Aalto University
School of Electrical Engineering

ELEC-E8740 — Linear Continuous-Time Dynamic Models

Simo Särkkä

Aalto University

November 1, 2022

Contents

Intended Learning Outcomes

After this lecture, you will be able to:

- describe the idea of dynamic modeling in sensor fusion,
- explain the process of constructing continuous-time state-space models,
- distinguish deterministic and stochastic state-space models,
- construct linear and nonlinear continuous-time state-space models.

Recap (1/2)

- The Gauss–Newton update can be scaled with additional parameter γ:

$$\hat{\mathbf{x}}^{(i+1)} = \hat{\mathbf{x}}^{(i)} + \gamma \Delta \hat{\mathbf{x}}^{(i+1)}.$$

- The parameter can be found via line search that minimizes

$$J_{\mathrm{WLS}}^{(i)}(\gamma) = J_{\mathrm{WLS}}\left(\hat{\mathbf{x}}^{(i)} + \gamma \Delta \hat{\mathbf{x}}^{(i+1)}\right).$$

- We can also use inexact line search which ensures that the cost is decreased a sufficient amount.
- In Levenberg–Marquardt (LM) algorithm we replace the linear approximation in Gauss–Newton with its regularized version.
- In LM algorithm, we find a suitable regularization parameter λ via an iterative procedure.

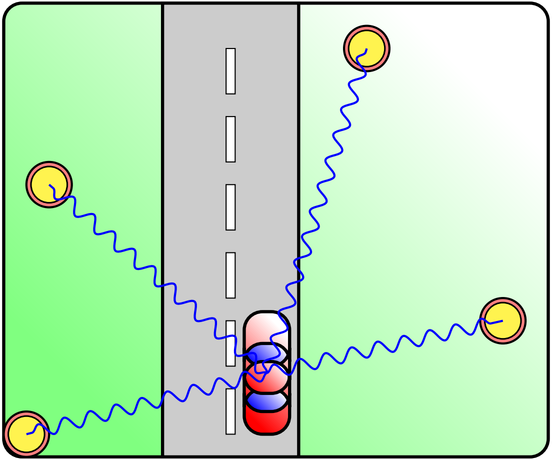
Recap (2/2)

- We can also consider regularized nonlinear problems with a simple trick:

$$
\begin{aligned}
J_{\text{ReLS}}(\mathbf{x}) &= (\mathbf{y} - \mathbf{g}(\mathbf{x}))^{\mathsf{T}} \mathbf{R}^{-1} (\mathbf{y} - \mathbf{g}(\mathbf{x})) + (\mathbf{m} - \mathbf{x})^{\mathsf{T}} \mathbf{P}^{-1} (\mathbf{m} - \mathbf{x}) \\
&= \left(\begin{bmatrix} \mathbf{y} \\ \mathbf{m} \end{bmatrix} - \begin{bmatrix} \mathbf{g}(\mathbf{x}) \\ \mathbf{x} \end{bmatrix} \right)^{\mathsf{T}} \begin{bmatrix} \mathbf{R}^{-1} & 0 \\ 0 & \mathbf{P}^{-1} \end{bmatrix} \left(\begin{bmatrix} \mathbf{y} \\ \mathbf{m} \end{bmatrix} - \begin{bmatrix} \mathbf{g}(\mathbf{x}) \\ \mathbf{x} \end{bmatrix} \right).
\end{aligned}
$$

- Quasi-Newton methods are more general optimization methods that approximate the Hessian in Newton's method.
- Various convergence criteria are available for terminating iterative optimization methods.

Motivation: Moving Targets (1/2)

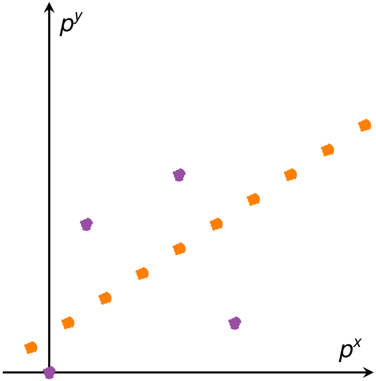


Motivation: Moving Targets (2/2)

- In practice, we often wish to track a moving target.
- One way is to recompute the position at every time step.
- This ignores the time continuity.
- We get a better result by modeling the temporal relationship of measurements.
- This can be done using (stochastic) differential equations and difference equations.

Localizing a Moving Target (1/4)

- Target moves rather than being stationary
- Sensors measure periodically, e.g., every second
- We can now either
  1. recompute the position estimate at every time, or
  2. use a dynamic model to connect the time points.



Localizing a Moving Target (2/4)

- Let us try a straight line model:

$$
\begin{aligned}
p^x(t) &= p^x(0) + v^x t, \\
p^y(t) &= p^y(0) + v^y t.
\end{aligned}
$$

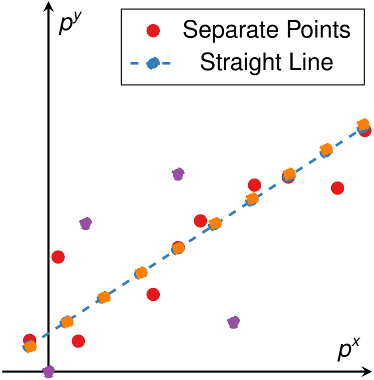
- Measurement model:

$$
\begin{aligned}
y_n(t) &= \sqrt{(p^x(t) - s_n^x)^2 + (p^y(t) - s_n^y)^2} + r_n(t) \\
&= \sqrt{(p^x(0) + v^x t - s_n^x)^2 + (p^y(0) + v^y t - s_n^y)^2} + r_n(t)
\end{aligned}
$$

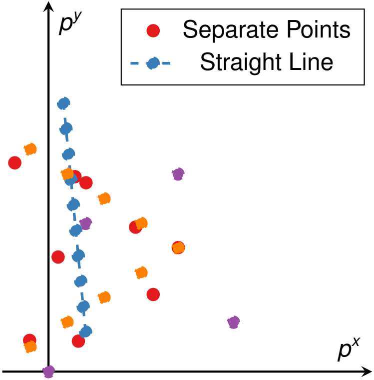
- We need to estimate 4 parameters:

$$
\mathbf{x} = \begin{bmatrix} p_t^x(0) & p_t^y(0) & v^x & v^y \end{bmatrix}^{\mathsf{T}}
$$

Localizing a Moving Target (3/4)



Localizing a Moving Target (4/4)

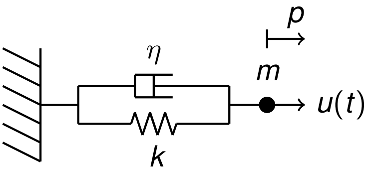


Localizing a Moving Target: Conclusions

- The static approach is not too well suited for time-varying processes
- A systematic method that relates (time-wise) related measurements is needed
- Solution: Use differential (and difference) equations to model the time-varying, i.e., dynamic, system

ODE Modeling of Dynamic Systems

- Ordinary differential equations (ODEs) can be used to describe many dynamic systems.
- Example: Spring-damper system:



- Second order ordinary differential equation:

$$ma(t) = -kp(t) - \eta v(t) + u(t)$$

- Other examples: Newtonion/Hamiltonian dynamics, kinematic models, heat and mass transfer, wave equations, . . .

Example: State-Space Representation of ODEs

- The second order ODE for spring:

$$ma(t) = -kp(t) - \eta v(t) + u(t)$$

- Equation system representation:

$$\begin{bmatrix} v(t) \\ a(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{\eta}{m} \end{bmatrix} \begin{bmatrix} p(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u(t)$$

- First order ODE equation system:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{\eta}{m} \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u(t)$$

- $\mathbf{x}(t) = \begin{bmatrix} p(t) & v(t) \end{bmatrix}^{\mathsf{T}}$ is the state of the system

Example: A Coffee Cup's Cooling (1/2)

- Newton's law of cooling for the coffee cup:

$$\frac{\mathrm{d}T_\mathrm{c}(t)}{\mathrm{d}t} = -k_1 (T_\mathrm{c}(t) - T_\mathrm{r}(t)),$$

- Newton's law of cooling for the room:

$$\frac{\mathrm{d}T_\mathrm{r}(t)}{\mathrm{d}t} = -k_2 (T_\mathrm{r}(t) - T_\mathrm{a}(t)) + h(t),$$

- Equation system:

$$\begin{aligned}
\frac{\mathrm{d}T_\mathrm{r}(t)}{\mathrm{d}t} &= -k_2 (T_\mathrm{r}(t) - T_\mathrm{a}(t)) + h(t) \\
\frac{\mathrm{d}T_\mathrm{c}(t)}{\mathrm{d}t} &= -k_1 (T_\mathrm{c}(t) - T_\mathrm{r}(t))
\end{aligned}$$

Example: A Coffee Cup's Cooling (2/2)

- The equation system:

$$
\begin{aligned}
\frac{\mathrm{d}T_\mathrm{r}(t)}{\mathrm{d}t} &= -k_2(T_\mathrm{r}(t) - T_\mathrm{a}(t)) + h(t) \\
\frac{\mathrm{d}T_\mathrm{c}(t)}{\mathrm{d}t} &= -k_1(T_\mathrm{c}(t) - T_\mathrm{r}(t))
\end{aligned}
$$

- In matrix form:

$$
\begin{bmatrix} \frac{\mathrm{d}T_\mathrm{r}(t)}{\mathrm{d}t} \\ \frac{\mathrm{d}T_\mathrm{c}(t)}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} -k_2 & 0 \\ k_1 & -k_1 \end{bmatrix} \begin{bmatrix} T_\mathrm{r}(t) \\ T_\mathrm{c}(t) \end{bmatrix} + \begin{bmatrix} k_2 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T_\mathrm{a}(t) \\ h(t) \end{bmatrix}
$$

- Compact state-space notation:

$$
\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}_u\mathbf{u}(t)
$$

A Linear System of Differential Equations (1/2)

General system of first order differential equations:

$$
\begin{aligned}
\dot{x}_1(t) &= a_{11} x_1(t) + a_{12} x_2(t) + \cdots + a_{1 d_x} x_{d_x}(t) \\
&\qquad + b_{11} u_1(t) + b_{12} u_2(t) + \cdots + b_{1 d_u} u_{d_u}(t) \\
\dot{x}_2(t) &= a_{21} x_1(t) + a_{22} x_2(t) + \cdots + a_{2 d_x} x_{d_x}(t) \\
&\qquad + b_{21} u_1(t) + b_{22} u_2(t) + \cdots + b_{2 d_u} u_{d_u}(t) \\
&\quad \vdots \\
\dot{x}_{d_x}(t) &= a_{d_x 1} x_1(t) + a_{d_x 2} x_2(t) + \cdots + a_{d_x d_x} x_{d_x}(t) \\
&\qquad + b_{d_x 1} u_1(t) + b_{d_x 2} u_2(t) + \cdots + b_{d_x d_u} u_{d_u}(t)
\end{aligned}
$$

A Linear System of Differential Equations (2/2)

- In matrix form:

$$\begin{bmatrix} \dot{x}_1(t) \\ \vdots \\ \dot{x}_{d_x}(t) \end{bmatrix} = \begin{bmatrix} a_{11} & \dots & a_{1 d_x} \\ \vdots & \ddots & \vdots \\ a_{d_x 1} & \dots & a_{d_x d_x} \end{bmatrix} \begin{bmatrix} x_1(t) \\ \vdots \\ x_{d_x}(t) \end{bmatrix} + \begin{bmatrix} b_{11} & \dots & b_{1 d_u} \\ \vdots & \ddots & \vdots \\ b_{d_x 1} & \dots & b_{d_x d_u} \end{bmatrix} \begin{bmatrix} u_1(t) \\ \vdots \\ u_{d_u}(t) \end{bmatrix}$$

- Compact state-space notation:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}_u \mathbf{u}(t)$$

- This is called the state-space form of the differential equation system, $\mathbf{x}(t)$ is the state of the system

Transforming ODEs to State-Space Form (1/2)

- *L*th order ODE in $z(t)$

$$\frac{\mathrm{d}^L z(t)}{\mathrm{d}t^L} = c_0 z(t) + c_2 \frac{\mathrm{d}z(t)}{\mathrm{d}t} + \cdots + c_{L-1} \frac{\mathrm{d}^{L-1} z(t)}{\mathrm{d}t^{L-1}} + d_1 u(t)$$

- Choose state components:

$$x_1(t) = z(t),\; x_2(t) = \frac{\mathrm{d}z(t)}{\mathrm{d}t},\; \ldots,\; x_{d_x}(t) = \frac{\mathrm{d}^{L-1} z(t)}{\mathrm{d}t^{L-1}}$$

- Then we have:

$$\begin{aligned}
\dot{x}_1(t) &= \frac{\mathrm{d}z(t)}{\mathrm{d}t} = x_2(t) \\
\dot{x}_2(t) &= \frac{\mathrm{d}^2 z(t)}{\mathrm{d}t^2} = x_3(t) \\
&\vdots \\
\dot{x}_{d_x}(t) &= \frac{\mathrm{d}^L z(t)}{\mathrm{d}t^L} = c_0 z(t) + c_2 \frac{\mathrm{d}z(t)}{\mathrm{d}t} + \cdots + c_{L-1} \frac{\mathrm{d}^{L-1} z(t)}{\mathrm{d}t^{L-1}} + d_1 u(t)
\end{aligned}$$

Transforming ODEs to State-Space Form (2/2)

- Rewritten in terms of states x_i:

$$\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
&\vdots \\
\dot{x}_{d_x}(t) &= c_0 x_1(t) + c_1 x_2(t) + \cdots + c_{L-1} x_{d_x}(t) + d_1 u(t)
\end{aligned}$$

- In matrix form:

$$\underbrace{\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_{d_x}(t) \end{bmatrix}}_{\triangleq \dot{\mathbf{x}}(t)} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & & \vdots \\ \vdots & & & \ddots & 0 \\ c_0 & c_1 & & \dots & c_{L-1} \end{bmatrix}}_{\triangleq \mathbf{A}} \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{d_x}(t) \end{bmatrix}}_{\triangleq \mathbf{x}(t)} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ d_1 \end{bmatrix}}_{\triangleq \mathbf{B}_u} u(t).$$

Deterministic Linear State-Space Model

- The dynamic model describes the evolution of the *state*:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$$

- The measurement model relates the *state* $\mathbf{x}_n = \mathbf{x}(t_n)$ at t_n to the *measurement* $\mathbf{y}_n$
- The linear measurement model is

$$\mathbf{y}_n = \mathbf{G}\mathbf{x}_n + \mathbf{r}_n.$$

- The deterministic linear state-space model combines the linear dynamic and measurement models

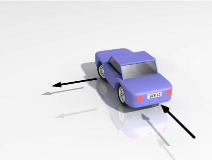
$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}_u\mathbf{u}(t),\\ \mathbf{y}_n &= \mathbf{G}\mathbf{x}_n + \mathbf{r}_n.\end{aligned}$$

Example: A Car Navigating in 2D (1)

- Newton's law gives:

$$
\begin{aligned}
m\, a^x &= F_\mathrm{p}^x \\
m\, a^y &= F_\mathrm{p}^y
\end{aligned}
$$

