International school
Linear System Theory, Control and Matrix Computations
Exercises for Lectures 3, 5, and 9

Ivan Markovsky

Problem 1 *Distance measure modulo translation, rotation, and scaling*

Consider two contours $\mathscr{C}_1$ and $\mathscr{C}_2$ in $\mathbb{R}^2$, specified by N matching points

$$p^{(i)} \in \mathscr{C}_1 \subset \mathbb{R}^2 \quad \leftrightarrow \quad q^{(i)} \in \mathscr{C}_2 \subset \mathbb{R}^2, \qquad i = 1, \ldots, N.$$

An example with $N = 6$ points is shown in Figure 1.

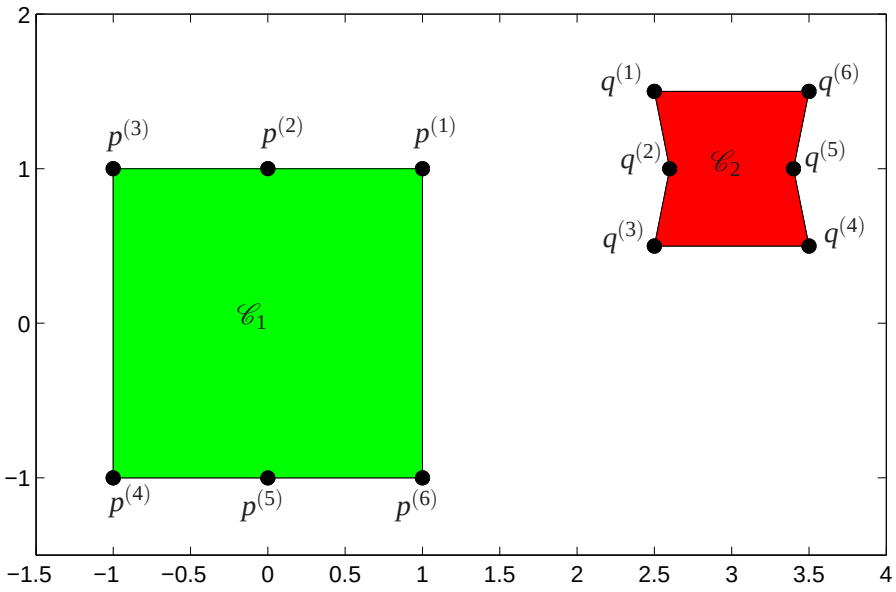

Figure 1: Contours $\mathscr{C}_1$ and $\mathscr{C}_2$ with 6 matching points $p^{(i)} \leftrightarrow q^{(i)}$.

Next, we define a distance measure $\text{dist}(\mathscr{C}_1, \mathscr{C}_2) \in [0, \infty)$ between $\mathscr{C}_1$ and $\mathscr{C}_2$ modulo translation, rotation, and scaling of the contours. Denote by $\mathscr{A}_{a,\theta,s}$ the operator that translates by the vector $a \in \mathbb{R}^2$, rotates by the angle $\theta \in [-\pi, \pi)$ rad, and scales by the factor $s > 0$, i.e.,

$$\mathscr{A}_{a,\theta,s}(p) = s \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} p + a.$$

A possible formulation of $\text{dist}(\mathscr{C}_1, \mathscr{C}_2)$ that gives a tractable problem is

$$\text{dist}(\mathscr{C}_1, \mathscr{C}_2) := \min_{\substack{a \in \mathbb{R}^2,\ s>0 \\ \theta \in [-\pi, \pi)}} \sqrt{\sum_{i=1}^{N} \left\| p^{(i)} - \mathscr{A}_{a,\theta,s}(q^{(i)}) \right\|_2^2}. \tag{1}$$

1. Prove that problem (1) is equivalent to the following least squares problem

$$\text{minimize}_{(a_1,a_2,b_1,b_2)\in\mathbb{R}^4} \quad \left\| \begin{bmatrix} p_1^{(1)} \\ p_2^{(1)} \\ \vdots \\ p_1^{(N)} \\ p_2^{(N)} \end{bmatrix} - \begin{bmatrix} 1 & 0 & q_1^{(1)} & -q_2^{(1)} \\ 0 & 1 & q_2^{(1)} & q_1^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & q_1^{(N)} & -q_2^{(N)} \\ 0 & 1 & q_2^{(N)} & q_1^{(N)} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} \right\|_2, \tag{2}$$

where the relation between the parameters (b_1, b_2) of (1) and the parameters of (θ, s) of (2) is given by

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = s \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} \theta \\ s \end{bmatrix} = \begin{bmatrix} \sin^{-1}(b_2/\sqrt{b_1^2+b_2^2}) \\ \sqrt{b_1^2+b_2^2} \end{bmatrix}. \tag{3}$$

2. Compute $d_{12} := \text{dist}(\mathscr{C}_1, \mathscr{C}_2)$ and $d_{21} := \text{dist}(\mathscr{C}_2, \mathscr{C}_1)$ and the corresponding transformation parameters a, θ, and s, where

$$\begin{bmatrix} p^{(1)} & p^{(2)} & p^{(3)} & p^{(4)} & p^{(5)} & p^{(6)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & -1 & 0 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 \end{bmatrix},$$

$$\begin{bmatrix} q^{(1)} & q^{(2)} & q^{(3)} & q^{(4)} & q^{(5)} & q^{(6)} \end{bmatrix} = \begin{bmatrix} 2.5 & 2.6 & 2.5 & 3.5 & 3.4 & 3.5 \\ 1.5 & 1.0 & 0.5 & 0.5 & 1.0 & 1.5 \end{bmatrix}.$$

Comment on the results.

Solution:

1. Consider a point $q \in \mathbb{R}^2$ and define the rotated point

$$q_{\mathrm{r}} := \begin{bmatrix} \cos\theta_{\mathrm{r}} & -\sin\theta_{\mathrm{r}} \\ \sin\theta_{\mathrm{r}} & \cos\theta_{\mathrm{r}} \end{bmatrix} q,$$

The key observation is that for $\theta_{\mathrm{r}} \in (-\pi, \pi)/0$,

$$\mathscr{A}_{a,\theta,s}(q) = \begin{bmatrix} q & q_{\mathrm{r}} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} + a,$$

where (b_1, b_2) and (θ, s) are in a one-to-one relation. The relation between the parameters is derived by solving the equation

$$\begin{bmatrix} q & q_{\mathrm{r}} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = s \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} q \tag{4}$$

for b_1 and b_2, given s and θ, and for θ and s, given b_1 and b_2. In the case of $\theta_{\mathrm{r}} = \pi/2$, the solution of (4), i.e., the relation between the original and transformed parameters is (3).

In terms of the parameters a_1, a_2, b_1, and b_2, problem (1) becomes the ordinary least squares problem

$$\text{minimize}_{(a_1,a_2,b_1,b_2)\in\mathbb{R}^4} \left\| \begin{bmatrix} p_1^{(1)} \\ p_2^{(1)} \\ \vdots \\ p_1^{(N)} \\ p_2^{(N)} \end{bmatrix} - \begin{bmatrix} 1 & 0 & q_1^{(1)} & q_{\mathrm{r}}^{(1)} \\ 0 & 1 & q_2^{(1)} & q_{\mathrm{r}}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & q_1^{(N)} & q_{\mathrm{r}}^{(N)} \\ 0 & 1 & q_2^{(N)} & q_{\mathrm{r}}^{(N)} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} \right\|_2 . \tag{5}$$

In the case of $\theta_{\mathrm{r}} = \pi/2$, (5) simplifies to (2).

