**Austrian – Bulgarian Automation Days**
Sofia, October 25-26, 2007

# Trends of Manufacturing Automation in Romania

**Theodor Borangiu, University Professor, PhD**
**National Centre of Research and Training in Robotics, Industrial Informatics and Material Engineering**
**University Politehnica of Bucharest, Dept. of Automation and Applied Informatics**
**E-mail: borangiu@cimr.pub.ro**

# Summary

1. **Generating CNC toolpaths from grey level image processing**
2. **Embedded numerical – adaptive control of machine tools**
3. **Robots for construction: from CAD to Real Time Control**
4. **Robot integration in manufacturing: Merged GVR – AVI tasks**

   **(Guidance Vision of Robots – Automated Visual Inspection)**
5. **New paradigms in manufacturing control: the holonic approach**

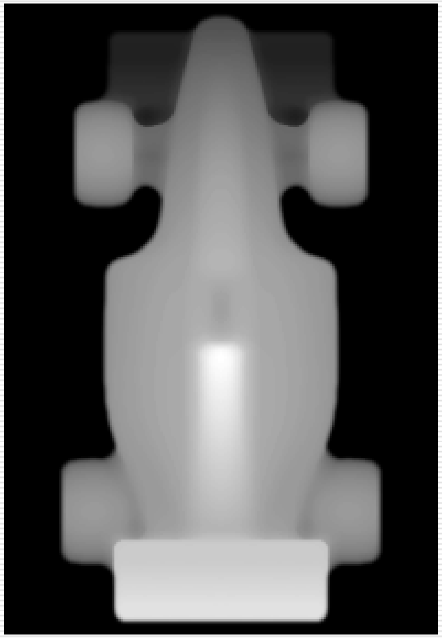
## 1. Generating CNC toolpaths from grey level image processing

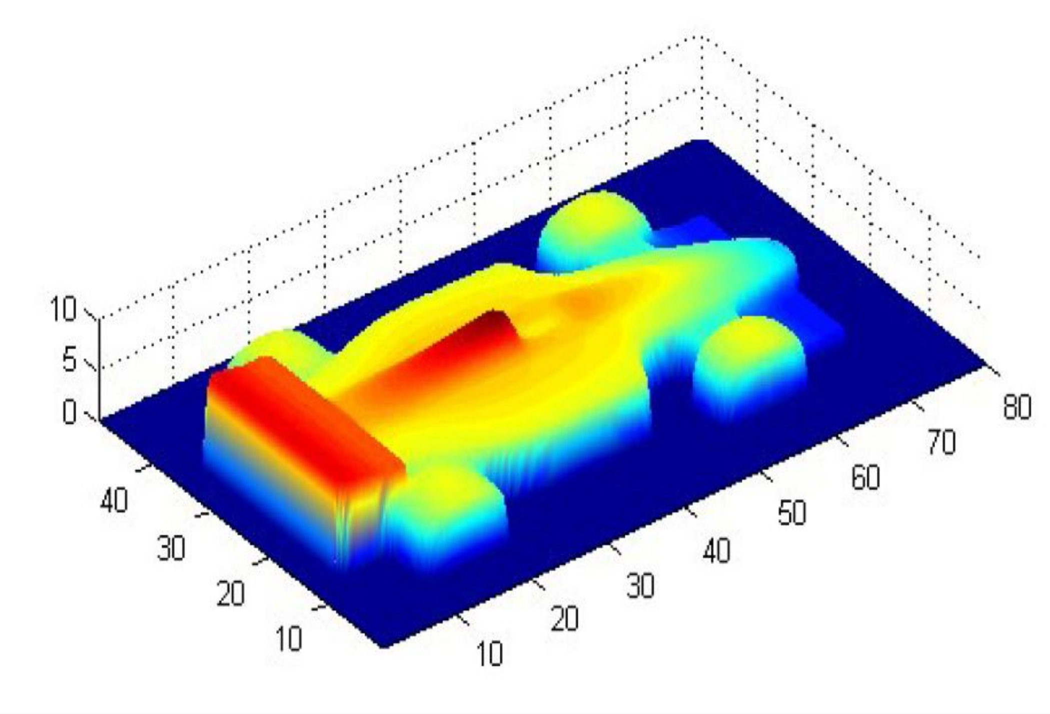
**Problem: Automatic generation of 3D machining surfaces with tool compensation from grey level image models**

- Height Map images
- Modelling the machining surface and tool shape
- Performing tool compensation
- Generating roughing toolpaths
- Generating finishing toolpaths
- Error analysis
- Future developments

# 1. Generating CNC toolpaths from grey level image processing

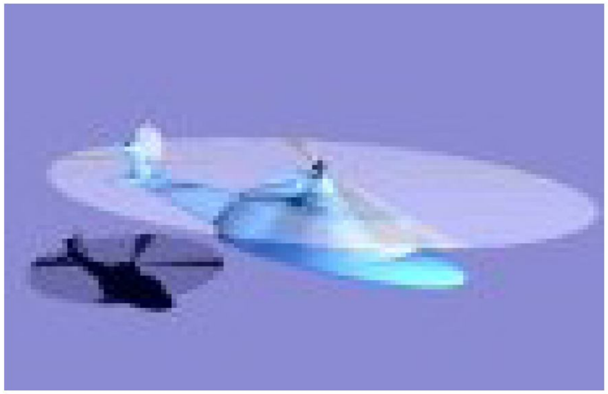
- **Height Map images**



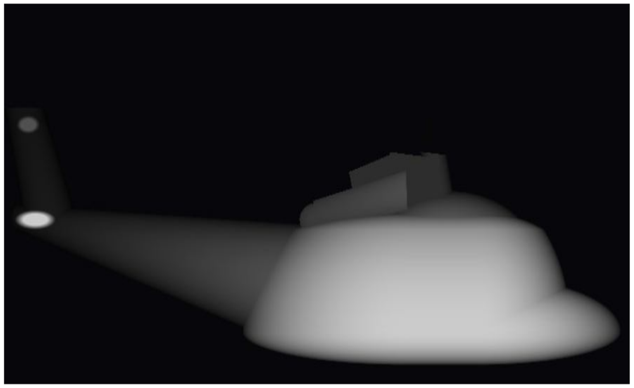


# 1. Generating CNC toolpaths from grey level image processing

## ❑ Obtaining height map images



3D Model in POV-Ray



Height Map Model

1) Remove light sources
2) Remove material textures
3) Use an ortographic camera
4) Apply a pigment with luminance proportional to the distance from the camera plane:
   - farthest point: pure black (luminance 0)
   - closest point: pure white (luminance 1)

# 1. Generating CNC toolpaths from grey level image processing

## ❑ Obtaining height map images

**The 3D surface of a model to be machined is scanned with a laser range finder device**:

- ➢ A vertical stripe of laser light is moved across the model object surface, and captured by a video camera. Along each horizontal scan line of the video frame, the *brightest* spot is taken to be the point at which the laser stripe "hits“ the surface (detection at sub-pixel resolution).
- ➢ The relative positions of the laser and the video camera are used to find the 3D coordinates of the brightest spot by triangulation.
- ➢ The *x*-coordinate of each point in the output depth image is determined by the position of the laser stripe for a particular video frame
- ➢ The *y*-coordinate corresponds to a raster line in the video frame,
- ➢ The depth value is computed from the brightness peak detected along the raster line in the video frame

# 1. Generating CNC toolpaths from grey level image processing

## 2.5D Surface Modelling

- Pixel graylevel at (i,j) encodes surface height at (x,y)
- Pixel-to-millimeter ratio:
    - $x = R\ i$
    - $y = R\ j$
- Minimum Z of the surface: black pixel
- Maximum Z of the surface: white pixel
- Graylevel value:
    - 8 bit integer: low precision, low storage space
    - 16 bit integer: good precision
    - Floating point: best precision, high storage space

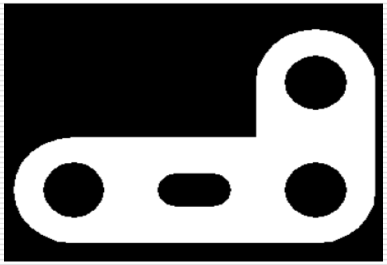
# 1. Generating CNC toolpaths from grey level image processing

## 2.5D Surface Modelling

**Simplest Case: 2 Dimensions**

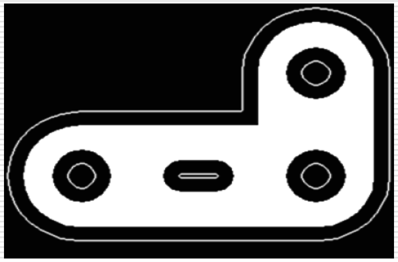
- Offsetting is equivalent to image dilation
- For efficiency, only contour pixels need to be considered
- Tool Path is generated by extracting the contour
- By image erosion, we obtain the machined shape

- Part model:



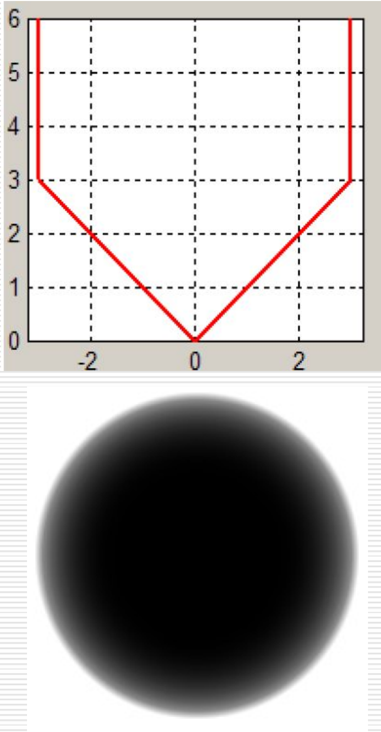
- Tool model:

- Tool compensation:

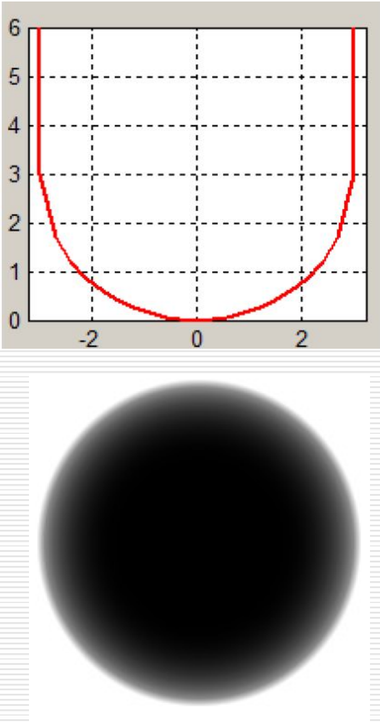


# 1. Generating CNC toolpaths from grey level image processing

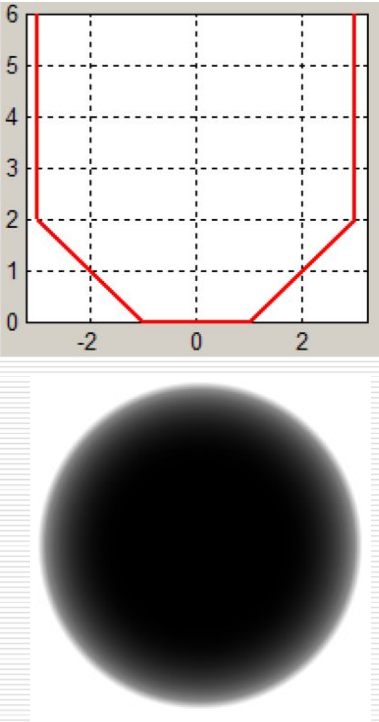
## ❑ Tool Shape Modelling



Conic Mill



Round End Mill



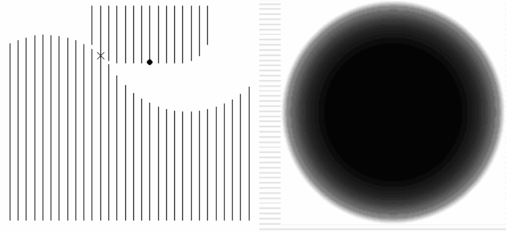
Bull End Mill

# 1. Generating CNC toolpaths from grey level image processing

## ❑ Tool Compensation

- ☐ Objective: Generating gouge-free tool paths
- ☐ Idea: For each (x,y) position, find the maximum depth at which the end mill can go down without cutting extra material

| | |
|---|---|
| Algorithm: | Graysale image dilation |
| Image: | Surface model |
| Structural element: | Tool model |



The principle of discrete cutter compensation based on gradient computing in the 2D grey scale cutter shape image:

**Idea**: at every location in the XY plane (i.e. any image pixel) the depth is computed which should be reached by the milling cutter, in order to be tangent to the surface.

