高技术通讯 2023年 第33卷 第11期:1181-1191

doi:10.3772/j.issn.1002-0470.2023.11.006

基于非零和博弈的自适应人机协作系统设计①

禹鑫燚② 罗惠珍 史栓武 魏 岩 欧林林③

(浙江工业大学信息工程学院 杭州 310023)

摘 要 为了提高人机协作的协调性,本文设计了基于非零和博弈的自适应人机协作系统,系统由互相解耦的内外环构成。在外环中,通过引入非零和博弈的方法设计人机协作策略,构建关于人力和机器人控制输入的能量函数,通过求解博弈中的纳什均衡达到最优控制。针对能量函数中的不确定参数,采用神经网络估计器进行更新,以估计人和机器人的力。并且通过设计神经网络函数的中心值,获得机器人控制力与跟踪误差的关系,保证控制方法的跟踪性。在更新过程中自适应调整刚度系数,实现人机柔顺协调。另外,在内环中设计了神经网络控制器,采用径向基神经网络,基于实时采集的机器人系统输入输出数据逼近控制器中未知非线性的机器人动力学模型,提高了系统跟踪精度。仿真结果验证了本文方法的有效性。

关键词 人机协作; 自适应阻抗控制; 非零和博弈; 神经网络

0 引 言

随着机器人越来越广泛地应用于装配、搬运、康复等物理协作任务[1-3]中,机器人在非结构化环境中的人机协作技术发挥着越来越重要的作用,因而人机协作得到了越来越多的关注。机器人适合做重复、高精度且具有一定危险性的工作,而人在面对突发事件时的灵活性和对未知环境的适应能力弥补了机器人的不足。为了在人机协作中充分利用机器人和人的优点,需要提高人和机器人协作的自适应性。

在人机协作控制领域,通常有阻抗控制[4]和导纳控制[5]2种有效控制方法。阻抗控制通过阻尼-弹簧-质量模型表示人和机器人的动态位置关系[6],测量机器人空间位置以及选取合适的阻抗参数,获得机器人输入力。因此,模型阻抗参数的选择至关重要,对于不同工作环境和操作人员需要不同的模型参数。由于环境的复杂性,难以获取合适的模型参数。文献[7]建立了阻抗模型的刚度系数与人机交互力的关系,通过自适应律调整刚度系数大小。文献[8]提出了一种辅助-对抗的方法,根据人在任务中的参与程度自适应调整阻抗参数。为了提高自适应性,变阻抗的方法被引入机器人领域,可以通过强化学习(reinforcement learning, RL)方法在线自适应调整阻抗参数[9]。文献[10]在每个关节上设计独立的RL输入补偿器。文献[11]利用输入输出数据重新构造状态,解决了未知环境动力学的最优阻抗控制问题。但现有的人机协作控制方法所设计的控制策略通常适用于某特定任务,缺乏多用途的系统方法。

在执行一些复杂多变的任务时,为了实现柔顺控制,人和机器人力的灵活调配起到关键作用。而在工业机器人的实际应用中,通常要考虑到任务的多样性,如车间装配在不同的阶段要完成不同的任务。因此需要设计通用的人机协作策略来完成人和机器人之间不同的协作任务,以适应任务变化。以

① 国家自然科学基金(62373329)和浙江省自然科学基金(LY21F030018)资助项目。
② 男,1979年生,博士;研究方向:机器人控制与规划;E-mail:yuxinyinet@163.com。
③ 通信作者,E-mail: linlinou@zjut.edu.com。
(收稿日期:2022-02-16)

往的研究设计了人机协作的自适应框架[12-14]。文献[15]根据人和机器人之间的差异程度在基于模型和无模型策略之间动态选择。这些研究指出，机器人需要根据人的意图做出判断，根据任务自适应调整引导或者跟随任务的角色。博弈论(game theory，GT)适用于分析多智能体系统[16-25]，能够用来构建通用的方法描述人机协作的双智能体博弈。非零和博弈(non-zero-sum games)是一种合作下的博弈[17]，适用于人机协作，可以通过给定目标函数和人机协作任务目标进行建模。文献[18]在已知线性系统目标函数的博弈中，通过求解黎卡提方程的方法获得最优控制。然而，该方法是针对确定目标函数生成的固定控制策略，在协作过程中可能无法达到平衡。因此通过评估交互性能，文献[19]采用策略迭代的方法降低计算成本并不断更新控制策略。策略迭代的方法被证明适用于已知模型或未知模型的线性系统[20-22]，且应用于已知或未知动态的博弈[23]。然而，在实际的人机协作场景中，机器人通常事先不知道人机协作的任务目标。

为了解决人机协作时人和机器人自动切换引导或跟随角色的问题，使得人和机器人能够根据任务相互协调，本文设计了自适应人机协作系统，系统由互相解耦的内外环构成。外环通过非零和博弈描述人机协作系统，并通过求解纳什均衡获得人机协作系统最优控制策略。首先对系统的阻抗模型重新建模，再构建关于人力和机器人控制输入的能量函数；针对能量函数中的不确定参数，采用神经网络拟合估计器更新，并自适应调整刚度系数，实现人机柔顺协调；设计神经网络函数的中心值保证控制方法的跟踪性。此外，在内环中设计径向基神经网络控制器，利用实时更新的机器人输入输出数据逼近机器人动力学模型，提高跟踪精度。

1 人机协作系统结构设计

人机协作系统由人和机器人组成。当没有外力干扰时，机器人会重复执行预定任务，如果人对机器人施加一个外力，那么机器人会服从人力沿着新的目标轨迹运动。通常，人施加的力是随机且多变的，需要完成的任务轨迹也不同。

本文设计的人机协作系统总体结构如图 1 所示，包括内环和外环。外环和内环的作用分别为平衡人机控制策略和提高跟踪精度。在外环中，通过引入非零和博弈论的方法，对人和机器人的交互行为进行博弈。此外利用神经网络拟合能量函数中的参数，确定阻抗模型中的刚度系数值。在内环回路中，设计了一个自适应神经网络控制器，利用神经网络补偿机器人动力学模型中的未知项。机器人将跟踪阻抗模型中求得的参考轨迹 $\boldsymbol{x}_{\mathrm{r}}$，而参考轨迹将跟踪实际期望轨迹 $\boldsymbol{x}_{\mathrm{d}}$，实现内外环互不干扰。

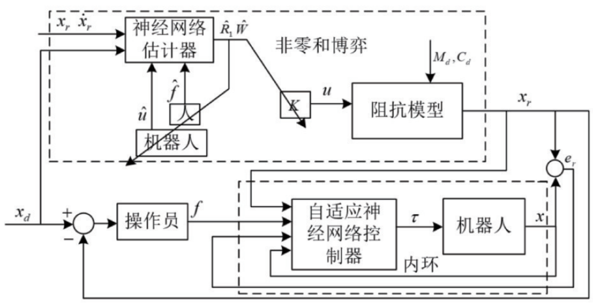


图 1 人机协作系统结构图

考虑如下机器人的笛卡尔空间动力学模型：

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{x}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{x}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{f} \tag{1}$$

其中。$\boldsymbol{q}$、$\dot{\boldsymbol{q}}$、$\ddot{\boldsymbol{q}} \in \boldsymbol{R}^n$ 分别是关节角、关节角速度、关节加速度；$\boldsymbol{x}$、$\dot{\boldsymbol{x}}$、$\ddot{\boldsymbol{x}} \in \boldsymbol{R}^m$ 分别为机器人末端执行器在笛卡尔空间的位置、速度、加速度；$\boldsymbol{M} = \boldsymbol{J}^{-\mathrm{T}}\boldsymbol{M}^*\boldsymbol{J}^{-1}$；$\boldsymbol{C} = \boldsymbol{J}^{-\mathrm{T}}(\boldsymbol{C}^* - \boldsymbol{M}^*\boldsymbol{J}^{-1}\dot{\boldsymbol{J}})\boldsymbol{J}^{-1}$；$\boldsymbol{G} = \boldsymbol{J}^{-\mathrm{T}}\boldsymbol{G}^*$；$\boldsymbol{M}^*(\boldsymbol{q}) \in \boldsymbol{R}^{n\times n}$ 为正定惯性矩阵，$\boldsymbol{C}^*(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \boldsymbol{R}^{n\times n}$ 为哥式和离心力矩阵，$\boldsymbol{G}^*(\boldsymbol{q}) \in \boldsymbol{R}^n$ 为重力矩阵，$\boldsymbol{J}(\boldsymbol{q}) \in \boldsymbol{R}^{m\times n}$ 为雅可比矩阵。$\boldsymbol{\tau}^*$ 为控制输入力矩，$\boldsymbol{f}$ 为人施加在机器人末端的交互力。n 为机器人关节数，m 为机器人自由度(degree of freedom，DOF)。

假设机器人的笛卡尔空间阻抗模型为

$$\boldsymbol{M}_{\mathrm{d}}\ddot{\boldsymbol{x}}_{\mathrm{r}} + \boldsymbol{C}_{\mathrm{d}}\dot{\boldsymbol{x}}_{\mathrm{r}} = \boldsymbol{u}(t) + \boldsymbol{f}(t) \tag{2}$$

其中，$\boldsymbol{x}_{\mathrm{r}}$、$\ddot{\boldsymbol{x}}_{\mathrm{r}}$、$\dot{\boldsymbol{x}}_{\mathrm{r}}$ 分别为虚拟参考轨迹、速度和加速度；$\boldsymbol{M}_{\mathrm{d}} \in \boldsymbol{R}^{m\times m}$ 和 $\boldsymbol{C}_{\mathrm{d}} \in \boldsymbol{R}^{m\times m}$ 分别为期望的惯性矩阵和阻尼矩阵；$\boldsymbol{u}(t) \in \boldsymbol{R}^m$ 为笛卡尔空间的控制输入力。

当机器人在协作过程中跟踪固定的期望轨迹

$\boldsymbol{x}_{\mathrm{d}}$ 时,把阻抗模型改写成以下形式:

$$\boldsymbol{M}_{\mathrm{d}}(\ddot{\boldsymbol{x}}_{\mathrm{r}}-\ddot{\boldsymbol{x}}_{\mathrm{d}})+\boldsymbol{C}_{\mathrm{d}}(\dot{\boldsymbol{x}}_{\mathrm{r}}-\dot{\boldsymbol{x}}_{\mathrm{d}})=\boldsymbol{u}(t)+\boldsymbol{f}(t) \tag{3}$$

其中为了便于控制器的设计,把式(3)中的阻抗模型改写成状态空间方程的形式:

$$\dot{\boldsymbol{z}}(t)=\boldsymbol{A}\boldsymbol{z}(t)+\boldsymbol{B}_1\boldsymbol{u}+\boldsymbol{B}_2\boldsymbol{f} \tag{4}$$

$$\boldsymbol{z}(t)=\begin{bmatrix}\boldsymbol{x}_{\mathrm{r}}(t)-\boldsymbol{x}_{\mathrm{d}}\\ \dot{\boldsymbol{x}}_{\mathrm{r}}(t)\end{bmatrix},\ \boldsymbol{A}=\begin{bmatrix}\boldsymbol{0}_{\mathrm{m}} & \boldsymbol{I}_{\mathrm{m}}\\ \boldsymbol{0}_{\mathrm{m}} & -\boldsymbol{M}_{\mathrm{d}}^{-1}\boldsymbol{C}_{\mathrm{d}}\end{bmatrix}$$

$$\boldsymbol{B}_1=\boldsymbol{B}_2=\begin{bmatrix}\boldsymbol{0}_{\mathrm{m}}\\ \boldsymbol{M}_{\mathrm{d}}^{-1}\end{bmatrix} \tag{5}$$

其中,$\boldsymbol{0}_{\mathrm{m}}$ 和 $\boldsymbol{I}_{\mathrm{m}}$ 分别代表 $m\times m$ 维的零矩阵和单位矩阵;$\boldsymbol{x}_{\mathrm{r}}$ 和 $\dot{\boldsymbol{x}}_{\mathrm{r}}$ 是阻抗模型的输出参考轨迹和速度,用来跟踪期望轨迹 $\boldsymbol{x}_{\mathrm{d}}$,同时在内环中使机器人真实的轨迹 $\boldsymbol{x}$ 跟踪 $\boldsymbol{x}_{\mathrm{r}}$。

将人和机器人作为 2 个智能体,利用非零和博弈论的方法描述人和机器人之间的协作,达到最优的合作关系。根据增广的状态 $\boldsymbol{z}$,人机协作系统的能量函数可以写成如下形式:

$$\boldsymbol{V}=\int_0^{\infty}(\boldsymbol{z}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{z}+\boldsymbol{u}^{\mathrm{T}}\boldsymbol{R}_1\boldsymbol{u}+\boldsymbol{f}^{\mathrm{T}}\boldsymbol{R}_2\boldsymbol{f})\,\mathrm{d}t \tag{6}$$

其中,

$$\boldsymbol{Q}=\begin{bmatrix}\boldsymbol{Q}_1 & \boldsymbol{0}_{\mathrm{m}}\\ \boldsymbol{0}_{\mathrm{m}} & \boldsymbol{Q}_2\end{bmatrix} \tag{7}$$

$\boldsymbol{Q}_1$、$\boldsymbol{Q}_2$ 分别为轨迹跟踪误差和速度的权值矩阵,$\boldsymbol{R}_1$ 和 $\boldsymbol{R}_2$ 分别为机器人和人控制力的权值矩阵。通过调节这些权值矩阵的大小,来确定控制目标。

如果能量函数式(6)已知,那么通过最小化能量函数,可以设计机器人控制器来实现人机协调[23]。然而,在实际的人机协作过程中,人的意图会时刻变化且不可预知,所以能量函数 $\boldsymbol{V}$ 的参数随时间变化,无法确定。因此,本文将在控制外环中引入神经网络估计器确定非零和博弈中人和机器人的控制律,从而更新阻抗模型中的刚度系数达到柔顺人机协作的目的。在内环中,针对未知非线性机器人模型参数,设计了自适应神经网络控制器,并提高跟踪精度。

2 外环中的非零和博弈控制方法

本节拟给出基于非零和博弈的外环控制方法,根据式(6),该系统中机器人和人的能量函数分别定义为

$$\begin{aligned}\boldsymbol{V}_i(\boldsymbol{z}(t),\boldsymbol{u},\boldsymbol{f})&=\int_t^{\infty}\boldsymbol{c}_i(\boldsymbol{z},\boldsymbol{u},\boldsymbol{f})\,\mathrm{d}s\\&=\int_t^{\infty}(\boldsymbol{z}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{z}+\boldsymbol{u}^{\mathrm{T}}\boldsymbol{R}_{i1}\boldsymbol{u}+\boldsymbol{f}^{\mathrm{T}}\boldsymbol{R}_{i2}\boldsymbol{f})\,\mathrm{d}s\end{aligned}$$

$$(i=1,2) \tag{8}$$

其中,$\boldsymbol{c}_i(\boldsymbol{z},\boldsymbol{u},\boldsymbol{f})=\boldsymbol{z}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{z}+\boldsymbol{u}^{\mathrm{T}}\boldsymbol{R}_{i1}\boldsymbol{u}+\boldsymbol{f}^{\mathrm{T}}\boldsymbol{R}_{i2}\boldsymbol{f}$,$\boldsymbol{R}_{i1}$ 和 $\boldsymbol{R}_{i2}$ 分别为机器人和人的正定权值矩阵。为了最小化能量函数 $\boldsymbol{V}_i$,可以利用求解纳什均衡的方法,得到最优的 $\boldsymbol{u}$、$\boldsymbol{f}$ 控制策略。

定理 1[26] 机器人和人博弈的纳什均衡策略 $\boldsymbol{u}^*$、$\boldsymbol{f}^*$ 满足以下不等式:

$$\begin{aligned}&\boldsymbol{V}_1(\boldsymbol{z},\boldsymbol{u}^*,\boldsymbol{f}^*)\leqslant\boldsymbol{V}_1(\boldsymbol{z},\boldsymbol{u},\boldsymbol{f}^*)\\&\boldsymbol{V}_2(\boldsymbol{z},\boldsymbol{u}^*,\boldsymbol{f}^*)\leqslant\boldsymbol{V}_2(\boldsymbol{z},\boldsymbol{u}^*,\boldsymbol{f})\end{aligned} \tag{9}$$

从纳什均衡的定义中不难看出,机器人和人都有自己的能量函数,并且不能通过单方面改变控制策略来提高其性能。在非零和博弈过程中,无论对方的策略如何,人和机器人都会选择某个确定的策略,不受影响。如果在另外一方选择确定的情况下,该选择的策略达到最优,则为纳什均衡。

