Contents

1 Introduction

- Curse of dimensionality: exponential growth of computational cost with RAUTE measurements

- Moving Horizon estimation: Only consider measurements in a certain time horizon

2 Problem formulation

- Consider problem with state and output noise

Classical state and parameter estimation problem

$$
\begin{aligned}
\min_{\boldsymbol{x},\, \boldsymbol{z},\, \boldsymbol{p},\, \boldsymbol{w}} \quad & \sum_{j=1}^{m} ||\boldsymbol{y}_j - \boldsymbol{h}(\boldsymbol{x}(\tau_j), \boldsymbol{z}(\tau_j), \boldsymbol{p})||_{V_j}^2 + \int_{t_0}^{T} ||\boldsymbol{w}(t)||_W^2 \; \mathrm{d}t \\
\text{s.t.} \quad & \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}) + \boldsymbol{w}(t), \qquad t \in [t_0, T], \\
& \boldsymbol{x}_0 = \boldsymbol{x}(t_0), \\
& \boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}), \qquad t \in [t_0, T].
\end{aligned}
$$

+ possibly further constraints

- $||\boldsymbol{r}||_A := \boldsymbol{r}^T A^T A \boldsymbol{r}$

- Matrices usually s.t. $A^T A = \sigma^{-1}$, σ variance matrix

2.1 Moving Horizon approach

- Only consider measurements in $[\hat{t} - t_{\mathrm{M}}, \hat{t}]$
- i.e. $\boldsymbol{y}_j, j = L := k - M + 1, ...k$
- +++: Problem mit angepasster Summe und Integral anschreiben
- Wish to incorporate old information, too -> Arrival cost: C_L

Continuous formulation

$$\begin{aligned} \min_{\boldsymbol{x},\ \boldsymbol{z},\ \boldsymbol{p},\ \boldsymbol{w}} \quad & C_L + \sum_{j=L}^{k} ||\boldsymbol{y}_j - \boldsymbol{h}(\boldsymbol{x}(\tau_j), \boldsymbol{z}(\tau_j), \boldsymbol{p})||_{V_j}^2 + \int_{\hat{t}-t_{\mathrm{M}}}^{\hat{t}} ||\boldsymbol{w}(t)||_W^2 \ \mathrm{d}t \\ \text{s.t.} \quad & \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}) + \boldsymbol{w}(t), \qquad t \in [\hat{t} - t_{\mathrm{M}}, \hat{t}], \\ & \boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}), \qquad t \in [\hat{t} - t_{\mathrm{M}}, \hat{t}]. \end{aligned}$$

+ possibly further constraints

- Usual choice for arrival cost:

$$C_j = \left|\left| \begin{array}{c} \boldsymbol{x}_j - \bar{\boldsymbol{x}}_j \\ \boldsymbol{p}_j - \bar{\boldsymbol{p}}_j \end{array} \right|\right|_{P_j}^2 .$$

- choices of $\bar{\boldsymbol{x}}_L, \bar{\boldsymbol{p}}_L, P_L$ crucial
- Problem is infinite dimensional -> needs discretization:
  - measurement times τ_j as time mesh
  - replace equation for $\dot{x}$ by update scheme

$$\boldsymbol{x}_{j+1} = \boldsymbol{F}(\boldsymbol{x}_j, \boldsymbol{z}_j, \boldsymbol{u}_j, \boldsymbol{p}) + \boldsymbol{w}_j, \ \ j = L, \ldots, k-1$$

where F iteration function (mostly derived from numerical integration)

Discrete formulation

$$\begin{aligned} \min_{\boldsymbol{x}_j,\ \boldsymbol{z}_j,\ \boldsymbol{p},\ \boldsymbol{w}_j} \quad & \left(\left|\left| \begin{array}{c} \boldsymbol{x}_L - \bar{\boldsymbol{x}}_L \\ \boldsymbol{p}_L - \bar{\boldsymbol{p}}_L \end{array} \right|\right|_{P_L}^2 + \sum_{j=L}^{k} ||\boldsymbol{y}_j - \boldsymbol{h}(\boldsymbol{x}(\tau_j), \boldsymbol{z}(\tau_j), \boldsymbol{p})||_{V_j}^2 + \sum_{j=L}^{k-1} ||\boldsymbol{w}_j||_W^2 \right) \\ \text{s.t.} \quad & \boldsymbol{x}_{j+1} = \boldsymbol{F}(\boldsymbol{x}_j, \boldsymbol{z}_j, \boldsymbol{u}_j, \boldsymbol{p}) + \boldsymbol{w}_j, \qquad j = L, \ldots, k-1, \\ & \boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}_j, \boldsymbol{z}_j, \boldsymbol{u}_j, \boldsymbol{p}), \qquad j = L, \ldots, k, \\ & \boldsymbol{x}_{j,\min} \leq \boldsymbol{x}_j \leq \boldsymbol{x}_{j,\max}, \qquad j = L, \ldots, k, \\ & \boldsymbol{z}_{j,\min} \leq \boldsymbol{z}_j \leq \boldsymbol{z}_{j,\max}, \qquad j = L, \ldots, k, \\ & \boldsymbol{w}_{j,\min} \leq \boldsymbol{w}_j \leq \boldsymbol{w}_{j,\max}, \qquad j = L, \ldots, k, \\ & \boldsymbol{p}_{\min} \leq \boldsymbol{p} \leq \boldsymbol{p}_{\max}. \end{aligned}$$

Designing the arrival cost

- Aim: Find C_{L+1} s. t. problem on $[\tau_{L+1}, \tau_{k+1}]$ is equivalent to problem on $[\tau_L, \tau_{k+1}]$
- Aim: C_{L+1} does not grow exceedingly
- -> Incorporate C_L, y_L, w_L into C_{L+1}
- Allow for parameter noise: $p_{j+1} = p_j + w_j^p$ -> new weighting matrix $\bar{W}$
- -> [MHE] decouples -> dynamic programming arguments apply
- -> ideal arrival cost C^*_{L+1} must fulfil

$$
\begin{aligned}
C^*_{L+1} = \min_{\boldsymbol{x}_L,\, \boldsymbol{p}_L} & \left(\left\| \begin{matrix} \boldsymbol{x}_L - \bar{\boldsymbol{x}}_L \\ \boldsymbol{p}_L - \bar{\boldsymbol{p}}_L \end{matrix} \right\|^2_{P_L} + \|\boldsymbol{y}_L - \boldsymbol{h}(\boldsymbol{x}(\tau_L), \boldsymbol{p}_L)\|^2_{V_L} + \left\| \begin{matrix} \boldsymbol{w}_L \\ \boldsymbol{w}_L^p \end{matrix} \right\|^2_{\bar{W}_L} \right) \\
\text{s.t.} \qquad & \boldsymbol{w}_L = \boldsymbol{x}_{L+1} - \boldsymbol{F}(\boldsymbol{x}_L, \boldsymbol{u}_L, \boldsymbol{p}_L), \\
& \boldsymbol{w}_L^p = \boldsymbol{p}_{L+1} - \boldsymbol{p}_L.
\end{aligned}
$$

- $x_{L+1} = x(\tau_{L+1}), p_{L+1}$ known from solution on $[\tau_{L+1}, \tau_{k+1}]$
- -> w_L not free anymore
- z_L can be expressed by x_l, p_L

Problem: h and x are still nonlinear -> would have to solve for $C*_{L+1}$ iteratively
Remedy: Linearize h and x around estimation $x*, p*$ from last iterate:

$$
\begin{aligned}
\boldsymbol{x}(\tau_{L+1}; \boldsymbol{x}_l, \boldsymbol{p}_L) \approx \boldsymbol{x}^*(\tau_{L+1}; \boldsymbol{x}^*(\tau_L), \boldsymbol{p}^*) + & \underbrace{\frac{d\boldsymbol{x}(\tau_{L+1}; \boldsymbol{x}^*(\tau_L), \boldsymbol{p}^*)}{d\boldsymbol{x}(\tau_L)}}_{:=X_x} (\boldsymbol{x}_L - \boldsymbol{x}^*(\tau_L)) \\
& + \underbrace{\frac{d\boldsymbol{x}(\tau_{L+1}; \boldsymbol{x}^*(\tau_L), \boldsymbol{p}^*)}{d\boldsymbol{p}}}_{:=X_p} (\boldsymbol{p}_L - \boldsymbol{p}^*) \\
& := \tilde{\boldsymbol{x}} + X_x \boldsymbol{x}_L + X_p \boldsymbol{p}_L
\end{aligned}
$$

and analogously

$$
\boldsymbol{h}(\boldsymbol{x}_L, \boldsymbol{z}_L, \boldsymbol{p}_L) \approx \tilde{\boldsymbol{h}} + H_x \boldsymbol{x}_L + H_p \boldsymbol{p}_L.
$$

IMPORTANT: The arr. cost we find that way is then also an approximation to the ideal one!

This transforms [ACP] to:

$$\min_{\boldsymbol{x}_L,\ \boldsymbol{p}_L} \left(\left\| P_L \begin{pmatrix} c\boldsymbol{x}_L - \bar{\boldsymbol{x}}_L \\ \boldsymbol{p}_L - \bar{\boldsymbol{p}}_L \end{pmatrix} \right\|_2^2 + \|V_L(\boldsymbol{y}_L - \tilde{h} - H_x x_L - H_p p_L)\|_2^2 + \left\| \bar{W}_L \begin{pmatrix} c x_{L+1} - \tilde{x} - X_x x_L - X_p p_L \\ p_{L+1} - p_L \end{pmatrix} \right\|_2^2$$

which we can write equivalently as

$$\min_{\boldsymbol{x}_L,\ \boldsymbol{p}_L} \left\| \underbrace{\begin{pmatrix} -P_L \begin{pmatrix} \bar{\boldsymbol{x}}_L \\ \bar{\boldsymbol{p}}_L \end{pmatrix} \\ V_L(\boldsymbol{y}_L - \tilde{\boldsymbol{h}}) \\ \bar{W}_L \begin{pmatrix} \tilde{\boldsymbol{x}} \\ 0 \end{pmatrix} \end{pmatrix}}_{:=b} + \underbrace{\left(\begin{array}{c|c} P_L & 0 \\ -(V_L H_X \mid V_L H_p) & 0 \\ -\bar{W}_L \begin{pmatrix} X_X & X_p \\ 0 & \mathbb{I} \end{pmatrix} & \bar{W}_L \end{array} \right)}_{:=M_{\mathrm{QR}}} \cdot \begin{pmatrix} \boldsymbol{x}_L \\ \boldsymbol{p}_L \\ x_{L+1} \\ \boldsymbol{p}_{L+1} \end{pmatrix} \right\|_2^2 .$$

Use QR-decomposition to obtain analytic solution:

$$M_{\mathrm{QR}} = Q \begin{pmatrix} \mathcal{R}_1 & \mathcal{R}_{12} \\ 0 & \mathcal{R}_2 \\ 0 & 0 \end{pmatrix}$$

Set

$$\begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{pmatrix} := Q^T b$$

With this the solution is

$$\begin{pmatrix} \boldsymbol{x}_L \\ \boldsymbol{p}_L \end{pmatrix} = -\mathcal{R}_1^{-1} \left(\rho_1 + \mathcal{R}_{12} \begin{pmatrix} \boldsymbol{x}_{L+1} \\ \boldsymbol{p}_{L+1} \end{pmatrix} \right),$$

with the optimal cost given as

$$C'(\boldsymbol{x}_{L+1}, \boldsymbol{p}_{L+1}) = \left\| \rho_3 \right\|_2^2 + \left\| \rho_2 + \mathcal{R}_2 \begin{pmatrix} \boldsymbol{x}_{L+1} \\ \boldsymbol{p}_{L+1} \end{pmatrix} \right\|_2^2 .$$

