



Aalto University
School of Electrical
Engineering

ELEC-E8740 — Continuous and Discrete Time Dynamic Models

Simo Särkkä

Aalto University

October 9, 2019

Contents

- 1 Intended Learning Outcomes and Recap
- 2 Motivation: Need for Dynamic Models
- 3 Linear ODEs and State-Space Models
- 4 Stochastic Modeling of Unknown Components
- 5 Nonlinear State-Space Models
- 6 Discrete-Time Models

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 **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

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

- 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.

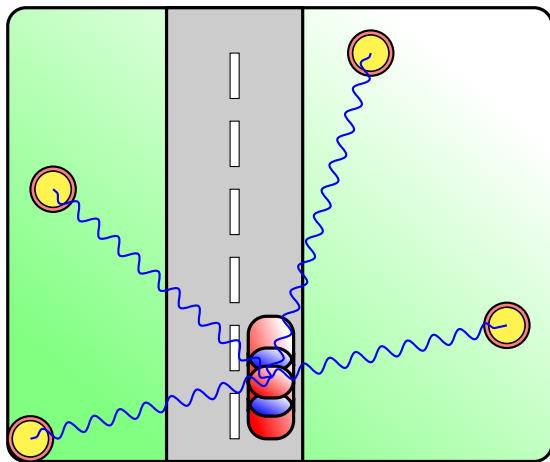
Recap (2/2)

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

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

- **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)

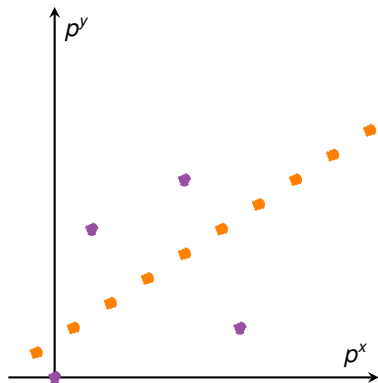


Motivation: Moving Targets (2/2)

- 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**:

$$p^x(t) = p^x(0) + v^x t,$$

$$p^y(t) = p^y(0) + v^y t.$$

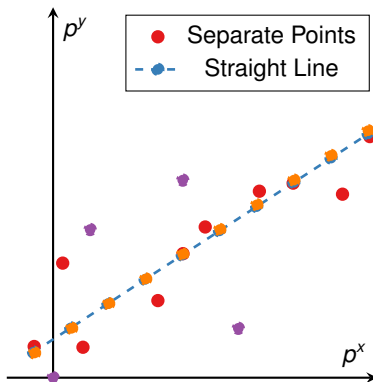
- Measurement model**:

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

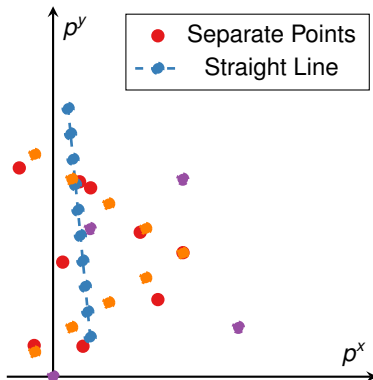
- We need to estimate **4 parameters**:

$$\mathbf{x} = [p_t^x(0) \quad p_t^y(0) \quad v^x \quad v^y]^T$$

Localizing a Moving Target (3/4)



Localizing a Moving Target (4/4)

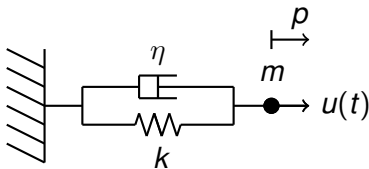


Localizing a Moving Target: Conclusions

- 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: Newtonian/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:

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

- First order ODE equation system:

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

- $\mathbf{x}(t) = [p(t) \quad v(t)]^T$ is the state of the system

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

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

$$\frac{dT_c(t)}{dt} = -k_1(T_c(t) - T_r(t)),$$

- Newton's law of cooling for the room:

$$\frac{dT_r(t)}{dt} = -k_2(T_r(t) - T_a(t)) + h(t),$$

- Equation system:

$$\begin{aligned}\frac{dT_r(t)}{dt} &= -k_2(T_r(t) - T_a(t)) + h(t) \\ \frac{dT_c(t)}{dt} &= -k_1(T_c(t) - T_r(t))\end{aligned}$$

