



## Sharing a system between simultaneous learners in remote laboratories

Arnaud Lelevé, Hcene Benmohamed, Patrick Prévot

### ► To cite this version:

Arnaud Lelevé, Hcene Benmohamed, Patrick Prévot. Sharing a system between simultaneous learners in remote laboratories. IFAC Workshop on Internet based Control Education (IBCE 04), Sep 2004, Grenoble, France. 6 p. hal-00187631

**HAL Id: hal-00187631**

**<https://hal.science/hal-00187631>**

Submitted on 5 Jun 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# SHARING A SYSTEM BETWEEN SIMULTANEOUS LEARNERS IN REMOTE LABORATORIES

LELEVE Arnaud, BENMOHAMED Hcene, PREVOT Patrick

*ICTT Laboratory – INSA Lyon, France  
first\_name.last\_name@insa-lyon.fr*

**Abstract:** E-Laboratories are important components of modern e-learning environments, specially in scientific and technical disciplines. Out of E-Laboratories, Remote Laboratories enable learners to train on remote real systems. Such systems are generally used by only one learner at a time but they would gain by being shared between several learners during a same lab. session. This paper presents solutions and their constraints to do so, by making an analogy with industrial production problems. Three system access scheduling algorithms are proposed and simulated on a remote lab. scenario.  
*Copyright © 2004 IFAC*

**Keywords:** Training, Laboratory education techniques, Remote Control, Algorithms, Work Organization

## 1 INTRODUCTION

Electronic laboratories (E-Labs) are important components of modern e-learning environments, especially in scientific and technical disciplines (Ursulet and Gillet, 2002). E-Labs are either R-Labs (Remote Laboratories), which offer remote access to real laboratory equipment and instruments (Saliyah, *et al.*, 2000), or V-Labs (Virtual Laboratories) (Wagner, 1999), which are based on simulations of real systems or phenomena. R-Labs are typically a transfer of classical *in-situ* laboratory towards distance learning environments. Their conception requires technical, pedagogical and computer science competencies. Due to these requirements, it appears to be more complex than other e-learning contexts such as on-line courses, virtual classrooms, e-projects, role-playing, ... typically centered around computer science and pedagogical needs. Motivations for remote laboratories development are:

- sharing heavy and expensive instruments and equipments between institutions,
- anytime and anywhere lab access,
- resorting to real systems for illustrations, during on-line courses or virtual classrooms,
- putting distant students in front of real situations and allowing them: to discover system behaviors, to train at using instruments, to verify scientific theories, ...

Such systems are generally usable by only one learner at a time but they would gain by being shared between several learners during a same lab. session. §2 gives more details about pros and cons. Starting from an analogy with a classical industrial production problem, we propose in §3 a few solutions which were simulated in a few different contexts. Results are commented on in §4.

## 2. SHARING REMOTE LABORATORIES

In classical schools, giving education to more and more people imposes to propose more room to host learners. Distant learning context transforms this hard constraint in a new soft constraint: proposing enough bandwidth to enable more learners to connect through the internet to their virtual campus. As resources become virtual, the main limit to increase the number of simultaneous learners (in asynchronous and synchronous activities) is the maximum number of them an instructor can supervise.

Remote laboratories add a supplementary hard constraint: a given number of lab apparatuses are provided (water level regulator, inverted pendulum, chemical experiment set-up, ...) and shared between every learner. This sharing is typically distributed in time: sessions of single (2 or 3) learner groups follow on from each other. On a larger scale, this may not be sufficient any more.

Three solutions are then conceivable.

*First one* consists in providing other identical apparatuses. This solution results in simultaneous groups, working on identical apparatuses. Each group has his given own apparatus during a whole session. But this is costly: at purchase and in maintenance. Furthermore, learners do not use apparatus during 100% of session time: moments of apparatus inactivity is a waste of time as nobody else can use them.

*A second solution* consists in sharing an apparatus in a finer grain time sharing policy: several groups use the same apparatus, simultaneously at session scale, but each one at his turn, during small slices of time. So, periods of apparatus inactivity are quite smaller

and more groups may use simultaneously the same system. This solution requires to manage in real time a concurrent access to a single resource in order to schedule a fair sharing with minimum wait-times. It also assumes that the R-Lab scenarios are sufficiently finely sliced and organized in a way which facilitates this sharing. This solution also requires that apparatus can be easily and quickly reconfigured (parameters, programs, current state) on group switching. This solution may appear inefficient when several groups are assigned a same scenario: every group will need the first system access at same time. How then to decide which one will be the first to manipulate ? A first approach to avoid this kind of troubling coincidence, consists in shifting session starting time group by group but this generally difficult to organize in a whole training schedule. Another approach would then consist in providing groups different scenarios (differently organized in time) or dynamical scenarios whose trajectory would depend on learner level evolution during the session (Benmohamed, *et al.*, 2003).

A third solution eventually consists in merging both previous ones: providing  $m$  identical apparatuses to  $n < m$  simultaneous learner groups. To be efficient, this assumes a group can use any of the apparatuses without consequence (even better, without noticing it). Using more than one apparatus simply brings more flexibility in sharing organization; we could think that assuming we could host  $n1$  groups on a single apparatus, we should be able to host  $n \geq m \cdot n1$  groups with  $m$  apparatuses.

This way, how to efficiently organize this sharing to ensure every group will have enough time to complete his scenario ?

### 3. SCENARIO SCHEDULING

#### 3.1 Industrial problem analogy

This problem is very similar to the organisation of an industrial production. Our lab systems, as machine tools in a workshop, are shared resources with non concurrent use. The production manager has several simultaneous orders to produce on his machines and has to schedule his production to be the most efficient as possible (criteria will be exposed later). Thus, E-Lab manager has to organise the use of his systems by learners, to procure them an efficient training.

#### 3.2 Modelling

Starting from this analogy, we can model learner groups as different orders to be delivered in time (session end time). These orders are consisted of chronological production steps requiring to be allocated to a specific machine. To carry the analogy further, we can imagine that knowledge and experience acquired as and when learners complete steps, is similar to perishable goods which do not tolerate waiting too long into a temporary intermediary stock.

A particularity of our E-Lab “production” is that groups alternate two kinds of activity: autonomous production (they do not require any machine) and production on lab. apparatus which is done on a single or a few machines. Out of respect for our analogy, we will model autonomous production by a production on a special machine: each group has its own machine to perform this kind of work. This way, they are able to use it whenever they do not need to manipulate.

#### 3.3 Objectives

In a workshop, main purpose is to produce at smaller cost by minimizing stocks and optimising resource use while respecting order delivery date. In our case, we try, in priority, to

- minimize time of waiting for a manipulation for every group,
- ensure that every group will have enough time to complete their work,
- minimize number of apparatuses in a lab.

Besides, in industry, changing pieces from a machine to another implies to create an intermediary stock in order to prevent a lack of raw material for next operation which would raise production lateness.

In our case, creating a stock corresponds to queuing learners up, which is not very comfortable for them. This is why our first objective is to limit queues.

#### 3.4 Assumptions

We consider manipulations on a single type of apparatus. Our laboratory is supposed to feature up to 3 cloned apparatuses. So, there is no constraint to assign a group to one apparatus or another one, from one manipulation step to the next one (in a workshop, we would have had to consider transportation cost between two different machines).

In this paper, we will assume reconfiguration is not necessary or takes so small a time that we do not deal with it in this first approach. In fact, we can assume reconfiguration time is included in learners playing time.

#### 3.5 Static approach

E-Lab scenario authors give initial timing forecasts for every step (autonomous work as well as manipulation step), according to each step difficulty and the fact learners become acquainted with apparatus use as long as they use it. As long as authors build their scenarios, they can ask for a test schedule by simulating  $n$  groups with different levels working on  $m$  lab. apparatuses, to make their forecasts more accurate. By the way, they limit the risk of critical situations when every group simultaneously requires to manipulate.

At performing time, a schedule can be established once for all, before first session. It will assume there will be a constant number of learner groups by session and always the same number of apparatus. But it will not work as soon an apparatus breaks down, a new one becomes available or authors

update their scenarios. To prevent such drawbacks, a more accurate schedule could be built at the beginning of each session, depending on the number of participants and the number of available apparatuses in each session.

Main drawback of establishing a static schedule is that learners will not respect exact timings as authors forecasted; schedule will be progressively more and more shifted and inefficient.

### 3.6 Dynamic approach

As in a workshop, production steps may take more (or sometimes less) time than forecasted (breakdown, unskilled operator, ...). This implies to be able to regularly reorganize production when gaps between forecasts and reality become too harmful for global production organization. In a remote lab, as we deal with humans, manipulation and autonomous study phases are ought to vary in time.

