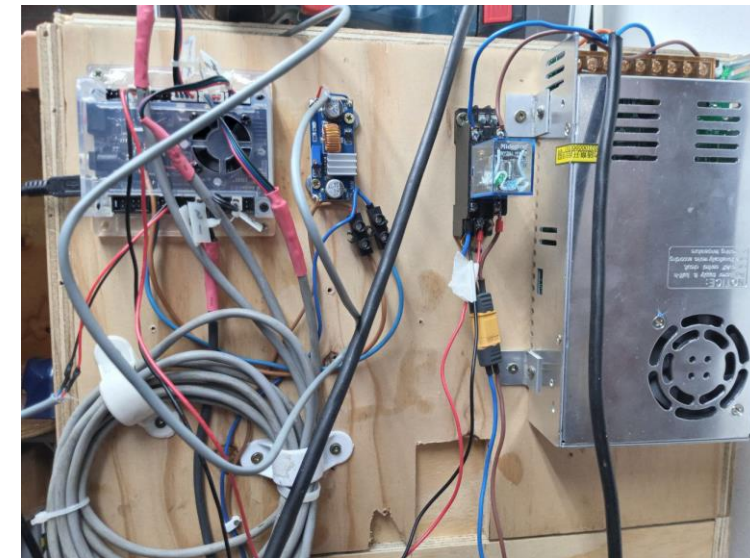
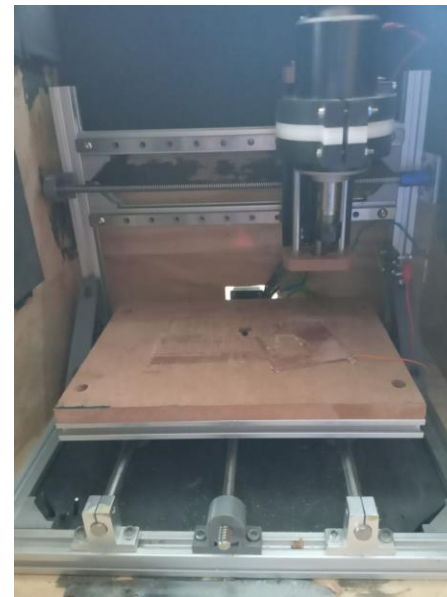
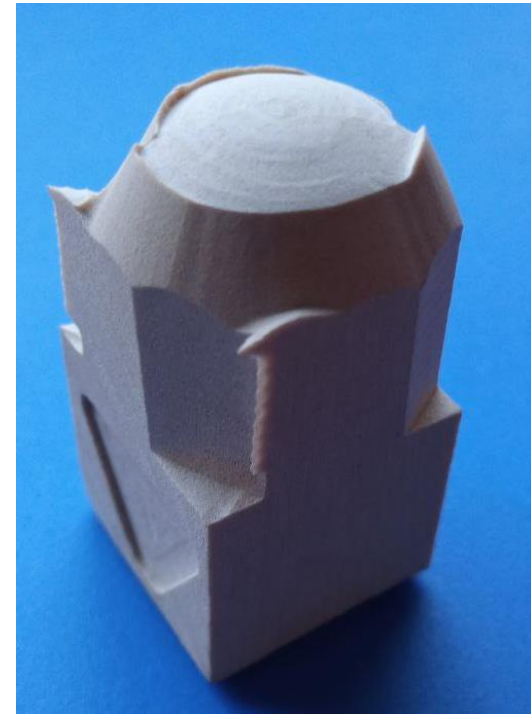


5-axis simultaneous machining on a Minitech Micromill using Fusion360 and Linuxcnc (continuation)

By Danie Gouws Jnr.

Continuation of previous talk

- Audio visual check?
- From DIY 5-axis woodpecker setup
- Prof helped with simultaneous operations on woodpecker 3018
- Disassemble-back to pcb cutting



Original Minitech Mill

- Details:
 - Secondhand, American 5-axis micromill designed for light, high precision lab use.
 - 50000 rpm 180W spindle
 - 5 micron resolution
 - Semiconductor micromachining
 - a-table, b-head configuration
- Original components:
 - 2 control boxes with stepper controllers, analog power supply
 - independent spindle and controller:
 - Very high quality design
 - No limit switches

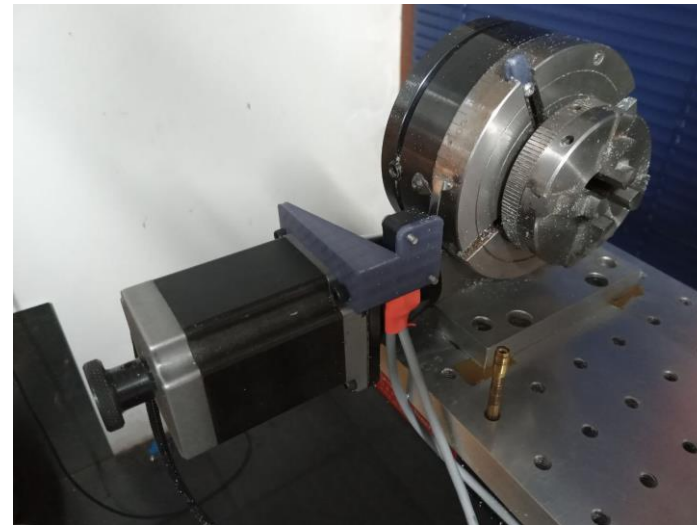
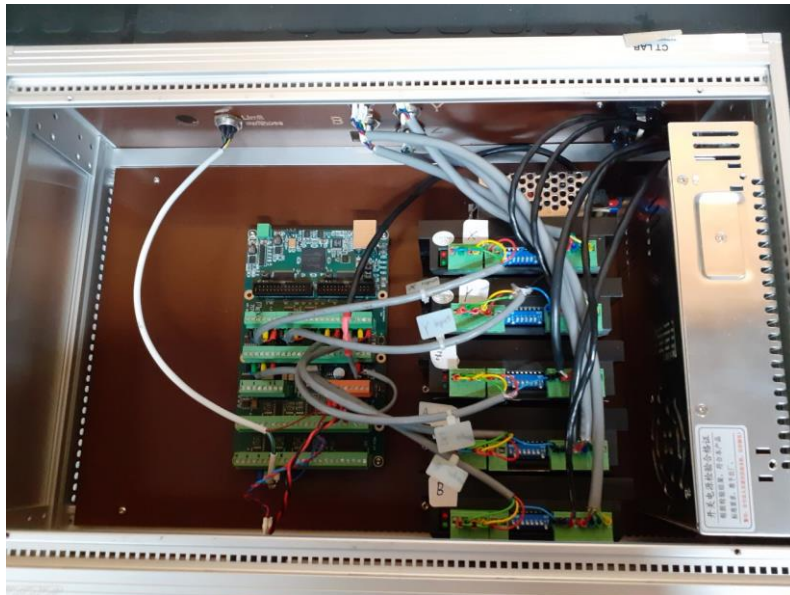


Mach3 or Linuxcnc?

- Originally mach3 controller:
 - Windows XP pc
 - Via parallel port interface x2
- Problems:
 - Display
 - Many irritations
 - Difficult to replace electronics if failure
- Stick with Mach3 or switch to Linuxcnc as on woodpecker?
- Ultimately decide to jump to Linuxcnc
- Cut wires at stepper motors and put all beyond in storage.

