Dynamic System Models in Discrete Time

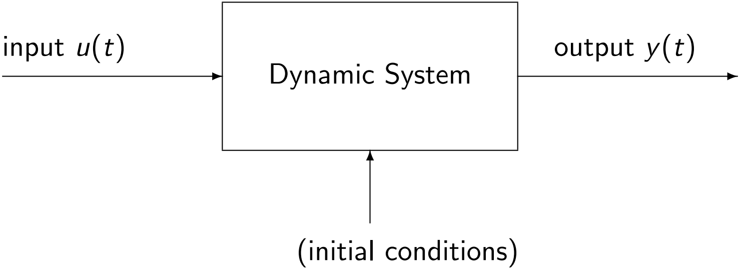
Moritz Diehl

Overview

- Deterministic Models
- Models with Measurement Noise (Output Errors)
- Models with Stochastic Disturbances (Equation Errors)

Deterministic Dynamic System Models

- A dynamic system model allows us to compute, for any horizon length N and sequence of inputs $u(1), \dots, u(N)$, the sequence of **system outputs** $y(1), \dots, y(N)$.
- Typically, we need to also specify the **initial conditions**.



Comments

Examples for Deterministic Models

- State Space Models:

$$x(t+1) = f(x(t), u(t))$$
$$y(t) = g(x(t), u(t)) \quad \text{for} \quad t = 1, 2, \ldots$$

  Initial conditions = initial state $x(1)$.

- Input Output Models:

$$y(t) = h(u(t), \ldots, u(t-n), y(t-1), \ldots, y(t-n))$$

  for $t = n+1, n+2, \ldots$.
  Initial conditions: $y(1), \ldots, y(n)$ and $u(1), \ldots, u(n)$.

Comments

State Space Form of Input-Output Systems

- can always transform from input-output form to state space
- state: $x(t) = (y(t-1), u(t-1), \ldots, y(t-n), u(t-n))^\top$
- state transition:

$$x(t+1) = f(x(t), u(t)) := \begin{bmatrix} h(u(t), \ldots, y(t-1), \ldots) \\ u(t) \\ y(t-1) \\ u(t-1) \\ \vdots \\ y(t-n) \\ u(t-n) \end{bmatrix}$$

- output equation: $y(t) = g(x(t), u(t)) := h(u(t), \ldots,)$.
- conversely, can transform some systems from state space to input-output form (e.g. LTI systems)

Comments

Linear Time Invariant Input Output Models

- General Difference Equation Form:

  $$y(t) = -a_1 y(t-1) - \ldots - a_n y(t-n) + b_0 u(t) + \ldots + b_n u(t-n)$$

  for $t = n+1, n+2, \ldots$.
  Initial conditions: $y(1), \ldots, y(n)$ and $u(1), \ldots, u(n)$.
- Even more general form with non-negative integers n_a, n_b and $a_0 \neq 0$:

  $$a_0 y(t) + \ldots + a_{n_a} y(t-n_a) = b_0 u(t) + \ldots + b_{n_b} u(t-n_b)$$

  Also called "Polynomial Model" and represented by rational **transfer function**

  $$G(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_{n_b} z^{-n_b}}{a_0 + a_1 z^{-1} + \ldots + a_{n_a} z^{-n_a}}$$

Comments

Discrete Time Transfer Functions

- Using $n = \max(n_a, n_b)$, by adding zero coefficients, one can bring each transfer function into the form

$$G(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_n z^{-n}}{a_0 + a_1 z^{-1} + \ldots + a_n z^{-n}} = \frac{b_0 z^n + b_1 z^{n-1} + \ldots + b_n}{a_0 z^n + a_1 z^{n-1} + \ldots + a_n}$$

- In MATLAB, an LTI system can be defined via the command `tf` (transfer function), with two vectors indicating the polynomial coefficients of **numerator** and **denominator**, and the sampling time T:

```
sys=tf([b0,b1, ..., bn], [a0,a1,..., an],T)
```

Comments

Polynomial Model Example: FIR Models

- Finite Impulse Response (FIR) models: $n_a = 0$.

$$y(t) = b_0 u(t) + \ldots + b_{n_b} u(t - n_b)$$

  Output is weighted sum of past inputs.

