Goals:
- Introduce receding horizon control (RHC) for constrained systems
- Describe how to use "differential flatness" to implement RHC
- Give examples of implementation on the Caltech ducted fan, satellites, etc

Reading:
- Notes: "Online Control Customization via Optimization-Based Control"

Control Architecture: Two DOF Design

- Nonlinear design
  - global nonlinearities
  - input saturation
  - state space constraints

- Local design
  - \( P \)
  - \( X_d \)
  - \( U_d \)
  - \( \delta u \)

- Use nonlinear trajectory generation to construct (optimal) feasible trajectories
- Use local control to handle uncertainty and small scale (fast) disturbances
- Receding horizon control: iterate trajectory generation during operation

Murray, Hauser et al
SEC chapter (IEEE, 2002)
Receding Horizon Control

Solve finite time optimization over \( T \) seconds and implement first \( \Delta T \) seconds

\[
\begin{align*}
\min_{u(t)} & = \min_{u(t)} \int_{t}^{t+T} \left( L(x(\tau), u(\tau)) d\tau + V(x(t+\Delta T)) \right) \\
x_0 &= x(t) \quad x_f = x_f(t+T) \\
\dot{x} &= f(x, u) \quad g(x, u) \leq 0
\end{align*}
\]

Requires that computation time be small relative to time horizons
- Initial implementation in process control, where time scales are fairly slow
- Real-time trajectory generation enables implementation on faster systems

Stability of Receding Horizon Control

RHC can destabilize systems if not done properly
- For properly chosen cost functions, get stability with \( T \) sufficiently large
- For shorter horizons, counter examples show that stability is trickier

Thm (Jadbabaie & Hauser, 2002). Suppose that the terminal cost \( V(x) \) is a control Lyapunov function such that

\[
\min_u (\dot{V} + L)(x, u) < 0
\]

for each \( x \in \Omega = \{ x; \ V(x) < r^2 \} \), for some \( r > 0 \). Then, for every \( T > 0 \) and \( \Delta T \in (0; T) \), the resulting receding horizon trajectories go to zero exponentially fast.

Remarks
- Earlier approach used terminal trajectory constraints; hard to implement in real-time
- CLF terminal cost is difficult to find in general, but LQR-based solution at equilibrium point often works well - choose \( V = x^T P \) \( x \) where \( P = \) Riccati soln
RHC Design: Choice of Cost Function

**Q:** How do we choose RHC cost to get desired performance
- RHC deals w/ constraints, but shifts design freedom into choice of weights

**Thm (Kalman, 1964)** Given any state feedback law $u = -Kx$, there exists a cost function such that the optimal controller for that cost generates the given feedback law

- Theorem can be used to show that finite time horizon cost function also exists
- Basic idea: solve the algebraic Riccati equation for $P$, $Q$, $R$ given $K$

$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$

$$Q^{-1}B^T P = K.$$  

- Kalman showed you can always find positive definite solution to these eqns
- “Extension” to finite horizon problem: set $P_T = P$ and use

$$J = \int_0^T x^T Q x + u^T R u \, dt + x^T(T) P_T x(T)$$

---

RHC Design Philosophy

**Remarks**
- Can extend linear state space results to NL systems with CLF-based control
- General theory of dynamic compensators (eg, loopshaping) still open

- Challenge: must be able to generate (optimal) trajectories fast…
Optimal Control Using Differential Flatness

Can also solve constrained optimization problem via flatness

\[ \min J = \int_{t_0}^{T} L(x, u) \, dt + V(x(T), u(T)) \]

subject to

\[ \dot{x} = f(x, u) \quad g(x, u) \leq 0 \]

\[ \begin{aligned} x &= x(z, \dot{z}, \ldots, z^{(q)}) \\ u &= u(z, \dot{z}, \ldots, z^{(q)}) \\ z &= \sum \alpha_{\lambda} p^{i}(t) \end{aligned} \]

