Graphical State Space Model
ShaoLin Lü

Content
Introduction
Kalman Filter
Factor Graph Representation
Graphical State Space Model
A Simple Example
Open Source
Conclution and Future works

# Graphical State Space Model

ShaoLin Lü

GraphOptimization Inc.

2021.10.17

# Content

1 Introduction

2 Kalman Filter

3 Factor Graph Representation

4 Graphical State Space Model

5 A Simple Example

6 Open Source

7 Conclution and Future works

# Introduction

Optimal estimation is a hit subject for engineers. In many fields, Kalman filter is the chosen one in the eyes of countless engineers. Recently, factor graph was frequently used to solve many optimal estimation problems in robotics community.

- Pearl: belief propogation
- Loeiger: sum-product algorithm
- Dellaert and Kaess: factor graph
- Open source library: such as Ceres, G2O, GTSAM, miniSAM and minisam

# Babel Tower of Optimal Estimation


Graphical State Space Model

ShaoLin Lü

Content

Introduction

Kalman Filter

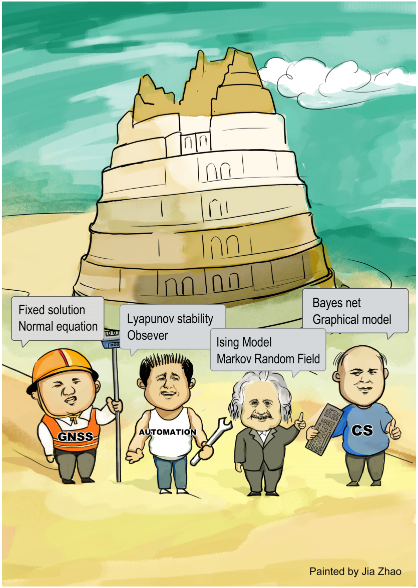
Factor Graph Representation

Graphical State Space Model

A Simple Example

Open Source

Conclution and Future works

# Graphcial models

Graphical State Space Model

ShaoLin Lü

Content

Introduction

Kalman Filter

Factor Graph Representation

Graphical State Space Model

A Simple Example

Open Source

Conclution and Future works

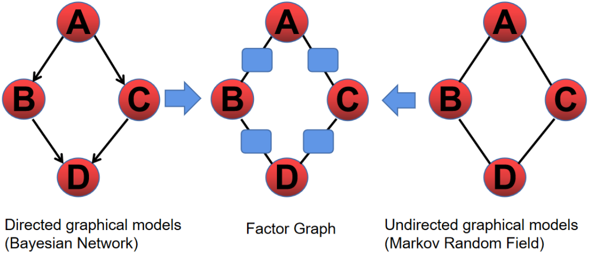



Figure 2: graphcial models

# Kalman Filter



Many problems in the real world can be formulated as a nonlinear system.

$$\dot{x} = f(x) + B(u) + q \tag{1}$$

$$y = C(x) + r \tag{2}$$

The linearization of the above system can be discribed as follows

$$\dot{x} = Ax + Bu + q \tag{3}$$

$$y = Cx + r \tag{4}$$

The above system can be discretized as follows

$$x_{k+1} = F_k x_k + B_k u_k + q_k, F_k \approx I_{n \times n} + AT \tag{5}$$

$$y_{k+1} = C_{k+1} x_{k+1} + r_k \tag{6}$$

# Standard Kalman Filter



Content
Introduction
Kalman Filter
Factor Graph Representation
Graphical State Space Model
A Simple Example
Open Source
Conclution and Future works

Initialize the estimation,

$$\hat{x}_0 = E(x_0) = x_0^+ \tag{7}$$

$$P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] \tag{8}$$

Update the prior mean and covariance

$$x_{k+1|k} = F_k x_k + B_k u_k \tag{9}$$

$$P_{k+1|k} = F_k P_k F_k^T + Q_k \tag{10}$$

Calculate Kalman gain

$$K_{k+1} = P_{k+1|k} C_{k+1}^T (C_{k+1} P_{k+1|k} C_{k+1}^T + R_k)^{-1} \tag{11}$$

Update the posteriori mean and covariance

$$x_{k+1} = x_{k+1|k} + K_{k+1}(y_{k+1} - C_{k+1} x_{k+1|k}) \tag{12}$$

$$P_{k+1} = (I - K_{k+1} C_{k+1}) P_{k+1|k} \tag{13}$$

# Kalman Filter



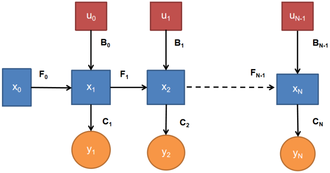



Figure 3: State space model representation of Kalman filter.

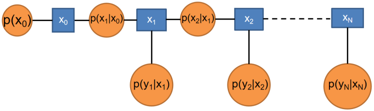



Figure 4: Factor Graph Representation of Kalman filter.

# The block equation of Kalman Filter

$$
P_w = diag \begin{bmatrix} P_0 \\ Q_0 \\ R_1 \\ Q_1 \\ R_2 \\ \\ \vdots \\ \\ Q_{N-1} \\ R_N \end{bmatrix}, \begin{bmatrix} I_{n\times n} & 0_{n\times n} & 0_{n\times n} & \cdots & 0_{n\times n} & 0_{m\times n} \\ -F_0 & I_{n\times n} & 0_{n\times n} & \cdots & 0_{n\times n} & 0_{n\times n} \\ 0_{m\times n} & C_1 & 0_{m\times n} & \cdots & 0_{m\times n} & 0_{m\times n} \\ 0_{n\times n} & -F_1 & I_{n\times n} & \cdots & 0_{n\times n} & 0_{n\times n} \\ 0_{m\times n} & 0_{m\times n} & C_2 & \cdots & 0_{m\times n} & 0_{m\times n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & \cdots & -F_{N-1} & I_{n\times n} \\ 0_{m\times n} & 0_{m\times n} & 0_{m\times n} & \cdots & 0_{m\times n} & C_N \end{bmatrix}
\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} \hat{x}_0^+ \\ B_0 u_0 \\ y_1 \\ B_1 u_1 \\ y_2 \\ \vdots \\ B_{N-1} u_{N-1} \\ y_N \end{bmatrix} \tag{14}
$$

# Kalman Filter

$$P_w = diag \begin{bmatrix} P_k \\ Q_k \\ R_{k+1} \end{bmatrix}, \begin{bmatrix} I_{n\times n} & 0_{n\times n} \\ -F_k & I_{n\times n} \\ 0_{m\times n} & C_{k+1} \end{bmatrix} \begin{bmatrix} x_k \\ x_{k+1} \end{bmatrix} = \begin{bmatrix} \hat{x}_k \\ B_k u_k \\ y_{k+1} \end{bmatrix} \tag{15}$$

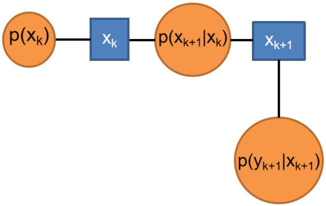



Figure 5: Factor Graph representation of Kalman filter at time $k+1$.

# Factor Graph Representation


Graphical State Space Model

ShaoLin Lü

Content

Introduction

Kalman Filter

Factor Graph Representation

Graphical State Space Model

A Simple Example

Open Source

Conclution and Future works


There are two kinds of methods for factor graph optimization: global optimization and sliding window optimization.
The difference lies in whether do marginalization or not.
When factor graph optimization is used in real-time processing case, sliding window optimization will be adopted.

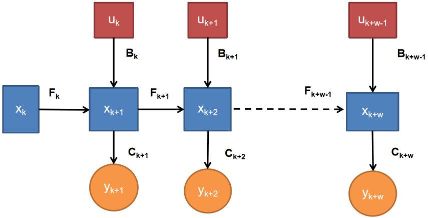



Figure 6: State Space model in Sliding window case.

# Factor Graph Representation

It should not be ignored, what factor graph can do while the traditional space model can not do:

- Model the system as a multi connected graph.
- Model the system distributedly not unifiedly.
- Model one part of the system as constants rather than time-series variables.

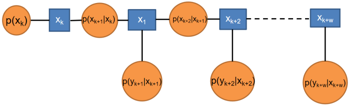



Figure 7: Factor Graph Representation in Sliding window case.

# Factor Graph Representation

The block tridiagonal equation goes as follows

$$P_w = diag\begin{bmatrix} P_k \\ Q_k \\ R_{k+1} \\ Q_{k+1} \\ R_{k+2} \\ \vdots \\ Q_{k+w-1} \\ R_{k+w} \end{bmatrix}, X_w = \begin{bmatrix} x_k \\ x_{k+1} \\ x_{k+2} \\ \vdots \\ \vdots \\ x_{k+w-1} \\ x_{k+w} \end{bmatrix}, b_w = \begin{bmatrix} \hat{x}_k^+ \\ B_k u_k \\ y_{k+1} \\ B_{k+1}u_{k+1} \\ y_{k+2} \\ \vdots \\ B_{k+w-1}u_{k+w-1} \\ y_{k+w} \end{bmatrix},$$

$$A_w = \begin{bmatrix} I_{n\times n} & 0_{n\times n} & 0_{n\times n} & \cdots & 0_{n\times n} & 0_{n\times n} \\ -F_k & I_{n\times n} & 0_{n\times n} & \cdots & 0_{n\times n} & 0_{n\times n} \\ 0_{m\times n} & C_{k+1} & 0_{m\times n} & \cdots & 0_{m\times n} & 0_{m\times n} \\ 0_{n\times n} & -F_{k+1} & I_{n\times n} & \cdots & 0_{n\times n} & 0_{n\times n} \\ 0_{m\times n} & 0_{m\times n} & C_{k+2} & \cdots & 0_{m\times n} & 0_{m\times n} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_{n\times n} & 0_{n\times n} & 0_{n\times n} & \cdots & -F_{k+w-1} & I_{n\times n} \\ 0_{m\times n} & 0_{m\times n} & 0_{m\times n} & \cdots & 0_{m\times n} & C_{k+w} \end{bmatrix}$$

$$A_w X_w = b_w \tag{16}$$

$$\hat{X}_w = (A_w^T P_w A_w)^{-1} A_w^T P_w b_w \tag{17}$$

# Graphical State Space Model



Content
Introduction
Kalman Filter
Factor Graph Representation
Graphical State Space Model
A Simple Example
Open Source
Conclution and Future works

If you want better results via factor graph optimization, perhaps you should try a different state space model. Consider the system as follows

$$\begin{bmatrix} \dot{x_c} \\ \dot{x_b} \end{bmatrix} = \begin{bmatrix} A_c & A_b \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ x_b \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} q \\ 0 \end{bmatrix} \quad (18)$$

$$y = \begin{bmatrix} C_c & C_b \end{bmatrix} \begin{bmatrix} x_c \\ x_b \end{bmatrix} + r \quad (19)$$

When Kalman filter is used to estimate the state, $[x_c, x_b]^T$, in the above equations,the prediction matrix will be

$$F_k \approx I_{n \times n} + AT$$
$$A = \begin{bmatrix} A_c & A_b \\ 0 & 0 \end{bmatrix} \quad (20)$$

Obviously, Kalman filter can not make use of the characteristics of this kind of this system.

# Graphical State Space Model

Graphical State Space Model
ShaoLin Lü
Content
Introduction
Kalman Filter
Factor Graph Representation
Graphical State Space Model
A Simple Example
Open Source
Conclution and Future works

The above system described can be discretized as follows

$$
\begin{aligned}
x_c(k+1) &= F_c(k)x_c(k) + F_b(k)x_b + B_k u_k + q_c(k) \\
F_c(k) &\approx I_{n\times n} + A_c T, F_b = A_b T \qquad (21) \\
y_{k+1} &= C_c(k+1)x_c(k+1) + C_b(k+1)x_b + r_k \qquad (22) \\
E(x_b) &= x_b^+ \qquad (23)
\end{aligned}
$$

This means that constant states can be modeled as a different form rather than a time-series form. This is the distributed discretization not the unified discretization.The limitation of Kalman filter is that it must use time-series forms to represent constant variables.

# Graphical State Space Model

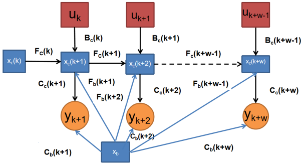



Figure 8: Graphical state space model represented in state space form for sliding window optimization case.