从式(8)中看出,机器人和人的控制策略相互耦合,所以本文通过强化学习中策略迭代[27]的方式来求解人机协作中的纳什均衡。假设能量函数式(8)连续且可微,分别对两边求导,结合式(4)则可被改写为以下非线性李雅普诺夫方程的形式:

$$\boldsymbol{c}_i(\boldsymbol{z},\boldsymbol{u},\boldsymbol{f})+\nabla\boldsymbol{V}_{\mathrm{i}}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{z}(t)+\boldsymbol{B}_1\boldsymbol{u}+\boldsymbol{B}_2\boldsymbol{f})=0 \tag{10}$$

其中 $\nabla\boldsymbol{V}_i^{\mathrm{T}}$ 是能量函数 $\boldsymbol{V}_i$ 关于 $\boldsymbol{z}$ 的偏导数的转置。为了确定策略迭代中的最优控制策略,分别引入如下的哈密顿函数和最优能量函数:

$$\begin{aligned}\boldsymbol{H}_i(\boldsymbol{z},\nabla\boldsymbol{V}_i,\boldsymbol{u},\boldsymbol{f})=&\boldsymbol{c}_i(\boldsymbol{z},\boldsymbol{u},\boldsymbol{f})+\nabla\boldsymbol{V}_i^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{z}(t)\\&+\boldsymbol{B}_1\boldsymbol{u}+\boldsymbol{B}_2\boldsymbol{f})\end{aligned} \tag{11}$$

$$\boldsymbol{V}_i^*(\boldsymbol{z})=\min_{\boldsymbol{u}_i\in\Psi(\Omega)}\int_t^{\infty}(\boldsymbol{z}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{z}+\boldsymbol{u}^{\mathrm{T}}\boldsymbol{R}_{i1}\boldsymbol{u}+\boldsymbol{f}^{\mathrm{T}}\boldsymbol{R}_{i2}\boldsymbol{f})\,\mathrm{d}s \tag{12}$$

为了便于表示,令 $\boldsymbol{u}_1=\boldsymbol{u}$,$\boldsymbol{u}_2=\boldsymbol{f}$。根据稳定性条件[23],当哈密顿函数关于控制策略的偏导数 $\dfrac{\partial\boldsymbol{H}}{\partial\boldsymbol{u}_i}=0$ 时,最优反馈控制策略满足:

$$u_i^* = \underset{u_i \in \Psi(\Omega)}{\operatorname{argmin}} H(z, \nabla V_i, u, f)$$

即

$$u_i^* = -\frac{1}{2} R_{ii}^{-1} B_i^{\mathrm{T}} \nabla V_i^* \tag{13}$$

如果对于能量函数已知的系统,可以直接利用策略迭代求解纳什均衡得到机器人和人的控制策略。然而,由于人的控制策略是无规则变化的,故能量函数 V_i 中的参数 R_{i1} 和 R_{i2} 无法确定。神经网络具有强大的拟合非线性曲线的能力,常用于未知的机器人控制系统[28]。因此引入神经网络估计器利用在线自适应的方法来确定适合的参数,从而得到人和机器人的控制律。

假设人和机器人有相同的能量函数 $V(z)$, 如式(6)所示,且连续可微,那么 $V(z)$ 可以通过神经网络被近似为

$$V(z) = W^{\mathrm{T}} S(z) + \varepsilon(z) \tag{14}$$

其中 W 是未知的期望权值矩阵, $S(z)$ 是神经网络激活函数, $\varepsilon(z)$ 是神经网络的近似误差。$V(z)$ 关于 z 的导数如下:

$$\begin{aligned}\nabla V &= \frac{\partial V(z)}{\partial z} = \left(\frac{\partial S(z)}{\partial z}\right)^{\mathrm{T}} W + \frac{\partial \varepsilon(z)}{\partial z} \\ &= \nabla S^{\mathrm{T}}(z) W + \nabla \varepsilon(z)\end{aligned} \tag{15}$$

其中 $\nabla S(z)$ 和 $\nabla \varepsilon(z)$ 分别为神经网络激活函数和近似误差的有界梯度。神经网络的权值矩阵 W 是未知的,能量函数的估计形式表示为

$$\hat{V}(z) = \hat{W}^{\mathrm{T}} S(z) \tag{16}$$

其中 $\hat{W}$ 是理想神经网络权值矩阵 W 的估计值。

由于人和机器人有相同的能量函数,则式(11)中的哈密顿方程可以写成:

$$\begin{aligned}H(z, \nabla V, u, f) = c(z, u, f) + \nabla V^{\mathrm{T}}(Az(t) \\ + B_1 u + B_2 f)\end{aligned} \tag{17}$$

且 $c(z,u,f) = z^{\mathrm{T}} Q z + u^{\mathrm{T}} R_1 u + f^{\mathrm{T}} R_2 f$, 将式(4)、(15)代入式(17)中,

$$\begin{aligned}H(z, \nabla V, u, f) &= c(z, u, f) + W^{\mathrm{T}} \nabla S(z)(A(z) \\ &\quad + B_1(z) u + B_2(z) f) \\ &\quad + \nabla \varepsilon^{\mathrm{T}}(z)(A(z) + B_1(z) u \\ &\quad + B_2(z) f) \\ &= c(z, u, f) + W^{\mathrm{T}} \sigma + \varepsilon_{\mathrm{H}}\end{aligned} \tag{18}$$

其中 $\sigma = \nabla S(z)(A(z) + B_1(z)u + B_2(z)f)$, $\varepsilon_{\mathrm{H}} = \nabla \varepsilon^{\mathrm{T}}(z)(A(z) + B_1(z)u + B_2(z)f)$, $A(z) = Az(t)$。

在人机协作的任务中, $c(z,u,f)$ 中的权值矩阵 Q、R_1 和 R_2 未知,又因为矩阵 R_1 和 R_2 是相互关联的,所以先令 R_2 为一个固定值,则 $c(z,u,f)$ 的估计值表示为

$$\hat{c}(z, u, f) = z^{\mathrm{T}} \hat{Q} z + u^{\mathrm{T}} \hat{R}_1 u + f^{\mathrm{T}} R_2 f \tag{19}$$

其中 $\hat{Q}$ 和 $\hat{R}_1$ 分别为 Q 和 R_1 的估计值。

那么,将哈密顿方程的近似值表示为

$$H(z, \nabla \hat{V}, u, f) = \hat{c}(z, u, f) + \hat{W}^{\mathrm{T}} \sigma := e \tag{20}$$

令式(18)中的 $c(z,u,f) + W^{\mathrm{T}} \sigma + \varepsilon_{\mathrm{H}} = 0$, 联合式(20)可将估计误差 e 表示为

$$e = (\hat{W} - W)^{\mathrm{T}} \sigma + (\hat{c} - c) - \varepsilon_{\mathrm{H}} = \tilde{W}^{\mathrm{T}} \sigma + \tilde{c} - \varepsilon_{\mathrm{H}} \tag{21}$$

其中 $\tilde{W} = \hat{W} - W, \tilde{c} = \hat{c} - c = z^{\mathrm{T}} \tilde{Q} z + u^{\mathrm{T}} \tilde{R}_1 u, \tilde{Q} = \hat{Q} - Q, \tilde{R}_1 = \hat{R}_1 - R_1$。为了便于表示,把参数 Q 和 R_1 包含在矩阵 θ 中,有如下定义,其中 $vec(\cdot)$ 表示对矩阵做列向量化操作。

$$\begin{aligned}\tilde{\theta} &= [vec^{\mathrm{T}}(\tilde{Q}) \quad vec^{\mathrm{T}}(\tilde{R}_1)]^{\mathrm{T}} \\ \theta &= [vec^{\mathrm{T}}(Q) \quad vec^{\mathrm{T}}(R_1)]^{\mathrm{T}} \\ \hat{\theta} &= [vec^{\mathrm{T}}(\hat{Q}) \quad vec^{\mathrm{T}}(\hat{R}_1)]^{\mathrm{T}}\end{aligned} \tag{22}$$

另外,令 $Y = [\bar{z}^{\mathrm{T}} \quad \bar{u}^{\mathrm{T}}]^{\mathrm{T}}$, $\bar{z}$ 和 $\bar{u}$ 分别表示如下:

$$\begin{aligned}\bar{z} = [&z^2(1), z(1)z(2), \cdots, z(1)z(3m), \\ &z(2)z(1), z^2(2), \cdots, z^2(3m)]^{\mathrm{T}}\end{aligned} \tag{23}$$

$$\begin{aligned}\bar{u} = [&u^2(1), u(1)u(2), \cdots, u(1)u(m), \\ &u(2)u(1), u^2(2), \cdots, u^2(m)]^{\mathrm{T}}\end{aligned} \tag{24}$$

然后用 $\tilde{\theta}$ 和 Y 来表示式(21)中的 $\tilde{c}$, 即:

$$\tilde{c} = \tilde{\theta}^{\mathrm{T}} Y \tag{25}$$

因此,结合式(18)和式(21),估计误差改写成如下形式:

$$e = \hat{W}^{\mathrm{T}} \sigma + \hat{\theta}^{\mathrm{T}} Y \tag{26}$$

为了能达到拟合效果,设计了一个二次残差函数 E[27],通过设计合适的 $\hat{W}$ 和 $\hat{\theta}$ 来最小化 E。

$$E = \frac{1}{2} e^{\mathrm{T}} e + \frac{\beta}{2\alpha_1} e_{\mathrm{f}}^{\mathrm{T}} e_{\mathrm{f}} \tag{27}$$

通过最小化二次残差 $\boldsymbol{E}$ 的第一项 $\frac{1}{2}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}$，可以使式(18)中的哈密顿方程最小化，即达到纳什均衡，从而求得最优控制策略 $\boldsymbol{u}^*$，$\boldsymbol{f}$。通过最小化 $\boldsymbol{E}$ 中的第2项 $\frac{\beta}{2\alpha_1}\boldsymbol{e}_{\mathrm{f}}^{\mathrm{T}}\boldsymbol{e}_{\mathrm{f}}$，使得人控制力的误差最小，那么估计损失 $\hat{\boldsymbol{c}}$ 可以更接近实际损失 $\boldsymbol{c}$，从而达到更好的协作。这里，$\boldsymbol{e}_{\mathrm{f}}=\hat{\boldsymbol{f}}-\boldsymbol{f}$ 是人力的误差，$\hat{\boldsymbol{f}}$ 和 $\boldsymbol{f}$ 分别表示人力的估计值和真实值，α_1 和 β 为正数。根据式(14)和(16)，最优机器人和人控制策略可以分别表示为

$$\boldsymbol{u}^* = -\frac{1}{2}\boldsymbol{R}_1^{-1}\boldsymbol{B}_1^{\mathrm{T}}\nabla\boldsymbol{V} = -\frac{1}{2}\boldsymbol{R}_1^{-1}\boldsymbol{B}_1^{\mathrm{T}}(\nabla\boldsymbol{S}^{\mathrm{T}}\boldsymbol{W}+\nabla\boldsymbol{\varepsilon}_{\mathrm{u}}) \tag{28}$$

$$\boldsymbol{f} = -\frac{1}{2}\boldsymbol{R}_2^{-1}\boldsymbol{B}_2^{\mathrm{T}}\nabla\boldsymbol{V} = -\frac{1}{2}\boldsymbol{R}_2^{-1}\boldsymbol{B}_2^{\mathrm{T}}(\nabla\boldsymbol{S}^{\mathrm{T}}\boldsymbol{W}+\nabla\boldsymbol{\varepsilon}_{\mathrm{f}}) \tag{29}$$

其中 $\nabla\boldsymbol{\varepsilon}_{\mathrm{u}} = -\frac{1}{2}\boldsymbol{R}_1^{-1}\boldsymbol{B}_1^{\mathrm{T}}\nabla\boldsymbol{\varepsilon}$，$\nabla\boldsymbol{\varepsilon}_{\mathrm{f}} = -\frac{1}{2}\boldsymbol{R}_2^{-1}\boldsymbol{B}_2^{\mathrm{T}}\nabla\boldsymbol{\varepsilon}$。人的控制力是可以通过力传感器测量得到的。另外，由式(13)得，估计的机器人控制策略 $\hat{\boldsymbol{u}}$ 和人控制策略 $\hat{\boldsymbol{f}}$ 分别表示为

$$\hat{\boldsymbol{u}} = -\frac{1}{2}\hat{\boldsymbol{R}}_1^{-1}\boldsymbol{B}_1^{\mathrm{T}}\nabla\boldsymbol{S}^{\mathrm{T}}\hat{\boldsymbol{W}} \tag{30}$$

$$\hat{\boldsymbol{f}} = -\frac{1}{2}\boldsymbol{R}_2^{-1}\boldsymbol{B}_2^{\mathrm{T}}\nabla\boldsymbol{S}^{\mathrm{T}}\hat{\boldsymbol{W}} \tag{31}$$

根据梯度下降算法，对式(27)分别求 $\hat{\boldsymbol{W}}$ 和 $\hat{\boldsymbol{\theta}}$ 的偏导，再乘上各自的系数，神经网络估计器的权值矩阵 $\hat{\boldsymbol{W}}$ 以及矩阵 $\hat{\boldsymbol{\theta}}$ 的更新率[27]为

$$\begin{aligned}\dot{\hat{\boldsymbol{W}}} &= -\alpha_1\frac{\partial\boldsymbol{E}}{\partial\hat{\boldsymbol{W}}}\\ &= -\alpha_1\boldsymbol{\sigma}(\hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\sigma}+\hat{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{Y})+\beta\nabla\boldsymbol{S}\boldsymbol{B}_2\boldsymbol{R}_2^{-\mathrm{T}}\boldsymbol{e}_{\mathrm{f}}\end{aligned} \tag{32}$$

$$\dot{\hat{\boldsymbol{\theta}}} = -\alpha_2\frac{\partial\boldsymbol{E}}{\partial\hat{\boldsymbol{\theta}}} = -\alpha_2(\hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\sigma}+\hat{\boldsymbol{\theta}}^{\mathrm{T}}\boldsymbol{Y}) \tag{33}$$

其中 α_2 为正常数。

在式(14)神经网络的设计中，采用了径向基函数。其中，径向基函数为 $\boldsymbol{S}(z)=[\boldsymbol{s}_1,\cdots,\boldsymbol{s}_N]^{\mathrm{T}}$，$N$ 代表神经节点的数量，激活函数选择了高斯函数的形式：

$$\boldsymbol{s}_i(z) = \exp\left[\frac{-(z-\boldsymbol{\mu}_i)^{\mathrm{T}}(z-\boldsymbol{\mu}_i)}{\eta_i^2}\right] \tag{34}$$

其中 $\boldsymbol{\mu}_i=[\boldsymbol{\mu}_{i,1},\boldsymbol{\mu}_{i,2},\cdots,\boldsymbol{\mu}_{i,2m}]$ 为激活函数的中心位置，η_i 为高斯函数的宽度（$i=1,\cdots,N$）。对径向基函数的梯度表示为 $\nabla\boldsymbol{S}(z)=[\nabla\boldsymbol{s}_1,\cdots,\nabla\boldsymbol{s}_N]^{\mathrm{T}}$，其中，

$$\nabla\boldsymbol{s}_i(z) = -\frac{2}{\eta_i^2}s_i(z)(z-\boldsymbol{\mu}_i)^{\mathrm{T}} \tag{35}$$

为了保证控制方法的跟踪性，式(34)中激活函数的中心位置 $\boldsymbol{\mu}_i$ 被设计为

$$\boldsymbol{\mu}_i = [(\boldsymbol{x}_{\mathrm{r}}-\boldsymbol{x}_{\mathrm{d}})^{\mathrm{T}},(\dot{\boldsymbol{x}}_{\mathrm{r}}+k(\boldsymbol{x}_{\mathrm{r}}-\boldsymbol{x}_{\mathrm{d}})^{\mathrm{T}})]^{\mathrm{T}} \tag{36}$$

