Apollo模型参考自适应控制MRAC (二)–MRAC例题

下面的例子是我硕士自适应控制课的三个关于Model Reference Adaptive Control的练习，我硕士导师上的这门课，我觉得对于理解自适应控制的原理非常有用，大家可以先看一下下述的几个例子，理解一下怎么设计MRAC控制器，Apollo的MRAC包括一阶系统和二阶系统，下面的几个例子都有涉及，并且有关于一阶和二阶系统微分方程和状态空间方法是怎么转换的
Adaptive Control

Tutorial 1

Adaptive Controller Derivation using Lyapunov Theory
第一个例子是针对一个一阶系统的，假设一阶系统的参考模型为

$$\dot{x}_p = a_p x_p + b_p \lambda u \qquad (1)$$

其中a_p, b_p是已知变量，$\lambda \neq 0$, 但是我们不知道其符号.
a) 通过Lyapunov函数的方法设计一个自适应控制器，使得上述一阶系统可以和参考模型的表现一致，参考模型的方程为:

$$\dot{x}_m = a_m x_m + b_m r \qquad (2)$$

其中$a_m < 0$, r是参考输入
b) 扩展之前开发的控制器，使得可以使用另外两个增益γ_x和γ_r来调整参数向量$\theta_x(t)$和$\theta_r(t)$的收敛率(rate of convergence)

如果实际系统P的特性可以完全跟随参考模型系统S，那就意味着当当前状态相同的时候$x_m = x_p$，我们希望通过选择$\dot{x}_m = \dot{x}_p$，结合(1)和(2)
我们可以得出

$$\begin{aligned} u &= \frac{a_m - a_p}{b_p \lambda} x_p + \frac{b_m}{b_p \lambda} r \qquad (3) \\ &= \theta_x^* x_p + \theta_r^* r \qquad (4) \end{aligned}$$

当我们对系统P使用上述输入u的时候，

$$\begin{aligned} \dot{x}_p &= a_p x_p + b_p \lambda \left(\frac{a_m - a_p}{b_p \lambda} x_p + \frac{b_m}{b_p \lambda} r\right) \qquad (5) \\ &= a_m x_p + b_m r \qquad (6) \end{aligned}$$

因为λ未知，所以θ_x^*和θ_r^*也是未知的，这意味着我们需要做初始值估计然后根据自适应变化率去修改，假设我们使用的自适应输入为

$$u = \hat{\theta}_x x_p + \hat{\theta}_r r \qquad (7)$$

同样的我们将上述输入带入系统的状态方程:

$$\begin{aligned} \dot{x}_p = a_p x_p + b_p\lambda((\hat{\theta}_x - \theta_x^* + \theta_x^*)x_p + (\hat{\theta}_r - \theta_r^* + \theta_r^*)r) \\ = a_p x_p + b_p\lambda\frac{a_m - a_p}{b_p\lambda}x_p + b_p\lambda\frac{b_m}{b_p\lambda}r + b_p(\tilde{\theta}_x x_p + \tilde{\theta}_r r) \\ = a_m x_p + b_m r + b_p\lambda(\tilde{\theta}_x x_p + \tilde{\theta}_r r) \end{aligned}$$

其中$\tilde{\theta}_x = \hat{\theta}_x - \theta_x^*$, $\tilde{\theta}_r = \hat{\theta}_r - \theta_r^*$,
我们定义误差状态量$e = x_p - x_m$, 我们可以得到error dynamics的表达式为

$$\dot{e} = a_m e + b_p\lambda(\tilde{\theta}_x x_p + \tilde{\theta}_r r) \qquad (8)$$

为了建立参数调整机制, 我们需要找到合理的Lyapunov函数
我们首先选取Lyapunov函数为如下形式:

$$V(e, \tilde{\theta}_x, \tilde{\theta}_r) = e^2 + b_p|\lambda|\tilde{\theta}_x^2 + b_p|\lambda|\tilde{\theta}_r^2$$

该函数是正定的, $V(e, \tilde{\theta}_x, \tilde{\theta}_r) > 0$
为了保证该Lyapunov函数是负半定的, 我们首先对上述V求导

$$\begin{aligned} \dot{V}(e, \tilde{\theta}_x, \tilde{\theta}_r) = 2e\dot{e} + 2b_p|\lambda|(\tilde{\theta}_x\dot{\tilde{\theta}}_x + \tilde{\theta}_r\dot{\tilde{\theta}}_r) \\ 2a_m e^2 + 2\tilde{\theta}_x(b_p\lambda e x_p + b_p|\lambda|\dot{\tilde{\theta}}_x) + 2\tilde{\theta}_r(b_p\lambda r + b_p|\lambda|\dot{\tilde{\theta}}_r) \\ = 2a_m e^2 \end{aligned}$$

我们选取如下参数

$$\begin{aligned} \dot{\tilde{\theta}}_x = -sgn(\lambda)e x_p \\ \dot{\tilde{\theta}}_r = -sgn(\lambda)e r \end{aligned}$$

此时$\dot{V}$是负半定的, $\dot{V} \leq 0$.

b) 为了能够调整参考变化的变化率, 我们新引入两个变量γ_x和γ_r, 我们重新定义一个正定的Lyapunov函数为以下形式:

$$V = e^2 + \frac{b_p|\lambda|}{\gamma_x}\tilde{\theta}_x^2 + \frac{b_p|\lambda|}{\gamma_r}\tilde{\theta}_r^2$$

对上述Lyapunov函数求导数

$$\begin{aligned} \dot{V} = 2a_m e^2 + 2\tilde{\theta}_x(b_p\lambda e x_p + \frac{b_p|\lambda|}{\gamma_x}\dot{\tilde{\theta}}_x) + 2\tilde{\theta}_r(b_p\lambda e r + \frac{b_p|\lambda|}{\gamma_r}\dot{\tilde{\theta}}_r) = \\ 2a_m e^2 \end{aligned}$$

为了使$\dot{V}$负半定, 我们选取下述参数变化率

$$\begin{aligned} \dot{\tilde{\theta}}_x = -\gamma_x sgn(\lambda)e x_p \\ \dot{\tilde{\theta}}_r = -\gamma_r sgn(\lambda)e r \end{aligned}$$

Tutorial 2

Lyapunov for State Space Systems
考虑非线性系统模型

$$
\begin{aligned}
\dot{x}1 &= x2 \\
\dot{x}2 &= ax_1^2 + bx2 + u \\
y &= x1
\end{aligned}
$$

其中状态变量$x1, x2 \in R$, 输入$u \in R$, 输出$y \in R$, 参数$a, b \in R$
a) 假设参数a和b已经, 设计状态反馈控制器$u(x_1, x_2)$使得上述系统的表现由下述参考模型决定:

$$
\begin{aligned}
y &= G_m(s)u_{ref} \\
G_m(s) &= \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}
\end{aligned}
$$

b) 基于前述问题, 我们设计自适应控制率, a和b是未知的并且证明系统是稳定的

a) 参考系统是一个标准的二阶系统, 我们定义两个参考模型状态变量x_{m1}和x_{m2}, 他们满足下述关系s:

$$
\begin{aligned}
\dot{x}_{m1} &= x_{m2} \\
\dot{x}_{m2} &= -\omega^2 x_{m1} - 2\zeta\omega x_{m2} + \omega^2 u_{ref}
\end{aligned}
$$

输入设置为

$$
u = -ax_1^2 - bx_2 - \omega^2 x_1 - 2\zeta\omega x_2 + \omega^2 u_{ref}
$$

当对系统采取上述输入的时候

$$
\dot{x} = \begin{bmatrix} \dot{x}1 \\ \dot{x}2 \end{bmatrix} = \begin{bmatrix} x2 \\ ax_1^2 + bx_2 + (-ax_1^2 - bx_2 - \omega^2 x_1 - 2\zeta\omega x_2 + \omega^2 u_{ref}) \end{bmatrix} = \begin{bmatrix} x2 \\ -\omega^2 x_1 - 2\zeta\omega x_2 + \omega^2 u_{ref} \end{bmatrix}
$$

如果参数a和b未知, 我们需要使用他们的估计数值$\hat{a}$和$\hat{b}$替代, 因此我们的控制器为

$$
u = -\hat{a}x_1^2 - \hat{b}x_2 - \omega^2 x_1 - 2\zeta\omega x_2 + \omega^2 u_{ref}
$$

把(1)带入系统模型

$$
\dot{x} = \begin{bmatrix} x2 \\ \tilde{a}x_1^2 - \tilde{b}x_2 - \omega^2 x_1 - 2\zeta\omega x_2 + \omega^2 u_{ref} \end{bmatrix}
$$

其中$\tilde{a} = a - \hat{a}$, $\tilde{b} = b - \hat{b}$. 定义误差状态变量为$e = [e_1 \ e_2]^T = x - x_m$,
我们得到如下的误差等式

$$
\dot{e} = \begin{bmatrix} e_2 \\ \tilde{a}x_1^2 + \tilde{b}x_2 - \omega^2 e_1 - 2\zeta\omega e_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} e + \begin{bmatrix} 0 \\ \tilde{a}x_1^2 + \tilde{b}x_2 \end{bmatrix}
$$

我们令$A_m = \begin{bmatrix} 0 & -\omega^2 \\ 1 & -2\zeta\omega \end{bmatrix}$, $\theta = \begin{bmatrix} \tilde{a} \\ \tilde{b} \end{bmatrix}$
我们选取Lyapunov函数$V(e, \tilde{\theta}) = e^T Pe + \tilde{\theta}^T \tilde{\theta}$, 其中$P$是Lyapunov方程正半定的对称矩阵

$$\dot{V}(e,\tilde{\theta}) = e^T P\dot{e} + \dot{e}^T Pe + 2\tilde{\theta}^T\dot{\tilde{\theta}}$$
$$= e^T P(\begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta w \end{bmatrix} e + \begin{bmatrix} 0 \\ \tilde{a}x_1^2 + \tilde{b}x_2 \end{bmatrix})$$
$$+ (e^T \begin{bmatrix} 0 & -\omega^2 \\ 1 & -2\zeta\omega \end{bmatrix} + \begin{bmatrix} 0 & \tilde{a}x_1^2 + \tilde{b}x_2 \end{bmatrix})Pe + 2\tilde{\theta}^T\dot{\tilde{\theta}}$$

$$\dot{V}(e,\tilde{\theta}) = e^T Qe + 2\tilde{\theta}^T \begin{bmatrix} x_1^2 \\ x_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} Pe + 2\tilde{\theta}^T(-\begin{bmatrix} x_1^2 \\ x_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} Pe) = -e^T Qe \le 0$$

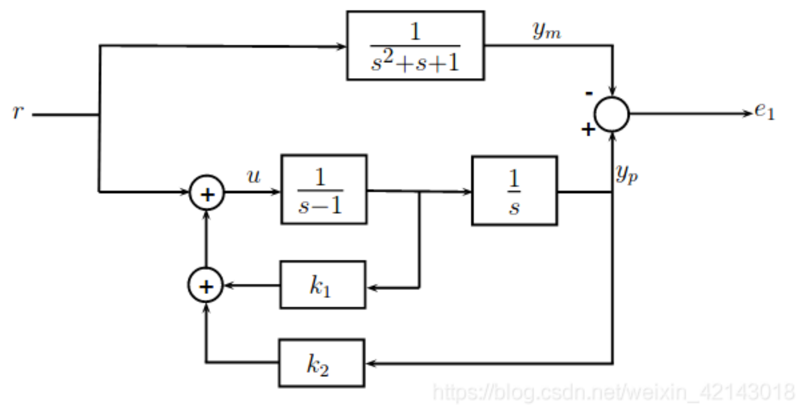
其中Q是正定矩阵, $PA_m + A_m^T P = -Q$, 参数变化率为

$$\dot{\tilde{\theta}} = -\begin{bmatrix} x_1^2 \\ x_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} Pe$$

