第四讲：最优控制的智能方法

最优控制介绍之四

张杰

人工智能学院
中国科学院大学

复杂系统管理与控制国家重点实验室
中国科学院自动化研究所

2017 年 9 月 21 日

关注：微信号“国科大最优控制”
课程微信群“国科大最优控制 2017”

飞跃
最优控制

国科大最优控制2017

该二维码7天内(9月15日前)有效，重新进入将更新

Table of Contents

Table of Contents

最优控制问题

问题 1 (最优控制问题)

1. 被控对象的状态方程为

$$\dot{x}(t) = f(x(t), u(t), t), \quad x(t_0) = x_0.$$

2. 容许控制, $u \in \mathcal{U}, \quad x \in \mathcal{X}$.
3. 目标集, $x(t_f) \in \mathcal{S}$

$$\mathcal{S} = [t_0, \infty) \times \{x(t_f) \in \mathbb{R}^n : m(x(t_f), t_f) = 0\}$$

4. 求分段连续的 u, 以最小化性能指标

$$J(u) = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t)\,\mathrm{d}t.$$

二次型性能指标

- Norbert Wiener 和 Albert C. Hall 在 20 世纪 40 年代就提出了设计闭环形式控制器以最小化跟踪误差的想法
- 20 世纪 60 年代初，Rudolf Kálmán 提出线性二次型最优控制问题

$$J(u) = \frac{1}{2} x^T(t_f) H x(t_f) + \frac{1}{2} \int_{t_0}^{t_f} \left[x^T(t) Q(t) x(t) + u^T(t) R(t) u(t) \right] \mathrm{d}t.$$

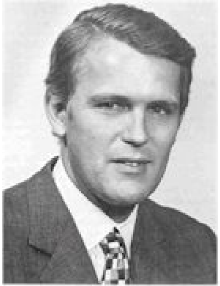


Figure: 匈牙利裔美国科学家 Rudolf Kálmán（卡尔曼）

课程内容

- 最优控制的数学理论
  - 经典变分法
  - 庞特里亚金极值原理
  - 动态规划方法
  - 微分博弈
- 最优控制的智能方法
  - 强化学习与自适应动态规划
  - 模型预测控制
  - 模糊控制
  - 平行控制与平行学习

无穷时间最优控制问题

问题 2 (无穷时间离散时间最优控制问题)

状态变量 $x(k):\mathbb{N}\to\mathbb{R}^n$，控制变量 $u(k):\mathbb{N}\to\mathbb{R}^m$

1. 被控对象的状态方程为

$$
\begin{aligned}
&x(k+1)=f_D(x(k),u(k)), \quad k=0,1,\ldots,\infty\\
&x(0)=x_0.
\end{aligned}
$$

2. 容许控制, $u\in\mathcal{U}$, $\quad x\in\mathcal{X}$.
3. 求最优控制 u，以最小化性能指标

$$
J(u;x_0)=\sum_{k=0}^{\infty} g_D(x(k),u(k)).
$$

一个例子

例 1

$$J(u) = \sum_{k=0}^{\infty}[x^2(k) + u^2(k)].$$

$$x(k+1) = x(k) + u(k), \quad k = 0, 1, 2, \ldots$$

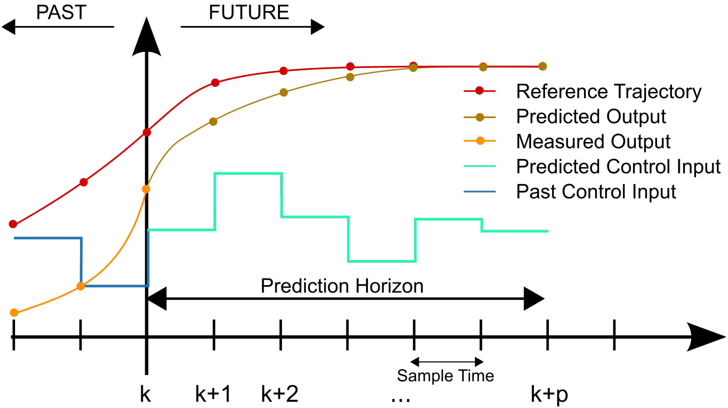
- 若存在最优控制能最小化 $J(u)$，则 $J(u)$ 必有限
- 令 $N \to \infty$，有 $x(k) \to 0, u(k) \to 0$

虽然没有设定明确的控制目标，状态却收敛于 0（渐进稳定）

Table of Contents

模型预测控制

- 预测模型【精确或不精确】
- 滚动优化【有限时间内开环最优控制】
- 反馈矫正【状态或模型】



一个最简单的例子

例 2

状态变量 $x(k):\mathbb{N}\to\mathbb{R}$，控制变量 $u(k):\mathbb{N}\to\mathbb{R}$。满足离散时间状态方程，

$$x(k+1)=x(k)+u(k),\quad x(0)=x_0.$$

最小化二次型性能指标：

$$J(u)=\sum_{k=0}^{N-1}[x^2(k)+u^2(k)]. \tag{1}$$

$N\to\infty$

若 $N=1,2$，容易计算！

简单分析

$$
\begin{aligned}
J(u) &= \sum_{k=0}^{\infty}[x^2(k) + u^2(k)] \\
&= \sum_{k=N}^{\infty}[x^2(k) + u^2(k)] + \sum_{k=0}^{N-1}[x^2(k) + u^2(k)].
\end{aligned}
$$

动态规划：最优控制则未来部分加和应为值函数 $V(x(N))$

//remark: 想想为何值函数没加时间?

