

机器人作业的离散事件建模与控制方法*

李 杰, 韦 庆, 常文森

(国防科技大学 自动控制系, 长沙 410073)

摘要: 针对传统机器人控制中规划器开环的不足, 首先设计了底层的非时间参考的连续路径规划与控制方法, 使规划器的规划值基于系统的状态之上, 避免了基于时间规划的系统由于意外事件造成的损害和任务的重新规划; 在此基础上, 又建立了整个作业任务的离散事件模型, 设计了离散事件路径规划方法及控制综合方法, 实现了事件反馈的闭环, 提高了系统处理不确定事件的能力和作业的自动化程度. 最后两台 PUMA 560 机械手搬运被识别工件的作业验证了本文算法的有效性.

关键词: 混合系统; 非时间参考的路径规划; 离散事件建模

中图分类号: TP242 **文献标识码:** A

Discrete event modeling and control approach for robotic tasks

LI Jie, WEI Qing, CHANG Wen-sen

(Department of Automatic Control, National University of Defense Technology, Changsha 410073, China)

Abstract: In order to conquering the disadvantage of the open loop planning in the traditional robot controller, this paper designs the non-time-based continuous path planning and control method first. It enables the controller's planning value based on the system's states to avoid the damage due to the unexpected events and the replanning of the task in the time-based system. And then we set up the discrete event model for the whole task and provide the discrete event path planning and control synthesis methods. Those methods closed the event feedback loop and improved the system's capability to couple with the unexpected events and the automation degree. The task for two PUMA 560 manipulators to carry the recognized objects verified the effectiveness of this arithmetic finally.

Key words: hybrid system; non-time-based path planning; discrete event modeling

1 引言(Introduction)

随着机器人应用领域的不断扩大, 其工作环境也日趋复杂, 确定环境中的机器人作业技术已经不能满足生产的需要. 除太空、深海、核工业等人类不宜接近的环境外, 一般的工业生产中环境的变化也是难以预料的. 在这种环境中作业, 机器人必须具有处理不确定性事件的能力, 在保证效率的同时还要保证安全性, 避免意外事件造成的损失.

在传统的机器人控制系统中, 规划器是开环的, 它的输出是时间的函数, 不包含机器人和环境的实时信息, 如图 1 所示. 这样当机器人被障碍卡阻时, 规划器的输出仍然随时间而变化. 由于控制误差的积累, 最终将导致机器人本体或被加工工件的损坏; 如果系统采取了保护措施, 避免了硬件系统的损坏, 但是要完成这次任务, 必须重新进行任务或路径的

规划, 因为本次的规划由于时间的演化已经无效. 这将严重影响机器人系统的工作效率, 特别是在人不能接近的环境中, 要恢复正常的工作状态往往需要大量的额外工作.

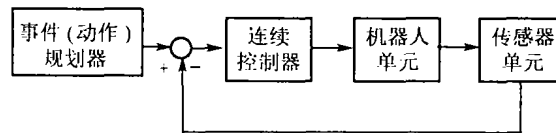


图 1 基于时间的机器人控制器
Fig. 1 Time-based robotic controller

为了提高机器人的适应性, 近年来科研工作者们提出了不少新的理论方法, 如离散事件动态系统方法, 它可以实现任务调度层次的反馈闭环, 但对于机器人连续运动过程中任意时刻的事件检测, 需要借助传感器才能完成. 由于受传感器数量和安装位置的限制, 检测的事件是很有限的. 例如, 当机械手

* 基金项目: 国家自然科学基金(69685009)和 863 智能机器人基础研究(9805 - 02)资助项目.
收稿日期: 2000 - 09 - 05; 收修改稿日期: 2001 - 07 - 10.

与环境碰撞时,只有发生在传感器部分的碰撞才能被检测,而对其它大部分面积上的碰撞都无能为力.

