International Journal of Control
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tcon20

Special issue on the use of computer algebra systems for computer aided control system design

N. P. Karampetakis & A. I. G. Vardulakis (Eds)
Published online: 20 Feb 2007.

International Journal of Control
Vol. 79, No. 11, November 2006, 1313–1320

Taylor & Francis
Taylor & Francis Group

Editorial

Special issue on the use of computer algebra systems for computer aided control system design

N. P. KARAMPETAKIS* and A. I. G. VARDULAKIS (Eds)

The importance of the continuing and growing need in the systems and control community for reliable algorithms and robust numerical software for increasingly challenging applications is well known and has already been reported elsewhere (*IEEE Control Systems Magazine*, Vol. 24, Issue 1). However, we have all had the experience of working on a mathematical project where an increased number of symbolic manipulations was needed. In a simple case, the required computation might have been to compute the Laplace transform or the inverse Laplace transform of a function, or to find the transfer function matrix for a given system topology where parameters are included. In a more demanding situation the required computation might have been to find the parametric family of solutions of a polynomial matrix Diophantine equation resulting from a variety of control problems such as those associated with stabilization, decoupling, model matching, tracking and regulation, or to compute the Smith McMillan form of a rational transfer function matrix in order to obtain a better insight into a number of structural properties of a system. The desire to use a computer to perform long and tedious mathematical computations such as the above led to the establishment of a new area of research whose main objective is the development: (a) of systems (software and hardware) for symbolic mathematical computations, and (b) of efficient symbolic algorithms for the solution of mathematically formulated problems. This new subject area is referred to by a variety of terms such as symbolic computations, computer algebra, algebraic algorithms to name a few. During the last four decades this subject area has accomplished important steps and it is still continuing its evolution process.

Computer algebra systems

Although Computer Algebra Systems (CAS) and Numerical Software (NS) have not been designed and developed for solving the same problems they could be considered as complementary tools rather than adversaries. CAS can be classified according to their functionality into (a) the general purpose CAS i.e. systems that incorporate functions for most subject areas of mathematics such as Macsyma, Reduce, Maple, Mathematica, etc, and (b) special purpose CAS which are systems that are specialized on a specific subject area of mathematics i.e. systems such as CoCoA (deals with computations in multivariate polynomial rings), DELiA (differential equations) etc. Apart from these two main categories there are also software packages of programs that have been built independently either in the programming environment of a general purpose CAS or in a special purpose CAS. To name a few: CALI is a REDUCE package that contains algorithms for computations in commutative algebra, Control System Professional (Bakshee 2003, Palancz *et al*. 2005) is a MATHEMATICA package that contains algorithms for control, NCAlgebra (Helton *et al*. 2000) is a non-commutative algebra package that runs under MATHEMATICA, SchematicSolver contains details for drawing, solving, simulating and implementing a system, Block Diagram Reduction Toolbox (drawing and calculating the transfer function of a system in terms of transfer functions of its components), PARADISE is a MATLAB toolbox for the design and analysis of robust control systems in which parametric models can be specified using Simulink etc. In table 1 we give an overview of some of the most important computer algebra systems that have been developed during the last four decades.

*Corresponding author. Email: karampet@math.auth.gr

International Journal of Control
ISSN 0020–7179 print/ISSN 1366–5820 online © 2006 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00207170600882346

Table 1. Computer algebra systems.

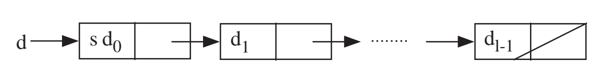
Year	Computer algebra system	Purpose
1961	SAINT	Indefinite integration
1964–66	ALTRAN, MATHLAB	Manipulation of polynomial and rational functions
1966–67	SIN	Symbolic integration
1968 – now	REDUCE http://www.uni-koeln.de/REDUCE	Starting from physics calculation. Solution for large scale formal problems in mathematics, science and engineering.
1968	MATHLAB-68	Improved version of Mathlab
1968 – now	MACSYMA http://www.macsyma.com	General purpose CAS
Late 1970's	muMATH	
1980 – now	MAPLE http://www.maplesoft.com	General purpose CAS
1980 – now	DERIVE	General purpose CAS, improved version of muMATH
1984 – now	SINGULAR http://www.mathematik.uni-kl.de/pub/~zca/Singular	A *CAS* for polynomial computations
1988 – now	SMP, MATHEMATICA http://www.wolfram.com	General purpose CAS
1989 – now	MuPAD http://www.mupad.de http://www.sciface.com	General purpose CAS
1991 – now	AXIOM http://www.nongnu.org/axiom	Successor to Stratchpad. General purpose CAS, that allows users to write algorithms over general fields or domains
	CAYLEY	Group theory
Late 1980s	MAGMA http://magma.maths.usyd.edu.au	General purpose CAS for algebra, number theory, algebraic geometry, algebraic topology, algebraic combinatorics etc.
1986–1997	GAP, GAP 2 (2000) http://www-gap.mcs.st-and.ac.uk	Groups, algorithms and programming, computational discrete algebra
	FORM	High energy physics calculation
1990–1996	LiE http://www-math.univ-poitiers.fr/~maavl/LiE	Lie algebra calculations
1992	MACAULAY 2 http://www.math.uiuc.edu/Macaulay2	Algebraic geometry and commutative algebra
Mid 1980s – 2000	PARI ftp://megrez.math.u-bordeaux.fr/pub/pari	Number theory

Advantages and disadvantages of the CAS

It was reported in Higham *et al*. (2004) that when solving a computational problem using a computer, the accuracy of the computed solution depends on three main factors:

(a) the machine arithmetic — i.e. the rounding unit and the range of this arithmetic,
(b) the computation problem — i.e. the sensitivity of its solution relative to changes in the data, and
(c) the computational algorithm — i.e. the numerical stability of the algorithm used.

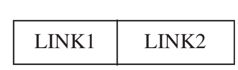
In a typical digital computer the specific word length used in order to store a number is limited by the number of distinct encodings which is typically 8, 16, 32, 36, 48 or 64 bits. These restrictions on the representation of objects have a major impact on the precision and the length of the numbers that can be used in numerical computations and are not sufficient for the purpose of symbolic computations. CAS use different techniques to store multiprecision integers and rational numbers and therefore are not suffering from precision and length problems. The largest integer that can be used in a typical digital computer that stores numbers in a word of 32 bits is $2^{31}-1=2147483647$ while in a CAS any multiprecision integer $d=s\sum_{i=0}^{l-1} d_i b^i$ can be represented by the linked list:



or by a dynamic array allocation:

$$\begin{bmatrix} sl & d_0 & d_1 & \ldots & d_{l-1} \end{bmatrix}$$

As an extension, a rational number can be represented by a node of the form



where LINK1 is a pointer to the numerator multiprecision integer (either a linked list or an allocation array) and similarly LINK2 is a pointer to denominator. Therefore, one of the major advantages in CAS is that we can use any precision that we want. On the other hand, the above representation of multi-precision numbers results into two disadvantages: (a) more memory is needed to store numbers, and (b) more computational power is required to process numbers. These two main disadvantages make CAS not very suitable for large scale problems, since, such problems, require much memory and speed. However, nowadays many attempts have been made for the solution of this kind of problems by the investigation of faster computational algorithms. By using multiprecision numbers we can use any accuracy we want and therefore avoid numerical stability problems as well as conditioning problems usually face in numerical packages.

Example 1: Consider the well known Kalman criterion of controllability, which states that the matrix pair $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ that corresponds to a state space model with n states and m inputs, is controllable if and only if the controllability matrix

$$\ell = (B \quad AB \quad A^2B \quad \ldots \quad A^{n-1}B)$$

has full rank. Following the example due to Paige (1981):

$$A = \operatorname{diag}\begin{pmatrix}2^0 & 2^{-1} & 2^{-2} & \ldots & 2^{-9}\end{pmatrix} \in \mathbb{R}^{10\times 10}$$
$$B = \begin{pmatrix}1 & 1 & 1 & \ldots & 1\end{pmatrix}^T \in \mathbb{R}^{10\times 1}.$$

The Kalman criterion does not yield a numerically viable test for controllability although the above is a symbolically viable test, as it is easily seen by the following Mathematica program, since it uses infinite precision or otherwise exact representation for integer and rational numbers:

```
In[1] := <<LinearAlgebra`MatrixManipulation`
In[2]:= A=Table[If[i == j, 2^(i-1), 0], {i, 1, 10}, {j, 1, 10}];
In[3]:= B=Table[{1}, {i, 1, 10}];
In[4]:= L=B;
In[5]:= Do [L= AppendRows [L, MatrixPower [A, i-1].B], {i, 2, 10}];
In[6]:= MatrixRank [L]
Out[6]:= 10
```

In the first line of the program we call the subpackage "MatrixManipulation" of Mathematica that is included in the package "Linear Algebra". In 2nd and 3rd line we create the matrix pair (A,B). In the 4th–5th line we create the controllability matrix ℓ and in the 6th line we compute the rank of ℓ.