预测模型

在 0 时刻，可观测状态 $x(0)$，选择【很大】的 N，取性能指标

$$\tilde{V}(x(N)) + \sum_{k=0}^{N-1}[x^2(k) + u^2(k)]$$

以状态方程

$$x(k+1) = x(k) + u(k), \quad k = 0, 1, 2, \ldots, N-1$$

求解该预测模型的最优控制，近似无穷时间最优控制问题的解

- 例如，可对于很大的 N，假定 $\tilde{V}(\cdot) = 0$【下例】
- 例如，可假定 $\tilde{V}(x(N)) = x^2(N)$【书 2.6.2】

预测模型与滚动优化

$N=2$，在任意时刻 $k=0,1,2,\ldots,$ 分别求解下列最优控制问题：

$$J(u)=\sum_{i=k}^{k+N-1}[x^2(i;k)+u^2(i;k)].$$

$$x(k;k)=x(k),$$

$$x(i+1;k)=x(i;k)+u(i;k),\quad i=k,k+1,\ldots,k+N-1,$$

在时刻 k，首先从环境或仿真中获取状态 $x(k)$。以 $x(i;k)$ 为状态变量，以 $u(i;k)$ 为控制变量，当 $N=2$ 时上述问题可以化简为：

$$\begin{aligned}&\min\{x^2(k;k)+u^2(k;k)+x^2(k+1;k)^2+u^2(k+1;k)\}\\=&\min\{x^2(k)+u^2(k;k)+[x(k)+u(k;k)]^2+u^2(k+1;k)\}\end{aligned}$$

最优解为 $u(k;k)=-x(k)/2$, $u(k+1;k)=0$.

反馈矫正

虽然根据预测模型求得的最优控制为

$$u(k;k) = -x(k)/2, \quad u(k+1;k) = 0.$$

仅实施第一个时段的控制变量，即 $u(k) = -x(k)/2$。
随后反馈矫正，构造预测模型，在本例状态方程不变，初值 $x(k+1;k+1) = x(k+1)$ 可由环境或仿真中观测。滚动优化可得

$$u(k+1) = -x(k+1)/2.$$

$N = 2$ 情况下，本例的模型预测控制课解得闭环形式控制策略，

$$\boxed{u(k) = -x(k)/2}$$

不同预测时段的模型预测控制

$N=2,3,5$ 时，分别得到：

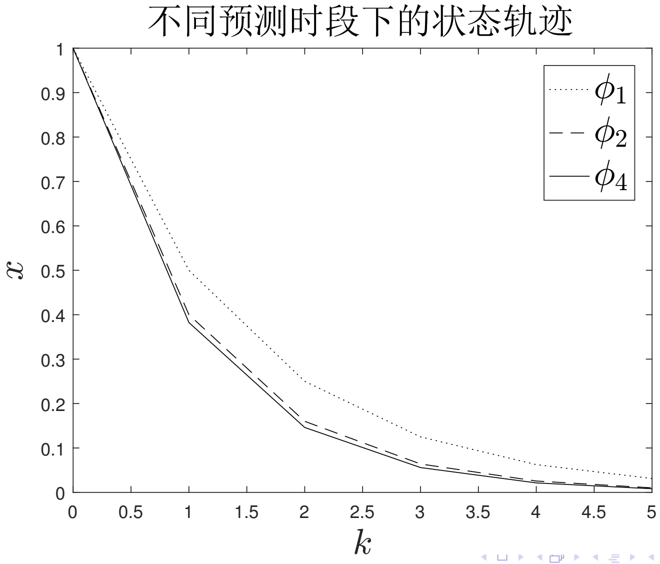
$$\begin{aligned} u(k) &= -0.5x(k) \\ u(k) &= -0.6x(k) \\ u(k) &= -0.618x(k). \end{aligned}$$

N 再增加，解得的控制策略收敛于 $-0.618x(k)$.

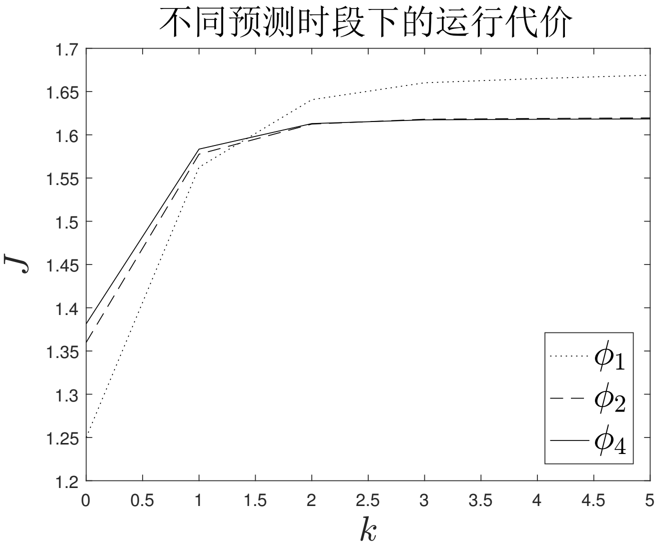
Remark 1

对于较大的 N，并不像 $N=2$ 一样容易计算，这将依赖于本课最优控制的各种方法技巧

不同预测时段下的状态轨迹



不同预测时段下的运行代价



模型预测控制

- 开环控制算法–> 闭环控制
- 有限时间近似无穷时间
- 模型可矫正
  - 利用回归、神经网络等，寻找与数据“最近似”的模型
  - 或，利用精确模型，仅用观测的状态矫正“预测”的状态

Table of Contents

神经元网络

受 Wiener 的影响，1943 年，Warren McCulloch 和 Walter Pitts 为人脑的神经网络构建了计算模型，提出将神经元具有的“皆有或皆无”（all-or-none）特性建模为阈值函数:

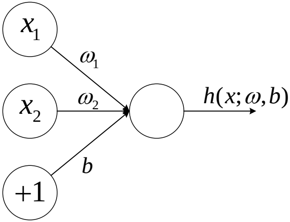
$$\phi(z)=\begin{cases}1, & z\geq\theta,\\ 0, & z<\theta.\end{cases}$$

最早的人工神经元，被称为 McCulloch-Pitts 模型（MCP 模型）

神经元网络

- 1958 年，Frank Rosenblatt 提出了感知器模型（perceptron）拓展至多元
- 1960 年，Bernard Widrow 在 ADALINE 模型（adaptive linear neuron）中将阈值移项，写成今天常见的偏移量形式

$$h(x;\omega,b)=\omega_1x_1+\omega_2x_2+b.$$



非线性的人工神经元

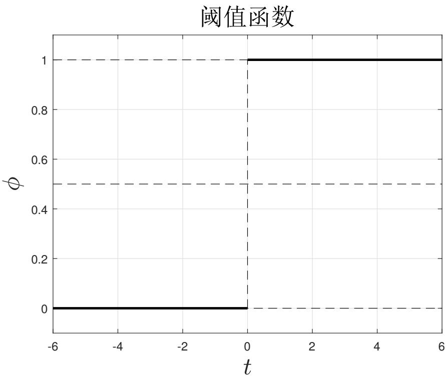
结合连续单调的激活函数，$\sigma : \mathbb{R} \to \mathbb{R}$，可得非线性的人工神经元

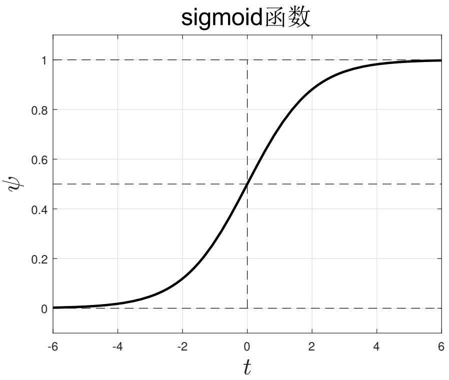
$$h(x; \omega, b) = \sigma(\omega_1 x_1 + \omega_2 x_2 + b). \tag{2}$$

例如，可令激活函数 σ 为 *sigmoid* 函数

$$\mathrm{sigmoid}(z) \stackrel{\mathrm{def}}{=} \frac{1}{1 + e^{-z}}. \tag{3}$$

阈值函数（左）与 sigmoid 函数（右）





多层神经元网络, MLP

神经网络就是人工神经元的联结。感知器叠加即得多层前馈神经网络（multi-layer feedforward perceptron），简称多层神经网络

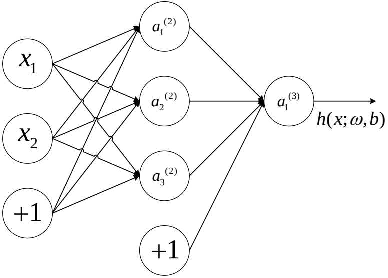


Figure: 单输出的多层神经网络

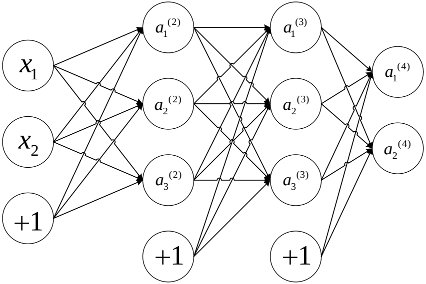


Figure: 多输出的多层神经网络

通用逼近器

定理 1 (MLP 是通用逼近器（universal approximator）)

