Motion and Measurement Models

And Some Things About It

Dalang Felix Sihitshuwam

Robotics and Computer Vision
Innopolis University

November 25, 2022

Outline

Motion Models

Introduction

Robot kinematic models are deterministic. We assume, that controls are definite and update the state exactly.

In reality, that's not the case. Control inputs are exposed to noise, and unmodeled external factors can affect robot state.

Probabilistic Motion Models are designed to cater for these uncertainties. Which is how the world works in reality.

$$p(x_t | u_t, x_{t-1})$$

Introduction

Robot kinematic models are deterministic. We assume, that controls are definite and update the state exactly.
In reality, that's not the case. Control inputs are exposed to noise, and unmodeled external factors can affect robot state.
Probabilistic Motion Models are designed to cater for these uncertainties. Which is how the world works in reality.

$$p(x_t | u_t, x_{t-1})$$

Introduction

Robot kinematic models are deterministic. We assume, that controls are definite and update the state exactly.
In reality, that's not the case. Control inputs are exposed to noise, and unmodeled external factors can affect robot state.
Probabilistic Motion Models are designed to cater for these uncertainties.
Which is how the world works in reality.

$$p(x_t|u_t, x_{t-1})$$

Introduction

Robot kinematic models are deterministic. We assume, that controls are definite and update the state exactly.
In reality, that's not the case. Control inputs are exposed to noise, and unmodeled external factors can affect robot state.
Probabilistic Motion Models are designed to cater for these uncertainties. Which is how the world works in reality.

$$p(x_t|u_t, x_{t-1})$$

Model Types

There are two basic model types

1. Velocity Motion Model
2. Odometry Motion Model

Velocity Motion Model

Assumes that the robot is actuated by independent translational (v) and rotational velocities (ω)

Velocity Motion Model

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} -\frac{v}{\omega} sin\theta + \frac{v}{\omega} sin(\theta + \omega\delta t) \\ \frac{v}{\omega} cos\theta - \frac{v}{\omega} cos(\theta + \omega\delta t) \\ \omega\delta t + \gamma\delta t \end{bmatrix}$$

Odometry Motion Model

Odometry is the use of data from motion sensors to estimate change in position over time.

Odometry is commonly obtained by integrating wheel encoders information

Interestingly, practical experience suggests that odometry, while still erroneous, is usually more accurate than velocity.

Odometry Motion Model

Odometry is the use of data from motion sensors to estimate change in position over time.

Odometry is commonly obtained by integrating wheel encoders information

Interestingly, practical experience suggests that odometry, while still erroneous, is usually more accurate than velocity.

Odometry Motion Model

Odometry is the use of data from motion sensors to estimate change in position over time.
Odometry is commonly obtained by integrating wheel encoders information
Interestingly, practical experience suggests that odometry, while still erroneous, is usually more accurate than velocity.

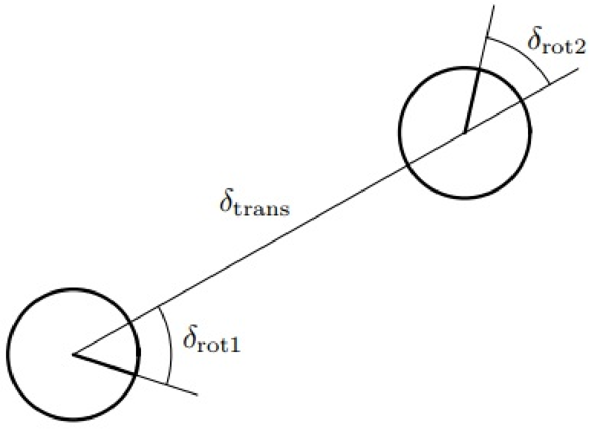
Odometry Motion Model

However, odometry is only available in retrospect, after the robot moved.
This poses no problem for filter algorithms, but makes this information unusable for accurate motion planning and control.

Odometry Motion Model

However, odometry is only available in retrospect, after the robot moved. This poses no problem for filter algorithms, but makes this information unusable for accurate motion planning and control.

Odometry Motion Model



We can get to any pose by executing 3 operations (rotate, translate and rotate, like the basic steps of the dubins path)

Odometry Motion Model

The odometry reports back to us a related advance from $\bar{x}_{t-1} = (\bar{x}\bar{y}\bar{\theta})$ to $\bar{x}_t = (\bar{x}'\bar{y}'\bar{\theta}')$

The bars indicate that these are odometry measurements embedded in the robot-internal coordinate whose relation to the world coordinate is unknown.

The key concept for utilizing this information in state estimation is the fact that the relative difference between $\bar{x}_{t-1}$ and $\bar{x}_t$, under a specific definition of the term "difference," is a good estimator for the difference of the true poses x_{t-1} and x_t

Odometry Motion Model

The motion information u_t is then given by the pair as shown below

$$u_t = \begin{pmatrix} \bar{x}_{t-1} \\ \bar{x}_t \end{pmatrix}$$

Relative odometry can then be extracted by transforming this information in a sequence of three steps (rotation, translation and rotation)

Odometry Motion Model

Thus there exist a unique set of three parameters to construct the relative motion between any pair of poses $\bar{s}$ and $\bar{s}'$

$$\begin{pmatrix} \delta_{rot1} \\ \delta_{trans} \\ \delta_{rot2} \end{pmatrix}$$

The motion model assumes that these parameters are corrupted by noise.

