AGH University of Science and Technology

Mechatronic systems

Maciej Petko

Department of Robotics and Mechatronics
D1-405, tel. 36-69
petko@agh.edu.pl

Mechatronics

What is mechatrinics?

Tomizuka:
The best practice of SYNTHESIS for engineers in various domains

It is a "philosophy" of design (development)

Each design decision should take into account all aspects of the device being designed

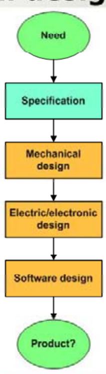
Design process

- **Inverse problem**
- **Often ill-posed**
- **Usually it is easier to estimate parameters of a designed device than to design a device with given parameters**
  - **Design decisions are based on heuristics (experience)**
  - **Design process is iterative (design, check, redesign, check,...)**

Design of mechatronic system

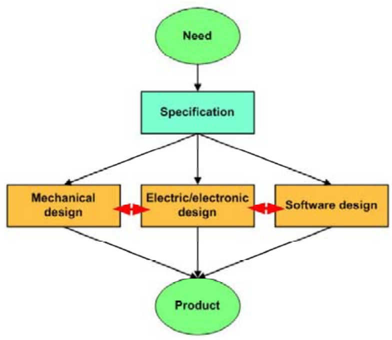
- **Mechatronic system is usually to complex to be designed by a single person**
- **Large group of experts in various domains does not automatically guarantee a success**

Traditional approach – sequential design



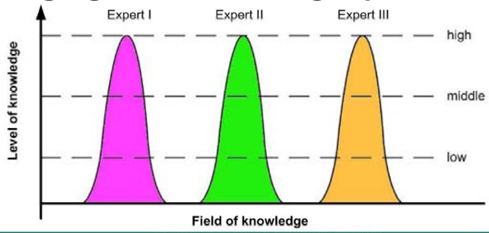
Mechatronic approach – concurrent design

- **Teamwork**
  - **Design experts**
  - **Mnfg. experts**
  - **Marketing exp.**
  - **Management**
  - **...**
- **Needed: verification method of components compatibility**
  - **Virtual prototyping**



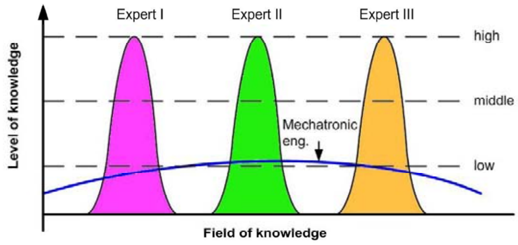
Communication in a team

- Engineer must be not only expert in his domain, but must be able to communicate with others
- Otherwise each talk in his own hermetic language – understanding impossible



Solution I

- Include into the team mechatronic engineers of wide but shallow knowledge

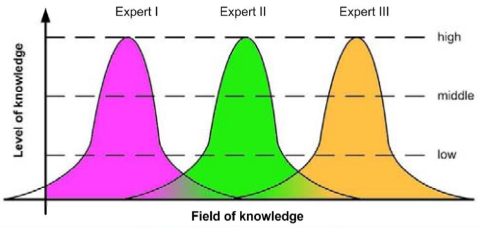


Solution I (contd.)

- Most of mechatronic courses at universities based on this idea
- Result: tendency to replace in teams “narrow” experts with deep knowledge by larger number of “general” mechatronic engineers
  - Savings expected
  - Projects failed at the first problem requiring advanced knowledge and experience in a particular domain

Solution II

- Education of specialists in traditional fields (mechanics, electronics, control,...) expanded by foundations of other domains and an interdisciplinary communication



Mechatronic education in Japan

- In a big company each design engineer is assigned to a large number of different projects one after another as a full member
  - Gains knowledge and experience in all domains of mechatronics
  - Eventually becomes a team leader
- Two conditions must be satisfied:
  - The company big enough to have different projects
  - The design engineer works long enough for the company to gain experience needed in this company (possible only in Japan)

Learning of mechatronics

- Mechatronics cannot be learned on a theoretical way but by
- participating
- in realization
- of complete projects
- ending with implementation,
- which is expensive

Benefits of concurrent, mechatronic design

- Shortening of the design phase
  - No need to wait for others
- Simplification and acceleration of implementation
  - Careful verification (virtual prototyping) → often the first prototype fulfills requirements and can be sold
- 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

Example of flexibility – breaks in a car

- In a classical hydraulic break, the intention of a driver, expressed by pressing the pedal, is transmitted to the brake shoes on a mechanical (hydraulic) way
- In modern, ABS breaks it is transmitted electrically, by a sensor in the pedal and electric motor driving the shoes
  - Application of a μP with a wheel movement sensor allows for improvement of the function by avoiding wheel blocking and shortening of the break system reaction time

Trends in mechatronic devices

- Decreasing of price/adding new functionalities
- Domination of software in realization of functions and providing quality
- Substitution of mechanical components by electric and electronic ones
- Demand for individualized devices
- Increasing role of HMI as the main differentiating factor
- More rigorous environment regulations
- Increasing number of microsystems as components

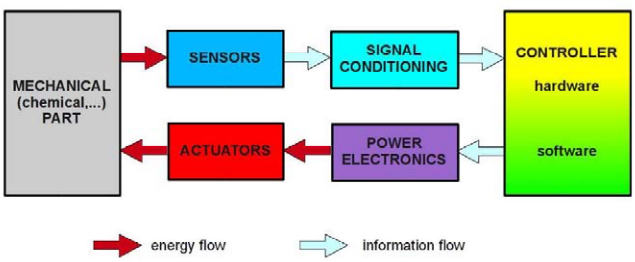
AGH University of Science and Technology

Structure of mechatronic systems

Maciej Petko

Department of Robotics and Mechatronics
D1-405, tel. 36-69
petko@agh.edu.pl

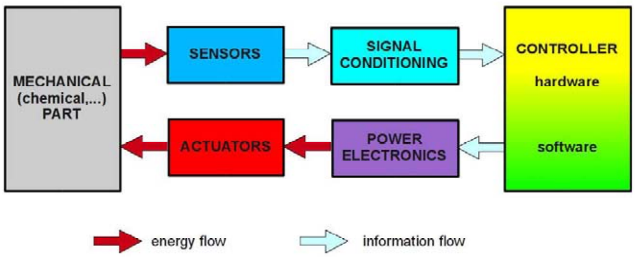
Typical structure of a mechatronic system



The "main" part of the system

- Mechanism
  - Robot manipulator
- Chemical
  - Process (fertilizers, petrochemical,...)
  - Fuel cells
- Complex
  - Combustion engine
  - Jet engine
  - Power plant turbine (turbogenerator)

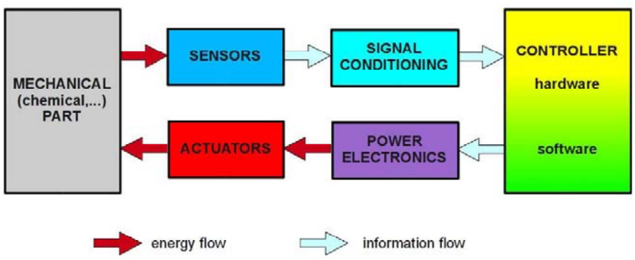
Typical structure of a mechatronic system



Sensors

- Provide information about the state of the "main" part of the system
- Converts energy into information
- Sensors output can be used as:
  - Feedback signal for control
  - Input signal for monitoring/diagnostic systems (eg. machine health)
  - Input signals in human-machine interface (HMI)
    - Sensing intentions of an operator

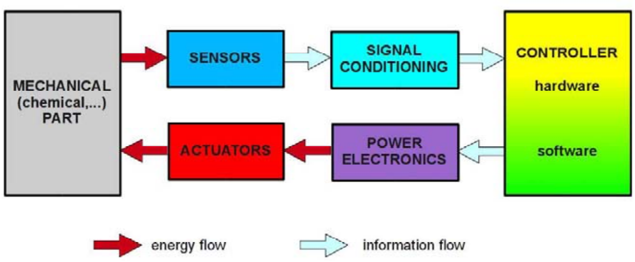
Typical structure of a mechatronic system



Signal conditioning

- Conversion to a form readable by the controller
  - eg. AD conversion
- Filtering
  - Noise
  - Unneeded information
  - Antialiasing
- Amplification
- Electrical matching
  - Voltage levels
  - Impedance

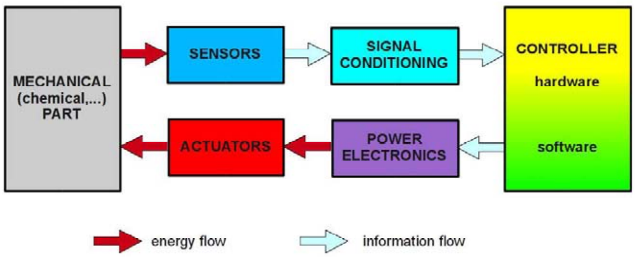
Typical structure of a mechatronic system



Actuators

- Supply energy in appropriate form for the "main" part of the system
  - Sometimes indirectly (eg. a valve)
- Can be used in HMI
- Converts energy
- From:
  - Electrical
  - Chemical
  - Mechanical
- Into:
  - Mechanical
  - ...

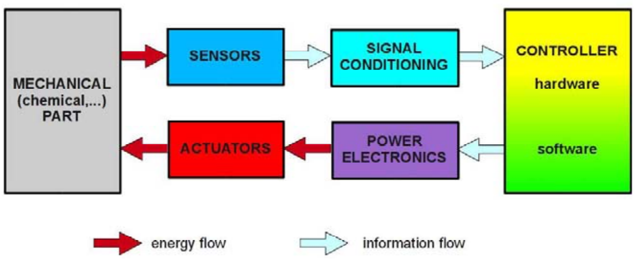
Typical structure of a mechatronic system



Power electronics

- **Translates control signals for actuators**
- **Usually supply energy for actuators**
  - **In a controlled way**
  - **Sometimes indirectly (eg. ignition in a combustion engine)**
- **Converts information into energy**
  - **Directly into electrical energy**

Typical structure of a mechatronic system



Controller

- **Executes algorithms (processes information):**
  - **Signal processing**
    - **Control**
    - **Monitoring (state estimation)**
    - **Presentation (display)**
  - **Predefined (unconditional) actions**
    - **Open-loop control**

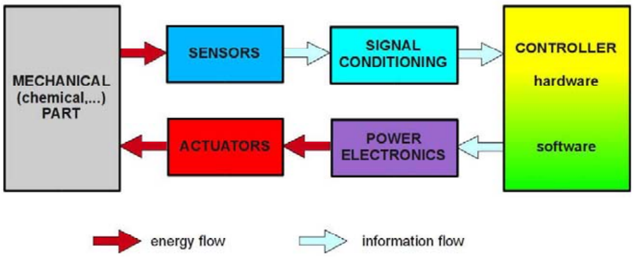
Controller

- **Software**
  - **Implementation of algorithms**
- **Hardware**
  - **For executing software**
    - **Then microprocessor(s) needed**
  - **For implementing parts of or entire algorithms**
    - **When entire algorithm then software (and microprocessor) not needed**

Controller hardware

- **Industrial computer**
  - **Modular**
  - **Operating system (OS)**
    - **Real-time (RTOS)**
- **Embedded controller**
  - **Microprocessor, microcontroller, DSP**
  - **FPGA, ASIC**
- **Fast prototyping hardware**
  - **Flexible, powerful (computationally)**
  - **Expensive**
  - **Different from target hardware**

Typical structure of a mechatronic system



Trends

- Substitute energy transmission by information transmission
  - Preferably information in digital form
    - “smart” sensors, “smart” actuators
- Supply and convert energy as close as possible to the “main” part of the system
  - eg. direct drives
- Move functions of the system from hardware to software
  - Cheaper
  - More flexible (customization)
  - More and more feasible

AGH University of Science and Technology

Examples of mechatronic systems – inertial sensors

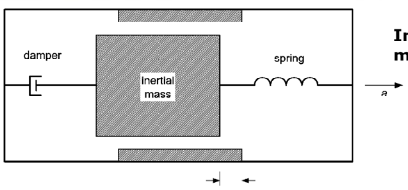
Maciej Petko

Department of Robotics and Mechatronics
D1-405, tel. 36-69
petko@agh.edu.pl

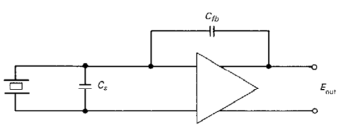
Accelerometer

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

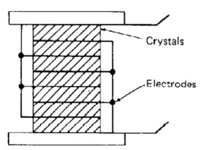
Piezoelectric accelerometer



Inertial (seismic) mass transducer

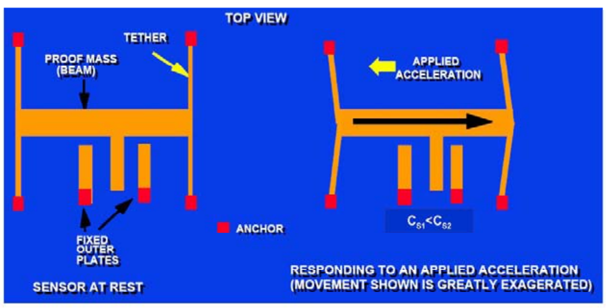


Charge amplifier

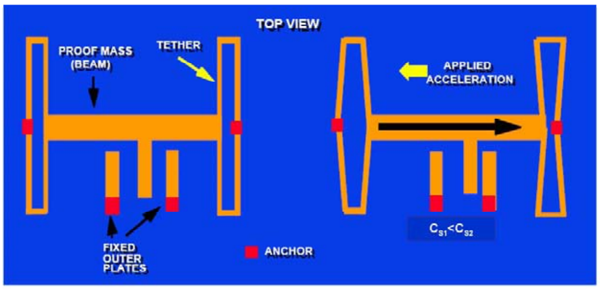


Stacked piezoelectric

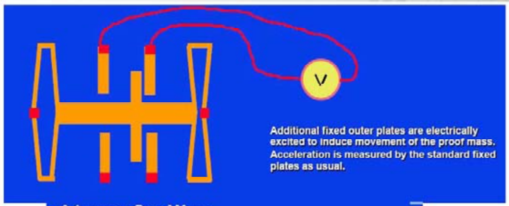
MEMS capacitive accelerometer – principle of operation



Improved version

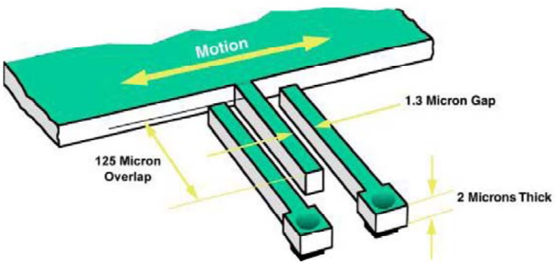


Accelerometer testing

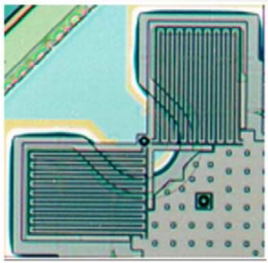


- 0.1μgrams Proof Mass
- 0.1pF per Side for the Differential Capacitor
- 20aF (10^{-18}f) Smallest Detectable Capacitance Change
- Total Capacitance Change for Full-scale is 10fF
- 1.3μm Gaps Between Capacitor Plates
- 0.2Å Minimum Detectable Beam Deflection (one tenth of an Atomic diameter)
- 1.6 μm Between the Suspended Beam and Substrate
- 10 to 22kHz Resonant Frequency of Beam

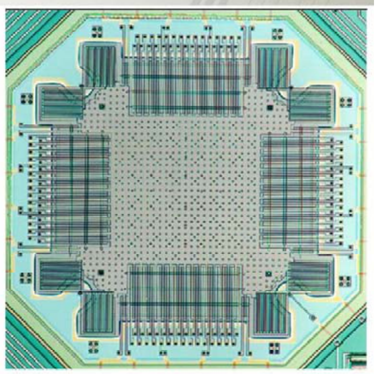
Single finger of the capacitor



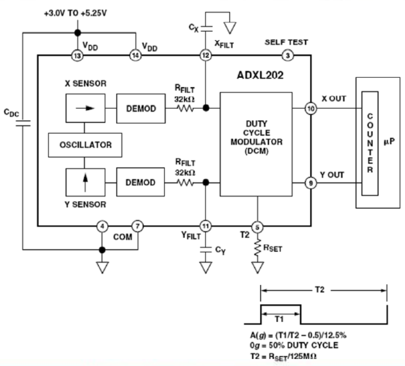
Entire mechanical part of ADXL202



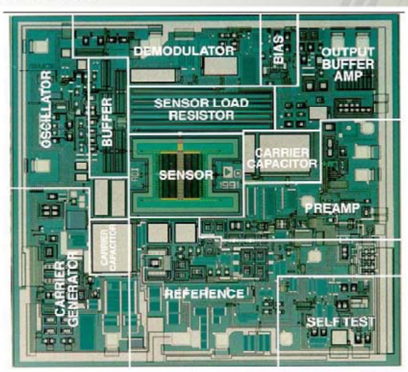
Spring in a corner of the structure



Sensor's electronics



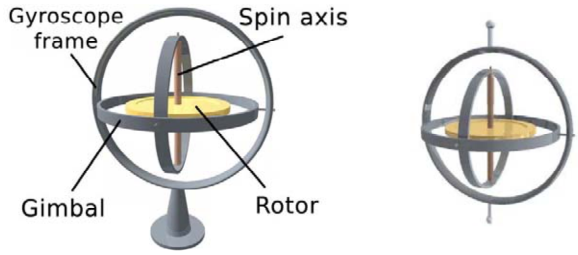
Accelerometer ADXL-50 Analog Devices



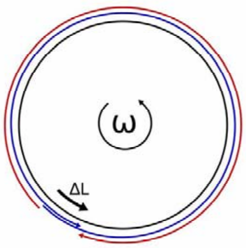
Gyroscope

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

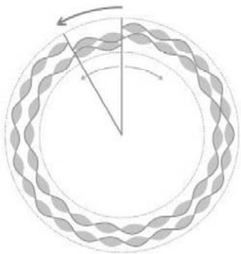
Mechanical gyroscope



Optical gyroscopes – based on Sagnac effect

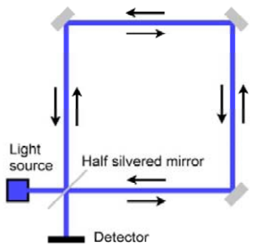


Light traveling opposite directions go different distances before reaching the moving source again



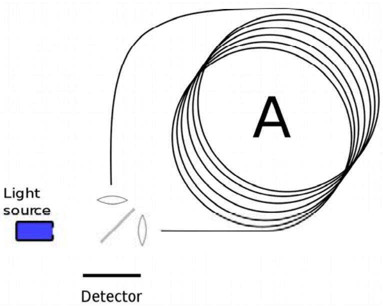
Schematic representation of the frequency shift when a ring laser interferometer is rotating

Sagnac interferometer



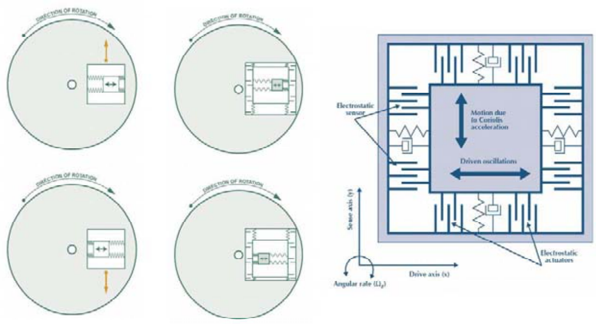
- A beam of light is split and the two beams follow a trajectory in opposite directions
- In the detector an interference pattern is obtained
- The position of the interference fringes is dependent on the angular velocity of the setup

Fibre optic gyroscope (FOG)



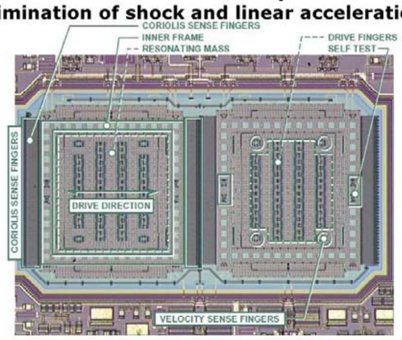
The interference on a Sagnac interferometer is proportional to the enclosed area. A looped fibre-optic coil multiplies the effective area by the number of loops. A coil of optical fibre can be as long as 5 km

MEMS gyroscope – principle of operation

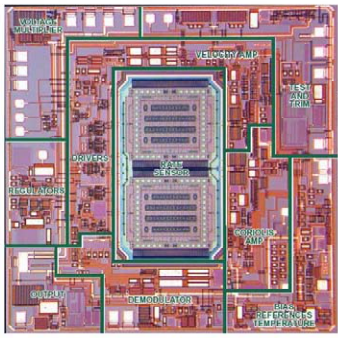


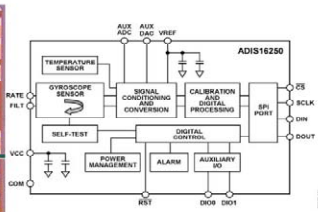
Differential system

Two transducers differentially connected for elimination of shock and linear acceleration



MEMS and signal conditioning on a single piece of silicone





- Synergy?
- Added value?

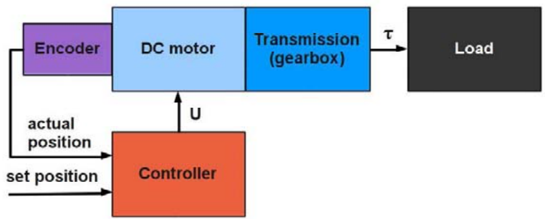
AGH University of Science and Technology

Examples of mechatronic systems – direct drives

Maciej Petko

Department of Robotics and Mechatronics
D1-405, tel. 36-69
petko@agh.edu.pl

Classical servomechanism



Permanent magnet DC motor

- Cannot operate at low speeds
  - Low torque
  - Low efficiency → temperature (rotor)
  - Nominal speed ~3000rpm
- Restricted power and torque
- Gearbox needed
- Brushes
  - Maintenance costs
  - Reliability

Gearbox

- Efficiency
- Vibrations
- Maintenance costs
- Reliability
- Elasticity
  - Sensor localization problem
- Large additional inertia
- Hard nonlinearities
  - Friction
  - Backlash
- Cost

Controller

- Comparatively simple
- Comparatively cheap

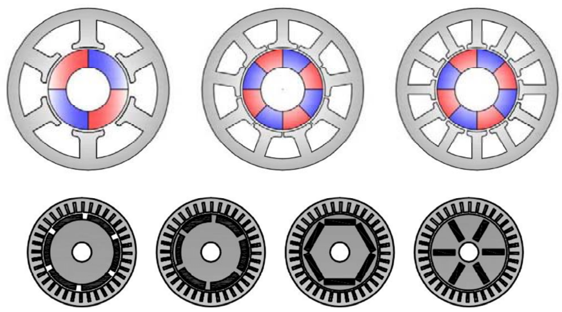
List of wishes

- High constant torque at low speed (at rest)
- High stiffness
- Low moment of inertia
- Low weight
- Compact
- High reliability
- Low maintenance costs

Direct drive

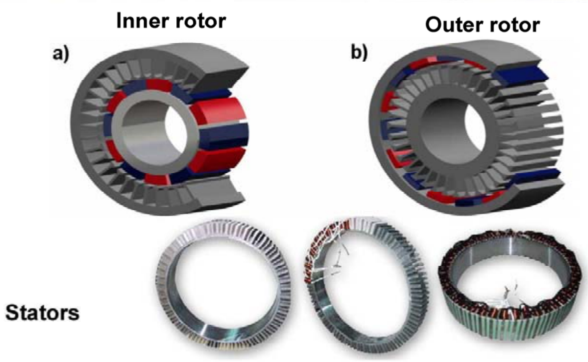
- Mechanical structure – inverted construction of a DC motor with brushes
- Rotor – multipole permanent magnet
- Stator – multiple windings connected in a delta (series) or wye "Y" ("star" - parallel) configuration
- Electronic commutation – the controller regulates relative position of two magnetic field vectors (rotor and stator)

Cross-section of a direct drive

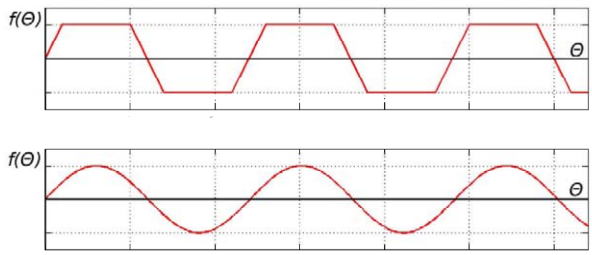


Placement of permanent magnets in a rotor

Configurations of direct drives



Magnetic field distribution in the slot between stator and rotor



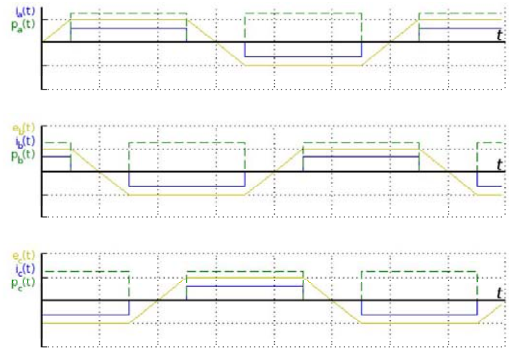
- Trapezoidal – BLDC
- Sinusoidal – PMSM

BLDC

Brushless Direct Current motor

- **At constant speed square phase currents**
- **At constant speed square back EMF**
- **Requires square supply voltage**

BLDC at constant speed

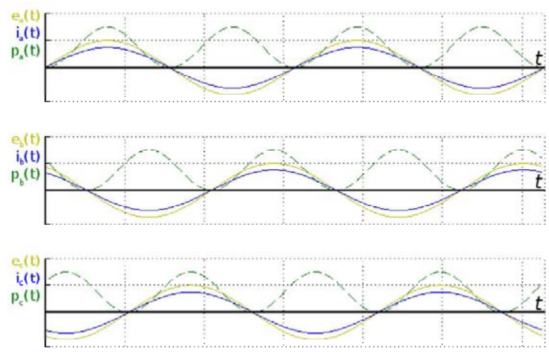


PMSM

Permanent Magnet Synchronous Motor

- **At constant speed sinusoidal phase currents**
- **At constant speed sinusoidal back EMF**
- **Requires sinusoidal supply voltage at constant speed**

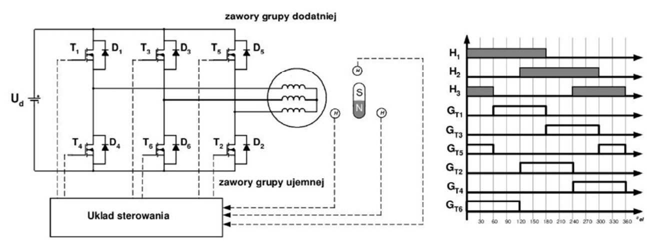
PMSM at constant speed



Control of direct drives

- **Requires "electronic commutator", usually microprocessor controller that switches stator coils**
  - **Current controller – high frequency PWM**
  - **Commutator coupled with a rotor (magnetic field) angular position sensor**
- **Often both functions in a single circuit – three-phase inverter with MOSFETs or IGBTs**

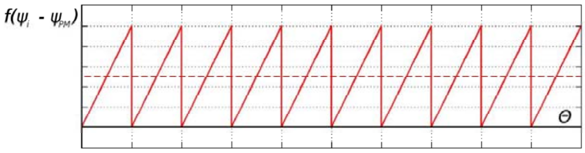
Control of BLDC



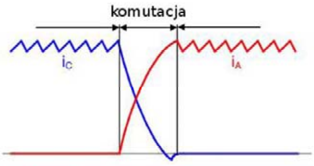
H – Hall sensors to switch transistors at each 60 electrical degrees
At each instant two transistors are on: one top, one bottom

BLDC torque ripple

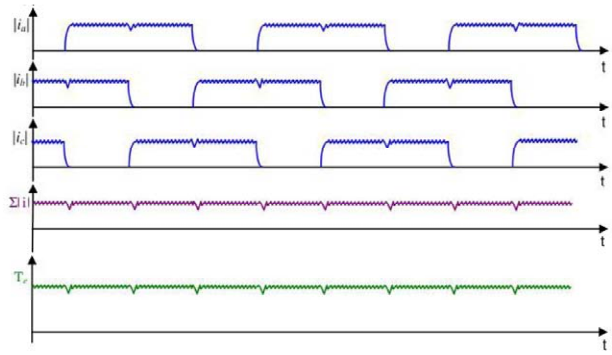
Stator magnetic field can have only several fixed orientations



Due to inductance, current in stator coils does not vanish immediately after switching off



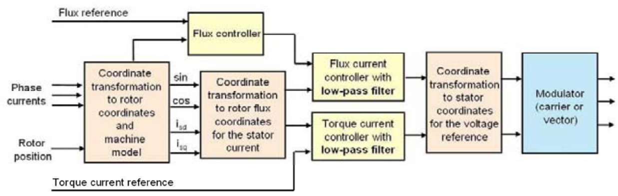
Effect of switching



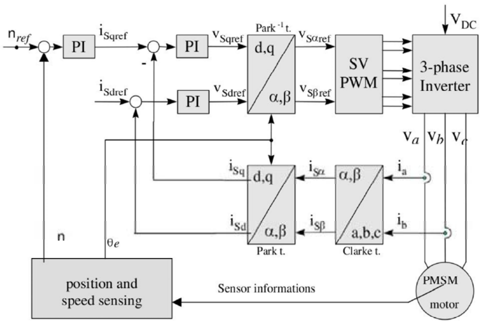
Control of PMSM

- **To maintain constant angle between stator and rotor magnetic flux vectors**
- **High frequency PWM – average currents in coils changes sinusoidally (at constant speed)**
- **Field Oriented Control (FOC)**
  - **Amplitude, frequency and phase of magnetic flux vectors**
  - **In transient and steady state**

Field Oriented Control



Field Oriented Control



d,q – rotor flux coordinates; α,β – stator coordinates

List of wishes

- **High constant torque at low speed (at rest)**
- **High stiffness**
- **Low moment of inertia**
- **Low weight**
- **Compact**
- **High reliability**
- **Low maintenance costs**

Mechatronic aspects

- Design?
- Synergy?
- Added value?

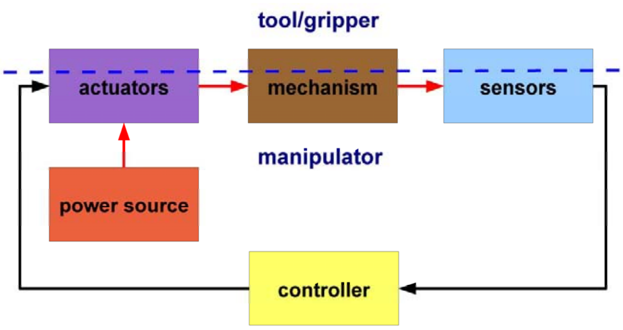
AGH University of Science and Technology

Examples of mechatronic systems – industrial robots

Maciej Petko

Department of Robotics and Mechatronics
D1-405, tel. 36-69
petko@agh.edu.pl

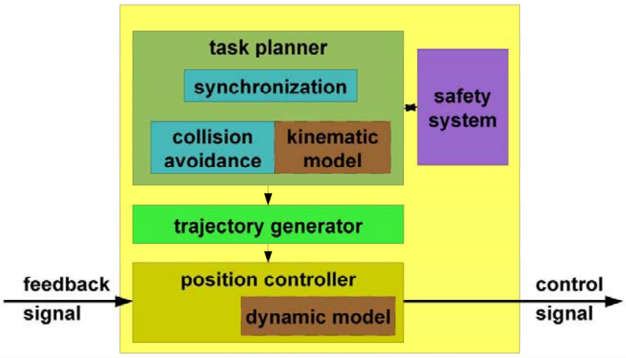
Structure of an industrial robot



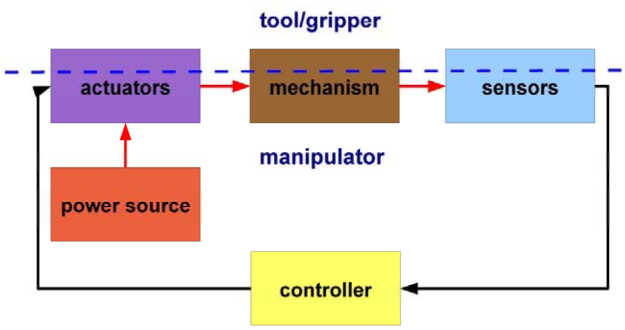
Mechanism

- Provides end-effector mobility and orientation in required workspace with required redundancy (DOF) in required time
- Consists of links and joints
- Kinematics
  - DOF, workspace, singular positions
  - transformation between joint (local) and global (usually Cartesian) coordinates
- Dynamics
  - to analyze mechanism movement in time

Controller



Structure of an industrial robot



AGH University of Science and Technology

Techniques of mechatronic design

Maciej Petko

Department of Robotics and Mechatronics
D1-405, tel. 36-69
petko@agh.edu.pl

Techniques of mechatronic design

- Design procedure depends on actual device, but there are commonly used techniques:
  - Virtual prototyping
  - Real-time simulation
    - Fast prototyping
    - Hardware-in-the-Loop-Simulation (HILS)
  - (fast prototyping on a target platform)
    - because of problems with implementation of signal processing algorithms
  - Implementation of control and signal processing algorithms

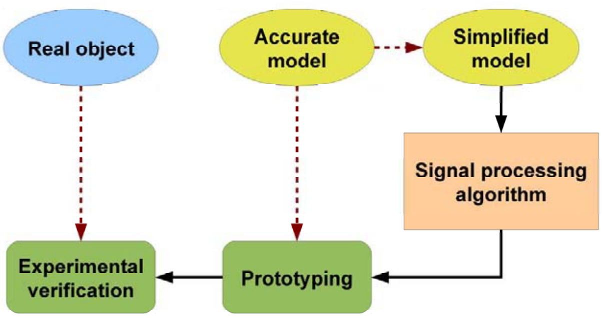
When those techniques are necessary?

- A controller (monitoring system) is design for a physically not yet existing system or experiments are dangerous
  - Virtual prototyping
- System is unstable or poorly damped → cannot be experimentally tested without a stabilizing closed-loop controller that cannot be selected through an experiment
  - Virtual prototyping + fast prototyping (for fine tuning)

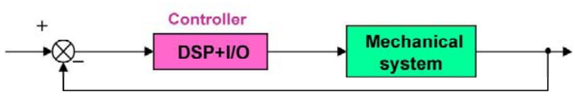
When those techniques are necessary? (contd.)

- System is complex, nonlinear, hard to theoretical modeling → experimental identification, sometimes iterative
  - Time consuming without fast prototyping
- Control, monitoring or other signal processing algorithm is implemented when the device is not yet existing physically
  - Experimental verification possible only through HILS

Modeling of an object

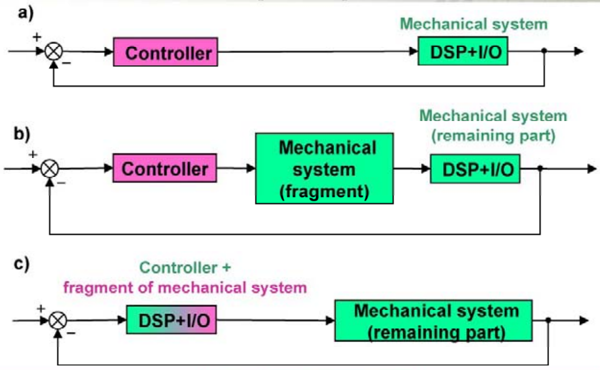


Fast prototyping

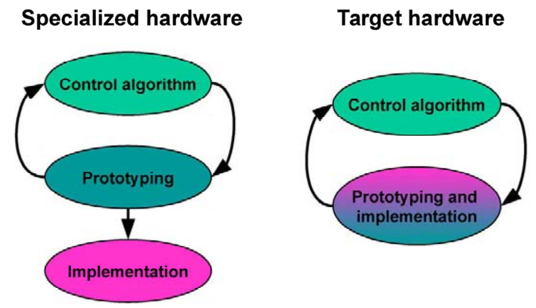


- Automatic code generation and compilation from a diagram
- Easy modification of parameters
- Display and recording of signals

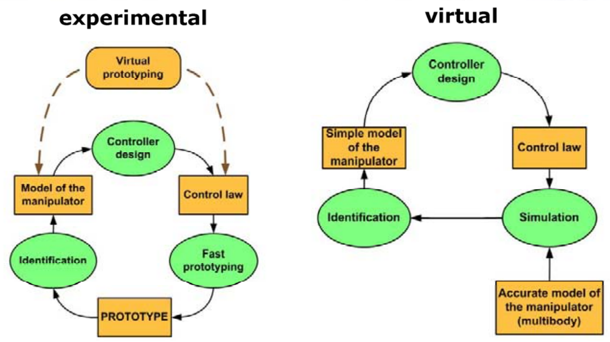
Hardware-in-the-Loop Simulation (HILS)



Fast prototyping on a target hardware platform



Identification in closed-loop



Hardware platform form implementation of algorithms

- Industrial computer
  - Single piece production
- Embedded system
  - Middle lot production
- Circuit based on ASIC/FPGA
  - Large lot production

- Specialized systems for fast prototyping
  - Flexible, large computing power
  - Expensive
  - Different from target hardware

Industrial computers

- **Easy and flexible configuration**
  - **Variety of peripherals available**
- **High reliability in industrial conditions**
- **Sufficient computing power**
- **Use of recognized software standards**
  - **(RT)OS**
  - **Reliability**
  - **Efficient and well-tested compilers**
  - **Availability of tools**
- **Expensive**
- **Suboptimal**

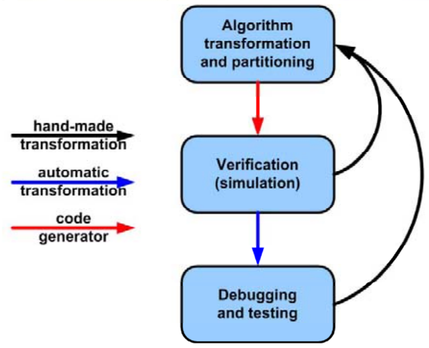
Embedded systems

- **Specialized, complete, non-reconfigurable microprocessor system**
- **Based on:**
  - **general-purpose CPU**
    - **Irregular, especially floating-point algorithms**
  - **DSP**
    - **Regular, simple algorithms fast processing large amounts of data**
  - **FPGA**

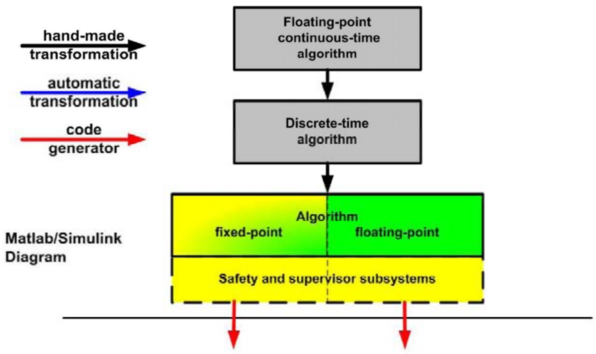
Transformations of algorithms being implemented

- **Time discretization**
  - **Always**
- **Amplitude quantization**
  - **Hardware implementation, recommended always**
- **Use of fixed-point arithmetic**
  - **Hardware implementation, recommended always**
- **Coding in HDL**
  - **Hardware implementation**

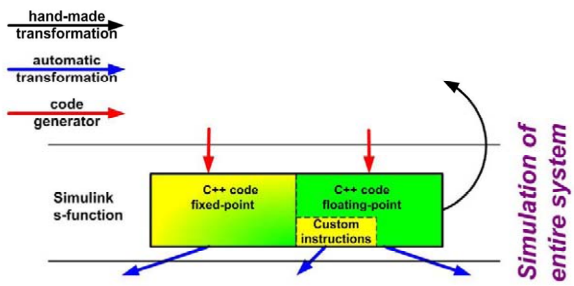
Mixed, hardware-software algorithm implementation



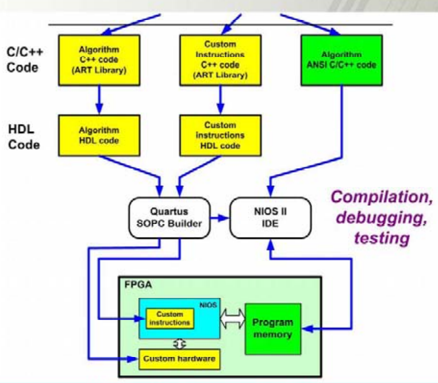
Algorithm transformation and partitioning



Verification (simulation)



Debugging and testing



AGH University of Science and Technology

An example of mechatronic design – a parallel robot

Maciej Petko

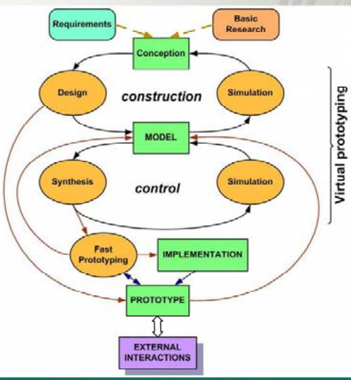
Department of Robotics and Mechatronics
D1-405, tel. 36-69
petko@agh.edu.pl

Parallel robot for milling

Why mechatronic approach?

- Each kinematic chain acts directly on a platform (end-effector) – analysis and design of individual limbs separately is impossible
- Closed kinematic chains improve properties (increase stiffness) of a robot, but complicate kinematic equations and control synthesis
- Modification of elements shape alternates character of kinematics equations
- Actuators usually play also a role of structural members of a manipulator and their mechanical properties influence dynamics (actuators depend on the construction and construction depends on actuators)

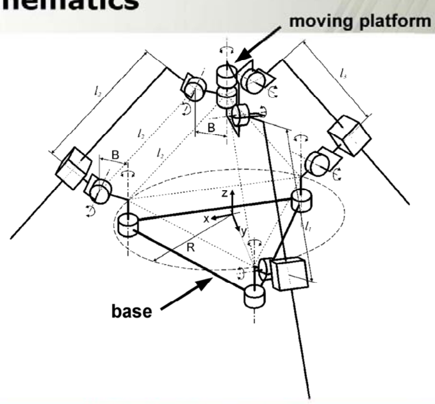
The general procedure



Assumptions

- 3 translational DOF
- Workspace Φ300 mm, h=300 mm
- Movement resolution 1 μm
- Accuracy best possible, target 10 μm
- HSM milling
- Milling force 50 N

Kinematics



structure 3RRPRR

Kinematics

Inverse problem

$$l_1 = \sqrt{x^2 + (y - R)^2 + z^2}$$

$$l_2 = \sqrt{\left(x - \frac{\sqrt{3}}{2}R\right)^2 + \left(y + \frac{1}{2}R\right)^2 + z^2}$$

$$l_2 = \sqrt{\left(x + \frac{\sqrt{3}}{2}R\right)^2 + \left(y + \frac{1}{2}R\right)^2 + z^2}$$

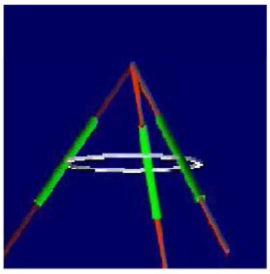
Forward problem

$$x = \frac{\sqrt{3}(l_2^2 - l_3^2)}{6R}$$

$$y = \frac{2l_1^2 - l_3^2 - l_2^2)}{6R}$$

$$z = \sqrt{l_1^2 - \frac{(l_2^2 - l_3^2)^2}{12R^2} - \left(\frac{2l_1^2 - l_3^2 - l_2^2}{6R} + R\right)^2}$$

Preliminary modeling of dynamics



Cutting process

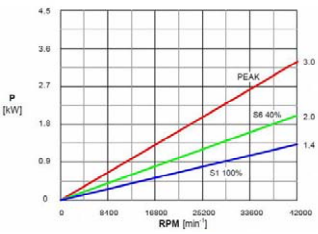
Main parameters:

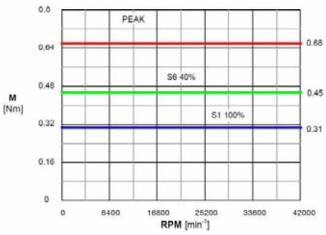
- **Instantaneous velocity of cutting edge – depends on machined material and tool (milling bit)**
  - **for aluminum and Φ5mm milling bit with 2 blades it gives 30-40 000 rpm spindle**
- **Feed per cutting edge – depends on the material and required quality of machined surface**
  - **for aluminum and 40 000 rpm it gives 1-2 m/min. feed rate**
- **Power – depends on volume removing rate (depth of milling)**
- **Torque – depends on the material, tool rotating speed, diameter and number of cutting edges, and feed per cutting edge**

Tool – HSM electrospindle

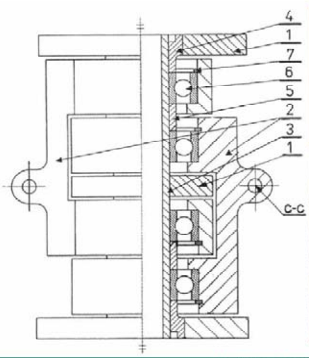
- **Integrated synchronous AC motor driven by a frequency inverter – up to 42'000 rpm**
- **3.0 kW/0.68 Nm peak, 2.0 kW/0.45 Nm @ S6 (40% Duty Cycle), 1.4 kW/0.31 Nm @ S1 (100% Duty Cycle)**
- **Mounting: standard taper shank DIN 69 871 – ISO 40**
- **Mass 1.4 kg**

IBAG HFK 95 S 40 CP spindle

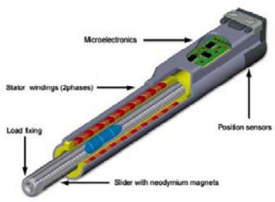




Moving platform

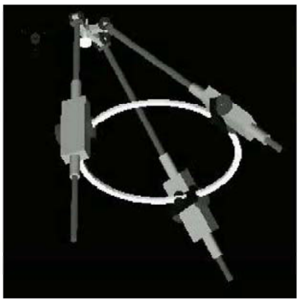


Actuators

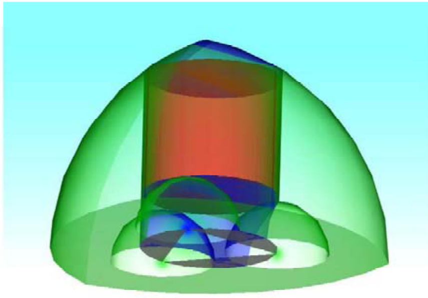


- **Linear BLDC motor**
- **Max. force 204 N, max accel. 147 m/s2, max velocity 2.2 m/s**

Preliminary modeling of dynamics II

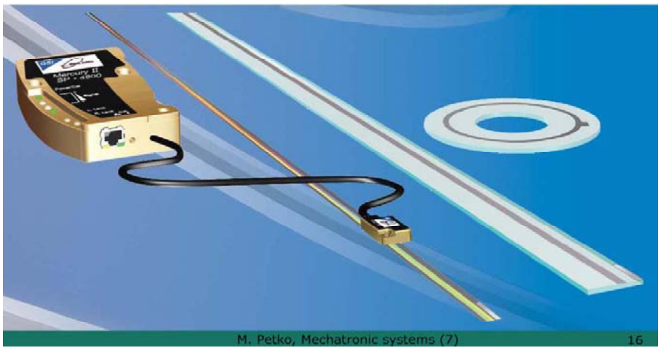


Workspace



Sensors

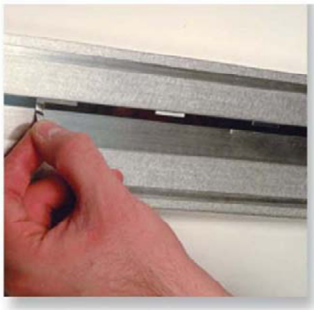
Mercury II reflective incremental encoder



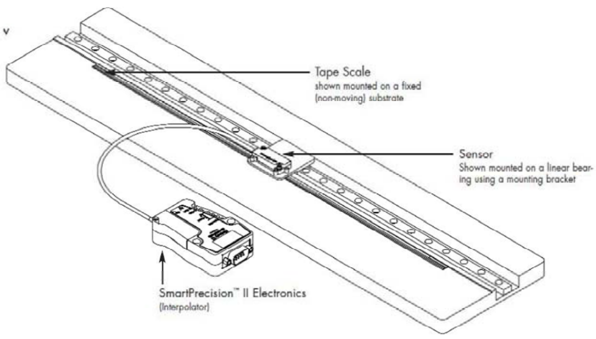
Mercury II

- **Resolution: linear: 5μm to 1.22nm, rotary: 20k to 268M CPR**
- **Accuracy: tape scale: ± 5μm/m, glass scales: linear: ± 1μm, rotary: up to ± 2.1 arc-sec**
- **Outputs: A-quad-B, Index Pulse, Dual Limits, and Alarm**
- **Same sensor for tape or glass, linear or rotary**
- **In robot used with tape scale and resolution 0.25 μm**

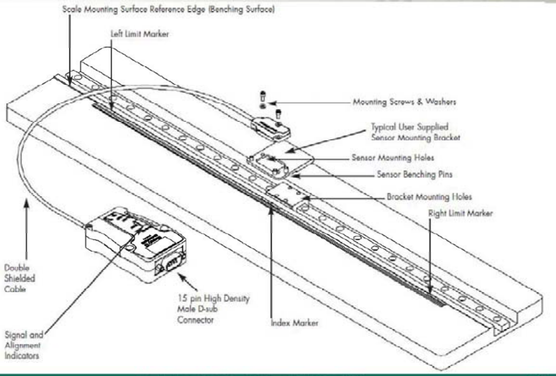
Index and limit markers



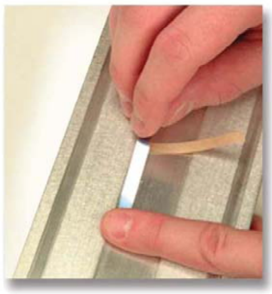
Mercury II



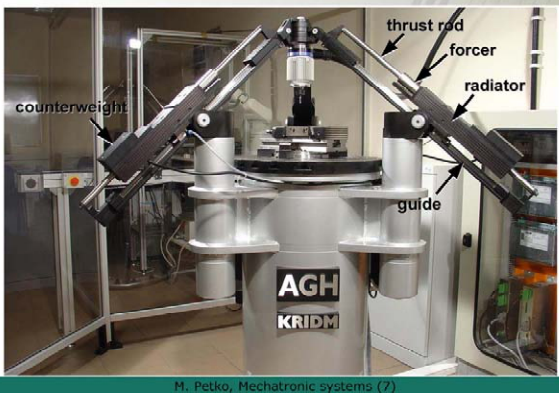
Mercury II



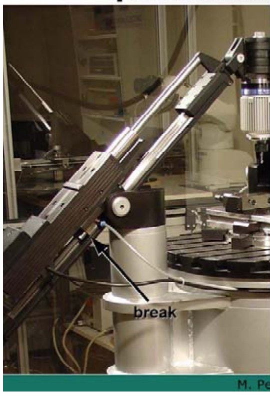
Tape scale

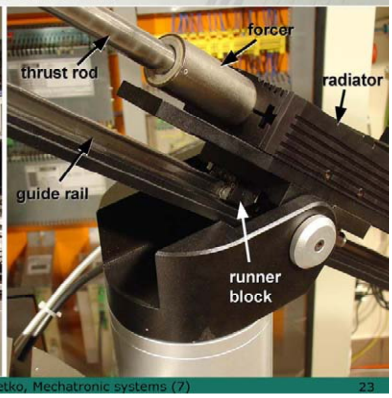


Limb construction

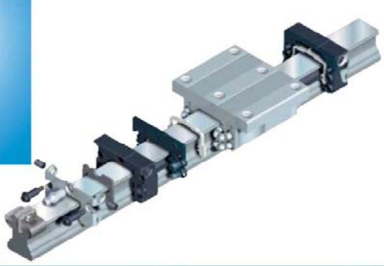


Guide rails, runner blocks, pneumatic breaks

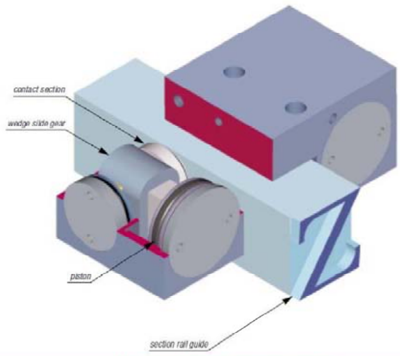




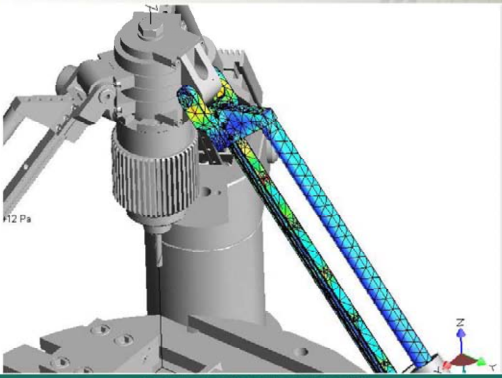
Guides



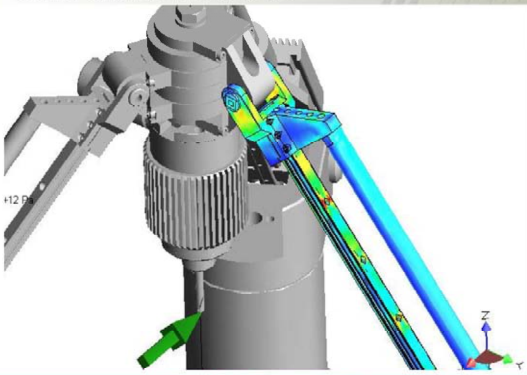
Pneumatic breaks



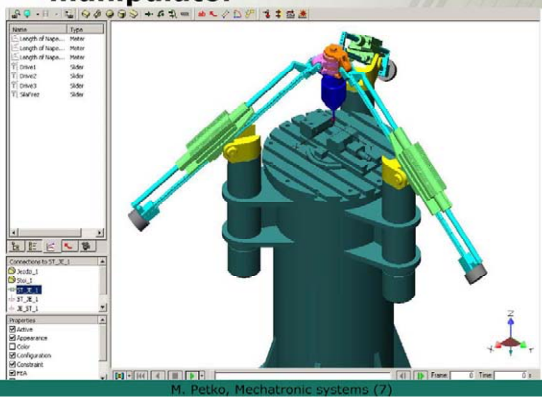
MES model of parts – strength and deformations



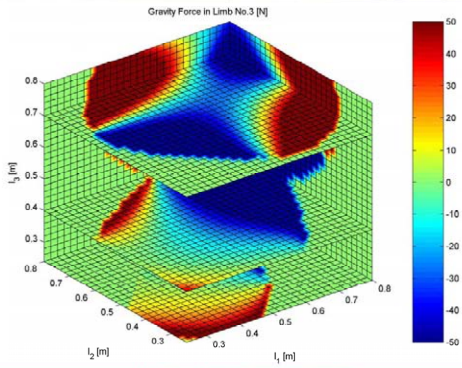
MES model of parts – strength and deformations



Multibody model of the manipulator



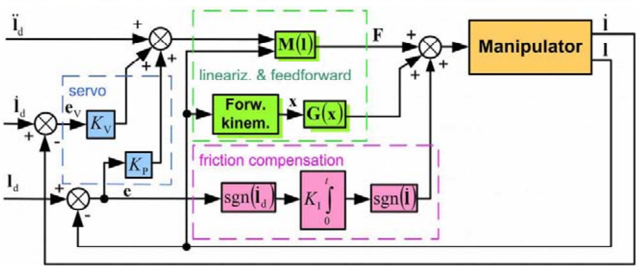
Gravity forces reduced to the actuator no. 3, in the workspace



Structural model of inverse dynamics

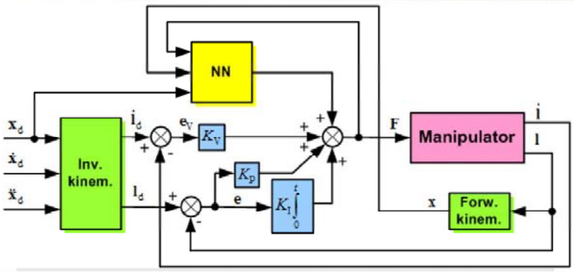
$$\begin{bmatrix} M_1(\mathbf{l}) & 0 & 0 \\ 0 & M_2(\mathbf{l}) & 0 \\ 0 & 0 & M_3(\mathbf{l}) \end{bmatrix} \cdot \begin{bmatrix} \ddot{l}_1 \\ \ddot{l}_2 \\ \ddot{l}_3 \end{bmatrix} + \begin{bmatrix} G_1(\mathbf{x}) \\ G_2(\mathbf{x}) \\ G_3(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

Stabilizing controller



$$\mathbf{F}(t) = \mathbf{M}(\mathbf{l}(t))\left(\ddot{\mathbf{l}}_d(t) + K_P\mathbf{e}(t) + K_V\mathbf{e}_V(t) + K_I\int_0^t \mathbf{e}(\tau)\ d\tau\right) + \mathbf{G}(\mathbf{x})$$

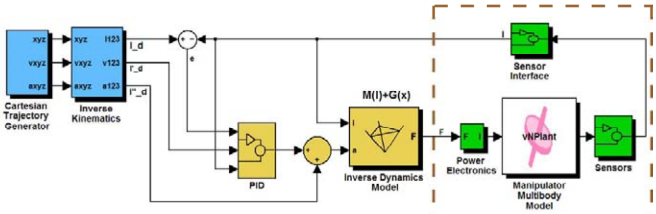
Neural controller



$$\mathbf{F}(t) = \sum_{j=1}^{3} \mathbf{W}_j \tanh\left(\sum_{l=1}^{15} \mathbf{w}_{j,l} \boldsymbol{\varphi}_l + \mathbf{w}_{j,0} \right) + \mathbf{W}_0 + K_P \mathbf{e}(t) + K_V \mathbf{e}_V(t) + K_I \int_0^t \mathbf{e}(\tau)\, d\tau$$

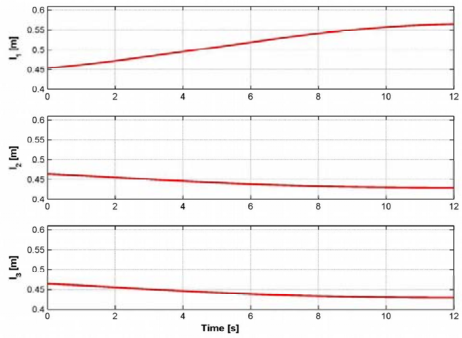
$$\boldsymbol{\varphi}(t) = \left[\mathbf{x}(t-2), \mathbf{x}(t-1), \mathbf{x}_d(t), \mathbf{F}(t-2), \mathbf{F}(t-1) \right]$$

Virtual prototyping

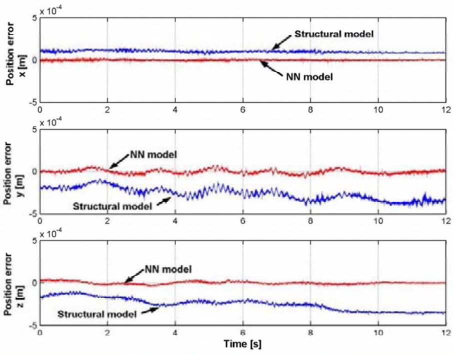


Overall system

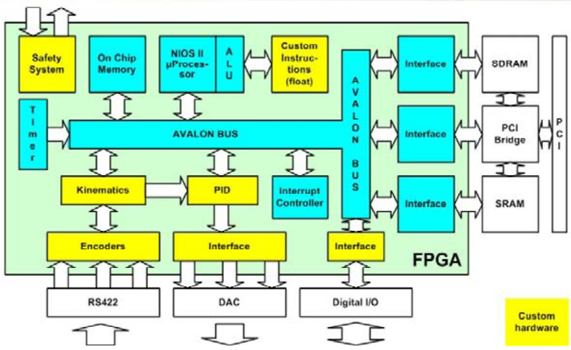
Experiment – trajectory



Trajectory tracking error



Architecture of the controller hardware



Comparison of Implementations

	Computed torque controller	Neural controller NN	PID	Forward kinematics	Total
Number of multiplications	181	91	30	11	148
Number of additions	59	66	27	7	107
Number of reciprocals	9	6	0	0	6
Number of square roots	7	0	0	1	2
Execution time – software only	2.9 ms	1 ms	0.4 ms	0.18ms	1.8 ms
Execution time – with hardware floating-point instructions	n. a.	25 µs	110 µs	2.3 µs	190 µs
Execution time – with hardware floating-point instructions, kinematics and PID	27 µs	25 µs	75 ns[a]	182 ns[a]	25 µs

[a] Executes in parallel to Nios II

Software vs. hardware realization

	LE	DSP	Execution time (hardware)	Execution time (software)
Multiplication	41	8	18 ns	2.3 µs
Addition	818	0	55 ns	2.7 µs
Reciprocal	1697	0	200 ns	2.8 µs
Square root	1149	0	182 ns	12 µs
Kinematics	1292	78	182 ns[a]	0.18 ms
PID	2150	0	75 ns[a]	0.5 ms

[a] Executes in parallel to Nios II

Usage of resources and calculation time

	Computed torque controller			Neural controller		
	Execution time	Resources LE	Resources DSP	Execution time	Resources LE	Resources DSP
Software only	2.5 ms	9023	10	1.8 ms	9023	10
With hardware floating-point instructions	n. a.	n. a.	n. a.	190 µs	11728	18
With hardware floating-point instructions, kinematics and PID	27 µs	15170	96	25 µs	15170	96

Calculation time of control algorithms

	FPGA with Nios II, 55 MHz	PowerPC, 400MHz	Celeron, 400MHz
Computed torque controller	27 µs	34 µs	34 µs
Neural controller	25 µs	29 µs	27 µs

Trajectory generators

- **In joint space**
  - **Polynomials of 3-4-3, 4-3-3-4 i 5 order**
- **In Cartesian space**
  - **From NC code**
  - **NC code describes only a path of the tool; a smooth trajectory along the path should be generated**