

Remote System Identification in the “Automatic Control Telelab” Environment

Marco Casini, Andrea Garulli, Domenico Prattichizzo, Antonio Vicino
Dipartimento di Ingegneria dell’Informazione
Università di Siena
Via Roma 56, 53100 Siena, Italy
{casini,garulli,prattichizzo,vicino}@ing.unisi.it

Abstract—This paper describes recent improvements of the Automatic Control Telelab (ACT), a remote laboratory of automatic control developed at the University of Siena. In particular, it focuses on a new tool which allows students to perform system identification experiments on remote processes by means of both classical (statistical) and set-membership (deterministic) techniques. Through this machinery, students can identify a suitable dynamic model of a process, design a controller and then test its behaviour remotely. The ACT home page is: <http://www.dii.unisi.it/~control/act>.

I. INTRODUCTION

Tele-laboratories are instances of the more general distance education problems which is attracting a wide attention in the academic and government communities. Automatic Control is one of the technical areas which most exploited the new technologies to develop new tools for distance learning [1]. A thorough treatment about control education by means of web technologies has been recently reported in [2].

Web-based laboratories are divided in two classes: virtual labs and remote labs. The main difference between them is that virtual labs allow one to remotely run simulations with possible animations of the controlled system [3], [4], [5], while remote labs are laboratories where students can remotely interact with real experiments. In remote labs, users can change control parameters, run the experiment, see the results and download data through a web interface. This is for instance the case of [6], where a remote lab for testing analog circuits is described. In [7], a remote chemical control process is implemented and in [8] several laboratory experiments are available.

The Automatic Control Telelab (ACT) developed at the University of Siena is an example of remote laboratory and is attracting the interest of many students from our campus and from other national and international institutions [9], [10].

The complexity of designing the hardware and software architectures of the remote laboratory dramatically increases when remote design of the controller through the Internet must be embedded in the system. This is for instance the case of the remote lab developed at the College of Engineering of Oregon State University [11], [12]. Another approach has been investigated in [13], where the user can run an experiment using a controller which runs not only on the server but also on the client, compiling and executing it on the user machine. In this case, some issues about network

reliability and delays have been addressed. The audio and video feedback is very important to increase the effect of telepresence, as shown in [14] where authors implemented a pan, tilt and zoom control of cameras grabbing the experiments of the remote lab.

Regarding the development system, a remote lab can be designed using a well-known software environment, such as LabVIEW [15] or Matlab/Simulink [16], [17], or it can be implemented through a new and on purpose designed software, as shown in [18] where a flow process is analyzed. One of the key features of the Automatic Control Telelab (ACT) is that students can design their own controllers through the Matlab/Simulink environment. This feature allows a remote user to synthesize his/her own controller without learning any special language other than the Matlab/Simulink software. It is the authors’ opinion that using standard tools like Matlab/Simulink will dramatically encourage the practice of control theory through remote labs.

The aim of this paper is to present a new development of the ACT: the “remote identification” environment. In this part of the ACT structure, students can perform system identification experiments on remote processes through the Internet. Students can choose which input to apply among a set of predefined inputs, or can design their own input signal in the Simulink environment in a very easy way. Once the experiment is performed, by means of a graphical user interface students can choose a model class and estimate the best model within the class, either in a classical statistical context [19], [20] or in a set-membership framework [21]. To the best of our knowledge, no other lab allows users to remotely identify a dynamic model of the process before controlling it. Understanding identification is a very important phase of the control education process and we believe that this new tool will consistently increase the distance learning capabilities of the ACT.

The paper is organized as follows. Section 2 illustrates the main features of the ACT. In Section 3 the new system identification feature is described, whereas in Section 4 the identification procedure is analyzed in depth. Remote sessions are reported in Section 5, while conclusions are drawn in Section 6.

II. THE AUTOMATIC CONTROL TELELAB

The Automatic Control Telelab is a remote laboratory mainly intended for educational purpose, and since 1999 it has been used in control systems classes [9], [22]. The aim of the project was to allow students to put in practice their theoretical knowledge of control theory in an easy way and without restrictions due to laboratory opening time and processes availability. The ACT is accessible 24 hours a day from any computer connected to the Internet by means of any common browser, e.g. Netscape Navigator or Microsoft Internet Explorer. No special software or plugin is required. Users can design new controllers by means of the Matlab/Simulink environment. However, if Matlab is not available on the local computer, the user can still run experiments by choosing among the available predefined controllers. A live video window is provided for each remote experiment session.

One of the main features of the ACT is the possibility of integrating any user-defined controller in the control loop of the remote process. The controller synthesis is based on the Matlab/Simulink environment. Since Matlab and Simulink packages are standard tools in control systems classes, there are no additional hurdles for a student who wants to design new controllers. In fact, a controller is just a Simulink model, similar to those commonly used to run simulations of dynamic systems.

At present, four processes are available for remote control: a DC motor, a tank for level control, a magnetic levitation system and a two degrees-of-freedom helicopter. The DC motor is used to control the angular position and velocity of a rotating axis. The level control process has been included because, in spite of its simplicity, it shows nonlinear dynamics, whereas the magnetic levitation process, being nonlinear and unstable, exhibits very interesting properties to be analyzed in control theory education. Finally, the two degrees-of-freedom helicopter, being a nonlinear and unstable MIMO system, can be used in graduate control system classes.

One of the problems, common to all the experiments involving hardware devices no matter if they are remote or not, is that each experiment can be accessed by one user at a time. To manage the experiment queue, the ACT web page shows which experiments are available as well as the maximum waiting time needed to access the busy experiments. The ACT home page is: <http://www.dii.unisi.it/~control/act>.

III. REMOTE SYSTEM IDENTIFICATION

The aim of the remote system identification facility is to find a suitable model which approximates the real plant.

From the experiment page, the user can choose the *Identification Experiment* option so that the *Identification Interface* appears as shown in Fig. 1 (here reference is made to the DC motor experiment in Fig. 2).

From this interface the input to the process (i.e. the DC motor voltage) can be selected among a set of predefined

Identification of a DC Motor		
Identification of a DC motor with speed control.		
Personal Data		
Name	Country	Email
<input type="text"/>	<input type="text"/>	<input type="text"/>
Predefined Inputs		
Download Model	Standard Inputs (selectable) Sample Time: 10 ms	Run Experiment
User-defined Inputs		
? Inputs Model	<input type="text"/>	Browse...
? Inputs Data	<input type="text"/>	Browse...
? Sample Time (msec)	<input type="text"/> Range=[10,100]	
? Download Template		Send Model

Fig. 1. The Identification Interface



Fig. 2. Picture of the DC motor

inputs (e.g. white noise, pseudo random binary sequences, sinusoidal and square waves, etc.) or can be designed, according to the user requirements, by using the Simulink environment. In this case, it is only necessary to download a Simulink template model and change the blocks inside the "ACT.INPUT" subsystem in a similar way to that discussed in [9] for user-defined controllers.

After the selection of the input signal, the *Identification Experiment Interface* appears. It is now possible to start the experiment: the user can see the development of input/output signal and observe the experiment through a webcam (see Fig. 3). It is also possible to modify on-line the applied input signal.

At the end of the experiment the *Identification Panel* is displayed (Fig. 4). Through this interface, it is possible to perform several operations to compute the identified model. In the left side of this interface the time plot as well as

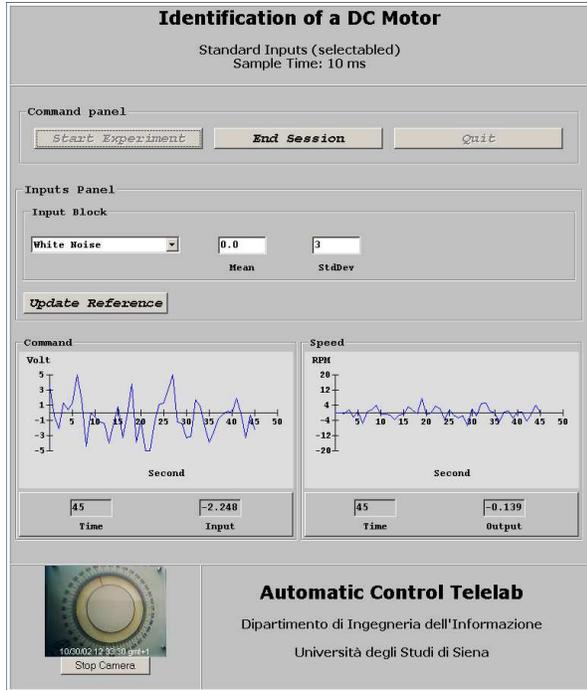


Fig. 3. The Identification Experiment Interface

the periodogram of the input/output signals are shown. At the bottom, it is possible to download the Matlab workspace file containing the experiment signals, in order to perform off-line analysis. The actual identification interface is on the right. It allows the user to choose the desired model structure and the order of its polynomials. Moreover, it is possible to choose to remove mean and trend as well as to choose which data are used to perform identification and which for validation purposes. Once all options are set, it is possible to perform the identification by pressing the *Run Identification Algorithm* button.

At the end of the identification procedure, results are shown in the *Identification Result Interface*, depicted in Fig. 5 (the figure regards the least-squares identification of an ARX model). In the left side of this interface it is possible to see the fitting plot, that is the comparison between the validation data and the model simulated output, and the frequency response of the estimated model. In the right side, the coefficients of the identified model are reported as well as three benchmark functions (Loss function, FPE and FIT, see [19]). The user can now decide to quit the identification session or to try a new model on the same data set.

IV. IDENTIFICATION PROCEDURES

The ACT allows the identification of models in both a statistic and a deterministic context. Regarding the statistic framework, it is possible to choose among *ARX*, *ARMAX*, and *OE* model classes (see [19]), while in the set-membership context *ARX* models are considered. In the following a brief

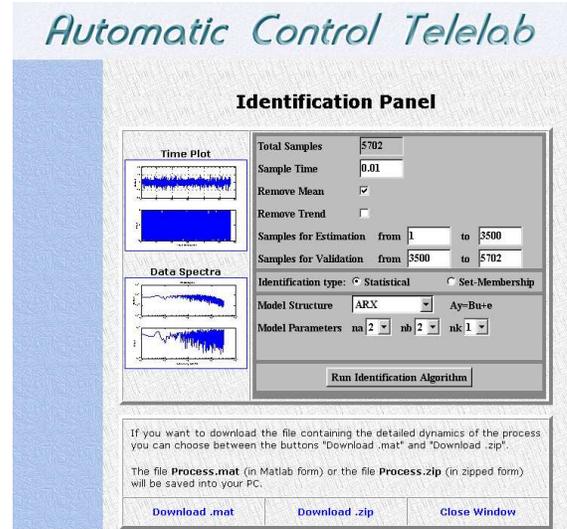


Fig. 4. The Identification Panel

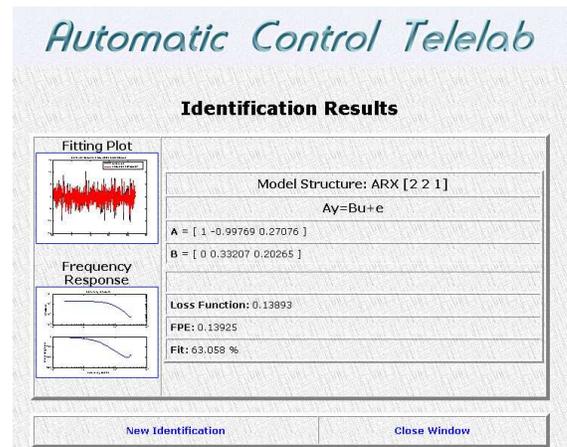


Fig. 5. The Identification Result Interface

description of ARX models identification is reported for both methods.

An ARX model is given by

$$y_k = - \sum_{i=1}^{n_a} a_i y_{k-i} + \sum_{i=1}^{n_b} b_i u_{k-i-n_k+1} + e_k \quad (1)$$

where u_k is the input, y_k is the output and e_k is the noise. The aim of identification is to estimate the parameters a_i, b_i on the base of the knowledge of a set of input/output data $\{u_1, y_1, \dots, u_N, y_N\}$.

The parameter estimation is performed in different ways, depending on the assumptions that are made on the noise e_k . In the classical statistical setting, noise is assumed to be a white stochastic process with zero mean, and a suitable norm of the prediction error associated to the chosen model class is minimized. Conversely, in the set-membership framework it is supposed that the noise is unknown-but-bounded: this

leads to the definition of a *set* of feasible parameter vectors which represents the parametric uncertainty associated to the estimated model.

A. Statistical identification

Equation (1) can be rearranged in the regression form

$$y_k = \phi_k \theta + e_k \quad (2)$$

where $\phi_k = [-y_{k-1}, \dots, -y_{k-n_a}, u_{k-n_k}, \dots, u_{k-n_k-n_b+1}]$, and $\theta = [a_1, \dots, a_{n_a}, b_1, \dots, b_{n_b}]'$ is the parameter vector to be estimated. Prediction error methods rely on the minimization of a suitable norm of the one-step prediction error associated to the ARX model, i.e. $\varepsilon_k = y_k - \phi_k \theta$. If the ℓ_2 norm is chosen, one has to compute

$$\min_{\theta} \sum_{k=1}^N \varepsilon_k^2$$

which leads to the popular *least squares* estimate

$$\theta_{LS} = \left(\sum_{k=1}^N \phi_k' \phi_k \right)^{-1} \sum_{k=1}^N \phi_k' y_k.$$

The properties of this estimate have been widely studied in the statistical context, and in the more general framework of quasi-stationary processes (see [19] for a thorough treatment).

In the *ACT Identification Panel*, when the ‘‘Statistical’’ identification type is selected, it is possible to choose the model structure ARX and to insert the number of parameters n_a , n_b , and the delay n_k . The same can be done for other model structures such as ARMAX and OE. The routines that the ACT uses to compute the optimal models in the statistical context are those implemented in [20].

B. Set-membership identification

The *set-membership* or *bounded-error* approach to parametric system identification is by now well settled (see e.g [23], [21], [24]). The main assumption is that the noise e_k is bounded in some norm. A typical case is that of ℓ_∞ -bounded noise, i.e.

$$|e_k| \leq \delta \quad \forall k. \quad (3)$$

From (2)-(3), one has that the set of all parameter vectors compatible with the available data and the chosen model structure is given by the *feasible parameter set*

$$\Theta = \{ \theta : |y_k - \phi_k \theta| \leq \delta, k = 1, \dots, N \}.$$

which is a polytope in the parameter space.

In the set-membership setting, two types of estimators are considered: *set estimators*, which give approximations of the feasible set Θ , and *pointwise estimators*, that select one parameter vector optimizing some functional related to Θ (see [21]). In the following, a set estimator and two point estimators are presented, which can be computed by the *ACT Identification Panel*.

Minimum outer box. This set estimator provides the minimum box containing the feasible set Θ . It requires the solution of $2n$ linear programming problems, given by

$$\begin{cases} \sup \theta_i \\ \inf \theta_i \end{cases} \quad i = 1, \dots, N$$

subject to $\theta \in \Theta$, which can be rewritten as

$$\begin{bmatrix} F \\ -F \end{bmatrix} \theta \leq \begin{bmatrix} y + \delta \mathbf{1} \\ -y + \delta \mathbf{1} \end{bmatrix},$$

where $y = [y_1, \dots, y_N]'$, $\mathbf{1} = [1, \dots, 1]'$ and

$$F = [\phi_1 \ \phi_2 \ \dots \ \phi_N]'$$

Central algorithm in ℓ_∞ norm. The central algorithm in ℓ_∞ norm is defined as the geometric center of the minimum box containing Θ . Therefore, it is the solution of the optimization problem

$$\theta_C = \arg \min_{\theta} \max_{\bar{\theta} \in \Theta} \|\theta - \bar{\theta}\|_\infty.$$

Projection algorithm in ℓ_∞ norm. An alternative pointwise estimate is given by the ℓ_∞ projection algorithm, defined as

$$\theta_P = \arg \min_{\theta \in \Theta} \|y - F\theta\|_\infty. \quad (4)$$

To compute θ_P , it is sufficient to solve one linear programming problem, i.e.

$$\theta_P = \arg \min_{\theta} \gamma$$

subject to

$$\begin{bmatrix} F & -1 \\ -F & -1 \end{bmatrix} \begin{bmatrix} \theta \\ \gamma \end{bmatrix} \leq \begin{bmatrix} y \\ -y \end{bmatrix}.$$

Note that the projection estimate θ_P requires a much lower computational effort than θ_C . Moreover, it does not depend on the bound δ in (3). Notice also that the least-squares estimate θ_{LS} corresponds to the ℓ_2 projection algorithm, which is obtained by using the ℓ_2 norm in (4) instead of the ℓ_∞ one (see [21] for details).

In the *Identification Panel*, when the ‘‘Set-membership’’ identification type is selected, it is possible to indicate the desired ARX model class and to choose between the two pointwise identification algorithms outlined above. In particular, when the central estimate θ_C is chosen, the ACT returns also the minimum outer box representing a (conservative) estimate of the parametric uncertainty associated to θ_C .

V. EXPERIMENTAL RESULTS

In this section, experimental results on remote identification of the DC motor are reported. Six different ARX models have been estimated, in both deterministic and statistic methods. Equation (1) has been rearranged as follows:

$$Ay = Bu + e \quad (5)$$

where $A = [1 \ a_1 \ a_2 \ \dots \ a_{n_a}]$, $B = [b_1 \ b_2 \ \dots \ b_{n_b}]$ and $y = [y_k \ y_{k-1} \ \dots \ y_{k-n_a}]'$, $u = [u_{k-n_k} \ \dots \ u_{k-n_k-n_b+1}]'$, $e = e_k$, being $k = \max\{n_a, n_b\} + 1, \dots, N$ (the first $\max\{n_a, n_b\}$ data have been used to initialize the algorithm).

The experiment has been performed using as input a Pseudo Random Binary Sequence (PRBS) satisfying $|u_k| \leq 5$. The experiment lasted 30 seconds and the sample time was set to 0.01 second. The first 2000 samples acquired were used for estimation while the last 1000 were used for validation of the obtained model. In particular, as validation index, the FIT on the validation data has been computed (for a deeper treatment see [19]). The FIT index is given by

$$FIT = \left(1 - \frac{\|\hat{y} - y\|_2}{\|y - \bar{y}\|_2}\right) \cdot 100 \quad (6)$$

where \hat{y} is the estimated output while \bar{y} is the output sample mean [20].

Table I shows the parameter vectors obtained by using the central algorithm in ℓ_∞ norm. The noise bound δ has been chosen as the smallest integer between 1 and 10 for which the feasible set turns out to be not empty, i.e. the LP problems admit a solution. Note that higher values of the noise bound δ for some models mean that the model is not appropriate to estimate the system. From Table I it ensues that the model which appears to be the most appropriate has $n_a = 2$ and $n_b = 2$.

Table II reports the bounds relative to the model parameters provided by the central algorithm. For example, the box containing the parameters for the case $n_a = 1$ and $n_b = 1$ is $A = [1 \ 0.39266 \pm 0.13082]$ and $B = [0 \ 0.39266 \pm 0.29905]$.

Note that for $n_a = 2$, $n_b = 2$ the resulting estimate uncertainty box is much smaller.

In Table III, the parameter vectors obtained applying the ℓ_∞ projection algorithm are reported. Again in this case, the best model has $n_a = 2$ and $n_b = 2$.

Time plots regarding the comparison between validation data and the model simulated output are reported in Fig. 6 for the central algorithm, concerning the case $n_a = 1$, $n_b = 1$ and $n_a = 2$, $n_b = 2$.

Regarding statistical identification, Table IV shows the estimated parameters vectors and the FIT index provided by the least-squares algorithm. Note that also in this case the best model is the one with $n_a = 2$ and $n_b = 2$.

TABLE I
IDENTIFICATION RESULTS (CENTRAL ESTIMATOR)

n_a	n_b	δ	A	B	Fit
1	1	5	[1 -0.80501]	[0 0.39266]	43.87 %
1	2	2	[1 -0.76413]	[0 0.39233 0.424]	82.26 %
2	1	3	[1 -1.3505 0.48092]	[0 0.39038]	76.57 %
2	2	1	[1 -1.0447 0.24862]	[0 0.40592 0.31999]	87.80 %
1	3	2	[1 -0.73955]	[0 0.4 0.452 0.1348]	86.03 %
3	1	3	[1 -1.44 0.867 -0.28]	[0 0.42202]	71.43 %

TABLE II
BOUNDS ON THE PARAMETERS (CENTRAL ESTIMATOR)

n_a	n_b	Error bounds on a_i	Error bounds on b_i
1	1	[0.13082]	[0.29905]
1	2	[0.069911]	[0.10384 0.11233]
2	1	[0.46552 0.38543]	[0.15696]
2	2	[0.09634 0.092721]	[0.04023 0.053808]
1	3	[0.11701]	[0.20282 0.22117 0.22784]
3	1	[0.63177 0.97816 0.48079]	[0.23269]

TABLE III
IDENTIFICATION RESULTS (PROJECTION ESTIMATOR)

n_a	n_b	A	B	Fit
1	1	[1 -0.76365]	[0 0.38011]	38.841 %
1	2	[1 -0.7594]	[0 0.38652 0.40863]	80.866 %
2	1	[1 -1.234 0.42702]	[0 0.38795]	62.185 %
2	2	[1 -1.0843 0.28293]	[0 0.40737 0.3071]	87.014 %
1	3	[1 -0.7432]	[0 0.389 0.441 0.121]	86.157 %
3	1	[1 -1.726 1.011 -0.201]	[0 0.43074]	51.725 %

TABLE IV
IDENTIFICATION RESULTS (LEAST SQUARES ESTIMATOR)

n_a	n_b	A	B	Fit
1	1	[1 -0.89271]	[0 0.42607]	49.357 %
1	2	[1 -0.78973]	[0 0.4098 0.4153]	81.087 %
2	1	[1 -1.408 0.58263]	[0 0.41881]	72.787 %
2	2	[1 -1.0304 0.24309]	[0 0.41081 0.31192]	88.173 %
1	3	[1 -0.72946]	[0 0.412 0.436 0.121]	87.251 %
3	1	[1 -1.609 1.058 -0.336]	[0 0.41896]	80.385 %

VI. CONCLUSIONS

The remote identification facility of the Automatic Control Telelab has been presented. This tool allows students to perform system identification experiments on remote processes through statistical and deterministic techniques. Comparisons between the two methods can be made, and the identified models can be used to remotely design a controller and to test it on the real plant. It is the authors' opinion that the remote identification facility increases the potentiality of the remote lab of automatic control, which can be used more efficiently in control system classes. Future efforts are directed towards the remote identification of unstable systems (e.g. magnetic levitation and helicopter simulator), which require closed-loop identification methods.

VII. REFERENCES

- [1] S. E. Poindexter and B. S. Heck. Using the web in your courses: What can you do? what should you do? *IEEE Control System*, 19(1):83–92, Feb. 1999.
- [2] S. Dormido. Control learning: present and future. In *15th IFAC World Congress b'02*, Barcelona, July 2002.

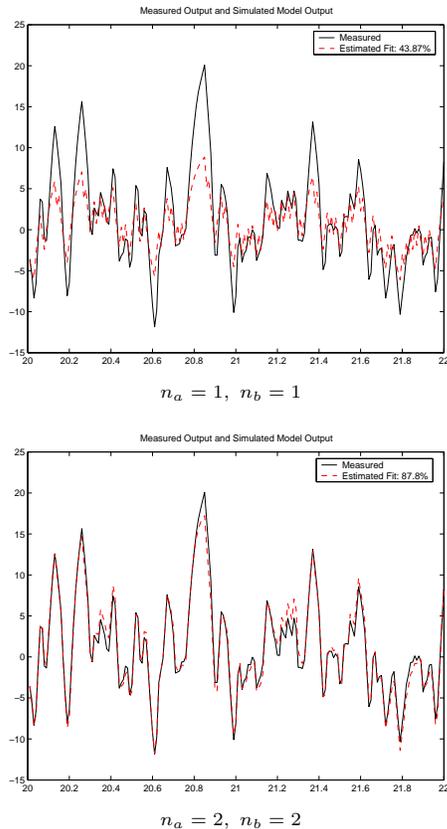


Fig. 6. Comparison between validation data and model simulated output (ℓ_∞ central estimate)

- [3] C.M Merrick and J.W. Ponton. The ecosse control hypercourse. *Computers in Chemical Engineering*, 20, Supplement:S1353–S1358, 1996.
- [4] K-M Lee, W. Daley, and T. McKlin. An interactive learning tool for dynamic systems and control. In *Proc. of International Mechanical Engineering Congress & Exposition*, CA, Nov. 1998.
- [5] C. Schmid. The virtual lab VCLAB for education on the web. In *Proc. of American Control Conference*, pages 1314–1318, Philadelphia, June 1998.
- [6] C.D. Knight and S.P. DeWeerth. World wide web-based automatic testing of analog circuits. In *Proc. of 1996 Midwest Symposium Circuits and Systems*, pages 295–298, Aug. 1996.
- [7] M. Shaheen, K. Loparo, and M. Buchner. Remote laboratory experimentation. In *Proc. of American Control Conference*, pages 1314–1318, Philadelphia, June 1998.
- [8] J. Henry. *Engineering lab on line*. University of Tennessee at Chattanooga. <http://chem.engr.utc.edu>.
- [9] M. Casini, D. Prattichizzo, and A. Vicino. The automatic control telab: a remote control engineering laboratory. In *Proc. of 40th IEEE Conference on Decision and Control*, pages 3242–3247, Orlando, Dec. 2001.
- [10] M. Casini, D. Prattichizzo, and A. Vicino. The automatic control telab: a user-friendly interface for distance learning. *IEEE Transactions on Education*, 46(2):252–257, 2003.
- [11] B. Aktan, C. A. Bohus, A. Crowl, and M. H. Shor. Distance learning applied to control engineering laboratories. *IEEE Transactions on Education*, 39(3):320–326, Aug. 1996.
- [12] A. Bhandari and M. Shor. Access to an instructional control laboratory experiment trough the world wide web. In *Proc. of American Control Conference*, pages 1319–1325, Philadelphia, June 1998.
- [13] J. W. Overstreet and A. Tzes. An internet-based real-time control engineering laboratory. *IEEE Control Systems Magazine*, 19(5):19–34, Oct. 1999.
- [14] J. Zhang, J. Chen, C.C. Ko, B.M. Chen, and S.S. Ge. A web-based laboratory on control of a two-degree-of-freedom helicopter. In *Proc. of Conference on Decision and Control*, pages 2821–2826, Orlando, Dec. 2001.
- [15] V. Ramakrishnan, Y. Zhuang, S.Y. Hu, J.P. Chen, C.C. Ko, Ben M. Chen, and K.C. Tan. Development of a web-based control experiment for a coupled tank apparatus. In *Proc. of American Control Conference*, pages 4409–4413, Chicago, June 2000.
- [16] T.F. Junge and C. Schmid. Web-based remote experimentation using a laboratory-scale optical tracker. In *Proc. of American Control Conference*, pages 2951–2954, Chicago, June 2000.
- [17] J. Apkarian and A. Dawes. Interactive control education with virtual presence on the web. In *Proc. of American Control Conference*, pages 3985–3990, Chicago, June 2000.
- [18] G. Choy, D.R. Parker, J.N. d’Amour, and J.L. Spencer. Remote experimentation: a web-operable two phase flow experiment. In *Proc. of American Control Conference*, pages 2939–2943, Chicago, June 2000.
- [19] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Upper Saddle River, NJ, 2nd edition, 1999.
- [20] L. Ljung. *System Identification Toolbox User’s Guide*. The Mathworks, Inc., 2000.
- [21] M. Milanese and A. Vicino. Optimal estimation theory for dynamic systems with set membership uncertainty: an overview. *Automatica*, 27(6):997–1009, 1991.
- [22] M. Casini. ACT: a remote laboratory of automatic control and identification. In “*Set-Membership Estimation: An Advanced Tool For System Identification*”, PhD. Thesis, pages 117–164. Università di Siena, 2003.
- [23] E. Fogel and F. Huang. On the value of information in system identification - bounded noise case. *Automatica*, 18:229–238, 1982.
- [24] M. Milanese, J. P. Norton, H. Piet-Lahanier, and E. Walter (eds.). *Bounding Approaches to System Identification*. Plenum Press, New York, 1996.