# 非完整机器人无死锁碰撞规避

1st RuochenZheng建模、优化与仿真部
旷视自动化与机器人中国北京
zhengruochen02@megvii.com

2nd SiyuLi建模、优化与仿真部
*Megvii* 自动化与机器人技术北京，中国
lisiyu02@megvii.com

arXiv:2305.04511v1 [cs.RO] 8 May 2023

摘要—**我们提出了一种用于非完整机器人多机器人系统中无死锁且无碰撞导航的方法。该问题通过二次规划求解，适用于大多数具有线性运动学约束的轮式移动机器人。我们引入屏蔽速度和屏蔽协作碰撞规避（MCCA）算法，以促进完全分布式的死锁避免行为。为验证该方法，我们提供了详细的实现，并针对差速驱动机器人引入了航向振荡规避。据我们所知，这是首个即使在机器人数量众多且存在狭窄通道的情况下，也能为死锁避免提供非常有前景且稳定结果的方法。**

## I. 引言

多机器人系统中的碰撞与死锁避免是此类系统投入实际应用前必须解决的一个基本问题。虽然集中式系统能够同时确定所有机器人的导航细节，但维数灾难会导致系统即使面对数量适中的机器人，也无法在可接受的时间内给出令人满意的解决方案。因此，我们认为，更有前景的方向是寻求一种分布式导航策略，使每个机器人能够充分利用其可获取的信息，为自身利益做出决策 [1]。

许多采用分散式互惠导航策略的研究使用速度障碍来表示速度需要避开的时空区域。速度障碍最初由 [2]，提出，使得以数学方式表示速度约束变得更加容易。一种颇具前景的框架——最优互惠碰撞避免（ORCA） [3],——已被证明能够以几何方式实现最优碰撞规避。对于每个机器人，线性规划（LP）问题会最小化所选速度与期望速度之间的偏差，并将允许速度构造为 ORCA 半平面。若不存在有效解，则会最小化违反半平面的最大距离。尽管互惠决策机制能够保证碰撞规避，但通常无法避免或解决死锁。环境中存在密集障碍物时，情况会更加糟糕。

受速度障碍和 ORCA 框架的启发，我们提出一种二次规划（QP）形式，用于解决具有线性运动学约束的非完整机器人碰撞规避问题。该形式是一种广义的其完整约束情形的形式与原始 ORCA 问题类似。该公式可应用于现实世界中大多数采用线性运动学模型的可控轮式移动机器人。

此外，本文提出了屏蔽速度和掩蔽协作碰撞避免（MCCA）半平面，以促进无死锁导航。我们通过提高某些机器人的优先级来实现临时的局部协作，并将屏蔽速度用作机器人更新并尝试接近的无死锁速度意图。屏蔽速度在机器人之间传播，会逐渐影响其他机器人的实际速度，从而持续促进无死锁导航。

我们以差速驱动机器人为例，展示了QP公式和MCCA方法的应用。结果表明，该方法能够高效地解决避碰问题；实验也证实，结合屏蔽速度、MCCA 半平面和机器人优先级判定，即使机器人之间的局部协作程度较低，也能可靠地解决各种场景中的死锁问题。该方法还适用于加速能力较低且难以操控的机器人。此外，由于无需显著增大机器人的半径，该公式使机器人在狭窄通道中具有更好的机动性。

本文其余部分的安排如下：第II节简要总结相关工作。第III节介绍用于解决非完整约束避碰问题的一般QP公式。第IV节介绍屏蔽速度和MCCA 半平面，并给出机器人优先级排序和无死锁导航算法。第V节结合QP公式和MCCA，为差速驱动机器人提供一个实例；第VI节则报告仿真结果。

## II. 相关工作

我们简要回顾速度障碍和最优互惠碰撞规避方面的相关工作。

作为早期探索， [2]提出了速度障碍物（VO）的概念，该方法针对圆形机器人和静态障碍物。VO 使得以数学形式表示速度约束变得简单。后来在 [4], 中，互惠速度障碍物（RVO）被.....

互惠速度障碍法（RVO）是对原始速度障碍法（VO）的一种扩展，旨在解决由 VO 引起的振荡问题。基本上，RVO 能够在速度发生变化时保持 VO 的稳定性，从而确保之后不会再次选择原始速度。它还证明了 RVO 在避免碰撞方面具有完备性。另一项改进 VO 的研究是 [5], ，该方法定义了有限时域速度障碍法（FVO），允许机器人避开其他机器人，但仅限于预先设定的时间间隔内。由此产生的 VO 呈截锥形。与原始 VO 相比，FVO 大幅扩大了解空间，因为它只需要在较短的时间间隔内避免碰撞。

ORCA [3]结合了互惠速度障碍法和有限时域速度障碍法的优势，为系统中的每个机器人求解一个线性规划（LP）问题，以确定其最优速度。假设机器人均为完整约束。对于每个机器人，求解一个 LP 问题，使所选速度与期望速度（由全局规划器给出）之间的偏差最小，并通过 ORCA 半平面构造允许的速度。[1] 鉴于该问题的独特结构，通过增量式地加入约束即可轻松求解 LP，并且能够保证得到全局最优解。下面介绍原始 ORCA 最常被提及的两项挑战。

第一项挑战源于完整约束假设。现实世界应用中的大多数机器人都受非完整约束。求得的速度可能无法直接、即时实现，因此目标轨迹与实际轨迹之间会出现偏差，从而可能导致碰撞。 [6], 提供了一种将差速驱动机器人的运动学纳入考虑的折中方案，该方案采用了 [7] 中所述的运动学适配方法，并引入了具有完全可控速度的有效中心，但代价是有效半径更大。不过，该方法使用的有效半径显著更大，且未探讨平移距离 $D$ 小得多的情况。 [8], 提出了一种类似的方法（非完整约束最优互惠碰撞规避），通过在非完整约束运动与理想的完整约束运动之间引入跟踪误差 $\epsilon$ ，扩大非完整约束机器人的半径。 [9] 和[10]中可以找到近期一些考虑互惠速度障碍法的非完整约束碰撞规避研究。

第二个挑战是多机器人系统中出现死锁和振荡的问题：两个或更多机器人被困在一起，长时间无法解决这种情况。虽然集中式决策机制（例如 [11], [12]和 [13]）最终可能解决问题，但对于机器人数量适中的情况而言，这种机制耗时过长且不稳定；此外，在离散时空中对这类机制进行评估和仿真也值得商榷。也有一些去中心化方法，例如[14]和 [15]。然而，这些机制大多无法应对狭窄通道。理想情况下，机器人之间的全局协调程度应尽可能低，同时又能在实时处理各种场景时有效避免死锁和振荡，因此我们将探索一种无死锁方法。

[1]如原始 ORCA 所建议，理想情况下应设置 $\boldsymbol{v}^{\text{opt}} = \boldsymbol{v}^{\text{current}}$；除非另有说明，本文均采用此设置。

