In: A. Bode, T. Ludwig, W. Karl, R. Wismüller (eds.): Euro-Par 2000 – Parallel Processing.
Lecture Notes in Computer Science 1900 (Springer-Verlag, 2000) 829-834.

A Parallel Optimization Scheme for Parameter Estimation in Motor Vehicle Dynamics

Torsten Butz[1], Oskar von Stryk[1], and Thieß-Magnus Wolter[2]

[1] Chair of Numerical Analysis (M2), Zentrum Mathematik,
Technische Universität München, D-80290 München, Germany
http://www-m2.ma.tum.de

[2] TESIS DYNAware, Implerstraße 26, D-81371 München, Germany
http://www.tesis.de

Abstract. For calibrating the vehicle model of a commercial vehicle dynamics program a parameter estimation tool has been developed which relies on observations obtained from driving tests. The associated nonlinear least-squares problem can be solved by means of mathematical optimization algorithms most of them making use of first-order derivative information. While the complexity of the investigated vehicle dynamics program only allows the objective gradients to be approximated by means of finite differences, this approach enables significant savings in computational time when performing the additionally required evaluations of the objective function in parallel. The employed low-cost parallel computing platform which consists of a heterogeneous PC cluster is well suited for the needs of the automotive suppliers and industries employing vehicle dynamics simulations.

1 Introduction

The numerical simulation of vehicle dynamics has gained considerable significance in automotive development, since it enables the thorough investigation of a novel vehicle in advance. Besides reducing the need for physical prototyping, real-time simulations may be used within hardware-in-the-loop test-benches which allow active control units, such as anti-lock braking systems and electronic stability programs, to be tested without danger for test driver and vehicle.

The development of complex electronic devices requires the virtual car to reproduce the behavior of the real vehicle in detail. Therefore, we employ a sophisticated vehicle model which comprises a suitable multibody system, including force elements and kinematical connections, as well as a realistic tire model. The use of a tailored modeling technique enables the entire vehicle dynamics to be described by a large system of ordinary differential equations.

Specifically for the use in a test-bench the calibration of the vehicle model need often be accomplished on the spot. For this purpose, nonlinear optimization algorithms and careful numerical differentiation can be combined to yield a parallel parameter estimation scheme which is suitable for low-cost computing platforms such as heterogeneous PC networks.

2 Simulation of Full Motor Vehicle Dynamics

The vehicle dynamics program *veDYNA* [1] which has been employed for the following investigations is developed and commercially distributed by TESIS DYNAware, München.

The vehicle model in *veDYNA* consists of a system of rigid bodies comprising the vehicle body, the axle suspensions and the wheels. In addition, partial models are employed to depict the characteristics of the drive train, the steering mechanism and the tires. The use of suitable minimum coordinates and generalized velocities avoids the need for algebraic constraints in the equations of motion [9]. Thus, the vehicle dynamics can fully be described by a system of 56 highly nonlinear ordinary differential equations. Due to the stiffness of the system its numerical integration is carried out by a semi-implicit Euler scheme.

For a realistic implementation of virtual test drives on the computer also models for the driver and the road have been developed [3].

The numerical results obtained from *veDYNA* show good agreement with real vehicle behavior. Simulations with time steps in the range of milliseconds may be carried out in real-time on reasonable PC hardware.

3 Estimation of Vehicle Parameters

The equations of motion for the vehicle model in *veDYNA* are summarized by

$$\dot{x}(t) = g\left(x(t), p, t\right) \tag{1}$$

with suitable initial values

$$x(t_0) = x_0 \ . \tag{2}$$

Here, $x(t) \in \mathbb{R}^{n_x}$ comprises the vehicle's state variables, and $p \in \mathbb{R}^{n_p}$ denotes the model parameters of interest which are constant for all times t. To adjust their values to the observed vehicle behavior, the nonlinear least-squares problem

$$\underset{p \in \mathbb{R}^{n_p}}{\text{minimize}}\ r(p) := \frac{1}{2}\left\|f(p)\right\|_2^2 := \frac{1}{2}\sum_{j=1}^{n_t}\sum_{i \in I_j}\left(\eta_{ij} - x_i(t_j, p)\right)^2 \tag{3}$$

must be solved. Here, η_{ij}, $i \in I_j$, are measurements of selected vehicle state variables at the times t_j throughout a driving test, and $x(t,p)$ denotes the corresponding numerical solution of (1), (2). Often additional box constraints

$$l_i \le p_i \le u_i, \quad i = 1, ..., n_p, \tag{4}$$

on the parameter range have to be considered which shall ensure optimization results compatible with the real vehicle properties.

For the solution of (3), (4) several gradient-based optimization methods as well as an evolutionary algorithm have been investigated [2]. In the sequel, we present results obtained from the Gauss-Newton method NLSCON [8], the Levenberg-Marquardt algorithm LMDER [7], the sequential quadratic programming method NLSSOL [5], and the implicit filtering code IFFCO [6], which is designed for solving noisy minimization problems.

4 Parallel Optimization

For the solution of the parameter estimation problem a program frame was implemented which integrates *veDYNA* in the course of the optimization [2]. Due to the complexity of the employed vehicle model and the closely coupled numerical integration, the required objective derivatives cannot be determined by automatic or internal numerical differentiation techniques, but have to be approximated by means of finite differences.

For the optimization with NLSCON, LMDER and NLSSOL the partial derivatives $\partial_i r(p) = f(p)^T \partial_i f(p)$ are obtained from the one-sided differences

$$(\partial_i f(p))_{\pm h_i} = \frac{f(p \pm h_i e_i) - f(p)}{\pm h_i} \tag{5}$$

depending on the feasibility of $p + h_i e_i$ or $p - h_i e_i$. Here, $e_i \in \mathbb{R}^{n_p}$ denotes the i-th canonical unit vector, and $h_i > 0$ is a finite difference increment which must be chosen carefully such as to account for truncation, condition, and rounding errors. The implicit filtering code IFFCO makes use of the central differences

$$(\partial_i r(p))_{2h_i} = \frac{r(p + h_i e_i) - r(p - h_i e_i)}{2h_i} \tag{6}$$

provided that both points are feasible; otherwise a one-sided difference is used as well. Accordingly, the computation of the gradient $\nabla r(p)$ requires n_p up to $2\,n_p$ additional evaluations of the objective function.

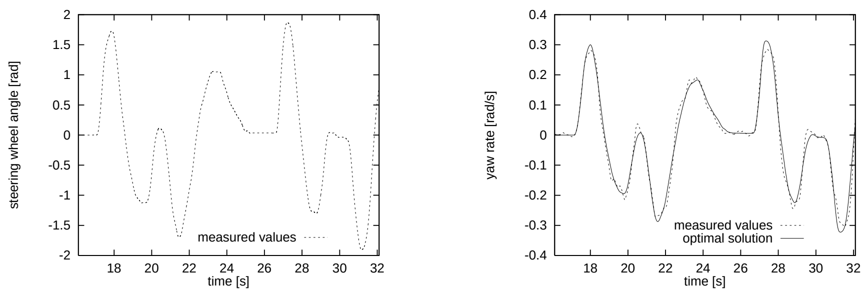
Since most effort is spent on the repeated integration of (1), (2), the computational time is much reduced by distributing these evaluations among further processors. For the one-sided differences the maximum speed-up is achieved, if n_p additional processors are available. In case of the centered differences one of the additionally required evaluations is performed by the client process, since the objective value at the current iterate need not be computed.

The communication between client and server processes across the network is handled by remote procedure calls. For this purpose, the ONC RPC library from Sun Microsystems, ported to Microsoft Windows, is used [4]. The exchange of data is done via the UDP transport protocol, since only arguments of moderate size are communicated.

5 Results

The above parameter estimation scheme was successfully employed to adjust the lateral vehicle dynamics properties in the *veDYNA* model of a passenger car [2]. Appropriate values for the remaining coefficients of the vehicle model had been validated by TESIS DYNAware beforehand.

The underlying data which was provided by an automotive supplier consisted of the steering wheel angle (cf. Fig. 1a) and the corresponding vehicle yaw rate recorded during multiple lane changes. The actual steering maneuver was preceded by a speed-up phase of 16.1 seconds. The sought vehicle parameters were given by the x-coordinate of the center of gravity and the cornering stiffnesses at the front and rear wheels which determine the lateral tire forces.



(a) (b)

Fig. 1. Steering wheel angle (a) and vehicle yaw rate (b) for the lane change maneuver.

The optimization was carried out on a heterogeneous Windows NT 4.0 and Windows 98 network at TESIS DYNAware, München. Initial guesses

$$p^0 = (-1.242, 27075.5, 27075.5)^T \tag{7}$$

were chosen according to the default parameter values for the employed *veDYNA* vehicle model. The associated least-squares residual was $r(p^0) = 0.96115$.

