Model Predictive Control

1. Basics

Compact set: Subset of Euclidean space being closed (*i.e., containing all its limit points*) and bounded (*i.e., having all its points lie within some fixed distance of each other*).
Matrix Inversion: $A^{-1} = \frac{1}{\det A} \cdot \operatorname{adj}(A)$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Eigenvalues: $\det(A - \lambda I) = 0$
Eigenvectors: $(A - \lambda I) \cdot v = 0$
Rule of Sarrus: $A_{3\times 3} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$

$$\det(A_{3\times 3}) = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} \\ - a_{31}a_{22}a_{13} - a_{32}a_{23}a_{11} - a_{33}a_{21}a_{12}$$

For system equations $y_i = f_i(x)$, you can find x with **Least Squares**:

$$x = \arg\min_x \left[\sum_{i=1}^{n} (y_i - f_i(x))^2\right]$$

($\rightarrow$ *Calculate derivative, set to zero and find corresponding value for* x)

2. Introduction

2.1. System Class

Time-invariant discrete-time dynamical control system
$x(k+1) = f(x(k), u(k)); \quad k = 0, 1, 2, \ldots \quad x(0) = x_0$
with state $x(k) \in \mathbb{R}^n$ and control input $u(k) \in \mathbb{R}^m$.

Notation:
$\underline{u}^N := \{u(0), u(1), \ldots u(N-1)\}$
$\underline{x}_{\underline{u}}^N(x_0) := \{x_0, x_{\underline{u}}(1, x_0), \ldots x_{\underline{u}}(N, x_0)\}$
or if context is clear for brevity:
$\underline{x}_{\underline{u}}^N(x_0) := \{x_0, x(1), \ldots x(N)\}$

2.2. Cost Function

Infinite Horizon:

$$J_\infty(x_0, \underline{u}^\infty) = \sum_{k=0}^{\infty} l(x_{\underline{u}}(k, x_0), u(k))$$

Finite Horizon:

$$J_N\left(x_0, \underline{u}^N\right) = \sum_{k=0}^{N-1} l(x_{\underline{u}}(k, x_0), u(k)) + J_f(x_{\underline{u}}(N, x_0))$$

with stage cost $l(x,u)$ and target cost $J_f(x)$.

2.3. Constraints

Input constraints: $u(k) \in \mathbb{U}$
State constraints: $x(k) \in \mathbb{X}, k = 1, 2, 3, \ldots; \ x(N) \in \mathbb{X}_f$

Admissible Controls: $\mathcal{U}_N(x_0) := \{\underline{u} | (x_0, \underline{u}) \in \mathbb{Z}\}$
Feasible Initial Values: $\mathcal{X}_N := \{x_0 \in \mathbb{X} | \mathcal{U}_N(x_0) \neq \emptyset\}$
with

$$\mathbb{Z}_N := \Big\{ (x_0, \underline{u}) \, | u(k) \in \mathbb{U}, x_{\underline{u}}(k, x_0) \in \mathbb{X}, \\ k = 0, 1, \ldots, N-1; x_{\underline{u}}(N, x_0) \in \mathbb{X}_f \Big\}$$

2.4. Optimization Problem

$$\mathbb{P}_N(x_0) : J_N^*(x_0) = \min_{\underline{u}} \{J_N(x_0, \underline{u}) \, | \underline{u} \in \mathcal{U}_N(x_0)\}$$

Assumption 1:
$f, \ l, \ J_f$ are continuous with $f(0,0) = 0, \ l(0,0) = 0, \ J_f(0) = 0$.

Assumption 2:
$\mathbb{X}$ is closed, $\mathbb{X}_f$ and $\mathbb{U}$ are compact and all sets contain the origin.

Under *Assumption 1* and *Assumption 2*, the optimization problem $\mathbb{P}_N(x_0)$ has a solution for all $x_0 \in \mathcal{X}_N$.
($\rightarrow$ Theorem of Weierstrass: Sets are sequentially compact *and* every bounded sequence of complex numbers contains at least one convergent subsequence.)

2.5. Controller

$\kappa_N(x_0) = u^*(0, x_0)$
with optimal control input $u^*(0, x_0)$ from solution of $\mathbb{P}_N(x_0)$,
$\underline{u}^* = \{u^*(0, x_0), u^*(1, x_0), \ldots u^*(N-1, x_0)\}$.

2.6. Basic time-invariant MPC algorithm

System: $x^+ = f(x, u)$
Cost: $J(x, \underline{u}) = \sum_{k=0}^{N-1} l(x(k), u(k)) + J_f(x(N))$
Constraints: $x(k) \in \mathbb{X}, u(k) \in \mathbb{U}$ for all $k \in \mathbb{N}_0$ and $x(N) \in \mathbb{X}_f$
where N is the prediction horizon.

- Measure x, determine $\mathcal{U}_N(x)$
- Solve $\mathbb{P}_N(x)$ and obtain $\underline{u}^*(x)$
- Control with $\kappa_N(x)$ such that $x^+ = f(x, \kappa_N(x))$
- Repeat for $x := x^+$

2.7. Constrained Optimization in a nutshell

Cost function: $\min F(z)$
Equality constraints: $g(z) = 0$
Inequality constraints: $h(z) \leq 0$

a) unconstrained
If z^* is minimum, then $\nabla F(z^*) = 0$.
b) equality constrained
Lagrange function $L = F(z) + \lambda^\top g(z)$ with Lagrange multiplier λ.
If z^* is minimum, then $\nabla_z L(z^*, \lambda^*) = 0$ and $\nabla_\lambda L(z^*, \lambda^*) = 0$.
c) inequality constrained
Lagrange function $L = F(z) + \mu^\top h(z)$ with KKT multiplier μ.
If z^* is minimum then $\nabla_z L(z^*, \mu^*) = 0$ and $\mu^* \geq 0, h(z^*) \leq 0$ and $\mu_i^* h_i(z^*) = 0$ for all i.

Optimum for cost function at horizon N for $J_N = 0$. If only one constraint is active, use **b)** to look at constrained edge of control input and derive the remaining optimal control inputs.

3. Dynamic Programming

3.1. Problem Statement

Time-invariant discrete-time dynamical control system *(can be nonlinear)*:
$x(k+1) = f(x(k), u(k)); \quad k = 0, 1, 2, \ldots \quad x(0) = x_0$
Cost: $V(x_0, \underline{u}) = \sum_{k=0}^{N-1} l(x(k), u(k)) + V_f(x(N))$
State constraints: $x(k) \in \mathbb{X}, \ k = 0, 1, \ldots, N-1, \ x(N) \in \mathbb{X}_f$
Input constraints: $u(k) \in \mathbb{U}, k = 1, 2, \ldots, N-1$

Find $\underline{u} = \{u(0), u(1), \ldots, u(N-1)\}$
where $u(k) = \mu_k(x(k))$ are control laws.

Advantages and Disadvantages:
$+$ We get control laws $\quad +$ Costly optimization is done offline
$-$ Curse of dimensionality

Comparisson to MPC:

- Dynamic Programming (DP) has one large horizon.
- MPC delivers control action for only one specific realisation.
- DP is optimal for full problem and target constraints are satisfied. MPC target constraints are reached asymptotically
  $\rightarrow$ *might have stability problem*

3.2. Notation

Cost-to-go: $V_i\left(x, \underline{u}^i\right) = \sum_{k=i}^{N-1} l(x(k), u(k)) + V_f(x(N))$
with $\underline{u}^i = \{u(i), u(i+1), \ldots, u(N-1)\}$
Optimal cost-to-go: $V_i^*(x) = \min_{\underline{u}^i \in \Upsilon_i(x)} V_i\left(x, \underline{u}^i\right)$ with

$$\Upsilon_i(x) := \\ \Big\{\underline{u}^i | \text{ for initial state } x(i) = x : u(k) \in \mathbb{U}, k = i, \ldots, N-1; \\ x(k) \in \mathbb{X}, k = i+1, \ldots, N-1; x(N) \in \mathbb{X}_f\Big\}$$

and $\Xi_i := \{x \in \mathbb{X} | \Upsilon_i(x) \neq \emptyset\}$

Recursive construction of feasible sets Ξ_i from behind:
$\Xi_N = \mathbb{X}_f$
$\Xi_i = \{x(i) \in \mathbb{X} | x(i+1) \in \Xi_{i+1} \text{ with } u(i) \in \mathbb{U}\}$,
$i = N-1, N-2, \ldots 0$

3.3. Bellman Recursion

Recursive calculation of optimal-cost V_i^*, starting with the terminal cost:
$V_N^*(x(N)) = V_f(x(N)) \ \rightarrow \ V_{N-1}^* = \min\{V_N^* + \ldots\} \ \rightarrow \ \ldots$
In general, with $l(x(i), u(i))$ as cost of only the current i:

$$V_i^*(x(i)) = \min_{u(i)} \Big\{ l(x(i), u(i)) + V_{i+1}^*(f(x(i), u(i))) \ | \\ u(i) \in \mathbb{U}, x(i) \in \mathbb{X}, f(x(i), u(i)) \in \Xi_{i+1} \Big\}, \\ i = N-1, N-2, \ldots, 0$$

Use system $f(x(i), u(i))$ to substitute $x(i+1)$ in $V_{i+1}^*(x(i+1))$
and solve $\frac{\partial V_i}{\partial u(i)} = 0$ to get optimal control input $u^*(i) = ax(i)$.
Insert $u^*(i)$ into $V_i^*(x(i))$ to get optimal cost-to-go.

4. Stability

4.1. Stability Concepts

System class: $x^+ = \tilde{f}(x) \quad \rightarrow$ Equilibrium point: $x_{eq} = \tilde{f}(x_{eq})$
Open loop: $x^+ = f(x) \quad \longleftrightarrow \quad$ Closed loop: $x^+ = f(x, u)$

Assumption:
If $\tilde{f}$ is not continuous, it is at least locally bounded.

Definition of stability in the sense of Lyapunov:
Equilibrium point $x_{\text{eq}} = 0$ of $x^+ = \tilde{f}(x)$ is locally stable, if $\forall \varepsilon > 0$ there exists a $\delta > 0$ s.t. for all $\|x(0)\| \leq \delta(\varepsilon)$, it holds that $\|x(k)\| \leq \varepsilon$ for all $k > 0$.

- Locally asymptotically stable, if in addition $\lim_{k \to \infty} \|x(k)\| = 0$ for $x(0)$ close to the origin.
- Globally asymptotically stable, if in addition $\lim_{k \to \infty} \|x(k)\| = 0$ for all $x(0) \in \mathbb{R}^n$.
- Asymptotically stable in $\mathcal{X}$, if in addition $\lim_{k \to \infty} \|x(k)\| = 0$ for all $x(0) \in \mathcal{X}$, where $\mathcal{X}$ is positive.

Definition of **positive invariance**:
$\mathcal{X}$ is positive invariant for $x^+ = \tilde{f}(x)$, if $\tilde{f}(x) \in \mathcal{X}$ for all $x \in \mathcal{X}$.
(*Once a system trajectory enters set* $\mathcal{X}$, *it will never leave it again.*)

Comparison Functions:
Necessary as candidate Lyapunov function is discontinuous.
$\rightarrow$ Optimal control $\underline{u}^*$ is solution of a optimum, convergence to different local minima possible.
$\rightarrow$ Special case: unconstrained MPC is continuous

Definition:

- A function α is a class $\mathcal{K}$ function, if it is continuous and strictly increasing with $\alpha(0) = 0$.
- A function α is a class $\mathcal{K}_\infty$ function, if it a class $\mathcal{K}$ function and in addition unbounded.
- A function α is a class $\mathcal{PD}$ (*positive definite*) function, if it is continuous with $\alpha(0) = 0$ and $\alpha(x) > 0$ for all $x \neq 0$.

Lyapunov's direct method:
A function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ is a global Lyapunov function for the equilibrium of $x^+ = \tilde{f}(x)$, if $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$, $\alpha_3 \in \mathcal{PD}$ exist, such that for all $x \in \mathbb{R}^n$ condition **(1)** and **(2)** holds.
If V is a global Lyapunov function for the equilibrium of $x^+ = \tilde{f}(x)$, then the equilibrium is globally asymptotically stable.

Lyapunov's direct method (**constrained**):
A function $V : \mathcal{X} \rightarrow \mathbb{R}$ with $\mathcal{X}$ invariant is a Lyapunov function for the equilibrium of $x^+ = \tilde{f}(x)$ on $\mathcal{X}$, if $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$, $\alpha_3 \in \mathcal{PD}$ exist, such that for all $x \in \mathcal{X}$ condition **(1)** and **(2)** holds.
If V is a Lyapunov function for the equilibrium of $x^+ = \tilde{f}(x)$ on $\mathcal{X}$, then the equilibrium is asymptotically stable on $\mathcal{X}$.

(1) Bounded: $\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|)$
(2) Descent: $V(f(x)) - V(x) \leq -\alpha_3(\|x\|)$

4.2. Assumptions for Stability

Assumption 3:
$l(x, u) > \alpha_l(\|x\|) \quad \forall x \in \mathcal{X}_N, \forall u \in \mathbb{U}$
$J_f(x) \leq \alpha_f(\|x\|) \quad \forall x \in \mathbb{X}_f$ where α_l, α_f are class $\mathcal{K}_\infty$ functions.

Assumption 4:
J_f is a Control Lyapunov Function (CLF), that means $J_f(0) = 0$, $J_f > 0$ and there exists $u \in \mathbb{U}$ such that $J_f(f(x,u)) - J_f(x) \leq -l(x, u) \ \forall x \in \mathbb{X}_f$.
If system is CLF, it is feedback stabilizable.

Assumption 5:
$\mathbb{X}_f$ is control invariant, that means, if $x \in \mathbb{X}_f$ then there exists $u \in \mathbb{U}$ such that $f(x, u) \in \mathbb{X}_f$.

4.3. Stability of MPC

Under Assumptions 1 to 5, the equilibrium x_{eq} is asymptotically stable in $\mathcal{X}_N$ for $x^+ = f(x, \kappa_N(x))$.

Proof: Choose Lyapunov function $V_N(x) = J_N(x, \underline{u}^*)$ and show property (1) and (2) of Lyapunov direct method.

4.4. Recursive Feasibility

Definition: MPC is said to be recursively feasible, if one can assure that there is a solution to $\mathbb{P}_N(x^+)$ having a solution of $\mathbb{P}_N(x)$.

If $\mathbb{X}_f$ is control invariant, then

- $\mathcal{X}_{j-1} \subseteq \mathcal{X}_j, \ j = 1, \ldots, N$
- $\mathcal{X}_{j-1}$ is control invariant, $j = 1, \ldots, N$
- MPC is recursively feasible

To deal with infeasibility, soften hard constraints.

To find feasible initial values $\mathcal{X}_1$ for an additional target constraint $\mathcal{X}_0 = \mathbb{X}_f = \{0\}$, solve $x^+ = f(x, u) = 0$ and check which states satisfy target constraint.

To determine $\mathcal{X}_1$ given an control invariant target set $\mathcal{X}_0 = \mathbb{X}_f$, put in system dynamics x^+ for x in $\mathbb{X}_f$ and check if state constraints are satisfied. $\mathcal{X}_1$ is then the set with adjusted system matrix and limits.

$$\mathcal{X}_1 = \left\{ \mathbf{A} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} a \\ b \end{bmatrix} \right\}$$

With control constraints $a_u \leq u \leq b_u$, give feasible values subject to the constraints. E.g. for a state $x = [x_1 \ x_2 \ x_3]^\top$
$\mathcal{X}_1 = \{x | a_u \leq x_1 \leq b_u, \ x_1 = x_3 = -x_2\}$

