Nonlinear Model Predictive Control Tools (NMPC Tools)

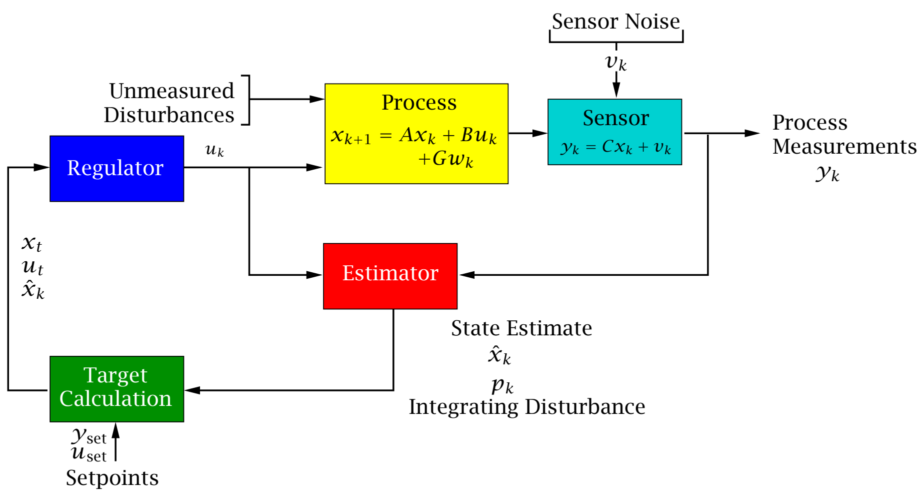
Rishi Amrit,* James B. Rawlings†

April 5, 2008

1 Formulation

We consider a control system composed of three parts([2]).

- Estimator
- Target calculator
- Regulator



The first part, the state estimator, determines an approximate current state of the system, knowing the history of injected inputs and measured

*amrit@wisc.edu
†rawlings@engr.wisc.edu

outputs. The state estimator is also used to estimate the integrating disturbance state. The second part is the steady-state target calculation, which adjusts the state and input targets to account for the integrated disturbance. The final part, the MPC regulator, is responsible for finding the best control profile given steady-state targets for the states and inputs. In some formulations, the regulator may instead be used to track a dynamic output trajectory.

1.1 Disturbance Model

We wish to provide the user with flexibility of defining the disturbance model. For the same reason, we define the disturbance model by specifying two matrices, B_d and C_d. The augmented model becomes:

$$\begin{aligned}
\underbrace{\begin{bmatrix} x \\ p \end{bmatrix}_{k+1}}_{\hat{x}_{k+1}} &= \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \underbrace{\begin{bmatrix} x \\ p \end{bmatrix}_{k}}_{\hat{x}_{k}} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} G & 0 \\ 0 & \Phi \end{bmatrix} \underbrace{\begin{bmatrix} w \\ \xi \end{bmatrix}_{k}}_{\hat{w}_{k}} \\
y_k &= \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix}_{k} + v_k
\end{aligned}$$

in which the index k represents the current sampling time, x_k is the state of the system, u_k is the input, w_k is the stochastic noise variable, and t_k is the time. We assume that w_k is normally distributed and has a zero mean. The term $B_d p_k$ is the integrated input disturbance. In cases of plant/model mismatch or nonzero mean disturbances, this term is nonzero; however, in the nominal case in which the plant and model are identical, this term vanishes. y_k is the measurement at time t_k. ξ_k is a normally distributed zero-mean vector. v_k is a stochastic Gaussian zero-mean noise term. $\hat{x}_k$ and $\hat{w}_k$ are the augmented state and disturbance vectors. Hence the model in terms of augmented state and disturbance vectors is:

$$\begin{aligned}
\hat{x}_{k+1} &= \hat{A}\hat{x}_k + \hat{B}u_k + \hat{G}\hat{w}_k \\
y_k &= \hat{C}\hat{x}_k + v_k
\end{aligned}$$

2 Moving Horizon Estimation

Modules: `lmhe`, `nmhe`

Linear MHE (Module: `lmhe`)

`lmhe` solves the following MHE problem. The model is linear and the user provides the model matrices.

$$\min_{w_{aug},v,x_{aug}} \sum_{k=0}^{N-1} \Big(\frac{1}{2}(w_k' Q_k w_k + v_k' R_k v_k + x_k' \Theta_k x_k + 2x_k' L_k w_k + \epsilon_k' Z_k \epsilon_k + \xi_k' \Psi_k \xi_k$$

$$+ q_k' w_k + r_k' v_k + \theta_k' x_k) + z_k' \epsilon_k + \psi_k' \xi_k\Big) + \frac{1}{2}\rho' P_N \rho + p_N' \rho$$

subject to:

$$\begin{aligned}
\hat{x}_0 &= \bar{x}_0 + \rho \\
\underbrace{\begin{bmatrix} x \\ p \end{bmatrix}_{k+1}}_{\hat{x}_{k+1}} &= \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \underbrace{\begin{bmatrix} x \\ p \end{bmatrix}_k}_{\hat{x}_k} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} G & 0 \\ 0 & I \end{bmatrix} \underbrace{\begin{bmatrix} w \\ \xi \end{bmatrix}_k}_{\hat{w}_k} \\
y_k &= \begin{bmatrix} C & C_d \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix}_k + v_k
\end{aligned}$$

$$\begin{aligned}
S_k w_k + E_{f_k} \hat{x}_k &\le s_k \\
H_k \hat{x}_k - \epsilon_k &\le h_k \\
\epsilon_k &\ge 0 \\
\Gamma_k v_k - \xi_k &\le \gamma_k \\
\xi_k &\ge 0 \\
S_\rho \rho &\le s_\rho
\end{aligned}$$

in which

- $\bar{x}_0$ is the a priori estimate of the initial state
- Optional constraints on state and state noise (specified by matrices S_k, Ef_k and s_k) are hard constraints.
- State constraint (specified by H_k and h_k) are softened

The module returns the estimated states, optimal ρ and the predicted state and measurement noise vectors.

2.1 Nonlinear MHE (Module: `nmhe`)

`nmhe` solves the following nonlinear MHE problem.

$$\min_{w,v,x} \sum_{k=0}^{N-1} (\frac{1}{2}(w_k' Q_k w_k + v_k' R_k v_k) + \frac{1}{2}\rho_N' P_N \rho_N + p_N' \rho_N$$

subject to:

$$\begin{aligned} x_0 &= \bar{x}_0 + \rho \\ x_{k+1} &= F(x_k, u_k) + G_k w_k + B_d p_k \\ y_k &= g(x_k) + v_k + C_d p_k \\ p_{k+1} &= p_k + \xi_k \\ H_k x_k &\leq h_k \text{ (constraint is softened)} \\ S_k w_k &\leq s_k \text{ (constraint is softened)} \\ G_k v_k &\leq g_k \end{aligned}$$

The user provides the system model as a system of ordinary differential equations

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

This model is integrated to yield the function $F(x_k, u_k)$ (See Appendix A).

3 Linear MPC

3.1 Target calculation

Depending on the number of inputs and outputs, the setpoint of the system may or may not be reachable. For cases when the system cannot achieve a steady state at the desired setpoint, a target steady state closest to the setpoint is estimated in a least squares sense.

Instead of solving separate problems to establish the target, a single formulation is preferred. The target tracking problem is formulated as a single quadratic program [1], that achieves the output target, if possible, and relaxes the the problem if the target is infeasible. For this purpose a soft constraint is formulated as follows:

$$\begin{aligned} y_{sp} - Cx_s - X_y p &\leq \eta \\ y_{sp} - Cx_s - X_y p &\geq -\eta \\ \eta &\geq 0 \end{aligned}$$

The constraint $y_{sp} = Cx_s + X_y p$ is relaxed by using the slack variable η. By suitably penalizing η, we guarantee that the relaxed constraint is binding when it is feasible.

The penalty is chosen to be a combination of a linear penalty $q_s \eta$ and a quadratic penalty $\eta' Q_s \eta$, in which the elements of q_s are strictly non negative and Q_s is positive definite. By choosing a sufficiently large q_s, the soft constraint can be guaranteed to be exact.

Module: `ltarget`: Solves the linear MPC target problem

$$\min_{x_s, u_s, \eta} \frac{1}{2}(\eta' Q_s \eta + (u_s - u_{sp})' R_s (u_s - u_{sp})) + q_s' \eta$$

where:

$$\begin{aligned}
(I - A)x_s &= Bu_s + B_d p \\
\eta &\geq y_{sp} - Cx_s + C_d p \\
-\eta &\leq y_{sp} - Cx_s + C_d p \\
\eta &\geq 0 \\
Du - Hx &\leq d
\end{aligned}$$

3.2 Regulator

In the regulation problem, we assume the stochastic variables w_k and v_k take on their mean values.

Module: `lmpc`: Solves the linear MPC problem

$$\min_{\tilde{x}, \tilde{u}, \epsilon} \sum_{k=0}^{N-1} \left(\frac{1}{2}(\tilde{x}_k' Q_k \tilde{x}_k + \tilde{u}_k' R_k \tilde{u}_k + 2\tilde{x}_k' M_k \tilde{u}_k + \epsilon_k' Z_k \epsilon_k) + q_k' \tilde{x}_k + r_k' \tilde{u}_k + z_k' \epsilon_k \right) + 1/2(\tilde{x}_N' P \tilde{x}_N + \epsilon_N' Z_N \epsilon_N) + p' \tilde{x}_N + z_N' \epsilon_N$$

subject to:

$$\begin{aligned}
\tilde{x}_{k+1} = A_k \tilde{x}_k + B_k \tilde{u}_k + F_k & \qquad k = 0, \ldots, N-1 \\
D_k \tilde{u}_k - G_k \tilde{x}_k \leq d_k & \qquad k = 0, \ldots, N-1 \\
H_k \tilde{x}_k - \epsilon_k \leq h_k & \qquad k = 0, \ldots, N \\
\epsilon_k \geq 0 & \qquad k = 0, \ldots, N
\end{aligned}$$

In all the Linear MPC modules, the user defines the problem matrices.

3.3 Quadratic terminal constraint regulator

Module: `lmpc_term`: Solves the linear MPC problem with a quadratic terminal constraint

$$\min_{\tilde{x},\tilde{u},\epsilon} \sum_{k=0}^{N-1} \Big(\frac{1}{2}(\tilde{x}_k' Q_k \tilde{x}_k + \tilde{u}_k' R_k \tilde{u}_k + 2\tilde{x}_k' M_k \tilde{u}_k + \epsilon_k' Z_k \epsilon_k) + q_k' \tilde{x}_k + r_k' \tilde{u}_k + z_k' \epsilon_k\Big) +$$
$$1/2(\tilde{x}_N' P \tilde{x}_N + \epsilon_N' Z_N \epsilon_N) + p' \tilde{x}_N + z_N' \epsilon_N$$

subject to:

$$\begin{aligned}
\tilde{x}_{k+1} &= A_k \tilde{x}_k + B_k \tilde{u}_k + F_k & k &= 0, \dots, N-1 \\
D_k \tilde{u}_k - G_k \tilde{x}_k &\leq d_k & k &= 0, \dots, N-1 \\
H_k \tilde{x}_k - \epsilon_k &\leq h_k & k &= 0, \dots, N \\
\epsilon_k &\geq 0 & k &= 0, \dots, N \\
(\tilde{x}_N - a_T)' \pi (\tilde{x}_N - a_T) &\leq b_T & & \qquad (1)
\end{aligned}$$

The quadratic constraint defined by equation (1), ensures that the terminal state resulting from the regulation optimization, lies within the specified region (elliptical in this case as the constraint is quadratic) around the final target.

4 Nonlinear MPC

4.1 Disturbance model formulation

The toolbox provides a flexible interface such that a wide variety of models can be simulated. The nonlinear modules require the user to specify the process model as a set of differential equations, which they use for sensitivity calculations. Also, this allows the user to specify disturbance models of their choice. The user defined differential equations are of the form:

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

The user has a choice of either pre-augmenting the integrating disturbance or specifying the linear disturbance multiplier in the following discrete time representation of the model:

$$x_{k+1} = F(x_k, u_k) + B_d p_k$$

The relationship between $F(x_k, u_k)$ and $f(x, u, t)$ is given in Appendix A. To demonstrate the disturbance model formulation using the above methodology, we now show formulations of the two common disturbance models, namely the input disturbance model and the output disturbace model.

4.1.1 Output disturbance model

The parameter B_d is 0 and C_d is identity. The user input takes the following form:

$$\begin{aligned}\dot{x} &= f(x,u,t)\\ y &= g(x) + C_d p\end{aligned}$$

When the module discretizes this model, the following discrete time representation is obtained:

$$\begin{aligned}x_{k+1} &= F(x_k,u_k) + Gw_k\\ p_{k+1} &= p_k + \xi_k\end{aligned}$$

4.1.2 Input disturbance model

Augment the state and the disturbance ad define the system as following:

$$\underbrace{\begin{bmatrix}\dot{x}\\ \dot{p}\end{bmatrix}}_{\dot{\tilde{x}}} = \underbrace{\begin{bmatrix}f(x,u+X_u p,t)\\ 0\end{bmatrix}}_{\tilde{f}(\tilde{x},u)}$$

Note that the parameter B_d is set 0 as the disturbance has been augmented. When the module discretizes this model, the following discrete time representation is obtained:

$$\underbrace{\begin{bmatrix}x_{k+1}\\ p_{k+1}\end{bmatrix}}_{\tilde{x}_{k+1}} = \underbrace{\begin{bmatrix}F(x_k,u_k+X_u p_k)\\ p_k\end{bmatrix}}_{\tilde{f}(\tilde{x}_k,u_k)} + \begin{bmatrix}G\\ I\end{bmatrix}\underbrace{\begin{bmatrix}w_k\\ \xi_k\end{bmatrix}}_{\text{Stochastic term}}$$

4.2 Target calculation

Same idea applies for target calculation in non linear systems as explained for linear systems. The model in this case is a nonlinear function of states and inputs.

Module: `ntarget`: Solves the nonlinear MPC target problem

$$\min_{x_s,u_s,\eta} \frac{1}{2}(\eta' Q_s \eta + (u_s - u_{sp})' R_s (u_s - u_{sp})) + q_s' \eta$$

subject to:

$$\begin{aligned}F(x_s,u_s) &= 0\\ \eta &\geq y_{sp} - y(x)\\ -\eta &\leq y_{sp} - y(x)\\ \eta &\geq 0\\ Du - Hx &\leq d\end{aligned}$$

The user provides the system model as a system of ordinary differential equations.

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

This model is integrated to yield the function $F(x_k, u_k)$ (See Appendix A). The user also provides the nonlinear function describing the dependence of output on the states.

$$y_k = g(x_k)$$

4.3 Controller

Module: `nmpc`: Solves the nonlinear MPC problem

$$\min_{x,u}((1/2)((x_k - x_{set})'Q_k(x_k - x_{set}) + (u_k - u_{set})'R_k(u_k - u_{set}) + \\ (u_k - u_{k-1})'S(u_k - u_{k-1})) + (1/2)(x_N - x_{set})'P(x_N - x_{set}))$$

subject to:

$$\begin{aligned}
x_{k+1} &= F(x_k, u_k) + G_k w_k + B_d p_k \\
y_k &= g(x_k) + v_k + C_d p_k \\
p_{k+1} &= p_k + \xi_k \\
H_k \tilde{x}_k &\leq h_k \text{ (constraint is softened)} \\
E_k y_k &\leq e_k \text{ (constraint is softened)} \\
D_k \tilde{u}_k &\leq d_k \\
G_k(\tilde{u}_k - \tilde{u}_{k-1}) &\leq g_k
\end{aligned}$$

The user provides the system model as a system of ordinary differential equations.

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

This model is integrated to yield the function $F(x_k, u_k)$ (See Appendix A). The user also provides the nonlinear function describing the dependence of output on the states.

$$y_k = g(x_k)$$

A Discrete time model from a system of ODE's

In the nonlinear modules of the package, the user interface allows the user to provide the system model as a set of ordinary differential equations

$$\begin{aligned} \frac{dx}{dt} &= f(x,u) \\ x(t_0) &= x_0 \end{aligned}$$

Let $S(t, x_0, u_c)$ be the solution to the model with input $u(t) = u_c$, where u_c is a constant on the interval $[0, t]$. Then the model function in discrete time is given by

$$F(x_k, u_k) = S(\Delta, x_k, u_k)$$

where Δ is the sampling time.

B Quadratic terminal constraint problem

B.1 Problem formulation

The linear problem regulator solves a quadratic objective satisfying linear constraints. To account for a quadratic terminal constraint, we add a terminal cost in the objective and increase it until the terminal constraint is satisfied. The problem formulation becomes:

$$\min_{\tilde{x},\tilde{u},\epsilon} \sum_{k=0}^{N-1} \left(\frac{1}{2}(\tilde{x}_k' Q_k \tilde{x}_k + \tilde{u}_k' R_k \tilde{u}_k + 2\tilde{x}_k' M_k \tilde{u}_k + \epsilon_k' Z_k \epsilon_k) + q_k' \tilde{x}_k + r_k' \tilde{u}_k + z_k' \epsilon_k \right) + \\ 1/2(\tilde{x}_N' P \tilde{x}_N + \epsilon_N' Z_N \epsilon_N) + p' \tilde{x}_N + z_N' \epsilon_N + \lambda(\tilde{x}_N - a_T)' \pi_N (\tilde{x}_N - a_T)$$

subject to:

$$\begin{aligned} \tilde{x}_{k+1} = A_k \tilde{x}_k + B_k \tilde{u}_k + F_k \qquad & k = 0, \ldots, N-1 \\ D_k \tilde{u}_k - G_k \tilde{x}_k \le d_k \qquad & k = 0, \ldots, N-1 \\ H_k \tilde{x}_k - \epsilon_k \le h_k \qquad & k = 0, \ldots, N \\ \epsilon_k \ge 0 \qquad & k = 0, \ldots, N \end{aligned}$$

Under appropriate convexity assumptions, $(\tilde{x}_N - a_T)' \pi_N (\tilde{x}_N - a_T)$ is a non-increasing function of λ for $\lambda \ge 0$. In the above objective, λ is varied until following constraint is satisfied

$$(\tilde{x}_N - a_T)' \pi_N (\tilde{x}_N - a_T) \le b_T$$

B.2 Algorithm

If the solution corresponding to $\lambda = 0$ does not satisfy the constraint, the algorithm then looks for a higher value of λ for which the constraint is satisfied. The module provides a calling sequence in which the user can pass a value of λ from the previous MPC iteration. Then a sufficiently big value of λ is searched. This is done by doubling the value until the constraint is satisfied. Bisection method is then applied to estimate the value of λ corresponding to which the solution satisfies the tolerances according to the following equation.

$$\frac{b_T}{(1 + tol_A)} \leq (\tilde{x}_N - a_T)'\pi_N(\tilde{x}_N - a_T) \leq b_T(1 + tol_A)$$

where tol_A is the tolerance specified by the user. The flow of the algorithm is given in Algorithm 1.

Algorithm 1 Quadratic terminal constraint regulator

Evaluate $\mathbf{x}^*$ for $\lambda = 0$
if $(\tilde{x}_N^* - a_T)'\pi_N(\tilde{x}_N^* - a_T) > b_T$ **then**
  if λ_g not provided by user **then**
    Initialize λ_g
  end if
  Evaluate $\mathbf{x}^*$ for $\lambda = \lambda_g$
  while $(\tilde{x}_N^* - a_T)'\pi_N(\tilde{x}_N^* - a_T) > b_T$ **do**
    $\lambda_g \leftarrow 2\lambda_g$
    Evaluate $\mathbf{x}^*$ for $\lambda = \lambda_g$
  end while
  $\lambda_L \leftarrow 0$, $\lambda_R \leftarrow \lambda_g$
  $\lambda_C \leftarrow (\lambda_L + \lambda_R)/2$
  Evaluate $\mathbf{x}^{\mathbf{C}}$ for $\lambda = \lambda_C$
  while $(\tilde{x}_N^C - a_T)'\pi_N(\tilde{x}_N^C - a_T) \notin [\frac{b_T}{(1+tol_A)}, b_T(1 + tol_A)]$ **do**
    if $(\tilde{x}_N^C - a_T)'\pi_N(\tilde{x}_N^C - a_T) \leq b_T$ **then**
      $\lambda_R \leftarrow \lambda_C$
    else
      $\lambda_L \leftarrow \lambda_C$
    end if
    $\lambda_C \leftarrow (\lambda_L + \lambda_R)/2$
    Evaluate $\mathbf{x}^{\mathbf{C}}$ for $\lambda = \lambda_C$
  end while
else
  x^* already satisfies constraint
end if

B.3 Example

A two state one input example was simulated to test the above algorithm.

$$A = \begin{bmatrix} 4/3 & -2/3 \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C = \begin{bmatrix} -2/3 & 1 \end{bmatrix} \tag{2}$$

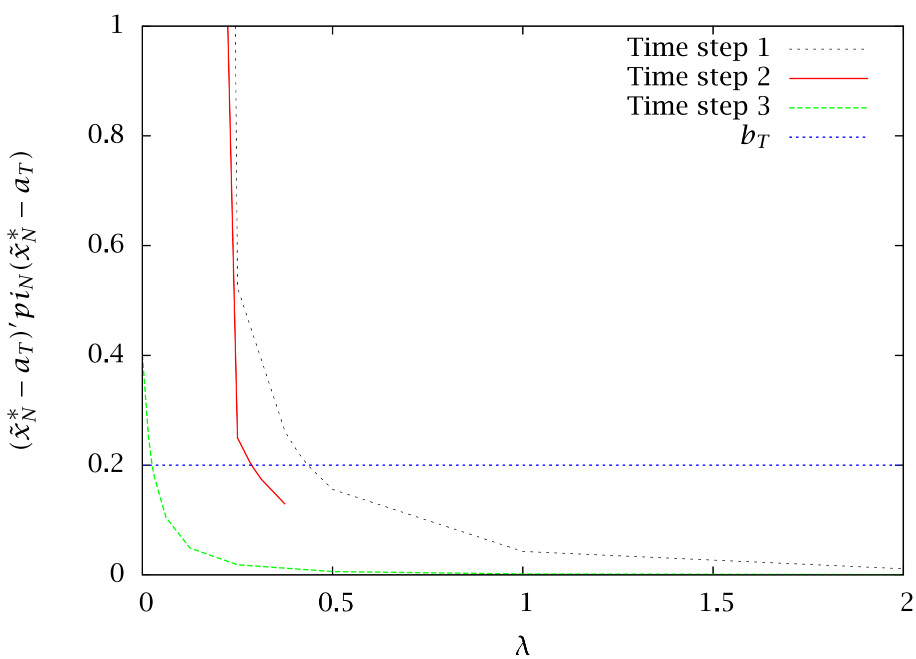
The controller objective matrices were chosen as:

$$Q = I, R = I, N = 4 \tag{3}$$

The input was constrained by $|u| \leq 1$, and the initial state was set to $[3, 3]'$. The following quadratic constraint matrices were chosen:

$$\pi_T = \begin{bmatrix} 70 & 0 \\ 0 & 70 \end{bmatrix}, b_T = 0.2 \tag{4}$$

The following terminal cost profiles were obtained for first 3 consecutive regulator optimizations. The optimal value of λ corresponds to $\tilde{x}_N \pi_N \tilde{x}_N' = b_T = 0.2$



Optimal solution corresponding to $\lambda = 0$ satisfies the constraint in all the optimizations at the following time steps, signifying that the state of the system was then close enough to the origin for the terminal state to lie in the desired terminal region.

References

[1] James B. Rawlings. Tutorial: Model predictive control technology. In *Proceedings of the American Control Conference, San Diego, CA*, pages 662–676, 1999.

[2] M. Tenny. *Computational Strategies for Nonlinear Model Predictive Control*. PhD thesis, University of Wisconsin–Madison, 2002.