The optimization produced a minimum residual $r(p^*) = 0.30324$ which was assumed for the parameter values

$$p^* = (-1.298, 16914.9, 15000.5)^T \tag{8}$$

computed by IFFCO. A comparison between the observed vehicle yaw rate and the corresponding simulation results for (8) is depicted in Fig. 1b. Good agreement is achieved between both characteristics. However, the small values of the estimated stiffnesses indicate that the kinematical axles of the vehicle model cannot depict the elastic properties of the actual suspension system exactly.

For reasons of comparison the numerical optimization was also carried out sequentially. In this case, the optimization including the computation of the gradients was done on a Dell 400 MHz PC where for each objective evaluation a CPU time of 8.1 seconds was needed.

In the parallel framework, the optimization was running on the same machine. The three function evaluations for the one-sided differences in NLSCON, LMDER and NLSSOL were performed on two Siemens 450 MHz PCs and a Dell 333 MHz notebook where the objective evaluations took 7.3 seconds and 8.8 seconds of CPU time respectively. The two additional evaluations required by IFFCO were carried out on a further Dell 333 MHz notebook and a Siemens 300 MHz PC. The corresponding CPU times were given by 8.8 and 9.7 seconds.

Table 1 shows a comparison between the results obtained from the different optimization codes [2]. The specified values consist of the least-squares residual at the respective optimal solution and the CPU times t_{seq} and t_{par} which were needed for the sequential and the parallel optimization. Their ratio, i. e., the achieved parallel speed-up, is given in the last column. Also listed is the number n_{seq} of objective evaluations during the entire optimization, and the share n_{par} that was performed by the client process in the parallel approach.

Table 1. Comparison of the computational results for the sequential and the parallel parameter estimation schemes.

Algorithm	$r(p^*)$	n_{seq}	t_{seq} [s]	n_{par}	t_{par} [s]	$t_{\text{seq}}/t_{\text{par}}$
IFFCO	**0.30324**	151	1223.6	64	543.1	2.25
LMDER	0.30341	52	**422.0**	25	207.8	2.03
NLSCON	0.30455	72	583.7	30	254.4	2.29
NLSSOL	0.30340	80	648.3	20	**185.3**	**3.50**

Applied to this problem the mentioned algorithms have produced meaningful parameter estimates with reasonably small residuals. The parallel execution of the finite difference computations reduces the required CPU times for all algorithms by more than a half. For NLSSOL, the computational time can be reduced to almost 25%, if equally fast remote processors are available. In case of the remaining optimization codes the achieved speed-up is significantly lower, since the employed line-search strategies do not allow a completely parallel treatment.

References

1. Anonymous: *veDYNA* User's Guide. TESIS DYNAware, München (1997)
2. Butz, T.: Parameter Identification in Vehicle Dynamics. Diploma Thesis, Zentrum Mathematik, Technische Universität München (1999)
3. Chucholowski, C., Vögel, M., von Stryk, O., Wolter, T.-M.: Real time simulation and online control for virtual test drives of cars. In: Bungartz, H.-J. et al. (eds.): High Performance Scientific and Engineering Computing. Lecture Notes in Computational Science and Engineering, Vol. 8. Springer-Verlag, Berlin (1999) 157–166
4. Gergeleit, M.: ONC RPC for Windows NT Homepage. World Wide Web, http://www.dcs.qmw.ac.uk/~williams/nisgina-current/src/rpc110/oncrpc.htm (1996)
5. Gill, P.E., Murray, W., Saunders, M.A., Wright, M.H.: User's Guide for NPSOL 5.0: A Fortran Package for Nonlinear Programming. Numerical Analysis Report 98-2, Department of Mathematics, University of California, San Diego (1998)
6. Gilmore, P.: IFFCO: Implicit Filtering for Constrained Optimization, User's Guide. Technical Report CRSC-TR93-7, Center for Research in Scientific Computation, North Carolina State University, Raleigh (1993)
7. Moré, J.J.: The Levenberg-Marquardt algorithm: implementation and theory. In: Dold, A., Eckmann, B. (eds.): Numerical Analysis. Lecture Notes in Mathematics, Vol. 630. Springer-Verlag, Berlin Heidelberg (1978) 105–116
8. Nowak, U., Weimann, L.: A family of Newton codes for systems of highly nonlinear equations. Technical Report TR 91-10, ZIB, Berlin (1991)
9. Rill, G.: Simulation von Kraftfahrzeugen. Vieweg, Braunschweig (1994)