

实验二：差速驱动机器人的运动学与控制

1 实验介绍

本次实验的目标是为差速驱动机器人编程实现一个闭环运动控制器，主要有以下三个子任务：

1) 给定机器人所需的前进速度和角速度，计算相应的车轮速度，以使机器人能在仿真平台中相应地驱动。要求在 MATLAB 中编程实现。

2) 开发一个闭环控制器，计算速度命令，将机器人驱动到指定的目标位置。要求在 MATLAB 中编程实现，并在模拟环境中进行测试。

3) 对上一步实现的控制器进行改进，使机器人能够恒定速度行驶，并且可以倒车。要求在 MATLAB 中编程实现，并在模拟环境中进行测试。

2 任务一：前馈控制（30 分）

差速控制机器人的运动学模型由以下两个公式描述：

$$v = \frac{r\dot{\phi}_r}{2} + \frac{r\dot{\phi}_l}{2} \quad (1)$$

$$\omega = \frac{r\dot{\phi}_r}{2l} - \frac{r\dot{\phi}_l}{2l} \quad (2)$$

其中， v 和 ω 分别表示机器人的前进速度和旋转速度。 $\dot{\phi}_l$ 和 $\dot{\phi}_r$ 分别表示左右轮子的转速。 r 和 l 分别表示车轮半径和内轮距的一半。

任务：利用公式 (1) 和公式 (2) 推导出 $\dot{\phi}_l$ 和 $\dot{\phi}_r$ 的计算公式，并在 `calculateWheelSpeeds.m` 中编写代码实现。

验证：在 MATLAB 中运行 `test/testCircleDrive.m`，对实现的前馈控制器进行验证。如果代码无误，输出结果应该为半径等于 0.5m 的圆型轨迹。

3 任务二：闭环控制（40 分）

对《自主移动机器人导论》一书中描述的线性状态反馈控制律进行实现。该控制律由公式 (3) 和公式 (4) 描述，其中变量与图1中对应。

$$v = k_\rho \rho \quad (3)$$

$$\omega = k_\alpha \alpha + k_\beta \beta \quad (4)$$

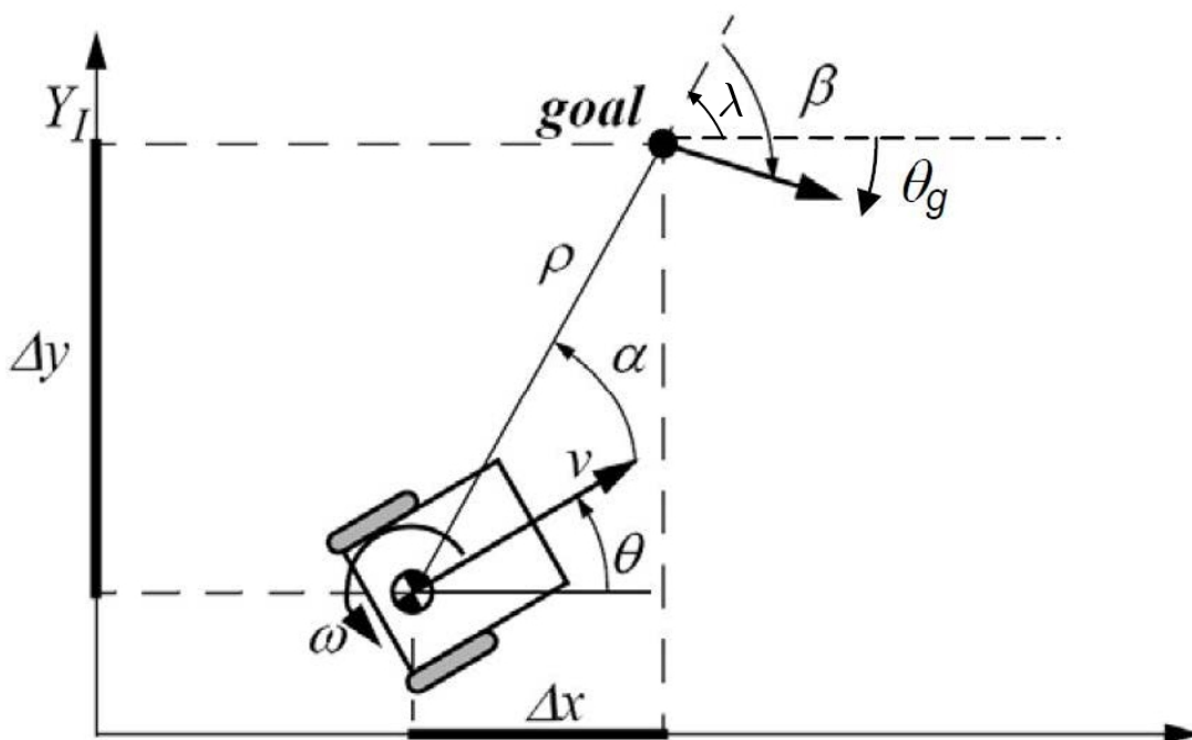




图 1: 参考姿态的状态反馈控制

任务: 在 `CalculateControlOutput.m` 中实现闭环控制器。需要用到的 Matlab 命令/文件有: `atan2` 和 `normalizeAngle.m`。需要用到的传入变量及其含义已在代码注释中给出, 需要实现的代码已用“TODO”标注。

验证: 启动 Coppeliasim, 加载场景 `scene/mooc_exercises.ttt`。在 MATLAB 中运行程序 `vrep/vrep-Simulation.m`。如果代码无误, 机器人应该开始向绿色的 `GhostBobTarget` 行驶, 并在与其相同的位姿时停下。在 Coppeliasim 的场景层次中选中 `GhostBobTarget`, 并在代码运行时使用 Coppeliasim 工具栏中的 “” 工具在场景视图中拖动绿色的 `GhostBobTarget` 以更改其位置, 使用 “” 工具在场景视图中拖动绿色的 `GhostBobTarget` 以更改其方向。多次运行, 以验证控制器是否正常工作。

4 任务三: 闭环控制 (改进) (30 分)

虽然上述控制律结构简单, 稳定性条件强, 但也存在一些缺点。主要存在以下两个问题:

1) 尽管机器人能够前后行驶, 但上述控制律只会输出前进速度, 因此在某些情况 (如目标在机器人身后的情况, 请使用传入的布尔参数 “`parameters.backwardAllowed`” 进行判断) 下会导致相当麻烦的运动。应推导并实现一种方法将上述控制律推广到前进和后退速度。

2) 在仿真平台上对任务二进行模拟时, 我们可以观察到, 当机器人接近目标位姿时, 其速度呈指数下降。应推导并实现一种方法, 使机器人执行与上述控制律相同的轨迹, 但保持恒定的前进速度。(提示: 传入的布尔参数 “`parameters.useConstantSpeed`” 表示此时机器人能否使用恒定速度, 传入的参数 “`parameters.constantSpeed`” 表示机器人能使用恒定速度时所使

用的恒定速度 v 。通过保持 v 和 ω 之间的商不变来实现机器人轨迹不变。)

任务: 为了解决上述问题,请改进 `calculateControlOutput.m` 中的代码。机器人应该能够根据它相对于目标姿势的位置,在前进/后退两个方向上行驶(15分),并且保持恒定速度(15分)。需要用到的传入变量及其含义已在代码注释中给出,需要实现的代码已用“TODO”标注。

验证: 启动 Coppeliasim,加载场景 `scene/mooc_exercises.ttt`。在 MATLAB 中运行程序 `vrep/vrep-Simulation.m`。如果代码无误,机器人应该开始向绿色的 GhostBobTarget 行驶,并在与其相同的位姿时停下。在 Coppeliasim 中更改 GhostBobTarget 的位置和方向,多次运行,以验证控制器是否正常工作。

5 需提交内容

1. 实验报告

- (1). 题目
- (2). 姓名-学号-班级
- (3). 实验内容分析: 比如对实验内容的理解、关键点、思路
- (4). 实验过程分析: 比如每一步的解析
- (5). 实验结果分析
- (6). 遇到的问题和心得

2. 实验源代码

3. 仿真视频