Nonlinear Control of Mechanical Systems in the Presence of Magnitude and Rate Saturation

Richard M. Murray

Mechanical Engineering/Control & Dynamical Systems

California Institute of Technology

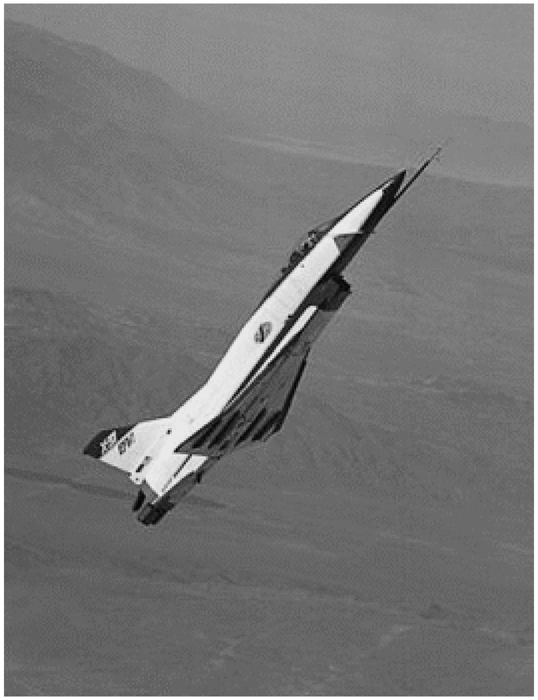
ONR Program Review

January 1999

Outline

- Motivation and approach
- Stabilization in the presence of magn/rate constraints
- Real-time trajectory generation w/ magn/rate constraints
- Summary and issues

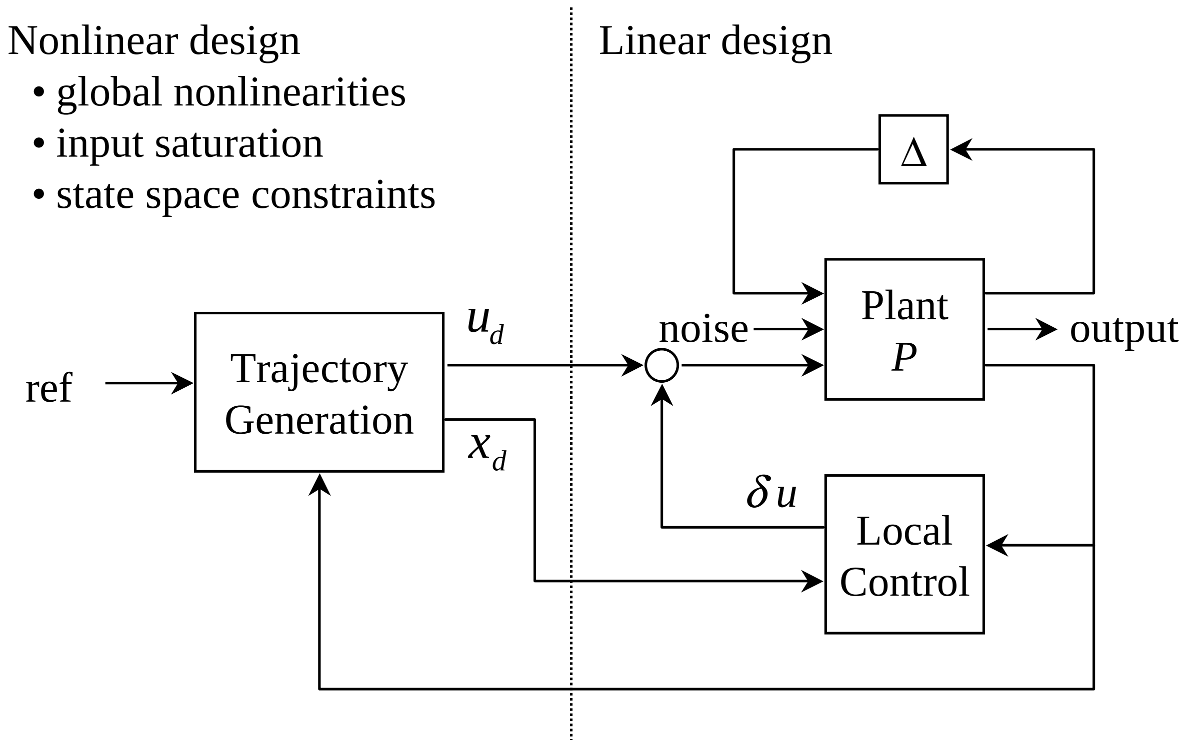
Motivation: Flight Control Trends



Features of highly aggressive, flight vehicles

- Tailless flight for LO $\Rightarrow$ non-conventional control surfaces
- Uninhabited flight: pilots at remote locations $\Rightarrow$ controllers have to do more
- Cost is increasingly a factor (especially for UAVs) + reduced development time

