MATCHED ARCHITECTURES FOR SIGNAL PROCESSING AND CONTROL

J. A. VAN WOERDEN, R. ZEELEN, C. VAN DEN BERG, A. W. BENSCHOP

TNO-TU Institute of Applied Physics
Postbus 155
NL-2600 AD Delft
The Netherlands

Received: May 13, 1992

Abstract

Fast processing environments for real-time data acquisition, data processing and control applications may be realised using very different architectures. State of the art systems generally employ multiprocessors and parallel processing having a dedicated architecture such as systolic arrays to support computation-intensive signal processing tasks such as, for instance, convolution, filtering, FFT, etc. Mostly, general purpose rather than application driven architectures are used whenever possible and the available literature is heavily concentrated on the first configuration.

At TPD-TNO, the research emphasis is on application driven architectures, and the objectives for the so-called 'matched' architecture designs are:

- Capability for a wide range of sizes, starting from small systems. The objective here is design for scalability
- Design for systems to be used in harsh environments
- Design for minimum connectivity, reduced communication bandwidth, incorporation of dedicated preprocessing, multibus systems, etc.

The real-time behaviour of general purpose architectures is not sufficiently predictable and they are not designed to perform acquisition tasks or data-intensive processing with high performance. Matched architectures, on the contrary, are designed for well defined applications and optimized for each application.

The key effort in matched architecture research is directed towards efficiently mapping algorithms to processing steps in hardware (and software) architectures. Essentially, the design process is iterative.

Keywords: adaptive signal processing, matched architecture.

Introduction

'Matched architectures' in itself is a meaningless phrase. Any architecture is matched to some extent. In this paper, the application of an approach to signal processing and control for real-time systems of any size is addressed and 'matched' refers to combined hardware and software architecture and algorithms.

The basic target system functionality is in data acquisition and signal processing with either classification or control as an output.

The approach will be illustrated with examples. All those are concentrated on signal processing and control using physical sensors or measuring techniques. The target systems may have very different computing power requirements and the dedicated design for functionality should address this aspect.

In the matched architecture approach, design tools are used to specify the system architecture and to predict its performance (CARRIERO and GELERNTER, 1989).

Rationale

The key effort in the 'matched architecture' approach is mapping algorithms to architectures, in hardware and software, for systems with multichannel data acquisition for and with so-called real-time performance.

Application driven architectures are the opposite of general purpose architectures. General purpose architectures are, of course, flexible. They may be employed in a variety of applications and are optimized for flexibility and programmability, but not for performance.

Application driven architectures, on the other hand, are only flexible for one field of application and are mainly optimized for processing power, data communication, scalability, cost or size (KENT, 1987).

Very large scale integration (VLSI) techniques with their inherent constraint, can easier be implemented in application driven environments than in general purpose applications, while providing a performance bonus.

Processing-intensive tasks operating on large data sets for products and applications in difficult environments are potential candidates for VLSI with silicon compiler design. A feature of this approach is that the 'connectivity' problem is solved in a structured manner.

The 'matched architecture' approach is concentrated on employing existing components in an existing or extended architecture, realising improved performance by optimizing both hardware and software.

The connectivity problem for large data sets is addressed by the following design methods:

- bandwidth reduction in communication by separation of data and control bus,
- employing more than 2 buses,
- control circulation between processors rather than data circulation,
- parallelisation in design with optimal interprocessor communication.

The components used most frequently are DSP's (Digital Signal Processors), because of the internal architecture, but gate-array/microcontroller combinations are dealt with as well.

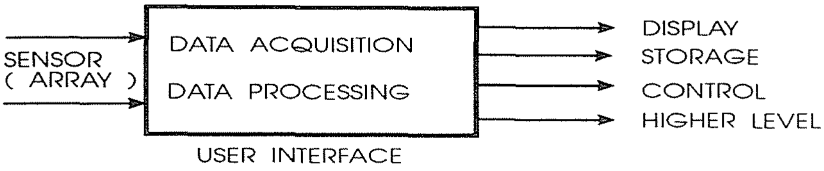

Fig. 1. Basic acquisition model

Basic Architecture

The general acquisition model is shown in *Fig. 1.*

The computer architecture underlying this general model depends on the tasks to be performed. Either a single processor or multi-processors in various configurations, with different degrees of parallelism might be used in this function.

