
IEOR 290A – Lecture 28

L2NW in LB MPC

1 Using L_2 -Regularized Nadaraya-Watson Estimator as Oracle

Suppose that we use the L2NW estimator as an oracle for LB MPC:

$$\mathcal{O}_n(\tilde{x}, \tilde{u}) = \frac{\sum_{i=0}^{n-1} (x_{i+1} - Ax_i - Bu_i) \cdot K \left(h^{-2} \cdot \left\| \begin{bmatrix} \tilde{x} \\ \tilde{u} \end{bmatrix} - \begin{bmatrix} x_i \\ u_i \end{bmatrix} \right\|^2 \right)}{\lambda + \sum_{i=0}^{n-1} K \left(h^{-2} \cdot \left\| \begin{bmatrix} \tilde{x} \\ \tilde{u} \end{bmatrix} - \begin{bmatrix} x_i \\ u_i \end{bmatrix} \right\|^2 \right)},$$

where $K(\cdot)$ is a kernel function. For notational convenience, we will define

$$\Xi_i = h^{-2} \cdot \left\| \begin{bmatrix} \tilde{x} \\ \tilde{u} \end{bmatrix} - \begin{bmatrix} x_i \\ u_i \end{bmatrix} \right\|^2,$$

and

$$Y_i = x_{i+1} - (Ax_i + Bu_i),$$

Similarly, we will define

$$\xi = \begin{bmatrix} \tilde{x} \\ \tilde{u} \end{bmatrix},$$

and

$$X_i = \begin{bmatrix} x_i \\ u_i \end{bmatrix}.$$

With this notation, we can write the L2NW estimator in much simpler notation:

$$\mathcal{O}_n(\tilde{x}, \tilde{u}) = \frac{\sum_{i=0}^{n-1} Y_i K(\Xi_i)}{\lambda + \sum_{i=0}^{n-1} K(\Xi_i)}.$$

It turns out that the deterministic properties of the L2NW estimator are well-suited for LB MPC. In particular, if $0 \in \mathcal{W}$ and $\lambda > 0$, then the L2NW estimator is differentiable and $\mathcal{O}(\tilde{x}, \tilde{u}) \in \mathcal{W}$ (meaning that it is bounded). This boundedness condition is important for achieving robust asymptotic stability (RAS) when using L2NW as an oracle, and the differentiability is important for numerical implementation because optimization software computes faster and more accurately when given gradients of functions in the objectives and constraints.

2 Stochastic Epi-Convergence of L2NW in LBMPC

It can be shown that under certain technical conditions (including $h_n \rightarrow 0$ and $\lambda = O(h_n)$), the L2NW estimator converges uniformly in probability. As a result, we have that if $g(x, u)$ is Lipschitz continuous, then the control law of LBMPC with L2NW converges in probability to the control law of an MPC that knows the true model. In the next subsection, we describe a simulation of LBMPC with L2NW applied to a nonlinear system. Note that even though the LBMPC optimization is non-convex in the scenario we present, using local optima gives good performance.

2.1 EXAMPLE: MOORE-GREITZER COMPRESSOR MODEL

The compression system of a jet engine can exhibit two types of instability: rotating stall and surge. Rotating stall is a rotating region of reduced air flow, and it degrades the performance of the engine. Surge is an oscillation of air flow that can damage the engine. Historically, these instabilities were prevented by operating the engine conservatively. But better performance is possible through active control schemes.

The Moore-Greitzer model is an ODE model that describes the compressor and predicts surge instability

$$\begin{aligned}\dot{\Phi} &= -\Psi + \Psi_c + 1 + 3\Phi/2 - \Phi^3/2 \\ \dot{\Psi} &= (\Phi + 1 - r\sqrt{\Psi})/\beta^2,\end{aligned}$$

where Φ is mass flow, Ψ is pressure rise, $\beta > 0$ is a constant, and r is the throttle opening. We assume r is controlled by a second order actuator with transfer function $r(s) = \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2} u(s)$, where ζ is the damping coefficient, w_n^2 is the resonant frequency, and u is the input.

We conducted simulations of this system with the parameters $\beta = 1$, $\Psi_c = 0$, $\zeta = 1/\sqrt{2}$, and $w_n = \sqrt{1000}$. We chose state constraints $0 \leq \Phi \leq 1$ and $1.1875 \leq \Psi \leq 2.1875$, actuator constraints $0.1547 \leq r \leq 2.1547$ and $-20 \leq \dot{r} \leq 20$, and input constraints $0.1547 \leq u \leq 2.1547$. For the controller design, we took the approximate model with state $\delta x = [\delta\Phi \ \delta\Psi \ \delta r \ \delta\dot{r}]'$ to be the exact discretization (with sampling time $T = 0.01$) of the linearization of (??) about the equilibrium $x_0 = [\Phi_0 \ \Psi_0 \ r_0 \ \dot{r}_0]' = [0.5000 \ 1.6875 \ 1.1547 \ 0]'$; the control is $u_n = \delta u_n + u_0$, where $u_0 \equiv r_0$. The linearization and approximate model are unstable, and so we picked a nominal feedback matrix $K = [-3.0741 \ 2.0957 \ 0.1195 \ -0.0090]$ that stabilizes the system by ensuring that the poles of the closed-loop system $x_{n+1} = (A + BK)x_n$ were placed at $\{0.75, 0.78, 0.98, 0.99\}$. These particular poles were chosen because they are close to the poles of the open-loop system, while still being stable.

We need the modeling error set \mathcal{W} , and this set \mathcal{W} was chosen to be a hypercube that encompasses both a bound on the linearization error, derived using the Taylor remainder theorem applied to the true nonlinear model, along with a small amount of subjectively-chosen “safety margin” to provide protection against the effect of numerical errors.

We compared the performance of linear MPC, nonlinear MPC, and LBMPC with L2NW for regulating the system about the operating point x_0 , by conducting a simulation starting

from initial condition $[\Phi_0 - 0.35 \ \Psi_0 - 0.40 \ r_0 \ 0]'$. The horizon was chosen to be $N = 100$, and we used the cost function

$$\psi_n = \|\tilde{x}_{n+N} - \Lambda\theta\|_P^2 + \|\bar{x}_s - \Lambda\theta\|_T^2 + \sum_{i=0}^{N-1} \|\tilde{x}_{n+i} - \Lambda\theta\|_Q^2 + \|\check{u}_{n+i} - \Psi\theta\|_R^2,$$

with $Q = \mathbb{I}_4$, $R = 1$, $T = 1e3$, and P that solves the discrete-time Lyapunov equation. (Note that the more complicated form of the cost function is due to the use of a reference governor concept that provides improved performance.) The L2NW used a bandwidth of $h = 0.5$, $\lambda = 1e-3$ and data measured as the system was controlled by LBMPC (i.e., there was no pre-training). Also, the L2NW only used three states $X_i = [\Phi_i \ \Psi_i \ u_i]$ to estimate $g(x, u)$; incorporation of such prior knowledge improves estimation by reducing dimensionality.

The significance of this setup is that the assumptions of the theorems about robust constraint satisfaction, feasibility, and robust asymptotic stability (RAS) are satisfied. This means that for both linear MPC and LBMPC: (a) constraints and feasibility are robustly maintained despite modeling errors, (b) closed-loop stability is ensured, and (c) control is input-to-state-stable (ISS) with respect to modeling error. In the instances we simulated, the controllers demonstrated these features.

Simulation results are shown in Fig. 1: LBMPC converges faster to the operating point than linear MPC, but requires increased computation at each step (0.3s for linear MPC vs. 0.9s for LBMPC). Interestingly, LBMPC performs as well as nonlinear MPC, but nonlinear MPC only requires 0.4s to compute each step. However, our point is that LBMPC does not require the control engineer to model nonlinearities, in contrast to nonlinear MPC. Our code was written in MATLAB and uses the SNOPT solver for optimization.

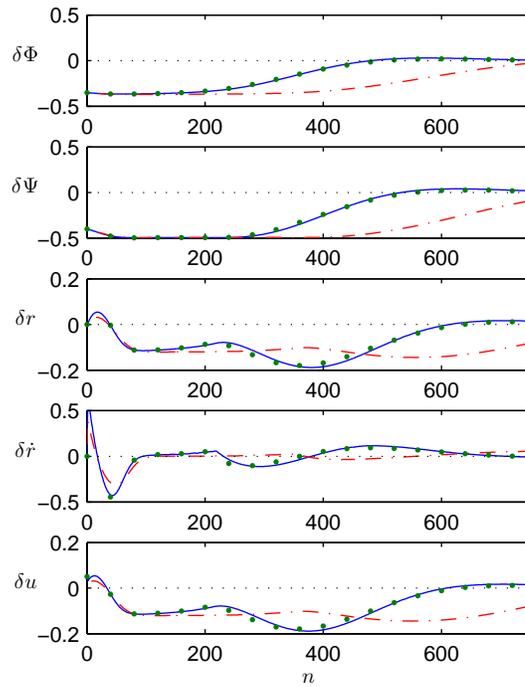


Figure 1: The states and control of LBMPC (solid blue), linear MPC (dashed red), and nonlinear MPC (dotted green) are shown. LBMPC converges faster than linear MPC.