INTERVAL SIMULATION OF NARMAX MODELS BASED ON COMPUTER ARITHMETIC

Priscila F. S. Guedes*, Márcia L. C. Peixoto*, Otávio A. R. O. Freitas*, Alípio M. Barbosa†, Samir A. M. Martins*, Erivelton G. Nepomuceno*

*GCOM - Grupo de Controle e Modelagem
UFBJ - Universidade Federal de São João del-Rei
Pça. Frei Orlando, 170 - Centro - 36307-352 - São João del-Rei, MG, Brasil

†Centro Universitário Newton Paiva
Rua José Cláudio Rezende, 420 - Estoril - 30494-230 - Belo Horizonte, MG, Brasil

Emails: pri12_guedes@hotmail.com, marciapeixoto93@hotmail.com, otavio_augusto_freitas@hotmail.com, alipiomonteiro@yahoo.com.br, martins@ufsj.edu.br, nepomuceno@ufsj.edu.br

Abstract—System identification is an important area of science, which aims to describe the characteristics of the system, representing them by mathematical models. Since many of these models can be seen as recursive functions, it is extremely important to control the errors in these functions, because small errors introduced in each computational step can grow exponentially due to the sensitivity to initial conditions present in this type of functions. One of the ways to control rounding and truncation errors is through interval arithmetic, since it is not possible to represent all numbers in the computer because of the finite representation in them. Thus, in arithmetic interval a number is represented by an interval in which the true number is within that interval. In this manuscript we developed an algorithm that performs the operations of interval arithmetic using basic functions. We have compared compared our results with the Matlab-toolbox Intlab. Numerical experiments have shown that our method is superior producing narrower intervals.

Keywords—Dynamical Systems, Error Propagation, Interval analysis, Intlab toolbox.

1 Introduction

System identification plays a fundamental role in the estimation of mathematical models from data (Ljung, 1998). The estimated models can then be used to analyse input-output relationships, to simulate the system in various situations and for controller design (Barbosa et al., 2015).

Some models can be seen as recursive functions in particular non-linear systems (Ferreira et al., 2006). Recursive functions are widely used to solve problems and systems, as these functions provide a description for a variety of problems (Feigenbaum, 1978). The computational simulations of these systems are subject to errors. To illustrate the error propagation, let us evaluate two natural interval extensions for the Sine Map, identified by the NARMAX model, the first is given by \( f(x_n) = 2.6868x_n - 0.2462x_n^3 \), and the second by its natural extension \( g(x_n) = 2.6868x_n - (0.2462x_n)x_n^2 \), with \( x_0 = 0.1 \). The result obtained is shown in Figure 1.

Figure 1: Simulation of Sine Map, with \( x_0 = 0.1 \) for two different natural extensions, (-x-) for \( f(x_n) \) and (-o-) for \( g(x_n) \).

Clearly, the two trajectories are completely different after 50 iterations. As can be seen by
means of Figure 1, the propagation of rounding errors may lead to wrong results, even for a very simple system. A good analysis of the errors arising in numerical calculations can contribute to identify the reliability of calculated results, avoid error risks and improve the accuracy of calculations (Nepomuceno, 2014; Qun, 2012).

The use of interval arithmetic (Moore, 1979) has been considered as an efficient method to deal with numerical errors. The idea is that instead of using a single floating-point value to represent a number, which would result in an error if the number is not representative on the machine, the value is exposed by lower and upper bounds, which define a representable range in the machine (IEEE, 2015). In general, little attention has been given to the propagation of error in the area of system identification. Some works related to this subject was (Nepomuceno and Martins, 2016; Guedes et al., 2017). However, in these works the error is investigated when using different extensions. Nepomuceno and Martins (2016) developed the lower bound error, a practical tool capable of increasing the reliability of the computational simulation of dynamic systems; this lower bound error is specified on basis of two natural extensions. Guedes et al. (2017) refined the lower bound error, which case it is determined for an arbitrary number of natural extensions. Peixoto et al. (2017) proposed a system identification process focusing on estimation of NARMAX models parameters using the contribution of interval arithmetic with the Intlab toolbox.

The main purpose of this paper is to present an interval bounds using rounding downwards and upwards, considering the rules of interval arithmetic and basic functions in Matlab. The method has been applied to Logistic Map (May, 1976), the NAR model of the Sine Map (Nepomuceno et al., 2003), and the ARX model of flexible transmission system (Landau et al., 1995). To validate the proposed method, the results obtained were compared with the Intlab toolbox.

The rest of the paper is organized as follows. In Section 2, we recall some preliminary concepts recursive functions and interval arithmetic. Then, in Section 3, we present the developed method. Section 4 is devoted to present the results, then the final remarks are given in Section 5.

2 Preliminary concepts

2.1 NARMAX polynomial

The NARMAX (Non-linear AutoRegressive Moving Average model with eXogenous inputs) model may describe nonlinear systems using difference equations, relating output with linear and nonlinear combinations of past inputs and outputs and can be written as (Chen and Billings, 1989)

\[
y(k) = F^l[y(k-1), \ldots, y(k-n_y), u(k-1), \ldots, u(k-n_u), e(k-1), \ldots, e(k-n_e)] + e(k),
\]

where \( y(k), u(k) \) and \( e(k) \) are, respectively, the output, the input and the noise terms at the discrete time \( k \in \mathbb{N} \). The parameters \( n_y, n_u \) and \( n_e \) are their maximum lag. And \( F^l \) is assumed to be a polynomial with nonlinearity degree \( \ell \).

2.2 Recursive functions

In recursive functions is possible to calculate the state \( x_{n+1} \), at a given time, from an earlier state \( x_n \)

\[
x_{n+1} = f(x_n),
\]

where \( f \) is a recursive function and \( x_n \) is a function state at the discrete time \( n \). Given an initial condition \( x_0 \), successive applications of the function \( f \) it is possible to know the sequence \( \{x_n\} \). The initial condition \( x_0 \) is called the orbit of \( x_0 \) (Gilmore and Lefranc, 2012).

2.3 Interval Arithmetic

Moore (1979) proposed the concept of interval arithmetic, based on the extension of the concept of real numbers to a range of real numbers. The aim of Moore was to develop a set of techniques capable of producing reliable results, considering rounding errors in the numerical calculation.

An interval \( X \) is denoted as \([X, X]\), i.e. \( X = \{x : X \leq x \leq X\} \), where \( X \) and \( X \), respectively, the lower and upper limit of the interval \( X \). In a degenerated interval, we have \( X = X \) and such an interval amounts to a real number \( x = X = X \).

For a given interval \( X = [X, X] \), its width is defined by \( \omega(X) = (X - X) \) and its centre is \( m(X) = \frac{1}{2}(X + X) \) (Rothwell and Cloud, 2012).

Interval arithmetic provides a method for applying the elementary operations of conventional arithmetic so that the result of the interval includes all possible results. Given \( X = [X, X] \) and \( Y = [Y, Y] \), the basic interval operations are defined by:

\[
X + Y = [X + Y, X + Y],
\]

\[
X - Y = [X - Y, X - Y],
\]

\[
X \cdot Y = [\min(S), \max(S)],
\]

where \( S = \{XY, X\bar{Y}, \bar{X}Y, \bar{X}\bar{Y}\} \). If 0 does not belong to \( Y \), then \( X/Y \) is given by

\[
X/Y = X \cdot (1/Y)
\]

and \( 1/Y = [1/Y, 1/Y] \).
Intlab is a toolbox for Matlab that supports real and complex intervals, vectors and matrices. The toolbox was developed so that the computer arithmetic satisfies the IEEE 754 arithmetic standard (IEEE, 1985) and, that a permanent switch of the rounding mode is possible. Arithmetical operations in Intlab are accurately ascertained to be correct, comprising input and output and standard functions. By that, it is feasible to supersede every operation of a standard numerical algorithm by the correlating interval operations (Rump, 1999).

3 Methods

The proposed method is based on the rules of interval arithmetic, so for each operation these rules are obeyed to obtain a reliable result with the interval arithmetic. In order to achieve the results using intervals, in addition to the arithmetical operations, each operation was performed rounding, so that achieve the lower and upper limits. To illustrate how the simulations were developed, let us consider two intervals. To illustrate how the simulations were developed, let us consider two intervals X = [0.1,0.3] and Y = [0.3,0.35]. Additionally, we perform addition and subtraction operations using the proposed method and the Intlab toolbox.

\[
X_{inf}=0.1; \\
Y_{inf}=0.3; \\
system\_dependent('setround',-Inf) \\
X_{sup}=0.3; \\
Y_{sup}=0.35; \\
system\_dependent('setround',Inf) \\
X_{inf}=0.4; \\
Y_{inf}=0.25; \\
system\_dependent('setround',Inf) \\
X_{sup}=0.6; \\
Y_{sup}=0.55; \\
system\_dependent('setround',Inf) \\
\]

4 Numerical experiments

In this section, we present the numerical experiments obtained by the proposed method and the Intlab toolbox. We select three cases studies, which chosen maps are for the systems Logistic Map (May, 1976), Sine Map (Nepomuceno et al., 2003) and Flexible Transmission Benchmark (Landau et al., 1995).

4.1 Logistic Map

The logistic map was described by May (1976) as:

\[
x_{n+1} = rx_n(1-x_n),
\]

where \( r \) is the control parameter, which belongs to the interval \( 1 \leq r \leq 4 \) and \( x_n \) to the interval \( 0 \leq x_n \leq 1 \). Algorithm 1 explains the simulations performed by the proposed method.

Algorithm 1: Pseudo-code of the Proposed Method for the Logistic map.

1: input Number of iterations (\( N \)), parameters and initial conditions
2: system\_dependent('setround',-Inf)
3: \( \hat{x}_{0,n} \leftarrow \hat{x}_{0,n} \)
4: \( r^- \leftarrow r \)
5: system\_dependent('setround',Inf)
6: \( \hat{x}_{0,n} \leftarrow \hat{x}_{0,n} \)
7: \( r^+ \leftarrow r \)
8: for \( n=1:N \) do
9: system\_dependent('setround',-Inf)
10: aux1n \leftarrow (1 - \hat{x}_n^+) \\
11: aux2n \leftarrow (r^- \times \hat{x}_n^-) \\
12: \hat{x}_{n+1}^- \leftarrow aux1n \times aux2n \\
13: auxn \leftarrow \hat{x}_{n+1}^- \\
14: system\_dependent('setround',Inf) \\
15: aux3n \leftarrow (1 - \hat{x}_n^-) \\
16: aux4n \leftarrow (r^+ \times \hat{x}_n^+) \\
17: \hat{x}_{n+1}^+ \leftarrow aux3n \times aux4n \\
18: system\_dependent('setround',0.5) \\
19: if \( \hat{x}_{n+1}^- > 0.5 \) then
20: system\_dependent('setround',-Inf) \\
21: \hat{x}_{n+1}^- \leftarrow \hat{x}_{n+1}^+ \\
22: system\_dependent('setround',Inf) \\
23: \hat{x}_{n+1}^+ \leftarrow auxn+1 \\
24: end if
25: end for
26: output \( \hat{x}_{n+1}^- \) and \( \hat{x}_{n+1}^+ \)

Table 1 shows the values of a simulation performed in Matlab. This table displays the width and midpoint of the interval, and for comparison shows the results made in the Intlab toolbox.

4.2 Sine Map

A unidimensional sine map is defined as

\[
x_{n+1} = \alpha \sin(x_n),
\]

where \( \alpha = 1.2 \pi \). A polynomial NAR identified for this system is given by (Nepomuceno et al., 2003)

\[
y_{n+1} = 2.6868y_n - 0.2462y_n^3.
\]

The algorithm 2 explains the simulations performed by the proposed method and the Table 2 shows the values of a simulation performed in Matlab.
Table 1: Comparison of width size for the simulation of logistic map. The reference is the width produced by means of Intlab (Rump, 1999). We have studied four cases, which the control parameter is $r = 3.99$ and initial conditions are as follow: (1): $x_0 = 0.2$, (2): $x_0 = 0.4$, (3): $x_0 = 0.6$ and (4): $x_0 = 0.8$.

