

Studying Adaptation and Homeostatic Behaviors of Kinetic Networks by using MATLAB

Tormod Drengstig

Department of Electrical Engineering and Computer Science,
University of Stavanger, Stavanger, Norway

Thomas Kjosmoen

Department of Electrical Engineering and Computer Science,
University of Stavanger, Stavanger, Norway

Peter Ruoff*

Centre for Organelle Research,
Faculty of Science and Technology,
University of Stavanger, Stavanger, Norway

December 3, 2009

*Corresponding author. Address: Centre for Organelle Research, Faculty of Science and Technology, University of Stavanger, N-4036 Stavanger, Norway, Tel.: (47) 5183-1887, Fax: (47) 5183-1750, Email: peter.ruoff@uis.no

Abstract

Organisms have the ability to counteract environmental perturbations and keep certain components within a cell homeostatically regulated. Closely related to homeostasis is the behavior of perfect adaptation where an organism responds to a step-wise perturbation by regulating some of its components, after a transient period, to their original pre-perturbation values. A particular interesting type of model relates to so-called robust behavior where the homeostatic or perfect adaptation property is independent of the magnitude of the applied step-wise perturbation. It has been shown that this type of behavior is related to the control-theoretic concept of integral feedback (or integral control). Using downloadable MATLAB examples, we demonstrate how robust perfect adaptation sites can be identified in reaction kinetic networks by linearizing the system, applying the Laplace transform and inspecting the transfer function. We also show how the homeostatic set-point in perfect adaptation is related to the presence of zero-order fluxes.

Key words: MATLAB, kinetic networks; metabolic control theory; control engineering; transfer functions; control coefficients; adaptation

1 Introduction

The capability of yeast and other organisms (and part of organisms) to adapt to environmental changes in nutrition (1–9), light (10–12), temperature (13, 14) or other stressors appear essential for an organism’s fitness and survival. There are various adaptation modes (15, 16), which range from no adaptation at all, to partial adaptation, perfect adaptation, and overadaptation (Fig. 1.1).

[Fig. 1.1 approximately here]

There is a considerable interest in perfect adaptation which describes the response during a step-wise perturbation by maintaining some of the variables (concentration/fluxes) to their original pre-perturbation values. Perfect adaptation has been found, for example in bacterial (2–8) and eukaryotic (9) chemotaxis, osmoregulation in yeast (17), photoreceptor responses (10, 11), MAP-kinase regulations (18–20), as well as temperature homeostasis in circadian and ultradian rhythms (generally referred to as temperature compensation).

There are two ways in which to consider how perfect adaptation may be understood. In the first case perfect adaptation is the result of a fine-tuning or balancing between rate parameters. This mode of adaptation is considered to be non-robust (21) because any change in a rate parameter will disrupt the balance and the adaptation behavior. In the second approach perfect adaptation is the result of a network property, which does not need a fine-tuning in most of the parameters. Yi et al. (22) showed that this second form of adaptation can be described in terms of an integral feedback (also called integral control); a concept used in control theory (23).

[Fig. 1.2 approximately here]

Fig. 1.2 illustrates the principles of integral feedback regulation. The error e between a reference signal (set-point) and the output (CV, controlled variable) is integrated, processed together with a perturbation and then fed back again to calculate the error again. In this way a closed loop is generated, where the output value of the system converges to the set-point and error e approaches zero. While the integral feedback principle is quite general, it is not obvious how to identify potential (robust) perfect adaptation sites in ki-

netic networks and how set-points can be interpreted in terms of a molecular mechanism.

In the following section we will illustrate, by using MATLAB, how robust perfect adaptation sites can be identified (24) and how zero-order fluxes can become important in defining set-points of homeostatic controllers (25).

2 Predicting Robust Perfect Adaptation Sites in Reaction Kinetic Networks

The procedure for identifying robust perfect adaptation sites consists of the following steps, which are described in detail in the next section, together with the MATLAB commands.

1. Define a state space model of the reaction kinetic network in question. This is a set of coupled first-order differential equations. The input variables on which step-wise perturbations are performed are generally the rate constants k_n , where n is an index for the reaction associated with k_n . The output variables are either the concentrations of the molecular components or reaction velocities (fluxes) or a combination of them (see below).
2. Perform a linearization of the network model (if possible). In some cases it is difficult or impossible to find a linearized model.
3. Laplace-transform the linearized network. In kinetics we generally think of a (perturbed) rate constant k_n or a concentration I_m of species m to be a function in time t , such as $k_n(t)$ or $I_m(t)$. The Laplace transform $F(s)$ of function $f(t)$ is defined as

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^{\infty} e^{-st} f(t) dt \quad (1)$$

and becomes a function of the complex-valued s - or frequency space. The advantage of working in s -space is that the differential equations are trans-

formed into algebraic equations, which are often easier to analyze and handle.

4. Calculate the transfer functions $H_{y_p, k_n}(s) = \frac{\Delta y_p(s)}{\Delta k_n(s)}$ between a small change in the Laplace transformed input elements $\Delta k_n(s)$ (i.e. rate constants) and the corresponding small change in the Laplace transformed output elements $\Delta y_p(s)$ (e.g. concentration or fluxes). The transfer function elements $H_{y_p, k_n}(s)$ are part of the transfer function matrix $\mathbf{H}(s)$ from the vector of input elements to the vector of output elements. In general, the elements of the transfer function matrix are written as:

$$H_{y_p, k_n}(s) = \frac{\Delta y_p(s)}{\Delta k_n(s)} = \frac{K \cdot \prod_{r=1}^n \left(-\frac{1}{z_r} s + 1 \right)}{\prod_{q=1}^m \left(-\frac{1}{p_q} s + 1 \right)} = \frac{n(s)}{d(s)} \quad (2)$$

where z_r are defined as the transfer function's zeros, p_q are the poles, and K is the gain.

As indicated by Eq. 2, $H_{y_p, k_n}(s)$ is described as the ratio between two polynomials, the numerator-polynomial $n(s)$ and the denominator-polynomial $d(s)$. The solution to $n(s) = 0$ (i.e. the position of the zeros z_r in the complex plane) indicates the type of adaptation behavior. Fig. 2.1 shows the different $n(s)$ polynomials that relate to the four adaptation types shown in Fig. 1.1. When, for a given $H_{y_p, k_n}(s)$, $n(s)$ has a zero in origo of the s -plane regardless of the values of any of the rate constants, the output shows robust perfect adaptation with respect to a stepwise increase of the rate constant considered as input. For a more detailed discussion about the influence of the transfer function's poles on adaptation kinetics we refer to Ref. (24).

[Fig. 2.1 approximately here]

2.1 Detailed Outline of the Principles

Consider a reaction kinetic network with M chemical components (I_m) and N reaction steps, where each step n is associated with a rate constant k_n . The network can be stimulated by changing one of the rate constants (k_n) by means of a step function. Such a stimulation may occur due to a signal

coming from a receptor acting specifically on k_n . In this respect, the rate constants are considered to be time-dependent. The kinetics of the network are described by the rate equations for each chemical component I_m :

$$\frac{dI_m(t)}{dt} = f_m(k_1(t), \dots, k_N(t), I_1(t), \dots, I_M(t)) \quad (3)$$

From this model we define P outputs, described as y_p (the *model output*), which are the different properties of the network we want to investigate. For instance, these outputs can be concentrations $I_m(t)$, fluxes $J_n(t)$, or other network properties which depend on concentrations and/or rate constants. Hence, the P nonlinear output models are given by:

$$y_p(t) = g_p(k_1(t), \dots, k_N(t), I_1(t), \dots, I_M(t)) \quad (4)$$

In order to find the *transfer function matrix* $\mathbf{H}(s)$ from the input (changes in rate constants) to the output candidates y_p , Eqs. 3 and 4 are first linearized around the (unperturbed) steady state (ss) values $\mathbf{I}_{ss} = [I_1, \dots, I_M]$, $\mathbf{y}_{ss} = [y_1, \dots, y_P]$, and the pre-perturbation values of the rate constants $\mathbf{k}_{ss} = [k_1, \dots, k_N]$ (note the independence of time t of the vector elements to indicate steady state values), giving the following linear state space model:

$$\Delta \dot{\mathbf{I}}(t) = A \cdot \Delta \mathbf{I}(t) + B \cdot \Delta \mathbf{k}(t) \quad (5)$$

$$\Delta \mathbf{y}(t) = C \cdot \Delta \mathbf{I}(t) + D \cdot \Delta \mathbf{k}(t) \quad (6)$$

where $\Delta \mathbf{k}(t) = [\Delta k_1(t), \dots, \Delta k_N(t)]^T$, $\Delta \mathbf{I}(t) = [\Delta I_1(t), \dots, \Delta I_M(t)]^T$ and $\Delta \mathbf{y}(t) = [\Delta y_1(t), \dots, \Delta y_P(t)]^T$ are vectors of small deviations around \mathbf{k}_{ss} , \mathbf{I}_{ss} and \mathbf{y}_{ss} , respectively. The *state matrix* $A \in \mathbb{R}^{M \times M}$, the *input matrix* $B \in \mathbb{R}^{M \times N}$, the *output matrix* $C \in \mathbb{R}^{P \times M}$ and the *feed-through matrix* $D \in \mathbb{R}^{P \times N}$ are defined as

$$A_{ij} = \left. \frac{\partial f_i}{\partial I_j} \right|_{ss} \quad (7)$$

$$B_{ij} = \left. \frac{\partial f_i}{\partial k_j} \right|_{ss} \quad (8)$$

$$C_{ij} = \left. \frac{\partial g_i}{\partial I_j} \right|_{ss} \quad (9)$$

$$D_{ij} = \left. \frac{\partial g_i}{\partial k_j} \right|_{ss} \quad (10)$$

Laplace-transforming the linearized model in Eqs. 5 and 6, gives the *transfer function* matrix $\mathbf{H}(s) \in \mathbb{R}^{P \times N}$ as (23):

$$\mathbf{H}(s) = \frac{\Delta \mathbf{y}(s)}{\Delta \mathbf{k}(s)} = C(sI - A)^{-1}B + D \quad (11)$$

where $I \in \mathbb{I}^{M \times M}$ is the identity matrix. $\mathbf{H}(s)$ describes the relationship between a *small change* in all possible inputs, i.e. the array of Laplace transformed rate constants $\Delta \mathbf{k}(s) = [\Delta k_1(s), \dots, \Delta k_N(s)]^T$, and the resulting *changes* in all possible outputs, i.e. $\Delta \mathbf{y}(s) = [\Delta y_1(s), \dots, \Delta y_P(s)]^T$.

2.2 Calculating Control Coefficients

In metabolic control analysis (26–29) sensitivities are generally calculated as dimension-independent control or sensitivity coefficients:

$$C_{k_n}^{y_p} = \frac{\partial \log y_p}{\partial \log k_n} \quad (12)$$

These sensitivity coefficients can also be calculated in s (frequency) space. The relationship between the s -dependent transfer function matrix and the frequency dependent control coefficient matrix is found to be (30)

$$\mathbf{C}_{\mathbf{k}}^{\mathbf{y}}(s) = \mathbf{H}(s) \bullet \frac{\mathbf{k}_{ss}}{\mathbf{y}_{ss}} \quad (13)$$

where the steady state control coefficient matrix becomes

$$\mathbf{C}_{\mathbf{k}}^{\mathbf{y}} = \mathbf{H}(0) \bullet \frac{\mathbf{k}_{ss}}{\mathbf{y}_{ss}} \quad (14)$$

by using element-wise multiplication, or the so-called Hadamard matrix multiplication ‘ \bullet ’ (31). An alternative approach to relate control/sensitivity coefficients to their s -dependent counterpart was described by Ingalls (32).

2.3 Illustrating the Principles

We will use motif M1 shown below to illustrate the MATLAB commands used to calculate the transfer functions and control coefficients.

$$[\textit{place picture of motif M1 here}] \quad (\text{M1})$$

The rate equations as in Eq. 3 become

$$\frac{dI_1(t)}{dt} = \dot{I}_1(t) = k_1(t) - k_2(t)I_1(t) + k_{-2}(t)I_2(t) \quad (15)$$

$$\frac{dI_2(t)}{dt} = \dot{I}_2(t) = k_2(t)I_1(t) - k_3(t)I_2(t) - k_{-2}(t)I_2(t) \quad (16)$$

Since concentration is the model output, we get $y_1(t) = I_1(t)$ and $y_2(t) = I_2(t)$ as described in Eq. 4. Defining the order of the reaction constants as $\Delta \mathbf{k}(t) = [\Delta k_1(t), \Delta k_2(t), \Delta k_{-2}(t), \Delta k_3(t)]^T$ gives the following MATLAB code for the implementation of Eqs. 3, 4, 7-10:

```
%file I122.m
clear all
close all

syms k1 k2 km2 k3 I1 I2

% differential equations
d_I1 = k1 - k2*I1 + km2*I2;
d_I2 = k2*I1 - km2*I2 - k3*I2;

% output equations
y1 = I1; y2 = I2;

% system matrix A
A11=diff(d_I1, I1); A12=diff(d_I1, I2);
A21=diff(d_I2, I1); A22=diff(d_I2, I2);
A=[A11 A12; A21 A22];

% input matrix B
B11=diff(d_I1, k1); B12=diff(d_I1, k2);
B13=diff(d_I1, km2); B14=diff(d_I1, k3);
B21=diff(d_I2, k1); B22=diff(d_I2, k2);
B23=diff(d_I2, km2); B24=diff(d_I2, k3);
B=[B11 B12 B13 B14; B21 B22 B23 B24];

C11 = diff(y1, I1); C12 = diff(y1, I2);
C21 = diff(y2, I1); C22 = diff(y2, I2);
C=[C11 C12; C21 C22];

D11=diff(y1, k1); D12=diff(y1, k2);
D13=diff(y1, km2); D14=diff(y1, k3);
```

```
D21=diff(y2,k1);D22=diff(y2,k2);
D23=diff(y2,km2);D24=diff(y2,k3);
D=[D11 D12 D13 D14;D21 D22 D23 D24];
```

which produces the following matrices for the linearized state space model:

$$A = \begin{bmatrix} -k_2 & k_{-2} \\ k_2 & -(k_3 + k_{-2}) \end{bmatrix}, \quad B = \begin{bmatrix} 1 & -I_1 & I_2 & 0 \\ 0 & I_1 & -I_2 & -I_2 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The MATLAB code for the transfer function matrix $\mathbf{H}(s) = \frac{\Delta \mathbf{y}(s)}{\Delta \mathbf{k}(s)}$ using Eq. 11 is shown below:

```
% identity matrix sI
syms s
sI = eye(2)*s;

% calculating the transfer function
Hs = C*inv(sI-A)*B+D;
```

The results can be presented in MATLAB workspace by typing `pretty(simple(Hs))` or more readable as

$$\mathbf{H}(s) = \frac{1}{s^2 + s(k_2 + k_3 + k_{-2}) + k_3 k_2} \cdots$$

$$\begin{bmatrix} s + k_3 + k_{-2} & -I_1(s + k_3) & I_2(s + k_3) & -I_2 k_{-2} \\ k_2 & I_1 s & I_2 s & -I_2(s + k_2) \end{bmatrix} \quad (17)$$

We see that the transfer function depends on the steady state value of I_1 and I_2 . Since these values depends on the rate constants, we first calculate and thereafter insert these expressions into the transfer function making the transfer function only dependent upon rate constants. The MATLAB commands used here are `solve` and `subs`

```
% state space calculations
ss = solve(d_I1 , I1 , d_I2 , I2 );
I1 = ss . I1 ;
I2 = ss . I2 ;
Hs1=subs(Hs,{ ' I1 ' , ' I2 ' },{ I1 , I2 })
```

where `ss` is a struct with two elements, i.e. the steady state expressions for I_1 and I_2 :

$$I_1 = \frac{k_{-2} k_1 + k_1 k_3}{k_3 k_2}, \quad I_2 = \frac{k_1}{k_3} \quad (18)$$

the steady state concentration control matrices using the following MATLAB code

```
k_y = [k1/I1 k2/I1 km2/I1 k3/I1
        k1/I2 k2/I2 km2/I2 k3/I2];
C = Hs.*k_y; % elementwise multiplication
C1 = subs(C,{'I1','I2'},{I1,I2})
Css = simplify(subs(C1,'s',0))
```

The results are presented below

$$\begin{aligned} \mathbf{C}_k^y(s) &= \mathbf{H}(s) \bullet \frac{\mathbf{k}_{ss}}{\mathbf{y}_{ss}} \\ &= \frac{1}{s^2 + s(k_2 + k_3 + k_{-2}) + k_3k_2} \cdot \\ &\quad \begin{bmatrix} \frac{(s+k_3+k_{-2})k_2k_3}{k_3+k_{-2}} & -(s+k_3)k_2 & \frac{(s+k_3)k_{-2}k_2}{k_3+k_{-2}} & \frac{-k_{-2}k_2k_3}{k_3+k_{-2}} \\ k_2k_3 & (k_3+k_{-2})s & -k_{-2}s & -(s+k_2)k_3 \end{bmatrix} \end{aligned} \quad (19)$$

and the steady state concentration control coefficient matrix as

$$\mathbf{C}_k^y = \begin{bmatrix} 1 & -1 & \frac{k_{-2}}{k_{-2}+k_3} & -\frac{k_{-2}}{k_{-2}+k_3} \\ 1 & 0 & 0 & -1 \end{bmatrix} \quad (20)$$

Applying the numerical values to the rate constants

```
C2 = subs(C1,{'k1','k2','km2','k3'},{1,2,3,4})
C3 = eval(simplify(C2)) % freq. dependent CC
C4 = dcgain(C3); % steady state CC
```

producing the following results in MATLAB

```
>> C4=dcgain(C3)
C4 =
    1.0000   -1.0000    0.4286   -0.4286
    1.0000    0         0        -1.0000
```

In the same manner as for the transfer function matrix, step responses, Bode plots, and pole/zero plots can now be found for $\mathbf{C}_k^y(s)$ in Eq.19.

The summation theorem applied to either of the concentration control coefficient matrices (i.e. the frequency dependent matrix in Eq. 19 or the steady

state matrix in Eq. 20) gives (summed over all N reactions):

$$\sum_{\text{all } N} \mathbf{C}_{\mathbf{k}}^{\mathbf{y}(s)} = \sum_{\text{all } N} \mathbf{C}_{\mathbf{k}}^{\mathbf{y}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

which is easily verified by summing the rows in the MATLAB results for `C4` shown above.

2.5 Structures as a Tool for Generalizing the Code

The basic data type in MATLAB is the numerical matrix; in fact, the name MATLAB stands for MATrix LABoratory. Numerical matrices are very useful for computations and linear algebra, but they are not the only data types MATLAB offers.

In our work, we have chosen to employ two additional data types to make the code easier to read, program, and manage: *Structures* (structs) and *cell arrays*. Whereas the basic matrices only allow numerical data, both the structs and the cell arrays allow us to group together data of different types such as scalars, matrices, symbols, strings, and indeed structs and cell arrays.

The structs are variables that have named *fields*, making it possible to create hierarchies of named variables, matrices, etc. We have used these structs to easier model our chemical networks, and to keep the symbolic and numerical data sets separate. The struct that represents the networks symbolically has been named `s` (for “symbolic”), while the struct containing the corresponding numerical values has been named `v` (for “values”). Having all the network data contained in two separate structures makes it trivial to save and load new sets of variables, and keep track of multiple data sets simultaneously.

As an example, the rate constants k_n , and indeed all other numbered variables, have been stored in the structs as cell arrays. The symbolic rate constants are accessed by using `s.k{n}`, and the corresponding values by using `v.k{n}`. Using cell arrays for such variables means we can quickly, not to mention independently of the current particular chemical network, check how many rate constants the network has. This allows us to write code that is more dynamic and which does not need to be tailor made for each network

to be evaluated, i.e. we can easily add more inputs or outputs and use the same framework.

In our work, we use the network models to study one or more of the following input/output relationships in our search for robust/non-robust perfect adaptation:

- rate constants/concentration
- rate constants/fluxes
- temperature/concentration
- temperature/fluxes

We specify the equations of the network in the `s` struct and the values for which we want to evaluate the network in the `v` struct.

For the example in motif M1, i.e. rate constant/concentration relationship, the network specification together with the differential equations will look like

```
v.intermediates = 2;
v.rate_constants = 4;
v.inputs = v.rate_constants;
v.outputs = v.intermediates;

v.k = {1, 2, 3, 4};

% differential equations
s.d_I{1} = s.k{1} - s.k{2}*s.I{1};
s.d_I{2} = s.k{2}*s.I{1} - s.k{3}*s.I{2};
```

The generic code (used for all motifs) for calculation of e.g. the system matrix A and the output matrix C in the `s` struct becomes

```
% system matrix A
for kk = 1:v.intermediates
    for jj = 1:v.intermediates
        s.d_I{kk}.dI{jj} = diff(s.d_I{kk}, s.I{jj});
        s.A(kk, jj) = s.d_I{kk}.dI{jj};
    end
end

% output matrix C
for kk = 1:v.outputs
    for jj = 1:v.intermediates
```

```

        s.dy{kk}.dI{jj} = diff(s.y{kk},s.I{jj});
        s.C(kk, jj) = s.dy{kk}.dI{jj};
    end
end

```

By using the structs and cell arrays, the specification for each motif needs approximately 20 individual lines of code, whereas the generic calculation for transfer function, control coefficients, search for non-robust perfect adaptation and others are programmed over approximately 1000 lines of code.

3 Defining the Set-Point in a Homeostatic Controller

Homeostasis is another aspect of how to view (robust) perfect adaptation of a controlled compound A . To see how the set-point in the integral feedback scheme (Fig. 1.2) can be defined in kinetic terms, we consider in Fig. 3.1 an homeostatic inflow controller (25), where species A is under negative stabilizing (33) feedback control by species E_{adapt} . We assume that A is synthesized by zero-order process with rate constant k_{synth} and is subject to unpredictable inflow perturbations by (varying) rate constant k_{pert} . Enzyme E_{tr} transforms A into another species, while enzyme E_{adapt} induced by A (through k_{adapt}) removes/degrades A . Enzyme E_{set} removes or inactivates E_{adapt} . Concentrations of E_{tr} and E_{set} are considered to be constant. All enzymatic reactions are described by standard Michaelis-Menten kinetics

$$v = \frac{V_{max}^E \cdot S}{K_M^E + S} \quad (21)$$

where v is the reaction velocity, S denotes the concentration of substrate, K_M^E is the Michaelis constant, and V_{max}^E is the maximum velocity described by $V_{max}^E = k_{cat}^E \cdot E$ with turnover number k_{cat}^E and enzyme concentration E .

[Fig. 3.1 approximately here]

The rate equations are:

$$\frac{dA}{dt} = k_{pert} + k_{synth} - \frac{V_{max}^{E_{adapt}} A}{K_M^{E_{adapt}} + A} - \frac{V_{max}^{E_{tr}} A}{K_M^{E_{tr}} + A} \quad (22)$$

$$\frac{dE_{adapt}}{dt} = k_{adapt}A - \frac{V_{max}^{E_{set}} E_{adapt}}{K_M^{E_{set}} + E_{adapt}} \quad (23)$$

Eq. 23 defines the error between the set-point in A -homeostasis, A_{set} , and the actual value in A by comparing Eq. 23 with the equation

$$\frac{dE_{adapt}}{dt} = k_{adapt}(A - A_{set}) \quad (24)$$

which gives the following A_{set}

$$A_{set} = \frac{V_{max}^{E_{set}}}{k_{adapt}} \cdot \frac{E_{adapt}}{K_M^{E_{set}} + E_{adapt}} \quad (25)$$

Eq. 25 indicates that $V_{max}^{E_{set}}/k_{adapt}$ is an upper bound for A_{set} and robust homeostasis in A with the set-point

$$A_{set} = \frac{V_{max}^{E_{set}}}{k_{adapt}} \quad (26)$$

is achieved when $K_M^{E_{set}} \ll E_{adapt}$, i.e. when there is a strong binding between E_{adapt} and its processing enzyme E_{set} leading to zero-order kinetics in the removal/inactivation of E_{adapt} (25).

To solve the rate equations (Eqs. 22 and 23) two m-files are created and put in the path of MATLAB. The first file `LShifc.m` contains initial concentrations to the dynamical variables $y(i)$, values to the rate parameters $k(i)$, the method of integration, and the simulation time. The file can also include plotting instructions as shown here:

```
%file: LShifc.m
clear all

% dynamic variables
% y(1) <-> A
% y(2) <-> E_adapt

%rate constants/rate parameters
% k(1) <-> k1
% k(2) <-> kcat(E_adapt)
% k(3) <-> KM(E_adapt)
% k(4) <-> k_adapt
% k(5) <-> kcat(E_set)
% k(6) <-> KM(E_set)
```

```

% k(7) <-> k_synth
% k(8) <-> kcat(E_tr)
% k(9) <-> KM(E_tr)
% k(10) <-> E_set
% k(11) <-> E_tr

% define rate constant values [k(1) k(2) ..... k(10)]
ks=[0.1 1.0 2.0 3.0 6.0e+6 1.0e-6 1.0 0.01 5.0 5.0e-7 0.1];

% simulation time
t=[0,50];

% initial concentrations
y0=[1.0 0.03];

% options for numerical integration
options = odeset('RelTol',0.000001,'MaxStep',0.01);
% solve model
[T Y]=ode15s(@hifc,t,y0,options,ks);

% making Figure 1
figure(1),
subplot(2,1,1),plot(T,Y(:,2),'k-',T,Y(:,1),'k-');
xlabel('time, \uau');
ylabel('concentration, \uau');
hold on
grid on
legend('E_{adapt}','A');
hold off

subplot(2,1,2),plot(Y(:,1),Y(:,2),'k-');
xlabel('A-concentration, \uau');
ylabel('E_{adapt}-concentration, \uau');
title(['inflow \uadhomeostatic \uadcontroller']);
hold on
legend('E_{adapt}-A-phase \uadplane');
hold on
grid on
hold off
...

```

The second file `hifc.m` defines symbolically the rate equations:

```

%file: hifc.m
function dy=hifc(t,y,k)
dy=zeros(2,1);
dy(1)=k(1)-k(2)*y(1)*y(2)/(k(3)+y(1))+k(7)-k(8)*k(11)*y(1)/(k(9)+y(1));
dy(2)=k(4)*y(1)-k(5)*y(2)*k(10)/(k(6)+y(2));

```

The model is run by placing the files `LShifc.m` and `hifc.m` somewhere in MATLAB's path, typing `LShifc` in the MATLAB console, and hitting the RETURN key. Fig. 3.2 shows the adaptation behavior of the inflow controller with the initial concentrations and rate constants given above.

[Fig. 3.2 approximately here]

3.1 Harmonic Oscillations in Homeostatic Controllers

Interestingly, the negative feedback in the A - E_{adapt} homeostatic system can lead to harmonic oscillations when the binding between A and E_{adapt} becomes strong (leading to low $K_M^{E_{adapt}}$ values) and, additionally, the removal of A by transforming enzyme E_{tr} is negligible (either by a large $K_M^{E_{tr}}$ value and/or by a low $V_{max}^{E_{tr}}$ value). In this case the rate equations (Eqs. 22 and 23) can be combined lead to the harmonic oscillator equation

$$\frac{\ddot{A}}{k_{cat}^{E_{adapt}} \cdot k_{adapt}} + A = A_{set} = \frac{V_{max}^{E_{set}}}{k_{adapt}} \quad (27)$$

indicating that A shows harmonic oscillations around A_{set} with a period length P given by

$$P = \frac{2\pi}{\sqrt{k_{cat}^{E_{adapt}} \cdot k_{adapt}}} \quad (28)$$

To observe these oscillations we make a slight change in the $K_M^{E_{adapt}}$ ($\mathbf{k}(3)$) value from 2.0 to 1.0e-6 by appending the following code in `LShifc.m`:

```
%file: LShifc.m
...
% repeat calculations, but now with low k(3)(KM (E_adapt)) value...
% define rate constant values [k(1) k(2) ..... k(10)]
ks=[0.1 1.0 1.0e-6 3.0 6.0e+6 1.0e-6 1.0 0.01 5.0 5.0e-7 0.1];

% simulation time
t=[0,50];

% initial concentrations
y0=[1.0 0.03];

% options for numerical integration
options = odeset('RelTol',0.000001,'MaxStep',0.01);
% solve model
[T Y]=ode15s(@hifc,t,y0,options,ks);

% making Figure 2
figure(2),
subplot(2,1,1),plot(T,Y(:,2),'k—',T,Y(:,1),'k—');
xlabel('time, \u03bcs');
ylabel('concentration, \u03bcM');
hold on
```

```

grid on
legend('E_{adapt}', 'A');
hold off

subplot(2,1,2), plot(Y(:,1), Y(:,2), 'k-');
xlabel('A-concentration, \uau');
ylabel('E_{adapt}-concentration, \uau');
title(['inflow \uhomeostatic \ucontroller'])
hold on
legend('E_{adapt}-A-phase \uplane');
hold on
grid on
hold off

```

In addition to Fig. 3.2, the following figure is generated showing harmonic oscillations in A and E_{adapt} .

[Fig. 3.3 approximately here]

These type of oscillations have been considered to occur in the negative feedback regulation of the p53-Mdm2 system (34, 35), where p53 is considered to be bound by Mdm2 to an upper (sub-apoptotic) level (36). Another interesting aspect of oscillations arising in homeostatic controllers may be related to the pulsatile manner how hormones are released leading to homeostatic control of important metabolites (37).

Supplementary Information

MATLAB files I122.m, LShifc.m, and hifc.m described in the text can be downloaded from <http://bioinfo.uu.no/adapt.zip>

References

- [1] Grylls, F. S., and J. S. Harrison, 1956. Adaptation of yeast to maltose fermentation. *Nature* 178:1471–2.
- [2] Berg, H. C., and P. M. Tedesco, 1975. Transient response to chemotactic stimuli in *Escherichia coli*. *Proc Natl Acad Sci U S A* 72:3235–9.
- [3] Alon, U., M. G. Surette, N. Barkai, and S. Leibler, 1999. Robustness in bacterial chemotaxis. *Nature* 397:168–71.

- [4] Bray, D., 2002. Bacterial chemotaxis and the question of gain. *Proc Natl Acad Sci U S A* 99:7–9.
- [5] Mello, B. A., and Y. Tu, 2003. Perfect and near-perfect adaptation in a model of bacterial chemotaxis. *Biophys J* 84:2943–56.
- [6] Berg, H. C., 2004. *E. coli* in Motion. Springer-Verlag, New York.
- [7] Mello, B. A., and Y. Tu, 2007. Effects of adaptation in maintaining high sensitivity over a wide range of backgrounds for *Escherichia coli* chemotaxis. *Biophys J* 92:2329–37.
- [8] Hansen, C. H., R. G. Endres, and N. S. Wingreen, 2008. Chemotaxis in *Escherichia coli*: A Molecular Model for Robust Precise Adaptation. *PLoS Computational Biology* 4:0014–0027.
- [9] Levchenko, A., and P. A. Iglesias, 2002. Models of eukaryotic gradient sensing: application to chemotaxis of amoebae and neutrophils. *Biophys J* 82:50–63.
- [10] Ratliff, F., H. K. Hartline, and W. H. Miller, 1963. Spatial and temporal aspects of retinal inhibitory interaction. *J Opt Soc Am* 53:110–20.
- [11] He, Q., and Y. Liu, 2005. Molecular mechanism of light responses in *Neurospora*: from light-induced transcription to photoadaptation. *Genes Dev* 19:2888–99.
- [12] Walters, R. G., 2005. Towards an understanding of photosynthetic acclimation. *J Exp Bot* 56:435–47.
- [13] Arthur, H., and K. Watson, 1976. Thermal adaptation in yeast: growth temperatures, membrane lipid, and cytochrome composition of psychrophilic, mesophilic, and thermophilic yeasts. *J Bacteriol* 128:56–68.
- [14] Margesin, R., 2009. Effect of temperature on growth parameters of psychrophilic bacteria and yeasts. *Extremophiles* 13:257–62.
- [15] Asthagiri, A. R., and D. A. Lauffenburger, 2000. Bioengineering models of cell signaling. *Annu Rev Biomed Eng* 2:31–53.
- [16] Koshland, J., D. E., A. Goldbeter, and J. B. Stock, 1982. Amplification and adaptation in regulatory and sensory systems. *Science* 217:220–5.

- [17] Muzzey, D., C. A. Gomez-Uribe, J. T. Mettetal, and A. van Oudenaarden, 2009. A systems-level analysis of perfect adaptation in yeast osmoregulation. *Cell* 138:160–71.
- [18] Asthagiri, A. R., C. M. Nelson, A. F. Horwitz, and D. A. Lauffenburger, 1999. Quantitative relationship among integrin-ligand binding, adhesion, and signaling via focal adhesion kinase and extracellular signal-regulated kinase 2. *J Biol Chem* 274:27119–27.
- [19] Hao, N., M. Behar, T. C. Elston, and H. G. Dohlman, 2007. Systems biology analysis of G protein and MAP kinase signaling in yeast. *Oncogene* 26:3254–66.
- [20] Mettetal, J. T., D. Muzzey, C. Gómez-Uribe, and A. van Oudenaarden, 2008. The Frequency Dependence of Osmo-Adaptation in *Saccharomyces cerevisiae*. *Science* 319:482–4.
- [21] Hong, C. I., E. D. Conrad, and J. J. Tyson, 2007. A proposal for robust temperature compensation of circadian rhythms. *Proc Natl Acad Sci U S A* 104:1195–200.
- [22] Yi, T. M., Y. Huang, M. I. Simon, and J. Doyle, 2000. Robust perfect adaptation in bacterial chemotaxis through integral feedback control. *Proc Natl Acad Sci U S A* 97:4649–53.
- [23] Wilkie, J., M. Johnson, and K. Reza, 2002. Control Engineering. An Introductory Course. Palgrave, New York.
- [24] Drengstig, T., H. R. Ueda, and P. Ruoff, 2008. Predicting Perfect Adaptation Motifs in Reaction Kinetic Networks. *J Phys Chem B* 112:16752–16758.
- [25] Ni, X. Y., T. Drengstig, and P. Ruoff, 2009. The control of the controller: molecular mechanisms for robust perfect adaptation and temperature compensation. *Biophys J* 97:1244–53.
- [26] Kacser, H., and J. A. Burns, 1979. Molecular democracy: who shares the controls? *Biochem Soc Trans* 7:1149–60.
- [27] Burns, J. A., A. Cornish-Bowden, A. K. Groen, R. Heinrich, H. Kacser, J. W. Porteous, S. M. Rapoport, T. A. Rapoport, J. W. Stucki, J. M.

- Tager, R. J. A. Wanders, and H. V. Westerhoff, 1985. Control analysis of metabolic systems. *Trends Biochem Sci* 19:16.
- [28] Heinrich, R., and S. Schuster, 1996. The Regulation of Cellular Systems. Chapman and Hall, New York.
- [29] Fell, D., 1997. Understanding the Control of Metabolism. Portland Press, London and Miami.
- [30] Drengstig, T., T. Kjosmoen, and P. Ruoff, 2009. On the Relationships between Sensitivity Coefficients and Transfer Functions in Reaction Kinetic Networks. *System Control Letters*, submitted .
- [31] Lutkepohl, H., 1996. Handbook of matrices. John Wiley & Sons.
- [32] Ingalls, B. P., 2004. A Frequency Domain Approach to Sensitivity Analysis of Biochemical Networks. *J. Phys. Chem. B* 108:1143–1152.
- [33] Beckskei, A., and L. Serrano, 2000. Engineering stability in gene networks by autoregulation. *Nature* 405:261–74.
- [34] Lahav, G., N. Rosenfeld, A. Sigal, N. Geva-Zatorsky, A. J. Levine, M. B. Elowitz, and U. Alon, 2004. Dynamics of the p53-Mdm2 feedback loop in individual cells. *Nat Genet* 36:147–50.
- [35] Geva-Zatorsky, N., N. Rosenfeld, S. Itzkovitz, R. Milo, A. Sigal, E. Dekel, T. Yarnitzky, Y. Liron, P. Polak, G. Lahav, and U. Alon, 2006. Oscillations and variability in the p53 system. *Mol Syst Biol* 2:2006 0033.
- [36] Jolma, I. W., X. Y. Ni, L. Rensing, and P. Ruoff. Harmonic Oscillations in Homeostatic Controllers: Dynamics of the p53 Regulatory System. *Biophys J* in press.
- [37] Chadwick, D. J., and J. A. Goode, 2000. Mechanisms and Biological Significance of Pulsatile Hormone Secretion. Wiley, New York.

Figure Legends

Fig. 1.1. Different adaptation behaviors of a system with respect to an applied step perturbation.

Fig. 1.2. Scheme of integral feedback/control of a perturbed system where the system output perfectly adapts to the setpoint. MV and CV are the manipulated and controlled variables, respectively. e denotes the error between the set-point and controlled variable. Gray symbols represent the notation by Yi et al. (22).

Fig. 2.1. The different adaptation behaviors in Fig. 1.1 are described by the solution of $n(s) = 0$. (1): when $n(s)$ is a constant, no adaptation exist; (2) partial adaptation is observed when the solution of $n(s) = 0$ lies in the left half of the complex s -plane; (3) the system shows perfect adaptation when the solution of $n(s) = 0$ lies in the origo; (4) overadaptation is observed when the solution of $n(s) = 0$ lies in the right half of the complex s -plane. The four transfer functions are described by $H_i(s) = \frac{n_i(s)}{d(s)}$, $i = 1, \dots, 4$ with the denominator $d(s) = (0.2s + 1)(0.35s + 1)(0.45s + 1)$ and numerators: $n_1(s) = 1.5$, $n_2(s) = (s + 1)$, $n_3(s) = s$, $n_4(s) = s - 1$.

Fig. 2.2. MATLAB generated plot showing the result of a step response on rate constants k_1 (In(1)), k_2 (In(2)), k_{-2} (In(3)), and k_3 (In(4)) with the resulting time-behavior of concentrations (amplitude) of I_1 (Out(1)) and I_2 (Out(2)). Please note that by default MATLAB implicitly assumes that the time scale by which rate constants are defined is given in seconds (sec).

Fig. 2.3. MATLAB generated Bode plot showing the magnitude of the outputs (in dB) and the output's phases (in degrees) as a function of the logarithm of the frequency (radians/second).

Fig. 2.4. MATLAB generated plot showing the zeros ($n(s) = 0$) as circles and poles ($d(s) = 0$) as crosses for each element in the transfer function matrix $\mathbf{H}(s)$.

Fig. 3.1. Homeostatic inflow controller keeping robust homeostasis in A .

Fig. 3.2. MATLAB generated plot showing (robust) adaptation in A .

Fig. 3.3. Harmonic oscillations generated in the homeostatic inflow controller. Due to the harmonic character of the oscillations no limit-cycle is observed but multiple trajectories in phase space occur (only one is shown) which depend on the initial concentrations.

Figure Images

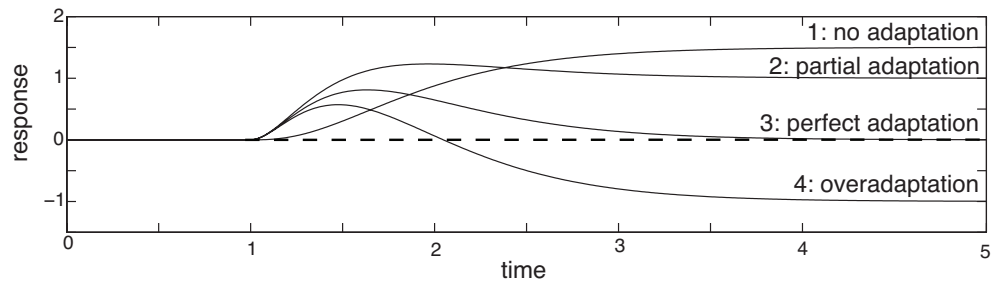


Fig. 1.1

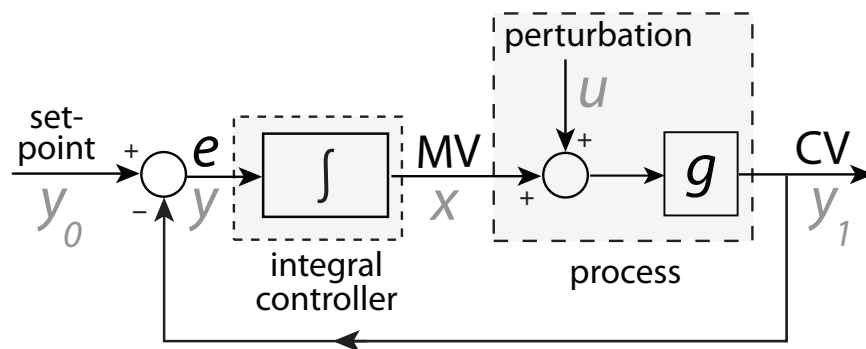


Fig. 1.2

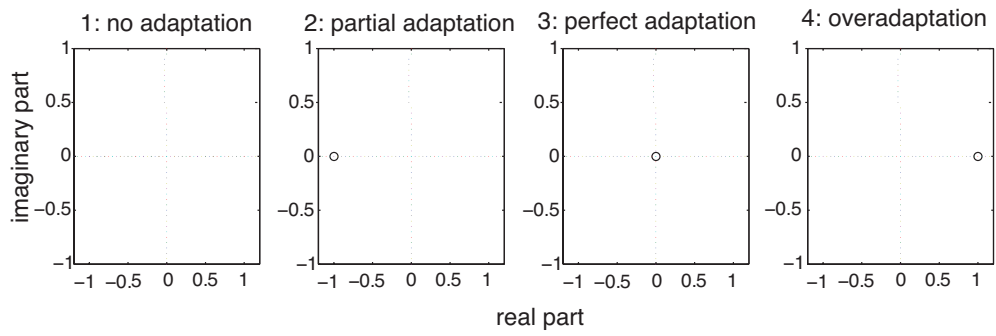


Fig. 2.1

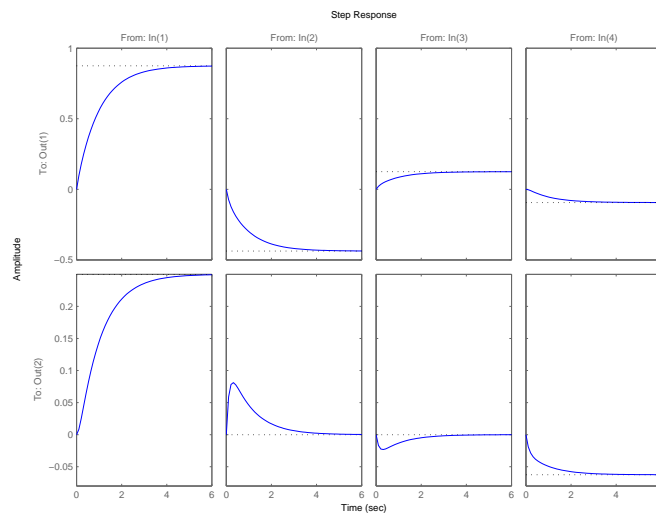


Fig. 2.2

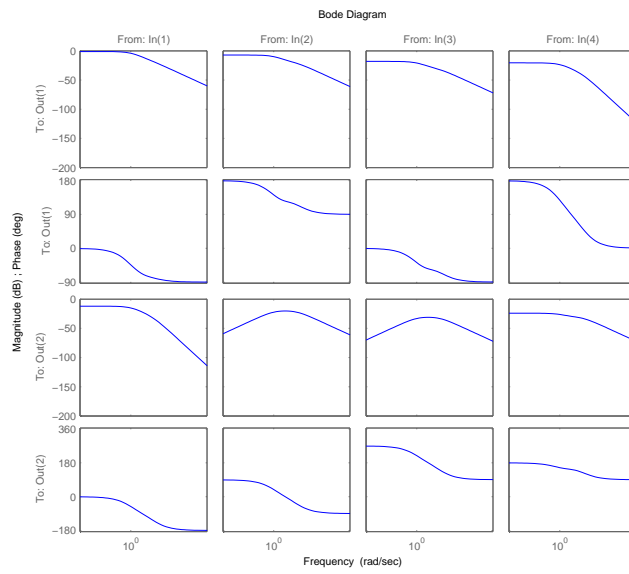


Fig. 2.3

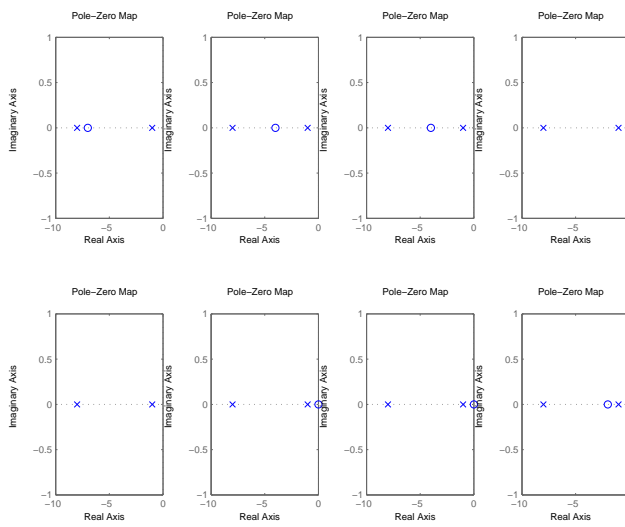


Fig. 2.4

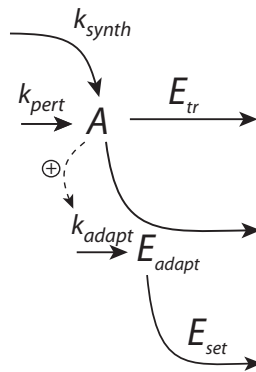


Fig. 3.1

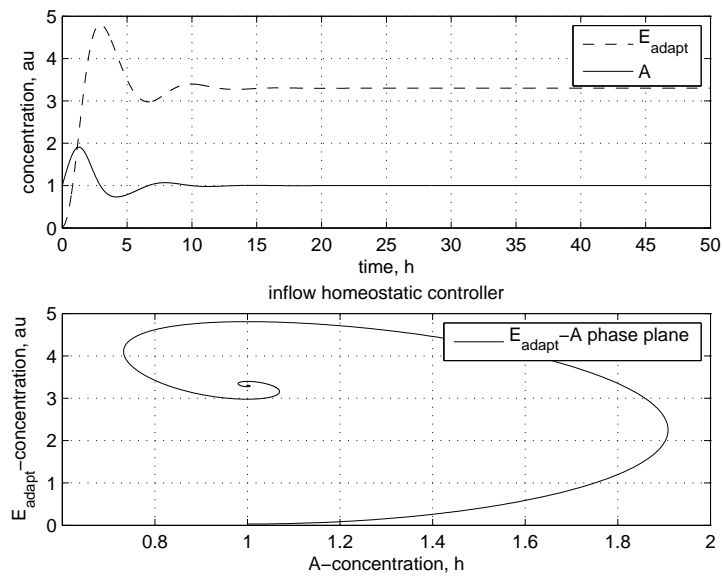


Fig. 3.2

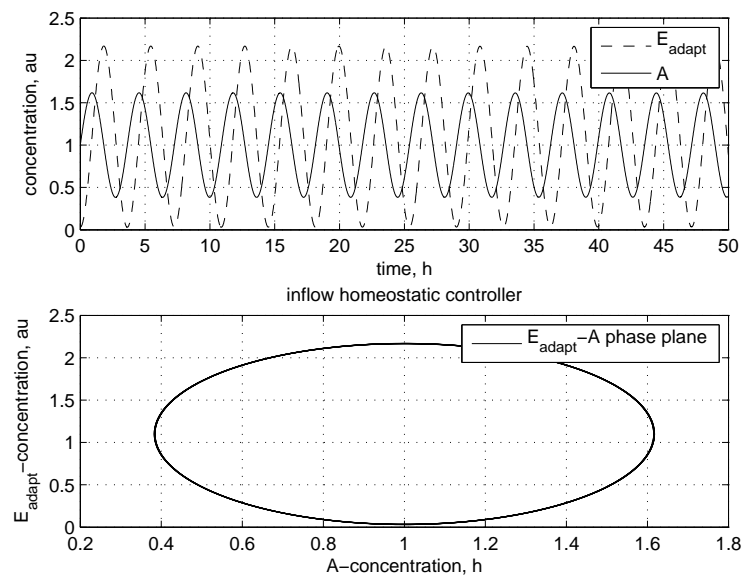
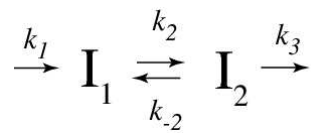


Fig. 3.3



Motif M1