Control Maps: Visualization of Local Robustness Properties for Control System Design

Ignacio Díaz ∗ Antonio Robles ∗ Abel A. Cuadrado ∗ Alberto B. Diez ∗ Manuel Domínguez ** Juan J. Fuertes **

Abstract: In this work the authors propose a method to visualize local robustness properties of a nonlinear control system by means of a projection mapping defined by a self-organizing map (SOM). This kind of visualization is potentially useful for control system design since local properties of the closed loop control system—sensitivity functions—are visualized, revealing operating points of critical performance, and suggesting tuning of the controllers or even adaptive control strategies by means of localized controllers. The proposed method is verified under simulation, showing a good coincidence between the small signal response around different operating points and maps of models built automatically from control system data under different working conditions. Finally, some concluding remarks, including limitations of the proposed method and suggestions for future work are given.

Keywords: Control system design; Nonlinear systems; Dynamic modelling; Robustness;

1. INTRODUCTION

Projection techniques have been largely used for modeling and visualization of nonlinear systems. These techniques consist in defining a mapping that projects a large number of process variables on a set of a few latent or explanatory variables that convey the most significant information about the process. Such projections define a mapping that relates the input space to a low dimensional latent space (often 2D or 3D for visualization) allowing to represent the process features in the same way as a cartographic map of the process (actually using similar techniques), often giving an extraordinary insight of the process nature.

Self-organizing map (SOM), with more than 5000 research articles—see Oja et al. [2003]—, is a popular nonlinear projection technique that has been extensively used in a large number of engineering applications including control, process monitoring and data visualization—see Kohonen et al. [1996].

However, in their basic form, projection techniques, and particularly the SOM, capture merely geometrical relationships between the process features by means of the latent variables. Therefore, they only allow to describe static features about the process. Fortunately, there exist ways to consider the dynamic behavior under projection techniques. In Kohonen [1993] the author already defines the idea of operator maps in which a dynamical model is assigned to each SOM unit. Also, a related approach called local dynamic modelling, allowing to estimate dynamic models from local data and assign them to SOM units, was proposed in Principe et al. [1998]. Later, the same author and others used these local models to define local controllers for aerospace UAV control in Cho et al. [2006]. In Barreto and Araújo [2004], the authors propose a rather different approach called VQTAM (vector-quantized temporal associative memory), using SOM regression on delayed inputs and outputs, in which no explicit model is stored in the SOM. In Barreto [2007] a good survey in methods to incorporate dynamic information in SOM is described. Surprisingly, despite one of the main potential of SOM is visualization, no significant works has been done to visualize process dynamics until recently. In Díaz Blanco et al. [2007, 2008] the authors presented a method that allowed to visualize a map of the dynamical behaviours of systems under different conditions—e.g. operating points—using self-organizing maps (SOM).

Closed loop systems, and particularly nonlinear control systems, exhibit dynamical behaviours that often become nontrivial even for control experts. Depending on the working point the loop dynamics behave locally in different ways, leading to different performance and stability properties. Sensitivity functions have been extensively used in control system design since they reveal, in an easy graphical way, complete information regarding stability, robustness as well as dynamic and steady state performance of the closed loop; furthermore, in addition, they serve as
guides to improve the control system performance during the design process. However, sensitivity functions are no longer descriptive when system operates far beyond the linearizing point, therefore making it difficult to develop systems that need to operate in a large range of operating points.

In this work, we propose a method to visualize the dynamical performance of nonlinear control systems in terms of maps, that allow the control engineer to assess the influence of the working point or other parameters on the local robustness properties of the control system. Despite the method has some limitations, the results are encouraging and suggest that it can be potentially very useful as a control system design aid.

This paper is organized as follows. In section 2, the local dynamic modeling method that allows to build a SOM map of the loop dynamics is briefly outlined. In section 3 the application of this method to visualize important features for control system design is presented. In section 4, sensitivity maps are obtained for a magnetic levitation system. Finally section 5 provides some general discussion and concludes the paper.

2. LOCAL DYNAMIC MODELING SOM

This section describes an extension of the SOM to model dynamic systems, based on the local dynamic modeling approach originally proposed by Principe et al. [1998], Cho et al. [2006], with some variations introduced in Díaz Blanco et al. [2007] for visualization of dynamics.

Let’s consider the following nonlinear system that express the present output as a function of past outputs and inputs as well as a set of parameters.

\[
y(k) = f(y(k-1),...,y(k-n_y),u(k),...,u(k-n_u),p_1(k),...,p_p(k))
\]

This model can be expressed in a more compact way as

\[
y(k) = f(\phi(k),p(k))
\]

where \(\phi(k) = [y(k-1),\ldots,y(k-n_y),u(k),\ldots,u(k-n_u)]^T\) is a vector of known data at sample \(k\), \(p = [p_1(k),\ldots,p_p(k)]^T\) is a vector of parameters, and \(f(\cdot,\cdot)\) is a given functional relationship that may be linear or nonlinear. Model (2) describes a dynamic relationship between the process inputs and outputs determined by the values of \(p_1(k),\ldots,p_p(k)\).

The process to model the system dynamics involves two stages:

- Mapping the space of operating points of the process using a projection algorithm (e.g. self-organizing map).
- Obtaining a local dynamic model for each point of the map that describes the behavior of the process around it.

2.1 Mapping the dynamics

Let’s consider a set of available process variables \(s(k) = [s_1(k),\ldots,s_u(k)]^T \in \mathcal{S}\) that are known or supposed to discriminate different dynamical process behaviors. These variables are called selectors of dynamics and are selected by the user on the basis of application domain knowledge.

In a first stage, a SOM with \(N\) units is trained in the selectors space \(\mathcal{S}\) to cluster the process dynamics. Each unit \(i\) is associated to a \(s\)-dimensional prototype vector \(m_i\) in \(\mathcal{S}\) and a position vector on a low dimensional regular grid, \(g_i\), in the output space. For each vector \(s(k)\) the best matching unit is computed as

\[
c(k) = \arg \min_i \|s(k) - m_i(t)\|
\]

Then, an adaptation stage is performed according to

\[
m_i(t+1) = m_i(t) + \alpha(t) h(c(k),i) [s(k) - m_i(t)]
\]

where \(\alpha(t)\) is the learning rate and \(h(c,i) = e^{-\frac{(c-i)^2}{2\sigma^2(t)}}\) is the neighborhood function. After convergence, the result is a set of prototype vectors that divide \(\mathcal{S}\) into a finite set of Voronoi regions each of which defines a different dynamic behavior.

2.2 Local model estimation

Once a SOM has been trained with the \(\{s(k)\}\), a model \(i\) may be estimated for each prototype \(m_i\) using all the pairs \(\{(y(k),\phi(k))\}\) such that

\[
\|g_{-i} - g_i\| \leq \sigma_{loc}
\]

that is, those pairs whose corresponding selectors \(s(k)\) are mapped onto a neighborhood of \(m_i\) of width \(\sigma_{loc}\).

If function \(f(\cdot)\) in model (1) is chosen to be linear, eq. (2) becomes

\[
y(k) = p_i^T \phi(k) + \varepsilon(k)\]

whose parameters \(p_i\) can be obtained using least squares to fit the aforementioned pairs \(\{(y(k),\phi(k))\}\).

The result of the whole process is a linear model, as the one described in (6), associated to each neuron \(i\), that can be rewritten in familiar notations for control engineers, such as the difference equation

\[
y(k) = \sum_{j=1}^{n_u} a_{j} y(k-j) + \sum_{j=0}^{n_u} b_{j} u(k-j) + \varepsilon(k)
\]

or as a transfer function

\[
G(z, p_i) = \frac{Y(z)}{U(z)} = \frac{\sum_{j=0}^{n_u} b_j z^{-j}}{1 - \sum_{j=1}^{n_u} a_j z^{-j}}
\]

Both forms are completely defined by the parameter vector \(p_i = [a_1,\ldots,a_{n_u},b_0,b_1,\ldots,b_{n_u}]\).

2.3 Prediction and Estimation

Once the model is trained, a local model \(f(\phi(k),p_i)\) is assigned to each neuron \(i\). It is possible to estimate \(y(k)\) given \(\phi(k)\) and the dynamic selectors \(s(k)\). This is accomplished in two steps: 1) obtaining the best matching unit, \(c(k) = \arg \min_i \|s(k) - m_i(t)\|\) and 2) applying the local model \(y(k) = f(\phi(k),p_{c(k)})\). Note that depending on the problem (one or multiple step ahead prediction) data vector \(\phi(k)\) may contain real or estimated past outputs.
3. APPLICATION TO CONTROL

A general feedback loop—Fig. 1—can be considered as a MIMO system, being influenced by external inputs such as the reference $r$, the load disturbance $d$, the measurement noise $n$. The remaining signals—process output $y$, control action $u$, error signal $e$, etc.—can be of interest to specify performance in controller design—see Aström and Murray [2010].

![Fig. 1. Generic feedback loop.](image)

A set of four transfer functions $(S, T, S_u, S_v)$ \(^1\) called “sensitivity functions” can be obtained. For linear(ized) systems, these functions summarize the control system performance with regard to robustness, stability, tracking, disturbance rejection and control action, among others—Goodwin et al. [2001], Aström and Murray [2010].

Assuming a proper selection of the sampling period—according to the system’s bandwidth—for system identification, the method described in section 2 allows to obtain a visual map of dynamical models just from input-output data of a system running under different dynamic behaviors. In the resulting map, each point represents a dynamical model—a discrete transfer function—, if local models are chosen to be linear.

For a control system such as the one described in Fig. 1, it is possible to

a) assign one of the inputs $\{r, d, n\}$ of the control loop to the input $u(k)$ of the dynamical model in the method of section 2,

b) assign one of the outputs $\{u, y, e, \nu, \eta\}$ of the control loop to $y(k)$, and finally

c) assign one or more variables defining the operating point to $s(k) = [s_1(k), s_2(k), \cdots, s_n(k)]^T$.

These assignments allow to build a map of any of the four aforementioned sensitivity functions and represent it for all the operating points that the feedback control system reaches for the training dataset. Maybe the most important sensitivity function in robust control design is $S(s) = 1/(1 + L(s))$, being $L(s)$ the loop transfer function, since it is closely related to robustness and stability properties of the closed loop system—see Aström and Murray [2010].

If the system has no prefilter ($F = 1$) and one takes $r$ as input and $e$ as output for the local dynamic model SOM algorithm of section 2, the outcome will be a map with a local sensitivity function for each node $i$

$$S^{(i)}(z) = \frac{E(z)}{R(z)}$$

for each $m_i$ that has been identified using process data around the working point defined by $m_i$.

Since a sensitivity transfer function is defined for each SOM node, any kind of insightful scalar feature that can be derived from the sensitivity function (e.g. maximum sensitivity $M_s$, frequency of the maximum sensitivity $\omega_m$, or others such as vector margin, robust phase and gain margins, etc.) can be represented using a new component plane. Moreover, even the whole frequency response of the sensitivity functions for all operating points can be represented using small glyphs.

Since all information is extracted from the same mapping learned by the SOM algorithm, contextual information such as the variables defining the working point or design parameters can be represented using similar maps, that can be seamlessly compared to the sensitivity maps, allowing to discover relationships with the control system performance.

4. RESULTS

The visualization method described in the previous sections was applied to a simulated nonlinear model of a magnetic levitation system presented in Shiao [2001].

![Fig. 2. Magnetic levitation control system.](image)

The dynamic response of the ball driven by the electromagnet involves the gravity and magnetic attraction forces. Applying Newton’s law, the dynamics of the ball motion are described by

$$m \frac{d^2x(t)}{dt^2} = mg - f(x, t)$$

Magnetic force can be derived from theoretical models

\(^1\) six functions for more general 2 degree of freedom controllers
\[ f(x,t) = K \cdot \left( \frac{i(t)}{x(t)} \right)^2 \]  

Even more realistic models can be attained changing the \( x^2 \) term for third or fourth degree polynomials adjusted to experimental data—see El Hajjaji and Ouladsine [2002]—but Shiao’s model has been considered good enough for the purpose of this work because it captures the essential nonlinearities of the plant.

Experimentally determined parameters have been borrowed from Shiao [2001], being \( m = 0.225 \text{ Kg} \), \( K = 7.938 \times 10^{-5} \text{ Nm}^2/\text{A}^2 \), and the sensor gain \( \beta = 200 \).

Saturation of \( \pm 8 \text{ A} \) in the current delivered by the driver has also been taken into account in the simulation model. The working point was chosen with the ball suspended 3 cm under the coil \( (x_0 = 0.03 \text{ m}) \), defining \( h = x_0 - x \) as the relative height of the ball—see Fig. 2. Once the model was linearized, a simple PID controller, \( C(s) \), was designed for this working point

\[ C(s) = 200 \cdot 15 \frac{(s + 6)(s + 25)}{s + 200} \]  

where the 200 factor (sensor gain) was included in the controller transfer function to normalize the block diagram to unity feedback. The block diagram of the control system is represented in Fig. 3 and Fig. 4 shows the controlled system root locus.

![Control system block diagram](image)

**Fig. 3. Control system block diagram.**

![Root locus of the linearized system](image)

**Fig. 4. Root locus of the linearized system.**

Simulation data of reference position \( r(t) \), the actual position \( h(t) \) of the ball and the error signal \( e(t) = r(t) - h(t) \) were obtained making the system work with small random variations around a smoothly changing reference position going far from the working point in both directions.

![Figure 5](image)

**Fig. 5. Reference value \( r(t) \) (top) and error signal \( e(t) \) (bottom) for different operating points around the linearization point \( (r = 0) \).**

In order to explore the changes in the control system performance with respect to the reference value, the local dynamic model SOM described in section 2 was applied. The following assignments were done:

\[
\begin{align*}
    u(k) &\leftarrow r(kT_m) \\
    y(k) &\leftarrow e(kT_m) \\
    s(k) &\leftarrow r(kT_m)
\end{align*}
\]  

where a sample period of \( T_m = 0.005 \text{ s} \) was chosen as a reasonable choice according to the system’s bandwidth—in the order of 10 Hz, depending on the operating point. A 15 × 15 SOM was trained in the selectors space, using the batch algorithm for 10 epochs and a neighborhood \( \sigma \) monotonically decreasing from 5 to 0.5. Once the selectors space was clustered, local models for each \( \mathbf{m}_i \) were estimated using LS identification for local linear models of the type described in eq. (7), for orders \( n_y = \{2, 3, 5, 10, 15, 20\} \), \( n_u = \{2, 5\} \) including their combinations.

<table>
<thead>
<tr>
<th>Iter.</th>
<th>( n_y )</th>
<th>( n_u )</th>
<th>( \sigma_{oc} )</th>
<th>RMSE_{loc} (\times 10^3)</th>
<th>RMSE_{lin} (\times 10^3)</th>
<th>AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>0.1342</td>
<td>0.1471</td>
<td>-17.83</td>
</tr>
<tr>
<td>#2</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>0.04657</td>
<td>0.1043</td>
<td>-19.95</td>
</tr>
<tr>
<td>#3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0.06284</td>
<td>0.1022</td>
<td>-19.35</td>
</tr>
<tr>
<td>#4</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>0.05221</td>
<td>0.1493</td>
<td>-19.64</td>
</tr>
<tr>
<td>#5</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>0.04484</td>
<td>0.1004</td>
<td>-20.02</td>
</tr>
<tr>
<td>#6</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>0.04285</td>
<td>0.09919</td>
<td>-20.11</td>
</tr>
<tr>
<td>#7</td>
<td>10</td>
<td>2</td>
<td>3</td>
<td>0.03737</td>
<td>0.1006</td>
<td>-20.39</td>
</tr>
<tr>
<td>#8</td>
<td>10</td>
<td>5</td>
<td>3</td>
<td>0.03864</td>
<td>0.09779</td>
<td>-20.32</td>
</tr>
<tr>
<td>#9</td>
<td>15</td>
<td>2</td>
<td>3</td>
<td>0.03602</td>
<td>0.1007</td>
<td>-20.46</td>
</tr>
<tr>
<td>#10</td>
<td>15</td>
<td>5</td>
<td>3</td>
<td>0.03743</td>
<td>0.09786</td>
<td>-20.38</td>
</tr>
<tr>
<td>#11</td>
<td>20</td>
<td>2</td>
<td>3</td>
<td>0.03577</td>
<td>0.1007</td>
<td>-20.47</td>
</tr>
<tr>
<td>#12</td>
<td>20</td>
<td>5</td>
<td>3</td>
<td>0.0372</td>
<td>0.09788</td>
<td>-20.39</td>
</tr>
</tbody>
</table>

Table 1. Table for model order selection.

The model with the least AIC (Akaike’s Information Criterion) was chosen. As seen in table 1, the best performing model according to its AIC index is the ARX(20,2). Note
that the local model estimator shows an error of only about 35% of the linear (global) model. In Fig. the real and estimated output $e(k), \hat{e}(k)$ are shown for both the linear global model (above) and the local dynamic model SOM.

As seen, the local dynamic model SOM is substantially more accurate. This stems from the fact that the obtained model stores a local dynamical model (in this case a more accurate. This stems from the fact that the obtained model stores a local dynamical model (in this case a

The map of $M_s$ in Fig. 7 shows the smallest values when the system operates near the reference position $r \approx 0$, showing that the stability is best in that region. In turn, the $M_s$ map shows larger values (red tones) when the ball is made to levitate below and above the reference position. On the other hand, the map of $\omega_{ms}$ clearly shows that the system’s bandwidth is directly related to the proximity of the ball to the electromagnet.

An alternative representation graphing the Bode diagrams of the sensitivity functions as small glyphs on each neuron position is shown in Fig. 8. This representation can provide extra information such as the presence of resonances on certain operating points. For instance, for ball positions near the electromagnet (described by the upper right corner) the sensitivity functions show an emerging resonance at frequencies below $\omega_{ms}$. The presence of a new low frequency oscillating mode – noticeable as a small overshoot above 0— can be distinguished in Fig. 9 on the response to the last step (around time $t = 48.75$ s). This can also be observed in the zoomed error signal $e(t)$ for operation far above the linearization point (bottom row, right subfigure).

To show the consistency of the previous results, the model was simulated for small steps in three regions: below the reference position ($r \approx -0.002$), above it ($r \approx 0.012$) and finally far above the reference position ($r \approx +0.024$), near the stability frontier. As seen in Fig. 9, the error signal shows slightly larger oscillations for positions below the reference position, smoothly decreasing as $r$ increases, and then abruptly increasing again for reference positions well above $r = 0$.

5. CONCLUSIONS

In this paper an original method to visualize nonlinear control system performance with respect to its operating point has been proposed. The method proposes using a local dynamic modeling SOM to learn the dynamic relationships between two (or more) variables of the control system for all operating points, allowing to exploit the excellent visualization capabilities of SOM for later visualization of the control system performance at different regions of operation. The method was demonstrated on a simulated model of a magnetic levitation control system, for which a map of the sensitivity function $S(j\omega)$ for different operating points was automatically learnt by the SOM. Several insightful visualizations have been derived from the learnt map, including component plane visualizations.
Fig. 9. Simulation of the error signal $e(t)$ for small steps ranging from negative to positive values of the reference position $r(t)$—units are in meters. The two bottom rows describe details of the above subfigure in three different working regions.

of the operating point (height of the ball), the maximum sensitivity $M_e$ and its frequency $\omega_{ms}$, and even maps of glyphs—small Bode diagrams of $S(j\omega)$. Since all the visualizations are derived from the same model, which is indeed a map of local transfer functions—more general dynamical models are allowed, however, direct comparison between visualizations is possible, allowing to discover nontrivial relationships among the operating point and the dynamic control performance in a natural way, therefore resulting in a potentially useful tool for control system design.

Anyway, despite the results presented in this paper are encouraging, further work remains to be done in testing the method on more difficult problems and solving some open questions, such as those arising from the nature and persistence of the excitation signal, or the types of nonlinearities that the algorithm is able to deal with. It has been found that, in some cases, the algorithm fails to produce understandable local models even when it accurately predicts the output. In most cases, however, a good application of typical system identification recommendations and a careful tuning of the algorithm parameters often solves the problem.

In sum, we believe that the results are encouraging and the ideas presented in this paper suggest a highly visual and intuitive approach to control system design of nonlinear systems and opens the way to further developments in new design methodologies.

REFERENCES


