Comparing partly and fully distributed control systems in a hydraulic mobile crane

Pasi Tuominen    Tapio Virvalo
Inst. of Hydraulics & Automation
Tampere University of Technology
P.O.Box 589, 33101 Tampere, FINLAND

ABSTRACT

Distributed control system offer many advantages compared to centralized system. There is at least two ways to distribute systems; partly and fully. With fully distribution there is some more benefits compared to partly one, but there are some extra demands for the used network protocol and the implementation requires good planning.

KEYWORDS

Fully distributed control, Distributed control, Independent Actuator, Crane

INTRODUCTION

Development of electronic and network systems has made possible to integrate more "intelligent" in mechanical component. Independent actuators (IA) and microprocessor controlled servosystems are introduced very widely by many manufacturers /1/,/2,/3/.

Figure 1. The independent actuator includes all the hardware and software needed for motion control of single axis.

DISTRIBUTED MOTION CONTROL

Motion control using distributed control system has not been so widely presented especially in the area of machine automation. There are many reasons for that. Motion control needs experience from many areas. Synchronization and real time demands are problems that are difficult to solve. Those problems are mainly due to network. Protocol used in distributed control system should have deterministic behavior. Also protocol should be efficient enough thinking about real time operation, most of all event handling capability should be good. Master - slave protocols and object oriented protocols are mainly used. Token - bus protocol has too big event handling delay /4/,/5/.
Master-Slave Structure (MS)
Master-slave structure in distributed control system is mostly used. Master-slave structure is easy to form and also master-slave protocol has been widely available by many companies \cite{6,7,8}. In hierarchical system it is usually uncontrollable situation if master will be broken. So, master-based control systems depend strongly on one node. Thinking about security this can be an additional risk. Needed number of transmission for updating all nodes is in host based system two times the number of nodes. The problem is that all processing that handle the control of whole machine have to be executed in one node, and that the same node have to control also the message sending in the system. So the clear bottleneck in the system is then the master. Parallel processing can be used only partly with host based architecture.

Fully Distributed System (FDS)
In FDS not only the network control but also the control of the whole machine is distributed. In the network there is not any master in the network that is controlling the system. All nodes are equal in pure fully distributed control system. Every node can send to another node. The expanding of the network is not so easy when FDS is used. Because every node is independent there is not so much need for efficient network. The need of network can be minimized with FDS. With FDS, the control of whole machine is distributed, and real parallel processing is possible.

POSSIBLE NETWORK PROTOCOLS IN HYDRAULIC MOBILE CRANE
In used hydraulic mobile crane is the characteristic frequency about 1 Hz. The control interval in single actuator should be in maximum 20 ms.

Figure 3. The control of machine takes time 100 ms, it can be interrupted by actuator controller in every 20 ms.

Mobile crane is usual mobile crane without any extra modification, except one - potentiometers has been installed inside the cylinders.

Figure 4. Used mobile crane. Cylinders includes potentiometers.
When distribution is used, is there need for network protocol. The network protocol should be deterministic and cheap. Also it should be available and efficient enough. Most of all event handling capability is important. Thinking about these requirements are sc. vehicle networks proved to be very suitable. CAN (Controlled Area Network) is mostly used. Also other that kind of protocols are available, like ABUS, VAN, J1850 etc., \(\ddagger\). In our crane CAN-based network is used.

**CAN (Controller Area Network)**

CAN is developed by Intel and Bosch. CAN use CSMA/CD + AMP protocol. It belongs to so called vehicle networks and in those the fastest possible class C (SAE - class definition) \(\ddagger\). CAN really is very fast; the protocol guarantee 125 \(\mu\)s latency time when transmission rate is 1 Mbit/s and 8 byte is transferred. Messages are objects that have different priorities. In simulated results CAN is proven to be fast enough.

**IMPLEMENTATION**

**Hardware solution**

In distributed systems the number of cables should be as small as possible. In this solutions is only two cables. One twisted pair cable for transmission and one three-wired cable for power needs. When hardware has been designed, the low-cost and integration qualities of electronics are taken account. INTEL’s 16 bits microcontroller 80C196 is quit a cheap, it has been chosen as main processor. Communication has been implemented with Philip’s CAN controller 82C200. It support basic-CAN protocol.

80C196

80C196 is 16-bit CMOS microcontroller. Main character of it are:

* 232 Byte Register File
* Register-to-register Architecture
* 28 Interrupt Sources/16 Vectors
* 2.3 \(\mu\)s 16 x 16 Multiply
* 4 \(\mu\)s 32/16 Divide
* 16-Bit Up/Down Counter with capture
* Pulse-Width-Modulated Output

* 10-Bit A/D Converter with S/H

In this implementation is 10-Bit A/D Converter used for position feedback. Filtered Pulse-Width-Modulation is used as a control output. This PWM drives commercial voltage-to-current transformation card, that drives valves, figure 5.

![Figure 5. Intels 80C196 measure the position with A/D converters and drives voltage-to-analog converter with filtered PWM output](image)

**Software solution**

There are many real time tasks that should be done simultaneously. These tasks are distributed different ways with different control architectures.

**Master-slave (MS)**

With MS architecture only the control of actuator is distributed, so needed software in slave is not so big as with fully distributed system. The control of machine and synchronisation is done by master. In our solution control-loop of crane is executed so fastly (10 times/s), that more synchronisation is not needed with master-based system. There are not any collisions between messages with this architecture. So MS-architecture is very deterministic. If crane should be connected to upper level is master natural connection point. With this kind of architecture should master have quit a big calculation capability.
Figure 6. In partial distributed control system only the control of actuators are distributed. System controller, master, send orders to actuators and also follow the function of whole machine.

Fully distributed system (FDS)
In this architecture all nodes are principally equal. In one node is executed simultaneously at least three tasks. With 80C196 is this implemented with internal timers, that can cause interrupt on wanted time.

There are more problems with FDS architecture than with master-based. One problem is the updating of informations. The other problem is synchronisation. Synchronization has to be done in machine control level. Used software can be dealt to software in joystick node and to software in actuator controller. The problem of updating has been dealt on two parts; the updating of control signal and the updating of positions signals. In control signal is command for next movement direction. It is one message, object, that has three data byte. One byte for one direction of movement. Signal-type is broadcast.

Figure 7. In one distributed node is many real time tasks that should execute simultaneously. Tasks have different priorities

Figure 8. Needed functions and calculation in joystick node

Figure 9. Updating of control signal in three dimensional case.

Another set of software is placed in nodes, that control actuators. The coordinate transformation is done there, and also rutin for
cylinder position transforming. In addition to those also the control of single actuator is executed in this node. The control of whole machine is distributed in three nodes. Needed functions and calculations are presented in figure 10.

**SOFTWARE IN NODE A**

<table>
<thead>
<tr>
<th>CYL_POS1, CYL_POS2, CYL_POS3</th>
<th>CYLINDER POSITIONS CYLINDER A, CYLINDER B CYLINDER C.</th>
</tr>
</thead>
<tbody>
<tr>
<td>θ₁, θ₂, θ₃</td>
<td>CALCULATION OF JOINT ANGLES</td>
</tr>
<tr>
<td>x, y, z</td>
<td>FORWARD KINEMATIC TRANSFORMATION</td>
</tr>
<tr>
<td>x + x₀, y + y₀, z + z₀</td>
<td>ADDITION OF CONTROL SIGNAL</td>
</tr>
<tr>
<td>CYL_POS1, CYL_POS2 CYL_POS3</td>
<td>INVERSE KINEMATIC TRANSFORMATION</td>
</tr>
<tr>
<td></td>
<td>NEW REFERENCE TO CYLINDER A</td>
</tr>
<tr>
<td></td>
<td>CONTROL OF CYLINDER A</td>
</tr>
</tbody>
</table>

Figure 10. The function that one node should do in fully distributed control system

Because the reliability is very important aspect in this case the software is located so, that single node is as intelligent as possible. System is controllable, even if one actuator node is destroyed.

Communication is more complex with FDS than with MS. CAN protocol support broadcast type transmission. So transmission delays does not depends only on used baudrate and length of messages, but also an bussload. Because there is not any single master, is it possible that two or more messages are transmitted on the same time. In simulated results CAN is proven to be fast enough, transmission take only 1% from total time per transmitted message if baudrate is 200 kb/s. Every node transmits two messages in every 100 ms. So total bussload is only 2% on used baudrate. Figure 11.

**Figure 11. Simulated transmission delays with CAN protocol when four nodes are used**

**Machine control level**

The control of machine means that machine can be moved in cartesian coordinate. User can move the head of the crane straight in x-direction, y-direction or z-direction. Also combination of those should be possible. Crane is controlled with the joystick.

**Communication tasks**

Communication task is very important part of software, especially with fully distributed control system. All common variables, objects, are updated through the network, it is important that this level is reliable and deterministic. CAN - processor can interrupt main - processor, when message is received.

**Trajectory level**

Trajectory generator calculates trajectory points if longer movements are executed. Generator control movement so that it is smooth and fast. If crane is controlled by a man, then generator is not needed. Trajectory generator is very useful if crane is connected to upper level of control, and
there are given some certain points in space where end-effector should go.

**Actuator control level**

The priority of actuator controller is the highest. Controller level tries to follow instruction produced by trajectory level as well as possible. In actuator control level many different control algorithms are possible. Conventional proportional controller is used as a reference in this research. Better control algorithm, like state controller and fuzzy controller will be tested in this environment. In this research work those controllers are not presented.

This level should be most reliable, and the control procedure should be executed very periodically without any delays. Implementation of this level required interrupt based algorithm, that should be non-maskable, because this level must be executed in all circumstances at 20 ms interval.

**COMMUNICATION SOLUTION**

CAN has proven to be most suitable solution in this case. There is many CAN produces commercially available. Even microprocessors including CAN protocol can be bought nowadays. Those products have anyway only 8-bit CPU and that is why those are not convenient for us. As a CAN protocol processor is choosen Philips 82C200. It support all object and it is cheap.

**CONCLUSION**

In hydraulic mobile crane distribution offer very convenient solution. Differences between MS and FDS are not so big when used crane is quit a simple. With FDS not any powerful master is needed, so it is little cheaper than MS. With MS parallel computing is not so effective than with FDS. An the other way structure of system is difficult to form with FDS. Fully distributed control systems offer economical solution to build the control of mobile crane. CAN-protocol is proven to be very suitable for this kind of purpose. In practice 16 - bits controllers can execute coordinate transformation and actuator controls fast enough. FDS offer one realistic way to build the control system for little or middle size machines like lifts and cranes. With more complicated systems is conventional MS more reliable solution.

**REFERENCES**


