ORBIT - Operating-Regime-Based Modeling and Identification Toolkit

Tor A. Johansen [a] Bjarne A. Foss [a]

[a] *Department of Engineering Cybernetics, The Norwegian University of Science and Technology, N-7034 Trondheim, Norway.*

ORBIT is a MATLAB-based toolkit for black-box and grey-box modeling of non-linear dynamic systems. The model representation is based on multiple local models valid in different operating regimes, that are smoothly blended into a global non-linear model. ORBIT is a computer-aided modeling environment that supports the interactive development of regime-based models on the basis of a mixture of empirical data and prior knowledge (similar to Takagi-Sugeno-type fuzzy models). ORBIT contains functions for robust parameter identification, including constrained identification, regularization and Bayesian identification. There are functions for structure identification, including selection of the local model orders as well as decompositions into operating regimes that give a good fit to the data. Model validation, simulation, visualization and automatic code generation are also supported.

Key words: Software tools, system identification, nonlinear systems, fuzzy systems.

1 Introduction

There is a growing interest in modeling and identification methods based on the explicit decomposition of the operating range of dynamic systems into operating regimes and the use of simple local models within each operating regime, see e.g. (Johansen and Foss 1997, Murray-Smith and Johansen 1997, Takagi and Sugeno 1985, Jacobs *et al.* 1991). Such dynamic models have found wide applicability in model predictive control (Foss *et al.* 1995, Chow *et al.* 1995), gain scheduling control (Hunt and Johansen 1997, Banerjee *et al.* 1995) and fuzzy control (Takagi and Sugeno 1985, Zhao *et al.* 1995). SINTEF and the Norwegian University of Science and Technology have developed a research tool to facilitate further research and development of this technology, as well as internal and external use in education and industrial research projects. The MATLAB-based software tool is called ORBIT (Operating-Regime-Based modeling and Identification Toolkit).

Mathematical modeling is the science of transforming the available empirical and mechanistic data and knowledge into a set of equations that are useful for the intended application. The blending of empiricism with an understanding of the mechanisms of the system is of fundamental importance. ORBIT is an interactive environment for computer-aided modeling and system identification, typically leading to mathematical models that can be viewed as grey-box models. Depending on the application and the available data and knowledge, ORBIT can be applied as anything between a purely data-driven automatic modeling tool and a graphical user interface (GUI) for manual specification of the model. Advanced methods for non-linear system identification form the core of ORBIT.

The regime-based models developed in ORBIT are closely related to Takagi-Sugeno type fuzzy models (Takagi and Sugeno 1985). A distinguished feature of ORBIT, compared for example to the MATLAB Fuzzy Logic Toolbox, is the strong focus on the application rather than on the fuzzy systems technology. In ORBIT an attempt has been made to hide as many as possible of the more technical details of fuzzy systems. For example, there is no flexibility in ORBIT with respect to the selection of the membership function parameterization or the fuzzy operators. On the other hand, the user has been given extensive flexibility on the application level. For example, it is possible to constrain the model parameters individually during identification, and to analyze and tune the individual local models. The authors believe that this makes ORBIT a useful tool for the application-oriented researcher.

This paper is structured as follows: In Section 2 the fundamentals of operating-regime-based modeling are reviewed. An overview of the ORBIT tool is given in Section 3. More details on the algorithms and functionality in ORBIT follow in Section 4. The paper ends with some concluding remarks in Section 5. A laboratory-scale identification example is used throughout the paper to illustrate the ideas.

2 Operating-Regime-Based Modeling

The purpose of this section is to provide a brief outline of the basics of operating-regime-based modeling. More details can be found in, e.g. (Murray-Smith and Johansen 1997).

2.1 Local models and operating regimes

Any model will have a limited range of validity. This range may be restricted by the modeling assumptions for a mechanistic model, or by the experimental conditions under which the data was logged for an empirical model. To emphasize this aspect, a model that has a range of validity that is smaller than the desired range of validity will be called a "local model", as opposed to a "global model" that is valid in the full range of operation. The region in which a local model is valid, is called an "operating regime".

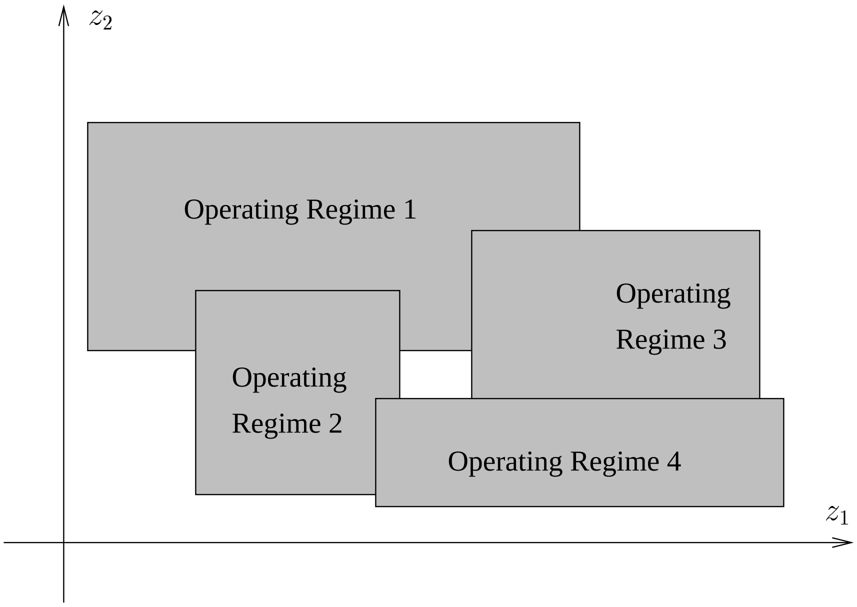

Fig. 1. Decomposition of operating range into regimes

The operating-regime-based modeling framework is conceptually illustrated in Fig. 1. The system's full range of operation (characterized by some vector $z = (z_1, ..., z_d))$ is covered by a number of possibly overlapping operating regimes. Within each operating regime the system is modeled by a local model, and the local models are blended into a global model using weighting functions. Notice that the operating regimes are not really regions with hard boundaries as illustrated in the figure, but rather regions with soft boundaries. This means that there will be a gradual transition between local models when moving between operating regimes. This is implemented as an interpolation technique, which can also be viewed as a fuzzy inference system (Takagi and Sugeno 1985).

One motivation behind this framework is that global modeling is complicated because one would need to describe the interactions between a large number of phenomena that appear globally. Local modeling, on the other hand, may be considerably simpler, because locally there may be a smaller number of phenomena that are relevant, and their interactions are simpler. For instance, while a non-linear model may be needed globally, linear models will be sufficient locally. A non-linear modeling problem can therefore be solved within the operating-regime-based modeling framework by decomposing the operating range into a number of operating regimes, and developing simple linear models within each operating regime.

2.2 Flexible modeling

One of the nice features of the operating-regime-based modeling framework is that the requirements in terms of system knowledge are quite reasonable. Often, quite elementary and qualitative system knowledge, combined with reasonable amounts of measured data, is sufficient to develop complex and accurate non-linear models based on operating regimes and local models.

Within this framework, modeling involves three major tasks:

(i) Select a decomposition into operating regimes. This includes the selection of the characteristic variables used to describe the operating regimes. For this purpose one can use a qualitative understanding of the system's behavior or internal phenomena, or one can select the model structure using some structure-identification algorithm and empirical data.
(ii) Select the structure of the local models. Again, one can use a combination of qualitative understanding and empirical data. The local model structures can be chosen as empirical model structures, or one can develop mechanistic local model structures.
(iii) Tune the local model parameters. For this part, empirical data and an identification algorithm are usually applied.

In some operating regimes, the system may be described by an empirical input/output model, while in other operating regimes it may be described by a mechanistic state-space model. In other words, the framework is flexible with respect to the type of system knowledge that can be applied.

Moreover, the framework supports incremental modeling and simple model maintenance, because the modification of a single local model or operating regime has a limited and quite predictable effect on the global model. In addition, one may use local models with different levels of accuracy, according to what is required in the different operating regimes.

2.3 Modeling without equations

One of the goals with the operating-regime-based modeling framework is to elevate the modeling problem somewhat from the equation level. With operating regime based models, the structure of the model is defined by the operating regimes and the structures of the local models. With the operating-regime-based modeling framework, it is possible to develop the model structure through a graphical representation of the operating regimes, rather than to restructure the model equations.

In addition to a certain structure, the model also consists of some model parameters. In the operating-regime-based modeling framework, these parameters are the parameters of the local models, so their scope is usually limited to a single local model. This simplification is useful, because the most complex equations one usually need to consider are the equations describing the local models.

2.4 Laboratory experiment

Consider the experimental setup illustrated in Fig. 2. The output of this system is the air temperature measurement $y(t)$, the input is the voltage over the resistor $u(t) \in [0, 10]$. In addition, the valve opening angle $v(t) \in [0°, 180°]$ is an independent measured variable that can be manipulated only by hand. On this system some experiments have been performed. First, consider the

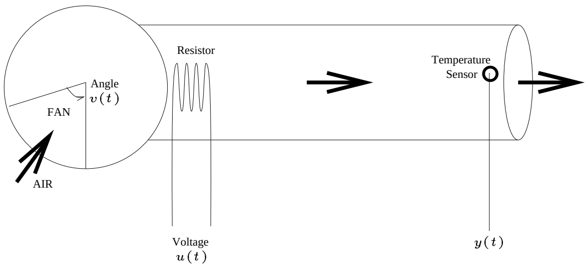

Fig. 2. Heat-transfer process: air is pulled into the 30 cm tube through a valve by a fan. The air is heated by a resistor, and its temperature is measured at the outlet.

responses to a 10 volt step input, for $v_1 = 20°, v_2 = 50°$, and $v_3 = 100°$ plotted in Fig. 3a. From these curves, two observations can be made:

(i) First, there seems to be two dominating time-constants, at least for small valve opening angles v. The fast mode (time constant about 1 second) is related to the heat capacity of the air in the tube, while the slower one (time constant equal to a few minutes) is related to the heat capacity of the tube itself and the rest of the equipment. In the modeling and identification experiments that follows, it will only be attempted to find a model with good prediction performance on the horizon of the faster of these modes, since this is the one that is of interest for control purposes.

(ii) Second, the steady-state gain seems to be a function of the valve opening angle v. On a shorter time-scale, cf. Fig 3b, it can be seen that the time-constant and time-delay are also functions of v. Similar experiments with fixed v but varying input step amplitude show that the steady-state gain depends on the input signal level, too, cf. the steady-state response in Fig. 4.

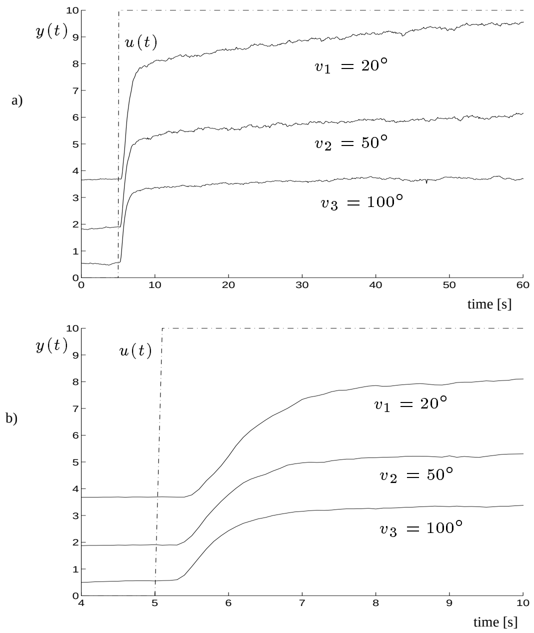

Fig. 3. a) Response to a 10 volt step input, and b) the same response on a shorter time-scale.

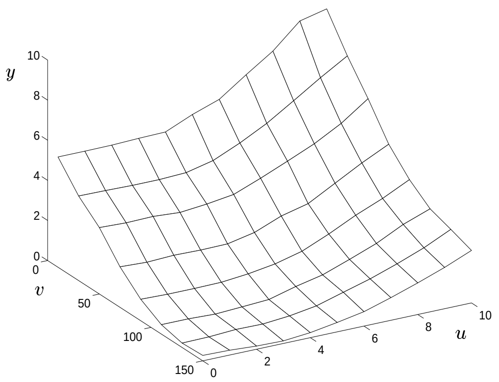

Fig. 4. Steady-state response.

For the purposes of identification, the data sequence in Fig 5 is used. The sampling interval is $\Delta t = 0.11$ s, and the sequence contains about 1300 samples. The input $u(t)$ is a random signal, exciting the dynamics locally about a sequence of random operating points that covers the full range of operation.

The valve opening $v(t)$ varies over the full range of operation in a pseudo-random manner. For the purpose of model validation, another data sequence with somewhat different excitation signals is used, see Fig. 6. The input varies in a qualitatively similar manner, while the valve opening angle varies more or less systematically to cover a wide range of frequencies and amplitude levels.

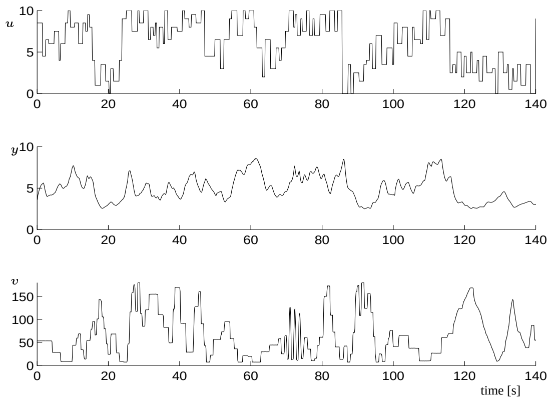

Fig. 5. Data sequence used for identification.

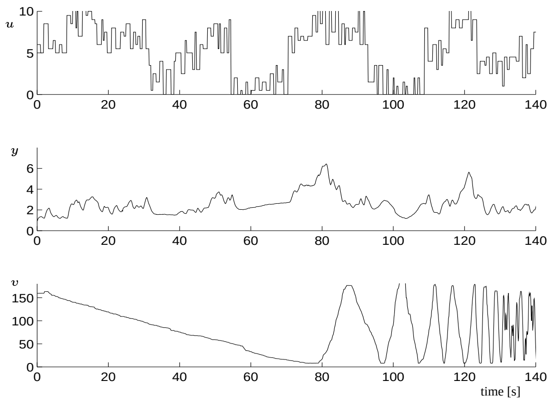

Fig. 6. Data sequence used for validation.

This laboratory experiment will be used throughout this paper in order to illustrate the ideas.

3 Overview of ORBIT

3.1 The ORBIT Environment

An overview of the ORBIT software environment can be seen in Fig. 7. ORBIT is implemented in MATLAB and fully documented in (Johansen 1997*c*). The ORBIT core contains the GUI, parameter- and structure-identification algorithms, model-validation algorithms, and model database management,

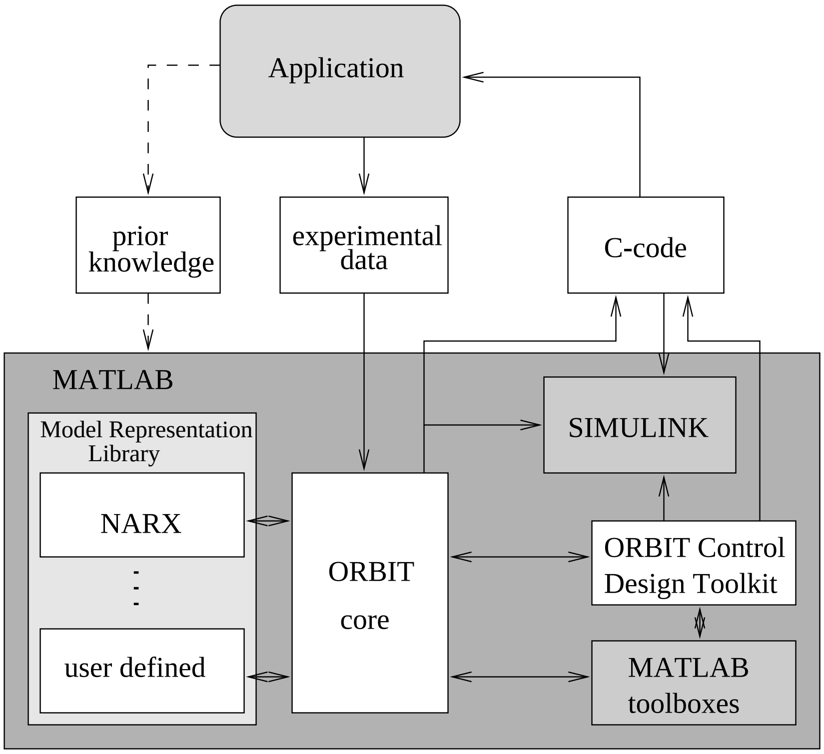

Fig. 7. The ORBIT Environment

as well as interfaces to various generic MATLAB tools and toolboxes. ORBIT can support a wide range of model representations. However, only the NARX representation (Johansen and Foss 1993) has currently been implemented as part of the standard model representation library. The advanced user is free to include customized or generic model representations in this library by programming the required MATLAB functions. The ORBIT Control Design toolkit (ORBITcd) (Johansen *et al.* 1998) supports the design of gain-scheduling-like non-linear controllers on the basis of ORBIT models. ORBIT models and controllers can be made available as SIMULINK S-functions and blocks for simulation. Using the built-in code-generation facility, models and controllers can be exported as C-functions for real-time application or fast simulation. Local models and data can also be interchanged with other MATLAB tools, including the MATLAB Control Toolbox, Signal Processing Toolbox, and LMI Toolbox. An application programmer's interface (API) allows other MATLAB programs to access the ORBIT model database. ORBIT is extendible, i.e. its core model representation and functions are documented. Experimental application data and prior knowledge form the basis of model development in ORBIT. These can be pre-processed and analyzed using generic MATLAB and SIMULINK functions before they are made use of in ORBIT.

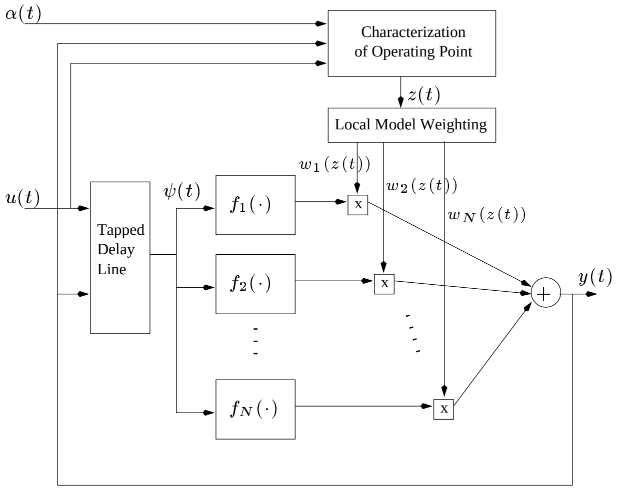

Fig. 8. The ORBIT NARX model representation.

3.2 The NARX Model Representation in ORBIT

Here, a useful model representation that is available in ORBIT will be described, namely the NARX model representation (Johansen and Foss 1993). Mathematically, this representation relates an input vector $u(t)$ to an output vector $y(t)$ by

$$
\begin{aligned}
y(t) &= \sum_{i=1}^{N} f_i(\psi(t);\theta_i)w_i(z(t)) + e(t) \\
\psi(t) &= (y(t-1), ..., y(t-n_y), u(t), ..., u(t-n_u))^T \\
z(t) &= H(y,u)(t)
\end{aligned}
$$

where $e(t)$ represents the unmodeled dynamics and noise. A block diagram for this model is provided in Fig. 8. The elements of this model representation are

- The functions $f_1, ..., f_N$ define a set of local models. These can be either of the most commonly applied linear type

$$
\begin{aligned}
f_i(\cdot) = a_{i,0} &+ A_{i,1}y(t-1) + ... + A_{i,n_y}y(t-n_y) \\
&+ B_{i,0}u(t) + ... + B_{i,n_u}u(t-n_u)
\end{aligned}
$$

  or user-programmed MATLAB functions implementing the mathematical function $f_i(\cdot)$.
- The local model parameters are lumped into a vector θ_i.
- The integer parameters n_y and n_u define the order of the NARX model.
- The positive definite weighting functions $w_1, ..., w_N$ define the relative weight of the local models at each operating point z. These functions will characterize the model's N operating regimes and satisfy

$$\sum_{i=1}^{N} w_i(z) = 1$$

for all z. The representation of these functions is described in detail in Section 3.3.
- The variables that characterize the operating regimes, z, are defined by a general non-linear operator H, which in its most general way can be implemented as a discrete-time SIMULINK S-function or block diagram. Hence, H can be built up from a mix of static and dynamic blocks.

In the laboratory heat-transfer experiment introduced in Section 2.4, the following local ARX model structure is chosen:

$$\begin{aligned} y(t) = a_0 + a_1 y(t-1) \\ b_{1,1}u(t-1) + \ldots + b_{1,4}u(t-4) \\ b_{2,1}v(t-1) + \ldots + b_{2,5}v(t-5) + e(t). \end{aligned} \tag{1}$$

3.3 Operating regimes and weighting functions

In ORBIT, the operating regimes and weighting functions are parameterized in terms of axis-parallel hyper-rectangles with soft edges. This representation is fairly general, since the user is free to select the operator H that defines the variables z that characterize the operating regimes.

The operating regimes can be viewed as fuzzy sets that are characterized by their membership functions $\mu_i(z)$. For example, a 2D axis-parallel rectangle with soft edges can be represented in terms of its projections onto the two axes, $\mu_i(z) = \mu_{i,1}(z_1)\mu_{i,2}(z_2)$. The weighting function are now defined as

$$w_i(z) = \frac{\mu_i(z)}{\sum_{j=1}^{N} \mu_j(z)}.$$

In the ORBIT GUI, it is the projections of the hyper-rectangles that are visualized and manipulated (except when z is 2-dimensional, when the rectangles can be explicitly visualized and manipulated).

From the discussion in Section 2.4 it is expected that the laboratory process will be nonlinear with respect to the two input variables $u(t)$ and $v(t)$. In other words, the parameters of the local model structure (1) should be functions of $u(t)$ and $v(t)$. Hence, the following characteristic variables $z(t) = (z_1(t), z_2(t)) = (u(t-2), v(t-4))$ are selected. A sample decomposition into three operating regimes (R4,R5 and R6) are shown in Figs. 9 and 10. Figure

9 visualizes the operating regimes as 2D-rectangles, while Figure 10 visualizes the same operating regimes in terms of their projections onto the two characteristic axes.

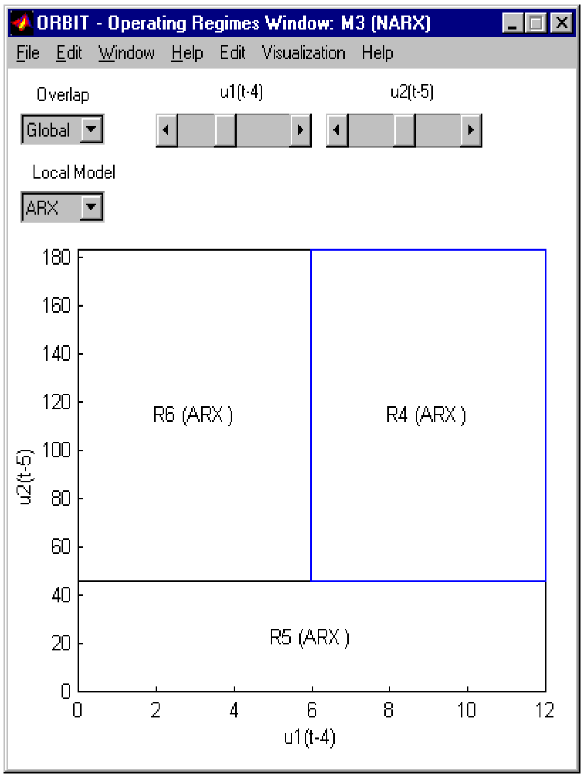

Fig. 9. Visualization of the three operating regimes as rectangles in 2D.

The Operating Regimes Window contains functions for the direct manipulation of operating regimes, including copy, move, resize, etc. Furthermore, the softness of the boundaries of the operating regimes can be specified individually, using the sliders in the Operating Regimes Window. Finally, there are functions for the selection and definition of variables that characterize the operating regimes, and their range.

4 ORBIT functionality

The functionality of ORBIT will be described with reference to the laboratory example. The different models found using ORBIT are summarized in the ORBIT Model Database Window, cf. Fig. 11, which contains five models:

- M1 - linear ARX models with structure (1).
- M2 - composite model consisting of four local linear ARX models.
- M3 - composite model consisting of three local linear ARX models.
- M4 - composite model with 28 local linear ARX models.

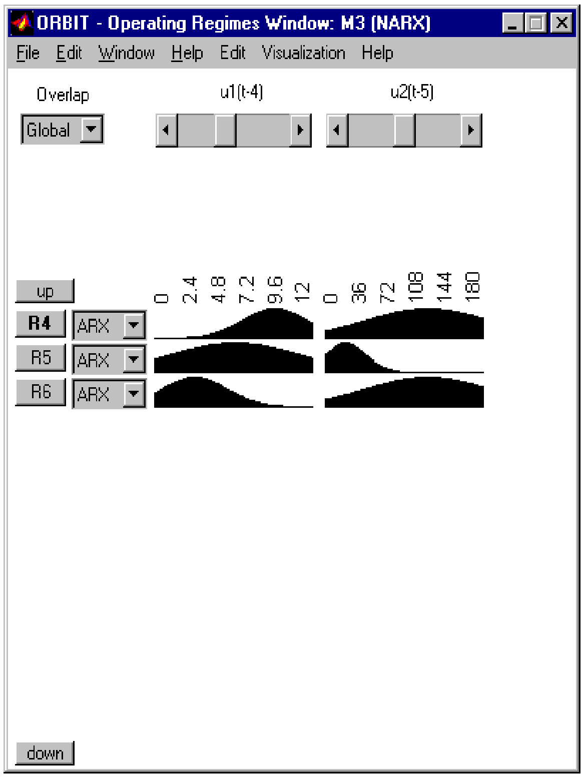

Fig. 10. Visualization of the three operating regimes as projections onto the characteristic dimensions.

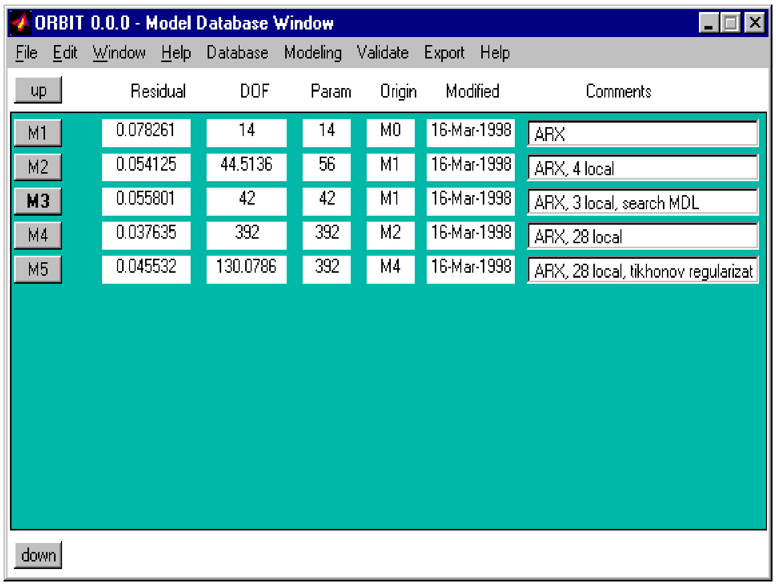

Fig. 11. The Model Database Window contains a list of all ORBIT models, and the functionality to manipulate them.

– M5 - model with the same structure as M4, but with different parameters.

The Model Database Window in Fig. 11 contains the functionality for saving, loading, copying and deleting models in the database.

4.1 Model-validation methods

Model validation is often viewed as a highly-application specific problem. This is recognized in ORBIT, and the models can be made available to external applications, or in the MATLAB/SIMULINK environment as S-functions or C-code. However, there are some commonly used validation methods that are supported by ORBIT. A database of models can be stored in ORBIT (see Fig. 11), and selected models can be compared by viewing simulation/prediction results (see Fig. 12). Statistical criteria such as the estimated prediction error can be visualized (see Fig. 13), and residuals and correlation functions can also be visualized and analyzed.

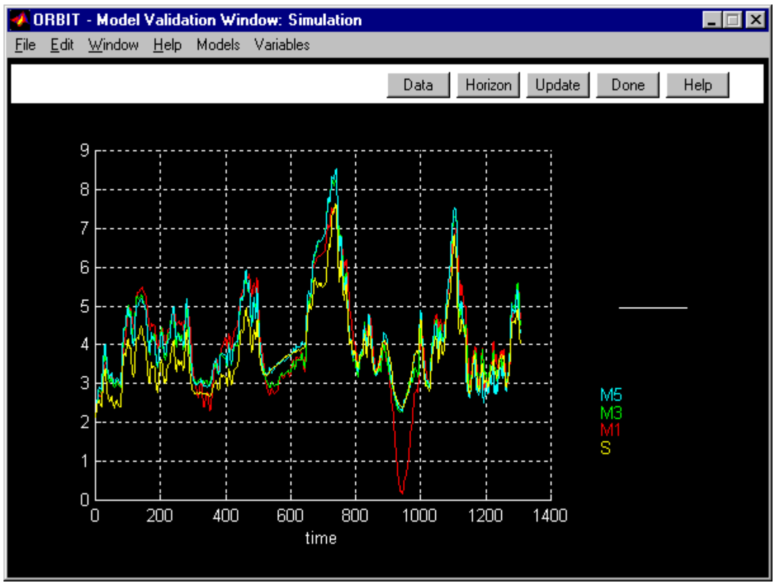

Fig. 12. The Model Validation Window allows the user to compare the prediction performance of the models M1, M3 and M5 models using simulation.

4.2 Parameter-identification methods

The parameter-identification problem consists of determining the local model parameters. In ORBIT it is defined as an extended optimization problem, see Fig. 14. The criterion is built up from the following components:

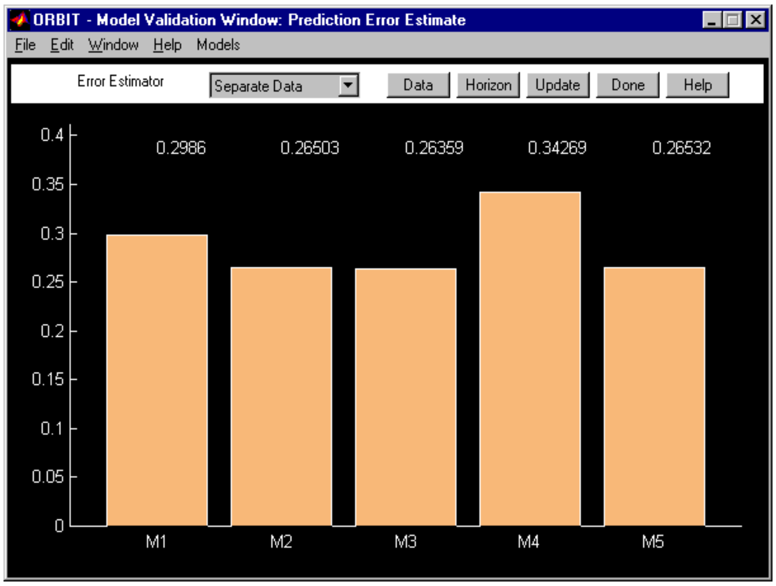

Fig. 13. The Model Validation Window allows the user to compare different models using statistical criteria. In this case the average 5-step-ahead prediction error on the validation data sequence is displayed.

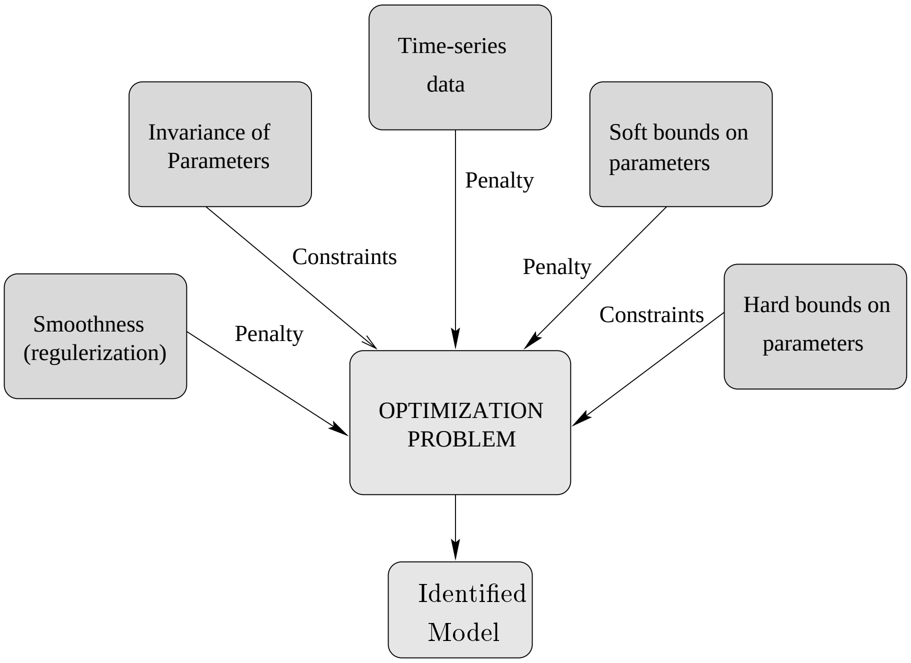

Fig. 14. The parameter identification is formulated as an extended optimization problem in ORBIT.

– Penalty on the mismatch between model prediction (possible multi-step prediction or simulation) and time-series data (prediction error). This defines the basic criterion found in many identification tools.
– A regularization penalty, which has the purpose of improving the robustness and removing any ill-conditioning of the identification problem due to an

over-parameterized or poorly identifiable model structure (Johansen 1997*b*). The following are implemented: SVD (singular value decomposition) based inversion of the Hessian, the ridge regression penalty, and an approximate Tikhonov stabilizer that penalizes non-smooth model behavior (parametric differences between neighboring local models) (Johansen 1996).

- Constraints or penalties due to prior knowledge about parameter values: a parameter can be totally unknown (no constraints or penalties) or completely known (equality constraint), or can have hard upper and lower bounds (inequality constraints) or soft upper and lower bounds (penalty function; can be viewed as a Bayesian identification method).
- The value of a parameter can be invariant, regardless of the operating point, i.e. the same in all local models. This is formulated as a set of equality constraints.

The mathematical formulation of the constraints and criterion are invisible to the user, but are built up internally from selections in the GUI, cf. Fig. 15. ORBIT contains three basic parameter-identification algorithms that can

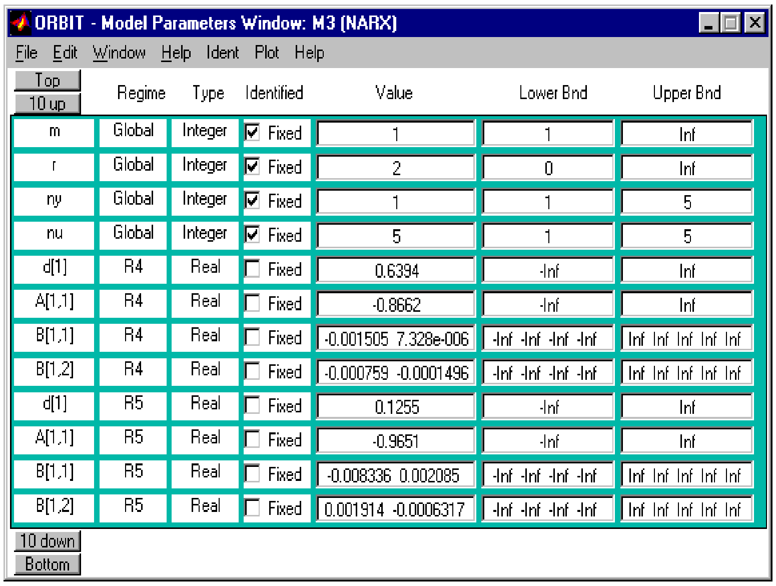

Fig. 15. The Model Parameters Window gives the user access to the model parameters and their properties.

be applied to solve the optimization problem(s) resulting from the data, prior knowledge and user selections:

- The **prediction error method** is the most general method, where the PE criterion

$$V_M(\theta) = \sum_{t=1}^{M} l(\varepsilon(t;\theta)) \tag{2}$$

is minimized using an SQP (sequential quadratic programming) algorithm. The prediction error is defined as

$$\varepsilon(t;\theta) = y(t) - \hat{y}(t|t-h;\theta)$$

where $\hat{y}(t|t-h;\theta)$ is the h-step-ahead predictor based on the global interpolated model, and θ are the model parameters. The PE criterion is defined in terms of a general user-specified penalty on the prediction error $l(\cdot)$ (not necessarily quadratic) and a user-specified prediction horizon h. Initial values for the SQP (sequential quadratic programming) algorithm can be specified, and the **constr** function from the MATLAB Optimization Toolbox is used to solve the resulting constrained non-quadratic optimization problem.
- **Least-squares method**. For the special case when the criterion function (2) is quadratic (requires the one-step-ahead predictor $h = 1$), the optimization problem becomes quadratic, and the constraints are always linear. The **qp** function from the MATLAB Optimization Toolbox is applied to solve the constrained quadratic programming problem.
- Rather than defining the identification problem in terms of a single global predictor $\hat{y}(t|t-h;\theta)$ that combines the local models, one can define a separate identification problem for each local model, based on local predictors $\hat{y}_i(t|t-h;\theta_i)$, where θ_i are the local model parameters for the local model with index i. Of course, only the data that are relevant to the current operating regime should be applied to identify the corresponding local model. This can be implemented as a **locally weighted least-squares method** when the criterion function is quadratic, i.e.

$$V_{M,i}(\theta_i) = \sum_{t=1}^{M} \left(y(t) - \hat{y}_i(t|t-1;\theta_i)\right)^2 w_i(z(t))$$

is minimized. This method has some interesting properties which are studied in detail in (Murray-Smith and Johansen 1997).

When applying regularization, ORBIT can compute the regularization parameter by optimizing the bias-variance trade-off (Johansen 1997*b*). The mean squared prediction error (MSE) can be estimated using a separate data sequence or methods such as the Final Prediction Error (FPE) or the Minimum Description Length (MDL). These well-known methods are extended to account for the existence of penalties and constraints in the criterion (Johansen 1997*a*).

The models M1, and M3 - M5 are found using the least squares algorithm. With the exception of model M5, they are all identified without regularization, but with some constraints to ensure that the first b-parameters are zero (due to the known time-delay). The model structure of M4 and M5 is strongly overparameterized. Hence, the model M4 identified using the pure least squares algorithm does not perform very well, see Figs. 12 and 13. On the other hand,

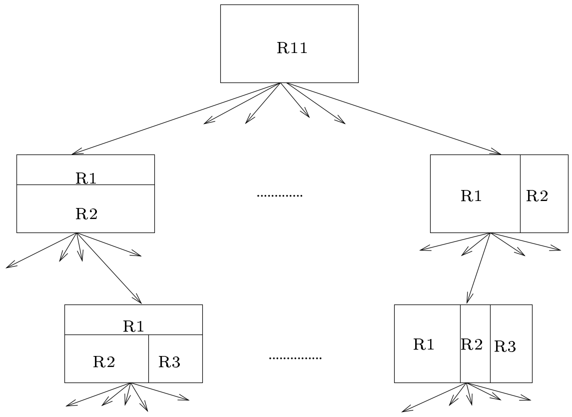

Fig. 16. The model structure tree resulting from successive operating regime decomposition.

the model M5 is identified using Tikhonov regularization, and thus has reduced the degrees of freedom from 392 to about 130. It can be seen from Figs. 12 and 13 that it performs well. The model M2 is identified using locally weighted least squares.

4.3 Structure-identification methods

The structural parameters of ORBIT models are

- The number of operating regimes, and their locations in the operating space.
- Any integer parameters in the local models, such as the order (n_u and n_y in the NARX representation).

ORBIT supports structural identification of these parameters on the basis of the optimization of statistical criteria based on separate validation data, FPE (Final Prediction Error) or MDL (Minimum Description Length) which all estimate the MSE (Mean Squared Prediction Error). The set of model structures defined by the possible operating regime decompositions is viewed as a tree, see Fig. 16. Notice that a finite number of candidate split points is selected along each dimension, so this leads to a finite number of model structures at each level in the tree. In addition, at each node in this tree there can be integer parameters related to the local models. ORBIT allows the user to interactively explore the model structure tree in Fig. 16. User-specified sub-trees can be searched to the desired depth, see Fig. 17. The number of candidate splits for each regime along each axis is determined by the "minimal discretization" parameter of this window. The "maximal discretization" parameter determines the maximum number of candidate splits over the full operating range. The heuristic local search method is described in detail in (Johansen and Foss 1995) and is based on (Sugeno and Kang 1988). Working interactively, the user may retain any promising models, and then validate and compare them using other methods, such as simulation and residual analysis. The user can also manipulate the operating regimes directly in the GUI or