*The shape of the milling cutter was modeled as a grey scale image, using the same scale factors as for the surface to be milled.*

# 1. Generating CNC toolpaths from grey level image processing

## Tool Compensation Results

### Advantages

- Gouge-free tool paths for many tool shapes
- Immediate generation of basic roughing and finishing tool paths
- Simple implementation, no need for complex 3D geometry computations

### Disadvantages

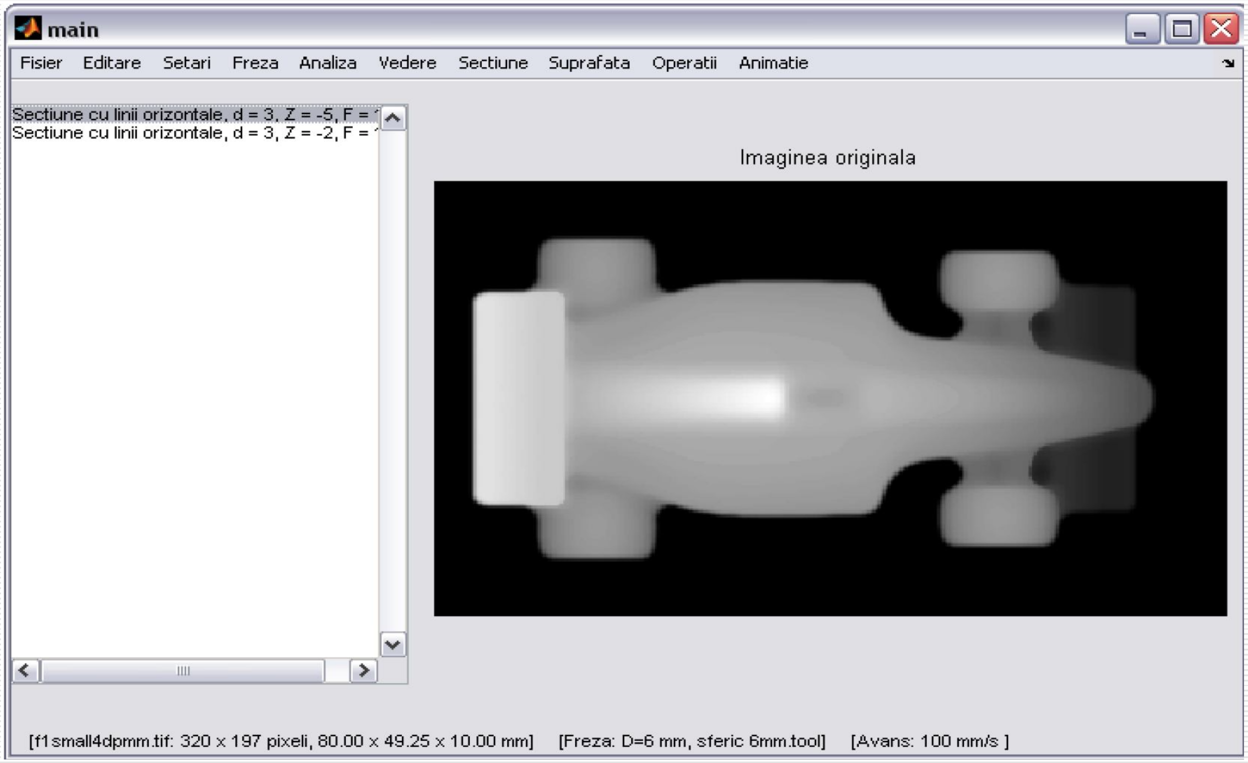
- High computation time
- High storage space for the image model
- Compromise between precision and speed!

### Increasing Speed

- It is not always necessary to compute the whole surface
- The algorithm can be parallelized

# 1. Generating CNC toolpaths from grey level image processing

## ❑ The Software – GUI Design



Sectiune cu linii orizontale, d = 3, Z = -5, F =
Sectiune cu linii orizontale, d = 3, Z = -2, F =

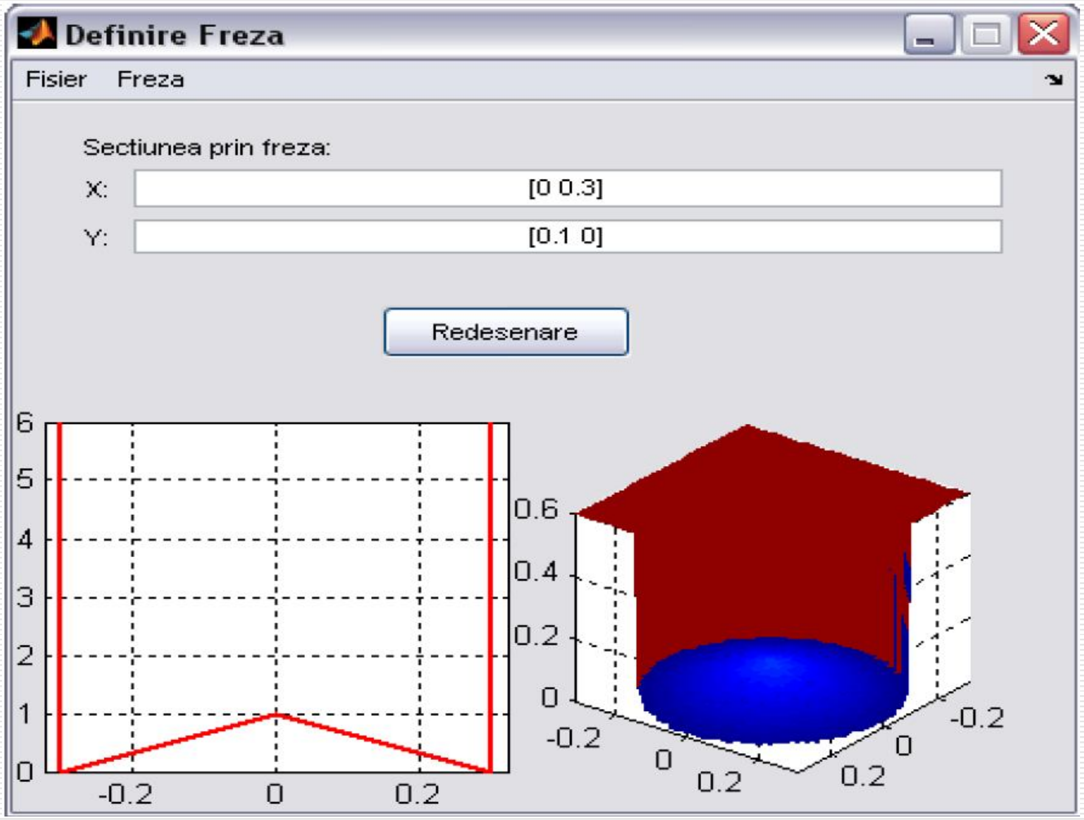
[f1small4dpmm.tif: 320 x 197 pixeli, 80.00 x 49.25 x 10.00 mm] [Freza: D=6 mm, sferic 6mm.tool] [Avans: 100 mm/s ]

## Software Features

- Grayscale model support
- Tool shape editor
- Roughing toolpath generation
- Finishing toolpath generation
- ISO CNC (G-Code) output

# 1. Generating CNC toolpaths from grey level image processing

## ❑ The Software – Tool Shape Editor



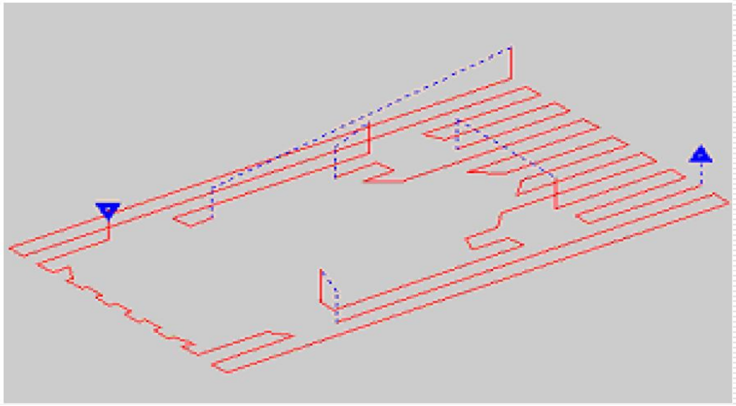
Sectiunea prin freza:

X: [0 0.3]

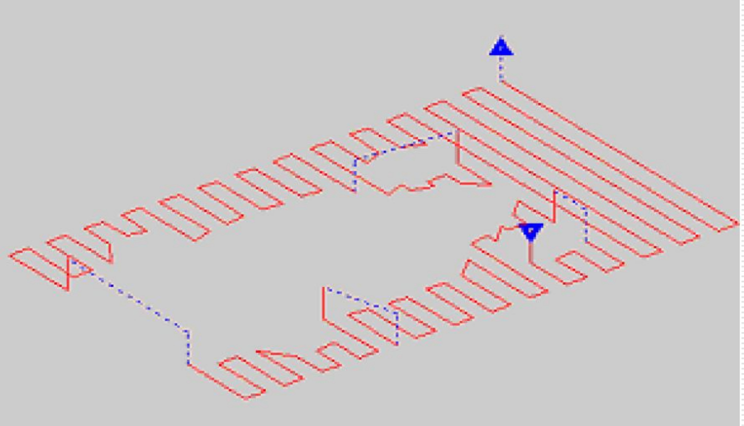
Y: [0.1 0]

# 1. Generating CNC toolpaths from grey level image processing

## ❑ Generating Roughing Toolpaths

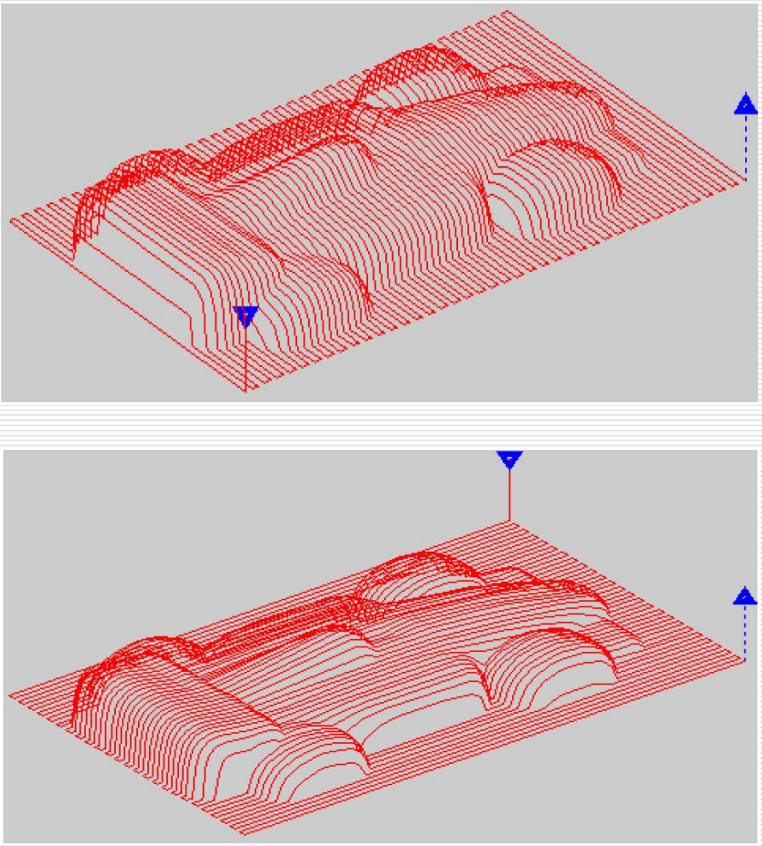


- ☐ Each roughing stage is performed at constant Z level
- ☐ At a given Z level, selecting the region where the cutter should clean up is an image thresholding operation
- ☐ For flat endmill cutters we use 2D offset compensation



# 1. Generating CNC toolpaths from grey level image processing

## ❑ Generating Finishing Toolpaths – 1st meth

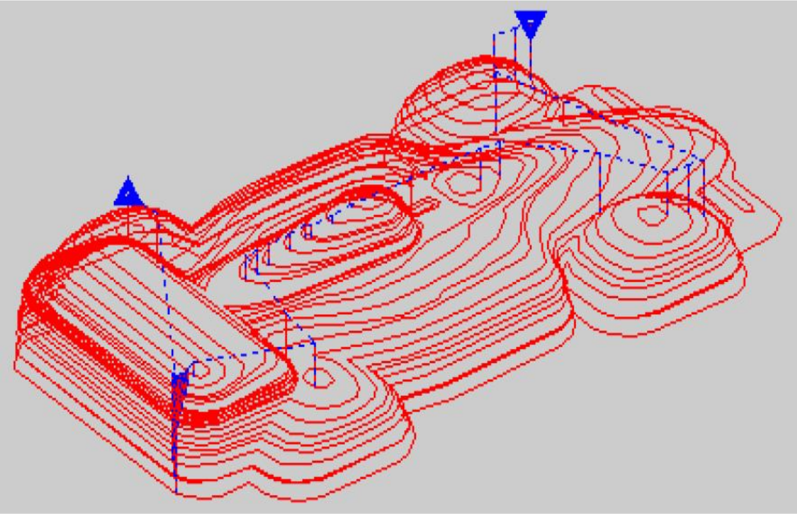


- In XY plane, the tool moves parallel with one axis or direction
- The tool moves on the “safe surface”
- There is no need to compute the whole “safe surface”

# 1. Generating CNC toolpaths from grey level image processing

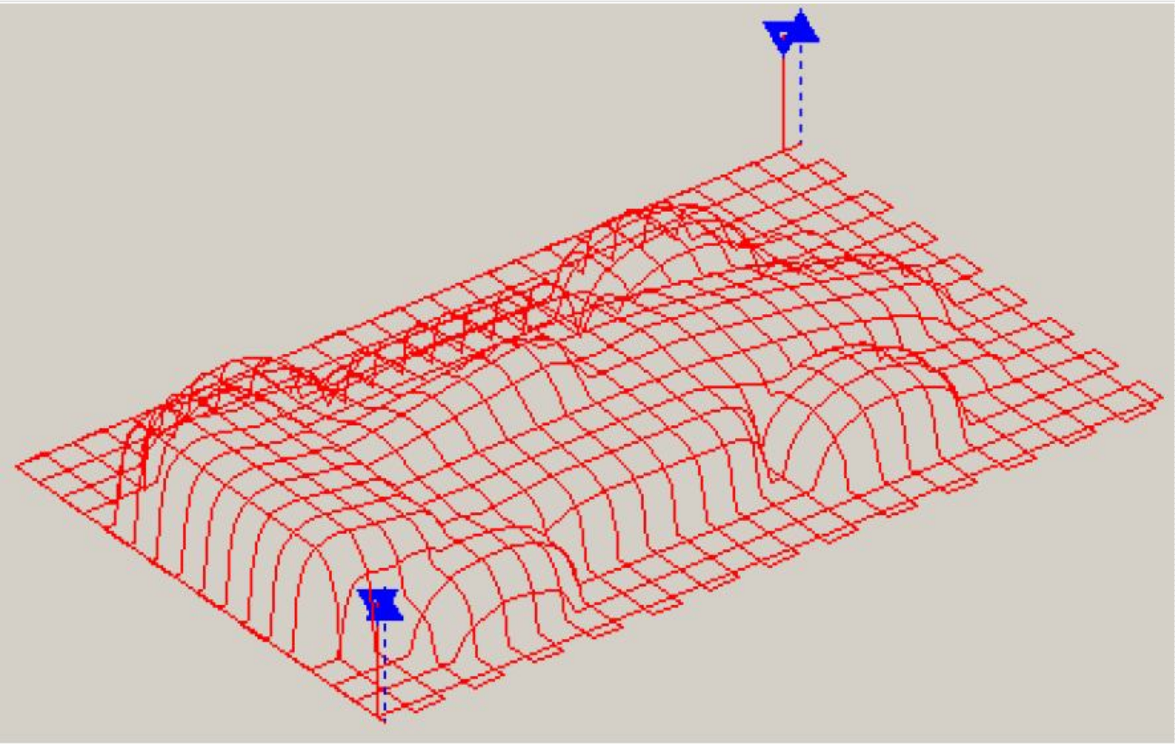
## ❑ Generating Finishing Toolpaths – 2nd method

- ☐ Tool paths are at constant Z levels
- ☐ Because of the tool shape, we cannot use 2D compensation any more
- ☐ The whole surface needs to be computed!



