第5章 机械手自适应迭代学习控制

迭代学习控制是通过迭代修正改善某种控制目标，它的算法较为简单，且能在给定的时间范围内实现未知对象实际运行轨迹以高精度跟踪给定期望轨迹，不依赖系统的精确数学模型。因而一经推出，就在机器人控制领域得到广泛的运用。

迭代学习控制(iterative learning control，ILC)是智能控制中具有严格数学描述的一个分支。1984 年，Arimoto[1] 等人提出了迭代学习控制的概念，该控制方法适合于具有重复运动性质的被控对象，它不依赖于系统的精确数学模型，能以非常简单的方式处理不确定度相当高的非线性强耦合动态系统。目前，迭代学习控制在学习算法、收敛性、鲁棒性、学习速度及工程应用研究上取得了巨大的进展[2,3]。

5.1 控制器增益自适应整定的机械手迭代学习控制

本节通过对文献[4]的控制方法进行详细推导及仿真分析，研究一类机械手力臂自适应迭代学习控制的设计方法。并针对该控制算法存在的问题，提出了相应的改进算法。

5.1.1 问题的提出

考虑 n 关节机械手，其动态方程如下：

$$\boldsymbol{D}(\boldsymbol{q}^j(t))\ddot{\boldsymbol{q}}^j(t)+\boldsymbol{C}(\boldsymbol{q}^j(t),\dot{\boldsymbol{q}}^j(t))\dot{\boldsymbol{q}}^j(t)+\boldsymbol{G}(\boldsymbol{q}^j(t),\dot{\boldsymbol{q}}^j(t))+\boldsymbol{T}_a(t)=\boldsymbol{T}^j(t) \tag{5.1}$$

其中，j 为迭代次数，$t\in[0,t_f]$，$\boldsymbol{q}^j(t)\in\mathbf{R}^n$ 和 $\ddot{\boldsymbol{q}}^j(t)\in\mathbf{R}^n$ 分别为关节角度，角速度和角加速度，$\boldsymbol{D}(\boldsymbol{q}^j(t))\in\mathbf{R}^{n\times n}$ 为惯性项，$\boldsymbol{C}(\boldsymbol{q}^j(t),\dot{\boldsymbol{q}}^j(t))\dot{\boldsymbol{q}}^j(t)\in\mathbf{R}^n$ 表示离心力和哥氏力，$\boldsymbol{G}(\boldsymbol{q}^j(t),\dot{\boldsymbol{q}}^j(t))\in\mathbf{R}^n$ 为重力加摩擦力项，$\boldsymbol{T}_a(t)\in\mathbf{R}^n$ 为可重复的未知干扰，$\boldsymbol{T}^j(t)\in\mathbf{R}^n$ 为控制输入。

机械手动态方程满足如下特性：

(1) $\boldsymbol{D}(\boldsymbol{q}^j(t))$为对称正定的有界矩阵；

(2) $\dot{\boldsymbol{D}}(\boldsymbol{q}^j(t))-2\boldsymbol{C}(q^j(t),\dot{\boldsymbol{q}}^j(t))$为斜对称阵，即满足 $\boldsymbol{x}^{\mathrm{T}}(\dot{\boldsymbol{D}}(\boldsymbol{q}^j(t))-2\boldsymbol{C}(\boldsymbol{q}^j(t),\dot{\boldsymbol{q}}^j(t)))\boldsymbol{x}=0$。

机械手动态方程满足如下假设条件：

(1) 期望轨迹 $\boldsymbol{q}_d(t)$在 $t\in[0,t_f]$内三阶可导；

(2) 迭代过程满足初始条件，即 $\boldsymbol{q}_{\mathrm{d}}(0)-\boldsymbol{q}^{j}(0)=0,\dot{\boldsymbol{q}}_{\mathrm{d}}(0)-\dot{\boldsymbol{q}}^{j}(0)=0,j\in\mathbf{N}$。

5.1.2 控制器设计

针对系统式(5.1)，如果满足机器人特性(1)和(2)以及假设(1)和(2)，则控制律设计为

$$\boldsymbol{T}^{j}(t)=K_{\mathrm{p}}^{j}\boldsymbol{e}(t)+\boldsymbol{K}_{\mathrm{d}}^{j}\dot{\boldsymbol{e}}(t)+\boldsymbol{T}^{j-1}(t),\quad j=0,1,\cdots,N \tag{5.2}$$

控制律中增益切换规则为

$$\boldsymbol{K}_{\mathrm{p}}^{j}=\beta(j)\boldsymbol{K}_{\mathrm{p}}^{0},\quad \boldsymbol{K}_{\mathrm{d}}^{j}=\beta(j)\boldsymbol{K}_{\mathrm{d}}^{0},\quad \beta(j+1)>\beta(j) \tag{5.3}$$

其中，$j=1,2,\cdots,N$，$\boldsymbol{T}^{-1}(t)=0$，$\boldsymbol{e}^{j}(t)=\boldsymbol{q}_{\mathrm{d}}(t)-\boldsymbol{q}^{j}(t)$，$\dot{\boldsymbol{e}}^{j}(t)=\dot{\boldsymbol{q}}_{\mathrm{d}}(t)-\dot{\boldsymbol{q}}^{j}(t)$，$\boldsymbol{K}_{\mathrm{p}}^{0}$ 和 $\boldsymbol{K}_{\mathrm{d}}^{0}$ 为PD控制中初始的对角增益阵，且都为正定，$\beta(j)$为控制增益，且满足 $\beta(j)>1$。

首先实现动态方程式(5.1)的线性化，沿着指令轨迹$(\boldsymbol{q}_{\mathrm{d}}(t),\dot{\boldsymbol{q}}_{\mathrm{d}}(t),\ddot{\boldsymbol{q}}_{\mathrm{d}}(t))$，采用泰勒公式，对式(5.1)进行线性化，采用泰勒公式，$\boldsymbol{D}(\boldsymbol{q})$线性化为

$$\boldsymbol{D}(\boldsymbol{q})=\boldsymbol{D}(\boldsymbol{q}_{\mathrm{d}})+\left.\frac{\partial\boldsymbol{D}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}}}(\boldsymbol{q}-\boldsymbol{q}_{\mathrm{d}})+\boldsymbol{O}_{\mathrm{D}}(\cdot)$$

其中，$\boldsymbol{O}_{\mathrm{D}}(\cdot)$为 $\boldsymbol{D}(q)$一阶展开式的残差，即

$$-\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}}=-\boldsymbol{D}(\boldsymbol{q}_{\mathrm{d}})\ddot{\boldsymbol{q}}-\left.\frac{\partial\boldsymbol{D}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}}}\boldsymbol{e}\ddot{\boldsymbol{q}}-\boldsymbol{O}_{\mathrm{D}}(\cdot)\ddot{\boldsymbol{q}}$$

$$\boldsymbol{D}(\boldsymbol{q}_{\mathrm{d}})\ddot{\boldsymbol{q}}_{\mathrm{d}}-\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}}=\boldsymbol{D}(\boldsymbol{q}_{\mathrm{d}})\ddot{\boldsymbol{q}}_{\mathrm{d}}-\boldsymbol{D}(\boldsymbol{q}_{\mathrm{d}})\ddot{\boldsymbol{q}}-\left.\frac{\partial\boldsymbol{D}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}}}\boldsymbol{e}\ddot{\boldsymbol{q}}-\boldsymbol{O}_{\mathrm{D}}(\cdot)\ddot{\boldsymbol{q}}$$

$$\boldsymbol{D}(\boldsymbol{q}_{\mathrm{d}})\ddot{\boldsymbol{e}}+\left.\frac{\partial\boldsymbol{D}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}}}\ddot{\boldsymbol{q}}\boldsymbol{e}=\boldsymbol{D}(\boldsymbol{q}_{\mathrm{d}})\ddot{\boldsymbol{q}}_{\mathrm{d}}-\boldsymbol{D}(\boldsymbol{q}_{\mathrm{d}})\ddot{\boldsymbol{q}}-\boldsymbol{O}_{\mathrm{D}}(\cdot)\ddot{\boldsymbol{q}}$$

由于

$$\left.\frac{\partial\boldsymbol{D}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}}}\ddot{\boldsymbol{q}}\boldsymbol{e}=\left.\frac{\partial\boldsymbol{D}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}}}(\ddot{\boldsymbol{q}}+\ddot{\boldsymbol{q}}_{\mathrm{d}}-\ddot{\boldsymbol{q}}_{\mathrm{d}})\boldsymbol{e}=\left.\frac{\partial\boldsymbol{D}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}}}\ddot{\boldsymbol{q}}_{\mathrm{d}}\boldsymbol{e}-\left.\frac{\partial\boldsymbol{D}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}}}\ddot{\boldsymbol{e}}\boldsymbol{e}$$

则

$$\boldsymbol{D}(\boldsymbol{q}_{\mathrm{d}})\ddot{\boldsymbol{e}}+\left.\frac{\partial\boldsymbol{D}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}}}\ddot{\boldsymbol{q}}_{\mathrm{d}}\boldsymbol{e}-\left.\frac{\partial\boldsymbol{D}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}}}\ddot{\boldsymbol{e}}\boldsymbol{e}=\boldsymbol{D}(\boldsymbol{q}_{\mathrm{d}})\ddot{\boldsymbol{q}}_{\mathrm{d}}-\boldsymbol{D}(\boldsymbol{q}_{\mathrm{d}})\ddot{\boldsymbol{q}}-\boldsymbol{O}_{\mathrm{D}}(\cdot)\ddot{\boldsymbol{q}} \tag{5.4}$$

同理

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})=\boldsymbol{C}(\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}})+\left.\frac{\partial\boldsymbol{C}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}}}(\boldsymbol{q}-\boldsymbol{q}_{\mathrm{d}})+\left.\frac{\partial\boldsymbol{C}}{\partial\dot{\boldsymbol{q}}}\right|_{\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}}}(\dot{\boldsymbol{q}}-\dot{\boldsymbol{q}}_{\mathrm{d}})+\boldsymbol{O}_{\mathrm{C}}(\cdot)$$

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{e}}+\left.\frac{\partial\boldsymbol{C}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}}}\dot{\boldsymbol{q}}_{\mathrm{d}}\boldsymbol{e}+\left.\frac{\partial\boldsymbol{C}}{\partial\dot{\boldsymbol{q}}}\right|_{\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}}}\dot{\boldsymbol{q}}_{\mathrm{d}}\dot{\boldsymbol{e}}-\left.\frac{\partial\boldsymbol{C}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}}}\dot{\boldsymbol{e}}\boldsymbol{e}-\left.\frac{\partial\boldsymbol{C}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}}}\dot{\boldsymbol{e}}\dot{\boldsymbol{e}}$$

$$=\boldsymbol{C}(\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}})\dot{\boldsymbol{q}}_{\mathrm{d}}-\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}-\boldsymbol{O}_{\mathrm{C}}(\cdot)\dot{\boldsymbol{q}} \tag{5.5}$$

$$\boldsymbol{G}(\boldsymbol{q},\dot{\boldsymbol{q}})=\boldsymbol{G}(\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}})+\left.\frac{\partial\boldsymbol{G}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}}}(\boldsymbol{q}-\boldsymbol{q}_{\mathrm{d}})+\left.\frac{\partial\boldsymbol{G}}{\partial\dot{\boldsymbol{q}}}\right|_{\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}}}(\dot{\boldsymbol{q}}-\dot{\boldsymbol{q}}_{\mathrm{d}})+\boldsymbol{O}_{\mathrm{G}}(\cdot)$$

$$\left.\frac{\partial\boldsymbol{G}}{\partial\dot{\boldsymbol{q}}}\right|_{\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}}}\dot{\boldsymbol{e}}+\left.\frac{\partial\boldsymbol{G}}{\partial\boldsymbol{q}}\right|_{\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}}}\boldsymbol{e}=\boldsymbol{G}(\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}})-\boldsymbol{G}(\boldsymbol{q},\dot{\boldsymbol{q}})+\boldsymbol{O}_{\mathrm{G}}(\cdot) \tag{5.6}$$

由式(5.4)、式(5.5)和式(5.6)得

$$\boldsymbol{D}(t)\ddot{\boldsymbol{e}}+[\boldsymbol{C}+\boldsymbol{C}_{1}]\dot{\boldsymbol{e}}+\boldsymbol{F}\boldsymbol{e}+\boldsymbol{n}(\ddot{\boldsymbol{e}},\dot{\boldsymbol{e}},\boldsymbol{e},t)=\boldsymbol{H}-(\boldsymbol{D}\ddot{\boldsymbol{q}}+\boldsymbol{C}\dot{\boldsymbol{q}}+\boldsymbol{G}) \tag{5.7}$$

其中

$$\boldsymbol{n}(\ddot{\boldsymbol{e}},\dot{\boldsymbol{e}},\boldsymbol{e},t)=-\frac{\partial \boldsymbol{D}}{\partial \boldsymbol{q}}\Big|_{\boldsymbol{q}_{\mathrm{d}}}\ddot{\boldsymbol{e}}\boldsymbol{e}-\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{q}}\Big|_{\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}}}\dot{\boldsymbol{e}}\boldsymbol{e}-\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{q}}\Big|_{\boldsymbol{q}_{\mathrm{d}},\dot{\boldsymbol{q}}_{\mathrm{d}}}\dot{\boldsymbol{e}}\dot{\boldsymbol{e}}+O_{\mathrm{D}}(\cdot)\ddot{q}+\boldsymbol{O}_{\mathrm{C}}(\cdot)\dot{\boldsymbol{q}}-\boldsymbol{O}_{\mathrm{G}}(\cdot)$$

忽略残差项 $\boldsymbol{n}(\ddot{\boldsymbol{e}},\dot{\boldsymbol{e}},\boldsymbol{e},t)$，将式(5.1)代入式(5.7)，得第 j 次迭代的机器人动力学方程为

$$\boldsymbol{D}(t)\ddot{\boldsymbol{e}}^{j}(t)+[\boldsymbol{C}(t)+\boldsymbol{C}_1(t)]\dot{\boldsymbol{e}}^{j}(t)+\boldsymbol{F}(t)\boldsymbol{e}^{j}(t)-\boldsymbol{T}_{\mathrm{a}}(t)=\boldsymbol{H}(t)-\boldsymbol{T}^{j}(t)$$

其中

$$\boldsymbol{D}(t)=\boldsymbol{D}(\boldsymbol{q}_{\mathrm{d}}(t))$$

$$\boldsymbol{C}(t)=\boldsymbol{C}(\boldsymbol{q}_{\mathrm{d}}(t),\dot{\boldsymbol{q}}_{\mathrm{d}}(t))$$

$$\boldsymbol{C}_1(t)=\frac{\partial \boldsymbol{C}}{\partial \dot{\boldsymbol{q}}}\Big|_{\boldsymbol{q}_{\mathrm{d}}(t),\dot{\boldsymbol{q}}_{\mathrm{d}}(t)}\dot{\boldsymbol{q}}_{\mathrm{d}}(t)+\frac{\partial \boldsymbol{G}}{\partial \dot{\boldsymbol{q}}}\Big|_{\boldsymbol{q}_{\mathrm{d}}(t),\dot{\boldsymbol{q}}_{\mathrm{d}}(t)}$$

$$\boldsymbol{F}(t)=\frac{\partial \boldsymbol{D}}{\partial \boldsymbol{q}}\Big|_{\boldsymbol{q}_{\mathrm{d}}(t)}\ddot{\boldsymbol{q}}_{\mathrm{d}}(t)+\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{q}}\Big|_{\boldsymbol{q}_{\mathrm{d}}(t),\dot{\boldsymbol{q}}_{\mathrm{d}}(t)}\dot{\boldsymbol{q}}_{\mathrm{d}}(t)+\frac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}}\Big|_{\boldsymbol{q}_{\mathrm{d}}(t)}$$

$$\boldsymbol{H}(t)=\boldsymbol{D}(\dot{\boldsymbol{q}}_{\mathrm{d}}(t))\ddot{\boldsymbol{q}}_{\mathrm{d}}(t)+\boldsymbol{C}(\boldsymbol{q}_{\mathrm{d}}(t),\dot{\boldsymbol{q}}_{\mathrm{d}}(t))\dot{\boldsymbol{q}}_{\mathrm{d}}(t)+\boldsymbol{G}(\boldsymbol{q}_{\mathrm{d}}(t))$$

针对第 j 次迭代和第 $j+1$ 次迭代，方程式(5.7)可写为

$$\begin{cases}\boldsymbol{D}(t)\ddot{\boldsymbol{e}}^{j}(t)+[\boldsymbol{C}(t)+\boldsymbol{C}_1(t)]\dot{\boldsymbol{e}}^{j}(t)+\boldsymbol{F}(t)\boldsymbol{e}^{j}(t)-\boldsymbol{T}_{\mathrm{a}}(t)=\boldsymbol{H}(t)-\boldsymbol{T}^{j}(t)\\ \boldsymbol{D}(t)\ddot{\boldsymbol{e}}^{j+1}(t)+[\boldsymbol{C}(t)+\boldsymbol{C}_1(t)]\dot{\boldsymbol{e}}^{j+1}(t)+\boldsymbol{F}(t)\boldsymbol{e}^{j+1}(t)-\boldsymbol{T}_{\mathrm{a}}(t)=\boldsymbol{H}(t)-\boldsymbol{T}^{j+1}(t)\end{cases}\tag{5.8}$$

为了简单起见，取 $\boldsymbol{K}_{\mathrm{p}}^{0}=\boldsymbol{\Lambda}\boldsymbol{K}_{\mathrm{d}}^{0}$，并定义

$$\boldsymbol{y}^{j}(t)=\dot{\boldsymbol{e}}^{j}(t)+\boldsymbol{\Lambda}\boldsymbol{e}^{j}(t)\tag{5.9}$$

文献[4]提出了如下定理：

定理 5.1 假设系统式(5.1)满足机器人特性(P1,P2)和假设条件(A1,A2)。采用控制律式(5.2)及其增益切换规则式(5.3)，则对于 $t\in[0,t_{\mathrm{f}}]$，有

$$\boldsymbol{q}^{j}(t)\xrightarrow{j\to\infty}\boldsymbol{q}_{\mathrm{d}}(t),\quad \dot{\boldsymbol{q}}^{j}(t)\xrightarrow{j\to\infty}\dot{\boldsymbol{q}}_{\mathrm{d}}(t)$$