有界区域 $\Omega \subset \mathbb{R}^n$，其上连续函数 $f(x) : \Omega \to \mathbb{R}^m$，给定任意的误差 $\epsilon > 0$，存在适当的隐层神经元数量，以及感知器参数，使得多层神经网络是函数 f 的近似：对任意的 $x \in \Omega$，

$$|h(x; \omega, b) - f(x)| < \epsilon.$$

- 神经网络用有限参数近似表示“无穷”定义域的函数
- 神经网络是参数化表示最优控制问题状态方程、值函数以及控制策略的优秀工具
- Werbos 用神经网络的反向传播算法计算神经网络的参数

最优性原理与策略迭代

根据最优性原理，有

$$V(x(k)) = \min_{u(k)}\{g_D(x(k), u(k)) + V(x(k+1))\}$$

1960 年，Ronald Howard 提出离散时间系统的策略迭代（policy iteration）：若已知一个容许控制策略 $u(k) = \phi_0(x(k))$，则对 $i = 0, 1, 2, \ldots,$ 先求解关于 V_i 的广义 *Bellman* 方程：

$$V_i(x(k)) = g_D(x(k), u(k)) + V_i(x(k+1)), \quad \forall x(k) \in X.$$

再解

$$\phi_{i+1}(x(k)) \leftarrow \operatorname*{argmin}_{u(k)} \Big\{ g_D(x(k), u(k)) + V_i(x(k+1)) \Big\}, \quad \forall x(k) \in X.$$

值迭代方法

策略迭代需要从一个容许控制策略出发，值迭代方法是一种十分便捷的近似，并不依赖于初始策略，令

$$V_0(x(k)) = 0, \forall x(k) \in X.$$

在每次迭代中 $i = 0, 1, \ldots$，首先计算迭代的近似最优控制律，$\forall x(k) \in X$，

$$\phi_i(x(k)) \leftarrow \underset{u(k)\in U}{\operatorname{argmin}} \left\{ g_D(x(k), u(k)) + V_i(f_D(x(k), u(k))) \right\}.$$

在此基础上，更新迭代的近似值函数：

$$V_{i+1}(x(k)) \leftarrow g(x(k), \phi_i(x(k))) + V_i(f_D(x(k), \phi_i(x(k)))).$$

$\boxed{V_0 \le V_1 \le \ldots}$

Remark 2

2016 年，魏庆来、刘德荣等给出其他初始值函数下的收敛证明

离散时间自适应动态规划

一般情况下，策略迭代方法难以解析求解。1977 年，Werbos 结合神经网络的反向传播算法，提出最早的自适应动态规划方法，用神经网路近似值函数和控制策略

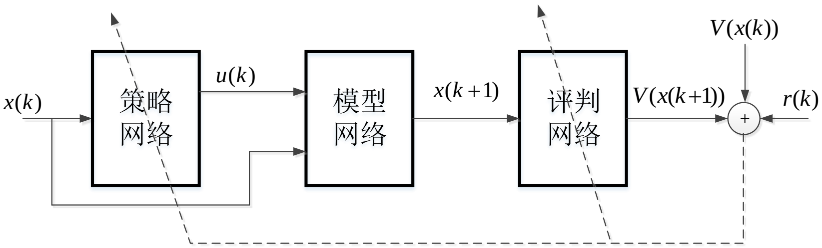


Figure: 启发式动态规划的三个模块

连续时间自适应动态规划

1979 年，George N. Saridis 提出利用逐次近似近似求解 HJB 方程，是 Howard 离散时间策略迭代的推广。对

$$J(u;x_0,t_0)=h(x(t_f))+\int_{t_0}^{t_f}\left[L(x(t))+\|u(t)\|^2\right]\mathrm{d}t,\quad x(t_0)=x_0.$$

$$\dot{x}(t)=A(x(t),t)+B(x(t),t)u(t),\quad t\in[t_0,t_f].$$

已知容许控制 $u(t)=\phi_0(x(t,t))$。$i=0,1,\ldots,$ 先解广义 *HJB* 方程：

$$\frac{\partial V_i}{\partial t}+\mathcal{H}(x,u,\frac{\partial V_i}{\partial x},t)=0,\quad t\in[t_0,t_f].$$

再迭代控制律：

$$\phi_{i+1}(x,t)\leftarrow-\frac{1}{2}B^{\mathrm{T}}(x,t)\frac{\partial V_i}{\partial x}(x,t),\quad i=1,2,\ldots$$

