

实验四：基于扩展卡尔曼滤波器的机器人定位

1 实验介绍

机器人定位是一项极其重要的任务，其目的在于实时计算机器人所在的位置和方向。上一个实验演示了如何获取扫描测距仪感知的线性结构的更抽象表示。本次实验将以实验三为基础，给定线性特征地图，对机器人实现根据其感知的线性结构来定位自身。

2 卡尔曼滤波定位（90 分）

本实验中用于定位的扩展卡尔曼滤波器（EKF）可分为预测（prediction）步骤和更新（update）步骤。下面分别介绍这两个步骤。

2.1 状态预测（50 分）

实验二讨论了差动驱动机器人的运动模型。已知状态 $\mathbf{x}_{t-1} = [x_{t-1}, y_{t-1}, \theta_{t-1}]^T$ ，上一时间步的车轮位移 $\mathbf{u}_t = [\Delta s_l, \Delta s_r]^T$ ，运动模型可用于获得当前状态 $\hat{\mathbf{x}}_t$ 的先验估计。其公式如下：

$$\hat{\mathbf{x}}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t) = \mathbf{x}_{t-1} + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos\left(\theta_{t-1} + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin\left(\theta_{t-1} + \frac{\Delta s_r - \Delta s_l}{2b}\right) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix} \quad (1)$$

可以合理地假设运动受到噪声的影响，因此需要将其建模为加性高斯噪声 $\nu \sim \mathcal{N}(0, \mathbf{Q})$ 应用于控制输入。其公式如下：

$$\mathbf{Q} = \begin{bmatrix} k|\Delta s_l| & 0 \\ 0 & k|\Delta s_r| \end{bmatrix} \quad (2)$$

从而得到状态协方差的先验估计，其公式如下。其中 \mathbf{F}_x 和 \mathbf{F}_u 分别表示状态转移函数关于状态 \mathbf{x}_{t-1} 和控制输入 \mathbf{u}_t 的雅可比矩阵。

$$\hat{\mathbf{P}}_t = \mathbf{F}_x \cdot \mathbf{P}_{t-1} \cdot \mathbf{F}_x^T + \mathbf{F}_u \cdot \mathbf{Q}_t \cdot \mathbf{F}_u^T \quad (3)$$

任务：推导状态转移函数相对于状态和控制输入的雅可比函数 $\hat{\mathbf{F}}_x$ 和 $\hat{\mathbf{F}}_u$ 。并在 **transition-Function.m** 中编程实现。

验证：在 MATLAB 中运行 **test/validateTransitionFunction.m**，此函数使用一系列控制输入，通过提供的运动模型前向估计状态。如果实现正确，该函数将使用运动模型绘制真实路径以及

噪声控制输入的前向估计。可以发现，在实验过程中，算法实现的估计值与真实路径分歧越来越大。这说明，对于存在噪声扰动的实际应用，仅依赖互感知信息是不够的。

提示：为方便编写代码，可进行如下定义：

$$\begin{aligned}\Delta s &= \frac{\Delta s_r + \Delta s_l}{2} \\ \Delta \theta &= \frac{\Delta s_r - \Delta s_l}{b}\end{aligned}\quad (4)$$

则原式变为：

$$\hat{\mathbf{x}}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t) = \mathbf{x}_{t-1} + \begin{bmatrix} \Delta s \cos\left(\theta_{t-1} + \frac{\Delta \theta}{2}\right) \\ \Delta s \sin\left(\theta_{t-1} + \frac{\Delta \theta}{2}\right) \\ \Delta \theta \end{bmatrix} \quad (5)$$

2.2 状态更新（40 分）

通过上一节的实验可知，控制输入的扰动将导致对机器人状态的估计会越来越不准确。因此，外部感知位置线索在机器人应用中得到了广泛的应用。在本实验中，机器人能够感知线性结构，并拥有一个地图 \mathbf{M} ，其中包含其操作环境中的所有线性结构，用一个称为世界坐标系的坐标系表示。

2.2.1 测量函数（20 分）

正如实验三中介绍的，直线可以参数化为 $\mathbf{m}^i = [\alpha^i, r^i]^T$ ，这种参数化同时应用于感知测量模型 $\hat{\mathbf{z}}_t$ 的输出以及地图 \mathbf{M} 的构建。然而，虽然参数化方式是相同的，但表示测量值和地图的坐标系是不同的。地图中的线条在世界坐标系中表示，而机器人感知的却是其自身坐标系中相对于自身不同姿势的线条测量值。因此，需要通过将地图中的线从世界坐标系转换为机器人自身坐标系来模拟测量。如下图所示，左下角为世界坐标系。