因为$V(e,\tilde{\theta}) > 0$, 并且$\dot{V}(e,\tilde{\theta}) \le 0$, 系统是稳定的

Tutorial 3

Choice of Adaptive Control Law
系统框图如下所示, 我们需要设计参数k_1和k_2的变化率使得输出e_1最终变化为0



a) 根据上述框图画求出参考模型和系统模型的微分方程
b) 把上述微分方程转化为状态空间方程, 求出参考模型系统矩阵A_m, Plant的系统矩阵A_p和向量b
c) 求出矩阵P的每个元素, 从而使得$A_m^T + PA_m = -Q, Q = I$
d) 选择合适的Lyapunov函数和矩阵P, 并且求出参数 k_1和k_2的自适应变化率使得误差e_1可以渐渐变为0.

上述问题和Apollo中系统的模型是有很大的相似之处的, 参考系统和实际系统都是二阶系统, 并且第二个状态是第一个状态的变化率, 这一点和Apollo的MRAC很相像
a) 参考系统的微分方程为

$$\ddot{y}_m + \dot{y}_m + y_m = r$$

实际系统的微分方程为

$$\ddot{y}_p + (-1 - k_1)\dot{y}_p + (-k_2)y_p = r$$

控制器的微分方程为$u = r + k_1\dot{y}_p + k_2 y_p$

b) 系统的状态方程可以直接从微分方程中得出
参考模型:

$$\dot{x}_m = A_m x_m + br$$

其中模型的状态量:$x_m = \begin{bmatrix} y_m \\ \dot{y}_m \end{bmatrix}$, 参考模型的系统矩阵 $A_m = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
为了获取系统的状态空间表达式, 我们做下述变换:
$\ddot{y}_p + (-1 - k_1)\dot{y}_p + (-k_2)y_p = r, u = r + k_1\dot{y}_p + k_2 y_p$
$\ddot{y}_p + (-1 - k_1)\dot{y}_p + (-k_2)y_p = u - k_1\dot{y}_p - k_2 y_p$
$\ddot{y}_p = u - k_1\dot{y}_p - k_2 y_p + (1 + k_1)\dot{y}_p + k_2 y_p$
$\ddot{y}_p = \dot{y}_p + u$
系统P的状态空间表达式为

$$\dot{x}_p = A_p x_p + bu$$

其中$x_p = \begin{bmatrix} y_p \\ \dot{y}_p \end{bmatrix}$, $A_p = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
控制器的形式为$u = r + \theta^T x_p$, $\theta^T = \begin{bmatrix} k_2 & k_1 \end{bmatrix}$
c) 我们需要验证A_m是稳定的
$|\lambda I - A_m| = \lambda^2 + \lambda + 1 = 0$
可以推导出$\lambda_{1,2} = \dfrac{-1 \pm \sqrt{1-4}}{2}$
$Re(\lambda_{1,2}) = -0.5$, 由此我们可以看出矩阵A_m是渐进稳定矩阵， 两个实数特征值都位于左半平面
我们选取$Q = I$, 然后去求解下述Lyapunov方程

$$\begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} p1 & p2 \\ p2 & p3 \end{bmatrix} \begin{bmatrix} p1 & p2 \\ p2 & p3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} = - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

解上述方程，我们可以得到$p_2 = 0.5, p_3 = 1, p_1 = 1.5$.
d) 我们将u的表达式插入系统方程中, 可以得到

$$\dot{x}_p = A_p x_p + b(r + \theta^T x_p)\dot{x}_p = (A_p + b\theta^T)x_p + br$$

根据匹配条件(matching condition) $A_m = A_p + b\theta^{*T}$, 我们可以得出理想参数值

$$\theta^{*T} = [-1 \ \ -2]$$

因为参数是估计得出的, 假设估计参数为$\hat{\theta}$, 则系统方程为

$$\dot{x}_p = (A_p + b\hat{\theta}^T)x_p + br$$
$$\dot{x}_p = (A_p + b\theta^T)x_p + b\theta^{*T}x_p - b\theta^{*T}x_p + br$$

假设$\tilde{\theta}$为参数误差$\tilde{\theta} = \hat{\theta} - \theta^* = [\tilde{k}_2 \ \tilde{k}_1]^T$

$$\dot{e} = \dot{x}_p - \dot{x}_m$$
$$\dot{e} = A_m x_p + b\tilde{\theta}^T x_p + br - A_m x_m - br$$
$$\dot{e} = A_m e + b\tilde{\theta}^T x_p$$

我们选取Lyapunov函数: $V(e, \tilde{\theta}) = e^T P e + \tilde{\theta}^T \tilde{\theta}$
求导数:

$$\dot{V} = e^T P(A_m e + b\tilde{\theta}^T x_p) + (x_p^T \tilde{\theta} b^T + e^T A_m^T) Pe + 2\tilde{\theta}^T \dot{\tilde{\theta}}$$

$e^T Pb\tilde{\theta}^T x_p = x_p^T \tilde{\theta} b^T Pe = \tilde{\theta}^T x_p e^T Pb$, 选取自适应变化率为$\dot{\tilde{\theta}} = -x_p e^T Pb$, 可以得到：

$$\begin{aligned} \dot{V} &= e^T (PA_m + A_m^T P)e + 2\tilde{\theta}^T (x_p e^T Pb + \dot{\tilde{\theta}}) \\ &= -e^T Qq \leq 0 \end{aligned}$$

最终我们的自适应变化率为

$$\dot{\tilde{\theta}} = \begin{bmatrix} \dot{k}_2 \\ \dot{k}_1 \end{bmatrix} = -x_p e^T Pb = -\begin{bmatrix} y_p \\ \dot{y}_p \end{bmatrix} \begin{bmatrix} e_1 & e_2 \end{bmatrix} \begin{bmatrix} \frac{3}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{aligned} \dot{k}_2 &= -(\frac{e_1 + 2e_2}{2}) y_p \\ \dot{k}_1 &= -(\frac{e_1 + 2e_2}{2}) \dot{y}_p \end{aligned}$$

评论 6

这个不太懂

你好，Tutorial 1中自适应的变化率中λ的符号是怎么确定的？Tutorial 3中k1,k2最优值已经求得，为什么还要求自适应变化率？

回复 1年前 ···

Watchcloselyworld

你好，文中的：我们选取Q=IQ=I, 然后去求解下述Lyapunov方程 与题干（c）中的求出矩阵PP的每个元素, 从而使得ATm+PAm=−QAmT+PAm=−Q, Q=IQ=I 是怎么联系在一起的？可以直接以（c）中的条件： Tm+PAm=−QAmT+PAm=−Q, Q=IQ=I 来列等式计算P吗

回复 1年前 ···

weixin_43427770

你好，有没有MATLAB的仿真程序，谢谢

回复 1年前 ···

...同步电机无速度传感器控制(三)——模型参考自适应法... 10-30
本篇对另外一种无速度传感器控制策略做详解——模型参考自适应MRAS无速度控制策略。 1

L1自适应控制-理论基础 enhaibulei的博客 1万+
L1自适应背景 L1自适应控制算法是一种快速鲁棒的自适应控制。该算法实际上是模型参考自

MIT自适应律MRAC的理解和MATLAB实现 凯哥大数据——刘凯的博客 5383
文章目录什么是MIT自适应律？基于MIT律的可调增益MRAC计算过程MATLAB仿真完整MAT

如何理解Apollo模型参考自适应控制MRAC zmhzmhzm的博客 1739
最近有了解到百度无人驾驶Apollo项目中的横向控制有用到MRAC(Model Reference Adaptiv