有 $V_0(x,t)\geq V_1(x,t)\geq\ldots.$

一个最简单的例子

例 3

状态变量 $x(k):\mathbb{N}\to\mathbb{R}$，控制变量 $u(k):\mathbb{N}\to\mathbb{R}$。满足离散时间状态方程，

$$x(k+1)=x(k)+u(k).$$

要将状态控制在原点附近并保持稳定。设计二次型性能指标:

$$J(u)=\sum_{k=0}^{\infty}[x^2(k)+u^2(k)]. \tag{4}$$

值迭代方法

在值迭代方法中，对任意的 $x \in \mathbb{R}$，令 $V_0(x) = 0$.
对任意 $x(k) \in \mathbb{R}$,

$$\phi_0(x(k)) \leftarrow \operatorname*{argmin}_{u(k)\in\mathbb{R}} \left\{ x^2(k) + u^2(k) + 0 \right\} = 0.$$

$$V_1(x(k)) \leftarrow \left\{ x^2(k) + \phi_0^2(x(k)) + 0 \right\} = x^2(k).$$

$$\phi_1(x(k)) \leftarrow \operatorname*{argmin}_{u(k)\in\mathbb{R}} \left\{ x^2(k) + u^2(k) + [x(k) + u(k)]^2 \right\} = -0.5x(k).$$

$$V_2(x(k)) \leftarrow \left\{ x^2(k) + \phi_1^2(x(k)) + [x(k) + \phi_1^2(x(k))] \right\} = 1.5x^2(k).$$

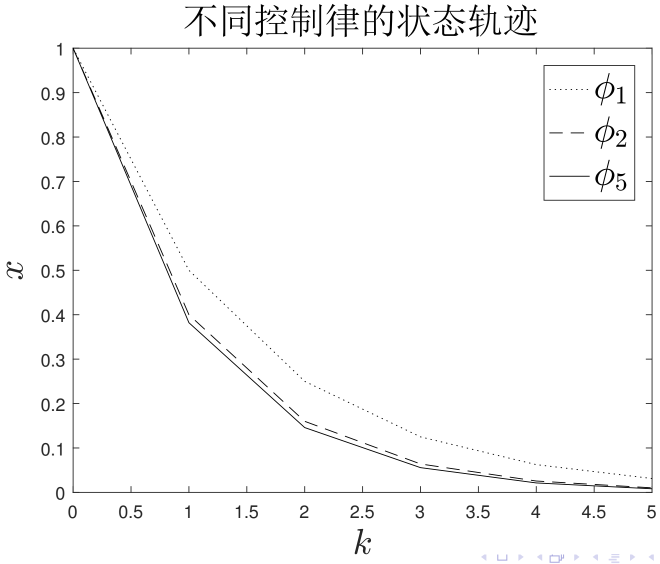
不同迭代次数的自适应动态规划值迭代方法

对 $i = 1, 2, 5$ 时，分别有

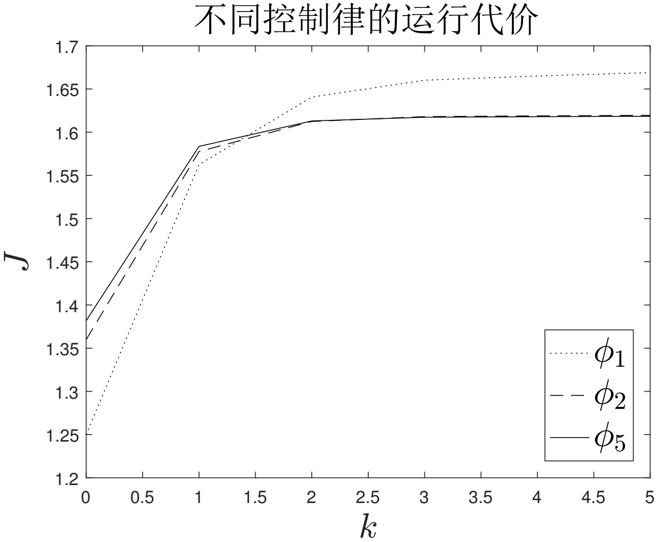
$$
\begin{aligned}
u(k) &= -\,0.5x(k) \\
u(k) &= -\,0.6x(k) \\
u(k) &= -\,0.618x(k).
\end{aligned}
$$

i 再增加，解得的控制策略收敛于 $-0.618x(k)$.

不同迭代次数控制律的状态轨迹



不同迭代次数控制律的运行代价



自适应动态规划

- 常用神经网络近似模型、值函数、控制策略
- 与 MPC 不同，N 增加可迭代利用
- 近似求解 Bellman/HJB 方程

Table of Contents

例子：导弹攻击固定目标的最优控制

例 4 (导弹攻击固定目标的最优控制)

- 初始时刻导弹三维坐标 $\mathbf{x}_0$，速度为 $\mathbf{v}_0$，状态方程

$$\dot{\mathbf{x}}(t) = \mathbf{v}(t),\ \mathbf{x}(t_0) = \mathbf{x}_0. \tag{5}$$

$$\dot{\mathbf{v}}(t) = \mathbf{u}(t),\ \mathbf{v}(t_0) = \mathbf{v}_0. \tag{6}$$

- 终止条件：t_f 时刻导弹击中目标的坐标 $\mathbf{x}_f$，速度 $\mathbf{v}_f$ 自由
- 最小化性能指标，例如最小能量

$$J(\mathbf{u}) = \int_{t_0}^{t_f} \frac{1}{2}\|\mathbf{u}\|^2 dt \tag{7}$$

例子：导弹攻击移动目标的最优控制

例 5 (导弹攻击移动目标的最优控制)