- Transfer function

$$\begin{aligned} G(z) &= b_0 + b_1 z^{-1} + \ldots + b_{n_b} z^{-n_b} \\ &= \frac{b_0 z^{n_b} + b_1 z^{n_b - 1} + \ldots + b_{n_b}}{z^{n_b} + 0 + \ldots + 0} \end{aligned}$$

- Concrete example: $y(t) = 2u(t) + 3u(t-1) + 4u(t-2)$.

$$G(z) = \frac{2z^2 + 3z + 4}{z^2 + 0 + 0}$$ MATLAB: sys=tf([2 3 4], [1 0 0],1)

Comments

Auto Regressive Models with Exogenous Inputs (ARX)

- For general polynomial models

$$G(z) = \frac{b_0 z^n + b_1 z^{n-1} + \ldots + b_n}{a_0 z^n + a_1 z^{n-1} + \ldots + a_n}$$

  with $n_a \neq 0$, different names are used in different fields:
  - **a**uto **r**egressive models with e**x**ogenous inputs (ARX)
  - Infinite Impulse Response (IIR) models
- There exist also auto regressive (AR) models without inputs, i.e. $n_b = 0$:

$$y(t) = -a_1 y(t-1) - \ldots - a_{n_a} y(t-n_a)$$

  Example: Fibonacci numbers 1,1,2,3,5,8,13,21, ...
  (however, such AR models have no transfer function)

Comments

Deterministic Simulation

- each deterministic model allows us to obtain $y(1), \ldots, y(N)$ by **forward simulation**.
- can define for $k = 1, \ldots, N$ the forward simulation map

$$y(k) = M(k; U, x_{\text{init}}, p)$$

  with initial conditions x_{init}, control trajectory U, and system parameters p.
- for LTI systems, simulation is particularly easy using the MATLAB command `lsim`.
- `lsim` expects an LTI system `sys`, and two vectors of length N: the input vector `u` and time vector `t`:

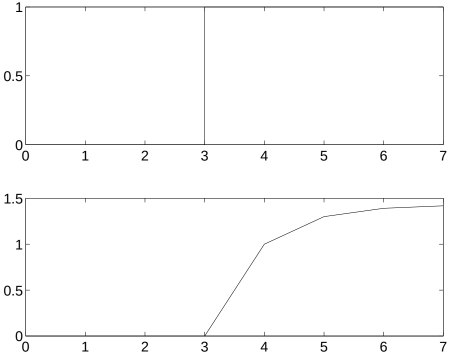
```
y=lsim(sys,u,t)
```

- `lsim` assumes by default zero initial conditions.

Comments

Example Code for using `lsim`

```
sys=tf([1],[1 -0.3],1);
u=[ 0 0 0 1 1 1 1 1]; t=[0 1 2 3 4 5 6 7];
y=lsim(sys,u,t);
subplot(2,1,1); stairs(t,u);
subplot(2,1,2); plot(t,y)
```



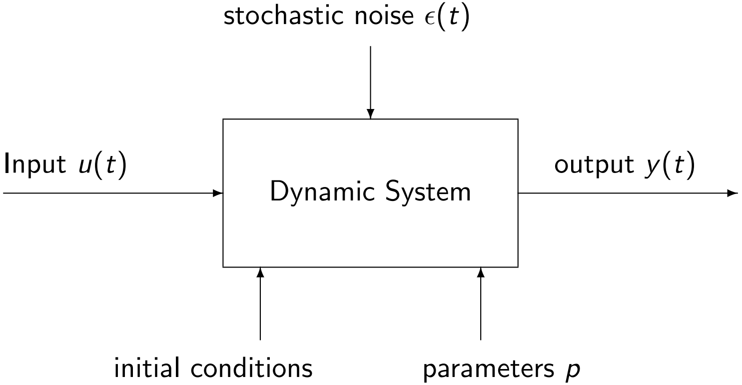
Comments

Overview

- Deterministic Models
- **Models with Measurement Noise (Output Errors)**
- Models with Stochastic Disturbances (Equation Errors)

