CS294-40 Learning for Robotics and Control Lecture 13 - 10/9/2008

Helicopter Dynamics Modeling and Kalman Filtering

Lecturer: Pieter Abbeel *Scribe: Andrew Wan*

1 Lecture outline

- Helicopter dynamics modeling
- Kalman filtering

2 Helicopter dynamics modeling

The helicopter state is: $(n, e, d, \dot{n}, \dot{e}, \dot{d}, qx, qy, qz, qw, p, q, r)$.

Its inputs are: $(u_{\text{aileron}}, u_{\text{elevator}}, u_{\text{rudder}}, u_{\text{collective}})$

where

- (n, e, d) parametize the coordinates of the helicopter in the earth-fixed North-East-Down frame.
- $(\dot{n}, \dot{e}, \dot{d})$ describe its velocity.
- (qx, qy, qz, qw) is a quaternion parametizing the orientation of the helicopter.
- (p, q, r) describes its rotational velocity.

Unknown forces and torques:
$$\begin{array}{ll} F_n(s,u) & T_x(s,u) \\ F_e(s,u) & T_y(s,u) \\ F_d(s,u) & T_z(s,u) \end{array}$$

We predict forces in the helicopter frame F_x, F_y, F_z as follows:

$$\begin{aligned} F_x &= C_u u + G_x & T_x &= T_x^0 + C_p p + C_{\text{aileron}} u_{\text{aileron}} \\ F_y &= F_y^0 + C_v v + G_y & T_y &= T_y^0 + C_q q + C_{\text{elevator}} u_{\text{elevator}} \\ F_z &= F_z^0 + C_w w + C_{\text{coll}} u_{\text{coll}} + G_z & T_z &= T_z^0 + C_r r + C_{\text{rudder}} u_{\text{rudder}} \end{aligned}$$

where (u, v, w) is the velocity expressed in the helicopter frame (u: forward, v: sideways to the righ, w: downward), and (G_x, G_y, G_z) is gravity in the helicopter frame.

The parameters characterizing a specific helicopter are C_u, C_v, C_w, $C_{collective}$, C_p, C_q, C_r, $C_{aileron}$, $C_{elevator}$, C_{rudder}, T_x^0, T_y^0, T_z^0, F_y^0, F_z^0.

(We assume the helicopter's mass and inertia are known. If not, we can estimate all parameters up to scale factors relating to mass and inertia.)

We find these values by collecting flight data:

$$\{(s_t, u_t)\}_{t=0}^T$$

We can find the values we need with linear regression (least squares), e.g., for the roll axis we would solve the following least squares problem:

$$\begin{bmatrix} T_x^{(0)} \\ T_x^{(1)} \\ \vdots \\ T_x^{(T)} \end{bmatrix} = \begin{bmatrix} 1 & p^{(0)} & u_{ail}^{(0)} \\ 1 & p^{(1)} & u_{ail}^{(1)} \\ \vdots & \vdots & \vdots \\ 1 & p^{(T)} & u_{ail}^{(T)} \end{bmatrix} \begin{bmatrix} T_x^0 \\ C_p \\ C_{\text{aileron}} \end{bmatrix}$$

2.1 Locally weighted models

In general, this linear dependence is not rich enough. Instead, we use a locally weighted model.

Previously, we solved a least squares problem of the following form:

$$\min_{\theta} \sum_i (y^{(i)} - \theta^\top x^{(i)})^2$$

Instead, for a given query point $\bar{x}$, we perform a weighted least squares, with weights $w^{(i)}(\bar{x})$ specific to the query point $\bar{x}$:

$$\min_{\theta(\bar{x})} \sum_i w^{(i)}(\bar{x})(y^{(i)} - (1 - x^{(i)})\theta(\bar{x}))^2 \tag{1}$$

This gives us a solution $\theta(\bar{x})$ specific to the query point $\bar{x}$.

We typically choose a $w^{(i)}$ such that points near $\bar{x}$ are weighted higher, e.g.:

$$w^{(i)}(\bar{x}) = e^{-1/2 \frac{(x^{(i)} - \bar{x})^2}{\delta^2}} \tag{2}$$

Choice of δ matters a lot in practice. For more details, see *Locally Weighted Linear Regression* (Atkeson, Moore, and Schaal).

In general, we have a multi-dimensional problem, and hence we have to choose a covariance matrix (rather than just a scalar δ).

This covariance matrix can be optimized through, e.g., leave one out cross validation (possibly a leave one out validation local to the query point).

