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. 2011-2013 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

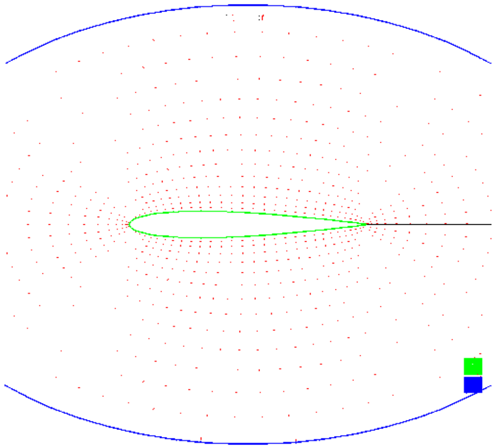
History: http://users.bergen.org/jdefalco/CNC/history.html

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

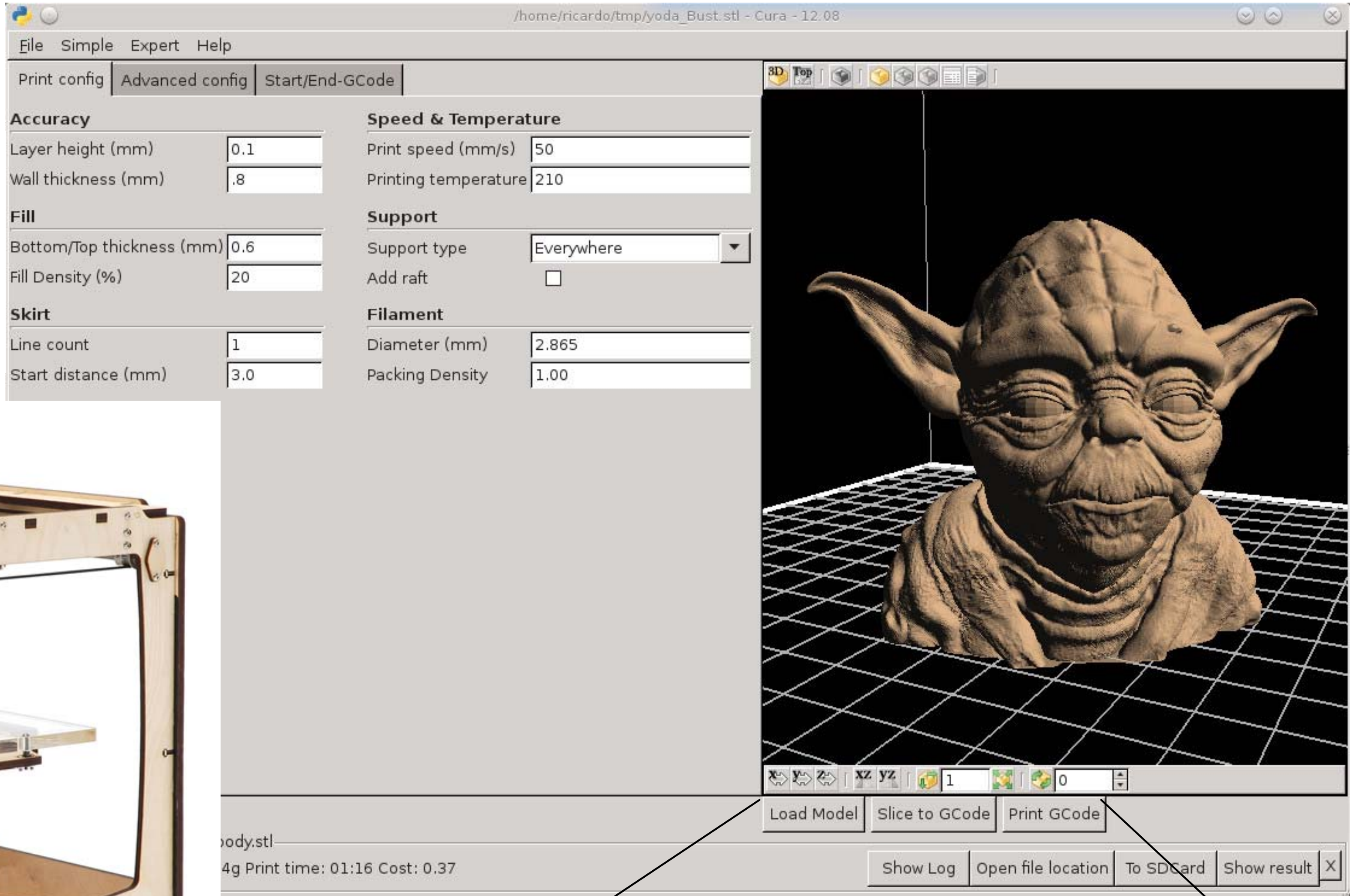
Prototype

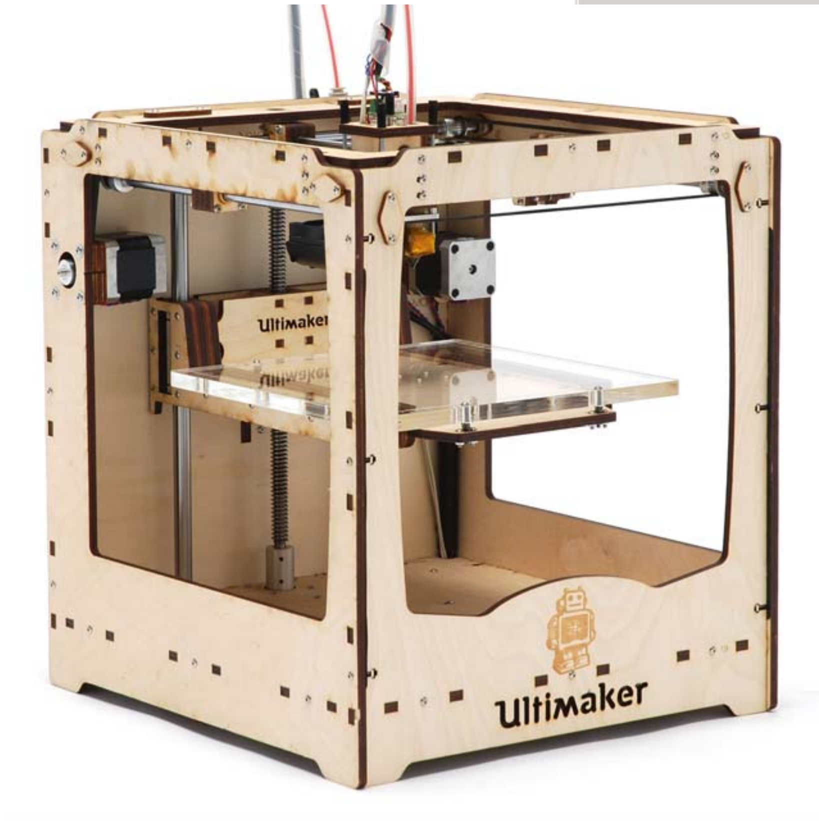
Nowadays, the machines work perfectly OK! the technological question is mostly about ***integration.***

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 describes 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.

Evolution in brief

CAD/CAM and CNC

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

Objectives

- To augment the accuracy, reliability, and the ability to introduce changes/new designs
- To augment the workload
- To reduce production costs
- To reduce waste due to errors and other human factors
- To carry out complex tasks (e.g. Simultaneous 3D interpolation)
- Augment 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

To 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)

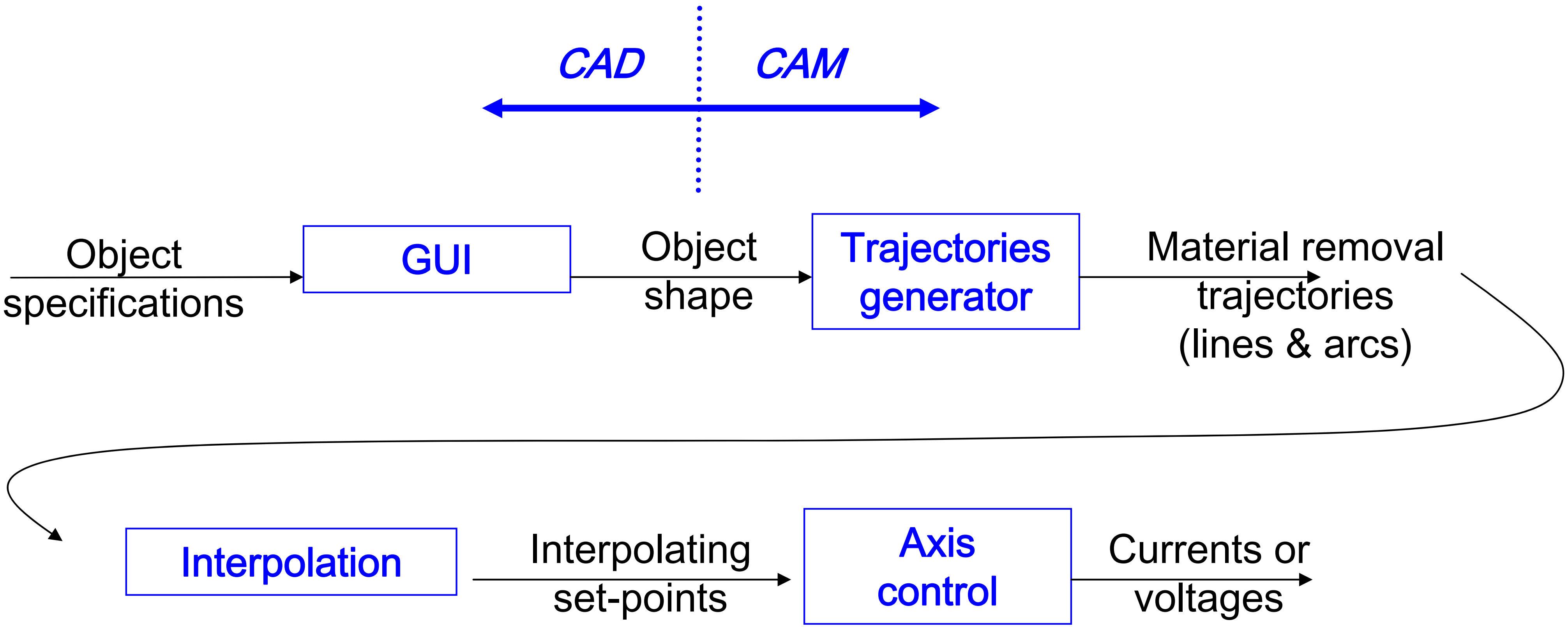
Allows the design in a computer environment.

CAM (Computer Aided Manufacturing)

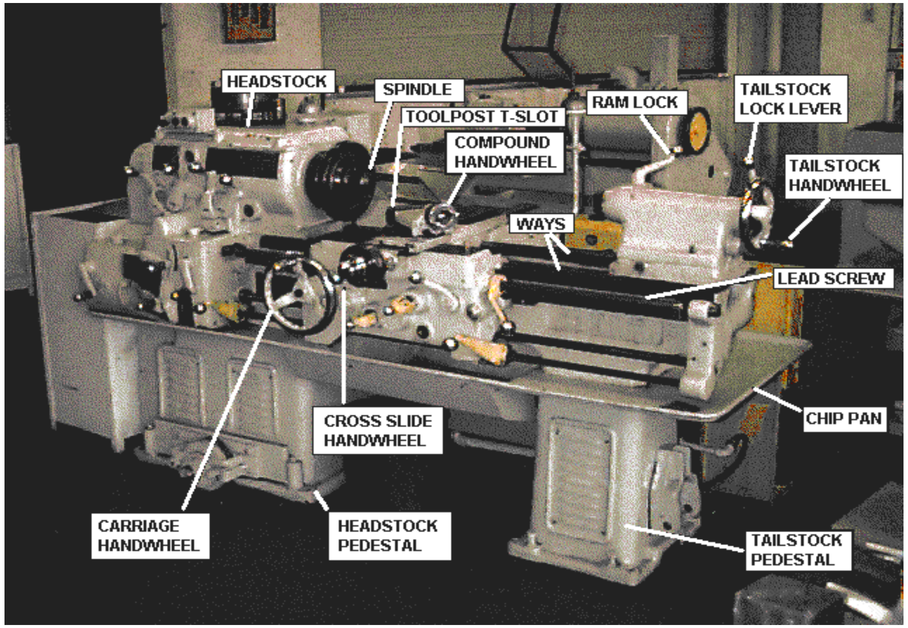
To manage programs and production stages on a computer.

CAD/CAM and CNC

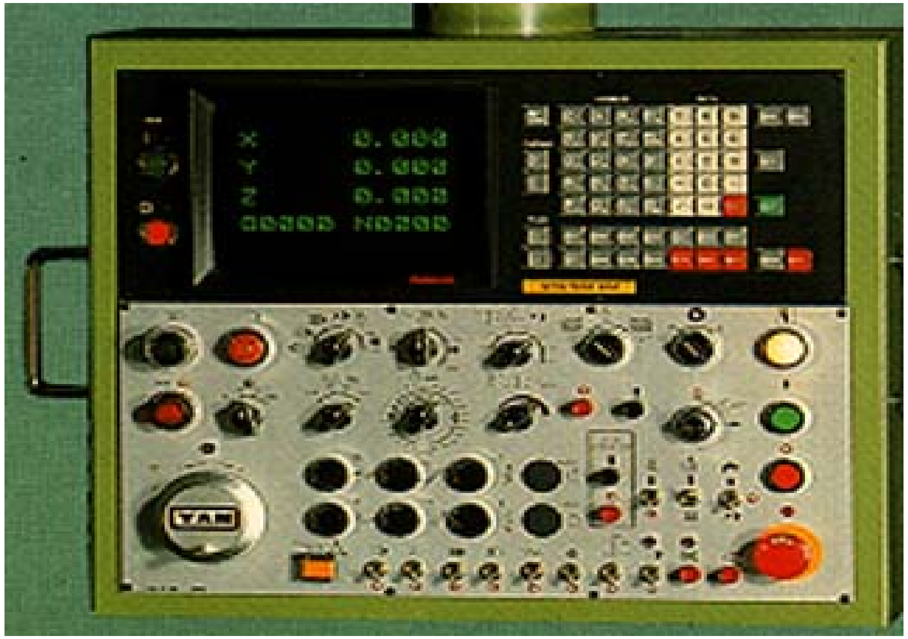
Methodology CAD/CAM




Bridgeport
TORQ-CUT
22

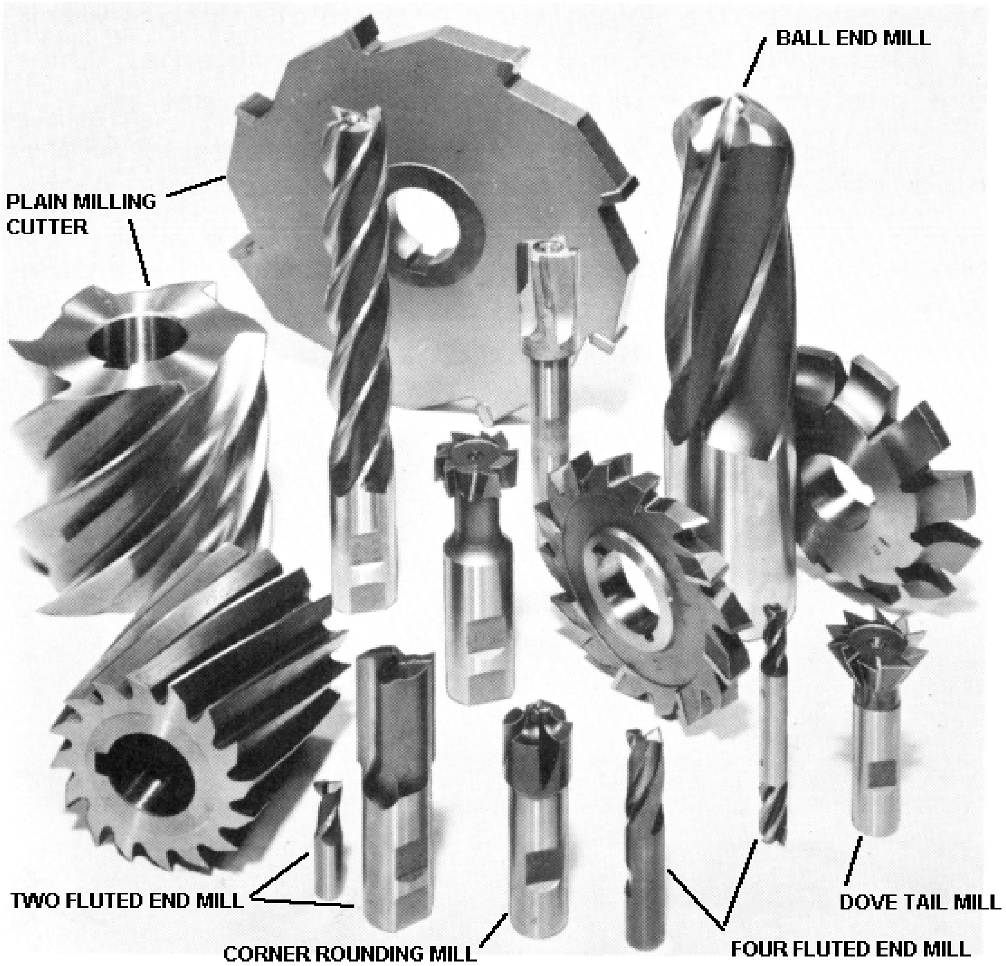

HEADSTOCK
SPINDLE
TOOLPOST T-SLOT
COMPOUND HANDWHEEL
RAM LOCK
TAILSTOCK LOCK LEVER
TAILSTOCK HANDWHEEL
WAYS
LEAD SCREW
CHIP PAN
CROSS SLIDE HANDWHEEL
CARRIAGE HANDWHEEL
HEADSTOCK PEDESTAL
TAILSTOCK PEDESTAL


SKIQ 20



CAD/CAM and CNC

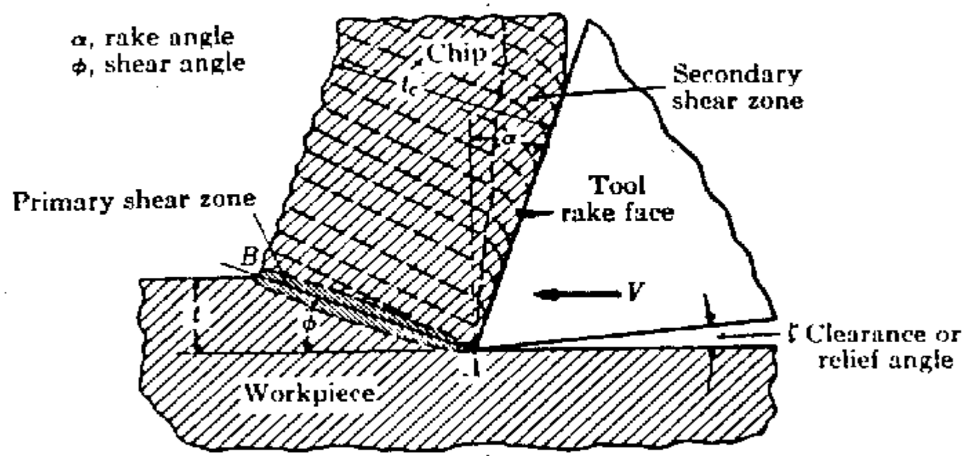
Tools:

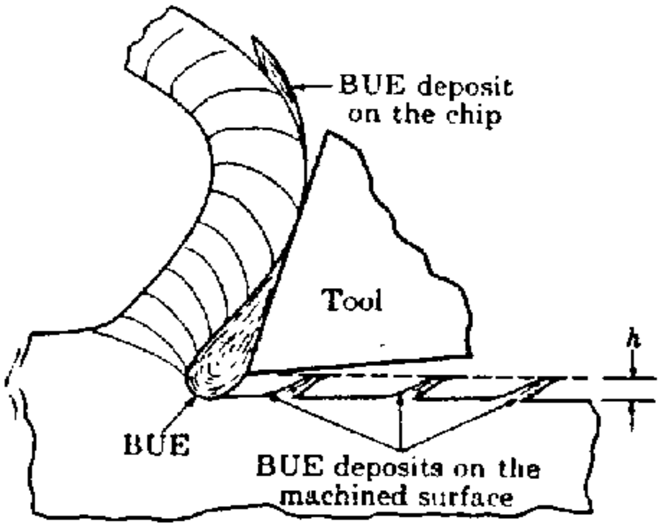


CAD/CAM and CNC

Tools:

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

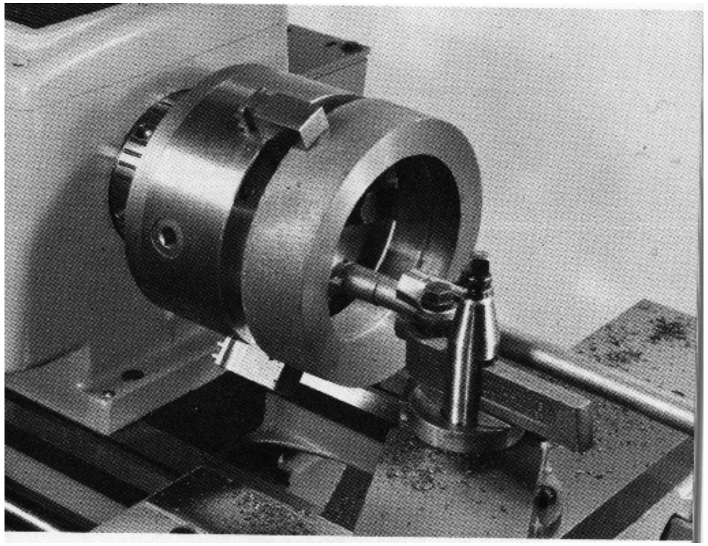


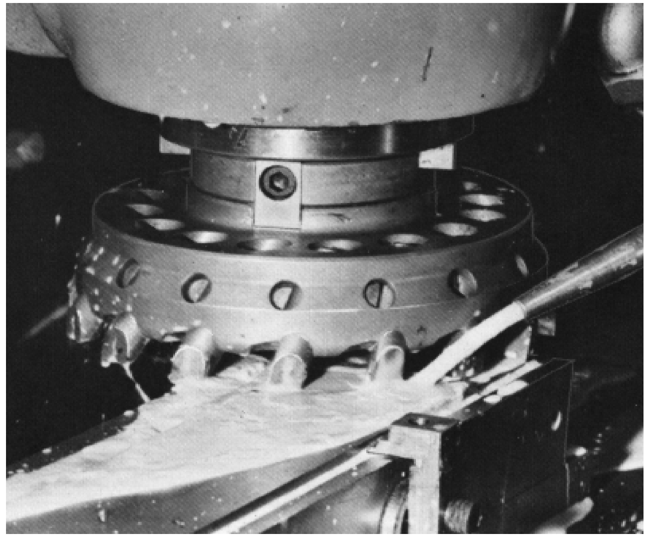


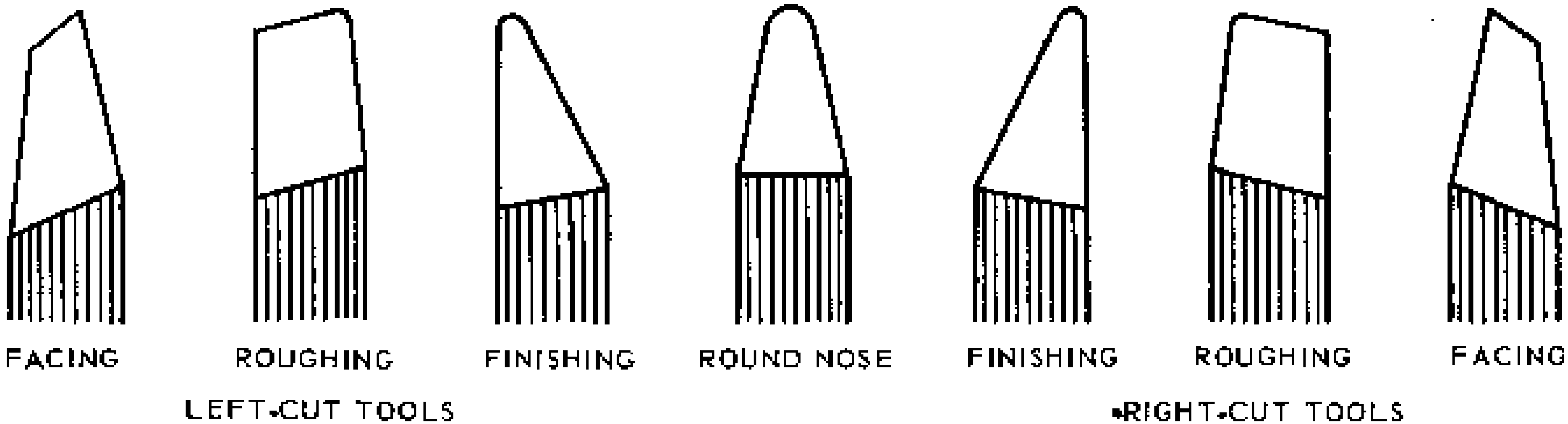
- **Speed of advance**
- **Speed of rotation**
- **Type of tool**

CAD/CAM and CNC

Tools:



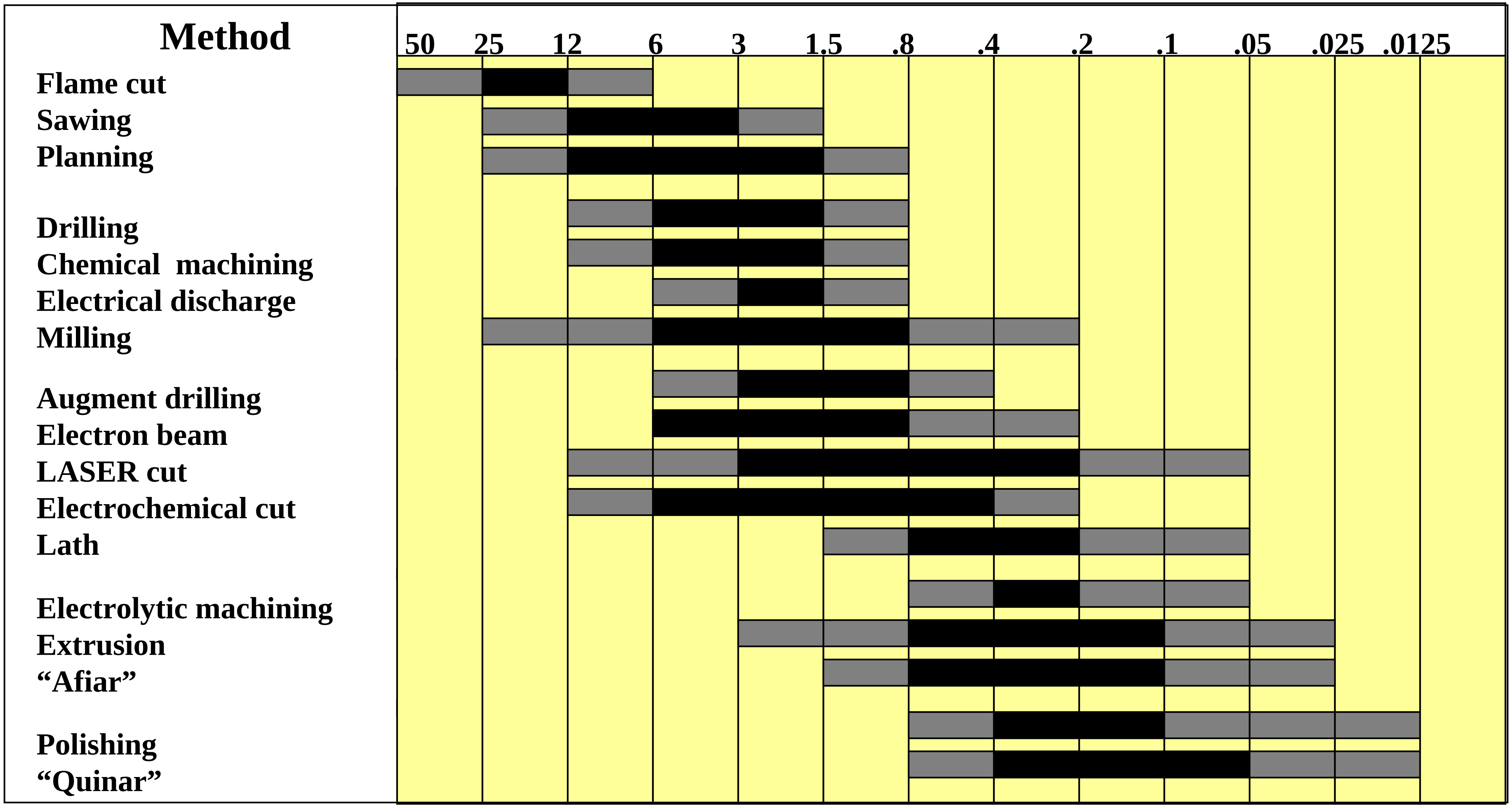




Specific tools to perform different operations.

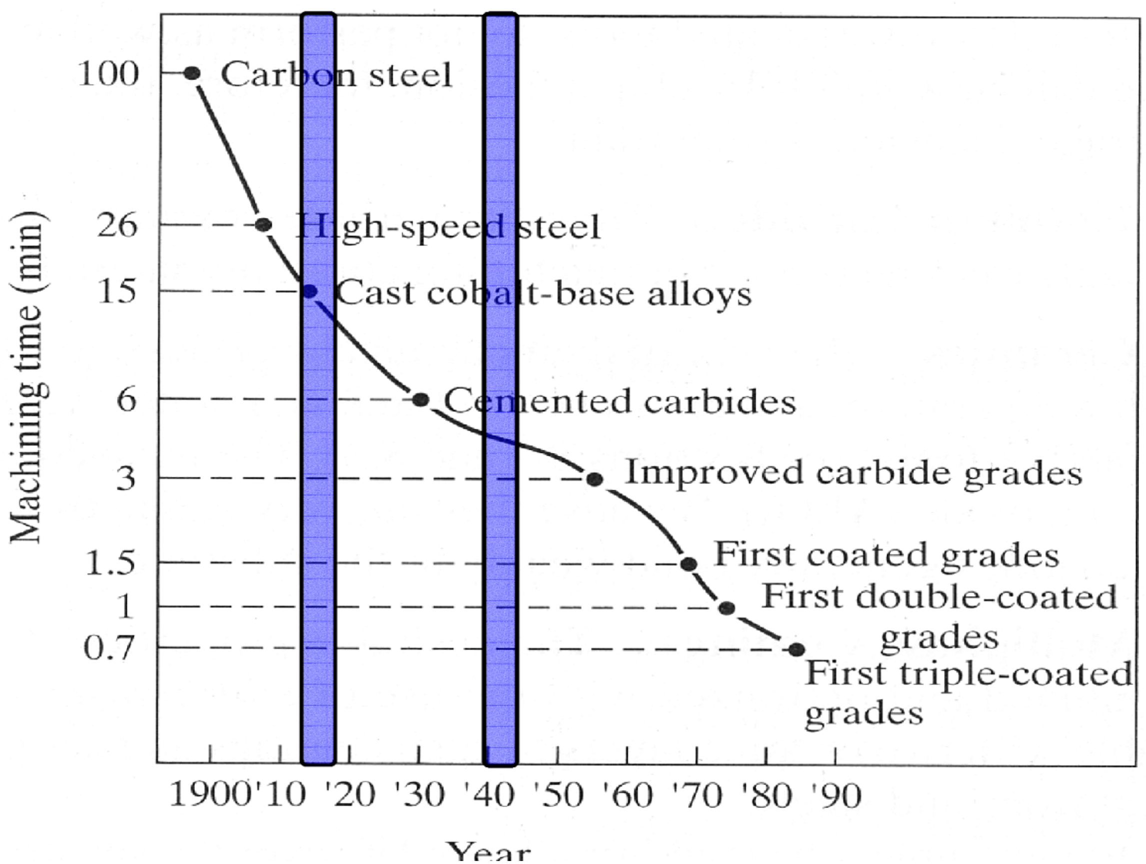
CAD/CAM and CNC

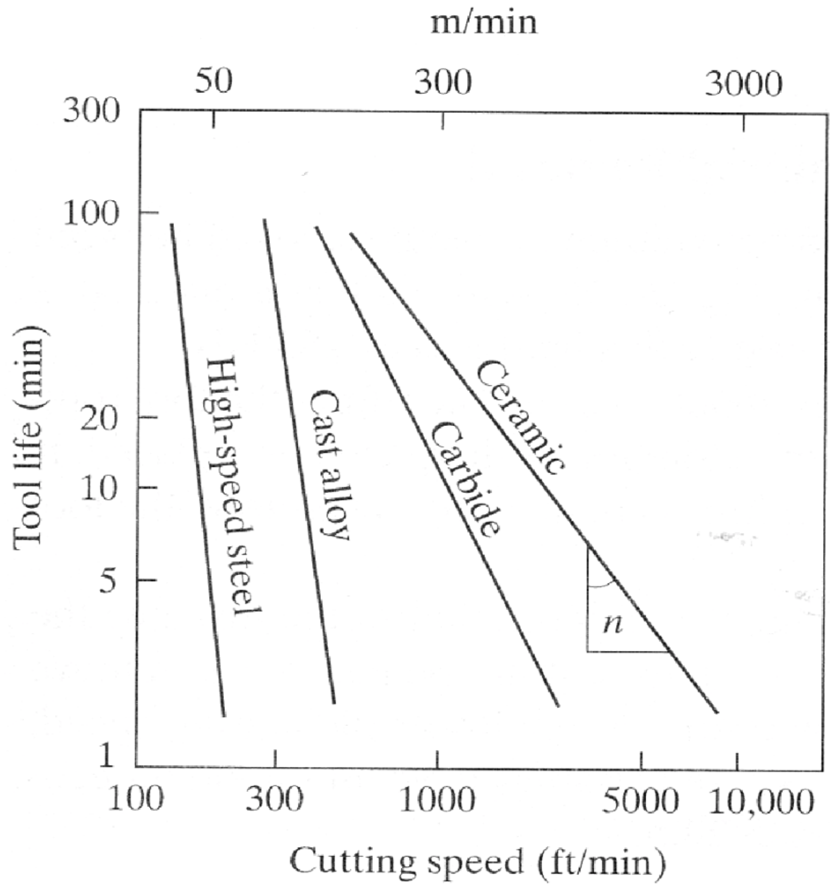
Tools: impact on the quality of finishing (μm)



CAD/CAM and CNC

Evolution of tools performance:





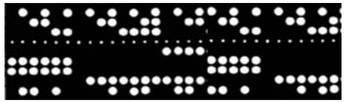
CAD/CAM and CNC

Industrial areas of application:

- Aerospace
- Machinery
- Electricity (board production)
- Automobiles
- Instrumentation
- Moulds

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

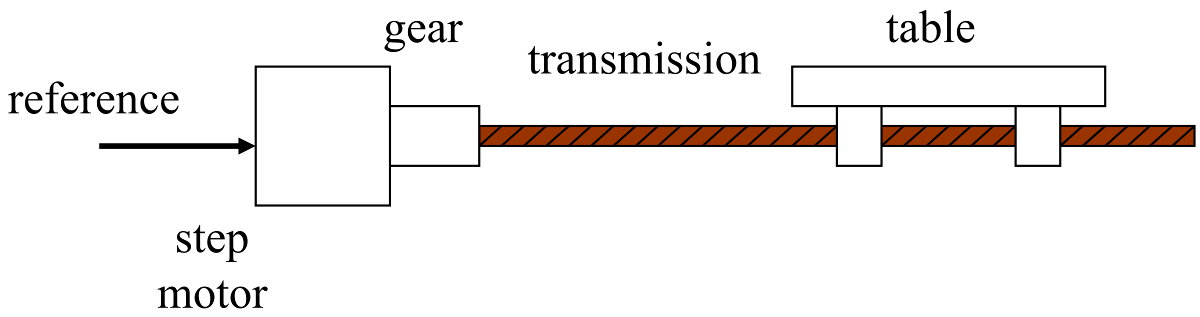
- Distributive numerical control
  - scheduling
  - Quality control
  - Remote monitoring

