Comparison of Parallel Simulation Techniques
Heterogeneous MP-system / FSIMUL-P

The simulation package FSIMUL-P is a supplement of the well known block oriented simulation language FSIMUL and permits parallel simulation of very complex industrial processes by means of distributed computer environments. The goal of developing the package was to increase computer power for simulation. Therefore, all blocks of FSIMUL-P are distributed within a multiple processor environment, so that each processor simulates only one subsystem of the model. The communication between the processors is based on virtual interprocessor-communication-channels.

The computer environment at the Institute of Automatic Control at the Ruhr-University of Bochum consists of UNIX-based workstations and PC-based workplaces, all connected by means of a TCP/IP network. In addition, a digital signal processor (DSP) card TMS320C40 and three transputer boards T800 as plug-in-cards (AT-bus) are available for the PCs. The master program FSIMUL-P was developed on a 386/486-based PC-system as a protected mode application in 32-bit technology. The worker process was written in C and compiled for all existing computer systems. The user has not to care for the management of the heterogeneous components of the simulation hardware.

A comfortable window oriented user interface enables an efficient work with the package. An integrated structure editor enables a very attractive model building. The firmware of FSIMUL-P consists of about 150 different block operations. The number of blocks used for a simulation model depends on the available system memory only. The user can program his complex simulation model in 16 implemented simulation levels. For every level, the user can decide for one of the processors available as part of the computing network. It is also possible to select between different numerical integration algorithms in each simulation level. For special simulation problems an extensive macro archive (macro library) is available.

The **Monte-Carlo study** of a damped second order mass-spring system achieved a very good speed performance by FSIMUL-P parallel simulation, because the communication overhead is very small. In FSIMUL-P only a static load balancing is implemented. A performance-monitor gives the user a good overview of the load balancing in the parallel system (CPU-times, communication, etc.). Table 1 provides an overview of the CPU-times for 100 sequential simulation runs in the time interval $t = [0;2]$; integration step size $h = 0.001$ sec; RK4-algorithm in each processor system.

processor system	simulation time
Master-PC (486 DX 50)	216.36 sec
Worker-PC (486 DX 50)	188.72 sec
Worker-PC (486 DX 66/2)	183.73 sec
HP840-Workstation (too slow)	256.51 sec
HP705-Workstation	163.09 sec
DSP TMS320C40	172.51 sec
T800 Transputer (too slow)	357.26 sec

Table 1

The CPU-speeds of all processors in the heterogeneous MP-system are very different, so that only the fastest processor systems are used for parallel simulation. The 100 simulation runs are equally distributed (25 each) on the DSP, HP705 and the two Worker-PCs. The Master-PC is only used for communication management, visualization of the average responses and performance-monitoring. In this configuration, 100 simulation runs are executed in 49.97 sec. Using 216.36 sec (see Table 1) as reference-value to calculate a speed-up factor, the resulting factor is $f = 4.32$. With respect to the necessary synchronisation of all processes, in FSIMUL-P, the T800-transputers would slow down the speed in parallel simulation.

For the **coupled predator-prey population** model the five populations are distributed on 3 or 5 processor-systems, so that every processor calculates only one or two populations. The Master-PC calculates the population v besides the simulation management and visualization of all population results because the population v has connection to all other populations. In the first configuration, the two Worker PCs, the DSP and the Workstation simulate particularly one of the other populations w, x and y, z. In the second case, one Worker-PC simulates the populations w and x, while the second Worker-PC simulates the populations y and z.

In addition we will examine the influence of the communication overhead through the parallelization. Table 2 shows the speed-up factors depending on communication intervals and configuration models.

configuration model	h	2h	5h	10h	20h
5 processor system	0.19	0.38	0.83	1.32	1.86
3 processor system	0.14	0.27	0.62	1.06	1.65

Table 2

The second order **partial differential equation** of a swinging rope is solved by discretization in 800 finite element modules. In FSIMUL-P, it would be possible to create a macro of one discretized section of rope. However, in this case, one would receive a system of order 1600 and a very big simulation structure of about 2400 blocks. Therefore, to solve this problem of the swinging rope by ordinary differential equation (ODE), a special block operation is implemented in FSIMUL-P.

Thus, one defines only the length 'L' of the rope-section, the number 'N' of discretization steps and the wave-propagation-speed factor 'a' of the rope. The ODE block solves the calculation of the swinging rope. The inputs of these blocks are the amplitudes u(0) and u(N) on both sides of the discretized sections; the multiplexed outputs are u(1), u(N/2), u(N/4), u(3N/4) and u(N-1).

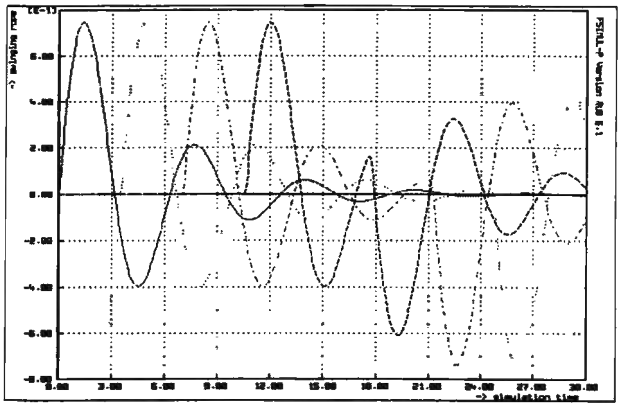

Figure 1: Parallel simulation of the swinging rope

Figure 1 shows the correct parallel simulation of the swinging rope problem (f = 3.85 with 4 Worker-PCs).

Figure 2 shows the effect that, by the parallelization of simulation models at the interconnection points, a loss of system dynamic is possible. This may happen if the communication interval is much bigger than the integration interval. The same probem occurs if an integration algorithm is used that needs to calculate intermediate values within the integration interval.

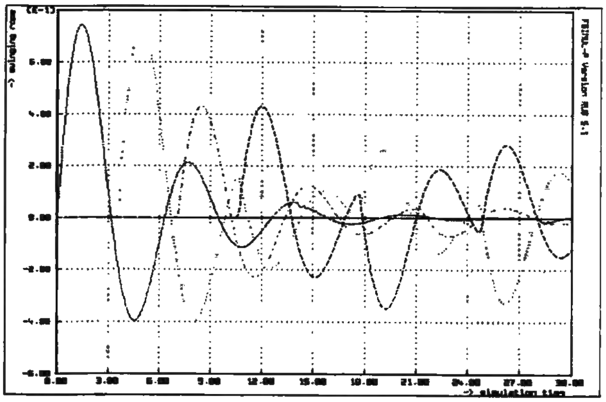

Figure 2: Parallel simulation with two processors, loss of system dynamic at interconnection point.

Peter Dellwig, Lehrstuhl für Regelungssysteme und Steuerungstechnik (Prof. Dr. K.-H. Fasol), Ruhr-Universität Bochum, D-44780 Bochum, Germany, Email: Peter.Dellwig@ruba.rz.ruhr-uni-bochum.dbp.de.