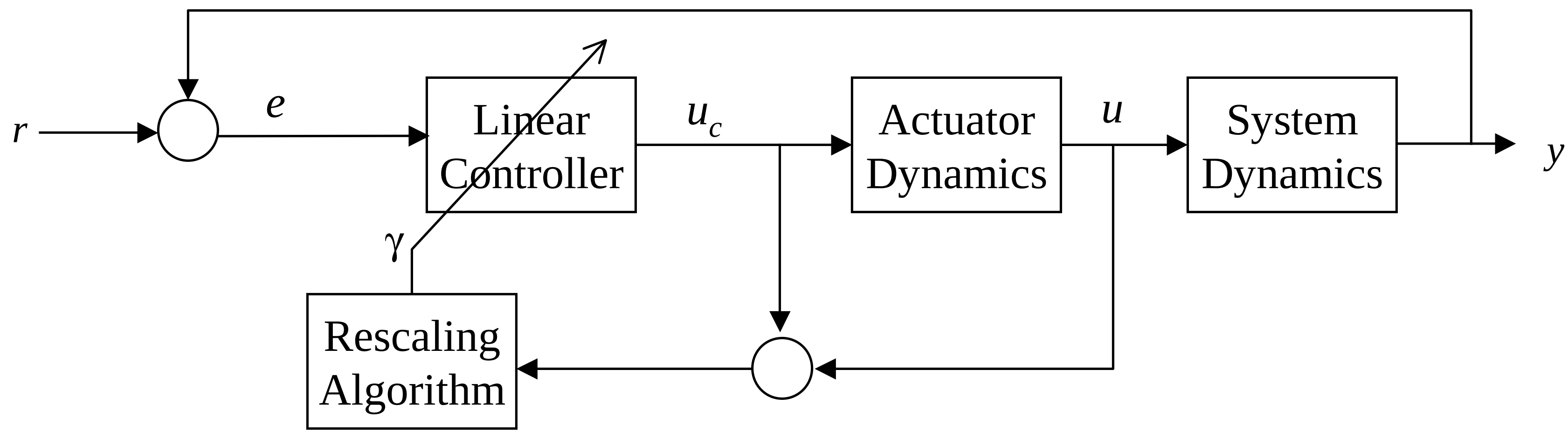
Approach: Two Degree of Freedom Design



- Use *real-time* trajectory generation to construct (suboptimal) feasible trajectories
- Use local linear/nonlinear control for tracking & robust performance

Stabilization in the Presence of Magn/Rate Constraints

Lauvdal, Murray, Fossen (GNC 97, CDC 97)



Basic idea: rescale the control law when can actuator saturates

- *Linear* scaling of gain does not work (frozen system unstable)
- *Nonlinear, dynamic* scaling is required for stability and performance

Features

- Leaves linear control design unchanged when away from limits (nonlinear wrapper)
- Requires state feedback controller (so far); theory available only for restricted cases

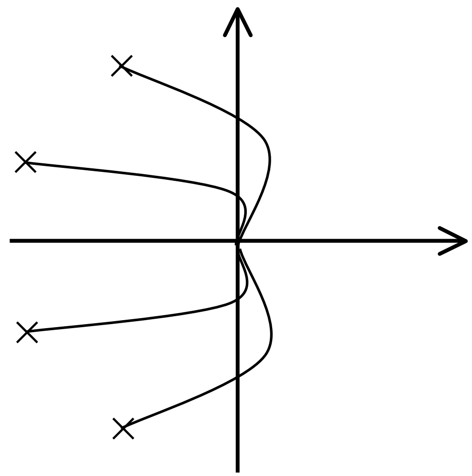
Related work by Teel, Megretski, Feron, and others

- Solve online Ricatti equation or use Lyapunov-based technique; state-based scaling

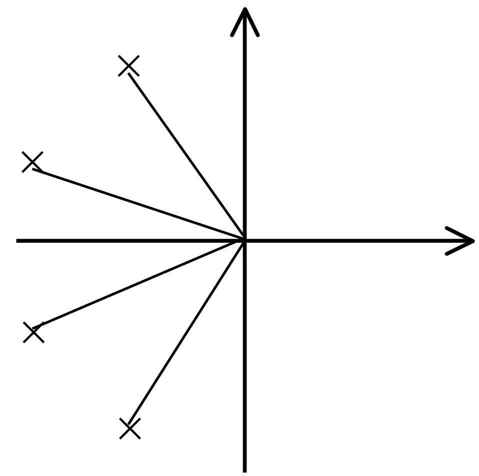
Main Idea: Nonlinear Rescaling to Preserve Damping Ratio