The design constraints are:

- the speed of the data acquisition;
- the amount of data to be acquired;
- the real time performance required;
- the communication with output devices.

Algorithms

Development of a system with a matched architecture involves both the matching of the hardware architecture to the specified computation requirement by choosing appropriate algorithms as well as matching algorithms, or the way of implementing them to a specific hardware architecture.

The 'art' of identification, inspection and control requires mastering physical techniques. In applications for agriculture and horticulture, the identification of characteristic features of plant cuttings requires algorithms for filtering, lighting and classification.

For optimum image acquisition, selecting the best spectral sensitivity and, sometimes, the use of non-optical techniques are found to be of vital importance. 'Ultrasonic vision' is a technique to be used if sound-waves can penetrate where light can not. Specific algorithms and data-processing are needed in an acoustic camera e.g., under development, for instance, to 'see' in cloudy water.

The problem facing control engineering is that observations are often of a limited nature and do not supply the optimum parameters required for the control algorithm. System identification and modelling are therefore essential, and knowledge engineering can provide good back-up. This knowledge engineering was needed for determining the friction and backlash in a robot joint to optimize the controller. A comparable application is found in actively controlling a car suspension by means of adjustable shocks.

The design of general purpose algorithms for data acquisition is well known (anti-aliasing, oversampling, filtering), but specific filters, time and frequency domain algorithms, detection technique algorithms, etc., have to be designed before a computer architecture can be 'matched' to it.

Algorithm design is a discipline driven activity in the first place. Algorithms to be executed on parallel systems must also be designed that way. If algorithms require a specified performance, an iterative process of algorithm design and porting the algorithms to a system has to take place, since system size, data speed and issues such as time processing in the frequency domain are involved.

System Design Method

The method for system design is hierarchical.

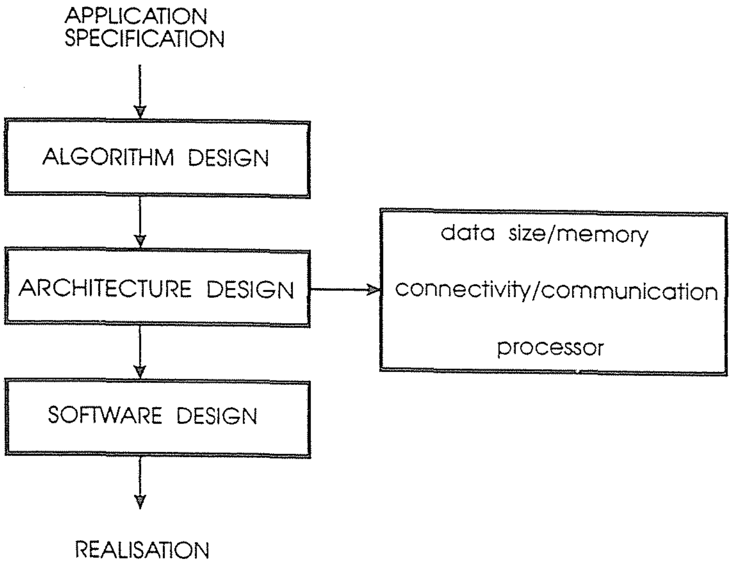

Fig. 2. System design method

The algorithms are designed first and analyzed for regular structures which may be executed in parallel. In the architecture design choices for memory organisation connectivity and computer power (processor) are made. Software design comes last.

Acquisition Architectures

The basic architecture is given in *Fig. 3.* This is a so-called general purpose architecture. A specific computer environment is used. Application driven architectures are relevant for real-time systems, scalable systems, etc.

