European Journal of Sport Science, vol. 1, issue 2

PerPot: A Metamodel for Simulation of Load Performance Interaction

Jürgen Perl

A metamodel is introduced, which on one hand can help to understand particular effects and phenomena in the interaction of load and performance in training processes. On the other hand, it can be used as a starting point for refinements to specific adaptation models. Finally, a software tool has been developed that supports different simulation approaches—for example, basic analysis of model parameter influences, diagnosis of the state of real adaptation systems, optimization of given load performance interactions, and planning optimal training schedules.

Key Words: adaptation, training, modeling, genetic algorithm, optimization, scheduling

Key Points:
• PerPot, a metamodel for modeling phenomena of physiological adaptation, is introduced.
• The PerPot software facilitates analysis and optimizes the interaction of load and performance.
• By use of genetic algorithms, training plans can be generated or improved.

Introduction

Aspects of load performance interaction have been discussed in detail during the last few years (e.g., see 2, 3). The Performance Potential Metamodel (PerPot) has been developed in order to simulate basic phenomena in the interaction of load and performance—for instance, the super compensation effect or collapse under overload. Deduced from such basic effects, PerPot consists of a number of potentials interacting with each other via delayed transport lines. Mathematical analyses show that the particular effects of time dependant oscillation and stabilization of performance can only be modeled using a structural property we call *antagonistic,* and which, in analogous ways, can be found in a lot of self-controlled natural systems. The reason for calling PerPot a metamodel is that it is not a model for just one system but for the class of all systems that control themselves using those antagonistic control structures. By refining the components of PerPot, particular models for special purposes can be derived from PerPot. In addition to this, PerPot can be used to improve neural networks for a better modeling of learning processes. In the following presentation, however, PerPot is used as a model for simulation of basic phenomena and effects. The article describes how dynamical training behavior can be understood from a structural point of view and how PerPot might help to plan and optimize training schedules.

The Model Structure of PerPot

As is shown in Figure 1, PerPot consists mainly of three potentials: The buffer potentials for strain and response are fed equally by the same training load. These buffer potentials in turn influence the performance potential in an antagonistic way: While the response potential raises the performance potential, the strain potential reduces it.

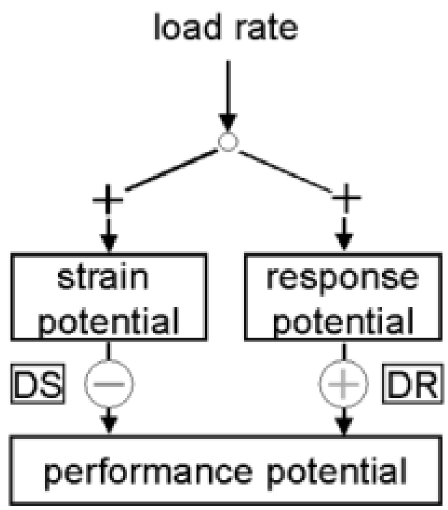

Figure 1 — PerPot model structure.

Doing it synchronously with equal delays, the whole interaction would have no effect. If, however, the delays of strain flow (DS) and response flow (DR) are different, the model shows a particular dynamic behavior, which depends on the current states of the potentials as well as on the current delay values (see Figure 2).

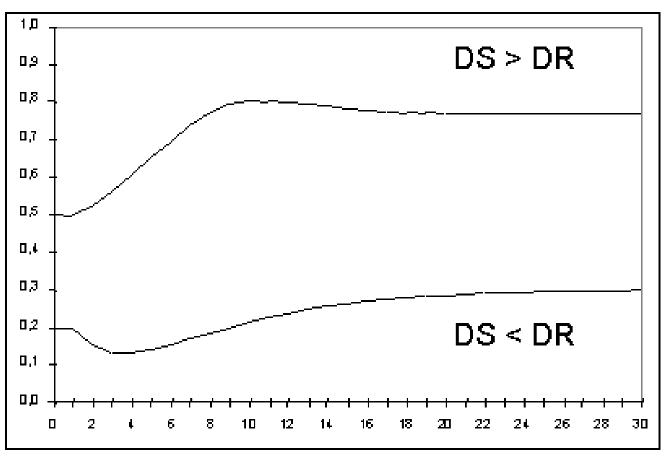

Figure 2 — Dynamic behavior of PerPot depending on the relation between the delay values.

Mathematical analyses show that this version of PerPot always converges into stable situations and without any effect of collapsing. The idea leading on is that of limitation of potentials (also see 1): Natural potentials normally have a limited capacity. So in case of overload, they might react with overflow effects. Figure 3 shows the simplest way to model the overflow effect that can happen with the strain potential: The surplus defines a delayed strain flow that reduces the performance potential.

