

# Digital Filter Stepsize Control in DASP3 and its Effect on Control Optimization Performance\*

*K. Meeker*<sup>†</sup>, *C. Homescu*<sup>†</sup>, *L. Petzold*<sup>‡</sup>, *H. El-Samad*<sup>§</sup>,  
*M. Khammash*<sup>§</sup> and *G. Söderlind*<sup>¶</sup>

## Overview

It has long been known that the solutions produced by adaptive solvers for ordinary differential (ODE) and differential algebraic (DAE) systems, while generally reliable, are not smooth with respect to perturbations in initial conditions or other problem parameters. Söderlind and Wang [12, 13] have recently developed a digital filter stepsize controller that has a theoretical basis from control and appears to result in a much smoother dependence of the solution on problem parameters. This property seems particularly important in the control and optimization of dynamical systems, where the optimizer is generally expecting the DAE solver to return solutions that vary smoothly with respect to the parameters. We have implemented the digital filter stepsize controller in the DAE solver *DASP3.1*, and used the new solver for the optimization of dynamical systems. The improved performance of the optimizer, as a result of the new stepsize controller, is demonstrated on a biological problem regarding the heat shock response of *Escherichia coli*.

---

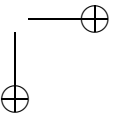
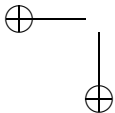
\*This work was supported by DOE DE-FG03-00ER25430, NSF/ITR ACI-0086061, NSF CTS-0205584 and U.S. Army DAAD19-03-D-0004.

<sup>†</sup>Dept. of Computer Science, University of California, Santa Barbara, U.S.A.

<sup>‡</sup>Dept. of Computer Science and Dept. of Mechanical Engineering, University of California, Santa Barbara, U.S.A.

<sup>§</sup>Dept. of Mechanical Engineering, University of California, Santa Barbara, U.S.A.

<sup>¶</sup>Centre for Mathematical Sciences, Lund University, Sweden.



# 1 Introduction

This paper is concerned with the control and optimization of dynamical systems. The problem is given by

$$\begin{aligned} \min_{\mathbf{u}} \Phi(\mathbf{u}) &= \int_0^T \Psi(\mathbf{y}, \mathbf{u}, t) dt \\ \text{subject to } \mathbf{F}(t, \mathbf{y}, \mathbf{y}', \mathbf{u}) &= 0, \quad \mathbf{y}(0) = \mathbf{y}_0(\mathbf{u}), \\ g(\mathbf{u}) &\leq 0 \end{aligned} \tag{1}$$

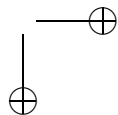
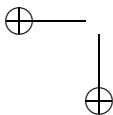
where  $\Psi$  is the objective function,  $\mathbf{F}$  is the DAE system which is a function of state  $\mathbf{y}$  and control  $\mathbf{u}$ , and  $g(\mathbf{u})$  are constraints. In the formulation considered here, the problem is solved by a *shooting* [1] type method: whenever the objective function needs to be evaluated for a given  $\mathbf{u}$ , we first solve the differential algebraic equations for  $\mathbf{y}$ . We selected a derivative-based optimization method for two reasons: a) the optimizer is fast and reliable, and b) the DAE solver *DASPK3.1* is able to efficiently compute the required derivatives via sensitivity analysis [9].

Derivative-based optimizers, while very efficient and robust, assume a substantial amount of smoothness in the problem. If the objective function depends on the numerical solution of a differential equation, it is possible that small discontinuities resulting from adaptive stepsize control decisions (in the ODE or DAE solver) could lead to a less than optimal performance of the optimizer. Hence it seems likely that the performance of the optimizer might be improved by selecting the stepsizes in such a way that the performance of the solver is less sensitive to perturbations in the problem. The digital filter stepsize controller proposed by Söderlind and Wang [12, 13] and implemented in *DASSL* has this property.

We implemented the new digital controller in the large-scale DAE solver *DASPK3.1* [9], which has a facility for sensitivity analysis. The resulting version of *DASPK*, denoted *DASPKmod*, was tested on several simulation and sensitivity problems [11]. It was found that its speed is generally comparable to that of *DASPK3.1*, but its performance is more predictable for small perturbations in the problem or code parameters.

To determine whether the new stepsize controller could lead to faster convergence in the control of dynamical systems, we tested the new solver in the optimization of an interesting problem from systems biology. We employed *DASPKmod* together with the derivative-based optimizer *KNITRO* [14] to solve this optimization problem. The numerical results demonstrate that the optimization process (combined with *DASPKmod*) took substantially fewer iterations compared with the case when *KNITRO* was used in conjunction with *DASPK3.1*.

The remainder of the paper is organized as follows. §2 contains a short description of the differential algebraic (DAE) solver *DASPK3.1*, including the computation of sensitivities (which are needed for the gradient-based optimization). In §3 we compare the characteristics of the original stepsize controller and the new digital filter stepsize controller. The test problem, which involves the optimization of the heat shock response in *E. coli*, is briefly presented in §5. Cast as a multiobjective optimization problem, it is solved using a powerful hybrid interior method,



KNITRO, summarized in §4. The numerical results are provided in §6. Our results are summarized in §7.

## 2 The DAE Solver DASPK

*DASPK3.1* [9] solves a differential algebraic system of the form

$$\begin{aligned}\mathbf{F}(t, \mathbf{y}, \mathbf{y}', \mathbf{u}) &= 0, \\ \mathbf{y}(t_0, \mathbf{u}) &= \mathbf{y}_0,\end{aligned}\tag{2}$$

where  $\mathbf{u}$  are parameters of the problem. The  $k$ -step backward differentiation formula (BDF) method employed by *DASPK3.1* approximates the derivative  $\mathbf{y}'$  using  $k$  past values of the solution  $\mathbf{y}$

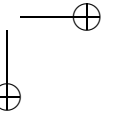
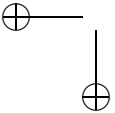
$$\mathbf{F}\left(t_{n+1}, \mathbf{y}_{n+1}, \frac{1}{h\beta_0} \sum_{i=0}^k \alpha_i \mathbf{y}_{n+1-i}, \mathbf{u}\right) = 0.\tag{3}$$

The order of the approximation can be varied by changing the number of past solution values used. A modified Newton method is used to solve the implicit equation for  $y_n$  at each time step. Selection of the time step is the focus of the next section, which describes two approaches for the stepsize controller. The linear systems may be solved by direct methods, or by preconditioned Krylov iteration.

The sensitivities are the derivatives of the state variables  $\mathbf{y}$  with respect to the problem parameters  $\mathbf{u}$ . They will be employed as gradient components, required by the derivative-based optimization approach. These derivatives are computed by the addition of sensitivity equations to the system being solved. The number of additional equations is  $n_s = n_y \cdot n_u$  [10]. Using the notation  $s_i = \frac{\partial \mathbf{y}}{\partial u_i}$ , the new system is given by

$$\begin{aligned}\mathbf{F}(t, \mathbf{y}, \mathbf{y}', \mathbf{u}) &= 0, \\ \frac{\partial \mathbf{F}}{\partial \mathbf{y}} s_i + \frac{\partial \mathbf{F}}{\partial \mathbf{y}'} s'_i + \frac{\partial \mathbf{F}}{\partial \mathbf{u}} &= 0, \quad i = 1, \dots, n_u.\end{aligned}\tag{4}$$

The sensitivity equations are generated using automatic differentiation software ADIFOR [2]. The staggered corrector method in *DASPK3.1* solves the entire system in two steps [6]. First it computes the approximation to the solution  $\mathbf{y}$  to (4) at the next time step using the BDF and Newton iteration. Then it solves the sensitivity equations (5) over the same time step using the BDF discretization and a second Newton iteration. The Jacobian matrix used to solve the original system and the sensitivity system are dependent only on the DAE solution, so they need only be computed once following the solution of the DAE on each time step.



### 3 Stepsize Controller

#### 3.1 Original DASPK approach

DASPK3.1 selects the stepsize  $h_{n+1}$  by

$$h_{n+1} = \left( \frac{\epsilon}{\hat{r}_n} \right)^{\frac{1}{k+1}} h_n, \quad (6)$$

where  $h_{n+1}$  is the next stepsize,  $\epsilon$  is a fraction of the desired error tolerance,  $\hat{r}_n$  is the estimated local error, and  $k$  is the order of BDF employed by DASPK3.1.

The dependence of the local error on the stepsize is described by

$$\hat{r}_n = \hat{\Phi}_n h_n^{k+1}, \quad (7)$$

where  $\hat{\Phi}_n$  is the norm of the principle error function. It has been observed in practice that the output stepsize sequence and the resulting error  $\hat{r}$  of the solver can exhibit large variations due to even very small perturbations in the input. For simulation alone, this is not generally an issue, as long as the solution is still within the error tolerance. But it becomes a critical matter of concern in problems such as optimization of DAE systems, where smoothness of the computed DAE solution is needed by the optimizer.

#### 3.2 Digital filter stepsize controller

Söderlind and Wang observed that the frequency response of the simple stepsize controller used in DASPK, as well as in most other popular DAE and ODE solvers, emphasizes high frequencies. They proposed in [12, 13] a stepsize low-pass filter to produce better frequency response while preserving stability. We have replaced the original stepsize controller in DASPK3.1 by their filter.

