Automation Systems

Discrete Event Control Systems and Networked Automation Systems

1st Lecture
Introduction and Overview
Automation Systems in general:

- Task: Control and supervision of technical processes by computing systems
  (or even better: by software running on computing devices)

- Several distinct application areas
  - factory automation
  - process automation
  - transportation automation
  - building automation
  - product automation

- Technical vs. Business Processes
  - Real-time constraints
  - Harsh environment
Application Domains

- **Home Appliances**
  - ASIC

- **Transportation**
  - Process Computer

- **Manufacturing**
  - PLC

- **Automotive**
  - Micro-Controller

- **Process Engineering**
  - Industrial -PC
Automation Pyramid

Management level

Process control level

Field level

Actuator-sensor level

Technical process

Automation pyramid

Intranet

Process bus

Fieldbus

Planning
Analysis
Evaluation

Program
Monitoring
Maintenance plan
Management

Program
Monitoring
Self trend
Threshold

Measured values
Influence values
Example: Air chamber

- Specifications
  1. If PS1 is triggered (Pressure < 6,1 bar), one motor should run
  2. If PS1 is not triggered (Pressure > 6,1 bar), no motor should run
  3. The two motors should run alternately
  4. If one motor is defect, the other one should takeover its tasks
  5. If PS2 is triggered (Pressure < 5,9 bar), both motors should run
Example: Pulper

- Specification
  - Filling solvent
  - Adding solid
  - Mixing and dissolving
  - Emptying the Tank
Example: Gate (1/2)

- Prof. Wagner, Hannover

Diagram:
- Wireless Receiver
- Key-operated Switch
- Remote control
- Key-operated switch
- Limit switch
- Light Barrier

Symbols:
- S1
- S3
- S4
- S5
- K1
- K2
- M
Example: Gate (2/2)

Specifications

- The gate can be opened and closed by controlling the electric motor.
- The gate can be opened from both outside and inside by the key-operated switch.
- Additionally, a remote control is provided, it has the same function as the key-operated switch.
- The completely opened gate should be automatically closed after 20 seconds waiting time.
- A red flashing light should notify the opening and closing of the rolling gate on both sides.
- For safety reasons, a light barrier is mounted. If the light barrier senses an object, the gate should not be closed.
- The rotary direction of the electric motor (clockwise rotation or anti-clockwise rotation) is controlled by two contactors.
- Two limit switches notify the states of „gate is opened“ and „gate is closed“. 
Comparison between Feedback control and Logic control

- **Definition according to DIN 19226 part 1 (translated by AES)**

  - Logic control is a process in a system, in which **one or more input signals** affect the output signals based on the specification of the system. Characteristics of the logic control is that the **control loop is either closed or opened**, furthermore, the output signals, affected by the input signals, are not continuous and do not retroact on themselves. **Note**: The concept of logic control is not only dedicated to the logic control process, but can be also used for the whole system, in which the logic control takes place.

  - Feedback control is a process, in which the **controlled output (single) is continuously compared** with the reference input and intended to follow the reference input exactly. Characteristic of the feedback control is that **the control loop is closed**, and the **controlled output affects itself continuously through the information path in the control loop**. **Note**: The process of feedback control can be also considered as continuous in case that it is composed of the similar frequently repeated individual process (i.e. sampling in sampling control). **The concept of feedback control is not dedicated to the feedback control process, but can be also used for the whole system, in which the feedback control takes place. The resulted action flow takes place in the control loop.**
Definitionen nach DIN 19226 Teil 1


Different perspectives

• View from continuous control
  - Feedback control = closed loop control
  - Logic control = open loop control

• View from Logic control
  - Feedback control = continuous (closed loop)
  - Logic control = discrete (closed or open loop)

• Comparison with a simple example
  - Filling level in a tank
  - Feedback control
    - controlled output = measured value = fill level in tank
    - actuator signal = filling flow
    - Actuator signal is linked up with the controlled output with a simple equation → closed loop
  - Logic control
    - Filling flow is either on or off
    - Two discrete filling levels will be measured
    - The exceeding or undershooting of one of the discrete filling levels triggers the closing or opening of the filling flow according to the algorithm. **But it is not the case, that the exceeding of the levels retroacts on itself**
Block diagram of feedback control

- **Reference input**
- **Algorithm**
- **Actuator**
- **Plant**
- **Controller output**
- **Controlled output**
- **Comparing element**
- **DISTURBANCES** (incl. EMI, environment, ...)
- **Sensors**
- Feedback variable

Controller and comparing element are connected to the algorithm, which in turn sends signals to the actuator. The actuator responds to the controller output, adjusting the plant. The feedback variable is monitored by the sensors, which feed back to the comparing element.
Block diagram of logic control

Controller
Algorithm

Actuators

Plant

Actuator feedback

Input signals
Actuating signals

Output signals

DISTURBANCES
(incl. EMI, Environment, ...)