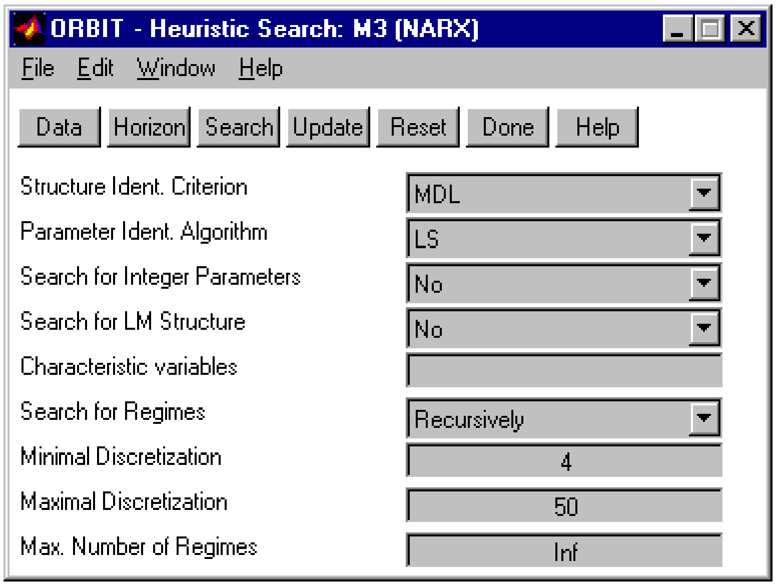

Fig. 17. The Structure Identification Window allows the user to search the model structure tree for promising model structures, using the heuristic search algorithm.

API, rather than relying on the search algorithm.

The model M3 is found using the heuristic search algorithm using pure least-squares parameter identification and the MDL criterion to find the structure. It contains three operating regimes, and it can be seen from the validation plots (Figs. 12 and 13) that is performs much better than the linear model.

4.4 Other functions

Other functions in ORBIT include the following aspects:

- There are interfaces to general-purpose MATLAB and SIMULINK tools.
- The parameters can be visualized as functions of the operating point.
- There is an Application Programmers Interface (API) which allows the user to access the ORBIT models and algorithms from the MATLAB command window without going through the GUI.
- The ORBIT Control Design toolkit (ORBITcd) (Johansen *et al.* 1998) supports the design of gain-scheduling-like non-linear controllers on the basis of ORBIT models.
- Local linear models can be exported from or imported into ORBIT in the LTI format of the MATLAB Control Systems Toolbox.

5 Concluding Remarks

ORBIT implements much of the current state of the art in operating-regime-based modeling and identification technology (including Takagi-Sugeno-Kang fuzzy models), in a flexible and integrated environment. ORBIT is being actively used in industrial research projects, e.g. (Hunt *et al.* 1996, Johansen *et al.* 1998), and in basic research and education on modeling, identification and control methods based on operating regimes.

References

Banerjee, A., Y. Arkun, R. Pearson and B. Ogunnaike (1995). H_∞ control of nonlinear processes using multiple linear models. In: *Proc. European Control Conference, Rome.* pp. 2671–2676.

Chow, C.-M., A. G. Kuznetsov and D. W. Clarke (1995). Using multiple models in predictive control. In: *Proc. European Control Conference, Rome.* pp. 1732–1737.

Foss, B. A., T. A. Johansen and Aa. V. Sørensen (1995). Nonlinear predictive control using local models - applied to a batch fermentation process. *Control Engineering Practice* **3**, 389–396.

Hunt, K. J. and T. A. Johansen (1997). Design and analysis of gain-scheduled local controller networks. *Int. J. Control* **66**, 619–651.

Hunt, K. J., J. C. Kalkkuhl, H. Fritz and T. A. Johansen (1996). Constructive empirical modelling of longitudinal vehicle dynamics with local model networks. *Control Engineering Practice* **4**, 167–178.

Jacobs, R. A., M. I. Jordan, S. J. Nowlan and G. E. Hinton (1991). Adaptive mixtures of local experts. *Neural Computation* **3**, 79–87.

Johansen, T. A. (1996). Robust identification of Takagi-Sugeno-Kang fuzzy models using regularization. In: *Proc. IEEE Conf. Fuzzy Systems, New Orleans.* pp. 180–186.

Johansen, T. A. (1997*a*). Constrained and regularized system identification. In: *Preprints IFAC Symposium on System Identification, Kitakyushu, Japan.* pp. 1467–1472.

Johansen, T. A. (1997*b*). On Tikhonov regularization, bias and variance in nonlinear system identification. *Automatica* **33**, 441–446.

Johansen, T. A. (1997*c*). ORBIT control design toolbox V1.10 – User's guide and reference. Technical Report STF72 F97312. SINTEF, Trondheim, Norway.

Johansen, T. A. and B. A. Foss (1993). Constructing NARMAX models using ARMAX models. *Int. J. Control* **58**, 1125–1153.

Johansen, T. A. and B. A. Foss (1995). Identification of non-linear system structure and parameters using regime decomposition. *Automatica* **31**, 321–326.

Johansen, T. A. and B. A. Foss (1997). Operating regime based process modeling and identification. *Computers and Chemical Engineering* **21**, 159–176.

Johansen, T. A., K. J. Hunt and H. Fritz (1998). A software environment for gain scheduled local controller network design. *IEEE Control Systems Magazine* **18**(2), 48–60.

Murray-Smith, R. and Johansen, T. A., Eds.) (1997). *Multiple Model Approaches to Modelling and Control.* Taylor and Francis, London.

Sugeno, M. and G. T. Kang (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems* **26**, 15–33.

Takagi, T. and M. Sugeno (1985). Fuzzy identification of systems and its application to modeling and control. *IEEE Trans. Systems, Man, and Cybernetics* **15**, 116–132.

Zhao, J., V. Wertz and R. Gorez (1995). Linear TS fuzzy model based robust stabilizing controller design. In: *IEEE Conf. Decision and Control, New Orleans.* pp. 255–260.