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

- The equation system:

$$\begin{aligned}\frac{dT_r(t)}{dt} &= -k_2(T_r(t) - T_a(t)) + h(t) \\ \frac{dT_c(t)}{dt} &= -k_1(T_c(t) - T_r(t))\end{aligned}$$

- In matrix form:

$$\begin{bmatrix} \frac{dT_r(t)}{dt} \\ \frac{dT_c(t)}{dt} \end{bmatrix} = \begin{bmatrix} -k_2 & 0 \\ k_1 & -k_1 \end{bmatrix} \begin{bmatrix} T_r(t) \\ T_c(t) \end{bmatrix} + \begin{bmatrix} k_2 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T_a(t) \\ h(t) \end{bmatrix}$$

- Compact state-space notation:

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

A Linear System of Differential Equations (1/2)

General system of first order differential equations:

$$\dot{x}_1(t) = a_{11}x_1(t) + a_{12}x_2(t) + \cdots + a_{1d_x}x_{d_x}(t) \\ + b_{11}u_1(t) + b_{12}u_2(t) + \cdots + b_{1d_u}u_{d_u}(t)$$

$$\dot{x}_2(t) = a_{21}x_1(t) + a_{22}x_2(t) + \cdots + a_{2d_x}x_{d_x}(t) \\ + b_{21}u_1(t) + b_{22}u_2(t) + \cdots + b_{2d_u}u_{d_u}(t)$$

⋮

$$\dot{x}_{d_x}(t) = a_{d_x1}x_1(t) + a_{d_x2}x_2(t) + \cdots + a_{d_xd_x}x_{d_x}(t) \\ + b_{d_x1}u_1(t) + b_{d_x2}u_2(t) + \cdots + b_{d_xd_u}u_{d_u}(t)$$

A Linear System of Differential Equations (2/2)

- In matrix form:

$$\begin{bmatrix} \dot{x}_1(t) \\ \vdots \\ \dot{x}_{d_x}(t) \end{bmatrix} = \begin{bmatrix} a_{11} & \dots & a_{1d_x} \\ \vdots & \ddots & \vdots \\ a_{d_x1} & \dots & a_{d_xd_x} \end{bmatrix} \begin{bmatrix} x_1(t) \\ \vdots \\ x_{d_x}(t) \end{bmatrix} + \begin{bmatrix} b_{11} & \dots & b_{1d_u} \\ \vdots & \ddots & \vdots \\ b_{d_x1} & \dots & b_{d_xd_u} \end{bmatrix} \begin{bmatrix} u_1(t) \\ \vdots \\ u_{d_u}(t) \end{bmatrix}$$

- 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 $z(t)$

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

- Choose state components:

$$x_1(t) = z(t), \quad x_2(t) = \frac{dz(t)}{dt}, \quad \dots, \quad x_{d_x}(t) = \frac{d^{L-1} z(t)}{dt^{L-1}}$$

- Then we have:

$$\dot{x}_1(t) = \frac{dz(t)}{dt} = x_2(t)$$

$$\dot{x}_2(t) = \frac{d^2 z(t)}{dt^2} = x_3(t)$$

⋮

$$\dot{x}_{d_x}(t) = \frac{d^L z(t)}{dt^L} = c_0 z(t) + c_2 \frac{dz(t)}{dt} + \dots + c_{L-1} \frac{d^{L-1} z(t)}{dt^{L-1}} + d_1 u(t)$$

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

- Rewritten in terms of states x_i :

$$\dot{x}_1(t) = x_2(t)$$

$$\vdots$$

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

- In matrix form:

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

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

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

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

Example: A Car Navigating in 2D (1)

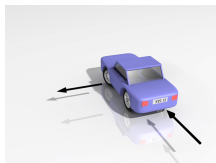
- Newton's law gives:

$$m a^x = F_p^x$$

$$m a^y = F_p^y$$

- Defining state $\mathbf{x} = [p^x \ p^y \ v^x \ v^y]^T$ leads to

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



- 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 :

