Linear Models for Output-Buffered Systems

Jim Ramsay, McGill University

University of Toronto, April 29, 2010

Outline

S. Ellner and J. Guckenheimer (2006), *Dynamic Models in Biology*.

"... contacts have been limited between the statistical research community and the scientific communities where dynamic models are used, such as economics and engineering – so limited that many other fields have independently developed statistical methods specific to their needs."

"... there is very little on dynamic models in the mainstream statistics literature or curriculum. An online search ... found 2 papers .. in the last 10 years of the *Journal of the American Statistical Association*, 4 in the last 10 years of *Biometrics*, and 5 in the last 10 years of *Biometrika*, which is under $\frac{1}{2}$ % of the total for those journals over the same time period."

What explains this gulf?

- Most dynamic systems are not fit to data; their designers are content with capturing gross shape features.
- When they are, the data-fitting algorithms used are primitive, unstable, and computationally intensive.
- Statisticians don’t know what to do with models defined by functional equations that cannot be put into explicit form.
- Statisticians and their clients tend not to “think” in terms of dynamic systems; time- and frequency-domain discrete time series models are dynamic, but tend to disguise the dynamic model.
- The modeling of input/output functional relationships is comparatively rare; most time series models capture only output structure.

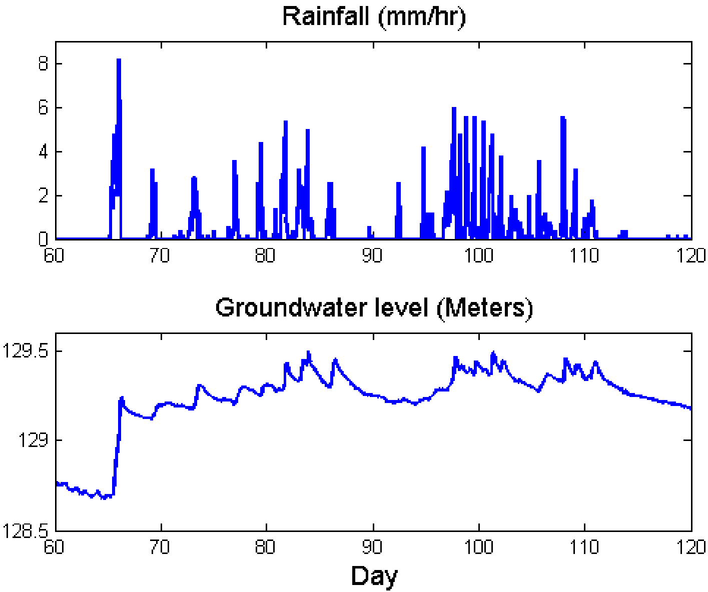
Where is this talk going?

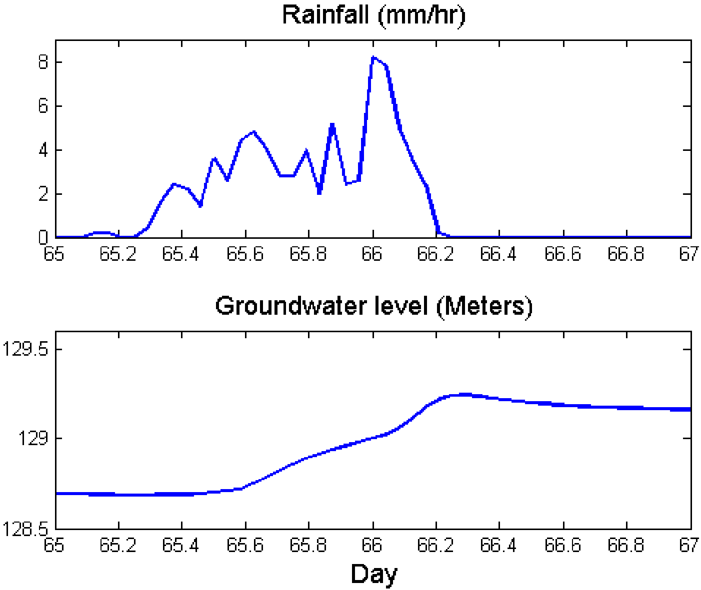
- Modeling groundwater level in response to rain in Vancouver illustrates a simple dynamical system.
- A quick intro to thinking dynamically is offered, in which a system buffers or diffuses sharp input changes.
- An approach to parameter estimation and inference for dynamical systems is sketched.
- Results for the groundwater data are displayed
- A system for spread of disease and a second order system are examined.