其中控制增益需要满足如下条件：

$$\begin{cases}l_p=\lambda_{\min}(\boldsymbol{K}_{\mathrm{d}}^{0}+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})>0\\ l_r=\lambda_{\min}(\boldsymbol{K}_{\mathrm{d}}^{0}+2\boldsymbol{C}+2\boldsymbol{F}/\boldsymbol{\Lambda}-2\dot{\boldsymbol{C}}_1/\boldsymbol{\Lambda})>0\\ l_p l_r\geqslant\left\|\|\boldsymbol{F}/\boldsymbol{\Lambda}-(\boldsymbol{C}+\boldsymbol{C}_1-\boldsymbol{\Lambda}\boldsymbol{D})\right\|_{\max}^{2}\end{cases}\tag{5.10}$$

其中 $\lambda_{\min}(\boldsymbol{A})$ 为矩阵 $\boldsymbol{A}$ 的最小特征值，$\|\boldsymbol{M}\|_{\max}=\max\|\boldsymbol{M}(t)\|$，$t\in[0,t_{\mathrm{f}}]$，$\|\boldsymbol{M}\|$ 为矩阵 $\boldsymbol{M}$ 的欧氏范数。

5.1.3 收敛性分析

参考文献[4]，下面给出了对定理 5.1 的收敛性分析和说明。定义 Lyapunov 函数为

$$V^{j}=\int_0^t\exp(-\rho\tau)\boldsymbol{y}^{j\mathrm{T}}\boldsymbol{K}_{\mathrm{d}}^{0}\boldsymbol{y}^{j}\,\mathrm{d}\tau\geqslant 0$$

其中，$\boldsymbol{K}_{\mathrm{d}}^{0}>0$ 为 PD 控制中 D 控制项的初始增益，ρ 为正实数。

由式(5.9)得

$$\delta\boldsymbol{y}^{j}=\boldsymbol{y}^{j+1}-\boldsymbol{y}^{j}=\dot{\boldsymbol{e}}^{j+1}+\boldsymbol{\Lambda}\boldsymbol{e}^{j+1}-(\dot{\boldsymbol{e}}^{j}+\boldsymbol{\Lambda}\boldsymbol{e}^{j})=\delta\dot{\boldsymbol{e}}^{j}+\boldsymbol{\Lambda}\,\delta\boldsymbol{e}^{j}\tag{5.11}$$

由式(5.8)得

$$\boldsymbol{D}(t)(\ddot{\boldsymbol{e}}^{j+1}(t)-\ddot{\boldsymbol{e}}^{j}(t))=-[\boldsymbol{C}(t)+\boldsymbol{C}_1(t)](\dot{\boldsymbol{e}}^{j+1}(t)-\dot{\boldsymbol{e}}^{j}(t))-\boldsymbol{F}(t)(\boldsymbol{e}^{j+1}(t)-\boldsymbol{e}^{j}(t))-(\boldsymbol{T}^{j+1}(t)-\boldsymbol{T}^{j}(t)) \tag{5.12}$$

由式(5.11)和式(5.12)得

$$\begin{aligned}\boldsymbol{D}\delta\dot{\boldsymbol{y}}^{j}&=\boldsymbol{D}\delta\ddot{\boldsymbol{e}}^{j+1}+\boldsymbol{D}\boldsymbol{\Lambda}\,\delta\dot{\boldsymbol{e}}^{j}=\boldsymbol{D}(\ddot{\boldsymbol{e}}^{j+1}-\ddot{\boldsymbol{e}}^{j})+\boldsymbol{D}\boldsymbol{\Lambda}\,(\dot{\boldsymbol{e}}^{j+1}-\dot{\boldsymbol{e}}^{j})\\&=-[\boldsymbol{C}(t)+\boldsymbol{C}_1(t)](\dot{\boldsymbol{e}}^{j+1}(t)-\dot{\boldsymbol{e}}^{j}(t))-\boldsymbol{F}(t)(\boldsymbol{e}^{j+1}(t)-\boldsymbol{e}^{j}(t))-\\&\quad(\boldsymbol{T}^{j+1}(t)-\boldsymbol{T}^{j}(t))+\boldsymbol{D}\boldsymbol{\Lambda}\,(\dot{\boldsymbol{e}}^{j+1}-\dot{\boldsymbol{e}}^{j})\\&=-[\boldsymbol{C}(t)+\boldsymbol{C}_1(t)]\delta\dot{\boldsymbol{e}}^{j}-\boldsymbol{F}(t)\delta\boldsymbol{e}^{j}-(\boldsymbol{K}_{\mathrm{p}}^{j+1}\boldsymbol{e}^{j+1}(t)+\boldsymbol{K}_{\mathrm{d}}^{j+1}\dot{\boldsymbol{e}}^{j+1}(t))+\\&\quad\boldsymbol{D}\boldsymbol{\Lambda}\,(\dot{\boldsymbol{e}}^{j+1}-\dot{\boldsymbol{e}}^{j})\end{aligned}$$

由 $\boldsymbol{K}_{\mathrm{p}}^{0}=\boldsymbol{\Lambda}\boldsymbol{K}_{\mathrm{d}}^{0}$ 和式(5.3)知 $\boldsymbol{K}_{\mathrm{p}}^{j+1}=\boldsymbol{\Lambda}\boldsymbol{K}_{\mathrm{d}}^{j+1}$，考虑到式(5.11)，可得

$$\begin{aligned}\boldsymbol{D}\delta\dot{\boldsymbol{y}}^{j}&=-[\boldsymbol{C}+\boldsymbol{C}_1](\delta\boldsymbol{y}^{j}-\boldsymbol{\Lambda}\,\delta\boldsymbol{e}^{j})-\boldsymbol{F}(t)\delta\boldsymbol{e}^{j}+\boldsymbol{D}\boldsymbol{\Lambda}\,(\dot{\boldsymbol{e}}^{j+1}-\dot{\boldsymbol{e}}^{j})-\\&\quad(\boldsymbol{\Lambda}\boldsymbol{K}_{\mathrm{d}}^{j+1}\boldsymbol{e}^{j+1}+\boldsymbol{K}_{\mathrm{d}}^{j+1}\dot{\boldsymbol{e}}^{j+1})\end{aligned}$$

由于

$$\boldsymbol{D}\boldsymbol{\Lambda}\,(\dot{\boldsymbol{e}}^{j+1}-\dot{\boldsymbol{e}}^{j})=\boldsymbol{D}\boldsymbol{\Lambda}\,[(\boldsymbol{y}^{j+1}-\boldsymbol{\Lambda}\boldsymbol{e}^{j+1})-(\boldsymbol{y}^{j}-\boldsymbol{\Lambda}\boldsymbol{e}^{j})]=\boldsymbol{D}\boldsymbol{\Lambda}\,\delta\boldsymbol{y}^{j}-\boldsymbol{D}\boldsymbol{\Lambda}^{2}\delta\boldsymbol{e}^{j}$$

$$\boldsymbol{\Lambda}\,\boldsymbol{K}_{\mathrm{d}}^{j+1}\boldsymbol{e}^{j+1}+\boldsymbol{K}_{\mathrm{d}}^{j+1}\dot{\boldsymbol{e}}^{j+1}=\boldsymbol{K}_{\mathrm{d}}^{j+1}(\boldsymbol{\Lambda}\boldsymbol{e}^{j+1}+\dot{\boldsymbol{e}}^{j+1})=\boldsymbol{K}_{\mathrm{d}}^{j+1}\boldsymbol{y}^{j+1}=\boldsymbol{K}_{\mathrm{d}}^{j+1}(\delta\boldsymbol{y}^{j}+\boldsymbol{y}^{j})$$

则

$$\begin{aligned}\boldsymbol{D}\delta\dot{\boldsymbol{y}}^{j}&=-(\boldsymbol{C}+\boldsymbol{C}_1)(\delta\boldsymbol{y}^{j}-\boldsymbol{\Lambda}\,\delta\boldsymbol{e}^{j})-\boldsymbol{F}\delta\boldsymbol{e}^{j}+\boldsymbol{D}\boldsymbol{\Lambda}\,\delta\boldsymbol{y}^{j}-\boldsymbol{D}\boldsymbol{\Lambda}^{2}\delta\boldsymbol{e}^{j}-\boldsymbol{K}_{\mathrm{d}}^{j+1}(\delta\boldsymbol{y}^{j}+\boldsymbol{y}^{j})\\&=-(\boldsymbol{C}+\boldsymbol{C}_1-\boldsymbol{\Lambda}\boldsymbol{D}+\boldsymbol{K}_{\mathrm{d}}^{j+1})\delta\boldsymbol{y}^{j}-(\boldsymbol{F}-\boldsymbol{\Lambda}\,(\boldsymbol{C}+\boldsymbol{C}_1-\boldsymbol{\Lambda}\boldsymbol{D}))\delta\boldsymbol{e}^{j}-\boldsymbol{K}_{\mathrm{d}}^{j+1}\boldsymbol{y}^{j}\end{aligned}$$

则

$$\boldsymbol{K}_{\mathrm{d}}^{j+1}\boldsymbol{y}^{j}=-\boldsymbol{D}\delta\dot{\boldsymbol{y}}^{j}-(\boldsymbol{C}+\boldsymbol{C}_1-\boldsymbol{\Lambda}\boldsymbol{D}+\boldsymbol{K}_{\mathrm{d}}^{j+1})\delta\boldsymbol{y}^{j}-(\boldsymbol{F}-\boldsymbol{\Lambda}\,(\boldsymbol{C}+\boldsymbol{C}_1-\boldsymbol{\Lambda}\boldsymbol{D}))\delta\boldsymbol{e}^{j} \tag{5.13}$$

由 V^{j} 的定义，得

$$V^{j+1}=\int_0^t\exp(-\rho\tau)\boldsymbol{y}^{j+1^{\mathrm{T}}}\boldsymbol{K}_{\mathrm{d}}^{0}\boldsymbol{y}^{j+1}\mathrm{d}\tau$$

定义 $\Delta V^{j}=V^{j+1}-V^{j}$，由式(5.3)和式(5.11)，并将式(5.13)代入，得

$$\begin{aligned}\Delta V^{j}&=\int_0^t\exp(-\rho\tau)(\delta\boldsymbol{y}^{j^{\mathrm{T}}}+\boldsymbol{y}^{j})^{\mathrm{T}}\boldsymbol{K}_{\mathrm{d}}^{0}(\delta\boldsymbol{y}^{j^{\mathrm{T}}}+\boldsymbol{y}^{j})\mathrm{d}\tau-\int_0^t\exp(-\rho\tau)\boldsymbol{y}^{j^{\mathrm{T}}}\boldsymbol{K}_{\mathrm{d}}^{0}\boldsymbol{y}^{j}\mathrm{d}\tau\\&=\int_0^t\exp(-\rho\tau)(\delta\boldsymbol{y}^{j^{\mathrm{T}}}\boldsymbol{K}_{\mathrm{d}}^{0}\delta\boldsymbol{y}^{j}+2\delta\boldsymbol{y}^{j^{\mathrm{T}}}\boldsymbol{K}_{\mathrm{d}}^{0}y^{j})\mathrm{d}\tau\\&=\frac{1}{\beta(j+1)}\int_0^t\exp(-\rho\tau)(\delta\boldsymbol{y}^{j^{\mathrm{T}}}\boldsymbol{K}_{\mathrm{d}}^{j+1}\delta\boldsymbol{y}^{j}+2\delta\boldsymbol{y}^{j^{\mathrm{T}}}\boldsymbol{K}_{\mathrm{d}}^{j+1}\boldsymbol{y}^{j})\mathrm{d}\tau\\&=\frac{1}{\beta(j+1)}\Big\{\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j^{\mathrm{T}}}\boldsymbol{K}_{\mathrm{d}}^{j+1}\delta\boldsymbol{y}^{j}\mathrm{d}\tau-2\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j^{\mathrm{T}}}\boldsymbol{D}\delta\dot{\boldsymbol{y}}^{j}\mathrm{d}\tau-\\&\quad2\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j^{\mathrm{T}}}((\boldsymbol{C}+\boldsymbol{C}_1-\boldsymbol{\Lambda}\boldsymbol{D}+\boldsymbol{K}_{\mathrm{d}}^{j+1})\delta\boldsymbol{y}^{j}+(\boldsymbol{F}-\boldsymbol{\Lambda}\,(\boldsymbol{C}+\boldsymbol{C}_1-\boldsymbol{\Lambda}\boldsymbol{D}))\delta\boldsymbol{e}^{j})\mathrm{d}\tau\end{aligned}$$

应用分部积分方法，根据初始条件(A2)知 $\delta\boldsymbol{y}^{j}(0)=0$，则

$$\begin{aligned}\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j^{\mathrm{T}}}\boldsymbol{D}\delta\dot{\boldsymbol{y}}^{j}\mathrm{d}\tau&=\exp(-\rho\tau)\delta\boldsymbol{y}^{j^{\mathrm{T}}}\boldsymbol{D}\delta\boldsymbol{y}^{j}\Big|_0^t-\int_0^t(\exp(-\rho\tau)\delta\boldsymbol{y}^{j^{\mathrm{T}}}\boldsymbol{D})'\delta\boldsymbol{y}^{j}\mathrm{d}\tau\\&=\exp(-\rho t)\delta\boldsymbol{y}^{j^{\mathrm{T}}}(t)\boldsymbol{D}(t)\delta\boldsymbol{y}^{j}(t)+\rho\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j^{\mathrm{T}}}\boldsymbol{D}\delta\boldsymbol{y}^{j}\mathrm{d}\tau-\\&\quad\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j^{\mathrm{T}}}\boldsymbol{D}\delta\dot{\boldsymbol{y}}^{j}\mathrm{d}\tau-\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j^{\mathrm{T}}}\dot{\boldsymbol{D}}\delta\boldsymbol{y}^{j}\mathrm{d}\tau\end{aligned}$$

将上式两端同项合并，得

$$2\int_0^t \exp(-\rho\tau)\delta y^{j\mathrm{T}}\boldsymbol{D}\delta\dot{\boldsymbol{y}}^j\,\mathrm{d}\tau=\exp(-\rho t)\delta\boldsymbol{y}^{j\mathrm{T}}(t)\boldsymbol{D}(t)\delta\boldsymbol{y}^j(t)+\rho\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j\mathrm{T}}\boldsymbol{D}\delta\boldsymbol{y}^j\,\mathrm{d}\tau-\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j\mathrm{T}}\dot{\boldsymbol{D}}\delta\boldsymbol{y}^j\,\mathrm{d}\tau$$

由特性(P2)，可得

$$\int_0^t\delta\boldsymbol{y}^{j\mathrm{T}}\dot{\boldsymbol{D}}\delta\boldsymbol{y}^j\,\mathrm{d}\tau=2\int_0^t\delta\boldsymbol{y}^{j\mathrm{T}}\boldsymbol{C}\delta\boldsymbol{y}^j\,\mathrm{d}\tau$$

则

$$\begin{aligned}\Delta V^j=\frac{1}{\beta(j+1)}\Big\{&-\exp(-\rho\tau)\delta\boldsymbol{y}^{j\mathrm{T}}\boldsymbol{D}(t)\delta\boldsymbol{y}^j(t)-\rho\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j\mathrm{T}}\boldsymbol{D}\delta\boldsymbol{y}^j\,\mathrm{d}\tau-\\&2\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j\mathrm{T}}(\boldsymbol{F}-\boldsymbol{\Lambda}(\boldsymbol{C}+\boldsymbol{C}_1-\boldsymbol{\Lambda}\boldsymbol{D}))\delta\boldsymbol{e}^j\,\mathrm{d}\tau-\\&\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^{j+1}+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\boldsymbol{y}^j\,\mathrm{d}\tau\Big\}\end{aligned}$$

由于

$$\begin{aligned}\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j\mathrm{T}}\boldsymbol{K}_{\mathrm{d}}^{j+1}\delta\boldsymbol{y}^j\,\mathrm{d}\tau&=\beta(j+1)\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j\mathrm{T}}\boldsymbol{K}_{\mathrm{d}}^0\delta\boldsymbol{y}^j\,\mathrm{d}\tau\\&\geqslant\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j\mathrm{T}}\boldsymbol{K}_{\mathrm{d}}^0\delta\boldsymbol{y}^j\,\mathrm{d}\tau\end{aligned}$$

利用式(5.11)，并将 $\delta\boldsymbol{y}^j$ 展开成 $\delta\dot{\boldsymbol{e}}^j+\boldsymbol{\Lambda}\delta\boldsymbol{e}^j$，得

$$\begin{aligned}\Delta V^j\leqslant\frac{1}{\beta(j+1)}\Big\{&-\exp(-\rho\tau)\delta\boldsymbol{y}^{j\mathrm{T}}\boldsymbol{D}(t)\delta\boldsymbol{y}^j(t)-\rho\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j\mathrm{T}}\boldsymbol{D}\delta\boldsymbol{y}^j\,\mathrm{d}\tau-\\&2\int_0^t\exp(-\rho\tau)\delta\dot{\boldsymbol{e}}^{j\mathrm{T}}(\boldsymbol{F}-\boldsymbol{\Lambda}(\boldsymbol{C}+\boldsymbol{C}_1-\boldsymbol{\Lambda}\boldsymbol{D}))\delta\boldsymbol{e}^j\,\mathrm{d}\tau-\\&2\boldsymbol{\Lambda}\int_0^t\exp(-\rho\tau)\delta\boldsymbol{e}^{j\mathrm{T}}(\boldsymbol{F}-\boldsymbol{\Lambda}(\boldsymbol{C}+\boldsymbol{C}_1-\boldsymbol{\Lambda}\boldsymbol{D}))\delta\boldsymbol{e}^j\,\mathrm{d}\tau-\\&\int_0^t\exp(-\rho\tau)\delta\dot{\boldsymbol{e}}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\dot{\boldsymbol{e}}^j\,\mathrm{d}\tau-\\&2\boldsymbol{\Lambda}\int_0^t\exp(-\rho\tau)\delta\boldsymbol{e}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\dot{\boldsymbol{e}}^j\,\mathrm{d}\tau-\\&\boldsymbol{\Lambda}^2\int_0^t\exp(-\rho\tau)\delta\boldsymbol{e}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\boldsymbol{e}^j\,\mathrm{d}\tau\Big\}\end{aligned}$$

应用分部积分方法，根据初始条件，有 $\delta\boldsymbol{e}^j(0)=0$，则

$$\begin{aligned}\int_0^t\exp(-\rho\tau)\delta\boldsymbol{e}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\dot{\boldsymbol{e}}^j\,\mathrm{d}\tau=&\exp(-\rho\tau)\delta\boldsymbol{e}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\boldsymbol{e}^j\Big|_0^t-\\&\int_0^t-\rho\exp(-\rho\tau)\delta\boldsymbol{e}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\boldsymbol{e}^j\,\mathrm{d}\tau-\\&\int_0^t\exp(-\rho\tau)\delta\dot{\boldsymbol{e}}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\boldsymbol{e}^j\,\mathrm{d}\tau-\\&\int_0^t\exp(-\rho\tau)\delta\boldsymbol{e}^{j\mathrm{T}}(2\dot{\boldsymbol{C}}_1-2\boldsymbol{\Lambda}\dot{\boldsymbol{D}})\delta\boldsymbol{e}^j\,\mathrm{d}\tau\end{aligned}$$