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3 \\
&\vdots \\
\dot{x}_n &= u
\end{aligned}
$$

$$u = k_1 x_1 + k_2 x_2 + \cdots + k_n x_n$$

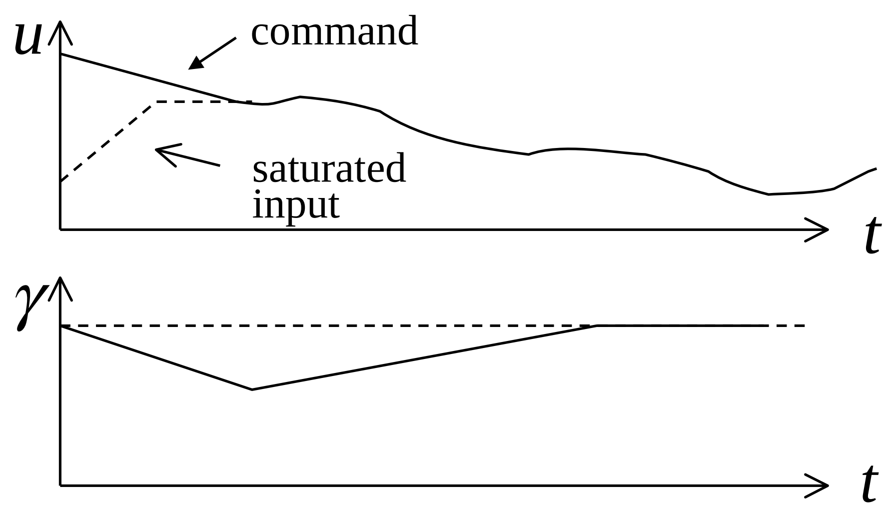


$$u = \gamma(k_1 x_1 + \cdots + k_n x_n)$$



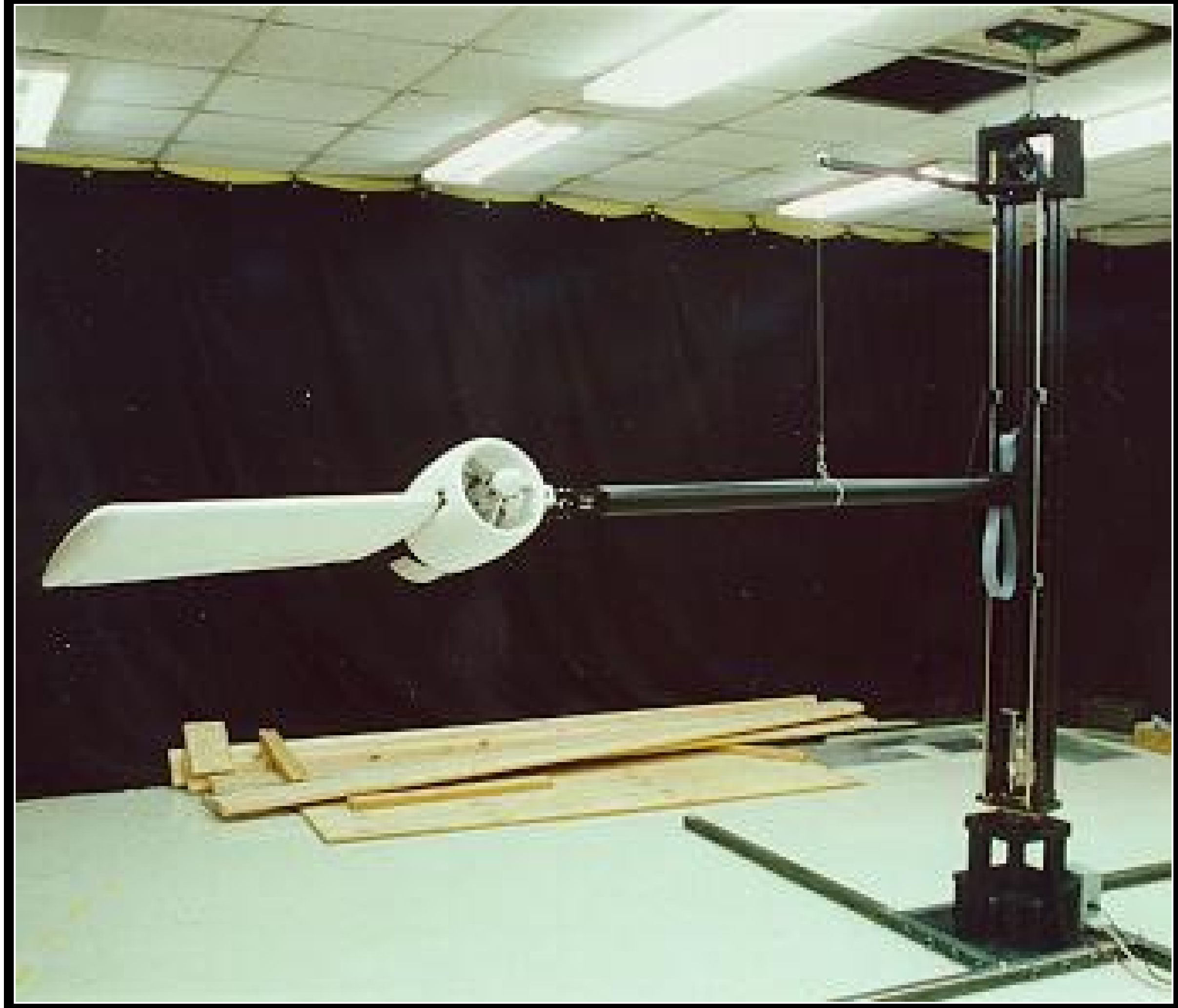
$$u = k_1 \gamma^n x_1 + \cdots + k_n \gamma x_n$$

$$\dot{\gamma} = \begin{cases} -\alpha(\gamma) & |u - u_c| > \varepsilon \\ +\beta & |u - u_c| \le \varepsilon \end{cases}$$



- Actual update law more complicated
- Balance decay/recovery rates
- Messy proof (see GNC, CDC papers)

Implementation on Caltech Ducted Fan



Features

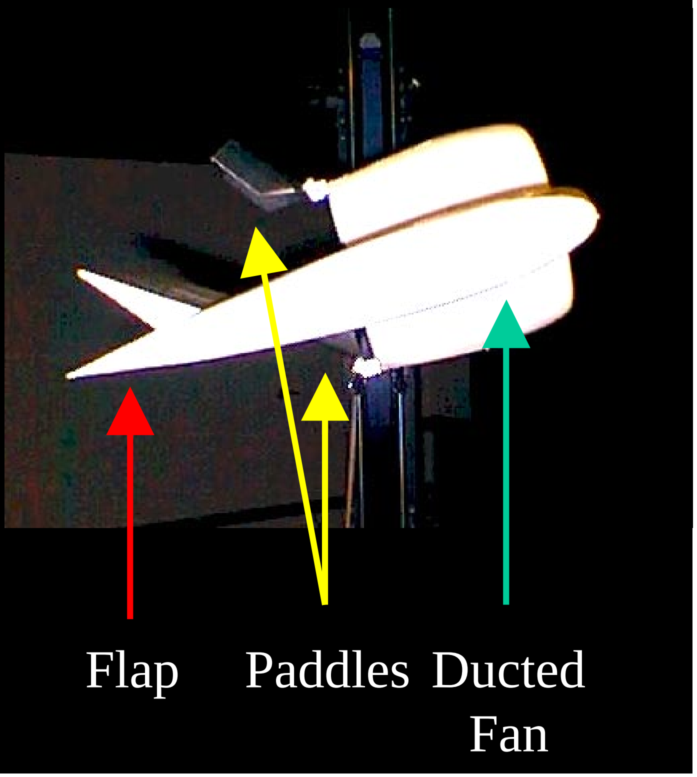
- Integrated fan/wing assembly
- Linear bearings, counterweight
- Floor, wrist slip rings

Usage

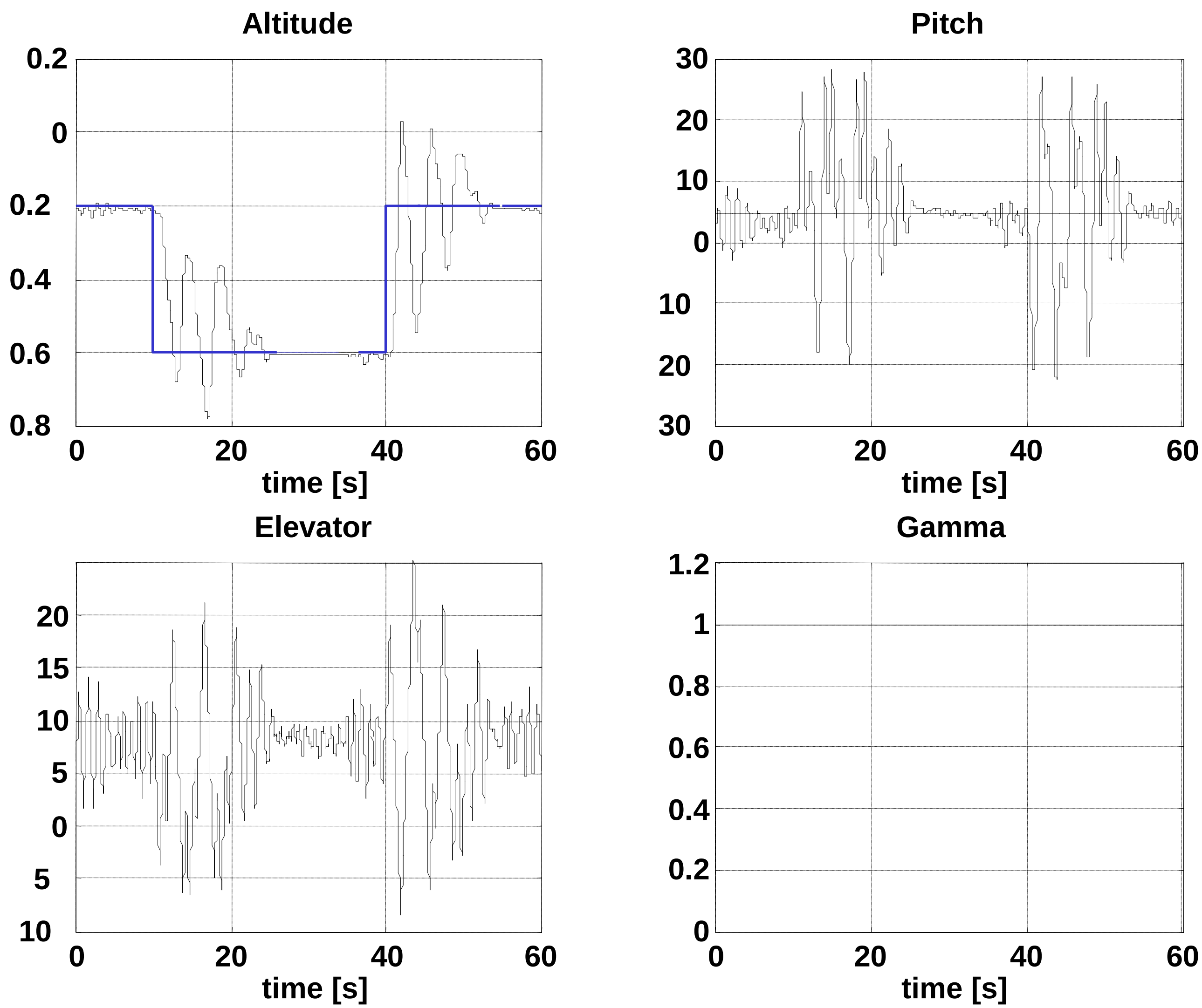
- Flatness implementation
- Magn/rate limit control

Dynamics

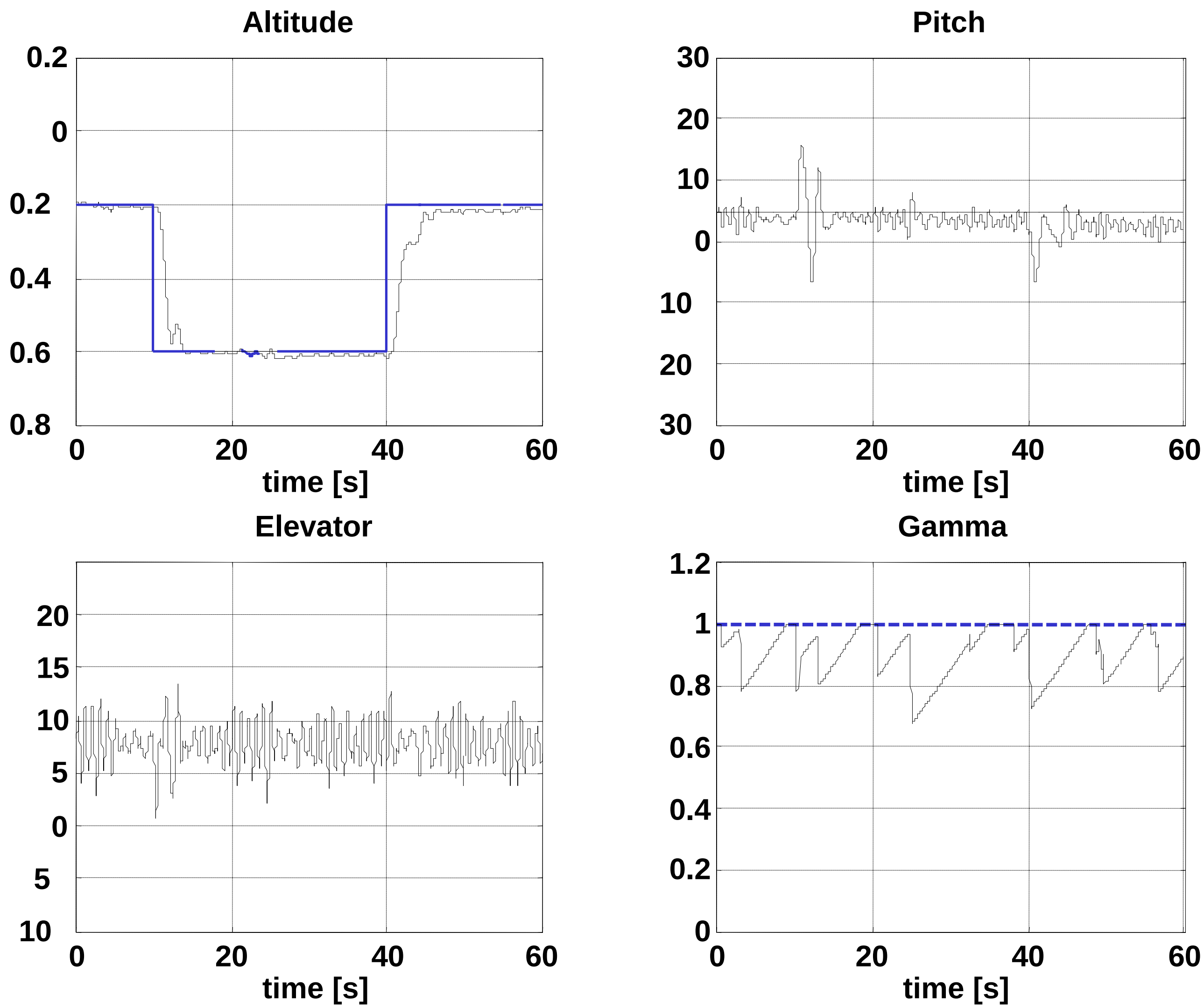
- Approx (2,4) integrator chains



Step altitude change: 20 deg/s rate limit, no rescaling

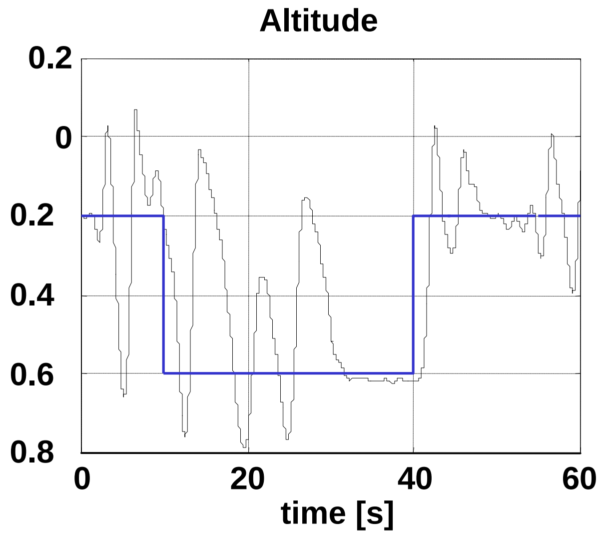


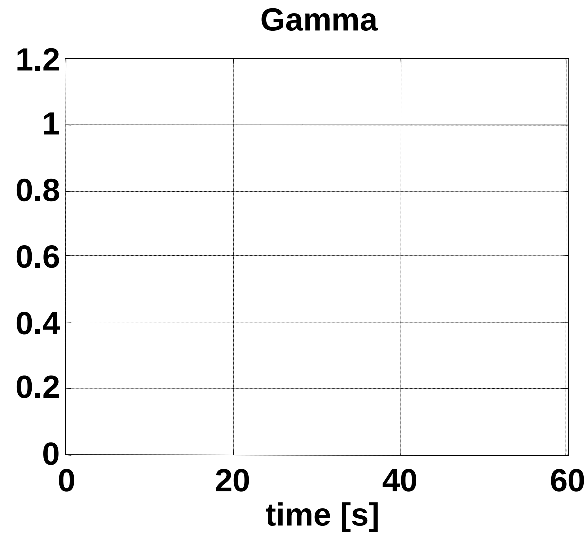
