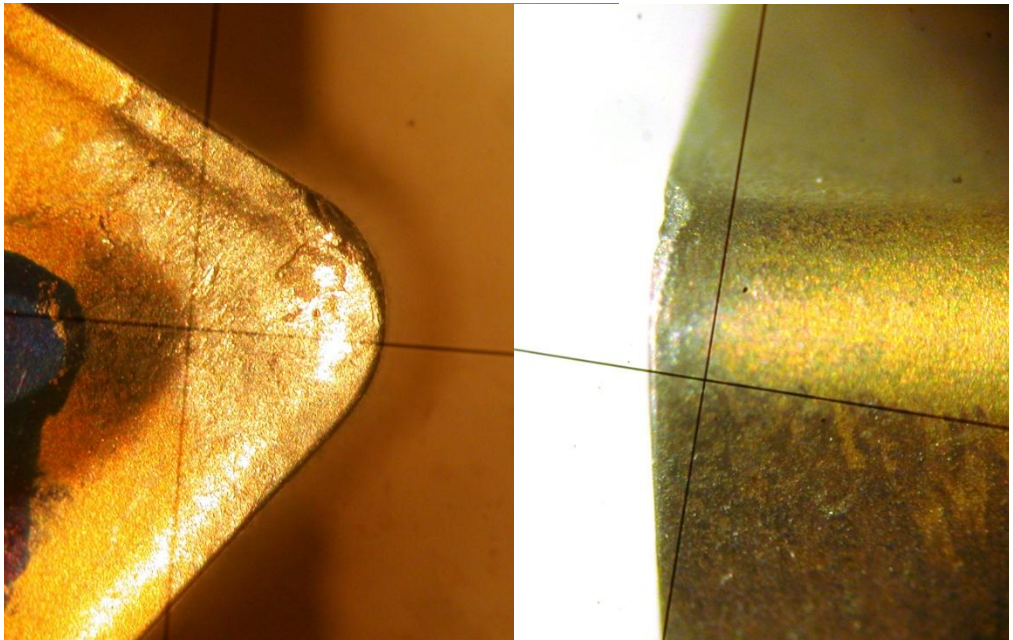
Monitoring and Control System for Finishing by Turning

Case: Tool Wear Monitoring in Interrupted Cutting

What is a Small Tool Fracture?

- Background:
  - A small fracture in interrupted turning is difficult to detect
    - → May effects surface quality
    - → May effect shape
- Additionally, some wear quickly leads to complete tool destruction

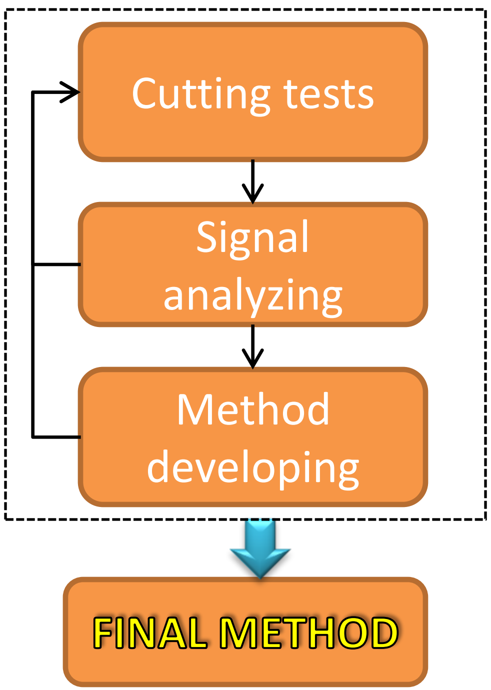
- Challenges:
  - Detecting the small fracture
  - Interrupted turning
  - Fast detection
  - Reliability



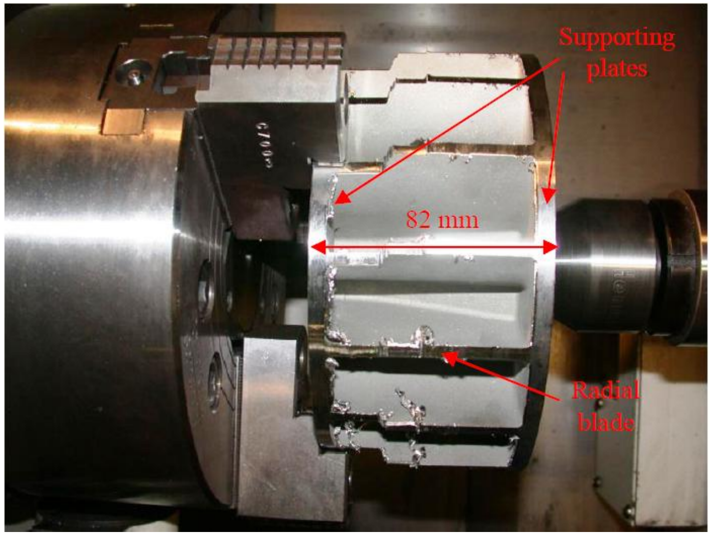
Design of Experiments

Cutting parameters

Factor	Value
Cutting Speed v_c [m/min]	*45 – 65*
Feed f [mm/rev]	*0.2 – 0.4*
Depth of Cut a [mm]	*0.5 – 1.5*
Insert Status	*New / Damaged / Broken*



- Insert: CNMG 120412-MR 2025/2035
- Tool holder: Sandvik PCLNL 2525M12 (positioning angle 95˚)
- Work piece: impeller, duplex steel G-X 2 CrNiMoN 25 6 3
- 6 mm cuts at 20 kHz (individual rotations measured, sample length varies by rotational speed)