5. MPC for Linear Systems

5.1. System Class for Linear MPC

System class: $x^+ = Ax + Bu$

Cost: $J_N(x, \underline{u}) = \frac{1}{2}\sum_{k=0}^{N-1} \|x(k)\|_Q^2 + \|u(k)\|_R^2 + \frac{1}{2}\|x(N)\|_{P_f}^2$

Constraints: $x(k) \in \mathbb{X}, u(k) \in \mathbb{U}, x(N) \in \mathbb{X}_f$, where $\mathbb{X}$, $\mathbb{U}$ and $\mathbb{X}_f$ are convex polytopes

5.2. LQ Control (non-receding finite horizon)

For linear systems with quadratic cost and no constraints.

System Class: $x^+ = Ax + Bu$

Cost:

$$V_0(x_0, u) = \frac{1}{2}\sum_{k=1}^{N-1} \underbrace{\|x(k)\|_Q^2}_{\substack{\text{penalize}\\ \text{bad performance}}} + \underbrace{\|u(k)\|_R^2}_{\substack{\text{penalize}\\ \text{actuator effort}}} + \frac{1}{2}\|x(N)\|_{P_f}^2$$

Control law: $u(k) = K(k)x(k)$ (K *is time-variant*)

while for $k = 0, \ldots, N-1$, with $P(N) = P_f$

$$K(k) = -\left(B^\top P(k+1)B + R\right)^{-1} B^\top P(k+1)A$$

and

$$P(k) = A^\top P(k+1)A + Q - A^\top P(k+1)B$$
$$\left(B^\top P(k+1)B + R\right)^{-1} B^\top P(k+1)A$$

Recursion for the Riccati Matrix:

$$P(k) = A^\top P(k+1)A + Q + \boldsymbol{K}(k)^\top B^\top P(k+1)A$$

In addition Lyapunov function:

$V_0^*(x_0) = \frac{1}{2}x_0^\top P(0)x_0$

5.3. LQ Control (infinite horizon)

System Class: $x^+ = Ax + Bu$

Cost: $V(x, \underline{u}) = \frac{1}{2}\sum_{k=0}^{\infty} \|x(k)\|_Q^2 + \|u(k)\|_R^2$

Control Law: $u(k) = K_\infty x(k)$, while

$K_\infty = -\left(B^\top P_\infty B + R\right)^{-1} B^\top P_\infty A$ and Riccati Equation:

$P_\infty = Q + K_\infty^\top R K_\infty + (A + BK_\infty)^\top P_\infty (A + BK_\infty)$

Stationary Riccati Matrix:

$$P_\infty = A^\top P_\infty A + Q + \boldsymbol{K}_\infty^\top B^\top P_\infty A, \qquad P_\infty \geq 0$$

5.4. MPC (constrained, receding horizon)

Stability of equilibrium $x_{eq} = 0$ under MPC, if

unconstrained: $P_f = P_\infty$

constrained:

1.) $P_f = P_\infty$

2.) constraint admissibility: $\mathbb{X}_f \subseteq \{x \in \mathbb{X} | Kx \in \mathbb{U}\}$

3.) positive invariance: $x \in \mathbb{X}_f \Rightarrow x^+ = (A + BK_\infty)x \in \mathbb{X}_f$

5.5. Underlying Optimization Problem (QP)

Using previews $x(k) = A^k x_0 + A^{k-1}Bu(0) + \ldots + Bu(k-1)$ for all k in horizon in cost function

$$J_N(x_0, \underline{u}) = \frac{1}{2}\sum_{k=0}^{N-1} \|x(k)\|_Q^2 + \|u(k)\|_R^2 + \frac{1}{2}\|x(N)\|_{P_f}^2$$

allows to transform the cost into

$$J_N(x, \underline{u}) = \frac{1}{2}\underline{u}^\top H(x_0)\underline{u} + c(x_0)^\top \underline{u} + d(x_0)$$

with $\underline{u} \in \mathcal{U}_N(x_0)$ polytopes QP problem (= quadratic cost with linear constraints) allows for efficient numerics.

6. Generalized Predictive Control (GPC)

6.1. System Class

Transfer function: $\frac{y(t)}{\Delta u(t-1)} = \frac{B(z^{-1})}{A(z^{-1})}$

System class:

$$A\left(z^{-1}\right)y(t) = B\left(z^{-1}\right)z^{-d}u(t-1) + C\left(z^{-1}\right)\frac{e(t)}{\Delta}$$

- Denominator: $A\left(z^{-1}\right) = 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_n z^{-n}$
- Denumerator: $B\left(z^{-1}\right) = 1 + b_1 z^{-1} + \ldots + b_m z^{-m}$, $m < n$
- $C\left(z^{-1}\right)$ for colored noise, in the following $C\left(z^{-1}\right) = 1$
- $e(t)$: white noise with zero mean

- Shift operator: $z^{-k}y(t) = y(t-k)$
- Dead time: z^{-d}, in the following $d = 0$
- $\Delta = 1 - z^{-1}$ (for $u(t) \leftrightarrow$ for $u(t-1)$: $\Delta = 1$)

Cost:

$$J = \sum_{j=1}^{N} \delta(j)(\underbrace{\hat{y}(t+j|t)}_{\substack{\text{predicted}\\ \text{output}}} - \underbrace{w(t+j)}_{\substack{\text{future ref.}\\ \text{trajectory}}})^2 + \sum_{j=1}^{M} \lambda(j)(\Delta u(t+j-1))$$

- Horizons: N is prediction horizon, M is control horizon. In the following $M = N$
- Weighting $\delta(j)$, $\lambda(j)$
- Control input $\Delta u = u(t) - u(t-1)$
- Prediction from time t to $t+j$:
  $\hat{y}(t+j|t) = G_j(z^{-1})\Delta u(t+j-1) + F_j(z^{-1})y(t)$

6.2. Diophantine Equation

Used to get rid of *inbetween predictions*.

$\rightarrow$ Fewer equations than unknown variables

$1 = E_j z^{-1}\tilde{A}\left(z^{-1}\right) + z^{-j}F_j\left(z^{-1}\right)$ with $\tilde{A} = \Delta A$

$E_j\left(z^{-1}\right)$: polynomial of degree $j-1$

$F_j\left(z^{-1}\right)$: polynomial of degree of $\tilde{A}$

Prediction:

$\underline{y} = \underline{Gu} + \underline{p}$ (prediction $\underline{p}$ often denoted as f)

where the choice of $\underline{y}$ and $\underline{u}$ as:

$$\underline{y} = [\hat{y}(t+1|t), \hat{y}(t+2|t), \ldots, \hat{y}(t+N|t)]^\top$$
$$\underline{u} = [\Delta u(t), \Delta u(t+1), \ldots, \Delta u(t+N-1)]^\top$$

defines G and $\underline{p}$:

$$G = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_0 \end{bmatrix}$$

$$\underline{p} = \underbrace{\underline{F}\left(z^{-1}\right)y(t)}_{\text{past values of } y} + \underbrace{\underline{G'}\left(z^{-1}\right)\Delta u(t-1)}_{\text{future \& past control inputs}}$$

with

$G_j\left(z^{-1}\right) = B\left(z^{-1}\right)E_j\left(z^{-1}\right)$

$$\underline{G'}\left(z^{-1}\right) = \begin{bmatrix} \left(G_1\left(z^{-1}\right) - g_0\right)z \\ \left(G_2\left(z^{-1}\right) - g_0 - g_1 z^{-1}\right)z^2 \end{bmatrix}$$

$\underline{F}\left(z^{-1}\right) = \left[F_1\left(z^{-1}\right), F_2\left(z^{-1}\right), \ldots, F_N\left(z^{-1}\right)\right]^\top$

and g_j j-th coefficients of polynomial G_j.

Excursion: Polynomials $E_j\left(z^{-1}\right)$ and $F_j\left(z^{-1}\right)$ can be obtained by dividing 1 by $\tilde{A}(z^{-1}) = \Delta A(z^{-1})$ until the remainder can be factorized as $z^{-j}F_j(z^{-1})$, e.g.:

$$(1 \qquad) : (\overbrace{1 - 2z^{-1} + 2z^{-2} - z^{-3}}^{\Delta A(z^{-1})}) = \overbrace{\overbrace{1}^{E_1} + 2z^{-1}}^{E_2} + \ldots$$

$$\begin{array}{ll} \underline{1 - 2z^{-1} + 2z^{-2} - z^{-3}} & \\ \quad 2z^{-1} - 2z^{-2} + z^{-3} & \Rightarrow z^{-1}F_1\left(z^{-1}\right) \\ \quad \underline{2z^{-1} - 4z^{-2} + 4z^{-3} - 2z^{-4}} & \\ \qquad 2z^{-2} - 3z^{-3} + 2z^{-4} & \Rightarrow z^{-2}F_2\left(z^{-1}\right) \qquad \ldots \end{array}$$

For $N = 2$:

$E_1(z^{-1}) = 1, \quad E_2(z^{-1}) = 1 + 2z^{-1}$

$\rightarrow \frac{z^{-1}F_1(z^{-1})}{z^{-1}} = F_1(z^{-1}) = 2 - 2z^{-1} + z^{-2}$

$\rightarrow \frac{z^{-2}F_2(z^{-1})}{z^{-2}} = F_2(z^{-1}) = 2 - 3z^{-1} + 2z^{-2}$

6.3. QP Problem

Cost (*with reference* w):

$$J = u\left(G^T QG + R\right)\underline{u} + 2(\underline{p} - \underline{w})^T QG\underline{u} + (\underline{p} - \underline{w})^T Q(\underline{p} - \underline{w})$$

is minimized by control sequence of future controls

$\underline{u} = -\left(G^T QG + R\right)^{-1} G^T Q(\underline{p} - w)$

where only $\underline{u}_1 = \Delta u(t)$ is applied as control.

E.g. for horizon $N = 3$:

$u(t) = u(t-3) + \Delta u(t-2) + \Delta u(t-1) + \Delta u(t)$

where $\Delta u(t)$ is the first element of $\underline{u}$.

7. Numerics

7.1. Nonlinear Programming (NP)

Cost function: $F(z)$

Equality constraints: $g(z) = 0$

Inequality constraints: $h(z) \leq 0$

Active Set: $\mathcal{A} = \{j | \mu_j > 0\}$ (*set of active constraints*)

Sequential: Recursive Elimination (*find optimal* $\underline{u}$)

\+ Small NP problem

− Sensitive dependance for larger N

Parallel: Full Discretization (*find optimal* $\underline{u}$ *and* $\underline{x}$)

\+ Simple to implement \+ Easier for state constraints

− Larger NP problem − Many constraints

Compromise: Multiple Shooting

Necessary conditions for a minimum:

If z^* is feasible minimum, then $\nabla_z L(z^*, \lambda^*, \mu^*) = 0$ and $\nabla_\lambda L(z^*, \lambda^*, \mu^*) = 0$ with $\mu^* \geq 0$, $h(z^*) \leq 0$ and $\mu_i^* h_i(z^*) = 0$ for all i with Lagrange function

$L = F(z) + \lambda^\top g(z) + \mu^\top h(z)$ with multipliers λ and μ.

7.2. Unconstrained Minimization: Newton's Method

Find minimum z^* numerically as follows

- Initialize: Guess $z^{(0)}$ close to z^* for $k = 0, 1, 2, \ldots$
- Update (calculate tangent and corresponding root value):
  $z^{(k+1)} = z^{(k)} + \alpha^{(k)}d^{(k)}$ where
  $d^{(k)} = -\left(\nabla^2 F\left(z^{(k)}\right)\right)^{-1}\nabla F\left(z^{(k)}\right)$ (search direction)
  and step length $\alpha^{(k)} = \operatorname{argmin} F(z^{(k)} + \alpha d^{(k)})$ (line search)
- Stop if $\|\nabla F(z^{(k)})\| < \epsilon_{tol}$ (*after one iteration if* F *is quadratic*)

7.3. Constrained Minimization: Quadratic Programming

Applies for QP problem class:

Quadratic cost function: $F(z) = \frac{1}{2}z^\top Hz + c^\top z$

Linear equality constraints: $g(z) = Ez + e = 0$

Liner inequality constraints: $h(z) = Iz + i \leq 0$

Find minimum z^* numerically as follows

- Initialize: Guess active set $\mathcal{A}_0$ for $k = 0, 1, 2, \ldots$
  If all active constraints are known, solvable in one iteration.
- Update optimization variables z with $\frac{\partial L}{\partial z}$.
  Inequality constraints from active set $\mathcal{A}_k$ are treated as equality constraints. Non-active constraints are neglected.
- Update active set:
  When for $j \in \mathcal{A}_k : \mu_j^{(k+1)} < 0$, delete const. j from active set
  When for $j \notin \mathcal{A}_k I_j z^{(k+1)} + i_j \geq 0$, add constraint j to active set and thus get $\mathcal{A}_{k+1}$
- Stop if $\mathcal{A}_k$ is not active anymore.

7.4. Constrained Minimization: Sequential QP

Combine Newton's Methods (linearization of NP problem to obtain QP problem) and QP method (choice of feasible active set of linearized problem).

7.5. Constrained Minimization: Interior Point

Cost function: $F(z)$

Equality constraints: $g(z) = 0$

Former inequality constraints, now equality constraints: $h(z) + s = 0$

New inequality constraints: $s \geq 0$

Lagrange Function $L = F(z) + y^\top g(z) + w^\top(h(z) + s) - \mu^\top s$

Necessary Conditions that are related to inequality constraints:

$s \geq 0$, $\mu \geq 0$ and $s^\top \mu = 0$

Relax complementary conditions $s^\top \mu = 0$ by introducing barrier parameter ϵ to $s^\top \mu = \epsilon$ and solve with $\epsilon \to 0$.

7.6. Soft Constraints

Softened minimization problem with penalty term $l(\epsilon)$:

$$\min_x f(x) + l(\epsilon), \text{ s.t. } g(x) \leq \epsilon, \ \epsilon \geq 0$$

where ϵ defines the degree of softness.

Quadratic penalty: $l\epsilon) = v \cdot \epsilon^2$ with $v > 0$

Feasible solution x^* for the original problem is not the same for the softened problem for any $v > 0$ and $\epsilon = 0$.

Linear penalty: $l(\epsilon) = u \cdot \epsilon$

Same solution as original problem for $u > \mu^* \geq 0$, where μ^* is the optimal Lagrange multiplier for the original problem.

8. Robust MPC

8.1. Robust Stability

Does state reach vicinity of the origin under bounded small disturbances?

Types of uncertainties: Parametric uncertainty, modeling errors, etc.

E.g. additive disturbance: $x^+ = f(x, u) + w$ where w is disturbance and $w \in \mathbb{W}$, where $\mathbb{W}$ is bounded.

Standard controllers (e.g. PID) have inherent robustness.

Nominal Robust Stability only if Lyapunov Function is continuous (e.g. for linear MPC).

8.2. Tube base MPC for Linear Systems

Nominal model: $z^+ = Az + Bu$

Disturbed (real) model: $x^+ = Ax + Bu + w$

Use tube $S(k)$, $k = 0, 1, 2, \ldots$ to check **constraint admissibility**:

Set $S(k)$ is such that all possible trajectories $x(k)$ (apply all possible disturbances) of real disturbed system fulfill $x(k) \in \{z(k) \oplus S(k)\}$, i.e. do not violate constraints.

MPC feedback control: $u^* = v^* + K(x - z)$, where K is found offline. Appropiate choice of K reduces the size of the tube.

Optimize cost to find v^* such that x^* is constraint admissible by constraint tightening using the tube.