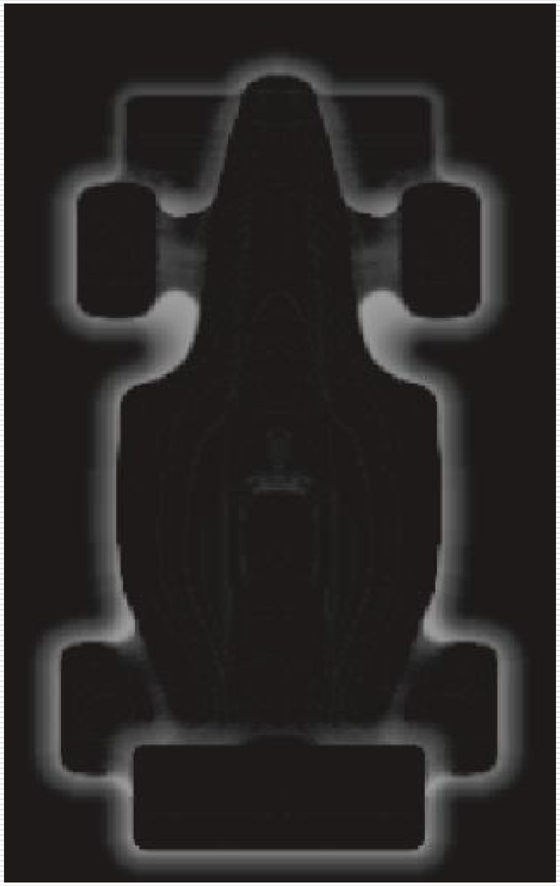
# 1. Generating CNC toolpaths from grey level image processing

- **Finishing - Combined**



# 1. Generating CNC toolpaths from grey level image processing

## ☐ Error Analysis 1



- ☐ A tool can be too big to machine the fine details
- ☐ At first, we can use a bigger tool to machine surfaces without details, and then a smaller tool to machine only the small details
- ☐ We can simulate one cutting operation and see what could not be machined

# 1. Generating CNC toolpaths from grey level image processing

## ❑ Error Analysis 2

- ☐ Model the toolpath like a grayscale image
- ☐ Erode the image using the tool model
- ☐ Compare the eroded image with the original model

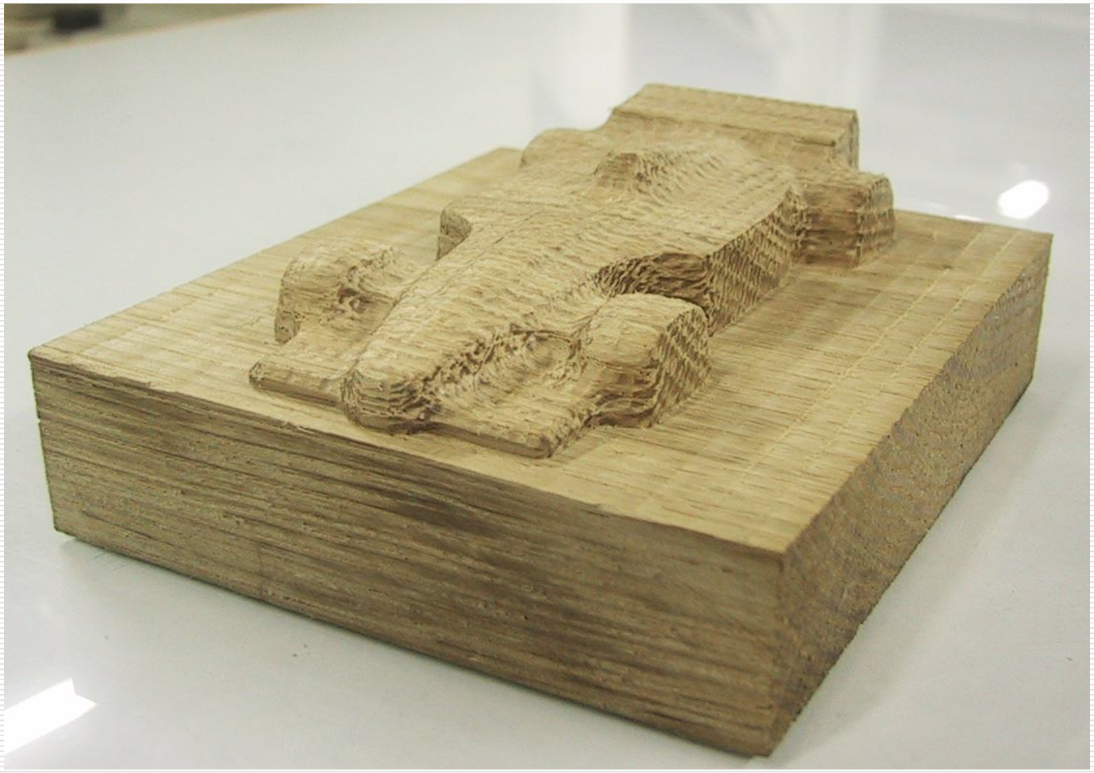
## 1. Generating CNC toolpaths from grey level image processing

- **ISO CNC Output**
  - Toolpaths are made of linear segments and circular arcs
  - Succesive segments may be approximated with circular arcs
  - Toolpath optimization: reduce the time for moving the head without cutting

```
M03
G0 X80 Y7.75
G1 Z-40 F100
G1 X65.5
G0 Z0
G0 X71.75 Y10.75
G1 Z-40 F100
G1 X80
G1 Y13.75
G1 X73
G1 Y16.75
G1 X80
G1 Y19.75
G1 X73.25
G0 Z0
G0 X0 Y0
M05
```

# 1. Generating CNC toolpaths from grey level image processing

## ❑ Sample Workpiece



# 1. Generating CNC toolpaths from grey level image processing

## ❑ Future Developments – Collision Detection

- ☐ The whole tool shape can be modelled, including the tool holder, to check if a tool path will cause a collision with the workpiece
- ☐ At every moment the amount of material can be computed; this is useful to check if the maximum allowed cutting depth of a tool is not exceeded.
- ☐ Vary feedrate and speed at roughing: save time, increase throughput

# 2. Embedded numerical-adaptive control of machine tools

## ❑ The inequality-constrained **O**ptimal **Co**ntrol of machining (ACO)

- ☐ **Quality Function**: the *cutting productivity*, for each one of the 2D closed *XY* roughing paths approximating one locus of spatial points of same grey level (points of depth $Z_i, Z_i = Z_{i-1} - \Delta_j$ in the part model image). The first paths to be machined = loci of nearest points relative to the range sensor (grey level value $z_0$ ), with cutting depth $\Delta_0$ . If the furthest image points relative to the sensor have a grey level $Z_f$ , then, by piecewise estimating the grey level gradient in the point depths range $Z_0 ... Z_f$ , the number of *C* cutting passes is computed to plan the $2_{1/2}$ approximation of 3D machining with cutting depths $\Delta_j$ , such that $\sum_j c_j \cdot \Delta_j = Z_f - Z_0$, and $\sum_j c_j = C$, and $c_j$ roughing paths $p_i$ have the same cutting depth $\Delta_j$ .
- ☐ Computation of **machining passes** considers: *form accuracy* (piecewise change of the cutting depth $\Delta_j$ on grey level gradient basis) and *material characteristics* (type and hardness of material to be removed impose upper limits on $\Delta_j$ ). **Result**: each roughing path $p_i$ should be realized at constant cutting depth $\Delta_j$ , feedrate $w_j = \mathrm{F}$ and spindle speed $n_j = \mathrm{S}$ to protect the tool and the machine's kinematical chain.
- ☐ **ACO approach**: maintains $\Delta_j$ constant at roughing path level, but *varies* in real time $w, n$ along trajectory $p_i$ – computed from image points of same grey level – optimize a cost function.

# 2. Embedded numerical-adaptive control of machine tools

## ❑ The optimal control strategy (ACO)

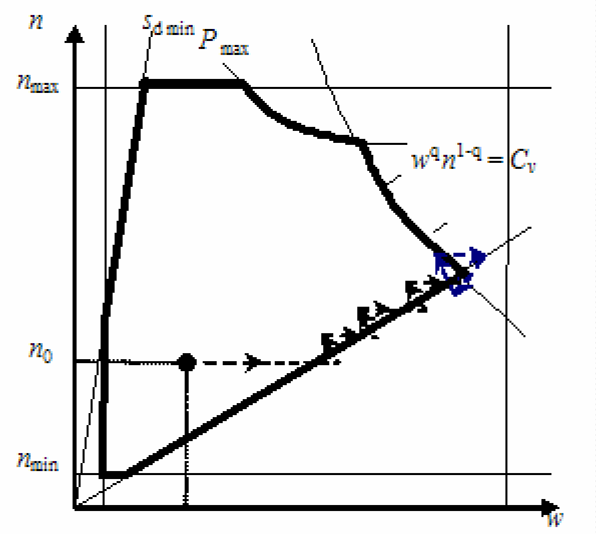


Fig. 7. Locating the optimum point at the intersection of a “mechanical load” constraint $C \cdot w - n = 0$ (torsion, deflection or pitch feed) with one of the type: electric power, feedrate or computed curve.

The optimization strategy for the milling process is based on two types of control actions:

1. Driving the working point $x = (w, n)$ towards the stationary one $x_0$ and maintaining it as close as possible to it ($x_0$ moves due to load disturbances i.e. to variations of material hardness or depth of material to be removed ).
2. Bringing back the current working point into the allowable domain bordered by constraints (2)?(7) and the computed curve (12), whenever these seven constraints are violated.

Table 1 Constraint grouping in response to violation of the allowable working domain boundaries

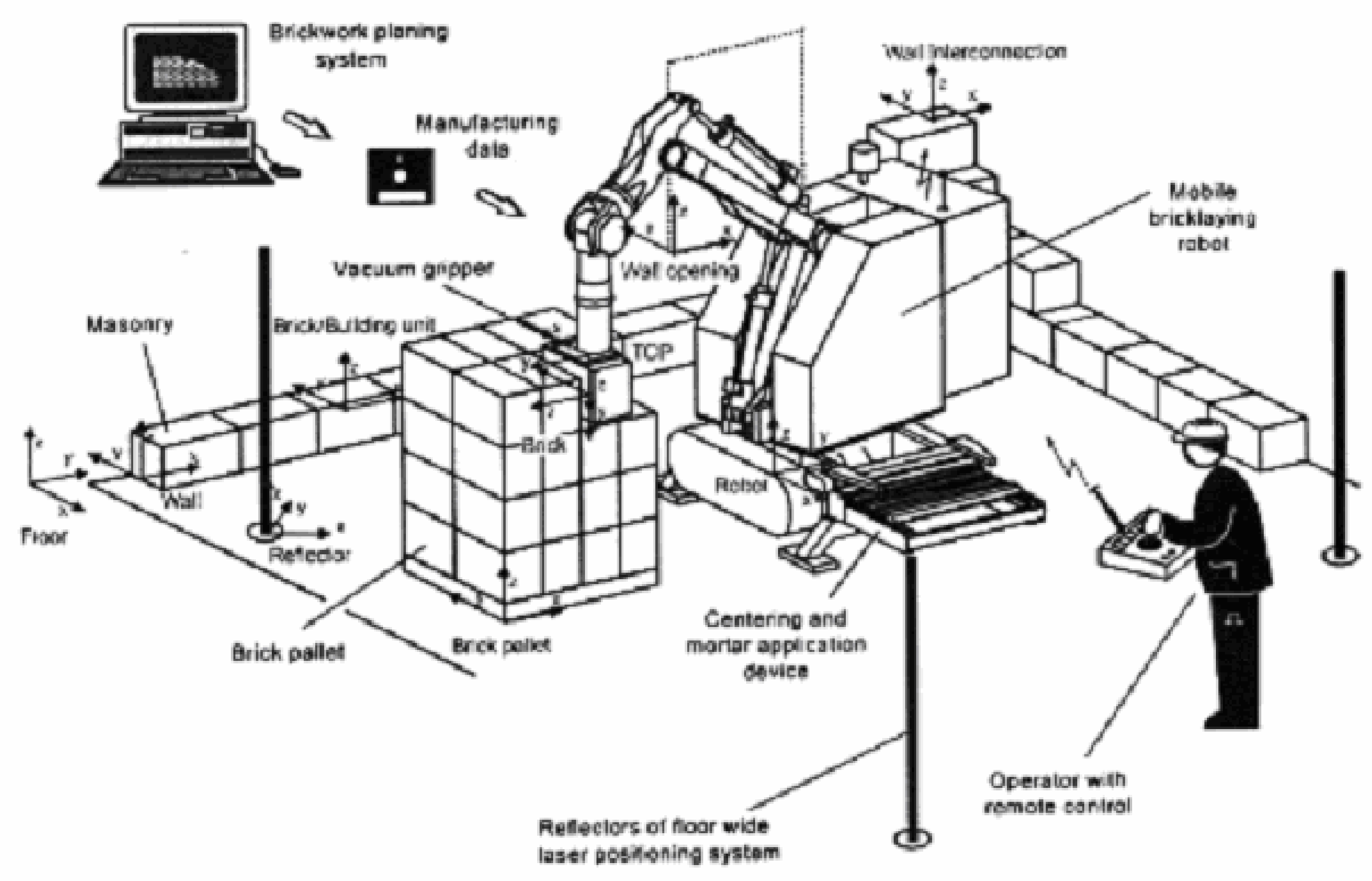
| Constraint class | Components | Controls actions at frontier violation |
|---|---|---|
| $R_1$: “max. chip load” type | (2), (3) and (4) right | $w \leftarrow w - \Delta w$<br>$n \leftarrow n + \Delta n$ |
| $R_2$: “max.power” type | (7), (12) and (5) right | $w \leftarrow w - \Delta w$<br>$n \leftarrow n - \Delta n$ |
| $R_3$: “unloaded” type | (4) left | $w \leftarrow w + \Delta w$<br>$n \leftarrow n - \Delta n$ |
| $R_4$: “speed” type | (6) | $w = \text{ct}$ and<br>$n \leftarrow n + \Delta n$<br>or $n \leftarrow n - \Delta n$ |
| $R_5$: “min. feed” type | (5) left | $w \leftarrow w + \Delta w$<br>$n = \text{ct}$ |

## 3. Robots for Construction – from CAD to RT Control

**Control trends in the building industry. R&D objectives**:

- **Requirements in construction industry**: quality, standardization and cost reduction: *replication of construction tasks* (bricklaying, windows placing, applying mortar and finishing operations – painting, polishing) and *increasing work productivity*.
- **Rationalization efforts** in the construction industry: attempt to *create information systems* progressively automating the building processes – e.g. bricklaying
- **New programming and control solutions** for robot systems for the building industry: implementing *CAD/CAM solutions* generating, managing and using at run time in robot programs production data extracted from civil engineering projects.
- **Research Objectives**: Design and implement an open and modular control solution for construction robots based on embedded systems integrated in a Service Oriented Architecture: (i) *AGV functionality* to access locations in the building site; (ii) *Arm dexterity* for construction tasks; (iii) *Open motion control HW & SW in an embedded architecture*; (iv) *KBTS* mapping technology data (materials, operations) to programs.

# 3. Robots for Construction – from CAD to RT Control



**Bricklaying task**: 7-d.o.f. mobile construction robot and its working environment

# 3. Robots for Construction – from CAD to RT Control

## Functional architecture:

•The mechanical device: a 5-d.o.f. *cylindrical arm*, partially closed-chain kinematic structure carried by a *mobile wheeled platform*.

•The resulting 7-d.o.f. mobile construction robot is able to move on *quasi-horizontal prepared surfaces* (floors), and generate a *workspace of 3.5 m height*.

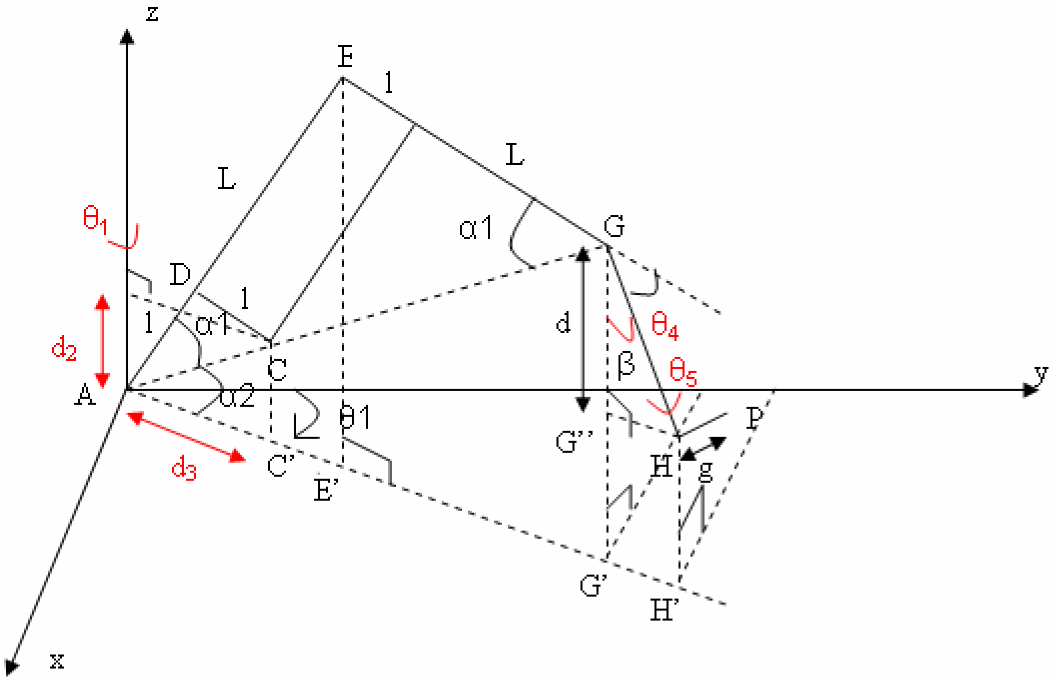
•The mobile robot is a *free-ranging* (non-guided) *wheeled vehicle*, capable to avoid obstacles (e.g. brick pallets) in a structured environment; its arm performs coordinated movements either in the *Cartesian space* or in the *5-dimension joint space* automatically at program execution or under manual control.

•A computer-based operator console (wireless laptop) is used both as:

- ➢ *teach pendant* for robot point learning and
- ➢ *robot terminal* for execution of monitor commands, program editing, debugging, and execution start-up and monitoring.

## 3. Robots for Construction – from CAD to RT Control

# Geometry Model of the 5-d.o.f. Cylindrical Arm



The closed-chain kinematic structure of the 5-d.o.f. cylindrical robot manipulator

## 3. Robots for Construction – from CAD to RT Control

### Generating Production Data for the Bricklaying Robot

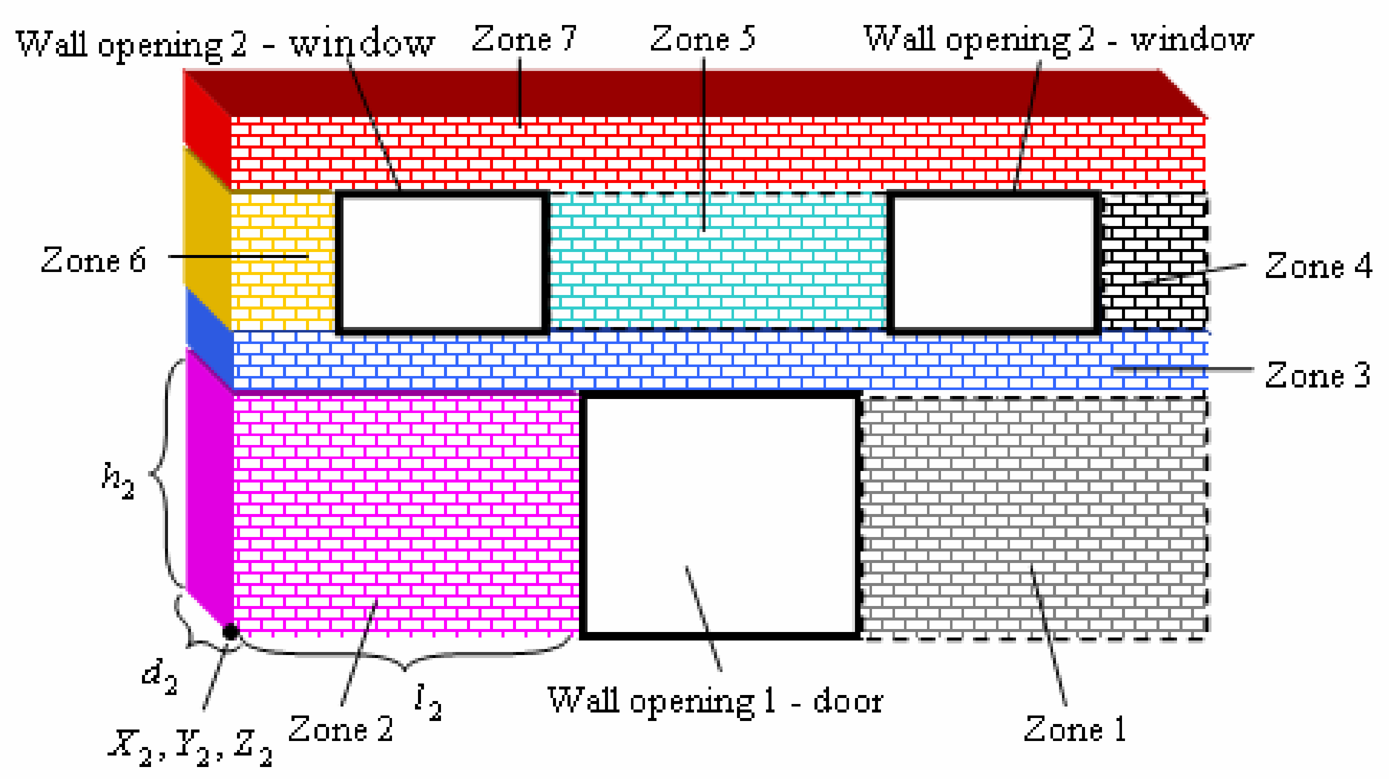
The process of *generating production data* for the bricklaying robot is done in 3 stages:

1. **Extraction of geometry data from the architecture project and construction specifications** (either from AUTOCAD files or manually input). 3D locations and dimensions of the walls, h-stockades (bulwark), wall openings, a.o, represent the input data. The output data computed in this stage refers to:

   (i) the *dimensions* and coordinates of each *elementary masonry zone* relative to a unique world frame, and
   (ii) the *specification of materials* necessary for the construction of elementary zones.

2. **Partition of the global masonry in wall segments** (relatively to a single floor of the building); this consists in joining several *elementary masonry zones* , , which may be completely included within the dexterous space of the cylindrical robot arm, with respect to a robotic bricklaying task associated to a *wall segment k*.

# 3. Robots for Construction – from CAD to RT Control



**Stage 1**: extraction of geometry data and locating data for *elementary wall zones*
(from architecture plans and construction specs.)

## 3. Robots for Construction – from CAD to RT Control

**3. Determining the material requirements** – number and types of bricks, the specification of *brick pallets* – location, size, form, organization, and the *data for application programs* – robot points, parameters of „pick-and-place” routines, execution order for wall segments, *planning of intermediate mobile robot locations* for navigating to successive wall segment locations or exiting the working area.

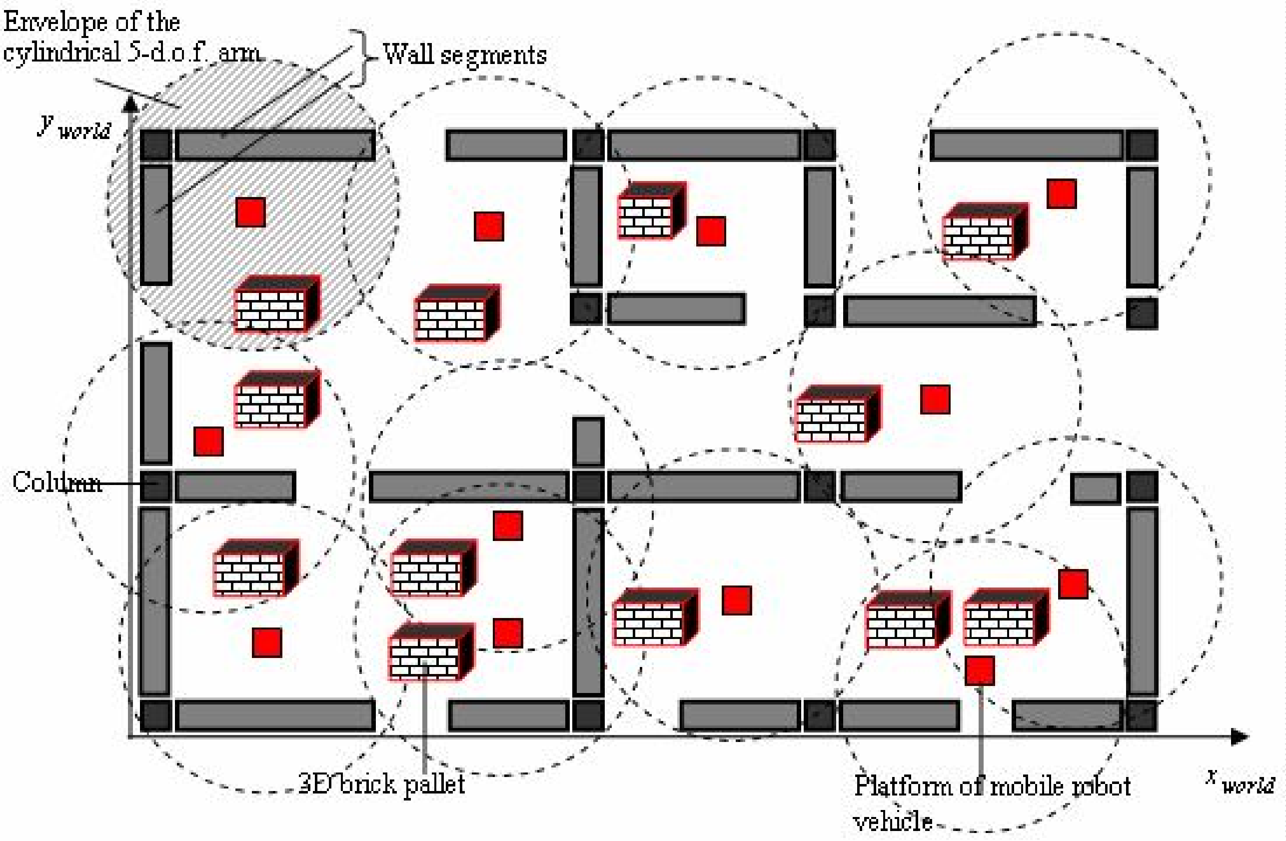
The **typical robot program sequence** allowing execution of the task associated to a current wall segment eventually provides access to more than one brick pallets.

I. *Manual/automatic guidance of the robot vehicle* to a world location from which access is granted both to the 3D brick pallet and to the ensemble of wall segments included in one elementary masonry zone.

II. *Auto locating of the mobile robot platform* using the laser scanner and range finder device mounted on the robot platform; the sensor scans over 360 degrees in a planar movement and, from the distances measured to three fixed reflectors (of known locations) determines its own location *X,Y* in the world frame.

III. *Performing the bricklaying task* (“pick and place” loop, 3D stack access, …)

# 3. Robots for Construction – from CAD to RT Control



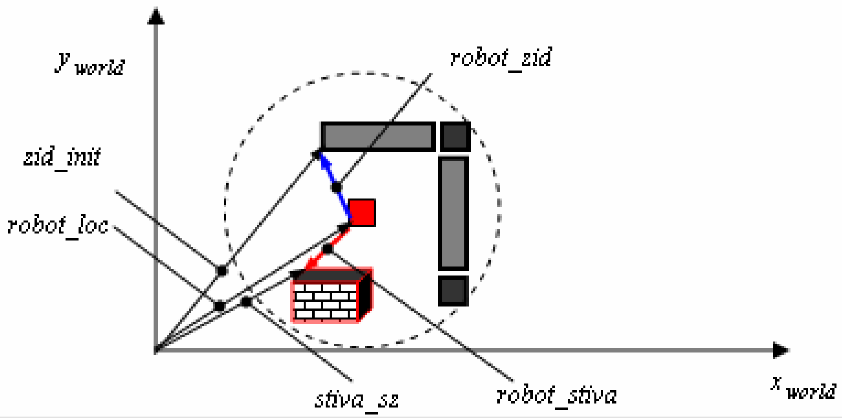
Partitioning a global masonry task in 12 bricklaying subtasks each one associated to a wall segment

# 3. Robots for Construction – from CAD to RT Control

*Computing the relative transformations* to access respectively the base of the brick pallet and the base of the current wall segment:

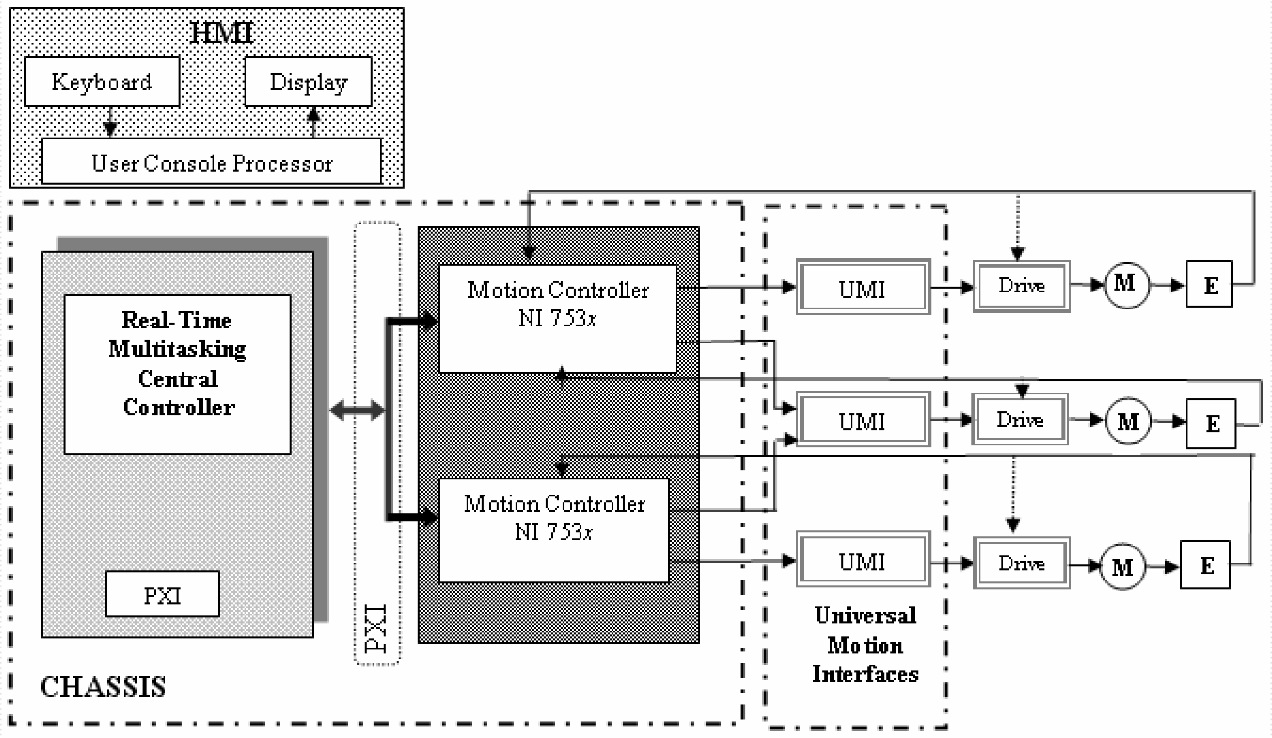
$$robot\_zid = (robot\_loc)^{-1} : zid\_init$$

$$robot\_stiva = (robot\_loc)^{-1} : stiva\_sz$$



## 3. Robots for Construction – from CAD to RT Control

- **Motion control architecture: multiprocessor, dual FPGA-DSP controllers**



12-axis *HW* implementing solution for robot motion control: 5-robot arm; 3-AGV; 4-platform inclination

# 3. Robots for Construction – from CAD to RT Control

## Modelling & embedded control of robot vehicle & arm motion

RT embedded NI PXI Controller (National Instruments)

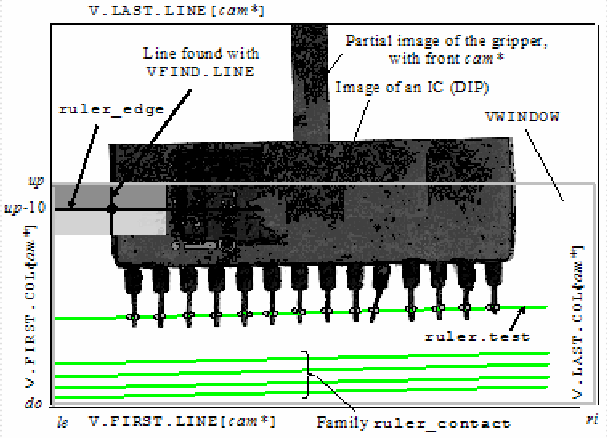
# 4. Robot Integration in Manufacturing: Merged GVR – AVI Tasks

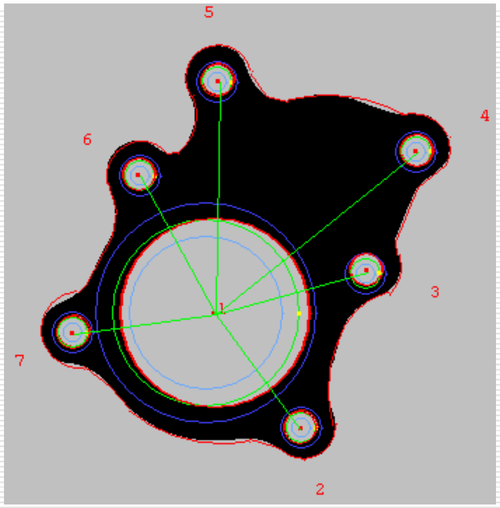
- **Need for implementing machine vision systems in robotics and manufacturing**: AI techniques are applied to create the best vision environment and adapt processing to lighting variations and part flow characteristics.

- **How did the distinction between Guidance Vision in Robotics (GVR) and Automatic Visual Inspection (AVI) become blurred?** → More and more inspection tasks require manipulation, and more and more component assembling / material processing tasks require quality inspection.

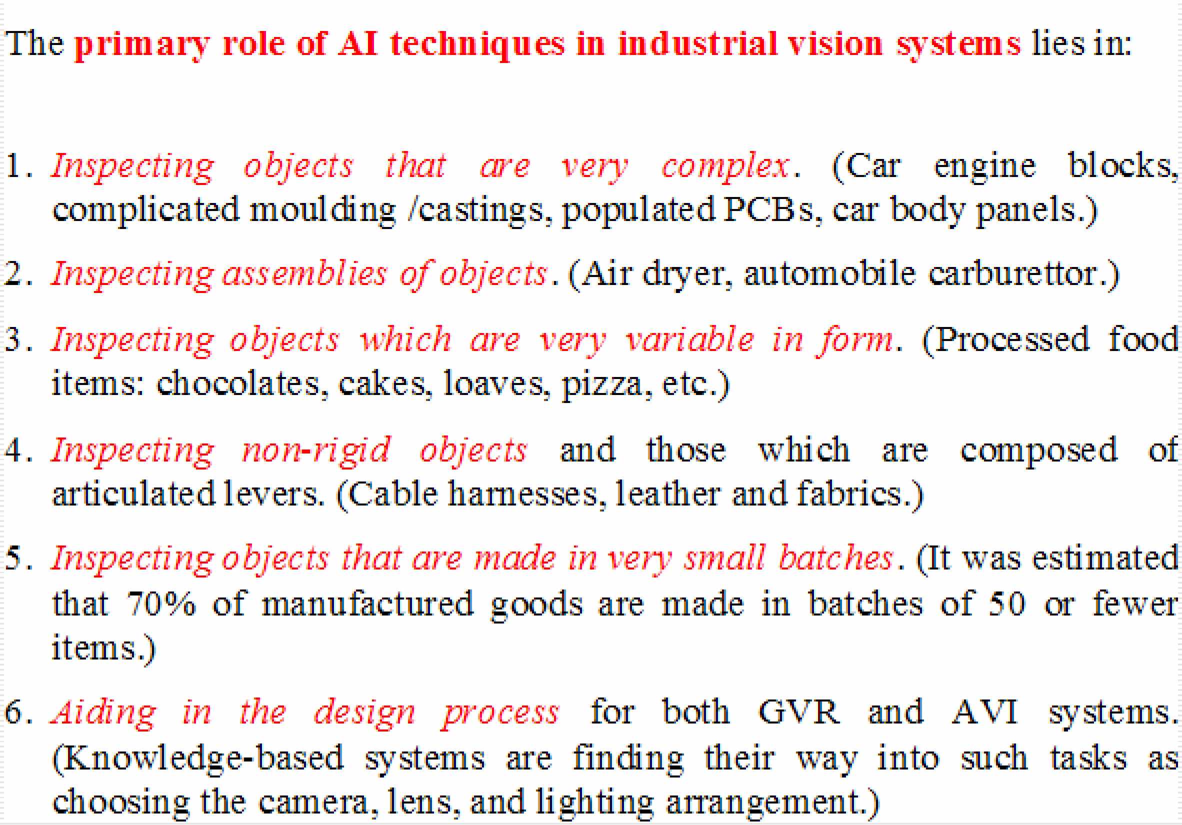
- **Why is intelligence needed for industrial vision systems?** Two technologies that have hitherto been almost disparate: *Image Processing* (**IP**) and *Artificial Intelligence* (**AI**) are currently being integrated. Of special interest are the tasks of *inspecting and manipulating industrial artefacts*.

- **Important vision applications in industry**: automotive, electronics, semiconductors, robotics, fabricated metal, printing, food/beverage, and pharmaceutical/medical.





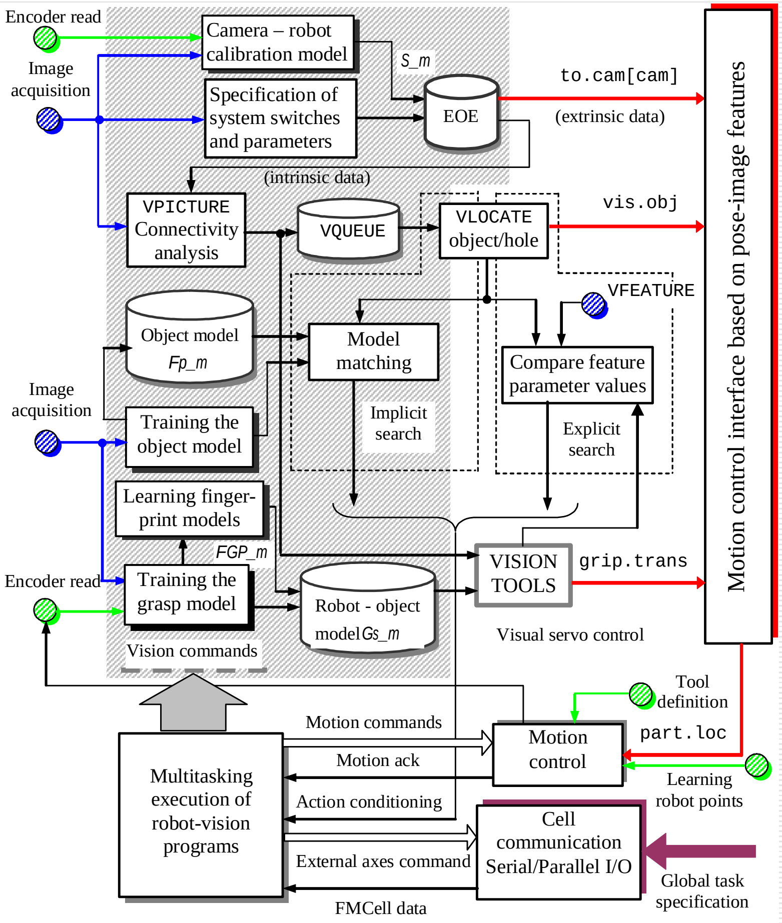
# 4. Robot Integration in Manufacturing: Merged GVR – AVI Tasks



The **primary role of AI techniques in industrial vision systems** lies in:

1. *Inspecting objects that are very complex*. (Car engine blocks, complicated moulding /castings, populated PCBs, car body panels.)
2. *Inspecting assemblies of objects*. (Air dryer, automobile carburettor.)
3. *Inspecting objects which are very variable in form*. (Processed food items: chocolates, cakes, loaves, pizza, etc.)
4. *Inspecting non-rigid objects* and those which are composed of articulated levers. (Cable harnesses, leather and fabrics.)
5. *Inspecting objects that are made in very small batches*. (It was estimated that 70% of manufactured goods are made in batches of 50 or fewer items.)
6. *Aiding in the design process* for both GVR and AVI systems. (Knowledge-based systems are finding their way into such tasks as choosing the camera, lens, and lighting arrangement.)

