Scripsie: Motion Modelling and Prediction

Aidan Landbserg

February 18, 2015

Dr. Van Daalen,

This section, for the moment, provides initially, a general background to explain the content that follows. It will very likely be integrated into another section in the final report, while other information will still be added to this particular section. The main goal is to set up the necessary theoretical basis in order to describe the motion model for you to examine. Also, I have purposefully omitted all relevant references as I am yet to complete all the literature I have obtained. I am though, well aware of which sections of this write up is to be referenced and will proceed to do so at a later stage.

1 Probabilistic Map Construction

The ultimate goal of the approach presented here, is to obtain a probabilistic three dimensional map of features, representing at every time instance both the state estimates of the of the camera as well as that of every feature observed. These features of interest are more commonly referred to as *landmarks* and the aforementioned terms will, from hereon in, be used synonymously . Most importantly though, the map is to contain the *uncertainty* associated with each of the aforementioned estimates.
The process regarding the construction of this map of features is to be implemented through the use of an (Extended) Kalman Filter. The map initially, completely void of any landmarks, is recursively updated according to the subsequent fusions of both predictions and measurements presented to the Kalman Filter. As new (potentially interesting) features are observed, the state estimates of both the camera as well as the landmarks are both updated - augmenting the state vector with additional features (if indeed they are observed) while deleting any landmarks that are of no longer of interest. The purpose of the map, ultimately, is to present real-time, repeatable localisation of the robot within the map. In order to obtain the best possible result, the algorithm should strive to obtain a sparse set of higher-quality landmarks rather than a dense set of ordinary landmarks within the environment.

1.1 State Representation

All relevant state estimates are embedded within the state vector $\hat{\mathbf{x}}_t$ which is comprised of two parts, the camera state $\hat{\mathbf{x}}_v$ and the feature estimates $\hat{\mathbf{y}}$ respectively. The camera state provides the estimate for its pose at each tilmestep and the landmark estimates provide the landmark's feature description as well its estimated position within the map.
Mathematically, the probabilistic map can be represented through the state vector $\hat{\mathbf{x}}_t$ and a covariance matrix P. $\hat{\mathbf{x}}_t$, as previously mentioned, is a a single column vector containing the estimates of the camera as well as the landmarks, and P is a square matrix. These quantities can be mathematically shown as follows:

$$\hat{\mathbf{x}}_t = \begin{pmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \\ \vdots \\ \hat{\mathbf{y}}_N \end{pmatrix}, \quad P_{NN} = \begin{bmatrix} P_{x,x} & P_{x,y_1} & P_{x,y_2} & \cdots & P_{x,y_N} \\ P_{y_1,x} & P_{y_1,y_1} & P_{y_1,y_2} & \cdots & P_{y_1,y_N} \\ P_{y_2,x} & P_{y_2,y_1} & P_{y_2,y_2} & \cdots & P_{y_2,y_N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{y_N,x} & P_{y_N,y_1} & P_{y_N,y_2} & \cdots & P_{y_N,y_N} \end{bmatrix}. \tag{1.1}$$

These quantities then, allow us to approximate the uncertainty regarding the generated feature map as a N-dimensional single multi-variate Gaussian distribution, where N, as stated above, is the total number of state estimates within the state vector.

Before continuing, it is important to consider and understand the notation used in the sections that follow. Two separate coordinate systems are to be considered, namely the *fixed* inertial reference fame system W and the cameras free coordinate frame system C. System variables defined within either of the aforementioned coordinate systems, are from here on in, to be designated a superscript to establish in which coordinate system it may be relevant (e.g. x^W). Vectors will be printed in bold and non-italics to better distinguish them from scalers. An example can be shown regarding the variable x: $\mathbf{x}$ denotes a vector while its scalar counterpart would be represented as x.

1.1.1 Camera Position State Representation

The following concept describes a suitable method to represent all relevant information regarding the cameras position and orientation in a three dimensional (3D) space. According to most implementations of robot localisation, there exists no concern to contrast between the concepts of a camera state $\hat{\mathbf{x}}_v$ and a camera position state $\mathbf{x}_p$: it is therefore important to note that a position state - containing the required information regarding a robots position - is merely an element of the camera state vector. The state camera vector - comprising of 10 individual states - is mathematically described as follows:

$$\hat{\mathbf{x}}_v = \begin{pmatrix} \mathbf{r}^W \\ \mathbf{q}^{WC} \\ \mathbf{V}^W \end{pmatrix}. \tag{1.2}$$

where $\mathbf{r}^W = (x\ y\ z)^T$ indicates the 3D cartesian position of the camera, $\mathbf{q}^{WC}$ the unit orientation *quarternion* -to be mathematically defined and described in the appendix- indicating the camera orientation *relative to the inertial reference frame* W and $\mathbf{V}^W$ indicating the linear velocities of the camera relative to the inertial reference frame W Often, the modelling of dynamic systems require that additional parameters - separate to those describing the position and orientation of the robot - be included in the state vector along with the position state vector. This is illustrated in the description above, with the position state vector $\mathbf{x}_p$ comprising of the 3D position vector, $\mathbf{r}^W$ and the unit orientation *quarternion*, $\mathbf{q}^{WC}$. The linear velocity $\mathbf{V}^W$, is the additional information required for system modelling.

1.1.2 Cartesian Feature Representation

As previously discussed, the aim is to describe a set of high-quality, well defined landmarks within the map. The map itself is to contain a 3D position of *each* observed landmark as well as a combined uncertainty. The feature estimates $\hat{\mathbf{y}}$ - comprising of N landmarks - is mathematically described through three individual cartesian coordinates - x, y and z respectively:

$$\hat{\mathbf{y}}_n = (x_n \quad y_n \quad z_n)^T. \tag{1.3}$$

where n represents a specific single landmark.

With reference to the theory on image processing, it can be discussed that the depth of a given landmark (in this case the z-coordinate) cannot be immediately determined, but rather approximated via triangulation given the landmark is observed over a sequence of (minimally) two known camera positions. The x and y measurements however, can be immediately determined from the image plane.

1.1.3 Control Input

1.2 State Transition and Motion Modelling

The solution to this specific implementation of the Simultaneous Localisation and Mapping (SLAM) problem, takes the following probabilistic form:

$$P\big(\hat{\mathbf{x}}_k, \hat{\mathbf{y}} \mid \mathbf{z}_{0:k}, \mathbf{u}_{0:k}, \mathbf{x}_0\big)[k] \tag{1.4}$$

with the aforementioned distribution described at each time instance of k. A brief description would yield that the distribution describes a joint density at time instance k, of the robot state as well as the landmark locations given all of the previously recorded observations and control inputs. This descriptions allows for the implementation of a recursive algorithm, namely, a discrete Kalman Filter. In order for a Kalman Filter to be successfully implemented, a state transition (motion) model as well as an observation model is required to individually describe the effects of the control input as well as the observations respectively. The Kalman Filter (as well as its variants) are more implicitly described later in the report.
It is important to note that the Kalman Filter estimates the state of a continuous- or discrete-time process that is described by a differential (continuous) or difference (discrete) equation. The Kalman Filter then updates the state estimate according to the measurements it obtains.

1.2.1 Motion Modelling

As previously discussed, the Kalman Filter requires a motion model in order to estimate the current state of the system. In short, the motion model describes the transition from the previous state to the following state with regard to the robots kinematic motion as well as the control inputs. The motion model in this particular instance can be described through a differential equation of the following form:

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k] + \mathbf{w}[k] \tag{1.5}$$

where $\mathbf{A}$ describes the manner in which state evolves from time k to $k+1$ without the influence of noise and controls, $\mathbf{B}$ describes how the control vector $\mathbf{u}[k]$ evolves from time step k to $k+1$ and $\mathbf{w}[k]$ is a Gaussian random variable representing the process noise.

The aforementioned model is then discretised using the forward method of discretisation: