自动驾驶中的SLAM技术

第一章作业思路提示

主讲人 陈梓杰

第一题

分别使用左右扰动模型，计算：$\dfrac{\partial \boldsymbol{R}^{-1}\boldsymbol{p}}{\partial \boldsymbol{R}}$.

左扰动：

$$
\begin{aligned}
\frac{\partial \mathbf{R}^{-1}\mathbf{p}}{\partial \mathbf{R}} &= \lim_{\phi \to 0} \frac{(\mathrm{Exp}(\phi)\mathbf{R})^{-1}\mathbf{p} - \mathbf{R}^{-1}\mathbf{p}}{\phi} \\
&= \lim_{\phi \to 0} \frac{\mathbf{R}^{-1}\mathrm{Exp}(-\phi)\mathbf{p} - \mathbf{R}^{-1}\mathbf{p}}{\phi} \\
&\approx \lim_{\phi \to 0} \frac{\mathbf{R}^{-1}(\mathbf{1} - \phi^{\wedge})\mathbf{p} - \mathbf{R}^{-1}\mathbf{p}}{\phi} \\
&= \lim_{\phi \to 0} \frac{-\mathbf{R}^{-1}\phi^{\wedge}\mathbf{p}}{\phi} \\
&= \lim_{\phi \to 0} \frac{\mathbf{R}^{-1}\mathbf{p}^{\wedge}\phi}{\phi} \\
&= \mathbf{R}^{-1}\mathbf{p}^{\wedge}
\end{aligned}
$$

右扰动：

$$
\begin{aligned}
\frac{\partial \mathbf{R}^{-1}\mathbf{p}}{\partial \mathbf{R}} &= \lim_{\phi \to 0} \frac{(\mathbf{R}\mathrm{Exp}(\phi))^{-1}\mathbf{p} - \mathbf{R}^{-1}\mathbf{p}}{\phi} \\
&= \lim_{\phi \to 0} \frac{\mathrm{Exp}(-\phi)\mathbf{R}^{-1}\mathbf{p} - \mathbf{R}^{-1}\mathbf{p}}{\phi} \\
&\approx \lim_{\phi \to 0} \frac{(\mathbf{1} - \phi^{\wedge})\mathbf{R}^{-1}\mathbf{p} - \mathbf{R}^{-1}\mathbf{p}}{\phi} \\
&= \lim_{\phi \to 0} \frac{-\phi^{\wedge}\mathbf{R}^{-1}\mathbf{p}}{\phi} \\
&= \lim_{\phi \to 0} \frac{(\mathbf{R}^{-1}\mathbf{p})^{\wedge}\phi}{\phi} \\
&= (\mathbf{R}^{-1}\mathbf{p})^{\wedge}
\end{aligned}
$$

第二题

深蓝学院 shenlanxueyuan.com

分别使用左右扰动模型，计算：$\dfrac{\partial \boldsymbol{R}_1\boldsymbol{R}_2^{-1}}{\partial \boldsymbol{R}_2}$.

左扰动：

$$\begin{aligned}
\frac{\partial \mathbf{R}_1\mathbf{R}_2^{-1}}{\partial \mathbf{R}_2} &= \lim_{\phi\to 0}\frac{\mathrm{Log}(\mathbf{R}_1(\mathbf{Exp}(\phi)\mathbf{R}_2)^{-1}) - \mathrm{Log}(\mathbf{R}_1\mathbf{R}_2^{-1})}{\phi} \\
&= \lim_{\phi\to 0}\frac{\mathrm{Log}(\mathbf{R}_1\mathbf{R}_2^{-1}\mathbf{Exp}(-\phi)) - \mathrm{Log}(\mathbf{R}_1\mathbf{R}_2^{-1})}{\phi} \\
&\approx \lim_{\phi\to 0}\frac{\mathrm{Log}(\mathbf{R}_1\mathbf{R}_2^{-1}) - \mathbf{J}_r^{-1}(\mathbf{R}_1\mathbf{R}_2^{-1})\phi - \mathrm{Log}(\mathbf{R}_1\mathbf{R}_2^{-1})}{\phi} \\
&= -\mathbf{J}_r^{-1}(\mathbf{R}_1\mathbf{R}_2^{-1})
\end{aligned}$$

右扰动：

$$\begin{aligned}
\frac{\partial \mathrm{Log}(\mathbf{R}_1\mathbf{R}_2^{-1})}{\partial \mathbf{R}_2} &= \lim_{\phi\to 0}\frac{\mathrm{Log}(\mathbf{R}_1(\mathbf{R}_2\mathrm{Exp}(\phi))^{-1}) - \mathrm{Log}(\mathbf{R}_1\mathbf{R}_2^{-1})}{\phi} \\
&= \lim_{\phi\to 0}\frac{\mathrm{Log}(\mathbf{R}_1\mathrm{Exp}(-\phi)\mathbf{R}_2^{-1}) - \mathrm{Log}(\mathbf{R}_1\mathbf{R}_2^{-1})}{\phi} \\
&= \lim_{\phi\to 0}\frac{\mathrm{Log}(\mathbf{R}_1\mathbf{R}_2^{-1}\mathrm{Exp}(-\mathbf{R}_2\phi)) - \mathrm{Log}(\mathbf{R}_1\mathbf{R}_2^{-1})}{\phi} \\
&\approx \lim_{\phi\to 0}\frac{\mathrm{Log}(\mathbf{R}_1\mathbf{R}_2^{-1}) - \mathbf{J}_r^{-1}(\mathbf{R}_1\mathbf{R}_2^{-1})\mathbf{R}_2\phi - \mathrm{Log}(\mathbf{R}_1\mathbf{R}_2^{-1})}{\phi} \\
&= -\mathbf{J}_r^{-1}(\mathbf{R}_1\mathbf{R}_2^{-1})\mathbf{R}_2
\end{aligned}$$

第三题

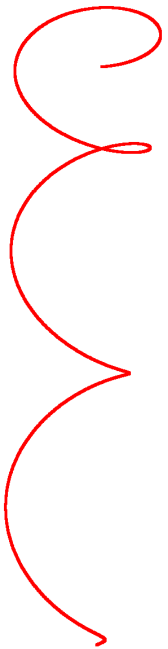
将实践环节中的运动学修改成带且一定角速度的平抛运动。车辆受固定的Z轴角速度影响，具有一定的初始水平速度，同时受$-Z$方向的重力加速度影响。请修改程序，给出动画演示。

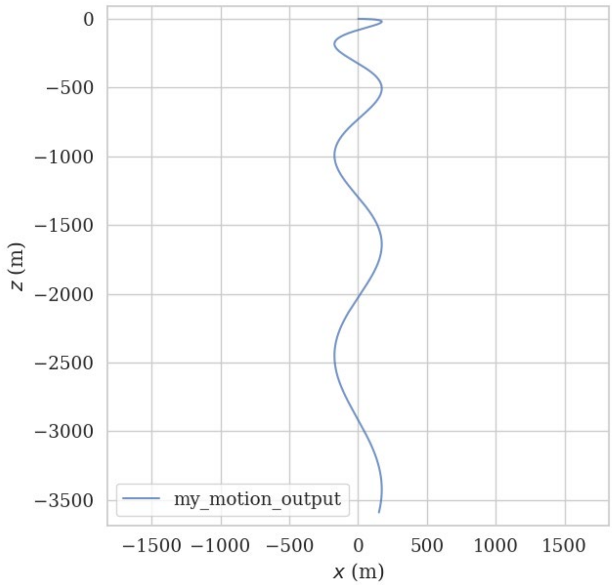
假设小车在世界系受到的加速度为$\mathbf{a}^w=\mathbf{g}^w$，以及固定的Z轴角速度$\boldsymbol{\omega}=[0,0,\omega_z]^\top$。位置、速度、姿态更新公式为：