By computing the schedule on the stream:

- whatever may be the differences between forecasts and reality, the planning is iteratively adapted to the reality,
- whenever a group prematurely stops or an apparatus breaks down, reorganisation will immediately take this event in consideration
- authors have just to verify their scenarii are feasible in session time in current conditions.

### 3.7 Optimisation algorithm

*First step* (“*scheduling*”, in fig. 1) consists in estimating requirements: “when and how long a group requires to manipulate”? It may be consolidated with data about how much they are in late compared to forecasts and how efficiently they used their previous manipulation time.

Considering these estimations, we have to determine an access order and affectation to a given apparatus. This step generates an **access schedule**, defining when and for how long each group has the right to access a given apparatus.

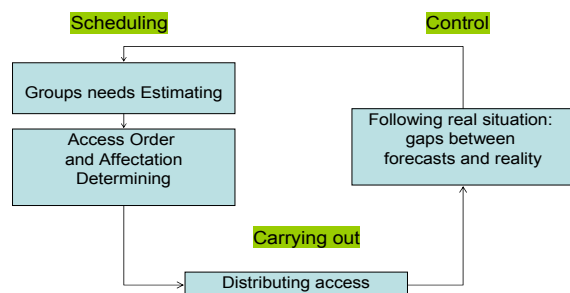


Fig. 1. Apparatus Access Scheduling Cycle

Apparatus access regulation software uses this access schedule (“*Distributing access*” box of fig. 1) to warn groups that they are about to be able to manipulate. If they really need it, they accept this time frame, otherwise, they can refuse it to beneficiate later (with priority over others) and tell the system when they should need it next time.

Once accepted, they manipulate and, when they finish, they free this access which can then be dealt to another waiting group. Each group is warned a

few minutes before the end of their time frame. Three situations may arise:

- they free this access before the end of time frame (anticipated freeing). Time not used for manipulation is then added to their future needs.
- They did not finish their manipulations: in this case, they ask for an extra time. By default, we can imagine the system gives 10% extra time.
- If they require even more time, trainer will be notified and will decide to give (or not) a certain amount of time, considering other groups needs.

*Third step* (“*Control*” of fig. 1) consists in evaluating gaps between forecasts and reality to enhance optimisation.

*Frequently used performance criteria*: they are used to measure how much scheduling is efficient. Table 1 proposes a comparison between industrial and E-Lab criteria.

Table 1 Scheduling criteria

Industry criteria	E-Labs criteria
1. Total cost of production	1. Global cost of E-labs sessions
2. Product current quantity	2. Number of learner groups
3. Equipement use ratio	3. Apparatus use ratio
4. Mean delay of orders	4. Mean delay of groups
5. % of delayed orders	5. % of waiting groups
6. global time of production	6. Total time of manipulation

*Parameters* for optimisation algorithm are :

- Total number of learners
- Number of learners per group
- Length of a session
- Total number of sessions
- Number of apparatuses

Once established parameters and chosen our criteria, we have to set up our priority rules. They permit to specify an equitable access order of groups, taking into account factors (importance of manipulation, extra time gathered since beginning of session, 3<sup>rd</sup> attempt of same manipulation, ...).

The most simple frequently used rule, but also the most penalizing one, is “*by arrival order*” (*BAO*): first arrived, first served. Main drawback is that groups requiring more time than others will delay other groups.

Another simple but fairer rule is “*by shorter (processing) time*” (*SPT*): the group which should take the minimum time for next step is chosen.

It is only required to know how much time each group requires for his following step. In industry, it is known to propose the smallest mean delay and, by the way, the best equipment use ratio. It also presents the smallest mean number of orders in waiting state. It gives so a better control. Also, total execution time is smaller than with previous rule. Its main drawback is that it neglects longer manipulations. So, to be more efficient, this rule should be merged with another one, otherwise some long manipulation requests will never be scheduled (or too late). It also does not take into account programmed product

delivery date but in our case, as groups work in a common session, every group has same delivery time, so this drawback is not so annoying. If we foresee to use our apparatus in a continuous time schema (different sessions may overlap), we could envisage another rule: “by nearest delivery date” for instance but in our hypothesis, every group belong to the same session so this rule has no interest in this case.

*What if groups get too much lateness?* Three reasons can explain too long queues:

- inadequate capacity: too many simultaneous groups compared to the number of available apparatuses,
- unsuitable scheduling algorithm,
- too long manipulation time per group.

Possible solutions to this problem are to:

- add a new cloned apparatus,
- extend session or reduce scenarii lengths,
- reduce number of manipulations in scenarii,
- reduce necessary time per manipulation (at global constant volume) in order to increase granularity

### 3.8 Appealing to virtual apparatuses:

In a workshop, a costly solution to make production easier consists in installing buffering stocks. In reality, manufacturers will prefer study a new scheduling. In our case, it would be interesting to replace buffering stocks, i.e. queues, by experimentations on a virtual system which eliminates problems of concurrent access.

For instance, in TIPY, learners could test and debug their algorithms on a software simulator before testing them on the real system.

It assumes that the simulator is perfectly compatible with the real system: learner should be able to use both without having to modify their programs.

## 4. SIMULATIONS

### 4.1 Purpose :

We wanted to illustrate the use of simple rules to schedule group access to apparatus, in order to optimise it. We used our experimental platform TIPY (Leleve, *et al.*, 2003) with a scenario written for the Vertical Store apparatus (see figure 2), in automation discipline. Following simulations don't appeal to virtual apparatus as suggested in §3.8

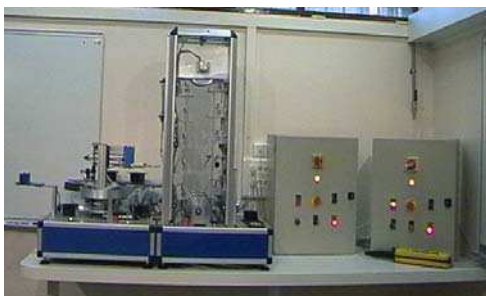


Fig. 2. TIPY Vertical Store E-Lab apparatus

In this context, learners are distant from system and grouped by 3 in front of their client PC. Tutor is distant from both learners and system as depicted in figure 3.

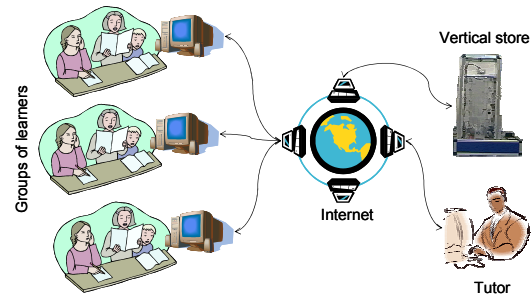


Fig. 3. TIPY actor repartition

### 4.2 Scenario structure

This scenario has been translated from real lab works currently used by our engineer students. It was conceived to keep students hard working during 2 sessions of 4h. In reality, a very few of them finish the entire scenario and the majority reaches three quarters of it.

TIPY E-Lab scenario had been extended compared to original one: according to students level, different steps are automatically chosen to fit the difficulty level with their needs. So, two learners may take different paths according to the time they spent and answers given to a Multiple Question Test at each end of previous step. Figure 4 gives a glimpse of its global structure with all possible steps and paths.

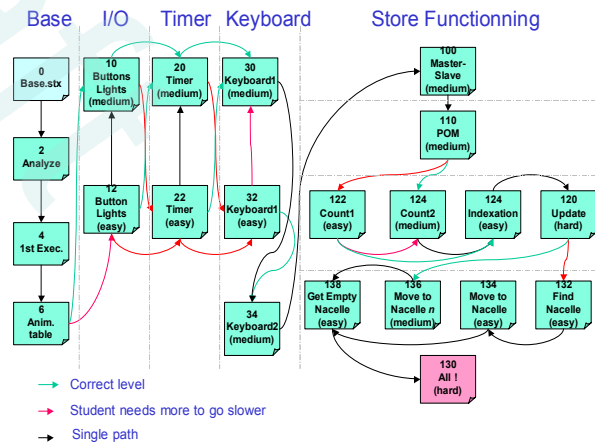


Fig. 4. TIPY dynamic scenario paths and steps

TIPY scenario structure was not fine enough to separate manipulation times from autonomous study times. We needed to slice each step in sub-steps which could be affected to an autonomous work or to an apparatus.

Table 2 gives details of an arbitrary step: (#34) which consists, for learners, in programming the vertical store to bring an empty gondola in front of the window, in order to put a piece into it in a near future.



**Table 2 Sample of step description**

Substeps	Description	Use apparatus	Time (mn)
34.1	Preparation	<input type="checkbox"/>	5
34.2	1st Test	<input checked="" type="checkbox"/>	4
34.3	Debug off line	<input type="checkbox"/>	3
34.4	2n test	<input checked="" type="checkbox"/>	3
34.5	Conclusion	<input type="checkbox"/>	2

#### 4.3 Simulation algorithms

Three algorithms were tested:

- BAO which can be summed up by following rule: “when more than one group wish to manipulate and an apparatus free itself, choose the first one which asked for it”.
- A variant of SPT (VSPT) represented by rule: “when more than one group wish to manipulate and an apparatus frees itself, choose the group which will use it during least time”. We can consider it is a mix between BAO and SPT.
- A “by least progress” based algorithm (BLP) using same kind of precedent rules but the chosen group is the one which has the least progressed in his work.

#### 4.4 Simulation parameters

According to the needs of simulation (2 to 6 simultaneous groups), we generated virtual use cases with different level groups, some using same paths (when levels are similar), others using different paths (when levels are different). Each group has its own velocity, translated by applying a constant ratio to step work times initially forecasted by the author (see table 3).

**Table 3 Groups, paths and levels**

Group id	Path level	Time ratio %
W1	Weak	160
W2	Weak	140
M1	Mean	140
M2	Mean	130
M3	Mean	120
S1	Strong	130
S2	Strong	120

**Table 4 Groups, by session size**

Session size	Group ids
2	W1, S1
3	W1, M2, S1
4	W1, M1, M3, S1
5	W1, W2, M1, M3, S1
6	W1, W2, M1, M3, S1, S2

As we needed to fetch data with different population by session, we set together some of them to create homogeneous sessions of 2 to 6 groups (see table 4).

#### 4.5 Simulation results

Schedules have been recorded. Figure 5 presents a sample built with BAO algorithm with 3 groups on a single apparatus. “W” dark gray cells depict waiting time for groups, while white cells in gray font display apparatus state (free or group id).

**2 groups on 1 apparatus:** with BAO, we obtain total wait time of 31,8 and 40,2 minutes, which corresponds to a global overtime of 7,2% (5,6 and 8,9%). Largest wait time was of 18,4 and 20 minutes but mean wait time is only around 0,6 minute. In fact, it was better than we expected. It means that the scenario has a good structure for this kind of use. VSPT and BLP algorithms give exactly same results because no real competition takes place between two groups. In fact, they simply resume manipulations one after the other.

Apparatus stays unused during 37% of total time, which means we could try to enhance this variable.

**3 groups on 1 apparatus:** competition starts. BAO results to an mean overtime of 22,5% (20,6→23,9%) but wait times are well distributed (mean ≈ 2 min, max 24 min). It is interesting to observe that mean wait time is twice as much as previous case but max wait time stays relatively constant. VSPT gives hardly better results: mean overtime of 20,8% (19,2→22,1%), mean wait time of 1,8 min but one group has to wait once up to 36,4 minutes (≈23 min for two others). It is uncomfortable for this group. Moreover, it was the weak one (longer manipulation times). The price to pay for this little enhancement does not seem interesting. Apparatus is unused during 18% of global time.

**3 groups on 2 apparatuses.** this situation virtually corresponds to 1.5 groups per apparatus. It is then obvious we obtain even better results than with 2 groups per apparatus: relative overtime < 1,5 %, mean wait times < 10 seconds and maximum at 3,7 min.

Apparatuses are logically under-used. In fact, they are only used during ~50% of session time. VSPT and BLA algorithms give same scheduling as BAO because there is no real competition as resources are numerous enough such as in first use case. Apparatus use is unbalanced (43 and 54% of unused time) but algorithms do not take into account such a constraint.

Group	Time (mn)	0,0	6,5	8,0	19,5	23,4	24,0	26,0	27,3	29,9	32,1	34,7	35,3	36,0	37,3	38,6	40,5
W1	Autonomous	0,0	2,0								4,1						4,3
	Apparatus	free	free	free	M1	S1	W		4,0		M1	S1	W		4,2		free
M1	Autonomous	0,0	2,0			4,1						4,3		6,0			
	Apparatus	free	free	free		S1	S1	W			4,2	S1	S1	S1	W1	W1	free
S1	Autonomous	0,0	2,0						4,1							4,3	6,0
	Apparatus	free	free	free	W	4,0			W1	W1	W	4,2			W1	W1	free

**Bold numbers: step #**

Gray text: Apparatus

**W : Wait state**

Fig. 5. Sample of simulation results: BAO with 3 groups on a single apparatus

4 to 6 groups on 2 apparatuses: relative overtime progressively raises from ~4% to ~18% with BAO algorithm. Groups get a fair mean wait time but the more numerous they are, greater are the disparities of max wait time (ranging from 9,7min to 42,3min at 6 groups). At 4 groups, VSPT gives same results as BAO but when number of groups increases, relative mean overtime get better than with BAO (-1% at 5, -13% at 6). His main drawback confirms itself: groups may wait very different maximum wait times (from 13,9 to 43,5 minutes at 6: worst than with BAO). For his part, BLA algorithm always gives worst results than two others: relative mean time increases from 20% (at 4 groups) to 43% (at 6 groups) compared to BAO one.

Concerning apparatuses, their use time progressively raises. At 4 groups, two apparatuses are unbalanced (29 and 40% of unused time) but results are more balanced after with global similar results between BAO and VSPT.

#### 4.6 Simulation conclusions

With scenarios which were not initially optimized to be used simultaneously, we obtain interesting results for 2 groups on 1 apparatus and up to 5 seems reasonable on 2 apparatuses. We can also wonder whether it is possible for the instructor to monitor more than 5 simultaneous groups in good conditions.

Worst results were found for BLA: as algorithm favours groups in late, those in advance are delayed and have to wait for slower ones. Even if it is generous, it has the disastrous impact on the whole timings (20 to 40% extra time compared to other algorithms). Mean results were found with BAO which was our reference as this is the first approach we thought was logical. Global best results were found for VSPT. The only drawback of this algorithm is that there may be a group which pays for all others the effort to reduce wait times.

Obtaining wait times of a few minutes is not really embarrassing in our point of view as it can be a good moment to have a talk between learners and tutor or write a report or even take a pause. Frequent wait times greater than 10 minutes appear to be problematic. It is important to remind that we did not use a real VSPT algorithm but a mix of BAO and VSPT. Knowing global results and specific phenomena raising from these 3 algorithms, we have now indications to develop algorithms more adapted to this problem. For instance, algorithms proposed in this paper act in present but have no memory nor real forecast capabilities. It would be interesting to integrate past, present and future data to enhance global scheduling and limit annoying wait times.

Even if these algorithms are not optimal, same ones can efficiently be used to monitor apparatus access in real time as to help the author to realize fine grain scenarios and evaluate their future scheduling.

## 5. CONCLUSION

This paper presents a method to schedule access to apparatus between several learner groups during a Remote Laboratory session. Based on an analogy with production engineering, we propose simple rules to limit wait times for learners and increase usage of apparatus. Simulations of use of these rules in different situations gave us interesting results concerning the interest of this study. Several studies will follow: new dedicated scheduling algorithms (based on past, present and future state estimation) and taking into account reconfiguration time when they become significant. Experimentation on our experimental platform is envisaged.

## REFERENCES

- Benmohamed H., Leleve A., Prevot P. and Meyer C. (2003), "Remote Laboratory: towards an integrated training system" in proc. of 4th proceedings of the International Conference on Information Technology Based Higher Education and Training (ITHET'03), Marrakech, Morocco, July 7-9 2003, pp. 110-115.
- Leleve A., Prevot P., Subai C., Noterman D. and Guillemot M. (2003), "Towards Remote Laboratory Platforms" in proc. of 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI'03), Orlando, Florida, USA, July 7-9 2003, pp. 110-115.
- Pinedo M. (1995), "*Scheduling, theory, algorithms and systems*", Prentice Hall, ISBN 0-13-706757-7
- Saliah H. H., Villardier L., Assogba B., Kedowide C., Wong T. (2000), "Resource Management Strategies for Remote Virtual Laboratory Experimentation" in proc. of 30<sup>th</sup> ASEE/IEEE Frontiers in Education Conference, Virginia Kansas City, MO, October 18-21 2000.
- Ursulet S. and Gillet D. (2002), "Introducing flexibility in traditional engineering education by providing dedicated on-line experimentation and tutoring resources" in proc. of the International Conference on Engineering Education, Manchester, UK, August 18-21, 2002.
- Wagner B. (1999), "From Computer-Based Teaching to Virtual Laboratories in Automatic Control" in proc. of 29<sup>th</sup> ASEE/IEEE Frontiers in Education Conference, San Juan, Puerto Rico, November 10-13, 1999