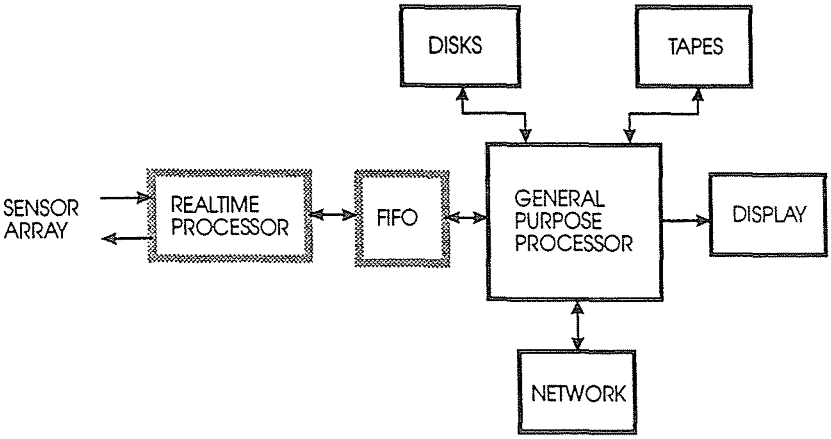

Fig. 3. Dedicated preprocessing acquisition model

The first step in developing application driven architectures is adding an extra processing device for fast preprocessing.

The real-time processor contains no operating system, has a dedicated acquisition and data reduction task and is connected to a general purpose environment via a loose coupling (FIFO). The general purpose environment has an operating system for system tasks and is either event driven (real-time) or time shared (not real-time). The same architecture can be realised with far superior performance if the real-time processor is an application accelerator (100 MIPS or 200 MFLOPS) and the FIFO is replaced by a local memory of 16 Mbyte.

A larger number of channels can be handled by acquisition units with more than one bus, with fast memory on the buses and data and control separation (Harvard structure). Either a dedicated (DSP) or a general purpose end computer can be used. Memory organisation and connectivity is here of even higher importance than processing power (BOWEN, 1992).

A special form of this architecture is a DSP or microcontroller based configuration with a programmable (fast) extended function. Flexibility is

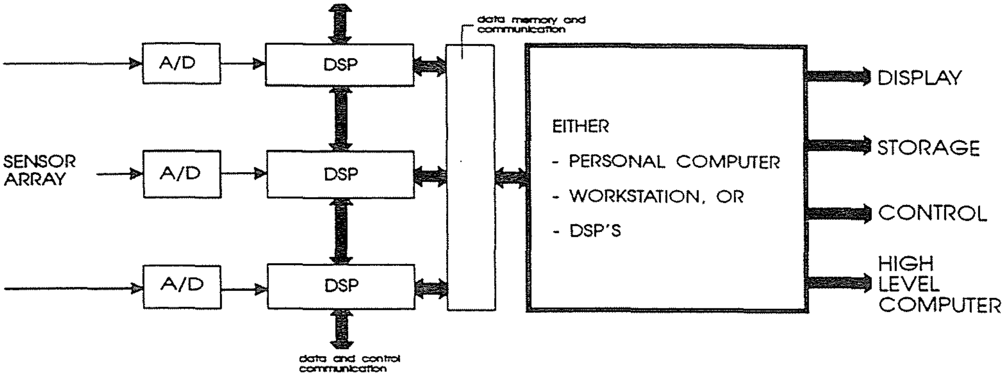

Fig. 4. Parallel acquisition model

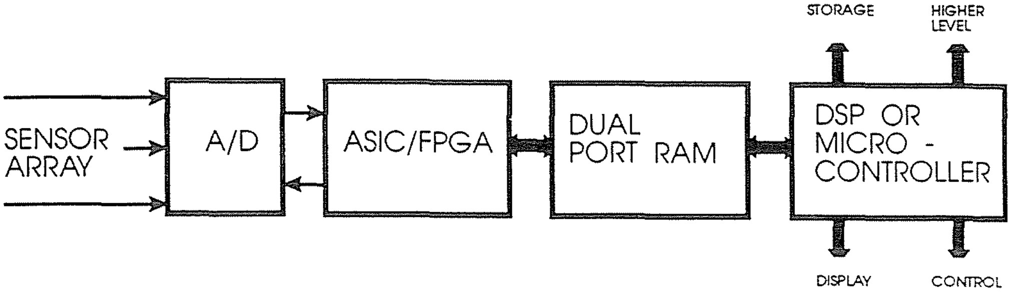

Fig. 5. 'Simple' architecture acquisition model

provided by the DSP (microcontroller) and computing speed by the ASIC (*Fig. 5*).

Reactive Systems

Embedded, targeted, concurrent, distributed real time systems are often referred to as 'reactive' systems (HAREL, 1988). They are not very easy to program, especially since the real-time feature is difficult to define in specific terms. Modelling techniques are in fact mandatory as well as techniques for emulating and inspecting a model generated.