Feedback variables

Sensors
<table>
<thead>
<tr>
<th></th>
<th><strong>Feedback control</strong></th>
<th><strong>Logic control</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variable</strong></td>
<td>Continuous</td>
<td>Discrete</td>
</tr>
<tr>
<td><strong>Mathematic</strong></td>
<td>Differential equations</td>
<td>Boolean algebra, Automata, Petri nets</td>
</tr>
<tr>
<td><strong>Feedback path in the system</strong></td>
<td>Closed synchronous loop</td>
<td>Asynchronous feedback of binary variables (→ Events)</td>
</tr>
<tr>
<td><strong>Feedback of the variables</strong></td>
<td>Variables in the loop retroact on themselves</td>
<td>Variables in the loop affect others</td>
</tr>
<tr>
<td><strong>Disturbances</strong></td>
<td>Unknown disturbances can be compensated</td>
<td>Only the already know disturbances and recordable interferences can be handled</td>
</tr>
<tr>
<td><strong>Loop gain</strong></td>
<td>Loop gain is defined → stability problem</td>
<td>No loop gain</td>
</tr>
<tr>
<td><strong>Number of signals</strong></td>
<td>&gt;95% control loops are single loop (1 Sensor, 1 Actuator)</td>
<td>Always multi-loop, i.e. often hundreds or more sensors and actuators → complexity</td>
</tr>
<tr>
<td><strong>Specification</strong></td>
<td>„always the same“, standardized: „controlled output follows reference input“</td>
<td>„always new“, cannot be standardized, normally done in a very comprehensive way</td>
</tr>
</tbody>
</table>
• 1800 first applied control in the industrial scale (mechanical) (steam-engine, watt’s fly-ball governor)
• 1920 electrical controller (Kammgarn)
• 1940 pneumatic control
• 1960 contactor control as well as relay control with wiring diagram
• 1969 first PLC (Modicon 084)
• 1970 PLC-control with programs in ladder diagram and instruction list (i.e. Step5)
• 1980 Microcontroller-control with programs in text-based programming languages
• 1990 IPC-based or μP-based control with graphical programming

Watt's fly-ball governor

Modicon 084 (1969)
Categorization of Logic control (1/3)

- **Task-oriented (DIN standardization translated by AES)**
  - combinatorial control
    - DIN 19226 part 5: logic control relates the input signals and output signals in such a manner, that the states of output signals are decided by the Boolean operations of input signals. Furthermore, control processes containing logical elements and storage- and time-functions but no stepwise actions can be also termed as combinatorial control.
    - Briefly: mapping the output signal assignment to input signal assignment

- **Sequential control**
  - DIN 19226 part 5: sequential control is a control process that includes inevitable stepwise actions. In the process, the transfer from one step to the next one depends on the planned transfer-condition. The step sequence can be programmed by specific methods, i.e. with jumps, loops, branches. The steps in the control process are often arranged to correspond to the process-related sequence.
    - Briefly: stepwise execution of an algorithm with time-related or process-related step enabling conditions.
Klassifizierung von Steuerungen (1/3)

- **Aufgabenorientiert**
  - Verknüpfungssteuerungen
    - Kurz: Zuordnung von Ausgangssignalbelegungen zu Eingangssignalbelegungen
  - Ablaufsteuerungen
    - Kurz: Schrittweise Abarbeitung eines Algorithmus mit zeitabhängigen oder prozessabhängigen Weiterschaltbedingungen
Categorization of Logic control (2/3)

- **Realization oriented**
  - hard-wired
    - HW-solution
    - safety engineering
    - < 5% of the whole engineering
  - stored-program
    - programmable controller with interchangeable memory (PLC, microcontroller)
    - freely programmable (PLC, Industrial-PC)
    - >95% of the whole engineering

- **Software oriented**
  - textual
    - Instruction List (IL), Assembler
    - Structured Text (ST), Pascal
    - C, C++, Java
  - graphical
    - Sequential Function Chart (SFC)
    - Function block language
    - Ladder Diagram (LD)
• Signal-oriented

➤ **Digital control**
  • Numerical values and codes, utilization of counters, arithmetic logic units, registers etc.
  • Example: control with digital signals, accessing by the use of device address etc.

➤ **Binary or Logic control**
  • No numerical values and codes, utilization of logic-, time-, storage-units.
  • Example: pulper, air-chamber

Logic Control: 1\textsuperscript{st} part of this course
Example: Emerson's PlantWeb (Delta V)

You can choose the level of redundancy your application requires, including:
- Redundant Ethernet network communications
- Redundant controllers
- Redundant power supplies
- Redundant H1 FOUNDATION fieldbus interface and bus power
- Redundant digital HART I/O
- Redundant MODBUS and other RS485 serial communications
- Redundant workstations

Rugged control and field interfaces
Built to withstand harsh environments and ensure safety, these interfaces:
- Class 1, Div. 2
- Zone 1 and 2
- Intrinsically safe options
- Corrosion resistance
- -40 to 70°C

Commercial off-the-shelf technologies
Proven, low-cost, easily integratable commercial technologies are the fundamental building blocks of the DeltaV system. Technologies proven across many industries and known by a wide pool of professionals:
- Windows workstation and server-based PCs
- Ethernet technology
- Bus standards

Digital field interfaces
Digital busses deliver big project savings in wiring and system footprint. Digital communications include:
- FOUNDATION fieldbus
- AS-i
- DeviceNet
- Profinet
- HART
Lessons learned from this Example

• Very large number of different networks
  ➢ Need for classification of networks
  ➢ Large numbers necessitate generic approach
  ➢ ISO/OSI 7-layer model of communication

• Most real applications combine several networks
  ➢ Need for connections between networks
  ➢ Connections are possible on different abstract layers
    (repeater, hub, bridge, switch, router)
Current Trend 1: Ethernet Everywhere (Jetter JetWeb)

- General Idea: Ethernet with TCP/IP replaces all other networks especially fieldbus systems on the lower levels of automation
Current Trend 1: Ethernet Everywhere (Jetter JetWeb)

Example: Current situation in a manufacturing system

- 4 types of networks
- additional direct links
Current Trend 1: Ethernet Everywhere (Jetter JetWeb)

Example: Future situation in a manufacturing system

- only Ethernet remains
Current Trend 1: Ethernet Everywhere (Pros and Cons)

Pros:
✓ Inexpensive: off-the-shelf components
✓ Simple structure: only one network technology
✓ Uniform Parameterization and Maintenance
✓ Transmission of large amounts of data possible
✓ Access to all components across different levels of automation

Cons:
✗ Nondeterministic behavior, no priorities
✗ No access for low cost components. Ethernet (+ TCP/IP) stack needs to be implemented on each participant of the bus.
Example: Rockwell (Allen-Bradley) NetLinx

- **NetLinx™**
- Programmable Device Support PC
- Desktop PC with excel
- **EtherNet / IP**
- Controller and Bridge
- Servo
- HMI
- **ControlNet**
- Linking Device
- Drive
- **DeviceNet**
- Modular I/O
- Micro PLC
- Sensor
- Block I/O
- SoSe 19
- Georg Frey
Example: in-car networks
Automation task:
To stop the motor when the trolley arrives at the sensor position.

Problem (Analysis and Design):
Determine the delay between the event "the trolley is detected (S1)" and the event "the motor stops (O1)".
Example of an NAS: Comparison of different Architectures

**Without fieldbus**
- PLC

**With classical fieldbus (RS485)**
- Profibus slave
- PLC master
- Profibus-DP

**Networked Automation System (Switched Ethernet as fieldbus)**
- DIOD
- PLC
- Switch
- Switched Ethernet
- Switched Ethernet network between I/O and PLC

Direct link between I/O and controller
RS485 link between I/O and controllers
Switched Ethernet network between I/O and PLC
Example of an NAS: End-to-end delay forecast

• From sensor to actuator
  ➢ Distributed IO devices (DIOD): time to
    • filter binary IO
    • code or decode Ethernet frames
  ➢ Switches: time to forward the Ethernet frame
  ➢ Communication protocol: interframe gap
  ➢ Controllers (PLC): time to
    • code or decode Ethernet frames
    • execute program

• All Ethernet resources are potentially shared

• delay not constant and difficult to determine a priori
Lessons learned from these Examples

- Delays / Packet Loss can influence Product Quality
- Delays / Packet Loss may lead to unsafe behavior

- Influence has to be determined (simulation, formal analysis)
- Design methods have to be adapted
- Influence of automation system operation has to be considered (cyclic vs. event-based)

Topics to be dealt with in this course:
Networks in Automation: 2nd part of this course
1. Introduction to Logic Control
2. The control design process
3. Basic formal interpretations
4. Realization of logic control systems
5. Domain specific languages (IEC 61131)
6. Formal specification using Petri Nets (SIPN)
7. Analysis of Petri Nets
8. Analysis of SIPN
9. Implementation of SIPN using IEC61131
10. Signals and Communication
11. Network Examples (Ethernet and CAN)
12. Design of Distributed Controllers (IEC 61499)
Summary of Lecture 1

- Classification of logic control based on different criteria
  - Definitions of sequential control/combinatorial control

- Discussion about Networks in Automation
  - Specific Network Types and Requirements

- Overview about the contents of the lecture