SPRING 2010

即時控制系統設計
Design of Real-Time Control Systems

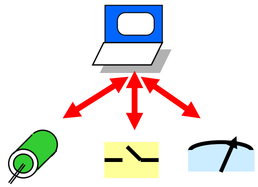
Lecture 35
Impact of Fieldbus on Communication in Robotic Systems

Feng-Li Lian

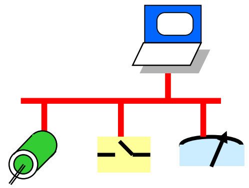
NTU-EE

Feb10 – Jun10

- Real-Time Control Systems
  - Controlled by one Computer Processor
    - Centralized control systems
    - Real-time operating systems
  - Controlled by one Communication Medium
    - Distributed control systems
    - Real-time communications



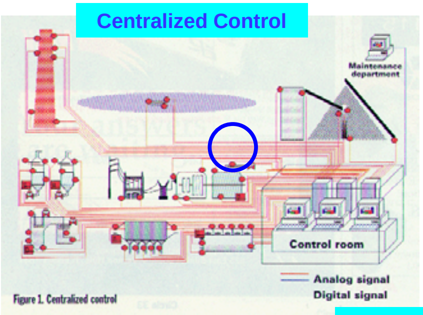
Centralized Control System

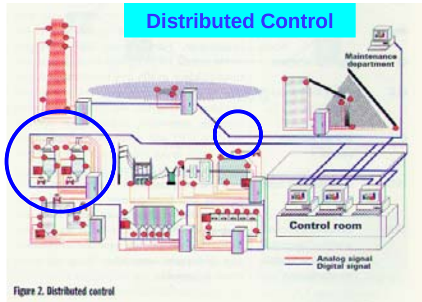


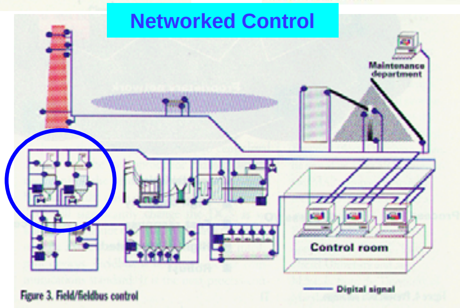
Distributed Control System

- S. Cavalieri, A.D. Stefano, and O. Mirabella,
- "Impact of fieldbus on communication in robotic systems,"
- IEEE-RA, Vol. 13, No. 1, pp. 30-48, Feb. 1997
- Abstract:
  - The aim of this paper is to discuss the problems relating to communication in distributed process control, highlighting the limits of traditional centralized communication systems based on point-to-point connections. The paper intends to show the real advantages that can be obtained by using a distributed commonbus communication system in time-critical process control applications, highlighting its impact on the architecture of control systems from the point of view of both hardware and software. Particular reference is made to the International Electrotechnical Commission/ Instrument Society of America (IEC/ISA) communication system which is close to becoming an international standard. In particular, the solution based on the use of the distributed common-bus communication protocol is applied to the control of an orange-picking robot, which had previously been implemented in a traditional way using point-to-point communications.

  - The choice of a case study referring to a robot control system seems an appropriate benchmark for the real potential of the fieldbus because the processes involved in a robot control system have highly critical time constraints which may stress the fieldbus communication system. The paper presents a qualitative analysis of the advantages that can be obtained by introducing the fieldbus into the orange picking robot control system. The analysis gives a detailed illustration of the hardware and software modification needed to replace the traditional point-to-point control systems of the orange picking robot with the fieldbus. Then an evaluation of the performance of fieldbus system is provided. The evaluation is made by modeling and simulating the distributed communication system with transition timed Petri nets (TTPN's). The results obtained in the study as a whole confirm the authors' conviction that, by careful analysis, it is possible and advantageous even in difficult cases to replace classical centralized point-to-point control systems with distributed systems based on use of the fieldbus.



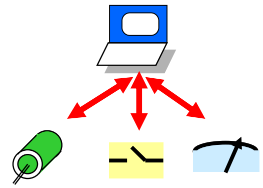




Analog
Digital

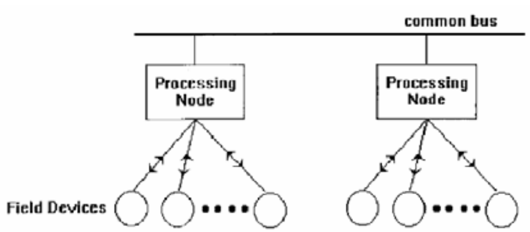
Source: InTech, Nov. 96

- Problems with a centralized architecture:
  1. Complex wiring is needed for point-to-point connection
     - Causing problems in the maintenance and documentation of communication systems
  2. Mainframe computers are characterized by great processing power
     - Cost may be very high
  3. The presence of critical fault points may jeopardize the functioning of large control blocks if malfunctioning
  4. Interchangeability is impossible
     - Systems are proprietary

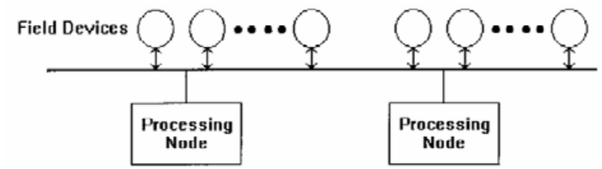


Centralized Control System

- Solution 1: Point-to-point control system:
  - Distribute the control functions of these systems over several processing nodes, each dedicated to a part of the control process and to a group of sensors/actuators
  - The nodes cooperate with each other, communicating through a shared physical channel
  - Problems:
    - Coding and timing

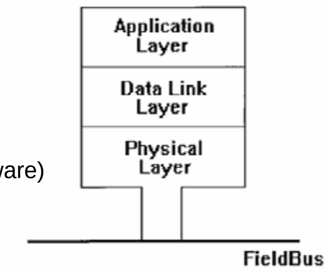


- Solution 2: Fully distributed control system:
  - All the elements (both control and field devices) are connected to a single communication channel
  - The sensors/actuators are therefore resources shared by all the processing nodes
  - Powerful communication protocol is required
    - Objects characterized a communication semantics
    - Medium access control
    - etc.



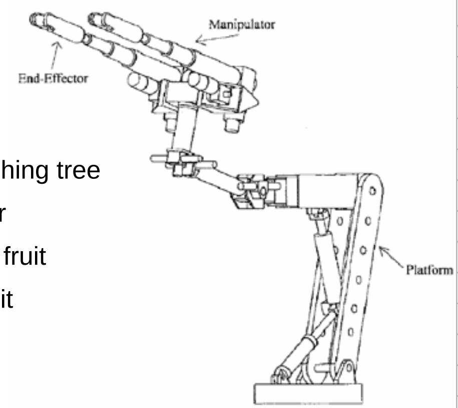
- Fieldbus architecture

1. Application layer:
   - Application process (AP)
     - > A set of resources (software and/or hardware)
   - Application relationship (AR)
     - > Client/server, publisher/subscriber

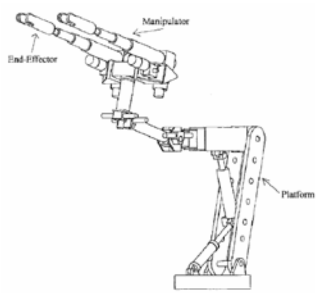


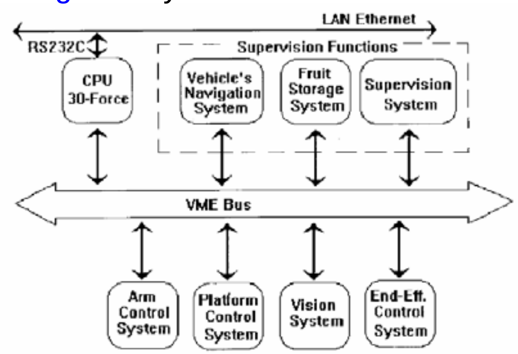
2. Data link layer:
   - To manage all time-critical communication requests
   - Link active scheduler (LAS)
     - > Distribute the available bandwidth
     - > Token
     - > Macrocycle: a scheduling table
3. Physical layer:

- Hardware:
  - 1 mobile vehicle:
    - Monochrome camera
  - 4 platforms:
    - Ultrasonic sensor for approaching tree
    - 2 robot arms with end-effector
    - 1 color camera for position of fruit
    - 1 laser sensor for depth of fruit
    - Storage container

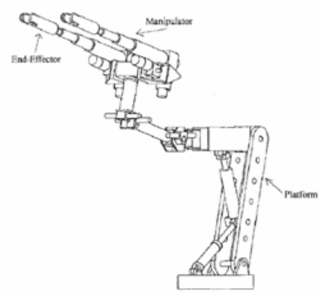


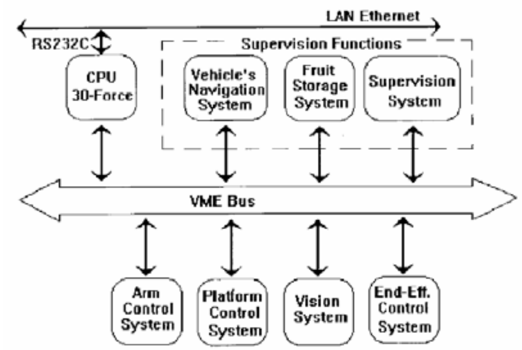
- Master Board:
  - Monitor the state of the system
  - Receive information from the slave boards
    - About the functioning and positioning of the picking units
  - Manage the vehicle's navigation system
  - Store the fruit



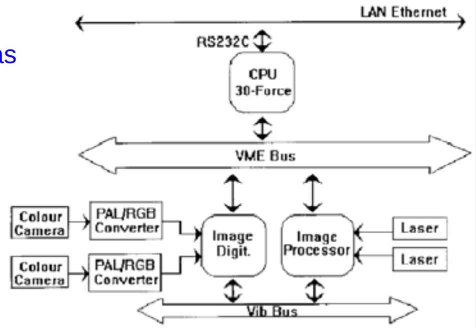


- Slave Board:
  - Vision
  - Platform control
  - Arm control
  - End-effector control

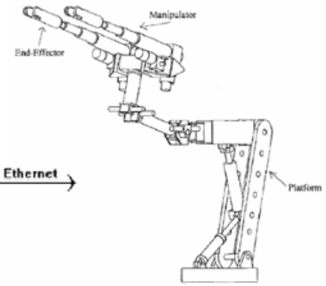


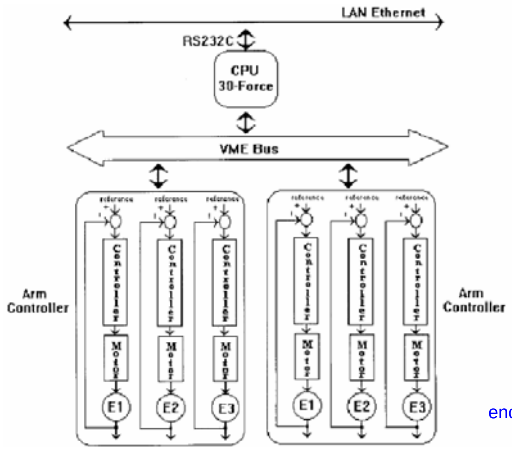


- Vision System:
  - Functions:
    - Acquire the scene
    - Process the images
    - Detect the position of the orange
    - Generate reference signal to guide the picking arms
  - Hardware:
    - 2 independent color cameras
    - Dedicated high-speed bus
  - Tasks:
    - Segmentation
    - Image processing
    - Scene analysis
    - Reference generation



- Arm Control System:
  - Functions:
    - 3 degrees of freedom



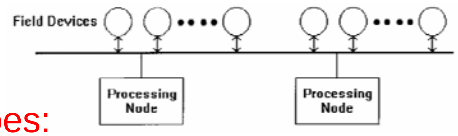


encoder

- Platform Control:
  - Functions:
    - Bring 2 arms up to the picking position
    - Adapt to the dimensions of the foliage of the tree
      - By means of proximity feedback from 2 ultrasonic sensors

- End-effector Control System:
  - Hardware:
    - 1 color cameras
      - Acquire the scene
    - 2 lasers
      - Identify the distance between the end-effector and the orange

