Reverse Engineered MPC for Tracking with Systems That Become Uncertain

Edward N. Hartley (edward.hartley@eng.cam.ac.uk)
Jan M. Maciejowski (jmm@eng.cam.ac.uk)

European Control Conference 2014, Strasbourg, France
Wednesday 25th June 2014, 10:20–10:40

Cambridge University Engineering Department

Aircraft Robustness of Inner-Loop Control Law to Loss of Airspeed Information

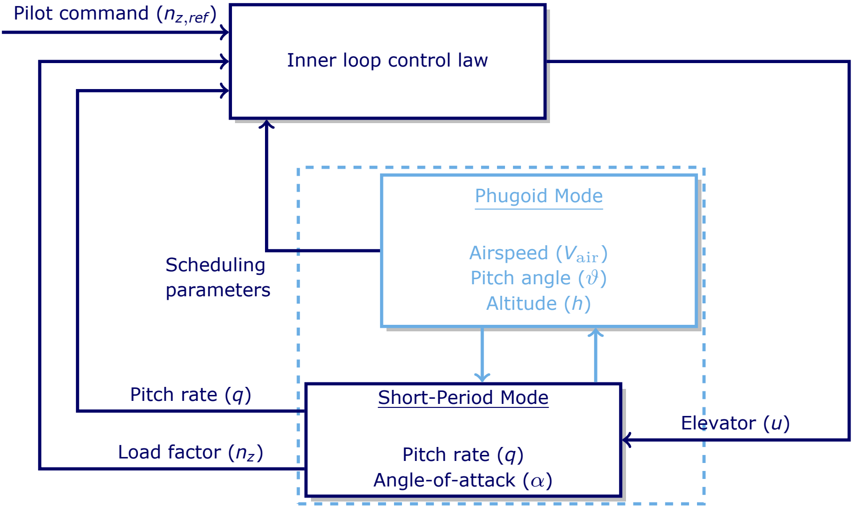
- Controls "short-period" mode.
- Tracks "load-factor" reference commanded by the pilot or outer-loop autopilot.

  Load factor closely related to normal acceleration.
- Commonly a gain-scheduled proportional-integral control law with feedback of pitch rate and load factor ("C^*")

  not controlling airspeed, but scheduled by airspeed.
- Constraints? Currently ad-hoc, but LTV-MPC applicable.

What if we no longer have the scheduling information?

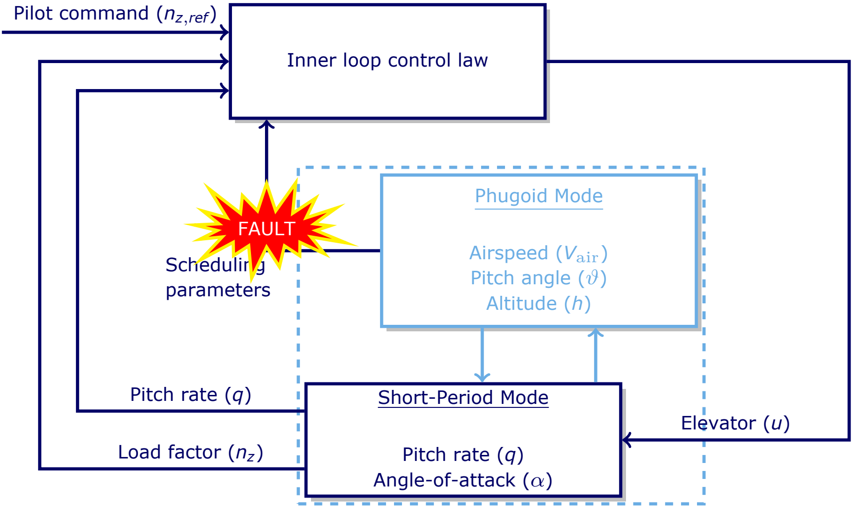
- e.g. due to a detected sensor failure



Pilot command ($n_{z,ref}$)
Inner loop control law
Phugoid Mode
Airspeed (V_{air})
Pitch angle (ϑ)
Altitude (h)
Scheduling parameters
Pitch rate (q)
Load factor (n_z)
Short-Period Mode
Pitch rate (q)
Angle-of-attack (α)
Elevator (u)

Background and Motivation

The Motivating Scenario



Parameter-Varying State-Space Model

$$
\begin{aligned}
x(k+1) &= A(\theta)x(k) + B(\theta)u(k) + d(\theta) \\
y_r(k) &= C_r x(k) \\
y_m(k) &= Cx(k)
\end{aligned}
$$

- θ represents the scheduling information
- When θ is measurable: linear time-varying system
- When θ is not measurable: uncertain system

Want to design a controller with the following properties

- Handles multivariable systems
- Respects asymmetric input and output constraints
- Has adequate small-signal closed-loop performance
- Modest computational requirements

- **Tracks non-zero setpoints**
- **Robustness to parametric uncertainty**
- **Interchangeable with a nominal high performance design**

Background and Motivation

Challenges

Parametric Uncertainty

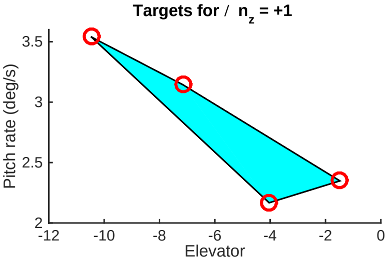
- Too large to approximate as additive?
- Looking at “robust” rather than “adaptive” methods

Computational requirements

- 250 ms sampling time
- Don’t want to solve LMIs online!
- Don’t want exponentially growing trees of predictions

Uncertain Equilibrium Pair

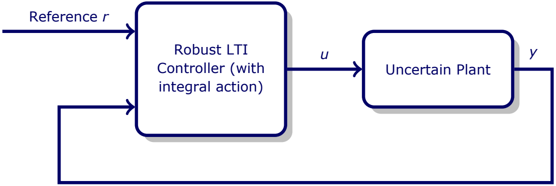
- Not regulating to the origin
- Cannot do change of variables to turn into regulation to the origin!



Proposed method
"Reverse Engineering"

Assumption

- A suitable (unconstrained) linear robust controller of an appropriate form already exists; or
- It is relatively easy to design such a controller.



Method

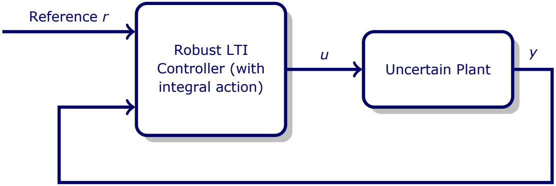
- Transform the baseline into an observer-based controller
- Partition into feedback and feedforward
- Enforce constraints using online optimisation

The Baseline Control Law

$$\left[\begin{array}{c} x_k(k+1) \\ \hline u(k) \end{array}\right] = \left[\begin{array}{c|cc} I & -C_r & I \\ \hline K_2 & K_1 & 0 \end{array}\right] \left[\begin{array}{c} x_K(k) \\ \hline x(k) \\ r(k) \end{array}\right]$$

Since this is an integral control law...

If $r(k)$ and θ are constant, then $\lim_{k\to\infty} y_r(k) \to r(k)$.



Nominal model

$$\left[\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & 0 \end{array}\right]$$

$$\hat{A} \approx A(\theta)$$
$$\hat{B} \approx B(\theta)$$
$$\hat{C} = I$$

Baseline Regulator

$$\left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array}\right]$$

$$A_K = I$$
$$B_K = -C_r$$
$$C_K = K_2$$
$$D_K = K_1$$

Disturbance Augmented Model

$$\bar{x} = \begin{bmatrix} x \\ w \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} \hat{A} & I \\ 0 & I \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} \hat{B} \\ 0 \end{bmatrix}, \quad \bar{C} = \begin{bmatrix} \hat{C} & 0 \end{bmatrix}$$

Baseline regulator re-written in (reduced-order) observer form

$$\bar{z}(k+1) = F\bar{z}(k) + G\bar{y}(k) + T\bar{B}\bar{u}(k)$$ Observer Dynamics

$$\hat{\bar{x}}(k) = H_2\bar{z}(k) + H_1\bar{y}(k)$$ State/Disturbance Estimate

$$\bar{u}(k) = K_c\hat{\bar{x}}(k) + D_Q(\bar{y}(k) - \bar{C}\hat{\bar{x}}(k))$$ Control Input

