Continuous-time Trajectory Representation for Simultaneous Localization and Mapping

Chao Qu

Department of Computer and Information Science
University of Pennsylvania

May 7, 2021

Outline

References

Discrete

1. Sibley (2006). A Sliding Window Filter for SLAM.
2. Dellaert *et al.* (2017). Factor Graphs for Robot Perception.
3. Kummerle (2013). State Estimation and Optimization for Mobile Robot Navigation

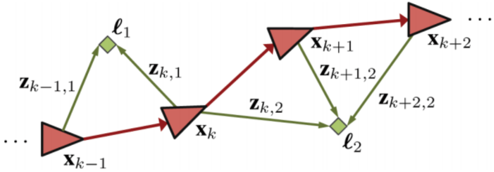
Continuous

1. Furgale *et al.* (2015). Continuous-time batch trajectory estimation using temporal basis functions. IJRR, 34, 1688 - 1710.
2. Tong *et al.* (2013). Gaussian Process Gauss-Newton for non-parametric simultaneous localization and mapping. IJRR, 32, 507 - 525.
3. Zheng *et al.* (2015). Decoupled Representation of the Error and Trajectory Estimates for Efficient Pose Estimation. RSS.

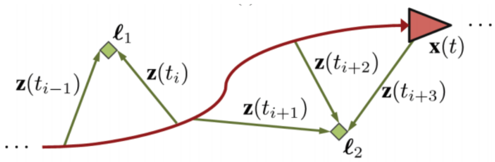
Introduction

Two Types of SLAM

- *Discrete-time SLAM*: A set of poses to represent robot trajectory $x_1, \ldots, x_K$.
- *Continuous-time SLAM*: A function parameterized by time $x(t)$.



(a) The conventional discrete-time formulation, which discretizes robot trajectory into a set of poses.



(b) Continuous-time formulation, where the robot trajectory is modeled as a smooth function of time.

Motivation

Problematic Situations for Discrete-time SLAM

- When a sensor is capturing data at a **high frequency**.
- When a sensor is scanning continuously while **moving**.
- When multiple sensor readings arrive **asynchronously**.
- When a robot is operating for a **prolonged period** of time.

Fundamental Issue

- **Naively** applying the discrete-time formulation will make the problem intractable.
- Continuous-time representation provides a solution to this problem.
- It allows us to use all measurements without inflating the state vector.

Discrete-time SLAM

In SLAM, we want to characterize our knowledge about the **unknown** variables X, given a set of **measurements** Z. The quantity that we are interested in the **posterior** probability $p(X|Z)$. The process of computing this term is called **probabilistic inference**.

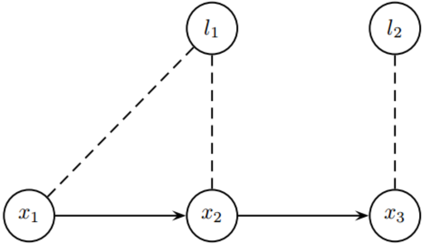
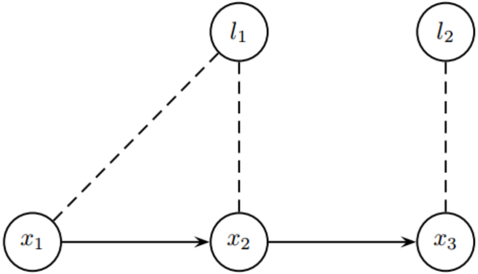
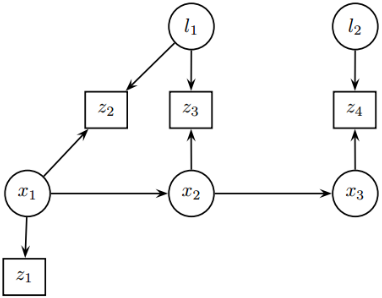


Figure 2: A toy SLAM problem with three robot poses and two landmarks.

Bayesian Networks



(a) Toy SLAM problem.

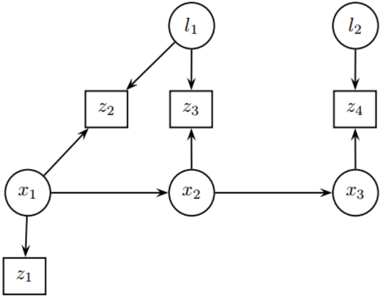


(b) Bayesian network model.

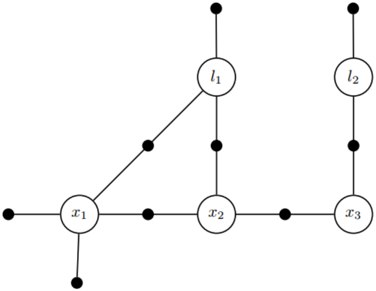
A **Bayes net** defines a joint probability density $p(\Theta)$ over all variables $\Theta = \{\theta_1, \ldots, \theta_n\}$

$$p(X, Z) = p(\Theta) := \prod_j p(\theta_j | \pi_j) \tag{1}$$

Factor Graphs



(a) Bayesian network model.



(b) Factor Graph model.

A factor graph F defines the factorization of a global function $\phi(X)$ as

$$P(X|Z) \propto \phi(X) = \prod_i \phi_i(X_i) \tag{2}$$

Problem Modeling

1. Prior Knowledge

$$x \sim \mathcal{N}(\check{x}, P), \quad p(x) = \frac{1}{\sqrt{|2\pi P|}} \exp\left\{-\frac{1}{2}\|x - \check{x}\|_P^2\right\} \tag{3}$$

2. Motion Model

$$x_k = f(x_{k-1}, u_k) + w_k, \quad w_k \sim \mathcal{N}(0, Q_k) \tag{4}$$

$$p(x_k | x_{k-1}, u_k) = \frac{1}{\sqrt{|2\pi Q_k|}} \exp\left\{-\frac{1}{2}\|f(x_{k-1}, u_k) - x_k\|_{Q_k}^2\right\} \tag{5}$$

3. Measurement model

$$z_i = h(x_i, m) + n_i, \quad n_i \sim \mathcal{N}(0, R_i) \tag{6}$$

$$p(z_i | x_i, m) = \mathcal{N}(z_i; h(x_i, m), R_i) = \frac{1}{\sqrt{|2\pi R_i|}} \exp\left\{-\frac{1}{2}\|h(x_i, m) - z_i\|_{R_i}^2\right\} \tag{7}$$

Problem Modeling (Matrix Form)

For simplicity we usually define a lifted form with $\theta = [x, m]^\top$ and

$$x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_K \end{bmatrix}, \; f(x,u) := f(x_{0:}, u_{1:K}) = \begin{bmatrix} \check{x}_0 \\ f(x_0, u_1) \\ \vdots \\ f(x_{K-1}, u_K) \end{bmatrix}, \; Q = \begin{bmatrix} P_0 & & & \\ & Q_1 & & \\ & & \ddots & \\ & & & Q_K \end{bmatrix} \tag{8}$$

$$z = \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix}, \quad h(\theta) := h(x_{0:K}, m) = \begin{bmatrix} h(x_1, m) \\ \vdots \\ h(x_N, m) \end{bmatrix}, \quad R = \begin{bmatrix} R_1 & & \\ & \ddots & \\ & & R_N \end{bmatrix} \tag{9}$$

With corresponding error functions

$$e_p(\theta) = m - \check{m}, \quad e_u(\theta) = f(x,u) - x, \quad e_z(\theta) = h(\theta) - z \tag{10}$$

MAP Inference

We are interested in the value X that maximizes the posterior probability density $p(X|Z)$

$$X^* = \arg\max_X p(X|Z) = \arg\max_X \frac{p(Z|X)p(X)}{p(Z)} = \arg\max_X p(Z|X)p(X) \tag{11}$$

$$\begin{aligned}
p(X|Z) &= p(x_{0:K}, m|u_{1:K}, z_{1:N}) && \text{(Posterior)} \quad (12)\\
&= \frac{p(x_{0:K}, m|u_{1:K})p(z_{1:N}|x_{0:K}, m, u_{1:K})}{p(z_{1:N}|u_{1:K})} && \text{(Bayes' rule)} \quad (13)\\
&= \frac{p(x_{0:K}, m|u_{1:K})\prod_{i=1}^{N} p(z_i|x_i, m)}{p(z_{1:N})} && \text{(Independence)} \quad (14)\\
&= \frac{p(m)p(x_0)\prod_{k=1}^{K} p(x_k|x_{k-1}, u_k)\prod_{i=1}^{N} p(z_i|x_i, m)}{p(z_{1:N})} && \text{(Markov)} \quad (15)
\end{aligned}$$

Nonlinear Least Squares

Maximizing the posterior probability is equivalent to minimizing its negative log

$$X^* = \arg\max_X p(X|Z) = \arg\min_X -\ln p(X|Z) \tag{16}$$

$$\theta^* = \arg\min_\theta \{-\ln p(x_{0:K}, m|u_{1:K}, z_{1:N})\}, \quad \theta = \begin{bmatrix} x \\ m \end{bmatrix} \tag{17}$$

Plug in our previously defined models

$$\theta^* = \arg\min_\theta \frac{1}{2} \left\{ \|m - \check{m}\|_P^2 + \|f(x,u) - x\|_Q^2 + \|h(\theta) - z\|_R^2 \right\} \tag{18}$$

$$= \arg\min_\theta \frac{1}{2} \|e(\theta)\|_\Sigma^2 \tag{19}$$

Linearization

Linearize the error function at the current best estimate $\bar{\theta}$

$$e(\theta) = (\bar{\theta} + \Delta\theta) = e(\bar{\theta}) + J\Delta\theta + O(\|\Delta\theta\|^2), \quad J = \left.\frac{\partial e(\theta)}{\partial \theta}\right|_{\theta=\bar{\theta}} \tag{20}$$

A new MAP estimator of the optimal increment $\Delta\theta^*$

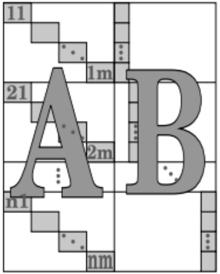
$$\Delta\theta^* = \arg\min_{\Delta\theta} \frac{1}{2}\|\bar{e} + J\Delta\theta\|_{\Sigma}^2, \quad \bar{\theta} \leftarrow \bar{\theta} + \Delta\theta^* \tag{21}$$

Setting the derivative to 0

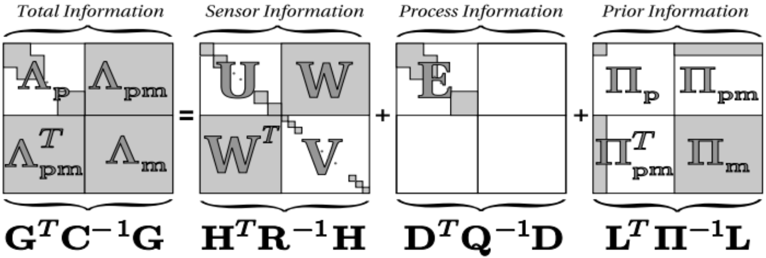
$$A\Delta\theta^* = J^\top\Sigma^{-1}J\Delta\theta^* = -J^\top\Sigma^{-1}\bar{e} = -b \tag{22}$$

$$(P^{-1} + G^\top Q^{-1}G + H^\top R^{-1}H)\Delta\theta^* = -(P^{-1}(\bar{m} - \check{m}) + G^\top Q^{-1}(\bar{f} - \bar{x}) + H^\top R^{-1}(\bar{h} - z)) \tag{23}$$

Sparse Jacobian and Hessian



(a) Structure of the measurement Jacobian



$$\underbrace{\begin{bmatrix} \Lambda_p & \Lambda_{pm} \\ \Lambda_{pm}^T & \Lambda_m \end{bmatrix}}_{\mathbf{G}^T\mathbf{C}^{-1}\mathbf{G}} = \underbrace{\begin{bmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^T & \mathbf{V} \end{bmatrix}}_{\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}} + \underbrace{\begin{bmatrix} \mathbf{E} & \\ & \end{bmatrix}}_{\mathbf{D}^T\mathbf{Q}^{-1}\mathbf{D}} + \underbrace{\begin{bmatrix} \Pi_p & \Pi_{pm} \\ \Pi_{pm}^T & \Pi_m \end{bmatrix}}_{\mathbf{L}^T\Pi^{-1}\mathbf{L}}$$

(b) Structure of the Hessian, as a composition from prior, motion and measurement.

Reduced Linear System

Reduce the map block on to the pose block (marginalization, Schur Complement)

$$\begin{bmatrix} A_{pp} & A_{pm} \\ A_{pm}^\top & A_{mm} \end{bmatrix} \begin{bmatrix} \Delta\theta_p^* \\ \Delta\theta_m^* \end{bmatrix} = \begin{bmatrix} -b_p \\ -b_m \end{bmatrix} \tag{24}$$

$$\begin{bmatrix} A_{pp} - A_{pm} A_{mm}^{-1} A_{pm}^\top & 0 \\ A_{pm}^\top & A_{mm} \end{bmatrix} \begin{bmatrix} \Delta\theta_p^* \\ \Delta\theta_m^* \end{bmatrix} = \begin{bmatrix} -b_p + A_{pm} A_{mm}^{-1} b_m \\ -b_m \end{bmatrix} \tag{25}$$

A_{mm} is block-diagonal, its inverse can be computed by inverting each block individually.

$$(A_{pp} - A_{pm} A_{mm}^{-1} A_{pm}^\top) \Delta\theta_p^* = -b_p + A_{pm} A_{mm}^{-1} b_m \tag{26}$$

$$A'_{pp} \Delta\theta_p^* = b'_p \tag{27}$$

$$A_{pm}^\top \Delta x_p^* + A_{mm} \Delta x_m^* = -b_m \tag{28}$$

Continuous-time SLAM

Two main factors determine how efficiently a SLAM problem can be solved are

1. the **size** of the state vector
2. the **sparsity** of the information matrix

Continuous-time posterior probability is

$$
\begin{aligned}
p(X|Z) &= p(x(t), m | u(t), z_{1:N}) && (29)\\
&= \frac{p(x(t), m | u(t)) p(z_{1:N} | x(t), m, u(t))}{p(z_{1:N} | u(t))} && (30)\\
&= \frac{p(x(t), m | u(t)) p(z_{1:N} | x(t), m)}{p(z_{1:N})} && (31)\\
&= \frac{p(m) p(x(t) | u(t)) \prod_{i=1}^{N} p(z_i | x(t_i), m)}{p(z_{1:N})} && (32)
\end{aligned}
$$

Parametric: Overview

The motion model $p(x(t)|u(t))$ is a continuous stochastic dynamical system described by the following differential equation

$$\dot{x}(t) = f(x(t), u(t)) + w(t), \quad w(t) \sim \mathcal{GP}(0, Q\delta(t - t')) \tag{33}$$

The probability density of the motion model is

$$p(x(t)|u(t)) \propto \exp\left\{-\frac{1}{2}\int_{t_0}^{t_K} \|\dot{x}(\tau) - f(x(\tau), u(\tau))\|_Q^2 d\tau\right\} \tag{34}$$

The motion term in the MAP estimator is

$$-\ln p(x(t)|u(t)) \propto L_u = \frac{1}{2}\int_{t_0}^{t_K} \|f(x(\tau), u(\tau)) - \dot{x}(\tau)\|_Q^2 d\tau \tag{35}$$

Parametric: Formulation

Approximate $x(t)$ as a weighted sum of a set of known temporal basis functions

$$\Phi(t) := \begin{bmatrix} \phi_1(t) & \cdots & \phi_M(t) \end{bmatrix}, \quad x(t) := \Phi(t)c \tag{36}$$

$$\theta(t) = \begin{bmatrix} x(t) \\ m \end{bmatrix} = \begin{bmatrix} \Phi(t)c \\ m \end{bmatrix} = \Psi(t)\beta, \quad \Psi(t) = \begin{bmatrix} \Phi(t) & \\ & 1 \end{bmatrix}, \quad \beta = \begin{bmatrix} c \\ m \end{bmatrix} \tag{37}$$

We can then linearize the motion term at $\bar{\beta}$

$$e_u(t) = f(x(t), u(t)) - \dot{x}(t) = g(t) \tag{38}$$

$$L_u = \frac{1}{2} \int_{t_0}^{t_K} \|e_u(\tau)\|_Q^2 d\tau \approx \frac{1}{2} \int_{t_0}^{t_K} \|\bar{e}_u(\tau) + J_u(\tau)\Delta c\|_Q^2 d\tau \tag{39}$$

$$\frac{\partial L_u}{\partial \Delta c} = \left(\int_{t_0}^{t_K} G(\tau)^\top Q^{-1} G(\tau) d\tau \right) \Delta c + \int_{t_0}^{t_K} G(\tau)^\top Q^{-1} \bar{g}(\tau) d\tau \tag{40}$$

Parametric: MAP Estimation

The MAP estimator of β is

$$\beta^* = \arg\min_{\beta} \frac{1}{2}\left\{ \|m - \check{m}\|^2_{P_m} + \int_{t_0}^{t_K} \|e_u(\tau)\|^2_Q d\tau + \|h(\Psi\beta) - z\|^2_R \right\} \tag{41}$$

Linearize at $\bar{\beta}$ and setting the derivative w.r.t. $\Delta\beta$ to 0

$$\left(P_m^{-1} + \int_{t_0}^{t_K} G(\tau)^\top Q^{-1} G(\tau) d\tau + \Psi^\top H^\top R^{-1} H \Psi \right) \Delta\beta^* \tag{42}$$

$$= -\left(P_m^{-1}(\bar{m} - \check{m}) + \int_{t_0}^{t_K} G(\tau)^\top Q^{-1} \bar{g}(\tau) d\tau + \Psi^\top H^\top R^{-1}(\bar{h} - z) \right) \tag{43}$$

Parametric: Discrete vs Continuous

	Discrete-time	Continuous-time
Posterior	$p(x_{0:K}, m \mid u_{1:K}, z_{1:N})$	$p(x(t), m \mid u(t), z_{1:N})$
Prior	$m \sim \mathcal{N}(\check{m}, P_m)$	same
Motion	$x_k \sim \mathcal{N}(f(x_{k-1}, u_k), Q_k)$	$\dot{x}(t) = f(x(t), u(t)) + w(t)$ $w(t) \sim \mathcal{GP}(0, Q\delta(t - t'))$
Measurement	$z_i \sim \mathcal{N}(h(x_i, m), R_i)$	$z_i \sim \mathcal{N}(h(x(t_i), m), R_i)$
A_u	$G^\top Q^{-1} G$	$\int_{t_0}^{t_K} G(\tau)^\top Q^{-1} G(\tau) d\tau$
b_u	$G^\top Q^{-1}(\bar{f} - \bar{x})$	$\int_{t_0}^{t_K} G(\tau)^\top Q^{-1} \bar{g}(\tau) d\tau$
A_z	$H^\top R^{-1} H$	$\Psi^\top H^\top R^{-1} H \Psi$
b_z	$H^\top R^{-1}(\bar{h} - z)$	$\Psi^\top H^\top R^{-1}(\bar{h} - z)$

Table 1: A comparison of discrete-time and continuous-time SLAM.

Parametric: Sparsity

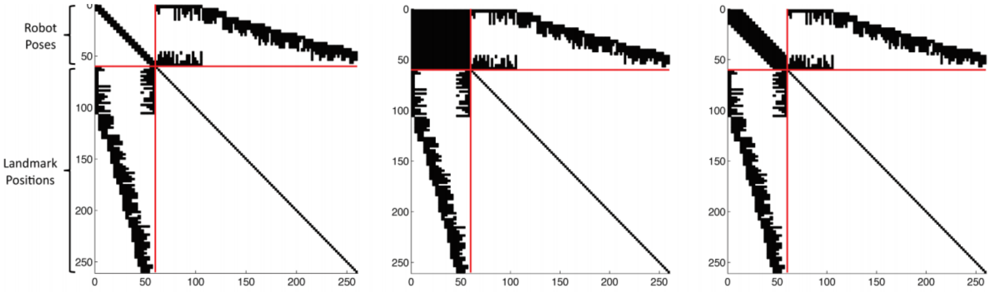


Figure 6: Sparsity patterns of the information matrix for a sample two-dimensional SLAM problem involving 20 robot poses and 100 landmarks.

Non-parametric: Overview

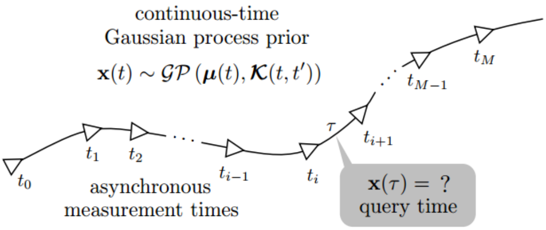


Figure 7: The trajectory of the robot can be represented by a continuous-time Gaussian process.

We directly model the motion prior as a Gaussian process

$$x(t) \sim \mathcal{GP}(\check{x}(t), K(t,t')), \quad \theta(t) \sim \mathcal{GP}(\check{\theta}(t), P(t,t')) \tag{44}$$

$$\theta(t) = \begin{bmatrix} x(t) \\ m \end{bmatrix}, \quad P(t,t') = \begin{bmatrix} K(t,t') & 0 \\ 0 & P_m \end{bmatrix} \tag{45}$$

Non-parametric: Discrete-time Solution

Discrete-time MAP estimator

$$\theta^* = \arg\min_{\theta} \frac{1}{2} \left\{ \|\theta - \check{\theta}\|_P^2 + \|h(\theta) - z\|_R^2 \right\} \tag{46}$$

with the following linear system

$$(P^{-1} + H^\top R^{-1} H)\Delta\theta^* = -(P^{-1}(\bar{\theta} - \check{\theta}) + H^\top R^{-1}(\bar{h} - z)) \tag{47}$$

where

$$P = \begin{bmatrix} K & 0 \\ 0 & P_m \end{bmatrix}, \quad K := [K(t_i, t_j)]\Big|_{1 \le i,j \le N} = \begin{bmatrix} K(t_1, t_1) & \cdots & K(t_1, t_N) \\ \vdots & \ddots & \vdots \\ K(t_N, t_1) & \cdots & K(t_N, t_N) \end{bmatrix} \tag{48}$$

Non-parametric: GP Interpolation

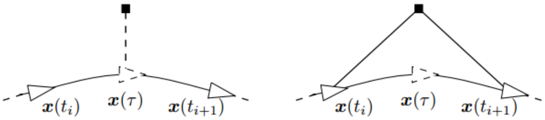


Figure 8: Measurement at state $x(\tau)$ (a) does not create an actual factor, the state $x(\tau)$ is instead interpolated by nearby states.

