ESC794: Special Topics: Model Predictive Control

Introduction

Hanz Richter, Professor
Mechanical Engineering Department
Cleveland State University
Recurring themes in control engineering practice

Outside textbooks and the comfortable realm of simulations, experience shows that:

■ Every system is to some extent nonlinear. Some nonlinearities like friction are difficult to model and interfere with desired performance.

■ Constraints on the input and/or states must be respected. Unplanned control saturation may lead to poor performance or oscillation.

■ Systems may have several control inputs, and it may be difficult to optimize the controller or its gains to match desired performance or even maintain stability.

■ Systems may change over time and may be affected by disturbances.
A difficult control problem: jet engines

All of the above challenges are found in many real systems. Aeronautical gas turbines are a notable example. The objective is to regulate thrust by using fuel flow and secondary actuators.

1. The model is highly nonlinear and only accessible to the designer as a numerical simulation (i.e., NASA’s T-MATS or C-MAPSS). There are no direct system properties to be inferred, except through local linearizations.
A difficult control problem: jet engines

1. Control inputs (fuel flow rates, valve openings) are inherently constrained. Certain states (rotor speeds) and outputs (turbine blade temperature) are subject to critical constraints related to safety and lifespan.

2. How to design a feedback controller that takes advantage of available actuators to maximize performance?

3. Engine dynamics are not the same at different Mach numbers, altitudes and outside air temperatures. Aging and deterioration appear as slowly-varying disturbances.
Robust/adaptive control of multivariable nonlinear systems

Researchers have proposed many major paradigms and specialized techniques to address (partially) the above four challenges. Among the most mainstream approaches, with verified applications:

1. Nonlinear control techniques for multivariable systems: sliding mode control, backstepping and other Lyapunov approaches.

2. Linearization-based robust linear control: LPV and gain scheduling, with $\mathcal{H}_\infty$ and $\mu$ synthesis.

3. Adaptive control, with fast adaptation and robust variants.

Although many potential approaches have been proposed, addressing optimality and constraints for nonlinear multivariable systems remains largely unfeasible with the above approaches.

For jet engines, the state-of-the-art is a gain-scheduled controller based on multiple local models (linearizations). Constraints are handled with heuristic approaches that turn controllers on or off based on proximity to constraints.

However, each of the above methods, plus MPC have been attempted in research contexts.
Ad-hoc constraint handling: min-max controller switching

This idea seems to be still the standard in commercial aeronautical engines (https://technology.nasa.gov/patent/LEW-TOPS-56).
Optimal Control

Model Predictive Control arises from optimal control, a method which can consider constraints and seeks performance maximization.

Given a dynamic system to be controlled

$$\dot{x} = f(t, x, u)$$

where the state and control must be kept in constraint sets $\mathbb{X}$ and $\mathbb{U}$, we can measure the performance obtained by applying a candidate control trajectory $u \in \mathbb{U}$ with a cost function:

$$J = \int_{t_0}^{t_f} l(x_u(t), u(t))dt$$

where $x_u(t)$ is the state trajectory (solution) resulting from applying $u$ between $t_0$ and $t_f$.

A basic optimal control problem is to calculate the control sequence $u^* \in \mathbb{U}$ that minimizes $J$ and satisfies $x_u(t) \in \mathbb{X}$ for $t \in [t_0, t_f]$. 
Symbols used in MPC tend to be complicated and vary with author. We will (mostly) follow Grüne and Pannek’s (G&P) notation.

- **\( U \)**: control value space (for example a flow control system with 2 valves may have \( U = \mathbb{R}^2 \))
- **\( X \)**: state value space (for example a triple integrator system has \( X = \mathbb{R}^3 \))
- **\( U^N \)**: set of finite control sequences with elements in \( U \), that is \( \{u(0), u(1) u(2), \ldots u(N − 1)\}, N \in \mathbb{N}_0 \)
- **\( X \subseteq X \)**: state constraint set. We want \( x(t) \in X \) at all times.
- **\( U(x) \subseteq U \)**: set of admissible control values at \( x \in X \). For example, the 2 valves may have \( U(x) = U = [0 1] \times [0 1] \)
- **\( U^N(x) \subseteq U^N \)**: set of admissible \( N \)-long control sequences at \( x \in X \).

