ORBIT - OPERATING REGIME BASED MODELING AND IDENTIFICATION TOOLKIT

Tor A. Johansen * **Bjarne A. Foss** **

* *SINTEF Electronics and Cybernetics, Automatic Control Department, N-7034 Trondheim, Norway. Email: Tor.A.Johansen@ecy.sintef.no*
** *Department of Engineering Cybernetics, The Norwegian University of Science and Technology, N-7034 Trondheim, Norway. Email: Bjarne.A.Foss@itk.ntnu.no*

Abstract: 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 being smoothly patched together into a global non-linear model. ORBIT is a computer-aided modeling environment that supports interactive development of regime based models on the basis of a mixture of empirical data and prior knowledge. 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 gives good fit to the data. Model validation, visualization and application is also supported.

Keywords: Software Tools, System Identification, Nonlinear Systems.

1. INTRODUCTION

There is a growing interest in modeling and identification methods based on explicit decomposition of the operating range of dynamical 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 dynamical 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 of the model. The blending of empiricism with an understanding of the mechanisms of the system is of fundamental importance (Johansen and Foss 1997). ORBIT is an interactive environment for computer aided modeling and system identification, typically leading to mathematical models that can be described as grey-box models. Depending on the application and the available data and knowledge, ORBIT can be applied as anything between an automatic data-driven modeling program and a graphical user interface (GUI) for manual specification of the model. Advanced methods for nonlinear system identification is the core of ORBIT.

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 *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 less 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.

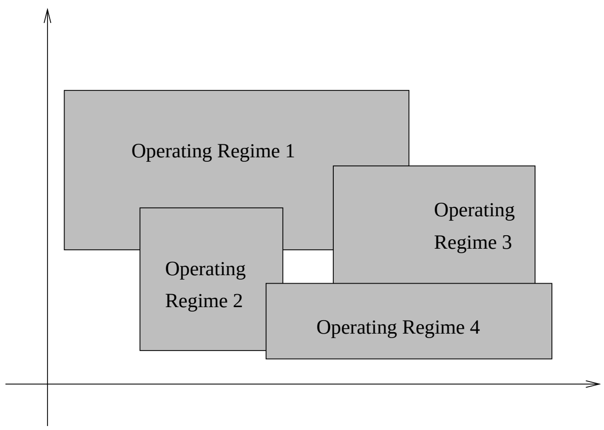

Fig. 1. Decomposition of operating range into regimes

The operating regime based modeling framework can be conceptually illustrated as in Figure 1. The system's full range of operation 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 combined 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.

One motivation behind this framework is that global modeling is complicated because one will 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 (Johansen and Foss 1997, Murray-Smith and Johansen 1997).

2.2 *Flexible Modeling*

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

Modeling involves three major tasks:

(1) Select a decomposition into operating regimes. For this purpose one can use 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.
(2) Select the structure of the local models. Again, one can use a combination of qualitative understanding and empirical data. The local model structures can either be chosen as empirical model structures, or you can develop mechanistic local model structures.
(3) 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 it in other operating regimes may be described by a mechanistic state-space model. In other words, the framework is flexible with respect to the type of process knowledge that can be applied.

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

3. ORBIT OVERVIEW

3.1 *The ORBIT Environment*

An overview of the ORBIT software environment can be seen in figure 2. ORBIT is implemented in MATLAB. Current versions run under MATLAB

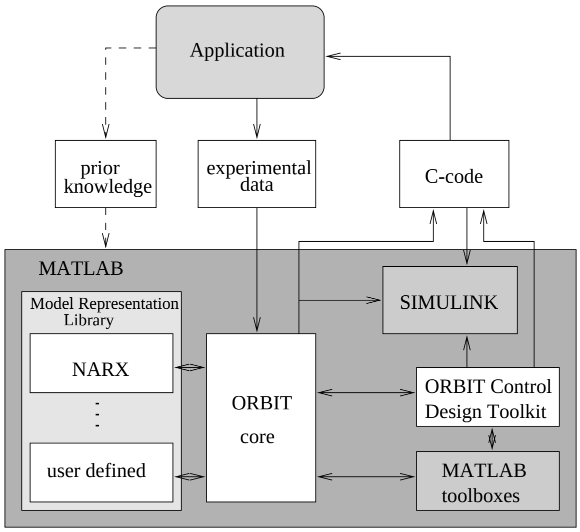

Fig. 2. The ORBIT Environment

4.2 on UNIX and PC platforms. The ORBIT core contains the GUI, parameter and structure identification algorithms and model validation algorithms, model database management 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) is currently implemented as part of the standard model representation library. The advanced user is free to include customised or generic model representations in this library by programming the required MATLAB functions. The ORBIT Control Design toolkit (ORBITcd) (Johansen *et al.* 1997) supports 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 model parameters can also be interchanged with other MATLAB tools, including the MATLAB Control Toolbox, Signal Processing Toolbox, and LMI Toolbox. An application programmers 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 preprocessed and analysed using generic MATLAB and SIMULINK functions before they are made use of in ORBIT.

3.2 *The NARX Model Representation in ORBIT*

Here we will describe a useful model representation that is available in ORBIT, namely the NARX model representation (Johansen and Foss

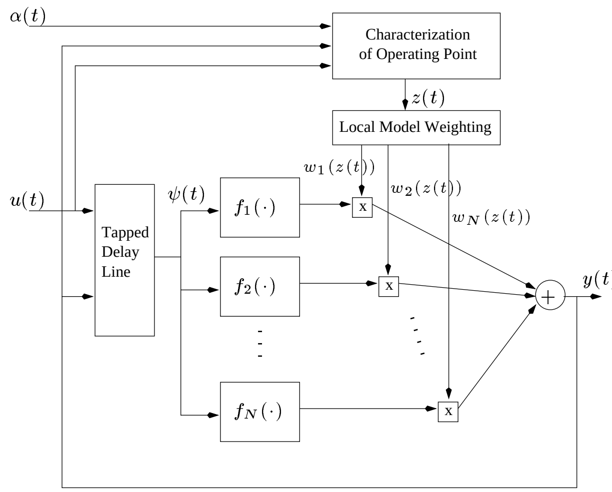

Fig. 3. The ORBIT NARX model representation.

1993). Mathematically, this representation relates an input vector $u(t)$ to an output vector $y(t)$ by

$$y(t) = \sum_{i=1}^{N} f_i(\psi(t);\theta_i)w_i(z(t)) + e(t)$$

$$\psi(t) = (y(t-1), ..., y(t-ny), u(t), ..., u(t-nu))^T$$

$$z(t) = H(y,u)(t)$$

where $e(t)$ represents the unmodeled dynamics and noise. A block diagram for this model is provided in Figure 3. The elements of this model representation are

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

  $$f_i(\cdot) = a_{i,0} + A_{i,1}y(t-1) + ... + A_{i,ny}y(t-ny) + B_{i,0}u(t) + ... + B_{i,nu}u(t-nu)$$

  or user-programmed MATLAB functions implementing $f_i(\cdot)$.
- The local model parameters are lumped into a vector θ_i.
- The integer parameters ny and nu define the order of the NARX model.
- The positive definite weighting functions $w_1, ..., w_N$ will define the relative weight of the local models at each operating point z. In practise, 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 function is described in detail in section 3.3.
- The variables that characterize the operating regimes, z, are defined by a general nonlinear operator H, which in its most general way can be implemented as a SIMULINK S-function or block diagram. Hence, H can be

built from a mix of static, discrete-event, and discrete-time blocks.

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 variables z that characterizes the operating regimes.

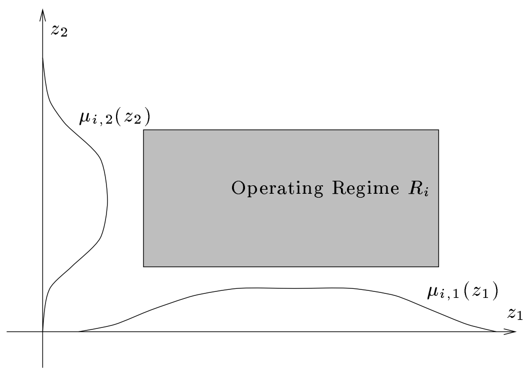

Fig. 4. The regimes are represented as hyper-rectangles with soft boundaries.

The operating regimes can be viewed as fuzzy sets which 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, $\mu_i(z) = \mu_{i,1}(z_1)\mu_{i,2}(z_2)$, cf. Figure 4. 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).

3.4 *Computer Aided Grey-Box Modeling*

ORBIT is based on the "computer-aided modeling" philosophy, supporting interactive development of models by step-wise structure and parameter identification algorithms, model validation tools as well as a GUI for manual model manipulation and entry points for structured knowledge. ORBIT supports modeling in the gap between the extremes of automatic data-driven modeling and completely manual model development.

4. ORBIT FUNCTIONALITY

The main modules of ORBIT, seen as different windows in the GUI, are illustrated in Figure 5 and will be described in more detail below.

4.1 *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 (Johansen 1997*a*). The criterion is built up from the following components:

- Penalty on mismatch between model prediction (possible multi-step ahead 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 remove any ill-conditioning of the identification problem due to an over-parameterized or poorly identifiable model structure (Johansen 1997*b*). We have implemented SVD 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), completely known (equality constraint), 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 of the operating point, i.e. the same in all local models. This is formulated as a set of equality constraints.

The above mentioned components that form the criterion are invisible to the user, but built up automatically from selections in the GUI. ORBIT contains three basic parameter identification algorithms that can 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. The PE criterion is defined in terms of a general user-specified criterion function (not necessarily quadratic), user-specified prediction horizon, constraints and penalties as described above, 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 is quadratic (requires one-step-ahead predictor), 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.

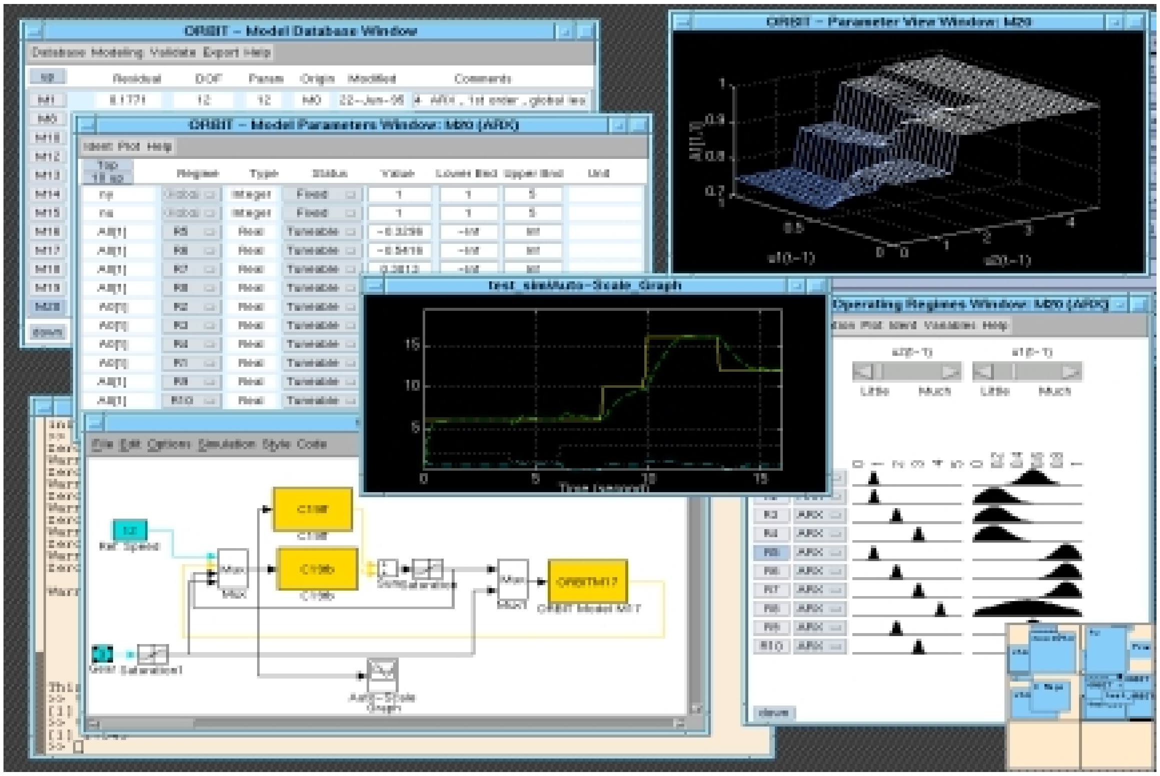

Fig. 5. Modeling with ORBIT.

- Rather than defining the identification problem in terms of a single global predictor that combines the local models, one can define a separate identification problem for each local model. Only the data that are relevant in the current operating regime are applied to identify the corresponding local model. This can be implemented as a **locally weighted least squares method** when the criterion function is quadratic. This method has some interesting properties that 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) and 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*).

4.2 *Structure Identification Methods*

The structural parameters of ORBIT models are

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

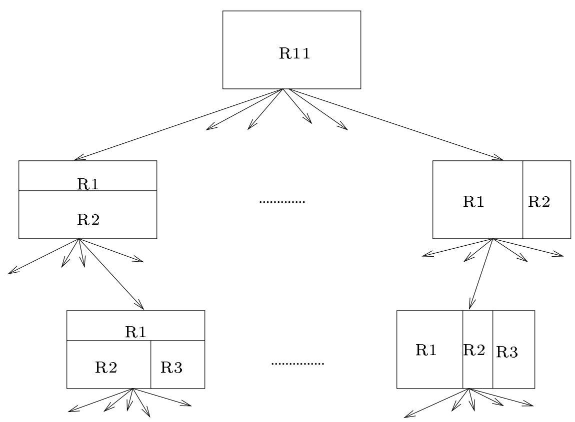

Fig. 6. The model structure tree resulting from successive operating regime decomposition.

ORBIT supports structural identification of these parameters on the basis of optimization of statistical criteria based on separate validation data, FPE or MDL that all estimate the MSE. The set of model structures defined by the possible operating regime decompositions is viewed as a tree, Figure 6. 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 Figure 6. User-specified sub-trees can be searched to the desired depth. The heuristic search method is described in detail in (Johansen and Foss 1995) and is based on (Sugeno and Kang 1988). Working interactively, the user may keep promising models and 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 API.

4.3 *Model Validation Methods*

Model validation is often viewed as a highly application specific problem. This is recognized in ORBIT, and the model can be made available to external applications and 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, and selected models can be compared by viewing simulation/prediction results. Statistical criteria such as FPE and MDL can be visualized, and residuals and correlation functions can also be visualized and analyzed.

4.4 *Other functions*

Other functions in ORBIT include

- Functions for direct manipulation of operating regimes, including copy, move, resize etc.
- The softness of the boundaries of the operating regimes can be specified individually.
- Selection and definition of variables that characterizes the operating regimes, and their range.
- There are interfaces to general-purpose MATLAB and SIMULINK tools.
- Database management for storing and recalling models.
- The parameters can be visualized as functions of the operating point.

The ORBIT Control Design toolkit (ORBITcd) (Johansen *et al.* 1997) supports design of gain-scheduling-like non-linear controllers on the basis of ORBIT models.

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), basic research and education on modeling, identification and control methods based on operating regimes.

6. 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. To be presented at the IFAC Symposium on System Identification, Kitakyushu, Japan.

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

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 (1997). A software environment for gain scheduled local controller network design. Submitted for publication.

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.