Let y be a subset of all poses that are in the state vector, x and y are jointly Gaussian

$$\begin{bmatrix} y \\ x \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \check{y} \\ \check{x} \end{bmatrix}, \begin{bmatrix} K_{yy} & K_{yx} \\ K_{xy} & K_{xx} \end{bmatrix}\right) \tag{49}$$

$$p(x|y) = \mathcal{N}(\check{x} + K_{xy}K_{yy}^{-1}(y - \check{y}), K_{xx} - K_{xy}K_{yy}^{-1}K_{yx}) \tag{50}$$

$$x = \check{x} + K_{xy}K_{yy}^{-1}(y - \check{y}), \quad \Delta x = K_{xy}K_{yy}^{-1}\Delta y \tag{51}$$

Non-parametric: MAP Estimation

Define $\gamma = [y, m]^\top$, we have the following linear relationship from GP interpolation

$$\theta = \check{\theta} + P_{xy}P_{yy}^{-1}(\gamma - \check{\gamma}) = \check{\theta} + \Pi(\gamma - \check{\gamma}) \tag{52}$$

$$\Delta\theta = P_{xy}P_{yy}^{-1}\Delta\gamma = \Pi\Delta\gamma \tag{53}$$

Now our MAP estimator becomes

$$\gamma^* = \arg\min_{\gamma} \left\{ \frac{1}{2}\|\theta - \check{\theta}\|_P^2 + \|h(\theta) - z\|_R^2 \right\} \tag{54}$$

$$= \arg\min_{\gamma} \left\{ \frac{1}{2}\|\Pi(\gamma - \check{\gamma})\|_P^2 + \|h(\check{\theta} + \Pi(\gamma - \check{\gamma})) - z\|_R^2 \right\} \tag{55}$$

Linearizing at $\bar{\gamma}$ and setting the derivative to 0 we have

$$(\Pi^\top P^{-1}\Pi + \Pi^\top H^\top R^{-1} H \Pi)\Delta\gamma^* = -(\Pi^\top P^{-1}(\bar{\theta} - \check{\theta}) + \Pi^\top H^\top R^{-1}(\bar{h} - z)) \tag{56}$$

Non-parametric: Equivalence to Parametric Form

Note that we have

$$\Delta\theta = \Psi\Delta\beta = \Pi\Delta\gamma, \quad P = \Psi B \Psi^\top \tag{57}$$

After some manipulation we see the two formulations are the same

$$\begin{aligned}
&(\Pi^\top P^{-1}\Pi + \Pi^\top H^\top R^{-1} H \Pi)\Delta\gamma^* && (58)\\
= &-(\Pi^\top P^{-1}(\bar{\theta} - \check{\theta}) + \Pi^\top H^\top R^{-1}(\bar{h} - z)) \qquad \text{(GPGN)} && (59)\\
&(\Pi^\top (\Psi B \Psi^\top)^{-1}\Pi + \Pi^\top H^\top R^{-1} H \Pi)\Pi^{-1}\Psi\Delta\beta^* && (60)\\
= &-(\Pi^\top (\Psi B \Psi^\top)^{-1}\Psi(\bar{\beta} - \check{\beta}) + \Pi^\top H^\top R^{-1}(\bar{h} - z)) && (61)\\
&(B^{-1} + \Psi^\top H^\top R^{-1} H \Psi)\Delta\beta^* && (62)\\
= &-(B^{-1}(\bar{\beta} - \check{\beta}) + \Psi^\top H^\top R^{-1}(\bar{h} - z)) \qquad \text{(Basis)} && (63)
\end{aligned}$$

Hybrid: Overview

1. **Question**: How do we choose the number of basis functions for the parametric model (or the number of poses for the non-parametric model)?
2. The core of SLAM is a **linearization-based**, gradient descent method, the complexity of which is a function of the dimension of the **error-state** vector. $A\Delta\theta^* = b$
3. Any linearization-based estimator relies on the computation of (1) **measurement residuals** and (2) a linear relationship between the residual and the **error state**.
4. We could use **different representations** for each of these. $x(t) = \hat{x}(t) + \delta x(t)$
5. The state estimate could be **discrete**, which makes few assumptions about the motion.
6. The state error could be **continuous**, which can reduce computation complexity.
7. The state error is generally much **smoother** than the trajectory, and can be well-modeled by a low-dimensional representation.

$$(P^{-1} + H^\top R^{-1} H)\Delta\theta^* = -(P^{-1}(\bar{\theta} - \check{\theta}) + H^\top R^{-1}(\bar{h} - z)) \tag{64}$$