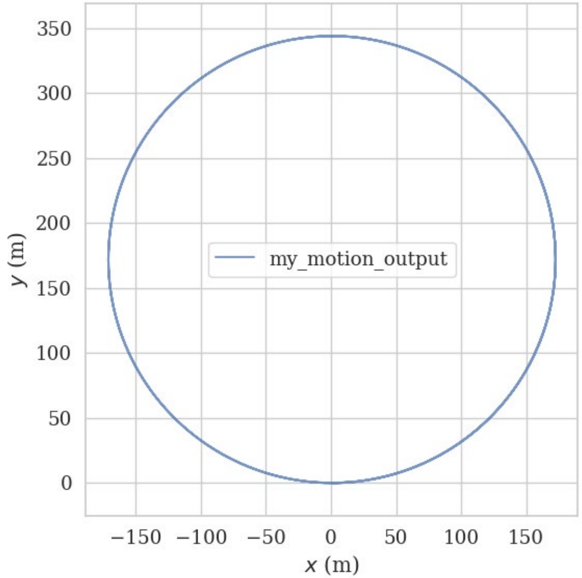
$$
\begin{aligned}
\mathbf{v}_k^w &= \mathbf{R}_b^w \mathbf{v}_k^b \\
\mathbf{p}_{b_{k+1}}^w &= \mathbf{p}_{b_k}^w + \mathbf{v}_k^w \Delta t + \frac{1}{2}\mathbf{a}^w \Delta t^2 \\
\mathbf{a}_k^b &= \mathbf{R}_{b_k}^{w\,\top} \mathbf{a}^w \\
\mathbf{v}_{k+1}^b &= \mathbf{v}_k^b + \mathbf{a}_k^b \Delta t \\
\mathbf{R}_{b_{k+1}}^w &= \mathbf{R}_{b_k}^w \mathrm{Exp}(\boldsymbol{\omega}\Delta t)
\end{aligned}
$$

第三题

深蓝学院
shenlanxueyuan.com







第四题

摘自**路古**同学的作业：

- 1.调整步长的方式：
  - Gauss-Newton算法使用局部近似的海森矩阵（或雅可比矩阵的转置乘以雅可比矩阵）来计算步长。它假设目标函数在当前估计的参数附近是近似线性的，并尝试沿着负梯度方向进行下降。
  - Levenberg-Marquardt算法通过添加一个调整项来修正海森矩阵，使其能够处理目标函数的非线性性质。这个调整项由一个参数λ控制，称为阻尼因子。通过调整阻尼因子的大小，Levenberg-Marquardt算法可以在高度非线性的区域和线性的区域之间平衡。
- 2.更新步长的计算：
  - Gauss-Newton算法计算步长的公式为：$\Delta x = -(J^T J)^{-1} J^T r$，其中$J$是雅可比矩阵，$r$是残差向量。这是一个线性方程求解的问题。
  - Levenberg-Marquardt算法计算步长的公式为：$(J^T J + \lambda I)\Delta x = -J^T r + O(\lambda)$，其中$\lambda$是阻尼因子，$I$是单位矩阵，$O(\lambda)$是一个调整项。这是一个带有调整项的线性方程求解问题。
- 3.收敛性和稳定性：
  - Gauss-Newton算法在目标函数局部线性可导且残差小的情况下通常收敛速度较快。然而，当目标函数存在高度非线性或残差较大的情况时，可能会导致收敛困难或发散。
  - Levenberg-Marquardt算法通过引入阻尼因子来提高收敛性和稳定性。当阻尼因子较小时，类似于Gauss-Newton算法；当阻尼因子较大时，类似于梯度下降算法。通过自适应地调整阻尼因子，Levenberg-Marquardt算法可以更好地应对不同类型的非线性问题。

Levenberg-Marquardt算法相对于Gauss-Newton算法具有更强的适应性和稳定性，但计算复杂度较高。

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening