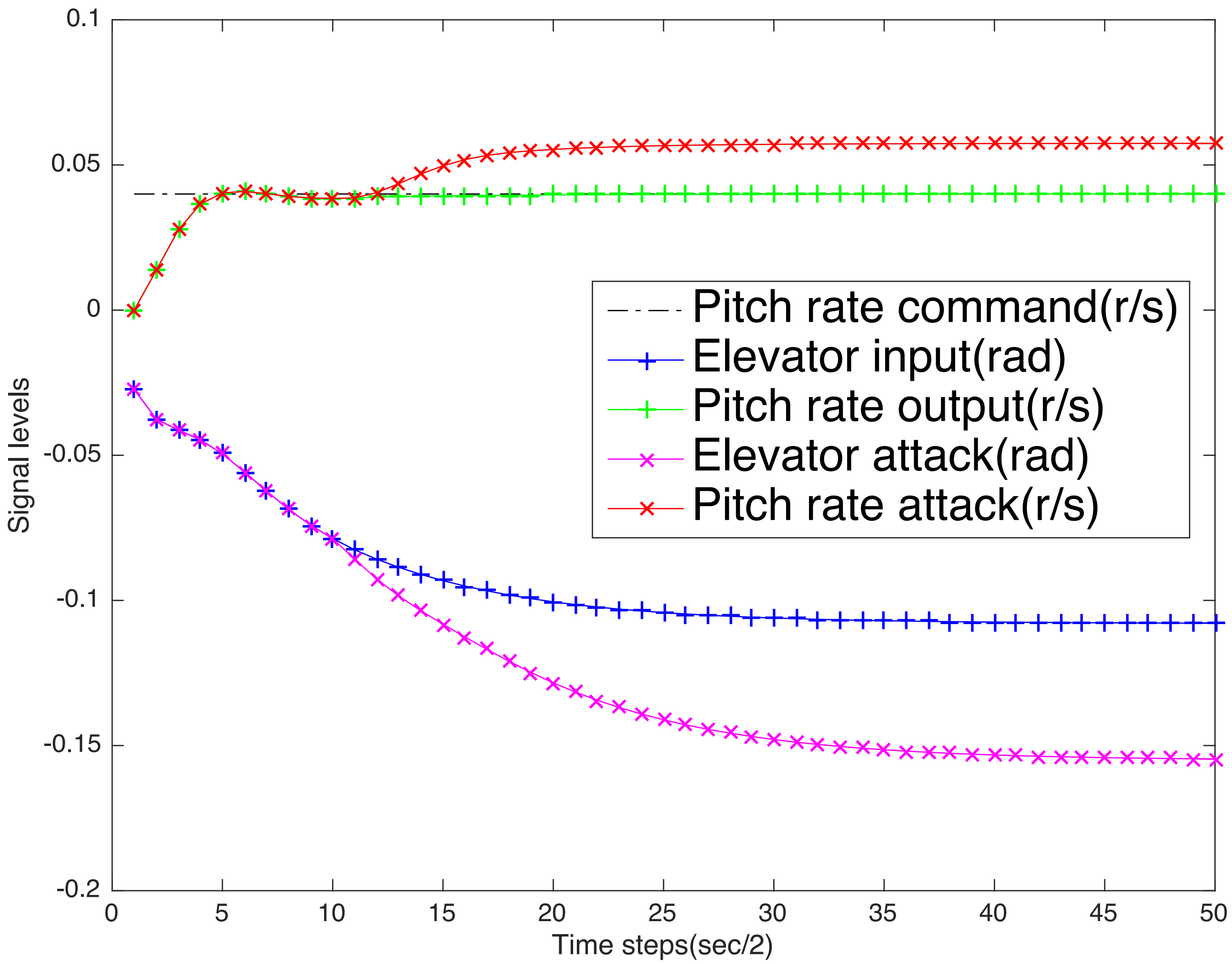
$$\hat{x}_{l+1|k} \in \hat{\mathbb{X}}, \ \hat{u}_{l|k} \in \mathbb{U}, \quad l = 0, \ldots, P-1$$

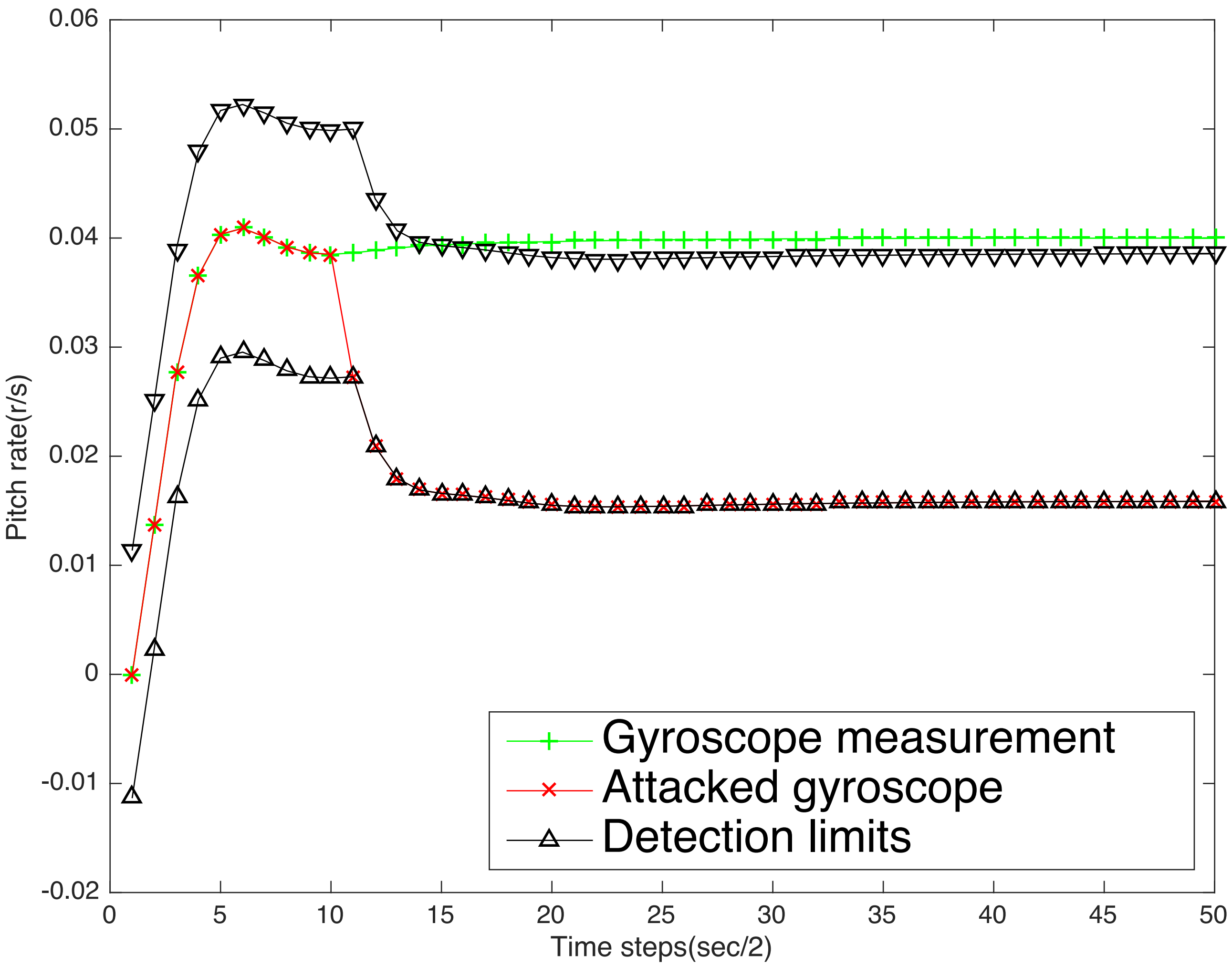


Resilience on Pitch Rate

Results



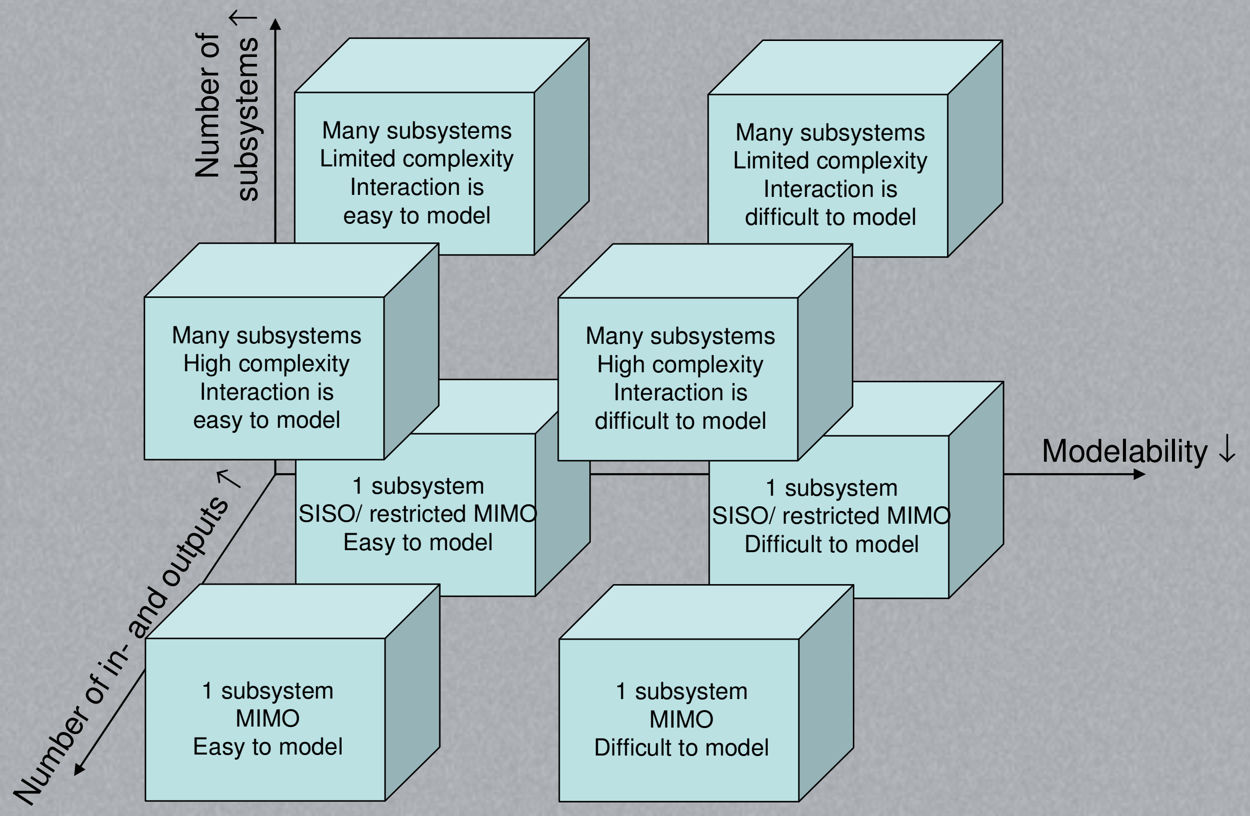


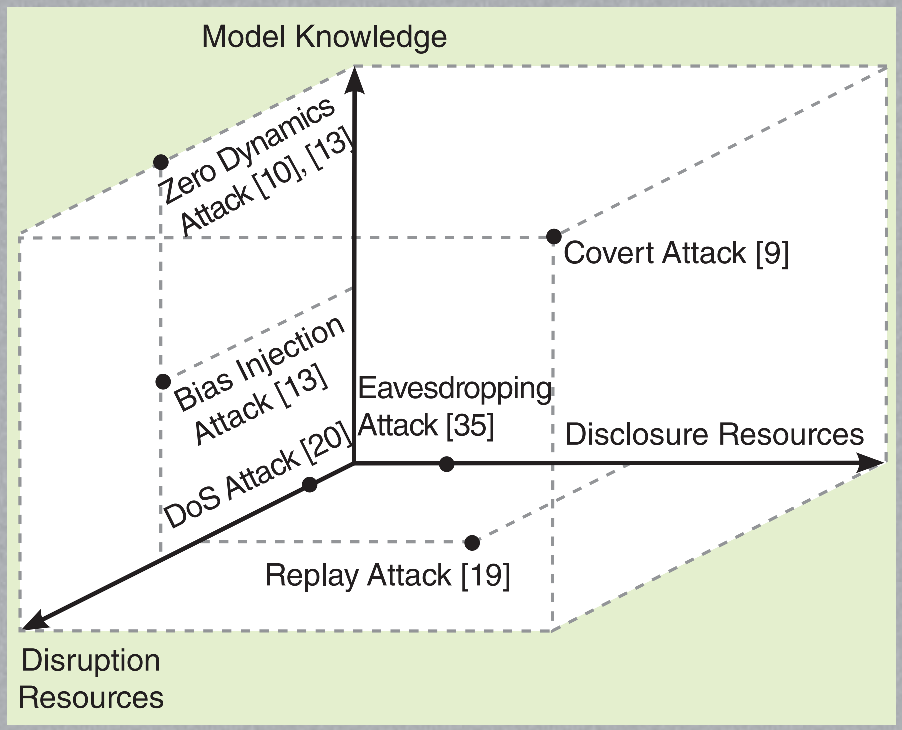


Perspectives (Big picture)

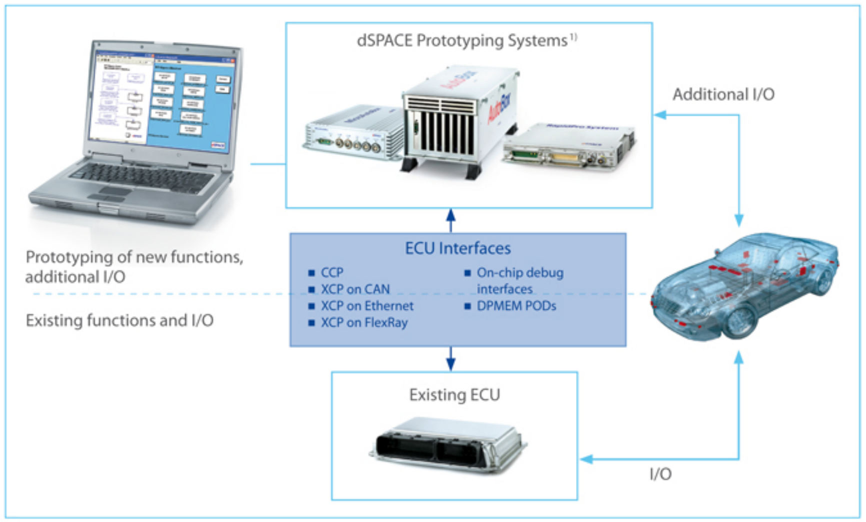
(i) **Integration**: New techniques of system design and compositions to distributed control implementation that preserves the robustness, performance and security properties.

(ii) **Adaptation**: Continuous learning and automatic synthesis of distributed models and controllers necessary.





Perspectives (Concrete)



1) **M**odeling: System identification of (non) linear and modular distributed systems.
2) Model-based **C**onstrained control: MPC, NMPC, DMPC.
3) Robustness to **T**ime-variance: Learning reference trajectory, distributed model parameters.
4) Certified system design: (i) Guarantees on recursive feasibility, stability and convergence. (ii) Resilient control and estimation over **V**ulnerability models.
5) **D**istributed systems: **F**ast MPC, NMPC, DMPC implementations on embedded platforms.