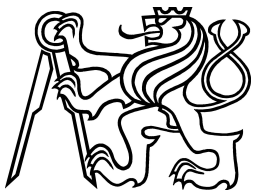


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA DOPRAVNÍ

Nela Fenclová

NÁSTROJE PRO ŘÍZENÍ A VYHODNOCOVÁNÍ MIKROINDENTAČNÍCH ZKOUŠEK

Diplomová práce

2015

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta dopravní
d ě k a n
Konviktská 20, 110 00 Praha 1

K618 .....Ústav mechaniky a materiálů

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Bc. Nela Fenclová

Kód studijního programu a studijní obor studenta:

N 3710 – ID – Inženýrská informatika v dopravě a spojích

Název tématu (česky): **Nástroje pro řízení a vyhodnocovaní mikroindentačnich zkoušek**

Název tématu (anglicky): Tools for Control and Evaluation of Microintentation Tests

Zásady pro vypracování

Při zpracování diplomové práce se řiďte osnovou uvedenou v následujících bodech:

- Tvrdost zjištěná na základě mikroindentačních zkoušek je jednou ze základních vlastností pevných látek, jakou můžeme uvádět pro jejich charakteristiku a srovnávání.
- Cílem diplomové práce je vytvoření nástroje pro řízení a vyhodnocovaní instrumentovaných mikroindentačních zkoušek pro mikroindentor sestrojený na ústavu K618 a zhodnocení přesnosti tohoto zařízení, případně návrh na jeho další úpravy.
- Řídicí software bude využívat realtime Linuxového jádra a jazyka pro programování CNC strojů (G-kód).
- Vyhodnocovací softwareový nástroj bude umožňovat tvorbu map lokálních změn mikrotvrdosti povrchu vzorků.

Rozsah grafických prací: Dle pokynů vedoucího

Rozsah průvodní zprávy: minimálně 55 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury:

P.E. Mix: Introduction to Nondestructive Testing: A Training Guide, Wiley-Interscience, 2005

A.C. Fischer-Cripps, Nanoindentation, Springer, 2004

W.C. Oliver, G.M. Pharr, J. Mater. Res. 7 (1992) 1564

W.C. Oliver, G.M. Pharr, J. Mater. Res. 19 (2004) 3

LinuxCNC Documentation, http://linuxcnc.org/docs/2.5/

Vedoucí diplomové práce: **Ing. Daniel Kytýř, Ph.D.**
Ing. Tomáš Fíla

Datum zadání diplomové práce: **28. července 2014**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání diplomové práce: **31. května 2015**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia

L. S.

prof. Ing. Ondřej Jiroušek, Ph.D.
vedoucí
Ústavu mechaniky a materiálů

prof. Dr. Ing. Miroslav Svítek
děkan fakulty

Potvrzuji převzetí zadání diplomové práce.

Bc. Nela Fenclová
jméno a podpis studenta

V Praze dne 28. července 2014

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně, a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Nemám závažný důvod proti užívání tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 30. května 2015 Podpis:

Poděkování

Na tomto místě bych ráda poděkovala Ing. Danielu Kytýřovi, Ph.D. a Ing. Tomáši Fílovi za jejich náměty, rady, připomínky a vstřícnost, kterou mi v průběhu psaní práce poskytovali.

Bylo mi potěšením spolupracovat s Mgr. Veronikou Petráňovou, díky které byla pořízena obrazová data z řádkovacího elektronového mikroskopu.

Dále bych ráda poděkovala své rodině a přátelům, kteří mi byli oporou při psaní této práce.

Práce vznikla za podpory Studentské grantové soutěže ČVUT číslo SGS15/225/OHK2/3T/16, rozvojového grantu 1051505F014 RPMT 2015 43e a institucionální podpoře RVO: 68378297.

V Praze, 30. května 2015

Nela Fenclová

Název práce:	Nástroje pro řízení a vyhodnocování mikroindentačních zkoušek
Autor:	Nela Fenclová
Studijní program:	Technika a technologie v dopravě a spojích
Obor:	Inženýrská informatika v dopravě a spojích
Druh práce:	Diplomová práce
Vedoucí práce:	Ing. Daniel Kytýř, Ph.D. Ústav mechaniky a materiálů, Fakulta dopravní, České vysoké učení technické v Praze Ing. Tomáš Fíla. Ústav mechaniky a materiálů, Fakulta dopravní, České vysoké učení technické v Praze
Abstrakt:	Předložená práce se zabývá návrhem, výrobou a zprovozněním indentačního zařízení schopného indentace při nízkém zatížení metodou Vickers. Za tímto účelem byly navrženy dva způsoby zatěžování, první mód pro zatěžování konstantní silou při osazení indentoru závažím a druhý mód pro zatěžování řízené silou. Pro řízení indentačního zařízení technologií CNC byl vybrán modul LinuxCNC verze 2.6.4. Pro oba způsoby zatěžování vzniklo vlastní uživatelské rozhraní. Pro kontrolu správné funkce indentoru a vyhodnocovacích procedur bylo provedeno měření na referenční tvrdoměrné destičce. Toto měření dokázalo plnou funkčnost indentačního zařízení při nízkých zatěžovacích silách. Chyba měření byla mnohem menší, než chyba předepsaná normou.
Klíčová slova:	instrumentovaná indentace, tvrdost podle Vickerse, řízení laboratorních zařízení, automatizované měření tvrdosti

Title: Tools for Control and Evaluation of Microindentation Tests

Abstract: The thesis deals with development of control and evaluation tools for micro and low load indentation tests. Presented experimental setup allows to perform automated indentation testing in i) dead weight testing mode and ii) force-driven instrumented indentation. Custom designed instrumented indentation device was equipped by a control software based on GNU/Linux with real-time kernel. Stand alone application for device control was developed using numerical control programming language together with glade virtual control panel for graphic user interface. Calibration measurement using reference hardness block proven a high reliability of the experimental setup. Overall measurement error was significantly lower than error prescribed by the standards.

Keywords: instrumented indentation, Vickers hardness, control of measuring equipment, hardness measurement automatization

Seznam použitých veličin a symbolů

symbol	jednotka	veličina
F	[N]	síla
d	[mm]	průměr vtisku
D	[mm]	průměr Brinellova indentoru
A	[mm^2]	plocha vtisku
HBS		tvrdost podle Brinella s indentorem z kalené oceli
HBW		tvrdost podle Brinella s indentorem z tvrdokovu
HM		tvrdost podle Meyera
HR		tvrdost podle Rockwella
F_0	[N]	předběžné zatížení
HV		tvrdost podle Vickerse
d_u	[mm]	průměr délky úhlopříček u Vickersovy zkoušky tvrdosti
l	[mm]	délka delší úhlopříčky u zkoušky tvrdosti podle Knoopa
HS		tvrdost podle Shoreho
h_1	[mm]	výška, ze které je pouštěno pádové těleso u zkoušky tvrdosti podle Shoreho
h_2	[mm]	výška odrazu pádového tělesa u zkoušky tvrdosti podle Shoreho
E	[GPa]	Youngův modul pružnosti
H		tvrdost

symbol	jednotka	veličina
h	[mm]	posunutí
S	[N/mm]	kontaktní tuhost
h_c	[mm]	kontaktní hloubka indentace
h_{max}	[mm]	maximální hloubka indentace
h_s	[mm]	vzdálenost mezi povrchem vzorku a kontaktem indentoru s povrchem
h_f	[mm]	hloubka vtisku po odtížení indentoru
$A(h_c)$	[mm^2]	kontaktní plocha indentoru
E_r	[GPa]	redukovaný Youngův modul
E_i	[GPa]	Youngův modul pružnosti indentoru
ν	[1]	Poissonovo číslo
ν_i	[1]	Poissonovo číslo indentoru
d_{str}	[mm]	aritmetický průměr délek úhlopříček Vickersova vtisku
H_{str}		aritmetický průměr tvrdostí podle Vickerse

Obsah

Seznam obrázků

Seznam tabulek

Kapitola 1

Úvod

Mechanické vlastnosti materiálů se nejčastěji zjišťují experimentálními metodami, které v současnosti prochází velkým vývojem. Experimentální metody se rozvíjejí díky možnosti počítačového řízení a přesnějšímu vyčítání jednotlivých výstupních veličin (například vyčítání síly siloměrem) zejména směrem k mikro-, nebo nano-oblasti. Pokrokem výpočetní techniky je také umožněno přesnější a automatizované vyhodnocování dat z experimentálních měření, tak aby doba vyhodnocování experimentu byla minimalizována.

Tato práce se zabývá zkouškami pro měření tvrdosti povrchu materiálu. Zkoušky tvrdosti jsou jednou ze základních mechanických zkoušek materiálů, které jsou často používány při výstupní kontrole velkých výrobních procesů z důvodů nedestruktivnosti dané metody. Zároveň se jedná o metody vědecké, sloužící nejen k ověřování, ale i ke stanovení mechanických vlastností daného materiálu. Při instrumentaci zkoušky tvrdosti je navíc možné získat informaci o dalších mechanických vlastnostech materiálu (jako například: Youngovu modulu pružnosti, mezi pevnosti, lomové houževnatosti atp.). Zkoušky tvrdosti nicméně podávají informaci pouze o povrchových mechanických vlastnostech materiálu, nikoliv o mechanických vlastnostech celého vzorku. Vzhledem k tomu, že tvrdost materiálu je závislá na celé řadě faktorů, mezi které patří i tvar indentoru či velikosti zatěžovací síly, není téměř žádný vzájemný vztah mezi hodnotami tvrdosti pořízenými různými druhy zkoušek tvrdosti.

Tato práce se zabývá návrhem, výrobou a zprovozněním funkčního zařízení schopného automatizované indentace pomocí hrotu Vickersova typu při nízkém zatížením, nebo mikroindentace. Motivací pro vznik zařízení byla potřeba robotizace měření tvrdosti s možností automatizovaného vytvoření mřížky jednotlivých vtisků pro efektivní zmapování změn hod-

not tvrdosti na dané ploše. Pro vyhodnocení měření je třeba vytvořit obrazová data jednotlivých vtisků, což je umožněno optickou soustavou indentoru. Pro kontrolu správné funkčnosti indentoru bylo provedeno měření spolehlivosti a určení chyby indentoru pomocí referenční tvrdoměrné destičky.

Kapitola 2

Zkoušky tvrdosti

Tvrdost materiálu je definována jako odolnost materiálu proti vnikání cizího tělesa a jedná se o jednu ze základních materiálových vlastností. Změřená hodnota tvrdosti materiálu je ovlivňena mnoha faktory, které závisí na vlastnostech vzorku, metodě indentace a vlastnostech indentoru. Mezi nejzásadnější faktory, které indentaci ovlivňují, patří [1]:

- velikost vzorku,
- elastické vlastnosti vzorku (modul pružnosti v tahu, modul pružnosti ve smyku, ...),
- plastické vlastnosti vzorku (mez kluzu, mez pevnosti, ...),
- síla působící na indentor,
- rychlost vtlačování indentoru,
- doba zatěžování maximální silou,
- tření mezi vzorkem a indentorem,
- teplota, za které indentace probíhá,
- vzdálenost vtisku a okraje vzorku,
- vzdálenost mezi jednotlivými vtisky,
- tvar indentoru,
- tvrdost indentoru,

- geometrická přesnost indentoru.

Pro měření tvrdosti se využívá celá řada zkušebních metod, které lze rozdělit podle způsobu provedení zkoušky na:

- vnikací, blíže popsané v kapitole 2.1,
- vrypové, blíže popsané v kapitole 2.2,
- odrazové, blíže popsané v kapitole 2.3.

Dalším možným dělením zkoušek tvrdosti je rozdělení podle charakteru zatěžovací síly na:

- kvazistatické,
- dynamické.

Hroty používané při indentaci se dělí podle tvaru na:

- ostré hroty,
- kulovité hroty.

2.1 Vnikací zkoušky tvrdosti

Vnikací zkoušky tvrdosti jsou základními materiálovými zkouškami, které se v materiálovém inženýrství a strojním průmyslu používají již více než 150 let [2]. Jejich základní princip spočívá ve vtlačovaní hrotu (indentoru) do povrchu zkoumaného vzorku. Indentor má pro jednotlivé zkoušky přesně danou geometrii, která je blíže popsána u jednotlivých typů zkoušek.

Obvyklá omezení u jednotlivých zkoušek se týkají rozteče mezi jednotlivými indenty a jejich vzdáleností od volných okrajů vzorku, a to kvůli vzájemnému ovlivnění deformačních oblastí vznikajících v objemu pod jednotlivými indenty.

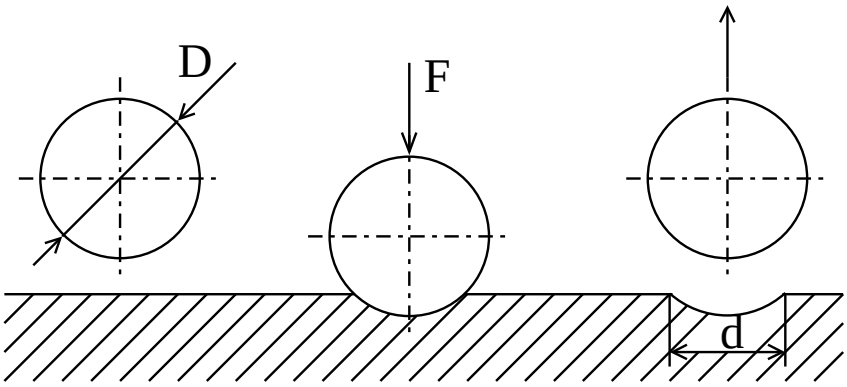
2.1.1 Zkouška tvrdosti podle Brinella

Moderní interpretace tvrdosti jako odporu materiálu proti vznikající plastické deformaci pod vtlačovaným indentorem, který je v porovnání se zkoušeným materiálem významně tvrdší, má původ v Brinellově práci z roku 1900 [3]. Johan August Brinell (švédský inženýr) vymyslel základ zkoušky, která je nyní po něm pojmenována.

Při zkoušce podle Brinella má indentor tvar kuličky, která se dříve vyráběla z vytvrzené oceli a později ze slinutého karbidu wolframu. Indentor je při této zkoušce vtlačován silou F do povrchu vzorku. Po odtížení indentoru zůstane v povrchu vzorku vtisk, který má tvar kulového vrchlíku. Tvrdost podle Brinella je potom vyjádřena jako poměr síly ku ploše vtisku:

$$HB = \frac{2F}{\pi D^2 \left[1 - \sqrt{1 - \left(\frac{d}{D}\right)^2}\right]}, \tag{2.1}$$

kde d je průměr vtisku v povrchu vzorku a D je průměr indentoru. Průběh zkoušky tvrdosti podle Brinella je schematicky zobrazen na obrázku 2.1



Obrázek 2.1: Schema průběhu zkoušky tvrdosti podle Brinella.

Jestliže je indentace provedena pomocí indentoru vyrobeného z vytvrzené oceli označuje se tvrdost podle Brinella jako HBS, jestliže je ovšem indentace provedena pomocí kulového hrotu ze slinutého karbidu wolframu, označuje se tvrdost pode Brinella jako HBW. Zkouška tvrdosti podle Brinella je popsána normou ČSN EN ISO 6506 [4].

Daná norma udává, že se hrot ve tvaru kuličky při zkoušce podle Brinella zatěžuje rovnoměrně bez rázů a chvění, doba od počátku zatěžování do plného zatížení zkušební

silou nesmí být kratší než 2 s a delší než 8 s. Doba výdrže na maximální síle má být v rozmezí 10 – 15 s [4].

Vzdálenost mezi středy jednotlivých vtisků Brinellova indentoru musí být alespoň 4 násobek průměru vtisku u vzorků vyrobených z oceli, litiny, mědi a jejích slitin a 6 násobek průměru vtisku u vzorků z lehkých kovů, olova, cínu a jejich slitin [4].

Vzdálenost od okraje vzorku musí být pro vzorky z oceli, litiny, mědi a její slitiny alespoň 2,5 násobek průměru vtisku u vzorků vyrobených z lehkých kovů, olova, cínu a jejich slitin musí být vzdálenost od okraje vzorku alespoň 3 násobek průměru vtisku [4].

Tloušťka vzorku musí být alespoň 8 násobek hloubky vtisku tak, aby na protilehlé straně k indentaci nebyly zřetelné žádné stopy po deformaci [4].

Průměr každého vtisku se změří ve dvou na sobě kolmých směrech a pro stanovení tvrdosti se počítá s průměrem těchto dvou hodnot [4].

2.1.2 Zkouška tvrdosti podle Meyera

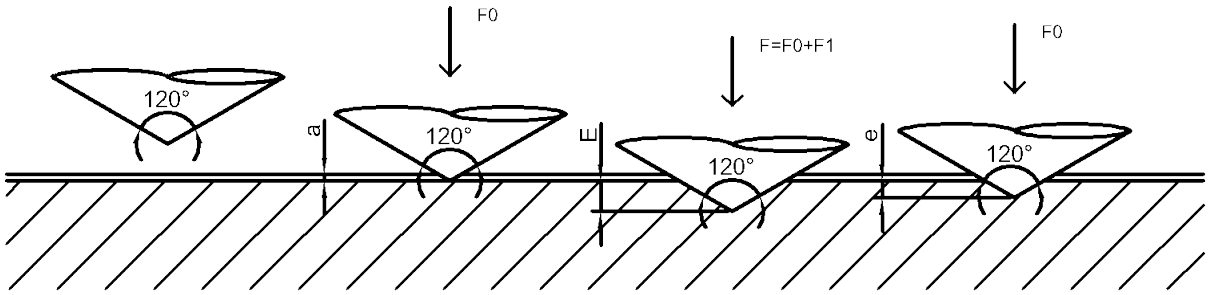
Na stejném principu jako Brinellova zkouška tvrdosti je založena i zkouška tvrdosti podle Meyera. Schéma průběhu zkoušky podle Meyera by tedy bylo stejné jako schéma zkoušky tvrdosti podle Brinella na obrázku 2.1. Meyerova zkouška tvrdosti je ale definována jako poměr síly ku ploše průmětu kulového vrchlíku vzniklého po indentaci do roviny povrchu vzorku [5]. Vypočítá se tedy jako:

$$HM = \frac{4F}{\pi d^2}. \tag{2.2}$$

2.1.3 Zkouška tvrdosti podle Rockwella

V roce 1914 byla patentována zkouška tvrdosti podle Rockwella, která byla poprvé komerčně využita na začatku 20. let 20. století [6]. Hodnota tvrdosti podle Rockwella je určována z rozdílu hloubky vtisku při působení dvěma úrovněmi zatížení (předběžného a celkového).

Při měření tvrdosti Rockwelovou metodou se používají dva typy indentorů – ocelová kulička a diamantový kužel. Kužel má vrcholový úhel 120 ° a poloměr zaoblení vrcholu 0,2 mm, kulička má průměr 1,5875 mm, nebo 3,175 mm (použití jednotlivých typů indentorů je shrnuto v tabulce 2.1).



Obrázek 2.2: Schema průběhu zkoušky tvrdosti podle Rockwella.

Průběh zkoušky začíná zatížením vzorku předběžnou silou 10 kp, nebo 3 kp[1], hloubka odpovídající tomuto zatížení je nastavena jako výchozí. Poté je k předběžnému zatížení přidáno další tzv. přídatné zatížení, které je po výdrži odstraněno tak, aby zůstalo pouze původní zatížení. V této poloze je odečten rozdíl výchozí a konečné hloubky a na stupnici tvrdoměru se rovnou zobrazí hodnota tvrdosti podle Rockwella. Průběh zkoušky podle Rockwella je znázorněn na obrázku 2.2.

Zkouška tvrdosti podle Rockwella je popsána normou ČSN EN ISO 6508 [7], která udává, že tvrdost podle Rockwella se značí jako HRx, kdy za x se dosadí písmeno A až T podle použitého hrotu a zatížení tak, jak je popsáno v tabulce 2.1. Volba mezi jednotlivými typy Rockwellovy zkoušky závisí zejména na předpokládané tvrdosti zkoušeného tělesa.

Tloušťka vzorku u zkoušky tvrdosti podle Rockwella musí být alespoň 10 násobek parametru e dle obrázku 2.2.

Doba zatěžování od předběžného zatížení k celkovému zatížení je pro metody HRA až HRK, viz tabulka 2.1, v rozmezí 2 až 8 s a pro ostatní metody 1 až 8 s. Celkové zatížení působí u materiálů, které nevykazují závislost plastické deformace na čase, 1 až 3 s, u materiálů, které vykazují omezenou závislost deformace na čase, 1 až 5 s a u materiálů, které vykazují silnou závislost plastické deformace na čase, 10 až 15 s [7].

Pro metody HRA až HRK, viz tabulka 2.1, musí být vzdálenost mezi středy dvou sousedních vtisků alespoň 4 násobek průměru vtisku a nejméně 2 mm. Vzdálenost vtisku od okraje vzorku musí být alespoň 2,5 násobek průměru vtisku [7].

Pro metody HRN a HRT platí minimální vzdálenost mezi středy dvou sousedních vtisků 3 násobek průměru vtisku a vzdálenost středu vtisku od okraje vzorku alespoň 2,5 násobek

[1]kilopond je dříve používaná jednotka síly, 1 kp = 9,81 N

Tabulka 2.1: Zkušební zatížení a tvary indentorů pro různé metody zkoušky tvrdosti podle Rockwella.

Stupnice tvrdosti	**Symbol tvrdosti**	**Vnikací těleso**	**Předběžné zatížení F_0 [N]**	**Celkové zatížení F [N]**	**Oblast použití**
A	HRA	Diamantový kužel	98,07 (10 kp)	588.4 (60 kp)	20 až 88 HRA
B	HRB	Ocelová kulička ∅ 1,5875 mm		980,7 (100 kp)	20 až 100 HRB
C	HRC	Diamantový kužel		1471 (150 kp)	20 až 70 HRC
D	HRD	Diamantový kužel		980,7 (100 kp)	40 až 70 HRD
E	HRE	Ocelová kulička ∅ 3,175 mm		980,7 (100 kp)	70 až 100 HRE
F	HRF	Ocelová kulička ∅ 1,5878 mm		588,4 (60 kp)	60 až 100 HRF
G	HRG	Ocelová kulička ∅ 1,5878 mm		1471 (150 kp)	30 až 94 HRG
H	HRH	Ocelová kulička ∅ 3,175 mm		588,4 (60 kp)	80 až 100 HRH
K	HRK	Ocelová kulička ∅ 3,175 mm		1471 (150 kp)	40 až 100 HRK
15 N	HR 15 N	Diamantový kužel	29,42 (3 kp)	147,1	70 až 94 HR 15 N
30 N	HR 30 N	Diamantový kužel		294,2	42 až 86 HR 30 N
45 N	HR 45 N	Diamantový kužel		441,3	20 až 77 HR 45 N
15 T	HR 15 T	Ocelová kulička ∅ 1,5878 mm		147,1	67 až 91 HR 15 T
30 T	HR 30 T	Ocelová kulička ∅ 1,5878 mm		294,2	29 až 82 HR 30 T
45 T	HR 45 T	Ocelová kulička ∅ 1,5878 mm		441,3	1 až 72 HR 45 T

průměru vtisku [7].

Vzhledem k jednoduchosti průběhu zkoušky tvrdosti podle Rockwella je tato zkouška vhodná pro použití kontroly kvality při výrobě velkých serií výrobků.

2.1.4 Zkouška tvrdosti podle Vickerse

V roce 1922 byla popsána zkouška tvrdosti podle Vickerse[2] [8]. Indentor při zkoušce tvrdosti podle Vickerse má tvar pravidelného čtyřbokého jehlanu vyrobeného z diamantu s vrcholovým úhlem mezi stěnami 136 °. Výpočet tvrdosti podle Vickerse je stejný jako u Brinellovy zkoušky, tedy podíl síly a plochy vtisku. Tvrdost podle Vickerse se tedy vypočítá jako:

$$HV = \frac{2F \sin \frac{136\,^\circ}{2}}{d_u^2} \approx \frac{1,8544F}{d_u^2}, \tag{2.3}$$

kde F je zatěžovací síla v kilopondech a d_u je aritmetický průměr délek úhlopříček vtisku u_1 a u_2 v milimetrech, který vznikl po odtížení indentoru, viz obrázek 2.3. Průmět vtisku indentoru má tvar čtyřúhelníku, jehož teoretickým tvarem je čtverec.

Pokud se pro výpočet využívají jednotky SI, je třeba převést sílu z kilopondů na Newtony a pro výpočet tvrdosti podle Vickerse použít vzorec:

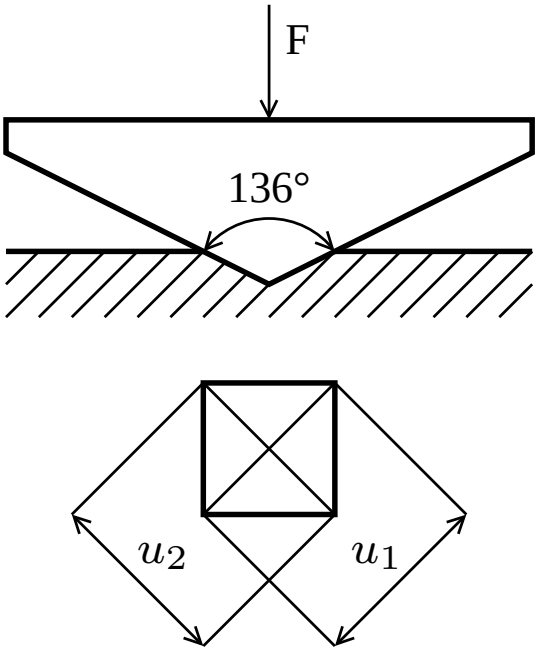
$$HV = \frac{0,1891F}{d_u^2}, \tag{2.4}$$

kde je síla zadána v Newtonech a d_u v milimetrech. Průběh zkoušky tvrdosti podle Vickerse je zobrazen na obrázku 2.3.

Podle velikosti zkušebního zatížení se zkouška tvrdosti podle Vickerse dělí na tři části [9]:

- zkouška tvrdosti podle Vickerse,
  - zkušební zatížení větší než 5 kp
- zkouška tvrdosti podle Vickerse při nízkém zatížení,
  - zkušební zatížení mezi 0,2 a 5 kp
- zkouška mikrotvrdosti podle Vickerse.

[2] Vickers – firma ve Velké Británii



Obrázek 2.3: Schema průběhu zkoušky tvrdosti podle Vickerse.

– zkušební zatížení mezi 0,01 a 0,2 kp

Zkouška tvrdosti podle Vickerse je popsána normou ČSN EN ISO 6507. Tvrdost podle Vickerse se označuje vypočtenou hodnotou tvrdosti, za kterou následuje označení HV pro tvrdost podle Vickerse a poté velikost zkušebního zatížení v kilopondech. Pokud se doba zatížení liší od předepsaných 10 až 15 s, označuje se tato doba za lomítko na konec. Označení 500 HV 30/20 znamená tvrdost podle Vickerse o hodnotě 500 při zatížení 294 N (30 kp) působícím po dobu 20 s [10].

Indentor musí být vtlačen do povrhu vzorku kolmo bez rázů a chvění a doba od začátku zatěžování po zatížení zkušební silou musí být mezi 2 a 8 s. Při zkouškách tvrdosti při nízkém zatížení nesmí překročit 10 s. Doba působení zkušební síly je mezi 10 a 15 s [10].

Pro vzorky vyrobené z oceli, litiny, mědi a jejích slitin platí, že vzdálenost vtisku od okraje vzorku musí být alespoň 2,5 násobek délky úhlopříčky a pro vzorky vyrobené z lehkých kovů, olova, cínu a jejich slitin musí být vzdálenost vtisku od okraje vzorku alespoň 3 násobek velikosti úhlopříčky [10].

Vzdálenost jednotlivých vtisků při zkoušení oceli, litiny, mědi a jejích slitin musí být minimálně 3 násobek délky úhlopříček a při zkoušení lehkých kovů, olova,cínu a jejich slitin musí být vzdálenost jednotlivých vtisků alespoň 6 násobek velikosti úhlopříčky [10].

Výhodou zkoušky tvrdosti podle Vickerse je, že vypočtené tvrdosti jsou prakticky nezávislé na velikosti zatížení, a že poměr hodnot tvrdostí odpovídá skutečným poměrům, tzn. že materiál o tvrdosti 200 HV má oproti materiálu o tvrdosti 100 HV dvojnásobnou tvrdost, proto se jedná o velmi univerzální zkušební metodu [11].

Nevýhodou zkoušky tvrdosti podle Vickerse je nutnost přesného změření délky úhlopříček a potřeba poměrně přesné úpravy povrchu vzorku před měřením.

2.1.5 Zkouška tvrdosti podle Knoopa

Zkouška tvrdosti podle Knoopa spočívá ve vtlačování indentoru tvaru čtyřbokého jehlanu s vrcholovými úhly $172,5°$ a $130°$ do povrchu zkoušeného vzorku danou silou. Vzniklý vtisk má tvar protáhlého kosočtverce, u kterého se změří pouze délka delší úhlopříčky (na rozdíl od zkoušky tvrdosti podle Vickerse 2.1.4, kdy se změřily délky obou úhlopříček). Tvrdost podle Knoopa se potom vypočítá jako:

$$HK = \frac{1,4509F}{l^2}, \tag{2.5}$$

kde F je síla v Newtonech a l je délka delší úhlopříčky v milimetrech.

Metoda se používá pro měření tvrdosti zejména tenkých povrchových vrstev z toho důvodu, že poměr délky úhlopříčky k hloubce vtisku je 30 (u zkoušky tvrdosti podle Vickerse je tento poměr 7). Protáhlý tvar vtisku vzniklého Knoopvým indentorem umožňuje umístit více indentů do dané vzdálenosti, než například Vickersův indentor. Díku tomu je vhodný pro měření gradientů tvrdosti.

Zkouška tvrdosti podle Knoopa je popsána normou ČSN ISO 4545 [12].

2.1.6 Shrnutí

Ze způsobu měření tvrdosti podle Brinella, Meyera, Vickerse a Knoopa je zřejmé, že pro určení tvrdosti je potřeba jisté míry dovednosti operátora pro správné změření rozměrů vtisku indentoru, k čemuž je potřeba alespoň optický mikroskop (při měření tvrdosti podle Vickerse za použití malých zatížení je pak vhodnější použití elektronové mikroskopie).

Brinellova, Vickersova i Rockwellova zkouška tvrdosti si velmi rychle vydobyly své místo pro kontrolu materiálových vlastností při výstupní kontrole materiálů využívaných pro automobilový, letecký nebo zbrojní průmysl, které byly v první polovině 20. století na velkém

vzestupu. Zkoušky tvrdosti byly, oproti do té doby standardně využívaným trhacím zkouškám, rychlejší, protože nebylo třeba vytvářet vzorky o speciálním tvaru a rozměrech. Nezbytný je pouze vzorek materiálu s relativně rovným povrchem. Další výhodou je, že zkoušky tvrdosti jsou ve své podstatě nedestruktivní, i když přesnější označení by bylo částečně destruktivní, protože povrch vzorku je trvale zdeformován. Pokud se použije zkouška tvrdosti podle Vickerse při malém zatížení, je možné pokrýt povrch vzorku sérií vtisků a tak zmapovat změny vlastností, které se mohou ve vzorku vyskytnout. Nevýhodou oproti trhací zkoušce ale je, že ze standardní indentace, tak jak byla popsána výše, není možno určit Youngův modul pružnosti materiálu a popsat konstituvní rovnice napětí – deformace. Modul pružnosti lze z indentační zkoušky zjistit až její instrumentací tak, jak je to blíže popsáno v kapitole 3, pro některé třídy materiálů existují také (velmi) přibližné přepočetní tabulky.

2.2 Vrypové zkoušky tvrdosti

Jedním z nejstarších způsobů měření tvrdosti byla vrypová zkouška, kterou provedl v roce 1722 Réamur. Réamur vytvořil zkušební tyč s proměnou tvrdostí po délce tyče. Hodnotu tvrdosti zkoušeného materiálu lze určit z polohy vrypu, kterou zkoušený materiál na tyči zanechal [1].

V roce 1822 sestavil Friedrich Mohse první stupnici tvrdosti, v níž seřadil 10 minerálů tak, že každý následující minerál je schopen vytvořit vryp do všech předchozích. Seřazení těchto deseti nerostů je znázorněno v tabulce 2.2.

2.2.1 Zkouška tvrdosti podle Martense

Zkouška tvrdosti podle Martense probíhá tak, že se diamantový hrot ve tvaru kuželu s vrcholovým úhlem 90° přitlačuje rostoucím zatížením do povrchu zkoušeného vzorku, který se pod hrotem pohybuje danou rychlostí. Tvrdost podle Martense odpovídá poté síle F nutné ke vzniku vrypu o šířce 0,01 mm.

2.2.2 Další vrypové zkoušky tvrdosti

Kromě již zmíněných vrypových zkoušek existuje ještě mnoho dalších typů vrypových zkoušek (scratch testů). Každá tato zkouška vznikla pro potřeby jednotlivých měření

Tabulka 2.2: Mohsova stupnice tvrdosti.

Tvrdost	**Minerál**	**Chemický vzorec**
1	mastek	$Mg_3Si_4O_{10}(OH)_2$
2	sůl kamenná	NaCl
3	kalcit – vápenec	$CaCO_3$
4	fluorit (kazivec)	CaF_2
5	apatit	$Ca_5(PO_4)_3(OH^-, Cl^-, F^-)$
6	ortoklas (živec)	$KAlSi_3O_8$
7	křemen	SiO_2
8	topaz	$Al_2SiO_4(OH^-, F^-)_2$
9	korund	Al_2O_3
10	diamant	C

různých materiálů a jsou mezi sebou jen velmi těžko porovnatelné. Scratch testy je možné rozdělit do čtyř skupin jako:

- jednohrotové scratch testy
  - nejpoužívanější tvary hrotů jsou koule nebo kužel
  - například již zmíněná Mohsova stupnice, viz tabulka 2.2
  - zkouška tvrdosti tužkami – vryp se vytváří tužkami různých tvrdostí. Porovnává se, která tužka se zlomí a nezanechá žádný vryp [13].
  - Briscoeův přístroj na vrypové zkoušky. Indentor je umístěn na rameni, pod kterým se pohybuje vzorek rychlostí od 0,001 do 40 mm/s. Na indentor je umístěn siloměr, který zaznamenává tečnou sílu působící na hrot. Normálová síla je vyvolaná závažím umístěným na držáku indentoru. Hrot má tvar koule, nebo kuželu [14].
  - mezi další zařízení pro vrypové zkoušky patří: přístroj od Gauthiera a Schirrera [15] a mnoho dalších.
- více-hrotové scratch testy
  - například Fords five-finger zkouška [16]

- vrypová zkouška hrotem umístěným na kotouči (pin-on-disc test)
  - Taberův vrypový test [17]
- nano – vrypová zkouška
  - nano – vrypovou zkoušku je možno provádět mnoha komerčně dostupnými přístroji jako například TI 950 TriboIndenter® (Hysitron, USA), nebo Nano Scratch Tester (Anton Paar TriTec, Švýcarsko).

2.3 Dynamické zkoušky tvrdosti – odrazové zkoušky tvrdosti a tvrdoměr Poldi

Odrazové zkoušky tvrdosti patří mezi dynamické zkoušky, kdy těleso o dané hmotnosti dopadá z konstantní výšky na povrch zkoumaného vzorku. Při nárazu se část kinetické energie spotřebuje na plastickou deformaci povrchu vzorku a zbývající energie se projeví odrazem tělesa. Hodnota tvrdosti se poté určí z výšky, které dosáhne odražené těleso. Nevýhodou dynamických zkoušek tvrdosti je jejich nízká přesnost oproti kvazistatickým metodám. Výhodou naopak je možnost jejich operativního využití v jakýchkoliv podmínkách a například již na hotových výrobcích jako kvalitativní zkouška na konci výrobní linky.

2.3.1 Zkouška tvrdosti podle Shoreho

U původního Shoreho skleroskopu se pohybuje pádové těleso ve tvaru válce zakončené diamantovým hrotem s poloměrem zaoblení 1 mm uvnitř skleněné trubky se stupnicí. Výška dosažená po odrazu pádového tělesa se odečte ze stupnice pomocí lupy. Počáteční výška pádového tělesa se nastavovala vysátím vzduchu z trubice nad tělesem, nebo pomocí pružiny.

U zkoušky tvrdosti podle Shoreho se používaly dvě stupnice označené jako HSC, nebo HSD a tyto stupice se lišily hmotností pouštěného tělesa a počáteční výškou, ze které je toto těleso pouštěno [9]. U stupnice HSC je hmotnost pádového tělesa 2,5 g a pádová výška 254 mm a tvrdost podle Shoreho se vypočítá jako:

$$HSC = \frac{10^4 h_2}{65 h_1}, \tag{2.6}$$

kde h_1 je původní výška, ze které je těleso pouštěno a h_2 je výška odrazu pádového tělesa od povrchu vzorku.

Druhé stupnici označené jako HSD odpovídá hmotnost pádového tělesa 36,2 g a počáteční výška 19 mm. Tvrdost podle Shoreho se v tomto případě vypočítá jako:

$$HSD = \frac{140 h_2}{h_1}, \tag{2.7}$$

kde h_1 je počáteční výška a h_2 je výška odrazu tělesa od povrchu zkoušeného vzorku.

2.3.2 Tvrdoměr Equotip

Tvrdoměr Equotip (firma Proceq, Švýcarsko) pracuje na principu zkoušky tvrdosti odrazem podle Leeba. Tvrdost pomocí zařízení Eqoutip se určí z rychlosti odrazu sondy od povrchu zkoušeného materiálu. Měření probíhá na základě vypuštění sondy kolmo proti povrchu měřeného vzorku. Tvrdost se stanoví z porovnání rychlosti sondy před a po dopadu na povrch vzorku. Tvrdoměr využívá předpokladu, že rychlost sondy po odrazu od tvrdšího materiálu bude vyšší než rychlost po odrazu od materiálu měkčího [18].

2.3.3 Tvrdoměr Poldi

Tvrdoměr Poldi je přenosný tvrdoměr, který porovnává velikost vtisku vzniklého rázem ve zkoušeném vzorku a velikost vtisku na materiálu o známé tvrdosti. Z tohoto poměru lze z tabulek vyčíst hodnotu tvrdosti podle Brinella a v některých případech také pevnost v tahu.

Zkouška probíhá tak, že se mezi porovnávací zkušební tyč o známé tvrdosti a zkoušený vzorek umístí kalená ocelová kulička o průměru 10 mm. Úderem kladiva do tvrdoměru se vytvoří vtisk jak do zkoušeného vzorku, tak do porovnávací zkušební tyče. Velikost vtisků je možné změřit lupou nebo přesněji pod mikroskopem. Tvrdost podle Brinella a pevnost v tahu je možné vyčíst z tabulek. Vzhledem k tomu, že se tvrdost určuje pouze z poměru vtisků, může se síla působící na kladivo při různých zkouškách lišit [19].

Tvrdoměr Poldi se využívá především pro rychlé, orientační měření tvrdosti velkých konstrukcí a výrobků, kde nelze použít normalizovanou zkoušku tvrdosti podle Brinella,

tak jak je popsána v části 2.1.1. Výhodou měření tvrdoměrem Poldi je jeho malá hmotnost a rozměry, což činí zkoušku velmi operativní. Nevýhodou naopak je jeho malá přesnost.

Kapitola 3

Instrumentovaná zkouška tvrdosti

Instrumentovaná indentační zkouška je indentace zařízením osazeným siloměrem a přesným vyčítáním polohy indentoru, tak aby bylo umožněno po celou dobu indentace zaznamenávat hloubku vtisku a sílu, která působí na povrch zkoušeného vzorku. Díky záznamu hodnot síly a posunutí je umožněno měřit další mechanické vlastnosti vzorku.

Zkouška tvrdosti povrchu materiálu se provádí vtlačováním tělesa do povrchu zkoušeného materiálu. Vtlačované těleso musí mít daný tvar a významně vyšší tvrdost (často diamantový, tvrdokovový, nebo vytvrzený ocelový hrot). Takto definované těleso je zatíženo silou, která se po určitou dobu udržuje konstantní. Po odtížení je možné na povrchu zkoušeného vzorku pozorovat vtisk vtlačovaného tělesa (hrotu) a změřit jej. Tvrdost se poté vypočte jako:

$$H = \frac{F}{A},$$

kde F je maximální zatížení a A je plocha vtisku. Jednotlivé typy zkoušek tvrdosti blíže popsané v kapitole 2 se liší v použití jiného tvaru vtlačovaného tělesa, nebo použitím jiného materiálu hrotu.

Ve výše popsaném průběhu měření se nijak neliší tradiční zkoušky tvrdosti od instrumentovaných zkoušek. Rozdílem mezi těmito přístupy ale je, že u tradičního měření tvrdosti je změřena velikost trvalé (plastické) deformace pouze jednou při jedné zatěžovací síle, kdežto u instrumentovaných zkoušek tvrdosti jsou síla a posun hrotu měřeny a/nebo řízeny po celou dobu kontaktu mezi hrotem a povrchem zkoumaného materiálu kontinuálně a paralelně. Používání instrumentované zkoušky tvrdosti neustále nabývá na významu při zjišťování mechanických vlastností celé řady materiálů: od kovů a keramiky k polymerům

a biologickým materálům.

Historie instrumentované indentace sahá do osmdesátých let dvacátého století. Od této doby došlo k velkému vývoji počítačové techniky a z toho důvodu je nyní jednodušší jak ovládání a řízení instrumentovaného indentoru, tak zpracování naměřených dat, které může být částečně, nebo zcela zautomatizované.

Instrumentovaná zařízení pro měření tvrdosti jsou často schopna pracovat s kombinací velmi malých rozlišení síly ($\approx 1\mu$N) i posunutí (≈ 0.2 nm) při současném velkém rozsahu použitelných sil a posunutí (od zatížení desítek μN po stovky mN a posunutí od desítek nm po desítky μm). Díky tomu může být jedno indentační zařízení použito pro měření téměř jakéhokoliv materiálu. Použití automatizovaného software pro řízení a vyhodnocení zkoušky potom umožňuje zrychlení a zejména zpřesnění měření. Dalšího zlepšení citlivosti přístroje, jakož i možnosti testování dalších materiálových charakteristik, je možné dosáhnout vybavením indentačního zařízení dynamickou oscilací hrotu [20].

Tradičně je instrumentovaná zkouška tvrdosti používána pro zjištění tvrdosti povrchu materiálu (H) a pro zjištění Youngova modulu pružnosti (E). Vedle Youngova modulu pružnosti a tvrdosti byla již instrumentovaná zkouška tvrdosti použita například také ke zjištění komplexního modulu u biomateriálů [21], [22], creepových vlastností u polymerů [23], meze kluzu a dislokačního chování u kovových materiálů [24], lomové houževnatosti u skleněných a keramických materiálů [25], mechanických vlastností tenkých vrstev [26] a zbytkového napětí [27]. Další možností použití instrumentované zkoušky tvrdosti jsou vrypové zkoušky, jako například hodnocení odolnosti žárových nástřiků proti vrypu a odolnosti kovu proti otěru.

Současně s nacházením nových možností využití instrumentované zkoušky tvrdosti je také třeba ověřovat výsledky této metody s tradičními způsoby zjišťování těchto vlastností. Příkladem může být ověření hodnoty Youngova modulu pružnosti spočítané z dat z instrumentované indentace a hodnoty Youngova modulu získaného z vyhodnocení tahové zkoušky provedené na vzorku ze stejného materiálu.

3.1 Průběh instrumentované indentační zkoušky

Zařízení pro instrumentovanou indentaci mohou být řízena silou nebo posuvem. Typický průběh indentační zkoušky je složen z pěti částí:

- po detekci kontaktu mezi indentorem a povrchem zkoušeného materiálu, se indentor

začne zatěžovat, dokud není dosažena maximální síla, nebo maximální posuv (podle toho, zda se jedná o řízení silou nebo posuvem).

- po danou dobu se udržuje síla na maximální hodnotě (obvykle se jedná o dobu 10 až 15 s)

- indentor se zdvihá od povrchu stejnou rychlostí jakou byl v první části zatěžován, dokud síla nedosáhne malé části maximální zatěžovací síly (obvykle 10% této hodnoty)

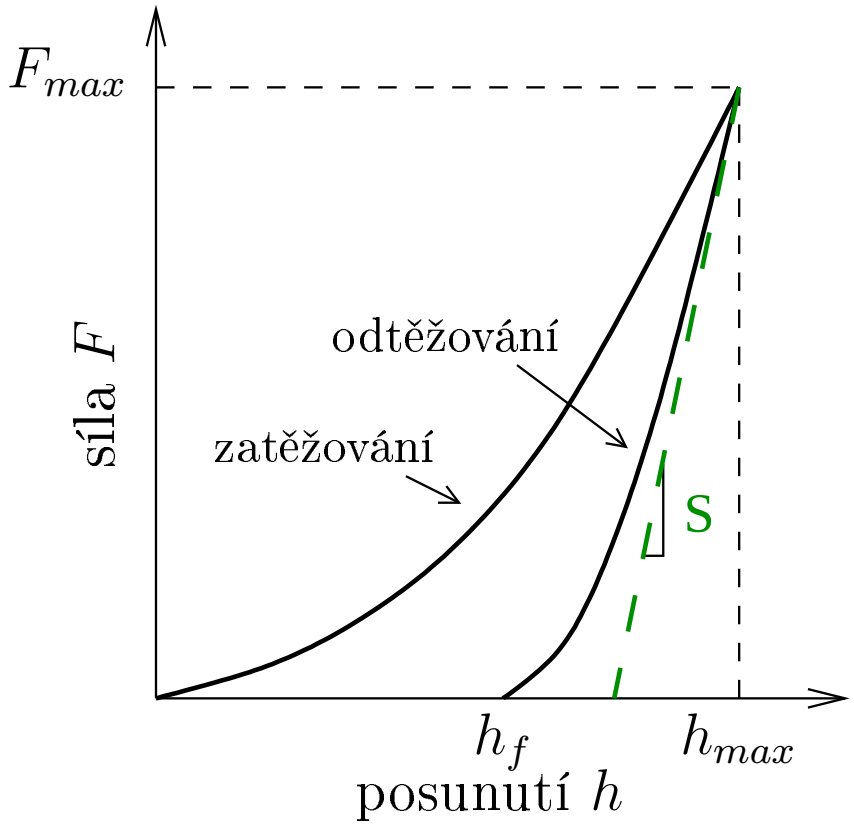
- tato síla se udržuje konstatní po dobu, kterou si zvolí uživatel. Důvodem začlenění této části je získání dostatečného množství dat k porovnání jaké množství naměřeného posuvu se vztahuje k termálnímu roztažení nebo smrštění indentačního zařízení a/nebo testovaného materiálu. Tento jev se nazývá teplotní posuv (thermal drift). Tato část může být vynechána, pokud se dá předpokládat, že teplotní posuv bude malý v porovnání s celkovým posuvem indentoru při indetaci.

- plné odtížení indentoru.

3.2 Postup měření

Při vtlačování indentoru do povrchu vzorku je u instrumentované indentace zaznamenáváno aktuální posunutí indentoru a síla, kterou indentor na povrch vzorku působí během celého průběhu zkoušky, tedy od okamžiku, kdy došlo ke kontaktu mezi hrotem a vzorkem až po odtížení indentoru (opětovnou ztrátu kontaktu). Typický průběh křivky síla – posunutí je zobrazen na obrázku 3.1.

První část křivky síla – posunutí je zatěžovací křivka, kdy se se zvyšujícím posunutím zvyšuje také síla, kterou indentor působí na povrch vzorku, dokud není dosažena maximální zadaná hodnota zatížení nebo posuvu. Při odtěžování indentoru se část deformace na povrchu vzorku opět vrátí do původního stavu (elastická deformace), obvykle ale zůstává část vtisku indentoru zdeformována (plastická deformace). Tvar odtěžovací křivky závisí na materiálových vlastnostech povrchu vzorku. Pokud by byla deformace vzniklá indentací čistě plastická, odtěžovací křivka by vedla vertikálně dolu. Pokud by byla deformace vzniklá indentací plně elastická, měla by odtěžovací křivka stejný průběh jako křivka zatěžovací.



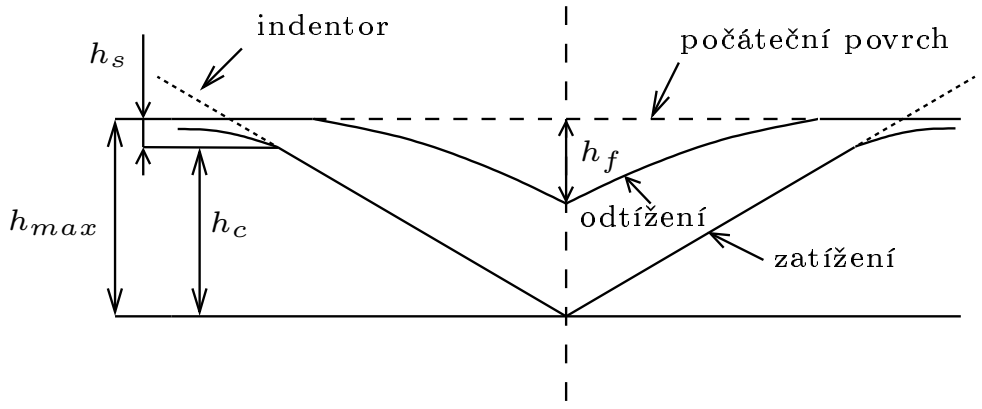
Obrázek 3.1: Typický tvar křivek zatěžování a odtěžovaní při instrumentované indentaci.

Pokud se jedná o zatěžování velmi malými silami, je nutná přesná kalibrace přístroje, kdy je třeba přesně určit bod kontaktu mezi hrotem indentoru a povrchem vzorku. Měření posuvu je nutno opravovat o tuhost přístroje a teplotní posuv, pokud nejsou zanedbatelné. V neposlední řadě je třeba měření síly zpřesnit o zatížení, které vyvolává hrot a jeho držák.

3.3 Vyhodnocení měření

Instrumentovaná indentační zkouška materiálu se vyhodnocuje z grafu závislosti síla – posunutí. Vyhodnocením lze získat materiálové vlastnosti zkoumaného vzorku, jako je hodnota tvrdosti a Youngova modulu pružnosti. Pro výpočet hodnoty Youngova modulu a tvrdosti je potřeba začít výpočet analýzou odtěžovací křivky ukázané na obrázku 3.1. Data síly a posunutí pořízenými při odtěžování proložíme funkcí:

$$F = B\,(h - h_f)^m\,, \tag{3.1}$$



Obrázek 3.2: Tvar povrchu vzorku při zatížení a odtížení indentoru s vyznačením kontaktní hloubky h_c.

kde F a h jsou uspořádané dvojice dat síla – posunutí a B, h_f a m jsou optimální konstanty získané proložením měřeného průběhu uvedeným matematickým modelem. Obvykle se pro prokládání dat nepoužívají data z celé odtěžovací křivky, ale pouze z její horní části (data po začátku odtěžování).

Kontaktní tuhost S je vztah mezi silou a posunutím na začátku odtěžovací fáze, (čistě elastická část). Z toho důvodu se při prokládání dat pořízených při odtěžování výrazem 3.1 používá nejčastěji jejich první polovina.

Pro kontaktní tuhost S platí při maximálním posunutí (maximální hloubce) h_{max}:

$$S = \left.\frac{\mathrm{d}F}{\mathrm{d}h}\right|_{h=h_{max}} = Bm(h_{max} - h_f)^{m-1}. \tag{3.2}$$

Jestliže známe hodnotu kontaktní tuhosti, můžeme spočítat hodnotu kontaktní hloubky indentace h_c, pro kterou platí:

$$h_c = h_{max} - h_s, \tag{3.3}$$

kde h_{max} je hodnota maximální hloubky indentace, která může být experimentálně změřena. Jádrem problému je určení hodnoty h_s, která určuje vzdálenost mezi povrchem vzorku a kontaktem. Na obrázku 3.2 je znázorněna h_c, h_{max} i h_s.

Hodnota h_s závisí na tvaru indentoru. Pro tvar povrchu mimo plochu kontaktu kuželového hrotu platí podle Sneddona vztah [28]:

$$h_s = \frac{(\pi - 2)}{\pi}(h_{max} - h_f), \tag{3.4}$$

kde h_f je hloubka vtisku po plném odtížení indentoru, a ne pouze h z toho důvodu, že Sneddonovo řešení počítá pouze s elastickými deformacemi.

Podle Sneddona [28] se dá vztah síla – posunutí pro hrot tvaru kužele popsat jako

$$(h_{max} - h_f) = 2\frac{F}{S}, \tag{3.5}$$

kde S je kontaktní tuhost. Po dosazení vzorce 3.5 do vzorce 3.4 a za F se dosadí maximální dosaženou sílu F_{max} se získá vztah:

$$h_s = \varepsilon\frac{F_{max}}{S}, \tag{3.6}$$

kde ε je geometrická konstanta, která se pro hrot tvaru kužele spočítá, jako:

$$\varepsilon = \frac{2(\pi - 2)}{\pi}, \tag{3.7}$$

nebo-li $\varepsilon = 0,72$. Pokud tento postup zopakujeme pro další tvary indentorů dostaneme geometrické konstanty, které jsou shrnuty v tabulce 3.1.

Tabulka 3.1: Teoretické hodnoty konstanty ε pro různé tvary indentorů.

Tvar indentoru	ε
kužel	0,72
plochý hrot	1
paraboloid	0,75

Při experimentálním ověření hodnot ε z tabulky 3.1 vyplývá, že přesnější hodnota ε u hrotu tvaru kužele je 0,75. Hodnota 0,75 se také využívá pro hroty tvaru jehlanu, kuželu a koule. V takovém případě se tedy kontaktní hloubka vypočítá jako:

$$h_c = h_{max} - 0,75\frac{F_{max}}{S}. \tag{3.8}$$

Určení hodnoty kontaktní hloubky způsobem popsaným výše bylo poprvé navrženo Warrenem Oliverem a Georgem Pharrem [29], proto když se v literatuře autoři odkazují na model Olivera a Pharra, odkazují na vztah 3.8.

Dalším krokem je určení kontaktní plochy indentoru, která se spočítá jako funkce kontaktní hloubky:

$$A_c = f(h_c).$$

Přesný tvar této funkce je závislý na tvaru indentoru a na hloubce indentace. Nejpoužívanějším indentorem při instrumentované indentaci je Berkovichův hrot. Berkovichův hrot je diamantový a má tvar tříbokého jehlanu. Pro ideální Berkovichův indentor, se kontaktní plocha vypočítá jako:

$$A_c = 24,56h_c^2. \tag{3.9}$$

Vzorec 3.9 se používá, pokud je celková indentační hloubka větší než 2 μm, protože od této hloubky je zanedbatelný rozdíl mezi reálným tvarem Berkovichova hrotu a ideálním tvarem Berkovichova hrotu. Pokud je ale celková indentační hloubka menší než 2 μm musíme počítat se zaoblením špičky hrotu a přidat do vzorce další člen. Kontaktní plocha se tak spočítá jako:

$$A = 24,56h_c^2 + Ch_c, \tag{3.10}$$

kde C je konstanta, která se zjišťuje empiricky indentací materiálu o známých vlastnostech, často se pro tyto účely používá křemenné sklo.

Pro různé tvary indentorů platí rozdílné funkce pro výpočet kontaktní plochy. Pro nejpoužívanější tvary hrotů jsou ty funkce pro výpočet kontaktní plochy shrnuty v tabulce 3.2.

Když známe hodnotu kontaktní plochy, můžeme již určit hodnotu indentační tvrdosti jako:

$$H = \frac{F_{max}}{A_c}. \tag{3.11}$$

Analýzou odtěžovací křivky lze kromě hodnoty tvrdosti zjistit také hodnota Youngova modulu zkoumaného materiálu jako:

$$E = \left(1 - \nu^2\right) \left[\frac{1}{E_r} - \frac{1 - \nu_i^2}{E_i}\right]^{-1}, \tag{3.12}$$

kde ν je Poissonovo číslo zkoušeného materiálu, ν_i je Poissonovo číslo indentoru, E_r je redukovaný modul a E_i je Youngův modul indentoru. Redukovaný modul se vypočítá jako:

Tabulka 3.2: Tvary indentorů a jim příslušná funkce pro výpočet kontaktní plochy.

Tvar indentoru	**Výpočet kontaktní plochy**	**Poznámky**
ideální Berkovichův hrot	$A_c = 24,56h_c^2$	používané, když je $h_c > 2\mu m$
reálný Berkovichův hrot	$A_c = 24,56h_c^2 + Ch_c$	hodnota C se určuje indentací známého materiálu
reálný krychlový hrot	$A_c = 2,60h_c^2 + Ch_c$	hodnota C se určuje indentací známého materiálu
kulový hrot	$A_c = 2\pi Rh_c$	R je poloměr hrotu, tato hodnota je buď známá, nebo se určí indentací známého materiálu
kuželový hrot	$A_c = \pi \operatorname{tg}^2 \psi h_c^2$	ψ je polovina vrcholového úhlu kuželu
kuželový hrot se zaoblenou špičkou	$A_c = \pi \operatorname{tg}^2 \psi h_c^2 + 2\pi Rh_c$	superpozice kontaktní plochy pro kulatý hrot a pro kuželovitý hrot
válcový hrot otočený podstavou ke vzorku	$A_c = \pi a^2$	a je poloměr podstavy a je konstatní při jakékoliv indentační hloubce

$$E_r = \frac{\sqrt{\pi}}{2}\frac{S}{\sqrt{A_c}}. \tag{3.13}$$

Pro výpočet Youngova modulu je nezbytné znát Poissonovo číslo zkoumaného vzorku. Citlivost spočteného Youngova modulu na hodnotu Poissonova čísla ale není vysoká. Analýza změn hodnoty Youngova modulu na nepřesnost způsobenou neznalostí Poissonova čísla ukazuje, že pokud je chyba Poissonova čísla 40 %, nepřesnost Youngova modulu je pouze do 5 % [30]. Pokud tedy neznáme přesnou hodnotu Poissonova čísla, můžeme použít hodnoty:

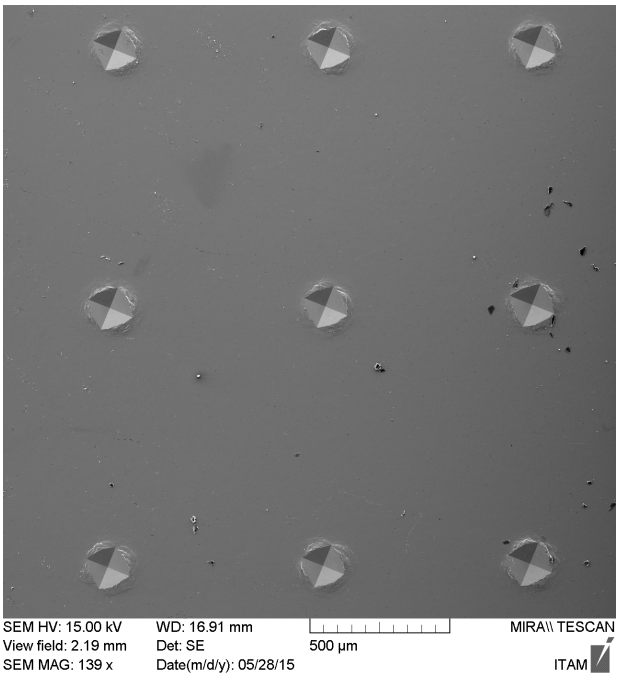
- 0,2 pro skla a keramiku,
- 0,3 pro kovy a
- 0,45 pro polymery.

Kapitola 4

Instrumentovaný indentor

Mikroindentační zařízení vzniklo na Ústavu mechaniky a meteriálů Fakulty dopravní ČVUT v Praze v rámci řešení grantu SGS15/225/OHK2/3T/16. Motivací pro vznik instrumentovaného indentoru byla potřeba robotizace měření tvrdosti na ploše s vytvořením libovolně velké mřížky vtisků pro tvorbu map změn tvrdosti. Robotizované měření není zatížené lidskou chybou v polohování, a tak vznikne rovnoměrná mřížka vtisků, ze kterých jsou také automaticky vytvořena obrazová data jednotlivých vtisků, která se dají jednoduše vyhodnocovat a poté vykreslit mapu změn tvrdosti v jednotlivých částech vzorku. Na obrázku 4.1 je znázorněna mřížka devíti vtisků vytvořená instrumentovaným indentorem vyvinutým v rámci této práci. Pro zvýšení flexibility zařízení je vhodné osadit indentor siloměrem tak, aby bylo umožněno měřit hodnotu síly, kterou indentor působí na povrch vzorku, po celou dobu měření. Druhým důvodem pro osazení indentoru siloměrem je možnost indentace libovolnou zatěžovací silou od mikroindentace, přes zatížení nízkou silou až po standardní indentaci. Rozmezí použitelných zatížení indentoru osazeným 50 N siloměrem U9C (HBM, Německo) je od 10 do 50 N. Pro maximální flexibilitu zařízení je také umožněno použít pro indentaci různé druhy indentorů. Pro první verzi indentoru byl zvolen Vickersův hrot, protože Vickersova zkouška je velmi univerzální a nabízí pokročilou možnost vyhodnocování pomocí skriptu v programovacím prostředí Matlab vzniklého v rámci mé bakalářské práce [31].

Podstata práce je návrh, výroba a zprovoznění (zapojení, kalibrace a software) funkčního zařízení pro indentaci s výsledky srovnatelnými s komerčně dostupnými Vickersovými indentory. Konstrukce zařízení byla provedena v softwaru Solidworks.



Obrázek 4.1: Mřížka devíti vtisků vytvořených instrumentovaným indentorem.

4.1 Podstata zařízení

Instrumentovaný indentor umožňuje zkoušení tvrdosti povrchu zkušebních vzorků. Při indentaci je použit Vickersův hrot, zařízení je ale variabilní a umožňuje výměnu Vickersova hrotu za jiný typ indentoru. Indentační zařízení je schopno provádět zkoušky tvrdosti podle Vickerse při nízkém zatížení, nebo zkoušky mikrotvrdosti, viz 2.1.4. Vzhledem k tomu, že je zařízení osazeno kamerou a přesným polohováním vzorku, je možné provádět indentaci na předem zvoleném místě na povrchu vzorku, nebo provádění série indentačních měření, kdy je možné pokrýt povrch vzorku sítí vtisků pro komplexní zmapování mechanických vlastností celé vybrané oblasti.

Zařízení umožňuje provádět dva typy zatěžování:

- indentace konstantní silou vyvozenou závažím,
- indentace řízená silou.

Při indentaci konstantní silou je na zařízení umístěno závaží. V tomto režimu může být zařízení osazeno siloměrem, není ale možné řídit indentor silou.

Při indentaci řízené silou je zařízeno osazeno siloměrem a je tedy možný záznam průběhu síly a posuvu během testu.

4.2 Technický popis zařízení

4.2.1 Mechanická část

Zařízení pro indentaci je osazeno dvěma plně motorizovanými polohovacími osami pro polohování vzorku a jednou plně motorizovanou indentační osou. Rám zařízení tvoří profily z hliníkové slitiny o průřezu 30 × 30 mm. Rám umožňuje měření rozměrnějších vzorků úpravou vzdálenosti indentační osy a v případě silou řízené indentace osazení siloměru. K rámu jsou připevněny desky z hliníkové slitiny, které pomocí drážek a tvarových spojení zajišťují kolmost indentační osy a povrchu vzorku. Jedna deska nese indentační osu a druhá polohovací osy.

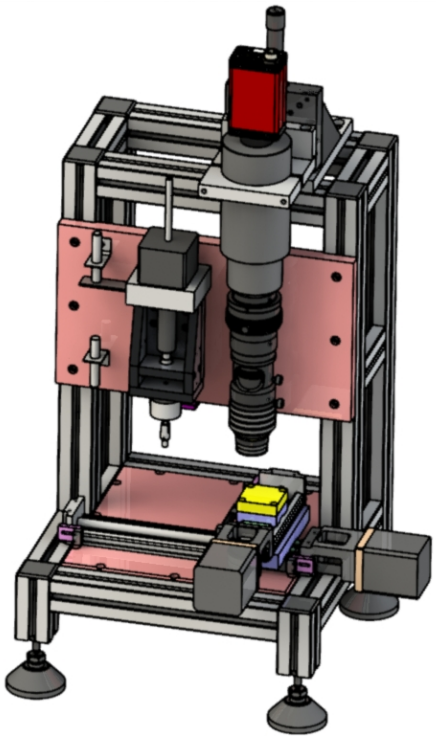
Polohování vzorku je zajištěno dvěma lineárními moduly KK40 (Hiwin, Japonsko) s kuličkovými šrouby a lineárním vedením s vymezenou vůlí. Přesnost nastavení polohy těchto os je 10 μm. Pro zajištění bezpečnosti pohybu je zařízení osazeno koncovými spínači. Pohyb polohovací osy je zajištěn krokovými motory specifikace dle NEMA17. Polohovací osy jsou navzájem kolmé (kolmost je zajištěna přes desku se zafrézovanými drážkami) a na vrchní osu je připevněna broušená deska pro umístění vzorku.

Indentační osa je tvořena přesným lineárním vedením s vysokou tuhostí a lehkým předpětím MGW12 (Hiwin, Japonsko) a pohyb osy je zajištěn přesným krokovým motorem 43H4N (HaydonKerk, USA) s přesností nastavení polohy 3 μm. Pro bezpečnost pohybu indentační osy je tato opatřena dvojicí indukčních (bezkontaktních) koncových spínačů. Na indentační osu je pomocí přesných tvarových spojení umístěna příruba pro připevnění indentoru, závaží (pro indentaci konstantní silou), siloměru (pro indentaci řízenou silou) a třetího indukčního spínače, který indikuje kontakt indentoru se vzorkem při indentaci konstantní silou.

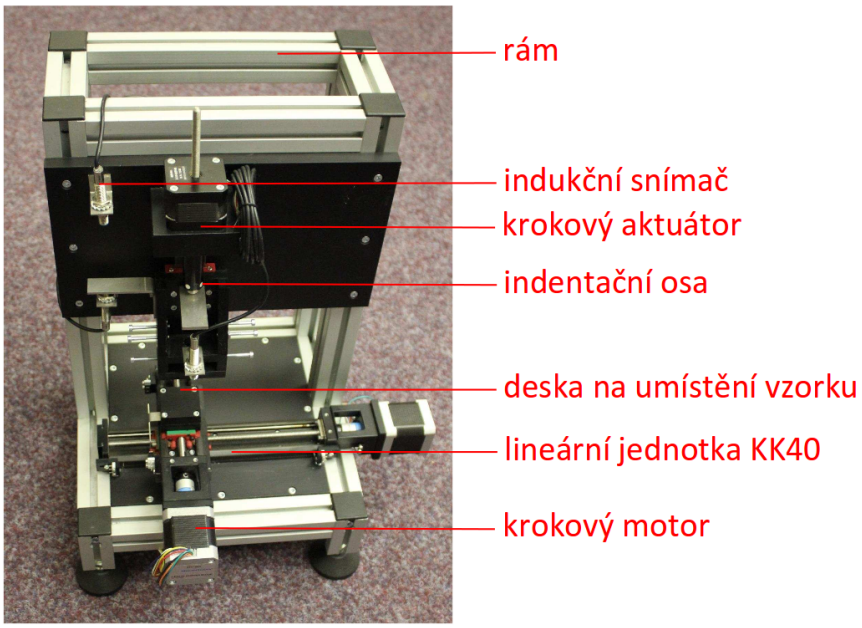
Indentor umožňuje indentaci silou 10 – 100 N. Vizualizace indentačního zařízení je na obrázku 4.2 a fotografie zařízení s popisem jednotlivých částí je na obrázku 4.3.

4.2.2 Elektrická/ řídící část

Indentor je řízen pomocí PC, ke kterému je připojeno následující příslušenství:



Obrázek 4.2: Vizualizace indentoru.



Obrázek 4.3: Fotografie indentoru s popisky jednotlivých částí.

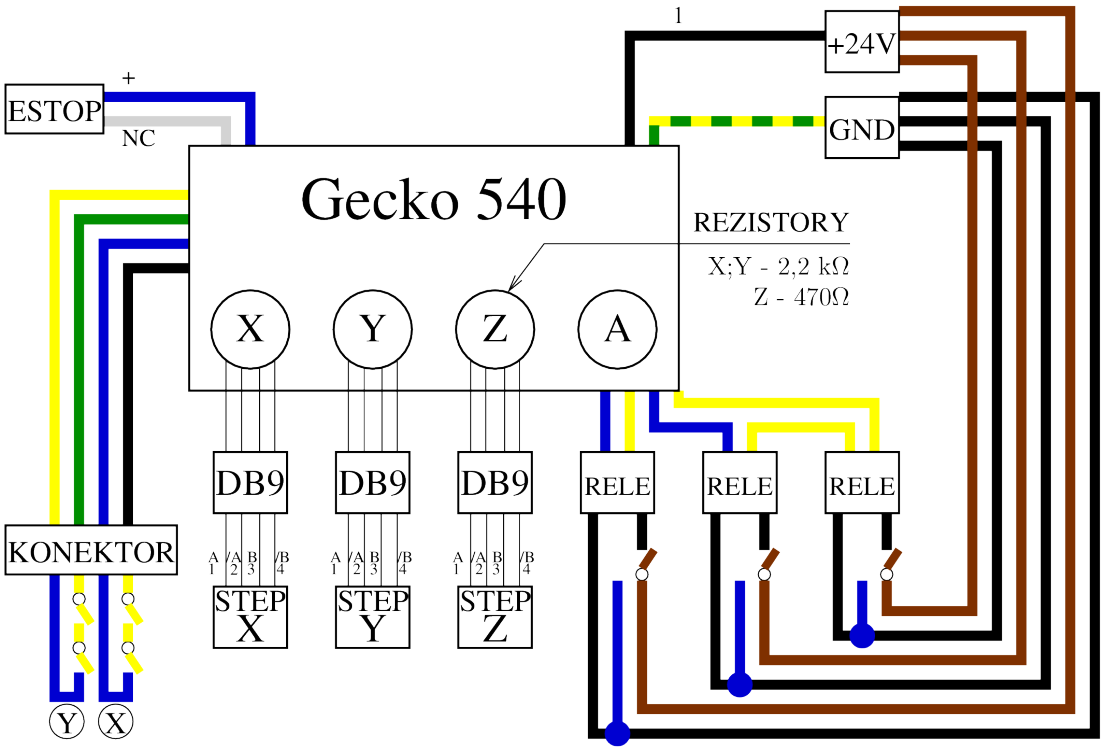
- řídící jednotka krokových motorů,
- vyčítací elektronika siloměru a
- kamera.

Propojení řídící jednotky krokových motorů a PC je realizováno pomocí kabelu pro LPT rozhraní, propojení vyčítání siloměru a PC pomocí USB rozhraní a kamera je připojena pomocí Ethernet rozhraní. Řízení krokových motorů probíhá v reálném čase technologií CNC (computer numerical control) pomocí operačního systému Linux s real-time jádrem. K řízení je využito modulu LinuxCNC, který je blíže popsán v kapitole 5 a řídící aplikace s uživatelským grafickým rozhraním popsaným v části 5.1. Zařízení je dále vybaveno tlačítkem nouzového zastavení, které v případě potřeby okamžitě hardwarově odstavuje napájení krokových motorů i aktuátoru.

Připojení krokových motorů, aktuátoru a tlačítka nouzového zastavení je zobrazeno na obrázku 4.4.

4.2.3 Optická část

Optická část je součástí indentačního zařízení z důvodu možnosti automatické tvorby obrazových dat v průběhu měření. Součástí optické soustavy je CCD kamera (Manta G-504B, AVT, Německo), modulární objektiv (Navitar 6000, USA) a osvětlovací jednotka (Schott KL2500, Německo) vybavená ohebnými světlovody. Kamera je ovládána pomocí pluginu vyrobeného přímo za účelem volání pomocí modulu LinuxCNC. Objektiv s kamerou jsou přichyceny k lineánímu stolku s mikrometrickým šroubem, který umožňuje ostření obrazu přibližováním a oddalováním od povrchu vzorku. Lineární stolek je přichycen k rámu indentoru a polohvání stolku se vzorkem umožňuje po vytvoření vtisku dojet pod kameru a vytvořit obrazová data.



Obrázek 4.4: Schéma zapojení krokových motorů, aktuátoru a tlačítka nouzového zastavení.

Kapitola 5

Řízení indentoru – LinuxCNC

Pro řízení indentoru, jakožto laboratorního zařízení, je vhodné převzít principy polohování z obráběcích CNC (computer numerical control) strojů. Laboratorní zařízení s obráběcími stroji jsou si v mnohém podobné:

- mají stejné druhy pohonů (krokové, nebo servo motory),
- musejí být schopny přesného polohování,
- jsou schopny synchronizovat pohyb os.

Z jednotlivých možností řízení strojů technologií CNC byl vybrán modul LinuxCNC verze 2.6.4. LinuxCNC (nebo-li Enhanced Machine Control) je software určený pro počítačové řízení v reálném čase technologií CNC. Slouží k řízení zejména obráběcích strojů, jako jsou frézky nebo soustruhy, slouží ale například také k řízení robotů. Vzhedem k tomu, že se jedná o open source kód, je jeho použití možné bezplatně. Současná verze LinuxCNC je licencovaná pod GNU General Public License a Lesser GNU General Public License (GPL a LGPL). LinuxCNC je kompatibilní s jakýmkoliv operačním systémem Linux s real-time jádrem. V našem případě byl použit LinuxCNC verze 2.6.4 ve spojení s operačním systémem Debian Wheezy, což je verze předkompilovaná komunitou.

Modul LinuxCNC v používané verzi je schopen ovládat najednou až 9 os a interpretovat programovací jazyk G – code (RS – 274), který se používá pro programování obráběcích strojů. Typickými příkazy G – codu jsou rychlý pohyb do daných souřadnic, pohyb danou rychlostí do zadaných souřadnic, pohyb po oblouku nebo změna souřadného systému. LinuxCNC nabízí také několik typů uživatelských rozhraní, včetně uživatelského rozhraní

pro dotykové disleje. Pro instrumentovaný mikroindentor bylo ovšem v rámci této práce vytvořeno vlastní uživatelské rozhraní, viz část 5.1.

LinuxCNC je software pro řízení polohování a jako takový nepodporuje tvorbu výkresů (CAD – Computer Aided Design, česky Počítačová podpora projektování) ani generování G – codu z výkresů (CAM- Computer Aided Manufacturing, česky Počítačová podpora obrábění).

5.1 Grafické uživatelské rozhraní

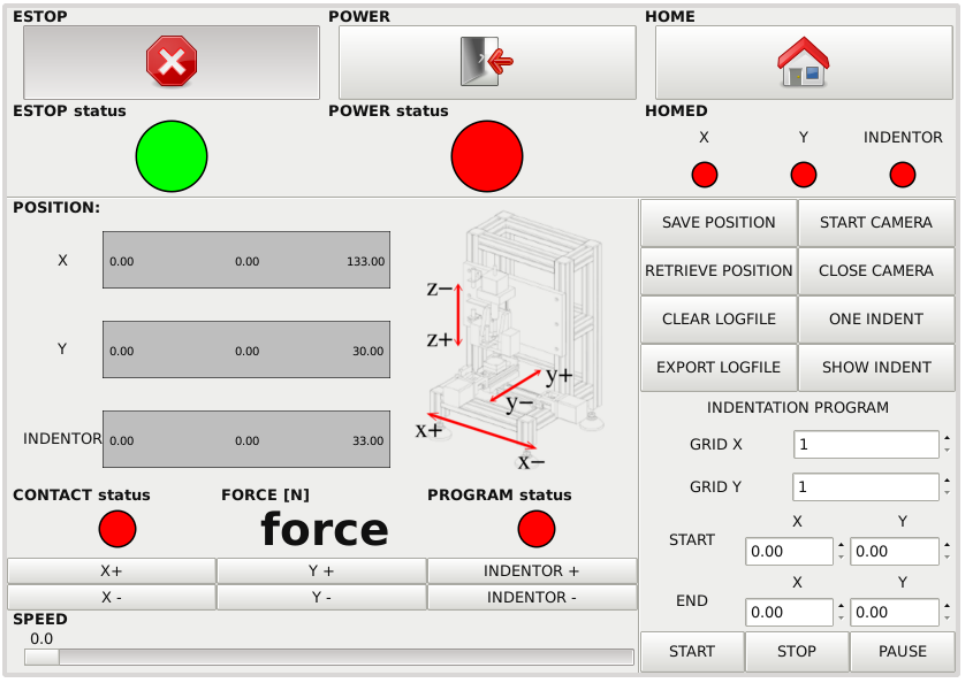
Vzhled grafického uživatelského rozhraní (anglicky Graphical User Interface, známé pod zkratkou GUI) byl realizován pomocí modulu projektu Glade Virtual Control Panel (GladeVCP). GladeVCP je součástí LinuxCNC, která umožňuje tvorbu vlastního grafického uživatelského rozhraní pro vlastní varianty řízení konkrétních strojů. Používá k tomu programovací jazyk Python spolu s Glade tvorbou rozhraní. Při tvorbě řídicího programu a skriptů se držíme logiky a teorie řízení pro obráběcí stroje a modifikujeme ji pro účely laboratorních zařízení.