- Defining state $\mathbf{x} = \begin{bmatrix} p^x & p^y & v^x & v^y \end{bmatrix}^\mathsf{T}$ leads to



$$
\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m & 0 \\ 0 & 1/m \end{bmatrix} \begin{bmatrix} F_p^x \\ F_p^y \end{bmatrix}
$$

- Assuming position measurements $\mathbf{y}_n$ gives

$$
\mathbf{y}_n = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}_n + \mathbf{r}_n.
$$

Uncertainty in Dynamic Models

- The deterministic input $u(t)$ might not be known
- The model does not capture every aspect of the process
- Solution: Add a stochastic process $w(t)$ as an input
- Example: Stochastic differential equation (SDE) of order L:

$$\frac{\mathrm{d}^L z(t)}{\mathrm{d}t^L} = c_0 z(t) + c_1 \frac{\mathrm{d}z(t)}{\mathrm{d}t} + \cdots + c_{L-1} \frac{\mathrm{d}^{L-1} z(t)}{\mathrm{d}t^{L-1}} + d_1 w(t)$$

Input Process $w(t)$

- Assumed to be zero-mean and stationary
- Characterized by its autocorrelation function...

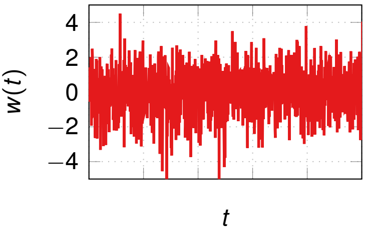
$$R_{ww}(\tau) = \mathrm{E}\{w(t+\tau)w(t)\}$$

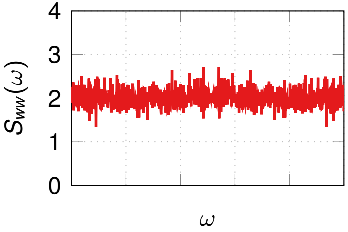
- ...or its power spectral density

$$S_{ww}(\omega) = \int R_{ww}(\tau) e^{-\mathrm{i}\,\omega\tau} \mathrm{d}\tau$$

White Processes

- "White noise" — equal contributions of each frequency
- Autocorrelation function: $R_{ww}(\tau) = \sigma_w^2 \delta(\tau)$
- Power spectral density: $S_{ww} = \sigma_w^2$
- Many forms of colored noise are filtered versions of white noise





Stochastic Linear State-Space Model

- Derivation of the stochastic dynamic model follows the same steps as for the deterministic case
- The stochastic process $\mathbf{w}(t)$ takes the place of the deterministic input $\mathbf{u}(t)$
- A system can have both deterministic and stochastic inputs
- Linear stochastic dynamic model:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}_w \mathbf{w}(t)$$

- Linear stochastic state-space model with measurements:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}_w \mathbf{w}(t) \\ \mathbf{y}_n &= \mathbf{G}\mathbf{x}_n + \mathbf{r}_n \end{aligned}$$

Example: A Car Navigating in 2D (2)

- Recall the deterministic dynamic model:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m & 0 \\ 0 & 1/m \end{bmatrix} \begin{bmatrix} F_p^x \\ F_p^y \end{bmatrix}$$

- F_p^x, F_p^y might be unknown when localizing the car
- Assume stochastic processes as the input:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}$$

- This is the Wiener velocity model in 2D

Summary

- Higher order ODEs and SDEs can be transformed to a first-order vector-valued equation system
- The deterministic linear state-space model is

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}_u\mathbf{u}(t)\\ \mathbf{y}_n &= \mathbf{G}\mathbf{x}_n + \mathbf{r}_n\end{aligned}$$

- The stochastic linear state-space model with stochastic input process $\mathbf{w}(t)$ is

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}_w\mathbf{w}(t)\\ \mathbf{y}_n &= \mathbf{G}\mathbf{x}_n + \mathbf{r}_n\end{aligned}$$

- The 2D Wiener velocity model is

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix}$$