Design of Servo Motion Control Systems

A servo motion control system has four main components (Fig. 1):

1. Servo Motor with position sensor; incremental or absolute encoder, or resolver
2. Servo Drive: amplifier with DC power supply
3. Motion Controller
4. Human Machine Interface (HMI) device

For each servo axis, a servo motor with position sensor and a servo drive are needed. Controller can be one for each axis or just one multi-axis controller which controls many axes. In current state of art systems, motion controller is a multi-axis controller (Fig. 2 and Fig. 3).

Other components may include

5. Mechanical motion transmission and coupling mechanisms to couple the motor to machine axis
6. Other sensors, i.e. tension sensor in web handling applications, vision system.
7. Other controllers such as PLCs or other motion controllers for the controller to communicate and coordinate with.

Most of the electrical components (HMI outside the electrical panel, motion controller, servo drives are inside the panel) are typically housed in an electrical enclosure. In addition, input power safety, filtering and AC/DC conversion devices, as well as wiring terminal blocks to make the proper wiring connections.

Fig. 1- Components of a single axis servo motion control system

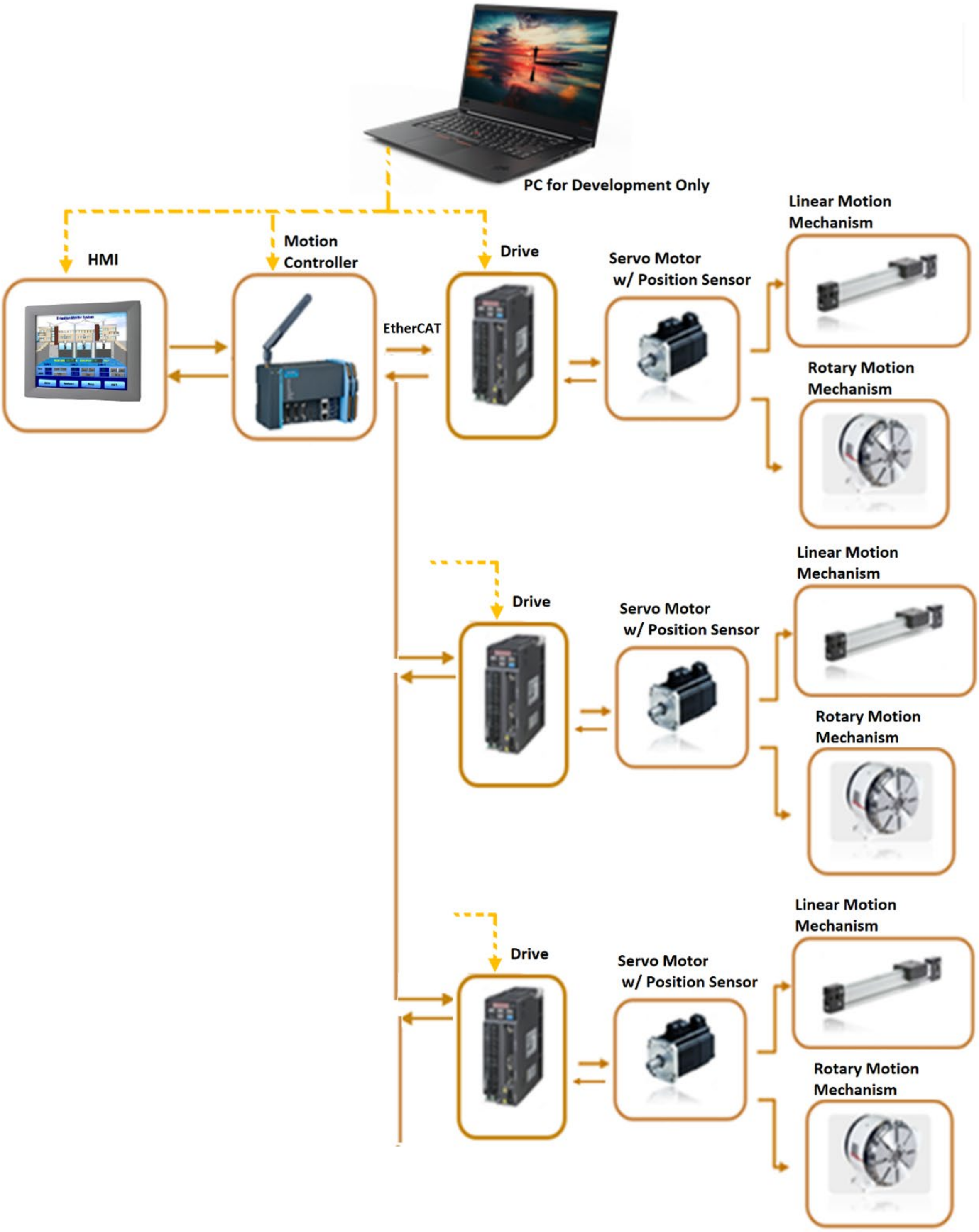


Fig 2- Components of a multi-axis servo motion control system

Multi axis motion control system includes an HMI, a PLC/Motion Controller, servo motors and drives (one pair for each axis), and the motion mechanisms. The HMI and PLC/Motion Controller hardware for single and multi-axis (i.e. up to 64 axis) motion control system are same

