# Industrial Automation

## (Automação de Processos Industriais)

## CAD/CAM and CNC

http://users.isr.ist.utl.pt/~jag/courses/api1213/api1213.html

Slides 2010/2011 Prof. Paulo Jorge Oliveira
Rev. 2011-2015 Prof. José Gaspar

# Syllabus:

**Chap. 4 - GRAFCET *(Sequential Function Chart)* [1 weeks]**

…

**Chap. 5 – CAD/CAM and CNC [1 week]**

Methodology CAD/CAM. Types of CNC machines.

Interpolation for trajectory generation.

Integration in Flexible Fabrication Cells.

…

**Chap. 6 – Discrete Event Systems [2 weeks]**

# Some pointers to CAD/CAM and CNC

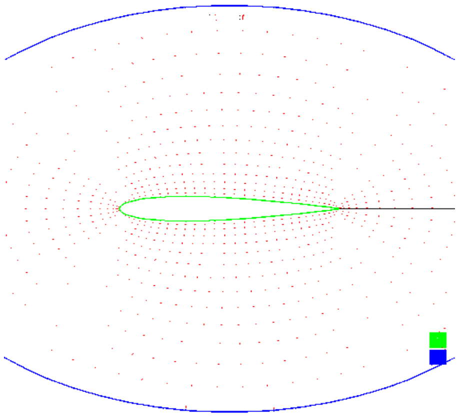
History: http://users.bergen.org/jdefalco/CNC/history.html

Tutorial:
http://users.bergen.org/jdefalco/CNC/index.html
http://www-me.mit.edu/Lectures/MachineTools/outline.html
http://www.tarleton.edu/~gmollick/3503/lectures.htm

Editors (CAD):
http://www.cncezpro.com/
http://www.cadstd.com/
http://www.turbocad.com
http://www.deskam.com/
http://www.cadopia.com/

Bibliography:
* **Computer Control of Manufacturing Systems**, Yoram Koren, McGraw Hill, 1986.
* **The CNC Workbook : An Introduction to Computer Numerical Control** by Frank Nanfarra, et al.

# CAD/CAM and CNC



Concept

Tool / Methodology

Prototype

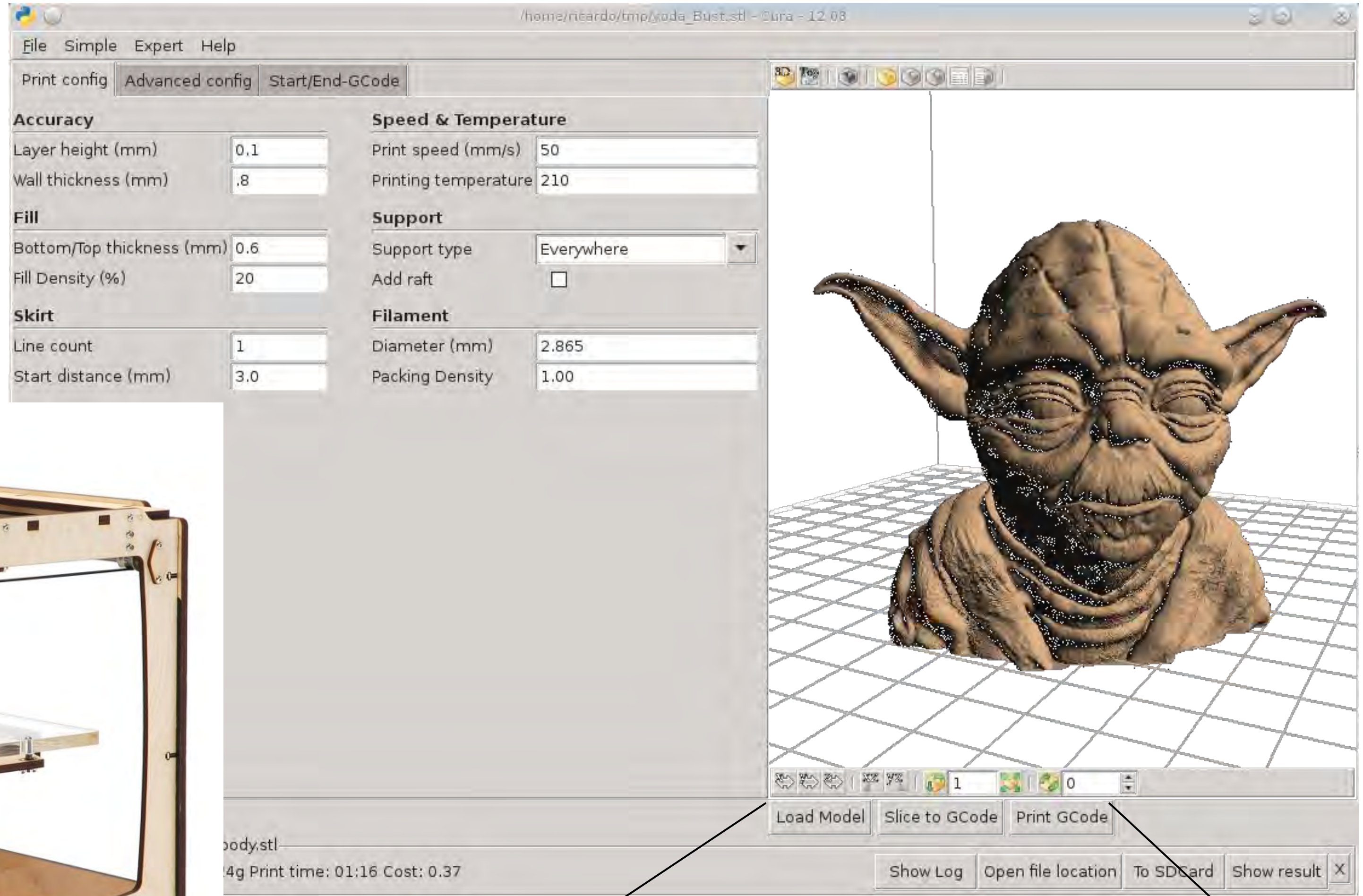
*Nowadays, machines are almost perfect! the technological question is mostly about* ***integration.***

# CAD/CAM and CNC at home!

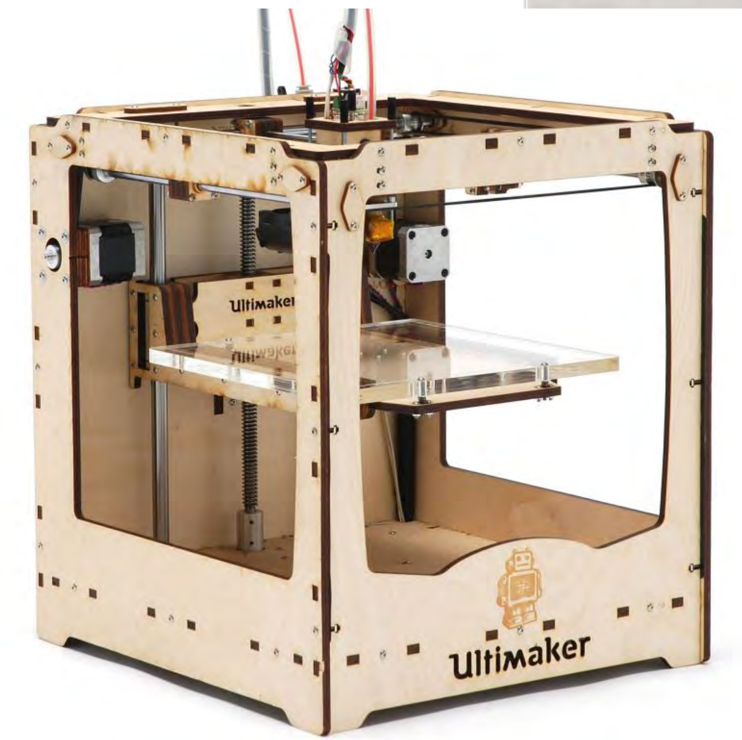
http://daid.github.com/Cura/

*Order in the internet, receive by mail and assemble yourself!*

http://www.ultimaker.com/







# Brief relevant history

## NC

1947 – US Air Force needs lead John *Parsons* to develop a machine able to produce parts described in 3D.

1949 – Contract with *Parsons Corporation* to implement to proposed method.

1952 – Demonstration at MIT of a working machine tool (NC), able to produce parts resorting to simultaneous interpolation on several axes.

1955 – First NC machine tools reach the market.

1957 - NC starts to be accepted as a solution in industrial applications , with first machines starting to produce.

197x – Profiting from the microprocessor invention appears the CNC.

*Footnotes:*

*1939-1945 – Second World War, 1947-1991 – Cold war;*
*1968 – Bedford/GM PLC, 1975-1979 – GRAFCET*

# Evolution in brief

## CAD/CAM and CNC

Modification of existing machine tools with **motion sensors** and **automatic advance** systems.

**Closed-loop** control systems for **axis control**.

Incorporation of the **computational advances** in the CNC machines.

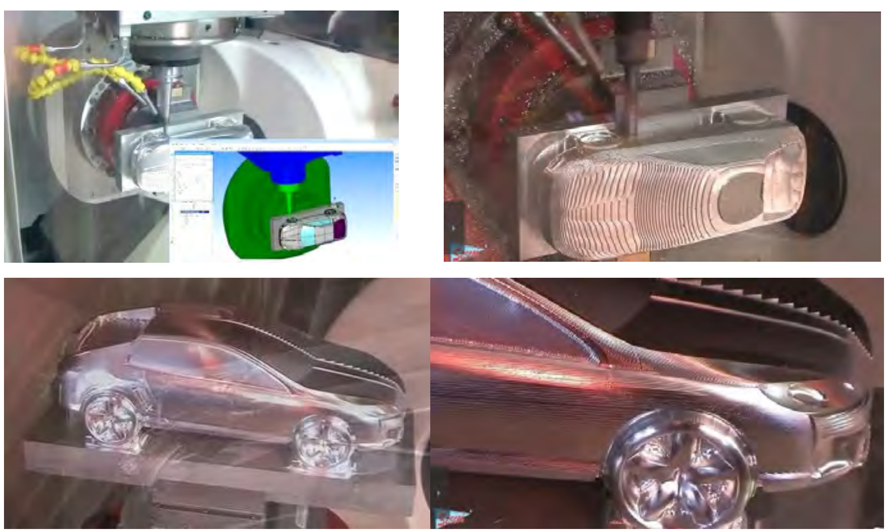
Development of **high accuracy interpolation** algorithms to trajectory interpolation.

Resort to **CAD systems to design parts** and to manage the use of CNC machines.

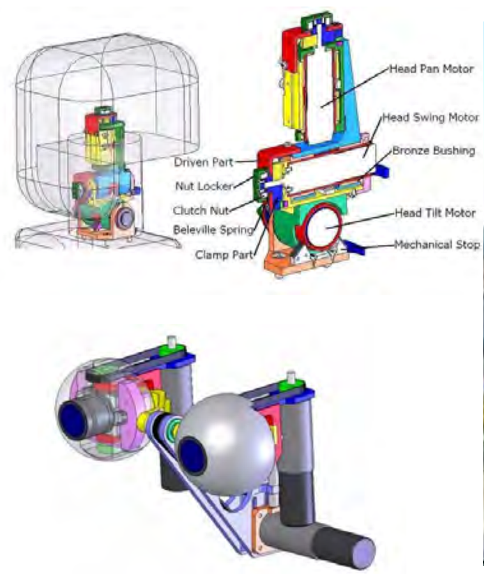
# CAD/CAM and CNC

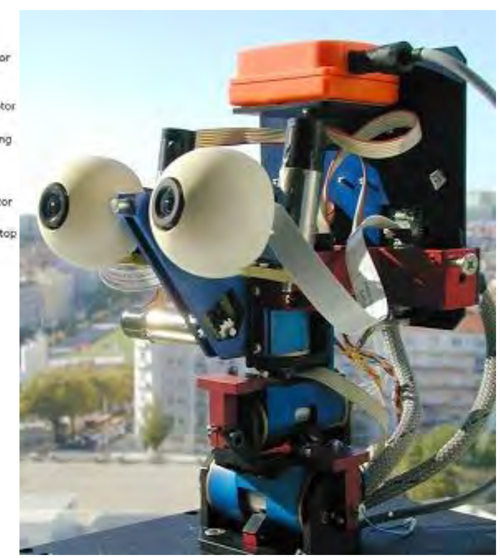
## Industrial areas of application:

- *Aerospace* *e.g. designing and testing wing and blade profiles*
- *Automobiles* *e.g. concept car design*
- *Moulds/Dies* *e.g. bottle caps, gears, hard shell luggage*
- *Electronics* *e.g. mounting components on PCBs*
- *Machinery* *e.g. iCub*



WorkNC CAD/CAM software by Sescoi



iCub head design at IST

# CAD/CAM and CNC

## Objectives

- Increase accuracy, reliability, and ability to introduce changes/**new designs**
- Increase workload
- Reduce production costs
- Reduce waste due to errors and other human factors
- Carry out **complex tasks** (e.g. Simultaneous 3D interpolation)
- Increase precision of the produced parts.

## Advantages

- Reduce the production/delivery **time**
- Reduce **costs** associated to parts and other auxiliary
- Reduce **storage** space
- Reduce time to start production
- Reduce machining time
- Reduce time to market (on the design/redesign and production).

## Limitations

- High initial **investment** (30k€ to 1500k€)
- Specialized **maintenance** required
- Does not eliminates the human errors completely
- Requires more specialized **operators**
- Not so relevant the advantages on the production of small or very small series.

# CAD/CAM and CNC Methodology CAD/CAM

Use technical data from a ***database*** in the design and production stages.
Information on parts, materials, tools, and machines are ***integrated***.

CAD (Computer Aided Design)

Allows the design in a computer environment.

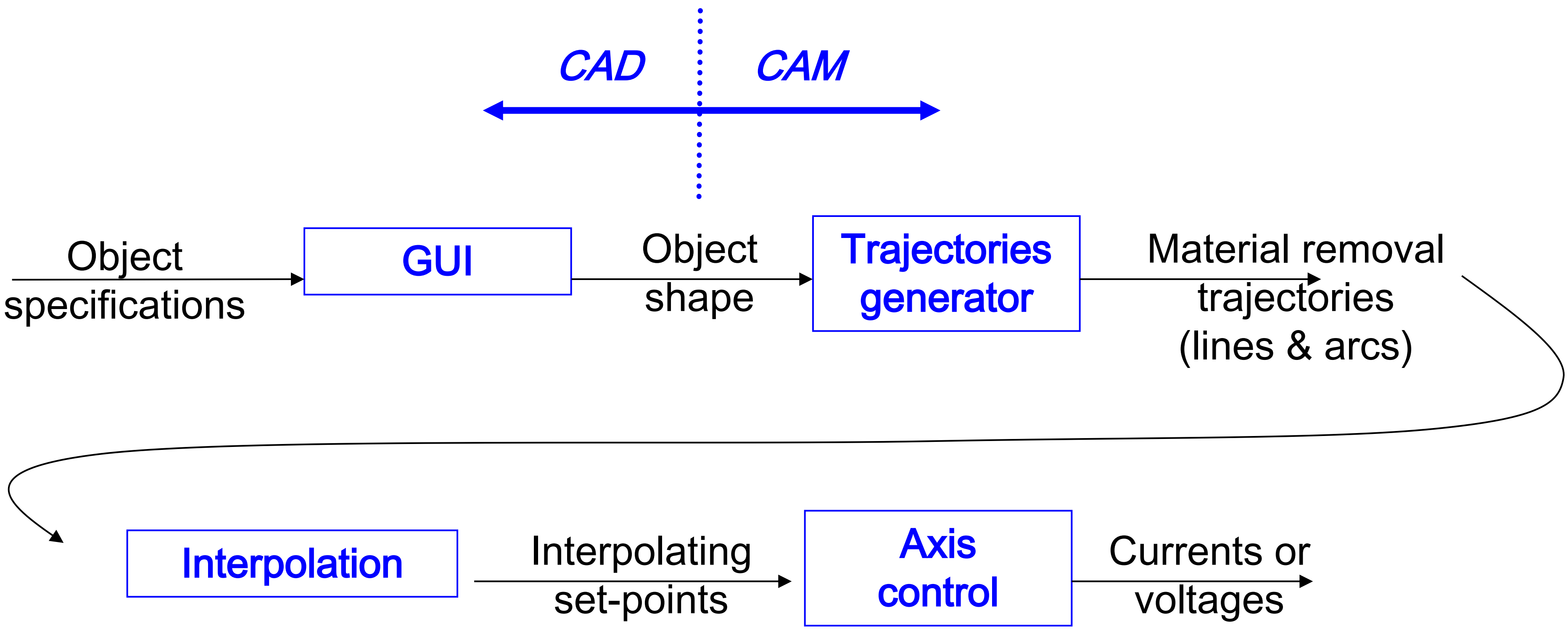
***Ideas → Design***

CAM (Computer Aided Manufacturing)

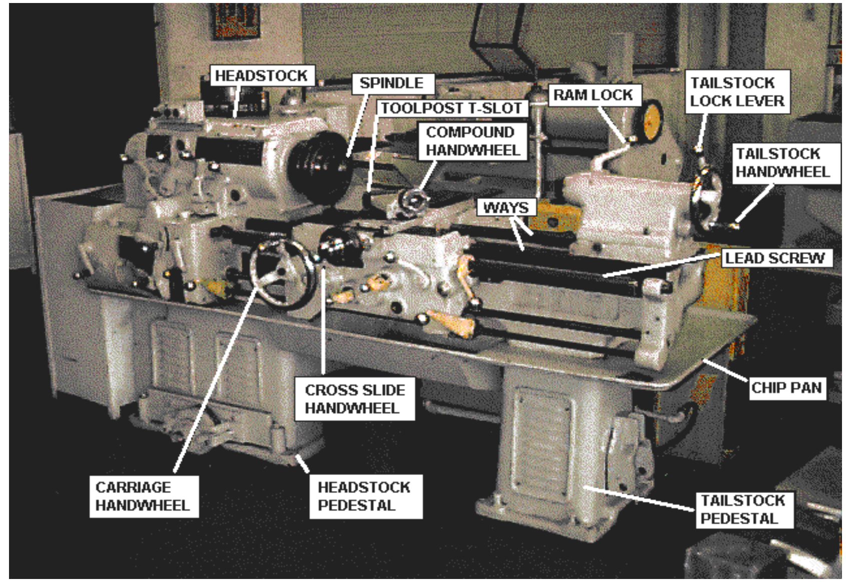
To manage programs and production stages on a computer.

***Design → Product***

# CAD/CAM and CNC Methodology CAD/CAM



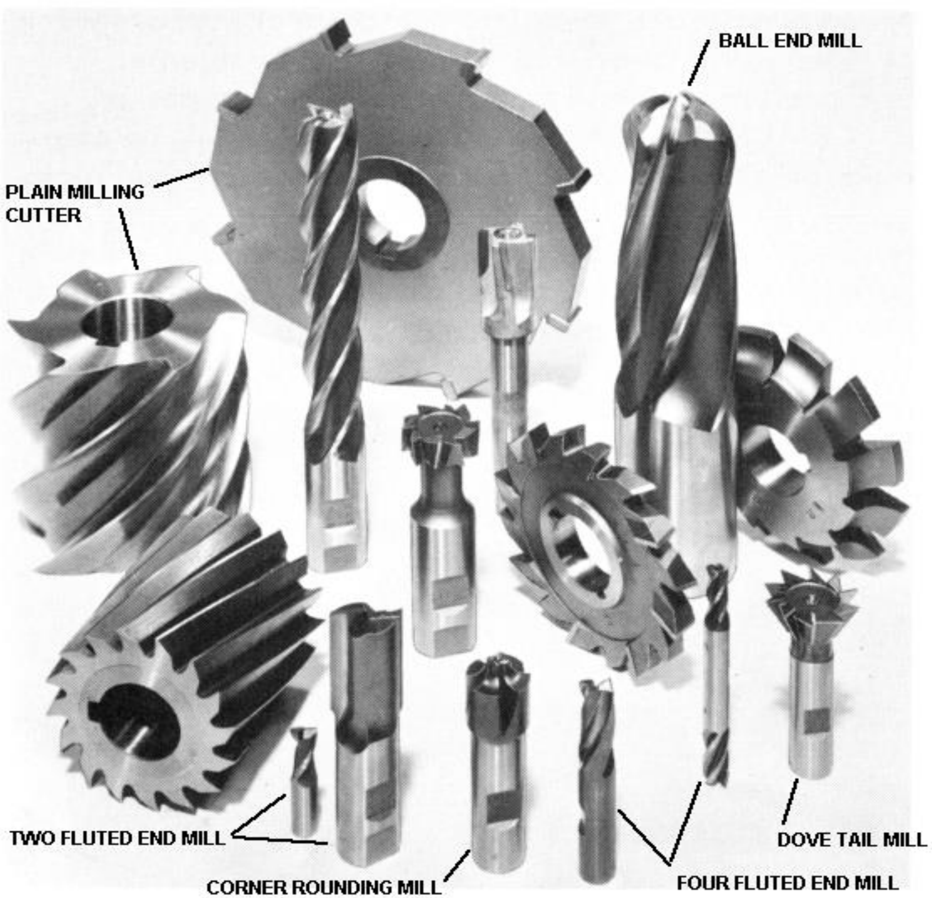

Bridgeport
TORQ-CUT
22


HEADSTOCK
SPINDLE
TOOLPOST T-SLOT
COMPOUND HANDWHEEL
RAM LOCK
TAILSTOCK LOCK LEVER
TAILSTOCK HANDWHEEL
WAYS
LEAD SCREW
CHIP PAN
CROSS SLIDE HANDWHEEL
CARRIAGE HANDWHEEL
HEADSTOCK PEDESTAL
TAILSTOCK PEDESTAL


SKIQ 20

# CAD/CAM and CNC

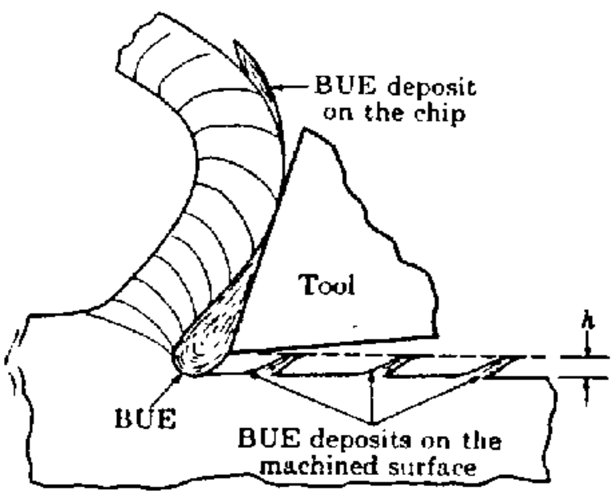
## Tools:

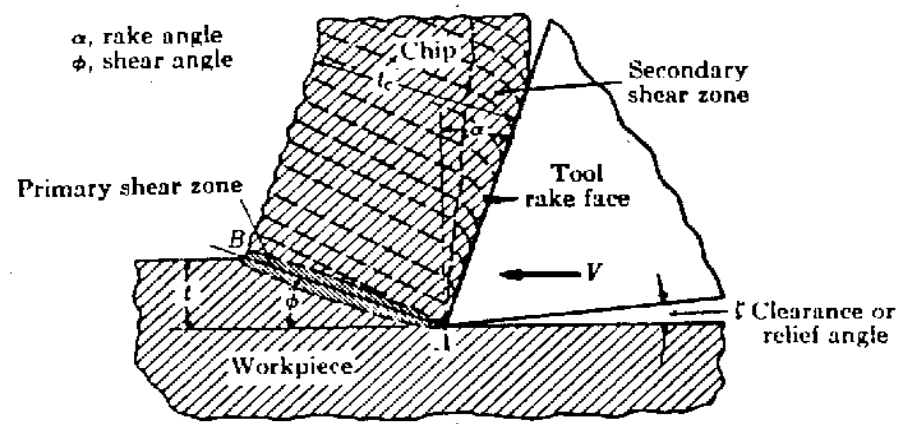



# CAD/CAM and CNC

## Tools:

**Attention to the constraints on the materials used ...**

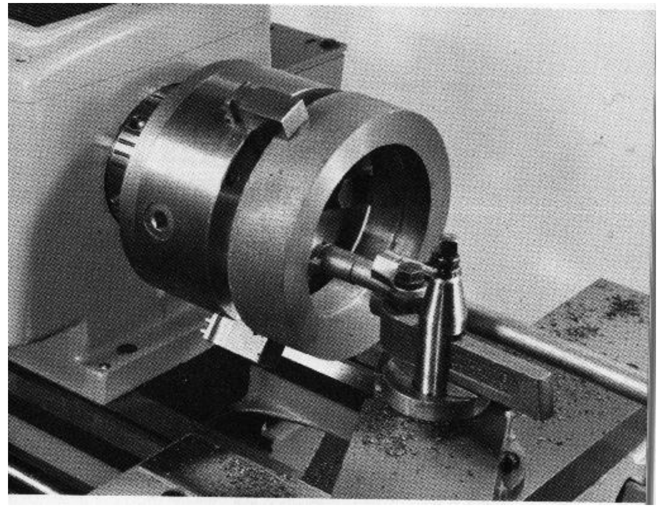


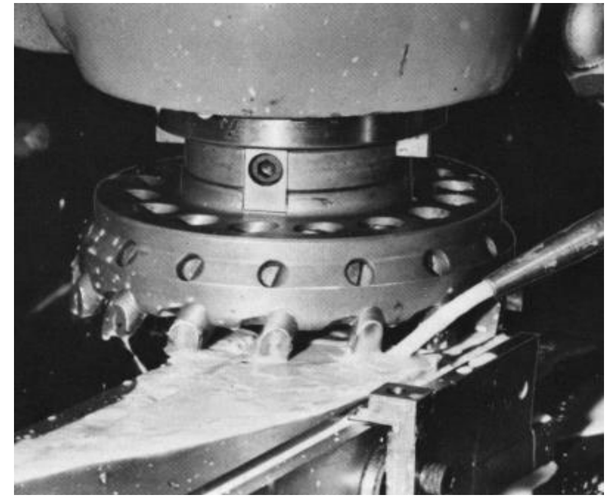



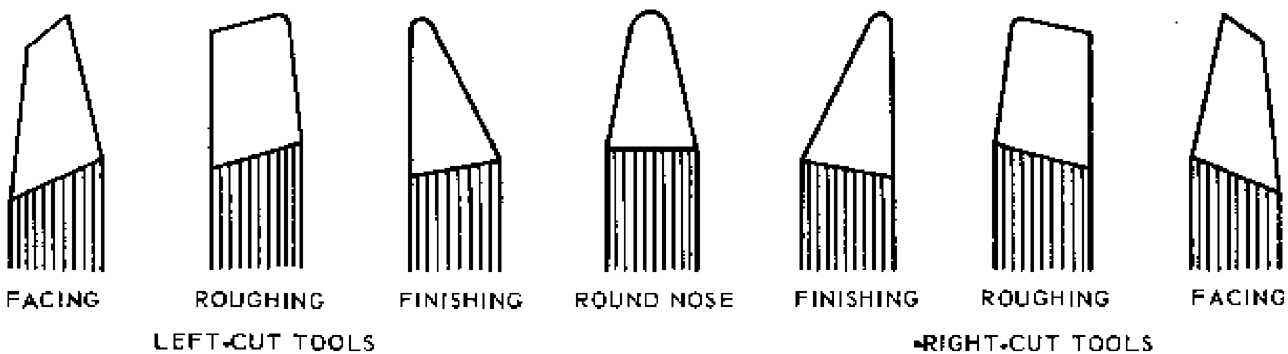

- **Speed of advance**
- **Speed of rotation**
- **Type of tool**

# CAD/CAM and CNC

## Tools:



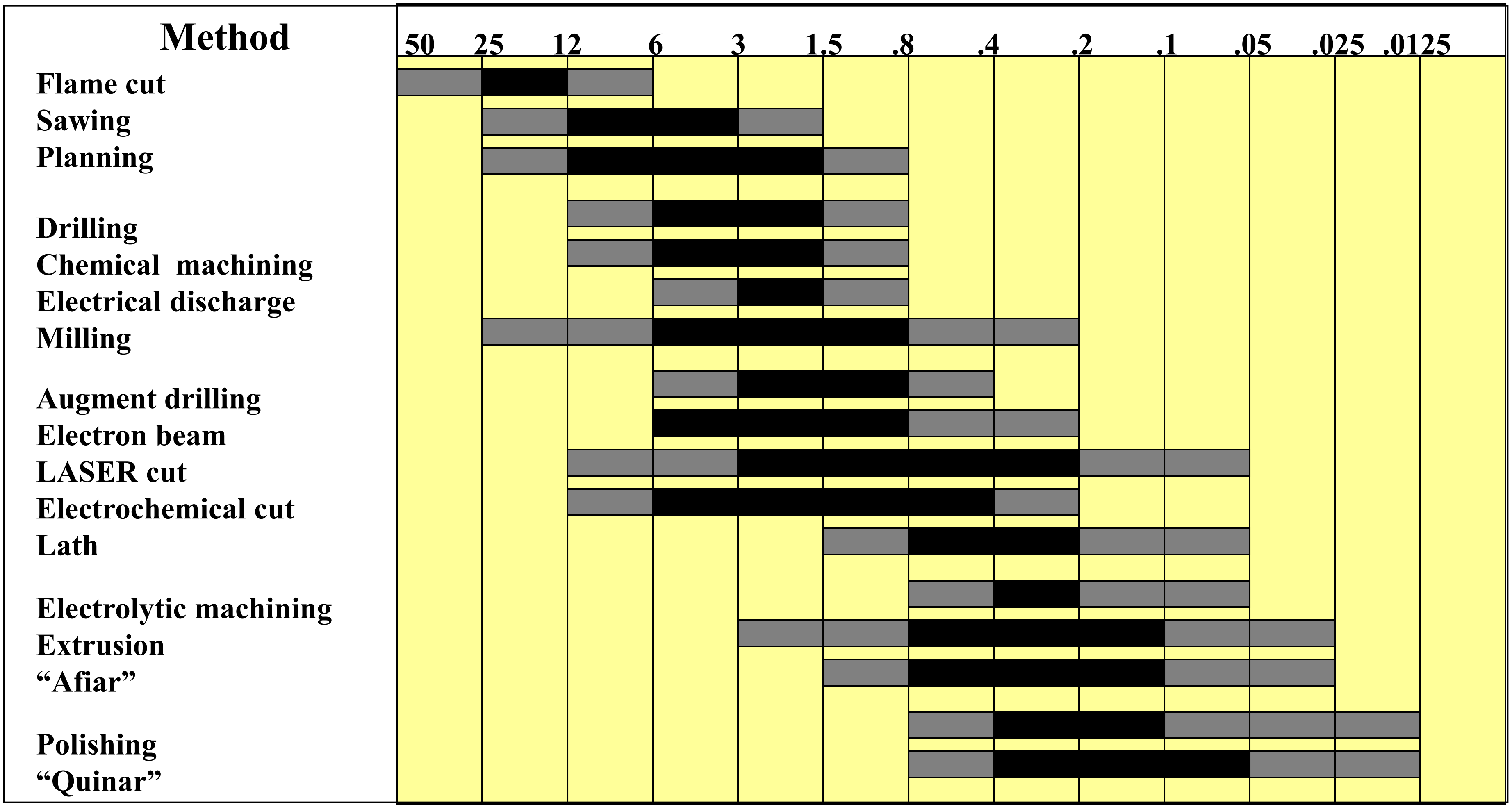





*Specific tools to perform different operations.*

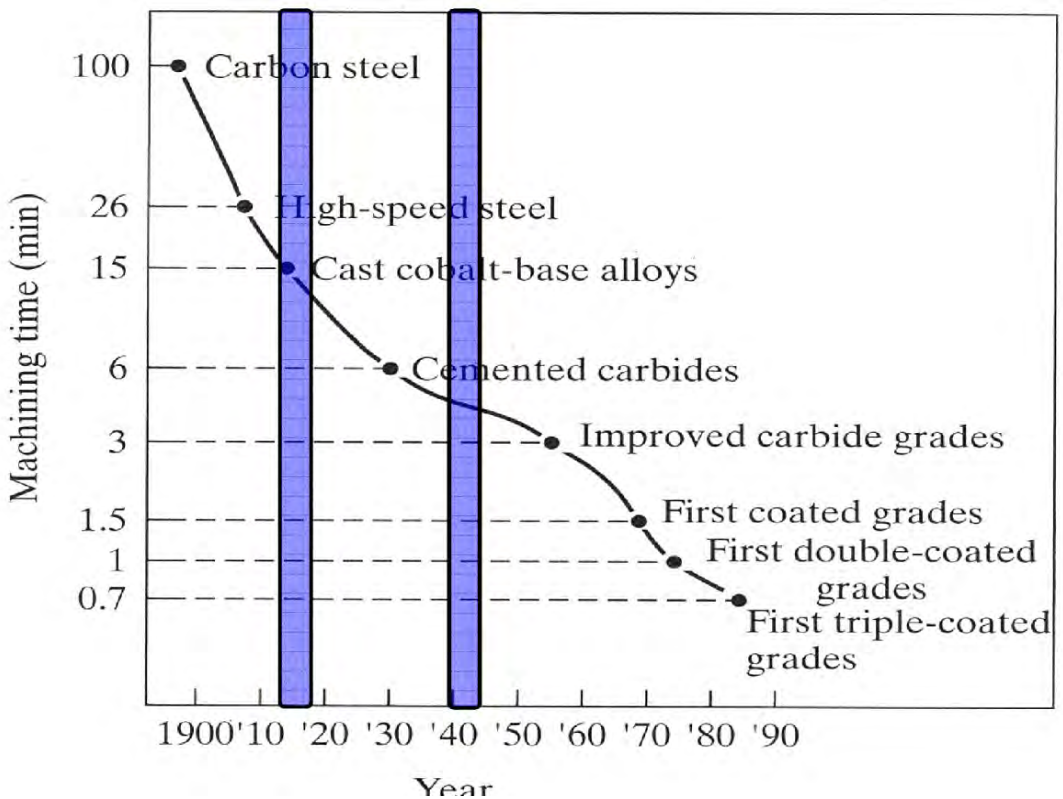
# CAD/CAM and CNC

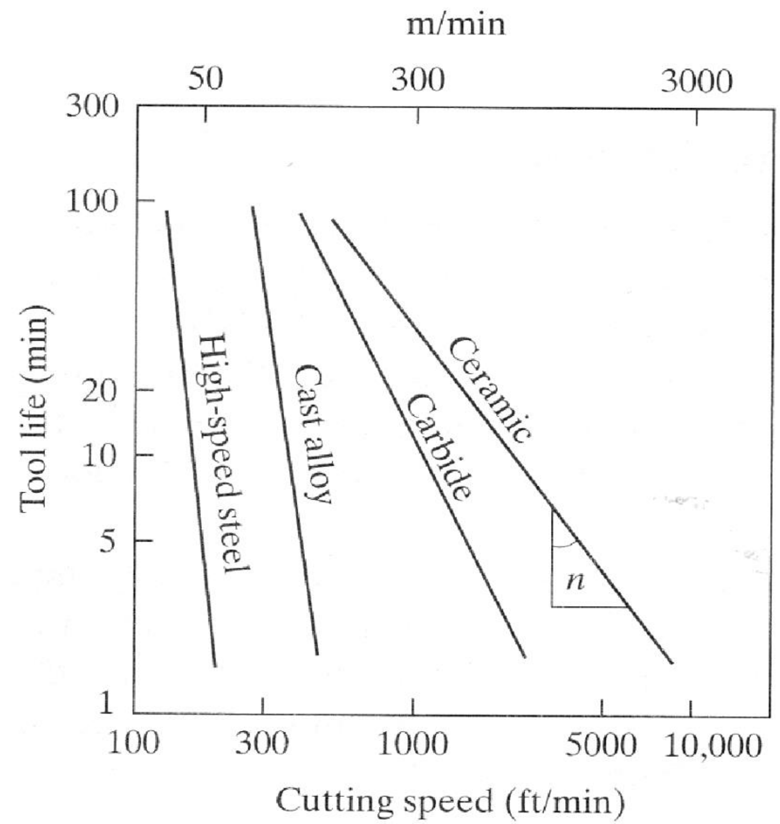
## Tools: impact on the quality of finishing (μm)



# CAD/CAM and CNC

## Evolution of tools performance:



# CAD/CAM and CNC

## Tools: Energy Requirements

Approximate Energy Requirements in Cutting Operations (at drive motor, corrected for 80% efficiency; multiply by 1.25 for dull tools).

| Material | Specific energy | |
|---|---|---|
| | $W \cdot s/mm^3$ | $hp \cdot min/in.^3$ |
| Aluminum alloys | 0.4–1.1 | 0.15–0.4 |
| Cast irons | 1.6–5.5 | 0.6–2.0 |
| Copper alloys | 1.4–3.3 | 0.5–1.2 |
| High-temperature alloys | 3.3–8.5 | 1.2–3.1 |
| Magnesium alloys | 0.4–0.6 | 0.15–0.2 |
| Nickel alloys | 4.9–6.8 | 1.8–2.5 |
| Refractory alloys | 3.8–9.6 | 1.1–3.5 |
| Stainless steels | 3.0–5.2 | 1.1–1.9 |
| Steels | 2.7–9.3 | 1.0–3.4 |

CAD/CAM and CNC

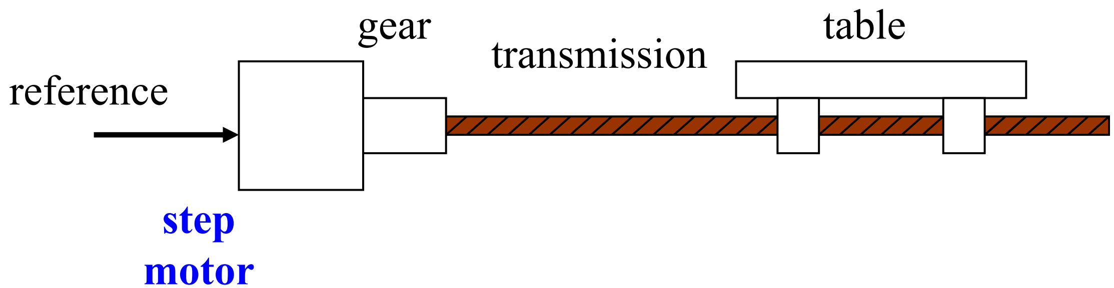
# Evolution of Numerical Control

- Numerical Control (NC)
  - Data on paper or received in serial port
  - NC machine unable to perform computations
  - Hardware interpolation
- Direct Numerical Control (DNC)
  - Central computer control a number of machines DNC or CNC
- Computer Numerical control (CNC)
  - A computer is on the core of each machine tool
  - Computation and interpolation algorithms run on the machine
- Distributive numerical control
  - Scheduling
  - Quality control
  - Remote monitoring

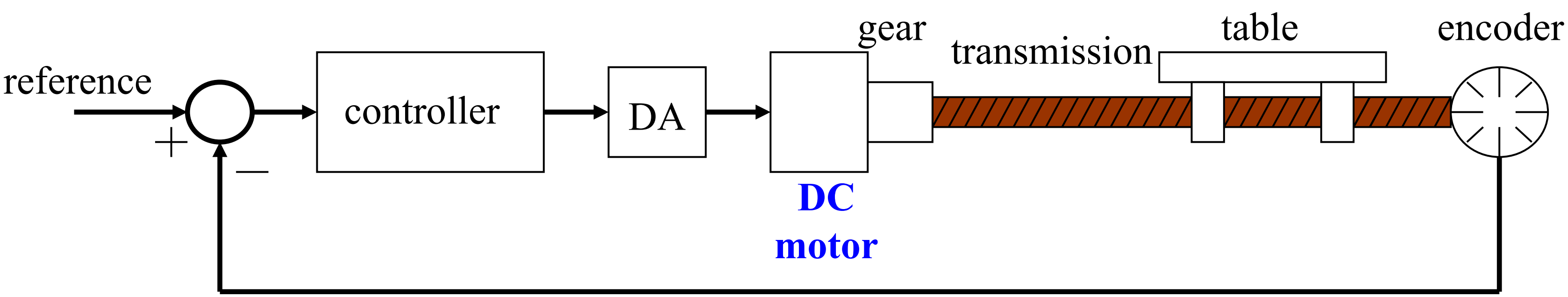
# CAD/CAM and CNC Numeric Control

## Architecture of a NC system: 1 axis

### Open-loop

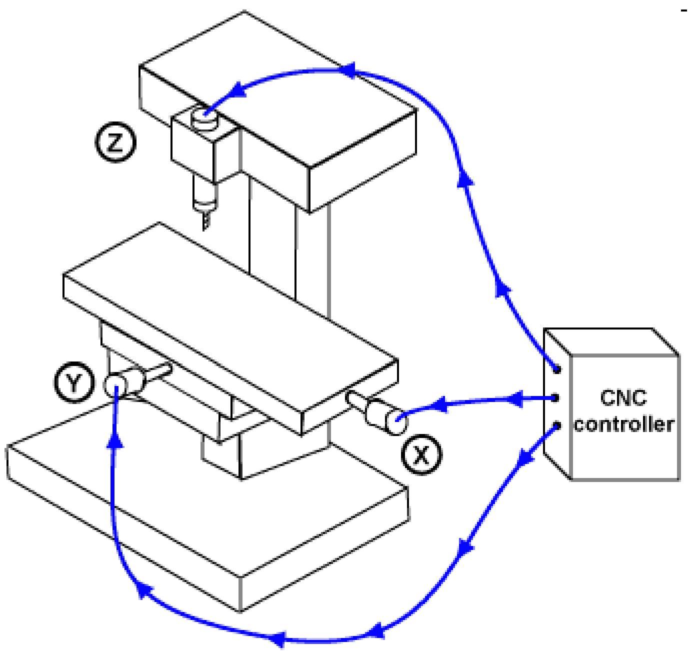



### Closed-loop

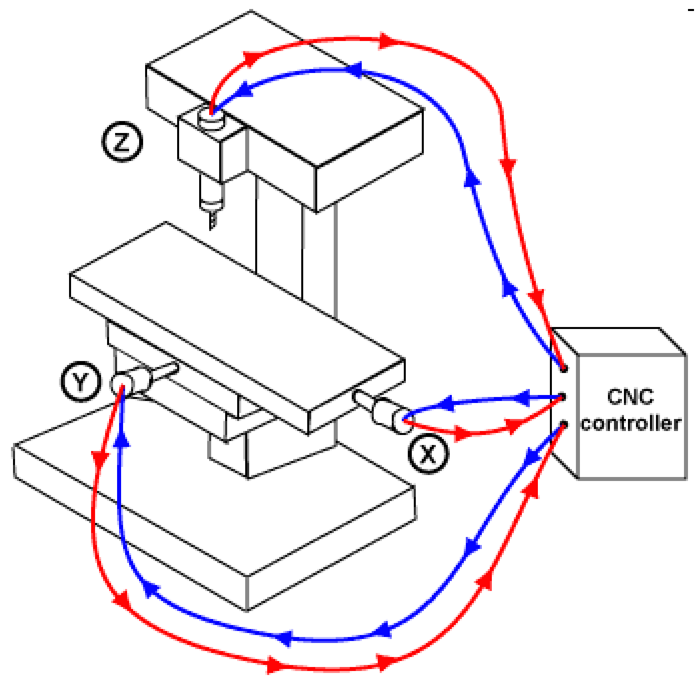


# CAD/CAM and CNC Numeric Control

## Architecture of a NC system: 3 axis




Open-Loop-System

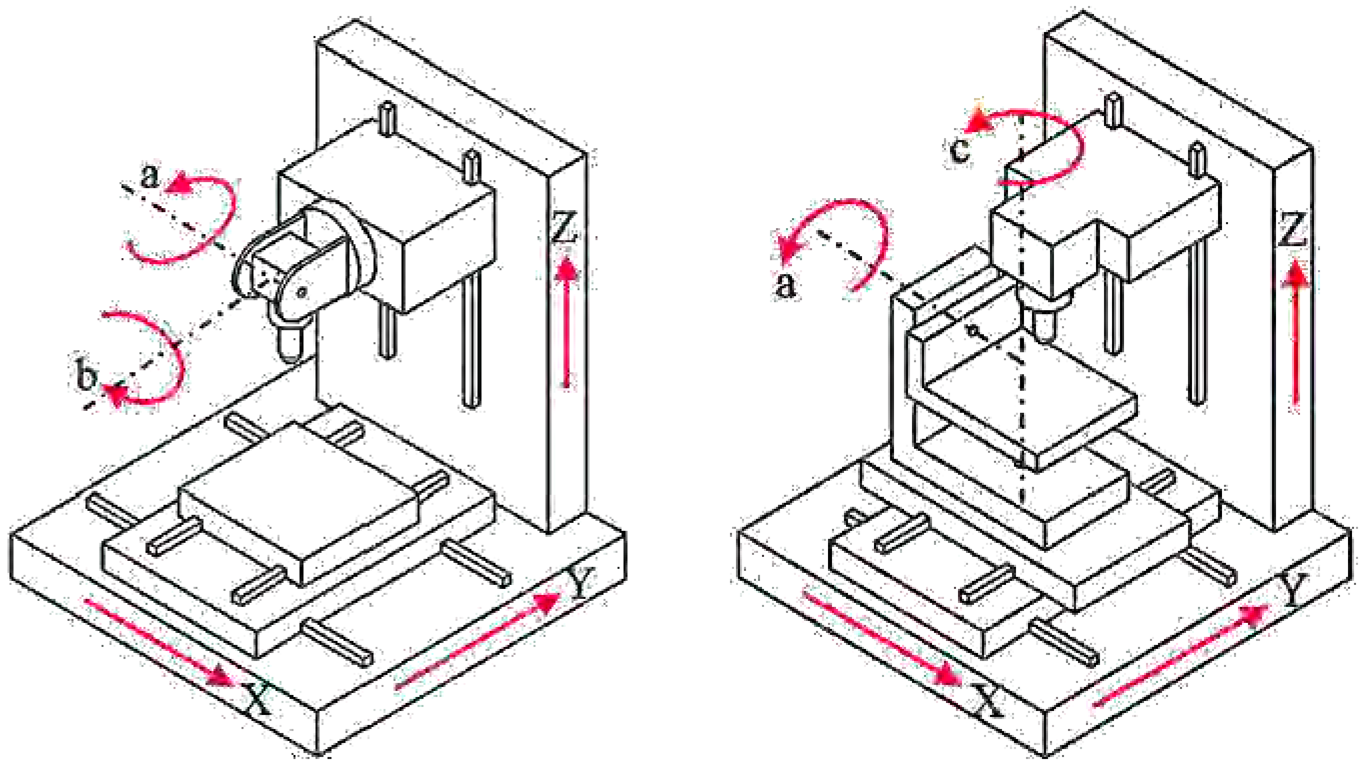



Closed-Loop-System

[http://www.kanabco.com/vms/cnc_control/cnc_control_03.html]

# CAD/CAM and CNC Numeric Control

## Architecture of a NC system: 5 axis

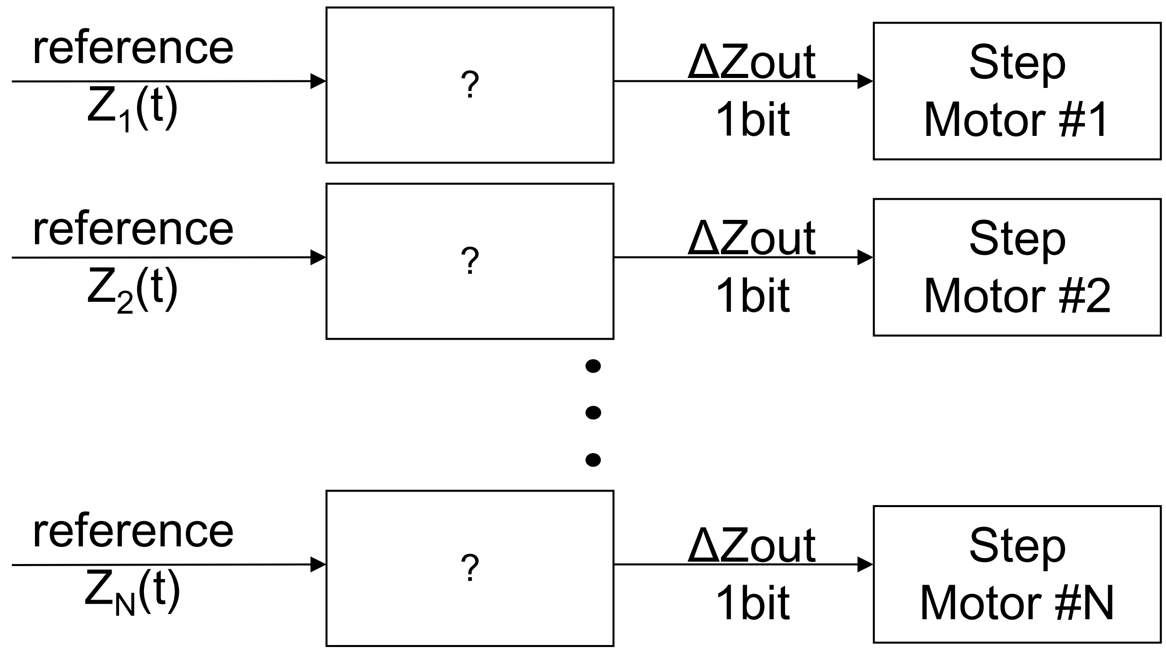



Standard configurations of the rotary axes on 5–axis CNC machines,
an ***orientable-spindle*** machine (left) and ***orientable-table*** machine (right) [Faroukia'14].

[Faroukia'14] "Inverse kinematics for optimal tool orientation control in 5-axis CNC machining", Rida T. Faroukia, Chang Yong Hanb, Shiqiao Lia, Computer Aided Geometric Design, v31n1 pp13-26 2014

# CAD/CAM and CNC

## Interpolation Motivation



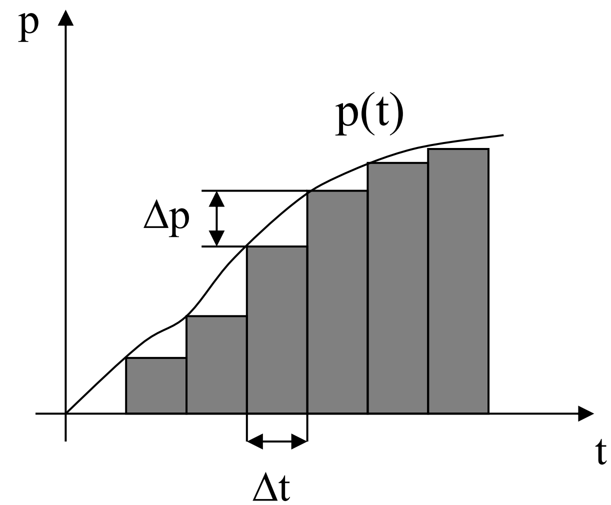

*Note1: The references are usually very **simple**, e.g. $Z_i(t)=a_i t+b_i$*

*Note2: Step motors count steps, i.e. are **numerical integrators** hence we have to convert Z(t) to an incremental representation $p_k$*

# CAD/CAM and CNC

## Interpolation: use incremental representation

*Motivation from numerical integration*




Area of a function

$$z(t) = \int_0^t p(\tau)d\tau \cong \sum_{i=1}^{k} p_i \Delta t$$

Introducing $z_k$, as the value of z at $t=k\Delta t$

$$z_k = \sum_{i=1}^{k-1} p_i \Delta t + p_k \Delta t = z_{k-1} + \Delta z_k, \qquad \Delta z_k = p_k \Delta t \qquad \Rightarrow \quad p_k = \Delta z_k / \Delta t$$

The integrator works at a rhythm of $f=1/\Delta t$ and the function p is given app. by:

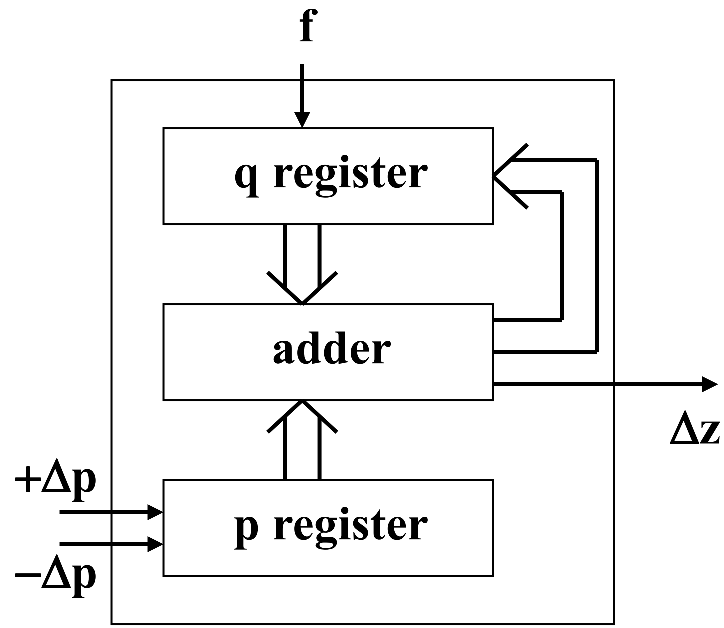
$$p_k = p_{k-1} \pm \Delta p_k$$

To be able to implement the integrator in registers with n bits, p must verify $p_k < 2^n$ .

*In the following we will use $p_k$ and $\Delta p_k$ instead of $z_k$ or $z(t)$.*

# CAD/CAM and CNC

## Implementation of a Digital Differential Analyzer (DDA)




The p register input is 0, +1= Δp or –1= –Δp.

The q register stores the **area integration** value

$$q_k = q_{k-1} + p_k.$$

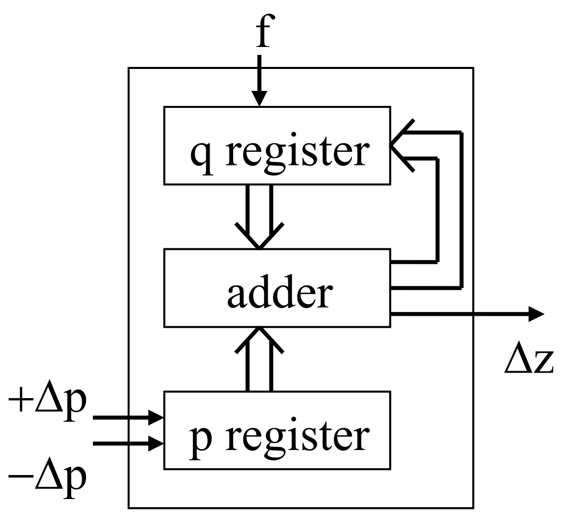
If the q register value exceeds ($2^n$-1) an overflow occurs and Δz=1:

$$\Delta z_k = 2^{-n} p_k$$

Defining C=f/$2^n$, and given that f=1/Δt, one has a scale factor from $p_k$ to $\Delta z_k$:

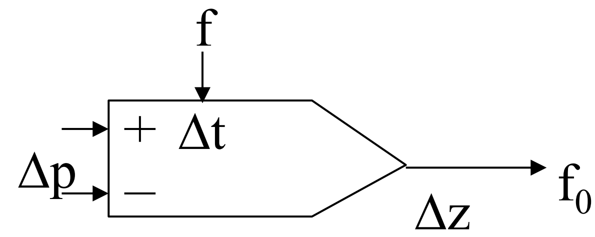
$$\Delta z_k = C p_k \Delta t$$

# CAD/CAM and CNC DDA for Linear Interpolation (1 axis):

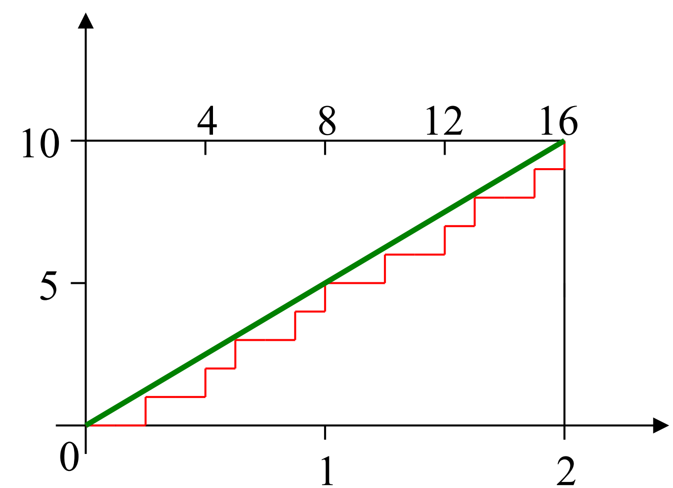



Example: let p=5, Δp=0 and assume q is a 3 bits register

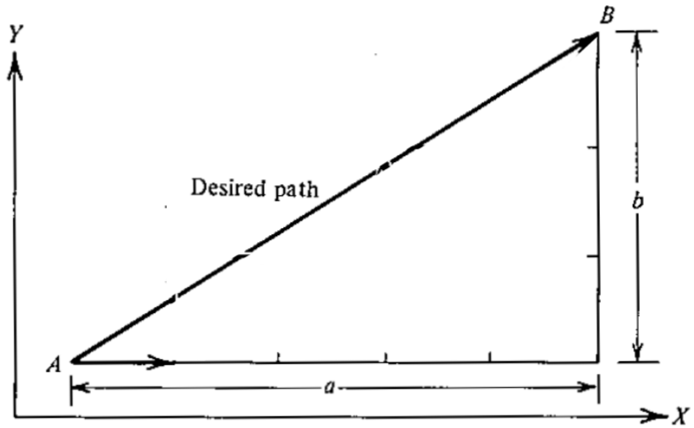
| Step | q | Δz | Σ Δz |
|---|---|---|---|
| 1 | 5 | | 0 |
| 2 | 2 | 1 | 1 |
| 3 | 7 | | 1 |
| 4 | 4 | 1 | 2 |
| 5 | 1 | 1 | 3 |
| 6 | 6 | | 3 |
| 7 | 3 | 1 | 4 |
| 8 | 0 | 1 | 5 |
| 9 | 5 | | 5 |
| | | ... | |



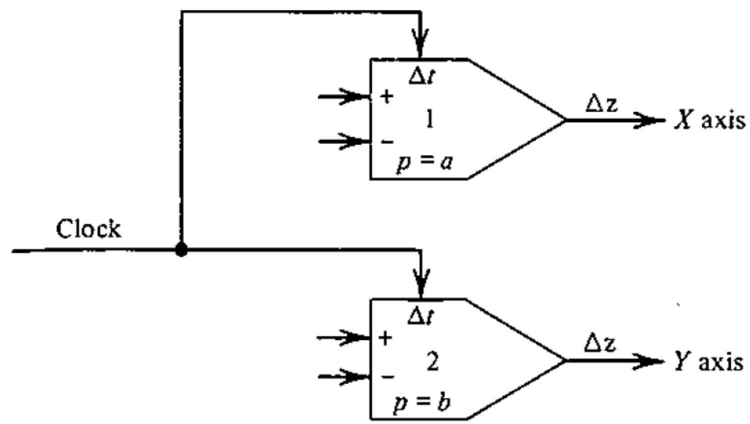

$$f_0 = \left(\frac{\Delta z}{\Delta t}\right)_k = Cp_k, \quad \text{where} \quad C = \frac{f}{2^n}$$



# CAD/CAM and CNC DDA for Linear Interpolation (2 axis):

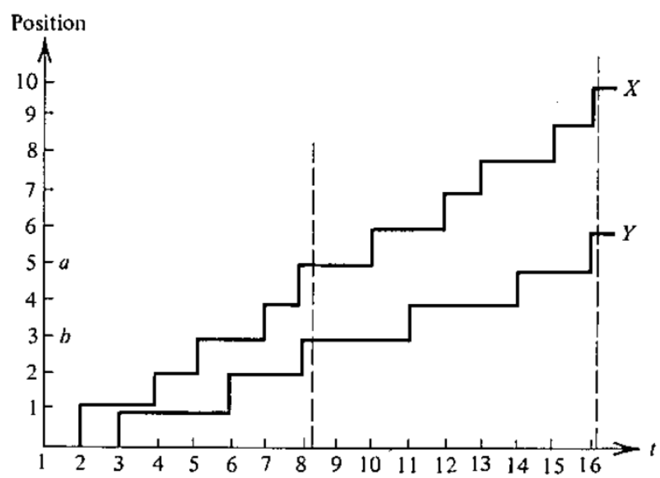


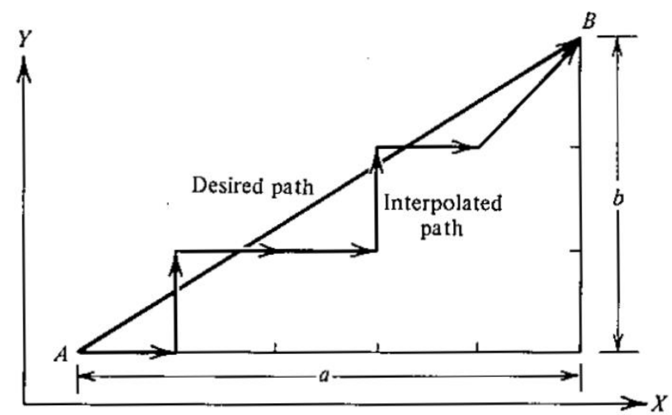

*(a) Specifications*




*(b) DDA solution*

*(c) Results*



# CAD/CAM and CNC Exponential Deceleration:

Let $p(t) = p_0 e^{-\alpha t}$ and $\frac{\Delta z}{\Delta t} = C p_k = C p_0 e^{-\alpha t}$.

The differential of p(t) is approximately

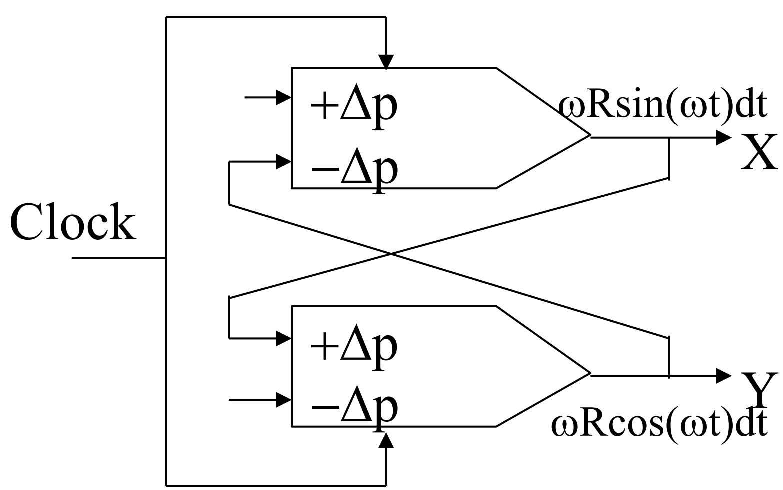
# CAD/CAM and CNC **Circular Interpolation:**

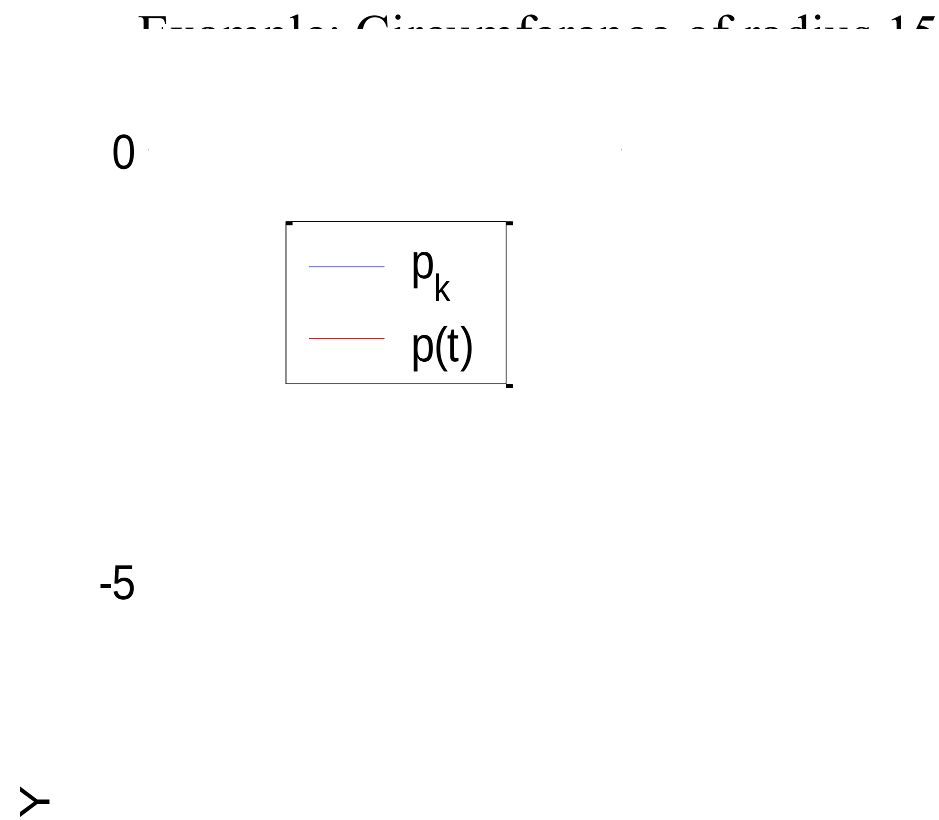
Let $(X-R)^2 + Y^2 = R^2$ or

$$X = R(1-\cos(\omega t))$$
$$Y = \quad R\sin(\omega t)$$

The differential is

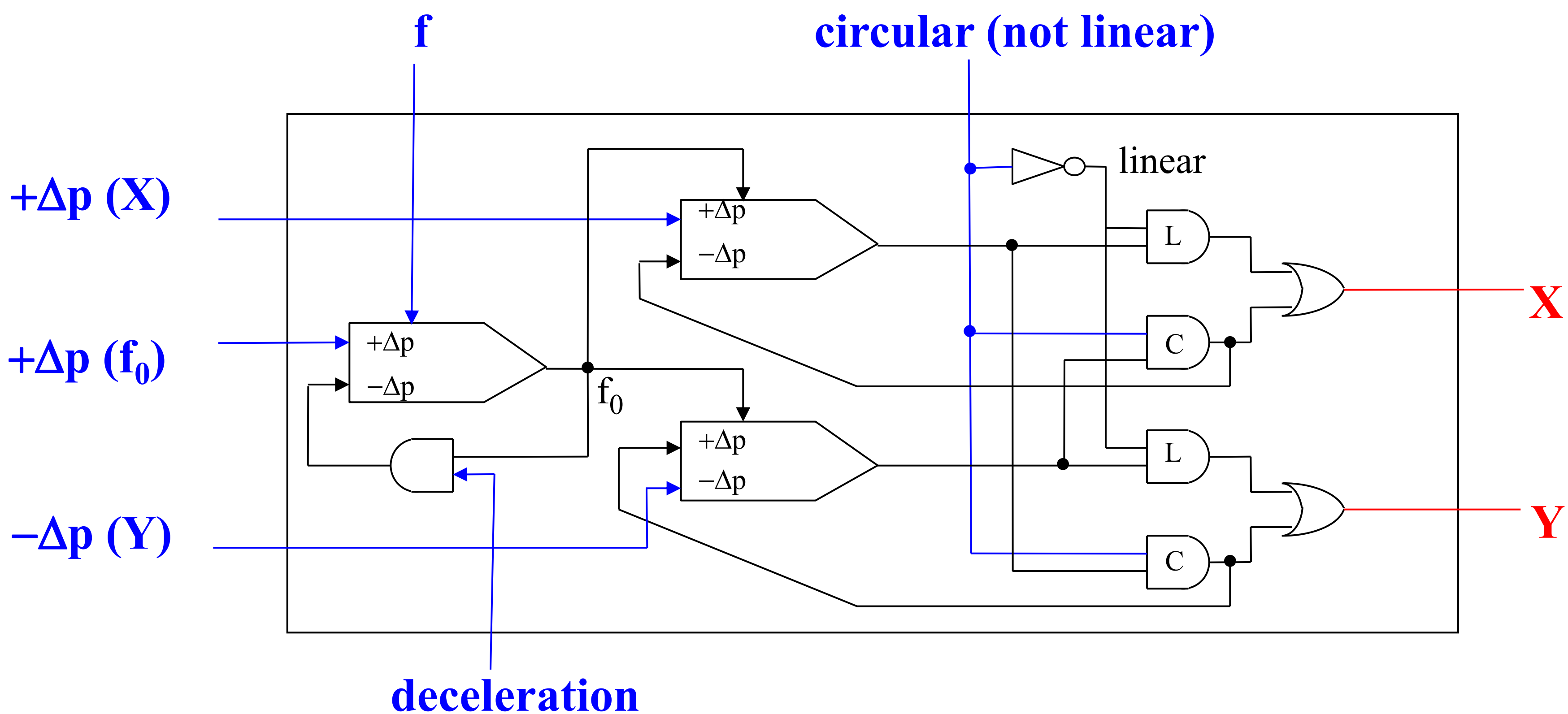
$$dX = \omega R\sin(\omega t)dt = d(-R\cos(\omega$$
$$dY = \omega R\cos(\omega t)dt = \ d(R\sin(\omega t$$



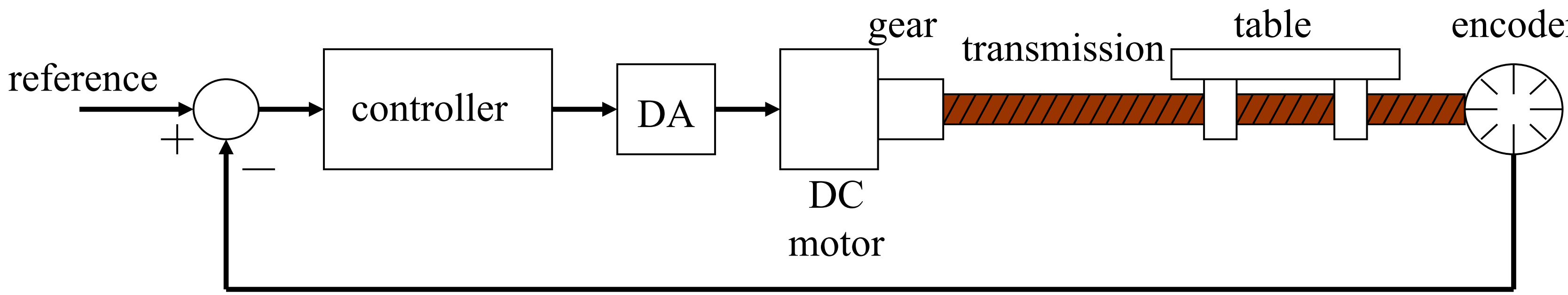
# CAD/CAM and CNC **Full DDA**

2D Line, 2D Arc, Acceleration / Deceleration

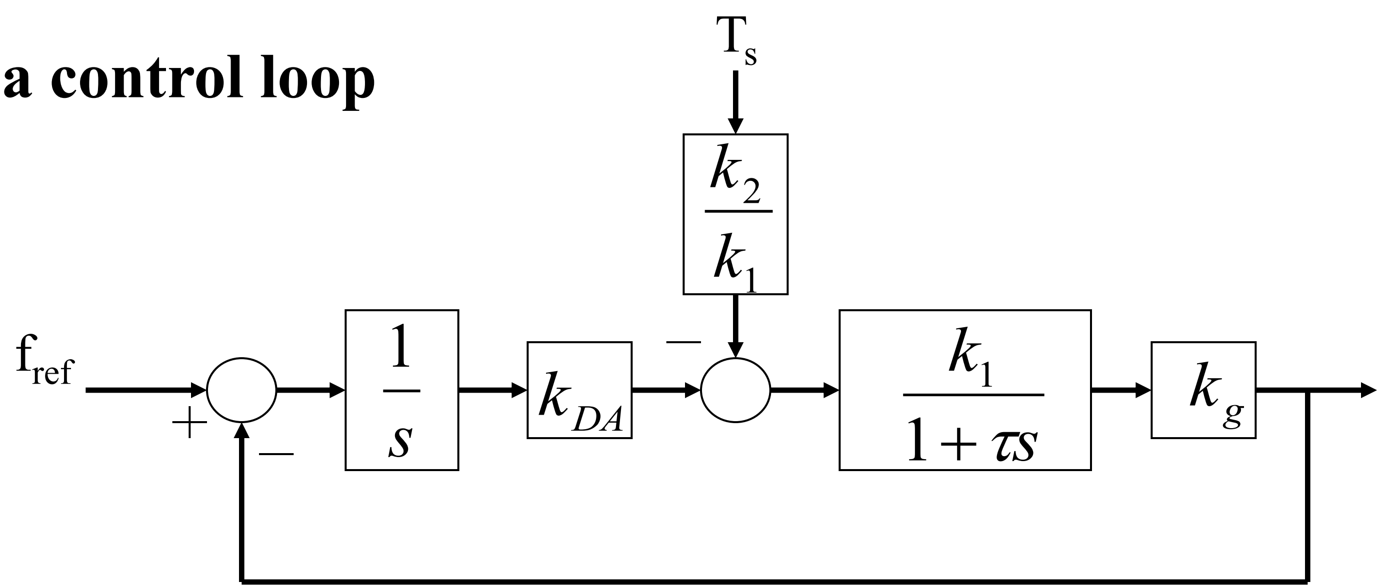


# CAD/CAM and CNC

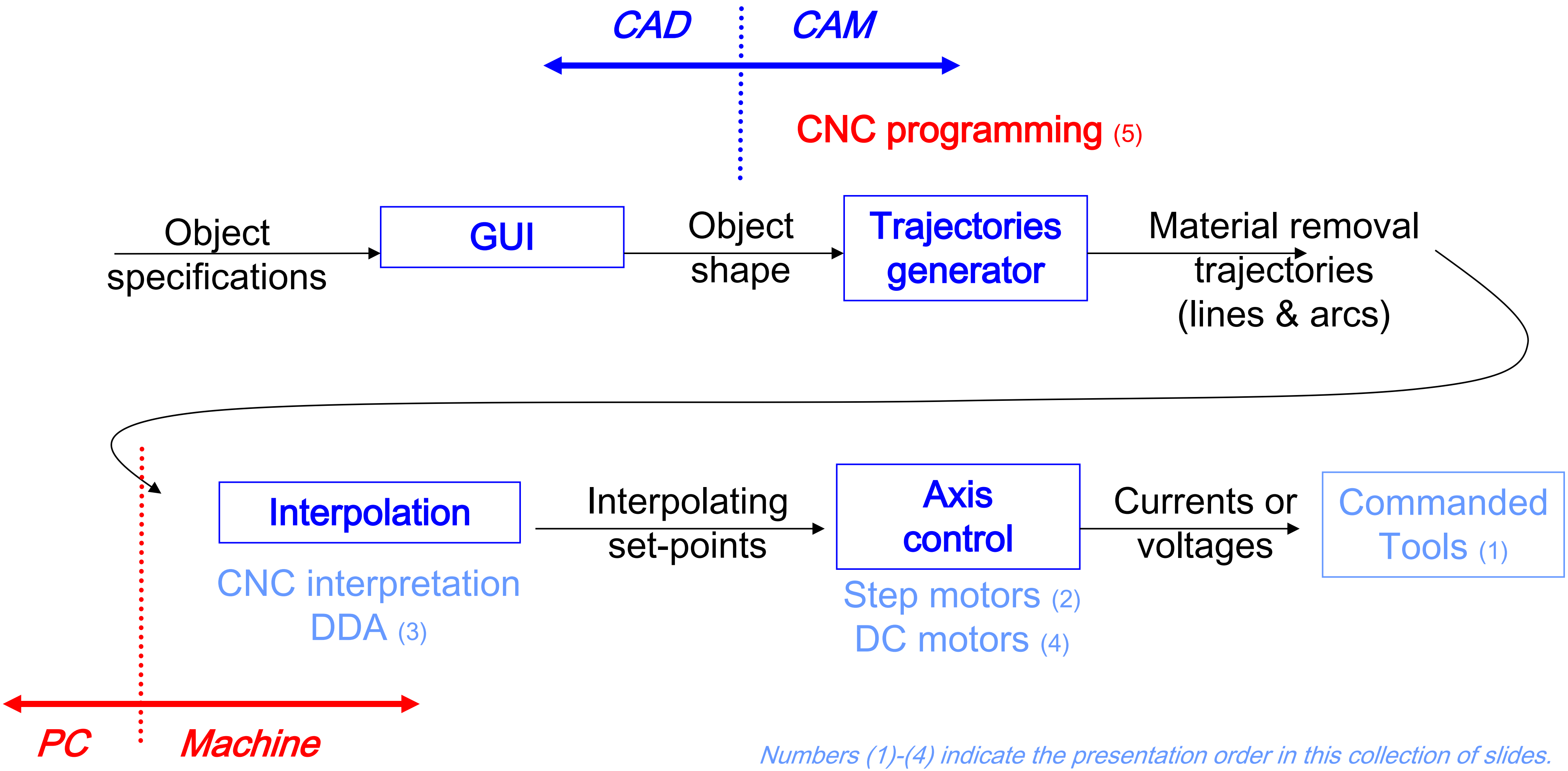
## CNC Axes Control




## Dynamics of a control loop



# Methodology CAD/CAM




*Numbers (1)-(4) indicate the presentation order in this collection of slides. In the following we introduce (5).*

# CAD/CAM and CNC - CNC Programming

*Summary of the previous slide:*

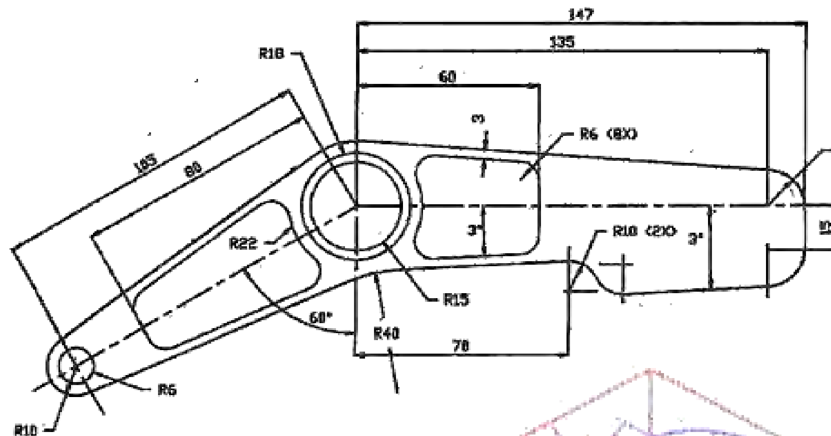
CNC machines know how to do **interpolation**, but not how to machine a complete part.

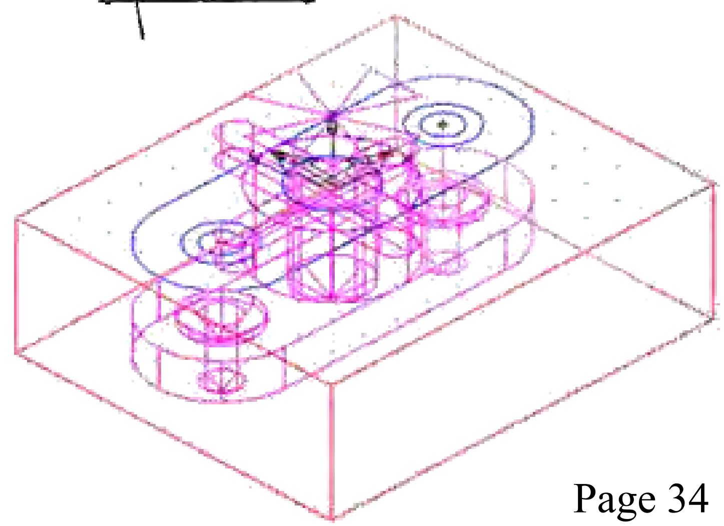
CAM helps to bridge the gap between **object shapes** and making **material removal trajectories** (to be interpolated).

In other words, one needs to do **CNC programming**.

*Steps 1, 2, ... 6, to execute a part*

**1.** Read and **interpret** the technical drawings



*In the following:* **G-code** (also RS-274), which has many variants, is the common name for the most widely used numerical control (NC) programming language.

# CAD/CAM and CNC - CNC Programming

**2.** Choose the most adequate **machine** for the several stages of machining

Relevant features:

- The workspace of a machine versus the part to be produced
- The options available on each machine
- The tools that can be used
- The mounting and the part handling
- The operations that each machine can perform

**3.** Choose of the most adequate **tools**

Relevant features:

- The material to be machined and its characteristics
- Standard tools cost less
- The quality of the mounting part is function of the number of parts to produce
- Use the right tool for the job
- Verify if there are backup tools and/or stored available
- Take into account tool aging

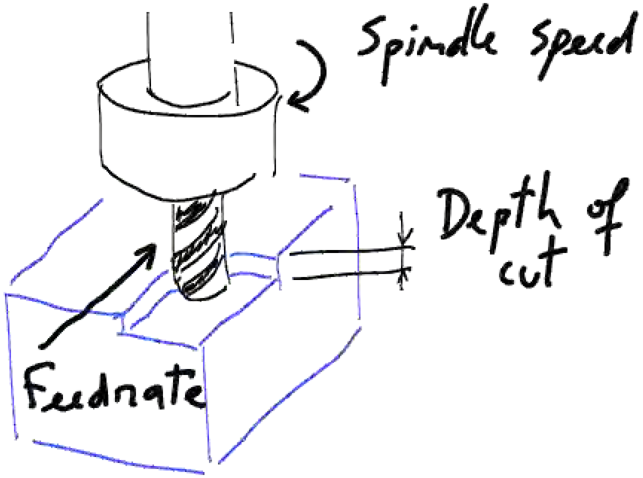
# CAD/CAM and CNC - CNC Programming

**4.** Cutting data

• Spindle Speed – speed of rotation of the cutting tool (rpm)

• Feedrate – linear velocity of advance to machine the part (mm/minute)

• Depth of Cut – depth of machining in z (mm)




**5.** Choice of the interpolation plane, in 2D ½ machines


5.1. Unit system
imperial / inches (G70) or
international millimeters (G71).

5.2. Command mode*
Absolute = use world coordinate system (G90)
Relative = move w.r.t. the current position (G91)

* There are other command modes, e.g. helicoidal.

# CAD/CAM and CNC - CNC Programming

## 6. Data Input

| | |
|---|---|
| N | Sequence Number |
| G | Preparatory Functions |
| X | X Axis Command |
| Y | Y Axis Command |
| Z | Z Axis Command |
| R | Radius from specified center |
| A | Angle ccw from +X vector |
| I | X axis arc center offset |
| J | Y axis arc center offset |
| K | Z axis arc center offset |
| F | Feed rate |
| S | Spindle speed |
| T | Tool number |
| M | Miscellaneous function |

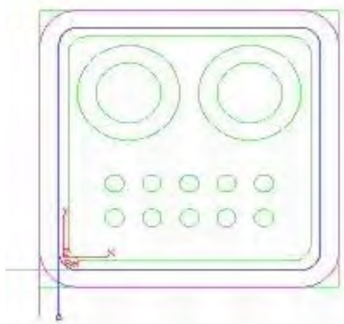
Example of a CNC program:

```
N30  G0 T1 M6
N35  S2037 M3
N40  G0 G2 X6.32 Y-0.9267 M8
N45  Z1.1
N50  Z0.12
N55  G1 Z0. F91.7
N60  X-2.82
N65  Y0.9467
N70  X6.32
N75  Y2.82
N80  X-2.82
N85  G0 Z1.1
...
```

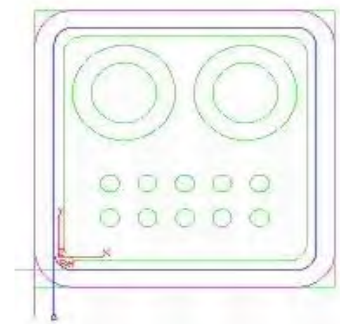
# CAD/CAM and CNC - CNC Programming

## Preparatory functions (inc.)

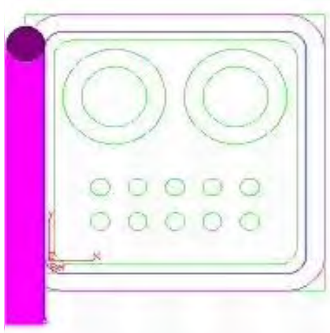
**G00 – GO**



**G01 – Linear Interpolation**



**G02 – Circular Interpolation (CW)**

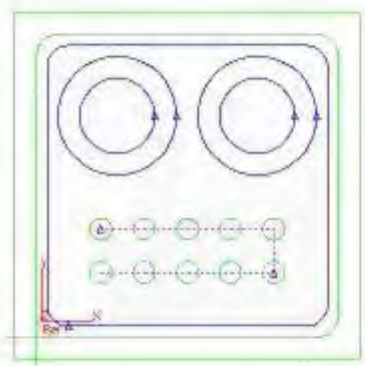


**G03 – Circular Interpolation (CCW)**

# CAD/CAM and CNC - CNC Programming

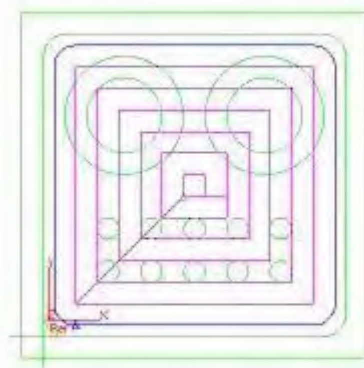
## Canned Cycles

G81 – Drilling cycle with multiple holes



## Special Cycles or Canned Cycles

G78 – Rectangular pocket cycle, used to clean a square shaped area



# CAD/CAM and CNC - CNC Programming

## Other preparatory functions

G04 - A temporary dwell, or **delay** in tool motion.
G05 - A permanent hold, or **stopping** of tool motion. It is canceled by the machine operator.
G22 - Activation of the stored **axis travel limits**, which are used to establish a safety boundary.
G23 - Deactivation of the stored axis travel limits.
G27 - Return to the machine **home** position via a programmed intermediate point
G34 - Thread cutting with an increasing lead.
G35 - Thread cutting with a decreasing lead.
G40 - Cancellation of any previously programmed tool radius compensation
G42 - Application of cutter radius compensation to the right of the workpiece with respect to the direction of tool travel.
G43 - Activation of tool length compensation in the same direction of the offset value
G71 - Canned cycle for multiple-pass turning on a lathe (foreign-made)
…

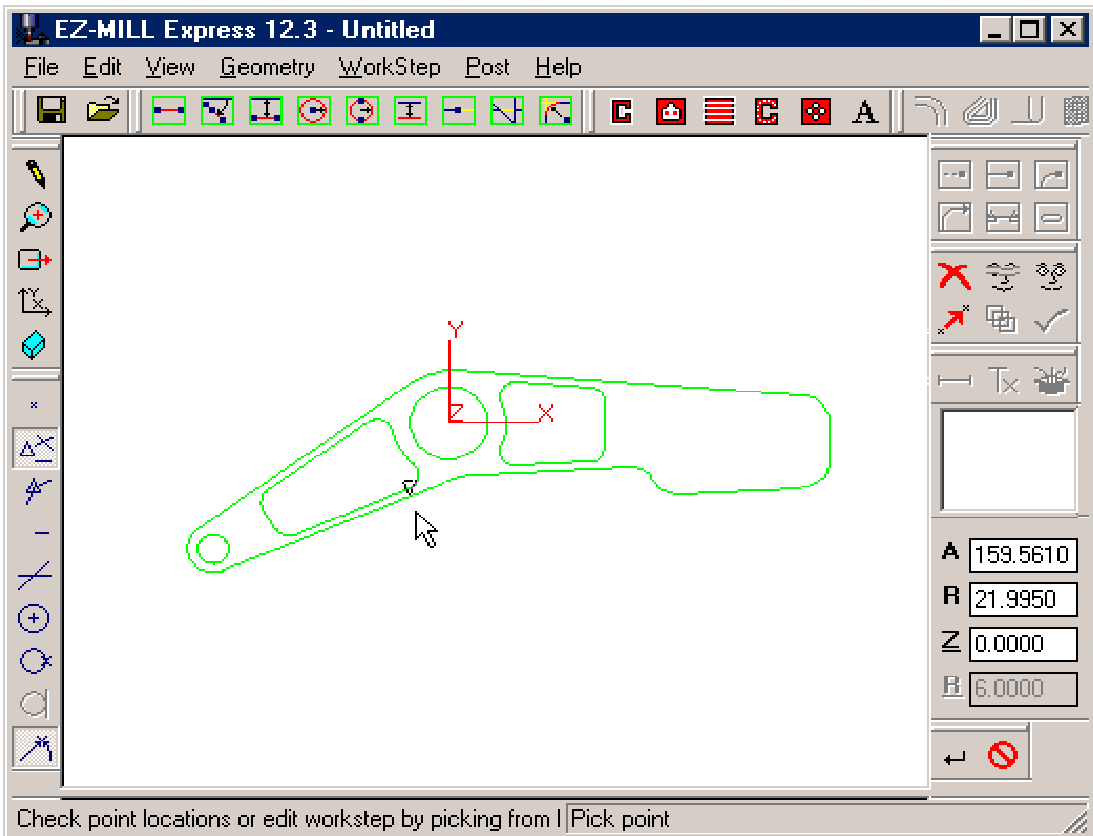
## Miscellaneous functions

M02 - Program end
M03 - Start of **spindle rotation clockwise**
M04 - Start of spindle rotation counterclockwise
M07 - Start of **mist coolant** (spray)
M08 - Start of **flood coolant** (e.g. oil)

# CAD/CAM and CNC

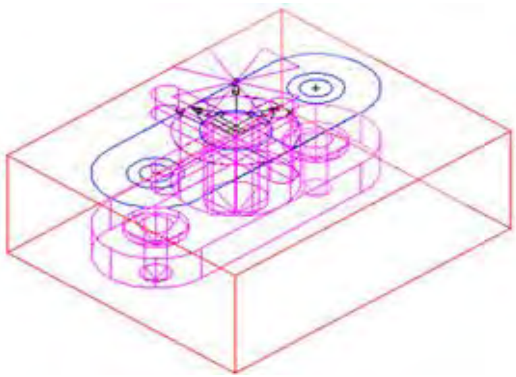
## Example of CNC programming

See http://www.ezcam.com/web/tour/tour.htm



# CAD/CAM and CNC

## Example of CNC programming



# CAD/CAM and CNC

## Advanced CNC programming languages

- Automatically Program Tool (APT), developed at MIT in 1954
- Derived from APT: ADAPT (IBM), IFAPT (France), MINIAPT (Germany)
- Compact II
- Autospot
- SPLIT

## Current trend in interpolation

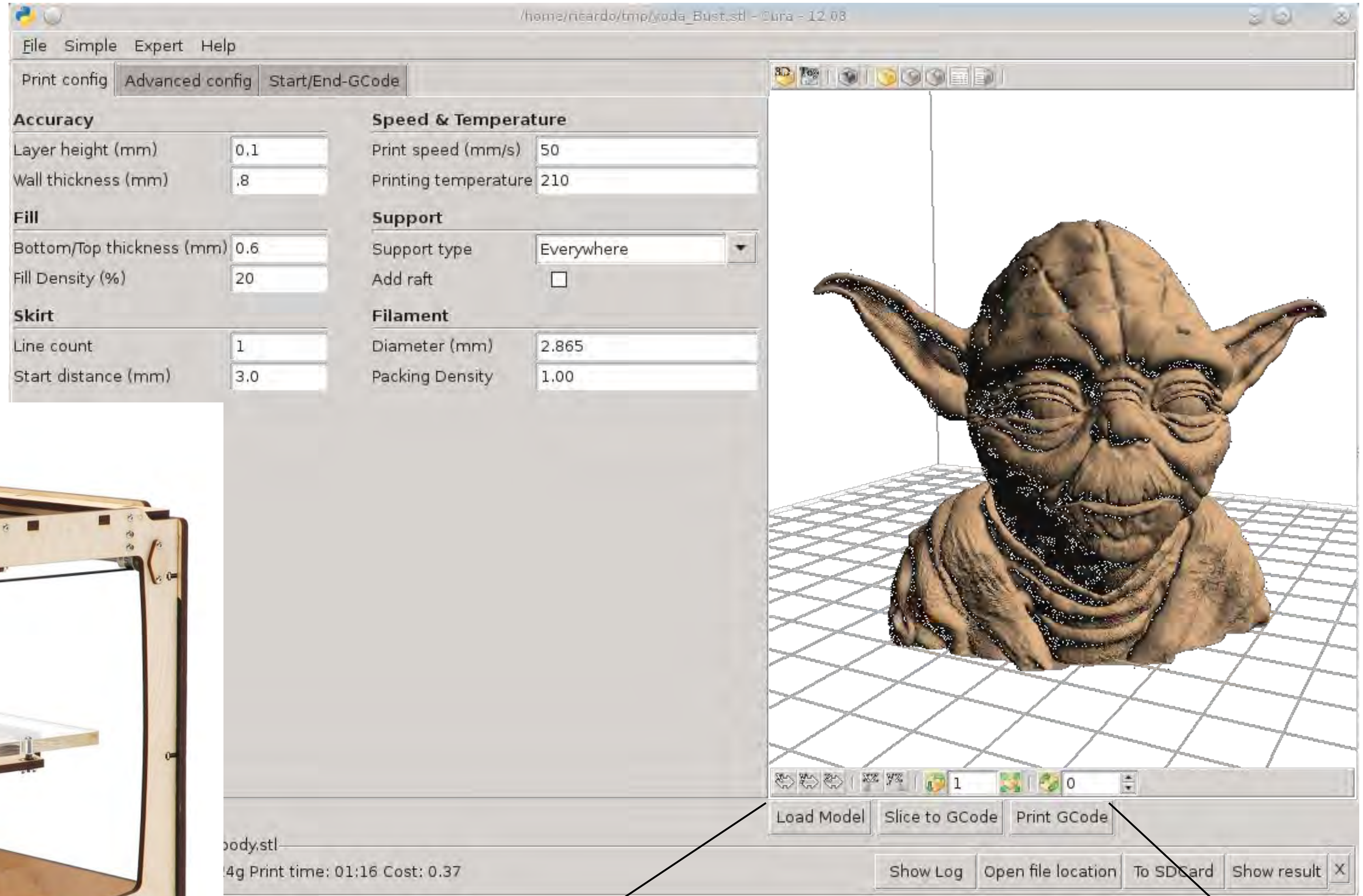
*Modern CAD systems have progressively gained the capability to describe a wide variety of complex shaped parts (like dies and molds) through parametric curves or surfaces like the Bezier, B-Spline or **non-uniform rational B-Spline (NURBS)**. (...) NURBS is one curve interpolator that draws considerable attention owing to the fact that NURBS offers a universal mathematical form for representing both analytical and free-form shapes [9]. In fact, most commercial CNC controller manufacturers (such as **Fanuc** [15] and **Siemens** [16]) incorporate such interpolation capabilities to their high-end CNC products.*

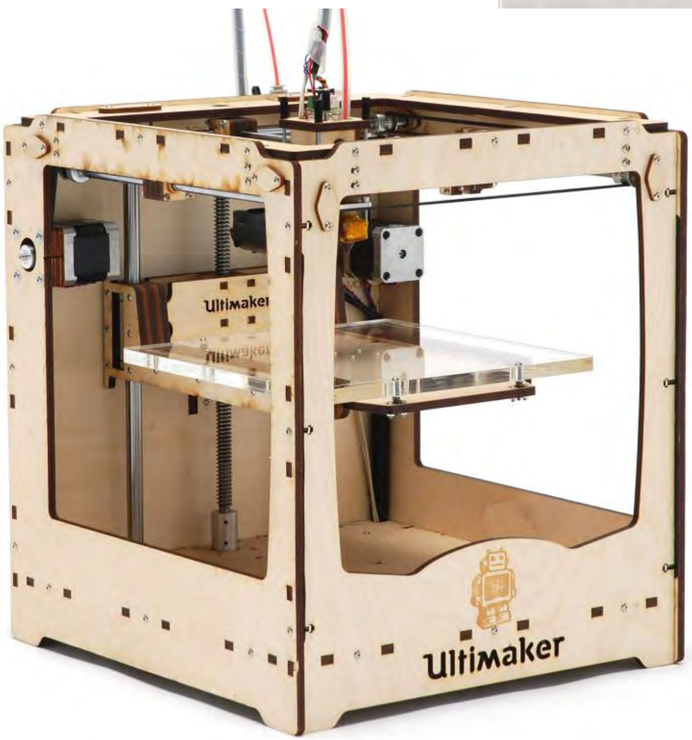
In "Direct command generation for CNC machinery based on data compression techniques", U. Yaman, M. Dolen, Robotics and Computer-Integrated Manufacturing 29 (2013) 344–356

# CAD/CAM and CNC at home!

http://daid.github.com/Cura/

*Order in the internet, receive by mail and assemble yourself!*
http://www.ultimaker.com/

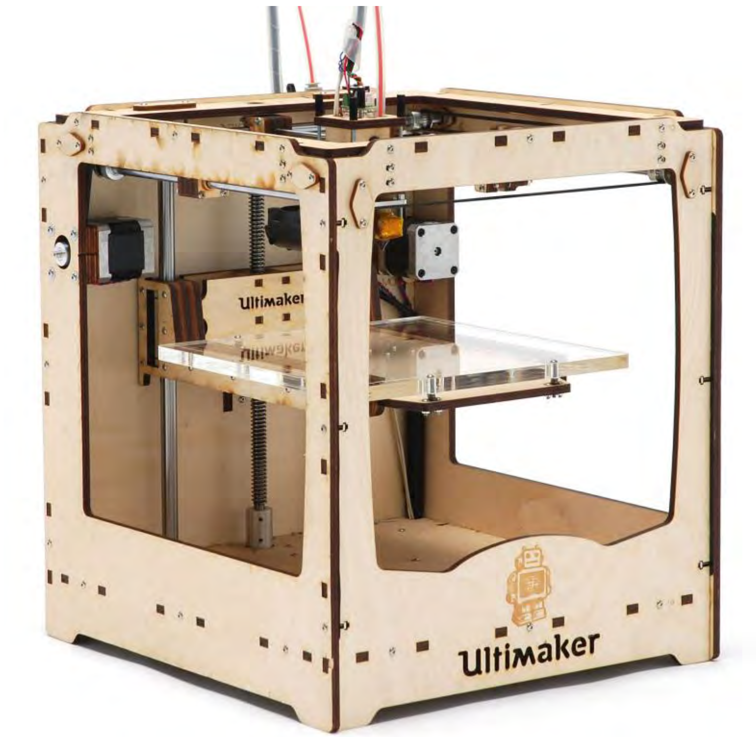


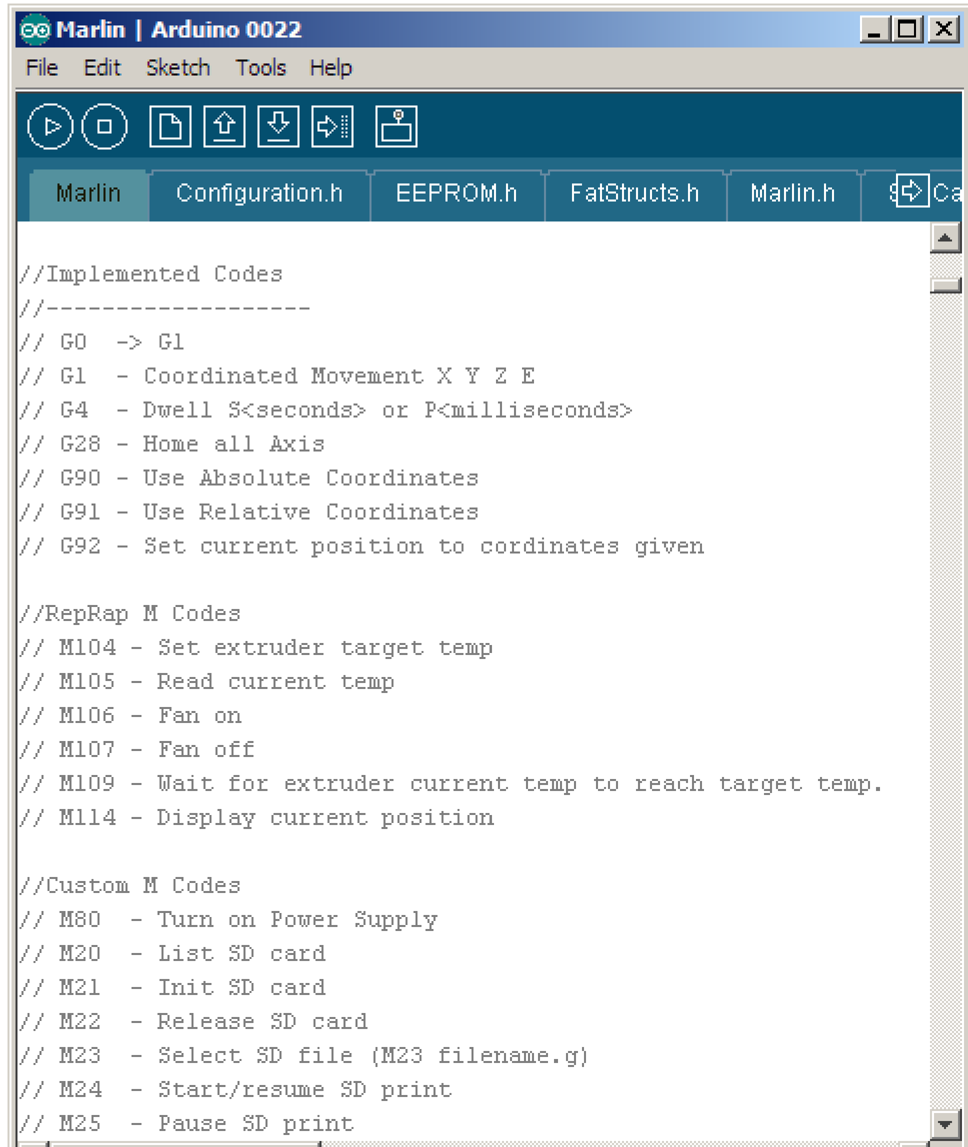

# CAD/CAM and CNC at home! - PC side, Slice to GCode

```
;TYPE:CUSTOM
M92 E865.888000
M109 S210.000000
;Sliced /home/ricardo/tmp/dump_body.stl at: Sun 28 Oct 2012 22:20:23
;Basic settings: Layer height: 0.1 Walls: 0.8 Fill: 20
;Print time:     1:16
;Filament used:     1.10m     9.24g
;Filament cost:     0.37
G21         ;metric values
G90         ;absolute positioning
M107        ;start with the fan off
G28 X0 Y0   ;move X/Y to min endstops
G28 Z0      ;move Z to min endstops
G92 X0 Y0 Z0 E0         ;reset software position to front/left/z=0.0
G1 Z15.0 F180
G92 E0                  ;zero the extruded length
G1 F200 E3
G92 E0                  ;zero the extruded length again
;G1 X100 Y100 F9000
G1 F9000
;LAYER:0
;TYPE:SKIRT
G1 X74.244 Y116.715 Z0.3 F9000.0
G1 F4200.0
G1 E4.525
G1 F9000.0
G1 X75.623 Y120.052 Z0.3 F1200.0 E4.5922
G1 X113.604 Y120.572 E5.2993
```

# CAD/CAM and CNC at home! - Machine side, Interpret GCode




https://github.com/bkubicek/Marlin
http://wiki.ultimaker.com/How_to_upload_new_firmware_to_the_motherboard



# CAD/CAM and CNC at home! - Machine side, Interpret GCode

Marlin | Arduino 0022

File Edit Sketch Tools Help

Marlin § | Configuration.h | EEPROM.h | FatStructs.h | Marlin.h

```
void loop()
{
  if(buflen<3)
    get_command();
        checkautostart(false);
  if(buflen)
  {
    process_commands();
    buflen = (buflen-1);
    bufindr = (bufindr + 1)%BUFSIZE;
  }
  //check heater every n milliseconds
  manage_heater();
  manage_inactivity(1);
  LCD_STATUS;
}


inline void get_command()
{
  while( Serial.available() > 0  && buflen < BUFSIZE) {
    serial_char = Serial.read();
    if(serial_char == '\n' || serial_char == '\r' || serial_char
    {
```

Marlin | Arduino 0022

File Edit Sketch Tools Help

Marlin § | Configuration.h | EEPROM.h | FatStructs.h | Marlin.h

```
inline void process_commands()
{
  unsigned long codenum; //throw away variable
  char *starpos = NULL;

  if(code_seen('G'))
  {
    switch((int)code_value())
    {
    case 0: // G0 -> G1

    case 1: // G1
      get_coordinates(); // For X Y Z E F
      prepare_move();
      previous_millis_cmd = millis();
      //ClearToSend();
      return;
      //break;

    case 4: // G4 dwell
      codenum = 0;
      if(code_seen('P')) codenum = code_value(); // milliseconds
      if(code_seen('S')) codenum = code_value() * 1000; // second
      codenum += millis();  // keep track of when we started wait
      while(millis()  < codenum ){
        manage_heater();
```

# CAD/CAM and CNC at home! - Machine side, Interpret GCode

```
void prepare_move()
{
  plan_buffer_line(destination[X_AXIS], destination[Y_AXIS],
    destination[Z_AXIS], destination[E_AXIS],
    feedrate*feedmultiply/60.0/100.);

  for(int i=0; i < NUM_AXIS; i++) {
    current_position[i] = destination[i];
  }
}
```

```
void plan_buffer_line(float x, float y, float z, float e, floa
// Add a new linear movement to the buffer.
// steps_x, _y and _z is the absolute position in mm.
// Microseconds specify how many microseconds the move should
// calculation the caller must also provide the physical lengt

  // Calculate the buffer head after we push this byte
  int next_buffer_head = (block_buffer_head + 1) %BLOCK_BUFFER

  // If the buffer is full: good! That means we are well ahead
  // Rest here until there is room in the buffer.
  while(block_buffer_tail == next_buffer_head) {
    manage_heater();
    manage_inactivity(1);
  }

  // The target position of the tool in absolute steps
  // Calculate target position in absolute steps
  long target[4];
  target[X_AXIS] = lround(x*axis_steps_per_unit[X_AXIS]);
  target[Y_AXIS] = lround(y*axis_steps_per_unit[Y_AXIS]);
  target[Z_AXIS] = lround(z*axis_steps_per_unit[Z_AXIS]);
```

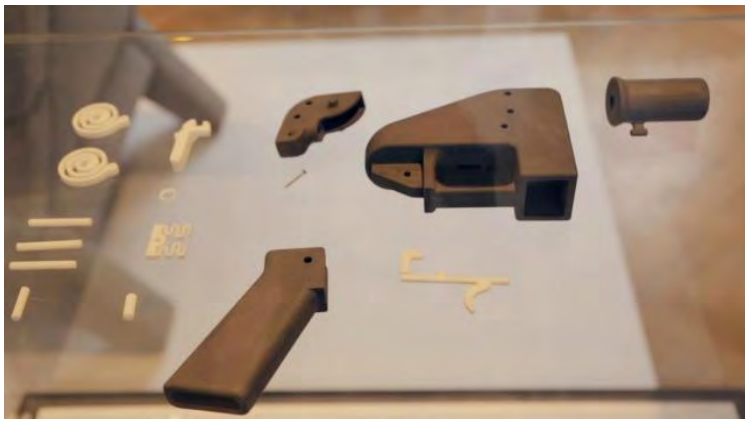
```
ISR(TIMER1_COMPA_vect)
// "The Stepper Driver Interrupt" - This timer interrupt is the workhorse.
// It pops blocks from the block_buffer and executes them by pulsing the st
{
  if(busy){ /*Serial.println("BUSY")*/;
    return;
  } // The busy-flag is used to avoid reentering this interrupt

  busy = true;
  sei(); // Re enable interrupts (normally disabled while inside an interru
#ifdef ULTIPANEL
  static int breakdown=0;
        if((breakdown++)%100==0)
   buttons_check();
/* [ErikDeBruijn] Perhaps it would be nice to use a piece of code like this
      if(sdactive){
        sprintf("SD printing byte %i%",(int) (sdpos/filesize*100)); // perh
        Serial.print(sdpos);
        Serial.print("/");
        Serial.println(filesize);
      }
*/
#endif
```

# CAD/CAM and CNC at home!

# CAD/CAM and CNC at home – a word of caution

## 3D-printed gun on display at V&A museum

By Sophie Curtis, The Telegraph, 17th Sep 2013



Victoria and Albert Museum (London), acquired, for display in their collection, the world's first 3D-printed gun, named "Liberator", developed and successfully fired by Texan law student Cody Wilson.

http://www.telegraph.co.uk/technology/news/10314763/3D-printed-gun-on-display-at-VandA-museum.html
http://www.dezeen.com/2013/09/26/movie-kieran-long-v-and-a-museum-london-3d-printed-gun/

## UK police raise specter of 3-D printer-made guns

By Laura Smith-Spark, CNN, 25th Oct 2013

The U.S. State Department banned the inventor of a plastic handgun, "The Liberator," from distributing its instructions.

Police in England said Friday they have seized what could be the parts for Britain's first firearm made using 3-D printing -- but later said more testing is needed to establish if this is the case.

http://edition.cnn.com/2013/10/25/world/europe/uk-police-3d-printer-gun/