3 State estimation

Generally, we need to keep track of the belief state:

$$\begin{aligned} \text{time update: } & P_{t|t}(x_t) \to P_{t+1|t}(x_{t+1}) \\ \text{measurement update: } & P_{t+1|t}(x_{t+1}) \to P_{t+1|t+1}(x_{t+1}) \end{aligned}$$

In the POMDP setting, we kept track of $P_{t|t-1}, P_{t|t}$ in a discrete state space with relatively small $|S|$. Now we consider continuous state spaces, particularly linear systems and observations with Gaussian noises:

$$\begin{aligned} \text{dynamics model: } & x_{t+1} = Ax_t + Bu_t + w_t, \ where \ w_t \sim N(0, \Sigma_w) \\ \text{measurement model: } & y_t = Cx_t + v_t, \ where \ v_t \sim N(0, \Sigma_v) \\ \text{initial state distribution: } & x_0 \sim N(\hat{x}_{0|-1}, P_{0|-1}) \end{aligned}$$

$x \sim N(\mu, \Sigma)$ means x is a Gaussian random variable with mean μ and covariance Σ. This means x has the following probability density:

$$P(x) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-1/2 (x-\mu)^\top \Sigma^{-1} (x-\mu)}.$$

Properties of Gaussians:

- $\mu = Ex$
- $\Sigma = \mathbb{E}\left[(x - \mu)(x - \mu)^\top\right]$
- A linear combination of Gaussian random variables is a Gaussian random variable.

We use the following notation:

$$\hat{x}_{t|t} = \mathbb{E}\left[x_t|y_0 \ldots y_t\right]$$
$$p_{t|t} = \mathbb{E}\left[(x_t - \hat{x}_{t|t})(x_t - \hat{x}_{t|t})^\mathsf{T}|y_0 \ldots y_t\right]$$

As $P(x_t|y_{0:t})$ is Gaussian, it is sufficient to find $\hat{x}_{t|t}$ and $P_{t|t}$ to characterize the probability density at time t conditioned on all measurements up to that t.

3.1 The recursive updates

The Kalman Filter recusively estimates state and covariance (error) at timesteps given measurements up to that time.

$$\begin{aligned}
x_{t+1} &= Ax_t + Bu_t + w_t \\
\hat{x}_{t+1|t} &= \mathbb{E}\left[x_{t+1}|y_{0:t}\right] = \mathbb{E}\left[Ax_t + Bu_t + w_t|y_{0:t}\right] \\
&= A\hat{x}_{t|t} + Bu_t + 0
\end{aligned}$$

$$\begin{aligned}
P_{t+1|t} &= \mathbb{E}\left[(x_{t+1} - \hat{x}_{t+1|t})(x_{t+1} - \hat{x}_{t+1|t})^\mathsf{T}|y_{0:t}\right] \\
&= \mathbb{E}\left[(Ax_t + Bu_t + w_t - A\hat{x}_{t|t} - Bu_t)(Ax_t + Bu_t + w_t - A\hat{x}_{t|t} - Bu_t)^\mathsf{T}|y_{0:t}\right] \\
&= \mathbb{E}\left[A(x_t - \hat{x}_{t|t})(x_t - \hat{x}_{t|t})^\mathsf{T}A^\mathsf{T}|y_{0:t}\right] + 2\mathbb{E}\left[w_t(x_t - \hat{x}_{t|t})^\mathsf{T}A^\mathsf{T}|y_{0:t}\right] + \mathbb{E}\left[w_t wt^\mathsf{T}|y_{0:t}\right] \\
&= A\mathbb{E}\left[(x_t - \hat{x}_{t|t})(x_t - \hat{x}_{t|t})^\mathsf{T}|y_{0:t}\right]A^\mathsf{T} + 0 + \Sigma_w \\
&= AP_{t|t}A^\mathsf{T} + \Sigma_w
\end{aligned}$$

$$\begin{aligned}
y_{t+1} &= Cx_{t+1} + v_{t+1} \\
\hat{y}_{t+1|t} &= \mathbb{E}\left[y_{t+1}|y_{0:t}\right] = C\hat{x}_{t+1|t} \\
&= \mathbb{E}\left[(y_{t+1} - \hat{y}_{t+1|t})(y_{t+1} - \hat{y}_{t+1|t})^\mathsf{T}|y_{0:t}\right] \\
&= \mathbb{E}\left[(Cx_{t+1} + v_{t+1} - C\hat{x}_{t+1|t})(Cx_{t+1} + v_{t+1} - C\hat{x}_{t+1|t})^\mathsf{T}|y_{0:t}\right] \\
&= \mathbb{E}\left[C(x_{t+1} - \hat{x}_{t+1|t})(x_{t+1} - \hat{x}_{t+1|t})^\mathsf{T}C^\mathsf{T}|y_{0:t}\right] + 2\mathbb{E}\left[v_{t+1}(C(x_{t+1} - \hat{x}_{t+1|t}))^\mathsf{T}|y_{0:t}\right] + \mathbb{E}\left[v_{t+1}v_{t+1}^\mathsf{T}|y_{0:t}\right] \\
&= CP_{t+1|t}C^\mathsf{T} + \Sigma_v
\end{aligned}$$