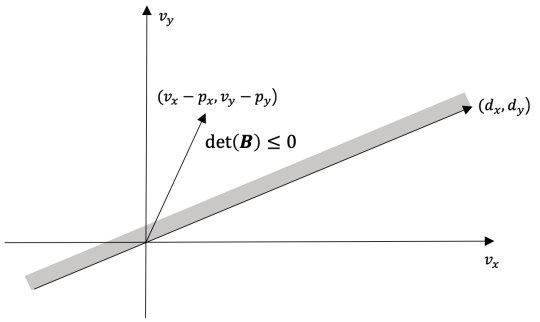



图 1：半平面及其允许速度示例。在速度空间中，一个允许的上半平面由位置 $\left(p_x^\zeta,\ p_y^\zeta\right)$ 和方向$\left(d_x^\zeta,\ d_y^\zeta\right)$ 描述。将位置 $\left(p_x^\zeta,\ p_y^\zeta\right)$ 平移至原点后，速度向量变为 $\left(v_x - p_x^\zeta,\ v_y - p_y^\zeta\right)$。如果速度位于该允许上半平面内部或边界上，则 $\boldsymbol{B} = \begin{bmatrix} v_x - p_x^\zeta & d_x^\zeta \\ v_y - p_y^\zeta & d_y^\zeta \end{bmatrix}$ 的行列式应为非正。

方向。

### III. 二次规划公式

与 ORCA 中的线性规划公式相比，我们希望碰撞规避公式具备以下关键特性。

- 可直接优化平移速度$\boldsymbol{v} = (v_x,\ v_y)$，使其接近 $\boldsymbol{v}$参考。
- 如果无法满足所有半平面约束，则应尽可能减小违反约束的程度。
- 控制输入和非完整运动学约束可以直接纳入其中。

可以构建一个相应的二次规划问题：

$$\min \quad \alpha_1 \left\| \boldsymbol{v} - \boldsymbol{v}^{\text{pref}} \right\|_2^2 + \alpha_2 \sum_{i \in \mathcal{O}} \delta_i^2 + \alpha_3 \sum_{j \in \mathcal{R}} \delta_j^2 \tag{1}$$

$$\text{s.t.} \quad \det\left(\boldsymbol{B}\right) \leq \delta_\zeta, \zeta \in \mathcal{O} \cup \mathcal{R}, \delta_\zeta \geq 0 \tag{1a}$$

$$\dot{\boldsymbol{q}} = \boldsymbol{S}(\boldsymbol{q}) \cdot \boldsymbol{r} \tag{1b}$$

$$\begin{bmatrix} \boldsymbol{F} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{r} \end{bmatrix} = 0. \tag{1c}$$

约束 (1a) 描述了当前机器人的允许速度半平面。每个半平面均由一个机器人 $\zeta \in \mathcal{R}$ 或静态障碍物 $\zeta \in \mathcal{O}$ 产生。图 1 解释了半平面与其允许速度之间的关系。通常，碰撞规避约束要求 $\det(\boldsymbol{B}) \leq 0$，这在拥挤的环境中可能难以实现。我们为 (1a) 中的每个约束引入一个辅助变量 $\delta_\zeta \geq 0$ ，并在目标函数中对正的辅助变量施加强惩罚，从而使碰撞规避约束始终得到满足，同时将速度违约降至最低。

约束 (1b) 和 (1c) 包含非完整约束运动学约束。非完整约束限制了机器人可能的速度或运动方向 [16]。考虑一个机器人的运动状态由广义坐标 $\boldsymbol{q} = [q_1, \cdots, q_n]^T$ 描述。 $\dot{\boldsymbol{q}} = [\dot{q_1}, \cdots, \dot{q_n}]^T$是

相应的速度向量。我们假设机器人具有 $m$ 个独立的运动控制电机或其他机构，记作 $\boldsymbol{r} = [r_1, \cdots, r_m]^T$。例如，对于差速驱动机器人，左轮速度和右轮速度是两个独立的运动控制变量。根据控制向量（控制输入）推导速度的过程称为前向状态估计或直接运动学模型，这一过程适用于每个时刻。在我们的公式 (1b) 中，假设这种关系是线性的，且 $\boldsymbol{S}(\boldsymbol{q}) = [\boldsymbol{s}_1(\boldsymbol{q}) \cdots, \boldsymbol{s}_m(\boldsymbol{q})]$ 是可达运动向量矩阵。用控制向量对这些向量进行线性组合，即可得到广义坐标下的速度。对于采用不同运动学模型的多数可控轮式移动机器人，例如差速驱动、三轮车式驱动、阿克曼驱动和拖车机器人 [16]，线性假设均成立。因此，所提出的公式适用于现实世界应用中的大多数移动机器人。约束 (1c) 表示针对速度向量或控制输入向量的定制线性约束，其中 $\boldsymbol{F}$ 和 $\boldsymbol{G}$ 是参数矩阵。定制约束的需求可能源于动力学和加速度限制、导航策略、法规等因素，从而提高该公式的通用性。

目标函数的第一部分是 $\boldsymbol{v} = (v_x, v_y)$与 $\boldsymbol{v}^{\text{pref}}$之间距离的平方。 $v_x$ 和 $v_y$ 是速度向量 $\dot{\boldsymbol{q}}$的一部分。 $\boldsymbol{v}^{\text{pref}}$ 由全局规划器给出。第二部分是对半平面违反的惩罚。因此，QP目标应在半平面违反为零或最小的条件下，使第一部分最小化。也就是说，在实践中我们令$\alpha_2$， $\alpha_3 \gg \alpha_1 > 0$ 。所提出的形式是一个定义明确的凸二次规划（CQP）问题，可以使用二次规划求解器在不到一毫秒的时间内高效求解。相比之下， [17] 中提出的一般方法需要大量数值计算，并且可能无法找到可行解。

如果从该形式中移除运动学约束（1b）和（1c），则可用于解决全向机器人的碰撞规避问题；此外，如果将 ORCA 半平面应用于约束（1a），我们称全向二次规划形式试图利用ORCA 半平面接近 $\boldsymbol{v}^{\text{pref}}$ [2]，，这与ORCA中的线性形式类似。

## IV. 掩蔽协作碰撞规避

互惠碰撞避免行为或共同责任带来的另一个挑战，可能是在拥挤环境或狭窄通道中导致死锁或振荡。死锁避免是一项全局目标，如果没有机器人偏离其原始路径，该目标就会失败。因此，应牺牲一些机器人以顾全大局。尽管在这种情况下鼓励采用集中式决策，但这很容易导致耗时的求解过程。在本节中，我们将展示屏蔽速度、MCCA 通过完全分布式决策和速度规划流程，结合半平面和机器人优先级判定，快速避免并解决可能出现的死锁。

### A. 屏蔽速度与 MCCA

我们将屏蔽速度定义为机器人更新并尝试趋近的无死锁意图，该意图也会公开给系统中的所有其他机器人。意图的更新会受到机器人正在避让其他机器人或正在被其他机器人避让这一状态的影响。假设所有机器人 $\mathcal{R}$ 都可以标记为高优先级或普通优先级， $A \in \mathcal{R}$ 和 $B \in \mathcal{R}$分别表示具有高优先级的头部机器人（需要被避让）和具有普通优先级的普通机器人（需要避让）。机器人 $A \in \mathcal{R}$ 能够在不考虑其他机器人的屏蔽速度的情况下更新其屏蔽速度 $\boldsymbol{v}_A^{\text{m}}$ ；而机器人 $B \in \mathcal{R}$ 则必须将其他机器人的屏蔽速度考虑在内，才能求得自己的屏蔽速度 $\boldsymbol{v}_B^{\text{m}}$。令

$$\boldsymbol{v}_A^{\overline{\text{m}}} = \underset{\boldsymbol{v}^{\text{m}} \in ORCA^{\tau}_{A|\mathcal{O}}}{\operatorname{argmin}} \left\| \boldsymbol{v}^{\text{m}} - \boldsymbol{v}_A^{\text{pref}} \right\|, \tag{2}$$

其中 $ORC\bar{A}_{A|\mathcal{O}} = \bigcap_{o \in \mathcal{O}} ORC\bar{A}_{A|o}$. And let

$$\boldsymbol{v}_B^{\text{m}} = \underset{\boldsymbol{v}^{\text{m}} \in MCCA^{\tau}_B}{\operatorname{argmin}} \left\| \boldsymbol{v}^{\text{m}} - \boldsymbol{v}_B^{\text{pref}} \right\|, \tag{3}$$

其中 $MCCA^{\tau}_B = ORCA^{\tau}_{B|\mathcal{O}} \cap \bigcap_{C \in \mathcal{R}} MCCA^{\tau}_{B|C}$。请注意，与原始 ORCA 不同，屏蔽速度不受 $D(\mathbf{0}, v_B^{\max})$ 的限制。更大的屏蔽速度会带来更高效的死锁消解行为。

方程 (2) 中的头部屏蔽速度 $\boldsymbol{v}_A^{\text{m}}$ 通过完整约束二次规划（如第III节所述）求解，以利用 ORCA 半平面避开静态障碍物 $ORCA^{\tau}_{A|\mathcal{O}}$，并接近$\boldsymbol{v}^{\text{pref}}$ 。这描述了这样一种意图：头部机器人 $A \in \mathcal{R}$ 试图接近其当前 $\boldsymbol{v}^{\text{pref}}$ ，而不考虑所有其他机器人。

公式 (3) 中的普通屏蔽速度 $\boldsymbol{v}_B^{\text{m}}$ 通过全向二次规划求解，以利用 MCCA 半平面$MCCA^{\tau}_B$ 考虑静态障碍物和所有其他机器人，从而趋近于 $\boldsymbol{v}^{\text{pref}}$ 。这描述了这样一种意图：普通机器人 $B \in \mathcal{R}$只有在避开所有静态障碍物 $ORCA^{\tau}_{B|\mathcal{O}}$ 以及所有其他机器人的屏蔽速度的条件下，才可能期望趋近于其期望速度。

屏蔽速度通过全向二次规划求解，原因有二。首先，屏蔽速度并非用于推导控制输入的实际速度，因此不受运动学约束的限制。相反，它用于屏蔽速度传播，并影响其他机器人的实际速度。其次，不考虑运动学约束时，求得的屏蔽速度通常更大，而且往往需要更多时间步才能达到，这会在持续时间和幅值两方面强化其他所有机器人的无死锁意图，而普通机器人必须对此加以考虑。

普通机器人 $B$ 的屏蔽速度会受到 $\mathcal{R}$ 和 $\mathcal{R}$ 中其他机器人的影响。在公式（4）中，我们提出了以下掩蔽协作碰撞规避（MCCA）

[2]在 ORCA 中生成半平面时，机器人与遵循共同责任原则的另一机器人之间存在相互许可的半平面，而机器人与静态障碍物之间的半平面则有所不同，机器人应对碰撞规避承担全部责任。

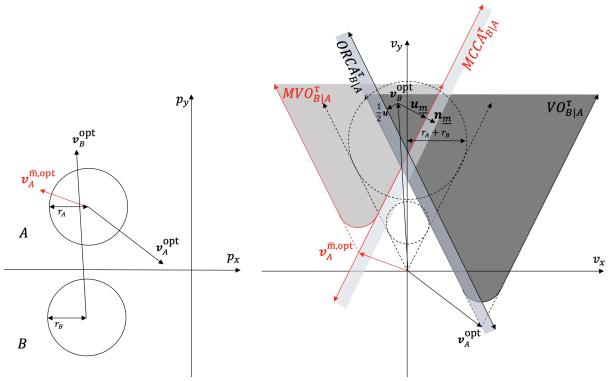



图 2：MVO 和 MCCA 半平面的示意图。对于 $B \in \mathcal{R}$，其相对于 $A \in \mathcal{R}$ 的 VO 由（$\boldsymbol{v}_A^{\text{opt}}$，$\boldsymbol{v}_B^{\text{opt}}$）构建，并应用 $A$ 与 $B$ 之间的共同责任原则（因此得到 $\frac{1}{2}\boldsymbol{u}$），最终形成 $ORCA^{\tau}_{B|A}$ 半平面。$B$ 相对于 $A$、$MVO^{\tau}_{B|A}$ 的 MVO 由（$\boldsymbol{v}^{\text{m}}_{,A}{}^{\text{opt}}$，$\boldsymbol{v}_B^{\text{opt}}$）构建。$\boldsymbol{u}_{\text{m}}$ 是从 $-\boldsymbol{v}_B^{\text{opt}} - \boldsymbol{v}^{\text{m}}_{,A}{}^{\text{opt}}$ 指向 $MVO^{\tau}_{B|A}$ 边界上最近点的向量，而 $\boldsymbol{n}_{\text{m}}$ 是该边界在点（$\boldsymbol{v}_B^{\text{opt}} - \boldsymbol{v}^{\text{m}}_{,A}{}^{\text{opt}}$）处的外法线。$MCCA^{\tau}_{B|A}$ 通过仅调整 $\boldsymbol{u}_{\text{m}}$ 来构建 $B$。

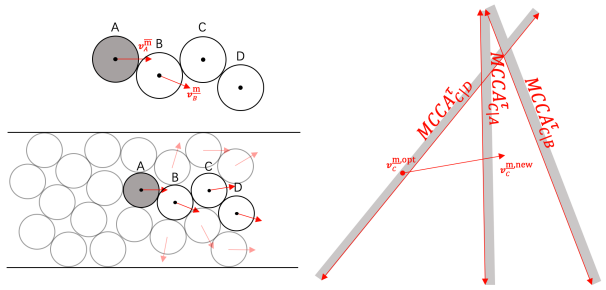



图 3：死锁情形下屏蔽速度的传播。头部机器人 $A$ 的屏蔽速度已传播至普通机器人 $B$，而普通机器人 $C$ 正在此时刻被传播。当普通机器人 $D$ 在不久后被传播时，$MCCA^{\tau}_{C|D}$ 将向右移动，从而扩大 $C$的屏蔽速度。我们可以将这一过程描述为：头部机器人的死锁规避意图从近到远向普通机器人逐步传播，后者则为头部机器人和较早被传播的普通机器人让出空间（向后传播）。

半平面，用于表示针对另一台机器人 $C$的允许屏蔽速度半平面 $B$：

$$MCCA^{\tau}_{B|C} = \left\{\boldsymbol{v}^{\text{m}} | (\boldsymbol{v}^{\text{m}} - (\boldsymbol{v}_B^{\text{opt}} + \boldsymbol{u}_{\text{m}})) \cdot \boldsymbol{n}_{\text{m}} \geq 0\right\}, C \in \mathcal{R}. \tag{4}$$

图 2 展示了普通机器人的屏蔽速度障碍（MVO）和 MCCA 半平面的概念。每个控制周期内，机器人都会针对所有其他机器人更新其屏蔽速度和 MCCA 半平面。为求得该机器人的实际避碰速度，我们调整通用的 QP 目标函数，使其

$$\min \alpha_1 \left\|\boldsymbol{v} - \boldsymbol{v}^{\text{pref}}\right\|_2^2 + \alpha_2 \sum_{i \in \mathcal{O}} \delta_i^2 + \alpha_3 \sum_{j \in \mathcal{R}} \delta_j^2 + \alpha_4 \sum_{k \in \mathcal{M}} \delta_k^2, \tag{5}$$

其中，$\mathcal{M}$ 是针对所有其他机器人的 MCCA 半平面集合，$\mathcal{O}$ 和 $\mathcal{R}$ 分别是针对障碍物和所有其他机器人的 ORCA 半平面集合。如前所述，$\mathcal{M} = \emptyset$ 适用于头部机器人，因为机器人无需避开其他机器人的屏蔽速度。因此，公式（1）中的目标对于领航机器人仍然成立。对于每个普通机器人，都要求 $\alpha_2$，$\alpha_3 \gg \alpha_4 \gg \alpha_1$，这样优先级最高的目标就是与静态障碍物和其他机器人保持安全距离；接下来必须避开所有其他机器人的屏蔽速度，以实现避免死锁的意图；最后，机器人可以追上其期望速度。在每个控制周期中，所有其他机器人的屏蔽速度都可能通过 MCCA 半平面影响当前机器人的实际速度；当该机器人的控制周期结束，其屏蔽速度和实际速度更新后，附近其他机器人的屏蔽速度和实际速度也可能在后续控制周期中受到影响。因此，如图 3 所示，屏蔽速度或无死锁意图可以在机器人之间于时间和空间中传播。

公式（4）中的 $\boldsymbol{u}_{\text{m}}$ 不除以 2，以确保具有普通优先级的机器人 $B$ 全权负责避开头部机器人的屏蔽速度。此外，这也有助于普通优先级机器人之间屏蔽速度的前向传播和后向传播。考虑到 $\boldsymbol{u}_{\text{m}}$ 的系数为 $0 < \beta < 1$，每当头部机器人 $A$ 的屏蔽速度传播到一组未受此前传播的普通机器人 MVO 影响的普通机器人 $\mathcal{L}$ 时，$\mathcal{L}$ 中机器人的 MCCA 半平面的平移量就会按 $\beta$ 的比例减小，这意味着前向传播很快也会减弱。反过来，后向传播也会受到影响，因为为了避开头部机器人，较早接收到传播的普通机器人可能需要更多空间，而 $\mathcal{L}$ 无法提供这些空间，因为 $\mathcal{L}$ 中机器人的屏蔽速度几乎不受影响。

### *B.* 机器人优先级判定

我们采用一种完全分布式方法，使每个机器人在每个时间步自主判定其优先级，然后才能应用屏蔽速度。图 4 展示了机器人优先级判定算法。由于领航机器人的屏蔽速度只尝试避开障碍物的 ORCA 半平面，因此，$\mathcal{R}$ 中两台领航机器人之间的屏蔽速度冲突无法解决，并可能导致死锁。因此，应首先通过优先级判定解决冲突，以确保领航机器人之间不会发生死锁。

在算法中，即使 $A_i$ 位于 $\mathcal{R}$中，若 $A_i$ 要成为领航机器人，也必须确保其作为领航机器人求得的屏蔽速度不会与 $\mathcal{R}$中的任何领航机器人发生冲突。因此，图 4 中的术语 $\boldsymbol{v}^{\text{m}}_{A_i}$ 适用于 $\mathcal{R}$ 中的所有机器人，以进行优先级判定。MVO 锥用于检测机器人 $A_i \in \mathcal{R}$ 与 $A_j \in \overline{\mathcal{R}}$ 从现在起任意时刻是否发生冲突，从而保证只要 $A_i$的屏蔽速度 $\boldsymbol{v}^{\text{m}}_{A_i}$ 不在 $MVO^{\infty}_{A_i|A_j}$ 中，这两台机器人在屏蔽速度空间中就能确定无死锁。此外，即使$\boldsymbol{v}^{\text{m}}_{A_i}$ 位于 $MVO^{\infty}_{A_i|A_j}$ 中，若屏蔽速度$\boldsymbol{v}^{\text{m}}_{A_i}$ 在屏蔽速度 $\boldsymbol{v}^{\text{m}}_{A_j}$ 上的投影与$\boldsymbol{v}^{\text{m}}_{A_j}$方向相同，或者等价地说，$\boldsymbol{v}^{\text{m}}_{A_i} \cdot \boldsymbol{v}^{\text{m}}_{A_j} < 0$，由此引发的死锁

**算法1：** $A_i$的机器人优先级判定

**输入** $\mathcal{R}$: 所有机器人 $A_i$: 当前机器人
**输出**已更新 $A_i$的优先级
初始化 $T_{A_i}=0,\ S_{A_i}=0,\ A_i\in\mathcal{R}$ —
**如果**($A_i$ 到达其目标) **那么**
  将 $A_i$ 设置为普通机器人， $T_{A_i}=0$、 $S_{A_i}=0$;
**else**
  **若** ($T_{A_i}>0$) **则**
    将 $A_i$ 设为普通机器人， $T_{A_i}=T_{A_i}-1$;
  **else**
    **若** ($\exists\ A_j\in\overline{\mathcal{R}}$ 满足 $\overline{\boldsymbol{v}}^{\mathrm{m}}_{A_i}\in MVO^{\infty}_{A_i|A_j}$ 且 $\overline{\boldsymbol{v}}^{\mathrm{m}}_{A_i}\cdot\overline{\boldsymbol{v}}^{\mathrm{m}}_{A_j}<0$ 且 $S_{A_i}<S_{A_j}$) **则**
      将 $A_i$ 设为普通机器人， $T_{A_i}=\eta$;
    **else**
      将 $A_i$ 设为领航机器人， $S_{A_i}=S_{A_i}+1$
    **end**
  **end**
**end**

图 4：机器人优先级判定算法。$\eta$ 是预设整数，表示新判定为普通机器人且与其他领航机器人发生冲突时的初始禁忌步数。只有当剩余禁忌步数 $T_{A_i}$ 为零时，该机器人才有机会提升优先级。 $S_{A_i}$ 表示 $A_i$ 明显处于高优先级的控制周期数，是一个累积指标，用于标记 $A_i$ 的重要性。一旦机器人到达当前目标， $T_{A_i}$ 和 $S_{A_i}$ 都会重置为零。

**算法 2：** MCCA

**输入** $\mathcal{R}$：所有机器人 $\mathcal{O}$：所有障碍物
**输出** 机器人控制输入
初始化;
循环过程
**对于** $each\ A_i\in\mathcal{R}$ **执行** 控制周期
  感知 $A_i$的中心位置/速度;
  感知障碍物线段;
  接收每个 $A_{j\neq i}\in\mathcal{R}$ 的状态（中心位置/速度、半径、屏蔽速度和优先级）;
  对 $A_i$执行算法1;
  构造 $ORCA^{\tau}_{A_i|\mathcal{O}}$;
  **对于** $A_{j\neq i}\in\mathcal{R}$， **执行**
    构造 $ORCA^{\tau}_{A_i|A_j}$;
    如果 $A_i\in\mathcal{R}$，则构造 $MCCA^{\tau}_{A_i|A_j}$ ;
  **end**
  从所有 $ORCA^{\tau}_{A_i|A_j}$ 和 $ORCA^{\tau}_{A_i|\mathcal{O}}$中获取 $ORCA^{\tau}_{A_i}$ ;
  获取 $MCCA^{\tau}_{A_i}$ 来自所有 $MCCA^{\tau}_{A_i|A_j}$ 以及 $ORCA^{\tau}_{A_i|\mathcal{O}}$ 如果 $A_i\in\mathcal{R}$;
  获取期望速度 $\boldsymbol{v}^{\mathrm{pref}}_{A_i}$ ;
  使用全向二次规划求解屏蔽速度 $\boldsymbol{v}^{\mathrm{m}}_{A_i}$ ;
  使用非完整约束二次规划求解控制输入 $\boldsymbol{r}_{A_i}$ ;
  将控制输入应用于 $A_i$的执行器
**end**

图 5：掩蔽协作碰撞规避算法。

$A_i$ 与 $A_j$ 不太可能发生，因为后方的机器人可以跟随前方的机器人。在 $\boldsymbol{v}^{\mathrm{m}}_{A_i}$ 位于 $MVO^{\infty}_{A_i|A_j}$ 且$\overline{\boldsymbol{v}}^{\mathrm{m}}_{A_i}\cdot\overline{\boldsymbol{v}}^{\mathrm{m}}_{A_j}\geq 0$的条件下，若机器人 $A_i$ 的累积重要性$S_{A_i}$ 高于任何此类 $A_j$，该算法就会将任何此类 $A_j$ 替换为$\mathcal{R}$，从而确保消除死锁隐患。引入累积重要性 $S_{A_i}$ 可避免每台机器人在高优先级与普通优先级之间频繁切换，并保持死锁消解的一致性。

### C. 求解过程

通过结合 QP公式与屏蔽速度的概念，我们在图 5 中总结了掩蔽协作式碰撞规避的求解过程。MCCA 的决策过程完全去中心化，每个机器人的状态对所有机器人均可用。

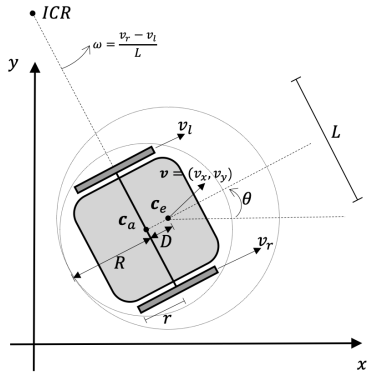



图 6：差速驱动机器人的运动学。左轮速度和右轮速度分别由独立的电机控制。 $r$ 是车轮半径。 $L$ 是两个车轮之间的距离。轮轴中心 $\boldsymbol{c}_a$ 的速度始终垂直于车轴，因此无法实现完全控制。有效中心 $\boldsymbol{c}_e$ 将机器人等效为一个半径为 $R+D$ 的圆盘，且 $\boldsymbol{v}=(v_x,\ v_y)$。为了便于计算角速度 $\omega$ [16]，引入瞬时旋转中心（ICR）。如图所示，我们将 $\boldsymbol{c}_e-\boldsymbol{c}_a$ 的方向定义为机器人航向。

## V. 差速驱动机器人的示例

鉴于上述介绍的特性，本节将详细介绍差速驱动机器人的实现方法。图 6 给出了差速驱动机器人的基本运动学模型。对于此类机器人，轮轴中心并非完全可控，因此我们采用 [7] 中的有效中心和半径。然而，仅为实现运动学适配，确保$D>0$ 即可。我们不谨慎选择数值以免导致圆盘半径显著增大，而是假设 $D/R\approx 0.1$ 或更小，该数值远小于 [6] 中建议的值（$D=R$）。

表 I：示例中的变量和参数

| **变量** | **描述** |
|---|---|
| $v_{\mathrm{l}}$ | 左轮线速度 |
| $v_{\mathrm{r}}$ | 右轮线速度 |
| $\boldsymbol{v}=(v_x,\ v_y)$ | $\boldsymbol{c}_e$的平移速度向量 |
| $\delta_\zeta$ | 半平面的辅助变量 $\zeta$ |
| $\omega$ | 机器人的角速度 |
| $\delta^{\omega}$ | 角速度的辅助变量 |
| **参数** | **描述** |
| $v^{\mathrm{c}}_{\mathrm{l}}$ | 当前左轮线速度 |
| $v^{\mathrm{c}}_{\mathrm{r}}$ | 当前右轮线速度 |
| $\theta$ | 当前航向 |
| $\boldsymbol{v}^{\mathrm{pref}}$ | 机器人的期望速度向量 |
| $v^{\max}$ | 车轮的最大线速度 |
| $a^{\max}$ | 车轮的最大线加速度 |
| $\Delta t$ | 时间步 |
| $\boldsymbol{v}^{\mathrm{opt}}_H$ | 完整约束优化速度 |
| $\theta^{\mathrm{opt}}_H$ | 完整约束优化航向 |
| $\mu$ | 角度控制级别 |

### A. 实际速度的非完整约束二次规划公式

公式（6）给出了用于求解实际速度的差速驱动二次规划公式，表 I 总结了其中涉及的变量和参数。其目标函数已根据公式（5）进行了调整。除基本的差速驱动运动学约束外，还添加了一种特殊的角度控制约束（6g），将其作为定制约束（1c），以解决差速驱动机器人特有的问题，该问题将在下一小节中介绍。

约束 (6a) 包含所有机器人（$\mathcal{M} = \emptyset$ 领航机器人）的 MCCA 半平面，以及针对所有机器人和障碍物的 ORCA 半平面。约束 (6b) 至 (6f) 描述了差速驱动机器人的运动学约束。$\boldsymbol{v} = (v_x,\ v_y)$ 是有效中心的速度向量，该中心与轴心相距 $D > 0$。约束 (6b)、(6c) 和 (6d) 建立了正运动学，其中 $v_\mathrm{l}$ 和 $v_\mathrm{r}$ 是控制输入。约束 (6e) 和 (6f) 对控制输入的最大速度和加速度作出限制。对于每个机器人，$a^{\max}\Delta t$ 是时间步长为 $\Delta t$时，两个相邻控制周期之间车轮线速度可能发生的最大变化。

$$\min \quad \alpha_1 \left\| \boldsymbol{v} - \boldsymbol{v}^{\mathrm{pref}} \right\|_2^2 + \tag{6}$$
$$\alpha_2 \sum_{i\in\mathcal{O}} \delta_i^2 + \alpha_3 \sum_{j\in\mathcal{R}} \delta_j^2 + \alpha_4 \sum_{k\in\mathcal{M}} \delta_k^2 + \alpha_5 \delta_\omega^2$$

$$\text{s.t.} \quad \det \begin{pmatrix} v_x - p_x^\zeta & d_x^\zeta \\ v_y - p_y^\zeta & d_y^\zeta \end{pmatrix} \le \delta_\zeta, \zeta \in \mathcal{O}\cup\mathcal{R}\cup\mathcal{M}, \delta_\zeta \ge 0 \tag{6a}$$

$$v_x = \left(\frac{\cos\theta}{2} + \frac{D\sin\theta}{L}\right) v_l + \left(\frac{\cos\theta}{2} - \frac{D\sin\theta}{L}\right) v_r \tag{6b}$$

$$v_y = \left(\frac{\sin\theta}{2} - \frac{D\cos\theta}{L}\right) v_l + \left(\frac{\sin\theta}{2} + \frac{D\cos\theta}{L}\right) v_r \tag{6c}$$

$$\omega = \frac{v_r - v_l}{L} \tag{6d}$$

$$|v_l|, |v_r| \le v^{\max} \tag{6e}$$

$$|v_l - v_l^{\mathrm{c}}|, |v_r - v_r^{\mathrm{c}}| \le a^{\max}\Delta t \tag{6f}$$

$$\int_0^T |\omega| - \frac{2a^{\max}\cdot t}{L}\, dt \le \theta' + \delta^\omega, \delta^\omega \ge 0, \tag{6g}$$

$$T = \frac{\theta' L}{2a^{\max}}, \theta' = \begin{cases} \frac{\angle(\theta,\theta_H^{\mathrm{opt}})}{\mu} & \text{if } \boldsymbol{v}_\omega \cdot \boldsymbol{v}_H^{\mathrm{opt}} > 0 \\ \angle(\theta,\theta_H^{\mathrm{opt}}) & \text{otherwise} \end{cases}$$

### B. 用于航向振荡规避的角度控制

当 $D$ 较小且考虑加速度时，我们发现了航向振荡问题。该现象如图 7 所示。航向振荡的原因有两方面。首先，我们考虑一个处于空环境中的差速驱动机器人，因此 $\boldsymbol{v}_H^{\mathrm{opt}} = \boldsymbol{v}^{\mathrm{pref}}$ 保持不变。图 7 中 $\tilde{\boldsymbol{v}}$ 与 $\boldsymbol{v}$ 之间的角度为 $\Delta\alpha = \arctan\frac{\omega D}{|v|}$。较小的 $D$ 会导致角度过小，以至于当 $\tilde{\boldsymbol{v}}$ 与 $\boldsymbol{v}_H^{\mathrm{opt}}$ 之间的角度趋近于零时，机器人

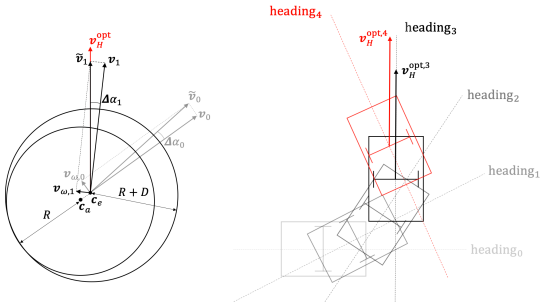



图 7：差速驱动机器人发生航向振荡。$\boldsymbol{v}_H^{\mathrm{opt}}$ 是每个时间步仅考虑约束 (6a) 时，二次规划 (6) 的解。非完整约束机器人无法立即达到 $\boldsymbol{v}_H^{\mathrm{opt}}$，但会在运动学约束下尝试达到该速度。每个时间步求得的机器人有效中心速度可分解为$\tilde{\boldsymbol{v}} = \boldsymbol{v}_\omega + \boldsymbol{v}$，其中 $\boldsymbol{v}_\omega$ 垂直于机器人航向方向，且 $|\boldsymbol{v}_\omega| = \omega D$。超调现象是指，当 $\tilde{\boldsymbol{v}}$ 非常接近$\boldsymbol{v}_H^{\mathrm{opt}}$ 时，机器人仍可能具有较大的角速度 $\omega$，从而使机器人航向进一步偏离目标全向速度及其航向（航向4与航向3）。

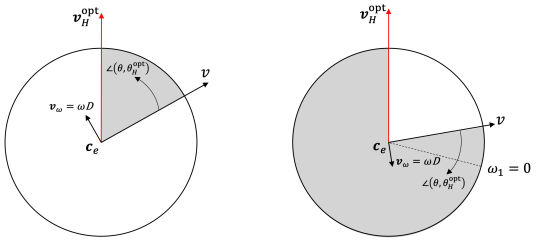



图 8：$\theta'$的两种情况。对于右侧的 $\boldsymbol{v}_\omega \cdot \boldsymbol{v}_H^{\mathrm{opt}} \le 0$，机器人不必像左侧的机器人那样穿过阴影角度区域，也可以在虚线处将 $\omega$ 减小至零，然后反向增大 $\omega$，以更快到达 $\boldsymbol{v}_H^{\mathrm{opt}}$（类似于 S 形转弯）。$\omega$ 反向后，就变成了左侧的第一种情况。

在控制输入将 $\omega$ 减速至零之前，它仍可能具有较大的角速度并发生过冲，从而导致航向振荡。其次，不变的 $\boldsymbol{v}_H^{\mathrm{opt}}$ 假设并不恰当。它会随着 $\boldsymbol{v}^{\mathrm{pref}}$ 以及其他机器人的状态变化而不断改变。如果求解得到的较新 $\boldsymbol{v}_H^{\mathrm{opt}}$ 与当前机器人航向非常接近，那么一旦其角速度无法及时降至零，机器人就会发生过冲。

为避免航向振荡，我们引入约束 (6g)。定义 $\angle(\theta,\ \theta_H^{\mathrm{opt}})$ 为机器人航向 $\theta$ 沿当前 $\boldsymbol{v}_\omega$ 方向转向以到达 $\boldsymbol{v}_H^{\mathrm{opt}}$ 所需转过的角度。基本上，角度控制约束会限制 $\omega$ 的增大，使当前 $\omega$ 能够在行进角度达到当前 $\angle(\theta,\ \boldsymbol{v}_H^{\mathrm{opt}})$之前，利用最大角加速度 $\frac{2a^{\max}}{L}$ 减小至零。

为进一步减少因 $\boldsymbol{v}_H^{\mathrm{opt}}$ 变化而导致航向振荡的第二种原因，我们会在角度小于 $\pi$ 或等价地小于 $\boldsymbol{v}_\omega \cdot \boldsymbol{v}_H^{\mathrm{opt}} > 0$ 时，将 $\theta'$ 除以 $\mu > 1$，以防止 $\omega$ 增加过多。然而，在以下情况下不会将 $\theta'$ 除以 $\mu > 1$。当角度 $\angle(\theta,\ \theta_H^{\mathrm{opt}})$ 至少为 $\pi$，或等价地为 $\boldsymbol{v}_\omega \cdot \boldsymbol{v}_H^{\mathrm{opt}} \le 0$ 时，差速驱动机器人可能会选择执行 S 形转弯，如图 8 所示。当 $\angle(\theta,\ \theta_H^{\mathrm{opt}})$ 更接近 $2\pi$ 时，机器人更有可能执行这种 S 形转弯动作。因此