2. For the contours in Figure 1, we have

$$d_{12} = 0.2626, \qquad \text{with} \quad a_{12} = \begin{bmatrix} -2.0690 \\ 6.2069 \end{bmatrix}, \quad s_{12} = 2.0690, \quad \theta_{12} = -1.5708\,\mathrm{rad}$$

and

$$d_{21} = 0.1265, \qquad \text{with} \quad a_{21} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \quad s_{21} = 0.48, \quad \theta_{21} = 1.5708\,\mathrm{rad}.$$

The contours $\mathscr{C}_1$, $\mathscr{A}_{a_{12},\theta_{12},s_{12}}(\mathscr{C}_2)$, $\mathscr{A}_{a_{21},\theta_{21},s_{21}}(\mathscr{C}_1)$, and $\mathscr{C}_2$ are shown in Figure 2. (The numerical results and the plots are generated by the functions `dist` and `test_dist`.) The example shows that, in general, $\mathrm{dist}(\mathscr{C}_1, \mathscr{C}_2) \neq \mathrm{dist}(\mathscr{C}_2, \mathscr{C}_1)$. Therefore, $\mathrm{dist}(\cdot,\cdot)$ is not a proper distance measure.

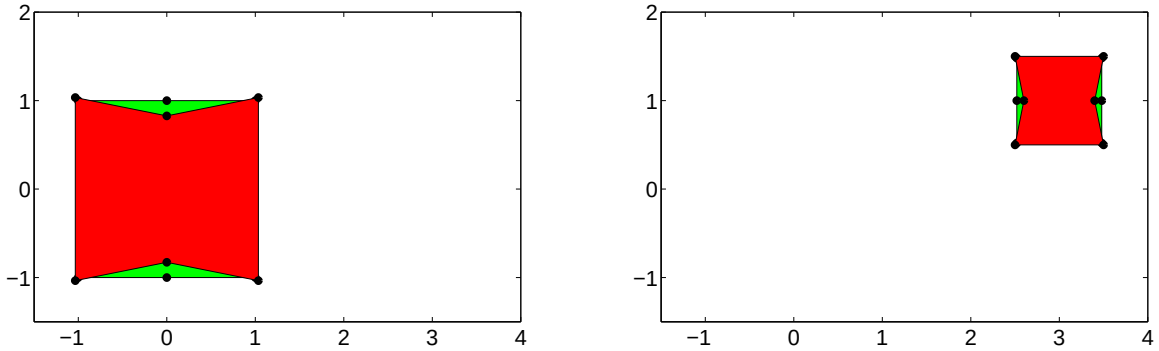

Figure 2: Left: Contours $\mathscr{C}_1$ and $\mathscr{A}_{a_{12},\theta_{12},s_{12}}(\mathscr{C}_2)$; Right: $\mathscr{A}_{a_{21},\theta_{21},s_{21}}(\mathscr{C}_1)$, and $\mathscr{C}_2$.

□

Problem 2 *Algorithms for exact system identification*

Consider an LTI system $\mathscr{B}_0$ and a trajectory $w_\mathrm{d} = (u_\mathrm{d}, y_\mathrm{d}) \in \mathscr{B}_0$ (the subscript 0 stands for "true data generating system"). In lecture 5, we outlined the following algorithms for exact system identification:

- $w_\mathrm{d} \mapsto R(\xi)$, where $\widehat{\mathscr{B}} := \ker\big(R(\xi)\big)$ is the identified model,
- $w_\mathrm{d} \mapsto H$, where H contains the first samples of the impulse response of $\widehat{\mathscr{B}}$,
- $w_\mathrm{d} \mapsto \mathscr{O}_{\ell_\mathrm{max}+1}(A,C) \mapsto (A,B,C,D)$, where (A,B,C,D) is an input/state/output representation of $\widehat{\mathscr{B}}$, and
- $w_\mathrm{d} \mapsto \big(x_\mathrm{d}(1), \ldots, x_\mathrm{d}(\mathtt{n}_\mathrm{max} + \mathtt{m} + 1)\big) \mapsto (A,B,C,D)$.

The purpose of this exercise is to apply the MATLAB implementations `w2r` and `uy2x2ss` of the algorithms $w_\mathrm{d} \mapsto R(\xi)$ and $w_\mathrm{d} \mapsto \big(x_\mathrm{d}(1), \ldots, x_\mathrm{d}(\mathtt{n}_\mathrm{max} + \mathtt{m} + 1)\big) \mapsto (A,B,C,D)$ for exact identification on the data `exactid_data.mat`. Assuming that the data generating system $\mathscr{B}_0$ is controllable and has an upper bound on the lag $\ell_\mathrm{max} = 5$, argue that the identified models, say $\widehat{\mathscr{B}}_1$ and $\widehat{\mathscr{B}}_2$, are MPUM for the data w_d. Conclude that $\widehat{\mathscr{B}}_1 = \widehat{\mathscr{B}}_2 = \mathscr{B}_0$. Verify that the models $\widehat{\mathscr{B}}_1$ and $\widehat{\mathscr{B}}_2$ are exact for w_d, i.e., $w_\mathrm{d} \in \widehat{\mathscr{B}}_1$ and $w_\mathrm{d} \in \widehat{\mathscr{B}}_2$, and are equivalent, i.e., $\widehat{\mathscr{B}}_1 = \widehat{\mathscr{B}}_2$.

Solution: According to the fundamental lemma of lecture 5, $\widehat{\mathscr{B}}_1 = \widehat{\mathscr{B}}_2 = \mathscr{B}_0$ provided 1) the data is exact, 2) the input is persistently exciting of order $\ell_{\max} + \mathtt{n}_{\max}$, and 3) $\mathscr{B}_0$ is controllable. Conditions 1 and 3 are not verifiable from the data but are given as a prior knowledge. In order to check condition 2, we need to verify that $\mathscr{H}_{\ell_{\max}+\mathtt{n}_{\max}}(u_{\mathrm{d}})$ is full row rank. For the data in the example, we have

```
>> rank(blkhank(u,10))
ans = 10
```

so condition 2 is satisfied and therefore by the assumptions stated in the exercise, $\widehat{\mathscr{B}}_1 = \widehat{\mathscr{B}}_2 = \mathscr{B}_0$.

The model parameters obtained by `w2r` with the data given in `exactid_data` are

$$R = \begin{bmatrix} 0.0427 & -0.0053 & -0.2618 & -0.0076 & 0.8329 & -0.2461 & -0.0000 & 0.4187 \end{bmatrix}.$$

In order to validate that $w_{\mathrm{d}} \in \widehat{\mathscr{B}}_1 := \ker\big(R(\sigma)\big)$, we need to check that

$$\begin{bmatrix} R_0 & R_1 & \cdots & R_\ell \end{bmatrix} \mathscr{H}_{\ell+1}(w_{\mathrm{d}}) = 0.$$

In MATLAB,

```
>> norm(r * blkhank(w,l+1))
ans = 5.2619e-015
```

which in the finite precision arithmetic can be considered as 0 and confirms that $w_{\mathrm{d}} \in \widehat{\mathscr{B}}_1$.

In order to validate that $w_{\mathrm{d}} \in \widehat{\mathscr{B}}_2 := \mathscr{B}_{\mathrm{i/s/o}}(A,B,C,D)$, we need to find an initial state x_{ini} that makes the system

$$y_{\mathrm{d}} - \underbrace{\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{T-1} \end{bmatrix}}_{\mathscr{O}_T(A,C)} x_{\mathrm{ini}} + \underbrace{\begin{bmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & \ddots & \ddots & \ddots & \\ CA^{T-1}B & \cdots & CAB & CB & D \end{bmatrix}}_{\mathscr{T}_T(A,B,C,D)} u_{\mathrm{d}} = 0 \tag{6}$$

as compatible as possible. If the residual is 0 (there is an exact solution), then $w_{\mathrm{d}} \in \widehat{\mathscr{B}}_2$. Otherwise, $w_{\mathrm{d}} \notin \widehat{\mathscr{B}}_2$. An inefficient but straightforward procedure for computing an x_{ini} is to form explicitly the extended observability matrix $\mathscr{O}_T(A,C)$, compute the zero initial conditions response $y_{\mathrm{f}} := \mathscr{T}_T(A,B,C,D)u_{\mathrm{d}}$, and solve the system

$$y_{\mathrm{d}} - y_{\mathrm{f}} = \mathscr{O}_T(A,C)x_{\mathrm{ini}}$$

in the least squares sense. For a MATLAB implementation of this procedure, see the function `inistate`. For the data in the example, we have

```
>> [xini,res] = inistate(w(:,1),w(:,2),sys); res
ans = 4.3175e-014
```

which confirms that $w_{\mathrm{d}} \in \widehat{\mathscr{B}}_2$.