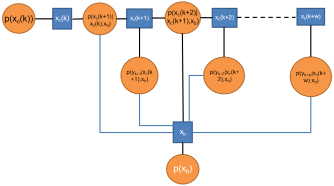
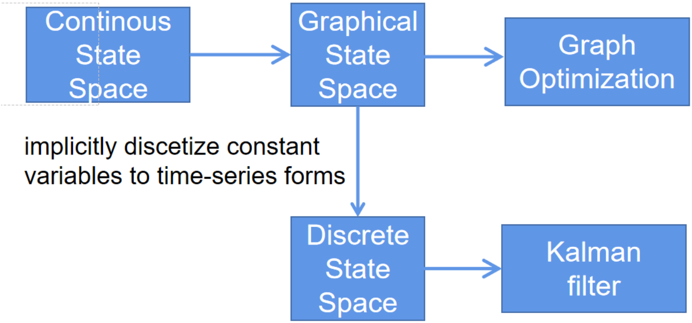



Figure 9: Graphical state space model represented in factor graph

# Flow Chart




Continous State Space
Graphical State Space
Graph Optimization
implicitly discetize constant variables to time-series forms
Discrete State Space
Kalman filter

# Graphical State Space Model



Content
Introduction
Kalman Filter
Factor Graph Representation
Graphical State Space Model
A Simple Example
Open Source
Conclution and Future works

$$P_w = diag\begin{bmatrix} P_b \\ P_c(k) \\ Q_c(k) \\ R_{k+1} \\ Q_c(k+1) \\ R_{k+2} \\ \vdots \\ Q_c(k+w-1) \\ R_{k+w} \end{bmatrix}, X_w = \begin{bmatrix} x_b \\ x_c(k) \\ x_c(k+1) \\ x_c(k+2) \\ \vdots \\ \vdots \\ x_c(k+w-1) \\ x_c(k+w) \end{bmatrix}, b_w = \begin{bmatrix} \hat{x}_b^+ \\ \hat{x}_c^+(k) \\ B_k u_k \\ y_{k+1} \\ B_{k+1}u_{k+1} \\ y_{k+2} \\ \vdots \\ B_{k+w-1}u_{k+w-1} \\ y_{k+w} \end{bmatrix},$$

$$A_w X_w = b_w \tag{24}$$

$$A_w = \begin{bmatrix} I_{n_b\times n_b} & 0_{n_b\times n_c} & 0_{n_b\times n_c} & 0_{n_b\times n_c} & \cdots & 0_{n_b\times n_c} & 0_{n_b\times n_c} \\ 0_{n_c\times n_b} & I_{n_c\times n_c} & 0_{n_c\times n_c} & 0_{n_c\times n_c} & \cdots & 0_{n_c\times n_c} & 0_{n_c\times n_c} \\ -F_b(k) & -F_c(k) & I_{n_c\times n_c} & 0_{n_c\times n_c} & \cdots & 0_{n_c\times n_c} & 0_{n_c\times n_c} \\ C_b(k+1) & 0_{m\times n_c} & C_c(k+1) & 0_{m\times n_c} & \cdots & 0_{m\times n_c} & 0_{m\times n_c} \\ -F_b(k+1) & 0_{n_c\times n_c} & -F_c(k+1) & I_{n_c\times n_c} & \cdots & 0_{n_c\times n_c} & 0_{n_c\times n_c} \\ C_b(k+2) & 0_{m\times n_c} & 0_{m\times n_c} & C_c(k+2) & \cdots & 0_{m\times n_c} & 0_{m\times n_c} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -F_b(k+w-1) & 0_{n_c\times n_c} & 0_{n_c\times n_c} & 0_{n_c\times n_c} & \cdots & -F_c(k+w-1) & I_{n_c\times n_c} \\ C_b(k+w) & 0_{m\times n_c} & 0_{m\times n_c} & 0_{m\times n_c} & \cdots & 0_{m\times n_c} & C_c(k+w) \end{bmatrix}$$

