Explicit constrained nonlinear MPC

Tor A. Johansen

Norwegian University of Science and Technology, Trondheim, Norway

Main topics

- Approximate explicit nonlinear MPC based on orthogonal search trees
- Approximate explicit nonlinear MPC based on local mp-QP solutions
- Discussion and summary
Part I

Approximate explicit constrained nonlinear MPC based on orthogonal search trees

- Nonlinear Constrained MPC formulation

- Theory: Stability, optimality, convexity

- Approximate mp-NLP via search tree partitioning

- Example
MPC formulation

Nonlinear system:

\[ x(t + 1) = f(x(t), u(t)) \]

Optimization problem (similar to Chen and Allgöwer (1998)):

\[ V^*(x(t)) = \min_U \left( J(U, x(t)) = \sum_{k=0}^{N-1} \left( ||x_{t+k}|t||^2_Q + ||u_{t+k}|t||^2_R + ||x_{t+N}|t||^2_P \right) \right) \]

with \( U = \{u_t, u_{t+1}, \ldots, u_{t+N-1}\} \) subject to \( x_{t}|t = x(t) \) and

\[
\begin{align*}
  & y_{\min} \leq y_{t+k}|t \leq y_{\max}, \quad k = 1, \ldots, N \\
  & u_{\min} \leq u_{t+k} \leq u_{\max}, \quad k = 0, 1, \ldots, N - 1, \\
  & x_{t+N}|t \in \Omega \\
  & x_{t+k+1}|t = f(x_{t+k}|t, u_{t+k}), \quad k = 0, 1, \ldots, N - 1 \\
  & y_{t+k}|t = Cx_{t+k}|t, \quad k = 1, 2, \ldots, N
\end{align*}
\]
MPC formulation, assumptions

A1. \( P, Q, R \succ 0 \).

A2. \( y_{\text{min}} < 0 < y_{\text{max}} \) and \( u_{\text{min}} < 0 < u_{\text{max}} \).

A3. The function \( f \) is twice continuously differentiable, with \( f(0, 0) = 0 \).

The compact and convex terminal set \( \Omega \) is defined by

\[ \Omega = \{ x \in \mathbb{R}^n \mid x^T P x \leq \alpha \} \]

where \( P \in \mathbb{R}^{n \times n} \) and \( \alpha > 0 \) will be specified shortly.
Terminal set

Similar to Chen and Allgöwer (1998):

**A4.** \((A, B)\) is stabilizable, where

\[
A = \frac{\partial f}{\partial x}(0, 0), \quad B = \frac{\partial f}{\partial u}(0, 0)
\]

Let \(K\) denote the associated LQ optimal gain matrix, such that \(A_0 = A - BK\) is strictly Hurwitz.

**Lemma 1.** If \(\kappa > 0\) is such that \(A_0 + \kappa I\) is strictly Hurwitz, the Lyapunov equation

\[
(A_0 + \kappa I)^T P (A_0 + \kappa I) - P = -Q - K^T R K
\]

has a unique solution \(P > 0\).
Terminal set, cont’d

Lemma 2. Let $P$ satisfy the conditions in Lemma 1. Then there exists a constant $\alpha > 0$ such that $\Omega = \{ x \in \mathbb{R}^n \mid x^T P x \leq \alpha \}$ satisfies

1. $\Omega \subset C = \{ x \in \mathbb{R}^n \mid u_{min} \leq -K x \leq u_{max}, y_{min} \leq C x \leq y_{max} \}$.

2. The autonomous nonlinear system

$$x(t + 1) = f(x(t), -K x(t))$$

is asymptotically stable for all $x(0) \in \Omega$, i.e. $\Omega$ is positively invariant.

3. The infinite-horizon cost for the system in 2, given by

$$J_\infty(x(t)) = \sum_{k=0}^{\infty} \left( \|x_{t+k|t}\|_Q^2 + \|K x_{t+k|t}\|_R^2 \right)$$

satisfies $J_\infty(x) \leq x^T P x$ for all $x \in \Omega$. 