Odometry Motion Model

Given odometry reading $u_t = (\bar{x}_{t-1}\bar{x}_t)^T$ and *prior* x_{t-1}, We calculate the predicted next state as follows

$$\delta_{rot1} = atan2(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$$

$$\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$$

$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$$

Odometry Motion Model

Finally we have the motion model

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \delta_{trans} cos(\theta + \delta_{rot1}) \\ \delta_{trans} sin(\theta + \delta_{rot1}) \\ \theta + \delta_{rot1} + \delta_{rot2} \end{bmatrix}$$

Note that these parameters are affected by noise and the *true* values have the noise subtracted.

Measurement Models

Measurement Model

Measurement models describe the formation process by which sensor measurements are generated in the physical world.

Measurement models (in the realm of Probabilistic Robots) account for the inherent uncertainty in the robot's sensors.

The measurement model is defined as a conditional probability distribution $p(z_t|x_t, m)$, where x_t is the robot pose, z_t is the measurement at time t, and m is the map of the environment.

The principle can be applied to any kind of sensor

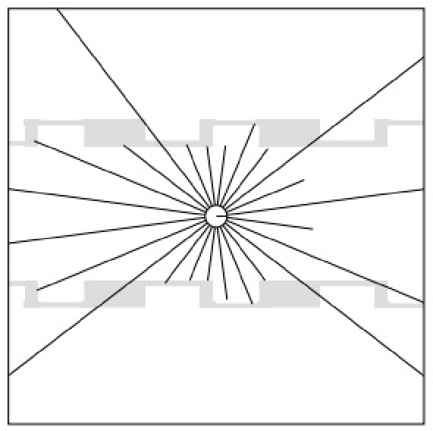
Measurement Model

Measurement models describe the formation process by which sensor measurements are generated in the physical world.
Measurement models (in the realm of Probabilistic Robots) account for the inherent uncertainty in the robot's sensors.
The measurement model is defined as a conditional probability distribution $p(z_t|x_t, m)$, where x_t is the robot pose, z_t is the measurement at time t, and m is the map of the environment.
The principle can be applied to any kind of sensor

Measurement Model

Measurement models describe the formation process by which sensor measurements are generated in the physical world.
Measurement models (in the realm of Probabilistic Robots) account for the inherent uncertainty in the robot's sensors.
The measurement model is defined as a conditional probability distribution $p(z_t|x_t, m)$, where x_t is the robot pose, z_t is the measurement at time t, and m is the map of the environment.
The principle can be applied to any kind of sensor

Measurement Model



Try to obtain a probability $p(z_t|x_t)$ rather than using a deterministic $z_t = f(x_t)$

Measurement Model

Some sensor models utilize raw sensor measurements while an alternative approach is to extract features from the measurements. Feature extractors extract a small number of features from high dimensional sensor measurements thereby enormously reduction of computational complexity. In robotics, it is also common to define places as features, such as hallways, vertical columns, intersections, etc.

Measurement Model

Some sensor models utilize raw sensor measurements while an alternative approach is to extract features from the measurements. Feature extractors extract a small number of features from high dimensional sensor measurements thereby enormously reduction of computational complexity. In robotics, it is also common to define places as features, such as hallways, vertical columns, intersections, etc.

Measurement Model

In many robotics applications, features correspond to distinct objects in the physical world such as door posts or window stills in indoor environments; outdoors they may correspond to tree trunks or corners of buildings. It is common to call those physical objects landmarks, to indicate that they are being used for robot navigation.

It is assumed that the sensor can measure the range and bearing of the landmark relative to the robot's local coordinate frame.

In addition, the feature extractor may generate a *signature* which could have a numerical value indicating colour, landmark type, height, etc.

Measurement Model

In many robotics applications, features correspond to distinct objects in the physical world such as door posts or window stills in indoor environments; outdoors they may correspond to tree trunks or corners of buildings. It is common to call those physical objects landmarks, to indicate that they are being used for robot navigation.
It is assumed that the sensor can measure the range and bearing of the landmark relative to the robot's local coordinate frame.
In addition, the feature extractor may generate a *signature* which could have a numerical value indicating colour, landmark type, height, etc.

Measurement Model

In many robotics applications, features correspond to distinct objects in the physical world such as door posts or window stills in indoor environments; outdoors they may correspond to tree trunks or corners of buildings. It is common to call those physical objects landmarks, to indicate that they are being used for robot navigation.
It is assumed that the sensor can measure the range and bearing of the landmark relative to the robot's local coordinate frame.
In addition, the feature extractor may generate a *signature* which could have a numerical value indicating colour, landmark type, height, etc.

Measurement Model

$$f(z_t) = \{f_t^1, f_t^2, \ldots\} = \{\begin{pmatrix} r_t^1 \\ \phi_t^1 \\ s_t^1 \end{pmatrix}, \begin{pmatrix} r_t^2 \\ \phi_t^2 \\ s_t^2 \end{pmatrix}, \ldots\}$$

Measurement Model

Given the location of the j^{th} landmark corresponding to the i^{th} feature, denoted as $m_{j,x}$ and $m_{j,y}$ which is it's coordinate in the global coordinate frame of the map and the robot pose $x_t = (x, y, \theta)^T$,

$$\begin{pmatrix} r_t^i \\ \phi_t^i \\ s_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ atan2((m_{j,y} - y), (m_{j,x} - x)) - \theta \\ s_j \end{pmatrix}$$

Measurement Model

Known Correspondence

To implement this measurement model, we need to define a variable that establishes correspondence between the feature f_t^i and the (known) landmark m_j in the map.

It is denoted as c_t^i. Correspondences can be generated however you want.

Acknowledgments

Appreciation goes to Dr. Geesara and all my noble colleagues.

References

1. Thrun, S., Burgard, W. & Fox, D. *Probabilistic Robotics.* 1st ed. (MIT Press, 2006).
2. Wikipedia. *Odometry.* `https://en.wikipedia.org/wiki/Odometry`.

The End