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

Input Process $w(t)$

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

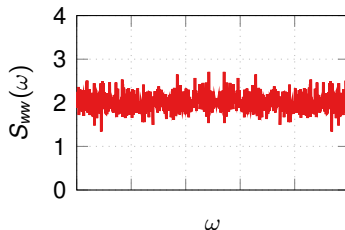
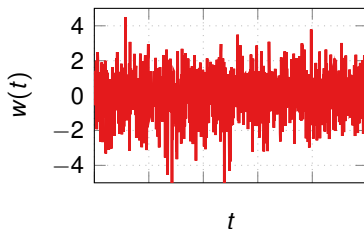
$$R_{ww}(\tau) = E\{w(t + \tau)w(t)\}$$

- . . . or its power spectral density

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

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**:

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

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

Example: A Car Navigating in 2D (2)

- Recall the **deterministic dynamic model**:

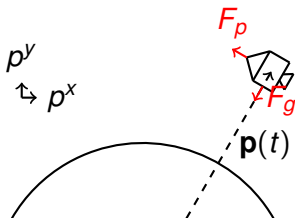
$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1/m & 0 \\ 0 & 1/m \end{bmatrix} \begin{bmatrix} F_p^x \\ F_p^y \end{bmatrix}$$

- 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**

Example: Dynamic Model for a Spacecraft (1/2)



- Gravitational acceleration:

$$g \approx g_0 \left(\frac{r_e}{|\mathbf{p}(t)|} \right)^2,$$

Example: Dynamic Model for a Spacecraft (2/2)

- Gravitational pull: $\mathbf{F}_g = -mg_0 r_e^2 \frac{\mathbf{p}(t)}{|\mathbf{p}(t)|^3}$
- Propulsion: $\mathbf{F}_p = F_p \frac{1}{|\mathbf{p}(t)|} \begin{bmatrix} -p^y(t) \\ p^x(t) \end{bmatrix}$
- Differential equation:

$$m\mathbf{a}(t) = -mg_0 r_e^2 \frac{\mathbf{p}(t)}{|\mathbf{p}(t)|^3} + \frac{1}{|\mathbf{p}(t)|} \begin{bmatrix} -p^y(t) \\ p^x(t) \end{bmatrix} u(t).$$

- State vector:

$$\mathbf{x}(t) = [p^x(t) \quad p^y(t) \quad v^x(t) \quad v^y(t)]^T.$$

Can not be written as $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}_u\mathbf{u}(t)$.

Nonlinear Differential Equation Systems

- Nonlinear ordinary differential equation system (b_{ij} may depend on $x_n(t)$):

$$\dot{x}_1(t) = f_1(x_1(t), x_2(t), \dots, x_{d_x}(t)) + b_{11}u_1(t) + \dots + b_{1d_u}u_{d_u}(t)$$

$$\dot{x}_2(t) = f_2(x_1(t), x_2(t), \dots, x_{d_x}(t)) + b_{21}u_1(t) + \dots + b_{2d_u}u_{d_u}(t)$$

⋮

$$\dot{x}_{d_x}(t) = f_{d_x}(x_1(t), x_2(t), \dots, x_{d_x}(t)) + b_{d_x1}u_1(t) + \dots + b_{d_xd_u}u_{d_u}(t)$$

- State vector: $\mathbf{x}(t) = [x_1(t) \quad x_2(t) \quad \dots \quad x_{d_x}(t)]^T$
- In vector form:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_{d_x}(t) \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{x}(t)) \\ f_2(\mathbf{x}(t)) \\ \vdots \\ f_{d_x}(\mathbf{x}(t)) \end{bmatrix} + \begin{bmatrix} b_{11}(\mathbf{x}(t)) & \dots & b_{1d_u}(\mathbf{x}(t)) \\ b_{21}(\mathbf{x}(t)) & & \vdots \\ \vdots & \ddots & \\ b_{d_x1}(\mathbf{x}(t)) & \dots & b_{d_xd_u}(\mathbf{x}(t)) \end{bmatrix} \mathbf{u}(t).$$

Nonlinear Continuous-Time State-Space Models

- **Deterministic** nonlinear dynamic model:

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

- **Stochastic** nonlinear dynamic model:

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

- Nonlinear **measurement model**:

$$\mathbf{y}_n = \mathbf{g}(\mathbf{x}_n) + \mathbf{r}_n$$

- Stochastic nonlinear **state-space model**:

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

$$\mathbf{y}_n = \mathbf{g}(\mathbf{x}_n) + \mathbf{r}_n$$

Example: Dynamic Model for a Spacecraft (2)

- Differential equation:

$$m\mathbf{a}(t) = -mg_0r_e^2 \frac{\mathbf{p}(t)}{|\mathbf{p}(t)|^3} + \frac{1}{|\mathbf{p}(t)|} \begin{bmatrix} -p^y(t) \\ p^x(t) \end{bmatrix} w(t).$$

- State vector:

$$\mathbf{x}(t) = [p^x(t) \quad p^y(t) \quad v^x(t) \quad v^y(t)]^T.$$

- Vector form:

$$\begin{bmatrix} v^x(t) \\ v^y(t) \\ a^x(t) \\ a^y(t) \end{bmatrix} = \begin{bmatrix} v^x(t) \\ v^y(t) \\ -g_0r_e^2 \frac{p^x(t)}{|\mathbf{p}(t)|^3} \\ -g_0r_e^2 \frac{p^y(t)}{|\mathbf{p}(t)|^3} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{p^y(t)}{m|\mathbf{p}(t)|} \\ \frac{p^x(t)}{m|\mathbf{p}(t)|} \end{bmatrix} w(t)$$
$$= \begin{bmatrix} f_1(\mathbf{x}(t)) \\ f_2(\mathbf{x}(t)) \\ f_3(\mathbf{x}(t)) \\ f_4(\mathbf{x}(t)) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{p^y(t)}{m|\mathbf{p}(t)|} \\ \frac{p^x(t)}{m|\mathbf{p}(t)|} \end{bmatrix} w(t),$$

Example: Robot Navigation in 2D (1/3)

- Quasi-constant turn model:

$$\dot{p}^x(t) = v(t) \cos(\varphi(t))$$

$$\dot{p}^y(t) = v(t) \sin(\varphi(t))$$

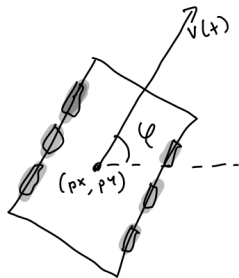
$$\dot{v}(t) = w_1(t)$$

$$\dot{\varphi}(t) = w_2(t)$$

- The state is

$$\mathbf{x}(t) = [p^x(t) \quad p^y(t) \quad v(t) \quad \varphi(t)]^T.$$

- Position measurement: picks $p^x(t)$ and $p^y(t)$
- Speed measurements (odometry): $v(t)$
- Magnetometer (compass): $\varphi(t)$.



Example: Robot Navigation in 2D (2/3)

- Gyroscope measures $\dot{v}(t)$.
- Accelerometer measures $\dot{\varphi}(t)$.
 - Word of warning: accelerometers are usually not accurate enough for this.
- Putting these into the equations we get the model

$$\dot{p}^x(t) = v(t) \cos(\varphi(t))$$

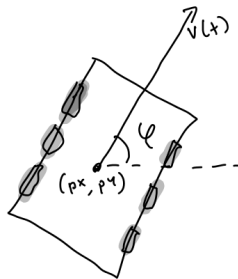
$$\dot{p}^y(t) = v(t) \sin(\varphi(t))$$

$$\dot{v}(t) = a_{\text{acc}}(t) + w_1(t)$$

$$\dot{\varphi}(t) = \omega_{\text{gyro}}(t) + w_2(t).$$

- The state is still

$$\mathbf{x}(t) = [p^x(t) \quad p^y(t) \quad v(t) \quad \varphi(t)]^T.$$



Example: Robot Navigation in 2D (3/3)

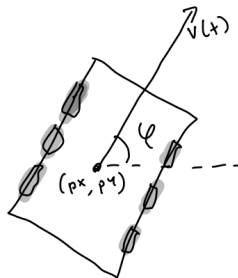
- Often we have the speed $v(t)$ directly available (e.g., from wheels)
- Then we can reduce the model to

$$\dot{p}^x(t) = v(t) \cos(\varphi(t))$$

$$\dot{p}^y(t) = v(t) \sin(\varphi(t))$$

$$\dot{\varphi}(t) = \omega_{\text{gyro}}(t) + w(t).$$

- The state is now
 $\mathbf{x}(t) = [p^x(t) \quad p^y(t) \quad \varphi(t)]^T$.
- This is a typical model used in 2D tracking.



Discrete-Time Processes and Difference Equations

- Some processes are only defined at discrete time points t_1, t_2, \dots
- The discrete-time equivalent of differential equations are **difference equations**
- The **difference** of two discrete points in time takes the role of the **derivative**

Vector Form of Difference Equation Systems

- Equation system of d_x linear difference equations:

$$x_{1,n} = a_{11}x_{1,n-1} + \cdots + a_{1d_x}x_{d_x,n-1} + b_{11}u_{1,n} + \cdots + b_{1d_u}u_{d_u,n}$$

$$x_{2,n} = a_{21}x_{1,n-1} + \cdots + a_{2d_x}x_{d_x,n-1} + b_{21}u_{1,n} + \cdots + b_{2d_u}u_{d_u,n}$$

\vdots

$$x_{d_x,n} = a_{d_x1}x_{1,n-1} + \cdots + a_{d_xd_x}x_{d_x,n-1} + b_{d_x1}u_{1,n} + \cdots + b_{d_xd_u}u_{d_u,n}$$

- Vector form:

$$\begin{bmatrix} x_{1,n} \\ \vdots \\ x_{d_x,n} \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1d_x} \\ \vdots & \ddots & \vdots \\ a_{d_x1} & \cdots & a_{d_xd_x} \end{bmatrix} \begin{bmatrix} x_{1,n-1} \\ \vdots \\ x_{d_x,n-1} \end{bmatrix} + \begin{bmatrix} b_{11} & \cdots & b_{1d_u} \\ \vdots & \ddots & \vdots \\ b_{d_x1} & \cdots & b_{d_xd_u} \end{bmatrix} \begin{bmatrix} u_{1,n} \\ \vdots \\ u_{d_u,n} \end{bmatrix}$$

- Compact notation:

$$\mathbf{x}_n = \mathbf{F}\mathbf{x}_{n-1} + \mathbf{B}_u\mathbf{u}_n$$

Deterministic Discrete-Time State-Space Model

- Linear discrete-time dynamic model:

$$\mathbf{x}_n = \mathbf{F}\mathbf{x}_{n-1} + \mathbf{B}_u\mathbf{u}_n$$

- Deterministic, linear discrete-time state-space model:

$$\mathbf{x}_n = \mathbf{F}\mathbf{x}_{n-1} + \mathbf{B}_u\mathbf{u}_n$$

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

with $E\{\mathbf{r}_n\} = \mathbf{0}$, $\text{Cov}\{\mathbf{r}_n\} = \mathbf{R}_n$, $\text{Cov}\{\mathbf{r}_n, \mathbf{r}_m\} = \mathbf{0}$ ($n \neq m$)

Conversion of L th Order Difference Equation (1/2)

- L th order difference equation (with single input u_n):

$$z_n = c_1 z_{n-1} + c_2 z_{n-2} + \dots + c_L z_{n-L} + d_1 u_n$$

- It is easier to choose \mathbf{x}_{n-1} on the RHS (c.f. continuous case)
- A possible choice:

$$x_{1,n-1} = z_{n-1}, x_{2,n-1} = z_{n-2}, \dots, x_{d_x,n-1} = z_{n-L}.$$

Conversion of L th Order Difference Equation (2/2)

- Difference equation system:

$$x_{1,n} = c_1 x_{1,n-1} + c_2 x_{2,n-1} + \cdots + c_L x_{d_x,n-1} + d_1 u_n$$

$$x_{2,n} = z_{n-1} = x_{1,n-1}$$

$$\vdots$$

$$x_{d_x,n} = z_{n-L+1} = x_{d_x+1,n-1}$$

- Vector form:

$$\begin{bmatrix} x_{1,n} \\ x_{2,n} \\ \vdots \\ x_{d_x,n} \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & \cdots & c_L \\ 1 & 0 & & \vdots \\ \vdots & \ddots & & \\ 0 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} x_{1,n-1} \\ x_{2,n-1} \\ \vdots \\ x_{d_x,n-1} \end{bmatrix} + \begin{bmatrix} d_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u_n,$$

Stochastic Linear State-Space Model (1/2)

- Dynamics are not entirely deterministic and inputs may not always be known
- Let the **process noise** \mathbf{q}_n (random variable) take the place of the input \mathbf{u}_n (or in addition to \mathbf{u}_n)
- Stochastic linear discrete-time dynamic model:

$$\mathbf{x}_n = \mathbf{F}\mathbf{x}_{n-1} + \mathbf{B}_q\mathbf{q}_n$$

Stochastic Linear State-Space Model (2/2)

- Stochastic linear discrete-time dynamic model:

$$\mathbf{x}_n = \mathbf{F}\mathbf{x}_{n-1} + \mathbf{B}_q\mathbf{q}_n$$

- The process noise follows

$$\mathbf{q}_n \sim p(\mathbf{q}_n)$$

with $E\{\mathbf{q}_n\} = 0$, $\text{Cov}\{\mathbf{q}_n\} = \mathbf{Q}_n$, and $\text{Cov}\{\mathbf{q}_m, \mathbf{q}_n\} = 0$
($m \neq n$)

- Stochastic linear discrete-time state-space model:

$$\mathbf{x}_n = \mathbf{F}\mathbf{x}_{n-1} + \mathbf{B}_q\mathbf{q}_n$$

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

Nonlinear Discrete-Time Dynamic Model

- Difference equations may also be nonlinear
- Nonlinear difference equation system (with process noise inputs):

$$x_{1,n} = f_1(x_{1,n-1}, x_{2,n-1}, \dots, x_{d_x,n-1}) + b_{11}q_{1,n} + \dots + b_{1d_q}q_{d_q,n}$$

$$x_{2,n} = f_2(x_{1,n-1}, x_{2,n-1}, \dots, x_{d_x,n-1}) + b_{21}q_{1,n} + \dots + b_{2d_q}q_{d_q,n}$$

⋮

$$x_{d_x,n} = f_{d_x}(x_{1,n-1}, x_{2,n-1}, \dots, x_{d_x,n-1}) + b_{d_x1}q_{1,n} + \dots + b_{d_xd_q}q_{d_q,n}$$

- Vector form:

$$\begin{bmatrix} x_{1,n} \\ \vdots \\ x_{d_x,n} \end{bmatrix} = \begin{bmatrix} f_1(x_{1,n-1}, x_{2,n-1}, \dots, x_{d_x,n-1}) \\ \vdots \\ f_{d_x}(x_{1,n-1}, x_{2,n-1}, \dots, x_{d_x,n-1}) \end{bmatrix} + \begin{bmatrix} b_{11} & \dots & b_{1d_q} \\ \vdots & \ddots & \vdots \\ b_{d_x1} & \dots & b_{d_xd_q} \end{bmatrix} \begin{bmatrix} q_{1,n} \\ \vdots \\ q_{d_q,n} \end{bmatrix}$$

Nonlinear Discrete-Time State-Space Model

- Compact notation of the dynamic model:

$$\mathbf{x}_n = \mathbf{f}(\mathbf{x}_{n-1}) + \mathbf{B}_q \mathbf{q}_n$$

- Nonlinear discrete-time state-space model:

$$\mathbf{x}_n = \mathbf{f}(\mathbf{x}_{n-1}) + \mathbf{B}_q \mathbf{q}_n$$

$$\mathbf{y}_n = \mathbf{g}(\mathbf{x}_n) + \mathbf{r}_n$$

where:

- $\mathbf{q}_n \sim p(\mathbf{q}_n)$, $E\{\mathbf{q}_n\} = 0$, $\text{Cov}\{\mathbf{q}_n\} = \mathbf{Q}_n$
- $\mathbf{r}_n \sim p(\mathbf{r}_n)$, $E\{\mathbf{r}_n\} = 0$, $\text{Cov}\{\mathbf{r}_n\} = \mathbf{R}_n$

Summary (1)

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

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

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

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

- The 2D Wiener velocity model is

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

Summary (2)

- Nonlinear continuous-time state-space model:

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

- Linear discrete-time state-space model:

$$\begin{aligned}\mathbf{x}_n &= \mathbf{F}\mathbf{x}_{n-1} + \mathbf{B}_q\mathbf{q}_n \\ \mathbf{y}_n &= \mathbf{G}\mathbf{x}_n + \mathbf{r}_n\end{aligned}$$

- Nonlinear discrete-time state-space model:

$$\begin{aligned}\mathbf{x}_n &= \mathbf{f}(\mathbf{x}_{n-1}) + \mathbf{B}_q(\mathbf{x}_{n-1})\mathbf{q}_n \\ \mathbf{y}_n &= \mathbf{g}(\mathbf{x}_n) + \mathbf{r}_n\end{aligned}$$