- Network parameters:
  - Protocol: IEC/ISA fieldbus
  - Bandwidth: 2.5 Mb/s



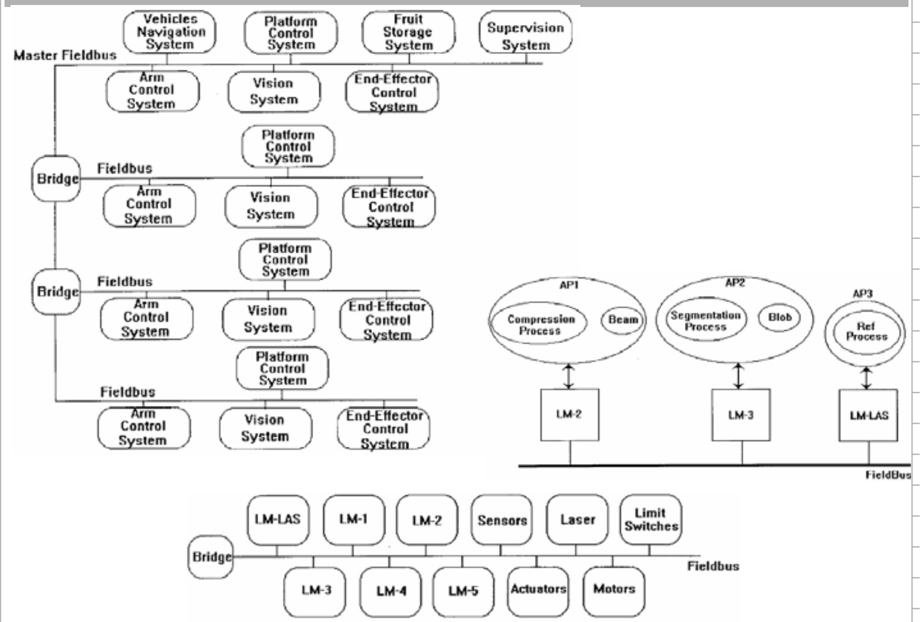
- Data/message traffic types:
  - Time-critical cyclic traffic
    - Periodical sampling of sensors etc.
  - Time-critical acyclic traffic
    - Alarm or command to actuators
  - Non-time-critical acyclic traffic
    - Processes with lower dynamics

Time-critical cyclic traffic

- Between the encoders and the regulation systems
  - > Angular speed: V_{max} (rad/s)
    - » From 0.0021 rad/s to 1.5 rad/s
  - > Resolution: R = 7200 counts/rev. (1 count = 4 pulses)
    - » a pulse is encoded with 2 bytes
  - > Frequency: $F = V_{max} * R * 4 / 6.28$ (Hz)
  - > Min Sampling Time: $S_t = 3.14 / (4 * R * V_{max})$ (sec)
    - » From 51.9 to 0.072 ms
  - > Message size: 7 (LAS) + 7 (control) + 2 (data) = 16 (bytes)
  - > BW for 0.072 ms: (16 * 8) / 0.072 ms = 1.8 Mb/s
    - » BW limit: 2.5 Mb/s

Time-critical cyclic traffic

- In vision systems
  - > Image Resolution: 256 * 256 pixels in RGB
  - > Frequency: every 40 ms
  - > 256 * 256 * 24 = 1.57 Mb every 40 ms = 39 Mb/s
  - > Compression:
    1. compression ratio: 1:25
    2. 24 bits -> 15 bits
  - > Reduce to 39 322 bits = 4916 bytes for every 40 ms
  - > Message size: 256 bytes max
  - > 4916 bytes decomposed into 20 messages
  - > Each message needs 14-byte overhead
  - > Total size: 4916 + 14*20 = 5196 bytes every 40 ms
  - > Bandwidth: 1.040 Mb/s



Navigation System	End-Effector Control System	Fruit Storage System	Crates Management System	Vision System	Platform Control System	Arm Control System
Diagnostic Signal n=1 Ta=1sec Acyclic	Diagnostic Signal n=1 Tc=30 msec Acyclic	Diagnostic Signal n=1 Ta=1 sec Acyclic	Diagnostic Signal n=1 Ta=0.5sec Acyclic	Video Image n=1 L=199 bytes Tc=40 msec Cyclic	Reference Signal n=1 Ta=20 msec Acyclic	Reference Signal n=6 Ta=20 msec Acyclic
Diagnostic Signal n=1 Ta=30 msec Acyclic	Reference Signal n=1 L=78 bytes Tc=20 msec Cyclic	Tube Conveyor Sensor n=2 Ta=60 msec Acyclic	Actuator Command n=4 L=16 bytes Tc=9 msec Cyclic		Limit Switch n=12 L=16 bytes Tc=6 msec Cyclic	Limit Switch n=12 L=16 bytes Tc=3 msec Cyclic
State of Vehicle n=5 Ta=50 msec Acyclic	End-Effector Command n=2 Acyclic	Limit Switch n=6 L=16 bytes Tc=9 msec Cyclic	Limit Switch n=8 L=16 bytes Tc=9 msec Cyclic		Ultrasound Input n=2 L=16 bytes Tc=6 msec Cyclic	
Electro-Valve Command n=2 Acyclic	Laser Activation n=1 Acyclic	Belt Command n=3 Ta=1 sec Acyclic	Digital Input n=2 L=16 bytes Tc=9 msec Cyclic		Supervision Signal n=1 L=16 bytes Tc=12 msec Cyclic	
Video Image n=1 L=78 bytes Tc=20 msec Cyclic	Grasping Sensor n=2 Ta=0.5 sec Acyclic					
Safety Input n=4 L=16 bytes Tc=12 msec Cyclic						
Distance Input n=4 L=16 bytes Tc=12 msec Cyclic						

- T_c: cyclic traffic
- T_a: acyclic traffic
- n: # of information
- L: # of bytes

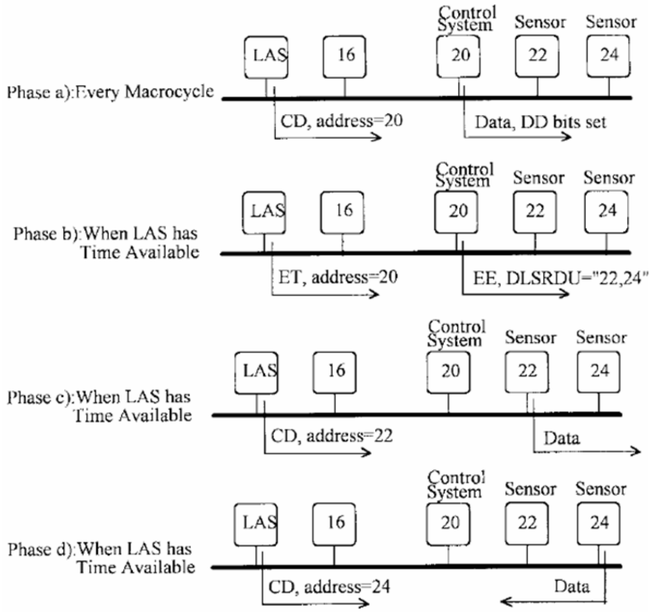
- Timing requirement:
  - From sensors to actuators:
    - > Data size: 2 bytes
    - > Message size: 2 + 14 = 16 bytes
    - > Transmission time: (16*8 (bit)) / 2.5 (Mb/s) = 0.0512 ms
  - Reference signals:
    - > Data size: 64 bytes
    - > Message size: 64 + 14 = 78 bytes
    - > Transmission time: (78*8 (bit)) / 2.5 (Mb/s) = 0.2496 ms
  - Image from the monochrome camera:
    - > Data size: 64 bytes
    - > Message size: 64 + 14 = 78 bytes
    - > Transmission time: (78*8 (bit)) / 2.5 (Mb/s) = 0.2496 ms

- Timing requirement:
  - Compressed image from the color camera:
    - > Data size: 4916 bytes every 40 ms
    - > # of bytes per cycle: 369 bytes
    - > Split it into 2 message:
    - > Message size: 185 + 14 = 199 bytes
    - > Transmission time: 1.27366 ms for these 2 messages

- Cyclic variables in different macrocycles

	Macrocycle 1	Macrocycle 2	Macrocycle 3
3msec	12	12	12
6msec	7	7	7
9msec	8	7	5
12msec	2	3	2
20msec	-	-	1
40msec	2	2	2
Total Cyclic Portion of Macrocycle	2.7584 msec	2.7584 msec	2.8544 msec
Total Acyclic Portion of Macrocycle	0.2416 msec	0.2416 msec	0.1456 msec

Scheduling Policy for Acyclic Traffic



- LAS: Link Active Scheduler
- CD: Compel Data
- DD: data request bit
- ET: Execute Transaction
- EE: End Execution
- DLS: Data Link Server
- RDU: Request Data Unit