Improvement of bioengineering courses through systems biology and bioprocess modeling

Yinjie Tang¹, Kirk Dolan²,³, Wei Liao²
Department of Energy, Environmental and Chemical Engineering; Washington University, St. Louis
Department of Biosystems & Agricultural Engineering, Michigan State University, East Lansing, MI
Department of Food Science & Human Nutrition, Michigan State University
Objectives

• To develop new curricula for bioengineering classes to bridge the gap between systems biology and bioprocess engineering.
• To train students how to use MATLAB, Simulink, and SBPD to model the bioprocessing and systems biology.
Motivations

Industrial biotechnology often uses microorganisms and enzyme catalysts to synthesize useful products. The benefits of biological reactions in large quantities cannot be realized without using systems biology, bioprocess dynamic control and modeling theory.
The critical frontiers for bioprocess engineering students are direct experience with:
• systems biological analysis;
• microbiology;
• metabolic engineering;
• bioreactor operation and control;
• modeling of dynamic behavior of metabolic reactions during fermentation.
Courses

1. Metabolic engineering (ChE596, WUSTL): This class teaches molecular tools for pathway modifications, systems biology, and metabolic modeling.

2. Process Control (ChE 462, WUSTL) and Process Control Laboratories (ChE 463, WUSTL) teaches process dynamics and control theory.

3. Bioprocess Engineering (ChE 453, WUSTL) focuses on enzyme kinetics, microbiology, bio-reaction modeling, metabolic models (stoichiometry models and flux balance models), process scale-up and product separations.

4. Microbial Systems Engineering (BE 360, MSU) teaches biological principles, computational tools for the analysis of microbial processes, kinetic analysis of bioprocesses, modeling of microbial processes, unit operations and scale-up.

5. Engineering Analysis and Optimization of Biological Systems (BE 835, MSU) has two parts: 1) numerical techniques and the forward problem; and 2) parameter estimation and inverse problems. Other topics include experimental design, sequential parameter estimation, model discrimination, and Monte Carlo simulation.
Fermentation Engineering Project (ChE453 and 463)
Note: students take both ChE453 (BioProcess Engineering) and ChE463 (Process Lab) in the spring semester. They did a fermentation project including three parts.

(Part 1: Fermentation Lab Experience)

(Part 2: Kinetic Modeling using MATLAB)

(Part 3: Metabolic Flux Analysis)
Par 1: Fermentation lab (one week)

- Students learn the fundamental concept for bioreactor operations and microbial cultures.
- Students learn how to handle and analyze biological samples using enzyme kit, GC-MS and spectrometry.
- Students learn how to design the experiments, record data, and write the report.
- Students experience team work and leadership.

Undergraduate students performed ethanol fermentation (2011).
Students apply MATLAB and develop kinetic models (using ode45) and perform the parameter fitting and statistical analysis (using nlinfit functions).

**Biomass**
\[
\frac{d(X)}{dt} = V \mu_{\text{max}} X \frac{S}{S + K_S} \frac{[O]}{K_O + [O]} \frac{N}{K_N + N} - k_d X
\]

**Ethanol**
\[
\frac{d(P)}{dt} = V X \frac{S}{K_I} + S + K_S
\]

**Sugar**
\[
\frac{d(S)}{dt} = F S_f - Y_{S/P} \frac{d(X)}{dt} - Y_{S/P} \frac{d(P)}{dt}
\]

**Nitrogen source**
\[
\frac{d(N)}{dt} = - \frac{\mu X}{Y_{X/N}}
\]

**Total mass**
\[
\frac{dV}{dt} = F
\]

Simulation and parameter fitting of fed batch fermentation data
Part 3: Metabolic flux analysis of ethanol fermentation

Determine intracellular metabolism during ethanol fermentation using simplified metabolic model and linear optimization.

Flux Balance Analysis is an important tool to determine:
1. Metabolic flux distributions.
2. Theoretical product yield.
3. The bottleneck pathway for product synthesis.
4. Mutants’ physiologies
5. Cell metabolism under different fermentation conditions.

Metabolic Flux: the *in vivo* enzymatic reaction rates in the metabolic network (a systems biology description).

TA: Xueyang Feng
(current a new professor at Virginia Tech)
Flux Balance Analysis (FBA)

- *in silico* simulation
- Linear programming (LP)
- Constraint based analysis
- Objective function: maximize biomass growth

\[
\text{maximize } \sum c_i \cdot v_i \\
\text{s.t. } S \cdot v = 0 \\
\text{lb} < v < \text{ub}
\]
Flux Balance Analysis (FBA)

16 fluxes, 8 intracellular metabolites

\[ \text{G6P} : v_1 = v_2 + v_3 + v_{16} \]

\[ \text{R5P} : v_2 = v_4 \]

\[ \text{Pyr} : 2 \cdot v_3 + v_4 = v_5 + v_{11} + v_{15} \]

\[ \text{AcCoA} : v_5 = v_6 + v_7 + v_{14} \]

\[ \text{ICIT} : v_7 = v_8 \]

\[ \text{AKG} : v_8 = v_9 + v_{12} \]

\[ \text{SUC} : v_9 = v_{10} \]

\[ \text{OAA} : v_{10} + v_{11} = v_7 + v_{13} \]

The transport fluxes were measured:

\[ v_1 = 11.0 \text{ mmol/g DCW/h} \]

\[ v_6 = 6.4 \text{ mmol/g DCW/h} \]

The building block fluxes can be assumed from biomass composition:

\[ v_{12} = 1.078 \cdot \mu \]

\[ v_{13} = 1.786 \cdot \mu \]

\[ v_{14} = 2.928 \cdot \mu \]

\[ v_{15} = 2.833 \cdot \mu \]

\[ v_{16} = 0.205 \cdot \mu \]

\( \mu \) is the biomass growth rate

17 variables

15 equations

Freedom = 2
Under assumption of $S \cdot v = 0$, we need to maximize $\mu$ to determine fluxes which produce the highest biomass growth.
Flux Balance Analysis (FBA)

Use “Optimization Toolbox” for Flux Analysis

Use “linprog” for FBA

Change to “Medium scale-simplex”

Put the objective vector

Options to stop the optimization

Maximize $\mu$

Constraints of fluxes

$$\text{obj} = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]^T$$

$$\text{lb} = [11.0 \ 0 \ 0 \ 0 \ 0 \ 6.4 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$$

$$\text{ub} = [11.0 \ 20 \ 20 \ 20 \ 20 \ 6.4 \ 20 \ 20 \ 20 \ 20 \ 20 \ 20 \ 20 \ 20 \ 20 \ 20]^T$$
FBA results

**Summary**
The three week project covers multi-scale cellular processes, which trains students with skills in both macroscopic bioreactor engineering and microscopic metabolism analysis.

---

### Optimization Tool

**Problem Setup and Results**

- **Solver**: Simplex
- **Algorithm**: Medium scale - simplex

<table>
<thead>
<tr>
<th>Constraints:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear inequalities:</td>
</tr>
<tr>
<td>Linear equalities:</td>
</tr>
<tr>
<td>Bounds:</td>
</tr>
</tbody>
</table>

**Start point:**
- Let algorithm choose point
- Specify point:

**Run solver and view results**

- Start
- Pause
- Stop

Current Iteration: 0

Optimization running.
Optimization terminated.
Objective function value: -1.5425689705319
Optimization terminated.

<table>
<thead>
<tr>
<th>Find point:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>17</td>
</tr>
</tbody>
</table>

---

**Fermentation Engineering**

**Lab Exp.**

**Reactor Models**

**Kinetic Modeling**

**Systems biology**

**Flux Analysis**
Creep compliance of a wheat protein film
Using formaldehyde cross-linker

- Creep compliance of a wheat protein film (determination of retardation time and free dashpot viscosity in the Jefferys model)

\[ J = J_1 (1 - \exp\left(\frac{-t}{\lambda_{ret}}\right)) + \frac{t}{\mu_0} \]

Where \( J \) is the strain, \( J_1 \) is the retarded compliance (Pa\(^{-1}\)); \( \lambda_{ret} = \mu_1/G_1 \) is retardation time (s); \( \mu_0 \) is the free dashpot viscosity (Pa s); \( t \) is the time.

- The recovery of the compliance is following the equation (\( t > t_1 \)):

\[ J = J_1 \exp\left(-\frac{t - t_1}{\lambda_{ret}}\right) \]

Where \( t_1 \) is the time the stress was released.
Creep compliance of a wheat protein film

Simulink model

Parameters: $J_1 = a = 0.38 \text{ Mpa}^{-1}$; $\lambda_{ret} = b = 510.6 \text{ s}$; $\mu_0 = c = 260800 \text{ Mpa s}$

\[
J = J_1 \left(1 - \exp\left(\frac{-t}{\lambda_{ret}}\right)\right) + \frac{t}{\mu_0}
\]

Example 3.

creepCompliance
Creep compliance of a wheat protein film

Simulation result
Simulink Modeling – Ethanol Fermentation

A. Fermenter

Regular fermenter

- Agitation
- Heater
- Products
- Biomass
- Aeration

Ethanol fermenter

- Heater
- Ethanol
- Biomass
B. Parameters of Fermenter

- The dimension of the tank: 5 x 7 m (diameter x height), effective height 5 m.
- Effective volume: 100 m³
- The inside area of the tank (A): 100 m²
- Tank material: Stainless steel
- Wall thickness: 0.02 m
- Thermal conductivity of stainless steel (k): 0.021 kw/m K
- Convection heat transfer coefficient of air (hₐ): 0.05 kw/m² K
- Convection heat transfer coefficient of liquid (hₐ): 0.5 kw/m² K
- Fermentation method: Batch
- Environmental temperature: 25 °C

*: from Perry Chemical Engineering Handbook
C. Heating system

❖ The required heat

\[ Q = MC_p(T - T_i) \]

\[ C_p = 3.95 \text{ kJ/kg K} \]
\[ M = 100,000 \text{ Kg} \]

Where \( Q \) is the heat required; \( M \) is the total mass of broth; \( C_p \) is the heat capacity of the broth; \( T \) is the target inside temperature; \( T_i \) is the real inside temperature.

❖ The heat loss

\[ q = UA(T_i - T_o) \]

\[ U = \frac{1}{1 + \frac{d}{h_a} + \frac{1}{k} + \frac{1}{h_l}} \]

\[ A = 100 \text{ m}^2 \]
\[ U = 0.045 \text{ kJ/m}^2 \text{ K} \]

Where \( T_o \) is the outside temperature; \( A \) is the inside area; \( U \) is the overall heat transfer coefficient. \( h_a \) and \( h_l \) are heat transfer coefficients for air and liquid inside of the tank. \( d \) is the thickness of the tank wall. \( k \) is the thermal conductivity of stainless steel.
The yeast strain: *S. cerevisiae*

The culture conditions: Temperature of 30°C, pH 5.0, anaerobic condition

The glucose concentration (g/L): 50-150

The initial inoculum (g/L): 1

The biomass yield from substrate (g/g): 0.15

The product yield from biomass (g/g): 5.33

Substrate constant (Ks, g/L): 0.025

Substrate maintenance rate constant (ms, g substrate/g biomass/hr): 0.036 = 0.00001 g/g/s

Product maintenance rate constant (mp, g product/g biomass/hr): ~0

μ_{max} (g/L/hr): 0.045 = 0.000013 (g/L/s)

The toxic power (η): 1

The maximum product concentration at which the growth is completely inhibited (P_{max}) (g/L): 112
D. Kinetics of ethanol fermentation

Substrate consumption

\[
\frac{ds}{dt} = -r_s = -\left[ \frac{\mu_{max}s}{Y_{XS}(K_s + s)} \left(1 - \frac{p}{p_{max}} \right)^\eta + m_s \right]x
\]

Ethanol production

\[
\frac{dP}{dt} = r_P = \left[ Y_{PX} \frac{\mu_{max}s}{K_s + s} \left(1 - \frac{p}{p_{max}} \right)^\eta + m_P \right]x
\]

Biomass accumulation

\[
\frac{dx}{dt} = r_x = \frac{\mu_{max}s}{K_s + s} \left(1 - \frac{p}{p_{max}} \right)^\eta x
\]
Simulink model

\[ Q = MC_p (T - T_i) \]
\[ q = UA (T_i - T_o) \]

Heating subsystem

\[ \frac{dS}{dt} = -r_s = - \frac{\mu_{\text{max}} s}{Y_{XS} (K_S + s)} \left(1 - \frac{p}{p_{\text{max}}} \right)^n + m_s \]

Kinetic subsystem

\[ \frac{dP}{dt} = r_p = \frac{\mu_{\text{max}} s}{K_S + s} \left(1 - \frac{p}{p_{\text{max}}} \right)^n + m_p \]

\[ \frac{dx}{dt} = r_x = \frac{\mu_{\text{max}} s}{K_S + s} \left(1 - \frac{p}{p_{\text{max}}} \right)^n x \]
Simulation results
Example from MSU BE 835

Objective: Teach students MATLAB skills for solving bioprocess problems

Graduate course was in two parts:

Part I (~40%)—The Forward Problem


Part II (~60%)—The Inverse Problem

Modeling: Forward or Inverse Problem?

**Forward Problem**
Given:
- microorganism,
- Time-temp history,
- Parameter values $D_0$, $z$

Compute log $N(t)$

$$
\log \frac{N(0)}{N(t)} = \frac{1}{D_0} \int_0^t 10 \left( \frac{T(t) - 250}{z} \right) dt
$$

**Inverse Problem**

- Experimental log $N(0)$, $\log N(t_1), \log N(t_2), \ldots \log N(t_n)$
- And a known model
- Initial guesses of $D_0$, $z$

Estimate $D_0$, $z$
Compute log reductions = $\log N_{\text{predicted}}$
Examples of inverse problems
Systematic Method

• **Step 1.** Choose the model.

• **Step 2.** Choose initial parameter values based on experience, literature, theory, linear approximation, etc.

• **Step 3.** Plot scaled sensitivity coefficients.

• **Step 4.** Perform inverse problem with OLS, sequential, or other method. Report statistics.

• **Step 5.** Plot residuals and test them against assumptions.

• **Step 6.** (Only for Arrhenius- or Bigelow-type secondary models.) Perform inverse problem for a range of Tref. Determine optimum reference temperature.

Statistical Assumptions about the errors

1. Additive;
2. Zero mean;
3. Constant variance;
4. Uncorrelated;
5. Normally distributed.
Sensitivity Coefficients = \( \frac{\partial \eta}{\partial \beta_i} \)

**Scaled Sensitivity Coefficients** = \( \beta_i \frac{\partial \eta}{\partial \beta_i} \)

Large response

\( \eta_1 \)  

Small response

\( \eta_2 \)

\( \eta \) is the model, log \( N \)

\( \beta_i \) is the ith parameter
What Insights can Scaled Sensitivity Coefficients (X’) Give?

• X’ can be plotted before the experiment is run as a map guiding the parameter estimation.

• The largest X’ will have the smallest relative error = standard deviation/estimate.

• Correlated X’ means potential estimation difficulties.

• Small X’(< ~5% of the span of \( \eta \)) means the parameter is insignificant and potentially can be eliminated from the model.
Parameter Estimation Method

Sequential estimation updates the parameter estimates as each data point is added.

• Provides insight to the estimation process.
• We expect the parameters to approach a constant before the end of the experiment.

\[ A_{i+1} = P_i X_{i+1}^T \]
\[ \Delta_{i+1} = \phi_{i+1} + X_{i+1} A_{i+1} \]
\[ K_{i+1} = A_{i+1} \Delta_{i+1}^{-1} \]
\[ e_{i+1} = Y_{i+1} - \hat{Y}_{i+1} \]
\[ b_{i+1}^* = b_i^* + K_{i+1} \left[ e_{i+1} - X_{i+1} \left( b_i^* - b \right) \right] \]
\[ P_{i+1} = P_i - K_{i+1} X_{i+1} P_i \]

Case Study—microbial inactivation

• *Salmonella* in 66% sugar liquid medium (Mattick et al., 2001)

• Four different dynamic temperature profiles:
  – Heating rate #1: 9°C/min
  – Heating rate #2: 5°C/min
  – Heating rate #3: 4.5°C/min
  – Heating rate #4: 2.7°C/min

Raw data: \( \log N/N_0 \) and \( T(t), 9^\circ C/\text{min}(\text{red}) \)
Raw data: $\log(N/N_0)$ and $T(t), \ 5^\circ C/min$(green)
Raw data: \( \log \frac{N}{N_0} \) and \( T(t) \), 4.5\(^\circ\)C/min(blue)
Raw data: \( \log(N/N_0) \) and \( T(t) \), 2.7\(^\circ\)C/min (black)
Model

Step 1. Primary: Weibull, differential form

where:

\[
\frac{d \log S(t)}{dt} = -b(T)nt^{n-1}
\]

\[S(t) = \frac{N(t)}{N_0}, \text{ survival fraction}\]

\[N(t) \text{ is microbial concentration, cfu/mL}\]

\[N_0 \text{ is mean initial microbial concentration, cfu/mL}\]

Initial value: \(\log S(0) = 0\)

Secondary: log logistic\n
\[b(T) = \ln \{1 + \exp[k(T - T_c)]\}\]

Where: \(k\) and \(T_c\) are parameters.

Summary: \(\log S(t)\) dependent variable, \(T\) & \(t\) are independent variables

Three parameters to be estimated:

\[k \ (^\circ C^{-1}), \ T_c \ (^\circ C)\ \text{and} \ n \ (\text{dimensionless})\]
What b(T) looks like

\[ b(T) = \log_e (1 + \exp[k(T - T_c)]) \]

- \( k = 0.5 \degree \text{C}^{-1} \) 
  - \( T_c = 55\degree \text{C} \)
- \( k = 0.3 \degree \text{C}^{-1} \) 
  - \( T_c = 70\degree \text{C} \)

Step 2. Initial Parameter Guesses

- From Mattick et al. (2001)
- \( k = 0.5 \, ^\circ\text{C}^{-1} \)
- \( T_c = 62 \, ^\circ\text{C} \)
- \( n = 0.6 \)
Step 3. Scaled Sensitivity Coefficients—9°C/min
Scaled Sensitivity Coefficients—5°C/min (green)

Scaled sensitivity coefficient, (log[cfu/mL])

- $X'_k$
- $X'_{Tc}$
- $X'_n$

Time (min)
Scaled Sensitivity Coefficients—4.5°C/min (blue)
Scaled Sensitivity Coefficients—2.7°C/min (black)
Step 4. Inverse Problem Solution Methods

• Solution to the differential equation: Runge-Kutta fourth and fifth order adaptive numerical method
  – MATLAB using ode45

• Solution to the inverse problem:
  – Ordinary Least Squares using MATLAB’s nlinfit function.
Step 4. Inverse Problem—Results for logS vs. t

\[ \log S = \log \left( \frac{N}{N_0} \right) \]
## Table of Results

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Estimate</th>
<th>Std Error</th>
<th>Rel Error</th>
<th>95% Conf Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k , (°C^{-1})$</td>
<td>0.67</td>
<td>0.152</td>
<td>22.80%</td>
<td>0.36</td>
</tr>
<tr>
<td>$T_c , (°C)$</td>
<td>54.13</td>
<td>0.883</td>
<td>1.63%</td>
<td>52.37</td>
</tr>
<tr>
<td>$n$</td>
<td>0.38</td>
<td>0.0477</td>
<td>12.44%</td>
<td>0.29</td>
</tr>
</tbody>
</table>

### Correlation Matrix

<table>
<thead>
<tr>
<th></th>
<th>$k , (°C^{-1})$</th>
<th>$T_c , (°C)$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k , (°C^{-1})$</td>
<td>1.000</td>
<td>-0.083</td>
<td>-0.943</td>
</tr>
<tr>
<td>$T_c , (°C)$</td>
<td>-0.083</td>
<td>1.000</td>
<td>0.400</td>
</tr>
<tr>
<td>$n$</td>
<td>-0.943</td>
<td>0.400</td>
<td>1.000</td>
</tr>
</tbody>
</table>
Step 5. Residuals

**Observed logS - Predicted logS**

**Standard Statistical Assumptions:**
- Additive
- Zero mean
- Constant variance
- Uncorrelated
- Normally distributed
Modify the model

• Residuals biased. Mean of residuals ≠ 0.
• Too many negative residuals, especially middle two heating rates.
• Initial values are not necessarily exactly the same, due to variability in logN(0).
• Modify by:
  – replacing logS with logN;
  – Adding four parameters: logN(0)₁, logN(0)₂, logN(0)₃, logN(0)₄.
Inverse Problem—Results for OLS for logN vs. t

![Graph showing the relationship between logN (log[cfu/mL]) and Time (min) vs. Temperature (°C).]
Residuals for logN vs. t

Standard Statistical Assumptions:
- Additive
- Zero mean
- Constant variance
- Uncorrelated
- Normally distributed
## Parameter Results for logN

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Std Error</th>
<th>Rel Error</th>
<th>95% Conf Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>AICc</td>
<td>-160.53</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSE (log10(cfu/mL))</td>
<td>0.374</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSE</td>
<td>0.140</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean of residuals</td>
<td>-3.50E-09</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Std Error</th>
<th>Rel Error</th>
<th>95% Conf Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>k (°C⁻¹)</td>
<td>0.67</td>
<td>0.18</td>
<td>26.43%</td>
<td>0.32</td>
</tr>
<tr>
<td>Tc (°C)</td>
<td>56.90</td>
<td>0.95</td>
<td>1.66%</td>
<td>55.02</td>
</tr>
<tr>
<td>n</td>
<td>0.42</td>
<td>0.06</td>
<td>13.72%</td>
<td>0.30</td>
</tr>
<tr>
<td>logN(0)1</td>
<td>6.77</td>
<td>0.10</td>
<td>1.51%</td>
<td>6.56</td>
</tr>
<tr>
<td>logN(0)2</td>
<td>6.63</td>
<td>0.12</td>
<td>1.82%</td>
<td>6.39</td>
</tr>
<tr>
<td>logN(0)3</td>
<td>6.72</td>
<td>0.08</td>
<td>1.17%</td>
<td>6.57</td>
</tr>
<tr>
<td>logN(0)4</td>
<td>6.75</td>
<td>0.09</td>
<td>1.34%</td>
<td>6.57</td>
</tr>
</tbody>
</table>
Step 5 continued. Sequential Results

Normalized Sequentially Estimated Parameter

logN(t)

k °C⁻¹
Tc
n
logN01
logN02
logN03
logN04
Sequential Results

Normalized Sequentially Estimated Parameter

\[ \text{logN}(t) \]

-1.5 to 2

Parameters:
- \( k \) °C^{-1}
- \( T_c \)
- \( n \)
- \( \text{logN01} \)
- \( \text{logN02} \)
- \( \text{logN03} \)
- \( \text{logN04} \)

Graphical representations show changes in these parameters over time.
Outputs

• Undergraduate and graduate engineering students learned both MATLAB and Simulink with application to bioengineering
• Slides and course syllabus are posted for free use at the website: http://tang.eece.wustl.edu/MATLAB_WUSTL.htm
• Two Journal articles published (in *Industrial & Engineering Chemistry Research, Inverse Problems in Science & Engineering*) by student using the methods learned in the course.
• Three book chapters published based on these methods.
• BE 835 selected in 2012 as a required course for graduate students in the department.
• Increased use of MATLAB and Simulink in undergraduate projects and graduate research, and improvement of the quality of the academic research.
• Yinjie Tang received a department chair’s award for outstanding teaching (2013)
• Two graduate students at WUSTL received TA awards for teaching process control classes (2012, 2013)
Acknowledgments

• Thanks to MathWorks for Education Grant Support