Finally, in order to verify that $\widehat{\mathscr{B}}_1 = \widehat{\mathscr{B}}_2$, (assuming that the systems are stable) we check $\|\widehat{\mathscr{B}}_1 - \widehat{\mathscr{B}}_2\|_\infty$

```
>> norm(sysh1-sysh2,'inf')
ans = 2.1356e-014
```

□

Problem 3 *Algorithms for approximate system identification*

The purpose of this exercise is to apply a simple version of the algorithm for approximate identification of lecture 9 on the flutter data `flutter.dat` from the Database for the Identification of Systems (DAISY)

`http://homes.esat.kuleuven.be/~smc/daisy/`

(The data is locally available together with the other functions necessary for this exercise.) Consider the misfit

$$\text{misfit}(w_{\text{d}}, \mathscr{B}) := \min_{\widehat{w}} \|w_{\text{d}} - \widehat{w}\|_2 \quad \text{subject to} \quad \widehat{w} \in \mathscr{B} \tag{7}$$

between the data w_{d} and a model $\mathscr{B}$. Geometrically, $\text{misfit}(w_{\text{d}}, \mathscr{B})$ is the orthogonal projection of w_{d} on $\mathscr{B}$. Assuming that $\mathscr{B}$ is controllable, let $\mathscr{B} = \text{image}\big(M(\sigma)\big)$ be a minimal image representation. In terms of the parameter M, the constraint $\widehat{w} \in \mathscr{B}$ becomes $\widehat{w} = M(\sigma)l$, for some latent variable l. In matrix form,

$$\widehat{w} = \mathscr{T}_T(M)l, \qquad \text{where} \quad \mathscr{T}_T(M) := \begin{bmatrix} M_0 & & & \\ M_1 & M_0 & & \\ \vdots & M_1 & \ddots & \\ M_\ell & \vdots & \ddots & M_0 \\ & M_\ell & & M_1 \\ & & \ddots & \vdots \\ & & & M_\ell \end{bmatrix} \in \mathbb{R}^{\mathtt{w}T \times (T+\ell)}.$$

Then the misfit computation problem (7) is equivalent to the standard least squares problem

$$\text{minimize} \quad \text{over } l \quad \|w_{\text{d}} - \mathscr{T}_T(M)l\| \tag{8}$$

and is implemented in MATLAB in the functions `misfit` and `misfit2`. The latter does not exploit the structure of the matrix $\mathscr{T}_T(M)$, while the former exploits the banded (but not Toeplitz) structure of $\mathscr{T}_T(M)$.

1. Consider the model $\mathscr{B} = \ker\big(R(\sigma)\big)$, defined by

   $$R(\xi) = R_0 + \xi R_1 + \xi^2 R_2 + \xi^3 R_3,$$

   where

   $$\begin{aligned} R_0 &= \begin{bmatrix} -1.0666 & -0.7235 \end{bmatrix}, \quad R_1 = \begin{bmatrix} 2.8745 & 2.3369 \end{bmatrix}, \\ R_2 &= \begin{bmatrix} -2.7278 & -2.5736 \end{bmatrix}, \quad R_3 = \begin{bmatrix} 0.8924 & 1.0000 \end{bmatrix}. \end{aligned} \tag{9}$$

   (This model is computed by the `w2r` function.) Compute the misfit between the flutter data and $\mathscr{B}$ with `misfit` and `misfit2`, measuring the computation time (see `help toc` in MATLAB), and compare the obtained answers. Comment on the results.