针对这个问题,Ning X.^[1,2]等人提出了一种基于机器人运动路径长度的连续轨迹规划方法,并设计了任务的双子代数模型.其基本思想是使规划器的输出基于机器人当前的移动距离.当机器人被卡阻时,规划器的输出停止变化,由于外在力矩与控制力矩的平衡,机器人就停留在当前位置,无需传感器检测碰撞;当阻力矩消失时,机器人又可继续作业,避免了任务的重规划.但他们的方法在实际控制时会有机器人无法启动或速度不连续的现象,而且他们的离散模型考虑了机器人每个动作的执行时间,如果机器人被卡阻的时间过长,规划器的规划值将出现时间上的混乱.

我们借鉴 Ning X.的“非时间参考”的连续路径规划思想,修正了控制力矩,使之成为了可以真正实用的控制策略;在离散层,我们用直观的 Petri 网模型代替了他们的双子模型,设计了在线的事件路径规划方法,解决了他们可能出现的时间混乱问题,实现了事件反馈的闭环,结构图如图 2 所示.

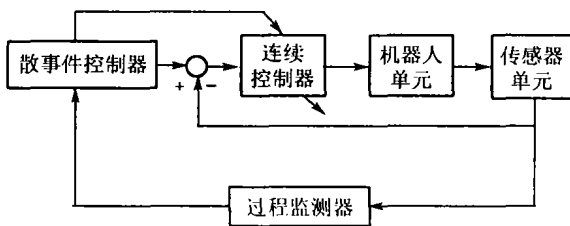


图 2 基于事件反馈的机器人控制器
Fig. 2 Event-based robotic controller

2 非时间参考的连续路径控制 (Non-time-based continuous path control)

2.1 单机器人非时间参考的路径规划 (Single robot non-time-based path planning)

机器人沿给定路径的运动可以描述为

$$\begin{cases} ds/dt = v, & |v| \leq v_m, \\ dv/dt = a, & |a| \leq a_m. \end{cases} \quad (1)$$

其中 s, v 和 a 分别为位置、速度和加速度, v_m 和 a_m 分别为最大速度和加速度约束.

用机器人的移动距离 s 代替时间参考 t , 并令 $w = v^2, u = da/ds$, 将 w 对 s 微分得到

$$\begin{cases} dw/ds = 2a, & |a| \leq a_m, |w| \leq w_m, \\ da/ds = u, & |u| \leq \mu_m. \end{cases} \quad (2)$$

其中 $w_m = v_m^2$ 为最大速度约束, μ_m 为最大加速度约束.

令 $x_1 = w, x_2 = a$, 则方程(2)对应的状态空间

形式为

$$\frac{d}{ds}X = FX + Bu. \quad (3)$$

其中 $X = [x_1 \ x_2]^T, F = \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. 显然该方程在给定边界条件下的解不唯一. 我们设计方程(3)的最短时间优化指标为

$$\min J = \min \int_{s_0}^{s_f} \frac{1}{v} ds = \min \int_{s_0}^{s_f} \frac{1}{\sqrt{x_1}} ds.$$

给定边界条件 $X(s_0) = 0, X(s_f) = 0$. 由 Pontryagin 最大值原理^[3]得到 u 的最优解为:

$$u = \begin{cases} u_m, & s \in [s_0, s_1], s_1 = s_0 + a_m/u_m, \\ 0, & s \in [s_1, s_2], s_2 = s_0 + w_m/2a_m, \\ -u_m, & s \in [s_2, s_3], s_3 = s_0 + w_m/2a_m + a_m/u_m, \\ 0, & s \in [s_3, s_4], s_4 = s_f - w_m/2a_m - a_m/u_m, \\ -u_m, & s \in [s_4, s_5], s_5 = s_f - w_m/2a_m, \\ 0, & s \in [s_5, s_6], s_6 = s_f - a_m/u_m, \\ u_m, & s \in [s_6, s_f]. \end{cases} \quad (4)$$

再利用关系 $a = \int_{s_0}^s u ds$ 和 $v = \sqrt{w} = \sqrt{\int_{s_0}^s 2a ds}$, 可以求得加速度和速度的最优值. 这些值只是期望轨迹的幅值,其方向要根据实际轨迹来确定.