- 导弹 (M) 状态方程和目标 (T) 状态方程分别为 ($\mathbf{u}_T$ 已知)

$$\dot{\mathbf{x}}_M(t) = \mathbf{v}_M(t),\ \dot{\mathbf{v}}_M(t) = \mathbf{u}_M(t). \tag{8}$$

$$\dot{\mathbf{x}}_T(t) = \mathbf{v}_T(t),\ \dot{\mathbf{v}}_T(t) = \mathbf{u}_T(t). \tag{9}$$

- 终止条件：t_f 时刻导弹击中目标，速度 $\mathbf{v}_f$ 自由

$$\mathbf{x}_M(t_f) = \mathbf{x}_T(t_f). \tag{10}$$

- 最小化性能指标，例如能量

$$J(\mathbf{u}_M) = \int_{t_0}^{t_f} \frac{1}{2}\|\mathbf{u}_M\|^2 dt \tag{11}$$

引入“相对位置”“相对速度”

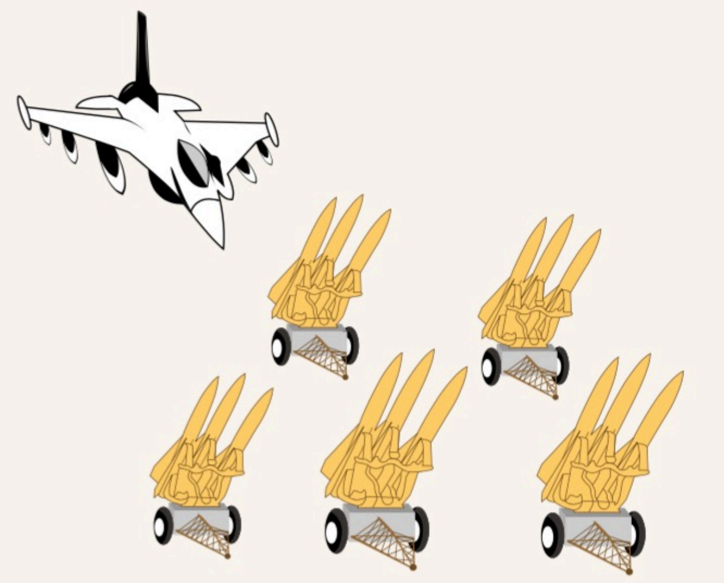
令 $\mathbf{x} := \mathbf{x}_M - \mathbf{x}_T$, $\mathbf{v} := \mathbf{v}_M - \mathbf{v}_T$. 状态方程变为

$$\dot{\mathbf{x}} = \mathbf{v}_M - \mathbf{v}_T = \mathbf{v}, \tag{12}$$

$$\dot{\mathbf{v}} = \mathbf{u}_M - \mathbf{u}_T. \tag{13}$$

终值条件 $\mathbf{x}(t_f) = 0$, $\mathbf{v}(t_f)$ free。性能指标不变
转化为和导弹攻击固定目标最优控制完全相同形式的问题，可使用极值原理或动态规划求解

被攻击的目标也使用最优控制躲避?



微分博弈的发展

- 1928 年 (On the Theory of Games of Strategy)，1944 年 (Theory of games and economic behavior) 两篇著作中，John Von Neumann 和 Osker Morgenstern 创立博弈论
- 1951 年起，Rand 公司在美国空军资助下，Rufus Issacs 研究对抗双方都能自由决策行动的追逃问题，形成了微分博弈的最初研究成果
- 60-70 年代，微分博弈理论逐渐完善，得到微分博弈值函数存在性等基础结果；1965 年，Issacs 整理出版了第一部微分博弈同名专著。也称动态博弈
- Saridis 称之为 “最坏情况设计”(1971 年)
- 2016 年，Google 公司的 AlphaGo 结合动态博弈和强化学习首次在围棋领域战胜人类世界冠军

从优化到博弈

定义 1 (函数极小值)

$\Omega \subset \mathbb{R}^N$ 是开集。称函数 $F \in C^1(\Omega)$ 在 x 达到局部极小值，若存在 $\epsilon > 0$ 使得:

$$F(x) \leq F(x'), \text{ if } \|x' - x\| < \epsilon, \forall x' \in \Omega.$$

定义 2 (纳什平衡 Nash Equilibrium, NE)

$F \in C^1(\Omega_1 \times \Omega_2)$，局中人 $i = 1, 2$ 的性能指标分别为 $F_i(x_1, x_2)$, $x_1 \in \Omega_1$,$x_2 \in \Omega_2$。x_1, x_2 是纳什平衡，若

$$F_1(x_1, x_2) \leq F_1(x_1', x_2), \forall x_1' \in \Omega_1, \tag{14}$$

$$F_2(x_1, x_2) \leq F_2(x_1, x_2'), \forall x_2' \in \Omega_2. \tag{15}$$

例子：囚徒困境

例 6 (囚徒困境)