CAD/CAM and CNC

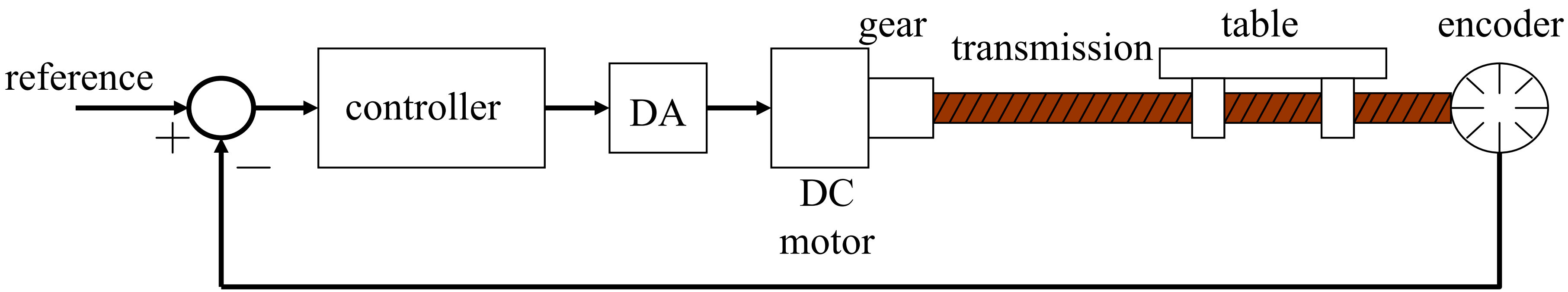
Numeric Control

Architecture of a NC system: 1 axis

Open-loop



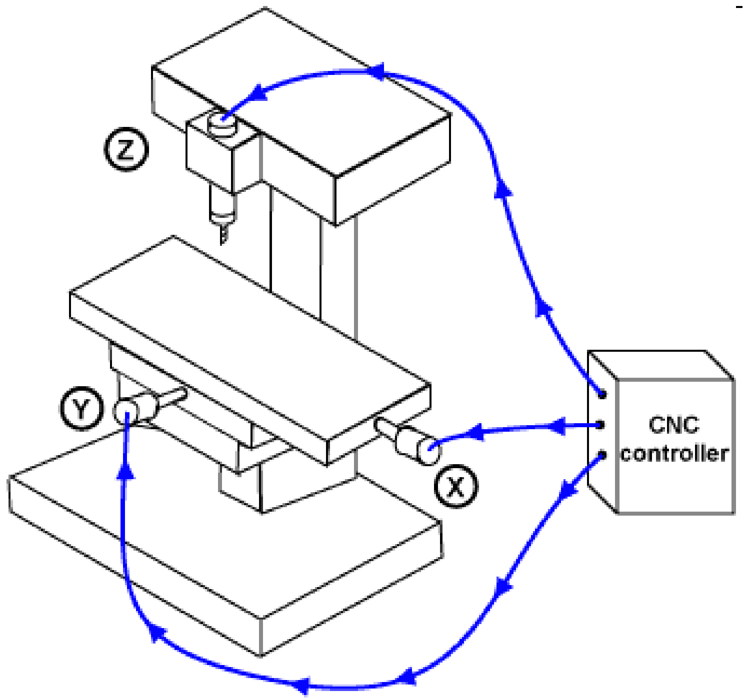
Closed-loop



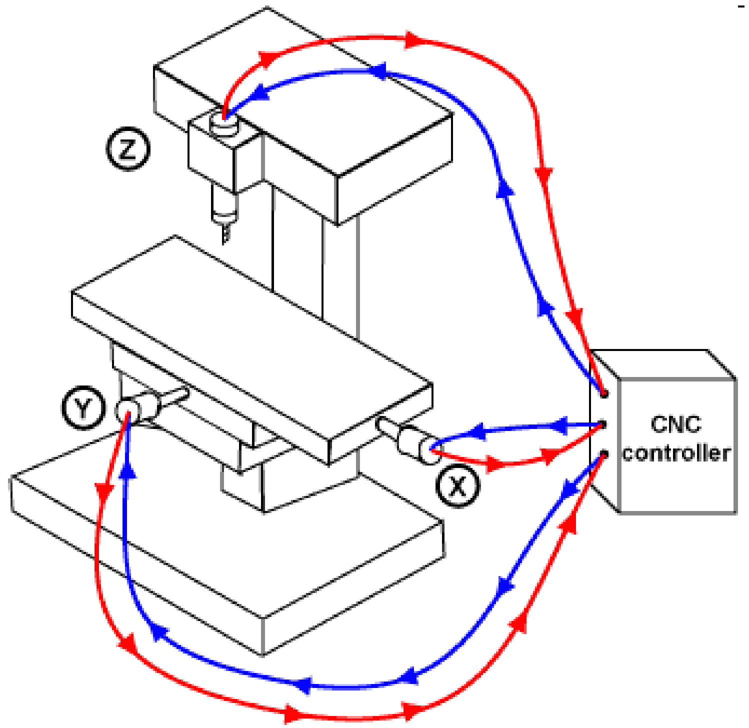
CAD/CAM and CNC

Numeric Control

Architecture of a NC system: 3 axis



Open-Loop-System

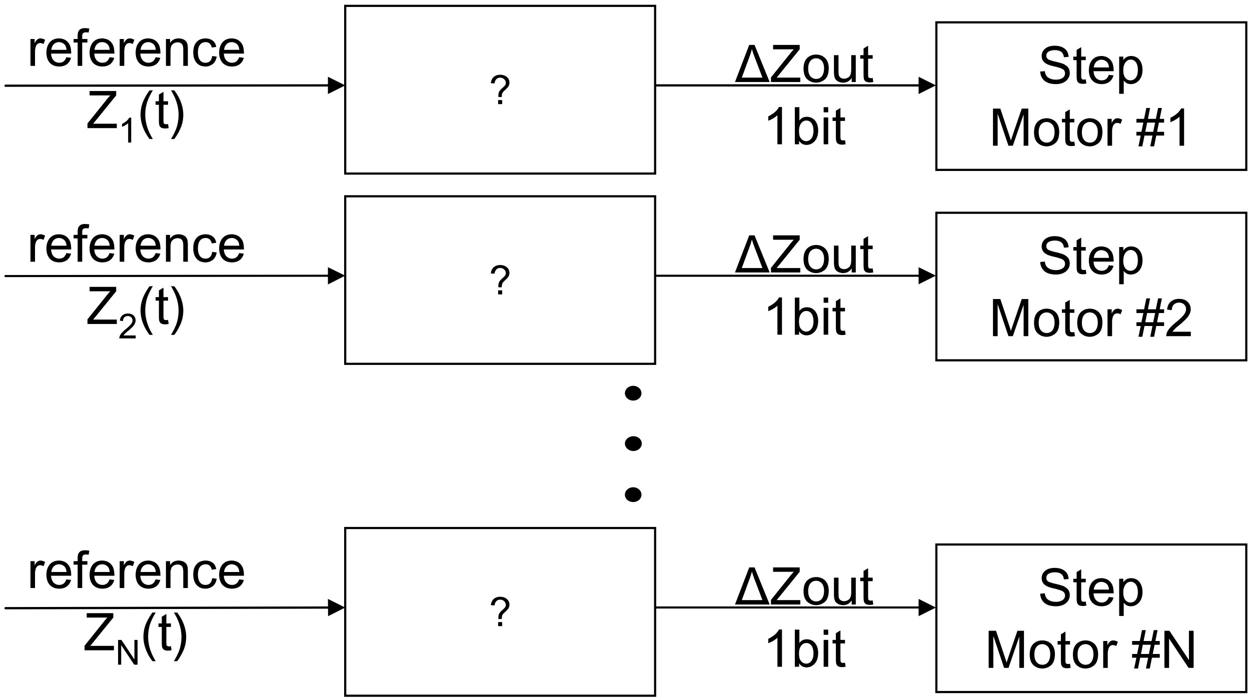


Closed-Loop-System

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

CAD/CAM and CNC

Interpolation Motivation



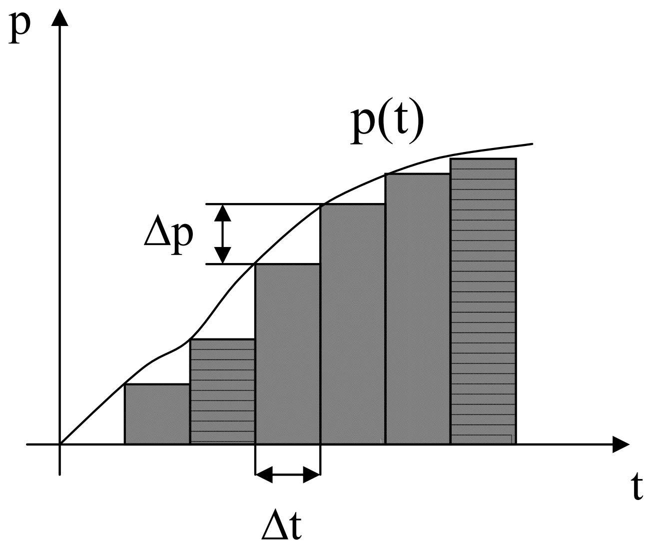
*Note1: The references are usually very **simple**, e.g.* $Z_i(t)=a_i t+b_i$

*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=kΔt

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

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

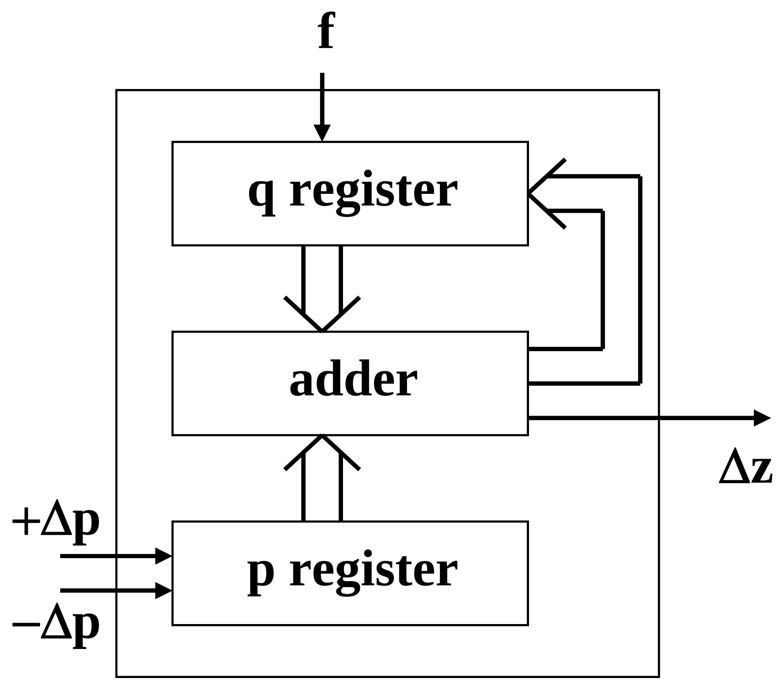
$$p_k = p_{k-1} \pm \Delta p_k$$

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 Δp_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:

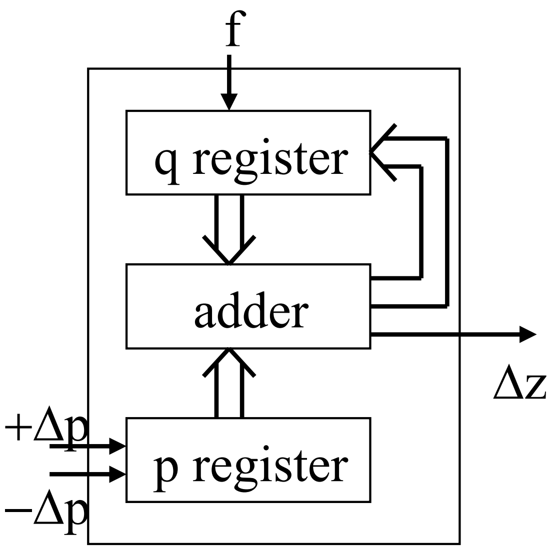
$$\Delta z_k = 2^{-n} p_k$$

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

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

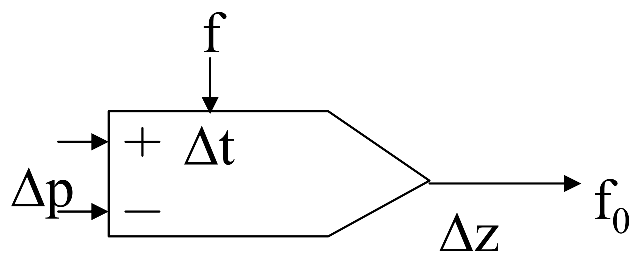
CAD/CAM and CNC

DDA for Linear Interpolation (1 axis):

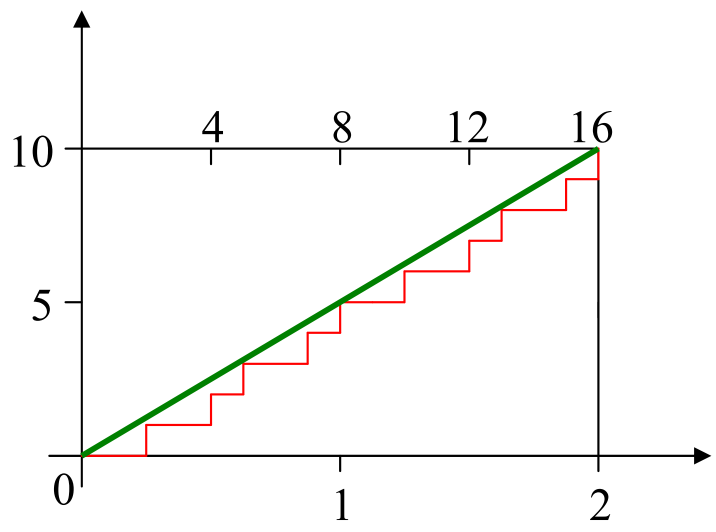


Let p=5, Δ p=0 and
assume q is a 3 bits register

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
		...	

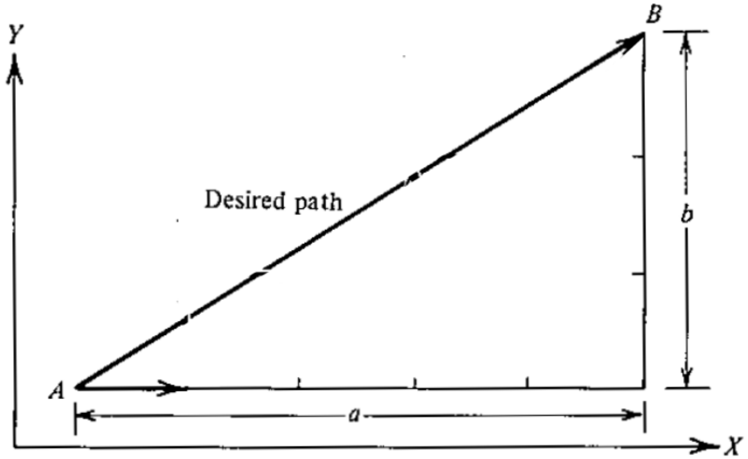


$$f_0 = \left(\frac{\Delta z}{\Delta t}\right)_k = C p_k, \quad \text{where} \quad C = \frac{f}{2^n}$$

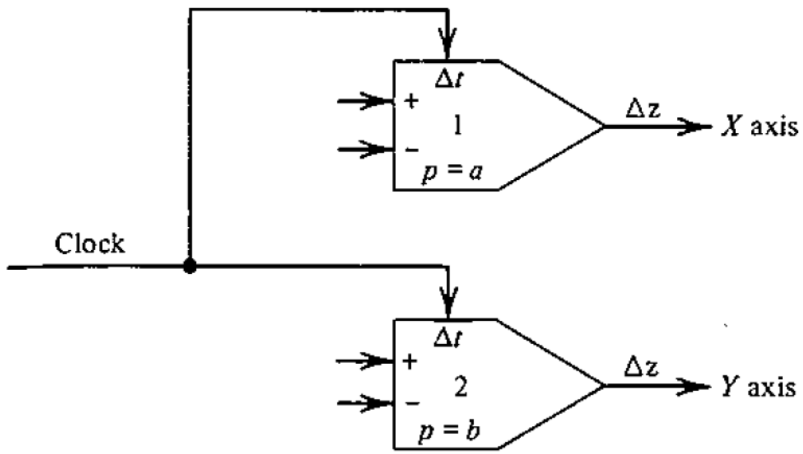


CAD/CAM and CNC

DDA for Linear Interpolation (2 axis):

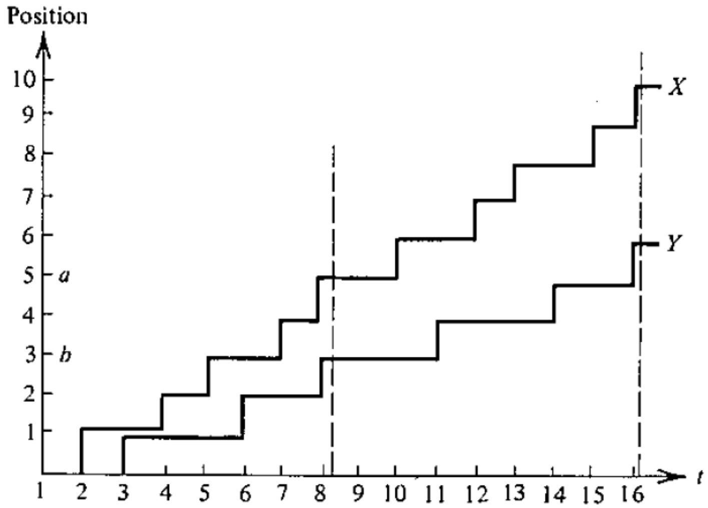


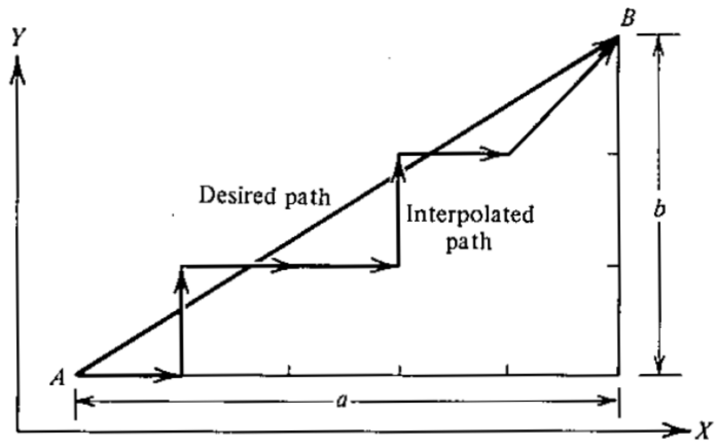
(a) Specifications



(b) DDA solution

(c) Results





CAD/CAM and CNC **Exponential Deacceleration:**

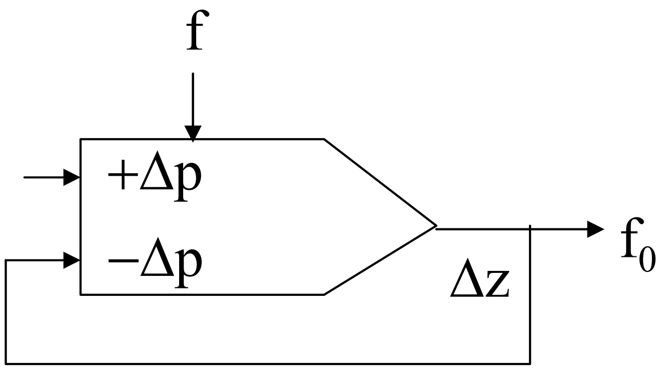
Let $p(t) = p_0 e^{-\alpha t}$ and $\frac{\Delta z}{\Delta t} = C p_k = C p_0 e^{-\alpha t}$.

The differential of p(t) is approximate

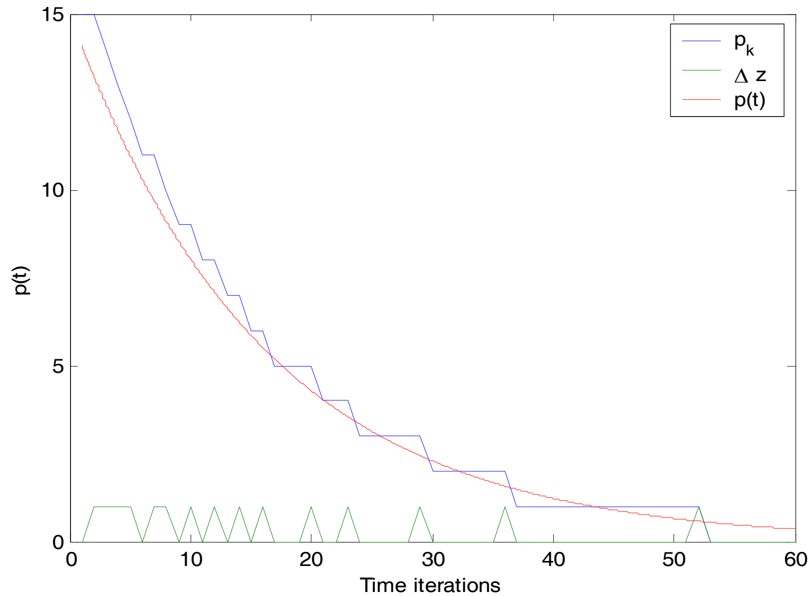
$$-\Delta p = \alpha p_k \Delta t$$

Setting C=α, i.e. $f = 2^n \alpha$, one has

$$-\Delta p = \Delta z$$



Example: $p(t)=15e^{-t}$



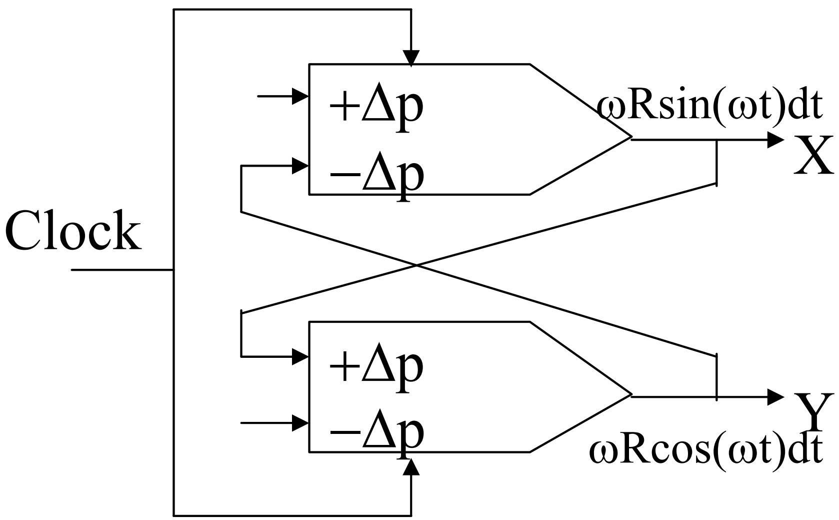
CAD/CAM and CNC **Circular Interpolation:**

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

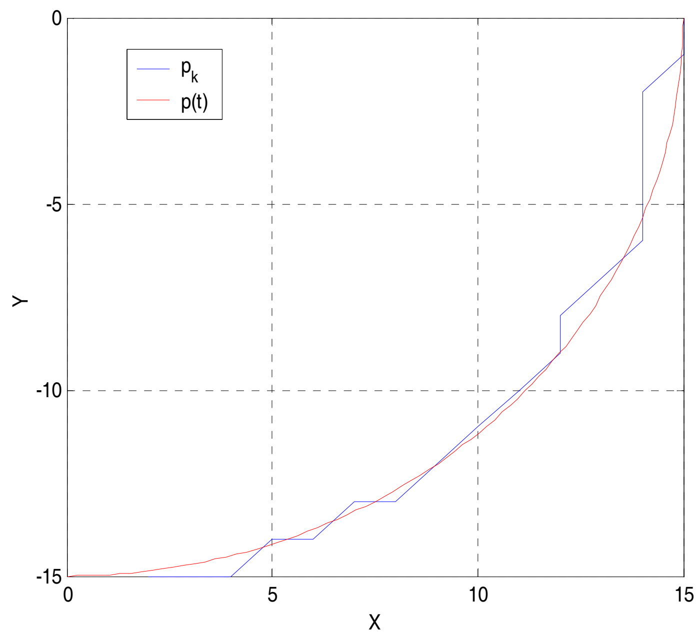
$$X = R(1-\cos(\omega t))$$
$$Y = R\sin(\omega t)$$

The differential is

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

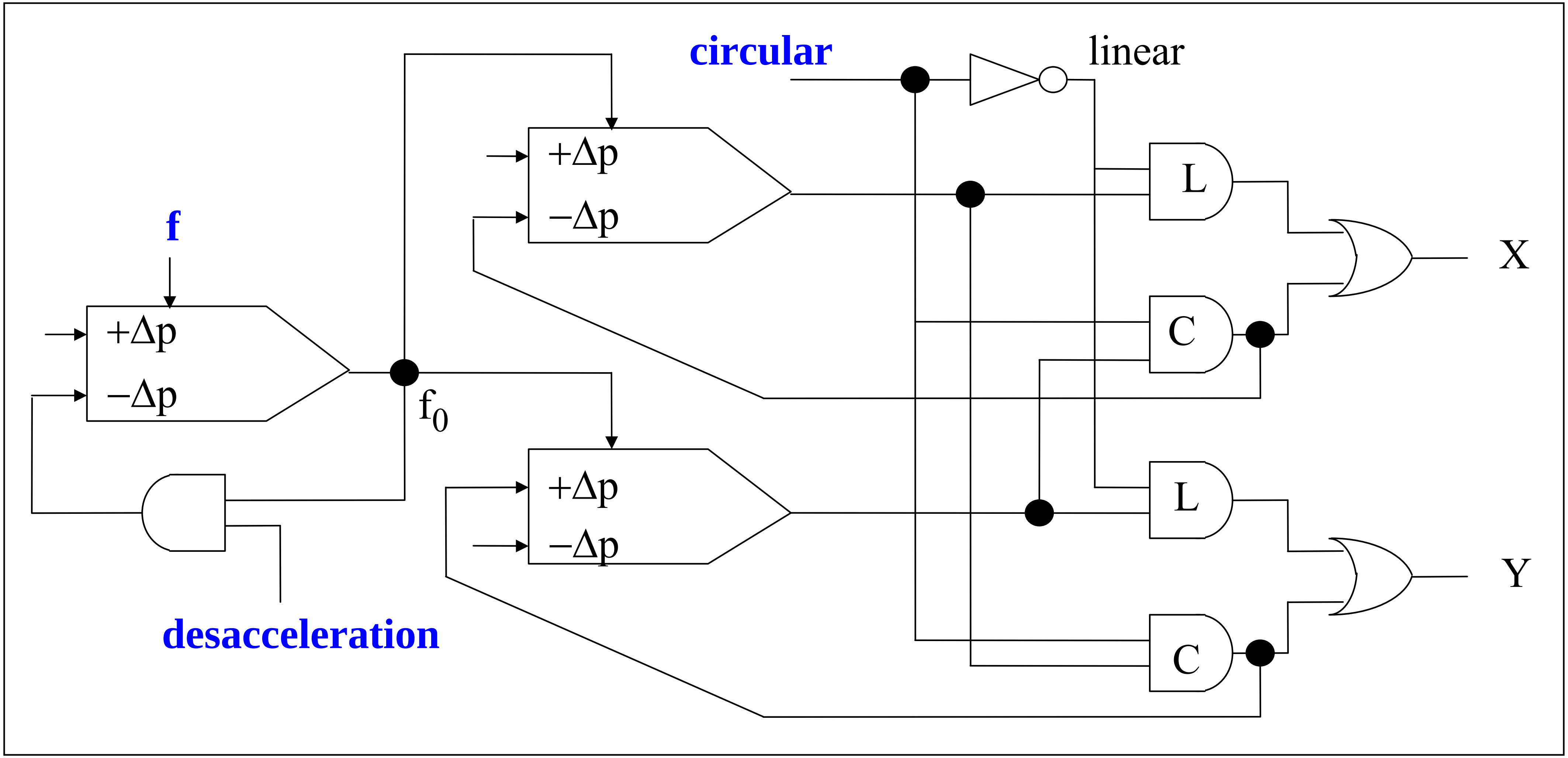


Example: Circumference of radius 15, centered at the origin.



