IEOR 290A – Lecture 25
The Oracle

1 Naming

In theoretical computer science, oracles are black boxes that take in inputs and give answers. An important class of arguments known as relativizing proofs utilize oracles in order to prove results in complexity theory and computability theory. These proofs proceed by endowing the oracle with certain generic properties and then studying the resulting consequences.

We have named the functions $\mathcal{O}_n$ oracles in reference to those in computer science. Our reason is that we proved robustness and stability properties of LBMPC by only assuming generic properties, such as continuity or boundedness, on the function $\mathcal{O}_n$. These functions are arbitrary, which can include worst case behavior, for the purpose of the theorems in the previous section.

Whereas before, we considered the oracles as abstract objects, here we discuss and study specific forms that the oracle can take. In particular, we can design $\mathcal{O}_n$ to be a statistical tool that identifies better system models. This leads to two natural questions: First, what are examples of statistical methods that can be used to construct an oracle for LBMPC? Secondly, when does the control law of LBMPC converge to the control law of MPC that knows the true model? It will turn out that the second question is complex, and will be discussed in a later lecture.

2 Parametric Oracles

A *parametric oracle* is a continuous function $\mathcal{O}_n(x, u) = \chi(x, u; \lambda_n)$ that is parameterized by a set of coefficients $\lambda_n \in \mathcal{T} \subseteq \mathbb{R}^L$, where $\mathcal{T}$ is a set. This class of learning is often used in adaptive control. In the most general case, the function χ is nonlinear in all its arguments, and it is customary to use a least-squares cost function with input and trajectory data to estimate the parameters

$$\hat{\lambda}_n = \arg\min_{\lambda \in \mathcal{T}} \textstyle\sum_{j=0}^{n} (Y_j - \chi(x_j, u_j; \lambda))^2, \tag{1}$$

where $Y_i = x_{i+1} - (Ax_i + Bu_i)$. This can be difficult to compute in real-time because it is generally a nonlinear optimization problem.

Example: It is common in biochemical networks to have nonlinear terms in the dynamics such as

$$\mathcal{O}_n(x, u) = \lambda_{n,1} \left(\frac{x_1^{\lambda_{n,2}}}{x_1^{\lambda_{n,2}} + \lambda_{n,3}} \right) \left(\frac{\lambda_{n,4}}{u_1^{\lambda_{n,5}} + \lambda_{n,4}} \right), \tag{2}$$

where $\lambda_n \in \mathcal{T} \subset \mathbb{R}^5$ are the unknown coefficients in this example. Such terms are often called Hill equation type reactions.

2.1 Linear Oracles

An important subclass of parametric oracles are those that are linear in the coefficients: $\mathcal{O}_n(x,u) = \sum_{i=1}^{L} \lambda_{n,i}\chi_i(x,u)$, where $\chi_i \in \mathbb{R}^p$ for $i = 1, \dots, L$ are a set of (possibly nonlinear) functions. The reason for the importance of this subclass is that the least-squares procedure (1) is convex in this situation, even when the functions χ_i are nonlinear. This greatly simplifies the computation required to solve the least-squares problem (1) that gives the unknown coefficients λ_n.

Example: One special case of linear parametric oracles is when the χ_i are linear functions. Here, the oracle can be written as $\mathcal{O}_m(x,u) = F_{\lambda_m}x + G_{\lambda_m}u$, where $F_{\lambda_m}, G_{\lambda_m}$ are matrices whose entries are parameters. The intuition is that this oracle allows for corrections to the values in the A, B matrices of the nominal model; it was used in conjunction with LBMPC on a quadrotor helicopter testbed that will be discussed in later lectures, in which LBMPC enabled high-performance flight.

3 Nonparametric Oracles

Nonparametric regression refers to techniques that estimate a function $g(x,u)$ of input variables such as x, u, without making *a priori* assumptions about the mathematical form or structure of the function g. This class of techniques is interesting because it allows us to integrate non-traditional forms of adaptation and "learning" into LBMPC. And because LBMPC robustly maintains feasibility and constraint satisfaction as long as Ω can be computed, we can design or choose the nonparametric regression method without having to worry about stability properties. This is a specific instantiation of the separation between robustness and performance in LBMPC.

Example: Neural networks are a classic example of a nonparametric method that has been used in adaptive control, and they can also be used with LBMPC. There are many particular forms of neural networks, and one specific type is a feedforward neural network with a hidden layer of k_n neurons; it is given by

$$\mathcal{O}_n(x,u) = c_0 + \sum_{i=1}^{k_n} c_i\sigma(a_i'[x'\ u']' + b_i), \tag{3}$$

where $a_i \in \mathbb{R}^{p+m}$ and $b_i, c_0, c_i \in \mathbb{R}$ for all $i \in \{1, \dots, k\}$ are coefficients, and $\sigma(x) = 1/(1+e^{-x}) : \mathbb{R} \to [0,1]$ is a sigmoid function. Note that this is considered a nonparametric method because it does not generally converge unless $k_n \to \infty$ as $n \to \infty$.

Designing a nonparametric oracle for LBMPC is challenging because the tool should ideally be an estimator that is bounded to ensure robustness of LBMPC and differentiable to allow for its use with numerical optimization algorithms. Local linear estimators are not guaranteed to be bounded, and their extensions that remain bounded are generally non-differentiable. On the other hand, neural networks can be designed to remain bounded and differentiable, but they can have technical difficulties related to the estimation of its coefficients. In future lectures, we will discuss one specific type of nonparametric oracle that works well with LBMPC both theoretically and in simulations.