| Case | n   | Proposed Method width | midpoint | Intlab width | midpoint |
|------|-----|------------------------|----------|--------------|----------|
| (1)  | 1   | 0                      | 0.2      | 0            | 0.2      |
|      | 5   | 9.770e-15              | 0.821645072876575 | 2.0095e-14 | 0.821645072876575 |
|      | 10  | 9.8366e-12             | 0.973482128268848 | 2.0389e-11 | 0.973482128268850 |
|      | 20  | 1.0059e-05             | 0.013337715656825 | 2.0851e-05 | 0.013337715656825 |
| (2)  | 1   | 0                      | 0.4      | 0            | 0.4      |
|      | 5   | 1.3212e-14             | 0.990570357273853 | 1.5876e-14 | 0.990570357273853 |
|      | 10  | 1.3349e-11             | 0.111714690634153 | 1.6068e-11 | 0.111714690634153 |
|      | 20  | 1.3652e-05             | 0.751597796573294 | 1.6432e-05 | 0.751597796573294 |
| (3)  | 1   | 0                      | 0.6      | 0            | 0.6      |
|      | 5   | 1.2768e-14             | 0.990570357273852 | 1.6320e-14 | 0.990570357273852 |
|      | 10  | 1.2898e-11             | 0.011714690634148 | 1.6517e-11 | 0.011714690634148 |
|      | 20  | 1.3190e-05             | 0.75159779656068  | 1.6892e-05 | 0.75159779656068  |
| (4)  | 1   | 0                      | 0.8      | 0            | 0.8      |
|      | 5   | 9.5479e-15             | 0.821645072876574 | 2.0206e-14 | 0.821645072876574 |
|      | 10  | 9.6391e-12             | 0.973482128268850 | 2.0502e-11 | 0.973482128268850 |
|      | 20  | 9.8575e-06             | 0.013337715653912 | 2.0967e-05 | 0.013337715653912 |

Table 2: Comparison of width size for the simulation of sine map. The reference is the width produced by means of Intlab (Rump, 1999). We have studied four cases, which initial conditions are as follow: (1): $x_0 = 0.1$, (2): $x_0 = 0.2$, (3): $x_0 = 0.5$, and (4): $x_0 = 0.8$.

| Case | n   | Proposed Method width | midpoint | Intlab width | midpoint |
|------|-----|------------------------|----------|--------------|----------|
| (1)  | 1   | 0                      | 0.1      | 0            | 0.1      |
|      | 5   | 1.2879e-14             | 3.408933569627769 | 1.3323e-14 | 3.408933569627769 |
|      | 10  | 5.5898e-11             | -0.847910701541987 | 5.9273e-11 | -0.847910701541987 |
|      | 20  | 1.3298e-04             | 2.811807282224221 | 1.4101e-04 | 2.811807282224221 |
| (2)  | 1   | 0                      | 0.2      | 0            | 0.2      |
|      | 5   | 4.9738e-14             | 1.052283645351666 | 5.7732e-14 | 1.052283645351666 |
|      | 10  | 7.3143e-10             | -0.135225347638312 | 8.5036e-10 | -0.135225347638312 |
|      | 20  | 1.5060e-02             | 0.407737577437317 | 1.7509e-02 | 0.407737577437317 |
| (3)  | 1   | 0                      | 0.5      | 0            | 0.5      |
|      | 5   | 9.5035e-14             | 3.229051816564168 | 1.0347e-13 | 3.229051816564168 |
|      | 10  | 6.1926e-10             | 1.811036103169525 | 6.7462e-10 | 1.811036103169525 |
|      | 20  | 9.7948e-03             | 3.162513358524606 | 1.0670e-02 | 3.162513358524606 |
| (4)  | 1   | 0                      | 0.8      | 0            | 0.8      |
|      | 5   | 2.2116e-13             | -1.371443155591735 | 2.6712e-13 | -1.371443155591735 |
|      | 10  | 1.6245e-09             | -3.378411778526505 | 1.9624e-09 | -3.378411778526505 |
|      | 20  | 2.6918e-01             | 0.207611416301658 | 3.2517e-01 | 0.207453930188374 |

### 4.3 Flexible Transmission Benchmark

The flexible transmission system of Landau et al. (1995) is presented in Figure 2. The system input is the reference for the axis position of the first pulley and the system output is the axis position of the third pulley measured by a position sensor, sampled and digitised. The following transfer function is considered as the plant model

$$ H(q^{-1}) = \frac{q^{-d}B(q^{-1})}{A(q^{-1})}. $$

We consider the following ARX model identified and validated through the real plant (Landau et al., 1995; Bombois et al., 2006):

$$ y(k) = 1.41833y(k-1) - 1.58939y(k-2) + 1.31608y(k-3) - 0.88642y(k-4) + 0.28261u(k-3) + 0.50666u(k-4). \tag{10} $$

The pseudo-code of the proposed method for the Flexible Transmission Benchmark is similar to the Sine map.

In the analysed cases, it was possible to verify that the proposed method which the rules of interval arithmetic are used, there is a decrease of
Table 3: Comparison of width size for the simulation of flexible transmission benchmark. The reference is the width produced by means of Intlab (Rump, 1999). We have studied four cases, which initial conditions are as follow: (1): \(x_0 = 0.1\), (2): \(x_0 = 0.2\), (3): \(x_0 = 0.6\), and (4): \(x_0 = 0.8\).

| Case | \(n\) | Proposed Method | Intlab |
|------|-------|----------------|--------|
|      |       | width midpoint | width midpoint |
| (1)  | 1     | 0.1102e-16 0.8151300000000000 | 3.3307e-16 0.8151300000000000 |
|      | 5     | 4.623e-15 1.475024309409214 | 7.5939e-15 1.475024309409214 |
|      | 10    | 3.2540e-10 -0.385319792715174 | 4.483e-10 -0.385319792715172 |
| (2)  | 1     | 3.3307e-16 0.8409900000000000 | 4.4409e-16 0.8409900000000000 |
|      | 5     | 6.1062e-14 1.432470784456269 | 8.1712e-14 1.432470784456269 |
|      | 10    | 3.6389e-10 -0.406693858836176 | 4.8523e-10 -0.406693858836172 |
| (3)  | 1     | 5.5511e-16 0.9444300000000000 | 1.1102e-15 0.9444300000000000 |
|      | 5     | 6.5281e-14 1.262256684644492 | 1.0791e-13 1.262256684644491 |
|      | 10    | 3.8598e-10 -0.492190123320189 | 6.3973e-10 -0.492190123320189 |
| (4)  | 1     | 8.8818e-16 0.9961500000000000 | 1.4433e-15 0.9961500000000000 |
|      | 5     | 8.3933e-14 1.177149634738603 | 1.1702e-13 1.177149634738603 |
|      | 10    | 4.9464e-10 -0.534938255562194 | 6.9073e-10 -0.534938255562196 |

Algorithm 2 Pseudo-code of the Proposed Method for the Sine map.

1: \textbf{input} Number of iterations (N) and initial conditions
2: \textbf{system\_dependent}('setround',-Inf)
3: \(\hat{x}_{0,n} \leftarrow \hat{x}_{0,n}\)
4: \textbf{system\_dependent}('setround',Inf)
5: \(\hat{x}_{0,n}^+ \leftarrow \hat{x}_{0,n}\)
6: \textbf{for} \(n=1:N\) \textbf{do}
7: \textbf{system\_dependent}('setround',-Inf)
8: \(\text{aux1}_n \leftarrow 2.6868 \times x_n\)
9: \(\text{aux2}_n \leftarrow x_n^3\)
10: \(\text{aux3}_n \leftarrow 2.2462 \times \text{aux2}_n\)
11: \textbf{system\_dependent}('setround',Inf)
12: \(\text{aux4}_n \leftarrow 2.6868 \times x_n^+\)
13: \(\text{aux5}_n \leftarrow x_n^+^3\)
14: \(\text{aux6}_n \leftarrow 2.2462 \times \text{aux5}_n\)
15: \textbf{system\_dependent}('setround',-Inf)
16: \(x_{n+1} \leftarrow \text{aux1}_n - \text{aux6}_n\)
17: \textbf{system\_dependent}('setround',Inf)
18: \(x_{n+1}^+ \leftarrow \text{aux4}_n - \text{aux3}_n\)
19: \textbf{end for}
20: \textbf{output} \(x_{n+1}^-\) and \(x_{n+1}^+\)

the interval when compared to the Intlab toolbox, since the rounding was performed in each operation and the rounding outward, that is, \(\pi\) must be rounded downward and \(\pi\) must be rounded upward. And in general, the midpoint average difference is around 6.028e – 12, 9.8825e – 6 and 6.25e – 13, for Logistic Map, Sine Map and Flexible Transmission Benchmark, respectively.

**5 Conclusion**

The control of error in numerical simulations has a great importance. Due to computational limitation, the computer does not generate exact answers, but an approximation. Therefore, the interval arithmetic it is possible to say that the result will be within an interval, which this interval encompasses the errors of the simulations, guaranteeing a better evaluation of the result. As previously discussed, there are few publications that report on this subject, mainly using interval arithmetic to examine these errors.

This article has presented a novelty to sim-
ulate recursive functions using basic concepts of arithmetic interval and primitive functions in Matlab. The interval calculus of three systems, the logistic map, the sine map, and the flexible transmission benchmark have been performed. To validate this method, the same simulation was executed in the Intlab toolbox. We have found narrower intervals using the proposed method. It is clear that the main advantage of the proposed method is to present a novelty to simulate recursive functions without a specific toolbox. More complex representations for system identification, such as neural networks, should be investigated in future work.

Acknowledgments

The authors thank PPGEL, CAPES, CNPq/INERGE, FAPEMIG and the Federal University of São João del-Rei for their support.

References

Barbosa, A. M., Takahashi, R. H. C. and Aguirre, L. A. (2015). Equivalence of non-linear model structures based on pareto uncertainty, IET Control Theory Applications 9(16): 2423–2429.

Bombois, X., Scorletti, G., Gevers, M., den Hof, P. V. and Hildebrand, R. (2006). Least costly identification experiment for control, Automatica 42(10): 1651 – 1662.

Chen, S. and Billings, S. A. (1989). Representations of non-linear systems: the NARMAX model, International Journal of Control 49(3): 1013–1032.

Feigenbaum, M. J. (1978). Quantitative universality for a class of nonlinear transformations, Journal of statistical physics 19(1): 25–52.

Ferreira, D. D., Nepomuceno, E. G. and Cerqueira, A. S. (2006). Nonlinear model validation using coherence, XII Latin-American Congress on Automatic Control-CLCA.

Gilmore, R. and Lefranc, M. (2012). The topology of chaos: Alice in stretch and squeezeland, John Wiley & Sons.

Guedes, P. F. S., Peixoto, M. L. C., Barbosa, A. M., Martins, S. A. M. and Nepomuceno, E. G. (2017). The lower bound error for polynomial NARMAX using an arbitrary number of natural interval extensions, DINCON 2017 – Conferência Brasileira de Dinâmica, Controle e Aplicações.

IEEE (1985). IEEE standard for binary floating-point arithmetic, Institute of Electrical and Electronics Engineers, New York. Note: Standard 754–1985.

IEEE (2015). Iee standard for interval arithmetic, IEEE Std 1788-2015 pp. 1–97.

Landau, I., Rey, D., Karimi, A., Voda, A. and Franco, A. (1995). A flexible transmission system as a benchmark for robust digital control, European Journal of Control 1(2): 77 – 96.

Ljung, L. (1998). System identification, Signal analysis and prediction, Springer, pp. 163–173.

May, R. M. (1976). Simple mathematical models with very complicated dynamics, Nature 261(5560): 459.

Moore, R. E. (1979). Methods and applications of interval analysis, Vol. 2, SIAM.

Nepomuceno, E. G. (2014). Convergence of recursive functions on computers, The Journal of Engineering 2014(10): 560–562.

Nepomuceno, E. G., Takahashi, R. H. C., Amaral, G. F. V. and Aguirre, L. A. (2003). Nonlinear identification using prior knowledge of fixed points: a multiobjective approach, Journal of Bifurcation and Chaos vol. 13: 1229–1246.

Nepomuceno, E. and Martins, S. (2016). A lower bound error for free-run simulation of the polynomial NARMAX, Systems Science & Control Engineering 4(1): 50–58.

Peixoto, M. L. C., Matos, M. T. R., Júnior, W. R. L., Martins, S. A. M. and Nepomuceno, E. G. (2017). Identification of dynamic systems with interval arithmetic, XIII Simpósio Brasileiro de Automação Inteligente - SBAI 2017, pp. 471–476 (In Portuguese).

Qun, Z. (2012). Interval control methods of rounding error in numerical calculation, 2012 7th International Conference on Computer Science Education (ICCSE), pp. 1304–1306.

Rothwell, E. J. and Cloud, M. J. (2012). Automatic error analysis using intervals, Education, IEEE Transactions on 55(1): 9–15.

Rump, S. (1999). INTLAB - INTerval LABoratory, in T. Csendes (ed.), Developments in Reliable Computing, Kluwer Academic Publishers, Dordrecht, pp. 77–104.