Outline

The Vancouver mud slide tragedy

- After a mud slide in North Vancouver that killed two people, the firm BGC Engineering was commissioned to monitor groundwater level.
- Rainfall raises groundwater level, and if it reaches a critical level at certain sites, people will be evacuated.
- Rainfall is measured hourly, and can be viewed as event data, or what statisticians call *marked point process* data.
- BGC Engineering asked us if we could develop an effective six-hour prediction of groundwater level (my son Sean works there).
- Was this a job for functional data analysis?


Rainfall (mm/hr)
8
6
4
2
0
60
70
80
90
100
110
120
Groundwater level (Meters)
129.5
129
128.5
60
70
80
90
100
110
120
Day


Rainfall (mm/hr)
0
2
4
6
8
65
65.2
65.4
65.6
65.8
66
66.2
66.4
66.6
66.8
67
Groundwater level (Meters)
128.5
129
129.5
65
65.2
65.4
65.6
65.8
66
66.2
66.4
66.6
66.8
67
Day

Direct functional regression won’t work

- Ramsay and Silverman (2005) *Functional Data Analysis* discuss the concurrent functional linear model

$$G(t) = G_0 + \alpha R(t) + \epsilon(t)$$

- But it easy to see that this won’t work because it proposes that sudden changes in rainfall $R(t)$ are passed on immediately to groundwater level $G(t)$.
- Instead, we see that G increases smoothly over a series of localized rainfalls, and declines slowly when they cease.
- We say that the response is `buffered`, or smoothed and delayed.

Outline

1. Introduction
2. How groundwater level responds to rainfall
3. **Buffered input-output systems**
4. The estimation procedure
5. Fitting groundwater level
6. Elementary spread of disease model: A system of diffusors
7. Second order linear diffusion buffers
8. Where we’ve been and where we can go next

Exponential decay or diffusion

- The shape of the decline in groundwater level G after an isolated rainfall event suggests exponential decay to a terminal level G_0

  $$G(t) = C\exp(-t/\tau) + G_0$$

  or

  $$G(t) = C\exp(-\beta t) + G_0, \quad \tau = 1/\beta,$$

- Time constant τ is about the time required to achieve 2/3 of the decay, 4τ is the number of time units to achieve about 98% of the decay, and β is the *speed* of the decay.

Exponential decay as a differential equation

- It is both more elegant and useful to express exponential decay as the differential equations

$$\tau \frac{dG}{dt} = -G(t) + G_0 \ \text{ or } \ \frac{dG}{dt} = -\beta[G(t) - G_0],$$

- or as

$$\tau DG + G = G_0 \ \text{ or } \ DG + \beta G = -G_0.$$

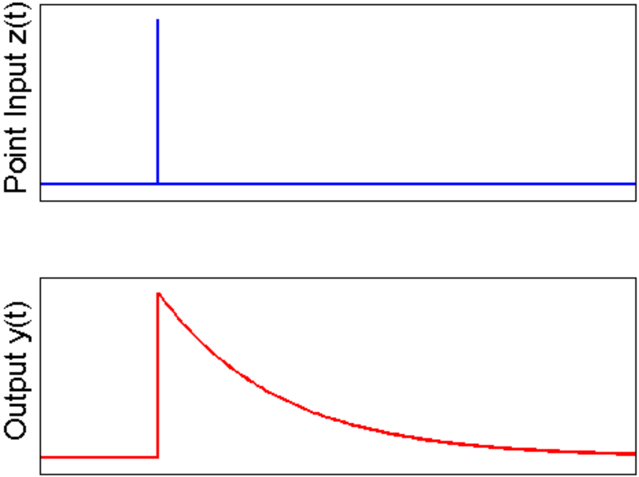
How a diffusion buffer modifies input

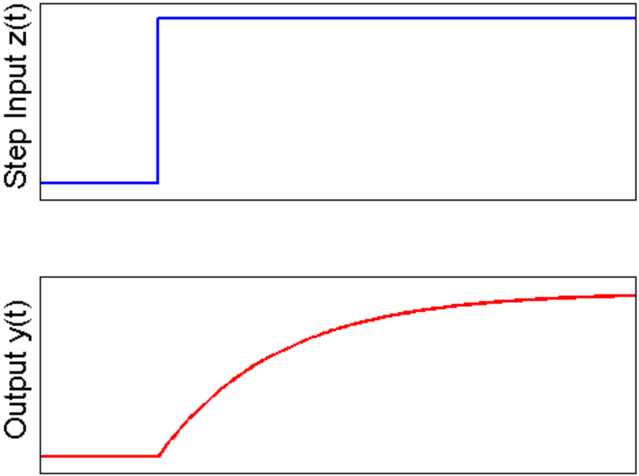
- This suggests a simple modification of our flawed concurrent functional regression model
- We replace

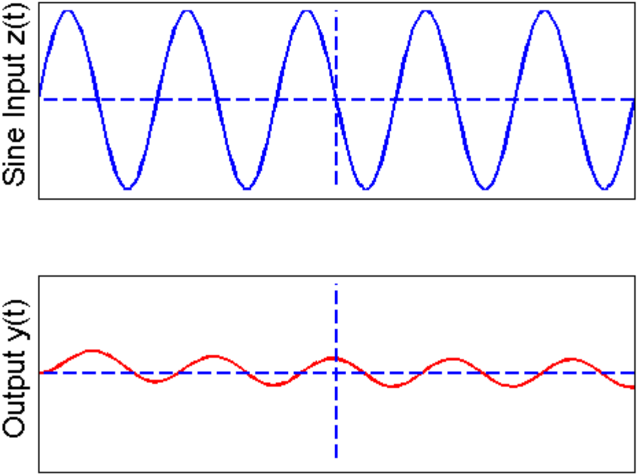
  $$G(t) = G_0 + \alpha R(t) + \epsilon(t)$$

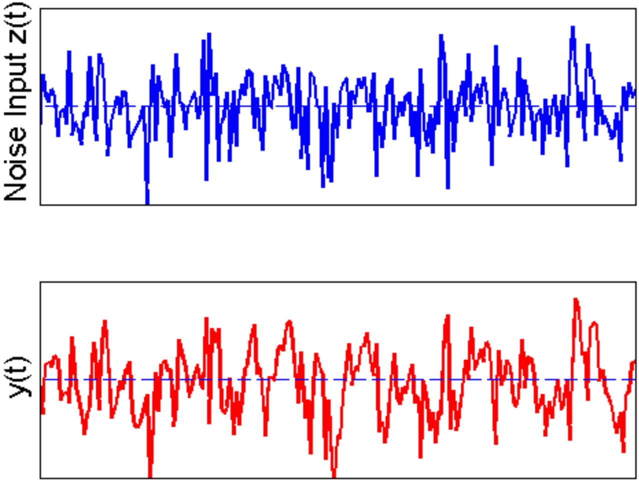
  by

  $$G(t) + \beta DG(t) = G_0 + \alpha R(t) + \epsilon(t)$$
- The following slides indicate how G would respond to four different schematic changes in R.


Point Input z(t)
Output y(t)


Step Input z(t)
Output y(t)


Sine Input z(t)
Output y(t)


Noise Input z(t)
y(t)

Outline

The Collocation Approach

- For each variable x_i in $\mathbf{x}$, we define a basis function expansion
  $$\hat{x}_i(t) = \mathbf{c}_i'\phi_i(t),$$
  where $\mathbf{c}_i$ and ϕ_i are a coefficient vector and a vector of basis functions, respectively.
- Enough basis functions are used to capture any transient localized features. This may require thousands of basis functions.
- Splines are usually the logical choice because of their local support and their capacity to model transient events. Far more knots than data points are often required.
- Discontinuities in a derivative at known locations may require multiple knots.

A Fitting Criterion Conditional on θ

- A criterion $J_i(\mathbf{c}_i|\boldsymbol{\theta}, \lambda_i)$ measures, for variable i,
    - the fidelity of $\hat{x}_i(\mathbf{t}_i|\mathbf{c}_i, \boldsymbol{\theta}, \lambda_i)$ to the data in $\mathbf{y}_i$,
    - how close $\hat{x}_i$ is to being an ODE solution for parameters $\boldsymbol{\theta}$,
    - where smoothing parameter λ_i controls the relative emphasis on these two objectives.
- For example

$$J_i(\mathbf{c}_i) = \|\mathbf{y}_i - \hat{x}_i(\mathbf{t}_i)\|^2 + \lambda_i \int [L_{i,\boldsymbol{\theta}}(\hat{x}_i(t))]^2 dt$$

where

$$L_{i,\boldsymbol{\theta}}(x_i) = Dx_i - f_i(\mathbf{x}, \mathbf{u}, t|\boldsymbol{\theta}).$$

- The first data-fitting term can be -negative log likelihood, or any appropriate function.

The parameter hierarchy

There are three classes of parameters to estimate:

- The coefficients $\mathbf{c}_i$ defining each basis function expansion.
- The model parameters θ defining the ODE.
- The smoothing parameters λ_i.

The roles of the three parameter levels

Coefficients $\mathbf{c}_i$ are *nuisance* parameters because

- they are not of direct interest
- their numbers are apt to vary with the length of the observation interval, density of observation, and other design factors.
- they can be orders of magnitude greater in number than the number of ODE parameters in θ.
- each c_{ik} controls the shape of x_i only *locally* over a small neighborhood of t_{ik}.

Structural and complexity parameters

- Parameters θ are *structural* in the sense of being of primary interest. Our clients want to know their values and precisions.
- Each θ_ℓ affects one or x_i's over a wide region of t-values.
- Smoothing parameters λ_i control the overall *complexity* of the model:
  - $\lambda_i \rightarrow 0 \Rightarrow$ high complexity in $\hat{x}_i$
  - $\lambda_i \rightarrow \infty \Rightarrow$ low complexity in $\hat{x}_i$

The parameter cascade algorithm

- Nuisance parameters are defined as *smooth functions* $\mathbf{c}_i(\theta, \boldsymbol{\lambda})$ of the structural and complexity parameters.
- Structural parameters are defined as *functions* $\theta(\boldsymbol{\lambda})$ of the complexity parameters.
- These functional relationships are defined *implicitly* by specifying a different conditional fitting criterion at each level of the parameter hierarchy.

The multi-criterion optimization strategy

- Nuisance parameter functions $\mathbf{c}_i(\boldsymbol{\theta}, \lambda_i)$ are defined by optimizing the regularized fitting criteria $J_i(\mathbf{c}_i)$ *each time* either $\boldsymbol{\theta}$ or $\boldsymbol{\lambda}$ is modified.

$$J_i(\mathbf{c}_i) = \|\mathbf{y}_i - \hat{x}_i(\mathbf{t}_i)\|^2 + \lambda_i \int [L_{i,\boldsymbol{\theta}}(\hat{x}_i(t))]^2 dt$$

- A purely data-fitting criterion $H(\boldsymbol{\theta})$ is then optimized with respect to the structural parameters $\boldsymbol{\theta}$ alone, such as

$$H(\boldsymbol{\theta}) = \sum_i w_i \|\mathbf{y}_i - \hat{x}_i(\mathbf{t}_i|\boldsymbol{\theta}, \lambda_i)\|^2$$

- Finally, at the top level, a complexity criterion, such as generalized cross-validation, $GCV(\boldsymbol{\lambda})$, is optimized with respect to $\boldsymbol{\lambda}$.

Advantages of parameter cascading

- Gradients and hessians at any level can be readily computed using the *Implicit Function Theorem*.
- Parameter estimates have bias and sampling variances as good as those obtained by other methods.
- Confidence interval estimates for parameters are easy to compute, and appear to have excellent bias and coverage properties.
- Compared to marginalizing out the nuisance parameters using MCMC or or other integration schemes, parameter cascading is
  - much faster,
  - much more stable,
  - much easier to program
  - can be deployed to the user community more conveniently.

Two fits for the price of one

- Data fitting function $\hat{x}_i(t)$ is only approximately a solution of the differential equation. It is a *data-regularized* approximation to a solution.
- But parameter estimates $\hat{\theta}$ and initial values $\hat{x}_i(0)$ permit us to numerically approximate an accurate solution $x_i^*(t)$ of the differential equation that fits the data as well as possible.
- Comparing the two fits can be instructive.

- When λ are small, the optimization criteria **J** and **H** are smooth functions of their arguments, easily optimized, and global optima are easier to locate.
- By ramping up λ slowly we
  - avoid the local minimum problem
  - force $\hat{x}_i(t)$ and $x_i^*(t)$ to coalesce

Outline

1. Introduction
2. How groundwater level responds to rainfall
3. Buffered input-output systems
4. The estimation procedure
5. **Fitting groundwater level**
6. Elementary spread of disease model: A system of diffusors
7. Second order linear diffusion buffers
8. Where we’ve been and where we can go next

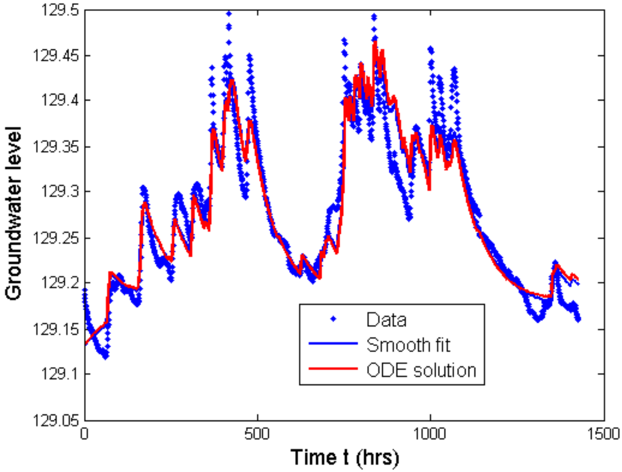
A simple differential equation

$$\frac{dG}{dt} = -\beta G(t) + \alpha R(t - \delta) - \mu$$

where

- β is the speed with which groundwater $G(t)$ reacts to a unit change in rainfall input $R(t)$,
- α defines the impact of a change in $R(t)$; the gain $K = \alpha/\beta$ is the final change in level achieved after a unit increase in $R(t)$, and
- μ is a baseline level, required here because the origin for level $G(t)$ is not meaningful.
- lag δ is the time for rainfall to reach the groundwater level, and is known to be about three hours.

The constant coefficient fit

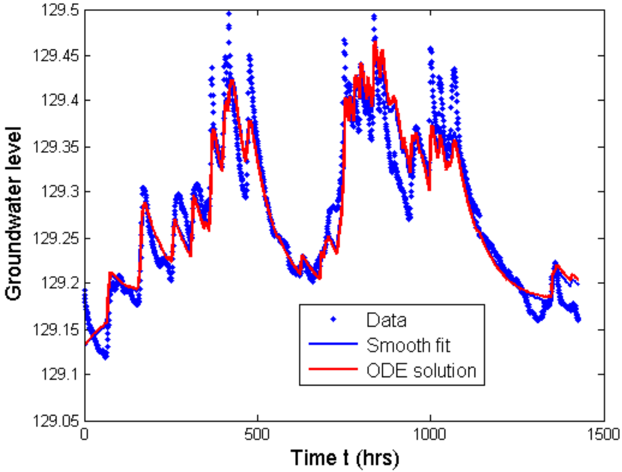


Allowing for long-term trend in ODE parameters

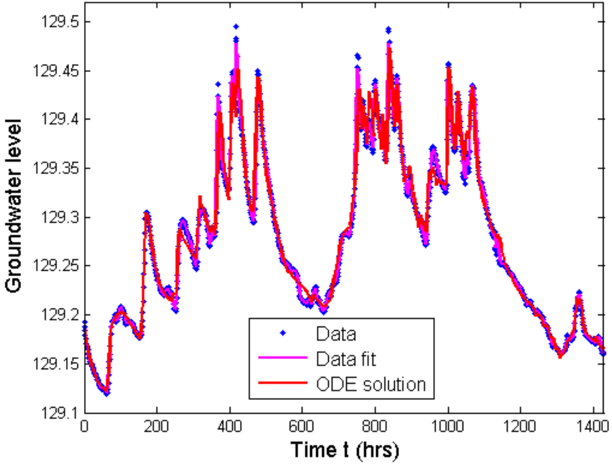
- As groundwater level $G(t)$ changes, the dynamics change, too, because we move through different types of sub-soil structures.
- We weren't given sub-soil transmission rates, so we needed to allow $\beta(t), \alpha(t)$ and $\mu(t)$ to vary slowly over time.

$$\frac{dG}{dt} = -\beta(t)G(t) + \alpha(t)R(t-\delta) + \mu(t)]$$

The constant coefficient fit



The variable coefficient fit



Lessons learned along the way

- The point or delta function nature of the rainfall data plays havoc with numerical initial value solvers, and severely violates Lipschitz continuity.
- Of course, this simple differential equation can be solved explicitly.
- But the solution was useless for data fitting because of it's multiple discontinuities and the large amount of data involved.
- The parameter cascading approach was much easier to work with because it used the differential equation itself rather than the solution.
- Its robustness was essential for working with this problem.

Lessons learned along the way

- The point or delta function nature of the rainfall data plays havoc with numerical initial value solvers, and severely violates Lipschitz continuity.
- Of course, this simple differential equation can be solved explicitly.
- But the solution was useless for data fitting because of it's multiple discontinuities and the large amount of data involved.
- The parameter cascading approach was much easier to work with because it used the differential equation itself rather than the solution.
- Its robustness was essential for working with this problem.

Outline

1. Introduction
2. How groundwater level responds to rainfall
3. Buffered input-output systems
4. The estimation procedure
5. Fitting groundwater level
6. **Elementary spread of disease model: A system of diffusors**
7. Second order linear diffusion buffers
8. Where we’ve been and where we can go next

The players and the script

- $S(t)$ indicates the number of people who can be infected (susceptible) at time t.
- $I(t)$ indicates the number who are infected.
- $R(t)$ indicates the number who were infected, are now recovered, and for a time, at least, are immune.
- The model is

$$\begin{aligned} DS(t) + \alpha S(t)I(t) &= 0 \\ DI(t) + [\gamma - \beta S(t)]I(t) &= 0 \\ DR(t) &= \gamma I(t) \end{aligned}$$

The S equation

$$DS(t) + \alpha I(t)S(t) = 0$$

- Susceptibles S decline to 0 at a rate $\alpha I(t)$
- We see that an input can also affect the speed of response, so that the number of infectives increases the rate of decay rather having a direct or additive effect.
- Otherwise, this is still a standard diffusor, but with a time-varying speed of response

The I equation

$$DI(t) + [\gamma - \beta S(t)]I(t) = 0$$

- Again a diffusor buffer, with a rate of decay that increases with the number of susceptibles, but also with a rate of growth γ that is constant.

The *R* equation

$$DR(t) = \gamma I(t)$$

- A diffusor buffer with the number of recovered's increasing linearly with the number of infected's.
- We learn two things in this simple system of diffusor buffers:
  - Diffusor buffers can be chained together in serial.
  - A variable can impact another variable either additively, or via its speed of response (or both).

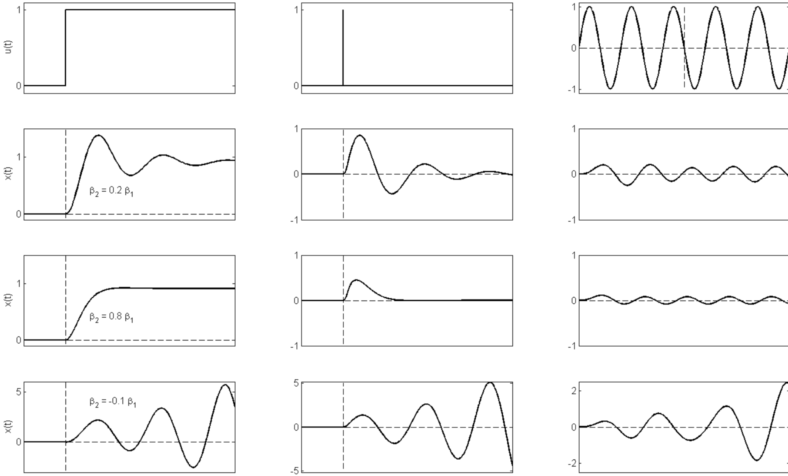
Outline

Two first order buffers in series

- Often two first order buffers are connected in series, as we saw with I and R.
- Pharmacokinetic/pharmacodynamic (PKPD) models use these routinely to model a body's uptake and elimination of a substance.
- Because the input to the first variable affects its rate of change, and it then affects the second variable's rate of change, the net effect is a second order buffer, modeling the behavior of the second derivative:

$$D^2x(t) + \beta_0 x(t) + \beta_1 Dx(t) = u(t)$$

- Here are some ways that a second order linear buffer can respond:


u(t)
x(t)
x(t)
x(t)
β2 = 0.2 β1
β2 = 0.8 β1
β2 = -0.1 β1
1
0
-1
5
-5
2
-2

Outline

Some perspectives

- A very large proportion of the dynamic systems that are used to model real data are systems of diffusion buffers.
- Each buffer systems can have inputs from either variables and from external variables.
- These inputs can be additive, and
- can also modulate the speed of the buffer's reaction.
- Partial differential equations are usually diffusion buffer systems in time combined with spatial buffers.
- The main challenge for the modeler is to identify the causal links.

Where to go to learn more

- Mathematical texts on differential equations can be less then helpful. They tend to assume smooth systems (Lipschitz continuity), over-emphasize systems of mainly mathematical or physical interest, and mostly consider only initial state values as data identifying the system.
- Some excellent model-oriented texts have emerged in recent years. Borelli, R. L. and Coleman, C. S. (1998) *Differential Equations: A Modeling Perspective.* Wiley, is a good example.
- The engineering community can be counted on for readable accounts, especially the chemical engineers. Marlin, T. (2000) *Process Control*. McGraw-Hill, is excellent.

Software for identifying dynamic systems from noisy data

- General purpose functions in both R and Matlab have been developed by Giles Hooker, and can be obtained at http://www.bscb.cornell/ hooker/.
- Matlab and R functional data analysis software is used. A set of tools Matlab and R is available from http://www.functionaldata.org.
- A useful resource is Ramsay, J. O., Hooker, G. and Graves, S. (2009) *Functional Data Analysis with R and Matlab*. Springer.