Modelling a system starting from the application is described in layers. System models consist of an interprocessor communication layer, a system software layer, and an application layer, on top of which a diagnostic model is designed most of the time. The specification of the software environment is part of this approach. Parallel systems are both heterogeneous and distributed (BOWEN, 1992).

Tools

Tools such as SCILL-IMAGE (software for image processing) are used for algorithm design for image processing. Matlab/SIMULINK is used for control design and simulation. The Comdisco workbench for signal processing function design is used both for system segmentation, modelling simulation and algorithm design.

As indicated algorithms are dedicated for applications and disciplines; they are designed iteratively with the system architecture.

Proven design tools such as Yourdon analysis and computer design tools are used for integrated systems. Performance measurement is carried out with simulators and special visualisation tools for the layers to be able to debug at appropriate levels.

Example 1: An Active Noise Control System for Reduction of Stochastic Noise

Introduction

The principle of Active Noise Control (ANC) has been known for many years. It is based on the phenomenon that a signal can be cancelled by superposition of a signal with the same amplitude but with opposite phase. This may suggest that Active Noise Control is a rather easy task. However, even in the case of simple systems (for instance ventilating ducts) things become much more complicated. For example, the amplitude and, even more important, the phase of the signal to be cancelled is generally unknown. The same holds for the acoustic transfer-functions, which may considerably vary over time due to temperature changes.

To apply Active Noise Control theory to real life applications, a fast control unit able to perform a variety of digital signal processing functions — such as adaptive filtering, FFT, transfer-function estimation — is needed.

Active Noise Control Principle

In *Fig. 6*, a typical set-up of a stochastic anti-noise system is depicted. A Detector Microphone, placed upstream of the anti-noise source, samples the sound field generated by the noise source, (primary sound field). A second microphone — the Error Microphone — placed downstream of the anti-

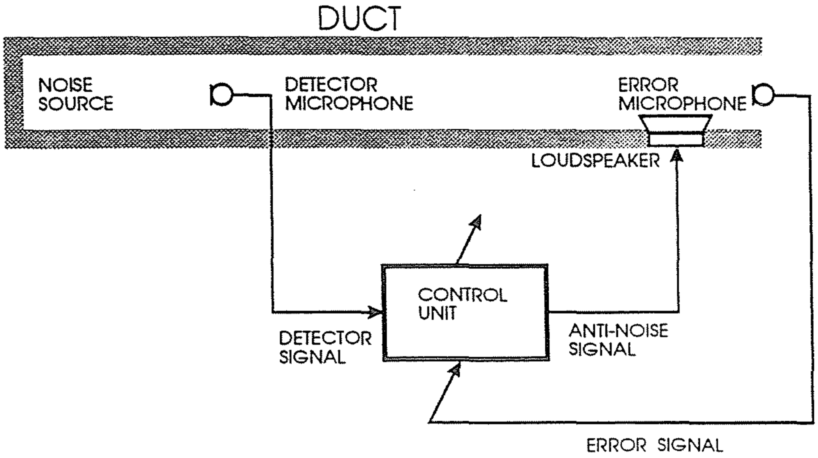

Fig. 6. Active noise control in a duct

noise source samples the so-called residual sound field. Both microphones are connected to a Control Unit.

The Control Unit uses the Detector Signal to generate the Anti-Noise Signal by applying a digital filter. The Anti-Noise Signal is used to drive the loudspeaker which in its turn generates a secondary sound field (see *Fig. 6*).

The main task of the Control Unit is to determine and adapt the coefficients of the digital filter to reduce the residual sound level. This is done by minimizing a cost-function based on Error Signal data. To calculate this cost-function, the Control Unit estimates transfer-functions which provide the unit with information about the acoustic environment to be controlled.

Since, in practice, the transfer-functions are not constant due to changes in the acoustical environment (e.g. changes in temperature, airflow, etc.) the Control Unit has to (re)estimate these functions at regular time-intervals.

Note that due to the use of transfer-function estimation techniques, it is not necessary to program application dependent information in advance (self-tuning regulator concept). This allows the same type of Control Unit to be used for different applications.

Stochastic Anti Noise Control — in contrast to periodic noise cancellation — imposes strict requirements on the calculation speed of the Control Unit, as it is not sufficient to control a limited number of harmonic frequencies only. Additionally due to the absence of periodicity in the Detector Signal, a considerable part of the processing has to be carried out within one sample period (typically less than 0.5 ms).

Functional Description

The Control Unit has to perform 4 main tasks:

1. Input Data Acquisition and Analog Pre-Processing
2. Active Noise Control-Algorithm Execution
3. Output Data Generation and Analog Post-Processing
4. User Communication

The *Input Data Acquisition and Pre-Processing task* is charged with the processing of the Detector and Error Signals. To allow for flexible anti-aliasing filtering, the data-acquisition is based on oversampling. The Input Data Acquisition and Pre-Processing task includes analog signal processing (e.g. signal buffering and anti-aliasing filtering). A/D conversion and digital signal processing (e.g. buffering of oversampled data, anti-aliasing filtering, re-sampling and buffering of down-sampled data).

Due to the use of oversampling, two time scales exist within this task: the oversample period time (50 μs → 20 kHz) and the sample period time (0.4 ms→ 2.5 kHz). Almost all (sub)-tasks are executed within one oversample period. The digital anti-aliasing filtering and the down-sampling, however, are performed within one sample period.

The *Active Noise Control-algorithm* is based on a three layer control-concept:

1. Controller layer,
2. Controller Design layer and
3. System Identification layer.

The Controller layer represents the lower control level, it applies an ARMA-filter to the Detector Signal data. The coefficients of this filter (2048 maximum) are determined by the Controller Design layer (medium control level). This layer is able to adapt to small changes in the primary sound field by using a LMS-type algorithm. In turn, this layer uses information processed by the System Identification layer (high control level). This layer determines the acoustic environment and adapts to small changes in it.

The Controller layer operates on a sample-to-sample basis in the time domain. It processes real, single precision floating point values. The other two layers operate in the frequency domain and process complex valued, single precision floating point values. The processing in these layers includes: Time Domain Windowing, Fast Fourier Transformation, and Power Spectrum Calculation of Detector and Error Signal data, Frequency Domain ARMA-filter Coefficient Update (LMS-algorithm) and Inverse Fast Fourier Transformation. Since the changes which should be tracked by those two upper layers are slow when compared to the sample frequency of the Control Unit, there is no need to complete processing within one sample period.

Control Unit

The matched architecture approach is described below.

To determine the processing power required to implement the tasks, simulations of some key (sub)-tasks were carried out. It turned out that it was not possible to use a single Digital Signal Processor (DSP).

Based on the different time scales of the tasks, a loosely coupled dual-processor architecture was proposed. One processor (DSP1) is charged with the tasks or sub-tasks which are not executed on a sample-to-sample basis: medium and high level Active Noise Control-Algorithm Execution and User Communication.

A second DSP (DSP2) takes care of the Input Data Acquisition and Analog Pre-Processing task, the low level Active Noise Control-Algorithm Execution task and the Output Data Generation and Analog Post-Processing task.

By distributing the (sub)-tasks over both processors in this way, only a limited bandwidth is required for processor-to-processor communication (transfer of Detector and Error Signal data and ARMA-filter coefficients).

Hardware Architecture:A two DSP based Control Unit hardware--architecture is used. Each DSP (Texas Instruments TMS320C30) has its own local program and data memory connected to a primary bus (DSP1 and DSP2 Local Memory). Both DSP's share a Dual-Port Memory which is used for processor-to-processor communication. Each DSP is connected to this memory by a second DSP bus (the expansion bus).

Further, DSP1 is connected to the Local User-Interface and the Remote User Interface. The former contains the interfaces for front-panel switches and indicators. The latter contains a Universal Asynchronous Receiver and Transmitter (UART) and RS-232C drivers. Both interface blocks are connected to DSP1 by the primary bus. The same is true for the DSP1-Control block. This block represents (local) control functions such as address-decoding and reset and ready signal generation.

The primary bus of DSP2 is connected to an Analog Pre-Processing block and an Analog Post-processing block. These blocks contain A/D and D/A converters, fixed analog low-pass filters and amplifiers. The DSP2-Control block is comparable to the one described for DSP1.

Software Implementation Aspects: Nearly all software functions have been written in the high-level language 'C'. This holds even for interrupt-functions and very time-critical functions performed by DSP2. Of course, this is only feasible if the C-compiler generates efficient and fast code. In case the 'C-function' turned out to be too slow, it was used as a prototype function to derive a fast Assembler routine by manual optimisation. This optimisation largely meant the introduction of parallel instructions. The same strategy was employed for special addressing modes supported by the DSP (e.g. circular addressing). An Assembler-coded function library was used to implement the FFT and Inverse FFT.

The total program size for DSP1 is ≈ 13 kquads (32 bit words) while the program of DSP2 requires less than 13 kquads.

The DSP software is described in more detail elsewhere (ZEELEN, 1991).

Example 2: Overhead Wire Inspection: an Opto-electronic System Facilitates Integral and Fast Inspection

A combination of geometric-optical design and advanced CCD detector arrays with fast electronics forms the base of an inspection system called ATON (Automatic Thickness measurement Overhead wires Netherlands railways).

This advanced system inspects overhead wires in the Netherlands (VAN GIGCH, 1991).

Specifications

The system requirements were very stringent. For instance, the measuring train should be able to perform inspections at 90 kilometres per hour to avoid disruptions of the normal timetable. Even at that speed, the wire has to be evaluated every centimetre, necessary for detecting so-called 'craters' in the wire caused by sparks; the diameter of these craters is around 1 centimetre.

The remaining thickness of the wire has to be determined to an accuracy of 0.25 millimetres, down to a minimum value of 7.5 millimetres. New wires are circular with a diameter of 12 millimetres; a wire that has been worn down to 7.5 millimetres should be replaced.

The position of the overhead wires with respect to the measuring train has to be measured and also the distance between the wires, which are always twinned.

Some practical specifications were:

- the field of view perpendicular to the direction of travel is 110 centimetres;
- there can be at most four wires within the field of view;
- the height of the wires above the track varies between 4.5 and 5.5 metres.

Opto-electronic method

From various possibilities an opto-electronic method was selected.

The overhead wires are illuminated from the carriage with a laser beam and the reflected light is focused on a detector array by an optical system. An important factor for the illumination optics were the scattering characteristics of the worn underside of the wires. In the direction of travel the reflection is almost purely specular, while in the perpendicular direction the light reflected was scattered over an angle of about ∩5 degrees. Therefore, cylindrical optics are used to ensure that, independently of the angular position of the wire, a major part of the reflected light will reach the detector.

For imaging the wire on the detector array a telescopic configuration is used. With this configuration, the magnification is independent of the distance of the wires. Variation of the height of the wires makes active focusing mandatory. A linear CCD (Reticon, 2048 elements) is used as detector.

A part of the optical system has been mounted on a positioning system which keeps the system and the laser focused on the underside of the wire.

A major effort was required to develop the electronic hard and software. At the maximum speed of 90 kilometres per hour a complete measurement has to be made every 0.4 milliseconds to meet the requirement of one measuring point per centimetre. For each measurement five CCD's of 2,048 pixels each have to be read out, and 2.500 measurements per second correspond with a data flow of 250 Mbits per second, to be processed in real time.

Digital Signal Processors

The output of the CCD's is stored temporarily in a buffer memory, large enough to hold data from five sequential (centimetre) measurements. For the processing, five DSP's (Digital Signal Processors) and five microprocessors are used. The data from each measurement are first read into the first DSP which determines the window in which the desired signal occurs. All other data are then discarded, leading to a reduction of the data flow to 12.5 Mbits/second (a factor of 20). These remaining data are processed by four other DSP's which extract the signal — one for each of the four wires that may be in view. These processors calculate the remaining thickness of the wire or establish the presence of craters. The development of the very fast algorithms required was one of the most difficult parts of the project.

Fig. 7 is an overview of the system.

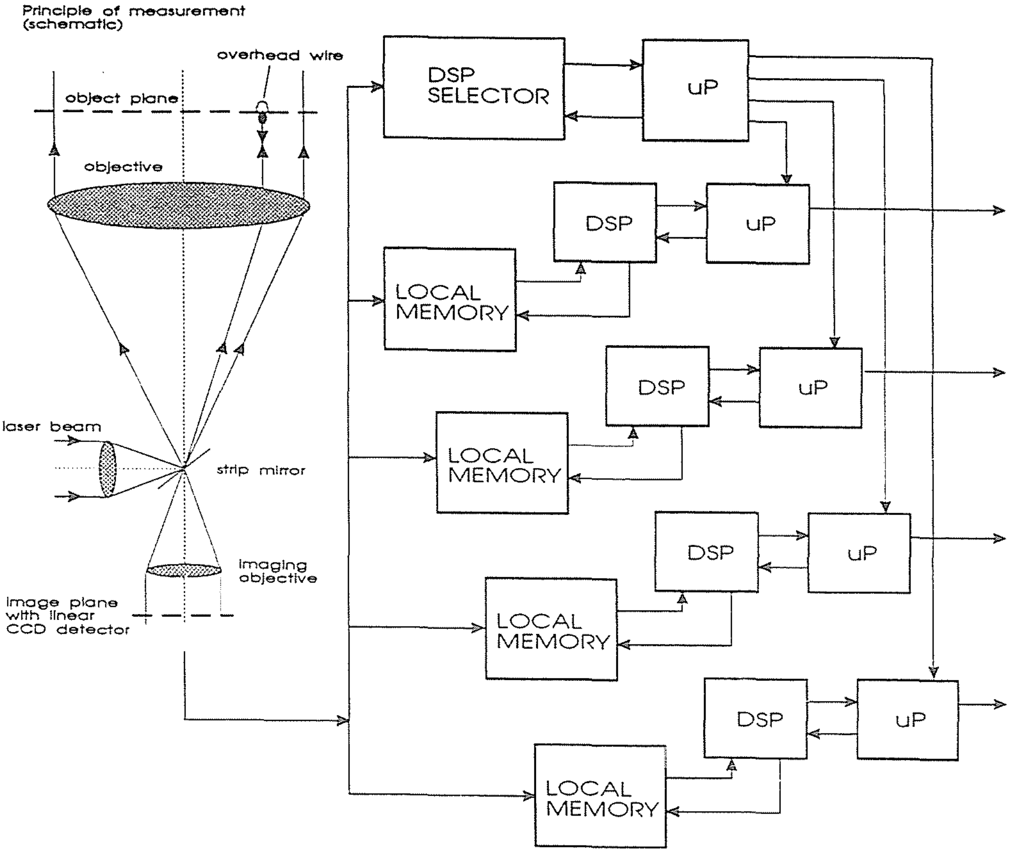

Fig. 7. Overview of the ATON measuring system

The DSP's perform feature extraction for each wire. Algorithms for threshold detection, averaging and reflection width measurement are implemented.

The algorithm is implemented in the DSP selector and parallel in the four DSP's. All algorithms were designed to be parallelized. The 'matched architecture' approach is realised in a physical separation of units (DSP's) running algorithms in parallel for feature extraction.

References

BOWEN, N. S. (1992): On the Assignment Problem of Arbitrary Process Systems to Heterogeneous Distributed Computer Systems. *IEEE Transactions on Computers,* Vol. 41, No. 3, pp. 257–273.

CARRIERO, N. – GELERNTER, D. (1989): LINDA in Context. *Comm. ACM*, Vol. 32, No 4, April 1989, pp. 444–458.

HAREL, D. E. A. et al. (1990): Satemate: 'A Working Environment for the Development of Complex Reactive Systems'. *IEEE Transactions on Software Engineering,* Vol. 16, No. 4, April 1990, pp. 403–414.

KENT, P. (1987): Application Dictates your Choice of a Multiprocessor Model. EDN, June 25, 1987, pp. 241–248.

ZEELEN, R. (1991): A Real Time DSP Based Active Noise Control System for Reduction of Stochastic Noise. *International Conference on DSP Applications and Technology,* Berlin, 28–31 October, 1991. pp. 640–648.

VAN GIGCH, J. M. (1989): ATON Measures Overhead Wires. Toegepaste Wetenschappen, (*TNO Magazine*) Vol. 5, No. 6, June pp 39–41. (In Dutch)

VAN GIGCH, J. M. – SMORENBURG, C. – BENSHOP, A. W.(1991): The Contact Wire Thickness Measuring System (ATON) of the Netherlands Railways *Rail International* Vol. 22, No. 4, April, pp. 20–31.