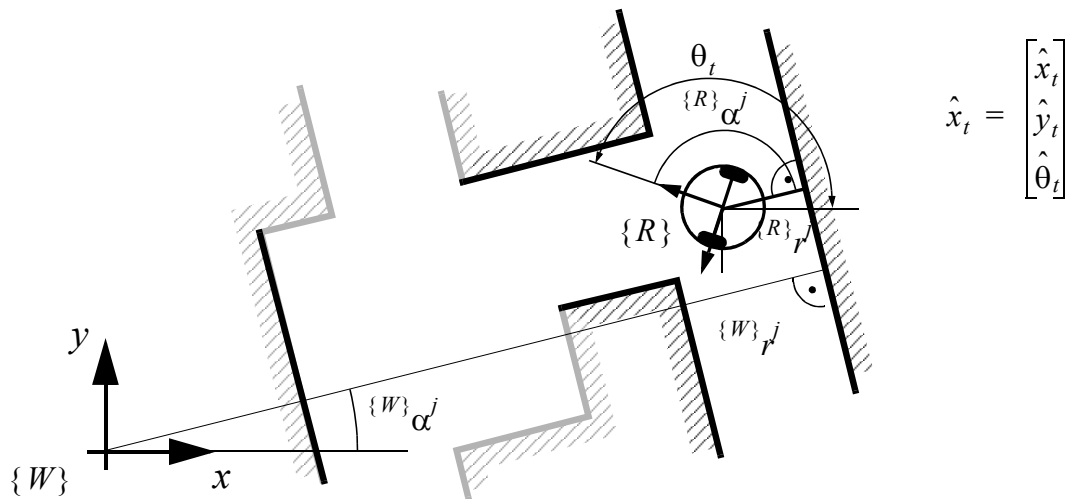


图 1: 坐标转换

任务：推导描述 \mathbf{m}^i 的测量模型 $\hat{\mathbf{z}}_t = h(\hat{\mathbf{x}}_t, \mathbf{m}^{(i)})$ ，及其雅各比矩阵 $\hat{\mathbf{H}}_{\mathbf{x}}$ ，并在 `measurement-Function.m` 中编程实现。

验证：在 MATLAB 中运行 `test/validateMeasurementFunction.m` 。通过函数打印的文本判断是否编写正确。

2.2.2 测量关联

为了正确应用卡尔曼滤波器更新，需要建立观测值和地图条目之间的关联。因此这里采用度量测量值 \hat{z}_t^i 和观测值 z_t^j 之间的马氏距离。其计算方式如下

$$\begin{aligned} \mathbf{v}_t^{ij} &= \mathbf{z}_t^j - \hat{\mathbf{z}}_t^i \\ \Sigma_{IN_t}^{ij} &= \hat{\mathbf{H}}_t^i \cdot \hat{\mathbf{P}}_t \cdot \hat{\mathbf{H}}_t^{iT} + \mathbf{R}_t^j \\ d_t^{ij} &= \mathbf{v}_t^{ijT} \cdot \left(\Sigma_{IN_t}^{ij} \right)^{-1} \cdot \mathbf{v}_t^{ij} \end{aligned} \quad (6)$$

在现实世界的机器人应用中，总是会出现与地图中的条目不对应的污染测量值。在引言中介绍的示例中，可能是之前关闭的门打开了，也可能是家具四处移动了。因此，需要引入一个 **validation gate** g ，只有低于该阈值的 d 才考虑对测量值进行关联（即观测到的线与地图中的线之间的距离不是太大时）。当多个地图条目 m 落入单个测量的 **validation gate** 时，测量与具有最小马氏距离的条目 m 相关联。另一方面，多个测量值可能与单个地图条目相关联。

2.2.3 更新估计（20 分）

本节之前的任务为实现扩展卡尔曼滤波器更新提供了基本的构建模块。最终需要对卡尔曼滤波进行更新。更新公式如下：

$$\begin{aligned} \mathbf{S} &= \mathbf{H}\mathbf{P}_{t-1}\mathbf{H}^T + \mathbf{R} \\ \mathbf{K} &= \mathbf{P}_{t-1}\mathbf{H}^T\mathbf{S}^{-1} \\ \mathbf{P}_t &= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}_{t-1} \\ \mathbf{x}_t &= \mathbf{x}_{t-1} + \mathbf{K}\mathbf{v} \end{aligned} \quad (7)$$

任务：实现单个完整的滤波步骤（包括状态预测、测量关联和状态更新），在 `filterStep.m` 中编程实现。

验证：在 MATLAB 中运行 `test/validateFilter.m`。通过函数打印的文本判断是否编写正确。

3 CoppeliaSim 仿真（10 分）

任务：在仿真环境中对上面实现的基于卡尔曼滤波器的机器人定位进行测试。

启动仿真平台，加载场景，在 MATLAB 中运行 `vrep/vrepSimulation.m` 开始模拟。机器人平台应开始沿圆形路径移动。在靠近实际机器人的地方，你应该会看到一个黄色的“幽灵”，它可以根据你的定位估计出姿势。

4 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