Fault-Tolerant Parallel Implementation of Decentralized Control for a Segmented Telescope Test-bed

S. FALLORINA, H. BOUSSALIS, K. RAD, C. LIU, J. DONG, A. KHOSHAFiAN
California State University, Los Angeles, CA 90032, USA

Abstract: - Control of large flexible structures such as segmented telescopes requires a reliable computing system that continues to operate satisfactorily despite the event of hardware or software failures. Through decentralized control, the overall structure can be controlled by multiple lower-order local controllers distributed on a multiprocessor system. The breakdown of the application into smaller individual computing agents facilitates a fault-tolerant control system that recovers from processor failures. This paper describes a pipelined parallel computing technique that enables the adaptive and dynamic task mapping for fault-tolerance. It also illustrates the real-time embedded system that provides architectural support to such dynamic task mapping.

Keywords: - Fault-Tolerance, Parallel Processing, Pipelined Task Mapping, Dynamic Task Mapping, Real-time Embedded Systems, Decentralized Control

1 Introduction

In order to detect faint light signals from space, the Next Generation Space Telescope (NGST) requires a larger light-gathering mirror [1]. But due to the manufacturing and space deployment difficulties of using a monolithic piece of glass, the primary mirror will be built up from smaller reflecting panels. However, a reflector built from segments relies on an active control system for precision alignment of the optical surface in a dynamic disturbance environment.

The operational complexity involved in such remote unmanned systems necessitates control and computing designs that can recover from system component failures. Faults may come from different constituent parts like sensors, actuators, and processors. The effects may range from a drop in performance to a full system failure. The objective of a fault-tolerant control system is minimal and graceful performance degradation. Dependable system integrity may be achieved through both fault-tolerant control and computing methods.

To study the control of complex dynamic behavior of such large segmented optical systems, NASA has funded a project to design and construct a test-bed in the Structures Pointing and Control Engineering (SPACE) Laboratory at the California State University, Los Angeles (CSULA). Of particular interest, is the application of decentralized control techniques. Decentralized control reduces the computational complexity involved with the control implementation of such large system. In addition, by exploiting the parallel nature of decentralization, failure analysis and the development of parallel programs can become simpler tasks.

Figure 1 shows the major features of the SPACE test-bed, which is designed to emulate a Cassegrain telescope of 2.4-meter focal length with performance comparable to an actual space-borne system [4].

The segmented primary mirror, supported by a lightweight truss, is composed of a ring of six actively controlled hexagonal panels arranged around a fixed central panel (Figure 2). To allow primary mirror shape control, the primary is fitted with an ensemble of 42 inductive sensors that provide measurements of relative panel displacement and angle. Each panel is mounted on three voice-coil linear actuators which provide three degrees of freedom for each segment. Decentralized control algorithms are implemented using commercial off-the-shelf (COTS) computer system.

---

1 This work was supported by NASA under Grant: URC NCC 4158.
Fault-Tolerant Decentralized Control

Fault tolerant systems employ a variety of technologies that improve system reliability. From a control design perspective, active reconfiguration of the control law on-line is achievable through Fault Detection and Isolation (FDI) and Reconfigurable Control (RC) schemes [10]. Another approach which is detailed here is decentralized control. The strategy of decentralized control is to control the entire system by controlling its subsystems. This involves decomposition techniques that result in physical or mathematical decentralization of the structure into lower-order subsystems. The properties of the interconnecting patterns can be further utilized to derive control laws for shape control of the primary mirror and vibration suppression of the overall structure.

This decentralized approach has many benefits. One clear advantage is the substantial reduction of computational load for an actual implementation compared to the centralized counterpart [5]. Meanwhile, the natural parallel structure of the set of subsystems makes it conducive for parallel processing. Each controller may perform operations upon a partitioned domain of the input signals. Such tasks can be mapped to a number of interconnected processors to achieve parallel processing. Furthermore, decentralization of the controller is potentially more robust, both with respect to model uncertainty and component failure. Should one or more subsystem failures, other local controllers may be able to compensate. Decentralized control provides a convenient framework to establish fault-tolerant control systems.

These advantages will be highlighted in the following decentralized control implementation for the segmented telescope test-bed. The discrete state equation (1) represents an $n^{th}$-order system with $m$ inputs and $r$ outputs, where $\Phi$ is the state transition matrix, $x(k)$ is the state vector, $u(k)$ is the input vector, and $y(k)$ is the output vector.

$$x_{n1}(k+1) = \Phi_{n1} x_{n1}(k) + \Psi_{n1} u_{n1}(k)$$

$$y_{n1}(k) = C_{n1} x_{n1}(k)$$

As illustrated in Figure 3, one decentralized control approach addresses the primary mirror shape control problem by first decomposing the model into six decoupled subsystems each representing the dynamics of one panel (i.e. subsystem 1 consists of panel 1, subsystem 2 consists of panel 2, and so on).
established methods such as $H_\infty$, adaptive, and neural network control [2, 3, 9]. In one implementation, a single 200th-order centralized controller that controls the entire primary mirror is replaced by six 12th-order local controllers, running simultaneously to maintain the precision primary mirror shape. This method reduces the required computational load and facilitates both parallel implementations and fault-tolerant systems studies. While failure of one or more controllers, with such a decoupled configuration, may render the corresponding controlled panels uncontrollable, the other panels may be unaffected. This reduced functionality can be avoided by a pipelined parallel computing technique described in this paper.

Another method decomposes the same primary mirror model into six subsystems, with each representing the dynamics of two overlapping panels (i.e. subsystem 1 consists of panels 1 and 2, subsystem 2 consists of panels 2 and 3, and so on). This approach includes more interactions between the segments resulting in better control performance, at the expense of increasing the controller order. The overlapping decentralized control implemented involves 14 to 18th-order controllers. However with such a configuration, failure of one or more controllers may be accommodated by other functioning controllers due to the overlapping control responsibility.

3 Fault-Tolerant Parallel Application Design

The decentralized control algorithms proposed can be easily parallelized. In this section, two different task mapping techniques are introduced. Between them, the pipelined task mapping described in Section 3.2 supports fault tolerance. I.e., if any of the processors fails to function, a re-mapping of tasks can be performed dynamically so all controllers can still work on the remaining processors that function normally.

3.1 Control Process Description

The single processor program of the primary mirror shape controller is presented on Figure 4. Each control process cycle starts by reading in 18 edge sensor signals which indicate the displacement and positions of the panels. These are used by the six controllers to calculate 18 control commands which are then converted into analog signals and sent to the actuators at the end of the cycle.

All these tasks need to take place within a specified sampling period. Through parallelization, the six decentralized controller tasks can be conveniently assigned to available processors as in Figure 5, thus increasing efficiency and speed-up. However, different controllers share some of the input data so these shared data must somehow be distributed. In addition, one master processor is still required to read the A/Ds, distribute the data, collect calculated control commands, and output them to the D/As.

3.2 Fault-Tolerant Pipelined Parallel Implementation

For supporting fault-tolerance, another possible parallel design approach may be implemented. This approach involves pipelining all the separate controllers in sequential order on each of the available processors as illustrated in Figure 6.

This technique promises to tolerate failure of one or more processors, since anyone of the functioning nodes are able to control all the
subsystems in different iterations. In this design, each processor keeps a copy of all of the constant matrices in its local memory; i.e., $A$’s, $B$’s, and $C$’s for all of the six panels. This approach reduces the traffic on the bus due to the accesses of the shared memory space. On the other hand, the input vectors, $u$’s, output vectors, $y$’s, and the state vectors, $x$’s, are stored in the global memory. Thus, each processor can access these global variables in a round-robin fashion. Note that, in such a real-time embedded system, the output vector of a specific decentralized controller is used to trigger the actuator to move the corresponding panel. Then, the sensor receives the signals from the test-bed and transforms the signals to the digitized input vector. The input vector is, then, used for the operation of the next iteration of the control. Thus, there is an automatic serialization between the accesses of the global vectors; no racing problem can occur. Note also that, message passing is avoided in this approach. Otherwise, re-routing of the messages can incur significant difficulty for the implementation as a processor crashes. Furthermore, if controllers are paired together as in Figure 6-b, then speed-up is achievable at the same time as fault tolerance.

This approach addresses the limitations of traditional parallel implementation of decentralized control using the decoupled subsystems. The reliability is achieved by the replication of tasks across multiple processors. However, the added redundancy lessens the opportunity for speed-up and better efficiency.

### 3.3 Computer Architecture

The fault-tolerant parallel implementation of the decentralized shape control of the primary mirror is to be realized on a distributed memory architecture detailed as follows. To implement the described control system, software drivers have to be significantly modified in order to work with the following system capabilities and limitations.

![Fig. 7- Computer Architecture](image)

The real-time embedded system, shown on Figure 7, utilizes an off-the-shelf Pentek 4285 board that is configured with four TMS320C40 chips [8]. Each processor has its own local memory in addition to a globally shared memory space. High-speed bi-directional communication ports allow direct message passing schemes between processors while the shared global random access memory (RAM) is accessed through a bus. The digital-to-analog (D/A) and analog-to-digital (A/D) converters are connected to the sensors and actuator amplifiers respectively. The control of primary mirror requires 18 sensor inputs and 18 actuator outputs. This multiple input-multiple output (MIMO) system...
attests to the computational requirements of this application.

The individual processing units utilized are 32-bit floating point TMS320C40 digital signal processors (DSP) that incorporate on-chip parallel processing features to obtain high individual performance as well as making it conducive to multiple processor configurations. In particular, the high-speed bi-directional communication ports provide zero glue logic, direct communication between processors. The C40s can also receive and send interrupts from the VMEbus and other processors.

The VMEbus has seven available IRQ lines (IRQ1 – IRQ7), with IRQ1 having the highest priority. Each of the 6102 A/D boards can send one interrupt to the DSP through the VMEbus at a specified IRQ line. Upon receiving interrupts, the data is read through the main bus and VMEbus by the master processor. However, when data is retrieved, that data is removed from memory; therefore, shared data must be passed or stored in shared memory. Furthermore, if multiple processors need to know that data has arrived, then multiple IRQ signals must be used since a single IRQ cannot be broadcasted.

Note that the critical task of data I/O and distribution is handled only by a single “master” processor. Should this processor stop working, its tasks must be transferred to a functioning processor. This dynamic task mapping is achieved through establishing a protocol that detects failure and reassigns of the role of master. When the data arrives at the start of each control cycle, all processors are interrupted so all are capable of responding. However, only the processor assigned as the master actually reads the data. At the end of each cycle, each processor reports its working status through shared memory. If the master processor fails to report, then another functioning processor from a pre-specified list takes up the master role.

4 Conclusion & Future Work
In this paper, two task mapping techniques have been described to support parallel computing of the decentralized control algorithms to meet real-time requirements. The next step is to implement such parallel programs that would still function in the event of failures. By capitalizing on the natural parallel structure of decentralized control, future experiments of the fault-tolerant parallel processing design will be conducted, wherein the system can recover from one or more processor failures and/or communication link breakdowns. Fault tolerance is achieved through both decentralized control and pipelined parallel computing methods.

5 Acknowledgments
This work was supported by NASA under Grant URC NCC 4158. Special thanks go to the SPACE lab team for their collaboration with this research.

References: