

判断两个凸多面体是否相交的一个快速算法*

任世军^{1,2} 洪炳熔¹ 孟庆鑫²

¹(哈尔滨工业大学计算机科学与工程系 哈尔滨 150001)

²(哈尔滨工程大学机电工程学院 哈尔滨 150001)

E-mail: rersj@hope.hit.edu.cn

摘要 在机器人路径规划中,碰撞检测算法占有十分重要的地位.在智能机器人仿真系统中,碰撞检测耗用的时间在整个路径规划过程所用时间中占有相当大的比例.于是,如何进一步提高碰撞检测的速度在智能机器人路径规划系统中就起到了非常关键的作用.而碰撞检测问题最终转化为判断三维空间中两个凸多面体是否相交的问题.就这一问题,给出了一种新的算法,其思想是取一个从一个凸多面体指向另一个多面体的向量,根据两个多面体中的面与这一向量的相对位置关系来寻找相交的平面.即有两个多面体的交点位于这一平面,若能找到一个相交平面则可以断定两个多面体相交.

关键词 路径规划,碰撞检测,机器人,线性不等式.

中图分类号 TP242

在机器人路径规划算法中,碰撞检测占有十分重要的地位,而在机器人的工作空间中往往用凸多面体来模拟机器人臂、障碍物等等.在机器人路径规划中,往往要检测机器人臂与障碍物以及机械臂与机械臂之间是否相撞,这就需要检测两个凸多面体形成的集合之间是否有交点.若有交点,说明机器人臂与障碍物相撞或者机械臂与机械臂自身相撞,否则说明机器人臂与障碍物并不相撞.以往的碰撞检测算法都是从空间几何学的角度^[1]或从最优化的角度^[2,3]出发,这样就需要大量的求交运算.在本文中,我们给出一种算法来检测空间中的两个凸多面体的碰撞问题.在机器人的仿真系统中,物体(包括机械臂和障碍物)的建模往往采用计算机图形学中的B-rep方式,即边界表示方式.一个物体由3个要素组成,即点、棱和面.在计算机中存储这样一个物体一般是存储其拓扑结构(指点、棱和面之间的关系),其各个顶点及其各个面的方程.在机械臂和障碍物的运动过程中都要实时地计算其每一个顶点及其各个面的方程,以确定该物体的位置.

设有 n 维空间中由线性不等式围成的两个凸多面体 A 和 B , 它们分别由式(1)和式(2)给出,

$$\begin{cases} p_{11}x_1 + p_{12}x_2 + \dots + p_{1n}x_n \leq q_1 \\ p_{21}x_1 + p_{22}x_2 + \dots + p_{2n}x_n \leq q_2 \\ \vdots \\ p_{m1}x_1 + p_{m2}x_2 + \dots + p_{mn}x_n \leq q_m \\ r_{11}x_1 + r_{12}x_2 + \dots + r_{1n}x_n \leq s_1 \\ r_{21}x_1 + r_{22}x_2 + \dots + r_{2n}x_n \leq s_2 \\ \vdots \\ r_{l1}x_1 + r_{l2}x_2 + \dots + r_{ln}x_n \leq s_l \end{cases} \quad (1)$$

$$\begin{cases} r_{11}x_1 + r_{12}x_2 + \dots + r_{1n}x_n \leq s_1 \\ r_{21}x_1 + r_{22}x_2 + \dots + r_{2n}x_n \leq s_2 \\ \vdots \\ r_{l1}x_1 + r_{l2}x_2 + \dots + r_{ln}x_n \leq s_l \end{cases} \quad (2)$$

设 $p_i = (p_{i1}, p_{i2}, \dots, p_{in})^T$, $r_i = (r_{i1}, r_{i2}, \dots, r_{in})^T$, $x = (x_1, x_2, \dots, x_n)^T$, 则式(1)可表示为

* 本文研究得到哈尔滨工业大学校管基金资助. 作者任世军, 1962年生, 副教授, 主要研究领域为人工智能, 机器人路径规划, 图论. 洪炳熔, 1937年生, 教授, 博士生导师, 主要研究领域为人工智能, 空间机器人, 虚拟现实. 孟庆鑫, 1939年生, 教授, 博士生导师, 主要研究领域为海洋智能机械, 水下机器人作业系统.

本文通讯联系人: 任世军, 哈尔滨 150001, 哈尔滨工业大学计算机科学与工程系

本文 1998-10-22 收到原稿, 1999-04-12 收到修改稿

$$\begin{cases} p_1^T \cdot x \leq q_1 \\ p_2^T \cdot x \leq q_2 \\ \vdots \\ p_m^T \cdot x \leq q_m \end{cases}$$

式(2)可表示为

$$\begin{cases} r_1^T \cdot x \leq s_1 \\ r_2^T \cdot x \leq s_2 \\ \vdots \\ r_l^T \cdot x \leq s_l \end{cases}$$

1 算法的理论基础

定理 1. 设 $g = (g_1, g_2, \dots, g_n)^T$ 为集合 A 的边界点, $t \in A$, 并且对于 $\forall \alpha (0 < \alpha \leq 1)$ 都有 $g + \alpha(t - g) \in A$, 那么一定存在 i_0 使得 $1 \leq i_0 \leq m, p_{i_0}^T \cdot (g - t) < 0$ 并且 $p_{i_0}^T \cdot g = q_{i_0}$.

证明: 由于 g 为集合 A 的边界点, 且 A 为凸多面体, 故有 $g \in A$.

首先, 我们证明一定存在 i_0 使得 $1 \leq i_0 \leq m$, 并且 $p_{i_0}^T \cdot g = q_{i_0}$. 因为若不是这样, 即对 $\forall i (1 \leq i \leq m)$, 都有 $p_i^T \cdot g < q_i$. 设 $\delta = \min \left\{ \frac{q_i - p_i^T \cdot g}{p_i^T \cdot e} \mid p_i^T \cdot e > 0, e = (e_1, e_2, \dots, e_n)^T \text{ 为单位向量}, 1 \leq i \leq m \right\}$, 由于对 $\forall i (1 \leq i \leq m)$ 和任意的单位向量 e , 我们有

$$\frac{q_i - p_i^T \cdot g}{p_i^T \cdot e} \geq \frac{\min_{1 \leq i \leq m} (q_i - p_i^T \cdot g)}{\|p_i\| \|e\|} \geq \frac{\min_{1 \leq i \leq m} (q_i - p_i^T \cdot g)}{\max_{1 \leq i \leq m} \|p_i\|} > 0,$$

故 $\delta > 0$. 对于任意的向量 x , 只要 $\|x - g\| < \delta$, 由于 $p_i^T \cdot x = p_i^T \cdot g + p_i^T \cdot (x - g)$, 并且 $p_i^T \cdot (x - g) \leq \|p_i\| \cdot \|x - g\| < \|p_i\| \cdot \delta \leq \|p_i\| \cdot \frac{q_i - p_i^T \cdot g}{p_i^T \cdot e}$, 我们取 $e = \frac{p_i}{\|p_i\|}$, 则有 $p_i^T \cdot (x - g) < \|p_i\| \cdot \frac{q_i - p_i^T \cdot g}{\|p_i\|} =$

$\|p_i\|^2 \cdot \frac{q_i - p_i^T \cdot g}{p_i^T \cdot p_i} = q_i - p_i^T \cdot g$, 从而 $p_i^T \cdot x < p_i^T \cdot g + (q_i - p_i^T \cdot g) = q_i$, 其中 $1 \leq i \leq m$, 即 $x \in A$. 从而 g 为 A 的内点, 矛盾.

其次, 我们证明对于所有满足 $p_i^T \cdot g = q_i$ 的 i 中一定存在一个 i_0 使得 $1 \leq i_0 \leq m$, 并且 $p_{i_0}^T \cdot (g - t) < 0$. 设 $I_1 = \{i \mid p_i^T \cdot g = q_i, 1 \leq i \leq m\}, I_2 = \{i \mid i \notin I_1, 1 \leq i \leq m\}$. 假设对于 $\forall i \in I_1$ 都有 $p_i^T \cdot (g - t) \geq 0$. 若存在 $i \in I_2$ 使得 $p_i^T \cdot (g - t) < 0$, 则取 α_1 满足 $\alpha_1 = \min \left\{ \frac{q_i - p_i^T \cdot g}{p_i^T \cdot (t - g)} \mid p_i^T \cdot (t - g) > 0, i \in I_2 \right\}$; 否则取 $\alpha_1 = 1$. 对于 $\forall i \in I_2$, 如果 $p_i^T \cdot (t - g) > 0$, 那么 $p_i^T \cdot [g + \alpha_1(t - g)] = p_i^T \cdot g + \alpha_1 \cdot p_i^T \cdot (t - g) \leq p_i^T \cdot g + \frac{q_i - p_i^T \cdot g}{p_i^T \cdot (t - g)} \cdot p_i^T \cdot (t - g) = q_i$. 如果 $p_i^T \cdot (t - g) \leq 0$, 那么 $p_i^T \cdot [g + \alpha_1(t - g)] = p_i^T \cdot g + \alpha_1 \cdot p_i^T \cdot (t - g) \leq q_i$. 由假设知, 对 $\forall i \in I_1$, 有 $p_i^T \cdot [g - \alpha_1(t - g)] = p_i^T \cdot g + \alpha_1 \cdot p_i^T \cdot (t - g) \leq p_i^T \cdot g = q_i$. 所以 $g + \alpha_1(t - g) \in A$, 与定理条件相矛盾. \square

定理 2. 如果存在 $s \in A, t \in B$, 使得 $s \in B, t \in A$, 那么若 $A \cap B \neq \emptyset$, 则一定存在 i_0 , 使得 $1 \leq i_0 \leq m, p_{i_0}^T \cdot (s - t) < 0$, 并且 $A \cap B \cap \{x \mid p_{i_0}^T \cdot x = q_{i_0}\} \neq \emptyset$. 也一定存在 i_1 , 使得 $1 \leq i_1 \leq n, r_{i_1}^T \cdot (s - t) > 0$, 并且 $A \cap B \cap \{x \mid r_{i_1}^T \cdot x = s_{i_1}\} \neq \emptyset$.

证明: 设 $A \cap B \neq \emptyset$. 我们用反证法. 如果对于 $\forall i (1 \leq i \leq m)$, 只要 $p_i^T \cdot (s - t) < 0$ 就有 $A \cap B \cap \{x \mid p_i^T \cdot x = q_i\} = \emptyset$, 亦即对于集合 $A \cap B$, 约束 $p_i^T \cdot x \leq q_i$ 不起作用, 那么若设 $A_1 = \{x \mid p_i^T \cdot x \leq q_i, p_i^T \cdot (s - t) \geq 0, 1 \leq i \leq m\}$, 则有 $A \cap B = A_1 \cap B$.

- (1) 若存在 α_0 , 使得 $0 < \alpha_0 < 1, s + \alpha_0(t - s) \in A \cap B$.
- (i) 由于 $t \in B$ 并且 B 是凸集, 所以对于 $\forall \alpha (\alpha_c \leq \alpha \leq 1)$ 都有 $s + \alpha(t - s) \in B$.
- (ii) 对于 $\forall \alpha (\alpha_0 \leq \alpha \leq 1) \forall i (1 \leq i \leq m)$. 若 $p_i^T \cdot (s - t) \geq 0$ 则由于 $s \in A$, 故有

$$p_i^T \cdot [s + \alpha(t-s)] = p_i^T \cdot s + \alpha p_i^T \cdot (t-s) \leq p_i^T \cdot s \leq q_i,$$

于是对于 $\forall \alpha (0 \leq \alpha \leq 1)$ 都有 $s + \alpha(t-s) \in A_1$.

由 (i) 和 (ii) 知, 对于 $\forall \alpha (0 \leq \alpha \leq 1)$ 都有 $s + \alpha(t-s) \in A_1 \cap B = A \cap B \subset A$. 当 $\alpha=1$ 时, $s + \alpha(t-s) = s + (t-s) = t$, 所以 $t \in A$. 矛盾.