With this controller, the stepsize is updated according to

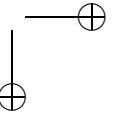
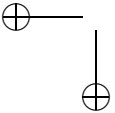
$$h_{n+1} = \left( \frac{\epsilon}{\hat{r}_n} \right)^{\beta_1} \left( \frac{\epsilon}{\hat{r}_{n-1}} \right)^{\beta_2} \left( \frac{h_n}{h_{n+1}} \right)^{-\alpha_2} h_n, \quad (8)$$

where  $k\beta_1 = k\beta_2 = \alpha_2 = \frac{1}{4}$ ,  $k = \hat{p} + 1$  and  $\hat{p}$  is the order of convergence.

The filter is named for the performance characteristics that it is controlling: the step size  $h$ , the *orders of dynamics*, *adaptivity*, and *filter* (according to this naming scheme, the original stepsize controller in DASPK3.1 is H110, while the new filter is denoted as H211). The H110 and H211 controllers differ in order of dynamics and filter order, but have the same order of adaptivity.

**Definition 1.** *The order of dynamics  $p_D$  of the closed loop system is defined to be the degree of  $q$  in the denominator of the stepsize transfer function  $H_{\hat{\Phi}}(q)$ .*

**Definition 2.** *Letting the error transfer function  $R_{\hat{\Phi}}(q)$  have all its poles strictly inside the unit circle, if  $|R_{\hat{\Phi}}(q)| = O(|q-1|^{p_A})$  as  $q \rightarrow 1$ , the order of adaptivity  $p_A$  of the controller is defined to be  $p_A$ .*



**Definition 3.** Letting the stepsize transfer function  $H_{\hat{\Phi}}(q)$  have all its poles strictly inside the unit circle, if  $|H_{\hat{\Phi}}(q)| = O(|q+1|^{p_F})$  as  $q \rightarrow -1$ , the step size filter order at  $q = -1$  is  $p_F$ .

The *stepsize* and *error transfer functions* are equations describing the stepsize and error response to controller input, and can be derived from the stepsize recursion relations (8) and (6), and the assumption that the local error is proportional to the step size

$$\hat{r}_n = \hat{\Phi}_n h_n^k, \quad (9)$$

where  $\hat{\Phi}_n$  is the norm of the principal error function. Taking logarithms of (8-9), solving for  $\log \hat{r}$  and  $\log h$ , and setting  $\log \epsilon$  (a constant) to zero, yields the stepsize and error transfer functions for the H110 controller

$$H_{\hat{\Phi}}(q) = -\frac{1}{kq}, \quad R_{\hat{\Phi}}(q) = \frac{q-1}{q},$$

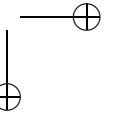
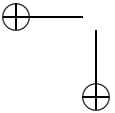
and the stepsize and error transfer functions for the H211 controller

$$H_{\hat{\Phi}}(q) = -\frac{\beta_1 q + \beta_2}{(q-1)(q+\alpha_2) + k(\beta_1 q + \beta_2)}, \quad R_{\hat{\Phi}}(q) = \frac{(q-1)(q+\alpha_2)}{(q-1)(q+\alpha_2) + k(\beta_1 q + \beta_2)}.$$

A controller is stable if all poles (roots of the denominator of  $H_{\hat{\Phi}}(q)$ ) are strictly inside the unit circle. The single pole  $q = 0$  of the elementary controller's stepsize transfer function  $H_{\hat{\Phi}}(q)$  is located at the origin so the H110 controller is highly stable. The order of dynamics quantifies how quickly and with what function the stepsize response decays to zero from an initial value, given no input disturbance (the homogeneous solution). The elementary controller has dynamic order one, so it decays linearly to a stepsize dependent on the initial value. The filter order determines the order of repeated averaging which smooths the stepsize. This elementary controller has filter order zero; therefore it does no smoothing of successive stepsizes. Given that the controller is stable, has a linear decay function, and a reasonably fast response, the next property to examine is the controller's frequency response in both the stepsize and error outputs.

The stepsize frequency response  $|H_{\hat{\Phi}}(e^{i\omega})| \equiv 1$  for  $\omega \in [0, \pi]$ , so it is independent of the frequency  $\omega$ . The output stepsize  $h$  will have the same spectral content as the input disturbance  $\hat{\Phi}$  because all frequencies have equal weighting. The error frequency response of a stable controller is  $|R_{\hat{\Phi}}(e^{i\omega})| = O(\omega^{p_a})$  as  $\omega \rightarrow 0$ . The output error  $\hat{r}$  will emphasize high frequencies of the input disturbance  $\hat{\Phi}$ . The combined result is the undesired emphasis on high frequencies, making the output stepsize and error rougher than the input disturbance, and prompting the design of a digital filter stepsize controller.