Explicit constrained nonlinear MPC
Multi-parametric nonlinear program (mp-NLP)

This and similar optimization problems can be formulated in a concise form

\[ V^*(x) = \min_U J(U, x) \quad \text{subject to} \quad G(U, x) \leq 0 \]

This defines an mp-NLP, since it is an NLP in \( U \) parameterized by \( x = x(t) \).

Define the set of \( N \)-step feasible initial states as follows

\[ X_F = \{ x \in \mathbb{R}^n \mid G(U, x) \leq 0 \text{ for some } U \in \mathbb{R}^{Nm} \} \]

We seek an explicit state feedback representation of the solution function \( U^*(x) \) to this problem: Implementation without real-time optimization!
How to represent $U^*(x)$ and $X_F$?

mp-LP/mp-QP: Exact piecewise linear (PWL) solution, on a polyhedral partition of $X_F$.

mp-NLP: No structural properties that can be exploited to determine a finite representation of the exact solution.

Keep the PWL representation as an approximation, as it is convenient to work with!
Orthogonal partitioning via a search tree

PWL approximation is flexible and leads to highly efficient real-time search.

How to construct the partition, and local linear approximations?

Explicit constrained nonlinear MPC
PWL approximate mp-NLP approach

How to construct the partition, and local linear approximations?

Construct feasible local linear approximation to the mp-NLP solution from NLP solutions at a finite number of samples in every region of the partition.

Construct the partition such that the approximate solution has a cost close to the optimal cost for all $x \in X_F$.

This is easier under convexity assumptions.
mp-NLP and convexity

A5. $J$ and $G$ are jointly convex for all $(U, x) \in \mathbb{U} \times X_F$, where $\mathbb{U} = [u_{min}, u_{max}]^N$ is the set of admissible inputs.

The optimal cost function $V^*$ can now be shown to have some regularity properties (Mangasarian and Rosen, 1964):

**Theorem 1.** $X_F$ is a closed convex set, and $V^* : X_F \rightarrow \mathbb{R}$ is convex and continuous. □

Convexity of $X_F$ and $V^*$ is a direct consequence of A5, while continuity of $V^*$ can be established under weaker conditions (Fiacco, 1983).
Optimal cost function bounds

Consider the vertices $\mathcal{V} = \{v_1, v_2, \ldots, v_M\}$ of any bounded polyhedron $X_0 \subseteq X_F$. Define the affine function $\bar{V}(x) = \bar{V}_0 x + \bar{t}_0$ as the solution to the following LP:

$$\min_{\bar{V}_0, \bar{t}_0} \left( \bar{V}_0 v + \bar{t}_0 \right) \text{ subject to } \bar{V}_0 v_i + \bar{t}_0 \geq V^*(v_i), \text{ for all } i \in \{1, 2, \ldots, M\}$$

Likewise, define the convex PWL function

$$\underline{V}(x) = \max_{i \in \{1, 2, \ldots, M\}} \left( V^*(v_i) + \nabla^T V^*(v_i) (x - v_i) \right)$$

**Theorem 2.** Consider any bounded polyhedron $X_0 \subseteq X_F$. Then $\underline{V}(x) \leq V^*(x) \leq \bar{V}(x)$ for all $x \in X_0$. $\square$

These bounds can be computed by solving $M$ NLPs.
Feasible local linear approximation

Similar to Bemporad and Filippi (2003):

**Lemma 3.** Consider any bounded polyhedron $X_0 \subseteq X_F$ with vertices $\{v_1, v_2, \ldots, v_M\}$. If $K_0$ and $g_0$ solve the convex NLP ($\beta \geq 0$ arbitrary):

$$
\min_{K_0, g_0} \sum_{i=1}^{M} \left( J(K_0 v_i + g_0, v_i) - V^*(v_i) + \beta \|K_0 v_i + g_0 - U^*(v_i)\|_2^2 \right)
$$

subject to $G(K_0 v_i + g_0, v_i) \leq 0, \quad i \in \{1, 2, \ldots, M\}$

then $\tilde{U}_0(x) = K_0 x + g_0$ is feasible for the mp-NLP for all $x \in X_0$. □