		B	
		招供 (C)	不招供 (N)
	招供 (C)	6/6	1/8
A	不招供 (N)	8/1	2/2

A 和 B 两人均可采取行动 N-不招供，或 C-招供

$$F_1(C,C)=6,\ F_1(N,C)=8,\ F_1(C,N)=1,\ F_1(N,N)=2,$$
$$F_2(C,C)=6,\ F_2(N,C)=1,\ F_2(C,N)=8,\ F_2(N,N)=2.$$

$$\boxed{F_1(C,C)<F_1(N,C),\ F_2(C,C)<F_2(C,N).}$$

博弈和倒推

这个理论涉及的与其说是一种从博弈的开始来看为最好的对策，不如说是一种从博弈的结局看来为最好的对策。在博弈的最后一着中，如果有可能，一个博弈参与者总是力求走能获胜的一着，其次要走能得平局的一着。他的对手，在走他这一着的前面一着时，总是力求要取一种着法，使得他不能走这获胜或得平局的一着……依次倒着推下去，都是如此。

—— 维纳

《控制论：或关于在动物和机器中控制和通讯的科学》第二版

反应函数法求解博弈平衡

定义 3 (反应函数)

对于任意给定的 $x_2 \in \Omega$，映射 $R_1(x_2) = \text{argmin}_{x_1 \in \Omega} F_1(x_1, x_2)$ 称为局中人-1 的反应函数 (reaction function, or best response)

Remark 3 (反应函数法求解纳什平衡)

若 $x_1 = R_1(x_2), x_2 = R_2(x_1)$，可知 x_1, x_2 为纳什平衡。可通过联立博弈双方的反应函数求解博弈的纳什平衡；此方法也可处理其他博弈平衡（如斯坦伯格平衡等）

古诺博弈: 反应函数法求解纳什平衡

例 7 (古诺寡头竞争模型, Cournot Model)

两家公司 $i=1,2$ 生产同类产品，生产数量为 $q_i \geq 0$，生产成本为 $c(q_i)=cq_i$，市场上产品单价 $p(q)=a-q$ 与市场上的产品总量 $q=q_1+q_2$ 有关。

两家公司都希望最大化各自的净利润

$$V_1(q_1,q_2)=p(q_1+q_2)q_1-c(q_1), \tag{16}$$

$$V_2(q_1,q_2)=p(q_1+q_2)q_2-c(q_2). \tag{17}$$

求 Best-response

固定公司 2 产量 q_2，公司 1 产量 q_1 应满足一阶条件

$$0 = \frac{\partial V_1}{\partial q_1} = \dot{p}(q_1 + q_2)q_1 + p(q_1 + q_2) - c,$$

$$R_1(q_2) = \frac{a - q_2 - c}{2}.$$

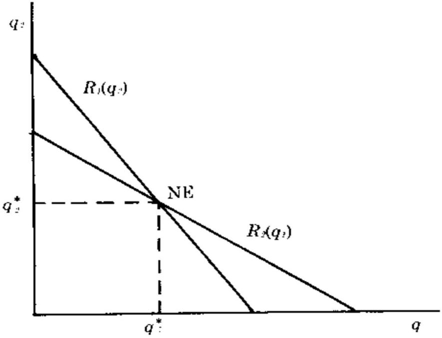
类似的，固定公司 1 产量，可得公司 2 的反应函数

$$R_2(q_1) = \frac{a - q_1 - c}{2}.$$

根据 Best-response 求得 NE

联立两个公司的反应函数得到古诺模型的纳什均衡

$$q_1^* = \frac{a-c}{3},\ q_2^* = \frac{a-c}{3}. \tag{18}$$



最优控制问题

问题 (典型最优控制问题)

1. 被控对象的状态方程为

$$\dot{x}(t) = f(x(t), u(t), t), \; x(t_0) = x_0.$$

2. 容许控制, $u \in U$
3. 目标集, $x(t_f) \in S$
4. 最小化性能指标

$$J(u) = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t) dt.$$

从最优控制到微分博弈

问题 3 (微分博弈问题)

1. 博弈双方的状态方程为

$$\dot{x}(t) = f(x(t), u_1(t), u_2(t), t),\ x(t_0) = x_0. \tag{19}$$

2. 容许控制 $u_1 \in U_1$, $u_2 \in U_2$
3. 二者均最小化各自的性能指标

$$J_1(u_1, u_2) = h_1(x(t_f), t_f) + \int_{t_0}^{t_f} g_1(x(t), u_1(t), u_2(t), t)dt \tag{20}$$

$$J_2(u_1, u_2) = h_2(x(t_f), t_f) + \int_{t_0}^{t_f} g_2(x(t), u_1(t), u_2(t), t)dt. \tag{21}$$

两车追逃博弈的例子

延续上节停车的例子。位置 x_1，速度 x_2，加速度 u。状态方程为：

$$\dot{x}_1(t) = x_2(t), \tag{22}$$

$$\dot{x}_2(t) = u(t). \tag{23}$$

$x(t_0) = x_0$ 在规定的 t_f 到达 $x(t_f) = x_f$，最小化控制能量：

$$J(u) = \int_{t_0}^{t_f} \frac{1}{2} u^2(t)\,\mathrm{d}t. \tag{24}$$

$t_0 = 0$, $t_f = 2$, $x_0 = [-2, 1]^{\mathrm{T}}$, $x_f = [0, 0]^{\mathrm{T}}$
在动态规划例子中，我们引入惩罚函数，消除终值约束，最小化：

$$J(u) = \frac{b}{2}\|x(t_f) - x_f\|_2^2 + \int_{t_0}^{t_f} \frac{1}{2} u^2(t) dt \tag{25}$$