The stability of the H211 controller is not as straightforward, and must be guaranteed by careful filter parameter selection [12]. The digital filter stepsize controller has a higher order of dynamics  $p_D = 2$  and a higher filter order  $p_F = 1$ . The effect of these improvements is to give it a faster and smoother response. The order of adaptivity is unchanged from the elementary controller, so the error frequency response again will emphasize high frequencies of the input disturbance.



However, the stepsize controller has been designed to shape the frequency response to be more uniform across all frequencies. This is done by placing the zeros (roots of the numerator of the stepsize transfer function) at frequencies we desire to suppress.

## 4 The Optimization Solver KNITRO

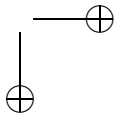
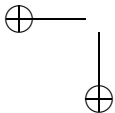
The heat shock optimization problem solved in this paper is a multiobjective optimization problem. In single-objective optimization, the goal is to find the best solution, which corresponds to the minimum or maximum value of the objective function. In multiobjective optimization with conflicting objectives, there is no single optimal solution. The interaction among different objectives gives rise to a set of compromise solutions in which no objective can be improved without degrading another objective. Such a solution is called “Pareto optimal”, after the economist who first analyzed income distribution in this way. Moreover, the underlying differential algebraic (DAE) system is very stiff.

Thus we needed to select a very robust algorithm from the class of gradient-based methods. Our choice was KNITRO, an optimization solver presented in [14, 3, 15]. Its excellent performance is due to the fact that it incorporates within the interior point method two powerful tools for solving nonlinear problems: sequential quadratic programming (SQP) and trust region techniques. SQP efficiently handles nonlinearities in the constraints. Trust region strategies allow the algorithm to treat convex and non-convex problems uniformly, permit the direct use of second derivative information and provide a safeguard in the presence of nearly dependent constraint gradients.

KNITRO is a hybrid interior method (also known as barrier method), where the original problem is replaced by a series of barrier subproblems controlled by a barrier parameter. The algorithm [3, 15] uses trust regions and a merit function to promote convergence. It performs one or more minimization steps on each barrier problem, then decreases the barrier parameter. The step computed at every iteration is decomposed into a normal step, whose goal is to improve feasibility, and a tangent step, toward optimality.

A typical iteration computes a primary step by solving primal-dual equations (using direct linear algebra) and performs a line search to ensure decrease in a merit function. In order to obtain global convergence in the presence of nonconvexity and Hessian or Jacobian singularities, the primary step is replaced, under certain circumstances, by a safeguarding trust region step. The second derivatives of the objective function and constraints are approximated using quasi-Newton updating.

The DAE solver is employed for two purposes in the optimization. First, it computes the underlying numerical solution, evaluates the integrand in the objective function, and through a quadrature equation, computes the integral appearing in the optimization objective function. Second, it calculates the sensitivities of the objective function, which form the gradients needed for the optimization.



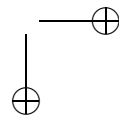
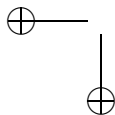
## 5 Heat Shock Model

Survival of organisms such as *E. coli* in extreme conditions, like exposure to high temperatures, has necessitated the evolution of stress response networks that detect and respond to environmental changes. The consequences of unmediated heat at the cellular level are the unfolding, misfolding, or aggregation of cell proteins, which threaten the life of the cell. Cells respond to heat stress by initiating the production of heat-shock proteins (HSPs), whose function is to refold denatured proteins into their native states. The heat shock response system is among the most important and challenging protective systems of the cell. This system consists of intricate control mechanisms for detecting the presence of heat related protein damage, and initiating an appropriate response through the synthesis of HSPs. In the cytoplasmic heat shock response system of the bacterium *E. coli*, HSPs at 30°C represent less than 2% of the total protein content in the cytoplasm, whereas at 46°C, close to the growth cut-off for the organism, HSPs account for 20-25% of the total protein content [7].

HSP levels are regulated through an intricate architecture of feedback loops centered around the  $\sigma^{32}$ -factor. The  $\sigma^{32}$ -factor regulates the transcription of the heat shock proteins under normal and stress conditions. The enzyme RNA polymerase (RNAP), bound to  $\sigma^{32}$ , recognizes the heat shock gene promoters and transcribes specific heat shock genes. This transcription process is controlled through the tight regulation of  $\sigma^{32}$ , whose synthesis, activity, and stability are modulated by intricate feedback and feedforward loops that incorporate information about the temperature and the folding state of the cell.

Building on the model of El-Samad et al. [4, 8], we formulated an optimization problem which asked whether the heat shock response is naturally balancing the cost of producing HSPs, versus the benefits in protein folding resulting from their presence in large number in the cell. Our motivation for addressing this problem is the following. A first glance at the heat shock response system would suggest that overproducing HSPs in anticipation of heat is the best strategy. However, this would only be true if the sole objective considered was the efficient folding of proteins. On the other hand, when one considers the HSPs' use of the cell's limited resources, it becomes apparent that an excess of HSPs can be expensive. In light of this analysis, the following questions arise naturally: *How does the heat shock response balance these two conflicting performance objectives? Is it solving an optimization problem whose result is a fine balance between the protective effect of HSPs and the metabolic burden of over-expressing them?*

We sought an answer to these questions by formulating and solving a multi-objective optimization problem where both the cost of producing and maintaining HSPs, in addition to the cost of unfolded proteins, are considered as optimization criteria. We used the mathematical model of the heat shock response reported in [8, 4]. The model employs first order kinetics (law of mass-action) to describe the various components of the heat shock system, namely the synthesis of HSPs and their regulator  $\sigma^{32}$ , in addition to protein folding and the association/dissociation activity of molecules. Reactions involving association/dissociation of molecules are often faster than synthesis and degradation of new molecules. Therefore, the model



yields itself naturally to a separation of time scales, hence numerical stiffness. This stiffness imposed the transformation of the differential equations that describe the fast states into algebraic constraints through a partial equilibrium approximation, yielding an index-one system of differential algebraic equations (DAE). The DAE describes the evolution of 11-differential variables, coupled to the evolution of 20-algebraic variables, and includes 32 parameters. This structure of the heat shock model depicts a real biological system with a substantial amount of complexity and high dimensionality, and makes an ideal testbed for the application of our new numerical techniques.

## 5.1 Pareto optimality

Solving our problem involves the simultaneous optimization of multiple objectives. The principle of multiobjective optimization is different from that of single-objective optimization. In single-objective optimization, the goal is to find the best solution, which corresponds to the minimum or maximum value of the objective function. But there is no single optimal solution in multiobjective optimization (due to conflicting objectives). Instead, we obtain a set of “compromise” solutions, largely known as the Pareto-optimal solutions.

Many other examples of multiobjective optimization can be found among real-world engineering design or decision-making problems. For example, in bridge construction, a good design is characterized by low total mass and high stiffness; in aircraft design simultaneous optimization of fuel efficiency, payload, and weight is required; in chemical plant design, or in design of a groundwater remediation facility total investment and net operating costs are considered; in the traditional portfolio optimization problem, minimum risk and maximum fiscal return are attempted.

Each solution of the Pareto optimal set is not dominated by any other solution. In going from one solution to another, it is not possible to improve on one objective without making at least one of the other objectives worse. Using multiobjective analysis, decision-makers can better assess the trade offs between different objectives. Although they cannot identify a solution that is clearly the best, they can discover a set of near optimal solutions, and have reasonable grounds to make sensible decisions and avoid clearly poor alternatives.

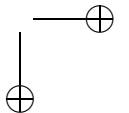
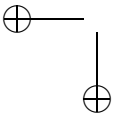
## 5.2 Multiobjective optimization formulation for heat shock

Using the framework of multiobjective optimization, the heat shock problem can be formulated as

$$\begin{aligned} \min_{\mathbf{u}} \mathbf{F}_{\alpha}(\mathbf{u}) &\equiv \int_0^T y_1^2(t, \mathbf{u}) dt + \alpha \int_0^T y_2^2(t, \mathbf{u}) dt \\ \text{subject to } g(\mathbf{u}) &\leq 0, \end{aligned} \quad (10)$$

where  $y_1$  and  $y_2$  are specific components of  $\mathbf{y}$ , the solution of the DAE describing the heat shock model.

The variables  $y_1$  and  $y_2$  correspond to the levels of unfolded proteins and, respectively, chaperons. For every value of the weight  $\alpha$ , the parameters of the



model, lumped in the control vector  $\mathbf{u}$ , should be determined in order to minimize the cost function  $F_\alpha(\mathbf{u})$ . The value of  $\alpha$  represents the relative weight (importance) for one of the objectives versus the other. It should be mentioned that different values of  $\alpha$  yield different optimal solutions. The control vector  $\mathbf{u}$  is composed of 32 model parameters for the heat shock model, e.g., binding and degradation coefficients or synthesis rates for different proteins,  $\sigma$  factors and chaperons [8].