- Project work by Mr. Andrea Vian
  - Multiple ways to detect

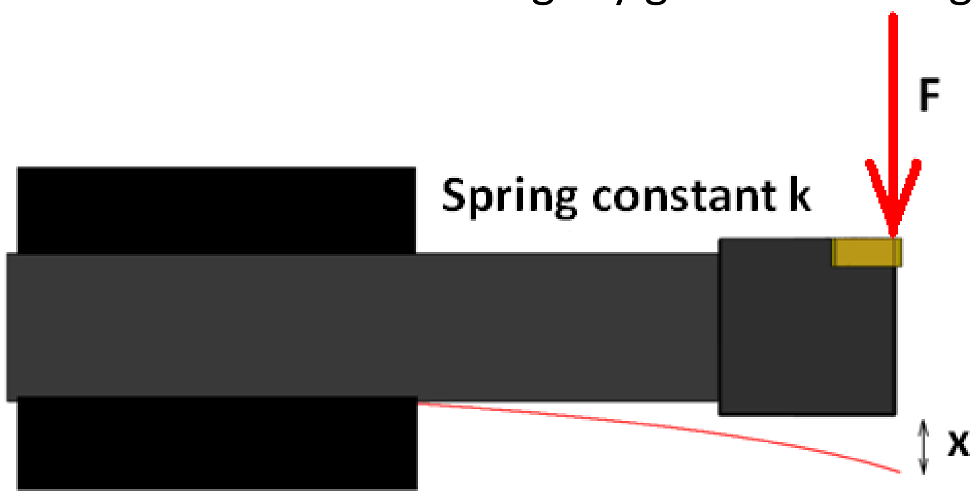
    → On the average 80 % correct

    → However, sensitive to change is conditions

OBSERVATION: Measure tool impact instead of trying to analyze cutting

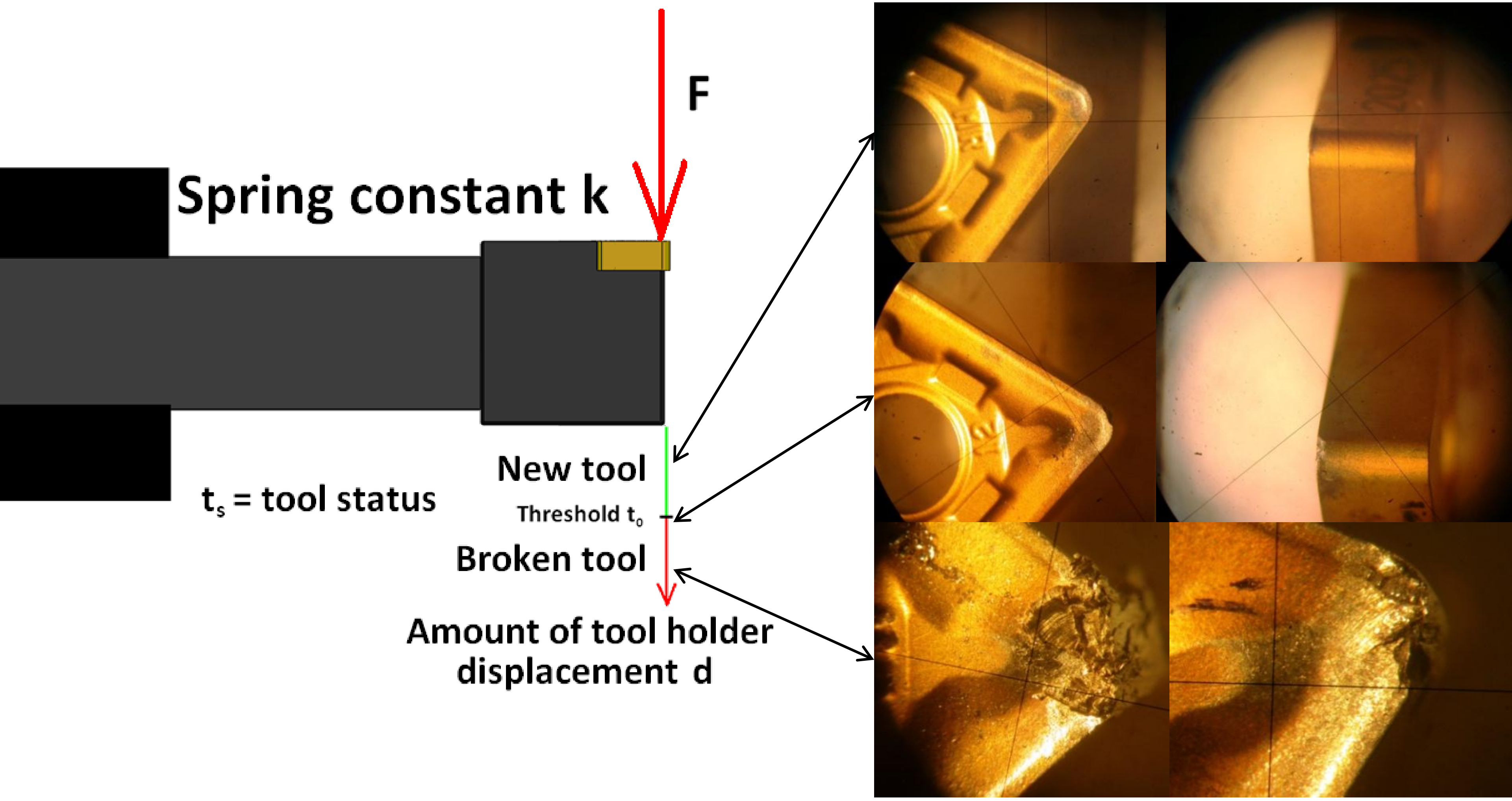
→ **A NEW METHOD BASED ON TOOL HOLDER DISPLACEMENT**