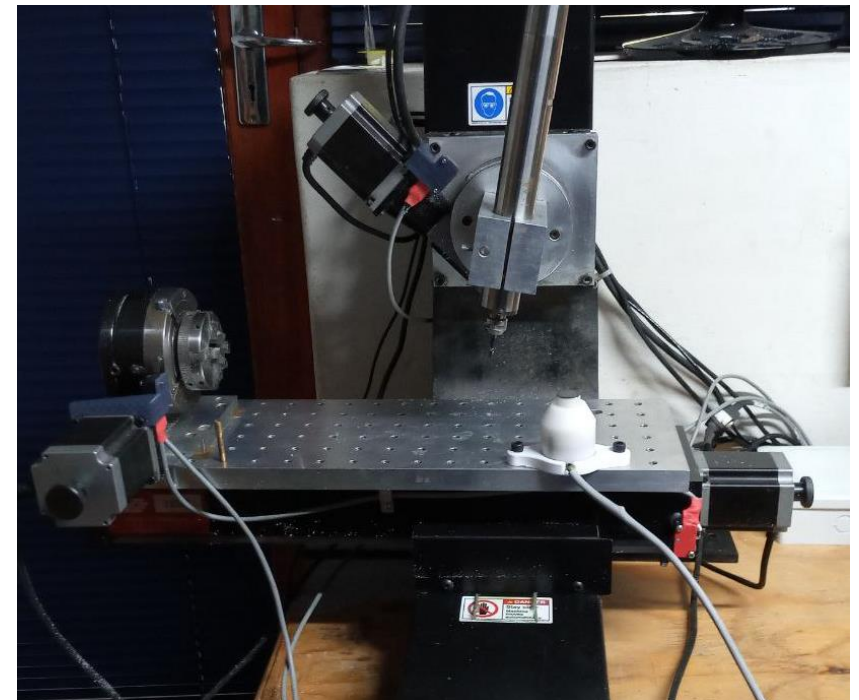
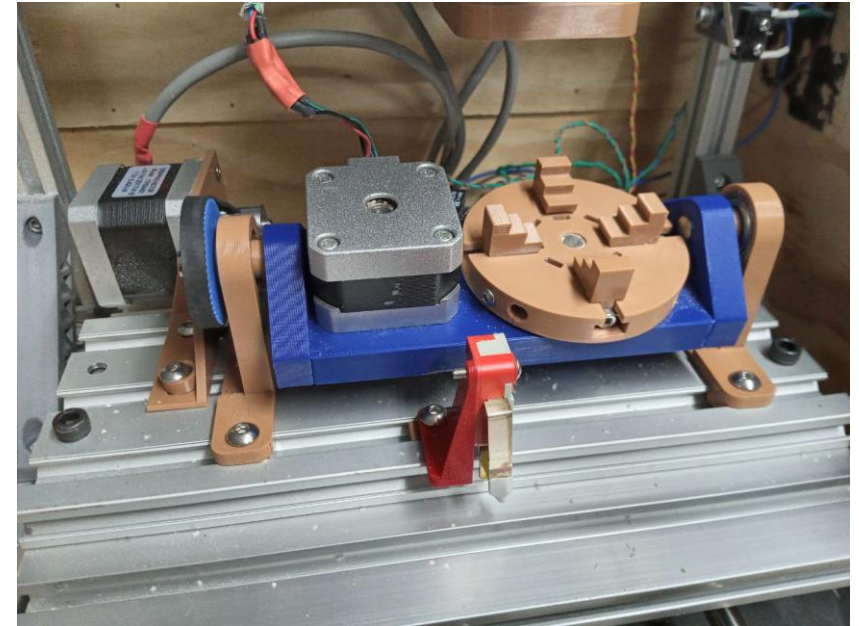
Setup

- Reused control box from woodpecker using Mesa7i76e card
- Add 5 limit switches
- Software setup- .hal and .ini
- Homing sequence, soft limits, etc until each joint works as expected