If system is flat, once again we get an algebraic problem:

\[ \begin{aligned} x &= x(z, \dot{z}, \ldots, z^{(q)}) \\ u &= u(z, \dot{z}, \ldots, z^{(q)}) \\ z &= \sum \alpha_{\lambda} p^{i}(t) \end{aligned} \]

\[ \min J = \int_{t_0}^{T} L(\alpha, t) \, dt + V(\alpha) \]

\[ g(\alpha, t) \leq 0 \]

Finite parameter optimization problem

- Constraints hold at all times ⇒ potentially over-constrained optimization
- Numerically solve by discretizing time (collocation)

Application: Caltech Ducted Fan

Flight Dynamics

\[ \begin{aligned} \dot{x} &= -D \cos \gamma - L \sin \gamma + F_{X_b} \cos \theta + F_{Z_b} \sin \theta \\ \dot{z} &= D \sin \gamma - L \cos \gamma - m_{\text{eff}} + F_{X_b} \sin \theta + F_{Z_b} \cos \theta \\ J \ddot{\theta} &= M_a - \frac{1}{r_s} I_p \Omega \dot{x} \cos \theta + M_T \\ \alpha &= \theta - \gamma, \quad \text{angle of attack} \\ \gamma &= \tan^{-1} \frac{-\dot{z}}{\dot{x}}, \quad \text{flight path angle} \end{aligned} \]

RHC Implementation

- System is approximately flat, even with aerodynamic forces
- More efficient to over-parameterize the outputs; use \( z = (x, y, \theta) \)
- Input constraints: max thrust, flap limits, flap rates
Example: Trajectory Generation for the Ducted Fan

Caltech Ducted Fan
- Ducted fan engine with vectored thrust
- Airfoil to provide lift in forward flight mode
- Design to emulate longitudinal flight dynamics
- Control via dSPace-based real-time controller

Trajectory Generation Task: point to point motion avoiding obstacles
- Use differential flatness to represent trajectories satisfying dynamics
- Use B-splines to parameterize trajectories
- Solve constrained optimization to avoid obstacles, satisfy thrust limits

From Real-Time Trajectory Generation to RHC

Three key elements for making RHC fast enough for motion control applications
- Fast computation to optimize over many variables quickly
- Differential flatness to minimize the number of dynamic constraints
- Optimized algorithms including B splines, colocation, and SQP solvers

Use of feedback allows substantial approximation
- OK to approximate computations since result will be recomputed using actual state
- NTG exploits this principle through the use of collocation

Can further optimize to take into account finite computation times

Tuning tricks
- Compute predicted state to account for computation times
- Optimize collocation times and optimization horizon
- Choose sufficiently smooth spline basis
Experiments: Caltech Ducted Fan

Real-Time RHC on Caltech Ducted Fan (Aug 01)
- NTG with quasi-flat outputs + Lyapunov CLF
- Average computation time of ~100 msec
- Inner (pitch) loop closed using local control law; RHC for position variables
- Inner/outer tradeoff: how much can be pushed into optimization

Highly Aggressive Constrained Turnaround
- Goal: -5 to 5 m/s. Final x position arbitrary, z within state constraint, Thrust vectoring within constraints
- Initial guess: Random
- Computation Time: 1.12 sec Sparc Ultra 10 83.3% CPU usage
- 6th order B-splines, seven intervals for each output, 30 equally spaced collocation points
- Full aerodynamic model
Example: Flight Control

dSPACE-based control system
- Two C30 DSPs + two 500 MHz DEC/Compaq/HP Alpha processors
- Effective servo rates of 20 Hz (guidance loop)

Franz, Milam et al
ACC 2002

Trajectory Generation for Non-Flat Systems

If system is not fully flat, can still apply NTG
\[
\begin{align*}
\dot{x} &= f(x,u) \\
z &= z(x,u,u,\ldots,u^{(q)}) \\
y &= h(x,u) \\
x &= x(z,\dot{z},\ldots,z^{(q)}) \\
u &= u(z,\dot{z},\ldots,z^{(q)}) \\
(x,u) &= \Gamma(y,\dot{y},\ldots,y^{(q)}) \\
0 &= \Phi(y,\dot{y},\ldots,y^{(p)})
\end{align*}
\]