Reminder: We want

$$C_{L+1} = \left\| \begin{array}{c} \boldsymbol{x}_{L+1} - \bar{\boldsymbol{x}}_{L+1} \\ \boldsymbol{p}_{L+1} - \bar{\boldsymbol{p}}_{L+1} \end{array} \right\|_{P_{L+1}}^2 .$$

We choose:

$$P_{L+1} := \mathcal{R}_2, \quad \begin{pmatrix} \bar{\boldsymbol{x}}_{L+1} \\ \bar{\boldsymbol{p}}_{L+1} \end{pmatrix} = -\mathcal{R}_2^{-1} \rho_2 .$$

Because then

$$\left\|\rho_2 + \mathcal{R}_2 \begin{pmatrix} \boldsymbol{x}_{L+1} \\ \boldsymbol{p}_{L+1} \end{pmatrix}\right\|_2^2 = \left\|\mathcal{R}_2^{-1}\rho_2 + \mathcal{R}_2^{-1}\mathcal{R}_2 \begin{pmatrix} \boldsymbol{x}_{L+1} \\ \boldsymbol{p}_{L+1} \end{pmatrix}\right\|_{\mathcal{R}_2}^2 = \left\| \begin{matrix} \boldsymbol{x}_{L+1} - \bar{\boldsymbol{x}}_{L+1} \\ \boldsymbol{p}_{L+1} - \bar{\boldsymbol{p}}_{L+1} \end{matrix} \right\|_{P_{L+1}}^2 .$$

Question: Does C_{L+1} influence grow exceedingly? **No!** Proof:

$$M^T M = \begin{pmatrix} *^1 & *^2 \\ *^3 & \bar{W}_L^T \bar{W}_L \end{pmatrix} \overset{\text{QR}}{=} \begin{pmatrix} \mathcal{R}_1^T \mathcal{R}_1 & \mathcal{R}_1^T \mathcal{R}_{12} \\ \mathcal{R}_{12}^T \mathcal{R}_1 & \mathcal{R}_{12}^T \mathcal{R}_{12} + \mathcal{R}_2^T \mathcal{R}_2 \end{pmatrix}$$

=>

$$\|v\|_{P_{L+1}}^2 = v^T P_{L+1}^T P_{L+1} v \overset{\text{our choice}}{=} v^T \mathcal{R}_2^T \mathcal{R}_2 v \leq \underbrace{v^T \mathcal{R}_{12}^T \mathcal{R}_{12} v}_{\geq 0} + v^T \mathcal{R}_2^T \mathcal{R}_2 v = \|v\|_{\bar{W}_L}^2$$

Resemblance to EKF:

3 Numerical solution

3.1 Direct multiple shooting

We consider:

$$\begin{aligned}
&\min_{\boldsymbol{u}\in U,\ \boldsymbol{x},\ \boldsymbol{z},\ \boldsymbol{p}} && J(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}) \\
&\text{s.t.} && \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}), \quad t \in [t_0, t_0 + T], \\
& && \boldsymbol{x}_0 = \boldsymbol{x}(t_0), \\
& && \boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}), \quad t \in [t_0, t_0 + T], \\
& && \boldsymbol{0} \geq \boldsymbol{r}(\boldsymbol{x}(t), \boldsymbol{z}(t), \boldsymbol{u}(t), \boldsymbol{p}), \quad t \in [t_0, t_0 + T].
\end{aligned}$$

With the multiple shooting approach we obtain the NLP

$$\begin{aligned}
&\min_{\boldsymbol{\xi}} \ J(\boldsymbol{\xi}) \\
&\text{s.t.} \ \boldsymbol{0} = \boldsymbol{x}(\tau_{j+1}; \boldsymbol{s}_j^x, \boldsymbol{s}_j^z, \boldsymbol{q}_j) - \boldsymbol{s}_{j+1}^x, \quad j = 0, \dots, m-1, \\
&\quad\ \ \boldsymbol{0} = \boldsymbol{x}(t_0) - \boldsymbol{s}_0^x, \\
&\quad\ \ \boldsymbol{0} = g(\boldsymbol{s}_j^x, \boldsymbol{s}_j^z, \boldsymbol{q}_j, \boldsymbol{p}), \quad j = 0, \dots, m, \\
&\quad\ \ \boldsymbol{0} \geq \boldsymbol{r}(\boldsymbol{s}_j^x, \boldsymbol{s}_j^z, \boldsymbol{q}_j, \boldsymbol{p}), \quad j = 0, \dots, m.
\end{aligned}$$