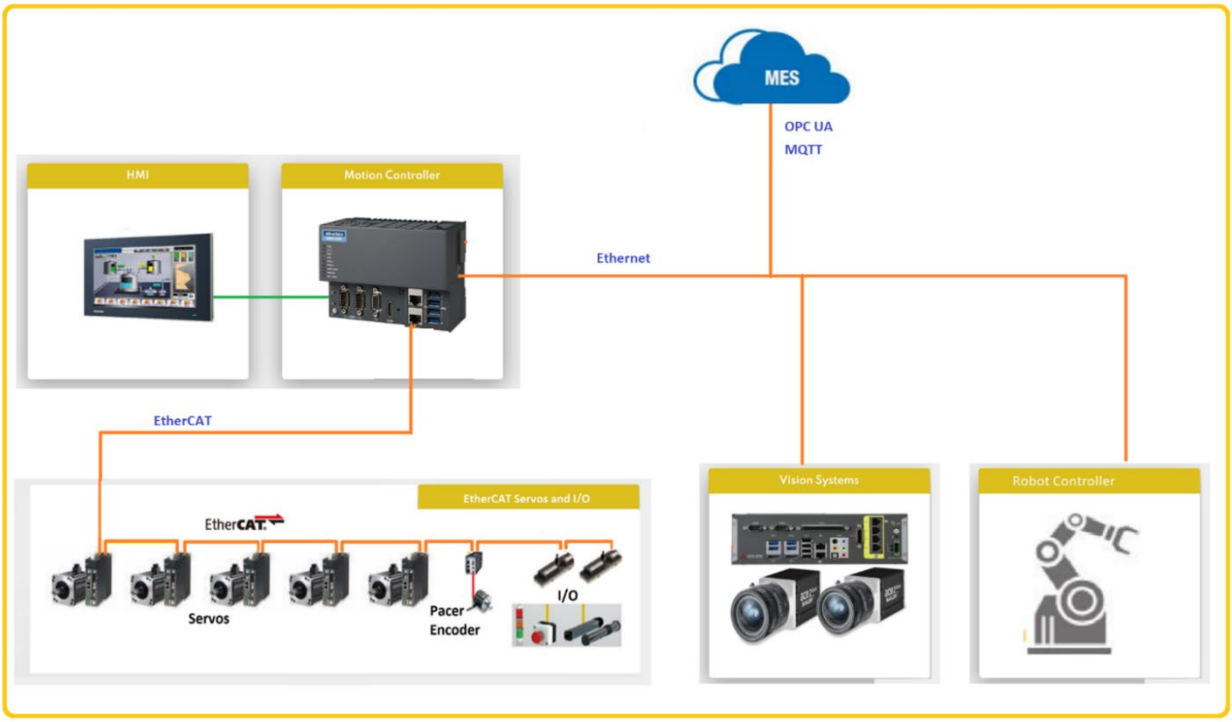


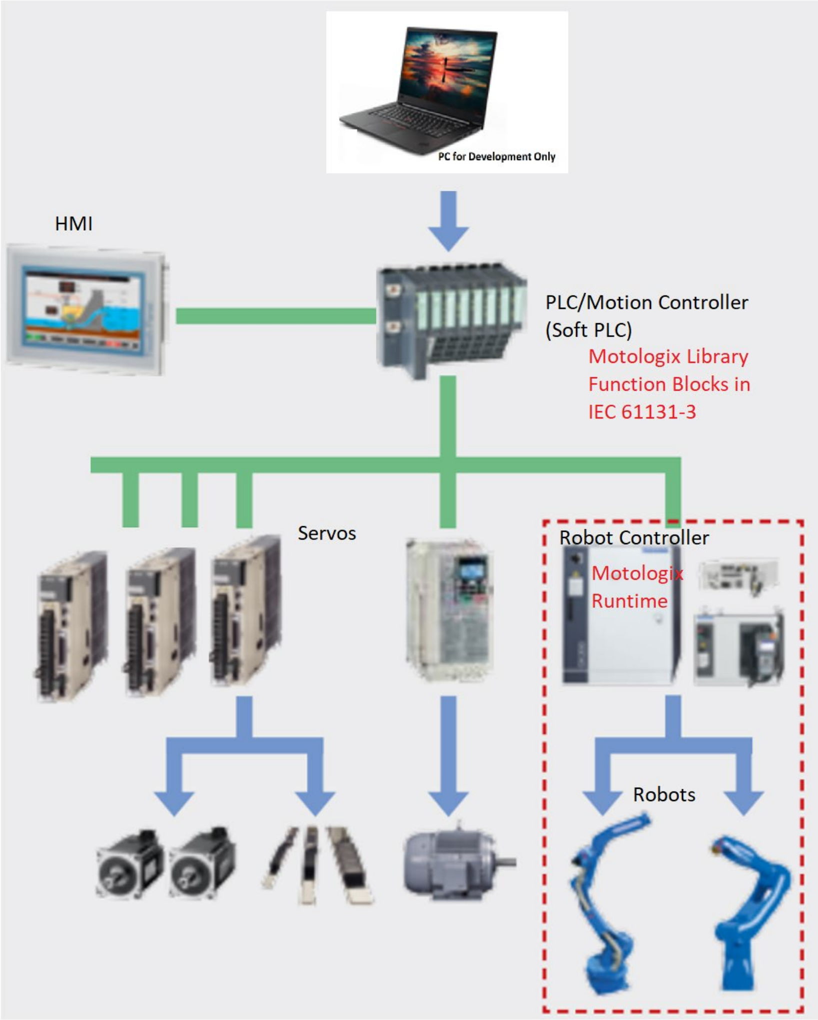
Fig. 3 - Multi-axis servo motion control system applications: robotics, XYZ tables, CNC machines, assembly machines, web handling machines, packaging machines, surgery robotics.

State of art factory automation is a system which provides access to the lowest level I/O device, i.e. turn ON/OFF a light on the factory floor, from anywhere in the world via Internet connectivity (Fig. 4). The automation system is part of the IoT network. However, extreme caution should be exercised in the proper use of the IoT and Internet access to the machinery remotely as this can lead to security breaches and create catastrophic safety problems. There are three main levels network communication in factory automation:

1. Internet (IoT) connectivity
2. Control level connectivity; i.e. EtherCAT, EtherNet/IP, Profibus DP etc to facilitate communication between subsystems
3. Device Level communications; EtherCAT, CAN bus, ModBus etc, for each controller to access the I/O devices, i.e. ON/OFF devices, servo drives.

For example, in an automation cell, a robot and its controller would act as part of the whole automation as sub-system. PLC/Motion Controller implements the coordinated motion control logic for specific application. As part of that logic, it would communicate with the Robot Controller to exchange information to make sure the Robot does its part of the job.

One good technological example for Robot and general purpose PLC/Motion Controller integration is with Yaskawa Motoman Robots. In that configuration, Yaskawa Motoman Robot Controller has its various robot control related functions implemented. It provides communication interface software for a higher level controller using IEC 61131-3 standard based function blocks, over EtherCAT or Ethernet/IP communication. The PLC/Motion Controller integrates the robot motion control as part of the overall machine control software using IEC 61131-3 function blocks, just like any other automation software solution. That way, end-user does not have to learn the robot supplier specific programming language for the robot. The end-user deal with the robot controller via PLC function blocks in IEC 61131-3 standard.



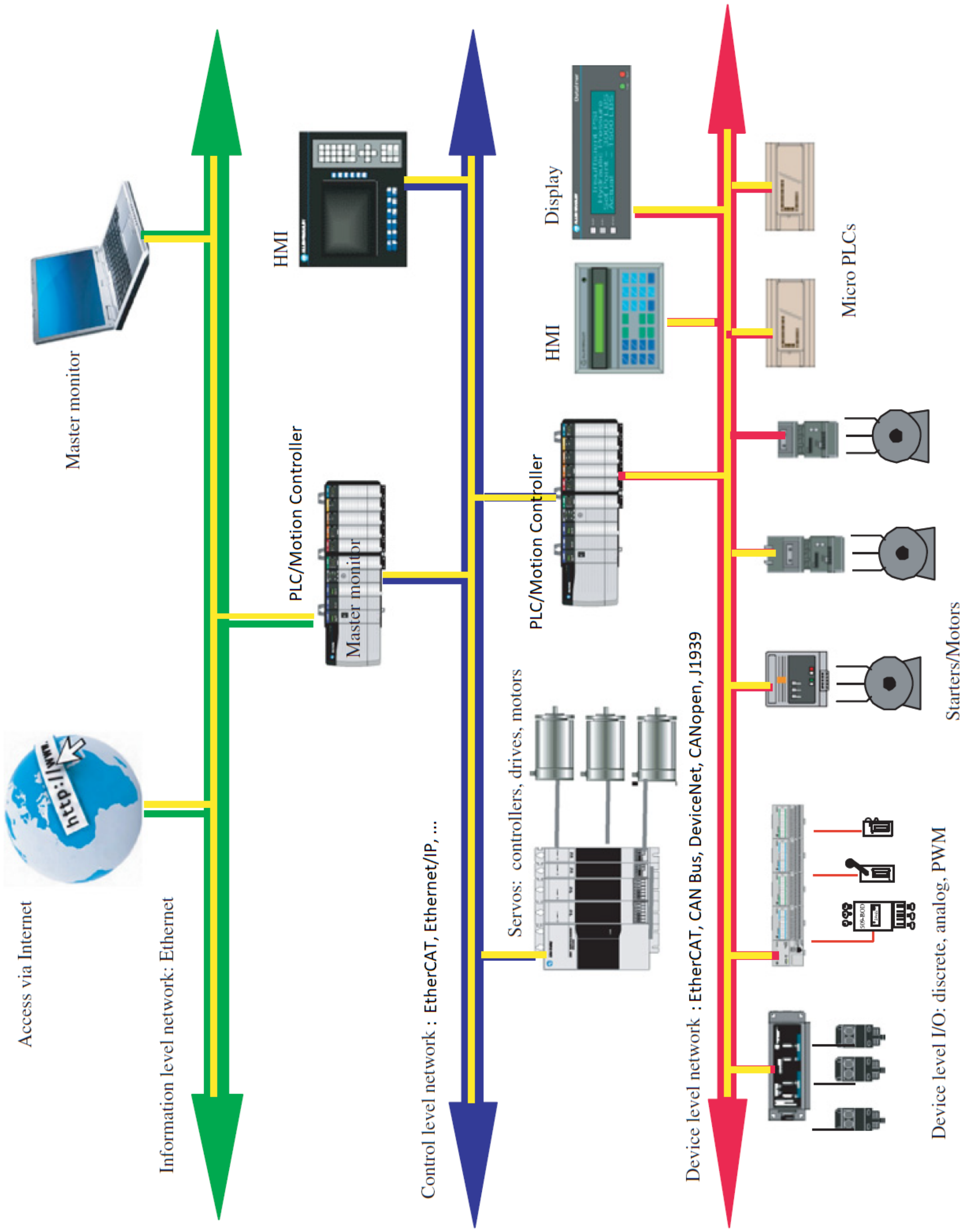


Fig. 4 – Communication network layers in factory automation

DESIGN STEPS

Step 1: Describe the operation of the machine; first as an overall modes of operation, then the details.

1. Required modes of operation (setup mode, manual mode, auto mode, power-up sequence things to do).

2. Required operator interface, that is what commands the operator will give, what status information the operator will need, what data the operator can change and monitor.

Step 2: Identify the required hardware components and design integration of the electromechanical system (mechanical assembly and electrical wiring of components).

1. Actuators: servo motors, DC motors, hydraulic or pneumatic cylinders.
2. Sensors: encoder feedback, proximity switches, photo-electric switches.
3. Controllers needed: PLCs, servo controller, sensor controller.
4. Operator interface devices.
5. Motion transmission mechanisms: gears, lead-screws, timing belts.