Where…

$$F = A_K - T\bar{B}C_K \qquad G = B_K - T\bar{B}D_K$$

$$K_c = C_K T + D_K\bar{C}$$

D_Q satisfies: $C_K = (K_c - D_Q\bar{C})H_2 \qquad D_K = (K_c - D_Q\bar{C})H_1$

$$T\bar{A} - (A_K - T\bar{B}C_K)T - (B_K - T\bar{B}D_K)\bar{C} = 0$$ **NON-UNIQUE**

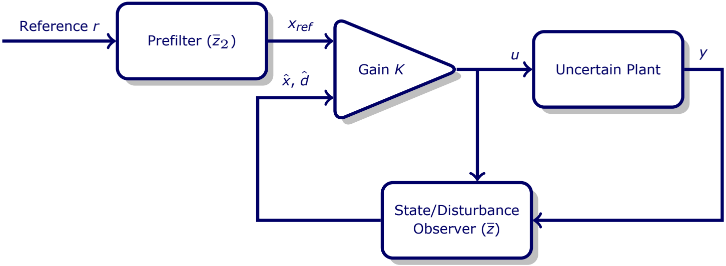
$$\begin{bmatrix} H_1 & H_2 \end{bmatrix}\begin{bmatrix} \bar{C} \\ T \end{bmatrix} = I$$ **NON-UNIQUE**

Tracking regulator

$$\bar{z}(k+1) = F\bar{z}(k) + G\bar{y}(k) + TB\bar{u}(k) \quad \text{Observer Dynamics}$$

$$\bar{z}_2(k+1) = F\bar{z}_2(k) + r(k) \quad \text{Prefilter Dynamics}$$

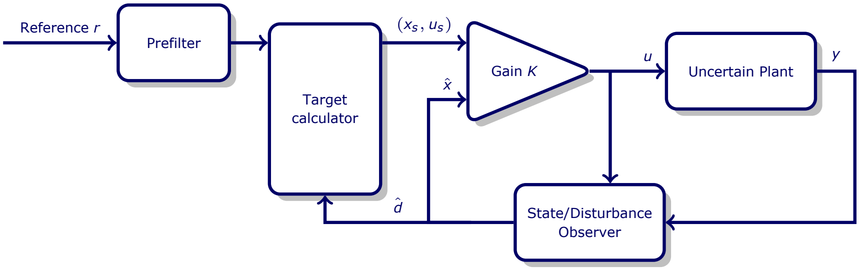
$$\begin{aligned}\bar{u}(k) &= K_c(H_2\bar{z}(k) + H_1\bar{y}(k) + H_2\bar{z}_2(k)) \\ &\quad + D_Q(\bar{y} - \bar{C}(H_1\bar{y}(k) + H_2\bar{z}(k) - H_2\bar{z}_2(k)))\end{aligned}$$



Comments so far...

- Uncertainty $\implies$ no separation principle
- State disturbance captures uncertain affine term and parameter uncertainty
- Reproducing the controller, not the closed-loop system: nominal model does not have to be accurate
- Non-symmetric Riccati equation non-unique (well known)
  - Realisation does not affect unconstrained input/output behaviour
  - Does affect internal signals
- Degrees of freedom in non-unique H_1 and H_2 will be used later.

Now want to transform one step further…



Taking the observer-form a step further

$$\hat{\hat{x}} = \begin{bmatrix} \hat{x}(k) \\ \hat{w}(k) \end{bmatrix}, \quad K_c = \begin{bmatrix} K_{cx} & K_{cd} \end{bmatrix}.$$

We want to re-write the observer-based control law as:

$$\overline{u}(k) = K_{cx}(\hat{x}(k) - x_s(k)) + u_s(k)$$

subject to:

$$(\hat{A} - I)x_s(k) + \hat{B}u_s(k) = -\hat{w}(k)$$
$$C_r x_s(k) = r_p = C_r x_{\mathrm{ref}}.$$

where

$$x_{\mathrm{ref}} = f(\hat{x}(k), y(k), \overline{z}_2(k))$$

(Prove by equating terms: see the paper for details!)