When system is not flat, use quasi-collocation
- Choose output \(y=h(x,u)\) that can be used to compute the full state and input
- Remaining dynamics are treated as constraints for trajectory generation
- Example: chain of integrators
\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= u \\
y_1 &= x_1 \\
y_2 &= x_2 \\
x_1 &= y_1 \\
x_2 &= y_2 \\
u &= \dot{y}_2
\end{align*}
\]
\begin{equation}
\text{Solve using NTG with } lb=ub
\end{equation}

Can also do full collocation (treat all dynamics as constraints)
\[
\begin{align*}
(x,u) &= \sum \alpha \psi^i(t) \\
\dot{x}(t_i) &= f(x(t_i),u(t_i))
\end{align*}
\]
Each equation gives constraints at collocation points => highly constrained optimization
Effect of Defect on Computation Time

Defect as a measure of flatness
- Defect = number of remaining differential equations
- Defect 0 ⇒ differentially flat

Sample problem: 5 states, 1 input
- $x_1$ is possible flat output
- Can choose other outputs to get systems with nonzero defect
- 200 runs per case, with random initial guess

Computation time related to defect through power law
- SQP scales cubically ⇒ minimize the number of free variables

Dramatic speedup through reduction of differential constraints

Example 2: Satellite Formation Control

Goal: reconfigure cluster of satellites using minimum fuel

Dynamics given by Hill’s equations (fully actuated ⇒ flat)
Satellite Formation Results

Station-keeping optimization
- Maintain a given area between the satellites (for good imaging) while minimizing the amount of fuel
- Idea: exploit natural dynamics of orbital equations as much as possible
- Input constraints: $\Delta V < 20 \text{ m/s/year}$

Results
- Use NTG to optimize over 60 orbits (~3 days), then repeat
- Results: at 45° inclination, obtain
  \[ 10.4 \text{ m/s/year} \]

<table>
<thead>
<tr>
<th>$i$ (deg)</th>
<th>$S$ (m²)</th>
<th>$\Delta V$ (m/s/year)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d &lt; 300 \text{ m}$</td>
<td>$S = 100 \text{ m}^2$</td>
<td>$\Delta V = 25.6 \text{ m/s/year}$</td>
</tr>
<tr>
<td>$i = 45 \text{ deg}$</td>
<td>$S = 200 \text{ m}^2$</td>
<td>$\Delta V = 47.2 \text{ m/s/year}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$i$ (deg)</th>
<th>$S$ (m²)</th>
<th>$\Delta V$ (m/s/year)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d &lt; 300 \text{ m}$</td>
<td>$S = 100 \text{ m}^2$</td>
<td>$\Delta V = 10.4 \text{ m/s/year}$</td>
</tr>
<tr>
<td>$i = 90 \text{ deg}$</td>
<td>$S = 200 \text{ m}^2$</td>
<td>$\Delta V = 21.4 \text{ m/s/year}$</td>
</tr>
</tbody>
</table>

Example 3: MVWT Control Design

Control design technique
1. LQR design of state space controller $K$ around reference velocity
2. Choose $P$, $Q$, $R$ using Kalman’s formula
3. Implement as a receding horizon control with input and state space constraints
   - RHC controller respects state space constraint
Receding horizon control (RHC) for constrained systems
• Allows nonlinear dynamics + input and state constraints
• Need to be careful with terminal conditions to insure stability

Differential flatness is an enabler for practical implementation of RHC
• Allows fast computation of (optimal) trajectories
• NTG can be used to implement RHC; works for (slightly) non-flat systems

Caltech ducted fan implementation illustrates applicability of results
• Real-time control on representative flight control platform with no inner loop
• Extensions to multi-vehicle testbed are being implemented