Can be computed from the solution of the same NLPs as above...
Sub-optimality bound

Since $\hat{U}_0(x)$ defined in Lemma 3 is feasible in $X_0$, it follows that

$$\hat{V}(x) = J(\hat{U}_0(x), x)$$

is an upper bound on $V^*(x)$ in $X_0$ such that for all $x \in X_0$

$$0 \leq \hat{V}(x) - V^*(x) \leq \varepsilon_0$$

where

$$\varepsilon_0 = - \min_{x \in X_0} \left( -\hat{V}(x) + V(x) \right)$$
Approximate mp-NLP algorithm

1. Initialize the partition to the whole hypercube, i.e. \( \mathcal{P} = \{ \mathbb{X} \} \). Mark the hypercube \( \mathbb{X} \) as unexplored.

2. Select any unexplored hypercube \( X_0 \in \mathcal{P} \). If no such hypercube exists, the algorithm terminates successfully.

3. Solve the NLP for \( x \) fixed to each of the vertices of the hypercube \( X_0 \) (some of these NLPs may have been solved in earlier steps). If all solutions are feasible, go to step 4. Otherwise, compute the size of \( X_0 \). If it is smaller than some tolerance, mark \( X_0 \) explored and infeasible. Otherwise, go to step 8.

4. If \( 0 \in X_0 \), choose \( \tilde{U}_0(x) = -Kx \) and go to step 5. Otherwise, go to step 6.

5. If \( X_0 \subseteq \Omega \), mark \( X_0 \) as explored and go to step 2. Otherwise, go to step 8.

6. Compute a feasible affine state feedback \( \tilde{U}_0 \), as an approximation to be used in \( X_0 \). If no feasible solution was found, go to step 8.

7. Compute the error bound \( \varepsilon_0 \). If \( \varepsilon_0 \leq \varepsilon \), mark \( X_0 \) as explored and feasible, and go to step 2.

8. Split the hypercube \( X_0 \) into two hypercubes \( X_1 \) and \( X_2 \) using some heuristic rule. Mark both unexplored, remove \( X_0 \) from \( \mathcal{P} \), add \( X_1 \) and \( X_2 \) to \( \mathcal{P} \) and go to step 2.
The origin is an unstable equilibrium point, with a stabilizable linearization. We discretize this system using a sampling interval $T_s = 0.1$. The control objective is given by the weighting matrices

$$Q = \frac{T_s}{2} I_{2\times2}, \quad R = T_s$$

The terminal penalty is given by the solution to the Lyapunov equation

$$P = \begin{pmatrix} 16.5926 & 11.5926 \\ 11.5926 & 16.5926 \end{pmatrix}$$

and a positively invariant terminal region is

$$\Omega = \{ x \in \mathbb{R}^2 \mid x^T P x \leq 0.7 \}$$

The prediction horizon is $N = 1.5/T_s = 15$. 

\textbf{Explicit constrained nonlinear MPC}
Example, partition with 105 regions

Real-time computations: 14 arithmetic operations per sample.
Example, solution and cost

Optimal solution $u^*(x)$ to the left and approximate solution $\tilde{u}(x)$ to the right.

Optimal cost function $V^*(x)$ to the left and sub-optimal cost $\tilde{V}(x)$ to the right.

Explicit constrained nonlinear MPC
Example, simulations

Explicit constrained nonlinear MPC
Properties of the algorithm

The PWL approximation generated is denoted \( \hat{U} : \mathbb{X}' \rightarrow \mathbb{R}^{N_m} \), where \( \mathbb{X}' \) is the union of hypercubes where a feasible solution has been found. It is an inner approximation to \( X_F \) and the approximation accuracy is determined by the tolerance in step 3.

**Theorem 3.** Assume the partitioning rule in step 8 guarantees that the error decreases by some minimum amount or factor at each split. The algorithm terminates with an approximate solution function \( \hat{U} \) that is feasible and satisfies

\[
0 \leq J(\hat{U}(x), x) - V^*(x) \leq \varepsilon
\]

for all \( x \in \mathbb{X}' \). \( \Box \)
Stability

The exact MPC will make the origin asymptotically stable (Chen and Allgöwer, 1998). We show below that asymptotic stability is inherited by the approximate MPC under an assumption on the tolerance $\varepsilon$:

A6. Assume the partition $\mathcal{P}$ generated by Algorithm 1 has the property that for any hypercube $X_0 \in \mathcal{P}$ that does not contain the origin

$$\varepsilon \leq \gamma \min_{x \in X_0} \|x\|_P^2$$

where $\gamma \in (0, 1)$ is given.

Theorem 4. The origin is an asymptotically stable equilibrium point, for all $x(0) \in \mathbb{X}$. $\square$
Non-convex problems

If convexity does not hold, global optimization is generally needed if theoretical guarantees are required:

1. The NLPs must be solved using global optimization in steps 3 and 6.

2. The computation of the error bound $\varepsilon_0$ in step 7 must rely on global optimization, or convex underestimation.

3. The heuristics in step 8 may be modified to be efficient also in the non-convex case.
Key references

T. A. Johansen, Approximate Explicit Receding Horizon Control of Constrained Nonlinear Systems, submitted for publication, 2002


Approximate explicit constrained nonlinear MPC based on local mp-QP solutions.

- Nonlinear Constrained MPC formulation
- Multi-parametric Nonlinear Programming (mp-NLP)
- An approximate mp-NLP algorithm
- Example: Compressor surge control
Nonlinear Constrained Model Predictive Control

Nonlinear dynamic optimization problem

\[ J(u[0, T], x[0, T]) \triangleq \int_0^T l(x(t), u(t), t) dt + S(x(T), T) \]

subject to the inequality constraints for \( t \in [0, T] \)

\[ u_{\text{min}} \leq u(t) \leq u_{\text{max}} \]
\[ g(x(t), u(t)) \leq 0 \]

and the ordinary differential equation (ODE) given by

\[ \frac{dx(t)}{dt} = f(x(t), u(t)) \]

with given initial condition \( x(0) \in X \subset \mathbb{R}^n \).
Multi-parametric Nonlinear Programming

The input $u(t)$ is assumed to be piecewise constant and parameterized by a vector $U \in \mathbb{R}^p$ such that $u(t) = \mu(t, U) \in \mathbb{R}^r$ is piecewise continuous.

The solution to the ODE is $x(t) = \phi(t, U, x(0))$, and the problem is reformulated as:

$$V(U; x(0)) \triangleq \int_0^T l(\phi(t, U, x(0)), \mu(t, U), t)dt + S(\phi(T, U, x(0)), T)$$

subject to

$$G(U; x(0)) \triangleq \begin{pmatrix} \bar{G}(U; x(0)) \\ U - U_{max} \\ U_{min} - U \end{pmatrix} \leq 0$$

This is an NLP in $U$ parameterized by $x(0)$, i.e. an mp-NLP. Solving the mp-NLP amounts to determining an explicit representation of the solution function $U^*(x(0))$. 

Explicit constrained nonlinear MPC
Explicit solution approach

mp-LP and mp-QP solvers are well established: $U^*(x)$ is piecewise linear (PWL).

In mp-NLPs (unlike mp-QPs) there are no structural properties to exploit, such that we seek only an approximation to the solution $U^*(x)$.

Main idea: Locally approximate the mp-NLP with a convex mp-QP. Iteratively sub-partition the space until sufficient agreement between the mp-QP and mp-NLP solutions is achieved.

This leads to a PWL approximation, which allows MPC implementation with a PWL function evaluation rather than solving an NLP in real time.
Assumptions

Consider any \( x_0 \in X \), and solution candidate \( U_0 \) with associated Lagrange multipliers \( \lambda_0 \) and optimal active set \( A_0 \).

**Assumption A1.** \( V \) and \( G \) are twice continuously differentiable in a neighborhood of \( (U_0, x_0) \).

**Assumption A2.** The sufficient first- and second-order conditions (Karush-Kuhn-Tucker) for a local minimum at \( U_0 \) hold.