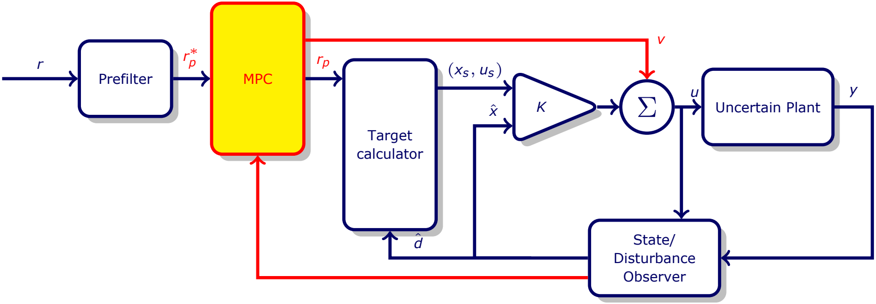
Steady state consistency

- Turns out that even though integrating control law is reproduced, the internal variables are not guaranteed to be consistent, i.e.

$$\lim_{k\to\infty} C_r x_s(k) \neq \lim_{k\to\infty} C_r x(k).$$

- Conditions found on non-unique H_1 and H_2 to enforce this: must choose the “correct” pseudoinverse of $\begin{bmatrix} C \\ T \end{bmatrix}$.
- Tedious algebra: see paper for details.

- Online MPC used to compute additive perturbation to:
  1. the reference input to the target calculator;
  2. the input applied to the plant.



- Very similar structure to method of Pannocchia (2004).
- Key difference: target calculator and gain are designed from an existing linear baseline control law

Prediction model for augmented plant

$$\begin{bmatrix} x(k+1) \\ \bar{z}(k+1) \end{bmatrix} = \mathcal{A}(\theta) \begin{bmatrix} x(k) \\ \bar{z}(k) \end{bmatrix} + \mathcal{B}(\theta) \begin{bmatrix} r_p(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} \mathbf{d}(\theta) \\ 0 \end{bmatrix}$$

- $v(k)$ is an additive input perturbation that the MPC manipulates
- $r_p(k)$ is a manipulated reference signal

Nominal constraints

- State constraints $\mathbb{X}$
- Input constraints $\mathbb{U}$

Control Invariant Set

$$\mathcal{C} \triangleq \Big\{(x(k), \bar{z}(k)) : \exists r_p \text{ satisfying constraints with } v(k) = 0, \\ \text{such that } (x(k+1), \bar{z}(k+1)) \in \mathcal{C}, \quad \forall \theta \in \Theta\Big\}.$$

Constrained MPC

When the variable θ is unknown, at each time step the online MPC formulation can compute $v(k)$ and $r_p(k)$ as:

$$\min_{r_p(k), v(k)} v(k)^T R_v v(k) + (r_p(k) - r_p^*(k))^T S (r_p(k) - r_p^*(k))$$

subject to $u(k) \in \mathbb{U}$, $x(k) \in \mathbb{X}$, and

$$\mathcal{A}(\theta) \begin{bmatrix} x(k) \\ \bar{z}(k) \end{bmatrix} + \mathcal{B}(\theta) \begin{bmatrix} r_p(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} \mathbf{d}(\theta) \\ 0 \end{bmatrix} \in \mathcal{C}, \forall \theta \in \Theta.$$

Nominal MPC

- When θ is known, a standard “linear-time-varying” MPC approach can be used to achieve better performance, failing over to the robust form when a fault occurs.
- Still use the reverse-engineered observer and target calculator
- Enforce the control invariant set constraint at every time step (or at least the first time step)

Plant Models

- Short-period longitudinal aircraft approximation extracted from publicly available B747 model
- Inputs in incremental form to allow rate constraints

$$\begin{bmatrix} q(k+1) \\ n_z(k+1) \\ u(k+1) \end{bmatrix} = A(\theta_i) \begin{bmatrix} q(k) \\ n_z(k) \\ u(k) \end{bmatrix} + B(\theta_i)\Delta u(k) + d(\theta_i)$$

Flight Points

Speed\Alt	5000 m	7500 m
160 m/s	1	
180 m/s		3
260 m/s	2	4

Constraints

- $-37T_s \leq \Delta u \leq 37T_s$ [deg/s]
- $-17 \leq u \leq 23$ [deg]
- $-2 \leq n_z \leq 1.5$ [g]
- $-2 \leq r_p \leq 1.5$ [g]

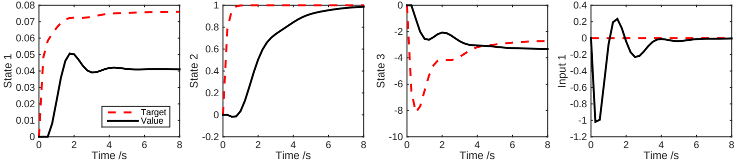
Baseline Control Law

- Designed by augmenting plant with integral of n_z tracking error and applying unconstrained version of RMPC of Kothare 1996: LMI-based feedback MPC to get a control gain
- Basically min-max LQR with multiple models, with an integrator
- Guaranteed to stabilise unconstrained plant for chosen realisations.

