Model predictive control (MPC)

- Optimal control problem
  Given a system with (stochastic) dynamics: \( x_{t+1} = f(x_t, u_t) \) Find the optimal policy \( \pi \) which minimizes the expected cost:
  \[ \mathbb{E} \left[ \sum_{t=0}^{\infty} g(x_t, u_t) \right] \]

- MPC:
  For \( t = 0, 1, 2, \ldots \)
  1. Solve
     \[ \min_{x_0, \ldots, x_H} \sum_{t=0}^{H} g(x_t, u_t) \]
     s.t. \( x_{t+1} = f(x_t, u_t) \forall t \in \{0, \ldots, H-1\} \)
  2. Execute \( u_t \) from the solution found in (1).

In practice, one often ends up having to solve:
\[ \min_{x_0, \ldots, x_H} \sum_{t=0}^{H} g(x_t, u_t) + \bar{g}(x_{t+1}, u_{t+1}) \]
\[ \text{s.t. } x_{t+1} = f(x_t, u_t) \forall t \in \{0, \ldots, H-1\} \]

Single shooting

- At core of MPC, need to quickly solve problems of the form:
  \[ \min_{x_0, x_1, \ldots, x_H, u_0, \ldots, u_H} \sum_{t=0}^{H} g(x_t, u_t) \]
  s.t. \( x_{t+1} = f(x_t, u_t), \forall t \in \{0, \ldots, H-1\} \)

- Single shooting methods directly solve for
  i.e., they solve:

  - Underneath, this typically boils down to berating:
    - For the current state sequence, simulate and find the state sequence
    - Take the 1st and 2nd derivatives w.r.t. \( x \)
  - Note: When taking derivatives, one ends up repeatedly applying the chain rule and the same Jacobians keep re-occurring
  - Beneficial to not waste time re-computing same Jacobians; pretty straightforward, various specifics with their own names (E.g., back-propagation.)

Single shooting drawback

- Numerical conditioning of the problem:
  - Influence on objective function of earlier actions vs. later actions
- What happens in case of a non-linear, unstable system?

Multiple shooting/Direct collocation

- Keep the state at each time in the optimization problem:
  \[ \min_{x_0, x_1, \ldots, x_H, u_0, \ldots, u_H} \sum_{t=0}^{H} g(x_t, u_t) \]
  s.t. \( x_{t+1} = f(x_t, u_t), \forall t \in \{0, \ldots, H-1\} \)
  \( h_k(x_t, u_t) \leq 0 \forall t \in \{0, \ldots, H-1\} \)

- Larger optimization problem, yet sparse structure.
- Special case: Linear MPC: \( f \) linear, \( h, g \) convex \( \rightarrow \) convex opt. problem, “easily” solved

Sequential Quadratic Programming (SQP)

- Goal: solve
  \[ \min_{x_0, x_1, \ldots, x_H, u_0, \ldots, u_H} \sum_{t=0}^{H} g(x_t, u_t) \]
  s.t. \( x_{t+1} = f(x_t, u_t), \forall t \in \{0, \ldots, H-1\} \)
  \( h_k(x_t, u_t) \leq 0 \forall t \in \{0, \ldots, H-1\} \)

- SQP: iterates over
  - Linearize \( f \) around current point \( (u, a) \), quadraticize \( g, h \) around current point
  - Solve the resulting Quadratic Programming problem to find the updated “current point” \( (u, a) \)

- Corresponds to:
  - Write out the first-order necessary conditions for optimality (the Karush-Kuhn-Tucker (KKT) conditions)
  - Apply Newton’s method to solve the (typically non-linear) KKT equations
Sequential Quadratic Programming (SQP)

- Not only method, but happens to be quite popular
- Packages available, such as SNOPT, SOCS.
- Many choices underneath:
  - Quasi-Newton methods
- Compared to single shooting:
  - Easier initialization (single shooting relies on control sequence)
  - Easy to incorporate constraints on state / controls
  - More variables, yet good algorithms leverage sparsity to offset this

Further readings

- Tedrake Chapter 9.
- Diehl, Ferreau and Haverbeke, 2008, Nonlinear MPC overview paper
- Francesco Borelli (M.E., UC Berkeley): taught course in Spring 2009 on (linear) MPC
- Packages:
  - SNOPT, ACADO, SOCS, ...
- We have ignored:
  - Continuous time aspects
  - Details of optimization methods underneath --- matters in practice b/c the faster the longer horizon
  - Theoretical guarantees

Non-minimum phase example

(Stoicne and Li, p. 195, Example II.2)

\[
\begin{align*}
3 x_1 + 4 y_2 &= 0 \\
\end{align*}
\]

The system is non-minimum phase, as it cannot be expressed as a single input-output relationship. Assume that initial steady state is \( x_1 = 0, x_2 = 0 \). The system is stable

\[
A = \begin{bmatrix} 3 & 0 \\ 0 & -4 \end{bmatrix}
\]

We can express a stable dynamics, \( \dot{y} = A y \). Note that the above equation has a zero which exactly cancels the unstable zero of the original system, i.e., permits exciting the nonminimum phase response in a natural way by utilizing constant inputs. By setting \( y = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \),

we see that this output is then subject to the natural dynamics for phase response.

Related intermezzo: Nonlinear control applied to kite-based power generation

Many companies pursuing this: Makani, KiteGen, SkySails, AmpyxPower, ...

Number from Diehl et al.: For a 500m\(^2\) kite and 10m/s wind speed (in sim) can produce an average power of more than 5MW

Technically interesting aspect in particular work of Diehl et al.: incorporate open-loop stability into the optimization problem.

- Only possible for non-linear systems
- The criterion quantifies how much deviation from the nominal trajectory would amplify/decrease in one cycle

Feedback linearization

Feedback linearization
Feedback linearization

Further readings:
- Slotine and Li, Chapter 6

Reminder: No office hours today.
[Feel free to schedule over email instead]

A system $x_{t+1} = f(x_t, u_t)$ if for all $x_0$ and all $x$, there exists a time $k$ and a control sequence $u_0, \ldots, u_{k-1}$ such that $x_t = x$.

Controllability [defn., linear systems]
- A system $x_{t+1} = f(x_t, u_t)$ is controllable if for all $x_0$ and all $x$, there exists a time $k$ and a control sequence $u_0, \ldots, u_{k-1}$ such that $x_t = x$.

Fact. The linear system $x_{t+1} = Ax_t + Bu_t$ with $x_t \in \mathbb{R}^n$ is controllable iff 
$[B \ AB \ A^2B \ \ldots \ A^{n-1}B]$ is full rank.

Lagrangian dynamics

[From: Tedrake Appendix A]
Lagrangian dynamics: example