**Assumption A3.** Linear independence constraint qualification (LICQ) holds, i.e. the active constraint gradients \( \nabla_{UG} A_0(U_0; x_0) \) are linearly independent.

**Assumption A4.** Strict complementary slackness holds, i.e. \( (\lambda_0)_{A_0} > 0 \).
Basic result (Fiacco, 1983)

The optimal solution $U^*(x)$ depends in a continuous manner on $x$: It may be meaningful with a PWL approximation to $U^*(x)$!

**Theorem 1.** If A1 - A3 holds, then

1. $U_0$ is a local isolated minimum.

2. For $x$ in a neighborhood of $x_0$, there exists a unique continuous function $U^*(x)$ satisfying $U^*(x_0) = U_0$ and the sufficient conditions for a local minimum.

3. Assume in addition A4 holds, and let $x$ be in a neighborhood of $x_0$. Then $U^*(x)$ is differentiable and the associated Lagrange multipliers $\lambda^*(x)$ exists, and are unique and continuously differentiable. Finally, the set of active constraints is unchanged, and the active constraint gradients are linearly independent at $U^*(x)$.
Local mp-QP Approximation

Minimize

\[ V_0(U; x) \triangleq \frac{1}{2}(U - U_0)^TH_0(U - U_0) \]
\[ + (D_0 + F_0(x - x_0))(U - U_0) + Y_0(x; x_0) \]

subject to

\[ G_0(U - U_0) \leq E_0(x - x_0) + T_0 \]

The cost and constraints are defined by the matrices

\[ H_0 \triangleq \nabla_{UU}^2 V(U_0; x_0), \quad F_0 \triangleq \nabla_{xU}^2 V(U_0; x_0), \quad D_0 \triangleq \nabla_U V(U_0; x_0) \]
\[ G_0 \triangleq \begin{pmatrix} \nabla_U \tilde{G}(U_0; x_0) \\ I \\ -I \end{pmatrix}, \quad E_0 \triangleq \begin{pmatrix} -\nabla_x \tilde{G}(U_0; x_0) \\ 0 \\ 0 \end{pmatrix}, \quad T_0 \triangleq \begin{pmatrix} -\tilde{G}(U_0; x_0) \\ U_{\max} - U_0 \\ U_0 - U_{\min} \end{pmatrix} \]
\[ Y_0(x; x_0) \triangleq V(U_0; x_0) + \nabla_x V(U_0; x_0)(x - x_0) + \frac{1}{2}(x - x_0)^T \nabla_{xx}^2 V(U_0; x_0)(x - x_0) \]
KKT conditions

Let the PWL solution to the mp-QP be denoted \( U_{QP}(x) \) with associated Lagrange multipliers \( \lambda_{QP}(x) \):

\[
H_0 \left( U_{QP}(x) - U_0 \right) + F_0(x - x_0) + D_0 + G_0^T \lambda_{QP}(x) = 0
\]
\[
\text{diag}(\lambda_{QP}(x)) \left( G_0(U_{QP}(x) - U_0) - E_0(x - x_0) - T_0 \right) = 0
\]
\[
\lambda_{QP}(x) \geq 0
\]
\[
G_0 \left( U_{QP}(x) - U_0 \right) - E_0(x - x_0) - T_0 \leq 0
\]

Assumption A5. For an optimal active set \( \mathcal{A} \), the matrix \( G_{0,\mathcal{A}} \) has full row rank (LICQ) and \( Z_{0,\mathcal{A}}^T H_0 Z_{0,\mathcal{A}} > 0 \), where the columns of \( Z_{0,\mathcal{A}} \) is a basis for \( \text{null}(G_{0,\mathcal{A}}) \).
QP-solution for fixed active set

**Theorem 2.** Let $X$ be a polyhedral set with $x_0 \in X$. If assumption A5 holds, then there exists a unique solution to the mp-QP and the critical region where the solution is optimal is given by the polyhedral set

$$\mathcal{X}_{0,\mathcal{A}} \triangleq \left\{ x \in X \mid \lambda_{QP,\mathcal{A}}(x) \geq 0, \quad G_0(U_{QP,\mathcal{A}}(x) - U_0) \leq E_0(x - x_0) + T_0 \right\}$$

Hence, $U_{QP}(x) = U_{QP,\mathcal{A}}(x)$ and $\lambda_{QP}(x) = \lambda_{QP,\mathcal{A}}(x)$ if $x \in \mathcal{X}_{0,\mathcal{A}}$, and the solution $U_{QP}$ is a continuous, PWL function of $x$ defined on a polyhedral partition of $X$. 
Comparing mp-QP and mp-NLP solutions

Local quadratic approximation makes sense!