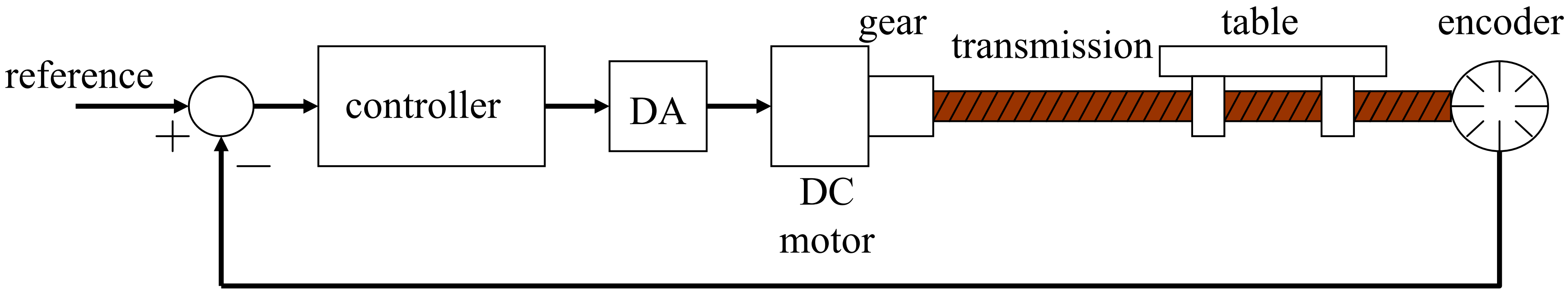
CAD/CAM and CNC Full DDA

2D Line, 2D Arc, Acceleration / Deacceleration

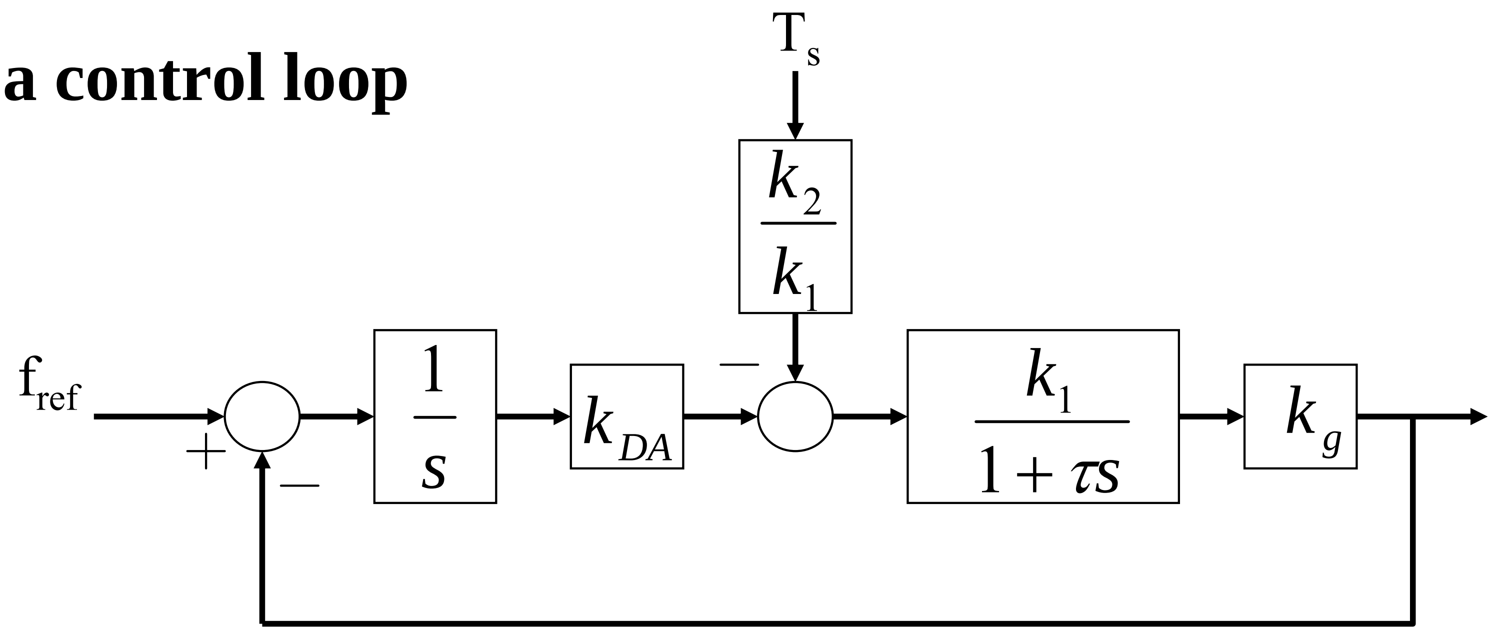


CAD/CAM and CNC

CNC Axes Control



Dynamics of a control loop

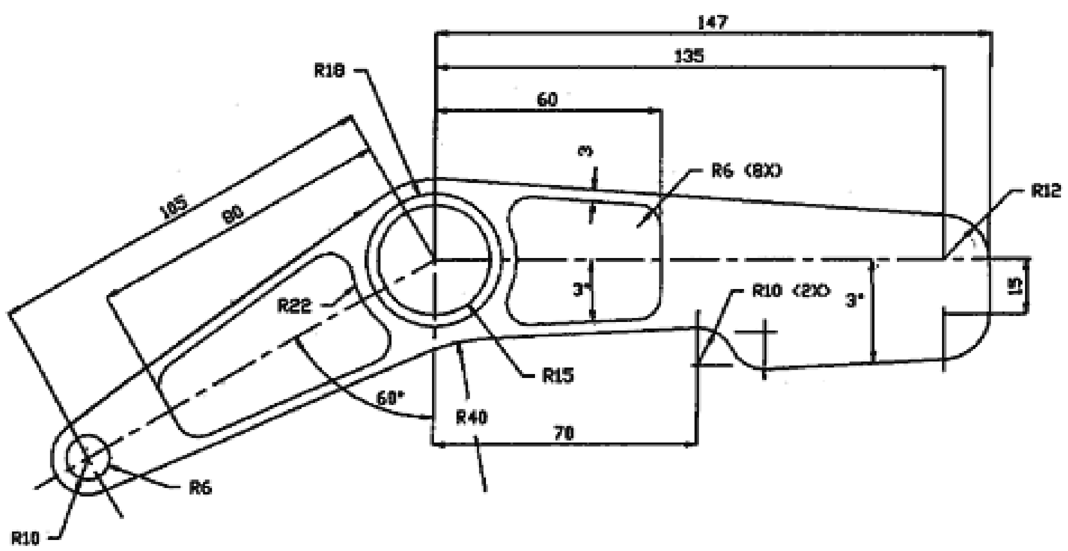


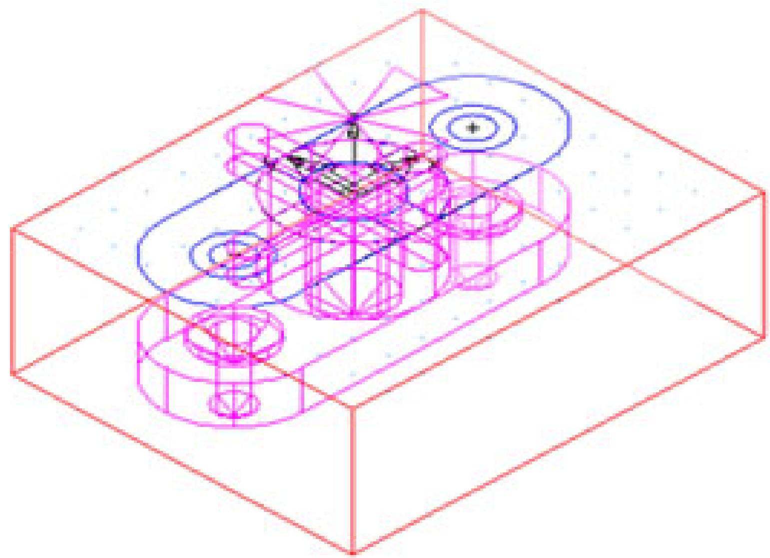
CAD/CAM and CNC

CNC Programming

Steps to execute a part

A) Read/interpret the technical drawings





CAD/CAM and CNC

CNC Programming

B) Choice of the most adequate **machine-tool** 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 available
- The mounting and the part handling
- The operations that each machine can perform

C) Choice 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

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

	Specific energy	
Material	$W \cdot s/mm^3$	$hp \cdot min/in.^3$
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

CNC Programming

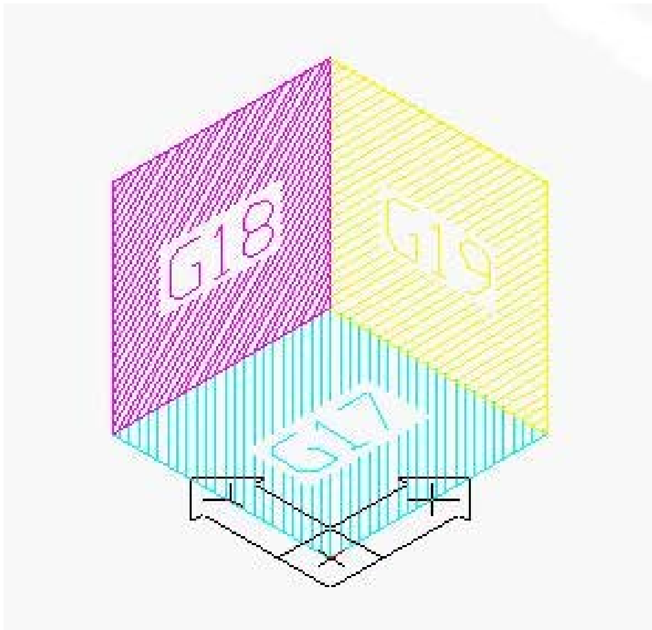
D) Cutting data

• 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)

E) Choice of the interpolation plane, in 2D ½ machines



CAD/CAM and CNC

CNC Programming

F_1) Unit system

imperial –inches (G70) or international millimeters (G71).

F_2) Command mode*

Absolute – relative to world coordinate system (G90)

Relative– movement relative to the actual position (G91)

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

CAD/CAM and CNC

CNC Programming

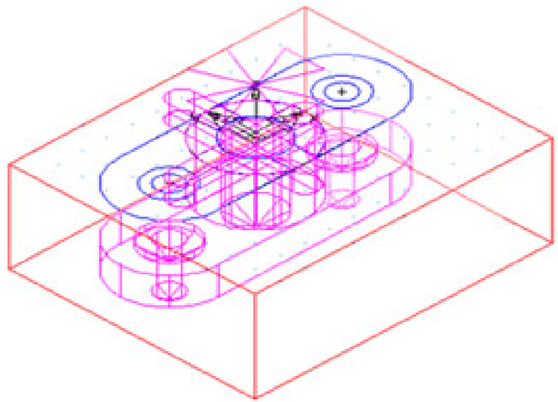
G) MANUAL DATA INPUT

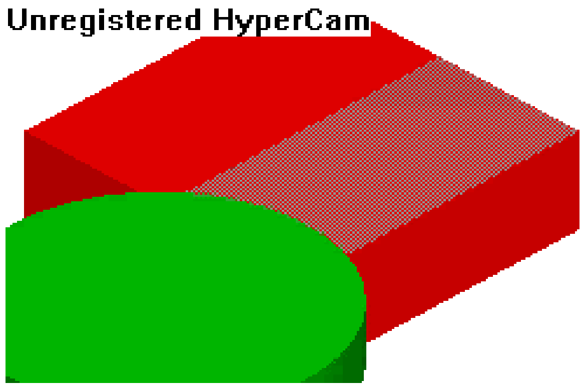
N	**Sequence Number**
G	**Preparatory Functions**
X	**X Axis Command**
Y	**Y Axis Command**
Z	**Z Axis Command**
R	**Radius from specified center**
A	**Angle ccw from +X vector**
I	**X axis arc center offset**
J	**Y axis arc center offset**
K	**Z axis arc center offset**
F	**Feedrate**
S	**Spindle speed**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