Step altitude change: 20 deg/s rate limit, dynamic rescaling



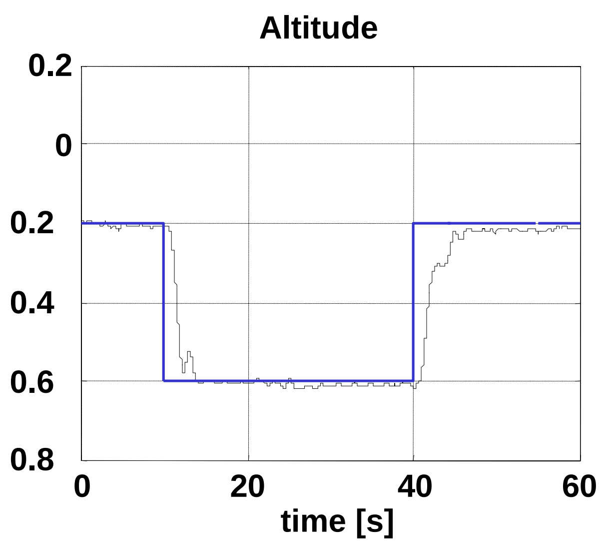
Comparison: 10 deg/s rate limit

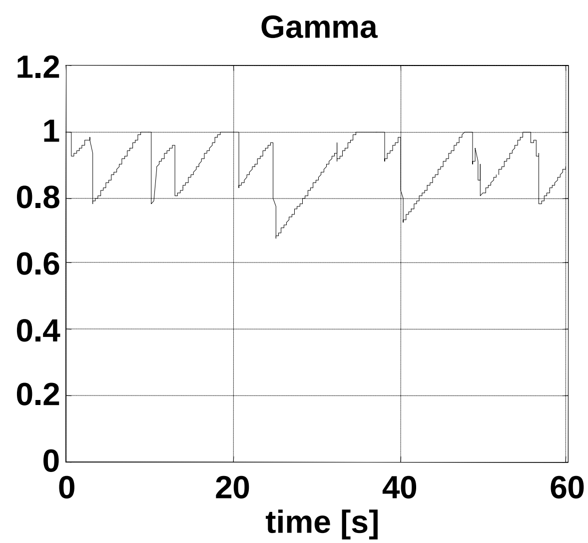
No rescaling





Dynamic rescaling





Richard Murray:

Need to add equation describing basis functions (alpha)

heory: Rescaling for Homogeneous Systems

P. Morin, L. Praly, R. Murray (NOLCOS 98)

$$(x) + g(x)u$$

Homogeneous order $\tau < 0$

Homogeneous order $\tau < 0$

With respect to dilation $r = (r_1, \ldots, r_n)$

Nominal control: $u = \alpha(x)$

Lyapunov function: $V(x) > 0, \dot{V} < 0$

Rescaled control:

$$v(x) = \underbrace{\lambda^{-\tau}(x)}_{\text{Rescale control}}\alpha\left(\underbrace{\delta_{\frac{1}{\lambda(x)}}(x)}_{\text{Scale state to } V=1}\right)$$

Features

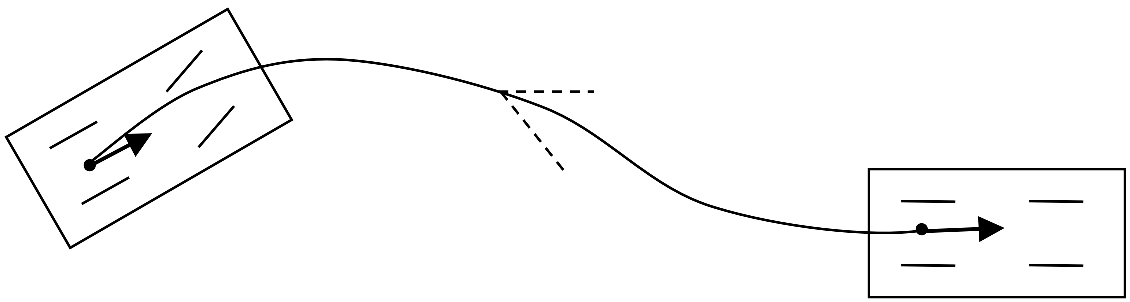
- Extends M'Closkey
- "Explains" Lauvdal
- Generalizes to non-homogeneous system (local approximation)

Thm GAS + $v(x)$ homogeneous order 0 ($\Rightarrow$ bounded)

Trajectory Tracking Using Differential Flatness

$$\begin{aligned} \dot{x} &= f(x, u) \\ z &= h(x, u, \dot{u}, \ldots, u^{(p)}) \\ |u| &< L \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} x &= x(z, \dot{z}, \ldots, z^{(q)}) \\ u &= u(z, \dot{z}, \ldots, z^{(q)}) \end{aligned}$$

Complicated constraints



Flat systems allow dynamic problems to be converted to algebra

- Solve spline problem with end constraints to generate feasible trajectory
- Input constraints become complicated, but *still* algebraic

Many interesting systems are flat

- All linear, feedback linearizable, chained, pure fbk form systems are flat
- Convential aircraft are *approximately* flat

Special Case: Linear Systems

Joint work with Sunil Agrawal (U. Del)

$$\dot{x} = Ax + Bu$$
$$z = Cx$$

Flat output

$$x_0 \overset{u_{[0,T]}}{\rightsquigarrow} x_f$$
$$|u| < L_1, \ |\dot{u}| < L_2$$

Linear everything $\Rightarrow$ solution is essentially a linear programming problem

$$x_0 \rightsquigarrow x_f \qquad \Rightarrow \qquad M\alpha = c$$
$$|u| < L_1, \ |\dot{u}| < L_2 \qquad \Rightarrow \qquad G(t)\alpha \leq d$$

Linear combinations of $\phi^{(j)}(t)$

Must hold for *every* time t

Q: Are there good (easy) ways to solve this problem in real-time

- Convert to discrete time? Basis functions?
- Keep in mind nonlinear case, if possible

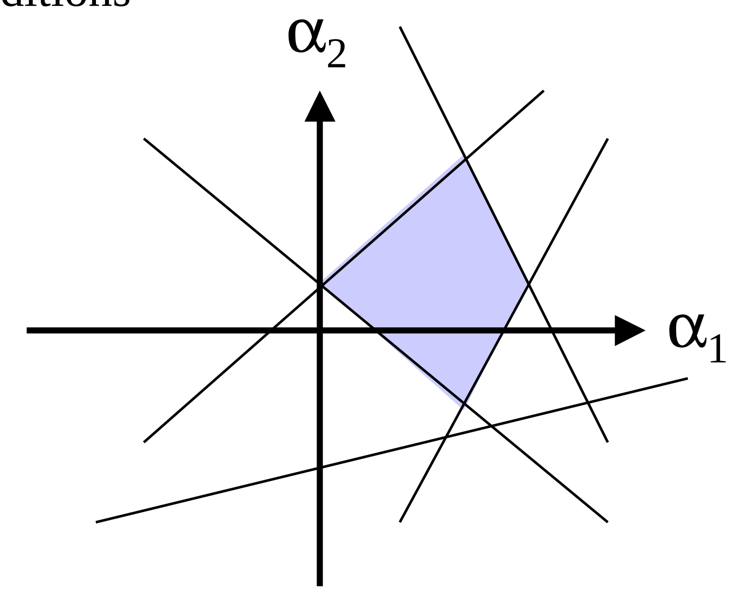
Trajectory Generation with Constraints

S. Agrawal, N. Faiz, R. Murray (IFAC 99)

$$z = \underbrace{\Phi_0(t)} + \underbrace{\sum \alpha_i \phi^i(t)} \qquad H_0 z + H_1 \dot{z} + \cdots + H_p z^{(p)} + d \le 0$$

Choose to satisfy constraints (null space)

Satisfies end conditions



Approach #1: Collocation (evaluate at $\{t_i\}$)

$$G(t_i)\alpha + d \le 0$$

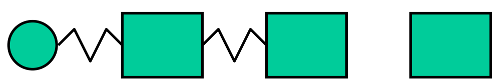
- Very similar to grasping problems
- Can track intersection points for speed

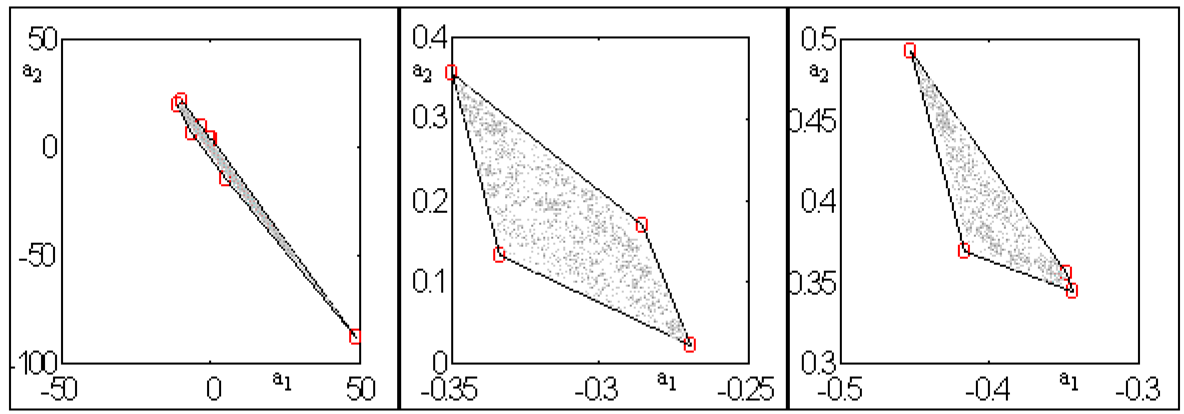
Approach #2: Use basis functions

- Gives guaranteed results, but can be conservative
- Choose basis functions based on dynamics (in progress)

Experimental Evaluation of Real-Time Algorithm

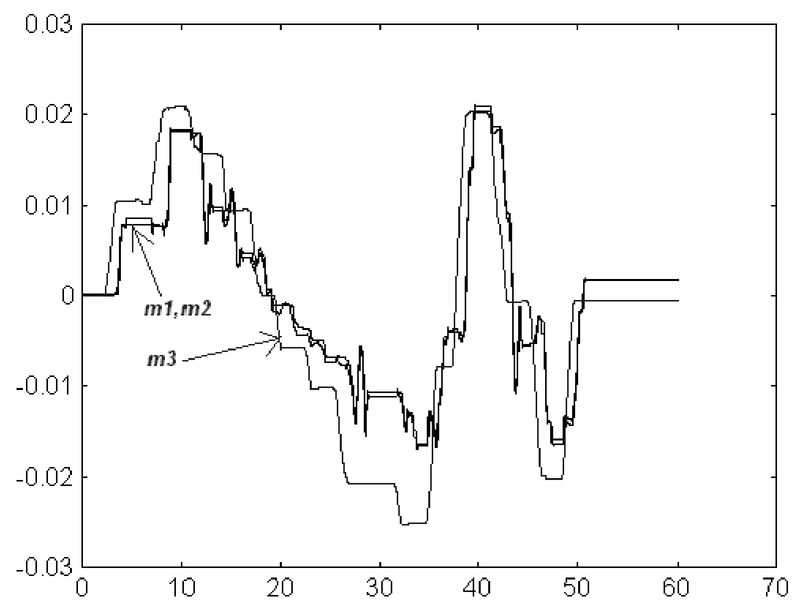
S. Agrawal, N. Faiz (U. Delaware)





Coupled spring-mass system

- Use third mass as control input; second mass tries to track first (pursuit problem)
- Constraint on mass positions and magnitude of input force
- Real-time trajectory generation, tracking and constraint update
  - 6 constraints, 3 collocation points, 2 modes $\Rightarrow$ 18 inequalities in 2 variables
  - 20 msec to compute trajectory



Summary and Future Work

Thrusts: stabilization and real-time traj gen in presence of magn/rate constraints

- Stabilization using dynamic rescaling
- Real-time trajectory generation for flat systems
- Motivated by increasing demand for aggressive motion control (uninhabited)

Current work (M. Milam, H. Streumper)

- Focused on real-time trajectory generation + surface allocation
- Comparison between real-time optimal & geometry-based approaches

Open issues

- Flatness based solutions are too restrictive; can we extend geometry to other classes
- Choice of basis functions to exploit dynamics/constraints, minimize computation
- State-based rescaling vs dynamic rescaling: need better comparison/understanding
- Missing *theory* for current industry allocation techniques (eg, RESTORE)