(2) 假设对于 $\forall \alpha (0 \leq \alpha \leq 1)$ 都有 $s - \alpha(t-s) \in A \cap B$. 即从 s 至 t 的线段中没有 $A \cap B$ 的点. 由于 $A \cap B$ 是凸集(两个凸集的交集仍为凸集)且非空, 所以, 可取一点 $g \in A \cap B$ 使得 g 到连接 s 至 t 的直线的距离最短. 设 $\alpha > 0$, 在空间中取一点 $g + \alpha(t-s)$. 因为连接点 s 至 t 的直线与连接点 g 与 $g + \alpha(t-s)$ 的直线相互平行, 所以, s, t, g 和 $g + \alpha(t-s)$ 位于同一个平面并且连接 $s, t, g + \alpha(t-s), g$ 的线段构成一个平行四边形. 我们说 $g + \alpha(t-s) \in A$, 否则, 由于 $s \in A$, 故连接 s 和 $g + \alpha(t-s)$ 的线段也属于 A . 同理, 连接 t 与 g 的线段属于 B . 于是上述梯形的两条对角线的交点既属于 A 又属于 B , 从而属于 $A \cap B$. 但这个交点到连接 s 至 t 的直线的距离比 g 到连接 s 至 t 的直线的距离更短, 矛盾. 于是, 对于 $\forall \alpha > 0$ 都有 $g + \alpha(t-s) \in A$. 故 g 为集合 A 的边界点. 根据定理 1 可知, 一定存在 i_0 使得 $1 \leq i_0 \leq m, p_{i_0}^T \cdot (g-t) < 0$ 并且 $p_{i_0}^T \cdot g = q_{i_0}$. 于是, 因为 $s \in A$, 所以 $p_{i_0}^T \cdot (s-t) = p_{i_0}^T \cdot [(s-g) + (g-t)] = p_{i_0}^T \cdot (s-g) + p_{i_0}^T \cdot (g-t) = (p_{i_0}^T \cdot s - q_{i_0}) + p_{i_0}^T \cdot (g-t) < 0$. 定理证毕. \square

2 判断空间中两个凸多面体是否相交的算法

在我们提出的算法中首先要用到文献[4]中提出的向量定位算法. 用以判断空间中由一组线性不等式围成的凸空间是否是空集. 若不是空集则返回其中的一点.

设 G 为一个动态点(即可以是一维、二维以及三维空间中的任意一点), A 为一组线性不等式, 用 $\text{xianliang}(A)$ 表示判断由 A 确定的集合是否是空集的向量定位算法, 则算法可以表示如下.

$\text{xianliang}(A)$

{

Step 1. 若 A 中含有矛盾常数不等式, 则返回 NULL.

Step 2. 将 A 中的不等式按照第 1 个变量的系数大于、等于和小于 0 的原则分成 3 个集合 A_1, A_2 和 A_3 .

Step 3. 若 $A_1 = \emptyset$ 并且 $A_3 = \emptyset$, 则去掉 A 中的第 1 个变量. 再次调用 $\text{xianliang}(A)$. 设 $G = \text{xianliang}(A)$. 将 G 与相应的第 1 个变量的任一实数组合在一起, 令其为 G . 返回 G .

Step 4. 若 $A_1 = \emptyset$ 或者 $A_3 = \emptyset$, 则将 A_2 中的第 1 个变量去掉, 设 $G = \text{xianliang}(A_2)$ (若 $A_2 = \emptyset$ 则取 G 为各个分量全是 0 的零向量). 若 $A_1 = \emptyset$, 则让第 1 个变量充分大. 若 $A_3 = \emptyset$, 则让第 1 个变量充分小. 这样即可求得满足所有不等式的点. 将得到的第 1 个变量与 G 一起构成新的 G , 并返回 G .

Step 5. 若 A_1 中的不等式的个数小于或者等于 A_3 中的不等式的个数, 则依次取 A_1 中的不等式, 令其为等式(设为 d), 化简其他不等式, 得到新的不等式组 \hat{A} , 使得 \hat{A} 中的变量个数少于 A 中的变量的个数. 设 $G = \text{xianliang}(\hat{A})$. 若 $G = \text{NULL}$, 则取下一个不等式. 继续执行. 否则, 用 G 和 d 可求出另外一个变量与 G 一起组成新的 G . 返回 G . 若所有不等式都取完, 则返回 NULL.

Step 6. 若 A_1 中的不等式的个数大于 A_3 中的不等式的个数, 则依次取 A_3 中的不等式, 令其为等式(设为 d), 化简其他不等式, 得到新的不等式组 \hat{A} , 使得 \hat{A} 中的变量个数少于 A 中的变量的个数. 设 $G = \text{xianliang}(\hat{A})$. 若 $G = \text{NULL}$, 则取下一个不等式. 继续执行. 否则, 用 G 和 d 可求出另外一个变量与 G 一起组成新的 G . 返回 G . 若所有不等式都取完, 则返回 NULL.

}

设 A 和 B 分别表示约束空间中的两个凸多面体的线性不等式组, 用 $\text{qiujiao}(A, B)$ 表示确定 A, B 的交集是否是空集的求交算法, 则算法可以表示如下.

$\text{BOOL qiujiao}(A, B)$

{

Step 1. 若约束 A 和约束 B 的不等式均为常数不等式且不含矛盾不等式, 则 return TRUE.

Step 2. 若约束 A 和约束 B 的不等式含有矛盾不等式, 则 return FALSE.

Step 3. $s = \text{xiangliang}(A), t = \text{xiangliang}(B)$.

Step 4. 若 $s = \text{NULL}$ 或 $t = \text{NULL}$, 则 return FALSE.

Step 5. 若 $s \in B$ 或 $t \in A$, 则 return TRUE.

Step 6. 如果约束 A 的不等式的个数少于约束 B 的不等式的个数, 那么依次选择约束 A 的不等式, 使得该不等式的法向量与向量 $(t-s)$ 的数量积大于 0. 用该不等式化简约束 A 和 B 的不等式得到不等式组 A_1 和 B_1 , 使得 A_1 和 B_1 中的变量个数少于 A 和 B 中变量的个数. 重复调用这一过程, 若 $\text{qiujiào}(A_1, B_1) = \text{TRUE}$, 则 return TRUE. 否则, 继续执行.

Step 7. 如果约束 B 的不等式的个数少于约束 A 的不等式的个数, 那么依次选择约束 B 的不等式, 使得该不等式的法向量与向量 $(t-s)$ 的数量积小于 0. 用该不等式化简约束 A 和 B 的不等式得到不等式组 A_1 和 B_1 , 使得 A_1 和 B_1 中的变量个数少于 A 和 B 中变量的个数. 重复调用这一过程, 若 $\text{qiujiào}(A_1, B_1) = \text{TRUE}$, 则 return TRUE. 否则, 继续执行.

Step 8. return FALSE.

};

在一个实际的机器人仿真系统中可以首先判断两个物体的包围盒(bounding box)是否相交, 若不相交, 则这两个凸多面体也不相交; 否则, 进一步用上面提出的算法来判定这两个凸多面体是否相交. 这样做可以提高碰撞检测的效率.

3 实例

例 1: 设在三维空间中给定两个凸多面体 A 和 B , 如图 1 所示.

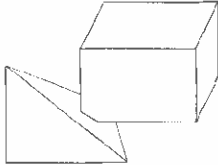


Fig. 1 Two intersected polyhedra
图1 两个相交的凸多面体



Fig. 2 Two polyhedra intersected on only edges not vertex
图2 两个只有棱而非点相交的凸多面体

这两个凸多面体的平面方程分别由方程组(3)和(4)给出.

$$\begin{cases} x \leq 0 \\ -y \leq 0 \\ -z \leq 0 \\ -x + y + z \leq 2 \end{cases} \quad (3)$$

$$\begin{cases} x \leq -0.5 \\ -y \leq -0.5 \\ -z \leq -0.5 \\ -x \leq 3 \\ y \leq 3 \\ z \leq 3 \end{cases} \quad (4)$$

由向量定位算法可以求出式(3)和式(4)中的两个点 $(0, 0, 0)$ 和 $(-0.5, 0.5, 0.5)$. 由于点 $(-0.5, 0.5, 0.5)$ 满足式(3), 所以这两个凸多面体相交.

例 2: 设在三维空间中给定两个凸多面体 A 和 B , 如图 2 所示.

这两个凸多面体的平面方程分别由方程组(5)和(6)给出,

$$\begin{cases} x \leq 0 \\ -y \leq 0 \\ -z + 2 \leq 0 \\ -x + y + z \leq 4 \end{cases}, \tag{5}$$

$$\begin{cases} x \leq -0.5 \\ -y \leq -0.5 \\ -z \leq -0.5 \\ -x \leq 3 \\ y \leq 3 \\ z \leq 3 \end{cases}. \tag{6}$$

由向量定位算法可以求出式(5)和式(6)中的两个点(0,0,-2)和(-0.5,0.5,0.5).由于式(5)中的不等式的个数少于式(6)中的不等式的个数,所以取式(5)中的不等式.由于此时t-s=(-0.5,0.5,2.5),只有第4个不等式相应的平面方程的法向量与t-s的数量积大于0,令其为等式,则式(5)和式(6)化简为

$$\begin{cases} x \leq 0 \\ -y \leq 0 \\ -x + y \leq 2 \end{cases}, \tag{7}$$

$$\begin{cases} x \leq -0.5 \\ -y \leq -0.5 \\ -x + y \leq 3.5 \\ -x \leq 3 \\ y \leq 3 \\ x - y \leq -1 \end{cases}. \tag{8}$$

用向量定位算法分别求出式(7)和式(8)中的两个点(0,0)和(-0.5,0.5).由于(-0.5,0.5)满足式(7),所以式(7)与式(8)的交不空,从而式(5)与式(6)的交也不空.

4 实验结果

我们在 IBM 486 上实现了我们提出的算法,通过大量的数据对算法进行了测试,结果见表 1.

Table 1 Experimental results of the algorithm (Experimented on IBM 486)

表 1 算法的实验结果(在 IBM 486 上执行)

| No. of A's constraints ^③ | Problem scale ^① | | No. of instance ^④ | Execution time ^② (s) | | |
|-------------------------------------|----------------------------|--|------------------------------|---------------------------------|----------------------|----------------------|
| | No. of B's constraints | | | maximum ^⑤ | minimum ^⑥ | average ^⑦ |
| 60 | 60 | | 30 | 0.03 | 0.01 | 0.01 |
| 80 | 60 | | 30 | 0.04 | 0.01 | 0.01 |
| 80 | 80 | | 20 | 0.06 | 0.02 | 0.03 |
| 100 | 100 | | 20 | 0.14 | 0.07 | 0.09 |
| 250 | 250 | | 20 | 1.93 | 1.2 | 1.52 |

①问题规模,②执行时间(秒),③约束数目,④算例数,⑤最多执行时间,⑥最少执行时间,⑦平均执行时间.

从表 1 的实验结果可以看出这个算法非常有效.在一个实际的机器人规划系统中执行这一算法之前,首先判断两个凸多面体的包围盒是否相交,然后再用这一算法进行判断.这样就能够进一步提高规划系统中碰撞检测的效率.

参考文献

1 Liu Lian-feng, Wang Yong-xi. A three dimensional algorithm on detecting collision between robot and its environment. Robot, 1996,18(1):50~54

- 〈刘连峰, 王泳喜. 空间路径规划中一种三维物体干涉检测算法. 机器人, 1996, 18(1): 50~54〉
- 2 Zeghloul S, Rambeaud P. A fast algorithm for distance calculation between convex objects using the optimization approach. *Robotica*, 1996, 14: 355~363
 - 3 Bobrow J E. A direct minimization approach for obtaining the distance convex polyhedra. *International Journal of Robotics Research*, 1989, 8(3): 65~75
 - 4 Ren Shi-jun, Hong Bing-rong. A fast algorithm to determine whether convex regions bounded by multiple linear constraints are empty. *Chinese Journal of Computers*, 1998, 21(10): 896~901
(任世军, 洪炳榕. 判定由线性不等式围成的凸空间是否为空的一个快速算法. 计算机学报, 1998, 21(10): 896~901)

A Fast Algorithm to Determine Whether the Intersection of Two Convex Regions Is Empty

REN Shi-jun^{1,2} HONG Bing-rong¹ MENG Qing-xin²

¹(Department of Computer Science and Engineering Harbin Institute of Technology Harbin 150001)

²(Department of Mechanical Engineering Harbin Engineering University Harbin 150001)

Abstract Collision detection algorithms play a very important role in the field of robot path planning. In a simulation system of intelligent robot, collision detection takes up a large portion of the time for the robot to plan a complete path from the initial position to the final position. So how to reduce the time the robot uses to detect collision becomes a key problem. But collision detection finally will transform to a problem to determine whether the intersection of two convex regions formed by linear inequalities is empty or not. The authors present a new algorithm in this paper. Firstly, a vector pointing from one polyhedron to the other is picked. Then the authors start to find an intersection plane of one polyhedron based on the scalar product of the norm vector of the plane and the picked vector. If such a plane is found, the intersection of the two convex polyhedra is not empty.

Key words Path planning, collision detection, robot, linear inequality.