2. *Misfit minimization* Next we consider the misfit minimization problem

   $$\widehat{\mathscr{B}}_{\text{gtls}} := \arg\min_{\widehat{\mathscr{B}}} \quad \text{misfit}(w_{\text{d}}, \mathscr{B}) \quad \text{subject to} \quad \widehat{\mathscr{B}} \in \mathscr{L}^{\mathtt{w}}_{\mathtt{m},\ell}, \tag{10}$$

   where the number of inputs $\mathtt{m} < \mathtt{w}$ and the lag ℓ are given natural numbers. Considering, again the image representation and restricting for simplicity to the SISO case, i.e., $\mathtt{w} = 2$ and $\mathtt{m} = 1$, (10) is equivalent to

   $$\text{minimize} \quad \text{over } M \in \mathbb{R}^{2(\ell+1)\times 1} \quad \text{misfit}\big(w_{\text{d}}, \text{image}\big(M(\sigma)\big)\big) \quad \text{subject to} \quad M \neq 0, \tag{11}$$

   which is a constrained nonlinear least-squares problem. Partition the parameter $M(\xi)$ of the image representation as $M =: \left[\begin{smallmatrix} p \\ q \end{smallmatrix}\right]$ and assumed that the polynomial p is monic (highest order coefficient is one). With this assumption, problem (11) becomes unconstrained

   $$\text{minimize} \quad \text{over } M' \in \mathbb{R}^{2\ell+1\times 1} \quad \text{misfit}\big(w_{\text{d}}, \text{image}\big(\left[\begin{smallmatrix} M' \\ 1 \end{smallmatrix}\right](\sigma)\big)\big). \tag{12}$$

The function `gtls` solves (12), using the function `fminsearch` from the Optimization Toolbox of MATLAB. (`fminsearch` implements the Nelder-Mead local optimization method.)

Partition the flutter data set into identification, e.g., first 60%, and validation, e.g., remaining 40%, parts. Compute a locally optimal model with lag $\ell = 3$ for the identification part of the data, using as an initial approximation (9). Validate the identified model by computing the misfit on the validation part of the data. Show the best fit of the validation data by the identified model.

3. *Compare with PEM* A classical method for system identification is the prediction error method (PEM) and a popular implementation of the PEM method is the function `pem` from the System Identification Toolbox of MATLAB. Similarly to the misfit minimization method (10), the PEM method is based on local optimization, starting from an initial approximation. However, the PEM method minimizes a different cost function. Using its default settings, the `pem` is applied on the identification part of the flutter data by

```
>> load flutter.dat
>> sys_pem = pem(iddata(flutter(1:600,2),flutter(1:600,1)),3);
```

(see `help pem` and `help iddata` for details on the usage of these functions). The following script validates the PEM model using the `misfit` function

```
>> sys_pem = ss(sys_pem); % convert sys_pem from the idss to the ss format
>> % apply misfit on the deterministic part of the model
>> [mv_pem,wh_pem] = misfit(flutter(601:end),sys_pem(1,1));
```

A validation function from the System Identification Toolbox is `compare`. The following script validates the GTLS model using `compare`

```
>> % convert sys_gtls from the ss to the idss format and apply compare
>> compare(iddata(flutter(601:end,2),flutter(601:end,1)),idss(sys))
```

Using the functions `misfit` and `compare`, validate the `pem` and `gtls` identified models on the validation part of the data. Repeat the experiment for different partitionings of the data into identification and validation parts. Comment on the results.

Solution:

1. Since a least squares problem of dimension $2T \times (T+\ell)$ is solved by general purpose methods, the computational complexity of `misfit2` is $O(T^3)$ flops. The efficient implementation of (8) that exploits the banded structure of the $\mathscr{T}_T(M)$ matrix has computational complexity $O(T)$ flops. The plot of the trajectory $\widehat{w}$ of (9) that best fits the data w_{d} in the misfit sense is given in Figure 3. The misfit is 9.5017.