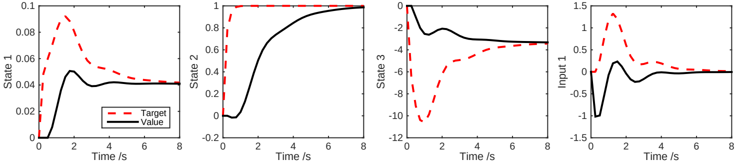
Reverse Engineering

- Nominal model for observer design: flight point 1.
- Dynamics separation: integrating modes in dynamics of $\overline{A} + \overline{B}K_c$

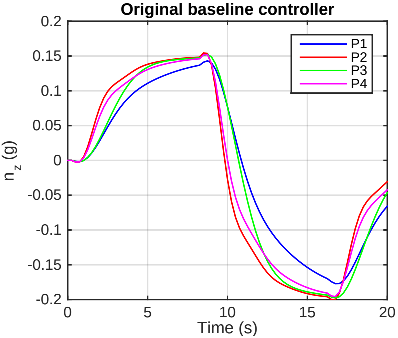
Mismatched model: arbitrary H_1, H_2 (inconsistent)



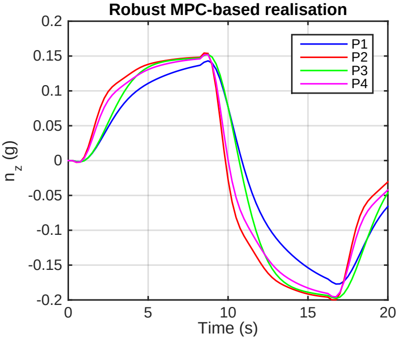
Mismatched model: proposed H_1, H_2 (consistent)



Baseline

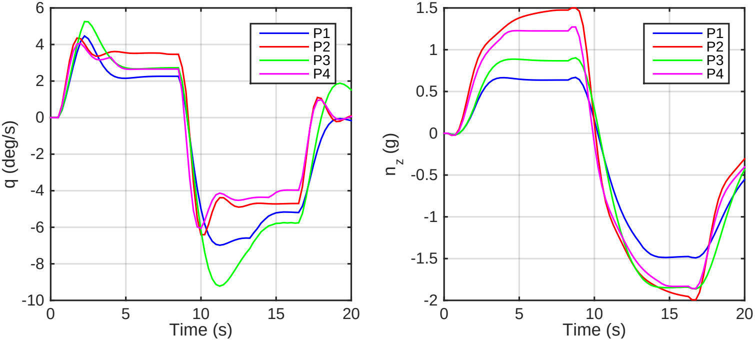


MPC



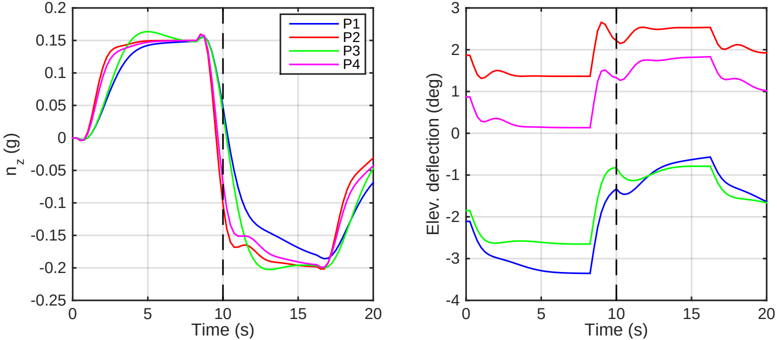
Demonstration

Robust enforcement of output constraints



Demonstration

Nominal to Robust Switchover



Conclusions

- An alternative way to design a constrained controller for **tracking non-zero setpoints** that is robust to **parametric uncertainty**
- Based on "**reverse engineering**" an existing robust control law into an observer-target-calculator-gain form
- **Constraint handling** facilitated by control invariant set
- Applied to flight control example

Future application challenges

- More detailed flight control example
- Complicating factors: sensor/filter dynamics, actuator dynamics
- Scheduling between altitudes