We denote by  $\mathbf{u}_\alpha^{OPTIM}$  the value of the parameter vector that generates the optimal solution for the corresponding weight  $\alpha$ . By plotting

$$\int_0^T y_1^2(t, \mathbf{u}_\alpha^{OPTIM}) dt \quad \text{vs.} \quad \int_0^T y_2^2(t, \mathbf{u}_\alpha^{OPTIM}) dt \quad (11)$$

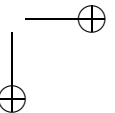
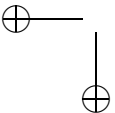
we obtain a Pareto optimal curve. Using this curve, we can then answer our questions about the optimality of the heat shock system by inspecting where the point of operation of the wild type system falls with respect to that curve. Fig. 1 shows the Pareto optimal curve (red) generated by the solution of our optimization problem. The plot also shows the cost of chaperons and unfolded proteins for wild type heat shock in *E. coli*. The closeness of this point to the optimal curve indicates that the heat shock response is operating in a near-optimal regime, perhaps as a result of a long evolutionary past that converged to a near-optimal solution. To check that our finding is substantial and rule out the possibility that the structure of the model itself is generating near-optimal solutions for all parameter values, we randomly generated combinations of the parameters and computed their costs. These randomly chosen parameter combinations mostly yielded operating points well inside the non-optimal region, therefore substantiating our conclusions. Three illustrative points are shown in Figure 1.

Since the main objective of our paper is to present the improvement in the optimizer's performance as a result of the new stepsize controller, we provide only a brief discussion of the Heat Shock Response and its numerical Pareto curve. For an in-depth description the reader is referred to El-Samad et al. [5].

## 6 Numerical Efficiency Comparison

To reduce the influence of the unsteady part of the heat shock model on the optimal solution, we solved the heat shock problem in two optimization stages. The goal of the first stage was to reach a steady-state solution for the DAE system. The values of the parameters  $\mathbf{u}$  and state variables  $\mathbf{y}$  for the *wild type heat shock* were given as initial conditions for the control parameter  $\mathbf{u}$ , respectively for the state variables for the DAE system describing the underlying process. The number of iterations was fixed at 1000 for each weight  $\alpha$  in the objective function (10). The number was selected based on numerical experiments which showed that for a larger number of iterations the cost functional decreases by an extremely small amount (less than 0.0001%) at each additional iteration, thus not promising enough improvement in the objective function to warrant the extra computational effort.

For the second stage of the optimization the initial conditions for the control vector  $\mathbf{u}$  were taken to be the optimal values obtained at the end of the first optimization run. The initial conditions for the state variables  $\mathbf{y}$  were obtained from



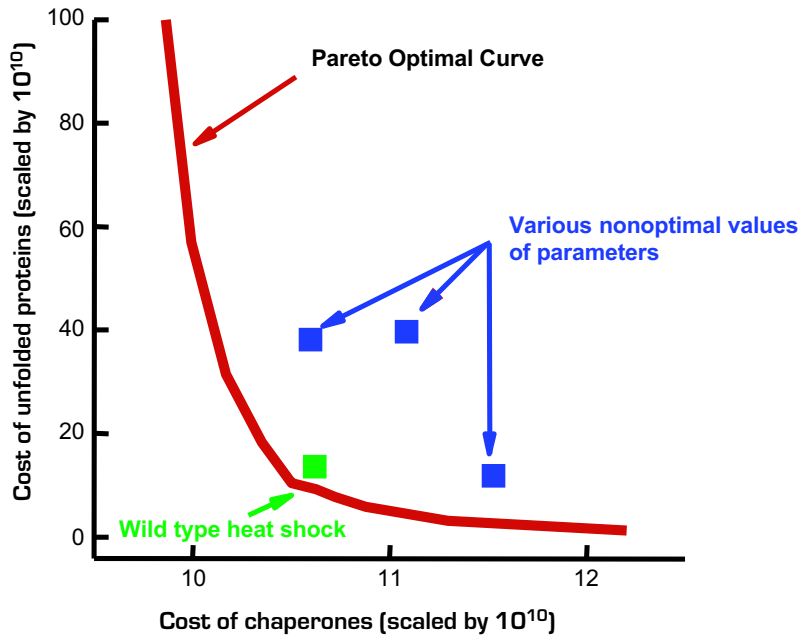
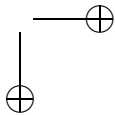
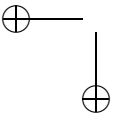


Figure 1. Pareto optimal design of the heat shock response.

the steady state of the heat shock model, which was found via a run of the model using the optimal values of  $\mathbf{u}$  from stage 1. For the second optimization stage, the number of iterations was bounded but not fixed. By allowing this number to vary, we were able to assess the improvements in the optimizer due to the digital stepsize controller.

We compare the optimal solutions and the computational efficiency with the solution computed with *DASPK3.1* and, respectively, *DASPKmod*. As mentioned before, for a multiobjective optimization problem it is not expected to have a single optimal solution, but rather to identify a set of good (near optimal) solutions that reduce the value of the objective functional while satisfying the constraints (the DAE system). The value of the objective function obtained using *DASPKmod* was similar (within the tolerance error) to the value of the objective function after optimization with *DASPK3.1* as the DAE solver. Next, we turn our attention to the computational costs of the two approaches. There is a clear advantage in both the computational time and the number of iterations for the optimization based on *DASPKmod*.

Figure 2 compares the efficiency of the two stepsize controllers for three values of the parameter  $\alpha$ . In the first stage the number of *DASPK* steps was smaller



with the digital filter stepsize controller than with the original stepsize controller, indicating that the average stepsize was larger with the new filter. However, the processor time taken to solve the problem was not consistently reduced. In the second stage the number of optimization steps and the corresponding time used to solve the problem were dramatically reduced. This was true even when the DASSL number of steps were not reduced when  $\alpha = 20$ . We conjecture that the performance improvement is due to the smoother behavior of the solution components with respect to perturbation of the problem parameters.

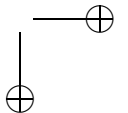
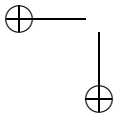
## 7 Conclusion

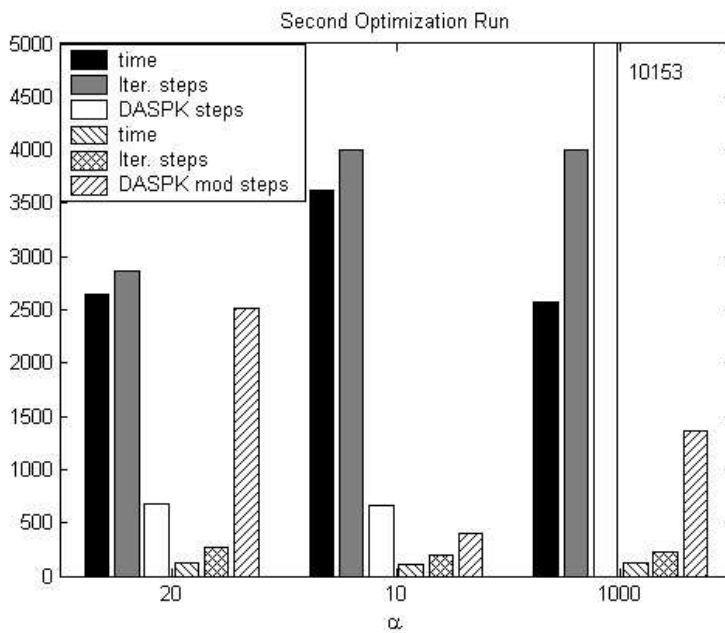
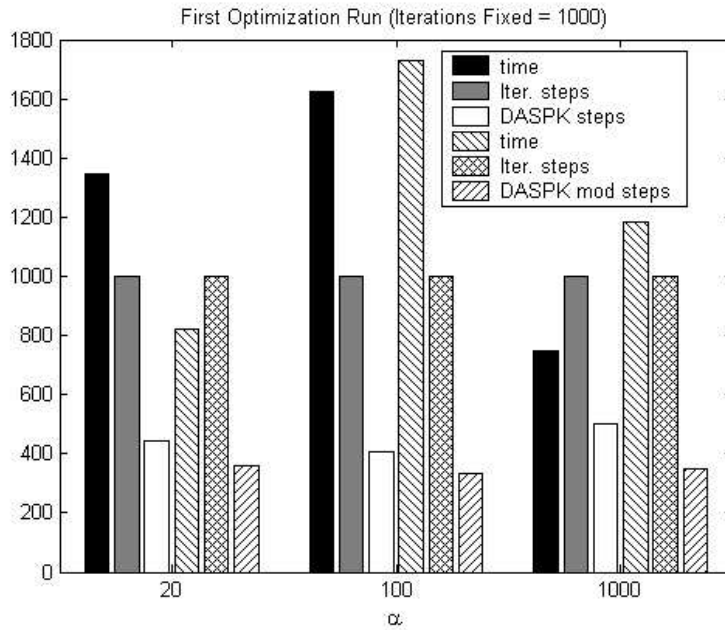
Many optimization problems consist of the minimization of a cost functional which is constructed from solutions of underlying differential systems, obtained using adaptive (DAE or ODE) solvers. Thus the iterative process is influenced by the numerical approach chosen to solve these differential systems, in particular by the smoothness of the selection of the time step size for the DAE or ODE solver.

We have obtained numerical results which show that, by choosing a smoother stepsize controller, the performance of the optimizer is greatly improved. Our methodology was tested on a real-life systems biology problem, the heat shock response of *E. coli*. The number of optimization iterations was reduced, while maintaining the same accuracy or better for the numerical optimal solution.

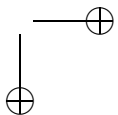
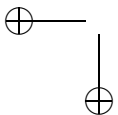
## Acknowledgements

We would like to thank Dr. Lina Wang of the Centre for Mathematical Sciences at Lund University, Sweden, for providing us with her digital stepsize controller implemented in DASSL.



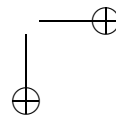
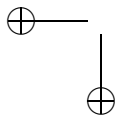


**Figure 2.** Performance of the optimizer versus the weight parameter  $\alpha$ . The underlying DAE system (2) for the heat shock problem is solved using either the original controller (DASP3.1) or the digital filter stepsize controller (DASP3.mod). The solid bars are the statistics for DASP3.1 and the striped bars denote DASP3.mod results. The statistics describe the CPU time (in seconds), the number of iteration steps and the number of DASP3 steps.



# Bibliography

- [1] U. M. ASCHER, R. M. MATTHEIJ AND R. D. RUSSELL, *Numerical solution of boundary value problems for ordinary differential equations*, SIAM Press, 1995.
- [2] C. BISCHOF, A. CARLE, G. CORLISS, A. GRIEWANK AND P. HOVLAND, *ADIFOR-Generating derivative codes from Fortran programs*, Sci. Program., 1 (1992), pp. 11–29.
- [3] R. H. BYRD, M. E. HRIBAR AND J. NOCEDAL, *An interior point algorithm for large-scale nonlinear programming*, SIAM J. Optim, 9 (1999), pp. 877–900.
- [4] H. EL-SAMAD, H. KURATA, J.C. DOYLE, C.A. GROSS AND M. KHAMMASH, *Surviving heat shock: control strategies for robustness and performance*, Proc. Nat. Acad. Sci. USA., 102 (2005), pp. 2736–2741.
- [5] H. EL-SAMAD, C. HOMESCU, M. KHAMMASH AND L. PETZOLD, *The heat shock response: optimization solved by evolution?*, Fifth International Conference on Systems Biology, ICSB 2004, Oct 9-13, 2004, Heidelberg, Germany.
- [6] W. FEEHERY, J. TOLSMA AND P. BARTON, *Efficient sensitivity analysis of large-scale differential-algebraic systems*, Appl. Numer. Math., 25 (1997), pp. 41–54.
- [7] GROSS, C.A., *Escherichia coli and Salmonella: cellular and molecular biology*, ASM Press, eds F.C. Neidhart, 1996, pp. 1384–1394.
- [8] H. KURATA, H. EL-SAMAD, T.M. YI, M. KHAMMASH AND J.C. DOYLE, *Feedback regulation of the heat shock response in E. Coli*, 40th IEEE Conference on Decision and Control, 2001, pp. 837–842.
- [9] S. LI AND L.R. PETZOLD, *Design of new DASPCK for Sensitivity Analysis*, Technical Report TRCS99-28, UCSB, 1999.
- [10] S. LI AND L. PETZOLD, *Software and algorithms for sensitivity analysis of Large-scale differential algebraic systems*, J. Comput. Appl. Math., 1-2 (2000), pp. 131–145.
- [11] K. MEEKER, *Digital filter stepsize control in DASPCK and its effect on control optimization performance*, M.Sc. Thesis, UCSB, 2004.



- [12] G. SÖDERLIND, *Digital filters in adaptive time stepping*, ACM T. Math. Software, 29 (2003), pp. 1–26.
- [13] G. SÖDERLIND AND L. WANG, *Adaptive time-stepping and computational stability*, ACM T. Comp. Logic., V (2002), pp. 1–14.
- [14] R. A. WALTZ AND J. NOCEDAL, *KNITRO User's Manual*, Technical Report OTC 2003/05, Optimization Technology Center, Northwestern University.
- [15] R. A. WALTZ, J. L. MORALES, J. NOCEDAL AND D. ORBAN, *KNITRO-Direct: A hybrid interior algorithm for nonlinear optimization*, Technical Report OTC 2003/10, Optimization Technology Center, Northwestern University.