Example 2: Consider for example the Ackermann formula which for a given matrix pair (A, B) evaluates a matrix F such that the pencil $sI-(A+BF)$ has specific eigenvalues i.e.,

$$F = [0 \quad 0 \quad \ldots \quad 1][B \quad AB \quad \ldots \quad A^{n-1}B]^{-1}a(A),$$

where

$$a(s) = \det[sI - (A + BF)] = s^n + a_1 s^{n-1} + \cdots + a_n$$

is the desired closed loop characteristic polynomial. According to Svensson (1993), Ackermann's method in the MATLAB Control Toolbox yields a feedback matrix of poor quality. More specifically, if we take a random (A, B) pair with only ten states and one input and assign random eigenvalues for the closed loop characteristic polynomial $a(s)=\det[sI-(A+BF)]$ symmetrically located with respect to the real axis, then using the MATLAB control function *acker* we recompute the eigenvalues of $A+BF$ with only 3 digits of accuracy. However, by using the Mathematica program in figure 1 we get the desired results without loss of any accuracy, since Mathematica uses exact arithmetic and does not suffer from loss of precision or significance when manipulating integer and/or rational numbers. It is known that, due to matrix multiplications (for the controllability matrix) and inversion of the controllability matrix ℓ, the use of a numerical approach for the computation of F leads to a severe loss of accuracy when a numerical approach is being used. In the first line of the Mathematica code above we call the subpackage "MatrixManipulation" of Mathematica that is included in the package "LinearAlgebra". In the second and third lines we create a random (A, B) pair with 10 states and 1 input. Each matrix has integer entries in the range $[-10, 10]$. In the fourth line we create the controllability matrix ℓ. In the fifth line we create the matrix function $a(A)$ where $a(s)=s^3(s+1)^6(s+3)$ and in the sixth line we use the Ackermann formula. Below we give the $\{1, 1\}$ element of the matrix F in order to see the precision

```
In[1] := <<LinearAlgebra`MatrixManipulation`
In[2]:= A=Table[Random[Integer,{-10,10}],
{i,1,10},{j,1,10}];
In[3]:= B=Table[Random[Integer,{-10,10}],
{i,1,10},{j,1}];
In[4]:= S1=B;Q=S1;Do[S1=A.S1;
Q=AppendRows[Q,S1],{i,1,9}];
In[5]:= k=MatrixPower[A,3].
MatrixPower[A+IdentityMatrix[10],6].
(A+3*IdentityMatrix[10]);
In[6]:= F={{0,0,0,0,0,0,0,0,0,1}}.Inverse[Q].k;
```

Figure 1. Mathematica program.

of the numbers that *Mathematica* is using for such kind of computations.

```
In[1]: = <<ControlSystems';
  (*we call the CSP package *)
```

$$\frac{10278979636146769009136123193667855749665483672936864194923 49}{5544438285328489874400541677185432379656154188635229715430 21}.$$

Another main advantage of CAS is that of symbolic computing or mixed symbolic-numeric computing, that means you can handle symbolic objects in the same way you handle numbers (determine determinants, inverses, transposes etc. of a matrix containing symbolic elements in contrast to a numerical environment, solve polynomial Diophantine equations, solve exact linear differential equations, find the transfer function matrices of systems that may include parameters in their descriptions etc.).

Example 3: Consider the state-space description of a cart with inverted pendulums, for small $|\theta_1|$ and $|\theta_2|$ given by

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\\ \dot{x}_3(t)\\ \dot{x}_4(t)\end{bmatrix} = \underbrace{\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ \frac{(M+m)g}{Ml_1} & \frac{mg}{Ml_1} & 0 & 0\\ \frac{mg}{Ml_2} & \frac{(M+m)g}{Ml_2} & 0 & 0\end{bmatrix}}_{A}\begin{bmatrix}x_1(t)\\ x_2(t)\\ x_3(t)\\ x_4(t)\end{bmatrix} + \underbrace{\begin{bmatrix}0\\ 0\\ -\frac{1}{Ml_1}\\ -\frac{1}{Ml_2}\end{bmatrix}}_{B} u(t)$$

$$x_1(t) = \theta_1(t),\ x_2(t) = \theta_2(t),\ x_3(t) = \dot{\theta}_1(t),\ x_4(t) = \dot{\theta}_2(t),$$

where M is the mass of the cart, l_1, l_2 are the lengths of the two inverted pendulums that both have bobs of mass m.

We can define the above system using the Control Systems Professional (CSP) package of Mathematica as follows

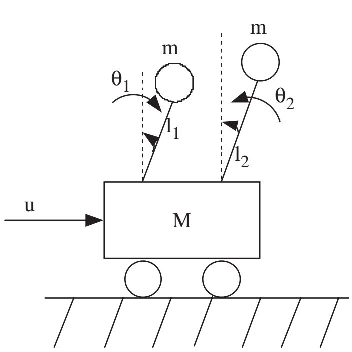


Figure 2. Cart with two inverted pendulums.

```
In[2]: = <<LinearAlgebra'
(* we call the Linear Algebra package *)
```

$$\texttt{In[3]} := \texttt{a} = \left\{\{0, 0, 1, 0\}, \{0, 0, 0, 1\}, \left\{\frac{(M+m)g}{Ml_1}, \frac{mg}{Ml_1}, 0, 0\right\}, \left\{\frac{mg}{Ml_2}, \frac{(M+m)g}{Ml_2}, 0, 0\right\}\right\};$$

(* we define the matrix A *)

$$\texttt{In[4]} := \texttt{b} = \left\{\{0\}, \{0\}, \left\{-\frac{1}{Ml_1}\right\}, \left\{-\frac{1}{Ml_2}\right\}\right\};$$

(* we define the matrix B *)

```
In[5]: = system = StateSpace[a, b];
(* we define the state space system with *)
(* the matrices A and B *)
```

Then we can easily check that the controllability matrix of the above system is given by

```
In[6]: = L = ControllabilityMatrix[system]
```

$$\texttt{Out[6]} = \left\{\left\{0, -\frac{1}{Ml_1}, 0, -\frac{g(m+M)}{M^2l_1^2} - \frac{gm}{M^2l_1l_2}\right\}, \left\{0, -\frac{1}{Ml_2}, 0, -\frac{g(m+M)}{M^2l_2^2} - \frac{gm}{M^2l_1l_2}\right\}, \left\{-\frac{1}{Ml_1}, 0, -\frac{g(m+M)}{M^2l_1^2} - \frac{gm}{M^2l_1l_2}, 0\right\}, \left\{-\frac{1}{Ml_2}, 0, -\frac{g(m+M)}{M^2l_2^2} - \frac{gm}{M^2l_1l_2}, 0\right\}\right\}$$

which has determinant

```
In[7]: = Factor[Det[L]]
```

$$\texttt{Out[7]} = -\frac{g^2(l_1 - l_2)^2}{M^4 l_1^4 l_2^4}$$

and thus the system is controllable iff $l_1 \neq l_2$. It is easy to find a state feedback of the form $u(t) = -Kx(t) + v(t)$ that will place the poles of the system to $\{-1, -1, -1, -3\}$ when $l_1 \neq l_2$.

```
In[8]: = f = StateFeedbackGains[system,
{-1, -1, -1, -3},
   Method → Ackermann]
```

$$\text{Out[8]} = \left\{\left\{\frac{-g^2 m l_1 + g^2 m l_2 + g^2 M l_2 + 12 g M l_1 l_2 + 3 M l_1^2 l_2}{g(l_1 - l_2)}, \frac{-g^2 m l_1 - g^2 M l_1 + g^2 m l_2 - 12 g M l_1 l_2 - 3 M l_1 l_2^2}{g(l_1 - l_2)}, \frac{2 M l_1 (3g + 5 l_1) l_2}{g(l_1 - l_2)}, -\frac{2 M l_1 l_2 (3g + 5 l_2)}{g(l_1 - l_2)}\right\}\right\}$$

```
In[9]: = Det[s * IdentityMatrix[4] − (a −
b.f)]//Factor
Out[9] = (1 + s)^3 (3 + s)
```

Apart from the main advantage of the exact arithmetic that CAS have, they also have other advantages such as:

- interactivity–you give questions and directly get your answers,
- arbitrary precision-arithmetic or even exact arithmetic can be used to utilize some of the direct (but numerically error proned) methods efficiently in control system design–apart of the multiprecision arithmetic you may use, you can select specific precision for your work,
- visualization of your results–you can view in 2-D and 3-D graphics your results (XY plots, XYZ plots, polar plots, log plots etc.) and import/export your results to known graphics formats,
- the existence of hundrends of built-in procedures covering both general and specialized areas of mathematics such as Laplace transforms, Inverse Laplace transforms, Z-transforms, inverse Z-transforms, Smith form of polynomial matrices,
- the existence of unique high-level programming languages and programming environments allowing specific procedures to be written,
- they are usually available for different types of computer platforms,
- CAS enables us to fully participate in the mathematical exploration, by exploring different scenarios by asking or being asked ''what-if'' type of questions,
- CAS help us to work with commutative or noncommutative algebra problems, where numerical environments are not suitable,

CAS can also be used in research

- to test conjectures — supporting or refute conjectures,
- to help us to carry out the solutions steps of a mathematical algorithm and avoiding the hand-calculation mistakes,
- to help us to design and build a *CAS* for a specialized task in our research area,
- to help us to customize and improve our algorithms for the solution of a problem,
- to give us closed form solutions, and thus provide deeper insight to the problem — we can find for example the family of stabilizing compensators of a system in a general parametric framework by retaining the extra degrees of freedom as symbols appearing as free parameters in the expression for the compensator. These free parameters can be then optimized to achieve specific goals associated with the internal stability and/or robustness of the resulting closed loop system,
- to assist in creating large mathematical tables such as integral tables, tables of special functions and so on,
- to help us to explore new algorithms and techniques,
- to give us time to focus on ideas and not on calculations — by helping us for example to theoretically analyse the structural properties of linear and nonlinear systems when some parameters are included in the model and we are looking for generic results instead of particular solutions for specific values of them.

However, CAS as part of their large memory requirements and relative poor performance are also accompanied by disadvantages such as

- difficulty to define the domain of the solution you are looking for,
- contain particularities that are learned only with experience,
- specialized areas are not covered by existing general purpose CAS,
- difficulty to give exact answers in problems where a closed form solution does not exist, e.g., exact solution of a general fifth order polynomial equation or a closed formula solution to a non-linear problem,
- difficulty to connect with other applications,
- difficulty to handle large scale problems due to the their excessive computational resources requirement, i.e. difficulty to handle large size of polynomials and polynomial matrices with high degrees and/or large dimensions,
- symbolic matrix analysis with a large number of symbols may become impractical,
- may give meaningless generic answers.

For the reasons described above, a combination of numerical and symbolic techniques is a very good way to reduce simultaneously the loss of accuracy and the required time (Hecker and Varga 2006, Karcanias *et al.* 2006, Söylemez and Ustoglu 2006)

CASs for engineering and control

CAS's are currently successfully used in several areas of engineering, such as robotics, non-linear dynamics, computational fluid dynamics, aerodynamics, control systems etc. For instance, computer algebra techniques have been used in robotics problems (Grabmeier *et al.* 2003), such as the solution of the inverse kinematic problem (Buchberger 1985, Lee and Liang 1988), and the direct kinematic problem (Husty 1996, Higham *et al.* 2004) in the so-called "piano-movers" or collision avoidance problems where a robot of certain geometry is supposed to move a payload with given shape through an environment containing a number of known obstacles without collision (Canny 1987, Basu *et al.* 1996), and to mechanical synthesis (Cohen and Heck 1995).

Recently, there has been a growing interest in the application of CAS to control design and analysis. Computer algebra techniques has been used for system modelling (Gawtgrio 1993, Gawthrop and Balance 1997, Varga and Looye 1999), and system analysis and synthesis methods. In Munro (1999) an up-to-date treatment of the significant impact of symbolic computing in the field of control engineering is given. Computer algebra techniques have been also used for the study of nonlinear control systems (Buchberger 1985, Akhrif and Blankenship 1987, van Essen 1994, Jager 1995, Kitamoto 1996, Lee and Liang 1988, Rothfuí *et al.* 1994, Schlacher and Kugi 2001, Higham *et al.* 2004), linear control systems (Munro 1999, Ogunye 1996, Ogunye and Penlidis 1996 and the references therein) and special problems of control such as the optimal control problem (Boulehmi and Calvet 1997, Stoutemyer 1979), the pole assignment problem (Söylemez and Munro 1998), the block reduction problem (Söylemez and Ustoglu 2004), matrix inequalities in control (de Oliveira and Helton 2003), solutions of ARMA representations (Jones *et al.* 2003), computation of the Smith–McMillan Form (Munro and Tspekis 1994), modelling and simulation of robot manipulators (Cetinkunt and Ittop 1992, Lin and Lewis 1994, Pota and Alberts 1995) etc. In Ogunye (1996) a comprehensive CAS package SYMCON is described created in Maple, for the design of multivariable discrete-time control systems using the polynomial equation approach. Wolfram Research the producer of one of the most popular CAS, Mathematica, in 1996 released the Control System Professional package (Bakshee 2003) — a conducive environment for solving control engineering problems — that is now shipping with numerous extensions as Control System Professional Suite.

A new action group for symbolic computation

Due to the interest in the application of computer algebra to control analysis and design an Action Group for Symbolic Computations for CACSD has been created as a part of the recently established IEEE Control System Society's Technical Committee on Computer Aided Control Systems Design see (http://anadrasis.math.auth.gr/cacsd/Home.htm. The main aim of this action group is to establish an information exchange forum for control related symbolic algorithms and software. The web page given above includes an open list with people, working in this subject area, calls for invited sessions, conferences on symbolic computations for CACSD, list of software, links to other relative web pages, and list of publications or even reprints, and announcements that are relevant to this area of research. The collection of all the above information: (a) would be an advantage for the organization of invited sessions at conferences as well as joint research projects, (b) will make known the existing research groups and their area of interests as well as existing CAS packages for CACSD.

Objectives and contents of the special issue

This special issue aims not at just a collection of papers on symbolic computations, but rather to address, at least to some extent, a coherent vision of the role of symbolic computations in control analysis and design problems. The articles are written in a way so that readers who are not experts in symbolic methods will be able to learn about these techniques. Illustrative examples are usually given to enlighten the main points of the discussions.

The papers in this special issue can be grouped into two categories: (a) the first 9 papers that use only symbolic methods, and b) the last 4 papers that use hybrid (mixed numerical-symbolic) methods.

Anai and Hara (2006) propose a new method of parameter space design for robust control synthesis, which guarantees the real stability radius constraint, accomplished by using quantifier elimination (QE).

Dolecek and Mitra (2006), present a *Matlab*-based symbolic sensitivity analysis of second-order IIR digital filters.

Fotiou *et al.* (2006) present two algebraic methods that help us to solve the parametric optimization problem for optimal control problems arising in model predictive control.

Jeannerod and Villard (2006), show that the asymptotically fastest known algorithms for some basic problems on univariate polynomial matrices, such as computation of rank, nullspace, determinant, generic inverse and reduced form, can be reduced to two computer algebra techniques, namely those of minimal basis computations and matrix fraction expansion/reconstruction, and to polynomial matrix multiplication.

Lutovac and Tošić (2006) present the role of symbolic computations in control engineering and signal processing. They provide illustrative application examples as appropriate to linear systems, non-linear systems, algorithm development, modelling, and simulation.

M.C. de Oliveira and Helton (2006), describe computer algebra algorithms, methodology and implementation which allow users to convert many system problems to Linear Matrix Inequalities (LMIs). The algorithms and ideas are implemented in a symbolic non-commutative algebra package NCAlgebra and the package *NCGB* that runs in Mathematica and has been developed by the authors.

Perdon *et al.* (2006) describe how computational algebra techniques can effectively use the geometrical approach in dealing with dynamical systems over rings. More specifically, they describe in details how to practically solve the distrubance decoupling and the block decoupling problem for delay differential systems, by using the symbolic computational package CoCoA.

Söylemez and Ustoglu (2006) present some classical control examples that illustrate the advantages of the computer algebra in the process of control system design, focusing to block diagram reduction, calculation of stabilizing compensators, dominant pole assignment and robust pole assignment.

Zheng *et al.* (2006) present the application of symbolic algebra techniques to the Mathematica implementation of a set of output-feedback pole assignment algorithms, for systems characterized by parametric uncertainty.

Karcanias *et al.* (2006) emphasize the significance of hybrid computations (mixed numerical and symbolic computations) in complex problems such as the computation of the greatest common divisor (GCD) of several polynomials that emerges in many fields of applications.

Liang and Chen (2006), develop a hybrid (symbolic and numerical) method based on Matlab Symbolic Math Toolbox (Moler and Costa 1976) to simulate some typical problems of boundary control of fractional order diffusion-wave equations.

Liu *et al.* (2006) describe the asymptotic time-scale and eigenstructure assignment (AETA) algorithm, which is one of the major applications of the structural decomposition approach in linear systems theory. Then they describe its software implementation in detail and show how the ATEA algorithm has been developed in such a way that facilitates the symbolic computation of the resulting feedback gains. Finally, they use simple applications to illustrate how the symbolic computation of ATEA based state feedback laws leads to feedback laws that are explicitly parameterized in the design parameter.

Hecker and Varga (2006) present symbolic manipulation techniques which are very useful in obtaining low-order linear fractional transformation (LFT) representations of linear parametric models. The results presented in this work are applied to the Research Civil Aircraft Model (RCAM), which is one of the most complicated existing parametric models in the literature, with great success.

Dedication

This special issue is dedicated to the memory of Professor Neil Munro who, very unfortunately, passed away in July 2004. For many years Professor Munro had dedicated his research to computer aided control systems design, in particular robust control design for linear systems using symbolic techniques with their novel applications to control systems design including pole assignment algorithms. This is indeed the area originated by him and has received wide-spread attention in our control system research community. In the last days of his life Professor Munro was working very hard to build a toolbox for CAD of control systems analysis and design in Mathematica.

For many years the had been the director of the UMIST Control Systems Centre. He made significant contributions to the development of the Centre and was highly respected by the international control research community. His contributions and papers on Multivariable Control Theory, Computer Aided Control Systems Design and Robust Design Methods have been highly regarded by the community and his papers have been very widely cited. He will be remembered as an excellent scientist, a wise mentor, a conscientious colleague and a reliable friend.

H. Wang
Control Systems Centre, School of Electrical and Electronics Engineering, The University of Manchester, 1P.O. Box 88, Manchester M60 1QD, UK, hong.wang@manchester.ac.uk.

References

O. Akhrif and G.L. Blankenship, "Computer algebra for analysis and design of nonlinear control systems", *Proceedings of the American Control Conference*, **1**, pp. 547–554, 1987.

H. Anai and S. Hara, "A parameter space approach to fixed-order robust controller synthesis by quantifier elimination", *International Journal of Control*, 79, within this special issue, 2006.

I. Bakshee, *Control System Professional*, Champaign: Wolfram Research, 2003.

S. Basu, R. Pollack and M.-F. Roy, "Computing roadmaps of semi-algebraic sets", in *28th ACM Symp. on Theory of Computing*, Philadelphia, Pennsylvania, USA, 1996, pp. 168–173.

M. Boulehmi and J.-L.Calvet, "Using computer algebra for solving some optimal control problems by dynamic programming", in Computer aided control systems design (CACSD'97), L. Boullart *et al.* Eds., *Proceedings of the 7th IFAC symposium*, Gent, Belgium, April 28–30, 1997. Amsterdam: Pergamon, 1997.

B. Buchberger, "Grobner bases: an algorithmic method in polynomial ideal theory", in *Multidimensional Systems Theory*, N.K. Bose, Ed., Chapter 6, D. Reidel Publishing Co., 1989, pp. 184–232.

J. Canny, *The complexity of motion planning*, Camridge, Mass: MIT Press, 1987.
S. Cetinkunt and B. Ittop, "Computer-automated symbolic modeling of dynamics of robotic manipulators with flexible links", *IEEE Transactions of Robotics and Automation*, 8, pp. 94–105, 1992.
A.M. Cohen and A. Heck, "Applied computer-algebra: experience from cam design", in *Computer Algebra in Science and Engineering*, J. Fleischer, J. Grabmeier, F.W. Hehl, and W. Kuchlin, Eds., *Proceedings ZIF Bielefeld, 28–31 August 1994*, Singapore: World Scientific Publishing, 1995, pp. 303–316.
G.J. Dolecek and S.K. Mitra, "Using MATLAB symbolic toolbox for the sensitivity analysis of IIR digital filters", *International Journal of Control*, 79, within this special issue, 2006.
H. van Essen, "Analysis and design of nonlinear control systems with the symbolic computation system MAPLE", *Journal A*, 35(2), pp. 3–8, 1994.
I.A. Fotiou, P. Rostalski, P.A. Parrilo and M. Morari, "Parametric Optimization and Optimal Control using Algebraic Geometry", *International Journal of Control*, 79, within this special issue, 2006.
J. Grabmeier, E. Kaltofen and V. Weispfenning, *Computer Algebra Handbook*, Berlin, Heidelberg, New York: Springer-Verlag, 2003.
B. de Jager, "The use of symbolic computation in nonlinear control: Is it viable?", *IEEE Trans. on Auto. Control*, 40, pp. 84–89, 1995.
C.-P. Jeannerod and G. Villard, "Asymptotically fast polynomial matrix algorithms for multivariable systems", *International Journal of Control*, 79, within this special issue, 2006.
P. Gawthrop, "Symbolic modeling in control", in *CAD for Control Systems*, D.A. Linkens, Ed., Dekker, 1993, pp. 127–146.
P.J. Gawthrop and D. Balance, "Symbolic algebra and physical model-based control", *Computing and Control Journal*, 8, pp. 70–76, 1997.
S. Hecker and A. Varga, "Symbolic manipulation techniques for low order LFT-based parametric uncertainty modeling", *International Journal of Control*, 79, within this special issue, 2006.
J.W. Helton, M. Stankus and R.L. Miller, *NCAlgebra*, San Diego, CA: Mathematics Department, University of California, 2000. http://www.math.ucsd.edu/~ncalg.
N. Higham, M. Konstantinov, V. Mehrmann and P. Petkov, "The sensitivity of computational control problems", *IEEE Control Systems Magazine*, 24, pp. 28–43, 2004.
M.L. Husty, "An algorithm for solving the direct kinematics of general stewart-platforms", *J. Mech. And Mach. Th.*, 31, pp. 365–380, 1996.
J. Jones, N.P. Karampetakis and A.C. Pugh, "Solutions of discrete ARMA-representations via Maple", *Applied Mathematics and Computation*, 139, pp. 437–489, 2003.
N. Karcanias, M. Mitrouli and D. Triantafyllou, "Matrix Pencil methodologies for computing the greatest common divisor of polynomials: Hybrid algorithms and their performance", *International Journal of Control*, 79, within this special issue, 2006.
T. Kitamoto, "The applications of computer algebra to the control of nonlinear systems (Japanese)", *RIMS Kokyuroku*, 941, pp. 143–150, 1996.
H.Y. Lee and C.G. Liang, "Displacement analysis of the general spatial 7-link 7R mechanism", *J. Mech. And Mach. Th.*, 3, pp. 219–226, 1988.
J. Liang and Y.Q. Chen, "Hybrid symbolic and numerical simulation studies of time-fractional order wave-diffusion systems", *International Journal of Control*, 79, within this special issue, 2006.
J. Lin and F.L. Lewis, "A symbolic formulation of dynamic equations for a manipulator with rigid and flexible links", *International Journal of Robotics Research*, 13, pp. 454–466, 1994.
X. Liu, Z. Lin and B.M. Chen, "Symbolic realization of asymptotic time-scale and eigenstructure, assignment design method in multivariable control", *International Journal of Control*, 79, within this special issue, 2006.
M.D. Lutovac and D.V. Tošić, "Symbolic design of control systems using Mathematica", *International Journal of Control*, 79, within this special issue, 2006.
M.D. Lutovac and D.V. Tošić, *SchematicSolver Version 2*, Available online at: www.schematicsolver.com.
C. Moler and P.J. Costa, *Symbolic Math Toolbox*, Natick, MA: The Mathworks Inc, 1976.
N. Munro and P. Tsapekis, "Some recent results using symbolic algebra", in *Proceedings of the IEEE/IFAC Joint Symposium on CACSD*, March 1994, Tuscon, Arizona, pp. 109–116, 1994.
N. Munro (Ed.), *Symbolic methods in control system analysis and design*, IEE Control Engineering Series, 56, Stevenage, Herts: The Institution of Electrical Engineers (IEE), 1999.
M.C. de Oliveira and J. William Helton, "Computer algebra tailored to matrix inequalities in control", *International Journal of Control*, 79, within this special issue, 2006.
A.B. Ogunye, "SYMCON: Symbolic computation controller package for use with MAPLE V", in *Proceedings of the 1996 IEEE International Symposium on CACSD*, Dearborn MI, September 15–18, pp. 488–494 1996.
A.B. Ogunye and A. Penlidis, "State space computations using MAPLE V", *IEEE Control Systems Magazine*, 16, pp. 70–77, 1996.
M.C. de Oliveira and J.W. Helton, "Computer algebra tailored to matrix inequalities in control", in *42nd IEEE Conference on Decision and Control*, 2003.
C.C. Paige, "Properties of numerical algorithms related to computing controllability", *IEEE Trans. Automat. Contr.*, 26, pp. 130–138, 1981.
B. Palancz, Z. Benyo and L. Kovacs, "Control System Professional Suite", *IEEE Control Systems Magazine*, 25, pp. 67–75, April 2005.
A.M. Perdon, M. Anderlucci and M. Caboara, "Efficient algorithms for geometric control of systems over rings", *International Journal of Control*, 79, within this special issue, 2006.
H.R. Pota and T.E. Alberts, "Multivariable transfer functions for a slewing piezoelectric laminate beam", *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control*, 117, pp. 352–359, 1995.
R. Rothfuι, J. Schaffner and M. Zeitz, "Computer-algebra program for analysis and design of nonlinear control systems", in Systems and networks: mathematical theory and applications, Helmke, Uwe *et al.*, Eds, *Proceedings of the 10th international symposium on the mathematical theory of networks and systems*, *MTNS '93*, held in Regensburg, Germany, August 2–6, 1993. Volume II: Invited and contributed papers. Berlin: Akademie Verlag. *Math. Res.* **79**, pp. 437–440, 1994.
K. Schlacher and A. Kugi, "Control of nonlinear descriptor systems, a computer algebra based approach", *Nonlinear control in the year 2000*, I. Alberto *et al.* Ed., Vol. 2. London: Springer. *Lect. Notes Control Inf. Sci.* **259**, pp. 379–395, 2001.
M.T. Soylemez and N. Munro, "Pole assignment and symbolic algebra: A new way of thinking", in *Proc. Control'98*, Swansea, pp. 1306–1310, UK: IEE, 1998.
M.T. Soylemez and I. Ustoglu, "Block diagram reduction using symbolic algebra", in *Proceedings of the 12th Mediterranean Conference on Control and Automation* (paper no. 1140), Kusadasi, Turkey, June 6–9, 2004.
M.T. Söylemez and I. Ustoglu, "Designing Control Systems using exact and symbolic manipulations of formulae", *International Journal of Control*, 79, within this special issue, 2006.
D.R. Stoutemyer, "Computer algebra for the calculus of variations, the maximum principle, and automatic control", *Int. J. Comput. Inform. Sci.*, 8, pp. 419–434, 1979.
T. Svensson, "Nonlinear control systems analysis using computer algebra", *Proceedings of the American Control Conference*, 3, pp. 3084–3088, 1993.
A. Varga and G. Looye, "Symbolic and numerical software tools for LFT-based low order uncertainty modeling", *Proc. CACSD'99 Symposium*, Kohala Coast, Hawaii, pp. 1–6, 1999.
X. Zheng, A.C. Zolotas and H. Wang, "Mathematica Implementation of Output-Feedback Pole Assignment for Uncertain Systems via Symbolic Algebra", *International Journal of Control*, 79, within this special issue, 2006.