其中，k 是大于0的常数。通过设计激活函数的中心位置，可以获得机器人控制律 $\boldsymbol{u}$ 与误差跟踪的关系 $\boldsymbol{u}=\boldsymbol{K}(\boldsymbol{x}_{\mathrm{d}}-\boldsymbol{x}_{\mathrm{r}})$。结合式(5)中的 $z(t)$ 和 $\boldsymbol{B}_1$，把 $\nabla\boldsymbol{S}(z)$ 代入式(30)可以得到：

$$\begin{aligned}\boldsymbol{u} &= \hat{\boldsymbol{R}}_1^{-1}\boldsymbol{B}_1^{\mathrm{T}}(z-\boldsymbol{\mu}_i)\boldsymbol{S}_\eta^{\mathrm{T}}(z)\hat{\boldsymbol{W}}\\ &= k\boldsymbol{S}_\eta^{\mathrm{T}}(z)\hat{\boldsymbol{W}}\hat{\boldsymbol{R}}_1^{-1}\boldsymbol{M}_{\mathrm{d}}^{-\mathrm{T}}(\boldsymbol{x}_{\mathrm{d}}-\boldsymbol{x}_{\mathrm{r}})\end{aligned} \tag{37}$$

其中 $\boldsymbol{S}_\eta^{\mathrm{T}}(z)=\left[\frac{s_1}{\eta_1^2},\cdots,\frac{s_N}{\eta_N^2}\right]^{\mathrm{T}}$，则：

$$\boldsymbol{K} = k\boldsymbol{S}_\eta^{\mathrm{T}}(z)\hat{\boldsymbol{W}}\hat{\boldsymbol{R}}_1^{-1}\boldsymbol{M}_{\mathrm{d}}^{-\mathrm{T}} \tag{38}$$

$\boldsymbol{K}$ 代表机器人的刚度系数，当 $\boldsymbol{K}$ 比较大的时候，可以较快地修正轨迹跟踪误差，相应地人拖动机器人也较为困难。由式(37)可知，通过更新迭代参数 $\hat{\boldsymbol{W}}$ 和 $\hat{\boldsymbol{R}}_1$ 来改变 $\boldsymbol{K}$ 值从而实现变阻抗控制。通过该方法，当需要跟踪预定轨迹时增大 $\boldsymbol{K}$ 来达到好的跟踪效果，当有人力存在时减小 $\boldsymbol{K}$ 值达到轻松协作的目的。机器人在完成任务时有自己的运动目标，但当人干预时，会把人的目标作为新目标，实现了协调的目的。

具体的自适应算法如下。

(1)设置权值矩阵 $\boldsymbol{R}_2$，初始化式(22)、(31)中的 $\hat{\boldsymbol{Q}}$、$\hat{\boldsymbol{R}}_1$ 和 $\hat{\boldsymbol{W}}$，激活函数为式(34)中的高斯函数，初始化更新律式(32)、(33)中的 α_1、α_2 和 β。

(2)当时间在 $t<t_f$ 时，给定期望轨迹 $\boldsymbol{x}_{\mathrm{d}}$，收集状态数据 $z=[(\boldsymbol{x}_{\mathrm{r}}-\boldsymbol{x}_{\mathrm{d}})^{\mathrm{T}}\dot{\boldsymbol{x}}_{\mathrm{r}}^{\mathrm{T}}]^{\mathrm{T}}$。分别通过式(30)和(31)更新 $\hat{\boldsymbol{u}}$ 和 $\hat{\boldsymbol{f}}$。利用式(32)和(33)更新 $\hat{\boldsymbol{W}}$ 和 $\hat{\boldsymbol{\theta}}$，再根据 $\hat{\boldsymbol{\theta}}$ 获得 $\hat{\boldsymbol{R}}_1$ 的值。

(3)根据更新的参数 $\hat{W}$ 和 $\hat{R}_1$，利用式(38)得到实时的机器人刚度系数 K，直到时间为 t_f 时停止迭代。

3 内环控制器设计

内环控制器的目的是设计力矩，在运动过程中，使机器人真实轨迹 x 跟踪外环中的参考轨迹 x_r，模型跟踪误差定义如下：

$$e_r = x_r - x \tag{39}$$

为了使模型跟踪误差项 e_r 最小，引入比例微分误差函数项 s，

$$s = \dot{e}_r + \Lambda e_r \tag{40}$$

其中 $\Lambda = \Lambda^T > 0$，Λ 为常数矩阵。对式(38)求导代入式(39)可得：

$$\dot{x} = -s + \dot{x}_r + \Lambda e_r \tag{41}$$

再对式(40)求导可得：

$$\ddot{x} = -\dot{s} + \ddot{x}_r + \Lambda \dot{e}_r \tag{42}$$

把式(40)和(41)代入动力学方程式(1)，可得：

$$\begin{aligned} &M(q)(-\dot{s} + \ddot{x}_r + \Lambda\dot{e}_r) + C(q, \dot{q}) \\ &(-s + \dot{x}_r + \Lambda e_r) + G(q) = \tau + f \end{aligned} \tag{43}$$

将式(42)中的 $M(q)\dot{s}$ 转换成如下的形式：

$$M(q)\dot{s} = -C(q, \dot{q})s + h(z) - \tau - f \tag{44}$$

其中，

$$\begin{aligned} h(z) = &M(q)(\ddot{x}_r + \Lambda\dot{e}_r) + C(q, \dot{q})(\dot{x}_r \\ &+ \Lambda e_r) + G(q) \end{aligned} \tag{45}$$

式(44)中包含了未知的动力学模型参数 $M(q)$、$C(q, \dot{q})$ 和 $G(q)$，在设计控制器的过程中，这些参数会影响人机协作的柔顺度。为了使机器人与人更好地配合，利用神经网络拟合动力学模型中不确定参数。$\hat{h}(z)$ 代表神经网络估计项，表示如下：

$$\hat{h}(z) = \hat{W}_n{}^T \phi(z) \tag{46}$$

其中 $z = [x, \dot{x}, \dot{x}_r, \ddot{x}_r, e_r, \dot{e}_r]$ 为神经网络的输入，$\hat{W}_n$ 是神经网络的估计权值矩阵，$\phi(z)$ 为激活函数。由于 $\hat{h}(z)$ 是 $h(z)$ 的估计函数，对于连续函数 $h(z)$，写成：

$$h(z) = W_n{}^T \phi(z) + \varepsilon_n(z) \tag{47}$$

其中 W_n 为神经网络权值矩阵，ε_n 为神经网络拟合误差。在图 1 的内环结构中，自适应神经网络控制器被设计为如下形式：

$$\tau = W_n{}^T \phi(z) + K_v s - f \tag{48}$$

其中 K_v 为控制增益。

基于文献[29]的研究结果，将神经网络权值矩阵的更新率设计成如下形式：

$$\dot{\hat{W}}_n = F\phi(z)s^T - k_n F \| s \| \hat{W}_n \tag{49}$$

其中 $F = F^T > 0$，$k_n > 0$。令 $\tilde{W}_n = W_n - \hat{W}_n$ 为权值估计误差。$\tilde{W}_n$ 的 F 范数 $\| \tilde{W}_n \|_F$ 需满足：

$$\| \tilde{W}_n \|_F > W_{nmax}/2 + \sqrt{W_{nmax}^2/4 + \varepsilon_N/k_n} \tag{50}$$

其中 $\| \tilde{W}_n \|_F \leqslant W_{nmax}$，$W_{nmax}$ 为矩阵 W_n 的最大特征根，ε_N 为正常数，使得 $\| \varepsilon_n \| \leqslant \varepsilon_N$。

定理 2 针对式(47)中设计的自适应神经网络控制器，采用式(48)的更新率且 $\| \tilde{W}_n \|_F$ 满足条件式(49)，则控制器式(47)能够使系统稳定，且能保证式(39)中的模型跟踪误差 e_r 有界。

证明 首先，定义李雅普诺夫函数为

$$L = \frac{1}{2}s^T M(q)s + tr(\tilde{W}_n{}^T F^{-1} \tilde{W}_n) \tag{51}$$

对式(50)求导可得：

$$\dot{L} = s^T M(q)\dot{s} + \frac{1}{2}s^T \dot{M}(q)s + 2tr(\tilde{W}_n{}^T F^{-1} \dot{\tilde{W}}_n) \tag{52}$$

将式(43)中 $M(q)\dot{s}$ 代入式(51)，再将式(46)、(47)代入其中得：

$$\begin{aligned} \dot{L} = &-s^T K_v s + \frac{1}{2}s^T(\dot{M}(q) - 2C(q, \dot{q}))s \\ &+ 2tr\tilde{W}_n(F^{-1}\dot{\tilde{W}}_n + \phi s^T) + s^T \varepsilon_n \end{aligned} \tag{53}$$

由于 W_n 是常值矩阵，对式 $\tilde{W}_n = W_n - \hat{W}_n$ 求导可得 $\dot{\tilde{W}}_n = -\dot{\hat{W}}_n$，根据式(48)得：

$$\dot{\tilde{W}}_n = -F\phi(z)s^T + k_n F \| s \| \hat{W}_n \tag{54}$$

把式(53)代入式(52)得：

$$\begin{aligned} \dot{L} &= -s^T K_v s + tr\tilde{W}_n{}^T(-\phi s^T + k_n \| s \| \hat{W}_n + \phi r^T) \\ &\quad + s^T \varepsilon_n \\ &= -s^T K_v s + k_n \| s \| tr\tilde{W}_n{}^T(W_n - \tilde{W}_n) + s^T \varepsilon_n \end{aligned} \tag{55}$$

由于

$$tr\tilde{\boldsymbol{W}}_{\mathrm{n}}^{\mathrm{T}}(\boldsymbol{W}_{\mathrm{n}}-\tilde{\boldsymbol{W}}_{\mathrm{n}})=(\tilde{\boldsymbol{W}}_{\mathrm{n}},\boldsymbol{W}_{\mathrm{n}})_{\mathrm{F}}-\|\tilde{\boldsymbol{W}}_{\mathrm{n}}\|_{\mathrm{F}}^{2}$$

$$(\tilde{\boldsymbol{W}}_{\mathrm{n}},\boldsymbol{W}_{\mathrm{n}})_{\mathrm{F}}-\|\tilde{\boldsymbol{W}}_{\mathrm{n}}\|_{\mathrm{F}}^{2}\leqslant\|\tilde{\boldsymbol{W}}_{\mathrm{n}}\|_{\mathrm{F}}\|\boldsymbol{W}_{\mathrm{n}}\|_{\mathrm{F}}-\|\tilde{\boldsymbol{W}}_{\mathrm{n}}\|_{\mathrm{F}}^{2}$$

因此

$$\begin{aligned}\dot{\boldsymbol{L}}\leqslant&-K_{\mathrm{vmin}}\|\boldsymbol{s}\|^{2}+k_{\mathrm{n}}\|\boldsymbol{s}\|\|\tilde{W}_{\mathrm{n}}\|_{\mathrm{F}}(\boldsymbol{W}_{\mathrm{nmax}}-\|\tilde{\boldsymbol{W}}_{\mathrm{n}}\|_{\mathrm{F}})\\&+\boldsymbol{\varepsilon}_{N}\|\boldsymbol{s}\|\\=&-\|\boldsymbol{s}\|(K_{\mathrm{vmin}}\|\boldsymbol{s}\|\\&+k_{\mathrm{n}}\|\tilde{\boldsymbol{W}}_{\mathrm{n}}\|_{\mathrm{F}}(\|\tilde{\boldsymbol{W}}_{\mathrm{n}}\|_{\mathrm{F}}-\boldsymbol{W}_{\mathrm{nmax}})-\boldsymbol{\varepsilon}_{N})\\=&-\|\boldsymbol{s}\|(k_{\mathrm{n}}(\|\tilde{\boldsymbol{W}}_{\mathrm{n}}\|_{\mathrm{F}}-\boldsymbol{W}_{\mathrm{nmax}}/2)2-k_{\mathrm{n}}\boldsymbol{W}_{\mathrm{nmax}}^{2}/4\\&+K_{\mathrm{vmin}}\|\boldsymbol{s}\|-\boldsymbol{\varepsilon}_{N})\end{aligned}\tag{56}$$

其中,K_{vmin} 为矩阵 $\boldsymbol{K}_{\mathrm{v}}$ 的最小特征根。把式(49)代入式(55)可得,李亚普诺夫函数的导数 $\dot{\boldsymbol{L}}<0$,从而证明所设计的控制器能够保证系统是稳定的。

4 仿 真

为了验证所提出方法的有效性,本节以二连杆机器人为仿真对象进行了仿真,如图 2 所示。本文的仿真平台为 CoppeliaSim (4.1.0,Windows),联合 Matlab (R2020a)进行仿真。在仿真过程中,移动最左侧的小球被认为施加力,从其初始位置到当前位置的向量表示人在二连杆末端小球上施加的力,其大小与长度成正比。黑色的圆和直线分别为二连杆机器人末端的期望路径。机器人按照预设的路径进行运动,当人给机器人一个外力时,机器人会改变原来的目标去跟随新的轨迹。同时,为了使人能更加容易拖动机器人,机器人的刚度系数也会自适应减小。为了说明本文方法的可行性,设置 2 组仿真,让机器人分别跟踪均匀水平直线和曲线路径。相关的控制参数设置如下:式(34)中的 $\boldsymbol{\eta}_i=100$,式(36)中的 $k=10$,式(3)中的阻抗参数为 $\boldsymbol{M}_{\mathrm{d}}=3\boldsymbol{I}_2\ \mathrm{kg}$,$\boldsymbol{C}_{\mathrm{d}}=30\boldsymbol{I}_2\ \mathrm{N/m}$。二连杆机器人参数为:连杆质量 $m_1=m_2=0.5\ \mathrm{kg}$,连杆长度 $l_1=l_2=0.3\ \mathrm{m}$,$\boldsymbol{I}_1=\boldsymbol{I}_2=1\ \mathrm{kg\cdot m^2}$。式(19)估计权值 $\hat{\boldsymbol{Q}}$ 中的 2 个矩阵的初始值设为 $\hat{\boldsymbol{Q}}_1=100\boldsymbol{I}_2\ \mathrm{m^{-2}}$,$\hat{\boldsymbol{Q}}_2=\boldsymbol{I}_2\ \mathrm{s^2/m^2}$,其中 $\hat{\boldsymbol{Q}}_1$ 和 $\hat{\boldsymbol{Q}}_2$ 分别为 $\boldsymbol{Q}_1$ 和 $\boldsymbol{Q}_2$ 的估计值,$\hat{\boldsymbol{R}}_1=\hat{\boldsymbol{R}}_2=10^{-3}\boldsymbol{I}_2\ \mathrm{N^{-2}}$。关于神经网络,初始值设为 $\boldsymbol{W}=40$,节点数为 6,更新率参数式(32)、(33)设为 $\alpha_1=0.1$,$\alpha_2=10^{-4}$,$\beta=10^{-3}$。更新率参数 α_1 和 α_2 的值会影响神经网络的更新速率,越大更新越快,同时也会使得刚度系数变化更快。

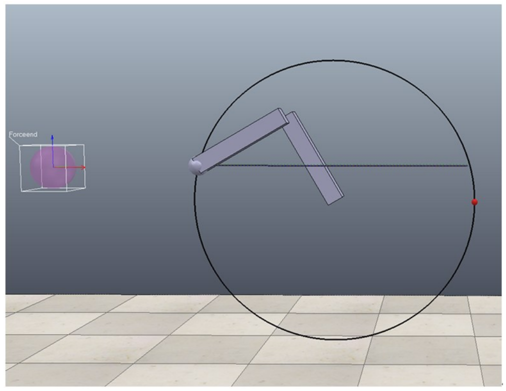
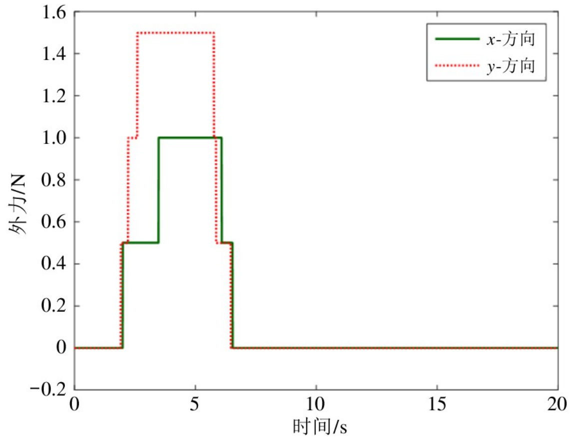