**NOTE:** Admissibility includes more requirements for a sequence than just belonging to \( U^N \). We discuss it now.

The key is to understand the difference between control values and control sequences reflected in the notation. Think of chain links vs. a whole chain.
Consider the first-order nonlinear system

\[ \dot{x} = -x + xu \]

and a cost function with \( l(x, u) = x^2 \), evaluated between \( t_0 = 0 \) and \( t_f = 1 \). Consider \( X = \mathbb{R} \) and require that \( |u(t)| \leq 1 \) for all \( t \in [0, 1] \).

1. Identify \( U \) and \( \mathbb{U} \). We are open to the possibility of \( u \) being discontinuous.
2. Solve the problem by solving the differential equation for all \( x(t_0) = x_0 \). Find the optimal cost \( J(x_0) \).
3. Sketch the optimal control \( u^* \) and optimal trajectories \( x^* \).
Example...

The previous example included optimality, input constraints and the system was nonlinear. However:

- The optimal control solution is a time trajectory $u^*(t)$ to be applied as calculated, without the possibility of adjustment due to unforeseen disturbances or model errors. It is an open-loop control strategy.

- The optimal control is defined only up to $t_f$. It does not address the need to operate the system for an indefinite period of time, like a thermostat or an automotive cruise control system do. It is a finite-horizon control input.

- Because of the above, there is no way to discuss system stability, which is an asymptotic (long-term) property.

- We were able to solve for $u^*(t)$ analytically because the simplicity of the system, the cost function and the constraints.
Model Predictive Control

Many variants and sub-classes of MPC exist, but the essential feature is the repeated solution of a finite-horizon optimal control problem. The infinite time horizon \([t_0, \infty)\) is divided in intervals of length \(\delta\), with instants \(t_k = t_0 + k\delta\), for \(k = 0, 1, \ldots, \infty\). The general MPC template is:

1. At \(t = t_k\) measure the state of the system, \(x(t_k)\). Solve a finite-horizon optimal control problem to determine the optimal control \(u^*\) and the optimal state trajectories \(x^*\) over \([t_k, t_k + T]\), where \(T\) is the horizon.

2. Apply only the first portion of \(u^*\) to the plant. That is, \(u^*(t)\) for \([t_k, t_k + \delta]\) is applied.

3. The system responds to the input and the process is repeated starting with \(t = t_k + \delta\). The optimal control and state from the previous MPC iteration are normally used as guesses in numerical solution algorithms.

An important observation is that although each optimal control iteration gives an open-loop trajectory, each \(u^*(t)\) determined at time \(t_k\) is implicitly dependent on \(x(t_k)\). Since this actual plant state is measured and used to find \(u^*(t)\) at each \(t_k\), a feedback action is introduced. In other words, the applied control defines an MPC feedback law \(\mu(x(t_k))\).
MPC and Strategic Games

MPC is much like chess:

1. At each turn, a player examines (predicts) several move scenarios that comply with the constraints (rules of the game).
2. Using move scenarios, experience, and value judgements, a player determines an optimal sequence of events, but can only make one move.
3. The system responds (the other player) to the move and the cycle is repeated.

MPC uses model-based *prediction* to solve the optimal control problem. A feedforward, or look-ahead element is present in addition to plant state feedback. Put this in contrast with PI control, which is error based: control actions depend only on the present and past plant outputs, not on predicted outputs (an error needs to occur for it to be corrected).
MPC and Driving

This comparison is popular: consider the problem of night driving on a road while staying on the correct lane.

- PI control is like doing this by looking only at the rearview mirror. We will correct for deviations once they happen.

- MPC is what we actually do. We see the trends in the painted lines and anticipate turns with lead action.