# 4. Robot Integration in Manufacturing: Merged GVR – AVI Tasks

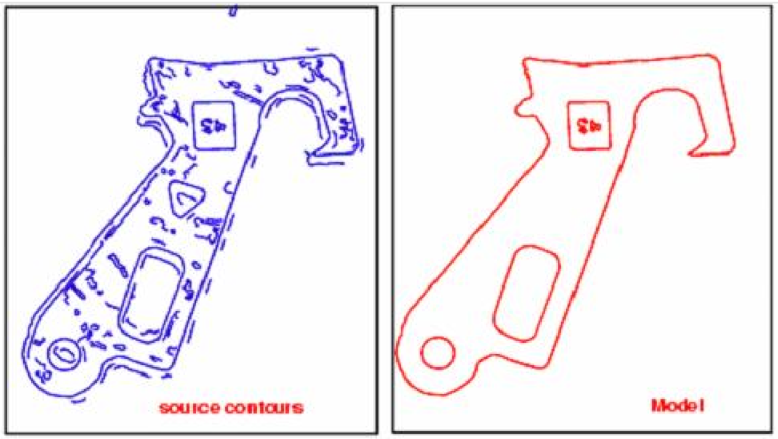


**Guidance vision** for **robot part grasping** (**RVAVI** function sets):

- **Scene functions**: accessing and use the *camera-robot calibration model* for *visually* blob locating and measuring in *robot world*
- **Scene-object functions**: learn, plan, and use *object models* for recognition
- **Robot-object functions**: learn and apply *object grasp models* for object classes and grasping styles
- **Robot-scene functions**: define, integrate, and check at run time *fingerprint models* against gripper-object related poses and scene occupancy for *collision-free* access to objects

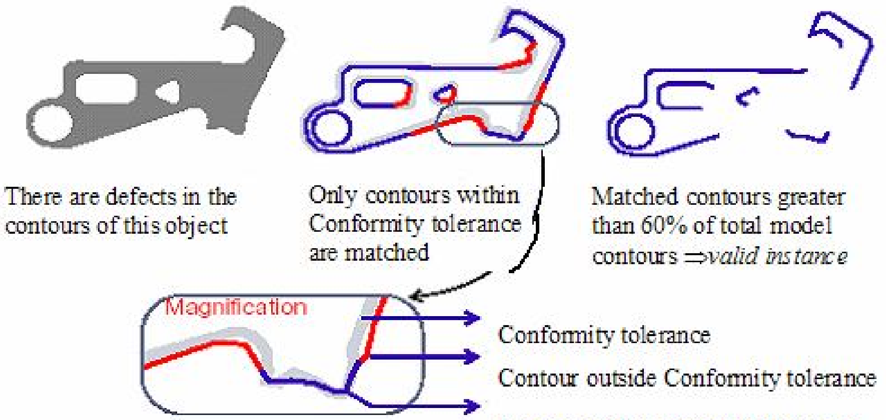
# 4. Robot Integration in Manufacturing: Merged GVR – AVI Tasks

Object Recognition Models (**ObjectFinder**)



STEP 1. *Source contours* are detected in the input greyscale image.

STEP 2. *Appropriate features* are selected to build the model.



There are defects in the contours of this object

Only contours within Conformity tolerance are matched

Matched contours greater than 60% of total model contours ⇒*valid instance*

Conformity tolerance

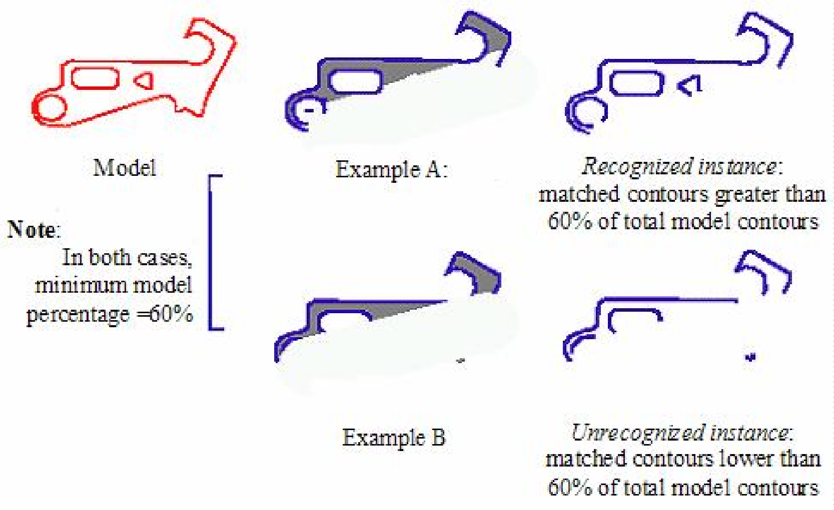
Contour outside Conformity tolerance

Contour within conformity tolerance

- **Conformity**: maximum local deviation between expected model contours and the contours actually detected in the input image.

# 4. Robot Integration in Manufacturing: Merged GVR – AVI Tasks

- **Verify%**: minimum percentage of model contours that need to be matched in the input image in order to consider the instance as valid. A higher value ⇒ faster recognition & higher rejection rate.


Model

Example A:

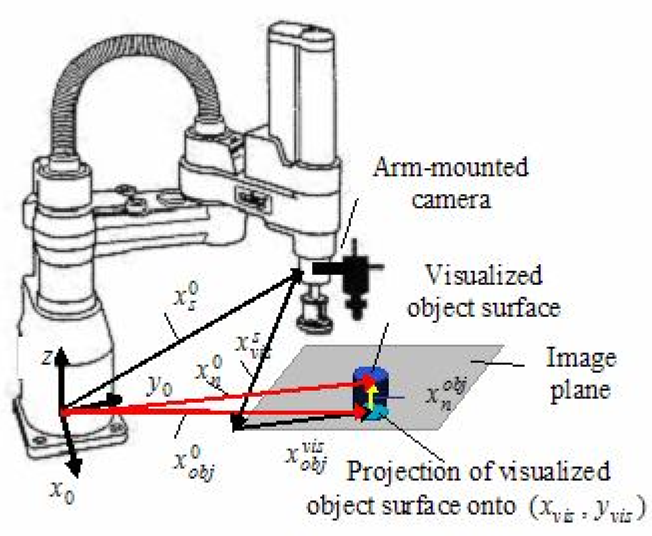
*Recognized instance*: matched contours greater than 60% of total model contours

**Note**: In both cases, minimum model percentage =60%

Example B

*Unrecognized instance*: matched contours lower than 60% of total model contours

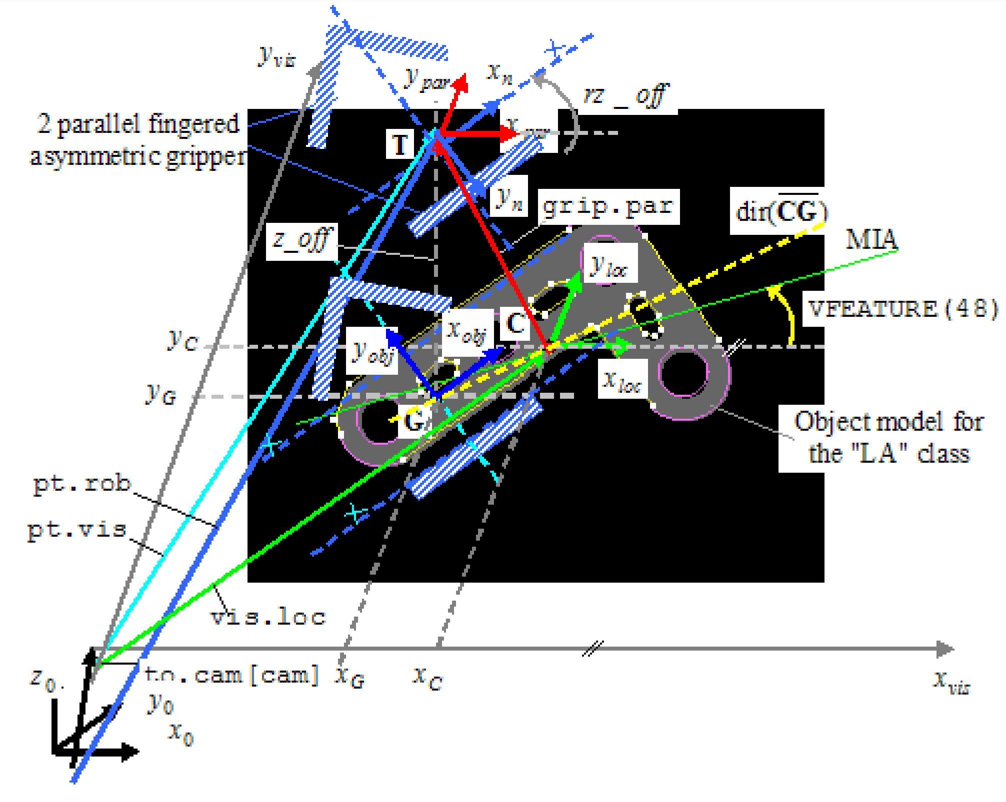
## Object Grasping Models



# 4. Robot Integration in Manufacturing: Merged GVR – AVI Tasks

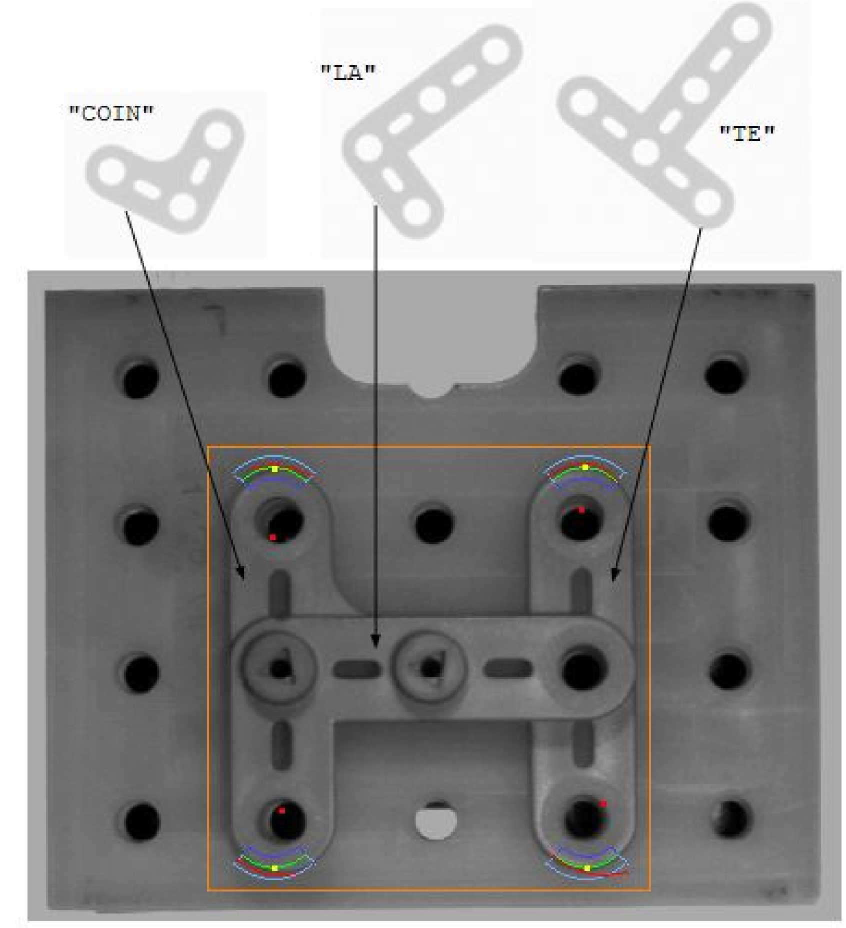
## Object Grasping Models

## Fingerprints Models for Collision Avoidance



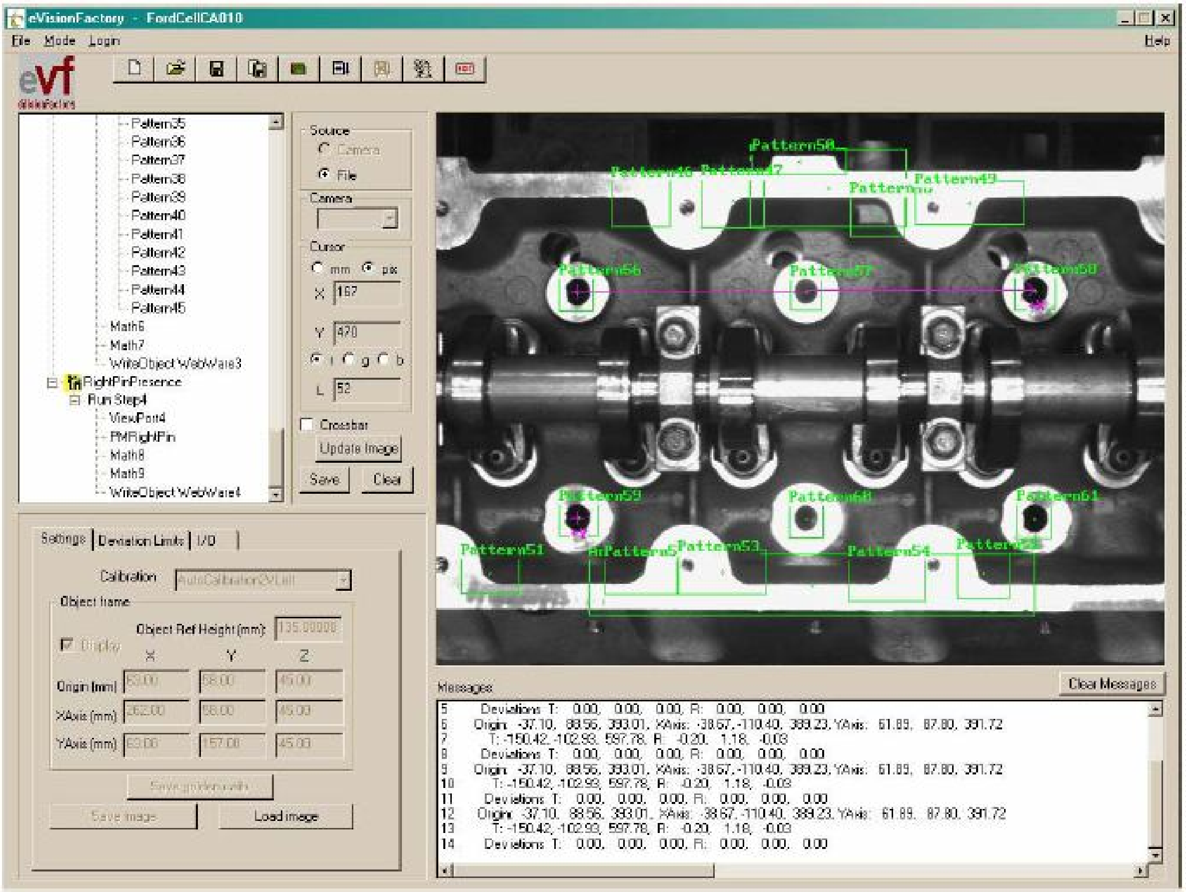
# 4. Robot Integration in Manufacturing: Merged GVR – AVI Tasks

Greyscale image of the final assembly of "COIN", "LA", and "TE" components, and graphic overlay of the vision tools used for inspection.



