POSE estimation through Rao-Blackwellized Particle filter.

Ruben Fiszel

June 5, 2017

Introduction

We will introduce here several filter for POSE estimation for highly dynamic objects and in particular drones. The order is from the most conceptually simple, to the most complex. The Rao-Blackwellized Particle filter can be viewed as a superset of many other filters, where those filters are just instances of a particle filter with only one particle.

Notes on notation and conventions

The referential by default is the fixed world frame.

- $\mathbf{x}$ designates a vector
- x_t is the random variable of x at time t
- $x_{t1:t2}$ is the product of the random variable of x between t1 included and t2 included
- $x^{(i)}$ designates the random variable x of the arbitrary particle i

POSE

POSE is the task of determining the position and orientation of an object through time. It is a subroutine of SLAM (Software Localization And Mapping). We can formelize the problem as:

At each timestep, find the best expectation of a function of the hidden variable state (position and orientation), from their initial distribution and observable random variables (such as sensor measurements).

- The state $\mathbf{x}$
- The function $g(\mathbf{x})$ such that $g(\mathbf{x}_t) = (\mathbf{p}_t, \mathbf{q}_t)$ where p is the position, q is the attitude as a quaternion.

- The observable variable $\mathbf{y}$ composed of the sensor measurements $\mathbf{z}$ and the control input $\mathbf{u}$

The algorithm inputs are:

- The distribution of initial position p_0 and orientation q_0
- control inputs u_t (the command sent to the flight controller)
- sensor measurements z_t coming from different sensors with different sampling rate
- information about the sensors (sensor measurements biases and matrix of covariance)

Quaternions

Quaternions are extensions of complex numbers but with 3 imaginary parts. They can be used to represent orientation, also referred to as attitude. Quaternions algebra make rotation composition simple and quaternions avoid the issue of gimbal lock. In all filters presented, they will be used to represent the attitude.

Filtering and smoothing

Smoothing is the statistical task of finding the expectation of the state variable from multiple observation variable ahead.

$$\mathbb{E}[g(\mathbf{x}_{0:t})|\mathbf{y}_{1:t+k}]$$

Which expand to,

$$\mathbb{E}[(\mathbf{p}_{0:t}, \mathbf{q}_{0:t})|(\mathbf{z}_{1:t+k}, \mathbf{u}_{1:t+k})]$$

k is a contant and the first observation is y_1

Filtering is a kind of smoothing where you only have at disposal the current observation variable ($k = 0$)

Kalman Filter

Kalman filters are optimal linear filters.

TODO

Linearity

Kalman filters are non optimal for our problem because our state has some non-linear components (attitude).

Indeed, rotations belong to $SO(3)$. It can be shown intuitively that they do not belong to a vector space because the sum of two unit quaternions is not a unit quaternion (not closed under addition). **TODO**

Extended Kalman Filters

EKF are linearized Kalman filters of the first order.

TODO

Unscented Kalman Filters

TODO

Particle Filter

Particle filters are monte carlo methods which in their general form ... **TODO**

TODO

Resampling

When the number of effective particles is too low ($N/10$), we apply systematic resampling

Rao-Blackwellized Particle Filter

Introduction

Compared to a plain PF, RPBF leverage the linearity of some components of the state by assuming our model gaussian conditionned on a latent variable: Given the attitude q_t, our model is linear. This is where RPBF shines: We use particle filtering to estimate our latent variable, the attitude, and we use the optimal kalman filter to estimate the state variable.

We introduce the latent variable $\boldsymbol{\theta}$

The latent variable $\boldsymbol{\theta}$ has for sole component the attitude:

$$\boldsymbol{\theta} = (\mathbf{q})$$

q_t is estimated from the product of the attitude of all particles $\theta^{(\mathbf{i})} = \mathbf{q}_t^{(i)}$ as the "average" quaternion $\mathbf{q}_t = avgQuat(\mathbf{q}_t^n)$. x^n designates the product of all n arbitrary particle. The average quaternion is not simply the average of its components ... **TODO**

We use importance sampling ... **TODO**

The weight definition is:

$$w_t^{(i)} = \frac{p(\boldsymbol{\theta}_{0:t}^{(i)} | \mathbf{y}_{1:t})}{\pi(\boldsymbol{\theta}_{0:t}^{(i)} | \mathbf{y}_{1:t})}$$

From the definition, it is proovable that:

$$w_t^{(i)} \propto \frac{p(\mathbf{y}_t | \boldsymbol{\theta}_{0:t-1}^{(i)}, \mathbf{y}_{1:t-1}) p(\boldsymbol{\theta}_t^{(i)} | \boldsymbol{\theta}_{t-1}^{(i)})}{\pi(\boldsymbol{\theta}_t^{(i)} | \boldsymbol{\theta}_{1:t-1}^{(i)}, \mathbf{y}_{1:t})} w_{t-1}^{(i)}$$

We choose the dynamic of the model as the importance distribution:

$$\pi(\boldsymbol{\theta}_t^{(i)} | \boldsymbol{\theta}_{1:t-1}^{(i)}, \mathbf{y}_{1:t}) = p(\boldsymbol{\theta}_t^{(i)} | \boldsymbol{\theta}_{t-1}^{(i)})$$

Hence,

$$w_t^{(i)} \propto p(\mathbf{y}_t | \boldsymbol{\theta}_{0:t-1}^{(i)}, \mathbf{y}_{1:t-1}) w_{t-1}^{(i)}$$

We then sum all $w_t^{(i)}$ to find the normalization constant and retrieve the actual $w_t^{(i)}$

State

$$\mathbf{x}_t = (\mathbf{a}_t, \mathbf{v}_t, \mathbf{p}_t)$$

- **a** acceleration
- **v** velocity
- **p** position

Initial position $\mathbf{p_0}$ at (0, 0, 0)

Observations

$$\mathbf{y}_t = (\mathbf{aA}_t, \boldsymbol{\omega G}_t, \mathbf{pV}_t, \mathbf{qV}_t, tC_t, \boldsymbol{\omega C}_t)$$

Measurements

- $\mathbf{aA}$ acceleration from the accelerometer in the body frame
- $\boldsymbol{\omega G}$ angular velocity from the gyroscope in the body frame
- $\mathbf{pV}$ position from the vicon
- $\mathbf{qV}$ attitude from the vicon

Control Inputs

- tC thrust (as a scalar) in the direction of the attitude from the control input.
- $\boldsymbol{\omega C}$ angular velocity in the body frame from the control input

Observations from the control input are not strictly speaking measurements but input of the state-transition model

Latent variable

$$\mathbf{q}_{t+1}^{(i)} = \mathbf{q}_t^{(i)} * R2Q((\boldsymbol{\omega C}_t + \boldsymbol{\omega C^{\epsilon}}_t) * dt)$$

where $\boldsymbol{\omega C^{\epsilon}}_t$ represents the error from the control input and is sampled from $\boldsymbol{\omega C^{\epsilon}}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\boldsymbol{\omega C}_t})$

We introduce some helper functions.

- $B2F(\mathbf{q}, \mathbf{v})$ is the body to fixed vector rotation. It transforms vector in the body frame to the fixed world frame. It takes as parameter the attitude q and the vector v to be rotated.
- $F2B(\mathbf{q}, \mathbf{v})$ is its inverse function (from fixed to body).
- $T2A(t)$ is the scaling from thrust to acceleration (by dividing by the weight of the drone: $\mathbf{F} = m\mathbf{a} \Rightarrow \mathbf{a} = \mathbf{F}/m$) and then multiplying by a unit vector $(0, 0, 1)$

dt is the lapse of time between t and the next tick (t+1)

Model dynamics

$\mathbf{w}_t$ is our process noise (wind, etc ...)

- $\mathbf{a}(t+1) = B2F(\mathbf{q}(t+1), T2A(tC(t+1))) + \mathbf{w}_{\mathbf{a}_t}$
- $\mathbf{v}(t+1) = \mathbf{v}(t) + \mathbf{a}(t) * dt + \mathbf{w}_{\mathbf{v}_t}$

- $\mathbf{p}(t+1) = \mathbf{p}(t) + \mathbf{v}(t) * dt + \mathbf{w}_{\mathbf{p}_t}$

Note that $\mathbf{q}(t+1)$ is known because the model is conditionned under $\boldsymbol{\theta}_{t+1}^{(i)}$.

The model dynamic define the state-transition matrix $\mathbf{F}_t(\boldsymbol{\theta}_t^{(i)})$, the control-input matrix $\mathbf{B}_t(\boldsymbol{\theta}_t^{(i)})$ and the process noise $\mathbf{w}_t(\boldsymbol{\theta}_t^{(i)})$ for the Kalman filter.

TODO: write the 3 matrices explicitely

Kalman prediction

$$\mathbf{m}_t^{-(i)} = \mathbf{F}_t(\boldsymbol{\theta}_t^{(i)})\mathbf{m}_{t-1}^{(i)} + \mathbf{B}_t(\boldsymbol{\theta}_t^{(i)})\mathbf{u}_t$$
$$\mathbf{P}_t^{-(i)} = \mathbf{F}_t(\boldsymbol{\theta}_t^{(i)})\mathbf{P}_{t-1}^{-(i)}(\mathbf{F}_t(\boldsymbol{\theta}_t^{(i)}))^T + \mathbf{w}_t(\boldsymbol{\theta}_t^{(i)})$$

Measurements model

The measurement model defines how to compute $p(\mathbf{y}_t|\boldsymbol{\theta}(i)_{0:t-1}, \mathbf{y}_{1:t-1})$

- Vicon:
  1. $\mathbf{p}(t) = \mathbf{pV}(t) + \mathbf{pV}_t^\epsilon$ where $\mathbf{pV}_t^\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\mathbf{pV}_t})$
  2. $\mathbf{q}(t) = \mathbf{qV}(t) + \mathbf{qV}_t^\epsilon$ where $\mathbf{qV}_t^\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\mathbf{qV}_t})$
- Gyroscope:
  3. $\mathbf{q}(t) = \mathbf{q}(t-1) + (\boldsymbol{\omega G}(t) + \boldsymbol{\omega G}_t^\epsilon) * dt$ where $\boldsymbol{\omega G^\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\boldsymbol{\omega G}_t})$
- Accelerometer:
  4. $\mathbf{a}(t) = B2F(\mathbf{q}(t), aA(t) + \mathbf{aA}_t^\epsilon)$ where $\mathbf{aA}_t^\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\mathbf{aA}_t})$
  5. $\mathbf{g}^f(t) = B2F(\mathbf{q}(t), \mathbf{aA}(t) + \mathbf{aA}_t^\epsilon) - \mathbf{a}(t)$ where $\mathbf{aA}_t^\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\mathbf{aA}_t})$

(1, 2, 4) define the observation matrix $\mathbf{H}_t(\boldsymbol{\theta}_t^{(i)})$ and the observation noise $\mathbf{v}_t(\boldsymbol{\theta}_t^{(i)})$ for the Kalman filter.

TODO: write the 3 matrices explicitely

$$p(\mathbf{y}_t|\boldsymbol{\theta}_{0:t-1}^{(i)}, \mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{z}_t^{(1,2,4)}; \mathbf{H}_t(\boldsymbol{\theta}_t^{(i)})\mathbf{m}_t^{(i)}, \mathbf{H}_t(\boldsymbol{\theta}_t^{(i)})\mathbf{P}_t^{-(i)}(\mathbf{H}_t(\boldsymbol{\theta}_t^{(i)}))^T + \mathbf{v}_t(\boldsymbol{\theta}_t^{(i)}))$$

$\mathbf{z}^{(1,2,4)}$ means component 1, 2 and 4 of $\mathbf{z}$.

Asynchronous measurements

Our measurements have different sampling rate so instead of doing full kalman update, we only apply a partial kalman update corresponding to the current type of measurement $\mathbf{z}_t$

Other sources or reweighting

(3 and 5) defines two other weight updates.

$$p(\mathbf{y}_t|\boldsymbol{\theta}_{0:t-1}^{(i)}, \mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{z}_t^{(3)}; (\mathbf{q}_t^{(i)} - \mathbf{q}_{t-1}^{(i)})/dt, \mathbf{R}_{\boldsymbol{\omega G}_t})$$

$$p(\mathbf{y}_t|\boldsymbol{\theta}_{0:t-1}^{(i)}, \mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{z}_t^{(5)}; F2B(\mathbf{q}_t^{(i)}, \mathbf{g}) + \mathbf{a}_t^{(i)}, \mathbf{R}_{\mathbf{aA}_t} + \mathbf{Pa}_t^{-(i)})$$

TODO: Check that matrix of covariance is correct for 5. Found covariance as covariance of sum of normal but seems too simple.

where $\mathbf{Pa}_t^{-(i)}$ is the variance of $\mathbf{a}$ in $\mathbf{P}_t^{-(i)}$ and $\mathbf{g}$ is the gravity vector.

Kalman update

TODO: plain kalman update matrix operations.

Algorithm summary

1. Initiate N particles with p_0, q_0 $simp(q_0)$, P_0 and $w = 1/N$
2. While new sensor measurements $(\mathbf{z}_t, \mathbf{u}_t)$

- foreach N particles (i):
    1. sample new latent variable $\boldsymbol{\theta_t}$ from $\mathbf{u}_t$
    2. Depending on the type of measurement:
        - **Gyroscope**:

          $$p(\mathbf{y}_t|\boldsymbol{\theta}(i)_{0:t-1}, \mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{z}_t^{(3)}; (\mathbf{q}_t^{(i)} - \mathbf{q}_{t-1}^{(i)})/dt, \mathbf{R}_{\boldsymbol{\omega G}_t})$$

        - **Vicon**:

          $$p(\mathbf{y}_t|\boldsymbol{\theta}(i)_{0:t-1}, \mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{z}_t^{(1,2,4)}; \mathbf{H}_t(\boldsymbol{\theta}_t^{(i)})\mathbf{m}_t^{(i)}, \mathbf{H}_t(\boldsymbol{\theta}_t^{(i)})\mathbf{P}_t^{-(i)}(\mathbf{H}_t(\boldsymbol{\theta}_t^{(i)}))^T + \mathbf{v}_t(\boldsymbol{\theta}_t^{(i)}))$$

          and update particle state with a partial/full kalman prediction and update
        - **Accelerometer**:

          $$p(\mathbf{y}_t|\boldsymbol{\theta}(i)_{0:t-1}, \mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{z}_t^{(1,2,4)}; \mathbf{H}_t(\boldsymbol{\theta}_t^{(i)})\mathbf{m}_t^{(i)}, \mathbf{H}_t(\boldsymbol{\theta}_t^{(i)})\mathbf{P}_t^{-(i)}(\mathbf{H}_t(\boldsymbol{\theta}_t^{(i)}))^T + \mathbf{v}_t(\boldsymbol{\theta}_t^{(i)}))$$

          from the acceleration information and update particle state with a partial/full kalman prediction and update, and then a new

          $$p(\mathbf{y}_t|\boldsymbol{\theta}(i)_{0:t-1}, \mathbf{y}_{1:t-1}) = \mathcal{N}(\mathbf{z}_t^{(5)}; F2B(\mathbf{q}_t^{(i)}, \mathbf{g}) + \mathbf{a}_t^{(i)}, \mathbf{R}_{\mathbf{aA}_t} + \mathbf{Pa}_t^{-(i)}) p(\mathbf{y}_t|\boldsymbol{\theta}(i)_{0:t-1}, \mathbf{y}_{1:t-1})$$

          from the orientation information

3. Update $w_t^{(i)}$ as $w_t^{(i)} = p(\mathbf{y}_t|\boldsymbol{\theta}(i)_{0:t-1}, \mathbf{y}_{1:t-1})w_{t-1}^{(i)}$

- Compute $\mathbf{p}_t$ and $\mathbf{q}_t$ as the expectation from the distribution approximated by the N particles.
- Resample if the number of effective particle is too low