- What does the “D” in PID control do? Is it anticipatory?
Major MPC shortcomings

MPC is just another tool in the repertoire of control paradigms. It has the following limitations, which continue to motivate much research:

- **Computational burden:** An optimal control problem must be solved at each instant $t_k$. The spacing $\delta$ must be chosen small enough to keep up with the dynamics of the system being controlled. With increasing system and constraint complexity, there may not be enough time to complete the solution before $\delta$ has elapsed.

- **Feasibility and existence of a solution:** Regardless of how fast it could be solved, the optimal control needs to be feasible (some control input has to exist that complies with the constraints). Also, even if feasible, an optimal control may not exist.

- **Robustness:** MPC is based on a model, but models are imperfect, leading to imperfect predictions and at best, sub-optimality and lack of constraint satisfaction. At worst, instability and destruction can occur. The topic continues to be researched, with some promising approaches.

Still, MPC has been widely applied, patented and commercialized. In this course, we will build and demonstrate a real-time implementation of MPC.
Brief Historical Timeline of MPC


- Like many other control approaches, MPC-like strategies were applied before they were analyzed and understood. Model Predictive Heuristic Control was proposed by Richalet, and Dynamic Matrix Control was used by Shell Oil in the 70’s.

- The chemical and process industries often involve control problems for plants with very slow dynamics. The time constant (residence time) is in the order of the tens of minutes.

- This explains how MPC was possible back then, with *those* computers.

- Early theoretical investigations: Kleinmann, Chen (70s-80s).

- A formal control-theoretical approach did not start until 1990, with landmark papers by Mayne and Michalska.
Clear-cut feasibility and stability analysis is now available for linear and nonlinear systems (Chen and Allgöwer, Bemporad ’94, Mayne ’00, among others).

As people sought to apply MPC to faster plants (mechanical systems), computational and other implementation issues came into focus.

Explicit, pre-computed or offline MPC approaches were developed in the early 2000s (Bemporad & Morari). The idea is to solve for $\mu(x_0)$ offline for all $x_0$ in a region of interest. The region is divided into sub-regions where $\mu(x_0)$ has constant gains, as approximation. The online implementation is then reduced to look-up tables.

Robust MPC approaches were proposed in the mid 2000s, and the topic continues to be active (Mayne, Allgöwer, Rakovic).

Distributed MPC was motivated by computational advantages and also to handle large scale interconnected systems. Several strategies and algorithms are available (Richards & How, Müller, Allgöwer). The topic is finding renewed interest in the context of the Internet of Things. Autonomous vehicles, land, air and sea could be controlled with these approaches.
Brief Historical Timeline of MPC...

- *Economic* MPC refers to optimization for objectives other than setpoint regulation or trajectory tracking, like energy consumption minimization or product maximization. It is one of the most recent and promising research areas with immediate applications to smart grids. Main proposers are Angeli, Müller, Allgöwer and Grüne, a very active research area.

- *Output* MPC focuses on the use of state estimators in combination with MPC, and *moving horizon estimation* involves a counterpart of MPC aimed at estimation instead of control, with the same receding constrained optimization spirit.

- A large volume of research output has focused on efficient numerical solution algorithms oriented at fast real-time implementations. Today, researchers report MPC iteration times in the microsecond range, by using a combination of efficient algorithms and powerful hardware like Field-Programmable Gate Arrays (FPGA).

- The class will read the 2014 survey paper by David Mayne progressively during the semester, and portions of the paper will be used in some assignments and evaluations.
Course Organization

Roughly, the course will be divided into 7 parts:

1. Introduction. Linear and nonlinear discrete-time systems, stability, optimal control and dynamic programming.
2. Constrained linear MPC and quadratic programming solution.
4. Some numerical methods for MPC. Implementation focus.
5. Overview of Distributed MPC
6. Overview of Economic MPC
7. Guided real-time MPC implementation on lab hardware (project).

Students are expected to be familiar with: (at a minimum)

- Linear control theory with state-space methods
- Matlab programming and Simulink
- Linear algebra

Courses in nonlinear systems and optimal control are desirable but not required.