# A Simple Example



The continous-time system of simple radar tracking is described by

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ h \end{bmatrix} + q \tag{25}$$

$$\tilde{\rho} = \begin{bmatrix} \frac{x}{\sqrt{x^2+h^2}} & 0 & \frac{h}{\sqrt{x^2+h^2}} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ h \end{bmatrix} + r \tag{26}$$

# A Simple Example



When the framework of Kalman filter is used, the discrete-time system goes as follows

$$\begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \\ h_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \\ h_k \end{bmatrix} + q_k \tag{27}$$

$$\tilde{\rho}_{k+1} = \begin{bmatrix} \frac{x_{k+1}}{\sqrt{x_{k+1}^2 + h_{k+1}^2}} & 0 & \frac{h_{k+1}}{\sqrt{x_{k+1}^2 + h_{k+1}^2}} \end{bmatrix} \begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \\ h_{k+1} \end{bmatrix} + r_{k+1} \tag{28}$$

# A Simple Example

If graphical state space model is used, the system state can be modeled distributedly.

$$\alpha_{k+1} = \frac{x_{k+1}}{\sqrt{x_{k+1}^2 + h^2}}, \beta_{k+1} = \frac{h}{\sqrt{x_{k+1}^2 + h^2}} \tag{29}$$

The discrete-time system is discribed by

$$A_w = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -T & -1 & 1 & 0 & \cdots & 0 & 0 \\ \beta_{k+1} & 0 & 0 & \alpha_{k+1} & 0 & \cdots & 0 & 0 \\ 0 & -T & 0 & -1 & 1 & \cdots & 0 & 0 \\ \beta_{k+2} & 0 & 0 & 0 & \alpha_{k+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & -T & 0 & 0 & 0 & \cdots & -1 & 1 \\ \beta_{k+w} & 0 & 0 & 0 & 0 & \cdots & 0 & \alpha_{k+w} \end{bmatrix} \tag{30}$$

# A Simple Example


Graphical State Space Model

ShaoLin Lü

Content

Introduction

Kalman Filter

Factor Graph Representation

Graphical State Space Model

A Simple Example

Open Source

Conclution and Future works


$$P_w = diag\begin{bmatrix} P_h \\ P_{\dot{x}} \\ P(k) \\ Q(k) \\ R_{k+1} \\ Q(k+1) \\ R_{k+2} \\ \vdots \\ Q(k+w-1) \\ R_{k+w} \end{bmatrix}, X_w = \begin{bmatrix} h \\ \dot{x} \\ x_k \\ x_{k+1} \\ x_{k+2} \\ \vdots \\ \vdots \\ x_{k+w-1} \\ x_{k+w} \end{bmatrix}, b_w = \begin{bmatrix} \hat{h}^+ \\ \hat{\dot{x}}^+ \\ \hat{x}^+(k) \\ 0 \\ y_{k+1} \\ 0 \\ y_{k+2} \\ \vdots \\ 0 \\ y_{k+w} \end{bmatrix},$$

$$A_w X_w = b_w \tag{31}$$

# Simulation Results

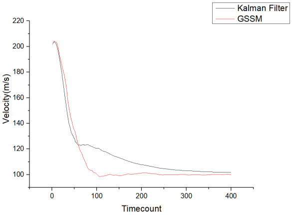
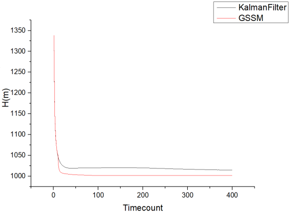



Figure 10: Velocity Estimation.



# Check the codes! Maybe I am wrong!



Content
Introduction
Kalman Filter
Factor Graph Representation
Graphical State Space Model
A Simple Example
Open Source
Conclution and Future works

http://github.com/shaolinbit/GraphicalStateSpaceModel

# Conclution and Future works

Graphical State Space Model

ShaoLin Lü

Content

Introduction

Kalman Filter

Factor Graph Representation

Graphical State Space Model

A Simple Example

Open Source

Conclution and Future works

In this paper, a new framework, named as graphical state space model, was proposed to estimate the state of a class of nonlinear system. A simple example was used to demonstrate the efficiency of this framework. In some cases, graphical state space model can extract more information than the standard Kalman disrete model. Detailed error analysis should be done in the future. What should be explorered next is that which kind of system can be estimated by this framework.