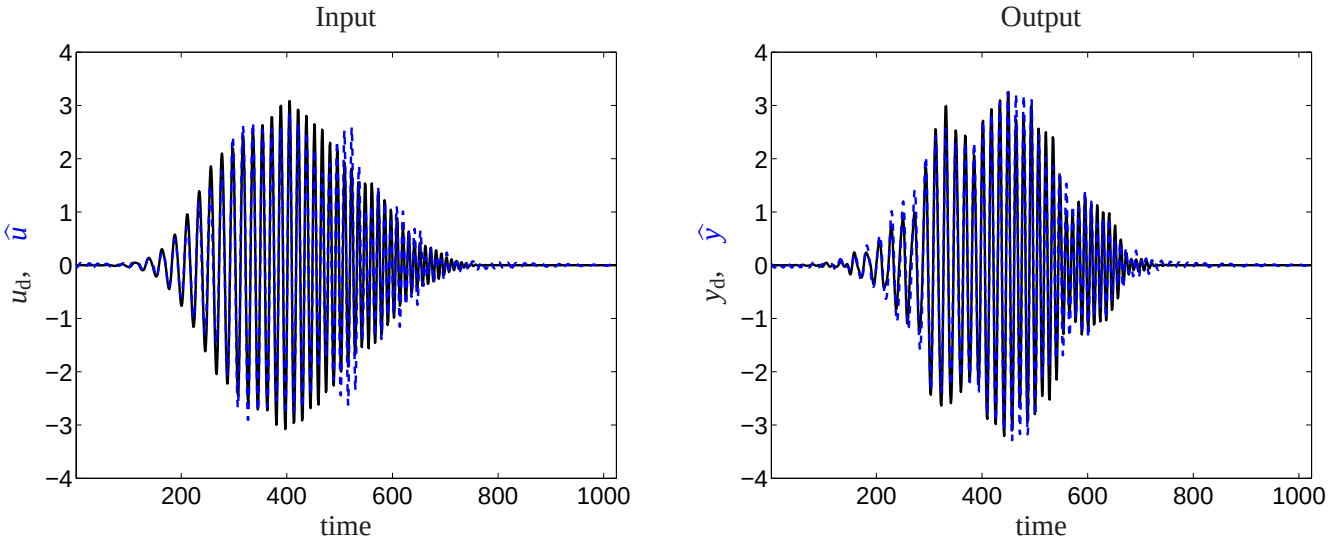

Figure 3: Best fit of the flutter data by the model (9).

2. and 3. Applied on the identification part of the data (first 600 samples) and initialized by the model (9), the `gtls` function computes a model that achieves misfit 1.9 with respect to he validation part of the data (remaining 424 samples). Using the default settings, the `pem` function, computes a model that achieves misfit 4.67 on the validation data.

   Figure 4 shows the best, in the sense (7), fit of the validation data by the `gtls` and `pem` models. The corresponding fit in the output error sense, computed by the `compare` function, is given in Figure 5. In this particular example `gtls` achieves better model than `pem` in both the misfit sense as well as in the output errors sense. There is no guarantee, however, that the same happens on other data sets or even on the same data when different partitioning of the data and/or different initial approximations are used.

   There are two main reasons for the dependence of the results on the simulation setting:

   - The misfit and output error minimization problems are nonconvex and the `gtls` and `pem` compute only locally optimal solutions. These solutions are sensitive to the initial approximations.
   - Even if a globally optimal minimum of the misfit and output error minimization problems are found, the corresponding models may (will almost certainly) not be optimal on the validation data.

   Good fit on the identification data would indeed correspond to a good fit on the validation data, if the data were generated by an LTI model, which is in the considered model class. In practice, however, this is likely to hold only approximately. Therefore, for data which is not well approximated by an LTI model the mismatch between the identification and validation fits may be significant.

   Despite item 1 above, experimental evidence may *suggest* that certain optimization methods are "more robust" to the initial approximation in finding better (or even global) minimum point. On the average such methods give better results than other methods. The Nelder-Mead optimization method needs weak assumptions (in particular smoothness of the cost function is not required) however is rather inefficient. The `pem` function uses algorithms for nonlinear least squares (e.g., the Levenberg–Marquardt algorithm), which assume smoothness and are more efficient, however, such algorithms tend to be less robust.

□

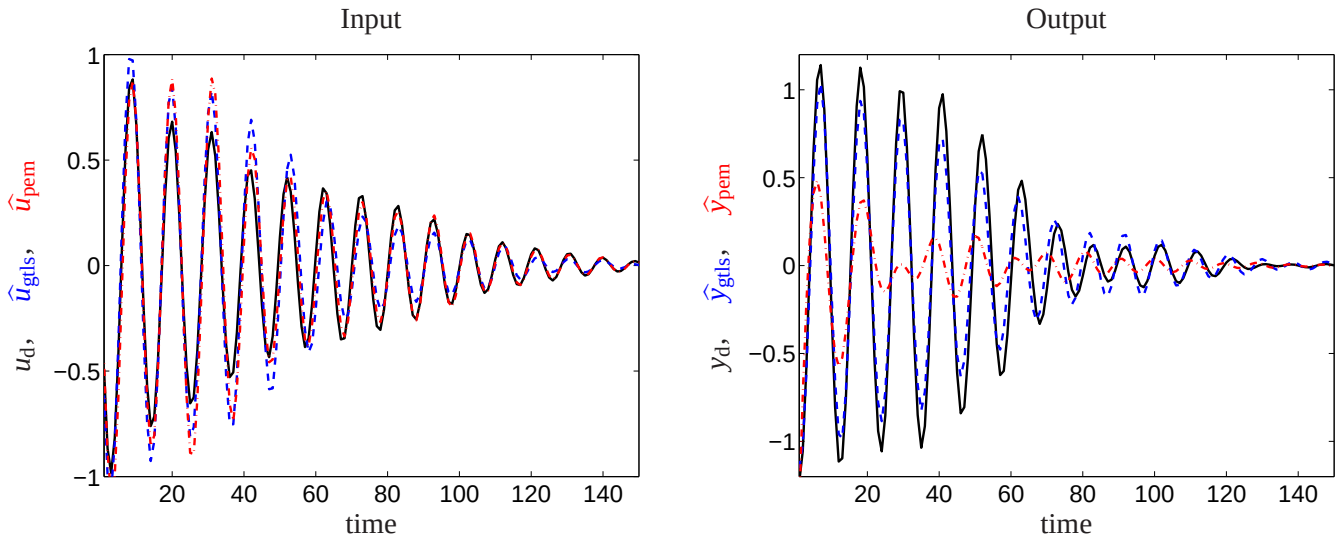

Figure 4: Best fit in the misfit sense (`misfit` function) of the validation data (solid line) by the GTLS (dashed line) and PEM (dashed-dotted line) models.

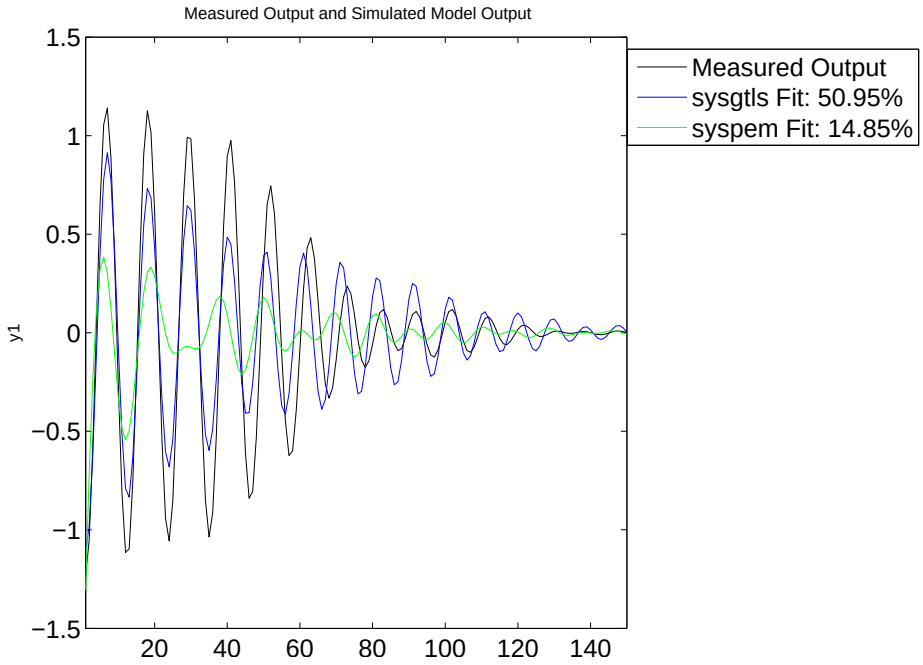

Figure 5: Best fit in the output error sense (`compare` function) of the validation data by the GTLS and PEM models.