2.2 非时间参考的机器人控制 (Non-time-based robotic control)

机器人经过前馈和反馈线性化后, PD 形式的反馈控制律 u 为:

$$u = A^d(t) + k_v(V^d(t) - V(t)) + k_p(X^d(t) - X(t)). \quad (5)$$

其中, $A^d(t), V^d(t)$ 和 $X^d(t)$ 分别为基于时间规划的机器人的期望加速度、速度和位置. k_v 和 k_p 分别为速度和位置反馈增益.

对于非时间参考的运动规划,对应的控制律为:

$$u = A^d(s) + k_v(V^d(s) - V(s)) + k_p(X^d(s) - X(s)). \quad (6)$$

其中 $A^d(s), V^d(s)$ 和 $X^d(s)$ 分别为基于路径长度规划的期望加速度、速度和位置. $V(s)$ 和 $X(s)$ 为实际测量的机器人的速度和位置.

式(5)和式(6)在形式上一样的完美,但仔细分析会发现,Ning X.的控制律(6)是有问题的.我们知道,稳定的 PID 控制中,机器人的运动实际上是由误差驱动的.在式(6)中, $A^d(s), V^d(s)$ 和 $X^d(s)$ 是机器人的移动路径 s 的函数.机器人从静止到运动

的瞬间, $s = s_0$, 由规划的边界条件得 $A^d(s_0) = V^d(s_0) = 0, X^d(s_0) = X(s_0)$, 代入式(6)得到: $u = 0$. 这意味着对机器人施加的控制力矩为 0, 机器人不会动作. 机器人不运动, s 就不会增加, $u = 0$ 依然成立. 因此, 控制律(6)无法使机器人产生运动, 这是一个矛盾. 另外, 当机器人以较低的速度运行时, 如果有个小的扰动使得机器人在某个控制周期没有向前运动(这是经常发生的), 则 s 不会增加, 下个周期的控制力矩也不会增加, 那么机器人就停在了当前位置, 或者以不连续的速度走走停停.

从上述分析看到, 要解决这两个问题, 必须在机器人的 PD 控制律之外施加一个产生初始运动并足够克服运动过程中的扰动的能量(单纯的摩擦补偿是不够的). 经过分析和设计, 我们提出在控制律(6)中增加如下的非线性项:

$$p_{\text{bias}} = \begin{bmatrix} \cos\alpha \\ \cos\beta \\ \cos\gamma \end{bmatrix} \cdot \frac{k_0}{1 + \exp(k_1 - k_2(s_f - s))}. \quad (7)$$

式中, $[\cos\alpha \ \cos\beta \ \cos\gamma]^T$ 为机器人的轨迹 $\vec{p_0 p_1}$ 在参考坐标系内的方向余弦; s_f 为终点对应的路径长度; k_0, k_1 和 k_2 为正常数, 其中 k_0 决定偏值力矩 p_{bias} 的大小, k_0 越大, 机器人的抗扰动能力越强; k_1 的作用是当机器人接近终点时, 使 $p_{\text{bias}} \rightarrow 0$, 减小它对终点处的控制精度的影响; k_2 是路径长度差值的加权系数. 当机器人的位置离终点较远时, p_{bias} 近似为一常数, 它驱使机器人运动并克服扰动; 而当离终点较近时, 该项的值接近 0, 不影响终点处的定位精度. p_{bias} 的加入解决了从规划到控制的实现问题.

对于姿态的控制, 可以用机器人转过的弧度作为动作参考, 控制律与上述类似.

2.3 多机器人间的非时间规划问题(Non-time-based planning for multi-robot)

前面介绍的是单机器人点到点(相邻两事件)间的规划问题. 当多机器人协作执行任务时, 有必要设计一个起同步作用的“共同的动作参考”. 这个参考反应所有机器人的状态, 当其中一个被阻挡时, 这个动作参考停止变化, 整个系统就象一个机器人一样停止运行.

设有 n 个机器人的动作要同步, 对应的动作集合表示为 $T = \{e_i \mid i = 1, 2, \dots, n\}$, 设第 i 台机器人的动作参考为 s_i , 完成动作 e_i 所要经过的路径长度为 l_i , 即有 $l_i = s_f - s_0^i$. 注意 $l_i \neq 0$. 设欲求的共同动作参考为 $s^*(T)$, 由它计算各机器人的动作参考的

步骤^[1]为:

1) 选择最长的路径:

$$l_{\max} = \max_{i \in \{1, \dots, n\}} l_i.$$

2) 计算归一化系数:

$$k_i = \frac{l_i}{l_{\max}} \text{ 对所有的 } i.$$

3) 求共同的动作参考:

$$s^*(T) = \max_{i \in \{1, \dots, n\}} \frac{s_i}{k_i}.$$

4) 计算各机器人的动作参考:

$$s_i = k_i \cdot s^*(T).$$

在控制过程中, 不断重复 3) 和 4), 就可以得到共同动作参考作用下各机器人的动作参考. 该过程中, 如果某个 s_i 停止增加, 由于步骤 3) 的作用, $s^*(T)$ 也停止增加, 则各 s_i 也不再变化.

设一个任务中需同步的动作集合为 $\Sigma = \{T_j \mid j = 1, 2, \dots, l\}$, 对应于任务 T_j 的动作参考为 $s^*(T_j)$, 则该任务“统一的动作参考”序列为:

$$S^* = \{s^*(T_j) \mid j \in 1, 2, \dots, l\}. \quad (8)$$

在任务的执行过程中, S^* 中的元素按事件发生的顺序逐个发挥作用.

3 离散事件控制器设计(Design of discrete event controller)

下面以两台机械手搬运工件为例, 介绍我们的 Petri 网^[4]建模方法及事件路径的动态规划方法.

双手搬运工件任务的描述为: 首先由视觉系统识别并定位工件; 然后手 1 在这个定位信息引导下抓取工件, 运动到指定位置; 手 2 运动到相同的位置, 从手 1 中接过工件, 并将工件送到另一指定位置; 最后双手回到初始位置, 等待下一次搬运任务.

双手搬运任务的 Petri 网模型如图 3 所示. 图中的位置 p_i 表示机械手的状态, 转换 t_i 表示机械手从一个状态运动到另一个状态. 位置中的令牌用“·”表示, 个数限制在 0 和 1 之间, 1 标识机械手处于该状态, 0 表示不在该状态. 图中各位置的具体含义如下:

- p_1 视觉系统得到了被识别工件的位置;
- p_2, p_3 分别表示机械手 2 和 1 在初始位置且空闲;
- p_4 机械手 1 位于工件上方;
- p_5 机械手 1 抓持工件;
- p_6 表示机械手 2 和 1 位于待交接工件的位置;

- p_7 机械手 1 已释放工件;
- p_8 机械手 2 抓持工件;
- p_9 机械手 2 位于释放工件的位置.

各转换的含义是:

- t_1 表示机械手 1 从空闲位置运动到工件上方的过程;
- t_2 表示机械手 1 抓取工件的过程;
- t_3 为双手到达交接工件的位置的过程;
- t_4 双手交接工件过程;
- t_5 机械手 2 到达释放工件位置的过程;
- t_6 为机械手 1 由交接位置到初始位置的过程;
- t_7 为机械手 2 释放工件的过程.

图中的位置集合为 $P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9\}$, 位置的个数 $\rho = 9$; 转换的集合为 $\Gamma = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$, 转换的个数 $\tau = 7$; 标识矢量为 $\gamma = [1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$ 表示两机械手处于空闲位置, 且已经定位了一个工件.

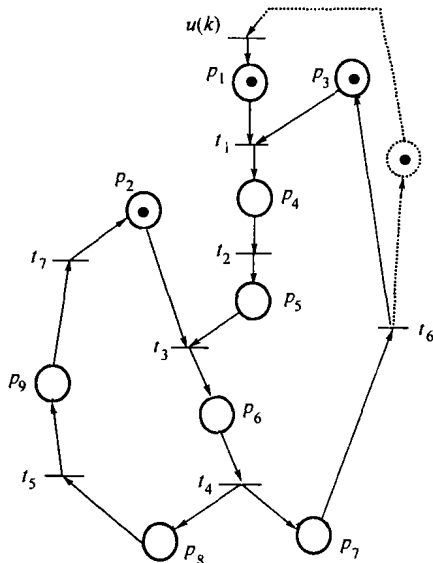


图 3 双手搬运的 Petri 网模型
Fig. 3 Petri net model for the carry task

上述转换中, t_3 包含双手的动作, 因此它需要同步并建立双手共同的动作参考; 另外在我们的模型中, 为了使网络稳定, 可以用图中虚线所示的反馈环来限制输入位置 p_1 内的令牌数量.

为了实现模型的在线计算, Petri 网可以用矩阵来表示^[5]. 我们定义一个输入矩阵 $N^- \in \mathbb{R}^{\rho \times \tau}$ 和一个输出矩阵 $N^+ \in \mathbb{R}^{\rho \times \tau}$ 分别表示网络的输入和输出函数. N^- 中的元素 $n_{i,j}^- = 1$ 表示各位置是第 j 个转换的输入, 否则它为 0; 类似地 N^+ 中的元素 $n_{i,j}^+ = 1$ 表示第 j 个转换的输出是第 i 个位置, 否则它为 0.

图 3 对应的输入和输出矩阵分别为:

$$N^- = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, N^+ = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{9}$$

式(9)中的两个矩阵决定着网络的演化. 我们定义下面的矢量为转换矢量:

$$t_j = [0 \ \dots \ 0 \ 1 \ 0 \ \dots \ 0] \in \mathbb{R}^\tau, 1 \leq j \leq \tau. \tag{10}$$

其中只有第 j 个元素为 1, 其它为 0, 它表示发生了事件 t_j . 那么, 事件发生后网络标识的变化为

$$\gamma(k) = \gamma(k-1) - N^- \cdot t_j + N^+ \cdot t_j \triangleq \gamma(k-1) + N \cdot t_j. \tag{11}$$

式中, $\gamma(k), \gamma(k-1)$ 分别为当前和上一时刻的标识; $N^- \cdot t_j$ 为由于事件 t_j 的发生引起的网络标识的减少; 而 $N^+ \cdot t_j$ 为由于事件 t_j 的发生引起的网络标识的增加; $N = N^+ - N^-$ 为由网络结构决定的标识变化矩阵, 我们称它为合成变化矩阵.

下面的问题是, 如果知道网络当前的标识, 那么下一时刻哪个事件要发生? 即网络的事件路径规划问题. 结合上面的输入输出矩阵, 我们提出了一种可通过编程实现的在线的事件路径的规划方法, 可根据当前标识决定下一时刻要发生的事件, 其步骤如下:

- 1) 计算各转换的输入位置数 I_j :

$$I_j = \sum_{i=1}^{\rho} n_{i,j}^-, j = 1, 2, \dots, \tau.$$

- 2) 计算各转换的条件数 C_j :

- a) $C_j = 0, i = 1$;
- b) 如果 $n_{i,j}^- > 0$: 若 $\gamma_i = 1$, 则 $C_j = C_j + 1$;
若 $\gamma_i = 0$, 则停止第 2) 步的计算.
- c) $i = i + 1$, 重复 b), 直到 $i = \rho$.

- 3) 判断应发生的事件:

如果 $C_j = I_j$, 则第 j 个事件发生; 如果 $C_j < I_j$, 条件不满足; 令 $j = j + 1$, 重复 2) 和 3), 直到 $j = \tau$.

通过上述的规划, 可以得到当前标识 $\gamma = [\gamma_1, \dots, \gamma_i, \dots, \gamma_\rho]$ 下欲发生的事件 j , 然后由式 (11), 就可以决定网络标识的变化.

知道了欲发生的事件 j 后,就可以选择式(8)中的动作参考,离散事件控制器就可以控制连续控制器去实现这个事件.一般来说,完成的动作不同,连续控制器的控制参数也不相同.我们可以将完成各事件对应的参数制成表格,存储在系统中.当某事件被确定后,离散事件控制器就可根据对应参数完成对连续控制器的规划和控制.这种控制综合方法简单,所需规划时间少,适合于实时控制.

4 实验(Experiment)

双机械手搬运被识别工件的实验是在我们的 GKD4^[6]系统上实现的.其中一块 MVME162 单板机作为上层的离散事件控制器;另两块 MVME165 单板机各为一手的连续控制器.我们将矩阵 N^- , N 以及对应各事件的参数值存储于离散事件控制器中.离散控制器动态决策出下一个要执行的事件,然后查表获得实现该事件的参数并下达给连续控制器.下层的连续控制器以非时间参考的连续路径规划方法控制机械手跟踪期望轨迹.被搬运工件的位置由视觉系统获得,当一个对象的位置被确定后,离散控制器就将其对应的标识置为 1,为避免出现过多的令牌,我们按图 3 虚线所示对视觉系统进行了反馈控制.

实验过程中,我们人为地给机械手设置障碍,比如用木棒阻挡机械臂的任何部位.对于不需要同步的动作,被阻挡的机械手会停止运动;对于需要同步的动作,当其中的一台机械手被阻挡停止运动后,另外一台机械手也同时停止运动.当移开障碍后,整个系统又恢复正常运行,不会象基于时间的规划那样使系统陷入瘫痪状态.

通过实验我们看到本文的方法有两个特点:一是双手可以按模型中令牌的流动并行运转,提高了系统的工作效率;二是无论机械手被卡阻的时间多

长,障碍撤除后系统都能正确的恢复运行,这是我们提出的非计时离散模型的优点.

总之,本文介绍的非时间参考的连续路径规划方法,可以使机器人实时地响应各种碰撞;采用离散事件反馈方法,可以使系统根据环境的变化动态决策自己的行为,提高了机器人系统在不确定性环境中操作的安全性和可靠性.

参考文献(References)

- [1] Ning X, Tam T J. Hybrid system approach for task planning and control of robotic manufacturing [A]. Proc. IFAC'96 World Congress [C]. San Francisco, California, USA, 1996
- [2] Ning X, Tam T J. Action synchronization and control of internet based telerobotic systems [A]. Proc. IEEE Int. Conf. Robotics and Automation [C]. Detroit, Michigan, USA, 1999, 219-224
- [3] Wu Qi. Principle of Automatic Control [M]. Beijing: Press of Tsinghua University, 1992
- [4] Peng Y J, Xiao Y H, Peng C, et al. Discrete Event Dynamic System [M]. Changsha: Press of Hunan Science Technology, 1993
- [5] McCarragher B J and Asada H. The discrete event modeling and trajectory planning of robotic assembly tasks [J]. ASME J. Dynamic System, Measurements and Control, 1995, 117(3): 394-400
- [6] Li Jie. The study of supervised control for coordinated multiple manipulators [D]. Changsha: National University of Defense Technology, 1999

本文作者简介

李杰 1971年生.1994年于国防科技大学自动控制系获学士学位,1999年获得控制理论与控制工程专业工学博士学位,现在香港科技大学作博士后研究工作.研究兴趣:多机器人协调,智能机器人. Email: eejieli.cn@yahoo.com.cn

韦庆 1969年生.现为国防科技大学自动控制系副教授.研究领域:机器人力控制,智能机器人.

常文森 1936年生.教授,博士生导师.研究领域:磁悬浮控制,智能机器人,移动机器人等.