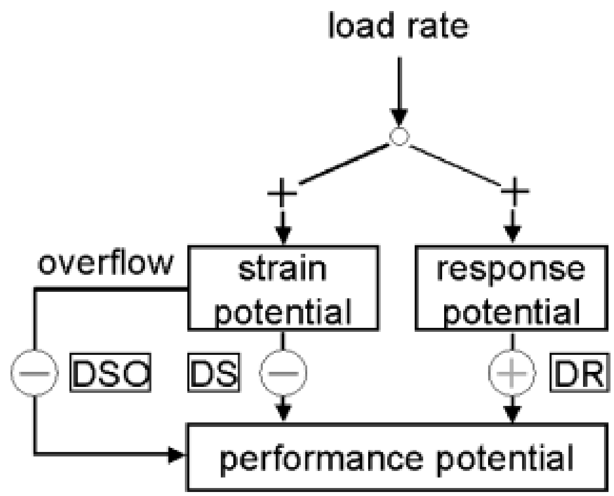

Figure 3 — PerPot completed by overflow component.

For getting an impression of how PerPot works, a PerPot animation is available. Depending on interactively adjustable load rate and delay parameters, it demonstrates how performance depends on load rate over time. In particular, it demonstrates the effect of load potential overflow and the emergency situation of the performance potential running empty.

\PerPot animation\

The idea of limited capacity and overflow allows for the introduction of a so-called reserve function that indicates the danger of a hidden collapse: The current reserve is measured as the difference of the strain potential capacity and the current value of the strain potential. If the reserve becomes negative, the overflow starts to reduce the performance potential, probably effecting an immediate collapse, as shown in Figure 4.

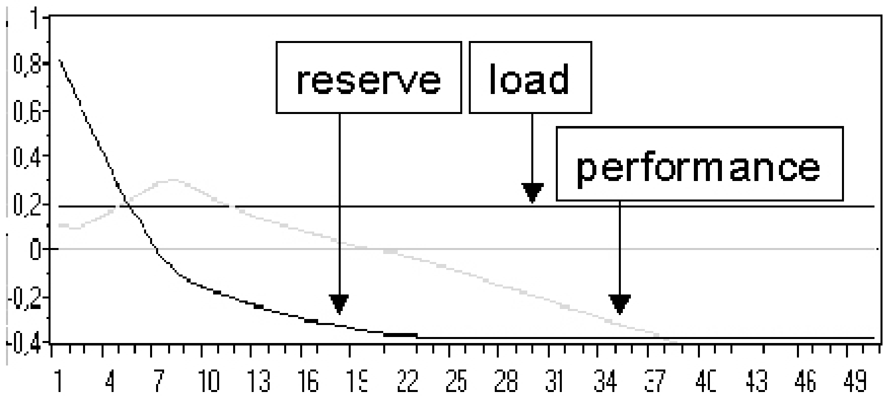

Figure 4 — Collapsing performance.

The following demonstrates how PerPot can be used for analysis of load performance interaction as well as for planning training schedules. All analyses are done by a particular PerPot software program that has been developed to support the model-based studies of phenomena in the field of load performance interaction. Some of the different user interfaces are introduced when dealing with the respective applications.

A first presentation of PerPot took place on the 4th Annual European College of Sport Science Symposium (see 4). A detailed description of the metamodel can be found in Mester/Perl (6), and "Antagonistic Adaptation Systems" (7) gives an overview of the model as well as of the software and its handling.

PerPot Based Phenomenon Analysis

Basic Parameter Effects

Figure 5 shows the starting user interface of PerPot, which allows the user to "play" with the different model parameters in order to get a feeling of how the model works and reacts.

Moreover, interactive parameter changing combined with online simulation of the model behavior can be a proper way to discover interesting areas or "hot spots" of the model in the event that its structure is too complex for systematic analysis.

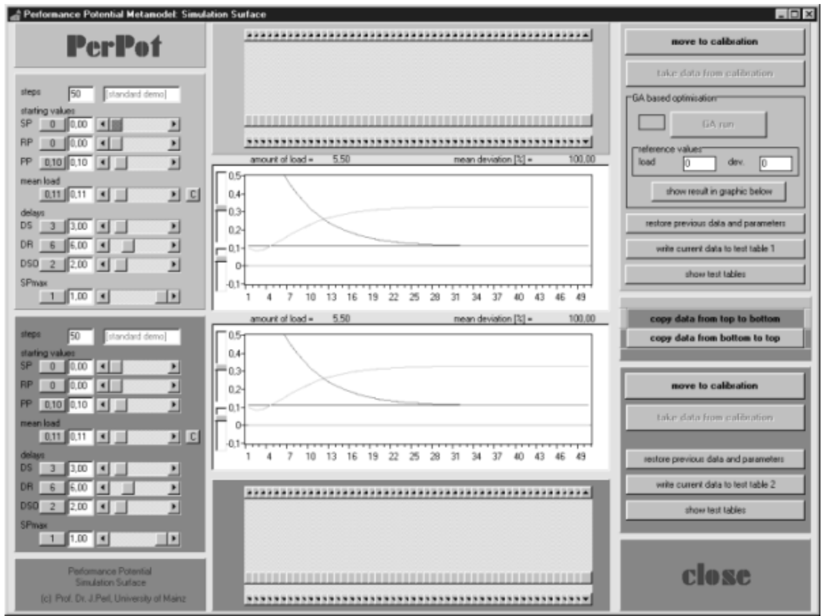

Figure 5 — Starting interface of the PerPot software.

The interface shown in Figure 5 contains two rather similar looking areas, which allow for parallel modeling and results that are easy to compare. The graphic in the middle of Figure 5 shows the current time-dependent values of load, performance, and reserve. Attached to the graphics, arrays of scrollbars are available for interactive changing of the load values, resulting in online changing of the load and reserve profiles. In the same way, the model parameters, in the form of delays and capacities, can be modified on the panels on the left-hand side. Again, modification to the model parameter results in an immediate adaptation of the graphic presentation. Figure 6 shows the effect of raising the load level from 0.11 to 0.18.

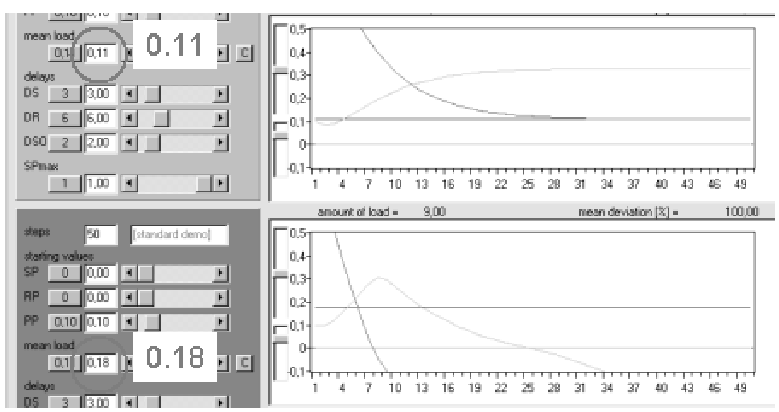

Figure 6 — Collapse by raising the load level to 0.18.

Note that on the starting user interface the delay parameters are time-independent constants. For deeper analysis of original data, constant parameters cannot be assumed. As is shown in the following, the use of time dependent delay profiles normally can be far better applied to the observed load performance interactions. (Some of the results detailed in the following section were first presented at the 5th Annual ECSS Symposium [see 4].)

Approximation of Performance Profiles

The problem with time-dependent delay parameter profiles in natural systems, however, is that they are unknown. One way to detect them is to run a closed-loop calibration, where the parameter profiles are changed as long as the simulated performance profile does not sufficiently coincide with the original performance profile. Mathematical analysis showed that—at least in the stable model states—the system is overestimated with the two parameters DS (strain delay) and DR (response delay). This means that the absolute values of these delay parameters are not as meaningful as the relation between them. Hence, in the following, the DS-profile is set constant while the DR-profile is optimized.

To check the quality of that algorithm, virtual data with generated profiles were used with quite satisfying results. The graphic in the lower part of Figure 7 shows the algorithmically detected DR-profile, which in fact was generated as a "3-pyramide-sequence." The lower graphic in Figure 7 shows the originally generated load profile, while the upper graphic is a (well-fitting) overlapping of the original and the simulated performance profiles.

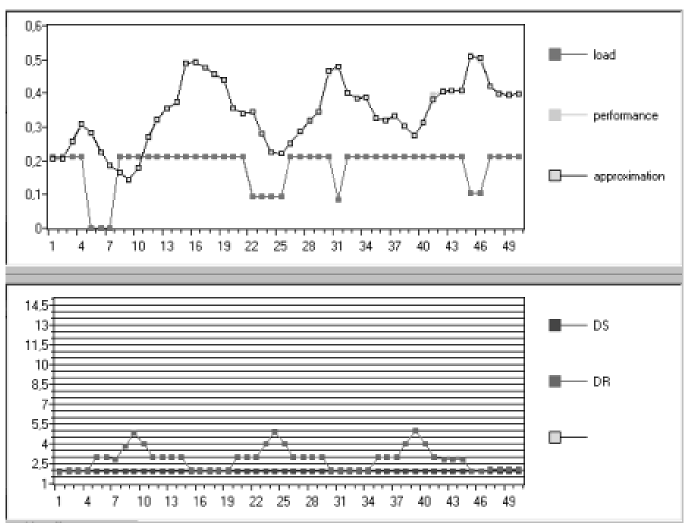

Figure 7 — Example of a calibration check.

Two examples are shown in Figure 8, where the PerPot generates a simulation based on the original data using the method of closed-loop calibration.

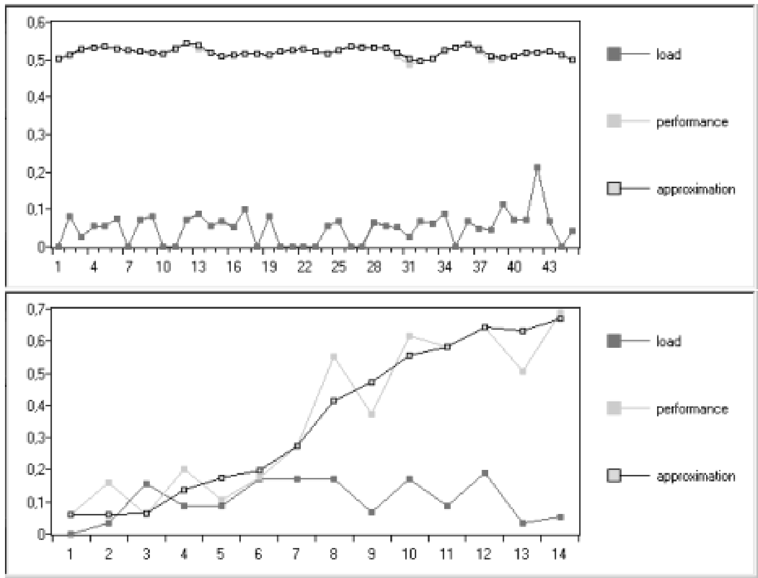

Figure 8 — Examples of PerPot simulation.

The user interface that can be used for calibration and simulation is shown in Figure 9. It contains two additional features, which cannot be explained in detail but should be mentioned: With respect to unified analysis, all model data are normalized to a range of 0 to 1; original data are not. Thus, the first step in calibration is to find the best fitting normalizing factors. Once calculated, these factors can be stored and loaded together with the load and performance data, using the buttons and features on the upper left-hand side of the interface. The lower left-hand side offers a couple of features supporting the calculation of time-dependent delay profiles. Sliding windows of arbitrary length can be used to calculate mean values of those parameters corresponding to time intervals. In particular, a length of 1 results in the step-by-step calculation that is used to calculate time-dependent delays (where, in Figure 9, constant ones are used).

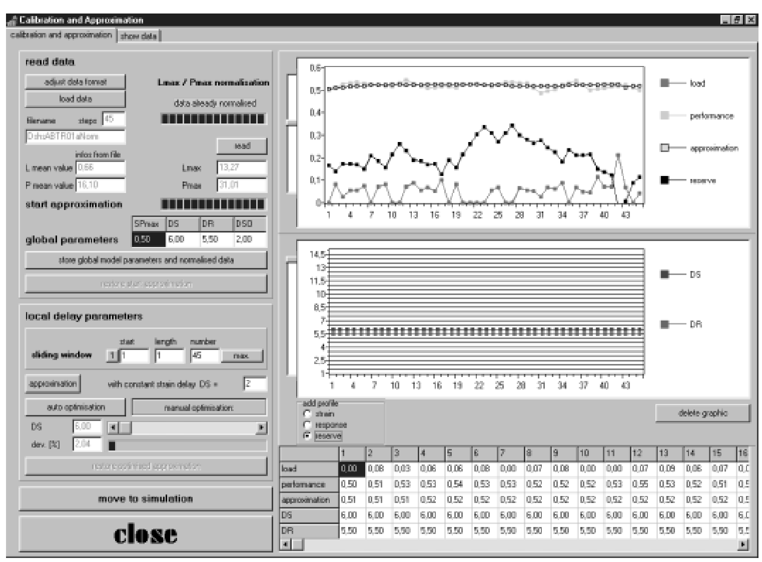

Figure 9 — User interface for calibration and simulation.

Profile Analysis
A main problem of load performance interaction is the analysis of the correlations between changes of load on the one hand, and resulting changes of performance on the other. Sometimes performance seems to rise with increasing load; sometimes the contrary effect can be observed. Obviously, those local effects are mainly caused or influenced by the delays. But load overflow situations or contrary behavior of the internal buffer potentials can also result in such effects. Model simulation can help to better understand what happens.

The upper graphic in Figure 10 shows load and performance profiles: In the left and middle boxes, performance in a typical way is delayed against load. These delays correspond to the delay values of 2, indicated in the lower graphic.

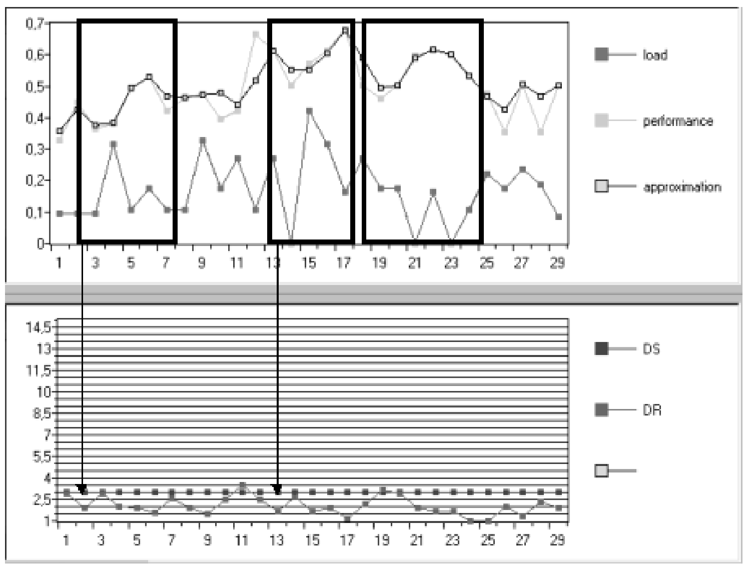

Figure 10 — Correspondences between delayed effects and delay values.

In the right box, however, performance decreases with increasing load. It could be assumed that a collapse effect—caused by a load rate that is too high—might be the reason. A closer look at the internal dynamic (Figure 11) shows that, in the contrary condition caused by a *too low* load rate, only the internal response potential RP ran empty. However, in the interval between step 15 and step 21, there is in fact a hidden collapse situation, as depicted in Figure 12: The reserve function becomes negative and so indicates an additional reduction in performance. Only a very high performance level and an immediate reduction of load prevent a breakdown.

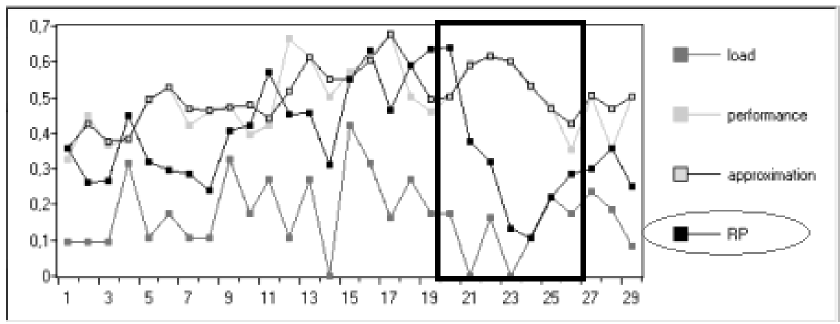

Figure 11 — Influence of internal potentials

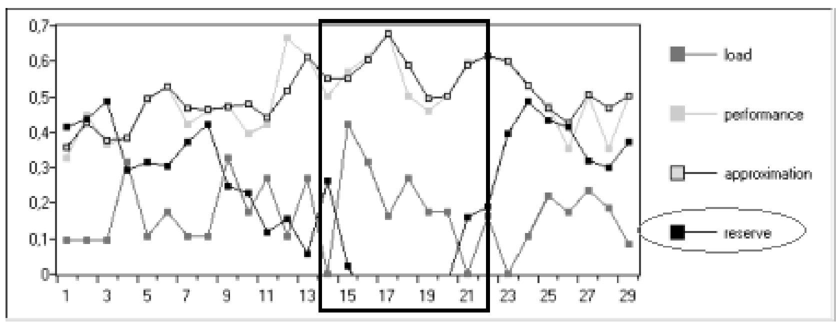

Figure 12 — Reserve profile indicating a hidden collapse.

Feature Analysis

The above examples demonstrate that understanding striking features often requires a deeper analysis of the system's state and behavior. It is very difficult to recognize dangerous situations and to react timely without first examining the dynamics of a system. Models and simulations could help to improve such preventive measures.

Not only are the local load and performance profiles of interest. As mentioned in the discussion above, delay profiles also play an important role in the dynamic behavior of PerPot. For example, a correlation analysis of the load and performance profiles from Figure 7 suggests that there is no correlation at all. As can be seen in Figure 13, ups and downs of load as well as of performance seem to correspond in an arbitrary way in any combination.

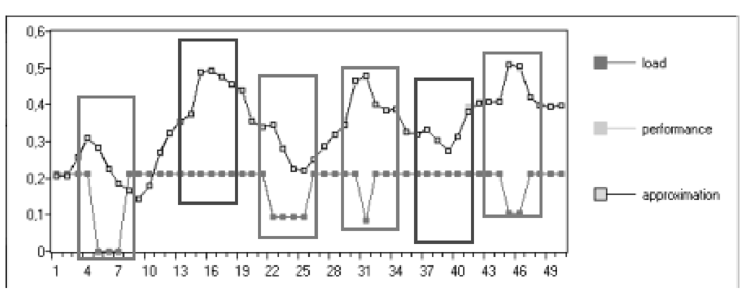

Figure 13 — Correlation between load and performance.

However, comparing load and performance profiles with the response delay profile leads to a quite different correlation: Performance is highly synchronous to the response delay (see Figure 14).

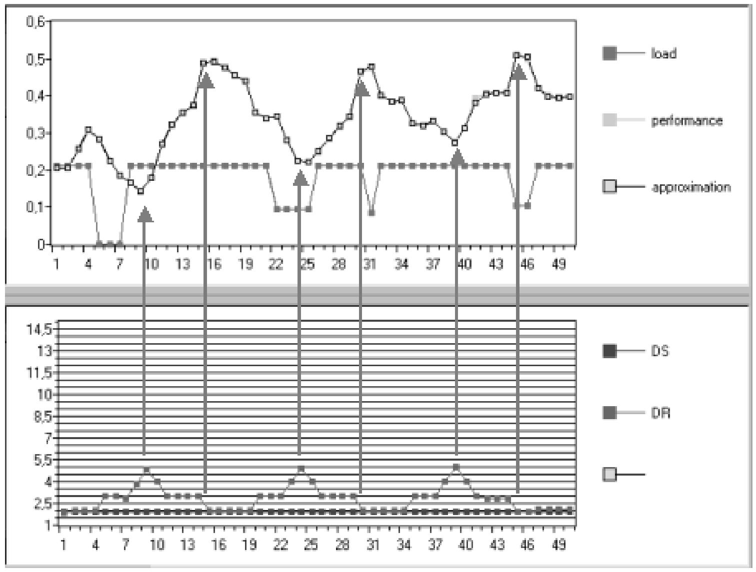

Figure 14 — Correlation between performance and response delay.

At first glance, this result might be surprising. From the point of view of model dynamics, though, it is not. It only says that varying time-dependent parameters influence the dynamic behavior in the same way as potential flows. Transformed to physiologic systems, this means that “parameters” can act like potentials. This is not a contradiction; rather, it corresponds with the idea of refinement of system components (introduced above), where a “parameter” can stand for a dynamic subsystem defining internal potentials and flows between them.

From this point of view, the PerPot model seems to reflect the physiological behavior of an athlete: In the short-run, training can directly improve performance. In the long-run, however, training influences parameters that, in turn, might improve the level of interaction between load and performance.

Outlook: Training Schedules

Normally the trainer’s task is to find a proper load plan in order to meet a projected performance profile, which often is determined by fixed events. Because of the delays and the antagonistic effects of load, developing an optimal load schedule can be a complicated task. On the one hand, the amount of load could be wrong, resulting in a low-level performance. On the other hand, the distribution of load over time could be wrong, resulting in a mismatch between events and high performance areas.

Algorithmic approaches can be used to determine optimal training schedules. In the following example, a so-called genetic algorithms (GA) is used. The basic idea of GA is that of evolutionary selection or “survival of the fittest,” where solutions are generated, combined with each other, and selected until best fitting “winners” are found. In particular, in complex problem situations, GAs can be used successfully if high-speed calculations are necessary and if a "close to optimality" standard is sufficient.

Model-Based Reduction of Training Load

The upper graphic in Figure 15 shows a load profile and, very close to each other, an original and a simulated performance profile. Between load and performance profiles, the simulated reserve profile can be seen. The lower graphic in Figure 15 shows the effect of GA-based optimizing: The performance approximation is even better (by 21%), the load is reduced (by 34%), and the reserve is clearly improved. Even the threatening collapse at step 42 could be avoided.

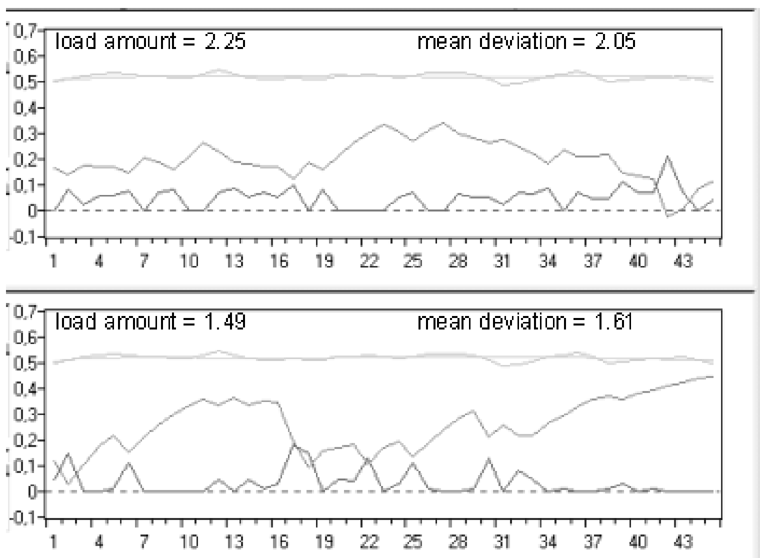

Figure 15 — GA based optimization of the load profile.

Model-Based Scheduling of Training Profiles

The approach described above can also be used to meet given performance profiles with optimized load schedules. To this end, PerPot offers a special design tool that allows the user to manually sketch the desired profile (see Figure 16, gray line), which then is approximated by GA calculating a proper load profile (see the upper above in Figure 17). The lower graphic in Figure 17 demonstrates the possibilities of additional manual improvement, where the work has started to do this by computer-based algorithms, too.

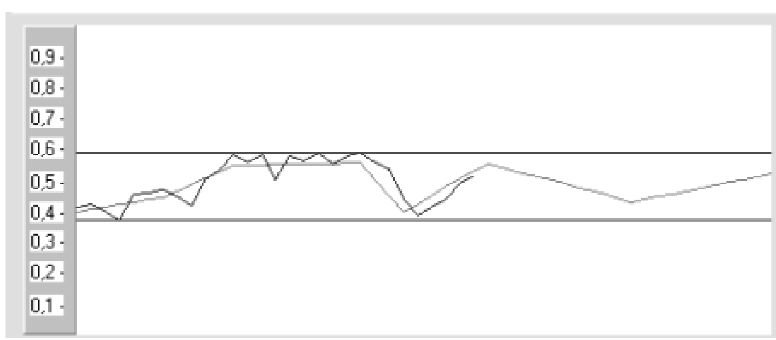

Figure 16 — Load profile (gray line) sketched for GA based approximation (blue line: pattern used as reference).

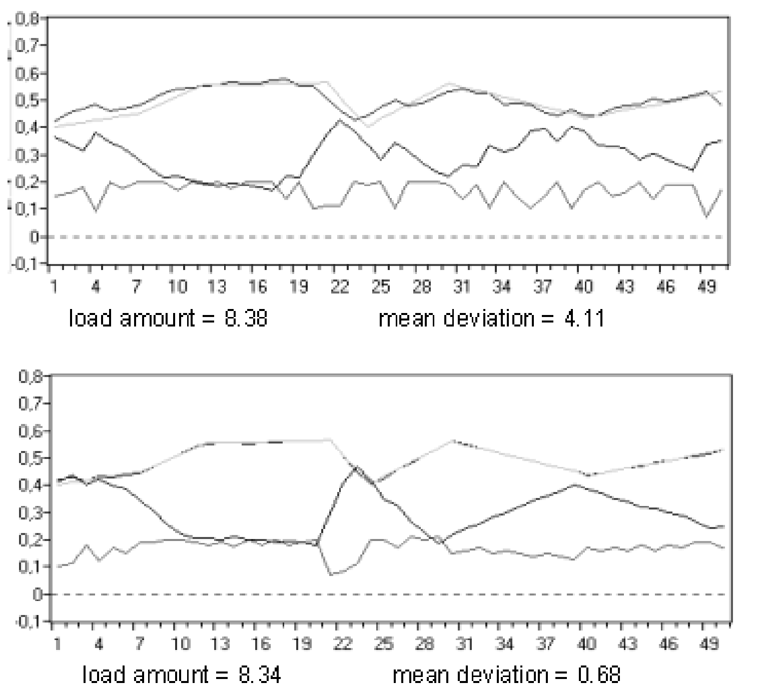

Figure 17 — GA-based scheduling and manual improvement.

As a final example, the GA-based scheduling method is applied to the situation presented in Figure 15. As Figure 18 illustrates, the performance profile is manually sketched as a rather straight line on the level of about 0.5. GA-based scheduling results in a load profile that is 54% smaller than the original load amount. Correspondingly, the reserve profile looks satisfactory and in particular shows no threat of collapse.

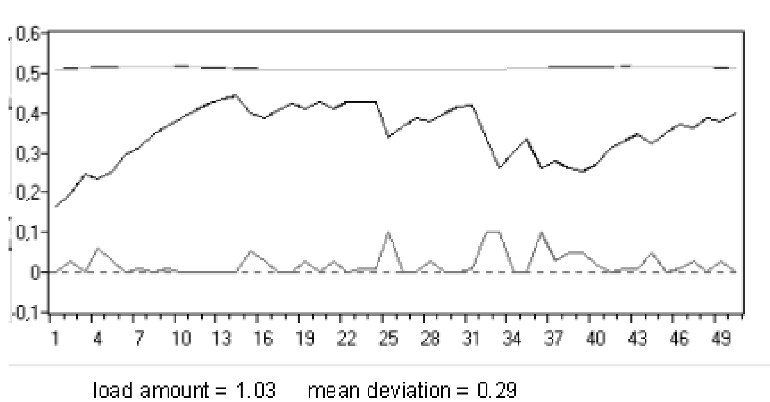

Figure 18 — Improving the load profile from Figure 15 by use of GA-based scheduling.

Currently, a number of projects with separate collaborators are in progress or planned for the near future to test the PerPot software program. These projects will facilitate an analysis of the utility of the approaches presented here.

References

1. Kreider RB, Fry AC, O'Toole ML. 1998. Overtraining in sport. Champaign, IL: Human Kinetics.
2. Maas S, Mester J. 1996. Diagnosis of individual physiological response in elite sport by means of time-series-analyses. In: Marconnet P et al., editors. Frontiers in sport science. The European perspective. Nice, France: University of Nice. p. 98-99.
3. Mader A. 1988. A transcription-translation activation feedback circuit as a function of protein degradation, with the quality of protein mass adaptation related to the average functional load. J Theor Biol 134:2-17.
4. Mester J, Hartmann U, Niessen M, Perl J. 2000. Time series analyses and metamodels for analyses of physiological adaptation. In: Avela J, Komi PV, Komulainen J, editors. 5th Annual Congress of the European College of Sport Science. University of Jyväskylä. p. 75.
5. Mester J, Perl J. 1999. Unconventional simulation and empirical evaluation of biological response to complex high training loads. In: Parisi P, Pigozzi F, Prinzi G, editors. Sport science '99 in Europe. Rome: University Institute of Motor Science. p. 163.
6. Mester J, Perl J. 2000. Grenzen der Anpassungs- und Leistungsfähigkeit aus systemischer Sicht—Zeitreihenanalyse und ein informatisches Metamodell zur Untersuchung physiologischer Adaptationsprozesse. Leistungssport 30, 1:43-51. (In German)
7. Perl J. 2000. Antagonistic adaptation systems: an example of how to improve understanding and simulating complex system behavior by use of meta-models and on-line simulation. 16th IMACS Congress. New Brunswick, NJ: Rutgers University.

About the Author

Jürgen Perl <Perl@informatik.uni-mainz.de> is with Johannes Gutenberg University, FB 17 - Institute of Information, D-55099 Mainz, Germany.

Jürgen Perl has been Chair for Applied Computer Science at the University of Mainz since 1984. Since 1990, he organized the biennial "Workshop on Sport & Computer Science" in Germany. Since 1995, he was speaker of the section "Computer Science in Sport" in the German Association of Sport Science. His main research interests and areas in the field of Computer Science in Sport include: modeling of processes in sport and sport games, computer-based recording and analysis systems in sports, and soft computing approaches in sport.