POSE estimation through Rao-Blackwellized Particle filter.

Ruben Fiszel

June 3, 2017

Introduction

The use of the RPBF is justified 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).

Compared to a plain PF, RPBF leverage the linearity of some components of the state by making our model linear conditionned on a latent variables.

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

RPBF

The aim of our filter is POSE. As such, we are interested in

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

(There is no observation of the initial position)

Where p is the position, q is the attitude as a quaternion.

Optimal filters exist for linear models (Kalman filters). Unfortunately, as stated previously, the attitude component of our model is non-linear. However, given the attitude q, we can make the assumption that our model is conditionally gaussian. 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 separate our variables in 3 kinds

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

The state is what we are estimating, the measurements and the control inputs are the data we are estimating them from.

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

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

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

- $\mathbf{a}$ acceleration
- $\mathbf{v}$ velocity
- $\mathbf{p}$ 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 $p(\mathbf{y}_t|\boldsymbol{\theta}(i)_{0:t-1}, \mathbf{y}_{1:t-1})w_{t-1}^{(i)}$

- 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.

Resampling

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

Algorithm summary

TODO

POSE

At each timestep, we get $p(t)$ as the average p from the state of all particles and $q(t)$ as the “average” quaternion (as defined previously) from the latent variable of all particles.