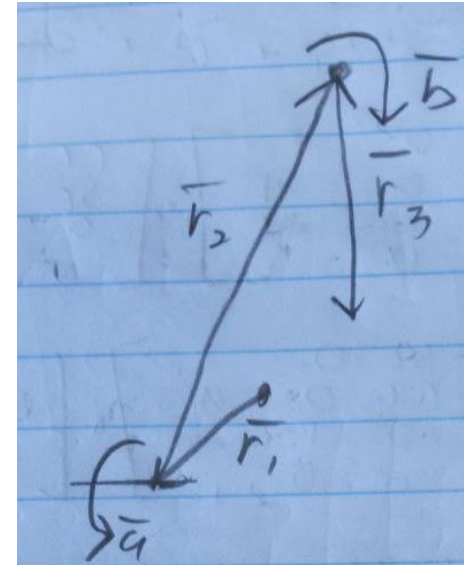
Kinematics required

- Kinematics
 - Relationship between how each joint (stepper motor and range of movement) should move in relation to gcode toolpath produced by standard CAM processor such as fusion360 which describes the position and orientation of the bit.
 - Forward and inverse transformation functions between joints and axes within a c-file. Some trigonometry included
- Woodpecker 3018
 - Trunnion table with a-table, c-table. Readily available.
 - same as for meso-mill currently being developed.
- Minitech
 - a-table, b-head
 - not available in standard linuxcnc.
 - Closest configuration is maxkins.c for c-table, b-head machine.
 - Necessary to derive from scratch



Kinematics derivation and implementation

- Used 33-page document on linuxcnc website by Prof
 - sketch
 - method of derivation
 - rotation matrixes
 - interpretation.
- With some effort managed to derive forward, inverse equations for Minitch.



$$\bar{r}_1 = \begin{bmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\bar{a} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & ca & sa & 0 \\ 0 & -sa & ca & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\bar{r}_2 = \begin{bmatrix} 1 & 0 & 0 & x_2 \\ 0 & 1 & 0 & y_2 \\ 0 & 0 & 1 & z_2 \end{bmatrix}$$

$$\bar{b} = \begin{bmatrix} cb & 0 & -sb & 0 \\ 0 & 1 & 0 & 0 \\ sb & 0 & cb & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} Q_x &= -sb \cdot z_3 + x_2 + x_1 \\ Q_y &= sa \cdot cb \cdot z_3 + ca \cdot y_2 + sa \cdot z_2 + y_1 \\ Q_z &= ca \cdot cb \cdot z_3 - sa \cdot y_2 + ca \cdot z_2 + z_1 \end{aligned}$$

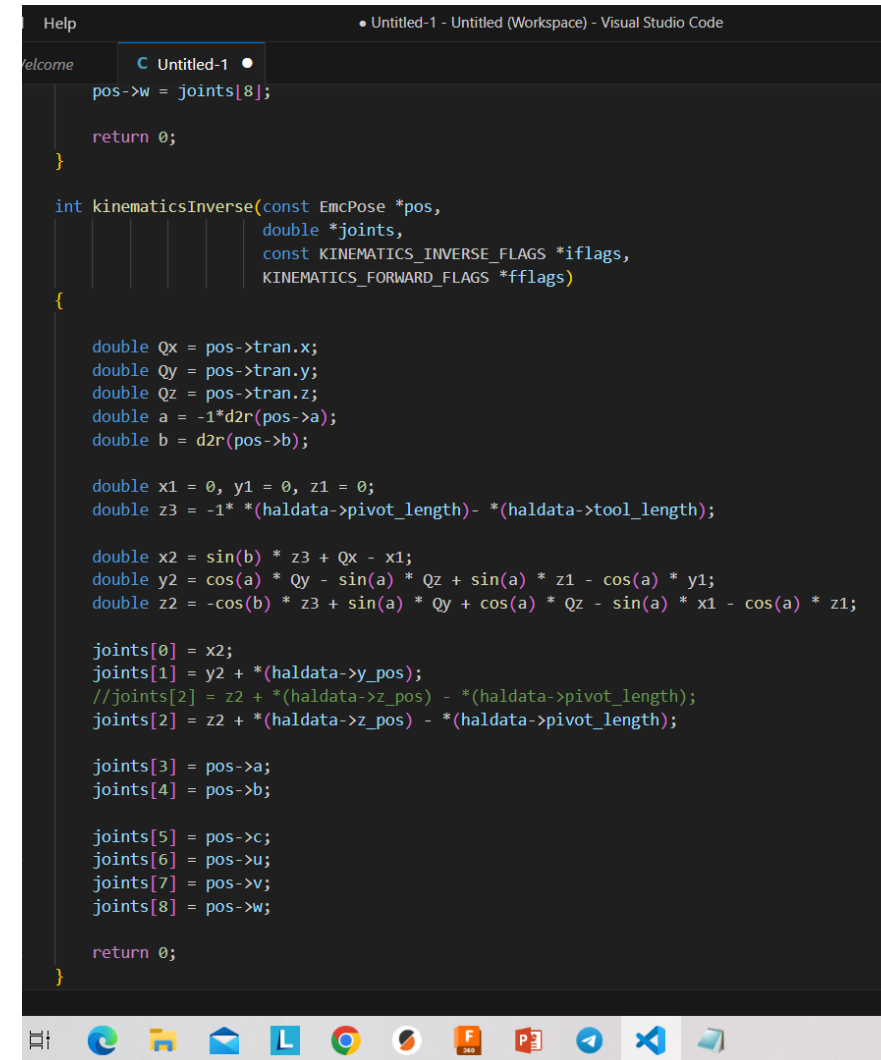
Forward transformation

$$\begin{aligned} x_2 &= sb \cdot z_3 + Q_x - x_1 \\ y_2 &= ca \cdot Q_y - sa \cdot Q_z + sa \cdot z_1 - ca \cdot y_1 \\ z_2 &= -cb \cdot z_3 + sa \cdot Q_y + ca \cdot Q_z - sa \cdot x_1 - (a \cdot z_1) \end{aligned}$$

Inverse transformation

Kinematics c file implementation

- Then interpret equations and implement into a c-file.
 - Used format of maxkins.c as template from linuxcnc github.
 - Added custom kinematics in windows code editor.
 - Trig equations within functions called by motion controller during run time.
 - Then took c-file to linuxcnc. Use halcompile on c-file through console. Specify in .ini file and run machine.



```
Help
• Untitled-1 - Untitled (Workspace) - Visual Studio Code
C Untitled-1
pos->w = joints[8];

return 0;
}

int kinematicsInverse(const EmcPose *pos,
                    double *joints,
                    const KINEMATICS_INVERSE_FLAGS *iflags,
                    KINEMATICS_FORWARD_FLAGS *fflags)
{
    double Qx = pos->tran.x;
    double Qy = pos->tran.y;
    double Qz = pos->tran.z;
    double a = -1*d2r(pos->a);
    double b = d2r(pos->b);

    double x1 = 0, y1 = 0, z1 = 0;
    double z3 = -1*(haldata->pivot_length) - *(haldata->tool_length);

    double x2 = sin(b) * z3 + Qx - x1;
    double y2 = cos(a) * Qy - sin(a) * Qz + sin(a) * z1 - cos(a) * y1;
    double z2 = -cos(b) * z3 + sin(a) * Qy + cos(a) * Qz - sin(a) * x1 - cos(a) * z1;

    joints[0] = x2;
    joints[1] = y2 + *(haldata->y_pos);
    //joints[2] = z2 + *(haldata->z_pos) - *(haldata->pivot_length);
    joints[2] = z2 + *(haldata->z_pos) - *(haldata->pivot_length);

    joints[3] = pos->a;
    joints[4] = pos->b;

    joints[5] = pos->c;
    joints[6] = pos->u;
    joints[7] = pos->v;
    joints[8] = pos->w;

    return 0;
}
```

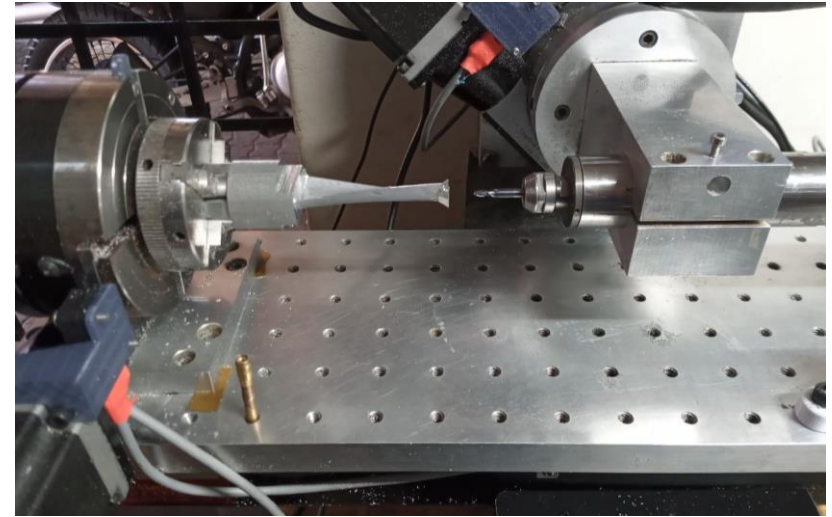
Debugging

- Vismach virtual machine
 - simulate machine axis movement for debugging.
 - Did not look into.
- Tested on actual machine but spindle replaced by pencil.
- “joint following error” indicate error in mathematics
 - point transformed from axis coordinates to joint and then back to axis do not agree
 - equations inconsistent.
- Found error in matrixes, corrected. Problems with positive direction of rotary axes. Works correctly.

Alignment and tool length probing

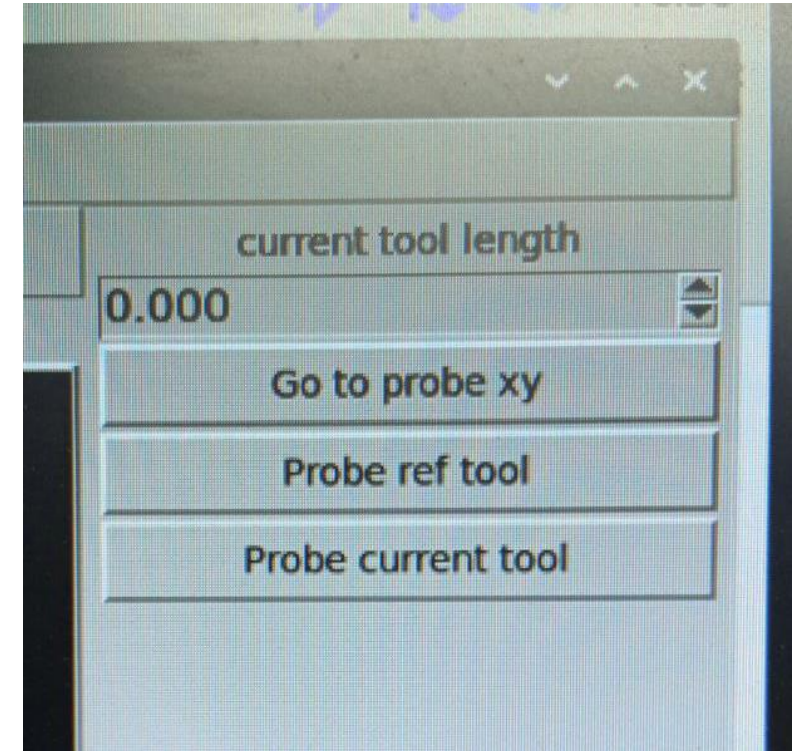
Alignment crucial for correct rotation:

- Position of a-axis in y, z joint coordinates
 - length from b-axis to tool-point
 - verticality of rotary axes
 - Start with only 1 tool
- Add tool offsets. Seems simple. Just add to length between pivot and tool point
 - Using tool-probe built by Dr. Des Horn
 - In Practice (from meso-mill)
 - Page of intense gcode for the probing subroutine/remap of M6
 - custom user control panel on Axis interface requires xml code
 - dozens of hal pins to connect various parts together.



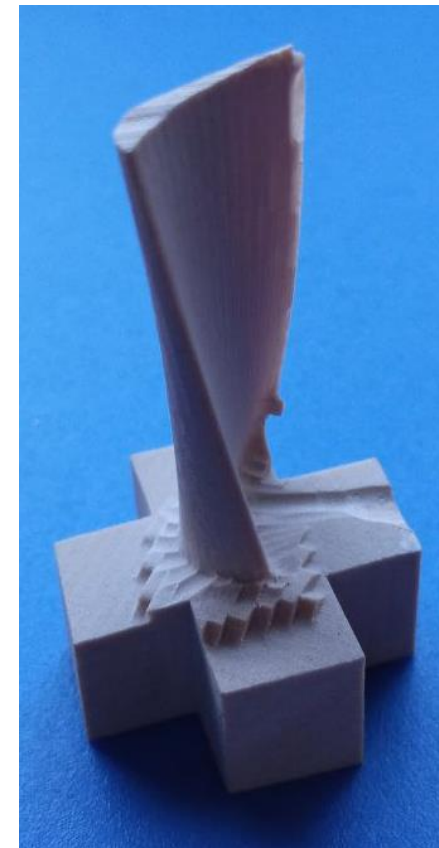
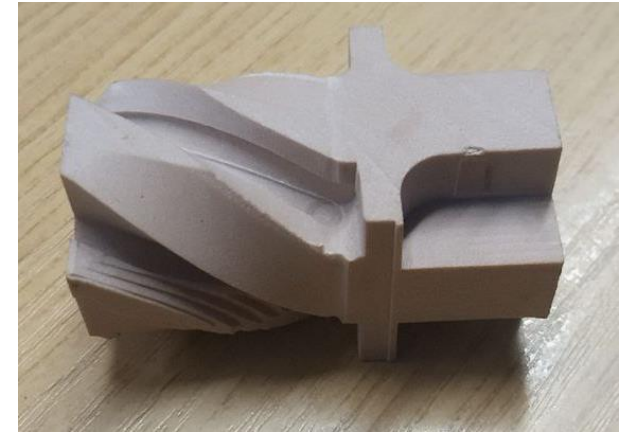
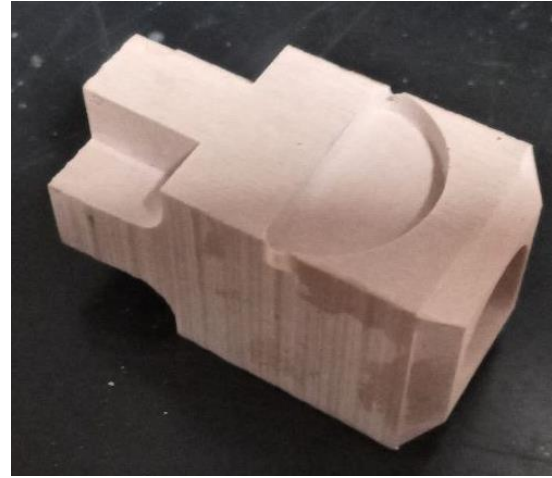
My tool length probe subroutine

- Ideally, use tool-probe subroutine from meso-mill
 - complete and actively supported set of code files.
- Not yet done. Used as reference.
- Used own setup.
 - 3 custom buttons
 - numeric input box connected to variable within kinematics file.
- Buttons
 - move to probe position
 - probe ref tool (spindle with collet removed)
 - probe current tool.
- G59, Halpin variable, spinbox widget
- Required manual enter into control panel, causes following from z jump error but works. Not integrated with tool table. Need to look at meso files. But working.



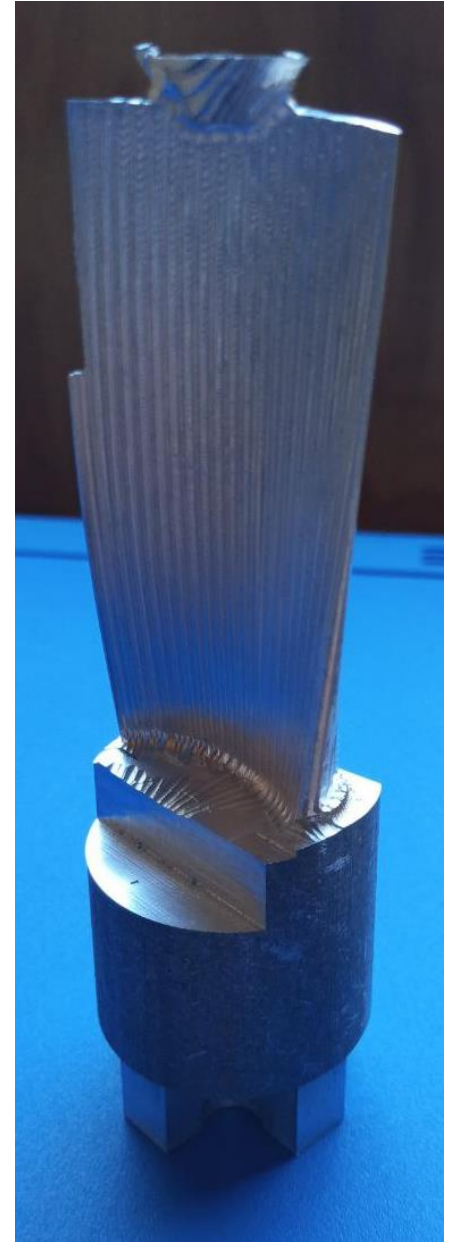
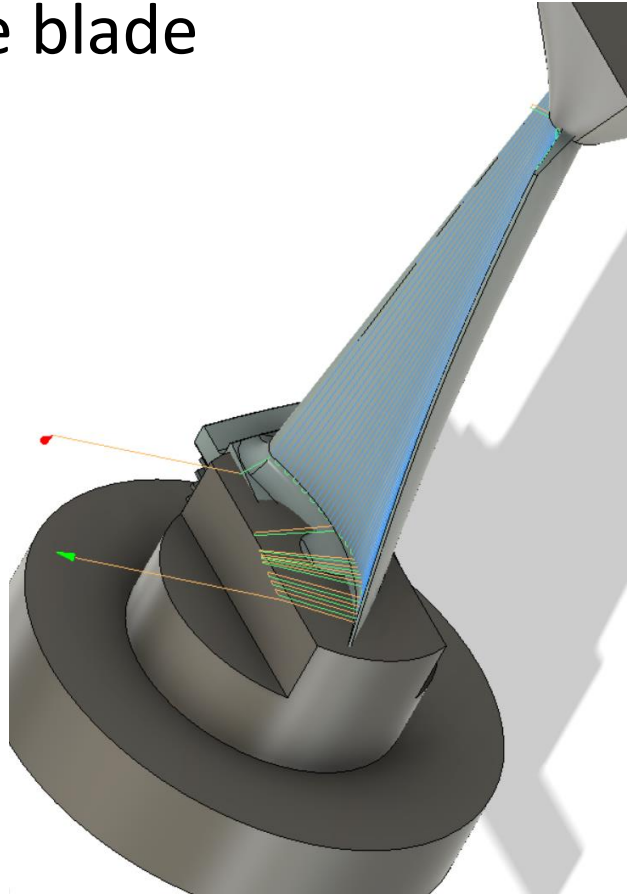
Test cuts

- Rough with indexed pockets.
- **material**
- Finish with 5-axis operations
- Corkscrew to test swarf



Aluminium cuts

- Same turbine blade model



Current situation

- Currently have working 5-axis system.
- Still need to connect spindle to Mesa card for automated control.
- Works best for soft material, small volume cuts. Cuts slowly for aluminium. Perhaps rough on machine like tormach, finish on Minitech for speedup.

Problems

- 5-axis Toolpaths between operation
 - Requires post-processor edits
 - Difficult to get working for every type of operation, with/without tailstock
- Connection of HAL variables to gcode:
 - Save z-coordinate into Halpin when command is issued within gcode subroutine
- Enclosure, fogbuster coolant, vacuum shoe...