General Stochastic Models

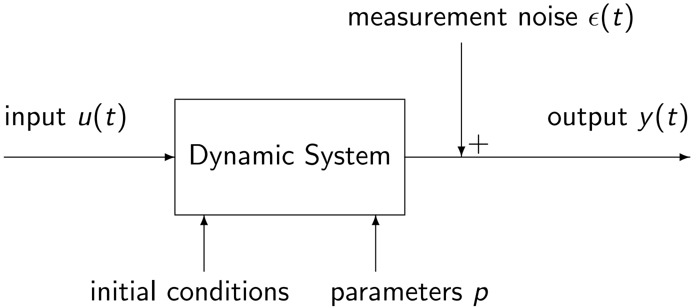
- In reality, we always have stochastic noise $\epsilon(t)$, e.g. external disturbances or measurement errors.
- also, we have unknown, but constant system parameters p.
- measured outputs $y(t)$ depend on both, $\epsilon(t)$ and p:



Comments

Special Case: Models with Output Errors

- simplest assumption: $\epsilon(t)$ is additive measurement noise
- use deterministic model: $\boxed{y(k) = M(k; U, x_{\text{init}}, p) + \epsilon(k)}$



Comments

Output Error Minimization

Assuming i.i.d. Gaussian noise, a maximum likelihood estimate for $\theta = (x_{\text{init}}^\top, p^\top)^\top$ can be obtained by nonlinear least squares:

$$\theta_{\text{ML}} = \arg\min_{\theta} \sum_{k=1}^{N} \left(y(k) - M(k; U, x_{\text{init}}, p) \right)^2$$

Here, $U = (u(1), \ldots, u(N))^\top$, and x_{init} represents initial conditions.

Advantage: often realistic noise assumption.

Disadvantage: M typically nonlinear, i.e. nonconvex optimization (impossible to find global minimum).

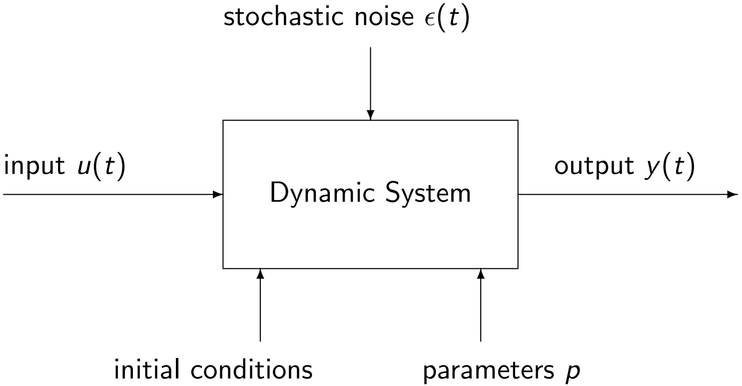
Comments

Overview

- Deterministic Models
- Models with Measurement Noise (Output Errors)
- **Models with Stochastic Disturbances (Equation Errors)**

General Stochastic Models

- stochastic noise $\epsilon(t)$ can enter model internally, not only at output equation
- measured outputs $y(t)$ depend on both, $\epsilon(t)$ and p:



Comments

Stochastic Models

- i.i.d. noise terms $\epsilon(t)$ enter the model as stochastic input.
- Stochastic State Space Models:

$$\begin{aligned} x(t+1) &= f(x(t), u(t), \epsilon(t)) \\ y(t) &= g(x(t), u(t), \epsilon(t)) \quad \text{for} \quad t = 0, 1, 2, \ldots \end{aligned}$$

- Stochastic Input Output Models:

$$y(t) = h(u(t), \ldots, u(t-n), y(t-1), \ldots, y(t-n), \epsilon(t), \ldots, \epsilon(t-n))$$

for $t = n+1, n+2, \ldots$.

Comments

Equation Error Models

- special case where i.i.d. noise $\epsilon(t)$ enters input-output equation as additive disturbance:

  $$y(t) = h(p, u(t), \ldots, u(t-n), y(t-1), \ldots, y(t-n)) + \epsilon(t))$$

  for $t = n+1, n+2, \ldots$.
- For i.i.d. Gaussian noise, maximum likelihood estimate for $\theta = p$ is given by

  $$\theta_{\mathrm{ML}} = \arg\min_{\theta} \sum_{k=n+1}^{N} \left(y(k) - h(p, u(t), \ldots, y(t-1), \ldots) \right)^2$$

- Note that $u(t)$ and $y(t)$ are the known measurements.
- The difference $y(k) - h(p, u(t), \ldots, y(t-1), \ldots))$ is called **equation error** or **prediction error**

Comments

Prediction Error Minimization

- Prediction error minimization (PEM) problem given by

$$\min_{\theta} \sum_{k=n+1}^{N} \left(y(k) - h(p, u(t), \ldots, y(t-1), \ldots) \right)^2$$

- this problem is convex if p enters linearly in f (**linear-in-the-parameters (LIP)**)
- thus, prediction error minimization is globally solvable for LIP models!

Comments

Examples for Linear-in-the-Parameters (LIP) Models

- General form of LIP model with equation error noise:

$$y(t) = \sum_{i=1}^{d} \theta_i \, \phi_i(u(t). \dots, y(t-1), \dots) + \epsilon(t)$$

  with basis functions $\phi_1, \dots, \phi_d$.
- Summarize as $\boxed{y(t) = \varphi(t)^\top \theta + \epsilon(t)}$ with regression vector $\varphi(t) = (\phi_1(\cdot), \dots, \phi_d(\cdot))^\top$.
- Prediction error minimization (PEM) problem then given by

$$\min_\theta \underbrace{\sum_{k=n+1}^{N} \quad (\ y(k) - \varphi(k)^\top \theta)\ )^2}_{= \|y_N - \Phi_N \theta\|_2^2}$$

- Solved by $\theta^* = \Phi_N^+ y_N$

Comments

Special Case: LIP-LTI Models with Equation Errors (ARX)

- general ARX model with equation errors given by

$$a_0 y(t) + \ldots + a_{n_a} y(t-n_a) = b_0 u(t) + \ldots + b_{n_b} u(t-n_b) + \epsilon(t)$$

- this is both LTI and LIP, combines best of two worlds...
- need to fix $a_0 = 1$ (otherwise, estimation problem is underdetermined)
- use $\theta = (a_1, \ldots, a_{n_a}, b_0, \ldots, b_{n_b})^\top$ and

$$\varphi(t) = (-y(t-1), \ldots, -y(t-n_a), u(t), \ldots, u(t-n_b))^\top$$

such that

$$y(t) = \varphi(t)^\top \theta + \epsilon(t)$$

- get unique θ easily using linear least squares!

Comments

Other Stochastic System Classes

- auto-regressive moving average with eXonegous input (ARMAX): noise term $\epsilon(t)$ goes through FIR filter before becoming equation error:

$$\begin{aligned} & a_0 y(t) + \ldots + a_{n_a} y(t - n_a) \\ = & b_0 u(t) + \ldots + b_{n_b} u(t - n_b) + \epsilon(t) + c_1 \epsilon(t-1) + \ldots c_{n_c} \epsilon(t - n_c) \end{aligned}$$

- auto-regressive moving average models without inputs (ARMA):

$$a_0 y(t) + \ldots + a_{n_a} y(t - n_a) = \epsilon(t) + c_1 \epsilon(t-1) + \ldots c_{n_c} \epsilon(t - n_c)$$

- when estimating the noise coefficients c_i, we always enter nonlinear least squares, because c_i are multiplied with unknown noise terms $\epsilon(t - i)$.