在这种情况下，我们不会将 $\theta'$ 减少 $\mu > 1$。

## VI. 实验

我们介绍了 MCCA 在各种场景下差速驱动机器人的实现和仿真结果。

### A. 实现细节

所有机器人均采用差速驱动，在三维空间中的尺寸为 950mm $\times$ 725 mm$\times$ 285mm。每个机器人配备两个半径为 $r = 0.1$ m 的动力轮，以及两个用于保持机器人平衡的被动脚轮。最大线速度为 2 m/s，最大线性加速度为 2 m/s$^2$。每个动力轮均由伺服电机控制；根据控制输入，伺服电机可使车轮达到最大线性加速度。$\boldsymbol{c}_a$ 与 $\boldsymbol{c}_e$ 之间的距离为 $D = 0.015$ m。机器人外接圆半径为$R = 0.485$ m。有效半径为 $R+D = 0.5$ m，仅比 $R$大 3%。每个控制周期的时间步为 $\Delta t = 0.25$ s。时间范围为 $\tau = 17$ s。公式（6）中的权重参数为 $\alpha_1 = 10^{-2}$、$\alpha_2 = 10^4$、$\alpha_3 = 10^2$、 $\alpha_4 = 1$、 $\alpha_5 = 2 \times 10^4$。角度控制参数为 $\mu = 9$。初始禁忌步数为 $\eta = 30$。

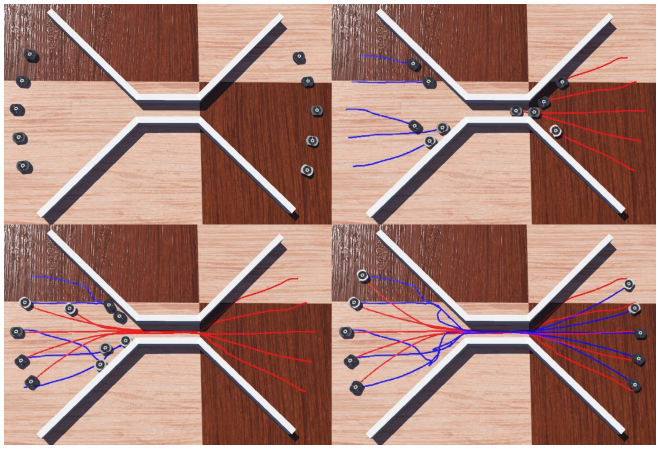
在每个仿真场景中，我们使用广播服务接收所有机器人的状态，并将这些状态广播给系统中的所有机器人。在每个时间步，每个机器人都会根据自身状态和其他机器人的广播状态来确定其控制输入。为开展本研究，仿真不包含感知和定位环节；在现实环境中，这通常通过同时定位与建图（SLAM）实现。不过，我们为每个机器人的位置和角度加入均匀分布噪声，其幅度分别为 0.01 米和 1 度。我们采用一种简单的方法，将机器人的期望速度 $\boldsymbol{v}^{\text{pref}}$ 设为从当前位置指向目标，且速度大小不超过最大速度。我们的仿真使用 Webots [18], 这一移动机器人仿真软件。我们使用 PROX-QP [19] 求解仿真中提出的二次规划问题。

我们设计这些仿真场景，以验证 MCCA 在两个方面的能力。首先，碰撞规避是最低要求。其次，死锁避免应当稳定且高效，并且即使在容易发生死锁的场景中也能发挥作用。此外，我们还验证了角度控制在差速驱动机器人避免航向振荡方面的有效性。

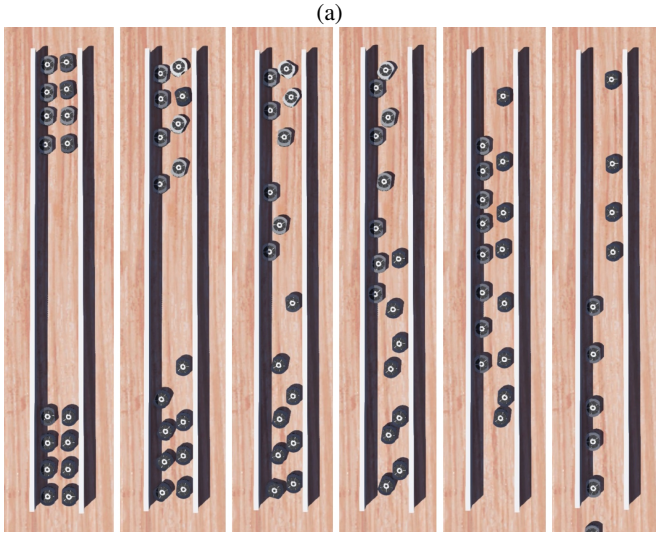
### B. 仿真结果

我们简要介绍这些场景。

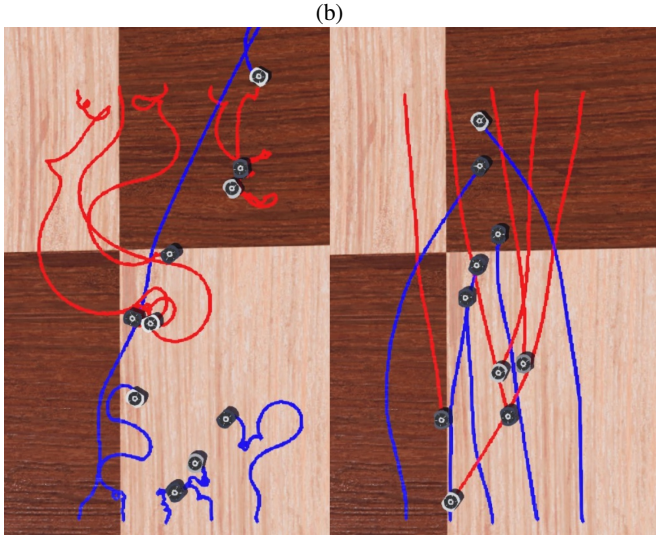
- 场景1：两组各有五台机器人的队伍，在单车道通道中超越对侧队伍的位置。
- 场景2：两组各有八台机器人，在双车道通道中行驶，并自发、顺畅地相互礼让。
- 场景3：十台机器人在多个单车道通道中反复往返。



(a)



(b)



(c)

图9：部分仿真结果。(a) 场景1：单车道通道中，一侧的5台机器人超越另一侧5台机器人的位置。(b) 场景2：双车道通道两侧各有8台机器人，试图到达对侧。(c) 场景7：为机动性较差的差速驱动机器人引入角度控制前后，其最大线性加速度从2m/s$^2$降至0.2m/s$^2$。

- 场景4：二十台机器人在障碍物密集分布的环境中反复往返。
- 场景5：四十台机器人在狭窄且拥挤的区域内反复往返行驶。
- 场景6：三十四台机器人汇入一条通道。
- 场景7：十台机动性较差的机器人反复往返行驶。

所有列出的场景均在随附视频中演示。场景3、4、5和7涉及反复往返行驶，在数百个仿真小时内始终保持无碰撞、无死锁，并且之后也依然如此。如果机器人持续超越另一侧的位置，场景1也同样如此。场景1、2和7的部分结果总结如下。

对于场景1，图9a展示了每台机器人的轨迹。如果两侧的机器人都涌入通道，可能会发生死锁，部分机器人会长时间来回移动。我们的方法能够在局部自发地避免死锁。一些机器人会绕行或在通道外等待，为通道内的机器人让路。对于场景2，图9b展示了一组屏幕截图，其中两车道通道两侧各有8台机器人相遇。这些机器人不仅会以类似拉链式合流的方式给另一侧的机器人让路，还能在没有集中控制或路径突变的情况下顺利完成这一过程。对于场景7，图9c表明，角度控制能够有效处理相较于最大速度而言加速度较低的机器人。若不采用角度控制，可以观察到这些机器人会突然停止、出现航向振荡和绕圈问题。角度控制有效地解决了这些问题。此外，机器人的半径仅略有增大，使这些机动性较差的机器人即使在拥挤区域和狭窄通道内也能平稳行驶。

## VII. 结论

本文首先提出了一种多目标通用二次规划公式，用于解决多机器人系统中的碰撞规避问题。该公式构成一个定义明确、易于求解的凸优化问题，适用于大多数可控轮式移动机器人，并可纳入线性定制约束。在此基础上，我们提出 MCCA 方法，在最大限度减少全局协调的同时，有效避免并解决死锁，从而维持完全分布式的决策机制。我们以数学方式定义了屏蔽速度和 MCCA 半平面，以实现让机器人在本地协同解决死锁情形的意图。通用 QP 与 MCCA 配合使用，确保每个机器人都能沿着无碰撞、无死锁且平滑的路径行进。为此，我们详细介绍了针对差速驱动机器人的实现方法，并通过在 QP 公式中加入定制的角度控制约束来解决航向振荡问题，使机动性较差的机器人能够在仿真中平稳行进，而无需显著增大有效半径。仿真结果显示，该方法在复杂且的环境中实现死锁避免的效果非常出色。极其困难的环境，这使我们相信，MCCA 方法在现实世界应用中具有巨大优势和潜力。

## 参考文献


[1] Y. Sheffi，《style id='2'>城市交通网络：基于数学规划方法的均衡分析。Prentice Hall，1984年1月。 [2] P. Fiorini 和 Z. Shiller，“使用速度障碍在动态环境中进行运动规划”，《国际机器人研究杂志》，第17卷，第7期，第760–772页，1998年。 [3] J. van den Berg、S. J. Guy、M. Lin 和 D. Manocha，“互惠 n 体碰撞规避”，载于机器人研究：第*14*届国际研讨会 *ISRR*。Springer，2011年，第3–19页。 [4] J. van den Berg、M. Lin 和 D. Manocha，“用于实时多智能体导航的互惠速度障碍”，载于*2008 IEEE* 国际机器人与自动化会议，2008年，第1928–1935页。 [5] S. J. Guy、J. Chhugani、C. Kim、N. Satish、M. Lin、D. Manocha 和 P. Dubey，“Clearpath：面向多智能体仿真的高度并行碰撞规避”，载于*2009 ACM SIGGRAPH/Eurographics* 计算机动画研讨会论文集，2009年，第177–187页。 [6] J. Snape、J. Van Den Berg、S. J. Guy 和 D. Manocha，“满足差速驱动约束的多机器人平滑无碰撞导航”，载于*2010 IEEE/RSJ* 智能机器人与系统国际会议。IEEE，2010年，第4584–4589页。 [7] B. Kluge、D. Bank、E. Prassler 和 M. Strobel，《style id='20'>拥挤的自然环境中人类与机器人的运动协调。柏林，海德堡：Springer Berlin Heidelberg，2005年，第207–219页。 [8] J. Alonso-Mora、A. Breitenmoser、M. Rufli、P. Beardsley 和 R. Siegwart，《style id='23'>多台非完整机器人最优互惠碰撞规避。柏林，海德堡：Springer Berlin Heidelberg，2013年，第203–216页。 [9] J. Alonso-Mora、P. Beardsley 和 R. Siegwart，“非完整机器人的协作式碰撞规避”，*IEEE*《机器人学汇刊》，第34卷，第2期，第404–420页，2018年。 [10] M. Sainte Catherine 和 E. Lucet，“一种改进的混合互惠速度障碍方法，用于无通信条件下的多机器人运动规划”，载于*2020 IEEE/RSJ* 智能机器人与系统国际会议（*IROS*），2020年，第5708–5714页。 [11] M. ˇC´ap、J. Gregoire 和 E. Frazzoli，“在延迟扰动下对多机器人规划进行可证明安全且无死锁的执行”，载于*2016 IEEE/RSJ* 智能机器人与系统国际会议（*IROS*），2016年，第5113–5118页。 [12] S. Dergachev、K. Yakovlev 和 R. Prakapovich，“结合 theta*、ORCA 与推送和旋转的多智能体导航方法”，载于交互式协作机器人，A. Ronzhin、G. Rigoll 和 R. Meshcheryakov 编。Cham：Springer International Publishing，2020年，第55–66页。[13] S. Dergachev 和 K. Yakovlev，“基于互惠碰撞规避和局部受限多智能体寻路的分布式多智能体导航”，载于*2021 IEEE* 第*17*届自动化科学与工程国际会议（*CASE*）。IEEE Press，2021年，第1489–1494页。[14] S. H. Arul 和 D. Manocha，“V-RVO：利用 Voronoi 图和互惠速度障碍实现分散式多智能体碰撞规避”，载于*2021 IEEE/RSJ* 智能机器人与系统国际会议（*IROS*），2021年，第8097–8104页。[15] H. Wang 和 M. Rubenstein，“行走、停止、计数与交换：具有理论保证的分散式多智能体寻路”，*IEEE*《机器人学与自动化快报》，第5卷，第2期，第1119–1126页，2020年。 [16] G. Klanˇcar、A. Zdeˇsar、S. Blaˇziˇc 和 I. ˇSkrjanc，“第2章——移动机器人的运动建模”，载于轮式移动机器人学，G. Klanˇcar、A. Zdeˇsar、S. Blaˇziˇc 和 I. ˇSkrjanc 编。Butterworth-Heinemann，2017年，第13–59页。 [17] D. Bareiss 和 J. Van den Berg，“广义互惠碰撞规避”，《国际机器人研究杂志》，第34卷，第12期，第1501–1514页，2015年。 [18] O. Michel，“Cyberbotics ltd. WebotsTM：专业移动机器人仿真”，《国际先进机器人系统杂志》，第1卷，第1期，第5页，2004年。 [19] A. Bambade、S. El-Kazdadi、A. Taylor 和 J. Carpentier，“PROX-QP：另一个面向机器人学及更广领域的二次规划求解器”，载于*RSS 2022*——机器人学：科学与系统，美国纽约，2022年6月。