将上式两端同项合并，并将两端同乘以 $\boldsymbol{\Lambda}$，得

$$2\boldsymbol{\Lambda}\int_0^t \mathrm{e}^{-\rho\tau}\delta\boldsymbol{e}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\dot{\boldsymbol{e}}^j\,\mathrm{d}\tau=\boldsymbol{\Lambda}\exp(-\rho t)\delta\boldsymbol{e}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\boldsymbol{e}^j+$$
$$\rho\boldsymbol{\Lambda}\int_0^t\exp(-\rho\tau)\delta\boldsymbol{e}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\boldsymbol{e}^j\,\mathrm{d}\tau+$$
$$2\boldsymbol{\Lambda}\int_0^t\exp(-\rho\tau)\delta\boldsymbol{e}^{j\mathrm{T}}(\boldsymbol{\Lambda}\dot{\boldsymbol{D}}-\dot{\boldsymbol{C}}_1)\delta\boldsymbol{e}^j\,\mathrm{d}\tau$$

则

$$\Delta V^j\leqslant\frac{1}{\beta(j+1)}\Big\{-\exp(-\rho\tau)\delta\boldsymbol{y}^{j\mathrm{T}}\boldsymbol{D}\delta\boldsymbol{y}^j(t)-\rho\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j\mathrm{T}}\boldsymbol{D}\delta\boldsymbol{y}^j\,\mathrm{d}\tau-$$
$$\boldsymbol{\Lambda}\exp(-\rho\tau)\delta\boldsymbol{e}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\boldsymbol{e}^j-$$
$$\rho\boldsymbol{\Lambda}\int_0^t\exp(-\rho\tau)\delta\boldsymbol{e}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\boldsymbol{e}^j\,\mathrm{d}\tau-\int_0^t\exp(-\rho\tau)w\,\mathrm{d}\tau\Big\}$$
$$\leqslant\frac{1}{\beta(j+1)}\Big\{-\exp(-\rho\tau)\delta\boldsymbol{y}^{j\mathrm{T}}\boldsymbol{D}\delta\boldsymbol{y}^j(t)-\rho\int_0^t\exp(-\rho\tau)\delta\boldsymbol{y}^{j\mathrm{T}}\boldsymbol{D}\delta\boldsymbol{y}^j\,\mathrm{d}\tau-$$
$$\boldsymbol{\Lambda}\exp(-\rho\tau)\delta\boldsymbol{e}^{j\mathrm{T}}l_p\delta\boldsymbol{e}^j-\rho\boldsymbol{\Lambda}\int_0^t\exp(-\rho\tau)\delta\boldsymbol{e}^{j\mathrm{T}}l_p\delta\boldsymbol{e}^j\,\mathrm{d}\tau-\int_0^t\exp(-\rho\tau)w\,\mathrm{d}\tau\Big\}$$

其中

$$w=\delta\dot{\boldsymbol{e}}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\dot{\boldsymbol{e}}^j+2\delta\dot{\boldsymbol{e}}^{j\mathrm{T}}(\boldsymbol{F}-\boldsymbol{\Lambda}(\boldsymbol{C}+\boldsymbol{C}_1-\boldsymbol{\Lambda}\boldsymbol{D}))\delta\boldsymbol{e}^j+$$
$$2\boldsymbol{\Lambda}\,\delta\boldsymbol{e}^{j\mathrm{T}}(\boldsymbol{\Lambda}\dot{\boldsymbol{D}}-\dot{\boldsymbol{C}}_1)\delta\boldsymbol{e}^j+\boldsymbol{\Lambda}^2\delta\boldsymbol{e}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\boldsymbol{e}^j+$$
$$2\boldsymbol{\Lambda}\,\delta e^{j\mathrm{T}}(\boldsymbol{F}-\boldsymbol{\Lambda}(\boldsymbol{C}+\boldsymbol{C}_1-\boldsymbol{\Lambda}\boldsymbol{D}))\delta e^j$$
$$=\delta\dot{e}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}_1-2\boldsymbol{\Lambda}\boldsymbol{D})\delta\dot{\boldsymbol{e}}^j+2\boldsymbol{\Lambda}\,\delta\dot{\boldsymbol{e}}^{j\mathrm{T}}(\boldsymbol{F}/\boldsymbol{\Lambda}-(\boldsymbol{C}+\boldsymbol{C}_1-\boldsymbol{\Lambda}\boldsymbol{D}))\delta\boldsymbol{e}^j+$$
$$\boldsymbol{\Lambda}^2\delta\boldsymbol{e}^{j\mathrm{T}}(\boldsymbol{K}_{\mathrm{d}}^0+2\boldsymbol{C}+2\boldsymbol{F}/\boldsymbol{\Lambda}-2\dot{\boldsymbol{C}}_1/\boldsymbol{\Lambda})\delta\boldsymbol{e}^j$$

取 $\boldsymbol{Q}=\boldsymbol{F}/\boldsymbol{\Lambda}-(\boldsymbol{C}+\boldsymbol{C}_1-\boldsymbol{\Lambda}\boldsymbol{D})$，则由式(5.10)，得

$$w\geqslant l_p\|\delta\dot{\boldsymbol{e}}\|^2+2\boldsymbol{\Lambda}\,\delta\dot{\boldsymbol{e}}^{\mathrm{T}}\boldsymbol{Q}\delta\boldsymbol{e}+\boldsymbol{\Lambda}^2l_r\|\delta\boldsymbol{e}\|^2$$

采用 Cauchy-Schwarz 不等式，有

$$\delta\dot{\boldsymbol{e}}^{\mathrm{T}}\boldsymbol{Q}\delta\boldsymbol{e}\geqslant-\|\delta\dot{\boldsymbol{e}}\|\,\|\boldsymbol{Q}\|_{\max}\|\delta\boldsymbol{e}\|$$
$$w\geqslant l_p\|\delta\dot{\boldsymbol{e}}\|^2-2\boldsymbol{\Lambda}\,\|\delta\dot{\boldsymbol{e}}\|\,\|\boldsymbol{Q}_{\max}\|\,\|\delta\boldsymbol{e}\|+\boldsymbol{\Lambda}^2l_r\|\delta\boldsymbol{e}\|^2$$
$$=l_p\Big(\|\delta\dot{\boldsymbol{e}}\|-\frac{\boldsymbol{\Lambda}}{l_p}\|\boldsymbol{Q}_{\max}\|\,\|\delta\boldsymbol{e}\|\Big)^2+\boldsymbol{\Lambda}^2\Big(l_r-\frac{1}{l_p}\|\boldsymbol{Q}\|_{\max}^2\Big)\|\delta\boldsymbol{e}\|^2\geqslant0$$

则 $\Delta V_j\leqslant0$，即

$$V^{j+1}\leqslant V^j$$

由于 $\boldsymbol{K}_{\mathrm{d}}^0$ 为正定阵，$V^j>0$ 且 V^j 有界，则当 $j\to\infty$ 时，$y^j(t)\to0$。由于 $\boldsymbol{e}^j(t)$ 和 $\dot{\boldsymbol{e}}^j(t)$ 为两个相互独立的变量，$\boldsymbol{\Lambda}$ 为正定常数阵，如取 $j\to\infty$，则 $\boldsymbol{e}^j(t)\to0$，$\dot{\boldsymbol{e}}^j(t)\to0$，$t\in[0,t_{\mathrm{f}}]$。

通过上面的分析，可得结论：

$$\boldsymbol{q}^j(t)\xrightarrow{j\to\infty}\boldsymbol{q}_{\mathrm{d}}(t),\quad\dot{\boldsymbol{q}}^j(t)\xrightarrow{j\to\infty}\dot{\boldsymbol{q}}_{\mathrm{d}}(t),\quad t\in[0,t_{\mathrm{f}}]$$

定理 5.1 中描述的控制算法的不足之处：针对的是重复性干扰，忽略了线性化残差项 $\boldsymbol{n}(\ddot{\boldsymbol{e}}^j,\dot{\boldsymbol{e}}^j,\boldsymbol{e}^j,t)$，且机械手动力学方程为确定的。针对这一问题，5.2 节中给出了改进的控制律。

5.1.4 仿真实例

针对双关节机械手动态方程式(5.1)进行仿真,方程中的各项取

$$\boldsymbol{D}(\boldsymbol{q})=\begin{bmatrix} i_1+i_2+2m_2r_2l_1\cos q_2 & i_2+m_2r_2l_1\cos(q_2) \\ i_2+m_2r_2l_1\cos(q_2) & i_2 \end{bmatrix}$$

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})=\begin{bmatrix} -m_2r_2l_1\dot{q}_2\sin(q_2) & -m_2r_2l_1(\dot{q}_1+\dot{q}_2)\sin(q_2) \\ m_2r_2l_1\dot{q}_1\sin(q_2) & 0 \end{bmatrix}$$

$$\boldsymbol{G}(\boldsymbol{q})=\begin{bmatrix} (m_1r_1+m_2l_1)g\cos q_1+m_2r_2g\cos(q_1+q_2) \\ m_2r_2g\cos(q_1+q_2) \end{bmatrix}$$

可重复的干扰为 $d_1(t)=a0.3\sin t$, $d_2(t)=a0.1(1-\mathrm{e}^{-t})$, $a=1$, $\boldsymbol{T}_a=[d_1 \quad d_2]^{\mathrm{T}}$。

系统参数取 $m_1=10$, $m_2=5$, $l_1=1$, $l_2=0.5$, $r_1=0.5$, $r_2=0.25$, $i_1=0.83+m_1r_1^2+m_2l_1^2$, $i_2=0.3+m_2r_2^2$。

角度期望轨迹 $q_1=\sin 3t$, $q_2=\cos 3t$, 取 $\boldsymbol{\Lambda}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$, 控制器参数设计为 $\boldsymbol{K}_{\mathrm{p}}^0=\boldsymbol{K}_{\mathrm{d}}^0=\begin{bmatrix}210 & 0\\ 0 & 210\end{bmatrix}$, $\beta(j)=2j$, $\boldsymbol{K}_{\mathrm{p}}^j=2j\boldsymbol{K}_{\mathrm{p}}^0$, $\boldsymbol{K}_{\mathrm{d}}^j=2j\boldsymbol{K}_{\mathrm{d}}^0$, $j=1,2,\cdots,N$。

系统的初始状态为 $\boldsymbol{x}=[3 \quad 0 \quad 0 \quad 1]^{\mathrm{T}}$, 取 $t_{\mathrm{f}}=5$, 迭代次数取 5 次。仿真结果见图 5.1 至图 5.5 所示。

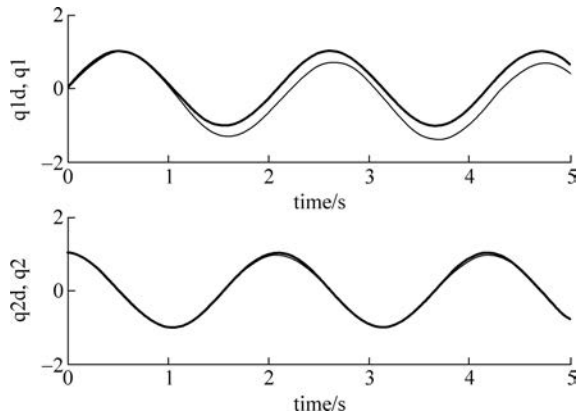


图 5.1 双关节 5 次迭代的角度跟踪过程

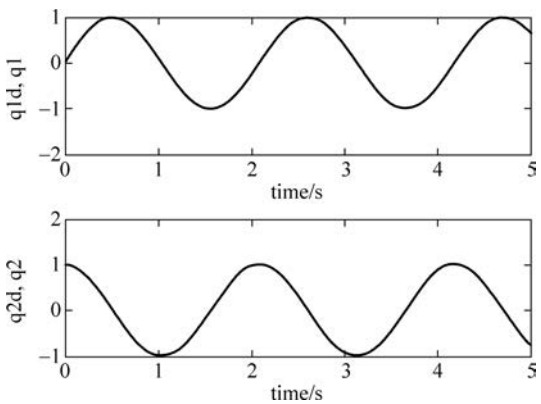


图 5.2 双关节第 5 次的角度跟踪

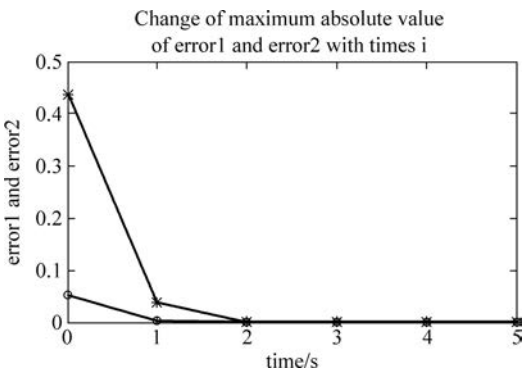


图 5.3 双关节 5 次角度跟踪的误差收敛过程

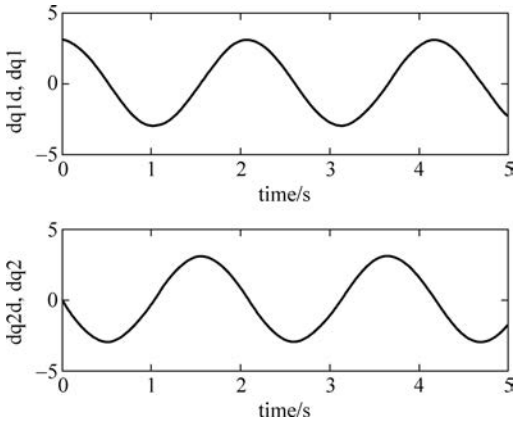


图 5.4　双关节第 5 次的角速度跟踪

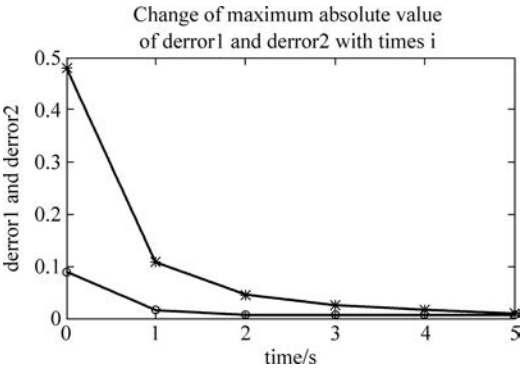


图 5.5　双关节 5 次角速度跟踪的误差收敛过程

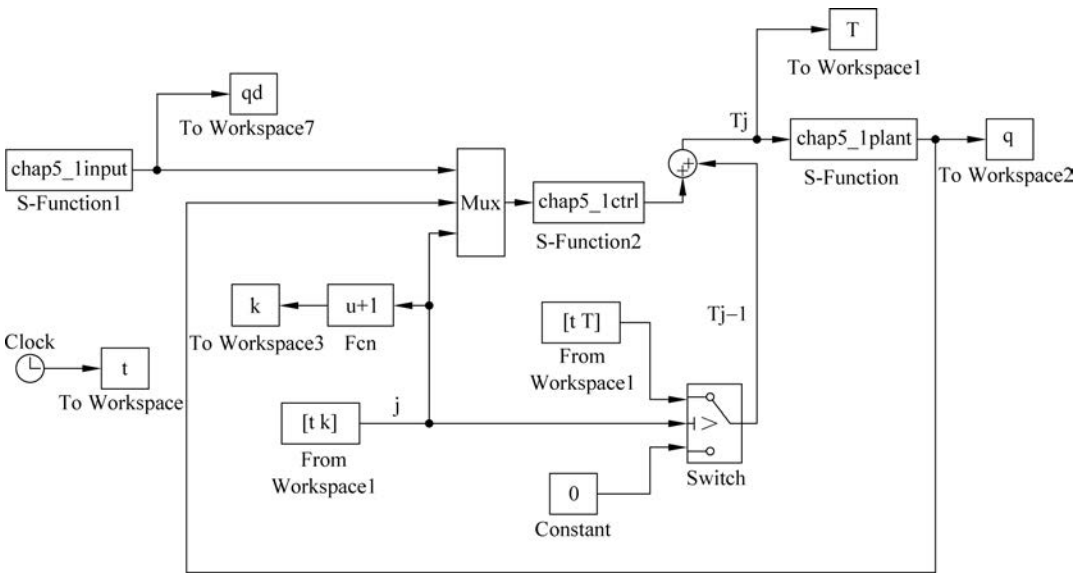
仿真程序：

（1）主程序：chap5_1main.m

```
% Adaptive switching Learning Control for 2DOF robot manipulators
clear all;
close all;

t = [0:0.001:5]';
T1(1:5001) = 0;
T1 = T1';
T2 = T1;
T = [T1 T2];
k(1:5001) = 0;
k = k';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M = 5;
for i = 0:1:M     % Start Learning Control
i
pause(0.01);

sim('chap5_1sim',[0,5]);

q1 = q(:,3);
q2 = q(:,4);
dq1 = q(:,1);
dq2 = q(:,2);
q1d = qd(:,1);
q2d = qd(:,2);
dq1d = qd(:,3);
dq2d = qd(:,4);
e1 = q1d - q1;
e2 = q2d - q2;
de1 = dq1d - dq1;
de2 = dq2d - dq2;

figure(1);
subplot(211);
hold on;
plot(t,q1,'b',t,q1d,'r');
xlabel('time(s)');ylabel('q1d,q1');
```

```
subplot(212);
hold on;
plot(t,q2,'b',t,q2d,'r');
xlabel('time(s)');ylabel('q2d,q2');

j = i + 1;
times(j) = i;
e1i(j) = max(abs(e1));
e2i(j) = max(abs(e2));
de1i(j) = max(abs(de1));
de2i(j) = max(abs(de2));
end            % End of i
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(2);
subplot(211);
plot(t,q1d,'r',t,q1,'b');
xlabel('time(s)');ylabel('q1d,q1');
subplot(212);
plot(t,q2d,'r',t,q2,'b');
xlabel('time(s)');ylabel('q2d,q2');

figure(3);
plot(times,e1i,'*-r',times,e2i,'o-b');
title('Change of maximum absolute value of error1 and error2 with times i');
xlabel('times');ylabel('error1 and error2');

figure(4);
subplot(211);
plot(t,dq1d,'r',t,dq1,'b');
xlabel('time(s)');ylabel('dq1d,dq1');
subplot(212);
plot(t,dq2d,'r',t,dq2,'b');
xlabel('time(s)');ylabel('dq2d,dq2');

figure(5);
plot(times,de1i,'*-r',times,de2i,'o-b');
title('Change of maximum absolute value of derror1 and derror2 with times i');
xlabel('times');ylabel('derror1 and derror2');
```

(2) Simulink 子程序：chap5_1sim. mdl。



（3）被控对象子程序：chap5_1plant.m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 1,
    sys = mdlDerivatives(t,x,u);
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 4;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 4;
sizes.NumInputs      = 2;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [3;0;0;1];
str = [];
ts  = [0 0];
function sys = mdlDerivatives(t,x,u)
a = 1.0;
d1 = a * 0.3 * sin(t);
d2 = a * 0.1 * (1 - exp( - t));

dq1 = x(1);
dq2 = x(2);
q1 = x(3);
q2 = x(4);

tol1 = u(1);
tol2 = u(2);

m1 = 10;m2 = 5;
l1 = 1;l2 = 0.5;
r1 = 0.5;r2 = 0.25;
i1 = 0.83 + m1 * r1^2 + m2 * l1^2;
i2 = 0.3 + m2 * r2^2;
g = 9.8;

D = [i1 + i2 + 2 * m2 * r2 * l1 * cos(q2) i2 + m2 * r2 * l1 * cos(q2);
   i2 + m2 * r2 * l1 * cos(q2) i2];
C = [ - m2 * r2 * l1 * dq2 * sin(q2), - m2 * r2 * l1 * (dq1 + dq2) * sin(q2);
    m2 * r2 * l1 * dq1 * sin(q2),0];
G = [(m1 * r1 + m2 * l1) * g * cos(q1) + m2 * r2 * g * cos(q1 + q2);m2 * r2 * g * cos(q1 + q2)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

D2 = inv(D);
T = [d1;d2];
A = - D2 * C;
Z = - D2 * G;

sys(1) = A(1,1) * x(1) + A(1,2) * x(2) + Z(1) + D2(1,1) * ( - T(1) + tol1) + D2(1,2) * ( - T(2) + tol2);
```

```
sys(2) = A(2,1) * x(1) + A(2,2) * x(2) + Z(2) + D2(2,1) * ( - T(1) + tol1) + D2(2,2) * ( - T(2) + tol2);
sys(3) = x(1);
sys(4) = x(2);
function sys = mdlOutputs(t,x,u)
sys(1) = x(1);                          % 第一个关节角速度 dq1
sys(2) = x(2);                          % 第二个关节角速度 dq2
sys(3) = x(3);                          % 第一个关节角度 q1
sys(4) = x(4);                          % 第二个关节角度 q2
```

(4) 控制器子程序：chap5_1ctrl.m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 2;
sizes.NumInputs      = 9;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];
function sys = mdlOutputs(t,x,u)
q1d = u(1);q2d = u(2);
dq1d = u(3);dq2d = u(4);
dq1 = u(5);dq2 = u(6);
q1 = u(7);q2 = u(8);
j = u(9);

e1 = q1d - q1;
e2 = q2d - q2;
de1 = dq1d - dq1;
de2 = dq2d - dq2;

Fai = eye(2);
Kd0 = [210 0;0 210];

% Iteration number
if j == 0
    beta = 1;
else
    beta = 2 * j;
end
sys(1) = beta * 210 * (e1 + de1);
sys(2) = beta * 210 * (e2 + de2);
```

(5) 指令程序：chap5_1input.m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
```

```
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 4;
sizes.NumInputs      = 0;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];
function sys = mdlOutputs(t,x,u)
q1d = sin(3 * t);
q2d = cos(3 * t);
dq1d = 3 * cos(3 * t);
dq2d = -3 * sin(3 * t);

sys(1) = q1d;
sys(2) = q2d;
sys(3) = dq1d;
sys(4) = dq2d;
```

5.2 基于增益自适应整定的机械手迭代学习控制的改进

5.2.1 算法的改进

5.1节中定理5.1描述的控制算法的不足之处为：它针对重复性干扰，忽略了线性化残差项 $\boldsymbol{n}(\ddot{\boldsymbol{e}}^j,\dot{\boldsymbol{e}}^j,\boldsymbol{e}^j,t)$，且机械手动力学方程为确定的。

为了弥补上述不足，本节中在控制律中加入鲁棒项，实现干扰为不重复、考虑线性化残差项 $\boldsymbol{n}(\ddot{\boldsymbol{e}}^j,\dot{\boldsymbol{e}}^j,\boldsymbol{e}^j,t)$ 的不确定机械手自适应迭代学习控制。

带有非重复干扰的不确定机器人系统动力学方程为

$$(\boldsymbol{D}(\boldsymbol{q}^j(t))+\Delta\boldsymbol{D}(\boldsymbol{q}^j(t)))\ddot{\boldsymbol{q}}^j(t)+(\boldsymbol{C}(\boldsymbol{q}^j(t),\dot{\boldsymbol{q}}^j(t))+\Delta\boldsymbol{C}(\boldsymbol{q}^j(t),\dot{\boldsymbol{q}}^j(t)))\dot{\boldsymbol{q}}^j(t)+$$
$$\boldsymbol{G}(q^j(t),\dot{\boldsymbol{q}}^j(t))+\Delta\boldsymbol{G}(\boldsymbol{q}^j(t),\dot{\boldsymbol{q}}^j(t))+\boldsymbol{T}_d^j(t)=\boldsymbol{T}^j(t) \tag{5.14}$$

被控对象式(5.14)可写为

$$\boldsymbol{D}(\boldsymbol{q}^j(t))\ddot{\boldsymbol{q}}^j(t)+\boldsymbol{C}(\boldsymbol{q}^j(t),\dot{\boldsymbol{q}}^j(t))\dot{\boldsymbol{q}}^j(t)+\boldsymbol{G}(\boldsymbol{q}^j(t),\dot{\boldsymbol{q}}^j(t))+\boldsymbol{d}^j(t)=\boldsymbol{T}^j(t) \tag{5.15}$$

其中

$$\boldsymbol{d}^j(t)=\Delta\boldsymbol{D}(\boldsymbol{q}^j(t))\ddot{\boldsymbol{q}}^j(t)+\Delta\boldsymbol{C}(\boldsymbol{q}^j(t),\dot{\boldsymbol{q}}^j(t))\dot{\boldsymbol{q}}^j(t)+\Delta\boldsymbol{G}(\boldsymbol{q}^j(t),\dot{\boldsymbol{q}}^j(t))+\boldsymbol{T}_d^j(t)$$

沿着指令轨迹 $(\boldsymbol{q}_d(t),\dot{\boldsymbol{q}}_d(t),\ddot{\boldsymbol{q}}_d(t))$，采用泰勒公式，则方程式(5.15)可线性化为

$$\boldsymbol{D}(t)\ddot{\boldsymbol{e}}+[\boldsymbol{C}+\boldsymbol{C}_1]\dot{\boldsymbol{e}}+\boldsymbol{F}\boldsymbol{e}+\boldsymbol{n}(\ddot{\boldsymbol{e}},\dot{\boldsymbol{e}},\boldsymbol{e},t)=\boldsymbol{H}-(\boldsymbol{D}\ddot{\boldsymbol{q}}+\boldsymbol{C}\dot{\boldsymbol{q}}+\boldsymbol{G}) \tag{5.16}$$

其中

$$\boldsymbol{n}(\ddot{\boldsymbol{e}},\dot{\boldsymbol{e}},\boldsymbol{e},t)=-\frac{\partial \boldsymbol{D}}{\partial \boldsymbol{q}}\Big|_{\boldsymbol{q}_d}\ddot{\boldsymbol{e}}\boldsymbol{e}-\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{q}}\Big|_{\boldsymbol{q}_d,\dot{\boldsymbol{q}}_d}\dot{\boldsymbol{e}}\boldsymbol{e}-\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{q}}\Big|_{\boldsymbol{q}_d,\dot{\boldsymbol{q}}_d}\dot{\boldsymbol{e}}\dot{\boldsymbol{e}}+\boldsymbol{O}_D(\cdot)\ddot{\boldsymbol{q}}+\boldsymbol{O}_C(\cdot)\dot{\boldsymbol{q}}-\boldsymbol{O}_G(\cdot)$$

将式(5.15)代入式(5.16),则

$$\boldsymbol{D}(t)\ddot{\boldsymbol{e}}^j(t)+[\boldsymbol{C}(t)+\boldsymbol{C}_1(t)]\dot{\boldsymbol{e}}^j(t)+\boldsymbol{F}(t)\boldsymbol{e}^j(t)+\boldsymbol{n}(\ddot{\boldsymbol{e}}^j,\dot{\boldsymbol{e}}^j,\boldsymbol{e}^j,t)-\boldsymbol{d}^j(t)=\boldsymbol{H}(t)-\boldsymbol{T}^j(t) \tag{5.17}$$

取 $\boldsymbol{d}_1{}^j(t)=-\boldsymbol{n}(\ddot{\boldsymbol{e}}^j,\dot{\boldsymbol{e}}^j,\boldsymbol{e}^j,t)+\boldsymbol{d}^j(t)$,则式(5.17)可写为

$$\boldsymbol{D}(t)\ddot{\boldsymbol{e}}^j(t)+[\boldsymbol{C}(t)+\boldsymbol{C}_1(t)]\dot{\boldsymbol{e}}^j(t)+\boldsymbol{F}(t)\boldsymbol{e}^j(t)-\boldsymbol{d}_1{}^j(t)=\boldsymbol{H}(t)-\boldsymbol{T}^j(t) \tag{5.18}$$

于是

$$\boldsymbol{D}(t)\ddot{\boldsymbol{e}}^{j+1}(t)+[\boldsymbol{C}(t)+\boldsymbol{C}_1(t)]\dot{\boldsymbol{e}}^{j+1}(t)+\boldsymbol{F}(t)\boldsymbol{e}^{j+1}(t)-\boldsymbol{d}_1{}^{j+1}(t)=\boldsymbol{H}(t)-\boldsymbol{T}^{j+1}(t) \tag{5.19}$$

设计鲁棒迭代学习控制律为

$$\boldsymbol{T}^j(t)=\boldsymbol{K}_p^j\boldsymbol{e}^j(t)+\boldsymbol{K}_d^j\dot{\boldsymbol{e}}^j(t)+\boldsymbol{T}^{j-1}(t)+E\,\mathrm{sgn}(\delta\boldsymbol{y}^{j-1}),\quad j=0,1,2,\cdots,N \tag{5.20}$$

其中,$\|\boldsymbol{d}_1^{j+1}(t)-\boldsymbol{d}_1^j(t)\|=\|\delta\boldsymbol{d}_1{}^j(t)\|\leqslant E$。

证明:收敛性分析的证明过程可参照 5.1 节的推导过程。

5.2.2 仿真实例

针对双关节机械手动态方程式(5.1)进行仿真,取不可重复的干扰为 $d_1(t)=a0.3\sin t$,$\boldsymbol{d}_2(t)=a0.1(1-\mathrm{e}^{-t})$,$a=1$ 为幅值为 1 的随机噪声,$\boldsymbol{T}_a=[d_1\quad d_2]^{\mathrm{T}}$。

角度期望轨迹 $q_1=\sin 3t$,$q_2=\cos 3t$,取 $\boldsymbol{\Lambda}=\begin{bmatrix}1&0\\0&1\end{bmatrix}$,控制器取式(5.20),控制参数取 $E=1.0$,$\boldsymbol{K}_p^0=\boldsymbol{K}_d^0=\begin{bmatrix}210&0\\0&210\end{bmatrix}$,$\beta(j)=2j$,$\boldsymbol{K}_p^j=2j\boldsymbol{K}_p^0$,$\boldsymbol{K}_d^j=2j\boldsymbol{K}_d^0$,$j=1,2,\cdots,N$。

系统的初始状态为 $\boldsymbol{x}=[3\quad 0\quad 0\quad 1]^{\mathrm{T}}$,迭代次数取 15 次。仿真结果如图 5.6 至图 5.10 所示。

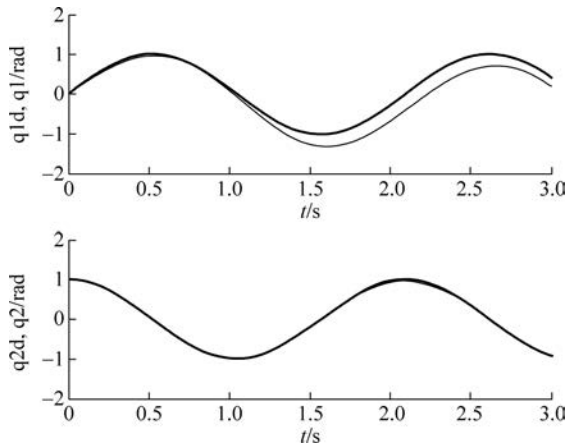


图 5.6 双关节 15 次迭代的角度跟踪过程

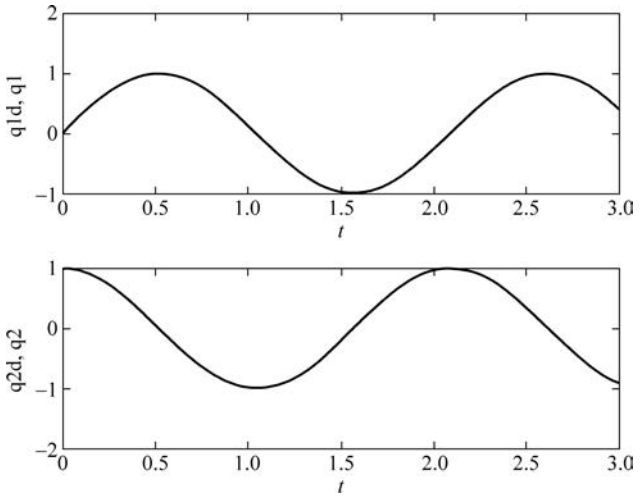


图 5.7　双关节第 15 次的角度跟踪

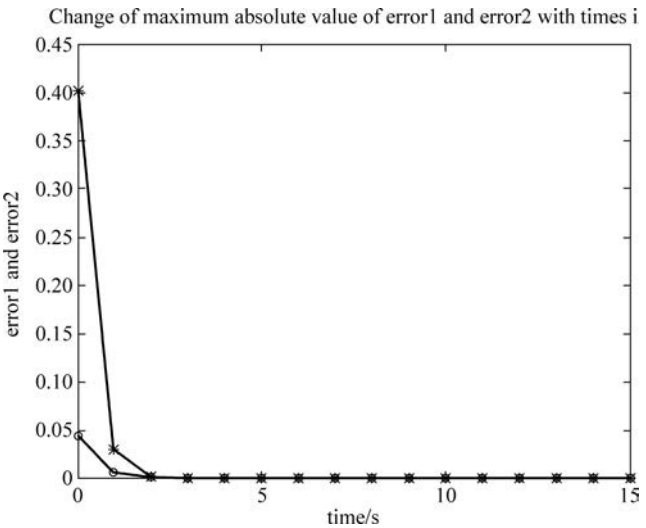


图 5.8　双关节 15 次角度跟踪的误差收敛过程

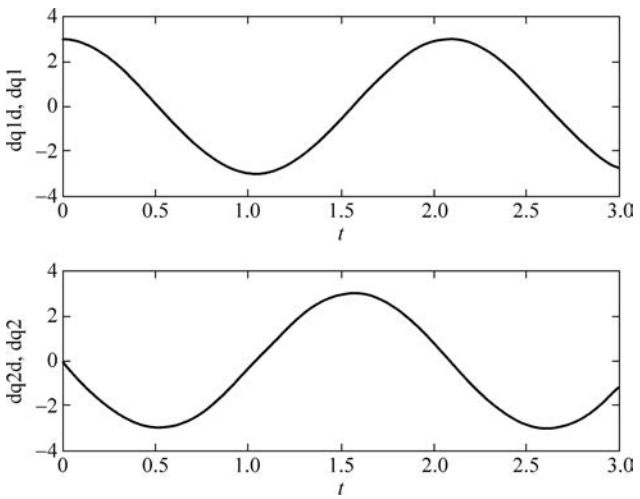


图 5.9　双关节第 15 次的角速度跟踪

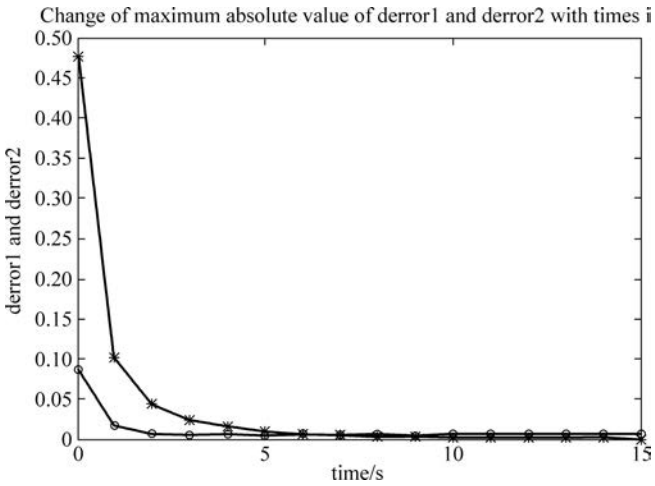


图 5.10　双关节 15 次角速度跟踪的误差收敛过程

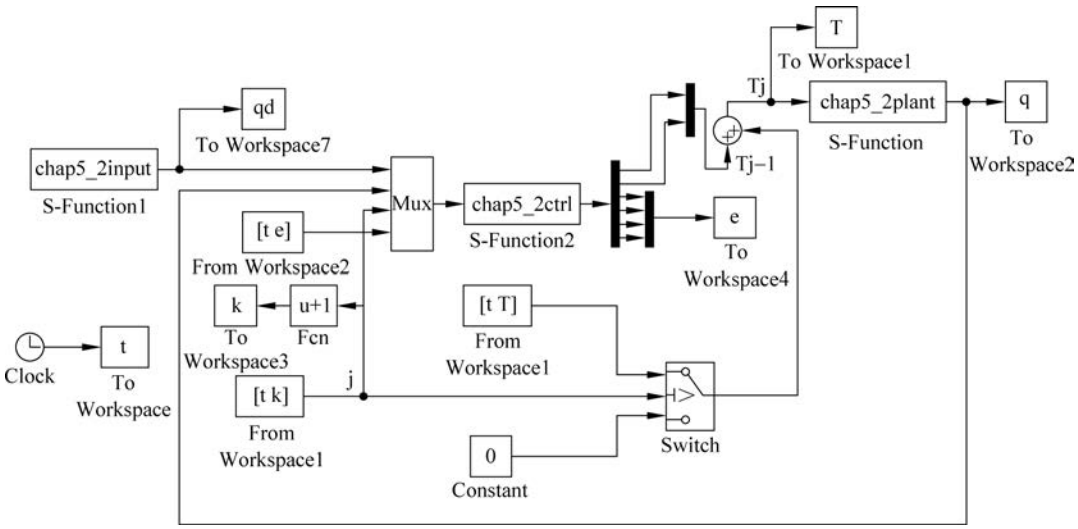
仿真程序：

(1) 主程序：chap5_2main.m。

```
clear all;
close all;
L = 3001;

t = [0:0.001:3]';
T1(1:L) = 0;
T1 = T1';
T2 = T1;
T = [T1 T2];

e1(1:L) = 0;
e1 = e1';
e2 = e1;
de1 = e1;
de2 = de1;
e = [e1 e2 de1 de2];

k(1:L) = 0;
k = k';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M = 15;
for i = 0:1:M
i
pause(0.01);

sim('chap5_2sim',[0,3]);

q1 = q(:,3);
q2 = q(:,4);
dq1 = q(:,1);
dq2 = q(:,2);
q1d = qd(:,1);
```

```
q2d = qd(:,2);
dq1d = qd(:,3);
dq2d = qd(:,4);
e1 = q1d - q1;
e2 = q2d - q2;
de1 = dq1d - dq1;
de2 = dq2d - dq2;

figure(1);
subplot(211);
hold on;
plot(t,q1,'b',t,q1d,'r');
xlabel('t (s)');ylabel('q1d,q1 (rad)');

subplot(212);
hold on;
plot(t,q2,'b',t,q2d,'r');
xlabel('t (s)');ylabel('q2d,q2 (rad)');

j = i + 1;
times(j) = i;
e1i(j) = max(abs(e1));
e2i(j) = max(abs(e2));
de1i(j) = max(abs(de1));
de2i(j) = max(abs(de2));
end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(2);
subplot(211);
plot(t,q1d,'r',t,q1,'b');
xlabel('t');ylabel('q1d,q1');
subplot(212);
plot(t,q2d,'r',t,q2,'b');
xlabel('t');ylabel('q2d,q2');

figure(3);
plot(times,e1i,'* - r',times,e2i,'o - b');
title('Change of maximum absolute value of error1 and error2 with times i');
xlabel('times');ylabel('error1 and error2');

figure(4);
subplot(211);
plot(t,dq1d,'r',t,dq1,'b');
xlabel('t');ylabel('dq1d,dq1');
subplot(212);
plot(t,dq2d,'r',t,dq2,'b');
xlabel('t');ylabel('dq2d,dq2');

figure(5);
plot(times,de1i,'* - r',times,de2i,'o - b');
title('Change of maximum absolute value of derror1 and derror2 with times i');
xlabel('times');ylabel('derror1 and derror2');
```

（2）Simulink 子程序：chap5_2sim. mdl。



（3）被控对象子程序：chap5_2plant. m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 1,
    sys = mdlDerivatives(t,x,u);
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 4;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 4;
sizes.NumInputs      = 2;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;
sys   = simsizes(sizes);
x0    = [3;0;0;1];
str   = [];
ts    = [0 0];
function sys = mdlDerivatives(t,x,u)
% a = 1;
a = rands(1);

d1 = a * 0.3 * sin(3 * pi * t);
d2 = a * 0.1 * (1 - exp( - t));

dq1 = x(1);      % 第一个关节角速度 dq1
dq2 = x(2);      % 第二个关节角速度 dq2
q1 = x(3);       % 第一个关节角度 q1
```

```
q2 = x(4);      % 第二个关节角度 q2

tol1 = u(1);
tol2 = u(2);

m1 = 10;m2 = 5;
l1 = 1;l2 = 0.5;
r1 = 0.5;r2 = 0.25;
i1 = 0.83 + m1 * r1^2 + m2 * l1^2;
i2 = 0.3 + m2 * r2^2;
g = 9.8;

D = [i1 + i2 + 2 * m2 * r2 * l1 * cos(q2) i2 + m2 * r2 * l1 * cos(q2);
   i2 + m2 * r2 * l1 * cos(q2) i2];
C = [ - m2 * r2 * l1 * dq2 * sin(q2), - m2 * r2 * l1 * (dq1 + dq2) * sin(q2);
    m2 * r2 * l1 * dq1 * sin(q2),0];
G = [(m1 * r1 + m2 * l1) * g * cos(q1) + m2 * r2 * g * cos(q1 + q2);m2 * r2 * g * cos(q1 + q2)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%

D2 = inv(D);
Ta = [d1;d2];
A = - D2 * C;
Z = - D2 * G;

sys(1) = A(1,1) * x(1) + A(1,2) * x(2) + Z(1) + D2(1,1) * ( - Ta(1) + tol1) + D2(1,2) * ( - Ta(2) + tol2);
sys(2) = A(2,1) * x(1) + A(2,2) * x(2) + Z(2) + D2(2,1) * ( - Ta(1) + tol1) + D2(2,2) * ( - Ta(2) + tol2);
sys(3) = x(1);
sys(4) = x(2);
function sys = mdlOutputs(t,x,u)
sys(1) = x(1);                          % 第一个关节角速度 dq1
sys(2) = x(2);                          % 第二个关节角速度 dq2
sys(3) = x(3);                          % 第一个关节角度 q1
sys(4) = x(4);                          % 第二个关节角度 q2
```

(4) 控制器子程序：chap5_2ctrl.m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 6;
sizes.NumInputs      = 13;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];
```

```
function sys = mdlOutputs(t,x,u)
q1d = u(1);q2d = u(2);
dq1d = u(3);dq2d = u(4);
dq1 = u(5);dq2 = u(6);
q1 = u(7);q2 = u(8);
j = u(9);

e1 = q1d - q1;
e2 = q2d - q2;
de1 = dq1d - dq1;
de2 = dq2d - dq2;

e1_1 = u(10);
e2_1 = u(11);
de1_1 = u(12);
de2_1 = u(13);

Fai = eye(2);
Kd0 = [210 0;0 210];

if j == 0
    betaj = 1;
else
    betaj = 2 * j;
end
E = 1.0;
delta_y = [de1 - de1_1 + e1 - e1_1;de2 - de2_1 + e2 - e2_1];

Kp0 = [210 0;0 210];
Kd0 = Kp0;
Kpj = betaj * Kp0;
Kdj = betaj * Kd0;
ej = [e1 e2]';
dej = [de1 de2]';

Deltaj = [E * sign(delta_y(1)) E * sign(delta_y(1))]';
Tj = Kpj * ej + Kdj * dej + Deltaj;

sys(1) = Tj(1);
sys(2) = Tj(2);
sys(3) = e1;
sys(4) = e2;
sys(5) = de1;
sys(6) = de2;
```

(5) 指令程序：chap5_2input.m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
```

```
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 4;
sizes.NumInputs      = 0;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];
function sys = mdlOutputs(t,x,u)
q1d = sin(3 * t);
q2d = cos(3 * t);
dq1d = 3 * cos(3 * t);
dq2d = - 3 * sin(3 * t);

sys(1) = q1d;
sys(2) = q2d;
sys(3) = dq1d;
sys(4) = dq2d;
```

5.3 基于切换增益的单关节机械手迭代学习控制

文献[5]针对多关节机械手提出了一种自适应迭代学习控制方法，该方法主要在经典的PD反馈控制的基础上通过迭代项克服机器人系统的未知参数和干扰带来的不确定性。本节在文献[5]的基础上，设计了单关节机械手的迭代学习控制算法，并给出了该方法的收敛性分析及仿真结果。

5.3.1 问题描述

单关节机器人系统的动力学模型为

$$I\ddot{\theta}_k + b\dot{\theta}_k + mgl\cos\theta_k = \tau_k + d_k(t) \tag{5.21}$$

其中，k 为正整数，表示迭代次数，θ_k、$\dot{\theta}_k$ 和 $\ddot{\theta}_k$ 分别代表关节第 k 次运行时的角度、角速度和角加速度，I 为转动惯量，b 为关节转动的粘性摩擦系数，m 为关节的质量，l 为关节的质心与关节的距离，$d_k(t)$为单关节机器人系统参数不确性项和干扰，τ_k 为作用于关节的控制输入。

假定关节的位置和速度可以通过反馈获得，则控制的任务就是设计一个控制律 $\tau_k(t)$使得 $\theta_k(t)$在任意 $t\in[0,T]$及任意 k 都有界，并且当 $k\to\infty$时 $\theta_k(t)$在任意时刻 $t\in[0,T]$都收敛于对应时刻的期望轨迹 $\theta_d(t)$。

根据上述描述，可以给出如下合理的假设：

假设 1：期望轨迹 $\theta_d(t)$及其一阶、二阶导数 $\dot{\theta}_d(t)$、$\ddot{\theta}_d(t)$和 $d_k(t)$有界；

假设 2：初始状态可重复，即 $\dot{\theta}_d(0)-\dot{\theta}_k(0)=\theta_d(0)-\theta_k(0)=0$；

假设 3：$|I\ddot{\theta}_{dk}-d_k|\leqslant\beta$，$|mgl\cos\theta|\leqslant k_g$，$b\leqslant k_c$；

假设 4：假设速度有界[8]，取$|\dot{\theta}_k|\leqslant M$。

5.3.2 自适应迭代学习控制器设计

考虑式(5.21)及假设 1～3,控制律设计为[5]

$$\tau_k = k_p e_k(t) + k_d \dot{e}_k(t) + \hat{\delta}_k(t)\mathrm{sgn}(\dot{e}_k(t)) \tag{5.22}$$

其中

$$\hat{\delta}_k(t) = \hat{\delta}_{k-1}(t) + \gamma\,|\dot{e}_k(t)\,| \tag{5.23}$$

且 $\hat{\delta}_{-1}(t)=0$,$e_k(t)=\theta_d(t)-\theta_k(t)$。如果 k_p,k_d,γ 均大于零,则 $e_k(t)$、$\dot{e}_k(t)$及 $\tau_k(t)$对于任意 k 都有界,且 $\lim\limits_{k\to\infty} e_k(t)=\lim\limits_{k\to\infty}\dot{e}_k(t)=0$,$t\in[0 \quad T]$。

为了简化起见,在下面的描述中有些变量省略了(t)。

5.3.3 收敛性分析

收敛性分析步骤如下。

1. 证明 W_k 的递增性

取如下的 Lyapunov 函数

$$W_k(t) = V_k(t) + \frac{1}{2\gamma}\int_0^t \tilde{\delta}_k^2(\tau)\mathrm{d}\tau \tag{5.24}$$

其中,δ 为用于描述模型信息的不确定项。

定义 $\delta=\beta+k_c M+k_g$,为了实现无须模型信息的控制,采用 $\hat{\delta}_k(t)$自适应估计 δ,并取 $\tilde{\delta}_k(t)=\delta-\hat{\delta}_k(t)$。定义 $V_k(t)$为

$$V_k(t) = \frac{1}{2}I\dot{e}_k^2(t) + \frac{1}{2}k_p e_k^2(t) \tag{5.25}$$

则

$$\begin{aligned}\Delta W_k &= W_k - W_{k-1} = V_k - V_{k-1} + \frac{1}{2\gamma}\int_0^t (\tilde{\delta}_k^2(\tau) - \tilde{\delta}_{k-1}^2(\tau))\mathrm{d}\tau \\ &= V_k - V_{k-1} - \frac{1}{2\gamma}\int_0^t (\bar{\delta}_k^2 + 2\tilde{\delta}_k\bar{\delta}_k)\mathrm{d}\tau\end{aligned} \tag{5.26}$$

其中,$\bar{\delta}_k=\hat{\delta}_k-\hat{\delta}_{k-1}$。

对 $V_k(t)$求一阶导数,然后对两边积分可得

$$V_k(t) = V_k(0) + \int_0^t (I\dot{e}_k\ddot{e}_k + k_p e_k\dot{e}_k)\mathrm{d}\tau \tag{5.27}$$

根据假设 2 可知 $V_k(0)=0$,利用式(5.21)及假设 1-4,可得

$$\begin{aligned}V_k(t) &= \int_0^t \dot{e}_k(I\ddot{\theta}_{dk} - I\ddot{\theta}_k + k_p e_k)\mathrm{d}\tau \\ &= \int_0^t \dot{e}_k(I\ddot{\theta}_{dk} + b\dot{\theta}_k + mgl\cos\theta_k - \tau_k - d_k + k_p e_k)\mathrm{d}\tau \\ &\leqslant \int_0^t \delta\,|\dot{e}_k\,|\,\mathrm{d}\tau + \int_0^t \dot{e}_k(-\tau_k + k_p e_k)\mathrm{d}\tau\end{aligned}$$

其中，$|\ddot{I\theta}_{dk}-d_k+mgl\cos\theta_k+b\dot{\theta}_k|\leqslant\beta+k_g+k_cM\leqslant\delta$。

将控制律式(5.22)代入上式，可得

$$\begin{aligned}V_k(t)&\leqslant\int_0^t\delta\,|\dot{e}_k|\,\mathrm{d}\tau+\int_0^t\dot{e}_k(-k_d\dot{e}_k-\hat{\delta}_k(t)\mathrm{sgn}(\dot{e}_k(t)))\mathrm{d}\tau\\&=\int_0^t\delta\,|\dot{e}_k|\,\mathrm{d}\tau+\int_0^t(-k_d\dot{e}_k^2-\hat{\delta}_k\,|\dot{e}_k|)\mathrm{d}\tau\end{aligned}\tag{5.28}$$

由式(5.23)可得

$$|\dot{e}_k|=\frac{1}{\gamma}(\hat{\delta}_k-\hat{\delta}_{k-1})=\frac{1}{\gamma}\bar{\delta}_k$$

则

$$V_k\leqslant-k_d\int_0^t\dot{e}_k^2\mathrm{d}\tau+\frac{1}{\gamma}\int_0^t\delta\bar{\delta}_k\mathrm{d}\tau-\frac{1}{\gamma}\int_0^t\hat{\delta}_k\bar{\delta}_k\mathrm{d}\tau=-k_d\int_0^t\dot{e}_k^2\mathrm{d}\tau+\frac{1}{\gamma}\int_0^t\tilde{\delta}_k\bar{\delta}_k\mathrm{d}\tau$$

将上式代入式(5.26)，可得

$$\begin{aligned}\Delta W_k&\leqslant-k_d\int_0^t\dot{e}_k^2\mathrm{d}\tau+\frac{1}{\gamma}\int_0^t\tilde{\delta}_k\bar{\delta}_k\mathrm{d}\tau-V_{k-1}-\frac{1}{2\gamma}\int_0^t(\bar{\delta}_k^2+2\tilde{\delta}_k\bar{\delta}_k)\mathrm{d}\tau\\&=-V_{k-1}-\frac{1}{2\gamma}\int_0^t\bar{\delta}_k^2\mathrm{d}\tau-k_d\int_0^t\dot{e}_k^2\mathrm{d}\tau\leqslant0\end{aligned}\tag{5.29}$$

上式说明 W_k 是非增序列，现只要证明 W_0 有界就可说明 W_k 是有界的。

2. 证明 $W_0(t)$ 的有界性

根据式(5.24)的定义，有

$$W_0(t)=V_0(t)+\frac{1}{2\gamma}\int_0^t\tilde{\delta}_0^2(\tau)\mathrm{d}\tau$$

则

$$\dot{W}_0(t)=\dot{V}_0(t)+\frac{1}{2\gamma}\tilde{\delta}_0^2(t)\tag{5.30}$$

根据式(5.27)，有 $V_0(t)=\int_0^t(I\dot{e}_0\ddot{e}_0+k_pe_0\dot{e}_0)\mathrm{d}\tau$，由式(5.21)可得

$$I\ddot{\theta}_0=\tau_0+d_0-b\dot{\theta}_0-mgl\cos\theta_0=k_pe_0+k_d\dot{e}_0+\hat{\delta}_0\mathrm{sgn}(\dot{e}_0)+d_0-b\dot{\theta}_0-mgl\cos\theta_0$$

则

$$\begin{aligned}\dot{V}_0(t)&=I\dot{e}_0\ddot{e}_0+k_pe_0\dot{e}_0=\dot{e}_0(I\ddot{e}_0+k_pe_0)=\dot{e}_0(I\ddot{\theta}_d-I\ddot{\theta}_0+k_pe_0)\\&=\dot{e}_0(I\ddot{\theta}_d-k_d\dot{e}_0-\hat{\delta}_0\mathrm{sgn}(\dot{e}_0)-d_0+b\dot{\theta}_0+mgl\cos\theta_0)\\&\leqslant\delta\,|\dot{e}_0|-k_d\dot{e}_0^2-\hat{\delta}_0\,|\dot{e}_0|\end{aligned}$$

由式(5.23)可知 $\hat{\delta}_0(t)=\gamma|\dot{e}_0(t)|$，将上式代入式(5.30)，可得

$$\begin{aligned}\dot{W}_0(t)&\leqslant\delta\,|\dot{e}_0|-k_d\dot{e}_0^2-\hat{\delta}_0\,|\dot{e}_0|+\frac{1}{2\gamma}\tilde{\delta}_0^2=-k_d\dot{e}_0^2+\tilde{\delta}_0\,|\dot{e}_0|+\frac{1}{2\gamma}\tilde{\delta}_0^2\\&=-k_d\dot{e}_0^2+\frac{1}{\gamma}\tilde{\delta}_0\hat{\delta}_0+\frac{1}{2\gamma}\tilde{\delta}_0^2=-k_d\dot{e}_0^2+\frac{1}{\gamma}\tilde{\delta}_0\left(\hat{\delta}_0+\frac{1}{2}\tilde{\delta}_0\right)\end{aligned}$$

由 $\hat{\delta}_0(t)=\delta-\tilde{\delta}_0(t)$，代入上式可得

$$\dot{W}_0(t) \leqslant -k_d \dot{e}_0^2 + \frac{1}{\gamma}\tilde{\delta}_0\delta - \frac{1}{2\gamma}\tilde{\delta}_0^2 \tag{5.31}$$

对于 $\lambda>0$，有如下不等式

$$\frac{1}{\gamma}\tilde{\delta}_0(t)\delta \leqslant \lambda\left(\frac{1}{\gamma}\tilde{\delta}_0(t)\right)^2 + \frac{1}{4\lambda}\delta^2 \tag{5.32}$$

成立，则可得

$$\begin{aligned}\dot{W}_0 &\leqslant -k_d \dot{e}_0^2 + \lambda\left(\frac{1}{\gamma}\tilde{\delta}_0(t)\right)^2 + \frac{1}{4\lambda}\delta^2 - \frac{1}{2\gamma}\tilde{\delta}_0^2(t)\\ &= -k_d \dot{e}_0^2 + \frac{1}{\gamma}\left(\frac{\lambda}{\gamma} - \frac{1}{2}\right)\tilde{\delta}_0^2(t) + \frac{1}{4\lambda}\delta^2\end{aligned} \tag{5.33}$$

取 $\frac{\lambda}{\gamma} - \frac{1}{2} \leqslant 0$，即 $\lambda \leqslant \frac{1}{2}\gamma$，则

$$\dot{W}_0 \leqslant -k_d \dot{e}_0^2 + \frac{1}{4\lambda}\delta^2 \leqslant \frac{1}{4\lambda}\delta^2$$

由式(5.34)，根据一致连续性判定定理(见附录)，若函数 W_0 在区间[0，T]上的导数有界，则 W_0 在[0，T]上一致连续，再根据闭区间上连续函数的有界定理(见附录)，W_0 在[0，T]上有界。

3. 证明误差的收敛性

由 W_0 在[0，T]上有界可知 W_k 有界，W_k 可写为

$$W_k = W_0 + \sum_{j=1}^{k}\Delta W_j \tag{5.34}$$

由式(5.29)可得 $\Delta W_k \leqslant -V_{k-1}$，则根据式(5.25)中 V_k 的定义，可得

$$\Delta W_k \leqslant -\frac{1}{2}I\dot{e}_{k-1}^2 - \frac{1}{2}k_p e_{k-1}^2$$

则

$$\sum_{j=1}^{k}\Delta W_j \leqslant -\frac{1}{2}\sum_{j=1}^{k}(I\dot{e}_{j-1}^2 + k_p e_{j-1}^2)$$

从而

$$W_k \leqslant W_0 - \frac{1}{2}\sum_{j=1}^{k}(I\dot{e}_{j-1}^2 + k_p e_{j-1}^2) \tag{5.35}$$

由上式可推出

$$\sum_{j=1}^{k}(I\dot{e}_{j-1}^2 + k_p e_{j-1}^2) \leqslant 2(W_0 - W_k) \leqslant 2W_0$$

由于 W_0 有界，则

$$\lim_{k\to\infty} e_k(t) = \lim_{k\to\infty}\dot{e}_k(t) = 0, \quad t \in [0 \quad T]$$

5.3.4 仿真实例

假设单力臂的质量均匀分布，质心距连杆的转动中心为 l，连杆运动的黏性摩擦系数为 b，并忽略弹性摩擦，外加干扰为 d_k，则根据牛顿定律得到其运动方程为

$$I\ddot{\theta}+b\dot{\theta}+mgl\cos\theta=\tau+d_k$$

其中，mg 为重力，I 为转动惯量，θ 为转动角度。

单自由度机械臂的系统参数为：机械臂质量 $m=1$，质心到关节的长度 $l=0.25$，转动惯量 $I=\frac{4}{3}ml^2$，重力加速度 $g=9.8$，黏性摩擦系数 $b=2.0$，取干扰和系统不确定项的总和为 $d_k=\sin t$。

控制器参数选为 $k_p=5.0$，$k_d=5.0$，$\gamma=20$。总的迭代次数为 20 次，每次的仿真时间为 1。仿真中，为了提高控制精度，Simulink 模块的仿真参数中的绝对误差和相对误差都取 10^{-5}。角度指令信号为 $\theta_d=\sin t$。采用控制律式(5.22)和自适应律式(5.23)，仿真程序中，取 $S=1$，仿真结果如图 5.11～图 5.15 所示。在控制律中，为了降低控制输入的抖振，并加快运算速度，可采用饱和函数代替切换函数，并取边界层厚度为 0.02，见仿真程序中的 $S=2$。

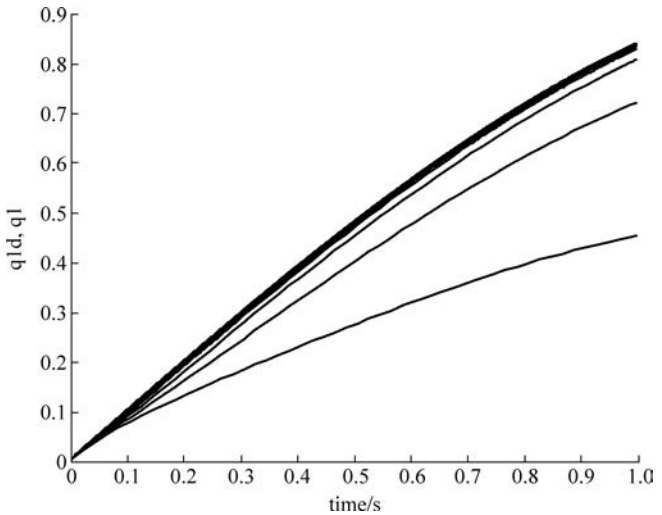


图 5.11　50 次迭代过程中角度的跟踪过程

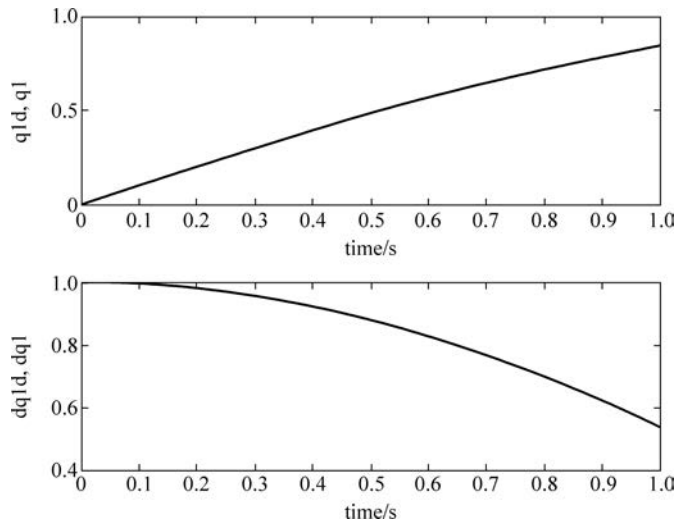


图 5.12　50 次迭代后角度和角速度跟踪

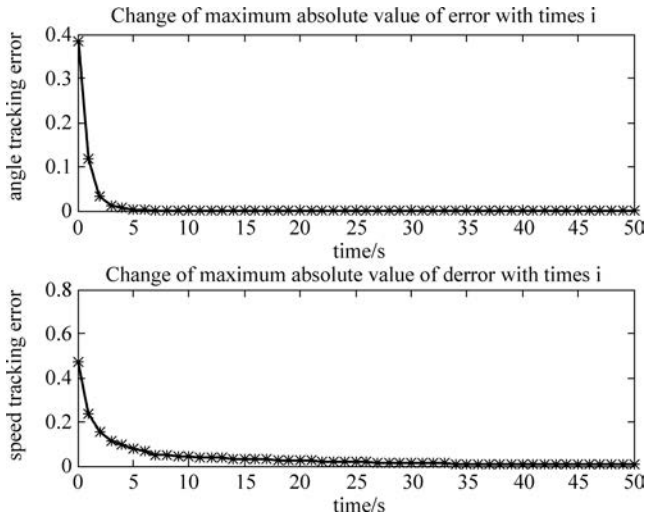


图 5.13　50 次迭代过程中误差和误差变化率的收敛过程

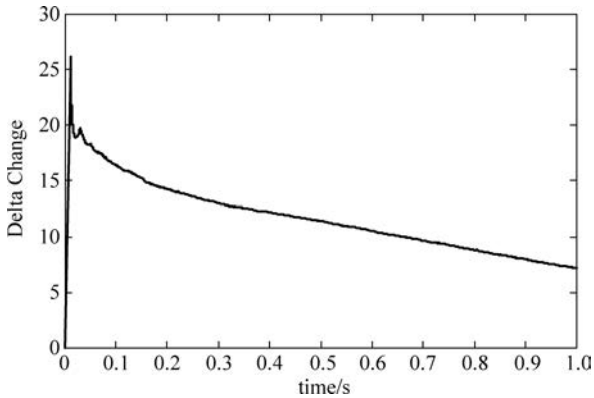


图 5.14　第 50 次迭代自适应项 δ 的变化

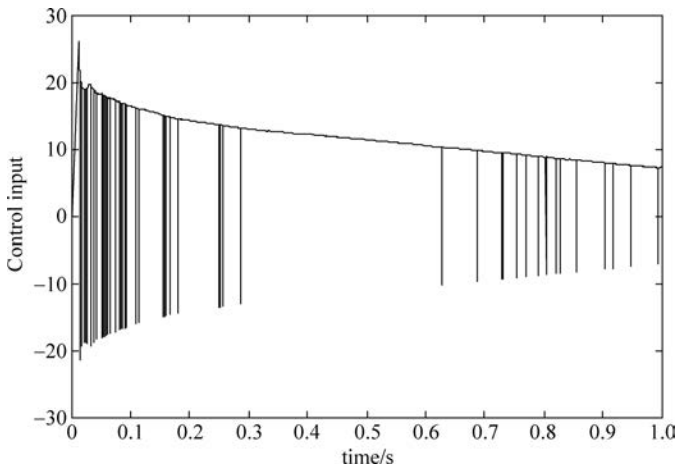


图 5.15　第 50 次迭代控制输入的变化

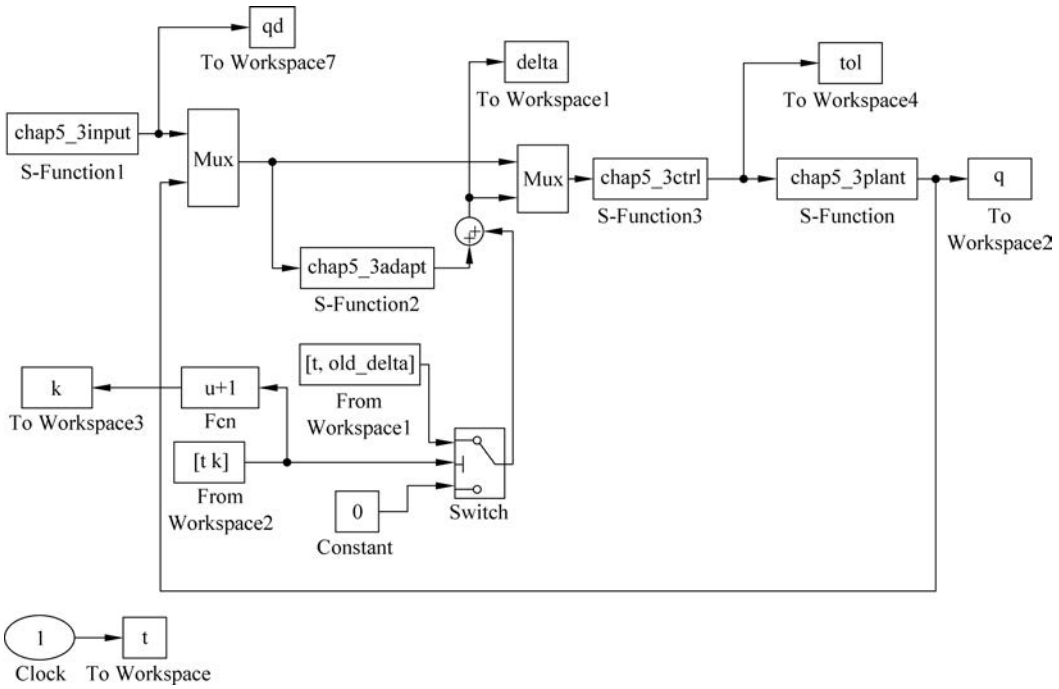
仿真程序：针对单力臂机械手的自适应迭代学习控制。

（1）主程序：chap5_3main.m。

```
clc;
clear all;
close all;

global old_delta
t = [0:0.01:1]';
k(1:101) = 0;
k = k';
delta(1:101) = 0;
delta = delta';

M = 50;
for i = 0:1:M
i
pause(0.01);

old_delta = delta;
sim('chap5_3sim',[0,1]);

q1 = q(:,1);
dq1 = q(:,2);
q1d = qd(:,1);
dq1d = qd(:,2);
e = q1d - q1;
de = dq1d - dq1;

figure(1);
hold on;
plot(t,q1,'b',t,q1d,'r');
xlabel('time(s)');ylabel('q1d,q1');

j = i + 1;
times(j) = i;

ei(j) = max(abs(e));
dei(j) = max(abs(de));
end

figure(2);
subplot(211);
plot(t,q1d,'r',t,q1,'b');
xlabel('time(s)');ylabel('q1d,q1');
subplot(212);
plot(t,dq1d,'r',t,dq1,'b');
xlabel('time(s)');ylabel('dq1d,dq1');

figure(3);
subplot(211);
plot(times,ei,'* - r');
title('Change of maximum absolute value of error with times i');
xlabel('times');ylabel('angle tracking error');
subplot(212);
```

```
plot(times,dei,'* - r');
title('Change of maximum absolute value of derror with times i');
xlabel('times');ylabel('speed tracking error');

figure(4);
plot(t,delta,'r');
xlabel('time(s)');ylabel('Delta Change');
figure(5);
plot(t,tol,'r');
xlabel('time(s)');ylabel('Control input');
```

(2) Simulink 子程序：chap5_3sim. mdl。



(3) 控制器子程序：chap5_3ctrl. m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 1;
sizes.NumInputs      = 5;
sizes.DirFeedthrough = 1;
```

```
sizes.NumSampleTimes  = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];
function sys = mdlOutputs(t,x,u)
q1d = u(1);dq1d = u(2);
q1 = u(3);dq1 = u(4);

e = q1d - q1;
de = dq1d - dq1;

Kp = 5.0;Kd = 5.0;

delta = u(5);
S = 2;
if S == 1
    tol = Kp * e + Kd * de + delta * sign(de);
elseif S == 2
    fai = 0.02;
    if de/fai > 1
       sat = 1;
    elseif abs(de/fai)<= 1
       sat = de/fai;
    elseif de/fai <- 1
       sat = -1;
    end
       tol = Kp * e + Kd * de + delta * sat;
end
sys(1) = tol;
```

（4）自适应律子程序：chap5_3adapt.m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 1;
sizes.NumInputs      = 4;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];
```

```
function sys = mdlOutputs(t,x,u)
q1d = u(1);dq1d = u(2);
q1 = u(3);dq1 = u(4);

de = dq1d - dq1;

Gama = 20;
delta = Gama * de' * sign(de); % Adaptive law
sys(1) = delta;
```

(5) 被控对象子程序：chap5_3plant.m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 1,
    sys = mdlDerivatives(t,x,u);
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 2;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 2;
sizes.NumInputs      = 1;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [0;1];
str = [];
ts  = [0 0];
function sys = mdlDerivatives(t,x,u)
g = 9.8;
m = 1.0;
l = 1.25;
b = 2.0;
I = 4/3 * m * l^2;

q = x(1);
dq = x(2);

tol = u(1);
dk = sin(t);
S = inv(I) * (tol + dk - m * g * l * cos(q) - b * dq);

sys(1) = x(2);
sys(2) = S;
function sys = mdlOutputs(t,x,u)
sys(1) = x(1);                                  % 角度
sys(2) = x(2);                                  % 角速度
```

（6）指令子程序：chap5_3input.m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 2;
sizes.NumInputs      = 0;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];
function sys = mdlOutputs(t,x,u)
q1d = sin(t);
dq1d = cos(t);

sys(1) = q1d;
sys(2) = dq1d;
```

5.4 基于切换增益的多关节机械手迭代学习控制

本节介绍一类多关节机器人力臂自适应迭代学习控制器的设计和仿真方法。

5.4.1 问题的提出

多关节机器人力臂动力学方程为

$$\boldsymbol{M}(\boldsymbol{q}_k(t))\ddot{\boldsymbol{q}}_k(t)+\boldsymbol{C}(\boldsymbol{q}_k(t),\dot{\boldsymbol{q}}_k(t))\dot{\boldsymbol{q}}_k(t)+\boldsymbol{G}(\boldsymbol{q}_k(t))=\boldsymbol{\tau}_k(t)+\boldsymbol{d}_k(t) \tag{5.36}$$

其中，$\boldsymbol{q}_k\in\mathbf{R}^n$，$\dot{\boldsymbol{q}}_k\in\mathbf{R}^n$，$\ddot{\boldsymbol{q}}_k\in\mathbf{R}^n$ 为关节角位移，角速度和角加速度量，$\boldsymbol{M}(\boldsymbol{q}_k)\in\mathbf{R}^{n\times n}$ 为机器人的惯性矩阵，$\boldsymbol{C}(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k)\dot{\boldsymbol{q}}_k\in\mathbf{R}^n$ 表示离心力和哥氏力，$\boldsymbol{G}(\boldsymbol{q}_k)\in\mathbf{R}^n$ 为重力项，$\boldsymbol{\tau}\in\mathbf{R}^n$ 为控制力矩，$\boldsymbol{d}_k\in\mathbf{R}^n$ 为各种未建模动态和扰动。

假设系统参数未知，且系统满足如下假设：

（A1）对于 $\forall t\in[0,T]$，指令轨迹 $\boldsymbol{q}_d(t)$，$\dot{\boldsymbol{q}}_d(t)$，$\ddot{\boldsymbol{q}}_d(t)$及干扰 $\boldsymbol{q}_d(t)$有界；

（A2）初始值满足 $\dot{\boldsymbol{q}}_d(0)-\dot{\boldsymbol{q}}_k(0)=\boldsymbol{q}_d(0)-\boldsymbol{q}_k(0)=0$。

且满足一般机器人模型所具有的如下 4 个特性：

（P1）$\boldsymbol{M}(\boldsymbol{q}_k)\in\mathbf{R}^{n\times n}$ 为对称正定且有界的矩阵；

（P2）$\dot{\boldsymbol{M}}(\boldsymbol{q}_k)-2\boldsymbol{C}(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k)$为对称矩阵，且 $\boldsymbol{x}^{\mathrm{T}}(\dot{\boldsymbol{M}}(\boldsymbol{q}_k)-2\boldsymbol{C}(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k))\boldsymbol{x}=0,\boldsymbol{x}\in\mathbf{R}^n$；

(P3) $\boldsymbol{G}(\boldsymbol{q}_k)+\boldsymbol{C}(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k)\dot{\boldsymbol{q}}_{\mathrm{d}}(t)=\boldsymbol{\Psi}(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k)\boldsymbol{\xi}^{\mathrm{T}}(t)$，$\boldsymbol{\Psi}(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k)\in\mathbf{R}^{n\times(m-1)}$ 为已知矩阵，$\boldsymbol{\xi}(t)\in\mathbf{R}^{m-1}$ 为未知向量；