机器人动力学与控制学习笔记（十）————自适应控制（模型 冰与火的青春的博客 6880
一、模型参考自适应 模型参考自适应是比较流行的自适应控制方式之一。模型参考自适应控

模型参考自适应控制 luckyoooooooo的博客 2547
模型参考自适应控制 考虑一阶线性模型，通过模型参考自适应的控制方法进行参考值的跟随

Apollo模型参考自适应控制MRAC (三)-MRAC代码分析 weixin_42143018的博客 4711
Apollo横向和纵向控制器都属于high level controller, 我们在设计控制器的过程中一直都是假

自适应控制---模型参考自适应控制（一）基于局部参数 热门推荐 ksjz123的博客 1万+
模型参考自适应---基于局部参数最优化的设计方法（MIT方案）

模型参考自适应MATLAB仿真程序 04-12
模型参考自适应控制的MATLAB 实例程序，可供新手参考学习。

模型参考自适应控制simulink仿真 11-06
模型参考自适应控制simulink仿真，写论文的时候自己搭的很基础的仿真。我参考的书是20

论文研究-基于PID和MRAC的电动汽车电机控制设计 .pdf 08-16
基于PID和MRAC的电动汽车电机控制设计，庞洋， 吴友宇， 在能源危机和环境污染日趋严

模型参考自适应仿真 11-10
基于simulink的模型参考自适应仿真图，可以用它来进行自适应概念的理解

模型参考自适应控制器(MRAC)系列: 1.基于Lyapunov稳定判据的自 橘子的控制小屋 1302
鲁棒模型参考自适应控制_1 引言 在设计一个控制系统时,其闭环稳定性,指令跟踪能力以及应

深度解析Apollo无人车感知和定位技术 黎国溥 3096
原文地址：http://www.realli.net/archives/26639 有关无人车的定位有两种，一种称之为绝对

Apollo 功能及源码讲解分析 yine的专栏 489
详见：http://www.iocoder.cn/categories/Apollo/

apollo5.5感知模块改进 qq_17364791的博客 538
apollo5.5感知模块整理 最近一直在研究百度的apollo源码感知部分，下面整理一下近期的一

模型参考自适应系统（一）： MRAS问题 仗剑走天涯的博客 8978
参考文献： 自适应控制，K.J.奥斯特隆姆,B.威顿马克.李清泉等译.1992，科学出版社. 4.1引

12	44万+	114万+	2万+	
原创	周排名	总排名	访问	等级

507	150	55	51	318
积分	粉丝	获赞	评论	收藏

私信 已关注

搜博主文章

热门文章

Apollo MPC OSQP Solver 7681

Apollo模型参考自适应控制MRAC(三)-MRAC代码分析 4704

Apollo模型参考自适应控制MRAC(二)-MRAC例题 3434

模型预测控制Paolo Falcone 博士面试 (二) - MPC控制的稳定性 3034

Hybrid A* 算法基本流程 2980

分类专栏

控制算法 8篇

博士面试 2篇

路径规划 2篇

C++

最新评论

Apollo MPC OSQP Solver
Newbee_Star: 大佬，debug模式怎么开呀

Hybrid A* 算法基本流程
GIANTi: 同问，RS曲线的曲率是不连续的，不知道混合A*中有没有...

线性模型预测控制二次规划(Qua...
凉了鸭: 不说我都没注意，太马虎了。。。

Apollo MPC OSQP Solver
是我你知道的: 这个是二次规划标准型里的线性部分的系数矩阵，...

Apollo MPC OSQP Solver
规划-YY: 楼主您好，我想问一下 在这个 osqp的c示例中，为什么二...

您愿意向朋友推荐“博客详情页”吗?

强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章

直线车道保持动力学分析 (Lane Keeping)

MPC自行车模型

线性模型预测控制二次规划(Quadratic Programming)的不同构造形式

2020年 11篇 2019年 1篇

目录