1. Mechatronic systems

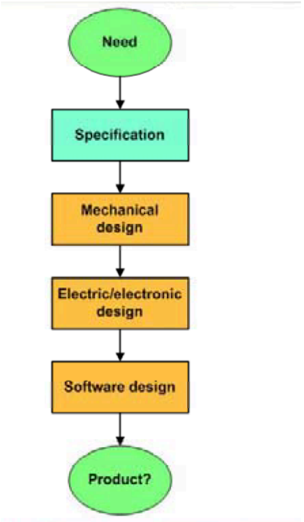
What is mechatronics?

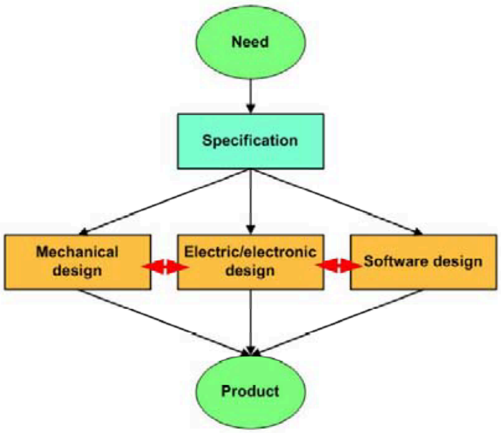
- It is a “philosophy” of design (development)

Design Process:

- Inverse Problem:
  - Often ill-posed.
  - Easier to estimate parameters than to design a device with predefined parameters.
- Heuristic-Based Decisions:
  - Design decisions rely on experience.
- Iterative Nature:
  - Design process involves iterations (design, check, redesign, check, ...).

Approaches





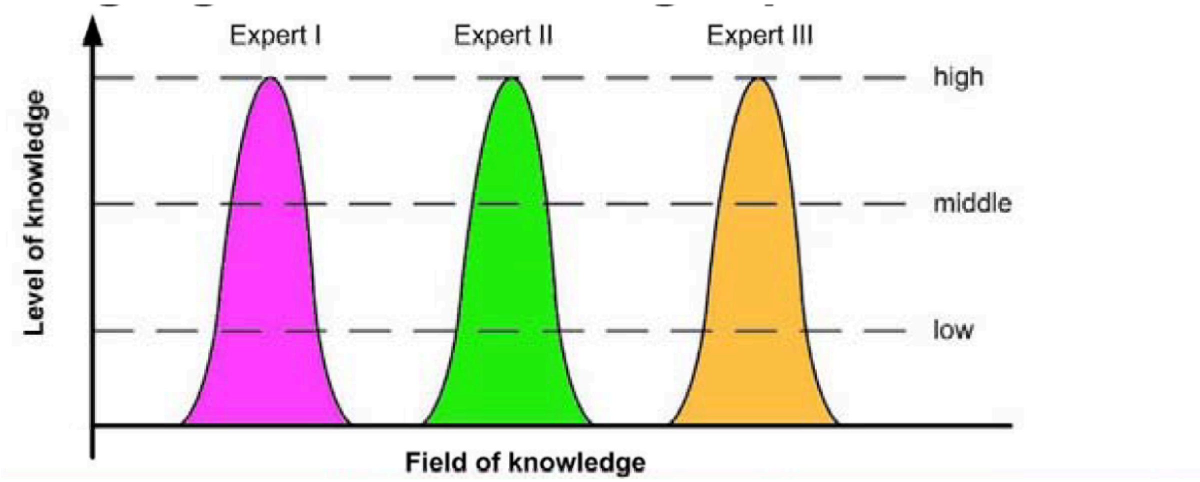
Concurrent (right)

+ Shortening of the design phase
+ No need to wait for others
+ Simplification and acceleration of implementation
+ Possibility of flexible realization of system's various function
+ When all components are designed at the same time, realization of functions can be moved easier between domains

- Needed: verification method of components compatibility: Virtual prototyping

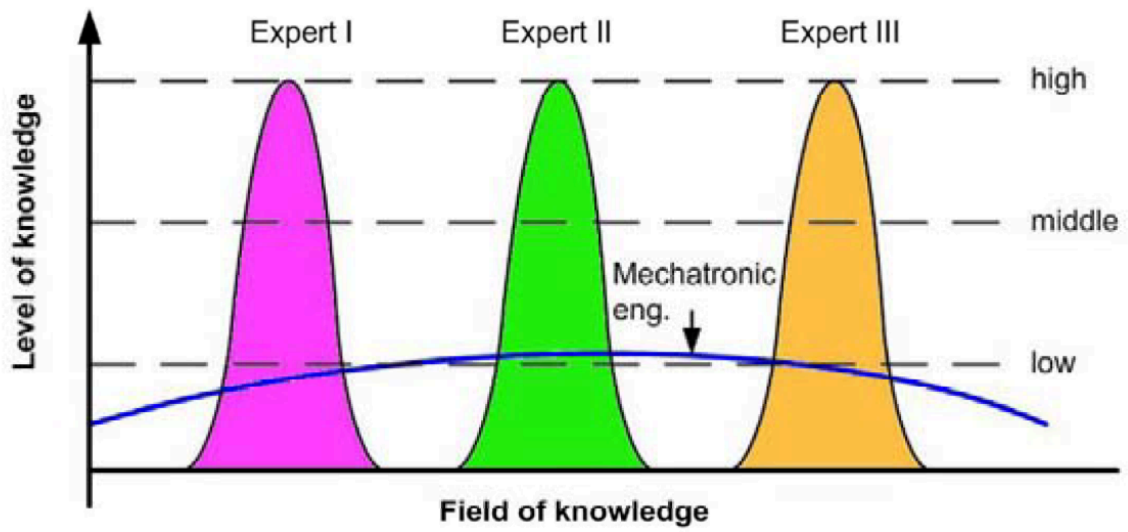
Communication in a team

- No communication everyone hermetic



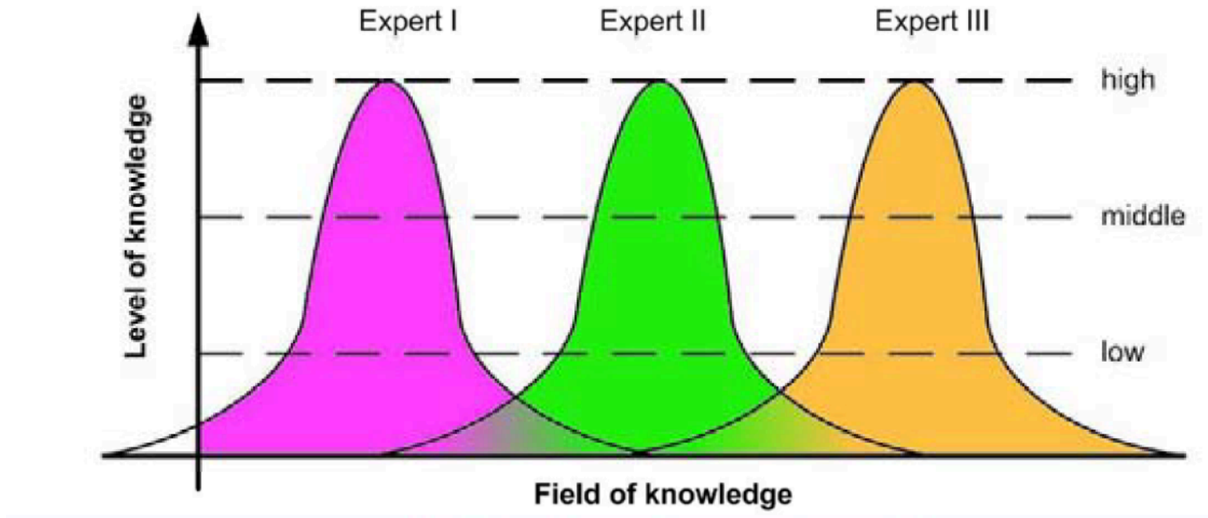
Solution I

- Replace experts by “general” mechatronic engineers
- Failure when advanced knowledge needed



Solution II

Each expert but all have interdisciplinary communication



Mechatronic Education in Japan:

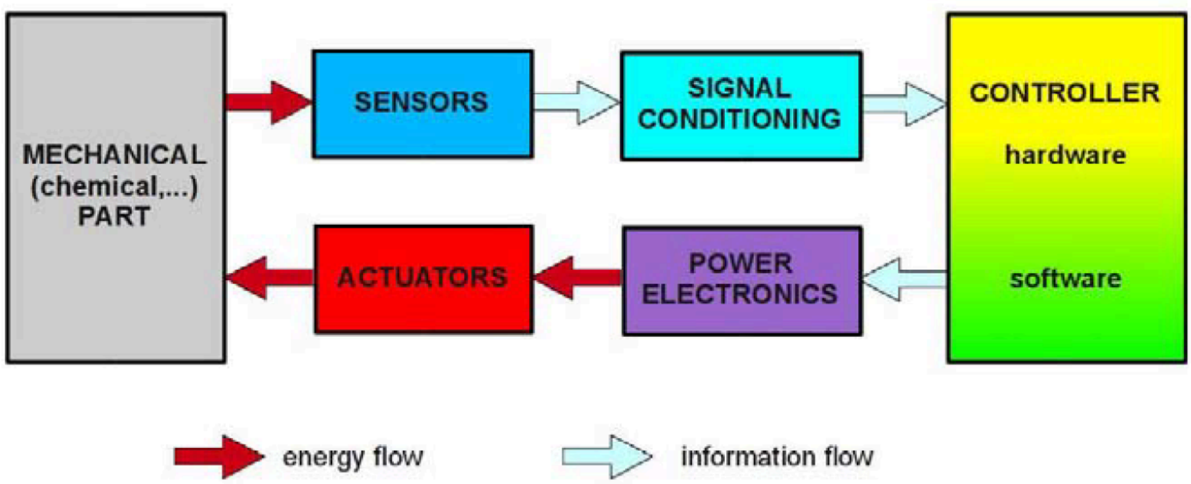
- Design engineers in large companies work on numerous projects sequentially, gaining comprehensive mechatronics expertise.
- Engineers eventually become team leaders after accumulating knowledge and experience across diverse domains.

Flexibility in Car Brakes Example:

- Classical hydraulic brakes transmit driver intentions mechanically.
- Modern ABS brakes use electrical transmission, reducing reaction time with a sensor in the pedal and an electric motor driving the shoes.

2. Structure of mechatronic systems

Typical structure of a mechatronic system



The “main” part of the system

- Mechanism: Robot manipulator
- Chemical Process: Fertilizers, petrochemicals
- Power Sources: Fuel cells, combustion engine, jet engine, power plant turbine (turbogenerator)

Sensors:
- Provide system status information
- Convert energy into data
- Outputs used for feedback, monitoring, and human-machine interface

Signal Conditioning:
- Prepares sensor output for controller
- Includes AD conversion, filtering, noise reduction, antialiasing, amplification, and electrical matching

Actuators:
- Supply energy to the system
- Can be indirect (e.g., valves)
- Convert electrical, chemical, or mechanical energy into mechanical energy

Actuator vs Motor:
Actuator: Component that produces mechanical movement or action.
Motor: Type of actuator that converts **electrical** energy into mechanical motion.

Power Electronics:
- Translate control signals for actuators
- Often supply energy for actuators in a controlled manner
- Convert information directly into electrical energy

Controller:
- Executes algorithms for signal processing, control, monitoring, and presentation
- Software implementation on hardware (microprocessor, etc.)
- Controller hardware can be an industrial computer, embedded controller, or fast prototyping hardware

3. Examples of mechatronic systems — inertial sensors

Accelerometer:

- Converts acceleration into a directly measurable quantity.
- Transforms this quantity into an electrical parameter.
- Processes the output by measuring the electrical quantity or parameter.

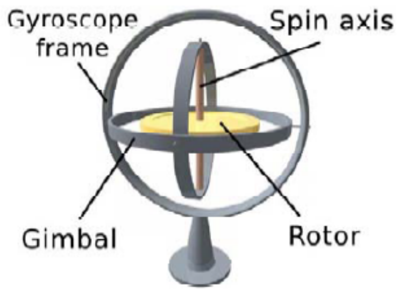
Piezoelectric Accelerometer:

- Converts acceleration into a directly measurable quantity using piezoelectric materials, which means proportional electricity to stress of the material

Gyroscope

1. Convert angular velocity into directly measurable quantity
2. Convert this quantity into electrical quantity or parameter
3. Process the output - measure the electrical quantity or parameter

Mechanical gyroscope



Sagnac Interferometer and Fiber Optic Gyroscope (FOG):

- Sagnac Interferometer splits light, creating interference fringes dependent on angular velocity.
- FOG enhances sensitivity by looping optical fiber, multiplying effective area. Coil length can be up to 5 km.

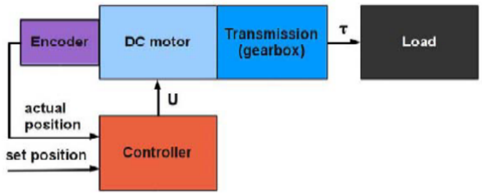
Differential system

Two transducers differentially connected for elimination of shock and linear acceleration

4. Examples of mechatronic systems - direct drives

Classical Servomechanism:

- Faces challenges at low speeds: low torque, efficiency, and potential temperature issues.
- Operates at 3000 rpm, with limited power and torque.
- Requires gearbox and brushes, leading to maintenance costs and reliability concerns.



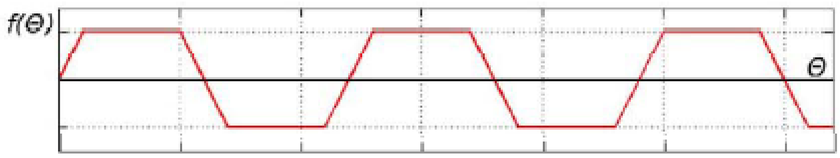
Gearbox:
Key considerations: efficiency, vibrations, maintenance costs, reliability, elasticity, sensor localization, inertia, nonlinearities, friction, backlash, and cost.

Controller:
Relatively simple and cost-effective.

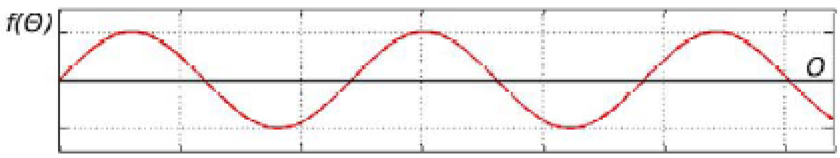
Direct Drive:
Resembles an inverted DC motor with brushes.
Uses electronic commutation for relative position control.

Magnetic Field Distribution:

a) Trapezoidal - BLDC



b) Sinusoidal - PMSM



BLDC (Brushless DC Motor) and PMSM (Permanent Magnet Synchronous Motor):
BLDC: Operates with square phase currents and back EMF at constant speed, requiring square supply voltage.

PMSM: Operates with sinusoidal phase currents and back EMF at constant speed, requiring sinusoidal supply voltage.

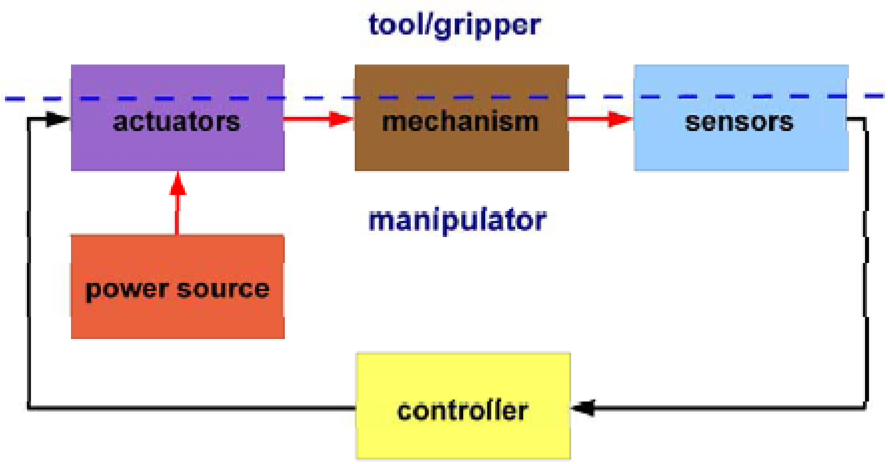
Control of Direct Drives:
Requires electronic commutator (microprocessor).
Involves high-frequency PWM and sensor for position.

BLDC Torque Ripple:
Variance in Torque output, something that should be considered

Control of PMSM:
Maintains constant angle between stator and rotor magnetic flux vectors.
Uses high-frequency PWM and Field Oriented Control (FOC) for flux vector control.

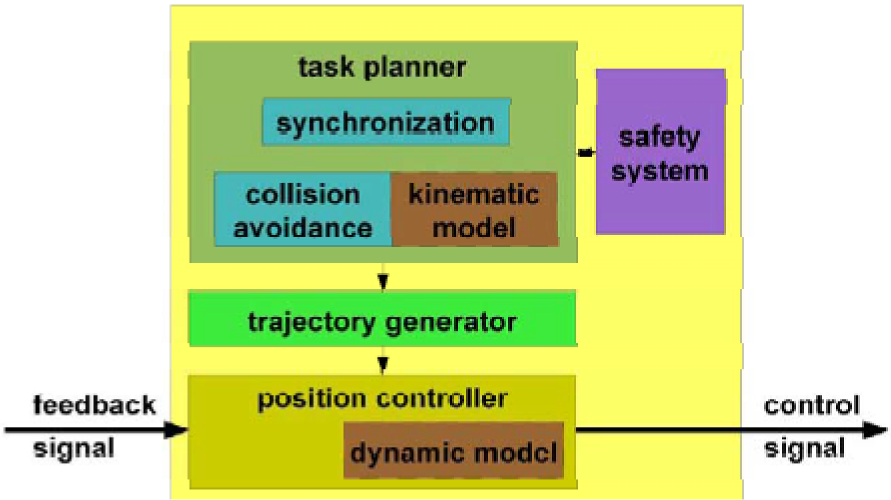
5. Examples of mechatronic systems — industrial robots

Structure of an industrial robot



Mechanism
- Enables end-effector mobility and orientation.
- Composed of links and joints.
- Kinematics:
    - Defines DOF, workspace, and transformations between local and global coordinates.
- Dynamics:
    - Analyzes mechanism movement over time.

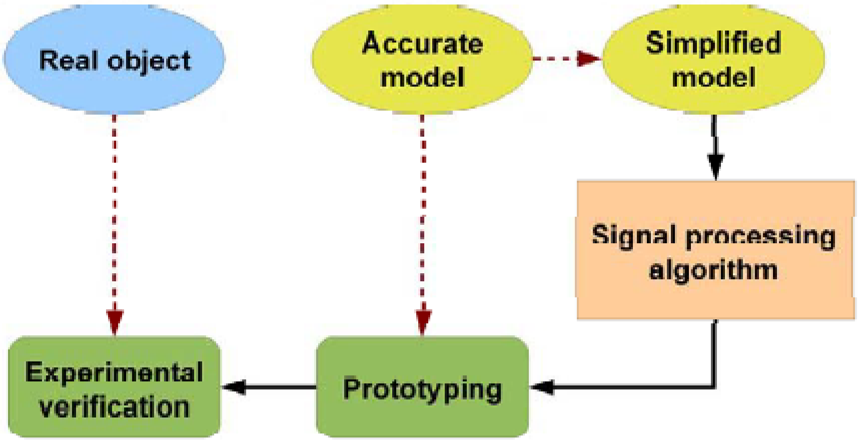
Controller



6. Techniques of mechatronic design

Commonly used techniques include virtual prototyping, real-time simulation, fast prototyping, and Hardware-in-the-Loop-Simulation (HILS).

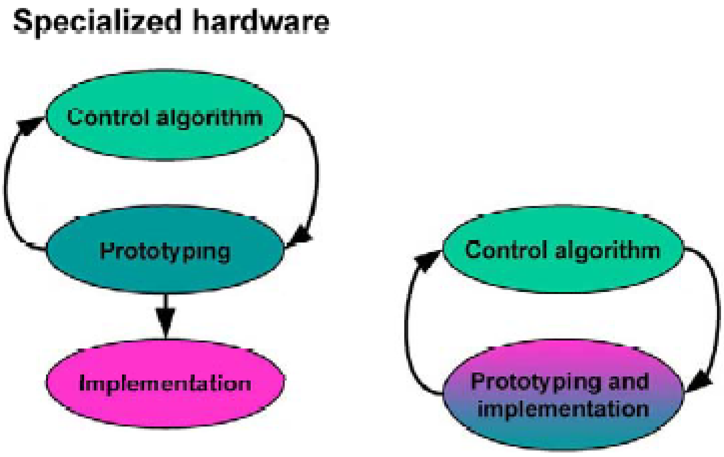
Modeling of an object



- Applied when designing a controller for a non-existing system, when experiments are hazardous, or when dealing with unstable or poorly damped systems.
- Fast prototyping is essential for complex, nonlinear systems that are hard to theoretically model.

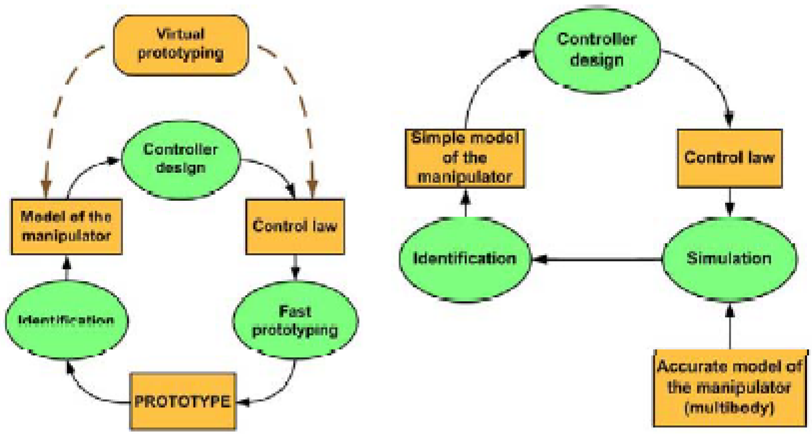
- HILS is necessary for experimental verification when a device is not physically present.
- Fast prototyping involves automatic code generation, easy parameter modification, and signal display/recording.
- Target hardware options include industrial computers for single-piece production, embedded systems for middle-lot production, and circuits based on ASIC/FPGA for large-lot production.
- Specialized hardware for fast prototyping offers flexibility and high computing power but is expensive and different from target hardware.

Specialized hardware vs Target hardware



Identification in closed-loop systems requires a balance between experimental and virtual methods.

Identification in closed-loop: experimental vs virtual



- Industrial computers provide flexibility, various peripherals, and reliability but can be expensive and suboptimal.

- Embedded systems are specialized, non-reconfigurable microprocessor systems based on general-purpose CPU, DSP, or FPGA.

7. An example of mechatronic design — a parallel robot

Why mechatronic approach?

- Each kinematic chain directly influences the end-effector, making separate analysis and design of limbs impractical.
- Closed kinematic chains enhance robot properties but complicate kinematic equations and control synthesis.
- Modifying element shapes changes kinematic equations.
- Actuators, often serving as structural members, impact manipulator dynamics, with mechanical properties influencing both construction and actuator design.

8. Mechatronic trends - all in one:

System Trends:

- Shift from energy to information transmission.
- Focus on digital information for "smart" sensors and actuators.
- Optimize energy supply closer to the main system.
- Transition functions from hardware to software.

Device Trends:

- Price reduction, added functionalities.
- Software dominance for functions and quality.
- Substitution of mechanical with electronic components.
- Growing demand for individualized devices.
- Increased role of Human-Machine Interface (HMI).
- Stricter environmental regulations.
- Rise in microsystems usage.