RaPID: THE END OF HEURISTIC PID TUNING

Peter Van Overschee, Christiaan Moons, Wim Van Brempt
Paul Vanvuchelen, Bart De Moor

Abstract: RaPID (Robust Advanced PID Control) is a new and original engineering tool for the design and implementation of optimal PID controllers. It integrates data acquisition, system identification and optimal PID control design. Special care has been taken in emphasizing engineering intuition and requirements as opposed to the mathematical details. The result is an extremely user-friendly design tool which allows for an easy model-based optimization of any PID controller.

Keywords: PID tuning, System identification, Optimal control, Real-time data acquisition

1. INTRODUCTION

RaPID (Robust Advanced PID Control) is a transportable computer system driven by user-friendly software which provides the control engineer with a tool to tune his controllers optimally. With **RaPID**, the time of heuristic PID tuning, largely based upon "fingerspitzengefuhl", is over.

PID controllers are the typical controllers at the lowest local level in industrial distributed control systems. Estimations indicate that more than 90 % of all controllers in the industry are PID. However in spite of the "lookup" tuning rules that have been around for more then 40 years [1,2] and the recent auto-tuning methods [3,4] a fraction of about 80 % is badly tuned and operates inappropriately or at least not optimally [5,6,7].

We believe this surprizing fact originates in the typical limitations of auto-tuners and tuning rule control design strategies. These methods indeed break down for three major reasons:

- The experiments are typically limited to simple step-like, open-loop experiments.

- Through graphical inspection, a two or three parameter model is then obtained. These simple experiments and models are often too limited to capture all relevant dynamics of the process.

- The control design based on these limited parameter models are straightforward strategies which allow no flexibility in the engineering specifications and present no alternative when they lead to unsatisfactory (unstable !) results.

RaPID solves these problems in a very elegant way:

- The data acquisition unit (Section 2) allows the engineer to apply, in addition to the simple traditional step-like signals, more sophisticated excitation signals that extract much more dynamic information from the system.

- In the identification unit (Section 3), the measured signals then serve as ingredients to automatically determine a mathematical optimal model, capturing all the relevant process dynamics.

- Finally, the control design unit (Section 4) finds optimal control parameters, based on engineering criteria such as overshoot, rise-time, settling time, load and/or disturbance rejection, energy consumption, robustness, maximal input signals, etc. Unlike the solutions of other packages, the final solution is not based on heuristics but takes into account the given engineering specifications.

- The designed controller can then be implemented and tested in the data acquisition unit with just one push of a button (Section 5).

The application areas of **RaPID** include (among others) the chemical and petro-chemical industry, the process, metallurgy and pulp & paper industry. However, it is also extremely well suited for fast and oscillatory processes in mechanical, electronic, mechatronic, hydraulic and pneumatic applications (upto 40 kHz sampling rate).

RaPID can be integrated in Matlab as a toolbox or runs as a stand alone application on a PC (Windows 95) or a workstation.

2. DATA ACQUISITION

The first step in the design of a model based controller is the collection of data. Hereto, the process input has to be excited with an actuator signal. The response of the process output variable to this excitation is subsequently recorded. In what follows, we describe the data acquisition window of **RaPID** (Figure 1), designed to facilitate this measurement process.

Figure 1: Data acquisition window. After the input is designed (bottom plot) it is applied to the process and the output is measured (top plot). The data is collected through file transfer (from DCS or PLC), from a Simulink model or from the **RaPID** real-time interface supporting sampling rates up to 40 kHz.

2.1 Input Signal Design

The input signal design window (Figure 2) allows for an easy design of various types of excitation signals. Apart from the more classical step, pulse, block and ramp like experiments, there is also a possibility to apply more sophisticated signals [8] as: Polynomial steps with limited speed and accelerations, Random (possibly filtered) noise, Pseudo Random Binary Noise Sequences (PRBNS), Swept sine and Multi sine signals. The main advantage of these signals over the classical experiments is that they can inject more energy into the system at a lower amplitude. Consequently, the system has to be less disturbed from its normal operating point to obtain the same results.

2.2 Measuring data

Once the input has been designed it has to be applied to the system. **RaPID** allows for three possible ways to do this:
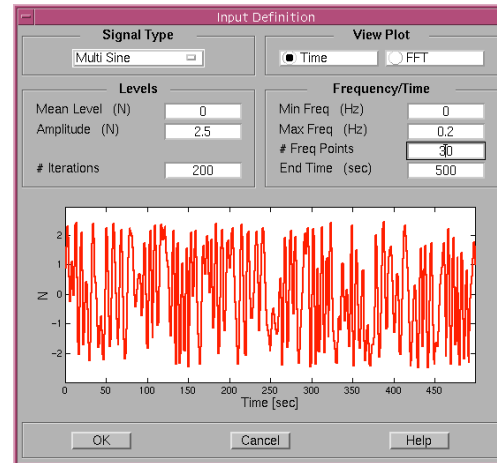

Figure 2: Input definition window including the design of Steps, Blocks, Pulses, Ramps, Polynomials, Random (filtered) signals, PRBN Signals, Swept Sines and Multi Sines. With a touch of a button the frequency contents of the signal can also be inspected. The more advanced signals allow to inject energy into the system at a lower amplitude.

File: The designed signal is saved in a file, transported on floppy, and then applied to the process (for instance with a DCS or PLC). Afterwards the measured signals are read back into **RaPID**. This mode can also be used to load previously measured data (Matlab or ASCII files).

Simulink: When **RaPID** is running as a toolbox in Matlab [9], physical process models implemented in Simulink can be used. **RaPID** has a seamless connection to Simulink, and with a simple hit of the "Start" button (Figure 1) the designed input signal is applied to the Simulink model, and the simulated data is acquired automatically.

Real Time: The third mode in which **RaPID** can collect data is through a Windows 95 integrated real time system developped especially for **RaPID**. In this case, a data acquisition board is plugged into the PC and is driven directly by the **RaPID** hard real-time software. The sampling rate can be up to 40 kHz, without loss of speed of **RaPID** (or any other Windows application) running simultaneously. In this mode, any process can be hooked up directly to the PC. Measurements (open and closed loop - see Section 5) are just a touch of the "Start" button away.

2.3 Viewing data

Once the data is measured, they can be visualized, zoomed into and inspected by just clicking with the mouse on the desired data point (these are features available for all **RaPID** plots). The data can also be analyzed for its minimal, maximal and mean value and its standard deviation.

3. IDENTIFICATION

The measured data is now used to derive a linear model for the process. The identification part of **RaPID** first pre-processes the data and then automatically computes a dynamical model which trades off model-fit versus model-complexity (Figure 3).
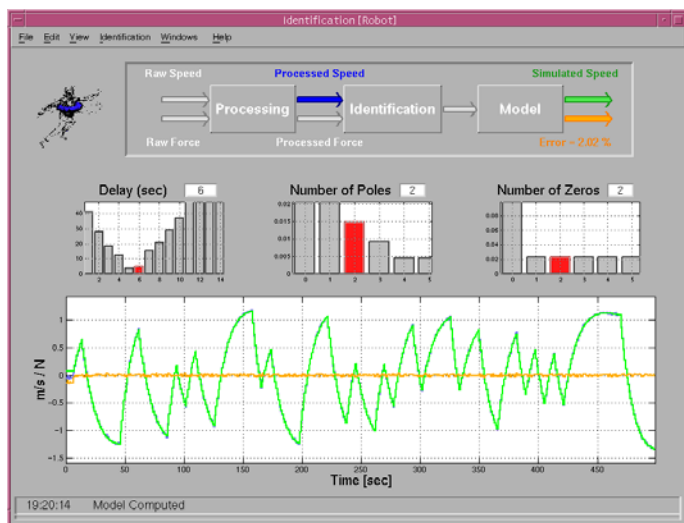

Figure 3: Identification window. The different identification steps are shown at the top. After the pre-processing, a model is identified and simulated. The bar plots in the middle indicate the selected structure (delay, number of poles and zeros). The preprocessed data, the simulated data and the error are shown in the bottom plot.

3.1 Pre-processing

The data can be made suitable for linear system identification with the pre-processing window (Figure 4). This includes truncation, detrending, filtering (low, high and band pass) and the optional a-priori inclusion of an integrator.

3.2 Identification

Once the data is suitably pre-processed, a model is identified (Figure 3). Important steps are the model structure selection and the model computation.
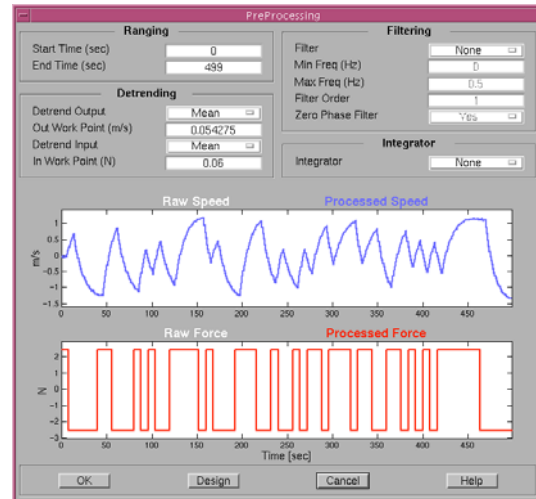

Figure 4: Pre-processing window with ranging, detrending, filtering and the possible inclusion of an integrator.

The model structure is composed of the input-output delay, the number of poles and number of zeros in the process. **RaPID** automatically makes a default choice, but the user can easily modify these parameters by just clicking on the bar plots in the middle of Figure 3.

The model is computed through a mixture of prediction error methods and subspace identification, two state of the art digital identification techniques [10,11]. The computation itself is implemented with fast NAG and LAPACK algorithms [12,13].

The resulting model is then used to simulate the data and is compared against the original data (Figure 3, bottom plot). Based on this comparison, the engineer can decide on an appropriate model structure. The time and frequency domain properties of the model can also be easily inspected in a separate window.

Advanced identification features include the possibility to fix DC gains and/or speeds, to change the stability radius of the identified model, to do a parameter sweep and to specify the model structure in even more detail.

4. PID CONTROL DESIGN

The identified model is now used for control design. First the control specifications and control structure have to be entered. Next, based on the identified model, an optimal controller is computed.
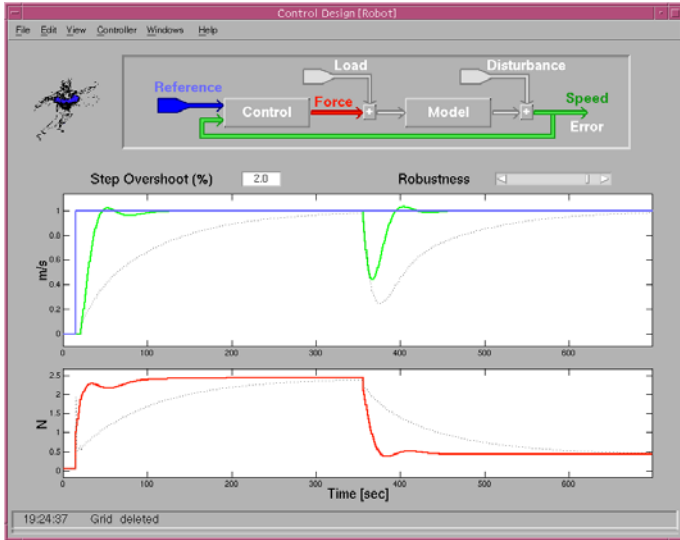
4.1 Controller Design

Figure 5: Controller design window. Response to step reference and step load disturbance (in the middle of the time span). The top plot shows the reference and output signal, the bottom plot the applied input signal. The thick line is the **RaPID** design. For comparison the Ziegler Nichols design is also shown as a dotted line.

Within **RaPID** the controller specifications can be entered easily and intuitively (Figure 5). The controller design has two major modes:

Tracking: The controller is mainly used for the accurate tracking of reference signals[1]. The engineer specifies the desired overshoot, robustness, maximal input amplitude and noise amplification. Furthermore, he can also select the rise time or settling time to be optimized (Figure 6).
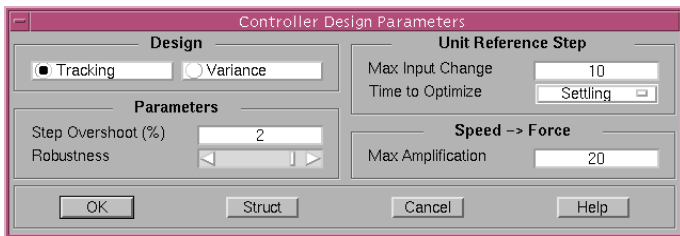

Figure 6: Controller design parameters. With the tracking design selected, the engineer has to enter the desired overshoot, robustness, maximal input change, time to optimize and noise amplification.

---

[1]Note that the reference signals can be generated in a similar manner as the input signals in Figure 2.

Variance: In this case the controller is mainly intended for the suppression of input load or output disturbance signals. The engineer chooses to minimize the settling time, the output energy or the overshoot due to a step-like load or output disturbance.

The tuning of the controller is based on a non-linear Sequential Programming Problem (SQP), combined with other heuristic optimization techniques [12,13] and leads to extremely well tuned PID controllers, in a neglible optimization time. By taking the robustness of the controllers into account in the design stage, they even perform well when the identified linear model deviates slightly from the real (possibly non-linear) process.

4.2 Controller Structure

RaPID contains a whole range of possible PID structures. Proportional, Integral, Derivative and Double Derivative terms can be combined in many different ways. The resulting structures include (among others) PI, PD, PID or even PIDD (double derivative) controllers.
One important difference with the classical PID structures is the standard inclusion of a feedforward term (Figure 7). The split between feedforward and feedback (two degree of freedom controller) has the major advantage that tracking and variance can be tuned independently. So, even though one mode (tracking or variance) is chosen in the controller design window (Figure 5), the two degree of freedom controller will perform well for both tracking and variance.
Note however that, when desired, a one degree of freedom controller can also be computed.
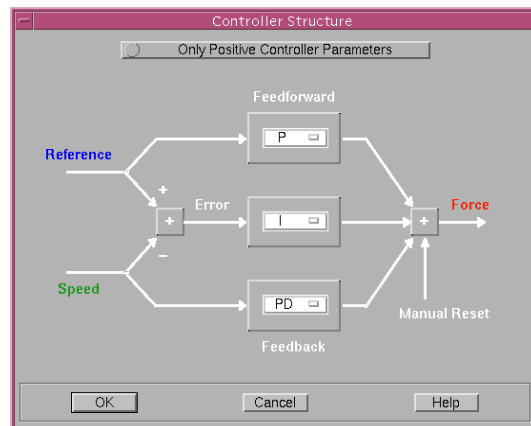

Figure 7: The top part is the reference feedforward, the bottom part the output feedback, and in the middle the common part is displayed. With the pop-up menu's the structure can be adjusted to almost any type of PID controller.

4.3 Additional features

The designed controller can easily be compared to the classical designs of [1,2,4] which are also implemented in **RaPID** (see also Figure 5).
Furthermore, the properties of the resulting closed loop system, such as time and frequency responses can be easily inspected. Complete confidence can thus be obtained before the controller is implemented in reality.
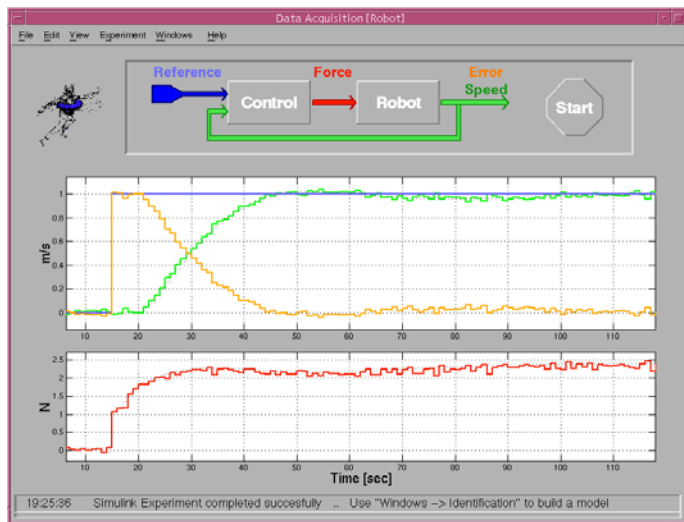
5. CONTROL IMPLEMENTATION

Figure 8: Data acquisition window in closed loop configuration. The top plot shows the step reference, the true (measured) closed loop output and the true error. The bottom plot is the actuating signal. Once again the Simulink or real-time experiment is only a push of the "Start" button away.

Once the controller has been computed it can be implemented in the data acquisition window (Figure 1), switched to closed loop operation (Figure 8). Depending on the mode **RaPID** is working in, the controller is implemented as follows:

File: If the controller has to be implemented in an external device supporting C, (pseudo)code is automatically generated to facilitate the implementation. The controller can also be automatiaclly converted to continuous time for implementation in DCS or PLC.

Simulink: The seamless Simulink interface discussed in Section 2 also allows for the implementation of the controller around the Simulink model. In this way, instantaneous closed loop simulations are performed on the physical process model.

Real Time: The **RaPID** real-time interface running under under Windows 95 supports the hard real-time implementation of the designed controller up to 40 kHz. No explicit down-loading is needed, and closed loop experiments are only a "Start" button away. A VME measurement system is also available.

All controllers are implemented with anti-windup and bumpless transfer.

6. CONCLUSIONS

RaPID is an integrated tool for the design and implementation of optimal, advanced PID controllers. Its main functionality is data acquisition (from file, Simulink or real-time) includeing design of experiments and controller implementation, system identification and control design.
Its flexibility and user-friendliness make it the ideal tool for optimization of industrial PID controllers.
Future extensions include fuzzy, neural and model predictive multivariable control. Direct comminucation with industrial control systems will soon be possible through CAN, Profibus or Serial Communication.
More information about **RaPID** and an interactive demo can be found on the website http://www.ismc.be.

ACKNOWLEDGEMENTS

REFERENCES

1. J.G. Ziegler and N.B. Nichols, "Optimum settings for automatic controllers", Trans, ASME, 64: 759-768, 1942.
2. K.L. Chien, J.A. Hrones and J.B. Reswick, "On the automatic control of generalized passive systems", Trans, ASME, 74: 175-185, 1952.
3. "Auto-tuned Control", Control and Instrumentation, 3: 33-37, 1990.
4. K.J. Aström and T. Hägglund, "PID Controllers: Theory, Design and Tuning", Instrument Society of America, Research Triangle Park, NC, U.S.A., 1995.

5. S. Yamamoto and I. Hashimoto, "Present status and future needs: The view from Japanese industry", in Arkun and Ray, Eds., Chemical Process Control-CPCIV. Proceedings of the Fourth International Conference on Chemical Process Control, Texas.

6. W.L. Bialkowski, "Dream versus reality: A view from both sides of the gap", Pulp and Paper Canada, 94:11, 1993.

7. D.B. Ender, "Process control performance: Not as good as you think", Control Engineering 40:10, pp. 180-190.

8. J. Schoukens and R. Pintelon, "Identification of Linear Systems: a Practical Guideline to Accurate Modeling", Pergamon Press, London, UK, 1991.

9. "Matlab Version 5", The Mathworks Inc., MA, U.S.A., 1996.

10. L. Ljung, "System Identification - Theory for the User", Prentice Hall, Englewood Cliffs, NJ, 1989.

11. P. Van Overschee and B. De Moor, "Subspace Identification for Linear Systems", Kluwer Academic Publishers, 1996.

12. "NAG C Library Version 4", Numerical Algebra Group Ltd, Wilkinson House, Oxford, UK, 1996.

13. E. Anderson, Z. Bai, C. Bischof and J. Demmel "LAPACK Users' Guide", Society for Industrial and Applied Mathematics, Philadelphia, 1995.

Peter Van Overschee

Katholieke Universiteit Leuven, ESAT/SISTA, Kardinaal Mercierlaan 94, 3000 Heverlee.
Tel: 016/321161 - Fax: 016/321970
peter.vanoverschee@esat.kuleuven.ac.be
http://www.ismc.be

Peter Van Overschee was born in Leuven, Brabant, Belgium on October 28, 1966. He obtained his degree of Electro-Mechanical Engineering in control theory in 1989 at the Katholieke Universiteit Leuven, Belgium. In 1990 he received his Master of Science in Electrical Engineering at Stanford University, California, USA. At the Katholieke Universiteit Leuven, he obtained his doctoral degree with the work "Subspace Identification: Theory - Implementation - Applications" in 1995. In 1994, he won the biannual Belgian Siemens Award and in 1996 the tri-annual best paper award of Automatica, a journal of the International Federation of Automatic Control (at the IFAC World Congress, San Francisco, 1996)
His current research interests are the theory and application of multivariable system identification. He was also involved in the design, development and implementation of the identification toolbox for Xmath (Integrated Systems Inc., Santa Clara, USA).

Christiaan Moons

Intelligent System Modeling and Control NV, Kardinaal Mercierlaan 94, 3000 Heverlee.
Tel: 016/321791 - Fax: 016/321970
christiaan.moons@ismc.be
http://www.ismc.be

Christiaan Moons was born in Lier, Belgium on November 6, 1966. He obtained his degree of Electro-Mechanical Engineering in control theory in 1990 at the Katholieke Universiteit Leuven, Belgium. He is a member of the board of administrators of ISMC NV (Intelligent Systems, Modeling and Control), a spin-off company specialized in modeling and control of multivariable industrial processes and mechatronics analysis and design.
Currently he is design manager of ISMC NV, where he is involved in the design of real time extensions of windows 95 and windows NT, and the hardware for rapid.

Wim Van Brempt

Intelligent System Modeling and Control NV, Kardinaal Mercierlaan 94, 3000 Heverlee.
Tel: 016/321095 - Fax: 016/321970

wim.vanbrempt@ismc.be
Wim Van Brempt obtained his degree of Electro-Mechanical Engineering in 1992 at the Katholieke Universiteit Leuven. At the same institute he obtained a postgraduate degree in Biomedical Engineering in 1994. Till 1996 he was involved in the coordination of an Automation Postgraduate and in the development of a department on Automation at the COR University in Ibague, Colombia - South-America. In 1997 he joined ISMC where his main activity lies in the modeling area.

Paul Vanvuchelen

Intelligent System Modeling and Control NV, Kardinaal Mercierlaan 94, 3000 Heverlee.
Tel: 016/321095 - Fax: 016/321970
paul.vanvuchelen@ismc.be
http://www.ismc.be
Paul Vanvuchelen was born in Genk, Limburg, Belgium on May 4, 1968. He obtained his degree of Electro-Mechanical Engineering in mechatronics in 1991 at the Katholieke Universiteit Leuven, Belgium. At the Katholieke Universiteit Leuven, he obtained his doctoral degree with the work "Virtual Engineering for Design and Control of Continuously Variable Transmissions" in 1997. He is general manager, senior consultant and member of the board of administrators of ISMC NV (Intelligent Systems Modeling and Control), a spin-off company of the K.U.Leuven, providing engineering services, software tools and systems for dynamic design and control.

His main interests are in the optimal use of physical modelling techniques and multivariable control algorithms, in order to solve dynamic design and control problems in industry.

Bart De Moor

Katholieke Universiteit Leuven, ESAT/SISTA, Kardinaal Mercierlaan 94, 3000 Heverlee.
Tel: 016/321715 - Fax: 016/321970
bart.demoor@esat.kuleuven.ac.be
http://www.ismc.be
Bart Lodewijk René De Moor was born in Halle, Brabant, Belgium on July 12, 1960. He received his doctoral degree in Applied Sciences in 1988 at the Katholieke Universiteit Leuven, Belgium. He was a visiting research associate (1988-1989) at the Departments of Computer Science (Gene Golub) and Electrical Engineering (Thomas Kailath) of Stanford University, California, USA. Since October 1989

he is a senior research associate of the Fund for Scientific Research Flanders (FWO-Vlaanderen) and associate professor at the Department of Electrical Engineering of the Katholieke Universiteit Leuven.

His research interests include numerical linear algebra (generalized svd, complementarity problems, structured total least squares, tensor algebra), system identification (subspace methods,structured total least squares), control theory (robust control, neural nets) and signal processing. He has more than 200 papers in international journals and conference proceedings. He received the Leybold-Heraeus Prize (1986), the Leslie Fox Prize (1989), the Guillemin-Cauer Best Paper Award (1990) of the IEEE Transactions on Circuits and Systems, the bi-annual Siemens prize (1994), the tri-annual best paper award of Automatica, a journal of the International Federation of Automatic Control (at the IFAC World Congress, San Francisco, 1996) and became a Laureate of the Belgian Royal Academy of Sciences (1992). He is a member of several boards of administrators of (inter)national scientific, cultural and commercial organizations, including ISMC NV (Intelligent Systems, Modeling and Control).

In 1991-1992 he was the chief of staff of the Belgian federal minister of science Ms.Wivina Demeester-DeMeyer and later on of the Belgian prime minister Wilfried Martens. Since 1994 he is the main advisor on science and technology policy of the Flemish minister-president Luc Van den Brande.