(P4) $\|\boldsymbol{C}(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k)\|\leqslant k_{\mathrm{c}}\|\dot{\boldsymbol{q}}_k\|$，$\|\boldsymbol{G}(\boldsymbol{q}_k)\|<k_{\mathrm{g}}$，$t\in[0,T]$，$k_{\mathrm{c}}$ 和 k_{g} 为正的实数。

5.4.2 三个定理及收敛性分析

定理 5.2 针对系统方程式(5.36)，如果采用控制律

$$\boldsymbol{\tau}_k(t)=\boldsymbol{K}_{\mathrm{P}}\tilde{\boldsymbol{q}}_k(t)+\boldsymbol{K}_{\mathrm{D}}\dot{\tilde{\boldsymbol{q}}}_k(t)+\boldsymbol{\varphi}(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\dot{\tilde{\boldsymbol{q}}}_k)\hat{\boldsymbol{\theta}}_k(t) \tag{5.37}$$

$$\hat{\boldsymbol{\theta}}_k(t)=\hat{\boldsymbol{\theta}}_{k-1}(t)+\boldsymbol{\Gamma}\boldsymbol{\varphi}^{\mathrm{T}}(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\dot{\tilde{\boldsymbol{q}}}_k)\dot{\tilde{\boldsymbol{q}}}_k(t) \tag{5.38}$$

其中，$\hat{\boldsymbol{\theta}}_{k-1}(t)=0$，$\tilde{\boldsymbol{q}}_k(t)=\boldsymbol{q}_{\mathrm{d}}(t)-\boldsymbol{q}_k(t)$，$\dot{\tilde{\boldsymbol{q}}}_k(t)=\dot{\boldsymbol{q}}_{\mathrm{d}}(t)-\dot{\boldsymbol{q}}_k(t)$，$\varphi(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\dot{\tilde{\boldsymbol{q}}}_k)\in\mathbf{R}^{n\times n}$，且 $\varphi(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k,\dot{\tilde{\boldsymbol{q}}}_k)\triangleq[\boldsymbol{\psi}(\boldsymbol{q}_k,\dot{\boldsymbol{q}}_k)\mathrm{sgn}(\dot{\tilde{\boldsymbol{q}}}_k)]$，矩阵 $\boldsymbol{K}_{\mathrm{P}}\in\mathbf{R}^{n\times n}$，$\boldsymbol{K}_{\mathrm{D}}\in\mathbf{R}^{n\times n}$ 和 $\boldsymbol{\Gamma}\in\mathbf{R}^{m\times m}$ 为正定对称矩阵。

那么，$\tilde{\boldsymbol{q}}_k(t)$，$\dot{\tilde{\boldsymbol{q}}}_k(t)$有界，且 $\lim\limits_{k\to\infty}\tilde{\boldsymbol{q}}_k(t)=\lim\limits_{k\to\infty}\dot{\tilde{\boldsymbol{q}}}_k(t)=0$，$t\in[0,T]$。

定理 5.2 的证明类似于 5.3 节的分析过程。文献[5]给出了定理 5.2 的证明过程，该证明过程分为以下 3 个步骤：

(1) $\Delta W_k(t)$的有界性证明：首先定义 Lyapunov 能量函数，然后通过证明 $\Delta W_k(t)\leqslant 0$，证明 $\Delta W_k(t)$为非递增序列。

(2) $W_0(t)$的连续性和有界性证明：首先证明 $\dot{W}_0(t)$的有界性，然后根据一致连续性判定定理(见附录)，证明 $W_0(t)$在$[0,T]$上一致连续，再根据闭区间上连续函数的有界定理(见附录)，证明 $W_0(t)$在$[0,T]$上有界。

(3) 误差的收敛性证明：由于 $W_0(t)$在$[0,T]$上有界，在 $W_k(t)$有界，进而可证明 $\tilde{\boldsymbol{q}}_k(t)$、$\dot{\tilde{\boldsymbol{q}}}_k(t)$有界，且 $\lim\limits_{k\to\infty}\tilde{\boldsymbol{q}}_k(t)=\lim\limits_{k\to\infty}\dot{\tilde{\boldsymbol{q}}}_k(t)=0$，$t\in[0,T]$。

在上述控制器的基础上，根据文献[5]，给出了如下两个控制器设计方法：

定理 5.3

$$\boldsymbol{\tau}_k(t)=\boldsymbol{K}_{\mathrm{P}}\tilde{\boldsymbol{q}}_k(t)+\boldsymbol{K}_{\mathrm{D}}\dot{\tilde{\boldsymbol{q}}}_k(t)+\boldsymbol{\eta}(\dot{\tilde{\boldsymbol{q}}}_k)\hat{\boldsymbol{\theta}}_k(t) \tag{5.39}$$

$$\hat{\boldsymbol{\theta}}_k(t)=\hat{\boldsymbol{\theta}}_{k-1}(t)+\boldsymbol{\Gamma}\boldsymbol{\eta}^{\mathrm{T}}(\dot{\tilde{\boldsymbol{q}}}_k)\dot{\tilde{\boldsymbol{q}}}_k(t) \tag{5.40}$$

定理 5.4

$$\boldsymbol{\tau}_k(t)=\boldsymbol{K}_{\mathrm{P}}\tilde{\boldsymbol{q}}_k(t)+\boldsymbol{K}_{\mathrm{D}}\dot{\tilde{\boldsymbol{q}}}_k(t)+\hat{\delta}_k(t)\mathrm{sgn}(\dot{\tilde{\boldsymbol{q}}}_k(t)) \tag{5.41}$$

$$\hat{\delta}_k(t)=\hat{\delta}_{k-1}(t)+\gamma\dot{\tilde{\boldsymbol{q}}}_k^{\mathrm{T}}(t)\mathrm{sgn}(\dot{\tilde{\boldsymbol{q}}}_k(t)) \tag{5.42}$$

定理 5.3 及定理 5.4 的收敛性分析过程类似于定理 5.2 的收敛性分析过程，详细分析见参考文献[5]。

5.4.3 仿真实例

被控对象为二关节机器人力臂，动力学方程见式(5.36)，即

$$\boldsymbol{M}(\boldsymbol{q}_k(t))\ddot{\boldsymbol{q}}_k(t)+\boldsymbol{C}(\boldsymbol{q}_k(t),\dot{\boldsymbol{q}}_k(t))\dot{\boldsymbol{q}}_k(t)+\boldsymbol{G}(\boldsymbol{q}_k(t))=\boldsymbol{\tau}_k(t)+\boldsymbol{d}_k(t)$$

其中上式各项表示为

$\boldsymbol{M}=[m_{ij}]_{2\times2}$，$m_{11}=m_1l_{c1}^2+m_2(l_1^2+l_{c2}^2+2l_1l_{c2}\cos q_2)+I_1+I_2$，$m_{12}=m_{21}=m_2(l_{c2}^2+l_1l_{c2}\cos q_2)+l_2$，$m_{22}=m_2l_{c2}^2+I_2$。$\boldsymbol{C}=[c_{ij}]_{2\times2}$，$c_{11}=h\dot{q}_2$，$c_{12}=h\dot{q}_1+h\dot{q}_2$，$c_{21}=-h\dot{q}_1$，$c_{22}=0$，$h=-m_2l_1l_{c2}\sin q_2$。$\boldsymbol{G}=[G_1 \quad G_2]^{\mathrm{T}}$，$G_1=(m_1l_{c1}+m_2l_1)g\cos q_1+m_2l_{c2}g\cos(q_1+q_2)$，$G_2=m_2l_{c2}g\cos(q_1+q_2)$；干扰项为$\boldsymbol{d}_k(t)=[d_m\sin(t) \quad d_m\sin(t)]^{\mathrm{T}}$，其中$d_m$为幅值为1的随机信号。

机器人系统参数为$m_1=m_2=1\mathrm{kg}$，$l_1=l_2=0.5\mathrm{m}$，$l_{c1}=l_{c2}=0.25\mathrm{m}$，$I_1=I_2=0.1\mathrm{kg}\times\mathrm{m}^2$，$g=9.81\mathrm{m/s}^2$。

两个关节的角度指令信号分别为$\sin(2\pi t)$和$\cos(2\pi t)$。为了保证被控对象初始输出与指令初值一致，取被控对象的初始状态为$\boldsymbol{x}(0)=[0 \quad 2\pi \quad 1 \quad 0]^{\mathrm{T}}$。

控制器参数选为$\boldsymbol{K}_{\mathrm{P}}=\boldsymbol{K}_{\mathrm{D}}=\begin{bmatrix}10 & 0\\ 0 & 10\end{bmatrix}$，自适应律参数取

$$\boldsymbol{\Gamma}=\begin{bmatrix}10 & 0 & 0 & 0 & 0\\ 0 & 10 & 0 & 0 & 0\\ 0 & 0 & 10 & 0 & 0\\ 0 & 0 & 0 & 10 & 0\\ 0 & 0 & 0 & 0 & 10\end{bmatrix}$$

总的迭代次数为5次，每1次的仿真时间为1。采用控制律式(5.37)和自适应律式(5.38)，仿真结果如图5.16～图5.19所示。

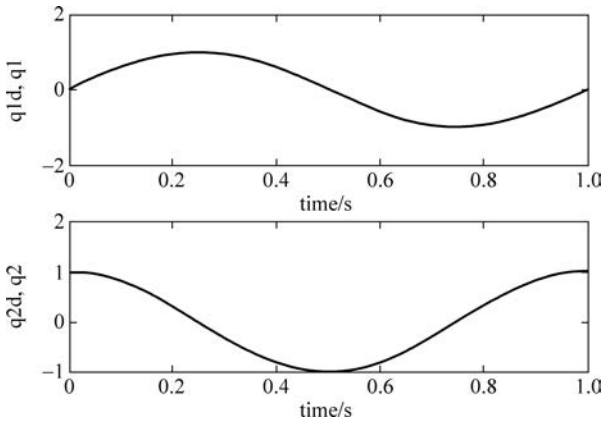


图5.16　5次迭代后的角度跟踪

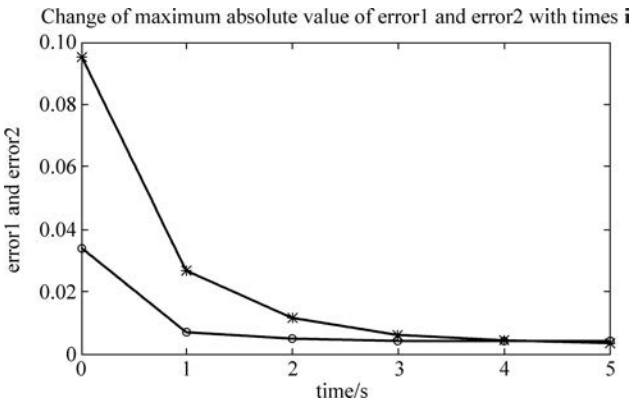


图5.17　5次迭代过程中角度跟踪误差绝对值的收敛过程

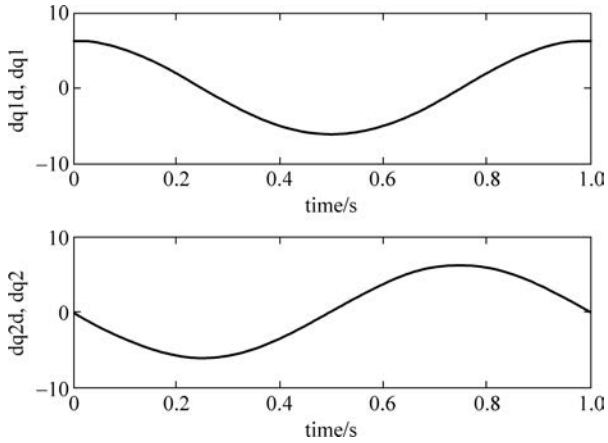


图 5.18　5 次迭代后的角速度跟踪

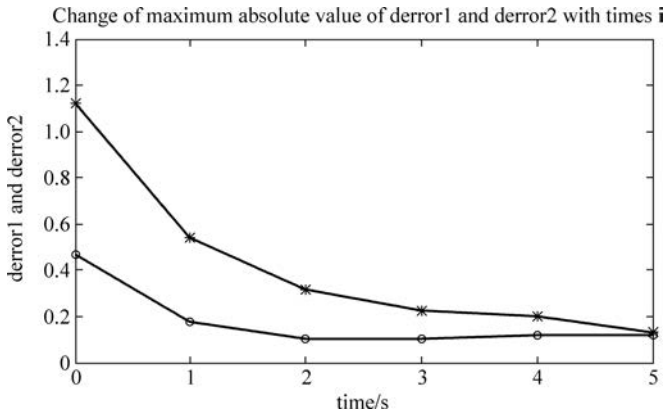


图 5.19　5 次迭代过程中角速度跟踪误差绝对值的收敛过程

仿真程序(一)：采用定理 5.2 的双力臂机械手自适应迭代学习控制。

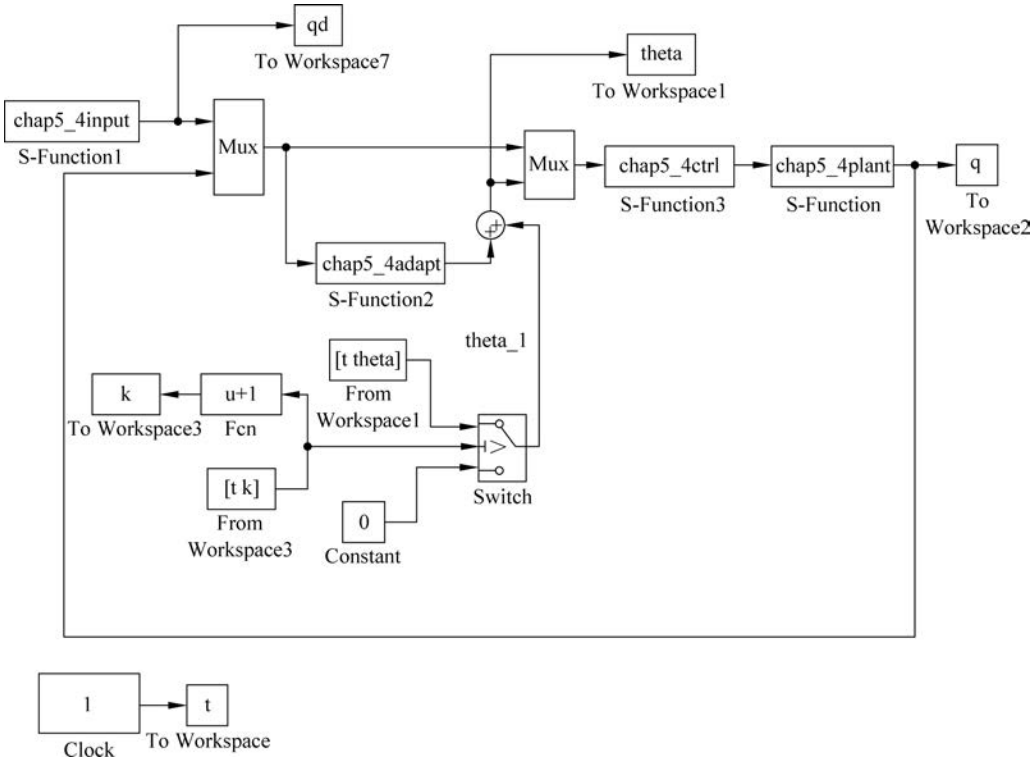
(1) 主程序：chap5_7main. m。

```
clear all;
close all;

t = [0:0.01:1]';
k(1:101) = 0;
k = k';

theta1(1:101) = 0;
theta1 = theta1';
theta2 = theta1;
theta3 = theta1;
theta4 = theta1;
theta5 = theta1;
theta = [theta1 theta2 theta3 theta4 theta5];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M = 5;
for i = 0:1:M
i
pause(0.01);
```

```
sim('chap5_4sim',[0,1]);

q1 = q(:,1);
dq1 = q(:,2);
q2 = q(:,3);
dq2 = q(:,4);
q1d = qd(:,1);
q2d = qd(:,2);
dq1d = qd(:,3);
dq2d = qd(:,4);
e1 = q1d - q1;
e2 = q2d - q2;
de1 = dq1d - dq1;
de2 = dq2d - dq2;

figure(1);
subplot(211);
hold on;
plot(t,q1,'b',t,q1d,'r');
xlabel('time(s)');ylabel('q1d,q1 (rad)');

subplot(212);
hold on;
plot(t,q2,'b',t,q2d,'r');
xlabel('time(s)');ylabel('q2d,q2 (rad)');

j = i + 1;
times(j) = i;
e1i(j) = max(abs(e1));
e2i(j) = max(abs(e2));
de1i(j) = max(abs(de1));
de2i(j) = max(abs(de2));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(2);
subplot(211);
plot(t,q1d,'r',t,q1,'b');
xlabel('time(s)');ylabel('q1d,q1');
subplot(212);
plot(t,q2d,'r',t,q2,'b');
xlabel('time(s)');ylabel('q2d,q2');

figure(3);
plot(times,e1i,'* -r',times,e2i,'o-b');
title('Change of maximum absolute value of error1 and error2 with times i');
xlabel('times');ylabel('error1 and error2');

figure(4);
subplot(211);
plot(t,dq1d,'r',t,dq1,'b');
xlabel('time(s)');ylabel('dq1d,dq1');
subplot(212);
plot(t,dq2d,'r',t,dq2,'b');
xlabel('time(s)');ylabel('dq2d,dq2');

figure(5);
plot(times,de1i,'* -r',times,de2i,'o-b');
```

```
title('Change of maximum absolute value of derror1 and derror2 with times i');
xlabel('times');ylabel('derror1 and derror2');
```

(2) Simulink 子程序：chap5_4sim. mdl。



(3) 控制器子程序：chap5_4ctrl. m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 2;
sizes.NumInputs      = 13;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];
function sys = mdlOutputs(t,x,u)
```

```
q1d = u(1);q2d = u(2);
dq1d = u(3);dq2d = u(4);

q1 = u(5);dq1 = u(6);
q2 = u(7);dq2 = u(8);

e1 = q1d - q1;
e2 = q2d - q2;
e = [e1 e2]';
de1 = dq1d - dq1;
de2 = dq2d - dq2;
de = [de1 de2]';

Kp = 10 * eye(2);
Kd = Kp;

theta1 = u(9);
theta2 = u(10);
theta3 = u(11);
theta4 = u(12);
theta5 = u(13);
theta = [theta1 theta2 theta3 theta4 theta5]';

fai11 = dq2 * sin(q2);fai21 = - dq1 * sin(q2);
fai12 = fai11 - fai21;fai22 = 0;
fai13 = cos(q1);fai23 = 0;
fai14 = cos(q1 + q2);
fai24 = cos(q1 + q2);
fai15 = sign(de1);
fai25 = sign(de2);
Fai = [fai11 fai12 fai13 fai14 fai15;
      fai21 fai22 fai23 fai24 fai25];

tol = Kp * e + Kd * de + Fai * theta;

sys(1) = tol(1);
sys(2) = tol(2);
```

（4）自适应律子程序：chap5_4adapt.m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 5;
```

```
sizes.NumInputs       = 8;
sizes.DirFeedthrough  = 1;
sizes.NumSampleTimes  = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];
function sys = mdlOutputs(t,x,u)
q1d = u(1);q2d = u(2);
dq1d = u(3);dq2d = u(4);

q1 = u(5);dq1 = u(6);
q2 = u(7);dq2 = u(8);

de1 = dq1d - dq1;
de2 = dq2d - dq2;
de = [de1 de2]';

fai11 = dq2 * sin(q2);fai21 = - dq1 * sin(q2);
fai12 = fai11 - fai21;fai22 = 0;
fai13 = cos(q1);fai23 = 0;
fai14 = cos(q1 + q2);
fai24 = cos(q1 + q2);
fai15 = sign(de1);
fai25 = sign(de2);
Fai = [fai11 fai12 fai13 fai14 fai15;
       fai21 fai22 fai23 fai24 fai25];
Gama = 10 * eye(5);
theta = Gama * Fai' * de;

sys(1) = theta(1);
sys(2) = theta(2);
sys(3) = theta(3);
sys(4) = theta(4);
sys(5) = theta(5);
```

(5) 被控对象子程序：chap5_4plant.m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 1,
    sys = mdlDerivatives(t,x,u);
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates   = 4;
sizes.NumDiscStates   = 0;
sizes.NumOutputs      = 4;
```

```
sizes.NumInputs      = 2;
sizes.DirFeedthrough = 0;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [0;2*pi;1;0];
str = [];
ts  = [0 0];
function sys = mdlDerivatives(t,x,u)
g = 9.81;
m1 = 1;m2 = 1;
l1 = 0.5;l2 = 0.5;
lc1 = 0.25;lc2 = 0.25;
I1 = 0.1;I2 = 0.1;

q1 = x(1);
dq1 = x(2);
q2 = x(3);
dq2 = x(4);

m11 = m1*lc1^2 + m2*(l1^2 + lc2^2 + 2*l1*lc2*cos(q2)) + I1 + I2;
m12 = m2*(lc2^2 + l1*lc2*cos(q2)) + I2;
m21 = m12;
m22 = m2*lc2^2 + I2;
M = [m11 m12;m21 m22];

h = -m2*l1*lc2*sin(q2);
c11 = h*dq2;
c12 = h*dq1 + h*dq2;
c21 = -h*dq1;
c22 = 0;
C = [c11 c12;c21 c22];

G1 = (m1*lc1 + m2*l1)*g*cos(q1) + m2*lc2*g*cos(q1 + q2);
G2 = m2*lc2*g*cos(q1 + q2);
G = [G1;G2];

d1 = rand(1)*sin(t);
d2 = rand(1)*sin(t);
d = [d1;d2];

tol = [u(1) u(2)]';

S = inv(M)*(tol + d - C*[dq1;dq2] - G);

sys(1) = x(2);
sys(2) = S(1);
sys(3) = x(4);
sys(4) = S(2);
function sys = mdlOutputs(t,x,u)
sys(1) = x(1);      % Angle1:q1
sys(2) = x(2);      % Angle1 speed:dq1
sys(3) = x(3);      % Angle2:q2
sys(4) = x(4);      % Angle2 speed:dq2
```

(6) 指令子程序：chap5_4input.m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 4;
sizes.NumInputs      = 0;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];
function sys = mdlOutputs(t,x,u)
q1d = sin(2 * pi * t);
q2d = cos(2 * pi * t);
dq1d = 2 * pi * cos(2 * pi * t);
dq2d = -2 * pi * sin(2 * pi * t);

sys(1) = q1d;
sys(2) = q2d;
sys(3) = dq1d;
sys(4) = dq2d;
```

仿真程序(二)：基于切换增益的多关节机械手迭代学习控制(采用定理 5.3)。

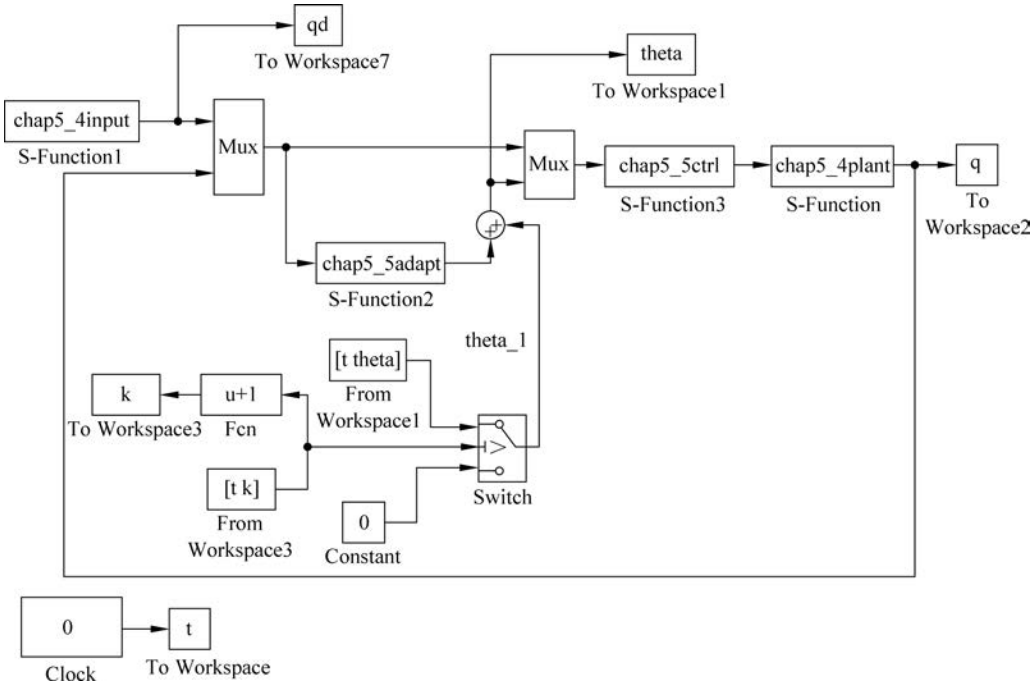
(1) 主程序：chap5_5main.m。

```
% Adaptive switching Learning Control for 2DOF robot manipulators
clear all;
close all;

t = [0:0.01:1]';
k(1:101) = 0;
k = k';

theta1(1:101) = 0;
theta1 = theta1';
theta2 = theta1;
theta = [theta1 theta2];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M = 5;
```

```
for i = 0:1:M      % Start Learning Control
i
pause(0.01);

sim('chap5_5sim',[0,1]);

q1 = q(:,1);
dq1 = q(:,2);
q2 = q(:,3);
dq2 = q(:,4);
q1d = qd(:,1);
q2d = qd(:,2);
dq1d = qd(:,3);
dq2d = qd(:,4);
e1 = q1d - q1;
e2 = q2d - q2;
de1 = dq1d - dq1;
de2 = dq2d - dq2;

figure(1);
subplot(211);
hold on;
plot(t,q1,'b',t,q1d,'r');
xlabel('t (s)');ylabel('q1d,q1 (rad)');

subplot(212);
hold on;
plot(t,q2,'b',t,q2d,'r');
xlabel('t (s)');ylabel('q2d,q2 (rad)');

j = i + 1;
times(j) = i;
e1i(j) = max(abs(e1));
e2i(j) = max(abs(e2));
de1i(j) = max(abs(de1));
de2i(j) = max(abs(de2));
end          % End of i
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(2);
subplot(211);
plot(t,q1d,'r',t,q1,'b');
xlabel('t');ylabel('q1d,q1');
subplot(212);
plot(t,q2d,'r',t,q2,'b');
xlabel('t');ylabel('q2d,q2');

figure(3);
plot(times,e1i,'* - r',times,e2i,'o - b');
title('Change of maximum absolute value of error1 and error2 with times i');
xlabel('times');ylabel('error1 and error2');

figure(4);
```

```
subplot(211);
plot(t,dq1d,'r',t,dq1,'b');
xlabel('t');ylabel('dq1d,dq1');
subplot(212);
plot(t,dq2d,'r',t,dq2,'b');
xlabel('t');ylabel('dq2d,dq2');

figure(5);
plot(times,de1i,'* -r',times,de2i,'o-b');
title('Change of maximum absolute value of derror1 and derror2 with times i');
xlabel('times');ylabel('derror1 and derror2');
```

(2) Simulink 子程序：chap5_5sim. mdl。



(3) 控制器子程序：chap5_5ctrl. m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
```

```
sizes.NumOutputs      = 2;
sizes.NumInputs       = 10;
sizes.DirFeedthrough  = 1;
sizes.NumSampleTimes  = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];
function sys = mdlOutputs(t,x,u)
q1d = u(1);q2d = u(2);
dq1d = u(3);dq2d = u(4);

q1 = u(5);dq1 = u(6);
q2 = u(7);dq2 = u(8);

e1 = q1d - q1;
e2 = q2d - q2;
e = [e1 e2]';
de1 = dq1d - dq1;
de2 = dq2d - dq2;
de = [de1 de2]';

Kp = 10 * eye(2);
Kd = Kp;

xite = [de1 sign(de1);de2 sign(de2)];

theta1 = u(9);
theta2 = u(10);
theta = [theta1 theta2]';

tol = Kp * e + Kd * de + xite * theta;

sys(1) = tol(1);
sys(2) = tol(2);
```

（4）自适应律子程序：chap5_5adapt.m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
```

```
sizes.NumOutputs      = 2;
sizes.NumInputs       = 8;
sizes.DirFeedthrough  = 1;
sizes.NumSampleTimes  = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];
function sys = mdlOutputs(t,x,u)
q1d = u(1);q2d = u(2);
dq1d = u(3);dq2d = u(4);

q1 = u(5);dq1 = u(6);
q2 = u(7);dq2 = u(8);

de1 = dq1d - dq1;
de2 = dq2d - dq2;
de = [de1 de2]';

Gama = 10 * eye(2);
xite = [de1 sign(de1);de2 sign(de2)];
theta = Gama * xite' * de;

sys(1) = theta(1);
sys(2) = theta(2);
```

(5) 被控对象子程序：chap5_4plant.m。

同“仿真程序(一)”。

(6) 指令子程序：chap5_4input.m。

同“仿真程序(一)”。

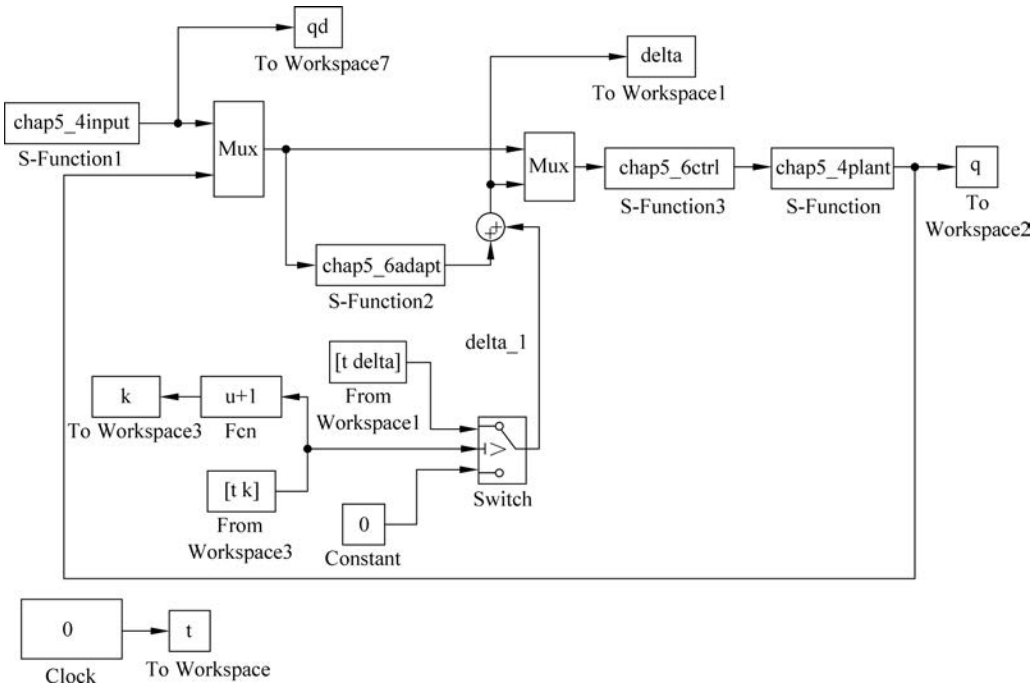
仿真程序(三)：基于切换增益的多关节机械手迭代学习控制(采用定理 5.4)。

(1) 主程序：chap5_6main.m。

```
% Adaptive switching Learning Control for 2DOF robot manipulators
clear all;
close all;

t = [0:0.01:1]';
k(1:101) = 0;
k = k';

delta(1:101) = 0;
delta = delta';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M = 10;
for i = 0:1:M     % Start Learning Control
i
pause(0.01);

sim('chap5_6sim',[0,1]);

q1 = q(:,1);
dq1 = q(:,2);
```

```
q2 = q(:,3);
dq2 = q(:,4);
q1d = qd(:,1);
q2d = qd(:,2);
dq1d = qd(:,3);
dq2d = qd(:,4);
e1 = q1d - q1;
e2 = q2d - q2;
de1 = dq1d - dq1;
de2 = dq2d - dq2;

figure(1);
subplot(211);
hold on;
plot(t,q1,'b',t,q1d,'r');
xlabel('t (s)');ylabel('q1d,q1 (rad)');

subplot(212);
hold on;
plot(t,q2,'b',t,q2d,'r');
xlabel('t (s)');ylabel('q2d,q2 (rad)');

j = i + 1;
times(j) = i;
e1i(j) = max(abs(e1));
e2i(j) = max(abs(e2));
de1i(j) = max(abs(de1));
de2i(j) = max(abs(de2));
end          % End of i
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(2);
subplot(211);
plot(t,q1d,'r',t,q1,'b');
xlabel('t');ylabel('q1d,q1');
subplot(212);
plot(t,q2d,'r',t,q2,'b');
xlabel('t');ylabel('q2d,q2');

figure(3);
plot(times,e1i,'* - r',times,e2i,'o - b');
title('Change of maximum absolute value of error1 and error2 with times i');
xlabel('times');ylabel('error1 and error2');

figure(4);
subplot(211);
plot(t,dq1d,'r',t,dq1,'b');
xlabel('t');ylabel('dq1d,dq1');
subplot(212);
plot(t,dq2d,'r',t,dq2,'b');
xlabel('t');ylabel('dq2d,dq2');

figure(5);
plot(times,de1i,'* - r',times,de2i,'o - b');
title('Change of maximum absolute value of derror1 and derror2 with times i');
xlabel('times');ylabel('derror1 and derror2');
```

（2）Simulink 子程序：chap5_6sim. mdl。



（3）控制器子程序：chap5_6ctrl. m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 2;
sizes.NumInputs      = 9;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];
function sys = mdlOutputs(t,x,u)
q1d = u(1);q2d = u(2);
dq1d = u(3);dq2d = u(4);

q1 = u(5);dq1 = u(6);
q2 = u(7);dq2 = u(8);
```

```
e1 = q1d - q1;
e2 = q2d - q2;
e = [e1 e2]';
de1 = dq1d - dq1;
de2 = dq2d - dq2;
de = [de1 de2]';

Kp = 10 * eye(2);
Kd = Kp;

delta = u(9);

tol = Kp * e + Kd * de + delta * sign(de);

sys(1) = tol(1);
sys(2) = tol(2);
```

（4）自适应律子程序：chap5_6adapt. m。

```
function [sys,x0,str,ts] = spacemodel(t,x,u,flag)
switch flag,
case 0,
    [sys,x0,str,ts] = mdlInitializeSizes;
case 3,
    sys = mdlOutputs(t,x,u);
case {2,4,9}
    sys = [];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates  = 0;
sizes.NumDiscStates  = 0;
sizes.NumOutputs     = 1;
sizes.NumInputs      = 8;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
x0  = [];
str = [];
ts  = [0 0];
function sys = mdlOutputs(t,x,u)
q1d = u(1);q2d = u(2);
dq1d = u(3);dq2d = u(4);

q1 = u(5);dq1 = u(6);
q2 = u(7);dq2 = u(8);

de1 = dq1d - dq1;
de2 = dq2d - dq2;
de = [de1 de2]';

Gama = 10;
delta = Gama * de' * sign(de);
sys(1) = delta;
```

(5) 被控对象子程序：chap5_4plant. m。

同“仿真程序(一)”。

(6) 指令子程序：chap5_4input. m。

同“仿真程序(一)”。

附录

(1) 一致连续性判定定理[6]：若函数 $f(x)$在区间$[a,b]$上的导数有界，则 $f(x)$在$[a,b]$上一致连续。

(2) 闭区间有界定理[7]：若函数 $f(x)$在闭区间$[a,b]$上连续，则它在$[a,b]$上有界。

参考文献

[1] Arimoto S, Kawamura S, Miyazaki F. Bettering Operation of robotics by leaning. Journal of Robotic System, 1984, 1(2): 123-140.

[2] 孙明轩，黄宝健. 迭代学习控制. 北京：国防工业出版社，1999.

[3] 谢胜利. 迭代学习控制的理论与应用. 北京：科学出版社，2005.

[4] Ouyang P R, Zhang W J, Gupta M M. An adaptive switching learning control method for trajectory tracking of robot manipulators. Mechatronics, 2006, 16: 51-61.

[5] Tayebi A. Adaptive iterative learning control for robot manipulators. Automatica, 2004, 40: 1195-1203.

[6] 甘宗怀，李秋林. 关于可导函数一致连续性的判定定理，高师理科学刊，2009，29(5)：38-39.

[7] 陈纪修，於崇华，金路. 数学分析(上册). 2 版. 北京：高等教育出版社，2004.

[8] A Mohammadi, M Tavakoli, H J Marquez, F Hashemzadeh. Nonlinear disturbance observer design for robotic manipulators, Engineering Practice, 2013, 21: 253-267.