- The deal:
  - Assume that the tool holder is a very stiff spring. When the cutting edge is no longer sharp, cutting forces increase.
  - This translates into slightly greater bending of the tool holder on impact



$$F = -kx$$

Tool breakage in pictures

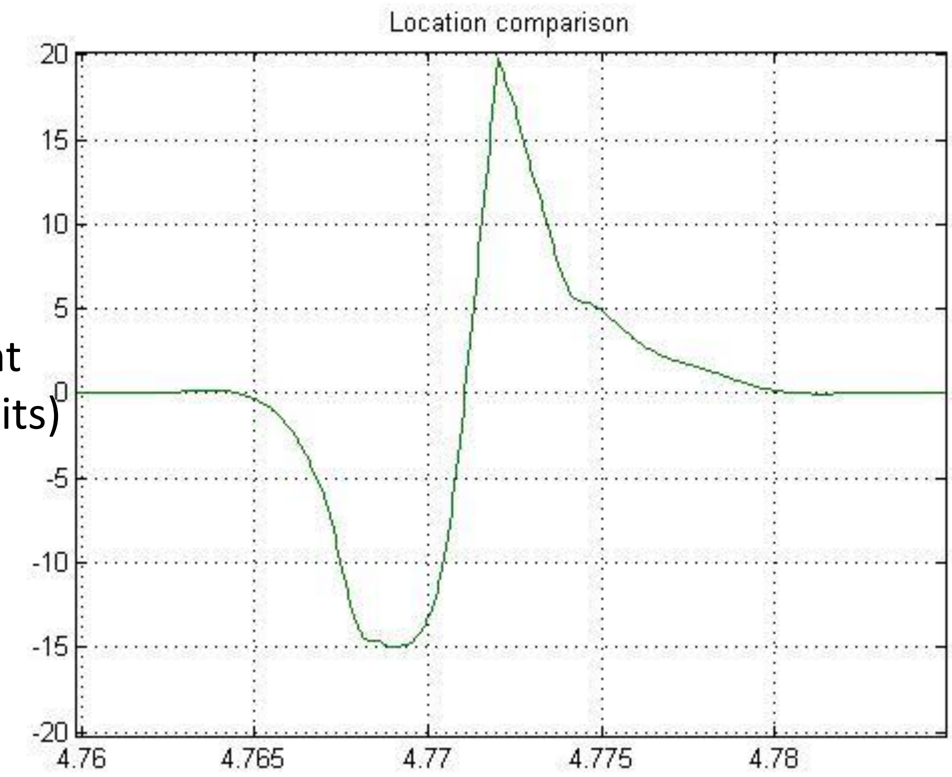


From Data to Information

- Acceleration sensor data very noisy
- Variable measurement error (zero point drifts)
- However, since the cutting (or in this case, punching) is interrupted regularly, tool position can be calibrated.
- Numerically integrate acceleration data to get tool position.
- Spring constant unknown, so use displacement data (no force estimate)

Example tool location result

Displacement
(arbitrary units)



Time

From Information to Knowledge

- Assume tool holder displacement is a function of cutting speed (v), depth of cut (a) and feed rate (f) as well as cross effects
  - Data available from lathe controller via Ethernet
- Estimate tool displacement

$$d(v_c, a_c, f_c, t) = C_0 + C_1 v_c + C_2 a_c + C_3 f_c + C_4 t + C_{12} v_c a_c + C_{13} v_c f_c + C_{14} v_c t + C_{23} a_c f_c + C_{24} a_c t$$

- Based on tool holder displacement, calculate tool status

$$t_s(v_c, a_c, f_c, t_0, d) = d - C_0 - C_1 v_c - C_2 a_c - C_3 f_c - C_4 t - C_{12} v_c a_c - C_{13} v_c f_c - C_{14} v_c t - C_{23} a_c f_c - C_{24} a_c t$$

Results

- Observation: Cutting speed has no effect on tool holder bending
  - This is consistent with the idea that the process is more like punching than cutting
- By using the value of 0.4 as a threshold for a slightly broken tool, and requiring two of three latest samples to be detected as broken, it was possible to identify small tool fracture at a 91% certainty.

Results, continued

- First and last measurements of an experiment were often misclassified (partial cut). Not counting these, 355 of 369 samples were correctly identified.
- Result calculated very fast (under three seconds from the start of first faulty measurement)
  - Implementation is a combination of Matlab and Win32 binary (small program to access the lathe control via DLL). Likely very slow.
  - Sample time, of course, is a major factor