Note:

$$\begin{aligned}
P(y_0, y_1, \ldots, y_T) =& P(y_0)P(y_1|y_0)P(y_2|y_{0:1})\ldots P(y_T|y_{0:T-1}) \\
=& \prod_t \frac{1}{(2\pi)^{d/2}\left|CP_{t+1|t}C^\mathsf{T}\Sigma_v\right|^{1/2}} e^{-1/2(y_{t+1}-\hat{y}_{t+1|t})^\mathsf{T}(CP_{t+1|t}C^\mathsf{T}+\Sigma_v)^{-1}(y_{t+1}-\hat{y}_{t+1|t})} \\
=& \text{ likelihood of the observable sequence } y_{0:T}
\end{aligned}$$

Later, we'll see that we can learn the parameters of the model (A, B, C) and the covariances automatically from the data by maximizing:

$$\max_{A,B,C,\Sigma_u,\Sigma_v} P(y_0, \ldots, y_T)$$

cross covariance is: $\mathbb{E}\left[(y_{t+1} - \hat{y}_{t+1|t})(x_{t+1} - \hat{x}_{t+1|t})^\mathsf{T}|y_{0:t}\right] = CP_{t+1|t}$

This gives us:

$$\begin{pmatrix} x_{t+1|t} \\ y_{t+1|t} \end{pmatrix} \sim N\left(\begin{bmatrix} \hat{x}_{t+1|t} \\ \hat{y}_{t+1|t} \end{bmatrix}, \begin{bmatrix} P_{t+1|t} & P_{t+1|t}C^\intercal \\ CP_{t+1|t} & CP_{t+1|t}C^\intercal + \Sigma_v \end{bmatrix} \right) \tag{3}$$

$$\tag{4}$$

To find the conditional distribution of $x_{t+1}|y_{0:t+1}$ we are left to solve a problem of the following form: We are given a joint Gaussian:

$$P(x, y) = \frac{1}{(2\pi)^{n/2}(2\pi)^{d/2}|\Sigma|^{1/2}} \; e^{-1/2 \begin{pmatrix} x - \mu x \\ y - \mu y \end{pmatrix}^\intercal \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix}^{-1} \begin{pmatrix} x - \mu x \\ y - \mu y \end{pmatrix}} \tag{5}$$

We need to find $P(x|y)$. We can do so as follows:

$$\begin{aligned} P(x|y) = & \frac{1}{(2\pi)^{n/2}\left|\Sigma_{x|y}\right|^{1/2}} e^{-1/2(x-\mu_{x|y})^\intercal \Sigma_{x|y}^{-1}(x-\mu_{x|y})} \\ & \Sigma_{x|y} = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx} \\ & \mu_{x|y} = \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y) \end{aligned}$$

Applying this to our Kalman filter problem gives us:

$$\hat{x}_{t+1|t+1} = \hat{x}_{t+1|t} + \underbrace{P_{t+1|t}C^\intercal(CP_{t+1|t}C^\intercal + \Sigma_v)^{-1}}_{\triangleq K_{t+1}}(y_{t+1} - \hat{y}_{t+1|t}) \tag{6}$$

$$P_{t+1|t+1} = P_{t+1|t} - P_{t+1|t}C^\intercal(CP_{t+1|t}C^\intercal + \Sigma_v)^{-1}CP_{t+1|t} \tag{7}$$

In summary, we estimate state x covariance P:

$$\begin{aligned} \hat{x}_{t+1|t} &= A\hat{x}_{t|t} + B\mu_t \\ P_{t+1|t} &= AP_{t|t}A^\intercal + \Sigma_w \\ \hat{x}_{t+1|t+1} &= \hat{x}_{t+1|t} + K_{t+1}(y_{t+1} - \hat{y}_{t+1|t}) \\ P_{t+1|t+1} &= P_{t+1|t} - P_{t+1|t}C^\intercal(CP_{t+1|t}C^\intercal + \Sigma_v)^{-1}CP_{t+1|t} \end{aligned}$$