Hybrid: Example

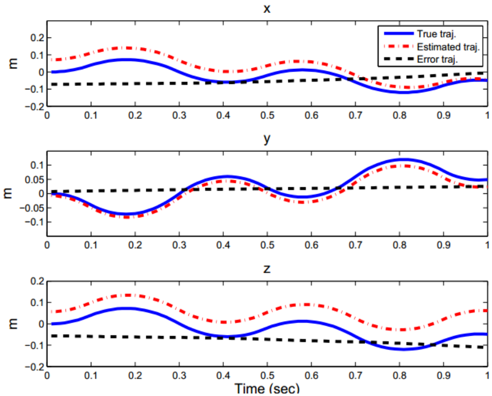


Figure 9: Example trajectory and estimation error during a one-second long interval of visual-inertial navigation.

Hybrid: Formulation

The trajectory estimate is defined in discrete time by $\hat{x} := \hat{x}_{1:N}$ and the trajectory error δx is defined in continuous time

$$\delta x \sim \mathcal{GP}(\delta\check{x}(t), K(t,t')), \quad \delta\theta \sim \mathcal{GP}(\delta\check{\theta}(t), P(t,t')) \tag{65}$$

The robot pose at a particular time t_i is the sum of the estimate and the error

$$x_i := x(t_i) = \hat{x}_i + \delta x(t_i), \quad \theta_i := \theta(t_i) = \hat{\theta}_i + \delta\theta(t_i) \tag{66}$$

Define $\gamma = [y, m]^\top$ and $\delta\gamma = [\delta y, 0]^\top$ we have

$$\delta\theta = \delta\check{\theta} + P_{xy}P_{yy}^{-1}(\delta\gamma - \delta\check{\gamma}) = \delta\check{\theta} + \Pi(\delta\gamma - \delta\check{\gamma}) \tag{67}$$

$$\Delta\delta\theta = P_{xy}P_{yy}^{-1}\Delta\delta\gamma = \Pi\Delta\delta\gamma \tag{68}$$

Hybrid: MAP Estimation

The MAP estimator can be derived from the discrete-time one

$$\theta^* = \arg\min_{\theta} \frac{1}{2} \left\{ \|\theta - \check{\theta}\|_P^2 + \|h(\theta) - z\|_R^2 \right\} \tag{69}$$

$$\delta\theta^* = \arg\min_{\delta\theta} \frac{1}{2} \left\{ \|\hat{\theta} - \check{\theta} + \delta\theta\|_P^2 + \|h(\hat{\theta} + \delta\theta) - z\|_R^2 \right\} \tag{70}$$

$$\delta\gamma^* = \arg\min_{\delta\gamma} \frac{1}{2} \|\hat{\theta} - \check{\theta} + \delta\check{\theta} + \Pi(\delta\gamma - \delta\check{\gamma})\|_P^2 \tag{71}$$

$$+ \frac{1}{2} \|h(\hat{\theta} + \delta\check{\theta} + \Pi(\delta\gamma - \delta\check{\gamma})) - z\|_R^2 \tag{72}$$

Linearizing at $\delta\bar{\gamma}$ and setting derivative to 0 we have

$$(\Pi^\top P^{-1}\Pi + \Pi^\top H^\top R^{-1} H \Pi)\Delta\delta\gamma^* \tag{73}$$

$$= -(\Pi^\top P^{-1}(\bar{\theta} - \check{\theta}) + \Pi^\top H^\top R^{-1}(\bar{h} - z)) \tag{74}$$

Conclusion

1. SLAM is all about solving a linear system $Ax = b$.
2. The size and pattern of A matters. We want it to be **small** and **sparse**.
3. Discrete-time SLAM maintains sparsity of A but cause it to grow quickly.
4. Continuous-time SLAM keeps the growth of A much slower.
5. But it will destroy the sparsity of A if we are not careful.
6. It is also tricky to pick the right model complexity beforehand.
7. A hybrid approach that combines both representations can handle arbitrarily complex trajectories.

Thanks! Questions?