两车追逃博弈的例子

例 8 (两车追逃博弈的例子)

延续上例场景，追逐者依然最小化其性能指标

$$J_1 = +\frac{b_1}{2}|x_1^{(1)}(t_f) - x_1^{(2)}(t_f)|^2 + \frac{1}{2}\int_{t_0}^{t_f}[u^{(1)}(t)]^2 \mathrm{d}t. \tag{26}$$

在固定的 t_f 时刻，尽量接近逃跑者，且兼顾能量消耗。逃跑者则希望 t_f 时刻距离越远越好，最小化性能指标

$$J_2 = -\frac{b_2}{2}|x_1^{(1)}(t_f) - x_1^{(2)}(t_f)|^2 + \frac{1}{2}\int_{t_0}^{t_f}[u^{(2)}(t)]^2 \mathrm{d}t. \tag{27}$$

$0 < b_1 < b_2$

两车追逃博弈的例子

为了计算方便，写成

$$J_1(u^{(1)}, u^{(2)}) = +\frac{b}{2}|x_1^{(1)}(t_f) - x_1^{(2)}(t_f)|^2 + \frac{1}{2}\int_{t_0}^{t_f} \frac{[u^{(1)}(t)]^2}{E_1}\mathrm{d}t, \quad (28)$$

$$J_2(u^{(1)}, u^{(2)}) = -\frac{b}{2}|x_1^{(1)}(t_f) - x_1^{(2)}(t_f)|^2 + \frac{1}{2}\int_{t_0}^{t_f} \frac{[u^{(2)}(t)]^2}{E_2}\mathrm{d}t. \quad (29)$$

其中 $E_1 > E_2 > 0$, $b_1 = E_1 b$, $b_2 = E_2 b$，分别为追逐者和逃跑者对双方距离对比控制能量的重视权重。

两车追逃博弈的例子

利用 PMP 或 DP 都可解得，

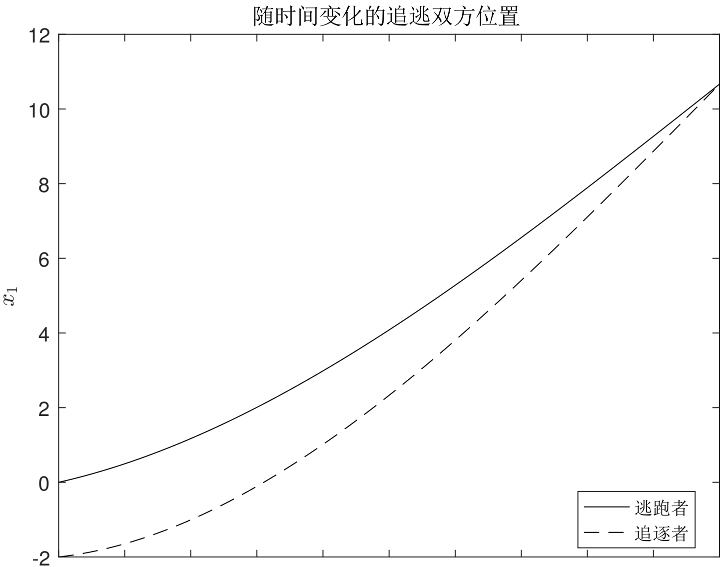
$$u^{(1)}(t) = -\frac{E_1(t_f - t)\left[x_1(t) + x_2(t)(t_f - t)\right]}{1/b + (E_1 - E_2)(t_f - t)^3/3}, \tag{30}$$

$$u^{(2)}(t) = \frac{E_2}{E_1} u^{(1)}(t). \tag{31}$$

这种情况下，终端时刻相对位置为

$$x_1(t_f) = \frac{x_0 + v_0(t_f - t_0)}{1 + b(E_1 - E_2)(t_f - t_0)^3/3}. \tag{32}$$

$E_1 = 0.8, E_2 = 0.5, b = 1000$，命中



小结

- 模型预测控制
  - 预测模型，滚动优化，反馈矫正
  - 利用反馈矫正将开环控制化为闭环形式
  - 有限时域的最优控制问题近似无穷时域
  - 预测模型往往“近似实际系统”
- 自适应动态规划
  - 策略迭代与逐次近似为基础
  - 神经网络等近似模型、值函数、控制策略
  - 迭代结果可利用
  - 近似求解 Bellman/HJB 方程
- 微分博弈
  - 最优控制 + 博弈 = 微分博弈
  - “最坏情况设计”，与 MPC“相反”
- 下节课：第二部分之一，变分法

关注：微信号“国科大最优控制”
课程微信群“国科大最优控制 2017”