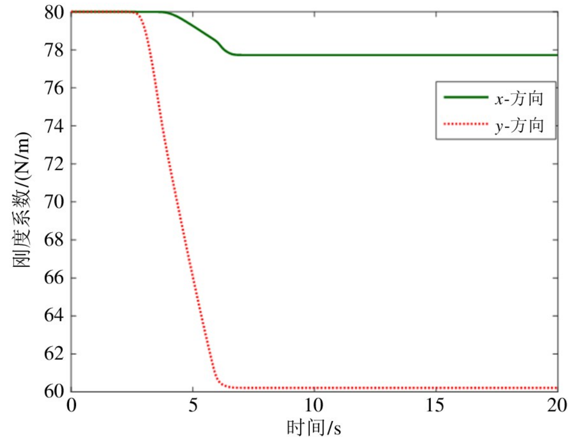
图 2 CoppeliaSim 二连杆仿真场景图

把参考路径设置为均匀水平直线,在不设置外力的情况下,机器人将沿着水平运动。在仿真的第 2 s 左右时,拖动小球施加外力,使机器人跟踪人的目标。施加的外力如图 3(a),分别表示 x 方向和 y 方向力的大小,在 7 s 左右释放。刚度系数 $\boldsymbol{K}$ 变化曲线图如 3(b)所示,当人施加力时,机器人会改变其预期的轨迹,同时刚度系数会适应人力逐渐减小,再趋于稳定。二连杆机器人的运动轨迹以及 x 和 y 两个方向关于时间的轨迹跟踪分别如图 3(c)、(d)和(e)所示。由图可知,在人不施加力时,能较好跟踪期望轨迹执行目标。

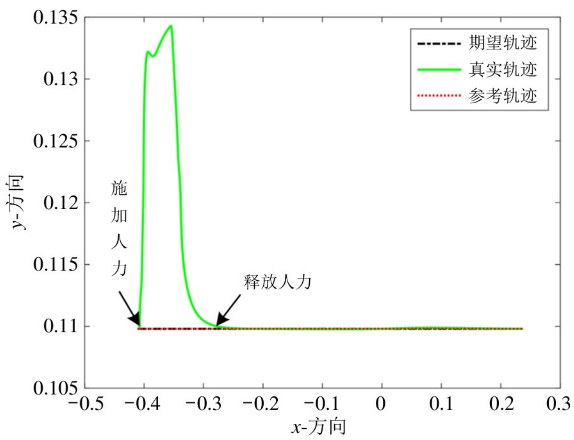
同样,当机器人沿着圆弧曲线轨迹运动时,图 4(a)为 x 和 y 两个方向力的变化曲线,在 1 s 左右施加外力,4 s 左右释放。图 4(b)为刚度系数 $\boldsymbol{K}$ 的变化曲线,且在施加外力后,刚度系数 $\boldsymbol{K}$ 也自适应减小。图 4(c)为二连杆机器人的曲线运动轨迹。由图可得,机器人先跟踪黑色曲线的部分,再跟随外力的方向,当外力释放后会继续跟踪原目标轨迹,$\boldsymbol{K}$ 的变化使人能更容易拖动机器人。图 4(d)和(e)分别为 x 和 y 两个方向关于时间的轨迹跟踪曲线。在不施加外力时,机器人能很好地跟踪目标轨迹。当机器人占据主导地位时,能较好地跟踪期望目标执行任务,当人加入任务中时,则人将成为为任务主导者。2 次仿真验证了本文方法的可行性和准确性。



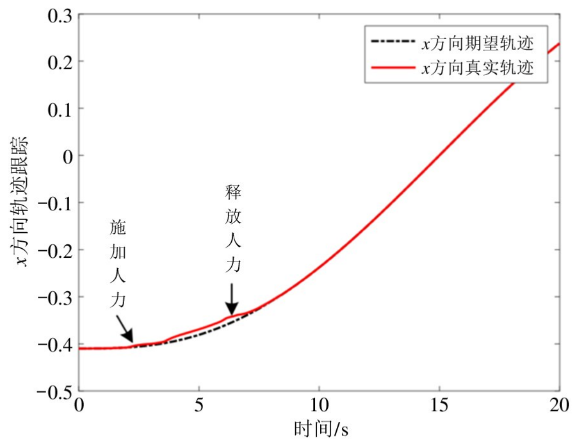
（a）x 方向和 y 方向施加的力



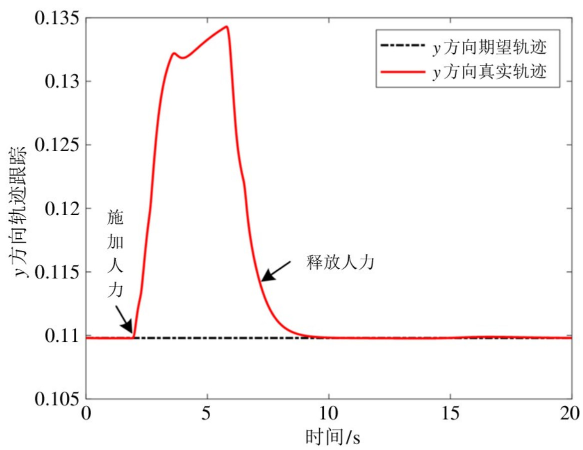
（b）刚度系数 $\boldsymbol{K}$ 变化



（c）机器人的直线运动轨迹图



（d）x 方向的轨迹跟踪

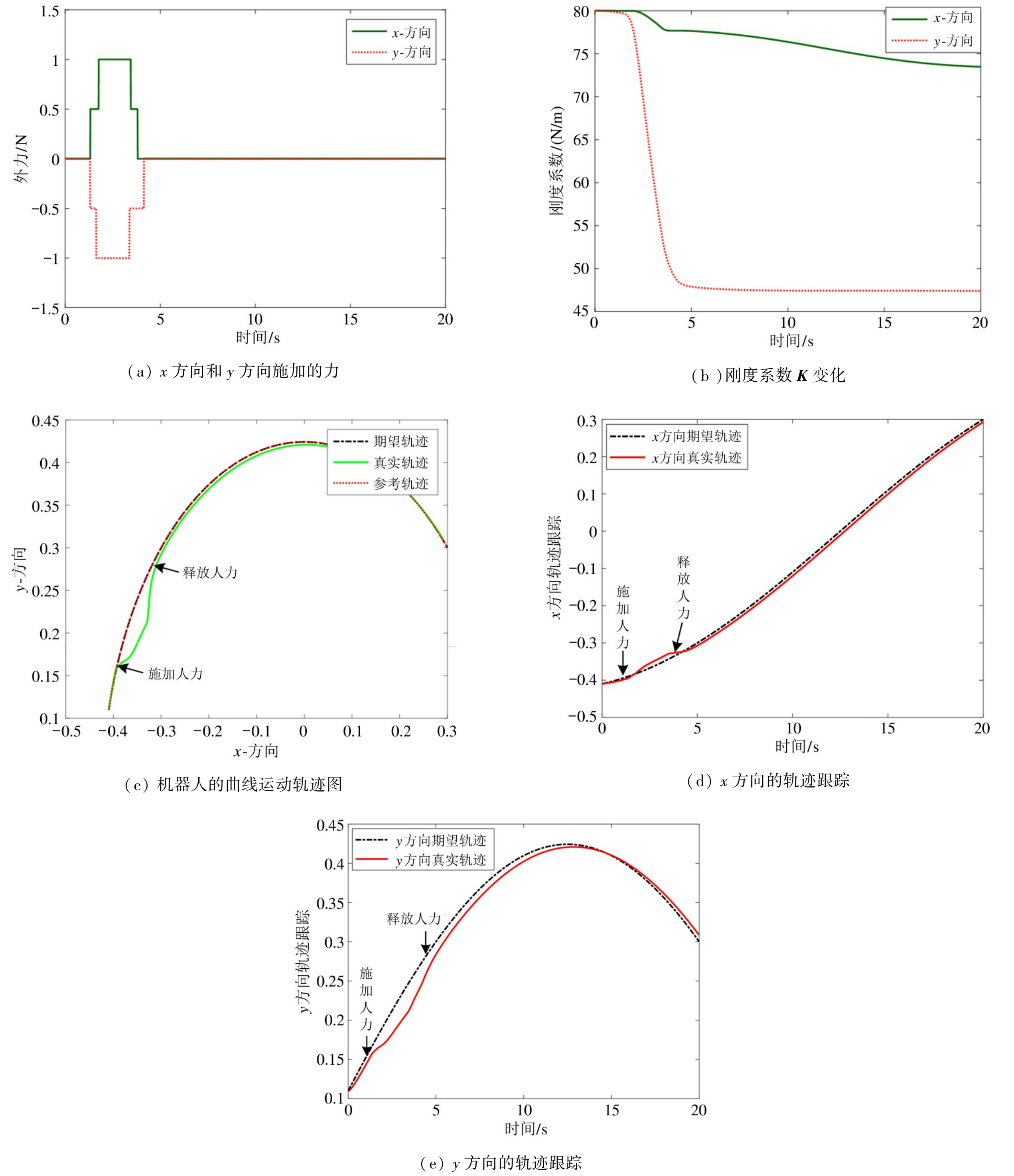


（e）y 方向的轨迹跟踪

图 3　轨迹为均匀水平直线的仿真图

5　结　论

本文提出了一种基于非零博弈论的人机协作的自适应系统，通过切换人和机器人在协作中的主导位置来提高协作的协调性。该系统由互相解耦的内外环构成。在外环中，引入了基于非零和博弈，构建关于人力和机器人控制输入的能量函数，求解人机交互力的纳什均衡以得到最优控制。能量函数中的



(a) x 方向和 y 方向施加的力

(b) 刚度系数 $\boldsymbol{K}$ 变化

(c) 机器人的曲线运动轨迹图

(d) x 方向的轨迹跟踪

(e) y 方向的轨迹跟踪

图4　轨迹为曲线的仿真图

不确定参数通过神经网络估计器迭代更新,在更新过程中自适应调整刚度系数,使其能够灵活地在人机协作或者仅机器人的情况下进行任务转换。通过设计径向基函数中心值,保证控制方法的跟踪性。在内环中,设计了神经网络控制器,采用径向基神经网络,机器人系统的输入输出数据被实时采集来逼近机器人动力学模型同时提高跟踪精度。仿真结果验证了本文所提方法的有效性。

参考文献

[1] ROVEDA L, PALLUCCA G, PEDROCCHI N, et al. Iterative learning procedure with reinforcement for high-ac-

curacy in robotized tasks[J]. IEEE Transactions on Industrial Informatics, 2017,14(4):1753-1763.

[2] SHIMMURA T, ICHIKARI R, OKUMA T. Human-robot hybrid service system introduction for enhancing labor and robot productivity[C]//Proceedings of the IFIP WG 5.7 International Conference on Advances in Production Management Systems. Novi Sad, Serbia:Springer, 2020:661-669.

[3] SHI D, ZHANG W X, ZHANG W, et al. Human-centered adaptive control of lower limb rehabilitation robot based on human-robot interaction dynamic model[J]. Mechanism and Machine Theory, 2021,162:104340.

[4] 毕如奇. 基于阻抗控制的工业机器人力控制方法[J]. 机械制造与自动化, 2021,50(6):178-180,198.

[5] 陈满意, 朱义虎,韩天勇, 等. 基于轨迹修正的曲面抛光机器人终端滑模导纳控制[J]. 计算机集成制造系统, 2022:1-16. http://kns.cnki.net/kcms/detail/11.5946.TP.202/1022.1517.018.html.

[6] HOGAN N. Impedance control: an approach to manipulation: Part II-Implementation[J]. Transactions of the ASME, 1985,107:8-16.

[7] RIENER R, LUNENRGER L, JEZERNIK S, et al. Patient-cooperative strategies for robot-aided treadmill training: first experimental results[J]. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2005,13(3):380-394.

[8] PÉREZ-IBARRA J C, SIQUEIRA A A G, SILVA-COUTO M A, et al. Adaptive impedance control applied to robot-aided neuro-rehabilitation of the ankle[J]. IEEE Robotics and Automation Letters, 2018,4(2):185-192.

[9] RIZVI S A A, LIN Z L. Reinforcement learning based Linear quadratic regulation of continuous-time systems using dynamic output feedback[J]. IEEE Transactions on Cybernetics, 2019,50(11):4670-4679.

[10] PANE Y P, NAGESHRAO S P, KOBER J, et al. Reinforcement learning based compensation methods for robot manipulators[J]. Engineering Applications of Artificial Intelligence, 2019,78:236-247.

[11] YANG J T, PENG C. Adaptive neural impedance control with extended state observer for human-robot interactions by output feedback through tracking differentiator[J]. Proceedings of the Institution of Mechanical Engineers Part I: Journal of Systems and Control Engineering, 2020,234(7):820-833.

[12] LIU X, GE S S, ZHAO F, et al. Optimized impedance adaptation of robot manipulator interacting with unknown environment[J]. IEEE Transaction on Control Systems Technology, 2020,29(1):411-419.

[13] MORTL A, LAWITZKY M, KUCUKYILMAZ A, et al. The role of roles: physical cooperation between humans and robots[J]. International Journal of Robotics Research, 2012,31(13):1656-1674.

[14] EVRARD P, KHEDDAR A, HOMOTOP Y. Switching model for dyad haptic interaction in physical collaborative tasks[C]//The 3rd Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. Salt Lake City, USA: IEEE, 2009:45-50.

[15] MEDINA J R, LAWITZKY M, MOLIN A, et al. Dynamic strategy selection for physical robotic assistance in partially known tasks[C]//Proceedings of the 2013 IEEE International Conference on Robotics Automation. Karlsruhe, Germany: IEEE, 2013:1180-1186.

[16] JARRASSE N, CHARALAMBOUS T, BURDET E. A framework to describe, analyze and generate interactive motor behaviors[J]. PLoS ONE, 2012,7(11):e49945.

[17] 刘莎, 张硕, 刘禄. 基于动态非零和博弈的无人机集群协同对抗方法研究[J]. 航空科学技术, 2022,33(2):75-83.

[18] KIRK D E. Optimal control theory: an introduction[M]. New York, USA: Dover, 2012.

[19] LEWIS F L, VRABIE D. Reinforcement learning and adaptive dynamic programming for feedback control[J]. IEEE Circuits System Magazine, 2009,9(3):32-50.

[20] VRABIE D, LEWIS F L. Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems[J]. Neural Networks, 2009,22(3):237-246.

[21] JIANG Y, JIANG Z P. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics[J]. Automatica, 2012,48(10):2699-2704.

[22] VRABIE D, PASTRAVANU O, ABU-KHALAF M, et al. Adaptive optimal control for continuous-time linear systems based on policy iteration[J]. Automatica, 2009,45(2):477-484

［23］ VAMVOUDAKIS K G, LEWIS F L. Multi-player non-zero-sum games: online adaptive learning solution of coupled Hamilton-Jacobi equations[J] Automatica, 2011,47(8):1556-1569.

［24］ ZHANG H, WEI Q, LIU D. An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games[J]. Automatica, 2011,47(1):207-214.

［25］ 陈银燕, 高安邦. 机器人导航路径的多种群博弈蚁群规划策略[J]. 机械设计与制造, 2021(1):272-276, 281.

［26］ TEE K P, YAN R, LI H. Adaptive admittance control of a robot manipulator under task space constrain[C]//Proceedings of the 2010 IEEE International Conference on Robotics and Automation. Anchorage, USA: IEEE, 2010: 5181-5186.

［27］ LI Y N, TEE K P, YAN R, et al. A framework of human-robot coordination based on game theory and policy iteration[J]. IEEE Transactions on Robotics, 2016,32(6):1408-1418.

［28］ 梁捷, 陈力, 梁频. 柔性臂空间机器人的神经网络自适应控制及振动模态分级模糊控制[J]. 计算机集成制造系统, 2012,18(9):1930-1937.

［29］ 刘金琨. 机器人控制系统的设计与MATLAB仿真[M]. 北京:清华大学出版社,2008.

Design of adaptive human-robot coordination system based on non-zero-sum game theory

YU Xinyi, LUO Huizhen, SHI Shuanwu, WEI Yan, OU Linlin

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023)

Abstract

In order to improve the coordination of human-robot cooperation, this paper proposes an adaptive system of human-robot interaction based on non-zero-sum game. The system consists of inner loop and outer loop which are decoupled. In the outer loop, the human-robot cooperation control is designed by introducing a non-zero-sum game method, and an energy function about human and robot force is constructed, and the optimal control is achieved by solving the Nash equilibrium. The neural network estimator is employed to update the uncertain parameters in the energy function and estimate the output force of human and robot. By designing the central value of the neural network function, the relationship between the robot force and the tracking error is obtained to ensure the tracking performance of the method. During the update process, the stiffness coefficient is adaptively adjusted when external force exists, so as to realize the compliance and coordination of human and robot. In addition, a neural network controller is designed in the inner loop, and the radial basis neural network is applied to approximate the unknown robot dynamics model using the collected input and output data of robot system, which improves the tracking accuracy of the system. Simulation results verify the effectiveness of this method.

Key words: human-robot coordination, adaptive impedance control, non-zero-sum game, neural network