3.2 DMS as framework for MHE

- Sampling times as shooting nodes
- Replace J with MHE cost function
- First constraints essentially identical
- Controls actually known -> instead: identify state noise w_j as q_j

Even more notation:

$$\begin{aligned}\boldsymbol{r}_k &= (\boldsymbol{x}_L, \boldsymbol{z}_L, \dots, \boldsymbol{x}_k, \boldsymbol{z}_k, \boldsymbol{w}_L, \dots, \boldsymbol{w}_{k-1}, \boldsymbol{p}) -> \textit{optimization variables}\\ \boldsymbol{D}_k &= (\bar{\boldsymbol{x}}_L, \bar{\boldsymbol{p}}_L, P_L, \boldsymbol{y}_L, V_L, \boldsymbol{u}_L, \dots, \boldsymbol{y}_k, V_k, \boldsymbol{u}_k) -> \textit{input data}\end{aligned}$$

In short we now write the NLP as:

$$\begin{aligned}&\min_{\boldsymbol{r}_k} \ \left\| J(\boldsymbol{r}_k; \boldsymbol{D}_k) \right\|_2^2\\ &\text{s.t.} \ \ \boldsymbol{0} = \boldsymbol{G}(\boldsymbol{r}_k; \boldsymbol{D}_k),\\ &\qquad \boldsymbol{0} \leq \boldsymbol{H}(\boldsymbol{r}_k; \boldsymbol{D}_k).\end{aligned}$$

3.3 Numerical solution of the NLP

Generalized Gauß-Newton method Iterate:

$$\boldsymbol{r}_k^{i+1} = \boldsymbol{r}_k^i + \Delta \boldsymbol{r}_k^i$$

where $\Delta \boldsymbol{r}_k^i$ is a solution of the quadratic subproblem

$$\begin{aligned}&\min_{\Delta \boldsymbol{r}_k^i} \ \left\| J(\boldsymbol{r}_k^i; \boldsymbol{D}_k) + \nabla_{\boldsymbol{r}} J(\boldsymbol{r}_k^i; \boldsymbol{D}_k)^T \Delta \boldsymbol{r}_k^i \right\|_2^2\\ &\text{s.t.} \ \ \boldsymbol{0} = \boldsymbol{G}(\boldsymbol{r}_k^i; \boldsymbol{D}_k) + \nabla_{\boldsymbol{r}} \boldsymbol{G}(\boldsymbol{r}_k^i; \boldsymbol{D}_k)^T \Delta \boldsymbol{r}_k^i,\\ &\qquad \boldsymbol{0} \leq \boldsymbol{H}(\boldsymbol{r}_k; \boldsymbol{D}_k) + \nabla_{\boldsymbol{r}} \boldsymbol{H}(\boldsymbol{r}_k; \boldsymbol{D}_k)^T \Delta \boldsymbol{r}_k^i\end{aligned}$$

which can be solved e.g. using the null space method.

3.4 MHE real-time iteration

Two key ideas: i) perform iteration only once per sampling -> justification: close to solution steps are small anyway

ii) divide into preparation and estimation phase -> possible as y_k enters the cost function only linearly

+++: details mit Handout durchgehen

4 Discussion

Advantages:

- Fulfils formal real-time paradigm
- performs very well in case studies

Disadvantages:

- no global convergence
- not known yet how to best choose the algorithm parameters
- no formal proof of stability

Summary

- Moving horizon: Restrict on time window
- Include old information into arrival cost -> carefully design Arr. cost
- Solve resulting NLP using GGN
- achieve RTI by doing only one GGN iteration and dividing into preparation and estimation phase