**Theorem 3.** Let $x_0 \in X$ and suppose there exists a $U_0$ satisfying assumptions A1 - A4. Then for $x$ in a neighbourhood of $x_0$

\[
U_{QP}(x) - U^*(x) = \mathcal{O}(\|x - x_0\|^2_2)
\]

\[
\lambda_{QP}(x) - \lambda^*(x) = \mathcal{O}(\|x - x_0\|^2_2)
\]
Approximation criteria

How small neighborhoods $X_0 \subset X$ are necessary?

The solution error bound is defined as ($w$ is a weight)

$$\rho \triangleq \max_{x \in X_0} \left| w^T (\mu(0, U_{QP}(x)) - \mu(0, U^*(x))) \right|$$

The cost error bound is defined as

$$\varepsilon \triangleq \max_{x \in X_0} |V(U_{QP}(x); x) - V^*(x)|$$

The maximum constraint violation is ($\omega$ is a weight)

$$\delta \triangleq \max_{x \in X_0} \omega^T G(U_{QP}(x); x)$$

Explicit constrained nonlinear MPC
Optimal cost function bounds

Consider the vertices $V = \{v_1, v_2, \ldots, v_M\}$ of any bounded polyhedron $X_0$. Define the affine function $\overline{V}(x) \triangleq \overline{V}_0 x + \overline{l}_0$ as the solution to the LP

$$\min_{\overline{V}_0, \overline{l}_0} \left( \overline{V}_0 v + \overline{l}_0 \right)$$

subject to

$$\overline{V}_0 v_i + \overline{l}_0 \geq V^*(v_i), \quad \text{for all } i \in \{1, 2, \ldots, M\}$$

Likewise, define the convex piecewise affine function

$$\underline{V}(x) \triangleq \max_{i \in \{1,2,\ldots,M\}} \left( V^*(v_i) + \nabla^T V^*(v_i)(x - v_i) \right)$$

**Theorem 4.** If $V$ and $G$ are jointly convex (in $U$ and $x$) on the bounded polyhedron $X_0$, then $\underline{V}(x) \leq V^*(x) \leq \overline{V}(x)$ for all $x \in X_0$. 

Explicit constrained nonlinear MPC
Optimal cost function bounds

Theorem 4 gives the following bounds on the cost function error

\[-\varepsilon_1 \leq V^*(x) - V(U_{QP}(x); x) \leq \varepsilon_2\]

where

\[\varepsilon_1 = \max_{x \in X_0} \left( V(U_{QP}(x); x) - V(x) \right)\]

\[\varepsilon_2 = \max_{x \in X_0} \left( \overline{V}(x) - V(U_{QP}(x); x) \right)\]
Approximate mp-NLP algorithm

Step 1. Let $X_0 := X$.

Step 2. Select $x_0$ as the Chebychev center of $X_0$, by solving an LP.

Step 3. Compute $U_0 = U^*(x_0)$ by solving a NLP.

Step 4. Compute the local mp-QP problem at $(U_0, x_0)$.

Step 5. Estimate the approximation errors $\epsilon$, $\rho$ and $\delta$ on $X_0$.

Step 6. If $\epsilon > \underline{\epsilon}$, $\rho > \bar{\rho}$, or $\delta > \bar{\delta}$, then sub-partition $X_0$ into polyhedral regions.

Step 7. Select a new $X_0$ from the partition. If no further sub-partitioning is needed, go to step 8. Otherwise, repeat Steps 2-7 until the tolerances $\underline{\epsilon}$, $\bar{\rho}$ and $\bar{\delta}$ are respected in all polyhedral regions in the partition of $X$.

Step 8. For all sub-partitions $X_0$, solve the mp-QP.

Explicit constrained nonlinear MPC
Example: Compressor surge control

Consider the following 2nd-order compressor model with \( x_1 \) being normalized mass flow, \( x_2 \) normalized pressure and \( u \) normalized mass flow through a close coupled valve in series with the compressor

\[
\begin{align*}
\dot{x}_1 &= B (\psi_e(x_1) - x_2 - u) \\
\dot{x}_2 &= \frac{1}{B} (x_1 - \Phi(x_2))
\end{align*}
\]

The following compressor and valve characteristics are used

\[
\psi_e(x_1) = \psi_{c0} + H \left( 1 + 1.5 \left( \frac{x_1}{W} - 1 \right) - 0.5 \left( \frac{x_1}{W} - 1 \right)^3 \right)
\]

\[
\Phi(x_2) = \gamma \text{sign}(x_2) \sqrt{|x_2|}
\]

with \( \gamma = 0.5 \), \( B = 1 \), \( H = 0.18 \), \( \psi_{c0} = 0.3 \) and \( W = 0.25 \), (Gravdahl and Egeland, 1997).
Example, cont’d

The control objective is to avoid surge, i.e. stabilize the system. This may be formulated as

\[
\begin{align*}
    l(x, u) &= \alpha(x - x^*)^T(x - x^*) + \kappa u^2 \\
    S(x) &= Rv^2 + \beta(x - x^*)^T(x - x^*)
\end{align*}
\]

with $\alpha, \beta, \kappa, \rho \geq 0$ and the setpoint $x_1^* = 0.40$, $x_2^* = 0.60$ corresponds to an unstable equilibrium point.

We have chosen $\alpha = 1$, $\beta = 0$, and $\kappa = 0.08$. The horizon is chosen as $T = 12$, which is split into $N = p = 15$ equal-sized intervals.

Valve capacity requires the constraint $0 \leq u(t) \leq 0.3$ to hold, and the pressure constraint $x_2 \geq 0.4 - v$ avoids operation too far left of the operating point.
The PWL mapping with 379 regions can be represented as a binary search tree with 329 nodes, of depth 9. Real-time evaluation of the controller therefore requires 49 arithmetic operations, in the worst case, and 1367 numbers needs to be stored in real-time computer memory.

Explicit constrained nonlinear MPC
Optimal solution and cost

Approximate solution

Exact solution

Explicit constrained nonlinear MPC
Simulation with controller activated at $t = 20$

Explicit constrained nonlinear MPC
Key references


Explicit linear MPC (including robust/hybrid)

- Solved using mp-LP/mp-QP/mp-MILP
- Exact piecewise linear representation
- Solvers efficient for constrained MPC problems with a few states and inputs and reasonable horizons
- Real-time implementation as binary search tree: $\mu$s computation times using fixed-point arithmetic
Discussion and summary, cont’d

Explicit non-linear MPC - emerging mp-NLP theory/technology based on mp-QP/mp-LP and NLP solvers.

**Bad news:** Non-convexity is challenging (not unexpected...)

**Good news:** i) Approximation theory (including stability) seems to be fairly straightforward to establish. ii) Real-time computational complexity of PWL approximation is comparable to linear MPC!
Advantages and disadvantages

+ Efficient real-time computations for small problems
+ Simple real-time SW implementation: Verifiability, reliability, safety-critical applications, small systems
+ Fixed-point arithmetic sufficient

- Requires more real-time computer memory
- Does not scale well to large problems
- Reconfigurability is not straightforward

Explicit solution is probably most interesting for nonlinear MPC problems: They are often smaller scale, and SW reliability and computational complexity issues are more pronounced.
Applications

- Embedded systems - low-level control, small systems, may be safety-critical (automotive, medical, marine etc.)

- Control allocation problems - static/quasi-dynamic, over-actuated mechanical systems (aerospace, marine, MEMS,...)

- Low-level process control - MPC-like functionality at unit-process level replacing PID-loops???