Step 3: Design electrical wiring diagrams: how each component is connected in the circuit (system).

Step 4: Develop application software for PLC/Motion Controller, HMI, Vision System, etc.

1. Top-down tree structure design of software.
2. Pseudo-code.
3. Code in specific programming language.

Step 5: Set up each programmable motion axis

1. Basic hardware check.
2. Power-up test.
3. Establish communication.
4. Default parameter set-ups: CW/CCW direction, torque limit, position limits,…
5. Servo tuning (*K*p,*K*v,*K*i,*K*vf ,*K*af ,…, values)**.**
6. Simple moves: Jog, Home, Single incremental motion (“Index”) parameters.

Step 6: Application specific coordinated motions and Auto Mode

Step 7: Debug, test, verify performance.

Step 8: Documentation: document clearly so that later on someone else can debug and/or modify the machine.

Power input and power signal conditioning (noise filter) as well as safety control of it via circuit breaker and contactor includes the following components (Fig. 5). These components must be sized to handle the rated voltage and number of phase (single phase or three phase) and rated current load for the system they support downstream.

Wires and cables varying high power signals, such as those connecting supply power and servo drives and servo motors, should be as far away as possible from the low-power signals such as sensors and controller IO signals in order to minimize the potential electromagnetic noise that would be induced on the low power signals by the high power signals. Basic wire-guideways (the “path” or “roads” that the respective wires are connected) should be planned in order to minimize the electromagnetic coupling by simply providing as much physical space as possible between the high power and low power signals.

Use of terminal blocks in the middle between two components to make a wire connection between a terminal on device 1 and another terminal on another device makes the tracing and troubleshooting the electrical wiring system easier.

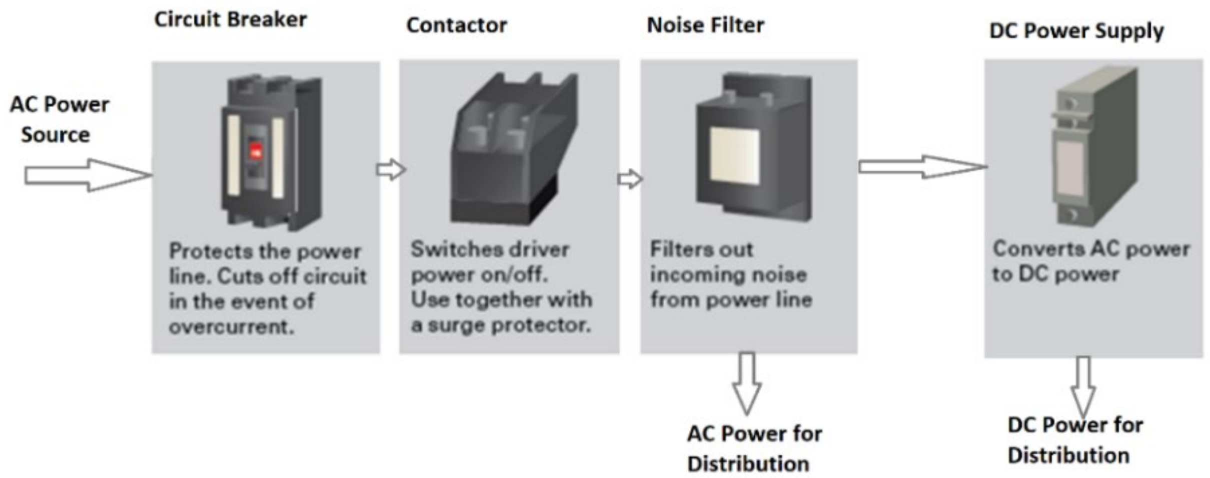


Fig. 5- Input power safety, noise filtering and AC/DC conversion in electrical control panels

From a design engineer's point of view, the design considerations must be given to each component as follows.

1. **Servo Motor:** State of art servo motor technology is undoubtedly brushless DC motors (sometimes referred as AC servo motors). Three phase stator winding and permanent magnet rotor assembly is the most commonly used type. The four most application related parameters of a servo motor are (Fig. 6, 7, 8)

    1. Rated power
    2. RMS and peak torque capacity
    3. Maximum speed
    4. Rotor inertia

Other parameters of interest are

5. Torque gain, hence RMS current and peak current it can handle
6. Rated terminal voltage that can be applied by the drive to control current
7. Winding resistance
8. Winding inductance

Mechanical dimensions of the motor; face mount dimensions (X and Y dimensions), motor length (Z-dimension) as well as motor shaft diameter and shaft modifications (keyway, tapered, solid round) are needed for mechanical assembly to the application machine.

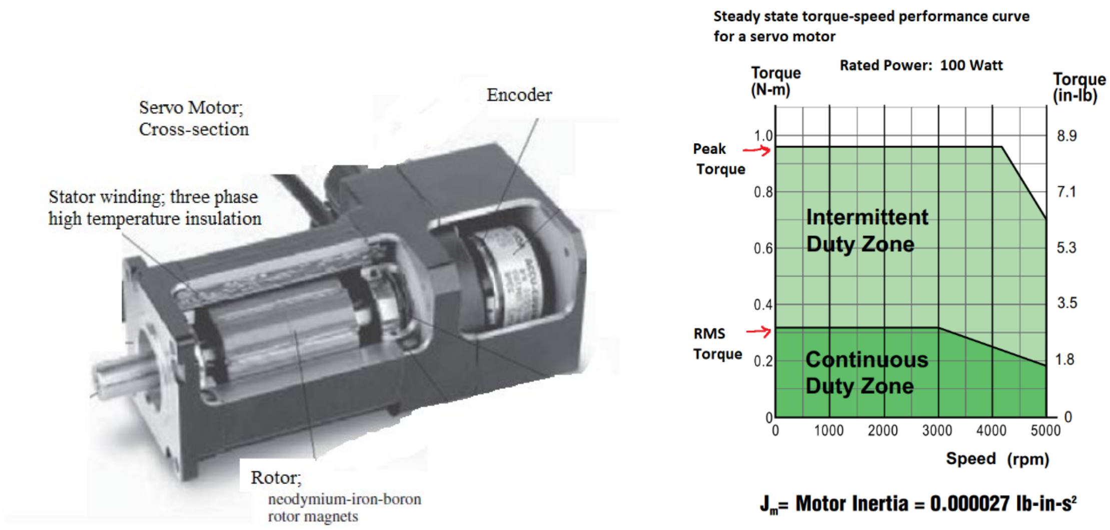
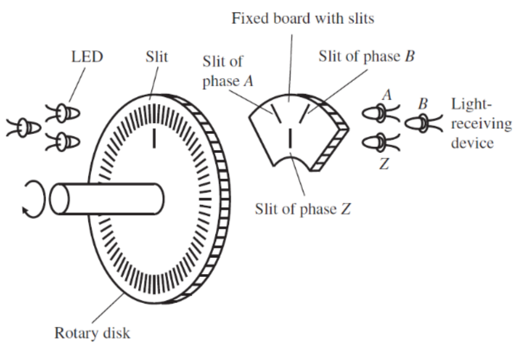


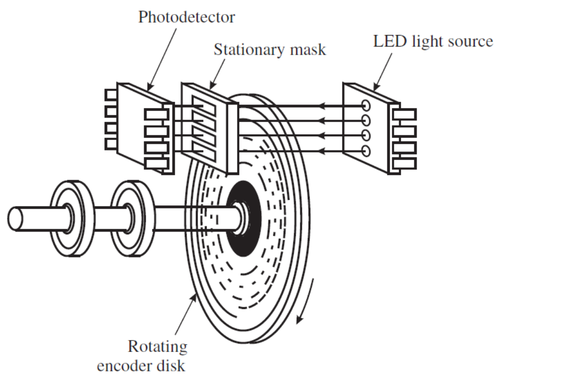
Fig. 6 – a) Servo motor cross-sectional view, b) Steady state torque-speed performance showing continuous and intermediate duty zone of operation.

Encoders are either incremental or absolute type with 17-bit to 24-bit or even 26-bit resolution per revolution (per 360 degrees of motor shaft rotation). For encoders design parameters are

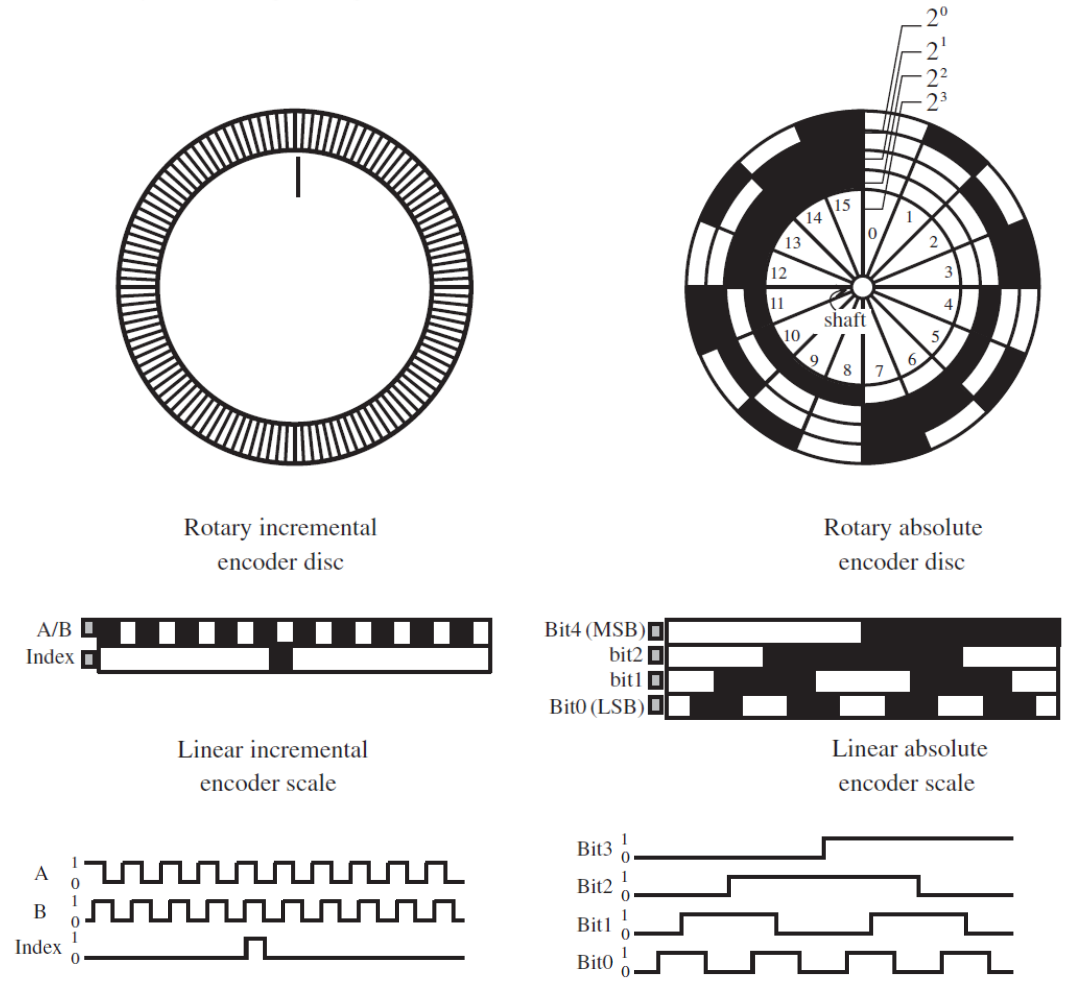
1. Encoder type: incremental or absolute (battery backed or not)
2. Resolution: [counts/revolution]
3. Encoder signal output type: Single ended, differential ended, EtherCAT communication interface



Components and operating principle of a rotary incremental encoder: rotary disk with slits, LEDs, phototransistors (light receiving devices) mask.



Components and operating principle of a rotary absolute encoder: rotary disk with absolute position coding in gray scale, LED set, phototransistor set, mask.



Disks of incremental (rotary and linear) and absolute encoders and the typical encoder output signals.

Fig. 7 – Encoders: incremental and absolute type, rotary and linear type, the output signals.

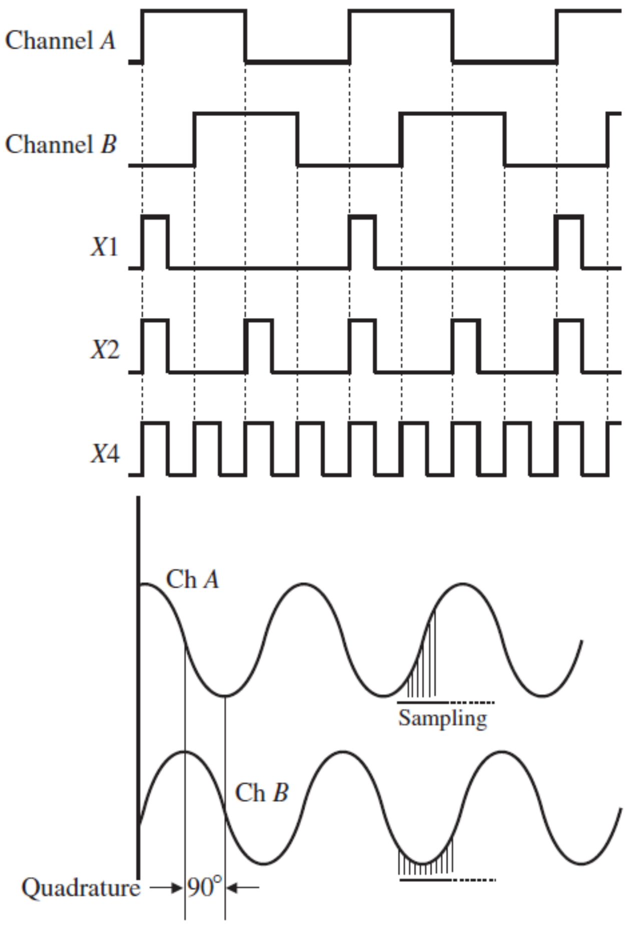


Fig. 8 – Quadrature and sinusoidal incremental encoder signals.

Quadrature decoding method of processing A and B channels of incremental encoder signal effectively increases the position measurement resolution by x4 (4 times the number of lines, that is the number of pulses per revolution in channel A and B). For example, an encoder 1024 lines (1024 pulses in Channel A, and 1024 pulses in channel B per revolution) provides 4096 = 4 x 1024 counts/rev.

Maximum frequency the encoder can handle divided by the number of counts per revolution determines the maximum speed the encoder can run. Notice that, as the number of counts per revolution (1/resolution) increases, the maximum speed rating decreases.

Very high precision incremental encoders provide analog signal output per line instead of ON/OFF state of signals, i.e. sinusoidal variation of the signal magnitude per line change in the rotor position. Then, the analog signal can be sampled to further increase the resolution of the position measurement. For example, if we have an incremental encoder with 4096 lines (12 bit resolution), and each line gives a sinusoidal output which is sampled at 8-bit resolution which means diving a single line signal to 2^8=256 levels, the effective resolution is then 4096 * 256 counts/revolution, which is 20-bit resolution.

The same concept is applied to the modern absolute encoders, hence increase their resolution to rather incredibly small values, i.e. 24-bit and 26-bit per revolution absolute encoders.

2. Servo Drive: Drive is a DC power supply plus amplifier (Fig. 9). The amplifier for a brushless servo motor is essentially a 6-power transistor set to control the current in each one of the three motor phases in the stator winding. The amplifier is size matched to the servo motor in order to provide the proper capacity of current and voltage. Maximum current capacity determines the maximum (peak) torque capacity of the motor at zero speed. Maximum voltage capacity determines the maximum speed capacity at zero torque. Motor steady state torque-speed performance curves are rated usually a little below these theoretical maximums.

There are at least three control loops closed for each servo axis:

1. Current loop
2. Speed loop
3. Position loop

In implementations of these loop in software, position and speed loop may be combined into one algorithm. Forth loop is added in application specific cases such as tension, pressure or force control loop. For example, in tension control of web motion, we use a tension sensor for a tension control loop. The output of the tension control loop becomes an additional signal or command signal to the position loop. Likewise, in robotic force sensor on the tool or injection molding applications pressure sensor on the injector cylinder is used for force control loop. For a successful application, each one of these servo loops (current, speed, position, and force/pressure/tension) must be tuned properly. That means parameters of the control algorithms must be set to numerical values that is appropriate for the application. That is called "servo tuning". The rule of thumb is that the each inner loop bandwidth should be at least 3 to 5 times faster than its outer loop bandwidth.

These loop functionalities may be implemented in the Drive (as almost every servo drive has powerful built-in microcontrollers/DSPs to implement these control logics and I/O interfaces) or in the motion controller. Individual division is of where the motion controller ends and drive functionality begins is up to the system designer since any commercial servo drive is capable of implementing all three or even four servo loops. Similarly, almost any servo motion controller is capable of implementing position and velocity loop as well as forth loop; tension, force etc. Current loop is almost always implemented in the Drive. The amplifier implements current control loop (i.e. PI algorithm) for each phase current and commutation of the current, that is the functional dependence of current distribution to each phase as a function of rotor position, which is known from position feedback device. However, there are rare commercial applications that the Drive provides only DC power supply and open loop power amplifiers (transistors) for voltage amplification and motion controller closes the current loop using the current feedback signal from the drive.

In software, a given servo drive and motion controller can be configured to decide which one implements which loop; in other words, where is the "red line" (in the figure below) drawn between them in terms of task division.

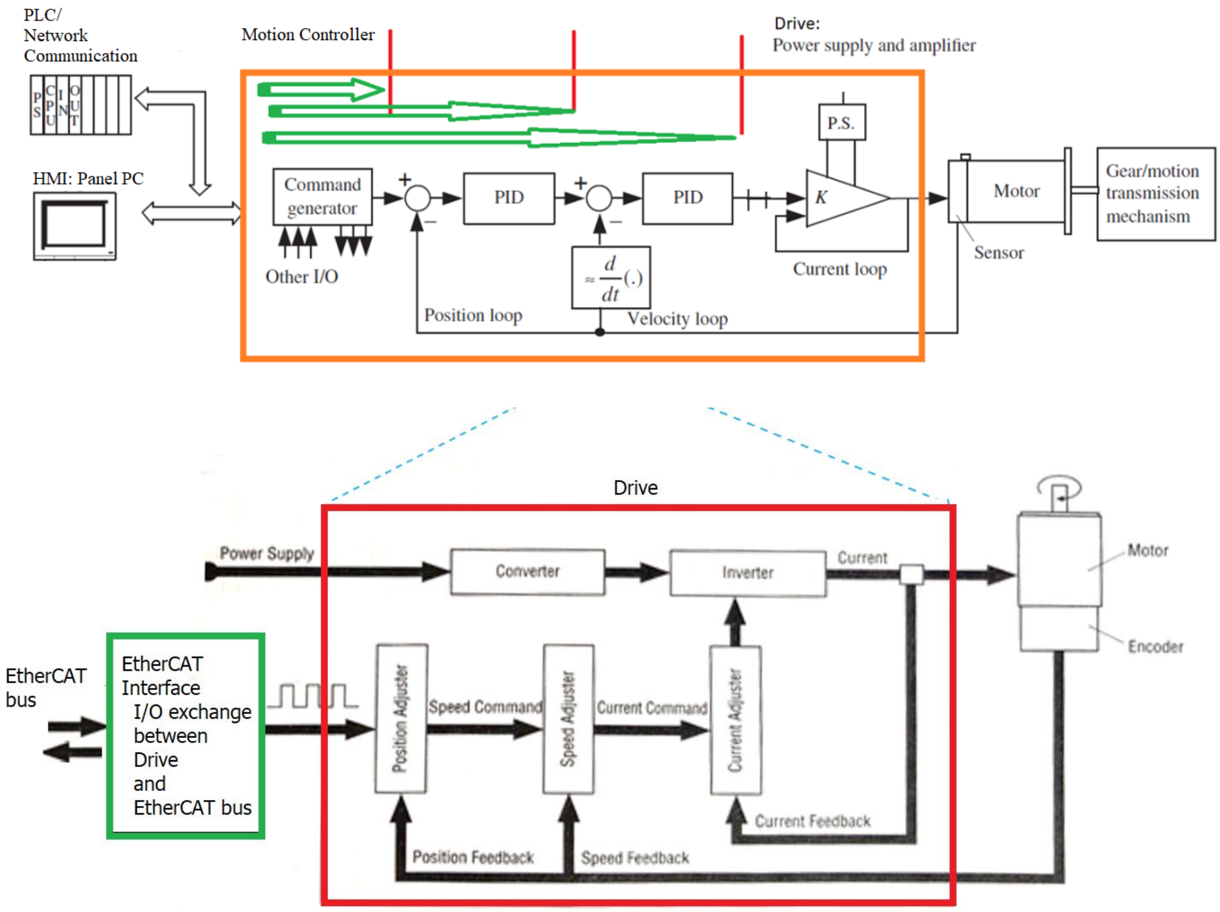


Fig. 9 – Closed loop servo control system components (top); functional components of a EtherCAT compatible servo drive (bottom).

3. Motion Controller: By analogy, the motion controller is the "brain", sensors and vision system are the "eyes", drives and motors are the "muscle", and HMI is the "human command center" of the system. With these configurable components, a servo motion control system is customized for **any machine motion control application.**
Motion controller selection involves the following choices:

- Number of servo axes it supports (i.e. 1, 2, 4, 8, 16, 32, 64, 128,…)
- Number of general purpose (analog and digital) IO it supports
- Interface type between controller and servo drives: i.e. hardwired IO lines for command signals and sensor and status signals, EtherCAT interface.
- Other communication modules and protocols it supports, i.e. Ethernet/IP, CAN, Modbus, USB.
- Software development environment (IDE) and languages supported: i.e. supplier specific languages, or IEC 61131-3 programming language being the fastest growing one. The software tools and their capabilities is one of the most important aspects in making a choice in the motion controller for an application. The expression in automation industry is that "**The name of the game in automation is software**".

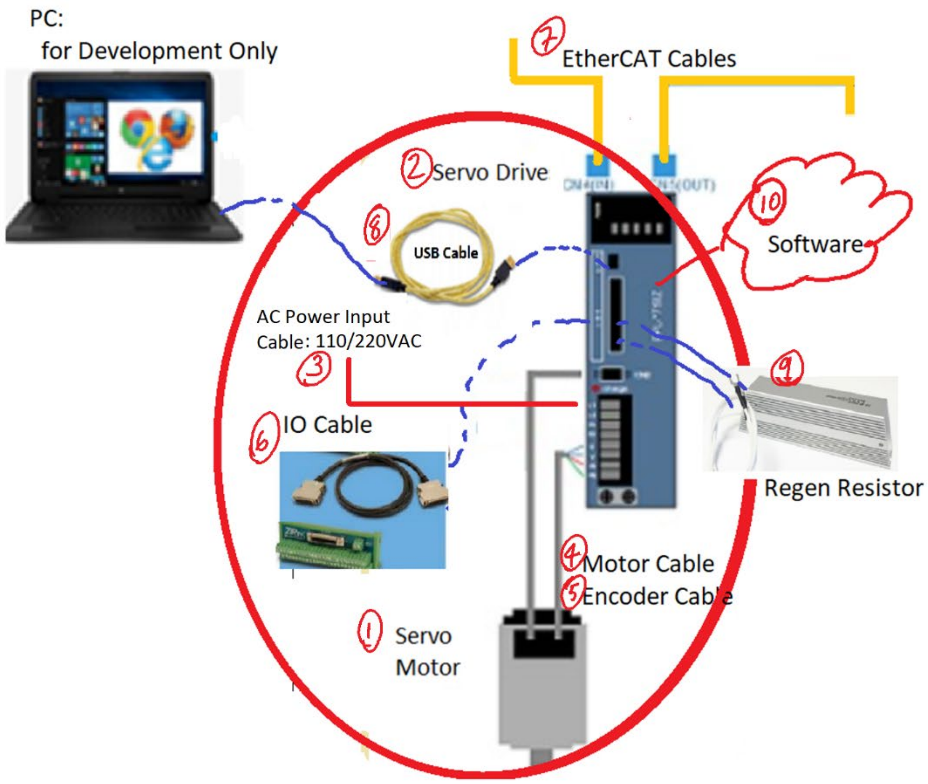
Motion controller based on industrial PC EtherCAT hardware and CODESYS software is one of the most powerful and flexible, supplier independent technologies. Hardware and software are based on global open standards, benefiting from economies of scale, flexibility, expandability, rugged, fanless embedded PC based hardware, supplier independence. Any EtherCAT compatible servo drive and IO module can be added from any combination of suppliers.

4. HMI: HMI device is interfaced to the controller via Ethernet TCP/IP, hence HMI can be local or remote anywhere in the world. HMI can be in two different forms:

a. Monitor (i.e. Touch Screen) with optional Keyboard and Mouse: In this case, the cost of the HMI device is lower. However, HMI device is simply an input (touch interface, keyboard, mouse) and output (monitor) device without any programming capability. All of the HMI logic and graphics is handled by the PLC/Motion Controller.
b. PC (i.e. Flat Panel PC, Laptop PC, or any PC): In this case, cost is a little higher compared Monitor+Keyboard+Mouse HMI. However, the HMI device, which is a PC, has not only the input-output devices but also full programmability. The HMI related graphics and logic can be implemented on the HMI PC, relieving the PLC/Motion Controller from this task and allow more computational time to be available for non-HMI control tasks. HMI PC and PLC/Motion Controller simply communicated with each other through a standard communication protocol such as OPC UA. The periodic update rate of list of variables between HMI PC and PLC/Motion Controller is typically in the range of 100msec to 300 msec. This is fast enough as a Human Operator's reaction time is about 300 msec.

Summary:

Configuring a single servo axis involves the selection of the following components (Fig. 10): four main components are servo motor, servo drive, motion controller and HMI. Additional components are "regenerative resistor" for some applications, cables (AC power input cable, motor-drive cable, encoder-drive cable, IO cable to wire custom axis level IOs to the drive, EtherCAT cable for real-time communication, USB cable used for off-line programming with PC) and, finally and very importantly the "software". There are two kinds of software involved: one to configure the servo drive (provided by the servo drive supplier) and one for the motion controller to program multi-axis motion control logic.



Single Axis Servo System Components:

1. Servo Motor
2. Servo Drive
3. AC Power Input Cable
4. Motor Cable
5. Encoder Cable
6. IO Cable
7. EtherCAT Cables
8. USB Cable
9. Regen Resistors and Cable
10. Software

Fig. 10 – Components of a single axis servo motion control system.

Servo motor and servo drive are power matched to mee the application power requirement.
The key parameters for servo motor are Peak and RMS Torque, rated speed, motor rotor inertia.
for servo drive are matched to the motor to provide required peak and RMS current, DC bus voltage.
Encoder resolution is determined to meet the application positioning accuracy: number of counts per revolution, incremental or absolute. The motor cable's wire diameter is sized for motor power.
The rest of the cables (communication and sgnals) are only configured for length.

References:

Cetin, S., Khandekar, F., Motion Control Software using CODESYS and EtherCAT, 2022.

Cetinkunt, S., Mechatronics with Experiments, John Wiley and Sons, 2012, Second Edition.

Motion Control System Overview by Yaskawa:
https://www.youtube.com/watch?v=-ZWUSbMKAfg

Servo Basics by Yaskawa:
https://www.youtube.com/watch?v=Gzo9m0tMD0A&feature=emb_rel_pause

Motor Basics by Yaskawa: https://www.youtube.com/watch?v=qpKybaLURy0

What is a Servo Motor and How it Works:
https://www.youtube.com/watch?v=ditS0a28Sko&t=658s

How a Servo Motor Works – Animation: https://www.youtube.com/watch?v=hg3TIFIxWCo

Drive Basic by Yaskawa: https://www.youtube.com/watch?v=3-cs4eEiBWo

How does an Electric Motor work? (DC Brush Type Motor)
https://www.youtube.com/watch?v=CWulQ1ZSE3c