CAD/CAM and CNC

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
...
```

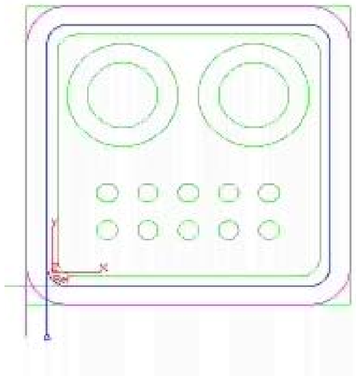




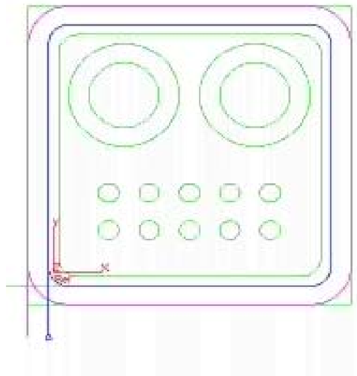
CAD/CAM and CNC

Preparatory functions (inc.)

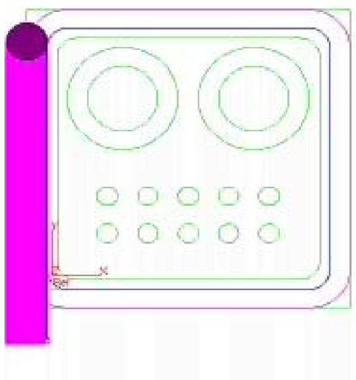
G00 – GO



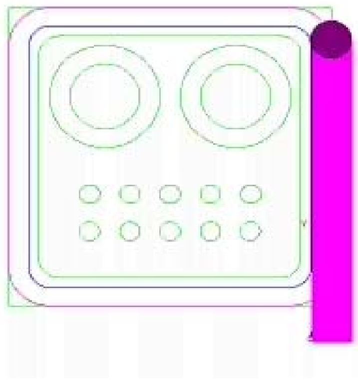
G01 – Linear Interpolation



G02 – Circular Interpolation (CW)



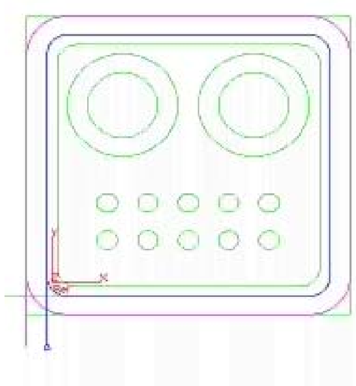
G03 – Circular Interpolation (CCW)



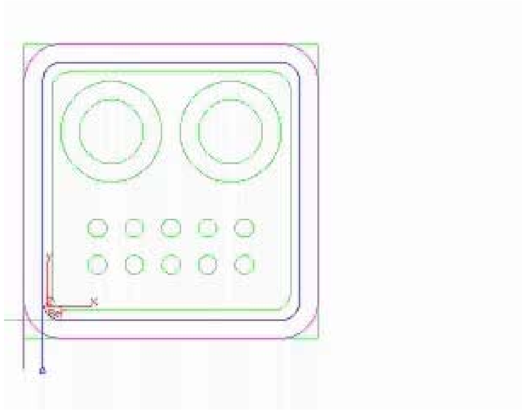
CAD/CAM and CNC

Preparatory functions (inc.)

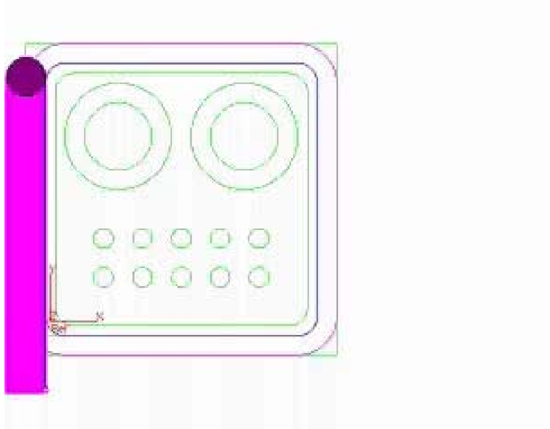
G00 – GO



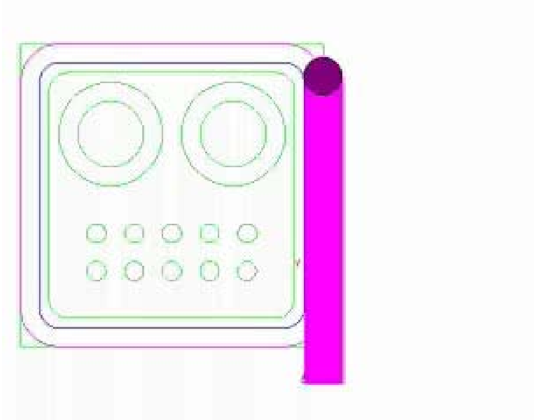
G01 – Linear Interpolation



G02 – Circular Interpolation (CW)



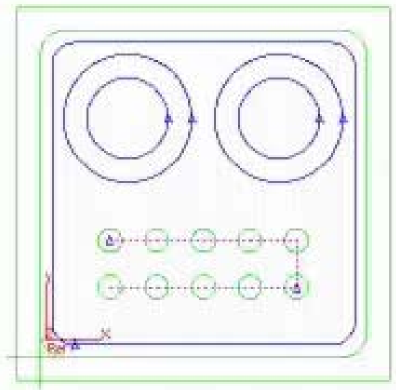
G03 – Circular Interpolation (CCW)



CAD/CAM and CNC

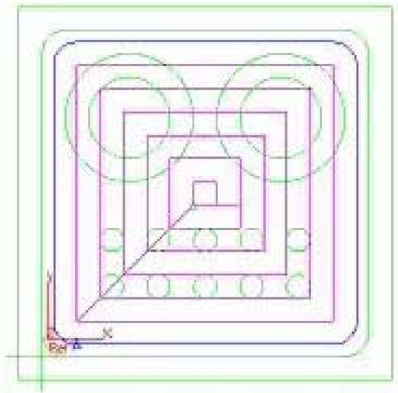
Canned Cycles

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

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)
•…

CAD/CAM and CNC

Miscellaneous functions

- M02 - Program end
- M03 - Start of spindle rotation clockwise
- M04 - Start of spindle rotation counterclockwise
- M07 - Start of mist coolant
- M08 - Start of flood coolant

CAD/CAM and CNC

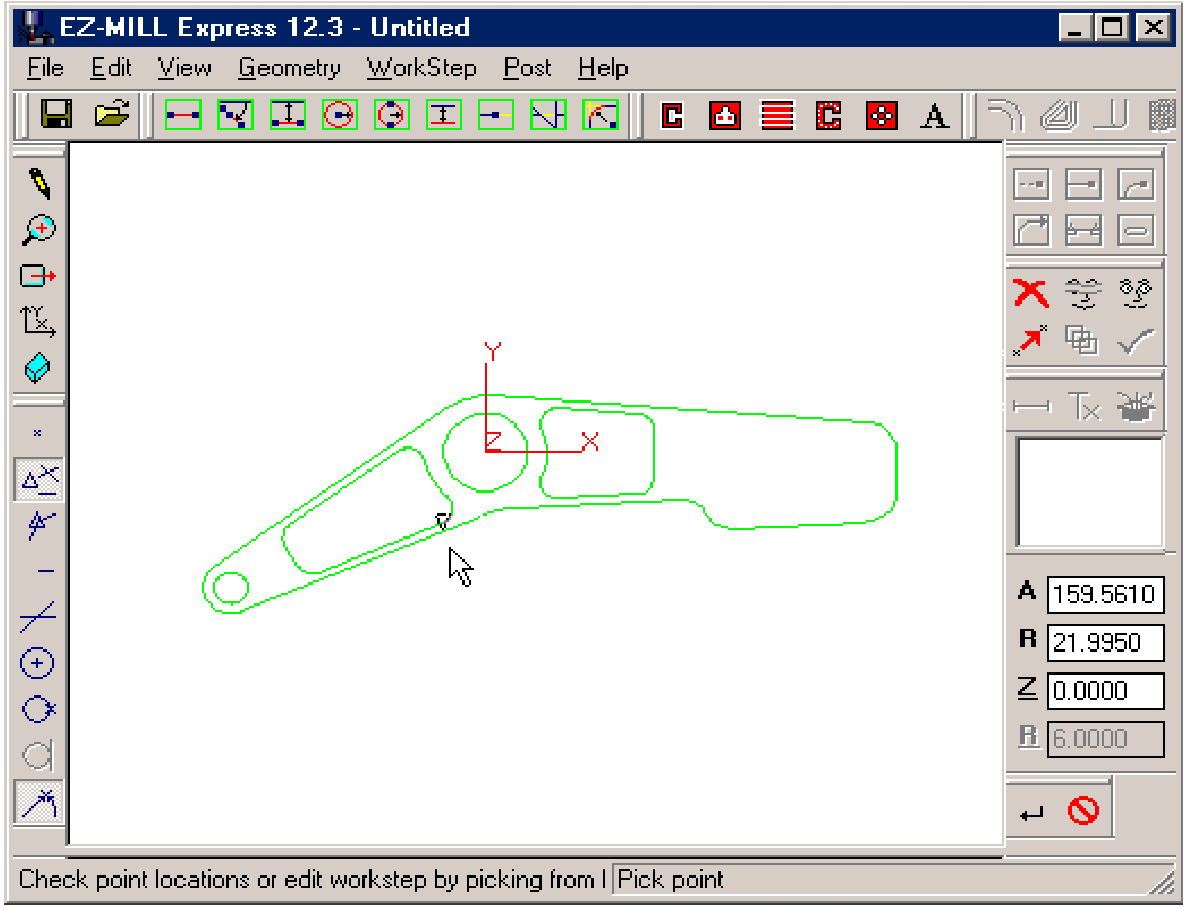
Tool change

Note: should be of easy access, when performed manually.

CAD/CAM and CNC

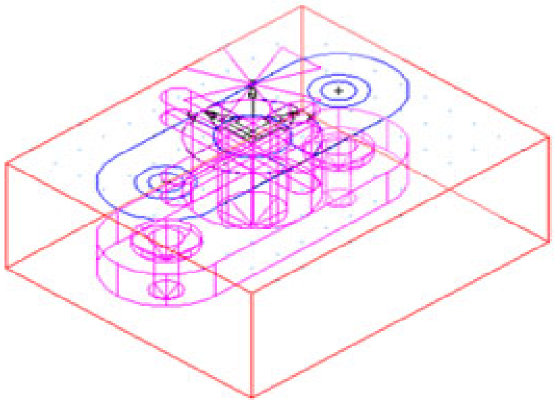
Example of CNC programming

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

CAD/CAM and CNC

Machine operation

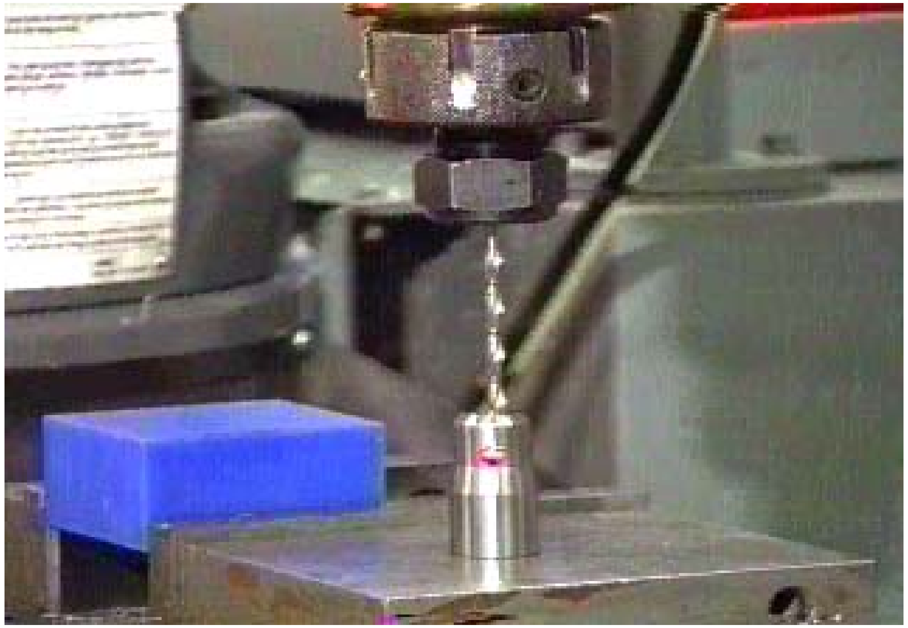
Rules of Security

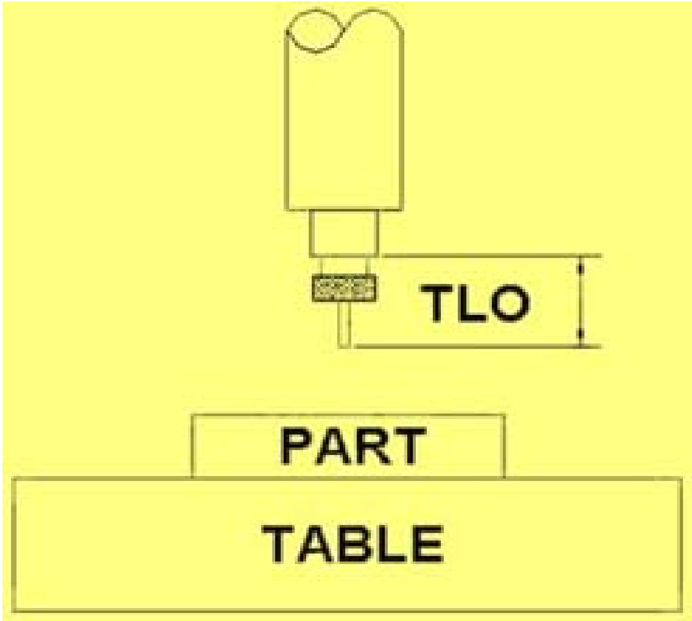
- Security is essential!
- The eyes must be always protected.
- The tools and parts must be handled and installed properly.
- Avoid the use of large cloths
- Clean the parts with a brush. Never with the hands.
- Be careful with you and the others.

CAD/CAM and CNC

Machine operation

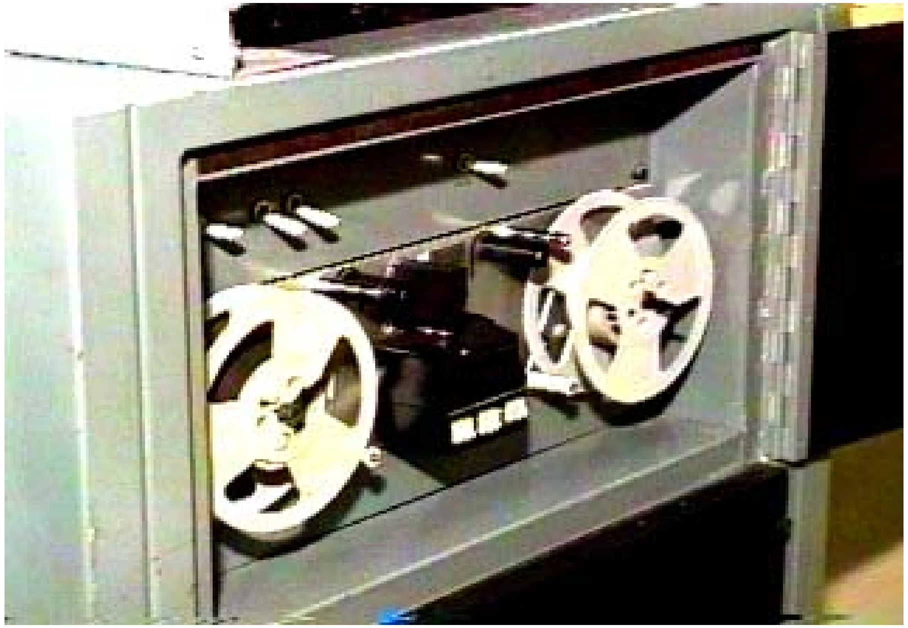
Verify tolerances and tools offsets for proper operation





CAD/CAM and CNC

Machine operation

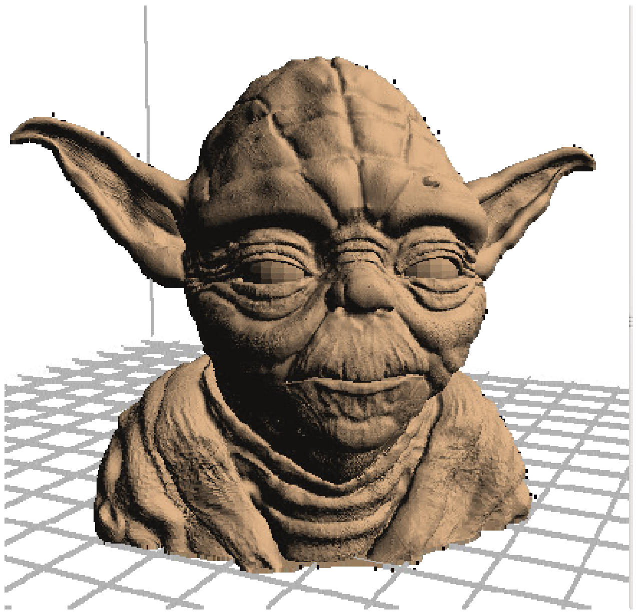


Load program

Follow up machine operation

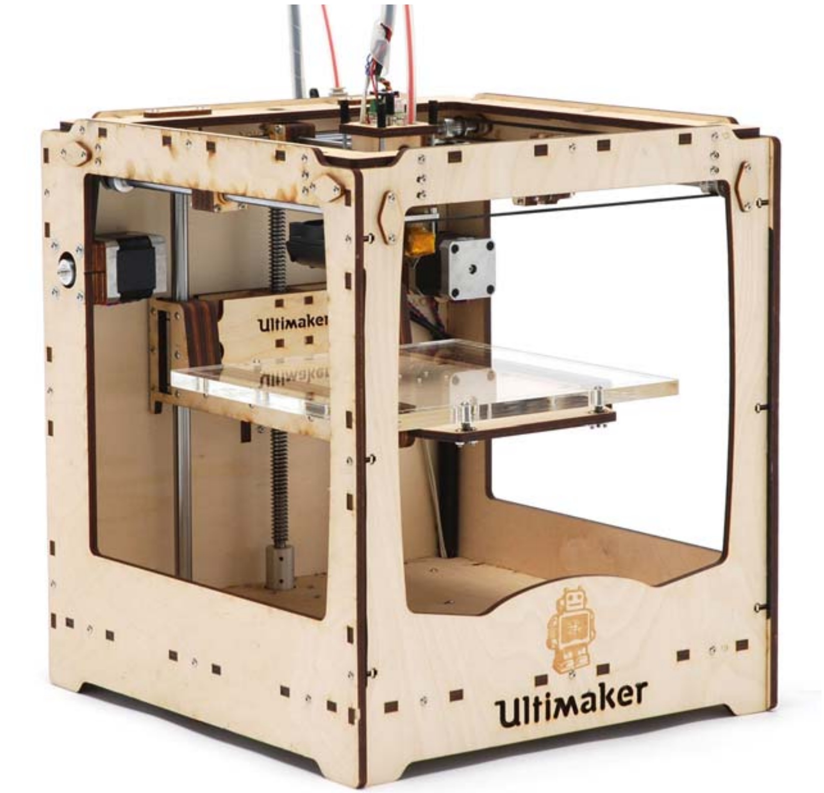
Verify carefully the produced part.

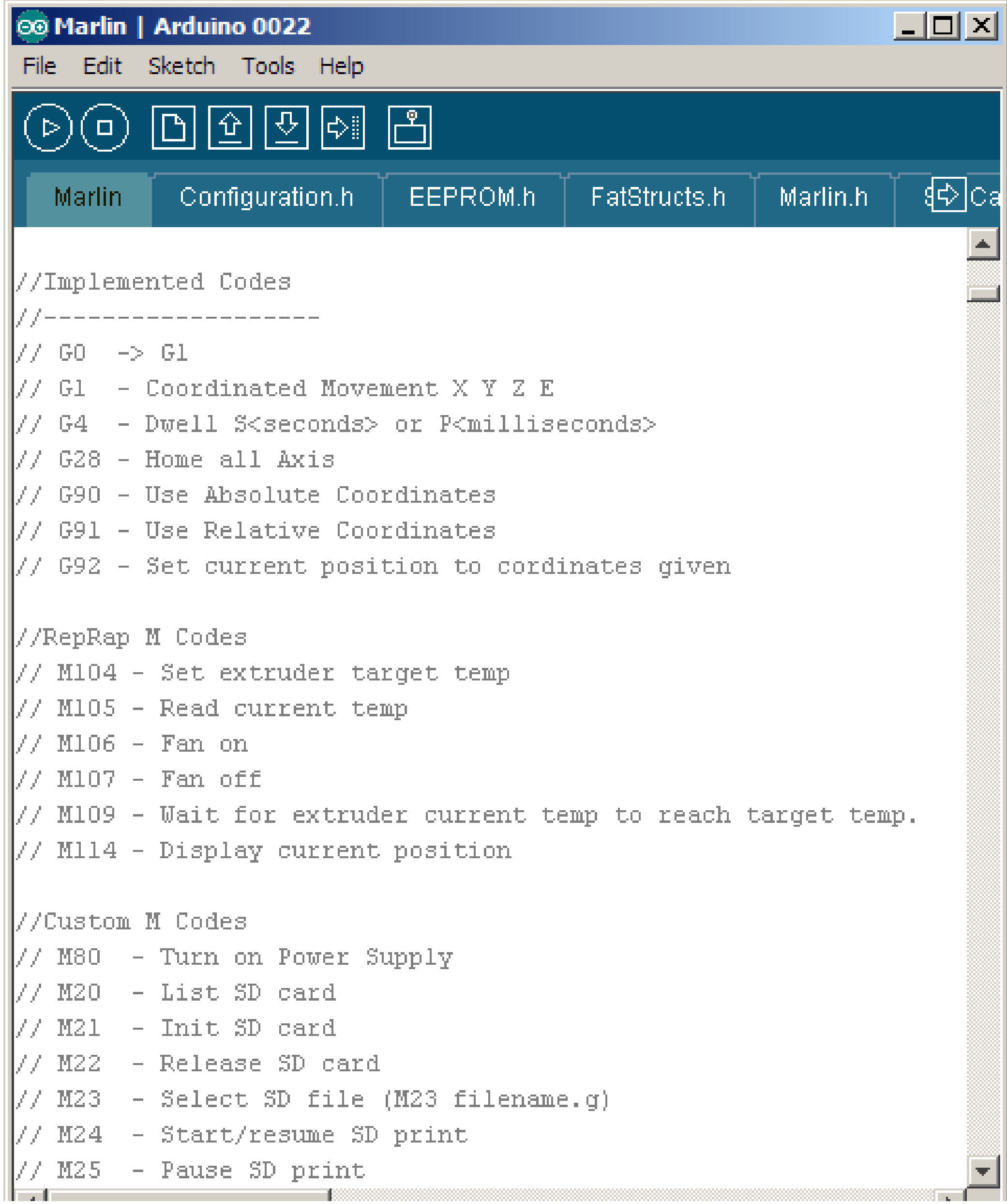
CAD/CAM and CNC at home!



```
;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
```

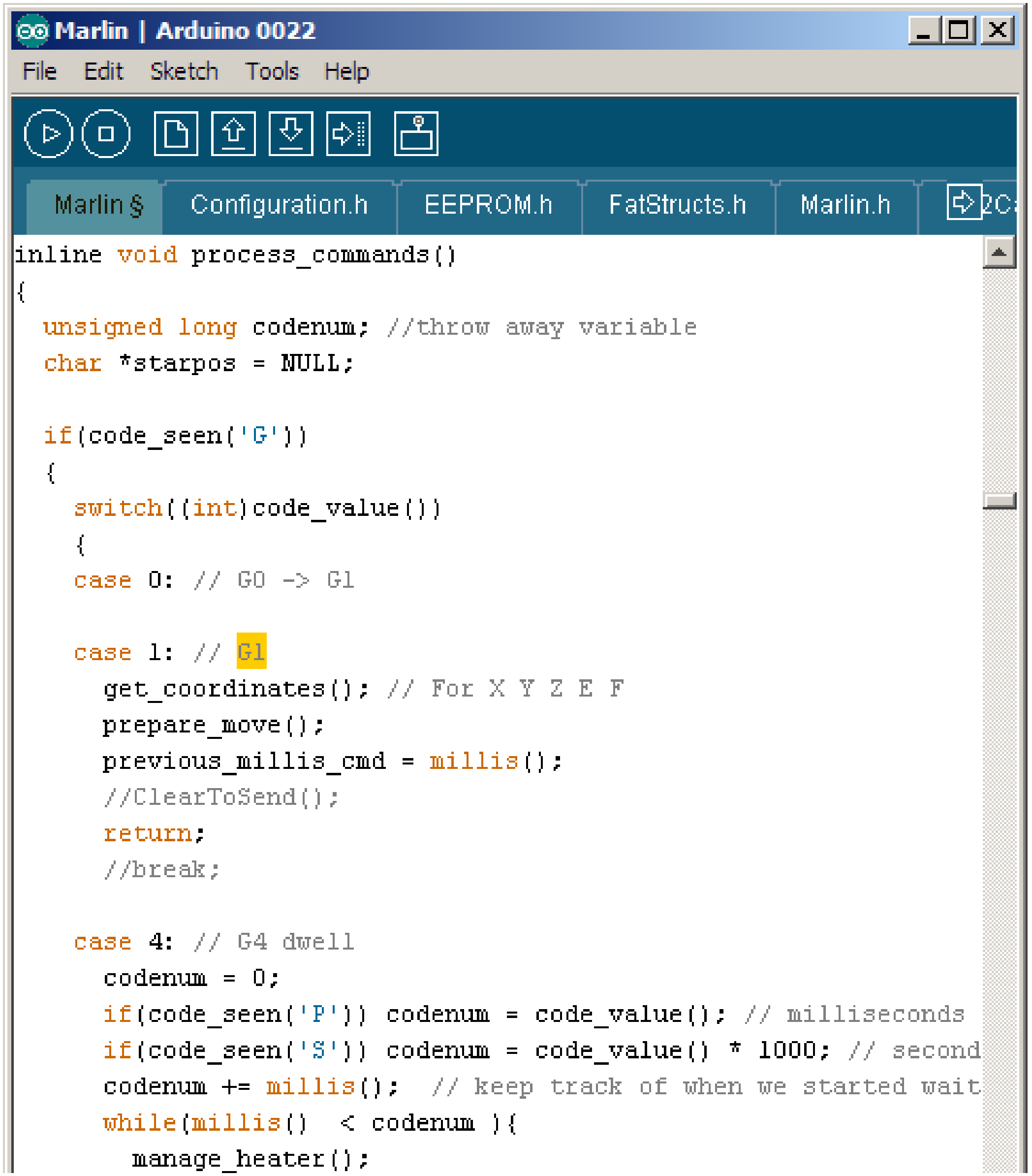
CAD/CAM and CNC at home!





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

CAD/CAM and CNC at home!



CAD/CAM and CNC at home!

```
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!