1 Learning aim

The learning aim is to learn an Inverse Kinematics model of the robot based on the robot dataset ds0.

The inverse kinematics model to be learnt takes as input the current robot pose $[x, y, \theta]$, the next robot pose $[x', y', \theta']$, the previous control command $u_{t_{\textbf{previous}}} = [v_{t-1}, \omega_{t-1}]$, and a time duration Δt. The output should be the required control command $[v, \omega]$ to achieve the next robot pose within the given Δt. This learned model will be compared to both the ground truth control commands given in the dataset, and the designed inverse kinematics model. The inverse kinematics model formulation derived from the motion model [1] is detailed here for reference.

$$\omega = \frac{\theta' - \theta}{\Delta t} \tag{1}$$

$$\text{For } \omega \neq 0 : v = \frac{\omega(y' - y)}{\cos\theta - \cos\theta + \omega\Delta t} \quad \text{or} \quad \frac{\omega - (x' - x)}{\sin\theta - \sin\theta + \omega\Delta t} \tag{2}$$

$$\text{For } \omega = 0 : v = \frac{y' - y}{\Delta t \sin\theta} \quad \text{or} \quad \frac{x' - x}{\Delta t \cos\theta} \tag{3}$$

Substituting equation 1 into 2, we get:

$$\text{For } \omega \neq 0 : v = \frac{(\theta' - \theta)(y' - y)}{\Delta t(\cos\theta - \cos\theta')} \quad \text{or} \quad \frac{(\theta - \theta')(x' - x)}{\Delta t(\sin\theta - \sin\theta')} \tag{4}$$

A maximum acceleration limit of $\dot{v}_{max} = 0.288\text{ms}^{-2}$ and $\dot{\omega}_{max} = 5.579\text{rads}^{-2}$ was imposed. Limits on the angular acceleration are applied immediately after calculation of ω (equation 1) and limits on the linear acceleration are applied after calculation of v (equations 2-4).

1. Calculate the linear and angular accelerations if the computed velocity command were to be issued:

$$\dot{v} = \frac{v_t - v_{t-1}}{\Delta t} \qquad \dot{\omega} = \frac{\omega_t - \omega_{t-1}}{\Delta t}$$

2. Check if these accelerations exceed the abovementioned limits. If yes, throttle the velocity command such that the acceleration stays within the imposed limits. sgn is the sign function.

$$\text{If } \dot{v} > \dot{v}_{max}, v_t = \text{sgn}(\dot{v})\dot{v}_{max}\Delta t + v_{t-1}$$
$$\text{If } \dot{\omega} > \dot{\omega}_{max}, v_t = \text{sgn}(\dot{\omega})\dot{\omega}_{max}\Delta t + \omega_{t-1}$$

2 Dataset formulation

The dataset is formulated from the robot's ground truth pose (ds0_Groundtruth.dat) and the issued control commands (ds0_Odometry.dat). From inspection of the dataset, the time steps in the ground truth pose and the time steps where control commands are issued are asynchronous. To learn the inverse kinematics function as described above, we would need to synchronize the time steps to build a dataset that maps inputs to outputs at each single time step. The method chosen to synchronise the time steps is outlined as follows:
Let the ground truth robot pose data be represented by q and the ground truth control commands be represented by u, where q and u are time series.
For each control command logged in ds0_Odometry.dat (excluding the last):

1. Set
$$\begin{aligned} t_{\textbf{start}} &= \text{current control command time} \\ t_{\textbf{end}} &= \text{next control command time} \\ \Delta t &= t_{\textbf{end}} - t_{\textbf{start}} \end{aligned}$$

2. Find the corresponding time step, or the ones right before and after in the ground truth pose data, for both $t_{\textbf{start}}$ and $t_{\textbf{end}}$
$$\begin{aligned} t_{\textbf{before\_start}} &= t \text{ in ground truth data where } t \leq t_{\textbf{start}} \\ t_{\textbf{after\_start}} &= t \text{ in ground truth data where } t \geq t_{\textbf{start}} \\ t_{\textbf{before\_end}} &= t \text{ in ground truth data where } t \leq t_{\textbf{end}} \\ t_{\textbf{after\_end}} &= t \text{ in ground truth data where } t \geq t_{\textbf{end}} \end{aligned}$$

3. Find the corresponding ground truth pose at $t_{\textbf{start}}$

If $t_{\textbf{before\_start}} = t_{\textbf{after\_start}}$:
There is a corresponding control command time step in the ground truth data
$[x, y, \theta] = q_{(t_{\textbf{before\_start}})}$
Else:
Perform interpolation to get the current pose
$$r_{\text{interpolate}} = \frac{t_{\textbf{start}} - t_{\textbf{before\_start}}}{t_{\textbf{after\_start}} - t_{\textbf{before\_start}}}$$
$$[x, y, \theta] = r_{\text{interpolate}} * (q_{(t_{\textbf{after\_start}})} - q_{(t_{\textbf{before\_start}})}) + q_{(t_{\textbf{before\_start}})}$$

4. Repeat step 3 with $t_{\textbf{end}}$ to get $[x', y', \theta']$

5. Build each row of the time-synchronised dataset as
$$\text{concatenate}([[x, y, \theta], [x', y', \theta'], u_{t_{\textbf{previous}}}, \Delta t, u_{t_{\textbf{now}}}])$$

6. Append the row to the dataset.

Using the above method, 95809 data samples were generated from ds0.

References

[1] Thrun, S., Burgard, W., Fox, D. & Arkin, R. (2005). *Probabilistic robotics*. MIT Press.