# 4. Robot Integration in Manufacturing: Merged GVR – AVI Tasks

## Anchor Feature Detection and Measurements



Source: ○ Camera ● File

Cursor: ○ mm ● pix

X 167

Y 470

● r ○ g ○ b

L 52

☐ Crossbar

Calibration: AutoCalibration2VLinl

Object frame

☑ Display

Object Ref Height (mm): 135.00000

|  | X | Y | Z |
|---|---|---|---|
| Origin (mm) | 63.00 | 58.00 | 45.00 |
| XAxis (mm) | 262.00 | 58.00 | 45.00 |
| YAxis (mm) | 63.00 | 157.00 | 45.00 |

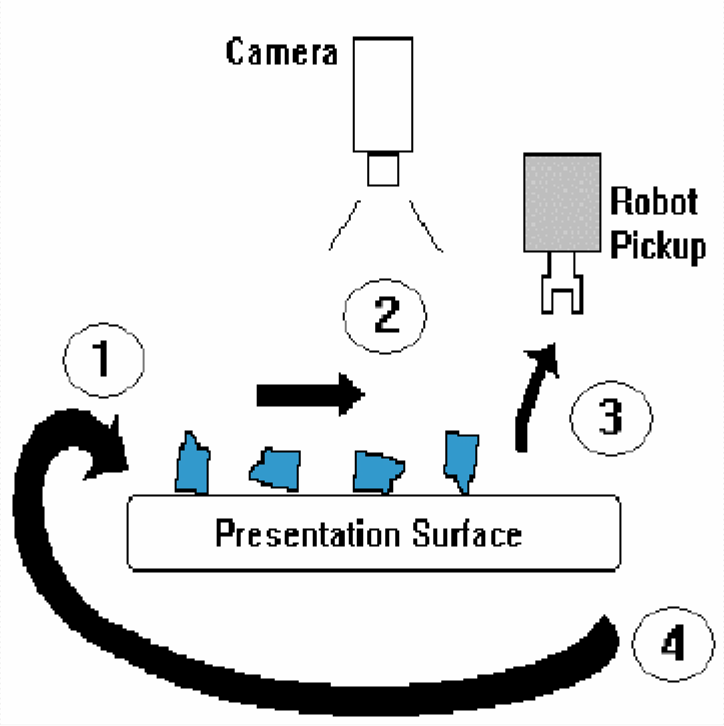
Messages

```
5     Deviations T:   0.00,   0.00,   0.00, R:   0.00,   0.00,   0.00
6    Origin: -37.10,  88.56,  393.01, XAxis: -38.67,-110.40,  389.23, YAxis:  61.89,  87.80,  391.72
7       T: -150.42,-102.93,  597.78, R:  -0.20,   1.18,  -0.03
8     Deviations T:   0.00,   0.00,   0.00, R:   0.00,   0.00,   0.00
9    Origin: -37.10,  88.56,  393.01, XAxis: -38.67,-110.40,  389.23, YAxis:  61.89,  87.80,  391.72
10      T: -150.42,-102.93,  597.78, R:  -0.20,   1.18,  -0.03
11    Deviations T:   0.00,   0.00,   0.00, R:   0.00,   0.00,   0.00
12   Origin: -37.10,  88.56,  393.01, XAxis: -38.67,-110.40,  389.23, YAxis:  61.89,  87.80,  391.72
13      T: -150.42,-102.93,  597.78, R:  -0.20,   1.18,  -0.03
14    Deviations T:   0.00,   0.00,   0.00, R:   0.00,   0.00,   0.00
```

Screenshot of the vision system user interface during part training, showing a cylinder head and the features used at run time by the 3D part locating kernel to calculate the object's 3D pose.

# 4. Robot Integration in Manufacturing: Merged GVR – AVI Tasks

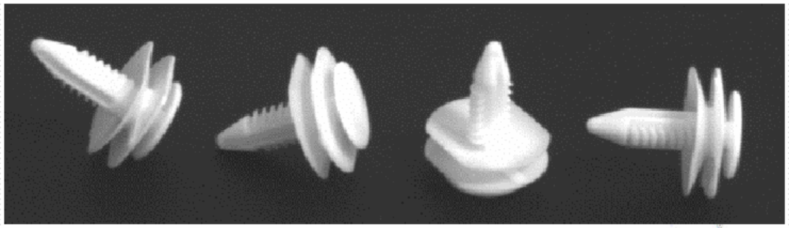
## Evolving Strategies for Flexible Part Feeding



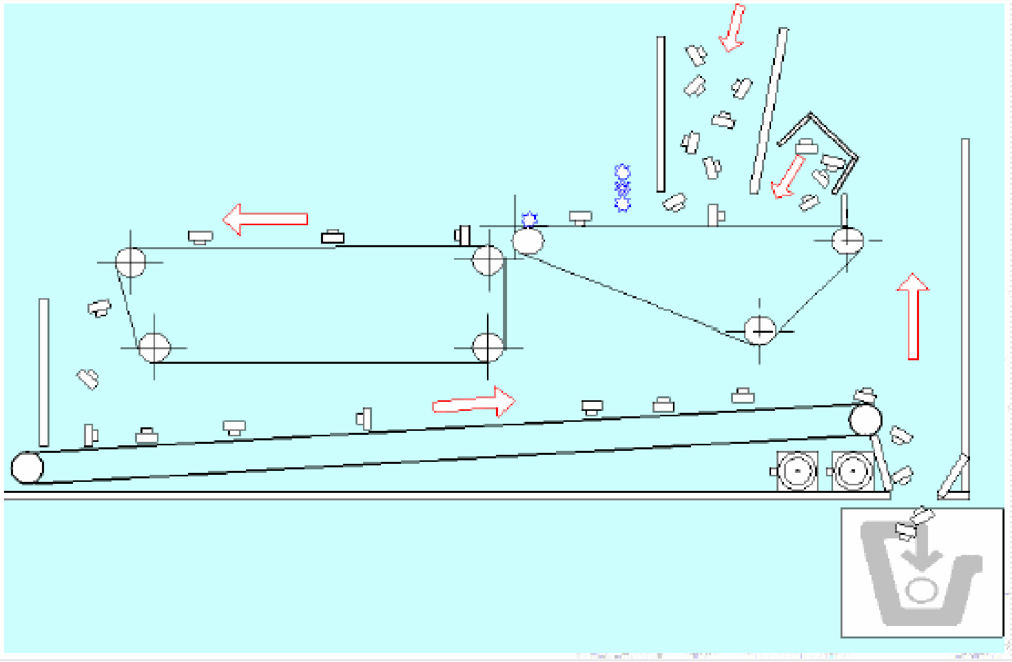
Basic flexible feeding concept: 1 – supply parts; 1 – locate desired parts; 3 – pick qualified parts; 4 – recycle parts that are not picked.

# 4. Robot Integration in Manufacturing: Merged GVR – AVI Tasks

Examples of stable states for some type of part on a conveyor belt.

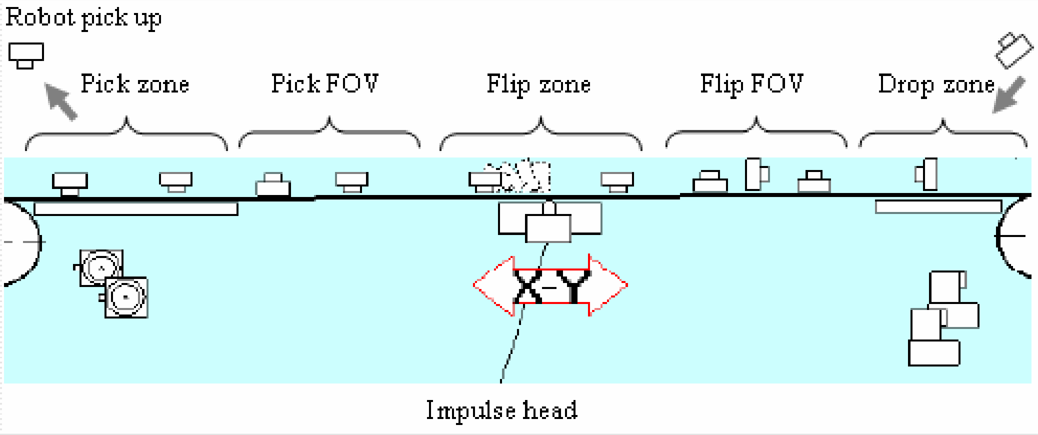


The solution for flexible feeding with continuous mode and part reorientation.

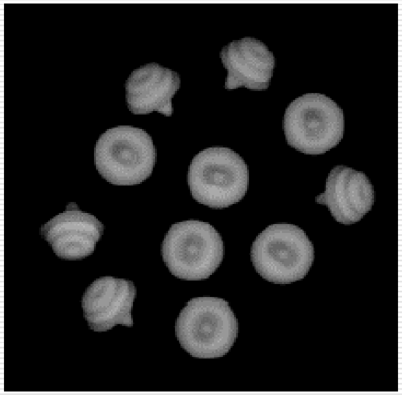


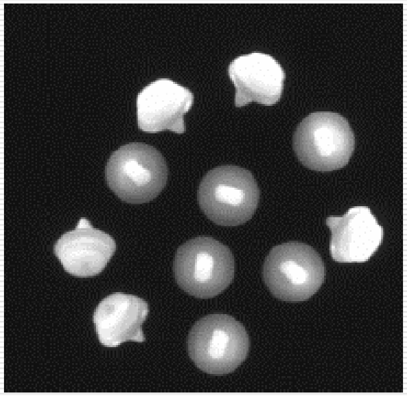
# 4. Robot Integration in Manufacturing: Merged GVR – AVI Tasks

*X-Y* flipping mechanism reorients parts for stable states – Structure.



*X-Y* flipping mechanism reorients parts for stable states – Results.



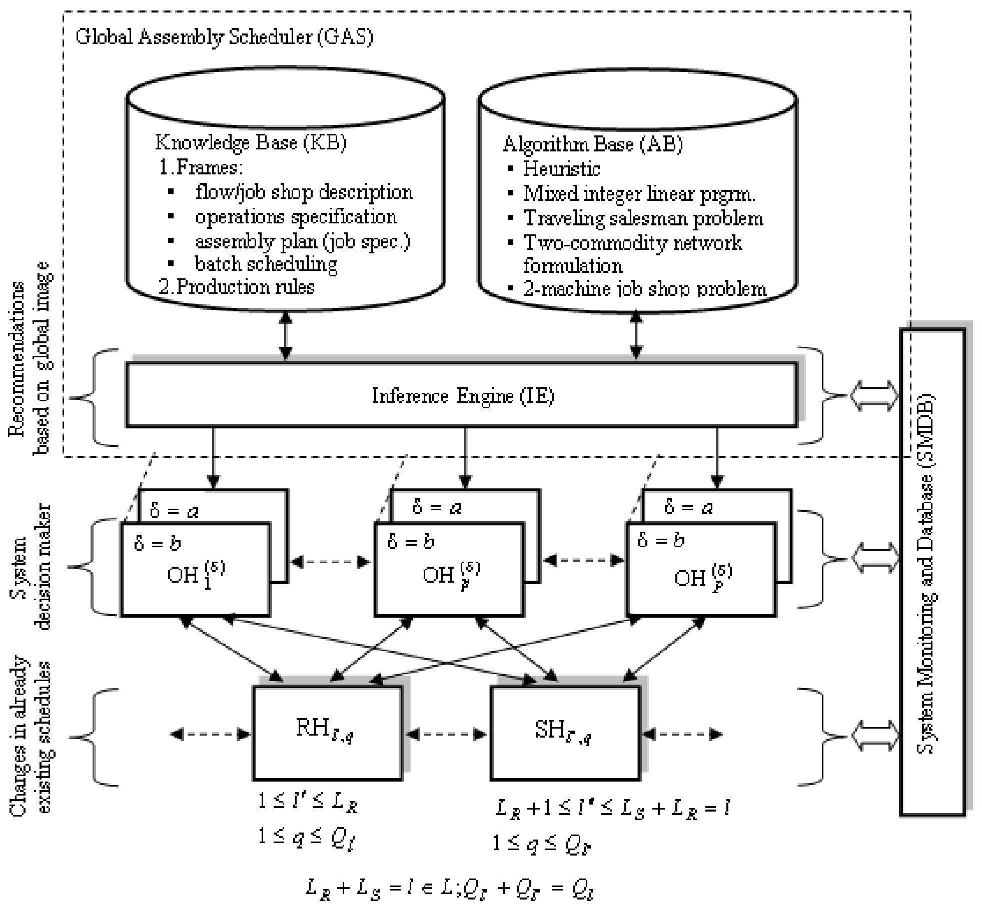


# 5. New Paradigms in Manufacturing Control: the Holonic Approach

## A Knowledge-based Holonic Control Architecture for Job Shop Robotized Assembly

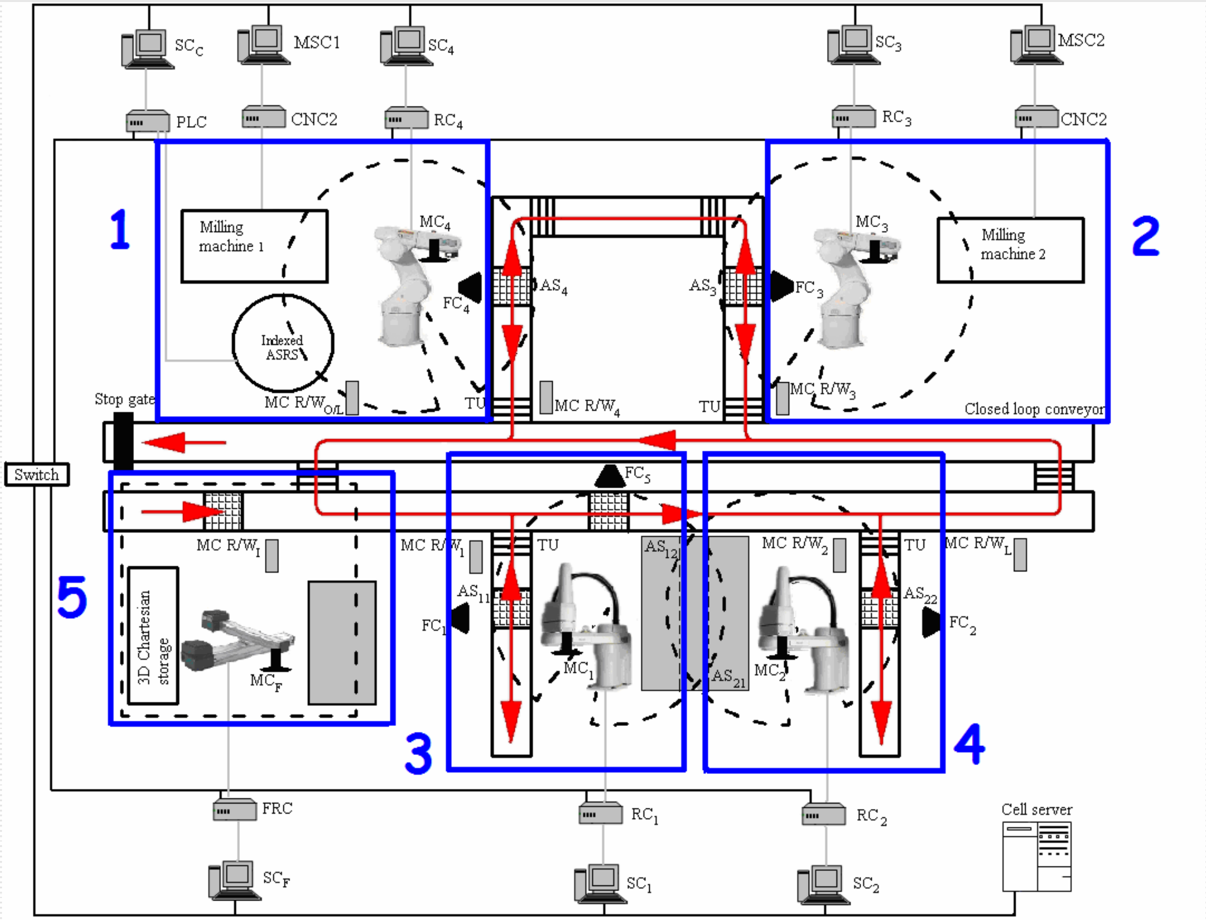
**The HOLARCHY:**

*Order* Holons
*Resource* Holons
*Product* Holons
*Expertise* Holons



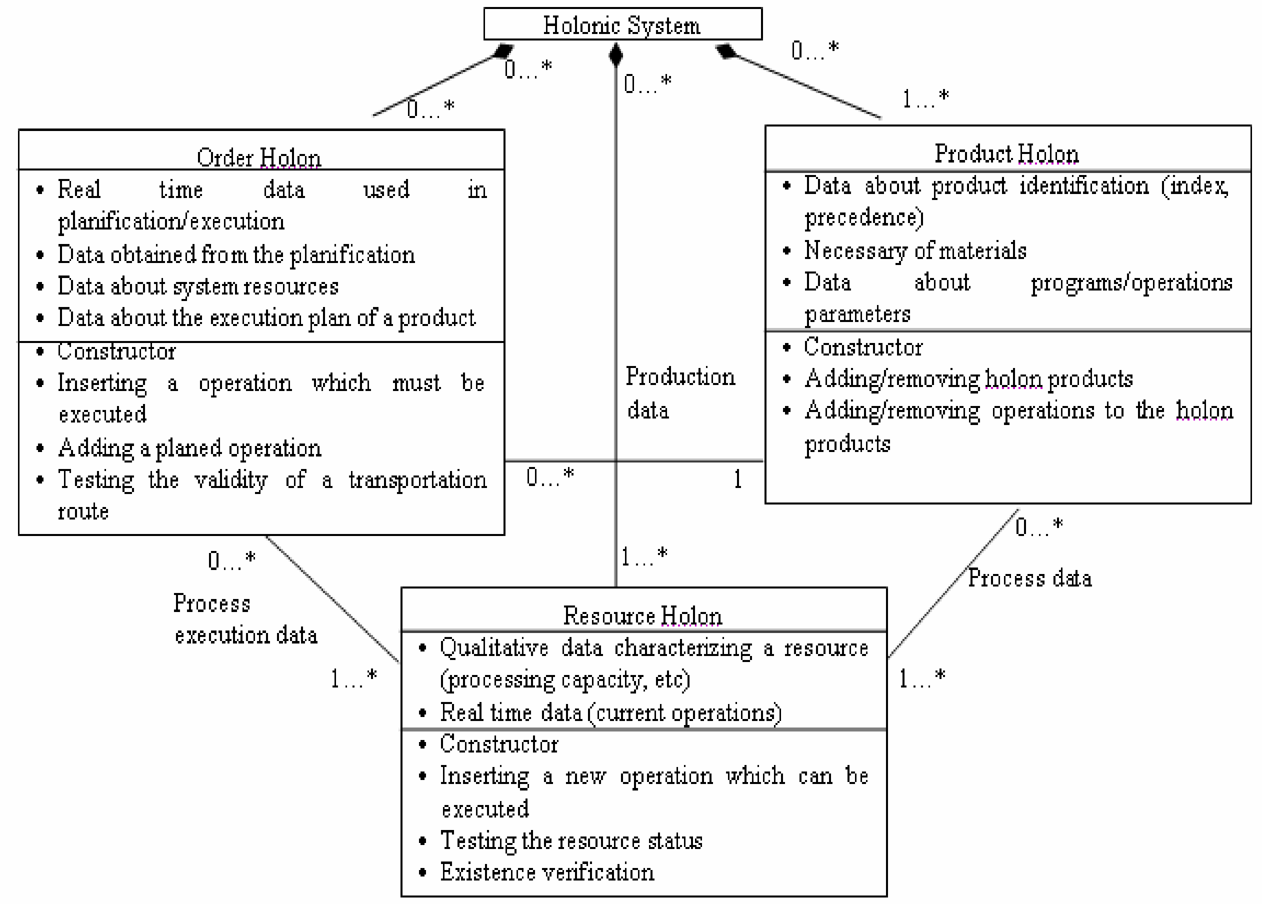
# 5. New Paradigms in Manufacturing Control: the Holonic Approach

Job Shop Manufacturing
with multiple networked
robot vision stations



# 5. New Paradigms in Manufacturing Control: the Holonic Approach

*Data and functions* embedded in the basic types of holons: **product**, **resource** and **order**



Order Holon
- Real time data used in planification/execution
- Data obtained from the planification
- Data about system resources
- Data about the execution plan of a product
- Constructor
- Inserting a operation which must be executed
- Adding a planed operation
- Testing the validity of a transportation route

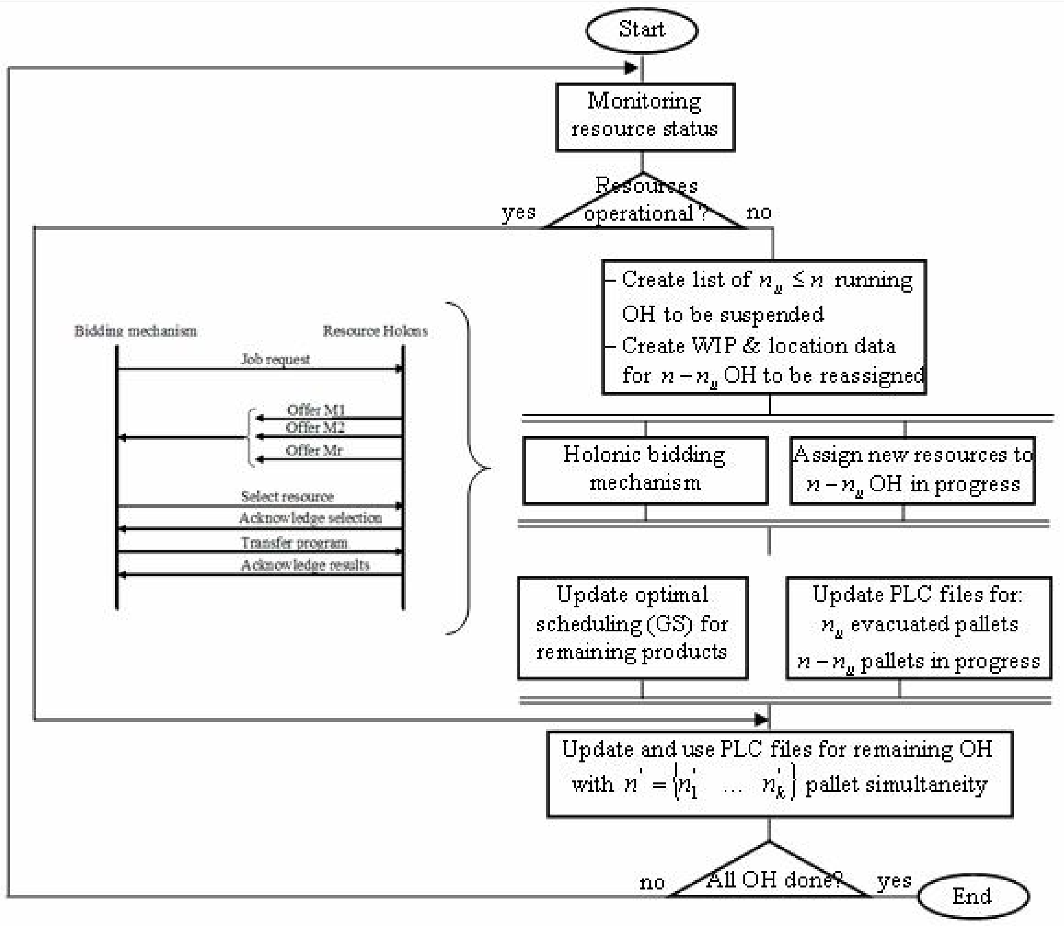
Product Holon
- Data about product identification (index, precedence)
- Necessary of materials
- Data about programs/operations parameters
- Constructor
- Adding/removing holon products
- Adding/removing operations to the holon products

Resource Holon
- Qualitative data characterizing a resource (processing capacity, etc)
- Real time data (current operations)
- Constructor
- Inserting a new operation which can be executed
- Testing the resource status
- Existence verification

# 5. New Paradigms in Manufacturing Control: the Holonic Approach

**The real time holonic production control mechanism:**

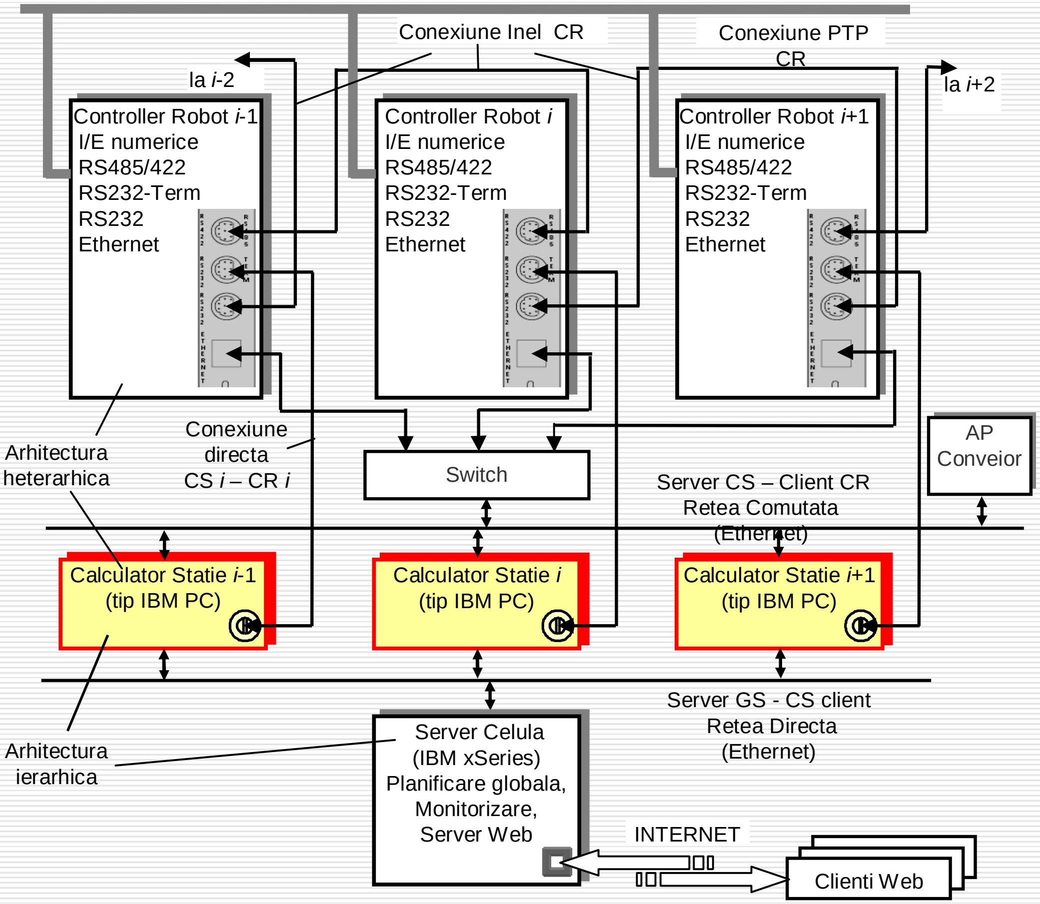
- ➢ Offer request
- ➢ Bidding for offer
- ➢ Awarding order
- ➢ Executing Order



# 5. New Paradigms in Manufacturing Control: the Holonic Approach

**Fault tolerant Control and Communication of networked robots:**

- Switched Ethernet
- Serial Direct
- Ring Inter Controller
- Point-to-Point I/O



## The End

# Thank you !