

冗余度机器人的运动规划碰撞算法

李刚俊

(成都电子机械高等专科学校机电系 成都 610031)

【摘要】在考虑了操作机的关节极限、自碰撞和静态障碍物的情况下,从给定的初始位形出发,发现一条到末端效应器目标位置和姿态的相连可到达路径。方法给出了机器人操作机点到点逆运动学问题求解算法,利用碰撞算法实现了冗余度机器人运动规划,仿真实验验证了该方法的有效性,并表明了该方法具有较大的实用价值。

关键词 碰撞算法; 冗余度机器人; 运动规划; 逆运动学

中图分类号 TP241

文献标识码 A

Collision-Based Motion Planning for Redundant Robot

LI Gang-jun

(Mechatronic Department, Chengdu Electromechanical College Chengdu 610031)

Abstract Based on collision detection algorithm, a motion planning method for globally solving for the point-to-point inverse kinematics of robotic manipulators with high redundancy is developed. It can capture the connectivity of the connected components of the free configuration space from initial configuration to the objective configuration for the manipulators with constraints due to the joint limits, self-collision and obstacles in the environment. The efficiency and the practical value of the method is illustrated.

Key words collision algorithm; redundant robot; motion planning; inverse kinematics

冗余度机器人能够完成避障、优化操作、改善性能等,同时也导致数学运算的复杂性。对于机器人系统来说,完成或实现某种作业实际上是使机器人实现可到达轨迹的控制问题,机器人控制问题包含两个相关的子问题——运动规划和运动控制。运动轨迹是机器人系统工作的依据,它决定了系统的工作方式和效率,机器人系统要完成某种操作作业,就必须对其运动轨迹进行规划。本文将对冗余度机器人的运动规划进行讨论,利用碰撞算法来实现冗余度机器人点到点可到达路径的规划。

1 目标函数的建立

所谓点到点就是给定一个初始机器人位形 $\hat{\theta}_0$, 一个希望的末端效应器位形 \hat{x}_g , 确定可到达的机器人位形 $\hat{\theta}$ 和与此相连的可行性路径 $f(\hat{\theta}) = \hat{x}_g$ 。

本文中强调对于机器人的任何几何形态的点到点运动学逆解问题,其几何形态无论是旋转关节或移动关节,存在关节限制、自碰撞限制且在环境中存在静态障碍。在研究运动规划的文献中,通常使用几种有效的逼近方法来搜索拥有极其复杂的拓扑结构的高维空间^[1-3],利用有限的节点的集合来捕捉自由位形空间的连接。分析强调问题的运动规划方面,即如果可行性路径存在,则确定一系列路径通过的“区域”,从而发现某一条可行性路径,并可进行更精确的优化。

首先将运动学逆解问题公式化为相对于可到达机器人位形空间的一个优化问题^[4], 并利用初始和希望末端效应器框架之间的一种新型计量方法作为目标函数。这一优化问题利用类似于Ariadne's clew算法的结构形式来求解^[5]。方法可被描述为两个子算法的组合, 它们以交互的方式执行。Explore算法探索始于起点的可到达位形空间, 不断将路标放置在里面。Search算法为一个局部规划, 主要用来执行碰撞检测, 该规划是为了获得始于新路标的目标位形。

根据Denavit-Hartenberg标记法, 利用 F_i 表示连接第 i 个关节的构件, T_{ac} 是齐次转移矩阵。用 d 来标记对应于两个末端效应器位形的两个末端效应器框架之间的一种测量距离, 比如说 F_a 和 F_b (相对于共同的坐标框架 F_c), 分别用 T_{ac} 和 T_{bc} 来代替。一个框架的单位矢量的顶点确定了工作空间中由3个物理点(顶点)组成的一个集合, 每个顶点表示3个轴中的一个轴。距离 d 定义为两个坐标架对应矢量顶点之间的Euclidean距离的平方和的平方根, 即:

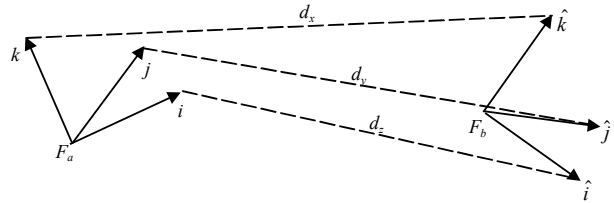


图1 d_x, d_y 和 d_z 的图例

$$d(T_{ac}, T_{bc}) = \sqrt{d_x^2 + d_y^2 + d_z^2}$$

这里 d_x, d_y 与 d_z 分别是单位矢量 x, y 和 z 的顶点之间的距离, 如图1所示。由于这种测量不依赖于基础坐标框架 F_c , 所以跟踪坐标可去掉。用符号 \cdot 代表 R^n 空间中的标准Euclidean模, 用 $dist$ 标记 R^n 中一点和一个集合间的距离, 则:

$$dist(\hat{\theta}, S) = \min_{\hat{\theta}' \in S} \|\hat{\theta} - \hat{\theta}'\| \tag{1}$$

用 A 标记机器人的手臂, $\hat{\theta} = (\theta_1, \theta_2, \dots, \theta_n)$ 代表机器人的一个位形, E 表示 A 的末端效应器, $\hat{x} = (x, y, z, \alpha, \beta, \gamma)$ 表示末端效应器的一个位形。 C_A 和 C_E 分别表示 A 和 E 的位形空间, $C_{Afree}(\hat{\theta})$ 表示包含点 $\hat{\theta}$ 的 C_A 的所有不碰撞可到达部分, 即路径相连。

可行性路径是能满足关节限制和碰撞限制的路径, 它完全位于 $C_{Afree}(\hat{\theta}_0)$ 内。路标是位形空间内的一点 $\hat{\theta}$, 并用 L_i 标记, 路标集记作 Q 。

首先将末端效应器位形 $f(\hat{\theta})$ 和目标末端效应器位形 \hat{x}_g 之间的距离定义为费用方程, 即把逆运动学问题当作优化问题, 从而建立目标函数:

$$CT(\hat{\theta}, \hat{x}_g) = d(T(f(\hat{\theta})), T(\hat{x}_g)) \tag{2}$$

将点到点的逆运动学问题作如下处理: 给定机器人初始位形 $\hat{\theta}_0$ 和希望的末端效应器位形 \hat{x}_g , 确定一个可到达的机器人位形 $\hat{\theta}_e$ 和一个可行性路径, 就是:

$$\hat{\theta}_e = \{ \hat{\theta} : \min CT(\hat{\theta}, \hat{x}_g), \hat{\theta} \in C_{Afree}(\hat{\theta}_0) \} \tag{3}$$

从中可以知道 $\hat{\theta}_e \in f^{-1}(\hat{x}_g) \Leftrightarrow CT(\hat{\theta}_e, \hat{x}_g) = 0$, 而且, 全局最小 $CT(\cdot)$ 不是唯一。为了计算优化问题的局部极小值, 首先利用一个局部算法来实现这一目标。

2 局部算法Search

由于 $\hat{\theta} = (\theta_1, \theta_2, \dots, \theta_n)$ 是机器人的一个位形, 可以将位形 $\hat{\theta}$ 中关节 i 的不碰撞区间(包含 θ_i)表示为: $[\theta_i - \Delta_i^{\min}, \theta_i + \Delta_i^{\max}] \subset R$, 其中 Δ_i^{\min} 和 Δ_i^{\max} 分别代表关节 i 在负方向和正方向上的最大增量。很显然可利用区间来表示关节限制, 通过简单的几何方法很容易地计算出碰撞限制并将其表示为区间, 这些区间的交集也就给出了决定关节 i 的可行性运动范围的理想区间 $[\Delta_i^{\min}, \Delta_i^{\max}]$ 。用 C_θ^i 表示关节 i 的可行性运动范围的区间集, 即:

$$C_\theta^i = \{ (\theta_1, \theta_2, \dots, \theta_{i-1}, \theta_i + \Delta, \theta_{i+1}, \dots, \theta_n) \mid \Delta \in [\Delta_i^{\min}, \Delta_i^{\max}] \}, C_\theta^i \in C_{Afree} \tag{4}$$

对于任意 $\hat{\theta} \in C_{Afree}$, 定义函数 g 如下:

$$g(\hat{\theta}, i) = \hat{\theta}^{\min} : \min CT(\hat{\theta}', \hat{x}_g) \tag{5}$$

其中 $\hat{\theta}' \in C_\theta^i$, 这是 C_θ^i 中的一个一维优化问题, 且是 C_θ^i 中唯一的一个全局极小值。应用该结果时, 除

关节*i*外的其他关节的值是不变的,也就是说如果 $\hat{\theta} = (\theta_1, \theta_2, \dots, \theta_i, \dots, \theta_{n-1}, \theta_n)$, 则 $g(\hat{\theta}, i) = (\theta_1, \theta_2, \dots, \theta_i^{\min}, \dots, \theta_{n-1}, \theta_n)$ 。从 $\hat{\theta}_0$ 到 $\hat{\theta}_n$ 的过程可以体现为:

$$g^n(\hat{\theta}_0) = g(g(g(\dots(g(\hat{\theta}_0, 1), 2) \dots, n-2), n-1), n) \quad (6)$$

就是Search算法:

```

search( $\hat{\theta}_0, \hat{x}_g$ )
 $\hat{\theta} = g^n(\hat{\theta}_0)$ ;
for ( $i=1; i < n; i++$ )
{ if ( $\hat{\theta} == \hat{\theta}_0$  or  $f(\hat{\theta}) == \hat{x}_g$ )
  { return( $\hat{\theta}$ );
    break;
  }
}
else return ( search( $\hat{\theta}, \hat{x}_g$ ) );

```

从 $\hat{\theta} = \hat{\theta}_0$ 开始, Search重复应用 $g^n(\hat{\theta})$, 每一次都利用前面的迭代结果作为当前迭代的起始点。如果条件 $CT(\hat{\theta}_{loc}, \hat{x}_g) = 0$ 成立的话, 则 $\hat{\theta}_{loc} = \text{search}(\hat{\theta}_0, \hat{x}_g)$ 是一个全局极小(即逆运动学问题的解)。否则, 发现的为一个局部极小值。在实际应用时, 可以对迭代次数加以限制或给出一个方法来限制迭代次数。而始于初始位形 $\hat{\theta}_0 \in C_{\text{Afree}}$ 的可到达位形空间的连接, 可由Explore算法来实现。

3 算法Explore

通过在所到达位形空间内产生自由路径和放置路标可获得上述空间的一个表示 $C_{\text{Afree}}(\hat{\theta}_0)$, 它始于初始路标位形 $L_1 = \hat{\theta}_0$ 。由Explore产生的路标节点的集合代表了 $C_{\text{Afree}}(\hat{\theta}_0)$ 中位形的集合, 连线的集合代表了连接这些位形的自由路径。

Explore要求全局优化, 用 Q_m 表示位于第*m*步、存在的路标的集合。首先初始化路标集为 $Q_1 = \{L_1\}$, 这里 $L_1 = \hat{\theta}_0$ 。把从 L_1 所能到达的最远的、位于 $C_{\text{Afree}}(\hat{\theta}_0)$ 中的那个位形取为 $\hat{\theta}_1$, 即 $\hat{\theta}_1: \max \|\hat{\theta} - L_1\|$, 其中 $\hat{\theta} \in C_{\text{Afree}}(\hat{\theta}_0)$, $\hat{\theta}_1$ 将作为第二个路标, 即 $L_2 = \hat{\theta}_1$ 和 $Q_2 = Q_1 \cup \{L_2\}$ 。下一次迭代放置第三个路标, 它是根据 $C_{\text{Afree}}(\hat{\theta}_0)$ 上从 Q_2 所能到达的最远距离而得到, 也就是说它到集合中的一个路标(共两个)的距离最大。对于第*m*次迭代, 可得到第(*m*+1)个路标, 它极大化到当前路标集的距离:

$$L_{m+1} = \{ \max \text{dist}(\hat{\theta}, Q_m) : \hat{\theta} \in C_{\text{Afree}}(\hat{\theta}_0) \} \quad (7)$$

可以采用一种基于准随机方法的全局优化来简化上述复杂的过程。它执行随机路径, 每一条路径都来至当前路标集, 并且完全位于 $C_{\text{Afree}}(\hat{\theta}_0)$ 内, 这些自由路径的端点被称为测试点, 那个至当前路标集最远的测试点被选为新的路标, 然后重复这一过程。对于第*m*次迭代, 用 $Q_m = \{L_1, L_2, \dots, L_m\}$ 表示已存在的路标集; 用 $T_m = \{r_1, r_2, \dots, r_m\}$ 表示从一个路标到另一个路标的路径的集合; 用 $B_m = \{h_1, h_2, \dots, h_m\}$ 代表存在的测试点的集合; 且用 $E_m = \{e_2, e_3, \dots, e_m\}$ 代表路径集。在第*m*步, 若 $h_k \in B_m$ 是距 Q_m 最远的测试点, 也就是有: $\rho_m = \text{dist}(h_k, Q_m) > \text{dist}(h_j, Q_m) \forall h_j \in B_m$, h_k 就成为第(*m*+1)个路标 $L_{m+1} = h_k$ 。随后又随机产生两个的新测试点, 进行第(*m*+1)次迭代。上述过程就是Explore算法。

```

explore ( $m$ )
 $\rho_m = 0$ ;
for ( $i = 1; i \leq m; i++$ )
{ min_dist = ;
  for ( $j = 1; j \leq m; j++$ )
  { distance =  $h_i - L_j$  ;

```

```

if ( distance < min_dist ) min_dist = distance ;
}
if ( min_dist > ρm )
{
    k = i ;
    ρm = min_dist ;
}
}
Lm+1 = hk , rm+1 = ek , em+1 = random-path-from ( Lm+1 ) ;
ek = random-path-from ( Lk ) , hm+1 = em+1 ( 1 ) , hk = ek ( 1 ) ;
return ( ρm ) ;

```

这里random-path-from ($\hat{\theta}$)产生一条开始于位形 $\hat{\theta}$ 的随机路径。反弹技术被用来生成随机路径，即每一条随机路径都来至于杆件从障碍物的反弹。Explore(m)采用已有的路标及路径作为其输入，而返回更新后的路标及路径。

综合利用以上两个算法，可以获得逆运动学规划算法程序Invs。

```

invs (  $\hat{\theta}_0, \hat{x}_g, n$  ) // n是迭代次数限制
m = 1 , L1 =  $\hat{\theta}_0$  , e1 = random-path-from ( L1 ) , h1 = e(1) ;
path_found = false ;
for ( m = 1 ; m <= n ; m ++ )
{
     $\hat{\theta}_{loc}$  = search ( Lm ,  $\hat{x}_g$  ) ;
    if ( CT( $\hat{\theta}_{loc}$  ,  $\hat{x}_g$ ) == 0 ) path_found = true ;
    else explore ( m ) ;
}
if ( path_found ) return (  $\hat{\theta}_{loc}$  ) ;
else return ( fail ) ;

```

通过连接路径的连接，可以很容易求得 $\hat{\theta}_{loc}$ ，并通过Search求得从 L_m 到 $\hat{\theta}_{loc}$ 的路径，从而得到从 $\hat{\theta}_0$ 到逆运动学解的路径解。

4 仿真结果

为了检验上面的算法，将其应用于一个7-DOF操作臂，所有关节都是旋转关节，每个关节都有一个关节限制。碰撞检测方法采用：对于指定的机器人连杆，当其旋转时，保持其它所有自由度固定不变，这样可以很有效地计算出它的非碰撞区间。为了保证每种检测情况确实存在解，随机生成几个非碰撞的机器人位形，记录它们所对应的末端效应器位形，将其作为测试的目标位形，并获得各种测试统计，如：成功率，平均路标数和平均运行时间。

将最大路标数限制为486，测试分为有和没有障碍两种情况，针对5种不同的障碍情况，选取100多个随机最终位形，利用相同的初始位形，路径规划算法总能在较短时间内发现一个解，如表1所示。

表1 运行时间统计

问题类型	路标数量		CPU时间/min	
	平均值	最大~最小	平均值	最大~最小
没有障碍	27	1~132	16.9	1~60
有障碍	41	1~189	35.7	1~122

5 结 论

本文的算法可以很容易地实现冗余度机器人的运动规划，在不同的环境情况下，路径规划算法都能成功运行。如果在使用过程中利用更有效的加速方法和碰撞检测技术，运行时间还可以大幅减少，甚至可以实现逆运动的实时计算，因而该算法具有较大的实用价值。

(下转第864页)

所有指令,首先,因为在技术上实现的成本将较大,其次,过于透明的市场也会降低交易效率或流动性^[4]。但如果交易所指定唯一交易商负责某股票的集合竞价,并允许该交易商享有“市场权力”(主要体现在具有信息优势,可以观测指令流),则应该注意适当提高市场透明性,以确保价格的信息效率。这种市场设计一般应用在做市商市场,对我国目前的指令驱动的限价指令簿市场并不适用,但如果以后出现采用做市商制度的创业板市场,则该设计就值得考虑。

4 结束语

本文分析了集合竞价的“黑箱”交易,证明了进行指示性价格披露、指定交易商定价或增加指令簿透明性可增加价格中的信息含量,提高股市的价格效率。因此,提供集合竞价过程中的即时信息的开放式集合竞价,更加有利于提高市场效率,降低操纵的可能性。目前我国新开设的“中小企业板块”就是采取开放式集合竞价过程决定每个交易日的开盘价,与本文的观点一致。此外,由于集合竞价时间是固定的,因此有可能被某些投资者利用,通过在集合竞价结束前下大额指令,操纵开盘价格。如果在规定的集合竞价开盘时间内,随机结束集合交易,而非固定时间结束,就可以防止操纵股价现象的发生,提高集合交易的有效性和价格信息含量,降低集合开盘交易的波动性。

参 考 文 献

- [1] Madhavan A. Trading mechanisms in securities markets[J]. *Journal of Finance*, 1992, 2(2): 607-641
- [2] Comerton-Forde C. Do trading rules impact on market efficiency? A comparison of opening procedures on Australian and Jakarta stock exchanges[J]. *Pacific-Basin Finance Journal*, 1999, (7): 495-521
- [3] Stoll H. The stock exchange specialist system: an economic analysis[M]. New York: Monograph Series in Finance and Economics, 1985, Vol. 2, Salomon Brothers Center, New York University, 1985
- [4] Madhavan A. Security prices and market transparency[J]. *Journal of Financial Intermediation*, 1996,(5): 255-283

编 辑 熊思亮

(上接第831页)

参 考 文 献

- [1] Chen P C, Hwang Y K. Sandros: A motion planner with performance proportional to task difficulty[C]. In *proc IEEE int Conf Robot Automat*, Nice, France, 1997. 2 346-2 353
- [2] Gupta K K, Zhu X. Practical global motion planning for many degree freedom: A novel approach within sequential framework[J]. *J Robot Syst*, 2001, 12 (2): 105-118
- [3] Kavraki L, Svetska P, Latombe J C, et al. Probabilistic roadmaps for path planning in high dimensional configuration space[J]. *IEEE Trans Robot Automat*, 1996, 12 (8): 566-580
- [4] Valero F, Mata J. A formulation for path planning of manipulators in complex environments by using adjacent configuration[J]. *Adv Robot*, 1997, 11(1): 33-56
- [5] Bessiere P, Ahuactzin J M. The ariadens clew algorithm: global planning with local methods[C]. In *Proc IEEE/RSJ Conf Intel Robots Syst*, Yokohama, Japan, 1993. 1 373-1 380

编 辑 孙晓丹