An Inequality Constrained Nonlinear Kalman-Bucy Smoother

Bradley Bell [1] James Burke [2] Gianluigi Pillonetto [3]

National Science Foundation (DMS-0505712)

National Institute of Health (NIBIB 2 P41 EB01975)

[1] Applied Physics Laboratory, University of Washington
[2] Mathematics, University of Washington
[3] Information Engineering, University of Padova

Applications

Applications

- Aircraft tracking (location and velocities)

Applications

- Aircraft tracking (location and velocities)
- Ship tracking

Applications

- Aircraft tracking (location and velocities)
- Ship tracking
- Underwater tracking

Applications

- Aircraft tracking (location and velocities)
- Ship tracking
- Underwater tracking
- Compartmental analysis of drug concentrations and clearance

Outline

- Unconstrained Kalman-Bucy smoother.

Outline

- Unconstrained Kalman-Bucy smoother.

  An optimization viewpoint (Bell 94)

Outline

- Unconstrained Kalman-Bucy smoother.

  An optimization viewpoint (Bell 94)

- Constrained Kalman-Bucy smoother.

Outline

- Unconstrained Kalman-Bucy smoother.

  An optimization viewpoint (Bell 94)

- Constrained Kalman-Bucy smoother.

- An interior point framework.

Outline

- Unconstrained Kalman-Bucy smoother.

  An optimization viewpoint (Bell 94)

- Constrained Kalman-Bucy smoother.

- An interior point framework.

- A toy ship tracking problem.

Outline

- Unconstrained Kalman-Bucy smoother.

  An optimization viewpoint (Bell 94)

- Constrained Kalman-Bucy smoother.

- An interior point framework.

- A toy ship tracking problem.

- Implementation of a constrained iterated Kalman-Bucy smoother.

Unconstrained Transition Model

$$x_k = g_k(x_{k-1}) + w_k, \quad k = 1, 2, \ldots, N \text{ (time)}$$

Unconstrained Transition Model

$$x_k = g_k(x_{k-1}) + w_k, \quad k = 1, 2, \ldots, N \text{ (time)}$$

$$x_0 = \hat{x}_0 + e_0$$

- $\hat{x}_0$ a known estimate, $e_0 \sim N(0, P_0)$

Unconstrained Transition Model

$$x_k = g_k(x_{k-1}) + w_k, \quad k = 1, 2, \ldots, N \text{ (time)}$$

$$x_0 = \hat{x}_0 + e_0$$

- $\hat{x}_0$ a known estimate, $e_0 \sim N(0, P_0)$
- $g : \mathbb{R}^n \to \mathbb{R}^n$ the transition function (known)

Unconstrained Transition Model

$$x_k = g_k(x_{k-1}) + w_k, \quad k = 1, 2, \ldots, N \text{ (time)}$$

$$x_0 = \hat{x}_0 + e_0$$

- $\hat{x}_0$ a known estimate, $e_0 \sim N(0, P_0)$
- $g : \mathbb{R}^n \to \mathbb{R}^n$ the transition function (known)
- $x_k \in \mathbb{R}^n$ the state sequence (unknown)

Unconstrained Transition Model

$$x_k = g_k(x_{k-1}) + w_k, \quad k = 1, 2, \ldots, N \text{ (time)}$$

$$x_0 = \hat{x}_0 + e_0$$

- $\hat{x}_0$ a known estimate, $e_0 \sim N(0, P_0)$
- $g : \mathbb{R}^n \to \mathbb{R}^n$ the transition function (known)
- $x_k \in \mathbb{R}^n$ the state sequence (unknown)
- $w_k \sim N(0, Q_k)$ transition noise

Unconstrained Transition Model

$$x_k = g_k(x_{k-1}) + w_k, \quad k = 1, 2, \ldots, N \text{ (time)}$$

$$x_0 = \hat{x}_0 + e_0$$

- $\hat{x}_0$ a known estimate, $e_0 \sim N(0, P_0)$
- $g : \mathbb{R}^n \to \mathbb{R}^n$ the transition function (known)
- $x_k \in \mathbb{R}^n$ the state sequence (unknown)
- $w_k \sim N(0, Q_k)$ transition noise
- $Q_k \in \mathcal{S}^n_+$ transition covariance (known)

Unconstrained Measurment Model

$$z_k = h_k(x_k) + v_k, \quad k = 1, 2, \ldots, N$$

Unconstrained Measurment Model

$$z_k = h_k(x_k) + v_k, \quad k = 1, 2, \ldots, N$$

- $z_k \in \mathbb{R}^m$ measurement vector (known)
- $h_k : \mathbb{R}^n \rightarrow \mathbb{R}^m$ the measurement function (known)

Unconstrained Measurment Model

$$z_k = h_k(x_k) + v_k, \quad k = 1, 2, \ldots, N$$

- $z_k \in \mathbb{R}^m$ measurement vector (known)
- $h_k : \mathbb{R}^n \to \mathbb{R}^m$ the measurement function (known)
- $v_k \sim N(0, R_k)$ measurement noise
- $R_k \in \mathcal{S}_+^n$ measurement covariance

Full Model

$$
\begin{aligned}
x_0 &= \hat{x}_0 + e_0 \\
x_k &= g_k(x_{k-1}) + w_k, \quad k = 1, 2, \ldots, N \\
z_k &= h_k(x_k) + v_k, \quad k = 1, 2, \ldots, N
\end{aligned}
$$

$$
e_0 \sim N(0, P_0), \quad w_k \sim N(0, Q_k), \quad v_k \sim N(0, R_k)
$$

Full Model

$$
\begin{aligned}
x_0 &= \hat{x}_0 + e_0 \\
x_k &= g_k(x_{k-1}) + w_k, \quad k = 1, 2, \ldots, N \\
z_k &= h_k(x_k) + v_k, \quad k = 1, 2, \ldots, N
\end{aligned}
$$

$$
e_0 \sim N(0, P_0), \quad w_k \sim N(0, Q_k), \quad v_k \sim N(0, R_k)
$$

Affine Case: $\nabla g = \text{constant}$ and $\nabla h = \text{constant}$

The Likelihood Function

$$\mathcal{L}(x_0, \ldots, x_N) \quad =$$

The Likelihood Function

$$\mathcal{L}(x_0, \ldots, x_N) \quad = \quad K \exp\left[-\frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0)\right]$$

The Likelihood Function

$$\mathcal{L}(x_0, \ldots, x_N) = K \exp\left[-\frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0)\right]$$

$$\bullet \prod_{k=1}^{N} \exp\left[-\frac{1}{2}\left(x_k - g_k(x_{k-1})\right)^T Q_k^{-1}\left(x_k - g_k(x_{k-1})\right)\right]$$

The Likelihood Function

$$\begin{aligned}\mathcal{L}(x_0, \ldots, x_N) &= K \exp\left[-\frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1} (x_0 - \hat{x}_0)\right] \\ &\bullet \prod_{k=1}^{N} \exp\left[-\frac{1}{2}\left(x_k - g_k(x_{k-1})\right)^T Q_k^{-1} \left(x_k - g_k(x_{k-1})\right)\right] \\ &\bullet \prod_{k=1}^{N} \exp\left[-\frac{1}{2}\left(z_k - h_k(x_k)\right)^T R_k^{-1} \left(z_k - h_k(x_k)\right)\right]\end{aligned}$$

The Unconstrained Likelihood Problem

The negative log-likelihood function

$$\begin{aligned} L(x_0,\ldots,x_N) &= \frac{1}{2}(x_0-\hat{x}_0)^T P_0^{-1}(x_0-\hat{x}_0) \\ &+\frac{1}{2}\sum_{k=1}^{N} \begin{array}{c} (x_k-g_k(x_{k-1}))^T\, Q_k^{-1}\,(x_k-g_k(x_{k-1})) \\ + \\ (z_k-h_k(x_k))^T\, R_k^{-1}\,(z_k-h_k(x_k)) \end{array} \end{aligned}$$

The Unconstrained Likelihood Problem

The negative log-likelihood function

$$L(x_0, \ldots, x_N) = \frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0) + \frac{1}{2}\sum_{k=1}^{N} \begin{array}{c} (x_k - g_k(x_{k-1}))^T Q_k^{-1} (x_k - g_k(x_{k-1})) \\ + \\ (z_k - h_k(x_k))^T R_k^{-1} (z_k - h_k(x_k)) \end{array}$$

$$\text{minimize } L(x) \qquad \text{s.t.} \quad x \in \mathbb{R}^{nN}$$

The Iterated Kalman-Bucy Smoother (Bell 94)

Kalman-Filter

$$\min_{x_N} \min_{x_{N-1}} \ldots \min_{x_1} \min_{x_0} L(x_0, x_1, \ldots, x_N)$$

The Iterated Kalman-Bucy Smoother (Bell 94)

Kalman-Filter

$$\min_{x_N} \min_{x_{N-1}} \ldots \min_{x_1} \min_{x_0} L(x_0, x_1, \ldots, x_N)$$

$$L(x_0, \ldots, x_N) \quad = \quad \frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0)$$

The Iterated Kalman-Bucy Smoother (Bell 94)

Kalman-Filter

$$\min_{x_N} \min_{x_{N-1}} \dots \min_{x_1} \min_{x_0} L(x_0, x_1, \dots, x_N)$$

$$\begin{aligned} L(x_0, \dots, x_N) \quad = \quad & \frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0) \\ & + \frac{1}{2}\left(z_0 - h_0(x_0)\right)^T R_0^{-1}\left(z_0 - h_0(x_0)\right) \end{aligned}$$

The Iterated Kalman-Bucy Smoother (Bell 94)

Kalman-Filter

$$\min_{x_N} \min_{x_{N-1}} \ldots \min_{x_1} \min_{x_0} L(x_0, x_1, \ldots, x_N)$$

$$\begin{aligned} L(x_0, \ldots, x_N) \quad = \quad & \frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0) \\ & + \frac{1}{2}\left(z_0 - h_0(x_0)\right)^T R_0^{-1}\left(z_0 - h_0(x_0)\right) \end{aligned}$$

$$z_0 = x_0, \quad h_0(x) = x, \quad R_0 = I$$

The Iterated Kalman-Bucy Smoother (Bell 94)

Kalman-Filter

$$\min_{x_N} \min_{x_{N-1}} \ldots \min_{x_1} \min_{x_0} L(x_0, x_1, \ldots, x_N)$$

$$\begin{aligned} L(x_0, \ldots, x_N) \quad = \quad & \frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0) \\ & + \frac{1}{2}\left(z_0 - h_0(x_0)\right)^T R_0^{-1} \left(z_0 - h_0(x_0)\right) \\ & + \frac{1}{2}\left(x_1 - g_1(x_0)\right)^T Q_1^{-1} \left(x_1 - g_1(x_0)\right) \end{aligned}$$

The Iterated Kalman-Bucy Smoother (Bell 94)

Kalman-Filter

$$\min_{x_N} \min_{x_{N-1}} \dots \min_{x_1} \min_{x_0} L(x_0, x_1, \dots, x_N)$$

$$\begin{aligned}
L(x_0, \dots, x_N) \quad = \quad & \frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0) \\
& + \frac{1}{2}\left(z_0 - h_0(x_0)\right)^T R_0^{-1}\left(z_0 - h_0(x_0)\right) \\
& + \frac{1}{2}\left(x_1 - g_1(x_0)\right)^T Q_1^{-1}\left(x_1 - g_1(x_0)\right) \\
& + \frac{1}{2}\sum_{k=1}^{N-1} \begin{array}{c} \left(z_k - h_k(x_k)\right)^T R_k^{-1}\left(z_k - h_k(x_k)\right) \\ + \\ \left(x_{k+1} - g_k(x_k)\right)^T Q_{k+1}^{-1}\left(x_{k+1} - g_k(x_k)\right) \end{array}
\end{aligned}$$

Basic Optimization Problem: Affine Case

$$\begin{aligned} \min_{(x,y,w)} \quad & (x-\hat{x})^T P^{-1}(x-\hat{x}) + (z-h(x))^T R^{-1}(z-h(x)) \\ & (y-g(x))^T Q^{-1}(y-g(x)) + f(y,w) \end{aligned}$$

Basic Optimization Problem: Affine Case

$$\min_{(x,y,w)} \quad (x-\hat{x})^T P^{-1}(x-\hat{x}) + (z-h(x))^T R^{-1}(z-h(x))$$
$$(y-g(x))^T Q^{-1}(y-g(x)) + f(y,w)$$

$$\min_{(y,w)} \quad \min_{x} \quad (x-\hat{x})^T P^{-1}(x-\hat{x}) + (z-h(x))^T R^{-1}(z-h(x))$$
$$(y-g(x))^T Q^{-1}(y-g(x)) + f(y,w)$$

Dimension Reduction: Affine Case

$$\min_{(y,w)} (y - \hat{y})^T \widehat{P}^{-1}(y - \hat{y}) + f(y, w)$$

Dimension Reduction: Affine Case

$$\min_{(y,w)} (y - \hat{y})^T \widehat{P}^{-1} (y - \hat{y}) + f(y, w)$$

where

$$U = [P^{-1} + \nabla h R^{-1} \nabla h^T]^{-1}, \qquad \widehat{P} = Q + \nabla g^T U \nabla g,$$

$$u = \hat{x} + U \nabla h R^{-1} (z - h(\hat{x})), \qquad \hat{y} = g(u).$$

Dimension Reduction: Affine Case

$$\min_{(y,w)} (y - \hat{y})^T \widehat{P}^{-1}(y - \hat{y}) + f(y, w)$$

where

$$U = [P^{-1} + \nabla h R^{-1} \nabla h^T]^{-1}, \qquad \widehat{P} = Q + \nabla g^T U \nabla g,$$

$$u = \hat{x} + U \nabla h R^{-1}(z - h(\hat{x})), \qquad \hat{y} = g(u).$$

Recursive application gives the Kalman Filter yielding the optimal x_N.

Dimension Reduction: Affine Case

$$\min_{(y,w)} (y - \hat{y})^T \widehat{P}^{-1}(y - \hat{y}) + f(y, w)$$

where

$$U = [P^{-1} + \nabla h R^{-1} \nabla h^T]^{-1}, \qquad \widehat{P} = Q + \nabla g^T U \nabla g,$$

$$u = \hat{x} + U \nabla h R^{-1}(z - h(\hat{x})), \qquad \hat{y} = g(u).$$

Recursive application gives the Kalman Filter yielding the optimal x_N.

The Kalman-Bucy smoother is the complete optimal solution obtained by back-solving first for x_{N-1}, then for $x_{N-2}, \ \ldots$.

Dimension Reduction: Affine Case

$$\begin{aligned}
L(x_0, \ldots, x_N) &= \frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0) + \frac{1}{2}\left(z_0 - h_0(x_0)\right)^T R_0^{-1}\left(z_0 - h_0(x_0)\right) \\
&+ \frac{1}{2}\left(x_1 - g_1(x_0)\right)^T Q_1^{-1}\left(x_1 - g_1(x_0)\right) \\
&+ \frac{1}{2}\sum_{k=1}^{N-1} \begin{array}{c} \left(z_k - h_k(x_k)\right)^T R_k^{-1}\left(z_k - h_k(x_k)\right) \\ + \\ \left(x_{k+1} - g_k(x_k)\right)^T Q_{k+1}^{-1}\left(x_{k+1} - g_k(x_k)\right) \end{array}
\end{aligned}$$

Dimension Reduction: Affine Case

$$
\begin{aligned}
L(x_0, \ldots, x_N) &= \frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0) + \frac{1}{2}\left(z_0 - h_0(x_0)\right)^T R_0^{-1}\left(z_0 - h_0(x_0)\right) \\
&\quad + \frac{1}{2}\left(x_1 - g_1(x_0)\right)^T Q_1^{-1}\left(x_1 - g_1(x_0)\right) \\
&\quad + \frac{1}{2}\sum_{k=1}^{N-1} \begin{array}{c} \left(z_k - h_k(x_k)\right)^T R_k^{-1}\left(z_k - h_k(x_k)\right) \\ + \\ \left(x_{k+1} - g_k(x_k)\right)^T Q_{k+1}^{-1}\left(x_{k+1} - g_k(x_k)\right) \end{array}
\end{aligned}
$$

$$
L_1(x_1, \ldots, x_N) = \frac{1}{2}(x_1 - \hat{x}_1)^T P_1^{-1}(x_1 - \hat{x}_1)
$$

Dimension Reduction: Affine Case

$$
\begin{aligned}
L(x_0, \ldots, x_N) &= \frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0) + \frac{1}{2}\left(z_0 - h_0(x_0)\right)^T R_0^{-1}\left(z_0 - h_0(x_0)\right) \\
&+ \frac{1}{2}\left(x_1 - g_1(x_0)\right)^T Q_1^{-1}\left(x_1 - g_1(x_0)\right) \\
&+ \frac{1}{2}\sum_{k=1}^{N-1} \begin{matrix} \left(z_k - h_k(x_k)\right)^T R_k^{-1}\left(z_k - h_k(x_k)\right) \\ + \\ \left(x_{k+1} - g_k(x_k)\right)^T Q_{k+1}^{-1}\left(x_{k+1} - g_k(x_k)\right) \end{matrix}
\end{aligned}
$$

$$
L_1(x_1, \ldots, x_N) = \frac{1}{2}(x_1 - \hat{x}_1)^T P_1^{-1}(x_1 - \hat{x}_1) + \frac{1}{2}\left(z_1 - h_1(x_1)\right)^T R_1^{-1}\left(z_1 - h_1(x_1)\right)
$$

Dimension Reduction: Affine Case

$$
\begin{aligned}
L(x_0,\ldots,x_N) \;&=\; \frac{1}{2}(x_0-\hat{x}_0)^T P_0^{-1}(x_0-\hat{x}_0) + \frac{1}{2}\left(z_0-h_0(x_0)\right)^T R_0^{-1}\left(z_0-h_0(x_0)\right) \\
&+\frac{1}{2}\left(x_1-g_1(x_0)\right)^T Q_1^{-1}\left(x_1-g_1(x_0)\right) \\
&+\frac{1}{2}\sum_{k=1}^{N-1} \begin{array}{c} \left(z_k-h_k(x_k)\right)^T R_k^{-1}\left(z_k-h_k(x_k)\right) \\ + \\ \left(x_{k+1}-g_k(x_k)\right)^T Q_{k+1}^{-1}\left(x_{k+1}-g_k(x_k)\right) \end{array}
\end{aligned}
$$

$$
\begin{aligned}
L_1(x_1,\ldots,x_N) \;&=\; \frac{1}{2}(x_1-\hat{x}_1)^T P_1^{-1}(x_1-\hat{x}_1) + \frac{1}{2}\left(z_1-h_1(x_1)\right)^T R_1^{-1}\left(z_1-h_1(x_1)\right) \\
&+\frac{1}{2}\left(x_2-g_2(x_1)\right)^T Q_2^{-1}\left(x_2-g_2(x_1)\right)
\end{aligned}
$$

Dimension Reduction: Affine Case

$$
\begin{aligned}
L(x_0,\ldots,x_N) \; &= \; \frac{1}{2}(x_0-\hat{x}_0)^T P_0^{-1}(x_0-\hat{x}_0) + \frac{1}{2}\left(z_0-h_0(x_0)\right)^T R_0^{-1}\left(z_0-h_0(x_0)\right) \\
&+\frac{1}{2}\left(x_1-g_1(x_0)\right)^T Q_1^{-1}\left(x_1-g_1(x_0)\right) \\
&+\frac{1}{2}\sum_{k=1}^{N-1} \begin{array}{c} \left(z_k-h_k(x_k)\right)^T R_k^{-1}\left(z_k-h_k(x_k)\right) \\ + \\ \left(x_{k+1}-g_k(x_k)\right)^T Q_{k+1}^{-1}\left(x_{k+1}-g_k(x_k)\right) \end{array}
\end{aligned}
$$

$$
\begin{aligned}
L_1(x_1,\ldots,x_N) \; &= \; \frac{1}{2}(x_1-\hat{x}_1)^T P_1^{-1}(x_1-\hat{x}_1) + \frac{1}{2}\left(z_1-h_1(x_1)\right)^T R_1^{-1}\left(z_1-h_1(x_1)\right) \\
&+\frac{1}{2}\left(x_2-g_2(x_1)\right)^T Q_2^{-1}\left(x_2-g_2(x_1)\right) \\
&+\frac{1}{2}\sum_{k=2}^{N-1} \begin{array}{c} \left(z_k-h_k(x_k)\right)^T R_k^{-1}\left(z_k-h_k(x_k)\right) \\ + \\ \left(x_{k+1}-g_k(x_k)\right)^T Q_{k+1}^{-1}\left(x_{k+1}-g_k(x_k)\right) \end{array}
\end{aligned}
$$

The Iterated Kalman-Bucy Smoother (Bell 94)

$$\text{minimize } \widehat{L}(x; y) \qquad \text{s.t.} \quad y \in \mathbb{R}^{nN}$$

The Iterated Kalman-Bucy Smoother (Bell 94)

$$\text{minimize } \widehat{L}(x; y) \qquad \text{s.t.} \quad y \in \mathbb{R}^{nN}$$

$$\begin{aligned} \widehat{L}_k(x; y) &= \frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0) \\ &+ \frac{1}{2}\sum_{k=1}^{N} \begin{array}{c} (y_k - \hat{g}_k(x_{k-1}; y_{k-1}))^T \, Q_k^{-1} \, (y_k - \hat{g}_k(x_{k-1}; y_{k-1})) \\ + \\ \left(z_k - \hat{h}_k(x_k; y_k)\right)^T R_k^{-1} \left(z_k - \hat{h}_k(x_k; y_k)\right) \end{array} \end{aligned}$$

The Iterated Kalman-Bucy Smoother (Bell 94)

$$\text{minimize } \widehat{L}(x; y) \qquad \text{s.t.} \quad y \in \mathbb{R}^{nN}$$

$$\begin{aligned}
\widehat{L}_k(x; y) &= \frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1} (x_0 - \hat{x}_0) \\
&\quad + \frac{1}{2} \sum_{k=1}^{N} \begin{array}{c} (y_k - \hat{g}_k(x_{k-1}; y_{k-1}))^T \, Q_k^{-1} \, (y_k - \hat{g}_k(x_{k-1}; y_{k-1})) \\ + \\ \left(z_k - \hat{h}_k(x_k; y_k)\right)^T R_k^{-1} \left(z_k - \hat{h}_k(x_k; y_k)\right) \end{array}
\end{aligned}$$

where

The Iterated Kalman-Bucy Smoother (Bell 94)

$$\text{minimize } \widehat{L}(x; y) \qquad \text{s.t.} \quad y \in \mathbb{R}^{nN}$$

$$\widehat{L}_k(x; y) = \frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0) + \frac{1}{2}\sum_{k=1}^{N} \begin{array}{c} (y_k - \hat{g}_k(x_{k-1}; y_{k-1}))^T \, Q_k^{-1} \, (y_k - \hat{g}_k(x_{k-1}; y_{k-1})) \\ + \\ \left(z_k - \hat{h}_k(x_k; y_k)\right)^T R_k^{-1} \left(z_k - \hat{h}_k(x_k; y_k)\right) \end{array}$$

where

$$\hat{g}_k(x_{k-1}; y_{k-1}) = g_k(x_{k-1}) + g_k'(x_{k-1})(y_{k-1} - x_{k-1})$$

The Iterated Kalman-Bucy Smoother (Bell 94)

$$\text{minimize } \widehat{L}(x; y) \qquad \text{s.t.} \quad y \in \mathbb{R}^{nN}$$

$$\begin{aligned}
\widehat{L}_k(x; y) \;&=\; \frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0) \\
&\quad + \frac{1}{2}\sum_{k=1}^{N} \begin{array}{c} (y_k - \hat{g}_k(x_{k-1}; y_{k-1}))^T \, Q_k^{-1} \, (y_k - \hat{g}_k(x_{k-1}; y_{k-1})) \\ + \\ \left(z_k - \hat{h}_k(x_k; y_k)\right)^T R_k^{-1} \left(z_k - \hat{h}_k(x_k; y_k)\right) \end{array}
\end{aligned}$$

where

$$\hat{g}_k(x_{k-1}; y_{k-1}) \;=\; g_k(x_{k-1}) + g_k'(x_{k-1})(y_{k-1} - x_{k-1})$$

$$\hat{h}_k(x_k; y_k) \;=\; h_k(x_k) + h_k'(x_k)(y_k - x_k)$$

A Large Scale Tridiagonal QP

$$\min \frac{1}{2} y^T C y + d^T y \qquad y \in \mathbb{R}^{nN}$$

A Large Scale Tridiagonal QP

$$\min \frac{1}{2} y^T C y + d^T y \qquad y \in \mathbb{R}^{nN}$$

$$C = \begin{bmatrix} C_1 & A_2^T & 0 & \\ A_2 & C_2 & A_3^T & 0 \\ 0 & \ddots & \ddots & \ddots \\ & 0 & A_N & C_N \end{bmatrix}$$

A Large Scale Tridiagonal QP

$$\min \frac{1}{2} y^T C y + d^T y \qquad y \in \mathbb{R}^{nN}$$

$$C = \begin{bmatrix} C_1 & A_2^T & 0 & \\ A_2 & C_2 & A_3^T & 0 \\ 0 & \ddots & \ddots & \ddots \\ & 0 & A_N & C_N \end{bmatrix}$$

$$A_k = -Q_k^{-1} g_k'(x_{k-1})$$

A Large Scale Tridiagonal QP

$$\min \frac{1}{2} y^T C y + d^T y \qquad y \in \mathbb{R}^{nN}$$

$$C = \begin{bmatrix} C_1 & A_2^T & 0 & \\ A_2 & C_2 & A_3^T & 0 \\ 0 & \ddots & \ddots & \ddots \\ & 0 & A_N & C_N \end{bmatrix}$$

$$\begin{aligned} A_k &= -Q_k^{-1} g_k'(x_{k-1}) \\ C_k &= Q_k + h_k'(x_k)^T R_k^{-1} h_k'(x_k) + g_{k+1}'(x_k)^T Q_{k+1}^{-1} g_{k+1}'(x_k) \end{aligned}$$

A Large Scale Tridiagonal QP

$$\min \frac{1}{2} y^T C y + d^T y \qquad y \in \mathbb{R}^{nN}$$

$$C = \begin{bmatrix} C_1 & A_2^T & 0 & \\ A_2 & C_2 & A_3^T & 0 \\ 0 & \ddots & \ddots & \ddots \\ & 0 & A_N & C_N \end{bmatrix}$$

$$A_k = -Q_k^{-1} g_k'(x_{k-1})$$

$$C_k = Q_k + h_k'(x_k)^T R_k^{-1} h_k'(x_k) + g_{k+1}'(x_k)^T Q_{k+1}^{-1} g_{k+1}'(x_k)$$

Solution: $\bar{y} = -C^{-1} d$

A Large Scale Tridiagonal QP

$$\min \frac{1}{2} y^T C y + d^T y \qquad y \in \mathbb{R}^{nN}$$

$$C = \begin{bmatrix} C_1 & A_2^T & 0 & \\ A_2 & C_2 & A_3^T & 0 \\ 0 & \ddots & \ddots & \ddots \\ & 0 & A_N & C_N \end{bmatrix}$$

$$\begin{aligned} A_k &= -Q_k^{-1} g_k'(x_{k-1}) \\ C_k &= Q_k + h_k'(x_k)^T R_k^{-1} h_k'(x_k) + g_{k+1}'(x_k)^T Q_{k+1}^{-1} g_{k+1}'(x_k) \end{aligned}$$

Solution: $\bar{y} = -C^{-1} d$
Efficient tridiagonal solvers are available, but better algorithms are needed.

The Constrained Model

The transition model:

$$
\begin{aligned}
x_k &= g_k(x_{k-1}) + w_k \quad (k = 1, \ldots, N) \\
x_0 &= \hat{x}_0 + e_0
\end{aligned}
$$

The Constrained Model

The transition model:

$$\begin{aligned} x_k &= g_k(x_{k-1}) + w_k \quad (k = 1, \ldots, N) \\ x_0 &= \hat{x}_0 + e_0 \end{aligned}$$

The measurement model:

$$z_k = h_k(x_k) + v_k \quad (k = 1, \ldots, N)$$

The constraints:

$$f_k(x_{k-1}, x_k) \leq 0, \quad \text{where } f_k : \mathbb{R}^n \rightarrow \mathbb{R}^\ell \quad (k = 1, \ldots, N)$$

The Constrained Model

The transition model:

$$
\begin{aligned}
x_k &= g_k(x_{k-1}) + w_k \quad (k = 1, \ldots, N) \\
x_0 &= \hat{x}_0 + e_0
\end{aligned}
$$

The measurement model:

$$
z_k = h_k(x_k) + v_k \quad (k = 1, \ldots, N)
$$

The constraints:

$$
f_k(x_{k-1}, x_k) \le 0, \quad \text{where } f_k : \mathbb{R}^n \to \mathbb{R}^\ell \quad (k = 1, \ldots, N)
$$

Goal: Develop a solver that uses the same linear algebraic tools as the unconstrained problem.

The Constrained Model

The transition model:

$$
\begin{aligned}
x_k &= g_k(x_{k-1}) + w_k \quad (k = 1, \ldots, N) \\
x_0 &= \hat{x}_0 + e_0
\end{aligned}
$$

The measurement model:

$$
z_k = h_k(x_k) + v_k \quad (k = 1, \ldots, N)
$$

The constraints:

$$
f_k(x_{k-1}, x_k) \leq 0, \quad \text{where } f_k : \mathbb{R}^n \to \mathbb{R}^\ell \quad (k = 1, \ldots, N)
$$

Goal: Develop a solver that uses the same linear algebraic tools as the unconstrained problem.
(Strongly PD tridiagonal systems)

The Constrained Gauss-Newton Model

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} y^T C y + d^T y \\ \text{subject to} & b + By + s = 0, \quad 0 \le s \end{array}$$

The Constrained Gauss-Newton Model

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}y^T C y + d^T y \\ \text{subject to} & b + By + s = 0, \quad 0 \le s \end{array}$$

C is tridiagonal.

The Constrained Gauss-Newton Model

$$\text{minimize} \quad \tfrac{1}{2} y^T C y + d^T y$$

$$\text{subject to} \quad b + By + s = 0, \quad 0 \leq s$$

C is tridiagonal.

$$B = \begin{bmatrix} D_1 f_1(x_1) & 0 & \cdots & 0 & 0 \\ D_1 f_2(x_1, x_2) & D_2 f_2(x_1, x_2) & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & D_{N-1} f_N(x_{N-1}, x_N) & D_N f_N'(x_{N-1}, x_N) \end{bmatrix}$$

The Constrained Gauss-Newton Model

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} y^T C y + d^T y \\ \text{subject to} & b + By + s = 0, \quad 0 \leq s \end{array}$$

C is tridiagonal.

$$B = \begin{bmatrix} D_1 f_1(x_1) & 0 & \cdots & 0 & 0 \\ D_1 f_2(x_1, x_2) & D_2 f_2(x_1, x_2) & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & D_{N-1} f_N(x_{N-1}, x_N) & D_N f'_N(x_{N-1}, x_N) \end{bmatrix}$$

B is a lower triangular bi-diagonal matrix.

QP First-Order Optimality Conditions

$$F(s, u, y) = 0 \qquad \text{with } 0 < s \text{ and } 0 < u$$

QP First-Order Optimality Conditions

$$F(s,u,y) = 0 \qquad \text{with } 0 < s \text{ and } 0 < u$$

where

$$F(s,u,y) = \begin{bmatrix} s + b + By \\ \\ \Lambda(s)\Lambda(u)e \\ \\ Cy + B^T u + d \end{bmatrix}$$

QP First-Order Optimality Conditions

$$F(s,u,y) = 0 \qquad \text{with } 0 < s \text{ and } 0 < u$$

where

$$F(s,u,y) = \begin{bmatrix} s + b + By \\ \\ \Lambda(s)\Lambda(u)e \\ \\ Cy + B^T u + d \end{bmatrix}$$

with

$$\Lambda(z) = \begin{bmatrix} z_1 & 0 & \\ 0 & \ddots & 0 \\ & 0 & z_N \end{bmatrix}$$

QP First-Order Optimality Conditions

$$F(s, u, y) = 0 \qquad \text{with } 0 < s \text{ and } 0 < u$$

where

$$F(s, u, y) = \begin{bmatrix} s + b + By \\ \\ \Lambda(s)\Lambda(u)e \\ \\ Cy + B^T u + d \end{bmatrix}$$

Primal Feasibility

with

$$\Lambda(z) = \begin{bmatrix} z_1 & 0 & \\ 0 & \ddots & 0 \\ & 0 & z_N \end{bmatrix}$$

QP First-Order Optimality Conditions

$$F(s,u,y) = 0 \qquad \text{with } 0 < s \text{ and } 0 < u$$

where

$$F(s,u,y) = \begin{bmatrix} s + b + By \\ \\ \Lambda(s)\Lambda(u)e \\ \\ Cy + B^T u + d \end{bmatrix} \qquad \begin{matrix} \text{Primal Feasibility} \\ \\ \text{Complementarity} \\ \\ \end{matrix}$$

with

$$\Lambda(z) = \begin{bmatrix} z_1 & 0 & \\ 0 & \ddots & 0 \\ & 0 & z_N \end{bmatrix}$$

QP First-Order Optimality Conditions

$$F(s, u, y) = 0 \qquad \text{with } 0 < s \text{ and } 0 < u$$

where

$$F(s, u, y) = \begin{bmatrix} s + b + By \\ \Lambda(s)\Lambda(u)e \\ Cy + B^T u + d \end{bmatrix} \qquad \begin{matrix} \text{Primal Feasibility} \\ \text{Complementarity} \\ \text{Stationarity} \end{matrix}$$

with

$$\Lambda(z) = \begin{bmatrix} z_1 & 0 & \\ 0 & \ddots & 0 \\ & 0 & z_N \end{bmatrix}$$

Relaxation and the Central Path

Let $\mu > 0$, and set $\mathbf{1}$ = vector of all ones.

Relaxation and the Central Path

Let $\mu > 0$, and set $\mathbf{1}$ = vector of all ones.
Consider solutions to the following eq. with $0 < s$ and $0 < u$:

$$\begin{bmatrix} 0 \\ \mu\mathbf{1} \\ 0 \end{bmatrix} = F(s,u,y) = \begin{bmatrix} s + b + By \\ \\ \Lambda(s)\Lambda(u)e \\ \\ Cy + B^T u + d \end{bmatrix}$$

Relaxation and the Central Path

Let $\mu > 0$, and set $\mathbf{1}$ = vector of all ones.
Consider solutions to the following eq. with $0 < s$ and $0 < u$:

$$\begin{bmatrix} 0 \\ \mu\mathbf{1} \\ 0 \end{bmatrix} = F(s, u, y) = \begin{bmatrix} s + b + By \\ \\ \Lambda(s)\Lambda(u)e \\ \\ Cy + B^T u + d \end{bmatrix}$$

The *central path* is the trajectory determined as the solutions of this equation with $0 < s$ and $0 < u$ as a function of $\mu > 0$:

Relaxation and the Central Path

Let $\mu > 0$, and set $\mathbf{1}$ = vector of all ones.
Consider solutions to the following eq. with $0 < s$ and $0 < u$:

$$\begin{bmatrix} 0 \\ \mu \mathbf{1} \\ 0 \end{bmatrix} = F(s, u, y) = \begin{bmatrix} s + b + By \\ \\ \Lambda(s)\Lambda(u)e \\ \\ Cy + B^T u + d \end{bmatrix}$$

The *central path* is the trajectory determined as the solutions of this equation with $0 < s$ and $0 < u$ as a function of $\mu > 0$:

$$(s(\mu), u(\mu), y(\mu))$$

Relaxation and the Central Path

Let $\mu > 0$, and set $\mathbf{1}$ = vector of all ones.
Consider solutions to the following eq. with $0 < s$ and $0 < u$:

$$\begin{bmatrix} 0 \\ \mu\mathbf{1} \\ 0 \end{bmatrix} = F(s, u, y) = \begin{bmatrix} s + b + By \\ \\ \Lambda(s)\Lambda(u)e \\ \\ Cy + B^T u + d \end{bmatrix}$$

The *central path* is the trajectory determined as the solutions of this equation with $0 < s$ and $0 < u$ as a function of $\mu > 0$:

$$(s(\mu), u(\mu), y(\mu))$$

We apply a predictor-corrector Newton method to follow this trajectory as $\mu \downarrow 0$.

A Primal-Dual Interior Point Approach

Let $\mu^j \downarrow 0$.

A Primal-Dual Interior Point Approach

Let $\mu^j \downarrow 0$.
Iteratively solve

$$F(s^j, u^j, y^j) + F'(s^j, u^j, y^j) \begin{bmatrix} \Delta s \\ \Delta u \\ \Delta y \end{bmatrix} = \begin{bmatrix} 0 \\ \mu^j \\ 0 \end{bmatrix}$$

A Primal-Dual Interior Point Approach

Let $\mu^j \downarrow 0$.
Iteratively solve

$$F(s^j, u^j, y^j) + F'(s^j, u^j, y^j) \begin{bmatrix} \Delta s \\ \Delta u \\ \Delta y \end{bmatrix} = \begin{bmatrix} 0 \\ \mu^j \\ 0 \end{bmatrix}$$

and set

$$\begin{aligned} s^j &= s^j + t\Delta s \; > 0 \\ u^j &= u^j + t\Delta u \; > 0 \\ y^j &= y^j + t\Delta y \; > 0 \end{aligned}$$

A Primal-Dual Interior Point Approach

Let $\mu^j \downarrow 0$.
Iteratively solve

$$F(s^j, u^j, y^j) + F'(s^j, u^j, y^j) \begin{bmatrix} \Delta s \\ \Delta u \\ \Delta y \end{bmatrix} = \begin{bmatrix} 0 \\ \mu^j \\ 0 \end{bmatrix}$$

and set

$$\begin{aligned} s^j &= s^j + t\Delta s \ > 0 \\ u^j &= u^j + t\Delta u \ > 0 \\ y^j &= y^j + t\Delta y \ > 0 \end{aligned}$$

$t > 0$ is a line search parameter

Row Reduce F' to a Tridiagonal System

The key to success is the structure of $F'(s, u, y)$:

Row Reduce F' to a Tridiagonal System

The key to success is the structure of $F'(s, u, y)$:

$$F'(s,u,y) = \begin{bmatrix} \mathrm{I} & 0 & B \\ \Lambda(u) & \Lambda(s) & 0 \\ 0 & B^T & C \end{bmatrix}$$

Row Reduce F' to a Tridiagonal System

The key to success is the structure of $F'(s, u, y)$:

$$F'(s,u,y) = \begin{bmatrix} \mathrm{I} & 0 & B \\ \Lambda(u) & \Lambda(s) & 0 \\ 0 & B^T & C \end{bmatrix}$$

Row reduce $F'(s, u, y)$:

Row Reduce F' to a Tridiagonal System

The key to success is the structure of $F'(s, u, y)$:

$$F'(s,u,y) = \begin{bmatrix} \mathrm{I} & 0 & B \\ \Lambda(u) & \Lambda(s) & 0 \\ 0 & B^T & C \end{bmatrix}$$

Row reduce $F'(s, u, y)$:

$$\begin{bmatrix} \mathrm{I} & 0 & B \\ \Lambda(u) & \Lambda(s) & 0 \\ 0 & B^T & C \end{bmatrix} \Longrightarrow$$

Row Reduce F' to a Tridiagonal System

The key to success is the structure of $F'(s, u, y)$:

$$F'(s,u,y) = \begin{bmatrix} \mathrm{I} & 0 & B \\ \Lambda(u) & \Lambda(s) & 0 \\ 0 & B^T & C \end{bmatrix}$$

Row reduce $F'(s, u, y)$:

$$\begin{bmatrix} \mathrm{I} & 0 & B \\ \Lambda(u) & \Lambda(s) & 0 \\ 0 & B^T & C \end{bmatrix} \Longrightarrow \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

Row Reduce F' to a Tridiagonal System

The key to success is the structure of $F'(s,u,y)$:

$$F'(s,u,y) = \begin{bmatrix} \mathrm{I} & 0 & B \\ \Lambda(u) & \Lambda(s) & 0 \\ 0 & B^T & C \end{bmatrix}$$

Row reduce $F'(s,u,y)$:

$$\begin{bmatrix} \mathrm{I} & 0 & B \\ \Lambda(u) & \Lambda(s) & 0 \\ 0 & B^T & C \end{bmatrix} \Longrightarrow \begin{bmatrix} \mathrm{I} & 0 & B & \\ & & & \\ & & & \end{bmatrix}$$

Row Reduce F' to a Tridiagonal System

The key to success is the structure of $F'(s, u, y)$:

$$F'(s,u,y) = \begin{bmatrix} \mathrm{I} & 0 & B \\ \Lambda(u) & \Lambda(s) & 0 \\ 0 & B^T & C \end{bmatrix}$$

Row reduce $F'(s, u, y)$:

$$\begin{bmatrix} \mathrm{I} & 0 & B \\ \Lambda(u) & \Lambda(s) & 0 \\ 0 & B^T & C \end{bmatrix} \Longrightarrow \begin{bmatrix} \mathrm{I} & 0 & B \\ 0 & \mathrm{I} & -\Lambda(s)^{-1}\Lambda(u)B \\ & & \end{bmatrix}$$

Row Reduce F' to a Tridiagonal System

The key to success is the structure of $F'(s,u,y)$:

$$F'(s,u,y) = \begin{bmatrix} \mathrm{I} & 0 & B \\ \Lambda(u) & \Lambda(s) & 0 \\ 0 & B^T & C \end{bmatrix}$$

Row reduce $F'(s,u,y)$:

$$\begin{bmatrix} \mathrm{I} & 0 & B \\ \Lambda(u) & \Lambda(s) & 0 \\ 0 & B^T & C \end{bmatrix} \Longrightarrow \begin{bmatrix} \mathrm{I} & 0 & B \\ 0 & \mathrm{I} & -\Lambda(s)^{-1}\Lambda(u)B \\ 0 & 0 & C + B^T\Lambda(s)^{-1}\Lambda(u)B \end{bmatrix}$$

What is the structure of $C + B^T \Lambda(s)^{-1} \Lambda(u) B$?

$$C + B^T \Lambda(s)^{-1} \Lambda(u) B =$$

What is the structure of $C + B^T \Lambda(s)^{-1} \Lambda(u) B$?

$$C + B^T \Lambda(s)^{-1} \Lambda(u) B = \begin{bmatrix} \hat{C}_1 & \hat{A}_2^T & 0 & \\ \hat{A}_2 & \hat{C}_2 & \hat{A}_3^T & 0 \\ 0 & \ddots & \ddots & \ddots \\ & 0 & \hat{A}_N & \hat{C}_N \end{bmatrix}$$

What is the structure of $C + B^T\Lambda(s)^{-1}\Lambda(u)B$?

$$C + B^T\Lambda(s)^{-1}\Lambda(u)B = \begin{bmatrix} \hat{C}_1 & \hat{A}_2^T & 0 & \\ \hat{A}_2 & \hat{C}_2 & \hat{A}_3^T & 0 \\ 0 & \ddots & \ddots & \ddots \\ & 0 & \hat{A}_N & \hat{C}_N \end{bmatrix}$$

where

What is the structure of $C + B^T \Lambda(s)^{-1} \Lambda(u) B$?

$$C + B^T \Lambda(s)^{-1} \Lambda(u) B = \begin{bmatrix} \hat{C}_1 & \hat{A}_2^T & 0 & \\ \hat{A}_2 & \hat{C}_2 & \hat{A}_3^T & 0 \\ 0 & \ddots & \ddots & \ddots \\ & 0 & \hat{A}_N & \hat{C}_N \end{bmatrix}$$

where

$$\begin{aligned} \hat{C}_k &= C_k + D_{k-1} f_k(x_{k-1}, x_k)^T \Lambda(s_{k-1})^{-1} \Lambda(u_{k-1}) D_{k-1} f_k(x_{k-1}, x_k) \\ &\quad + D_k f_k(x_{k-1}, x_k)^T \Lambda(s_k)^{-1} \Lambda(u_k) D_k f_k(x_{k-1}, x_k) \end{aligned}$$

What is the structure of $C + B^T \Lambda(s)^{-1} \Lambda(u) B$?

$$C + B^T \Lambda(s)^{-1} \Lambda(u) B = \begin{bmatrix} \hat{C}_1 & \hat{A}_2^T & 0 & \\ \hat{A}_2 & \hat{C}_2 & \hat{A}_3^T & 0 \\ 0 & \ddots & \ddots & \ddots \\ & 0 & \hat{A}_N & \hat{C}_N \end{bmatrix}$$

where

$$\begin{aligned} \hat{C}_k &= C_k + D_{k-1} f_k(x_{k-1}, x_k)^T \Lambda(s_{k-1})^{-1} \Lambda(u_{k-1}) D_{k-1} f_k(x_{k-1}, x_k) \\ &\quad + D_k f_k(x_{k-1}, x_k)^T \Lambda(s_k)^{-1} \Lambda(u_k) D_k f_k(x_{k-1}, x_k) \end{aligned}$$

and

What is the structure of $C + B^T \Lambda(s)^{-1} \Lambda(u) B$?

$$C + B^T \Lambda(s)^{-1} \Lambda(u) B = \begin{bmatrix} \hat{C}_1 & \hat{A}_2^T & 0 & \\ \hat{A}_2 & \hat{C}_2 & \hat{A}_3^T & 0 \\ 0 & \ddots & \ddots & \ddots \\ & 0 & \hat{A}_N & \hat{C}_N \end{bmatrix}$$

where

$$\begin{aligned} \hat{C}_k &= C_k + D_{k-1} f_k(x_{k-1}, x_k)^T \Lambda(s_{k-1})^{-1} \Lambda(u_{k-1}) D_{k-1} f_k(x_{k-1}, x_k) \\ &\quad + D_k f_k(x_{k-1}, x_k)^T \Lambda(s_k)^{-1} \Lambda(u_k) D_k f_k(x_{k-1}, x_k) \end{aligned}$$

and

$$\hat{A}_k = A_k + D_{k-1} f_k(x_{k-1}, x_k) \Lambda(s_{k-1})^{-1} \Lambda(u_{k-1}) D_{k-1} f_{k-1}(x_{k-1}, x_k)^T$$

A Toy Ship tracking Example

Track a ship traveling close to shore.

A Toy Ship tracking Example

Track a ship traveling close to shore.

Given:

A Toy Ship tracking Example

Track a ship traveling close to shore.

Given:

- The distance measurements from two fixed on-shore stations to the ship, with error:

A Toy Ship tracking Example

Track a ship traveling close to shore.

Given:

- The distance measurements from two fixed on-shore stations to the ship, with error: $(0, 0)$ and $(2\pi, 0)$.

A Toy Ship tracking Example

Track a ship traveling close to shore.

Given:

- The location of the shoreline:

- The distance measurements from two fixed on-shore stations to the ship, with error: $(0, 0)$ and $(2\pi, 0)$.

A Toy Ship tracking Example

Track a ship traveling close to shore.

Given:

- The location of the shoreline: The graph of $1.25 - \sin(t)$.

- The distance measurements from two fixed on-shore stations to the ship, with error: $(0, 0)$ and $(2\pi, 0)$.

A Toy Ship tracking Example

Track a ship traveling close to shore.

Given:

- The location of the shoreline: The graph of $1.25 - \sin(t)$.

  (u, v) is on the shore if $v = 1.25 - \sin(u)$.

- The distance measurements from two fixed on-shore stations to the ship, with error: $(0, 0)$ and $(2\pi, 0)$.

A Toy Ship tracking Example

Track a ship traveling close to shore.

Given:

- The location of the shoreline: The graph of $1.25 - \sin(t)$.

  (u, v) is on the shore if $v = 1.25 - \sin(u)$.

  (u, v) is off-shore if $v \geq 1.25 - \sin(u)$.

- The distance measurements from two fixed on-shore stations to the ship, with error: $(0, 0)$ and $(2\pi, 0)$.

A Toy Ship tracking Example

Track a ship traveling close to shore.

Given:

- The location of the shoreline: The graph of $1.25 - \sin(t)$.

  (u, v) is on the shore if $v = 1.25 - \sin(u)$.

  (u, v) is off-shore if $v \geq 1.25 - \sin(u)$.

  Thus, an off-shore constraint is given by

$$f(u, v) = 1.25 - \sin(u) - v \leq 0 \ .$$

- The distance measurements from two fixed on-shore stations to the ship, with error: $(0, 0)$ and $(2\pi, 0)$.

Dynamics

Dynamics

$$X_2(t) = t \quad \text{first component of position}$$

$$X_4(t) = 1.3 - \sin(t) \quad \text{second component of position}$$

Dynamics

$$X_1(t) = 1$$ first component of velocity

$$X_2(t) = t$$ first component of position

$$X_4(t) = 1.3 - \sin(t)$$ second component of position

Dynamics

$$\begin{aligned}
X_1(t) &= 1 && \text{first component of velocity} \\
X_2(t) &= t && \text{first component of position} \\
X_3(t) &= -\cos(t) && \text{second component of velocity} \\
X_4(t) &= 1.3 - \sin(t) && \text{second component of position}
\end{aligned}$$

Dynamics

$$\begin{aligned} X_1(t) &= 1 && \text{first component of velocity} \\ X_2(t) &= t && \text{first component of position} \\ X_3(t) &= -\cos(t) && \text{second component of velocity} \\ X_4(t) &= 1.3 - \sin(t) && \text{second component of position} \end{aligned}$$

Model the velocity error components as independent Brownian motions plus an initial velocity.

Dynamics

$$
\begin{aligned}
X_1(t) &= 1 && \text{first component of velocity} \\
X_2(t) &= t && \text{first component of position} \\
X_3(t) &= -\cos(t) && \text{second component of velocity} \\
X_4(t) &= 1.3 - \sin(t) && \text{second component of position}
\end{aligned}
$$

Model the velocity error components as independent Brownian motions plus an initial velocity.

Model observed position as the integral of the velocity error plus an initial location.

Discrete Transition Model

Approximate the stochastic integral by Euler's method with constant spacing Δt. This gives the discrete process.

Discrete Transition Model

Approximate the stochastic integral by Euler's method with constant spacing Δt. This gives the discrete process.
The transition model becomes

$$g_k(x_{k-1}) = \begin{pmatrix} x_{1,k-1} \\ x_{2,k-1} + x_{1,k-1}\Delta t \\ x_{3,k-1} \\ x_{4,k-1} + x_{4,k-1}\Delta t \end{pmatrix}$$

with

$$Q_k = \begin{pmatrix} \Delta t & \Delta t^2/2 & 0 & 0 \\ \Delta t^2/2 & \Delta t^3/3 & 0 & 0 \\ 0 & 0 & \Delta t & \Delta t^2/2 \\ 0 & 0 & \Delta t^2/2 & \Delta t^3/3 \end{pmatrix} .$$

Discrete Transition Model

Approximate the stochastic integral by Euler's method with constant spacing Δt. This gives the discrete process.
The transition model becomes

$$g_k(x_{k-1}) = \begin{pmatrix} x_{1,k-1} \\ x_{2,k-1} + x_{1,k-1}\Delta t \\ x_{3,k-1} \\ x_{4,k-1} + x_{4,k-1}\Delta t \end{pmatrix}$$

with

$$Q_k = \begin{pmatrix} \Delta t & \Delta t^2/2 & 0 & 0 \\ \Delta t^2/2 & \Delta t^3/3 & 0 & 0 \\ 0 & 0 & \Delta t & \Delta t^2/2 \\ 0 & 0 & \Delta t^2/2 & \Delta t^3/3 \end{pmatrix} .$$

Initialize with

$$g_1(x_0) = X(t_1)\ ,\ Q_1 = 100\,\mathrm{I}_{4\times 4}\ .$$

Measurement Model

Constant variance σ^2.

Shore distance measurement locations $(0, 0)$ and $(2\pi, 0)$.

$$h_k(x_k) = \begin{pmatrix} \sqrt{x_{2,k}^2 + x_{4,k}^2} \\ \\ \sqrt{(x_{2,k} - 2\pi)^2 + x_{4,k}^2} \end{pmatrix}$$

and

$$R_k = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix} .$$

Constraints

The ship cannot cross land:

$$X_4(t) \geq 1.25 - \sin[X_2(t)]$$

so

$$f_k(x_k) = 1.25 - \sin(x_{2,k}) - x_{4,k} \ .$$

Initialization Data

$$N = 50 \qquad \text{number of measurement times}$$

Initialization Data

$N = 50$		number of measurement times
$\Delta t = 2\pi/N$		spacing between time points

Initialization Data

$N = 50$		number of measurement times
$\Delta t = 2\pi/N$		spacing between time points
$t_k = k\Delta t$		time corresponding to k-th measurement

Initialization Data

$$\begin{aligned}
N &= 50 && \text{number of measurement times} \\
\Delta t &= 2\pi/N && \text{spacing between time points} \\
t_k &= k\Delta t && \text{time corresponding to k-th measurement} \\
\sigma &= .25 && \text{standard deviation of measurement noise}
\end{aligned}$$

Initialization Data

$N = 50$		number of measurement times
$\Delta t = 2\pi/N$		spacing between time points
$t_k = k\Delta t$		time corresponding to k-th measurement
$\sigma = .25$		standard deviation of measurement noise
$v_k \sim N(0, R_k)$		simulated measurement noise

Initialization Data

N	$=$	50	number of measurement times
Δt	$=$	$2\pi/N$	spacing between time points
t_k	$=$	$k\Delta t$	time corresponding to k-th measurement
σ	$=$	$.25$	standard deviation of measurement noise
v_k	$\sim$	$N(0, R_k)$	simulated measurement noise
z_k	$=$	$h_k[X(t_k)] + v_k$	simulated measurement value

Initialization Data

$$
\begin{aligned}
N &= 50 && \text{number of measurement times}\\
\Delta t &= 2\pi/N && \text{spacing between time points}\\
t_k &= k\Delta t && \text{time corresponding to k-th measurement}\\
\sigma &= .25 && \text{standard deviation of measurement noise}\\
v_k &\sim N(0, R_k) && \text{simulated measurement noise}\\
z_k &= h_k[X(t_k)] + v_k && \text{simulated measurement value}
\end{aligned}
$$

Initial state: $x_0 = (0, 0, 0, 1)^T$.

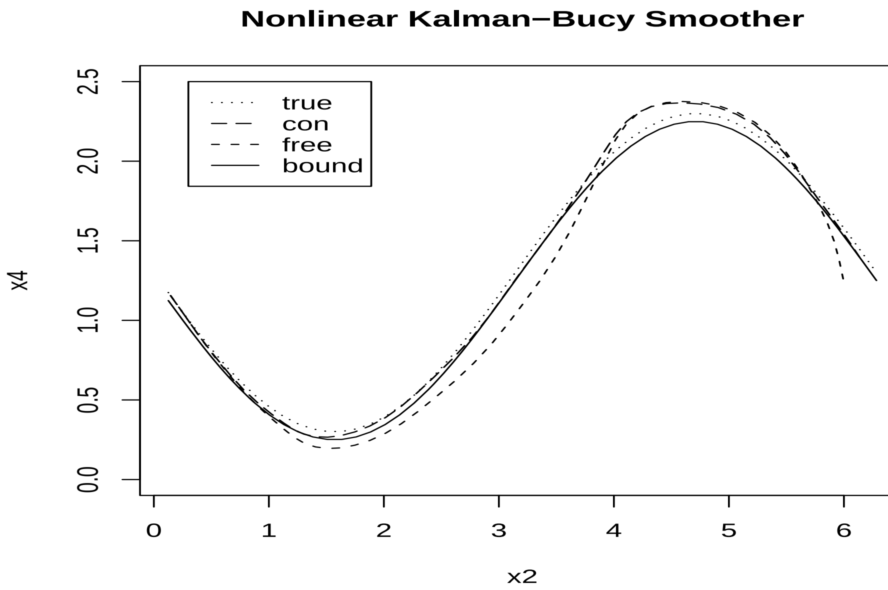
Initialization Data

$$
\begin{array}{lcll}
N & = & 50 & \text{number of measurement times} \\
\Delta t & = & 2\pi/N & \text{spacing between time points} \\
t_k & = & k\Delta t & \text{time corresponding to k-th measurement} \\
\sigma & = & .25 & \text{standard deviation of measurement noise} \\
v_k & \sim & N(0, R_k) & \text{simulated measurement noise} \\
z_k & = & h_k[X(t_k)] + v_k & \text{simulated measurement value}
\end{array}
$$

Initial state: $x_0 = (0, 0, 0, 1)^T$.

Initial constraint violation:

$$f_k(x_k^0) = 1.25 - \sin(x_{2,k}) - x_{4,k} = .25 > 0$$



Nonlinear Kalman−Bucy Smoother
true
con
free
bound
x4
x2
0.0
0.5
1.0
1.5
2.0
2.5
0
1
2
3
4
5
6

References

Bell, B.M., The iterated Kalman smoother as a Gauss-Newton method, *SIAM Journal on Optimization*, 4(3): 626-636, 1994.

Bell, B.M., The marginal likelihood for parameters in a discrete Markov process, *IEEE Transactions on Signal Processing*, 48(2000) 870-873.

Bell, B.M., J.V. Burke, G. Pillonetto, An inequality constrained nonlinear Kalman-Bucy smoother by interior point likelihood maximization, *Automatica*, 45:25-33, 2009.

The Nonlinear Algorithm

An SQP algorithm with ℓ_1 penalty linesearch.

$$\phi(x) = \sum_{k=1}^{N} \sum_{i=1}^{\ell} \max[f_k(x_k)_i, 0],$$

$$\tilde{\phi}(x; y) = \sum_{k=1}^{N} \sum_{i=1}^{\ell} \max[f_k(x_k)_i + f_k'(x_k)_i(y_k - x_k), 0]$$

Step 0: Initialization: Choose $x^0 \in \mathbb{R}^{nN}$ and $\alpha_0 > 0$. Set $\nu = 0$.

Step 1: Solve (inexactly) the QP subproblem for y^ν.

Step 2: Check convergence criteria.

Step 3: (Update the Penalty Parameter) Set $\hat{\alpha}_\nu = \alpha_\nu$. Define the value

$$\zeta_\nu = (y^\nu - x^\nu)^T C^\nu (y^\nu - x^\nu) + (a^\nu)^T (y^\nu - x^\nu)$$

If $\zeta_\nu \leq \hat{\alpha}_\nu \phi(x^\nu)$, set $\alpha_{\nu+1} = \hat{\alpha}_\nu$; otherwise, $\alpha_{\nu+1} = \max[\zeta_\nu / \phi(x^\nu)\ ,\ 2\hat{\alpha}_\nu]$.

Step 4: Compute the line search step size λ_p:

$$\begin{aligned}
\eta_\nu &= (a^\nu)^T(y^\nu - x^\nu) + \alpha_{\nu+1}[\tilde{\phi}(x^\nu; y^\nu) - \phi(x^\nu)] \\
H_\nu(\lambda) &= L[x^\nu + \lambda(y^\nu - x^\nu)] + \alpha_{\nu+1}\phi[x^\nu + \lambda(y^\nu - x^\nu)] \\
\lambda_n u &= \max\left\{2^{-q} \;\middle|\; q \in \mathbb{Z}_+ \text{ and } H_\nu(2^{-q}) - H_\nu(0) \le 2^{-q}\eta_\nu/10\right.
\end{aligned}$$

Step 5: Set $x^{\nu+1} = x^\nu + \lambda_\nu(y^\nu - x^\nu)$, then set $\nu = \nu + 1$ and go to Step 1.

Convergence Assumptions

1. The QP subproblems are all feasible. (Not strictly required)
2. The sequence $\{y^\nu\}$ is bounded.
3. The matrices $g_k'(x_{k-1}^\nu)$ are all invertible.
4. All cluster points of $\{x^\nu\}$ satisfy the generalized MFCQ.

Convergence Theorem

- The sequences
$$\{x^\nu\} \quad \text{and} \quad \{C_\nu\}$$
are bounded.
- $\{C_\nu\}$ is contained in a compact set of real symmetric positive definite matrices.
- Every cluster point is a KKT point.