Byla vytvořena dvě GUI, každé pro jinou aplikaci indentoru:

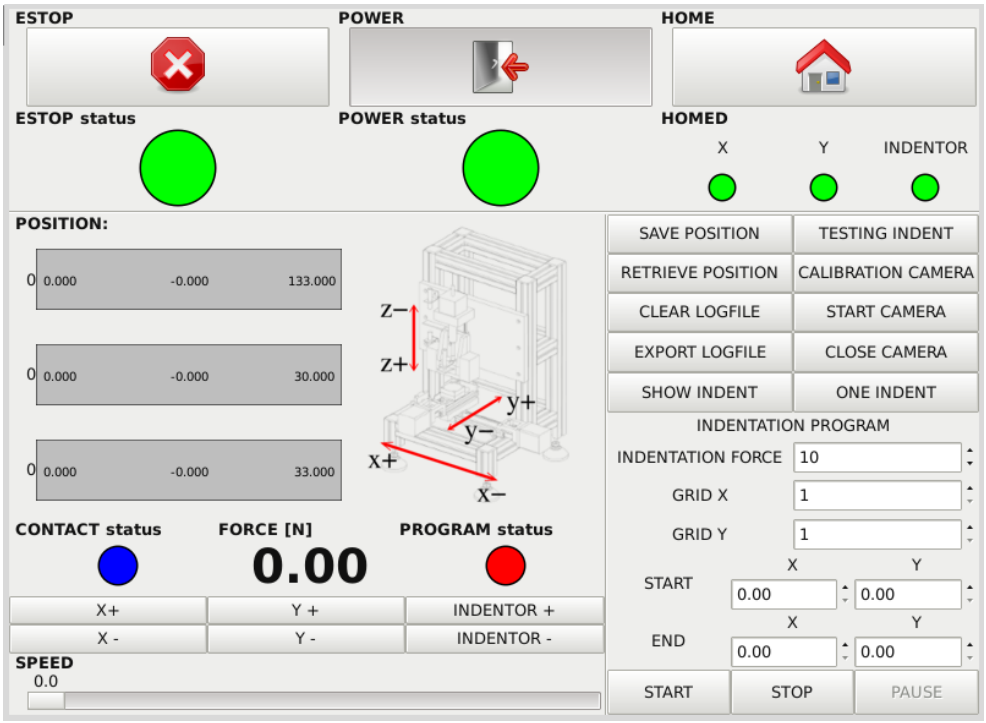
- pro indentaci konstantní silou, kdy jsou na indentor zavěšena závaží, je GUI navrženo jako na obrázku 5.1,
- pro indentaci řízenou silou (indentor osazený siloměrem), je vzhled GUI na obrázku 5.2.

Obě grafická rozhraní jsou rozdělena na tři části. Vrchní část a levá strana spodní části jsou shodné jak pro indentaci konstantní silou, tak pro indentaci řízenou silou. GUI je provedeno tak, že při spuštění indentoru je aktivní pouze horní část popsaná níže, kdežto spodní část je aktivní až po splnění podmínky bezpečného provozu. To znamená, že tlačítko nouzového vypnutí není aktivováno, tlačítko pro zapnutí indentoru je aktivováno a všechny osy mají nastavenou výchozí pozici.

V horní části jsou umístěna tlačítka a kontrolní diody. Zleva se jedná o tlačítko a kontrolní diodu tlačítka nouzového zastavení. Kontrolní dioda má dvě polohy, červená indikuje, že je aktivní tlačítko nouzového zastavení (jakýkoliv pohyb os indentoru není možný, krokové motory i aktuátor jsou hardwarově odstaveny), a zelená indikuje, že tlačítko nouzového zastavení není aktivní (indentace je možná). Dalším tlačítkem je zapnutí, jehož



Obrázek 5.1: Grafické uživatelské rozhraní pro indentaci konstantní silou se závažím.



Obrázek 5.2: Grafické uživatelské rozhraní pro indentaci řízenou silou (osazeno siloměrem).

kontrolní dioda červeně indikuje, pokud je indentor vypnutý, a zeleně, že je zapnutý. Posledním tlačítkem je tlačítko, které nastaví všechny tři osy do výchozí pozice. Toto nastavení je blíže popsáno v části 5.2.1. Pokud má daná osa (x, y, nebo osa indentoru) nastavenou výchozí pozici svítí dioda pod názvem příslušné osy zeleně, pokud osa ještě nastavenou výchozí pozici nemá, svítí tato dioda červeně.

Spodní segment je rozdělen na část levou a část pravou, jak je možné vidět na obrázcích 5.1 a 5.2. Levá část obsahuje informaci o současné poloze jednotlivých os a obrázek indentačního zařízení, na kterém je znázorněna orientace posuvu v jednotlivých směrech. V levé spodní části jsou dále umístěny dvě diody, kdy jedna indikuje, zda je hrot indentoru v kontaktu s povrchem materiálu (červeně), či nikoliv (modře). Druhá dioda signalizuje, zda běží indentační program (červeně), či nikoliv (zeleně). Mezi těmito diodami je zobrazena hodnota síly vyčtená ze siloměru zaokrouhlená na dvě desetinná místa. Pro polohování jednotlivých os je v levé spodní části dále umístěno šest tlačítek pro pohyb os (pro každou osu oba směry) a posuvník, na kterém se určí rychlost posunu os.

Pravá spodní část se již liší pro jednotlivé typy zatěžování. Pro indentaci konstantní silou jsou zde tlačítka:

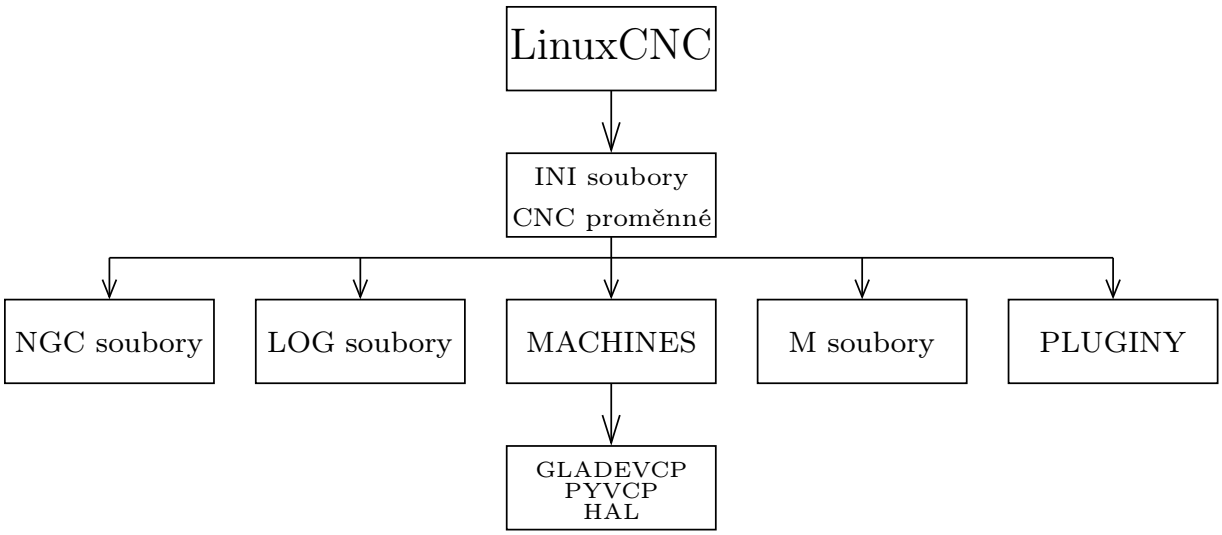
- uložit pozici – uložení aktuální pozice všech os do proměnných,
- obnovit pozici – obnovení pozice všech os, tak jak byly uložené tlačítkem ulož pozici,
- vyčistit log soubor – vymazání log souboru, do kterého se ukládají v dané sekvenci informace o poloze všech os a síle na siloměru,
- exportovat logfile – uložení souboru s naměřenými daty do dané složky v počítači,
- spustit kameru – spustí v novém okně aktuální obraz kamery s křížem pro lepší orientaci,
- vypnout kameru – tlačítko vypne výstup z kamery,
- jeden vtisk – toto tlačítko spustí indentační program pro vytvoření jednoho vtisku, program je blíže popsán v částech 5.3 a 5.4,
- ukázat indent – spustí program, který ukáže indent

Dále se v pravé spodní části nachází indentační program, který umožňuje automaticky provádět měření pomocí pravoúhlého gridu v libovolné oblasti vzorku a libovolným počtem indentů, zadání souřadnice x a y, kde má být první vtisk a poslední vtisk a tři tlačítka pro spuštění programu, pauzu běžícího programu a zastavení programu. Příklad mřížky vtisků vytvořený tímto indentačním programu je zobrazen na obrázku 4.1.

Při indentaci řízené silou je v pravé spodní části navíc tlačítko testovací vtisk, které spustí program vytvářející první testovací vtisk pro nastavení rychlosti indentace a tlačítko kalibrace Manty (více o této kalibraci viz část 5.2.2). Další odlišností GUI při indentaci řízené silou je zadání síly, kterou se má indentace provádět.

5.2 Koncept ukládání souborů pro LinuxCNC

V rámci tvorby ovládání indentoru byl navržen nový koncept adresářové struktury pro LinuxCNC z důvodu existence již několika strojů vzniklých na Oddělení biomechaniky ÚTAM AV ČR v.v.i. V konceptu se počítá s tím, že všechny existující stroje řízené metodou CNC budou využívat tento koncept tak, aby byla zajištěna univerzalita využívání totožných skriptů u všech strojů a zjednodušila se tak aktualizace při jejich úpravách. Na obrázku 5.3 je znázorněná koncepce rozvržení složek pro ukládání souborů potřebných pro běh strojů pomocí LinuxCNC.



Obrázek 5.3: Koncepce rozvržení složek pro ukládání souborů pro LinuxCNC.

5.2.1 INI soubory

Ve složce LinuxCNC jsou uloženy INI soubory jednotlivých strojů. INI soubor mikroindentoru s konstantním zatížením je v příloze A.1.1 a INI soubor indentoru, který je řízený silou je v příloze A.1.2. INI soubor obsahuje základní nastavení stroje a je zde obsažena informace o [32]:

- názvu stroje,
- umístění a vzhledu grafického uživatelského rozhraní, viz část 5.1,
- cestě k proměnným, které jsou trvale ukládány i po ukončení práce stroje,
- cestě k M souborům, více v části 5.2.7,
- cestě k HAL souborům, více viz část 5.2.5,
- uživatelem definované MDI příkazy, které se aktivují pomocí HAL pinů,
- údaje o počtu os, volbě jednotek, maximálních použitelných rychlostech,
- nastavení jednotlivých os.

Indentor má v INI souboru nastavené osy x, y a indentační osu. Použitými jednotkami posuvu jsou mm. Osa x má kvůli své fyzické délce nastavený maximální softwarový limit pohybu na 133 mm, pro optimální běh os je nastavena maximální rychlost pohybu 5 mm/s a maximální zrychlení 30 mm/s^2. Osa y má stejné hodnoty maximální rychlosti i zrychlení, maximální limit posuvu je nastaven na 30 mm. Indentační osa má nastavenou maximální rychlost pohybu na 3 mm/s, maximální zrychlení opět 30 mm/s^2 a maximální limit posuvu osy je nastaven na 33 mm.

V INI souboru je také definována sekvence pro nastavení výchozí pozice jednotlivých os. Nejprve je nalezena výchozí pozice osy indentoru a poté výchozí pozice zbylých dvou os, tak aby nemohlo dojít k poničení hrotu a siloměru tečnou silou při současném pohybu všech tří os. Nalezení výchozí pozice probíhá tak, že se osa pohybuje směrem na začátek, dokud se nesepne koncový spínač. Při sepnutí spínače se osa začne pomalu pohybovat opačným směrem do rozepnutí spínače a od tohoto bodu ještě o 0,5 mm.

5.2.2 CNC proměnné

Do textového souboru CNC proměnné se ukládají hodnoty, které si stroj při svém spuštění načítá pro svou správnou funkci. Tento soubor je jedinou možností, jak si stroj řízený pomocí modulu LinuxCNC uchová jednotlivé pozice, protože všechny ostatní proměnné jsou po vyputí stroje ztraceny. Tohoto souboru proměnných se využívá pro polohování posuvného stolku se vzorkem, kdy je třeba znát vzdálenost mezi indentorem a objektivem kamery, aby bylo umožněno automatické pořízení fotografií všech vtisků. Vzhledem k tomu, že je možné tuto informaci uchovat v souboru CNC proměnné, není potřeba při každém spuštění stroje kalibrovat přesnou pozici kamery.

Pozici kamery je ovšem třeba kalibrovat při vyjmutí a opětovném navrácení kamery (pozice kamery se změní) pomocí skriptu v příloze A.2.5. Tento skript funguje tak, že se při spuštění stroje nastaví počáteční souřadnice x start y start, vytvoří se zkušební vtisk a poté se manuálně najede s tímto vtiskem pod objektiv kamery. Ve chvíli, kdy je vtisk umístěn ve středu obrazu snímaného kamerou se spustí kalibrační skript, který si uloží potřebné vzdálenosti do souboru CNC proměnné.

5.2.3 NGC soubory

Do složky NGC soubory se ukládají všechny soubory G – code. G – code je programovací jazyk, který se obecně používá pro polohování CNC strojů, kde všechny příkazy jsou uvozeny písmenem G, následovaným číslem, které rozlišuje jednotlivé příkazy. Kompletní dokumentaci G – code lze nalézt v [33]. Pro vytvoření podmínek a cyklů jsou v LinuxCNC vytvořeny tzv. O – code. Dvojice O – code je také pro uložení celých skriptů, nebo jejich částí a poté možnosti jejich zavolání v jiném skriptu.

5.2.4 LOG soubory

Ve složce LOG soubory jsou uloženy skripty pro ukládání dat z měření, soubor pro zapisování měřených hodnot a skripty pro zobrazování zatěžovacích křivek v reálném čase psané v syntaxi Bash (Unix Shell interpreter) a v příkazovém jazyce Gnuplot.

5.2.5 Machines (stroje)

Ve složce stroje (Machines) jsou uloženy v jednotlivých podsložkách všechny stroje. Tyto podsložky obsahují soubor se vzhledem grafického uživatelského rozhraní vytvořeném buď v GladeVCP nebo PyVCP.

Ve složce stroje jsou dále HAL dokumenty. HAL je zkratka pro Hardware Abstraction Layer (česky hardwarová abstrakční vrstva) a jedná se o vrstvu, která načítá jednotlivé „stavební bloky“ a umožňuje jim se spojit k dosažení komplexního systému. Hal je založen na stejných principech, které jsou použity při navrhování hardwarových řídících obvodů. Příkladem fungování HALu může být spojení dvou signálů pomocí funkce `and`. V prvním kroku jsou vybrány signály, které chceme spojit pomocí logické funkce `and`. Dalším krokem je jejich propojení pomocí vhodné funkce, v našem případě funkce `and`. A posledním krokem je finální uskutečnění spojení dvou signálů pomocí logické funkce `and`.

Na jednoduchém příkladu bylo ukázáno, jak pomocí HALu dochází k propojování jednotlivých komponent a na tomto základě pak funguje celá řídící logika daného stroje [34].

Halové soubory indentoru řízeného silou jsou v příloze A.4 a halové soubory indentoru s konstantním zatížením jsou v příloze A.3.

5.2.6 Pluginy

Ve složce Pluginy jsou uloženy zásuvné moduly (neboli pluginy) pro čtení systémového času a pro vyčítání siloměru. Zásuvný modul je software, který pracuje jako doplněk jiné aplikace pro rozšíření její funkce.

5.2.7 M soubory

G – code obsahuje kromě funkcí začínajících na písmeno G, tak jak byly popsány v části 5.2.3 navíc příkazy začínajících na písmeno M (tzv. M – code). M – code pracuje s funkcemi, které jsou definovány uživatelem. Tyto M – code nesou označení M100 až M199 a funkce mohou obsahovat libovolné procedury podporované operačním systémem. Funkce M – code jsou vytvořené v syntaxi Bash (Unix Shell interpreter).

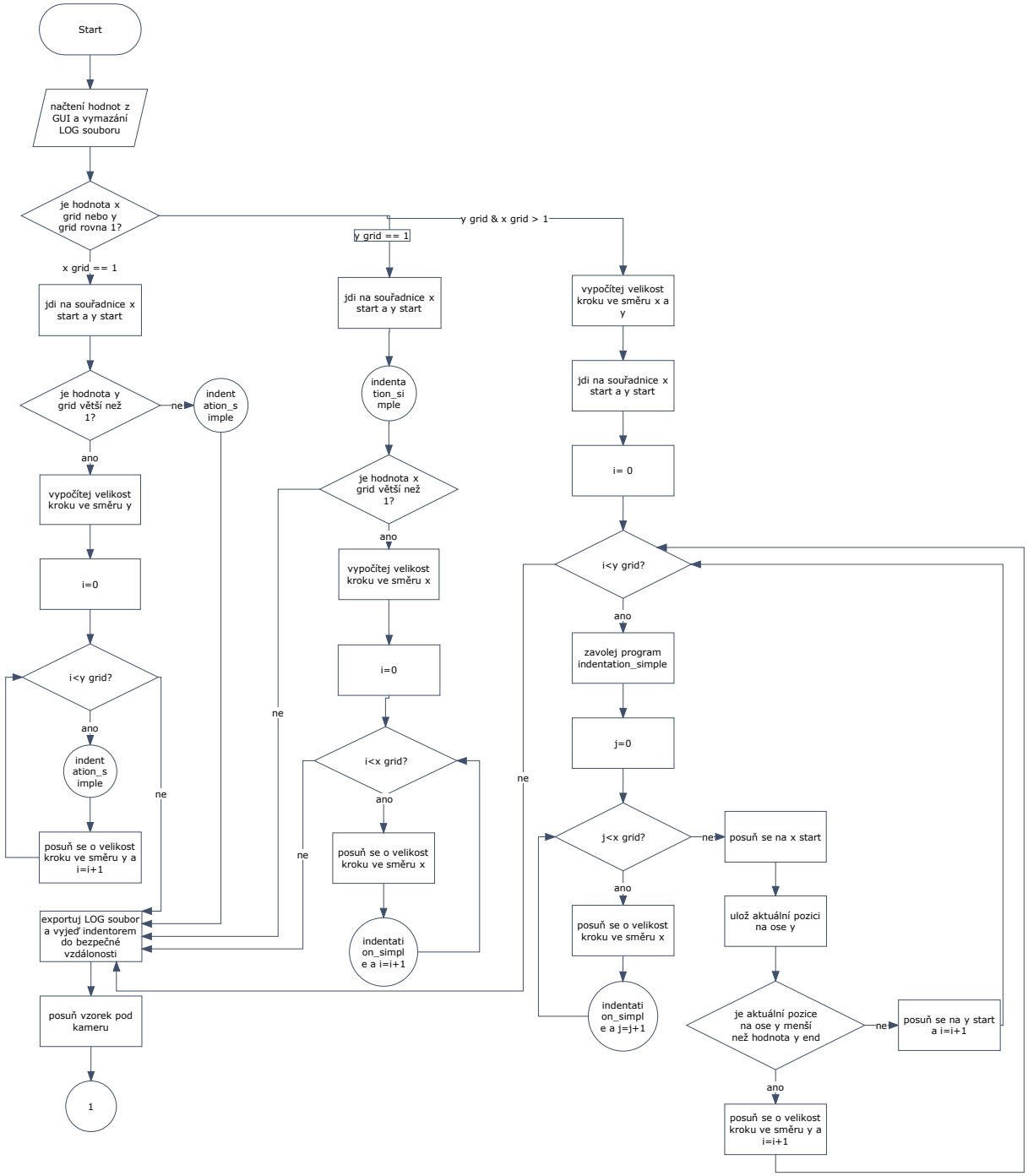
Pro indentor byly vytvořeny nebo použity (z již existujících strojů vyrobených na Oddělení biomechaniky Ústavu teoretické a aplikované mechaniky AV ČR, v.v.i.) M funkce, které jsou volány pomocí GUI popř. automaticky během měřícího programu:

- M123 (viz příloha A.5.1)
  - zobrazí živý náhled grafu působící síly,
- M124 (viz příloha A.5.2)
  - změní jméno log souboru na unikátní jméno podle data a času měření a uloží ho do příslušného adresáře,
- M125 (viz příloha A.5.3)
  - odstraní log soubor. Tato M funkce se používá před začátkem měření, aby v log souboru byly pouze hodnoty z příslušného měření,
- M126 (viz příloha A.5.4)
  - uloží výstup z kamery do formátu png,
- M127 (viz příloha A.5.5)
  - spustí kameru v samostatném okně,
- M128 (viz příloha A.5.6)
  - vypne kameru spuštěnou pomocí funkce M127.

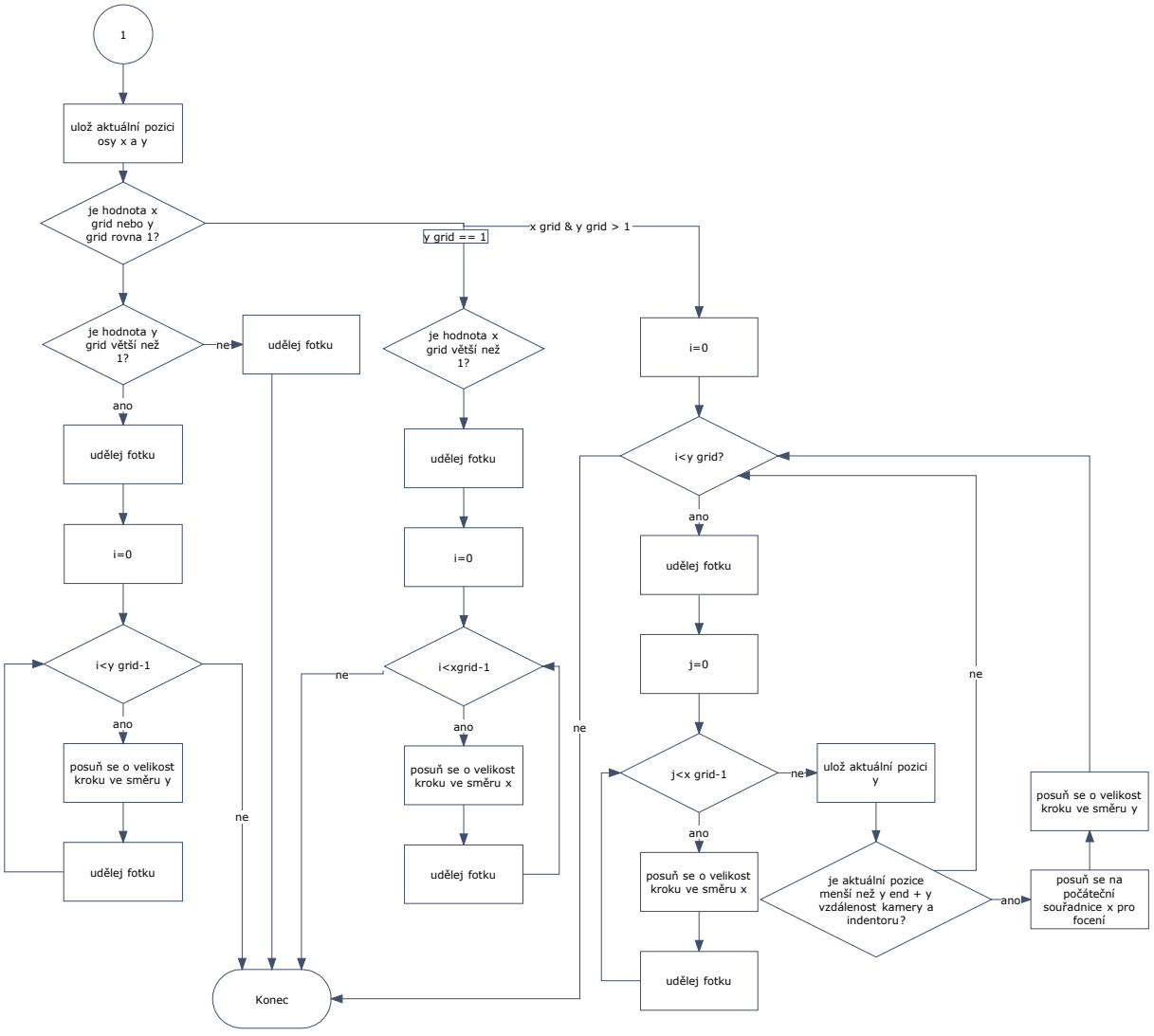
M funkce jsou propojeny s tlačítky na GUI pomocí MDI příkazů, které jsou definovány v INI souboru a vlastní propojení probíhá v souboru `custom_postgui.hal` v příloze A.3.2 a A.4.2. Funkce M123 a M126 nejsou přímo provázány s GUI, ale jsou volány v průběhu indentačního programu. Využití je ukázáno v jednotlivých vývojových diagramech v části 5.3.

5.3 Indentační programy indentoru se zatížením konstantní silou

Pro indentaci pomocí konstatní síly byla vytvořena série skriptů zapsaných pomocí G – code. Základní procedurou pro indentaci je program `microindentor_simple.ngc` (viz příloha A.2.1). Vývojový diagram tohoto programu je znázorněn na obrázcích 5.4 a 5.5.



Obrázek 5.4: První část vývojového diagramu indentačního programu `microindentor_simple.ngc`.



Obrázek 5.5: Druhá část vývojového diagramu indentačního programu `microindentor_simple.ngc`.

Tato indentační procedura volá ve svém průběhu program pro vytváření jednotlivých vtisků nazvaný `indentation_simple.ngc` (viz příloha A.2.2). Vývojový diagram tohoto programu je na obrázku 5.6.

Dalším indentačním programem je program `one_indent_simple.ngc` (viz příloha A.2.9), který na vybraném místě vzorku (výběr vhodného místa probíhá pomocí kamery) vytvoří jeden vtisk do povrchu vzorku. Tento program opět v průběhu volá program `indentation_simple.ngc`. Vývojový diagram programu na vytvoření jednoho vtisku je znázorněn na obrázku 5.7.

Dalším programem použitým jak při indentaci konstantní silou, tak při indentaci řízené silou, je program `show_indent.ngc` (viz příloha A.2.10). Tento program posune stolek se vzorkem ze současné pozice pod kameru a kameru spustí, aby bylo možné prohlédnout si aktuálně vytvořený vtisk.

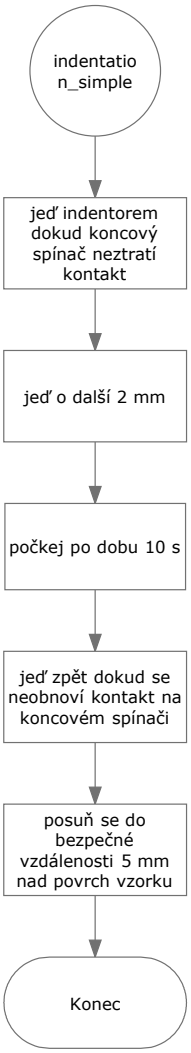
5.4 Indentační programy indentoru řízeného silou

Indentační programy pro indentaci řízenou silou musely být oproti indentačním programům při indentaci konstantní silou (popsáno v podkapitole 5.3) změněny kvůli odlišné konstrukci indentačního zařízení. Největší změnou oproti indentaci konstantní silou je potřeba vytvoření zkušebního vtisku, aby bylo možné dodržet normu pro indentaci podle Vickerse, která vyžaduje, aby maximálního zatížení bylo dosaženo do 10 s při indenatci nízkým zatížením. Zkušební indent zaznamená pozici indentoru při kontaktu indentoru s povrchem vzorku a pozici indentoru při dosažení zkušebního zatížení, které zadal uživatel a z těchto hodnot vypočítá rychlost, kterou musí probíhat indentace, aby bylo zkušební zatížení dosaženo do 10 s. Vývojový diagram programu na vytvoření testovacího vtisku je zobrazen na obrázku 5.8.

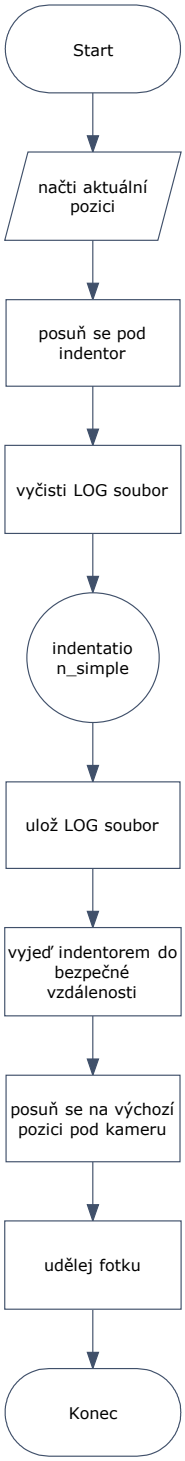
Hlavním indentačním programem pro indentaci řízenou silou je program `microindentor_instrumented.ngc`. Vývojový diagram tohoto programu je zobrazen na obrázcích 5.9 a 5.10.

Tento indentační program volá ve svém průběhu pro vytváření jednotlivých vtisků program nazvaný `indentation_instrumented.ngc`. Vývojový diagram tohoto programu je na obrázku 5.11

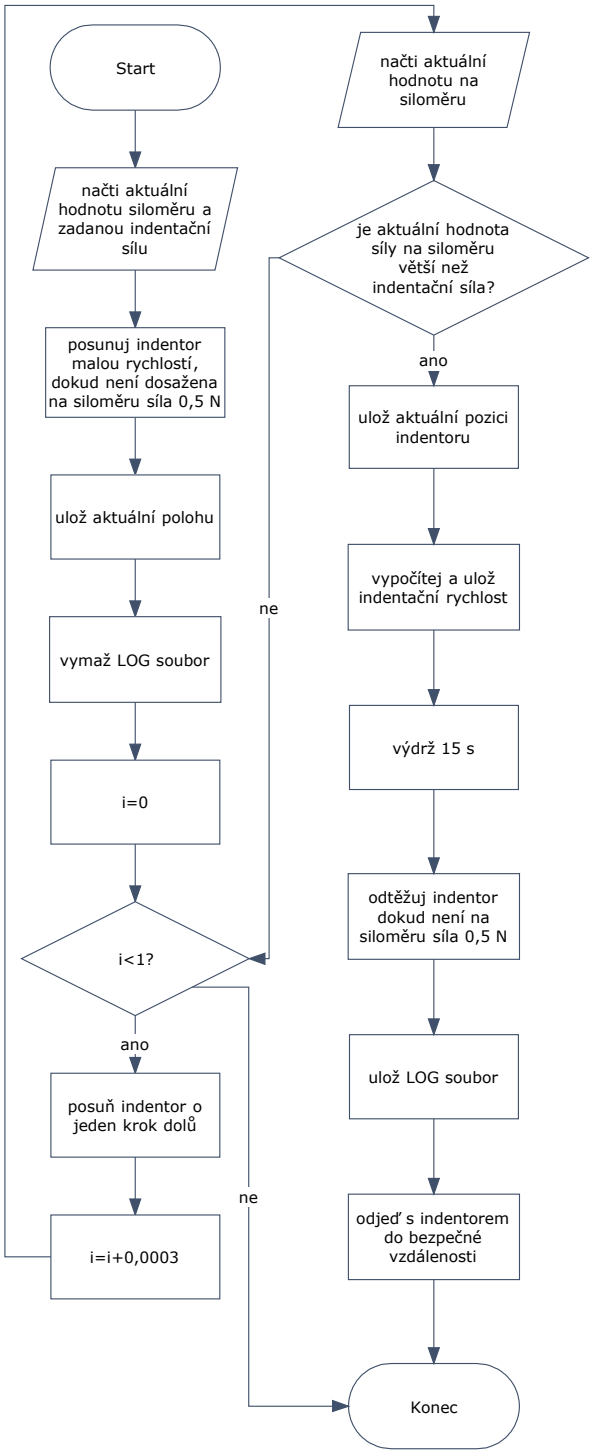
Indentační program pro vytvoření jednoho indentu `one_indent_instrumented` se liší od programu `one_indent_simple` pouze voláním buď indentačního programu



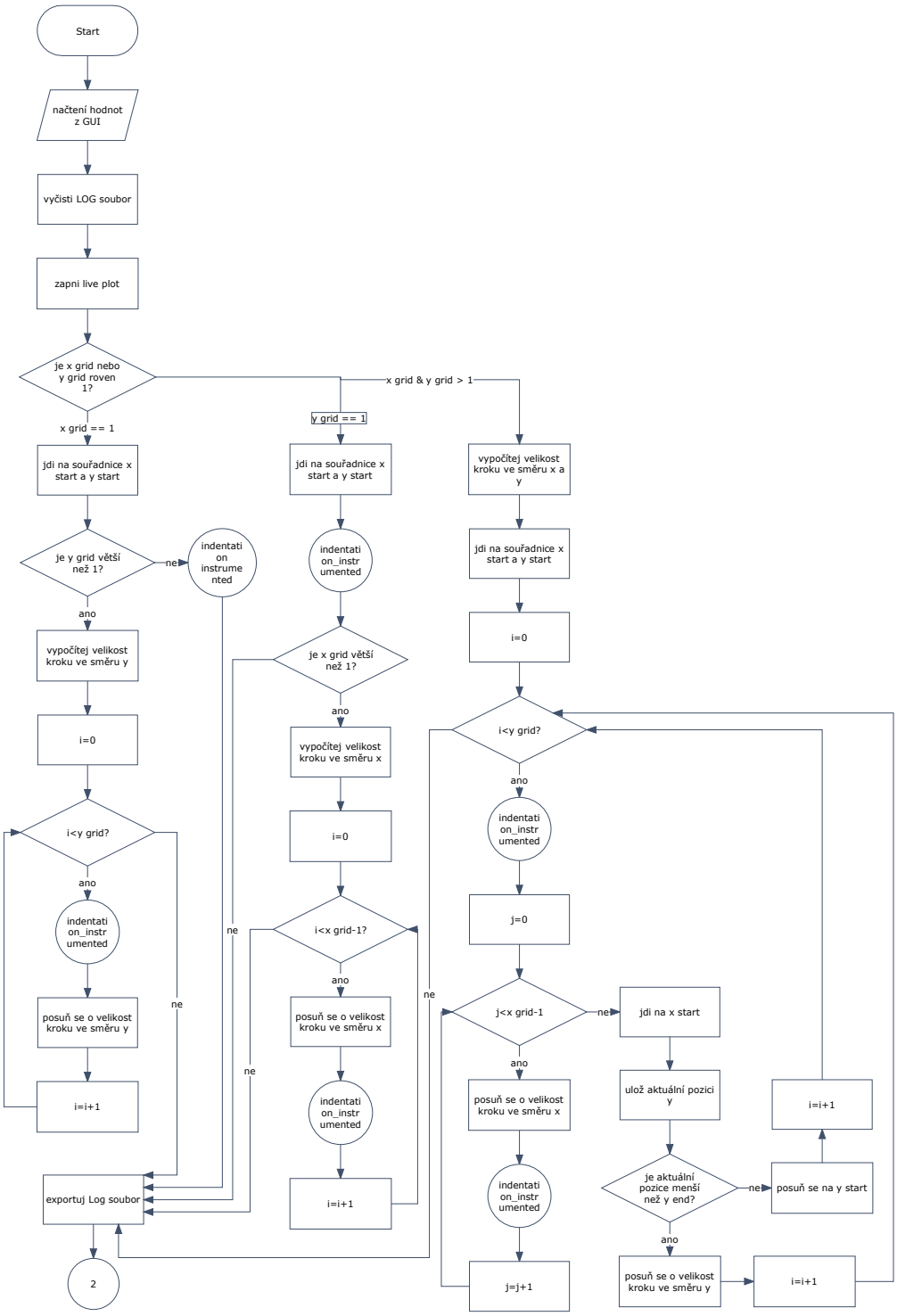
Obrázek 5.6: Vývojový diagram indentačního programu `indentation_simple.ngc`.



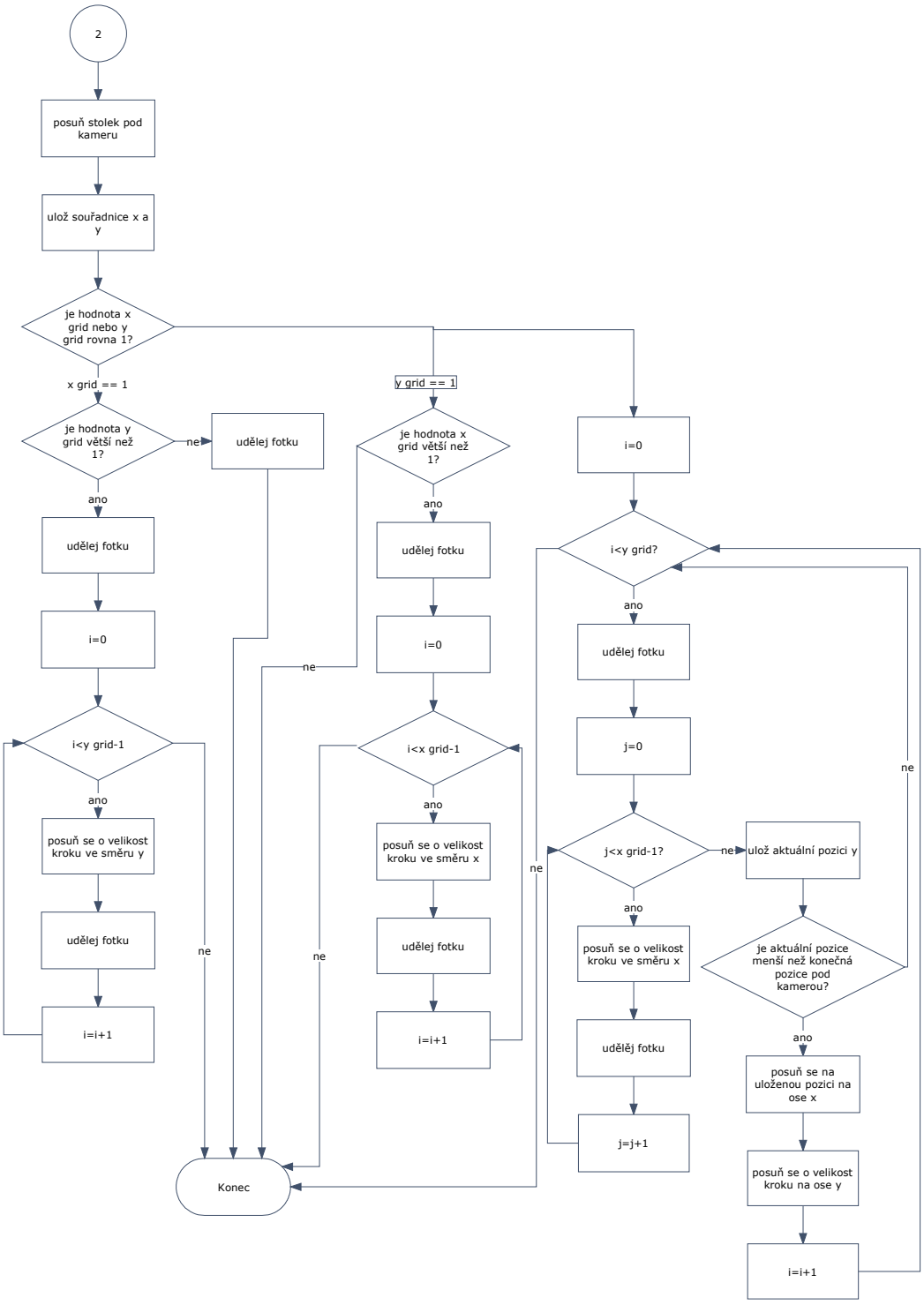
Obrázek 5.7: Vývojový diagram indentačního programu `one_indent_simple.ngc`.



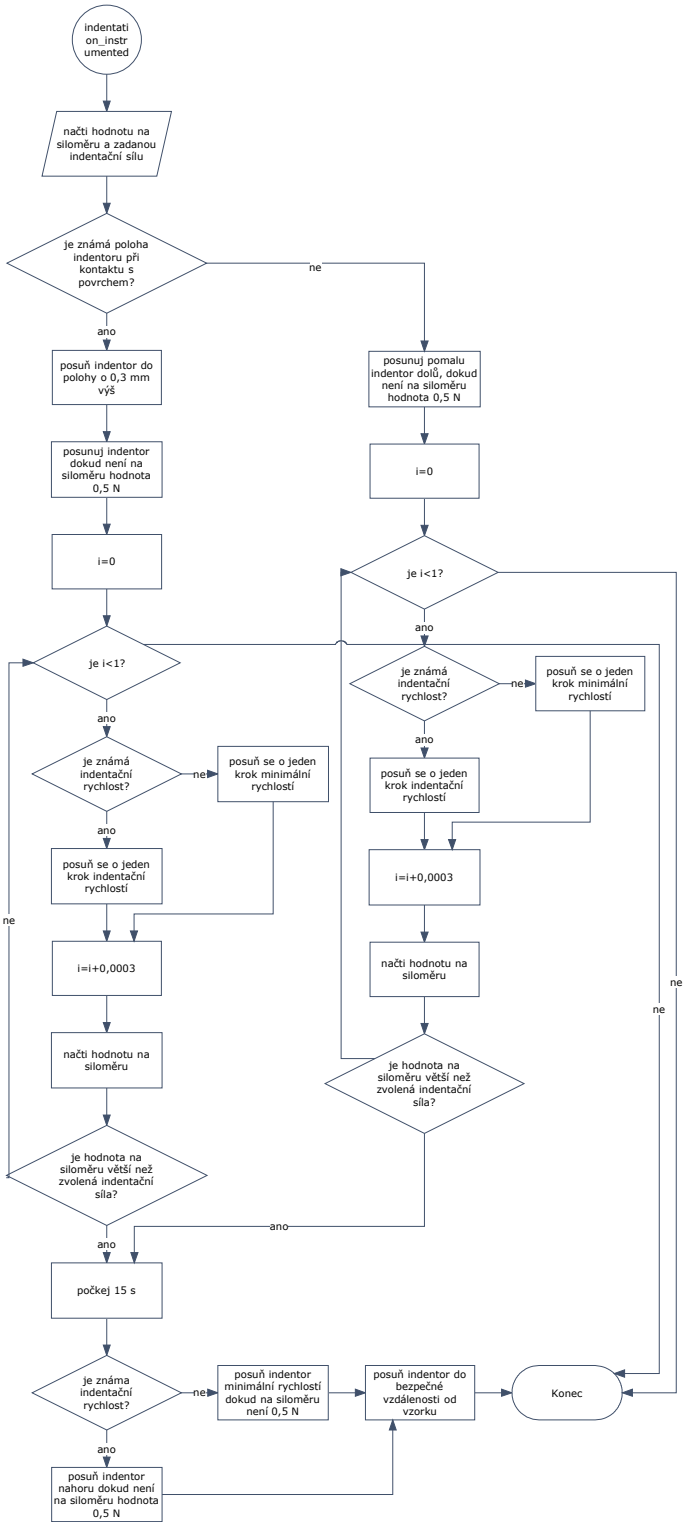
Obrázek 5.8: Vývojový diagram indentačního programu `testing_indent.ngc`.



Obrázek 5.9: První část vývojového diagramu indentačního programu `microindentor_instrumented.ngc`.



Obrázek 5.10: Druhá část vývojového diagramu indentačního programu `microindentor_instrumented.ngc`.



Obrázek 5.11: Vývojový diagram indentačního programu `indentation_instrumented.ngc`.

`indentation_instrumented.ngc` nebo `testing_indent.ngc` namísto `indentation_simple.ngc`.

Kapitola 6

Měření na tvrdoměrné destičce

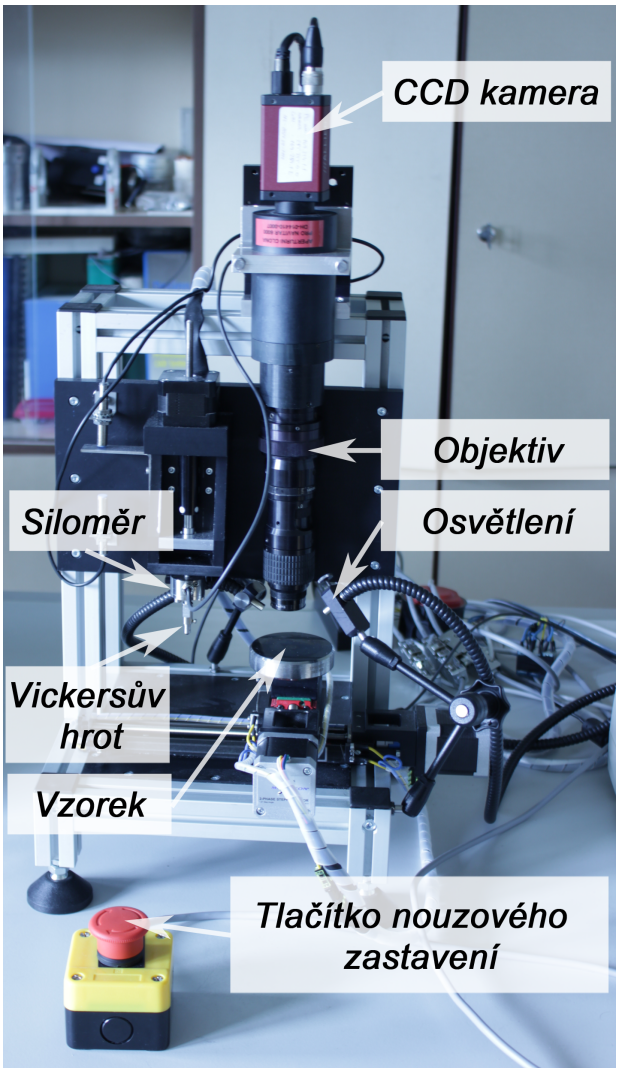
Pro ověření funkčnosti navrženého zařízení a funkčnosti měřících procedur, bylo provedeno měření na materiálu o známé tvrdosti – referenční tvrdoměrné destičce. Měření bylo provedeno na sérii pěti vtisků a byla vyhodnocena spolehlivost tvrdoměru a změřena chyba tvrdoměru v souladu s normou pro ověřování funkčnosti indentorů.

6.1 Referenční tvrdoměrná destička

Pro měření byla použita referenční tvrdoměrná destička (ITW Test & Measurement GmbH, Německo), pro kterou vydala certifikát Euro Products Calibration Laboratory. Kalibrace destičky byla provedena tak, že se do destičky vytvořilo deset vtisků Vickersovým indentorem se zatížením 3 kp na různých místech vzorků znázorněných v kalibračním certifikátu dodaným s tvrdoměrnou destičkou. Deklarovaná střední hodnota tvrdosti kalibrační destičky je 299,2 HV 3. Maximální změřená hodnota tvrdosti byla 301,2 HV 3 a minimální změřená hodnota tvrdosti byla 297,2 HV 3. Chyba měření byla stanovena na 1,5 HV.

6.2 Průběh měření

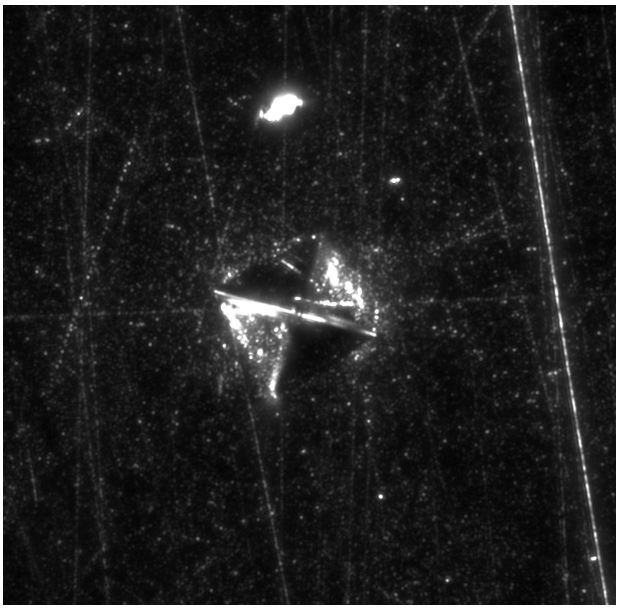
Experimentální sestava jejíž konstrukce byla popsána v kapitole 4 a řízení v kapitole 5 použitá pro kalibrační měření je zobrazena na obrázku 6.1.



Obrázek 6.1: Experimentální sestava instrumentovaného indentoru a tvrdoměrné destičky.

Tvrdoměrná destička se umístí na posuvný stolek indentoru a spustí se řídící software indentoru řízeného silou. Pomocí příslušného tlačítka se spustí kamera a polohováním posuvného stolku se nalezne vhodné místo pro umístění zkušebního vtisku. Zkušební vtisk byl vytvořen z důvodu určení souřadnice kontaktu indentoru s povrchem vzorku a na stanovení rychlosti pohybu indentoru kvůli splnění podmínky rychlosti indentace (plná zatěžovací síla musí být při indentaci podle Vickerse dosažena do 10 s). Po vytvoření zkušebního vtisku bylo možné vytvořit sérii pěti vtisků potřebnou při kontrolním měření tvrdoměru. Poté byla spuštěna kamera a nalezeno místo pro provedení vtisku a v tomto místě místě byl vytvořen vtisk zatěžovací silou 30 N pomocí indentačního programu `one_indent_instrumented.ngc`

(více o tomto zatěžovacím programu viz 5.4). Při vytváření obrazových dat jednotlivých vtisků je zásadní správné nasvícení a zaostření vtisku, tak aby byly hrany vtisku jasně identifikovatelné a bylo umožněno správné vyhodnocení měření. Z toho důvodu se po vytvoření vtisku opět spustila kamera, opravilo se nasvícení a zaostření a uložil se obrázek vtisku. Tento postup byl zopakován pětkrát. Obrazová data jednoho z vtisků jsou zobrazena na obrázku 6.2



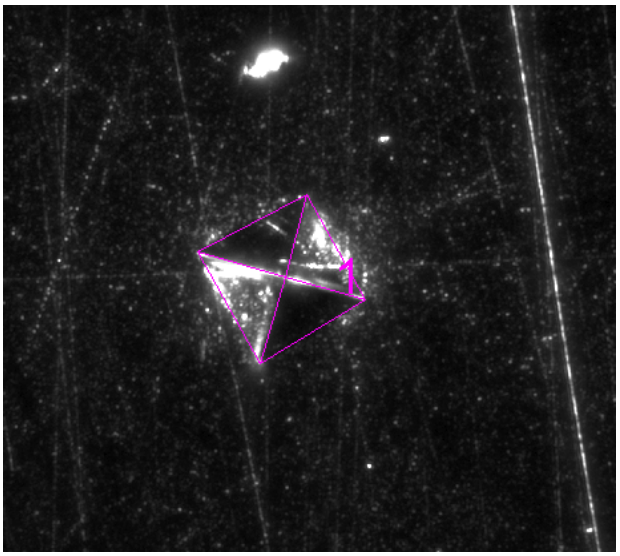
Obrázek 6.2: Vtisk vzniklý při indentaci tvrdoměrné destičky silou 30 N

6.3 Vyhodnocení měření

Vyhodnocení měření bylo provedeno z obrazových dat pořízených optickou soustavou indentoru pomocí standardní metody vyhodnocování Vickersovy zkoušky tvrdosti při nízkém zatížení. Měření nebylo vyhodnocováno metodou Olivera a Pharra z důvodů popsaných v části 6.3.2.

6.3.1 Vyhodnocení měření z obrazových dat

Vyhodnocení měření bylo provedeno pomocí skriptu pro vyhodnocení indentace podle Vickerse (viz příloha B.1.1), který vznikl v rámci mé bakalářské práce [31]. Skript pro výpočet tvrdosti ve svém běhu volá další funkci pro nahrání obrazových dat (viz příloha



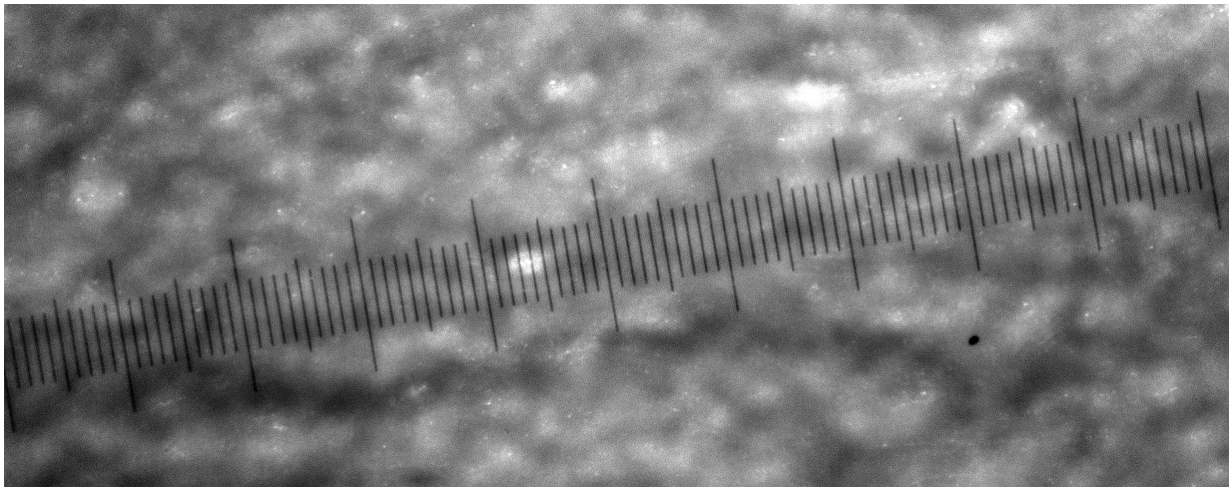
Obrázek 6.3: Označený vtisk po indentaci tvrdoměrné destičky.

B.1.2). Optickou soustavou byla pořízena obrazová data jednotlivých vtisků provedených do referenční tvrdoměrné destičky. Jednotlivá obrazová data jsou pomocí skriptu nahrána a pomocí myši se označí střed vtisku a poloha všech čtyř vrcholů vtisku. Výsledná hodnota tvrdosti podle Vickerse, délka úhlopříček a obrázek označeného vtisku jsou uloženy do vytvořené složky s výsledky. Obrázek označeného vtisku z obrázku 6.2 je zobrazen na obrázku 6.3

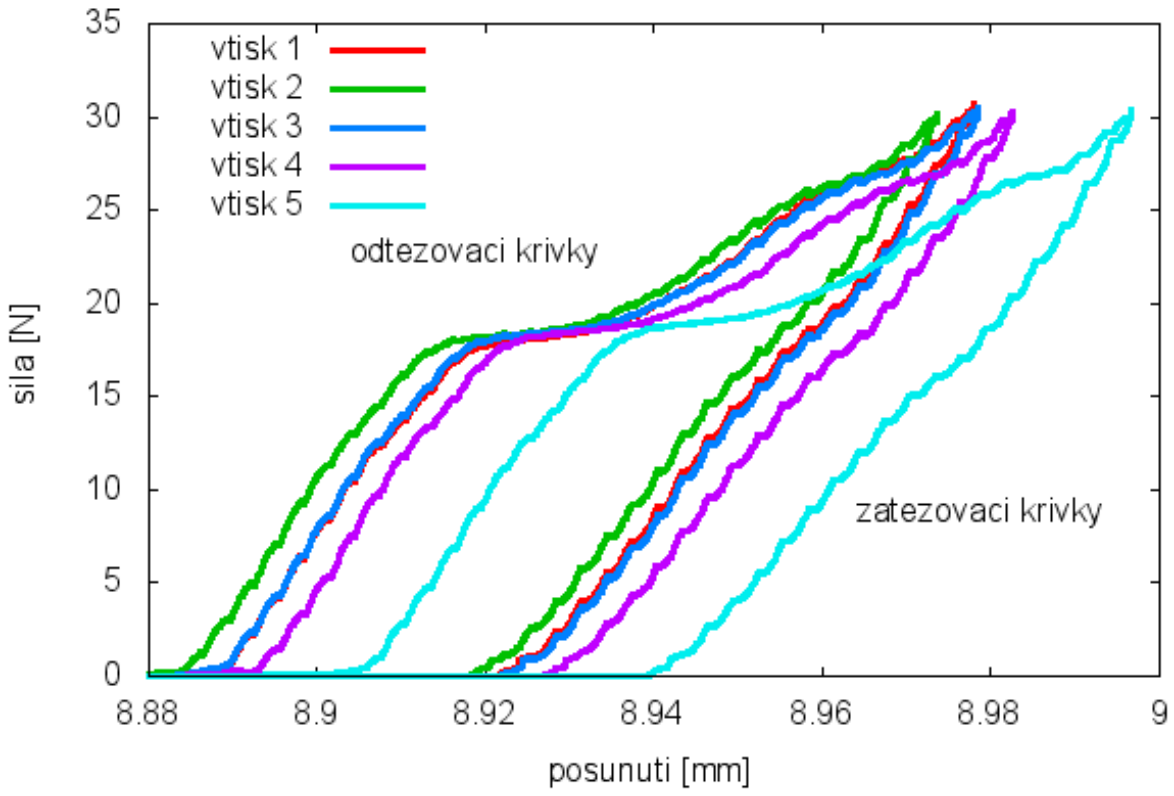
Pro správné vyhodnocení délky úhlopříček a tím i tvrdosti je nutné znát velikost jednoho pixelu obrazových dat v mikrometrech. Díky tomu, že ostření kamery probíhá pouze posunem polohovacího stolku s mikrometrickým šroubem je možné pořídit obrazová data s vyznačeným jedním milimetrem (viz obrázek 6.4) a vyhodnotit pomocí funkce `pixelsize.m` (viz příloha B.2.1) kolik mikrometrů odpovídá jednomu pixelu. Při tomto měření 1 px odpovídá 0,408 μm.

6.3.2 Vyhodnocení měření pomocí metody Olivera a Pharra

Vyhodnocení indentační zkoušky probíhalo pouze pomocí obrazových dat a ne ze záznamu síly a posunutí metodou Olivera a Pharra z důvodu nepřesného polohování aktuátoru indentační osy po přepólování pohybu do opačného směru. Průběh křivek zatěžování a odtěžování jednotlivých vtisků je zobrazen na obrázku 6.5. Z tohoto obrázku je zřejmé, že chyba odtěžovacích křivek oproti teoretickému tvaru zobrazenému na obrázku 3.1 je natolik



Obrázek 6.4: Zobrazení jednoho milimetru na měrce k výpočtu počtu mikrometrů na jeden pixel.



Obrázek 6.5: Zatěžovací a odtěžovací křivky jednotlivých vtisků indentoru.

velká, že vyhodnocování odtěžovací křivky metodou Olivera a Pharra není možné.

Kapitola 7

Ověření přesnosti tvrdoměru Vickers

Pro kontrolu indentoru bylo provedeno měření a jeho vyhodnocení podle postupu popsaného v [9] podle normy pro ověřování tvrdoměrů Vickers [10]. Toto měření se skládá ze dvou částí:

- měření spolehlivosti tvrdoměru (viz část 7.1) a
- měření chyby tvrdoměru (viz část7.2).

Pro tato měření se na tvrdoměrné destičce provede série pěti vtisků.

7.1 Spolehlivost tvrdoměru

Spolehlivost tvrdoměru se určuje z rozdílu nejvyšší a nejnižší střední hodnoty délek úhlopříček na vzniklých vtiscích. Na každém z pěti vtisků se změří délky úhlopříček vtisku u_1 a u_2 a spočítá se jejich aritmetický průměr d_{u_i}. Tyto hodnoty se podle vzrůstající délky seřadí d_{u_1} až d_{u_5} a spolehlivost tvrdoměru je poté dána rozdílem nejdelší a nejkratší průměrné délky úhlopříček $d_{rozdil} = d_{u_5} - d_{u_1}$. Pro indentaci nízkým zatížením a tvrdoměrnou destičku o hodnotě tvrdosti 299,2 HV 3 je dáno, že rozdíl nejdelší a nejkratší délky úhlopříček musí být menší nebo roven $0,03 \cdot d_{str}$, kde d_{str} je roven:

$$d_{str} = \frac{1}{5} \sum_{i=1}^{5} d_{u_i}. \tag{7.1}$$

Indentace byla řízena silou a indentační síla byla nastavena na 30 N.

7.1.1 Výsledky kontroly spolehlivosti tvrdoměru

Na sérii pěti vtisků byla provedena indentace tak, jak je popsáno v části 7.1. Délky úhlopříček jednotlivých vtisků a jejich aritmetický průměr jsou seřazeny v tabulce 7.1 od nejkratší po nejdelší.

u_1 [mm]	u_2 [mm]	d_u [mm]	HV
0,13693	0,13500	0,13597	306,87
0,13853	0,13428	0,13640	304,90
0,13769	0,13598	0,13683	303,00
0,13706	0,13717	0,13712	301,73
0,13941	0,13589	0,13765	299,43

Tabulka 7.1: Délky jednotlivých úhlopříček a jejich aritmetický průměr seřazený od nejmenšího k největšímu a hodnoty odpovídající tvrdosti podle Vickerse.

Rozdíl nejdelšího a nejkratšího aritmetického průměru délek úhlopříček je:

$$d_{rozdil} = d_5 - d_1 = 0,13765 - 0,13597 = 0,00168, \tag{7.2}$$

kde d_5 je nejdelší aritmetický průměr délek úhlopříček a d_1 je nejkratší aritmetický průměr délek úhlopříček. Pro tvrdoměrnou destičku o hodnotě tvrdosti 299,2 HV 3 platí, že d_{rozdil} musí být menší nebo roven $0,03 \cdot d_{str}$, kde po dosazení do rovnice 7.1 je d_{str}:

$$d_{str} = \frac{0,13597 + 0,13640 + 0,13683 + 0,13712 + 0,13765}{5} = 0,13679. \tag{7.3}$$

Pro dostatečnou spolehlivost tvrdoměru musí být d_{rozdil} menší nebo roven spolehlivosti $0,03 \cdot d_{str}$:

$$spolehlivost = 0,03 \cdot d_{str} = 0,03 \cdot 0,13679 = 0,00410 \tag{7.4}$$

Spolehlivost tvrdoměru je dostatečná, protože platí že:

$$\begin{aligned} d_{rozdil} &\leqslant spolehlivost \\ 0,00168 &\leqslant 0,00410. \end{aligned} \tag{7.5}$$

7.2 Chyba tvrdoměru

Druhým způsobem kontroly indentoru je měření chyby tvrdoměru. V tomto případě se vyhodnotí z průměrných délek úhlopříček tvrdost podle Vickerse podle vzorce 2.4. Maximální přípustná chyba tvrdoměru vyjádřená v procentech pro tvrdoměrnou destičku o tvrdosti 299,2 HV 3 je 3 %. Chyba v procentech tvrdoměru se vypočte jako:

$$chyba = \frac{(H_{str} - H) \cdot 100}{H}, \tag{7.6}$$

kde H je tvrdost tvrdoměrné destičky a H_{str} je dána jako:

$$H_{str} = \frac{1}{5} \sum_{i=1}^{5} H_i, \tag{7.7}$$

kde H_i jsou vypočtené hodnoty tvrdosti podle Vickerse jednotlivých vtisků.

7.2.1 Výsledky kontroly chyby tvrdoměru

Vypočtené hodnoty tvrdosti podle Vickerse jednotlivých vtisků jsou shrnuty v tabulce 7.1. Hodnota H_{str} po dosazení do vzorce 7.7 je tedy:

$$H_{str} = \frac{306,87 + 304,90 + 303,00 + 301,73 + 299,43}{5} = 303,19. \tag{7.8}$$

Po dosazení do vzorce 7.6 se stanoví chyba tvrdoměru na:

$$chyba = \frac{(303,19 - 299,2) * 100}{299,2} = 1,332. \tag{7.9}$$

Chyba tvrdoměru 1,332 % je menší než přípustné 3 %.

7.3 Shrnutí výsledků kontroly instrumentovaného indentoru

Instrumentovaný indentor splňuje podmínky ověřování tvrdoměrů Vickers pro indentaci nízkým zatížením 30 N. Spolehlivost indentoru byla stanovena na 0,00168, kdy tato hodnota měla být podle normy pro ověřování tvrdoměrů Vickers menší než 0,00410. Vypočtená chyba tvrdoměru byla 1,332 %, kdy normou pro ověřování tvrdoměrů podle Vickerse je tato hodnota stanovena na 3 % [10].

Pro ověření instrumentovaného indentoru na mikroindentaci (zkušební zatížení 0,01 až 0,2 kp) by bylo nutné pořídit další tvrdoměrnou destičku určenou pro tato zatížení. Pořízení další tvrdoměrné destičky ovšem nebylo z finančních důvodů možné.

7.4 Možná vylepšení instrumentovaného indentoru

Pro správné určení tvrdosti povrchu vzorku je zásadní správné změření délek úhlopříček, což vyžaduje kvalitní obrazová data jednotlivých vtisků indentoru. Docílení lepších obrazových dat je možné kruhovým osvětlením a ne doposud používaným dvouramenným světlovodem studeného světla. Zlepšení obrazových dat je také možné docílit použitím specializovaného objektivu. Pro přesné určení délky úhlopříček je také možné pořízení obrazových dat jednotlivých vtisků pomocí řádkovací elektronové mikroskopie. Osazení instrumentovaného indentoru kamerou a objektivem ovšem umožňuje ucelené řešení bez nutnosti přenášení vzorku a úpravu pro elektronovou mikroskopii.

Pro umožnění vyhodnocení dat metodou Olivera a Pharra je nezbytné přesné vyčítání posunu indentační osy. To je možné buď přidáním vyčítání polohy indentační osy opticky nebo magneticky. Ideálním řešením by ovšem bylo nahrazení aktuátoru indentační osy za piezo posuv, který je ovšem velmi finančně nákladný.

Kapitola 8

Závěr

Cílem předkládané práce bylo vytvořit funkční zařízení schopné indentace pomocí hrotu Vickersova typu při nízkém zatížení, nebo mikroindentace. Za tímto účelem bylo navrženo a zkonstruováno indentační zařízení se dvěma módy zatěžování. Prvním módem je indentace konstantní silou, kdy je indentor osazen závažím. Druhým módem je indentace řízená silou, kdy je indentor osazen siloměrem. Pro oba módy zatěžování bylo navrženo vlastní grafické uživatelské rozhraní umožňující ovládání indentoru, optické soustavy pro pořizování obrazových dat a tvorbě indentačních programů (tvorba mřížky vtisků na ploše, nebo vytvoření pouze jednoho vtisku v daném místě vybraném z obrazu kamery). Řízení indentoru bylo realizováno pomocí modulu LinuxCNC, který je určen pro počítačové řízení v reálném čase technologií CNC.

Pro kontrolu přesnosti indentoru bylo provedeno měření spolehlivosti a určení chyby indentoru pomocí referenční tvrdoměrné destičky. Na referenční tvrdoměrné destičce byla provedena série pěti vtisků, které byly vyhodnoceny z obrazových dat jednotlivých vtisků pomocí skriptu v programovacím prostředí Matlab. Data z indentační zkoušky nebyla vyhodnocována pomocí metody Olivera a Pharra z důvodu nepřesnosti polohování indentační osy, a tím nedostatečně přesnému průběhu odtěžovací křivky.

Spolehlivost indentoru je určena z rozdílu délek úhlopříček jednotlivých vtisků a pro tvrdoměrnou destičku o jmenovité tvrdosti 299,2 HV 3 platí, že spolehlivost by měla být menší nebo rovna hodnotě 0,00410 a pro instrumentovaný indentor vyšla hodnota spolehlivosti 0,00168. Platí tedy že, $0,00168 < 0,00410$. Chyba tvrdoměru je určena z chyby hodnot tvrdosti jednotlivých vtisků a pro indentaci referenční tvrdoměrné destičky o jmenovité tvrdosti 299,2 HV 3 by neměla být chyba větší než 3 %. Chyba tvrdosti indentoru

byla stanovena na 1,332 %, platí tedy, že $1,332 < 3$. Indentor je tedy plně schopen indentace při nízkém zatížení a vyhovuje požadavkům norem na laboratorní zařízení.

Přesnost měření je závislá zejména na kvalitě obrazových dat, proto je do budoucna plánováno vylepšení optické soustavy indentoru o kruhové osvětlení a nový objektiv. Umožnění vyhodnocování dat z indentačního měření metodou Olivera a Pharra by bylo možné po zpřesnění vyčítání polohy indentační osy, což by bylo možné optickým, nebo magnetickým vyčítáním polohy. Ideálním řešením by bylo opatření indentační osy piezo aktuátorem, toto řešení je ovšem velmi finančně nákladné.

V rámci této práce vznikl instrumentovaný indentor umožňující plně funkční a automatizovanou indentaci metodou dle Vickerse. Zařízení bylo v roce 2014 zaregistrováno jako funkční vzorek [35].

Literatura

[1] Pavel Doležal a Bohumil Pacal. Hodnocení mikrotvrdosti struktur materiálů. http://ime.fme.vutbr.cz/images/umvi/opory/hmsm/index.htm

[2] F.Crace Calvert a Richard Johnson. On the hardness of metals and alloys. *Journal of the Franklin Institute*, 67(3):198 – 203, 1859. http://www.sciencedirect.com/science/article/pii/0016003259902649

[3] J.A. Brinell. Way of determining the hardness of bodies and some applications of the same. *Teknisk Tidskrift*, (5), 1900.

[4] *ČSN EN ISO 6506: Kovové materiály - Zkouška tvrdosti podle Brinella*, 2000.

[5] E. Meyer. Investigations of hardness testing and hardness. (9), 1908.

[6] Ian M. Hutchings. The contributions of david tabor to the science of indentation hardness. *Journal of Materials Research*, 24(3):581–589, 2009.

[7] *ČSN EN ISO 6508: Kovové materiály - Zkouška tvrdosti podle Rockwella*, 2000.

[8] R. Smith a Sandland. An accurate method of determining the hardness of metals, with particular reference to those of a high degree of hardness. *Proc. Instn. Mech. Engrs.*, 1(623), 1922.

[9] Jan Ludvík, Karel Bílek a Štěpán Ludvík. Zkoušky tvrdosti, *Metrotest Kladno*, 2010. http://www.metrotest.cz/files/zkousky_tvrdosti.pdf

[10] *ČSN EN ISO 6507: Kovové materiály - Zkouška tvrdosti podle Vickerse*, 2000.

[11] Václav Machek a Jaromír Sodomka. *Vlastnosti kovových materiálů*. Nakladatelství ČVUT, Praha, 1 edition, 2007.

[12] *ČSN ISO 4545: Kovové materiály - Zkouška tvrdosti podle Knoopa*, 2000.

[13] Min Hao Wong. The development of scratch test methodol ogy and characterization of surface damage of polypropylene. Texas A&M University, 2003. http://repository.tamu.edu/bitstream/handle/1969.1/1194/etd-tamu-2003B-2003070711-MINH-1.pdf?sequence=1

[14] Brian J. Briscoe, Paul D. Evans, Enrico Pellilo a Sujeet K. Sinha. Scratching maps for polymers. *Wear*, 200(12):137 – 147, 1996. http://www.sciencedirect.com/science/article/pii/S0043164896073140

[15] C. Gauthier a R. Schirrer. Time and temperature dependence of the scratch properties of poly(methylmethacrylate) surfaces. *Journal of Materials Science*, 35(9):2121–2130, 2000.

[16] RoseA. Ryntz a Dottie Britz. Scratch resistance behavior of automotive plastic coatings. *Journal of Coatings Technology*, 74(925):77–81, 2002.

[17] Jian Sun, Harold Mukamal, Zhiqiang Liu a Weidian Shen. Analysis of the taber test in characterization of automotive side windows. *Tribology Letters*, 13(1):49–54, 2002.

[18] W. Verwaal a A. Mulder. Estimating rock strength with the equotip hardness tester. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 30(6):659 – 662, 1993. http://www.sciencedirect.com/science/article/pii/0148906293912269

[19] Tvrdoměr poldi. *Strojírny Poldi spol. s.r.o.* Kladno. http://www.strojpoldi.cz/web-galerie-soubory/download-katalogy/Poldi_kladivko_cesky.pdf

[20] Mark R. VanLandingham. Review of instrumented indentation. *Journal of Research of the National Institute of Standards and Technology*, 108(4):249 – 265, 2003. http://nvlpubs.nist.gov/nistpubs/jres/108/4/j84vanl.pdf

[21] Donna M. Ebenstein a Lisa A. Pruitt. Nanoindentation of biological materials. *Nano Today*, 1(3):26 – 33, 2006. http://www.sciencedirect.com/science/article/pii/S1748013206700779

[22] Daniel Kytýř, Nela Fenclová, Petr Koudelka, Tomáš Doktor, Josef Šepitka a Jaroslav Lukeš. Mapping of local changes of mechanical properties in trabecular interconnections. *Key Engineering Materials*, in press.

[23] H. Lu, B. Wang, J. Ma, G. Huang a H. Viswanathan. Measurement of creep compliance of solid polymers by nanoindentation. *Mechanics of Time-Dependent Materials*, 7(3-4):189–207, 2003.

[24] T. Ohmura, A.M. Minor, E.A. Stach a J.W. Morris. Dislocation – grain boundary interactions in martensitic steel observed through in situ nanoindentation in a transmission electron microscope. *Journal of Materials Research*, 19:3626–3632, 2004. http://journals.cambridge.org/article_S0884291400087719

[25] Tim Burgess a M. Ferry. Nanoindentation of metallic glasses. *Materials Today*, 12:24 – 32, 2009. http://www.sciencedirect.com/science/article/pii/S1369702109700392

[26] S J Bull. Nanoindentation of coatings. *Journal of Physics D: Applied Physics*, 38(24):R393, 2005. http://stacks.iop.org/0022-3727/38/i=24/a=R01

[27] J. G. Swadener, B. Taljat a G.M. Pharr. Measurement of residual stress by load and depth sensing indentation with spherical indenters. *Journal of Materials Research*, 16:2091–2102, 2001. http://journals.cambridge.org/article_S088429140006893X

[28] Ian N. Sneddon. The relation between load and penetration in the axisymmetric boussinesq problem for a punch of arbitrary profile. *International Journal of Engineering Science*, 3:47–57, 1965.

[29] W.C. Oliver a G.M. Pharr. An improved technique for determining hardness and elastic modulus using load and displacement sensing indentation experiments. *Journal of Materials Research*, 7(06):1564–1583, 1992.

[30] J. Hay. Introduction to instrumented indentation testing. *Experimental Techniques*, 33(6):66–72, 2009.

[31] Nela Fenclová. *Bakalářská práce: Poloautomatické mapování tvrdosti povrchů materiálů*. Praha, 2013.

[32] Ini configuration. *LinuxCNC Documentation*
http://www.linuxcnc.org/docs/html/config/ini_config.html#sub:HALUI-section

[33] G codes. *LinuxCNC Documentation*
http://linuxcnc.org/docs/html/gcode/gcode.html

[34] Hal introduction. *LinuxCNC Documentation*
http://linuxcnc.org/docs/html/hal/intro.html

[35] T. Fíla, D. Kytýř a N. Fenclová. Instrumentovaný mikroindentor [Funkční vzorek], 2014.

Příloha A

Zdrojové kódy pro řízení indentoru

A.1 INI soubory

A.1.1 microindentor_simple.ini

```
# Generated by stepconf at Mon Jul  1 17:13:37 2013
# If you make changes to this file, they will be
# overwritten when you run stepconf again

[EMC]
MACHINE = microindentor_simple
DEBUG = 0

[DISPLAY]
DISPLAY = axis
EDITOR = gedit
POSITION_OFFSET = RELATIVE
POSITION_FEEDBACK = ACTUAL
MAX_FEED_OVERRIDE = 1.2
INTRO_GRAPHIC = linuxcnc.gif
INTRO_TIME = 0.5
PROGRAM_PREFIX = ./ngc_files
INCREMENTS = 5mm 1mm .5mm .1mm .05mm .01mm .005mm
```

```
GLADEVCP = ./machines/microindentor_simple/microindentor_simple.
    ↪ ui

[FILTER]
PROGRAM_EXTENSION = .png,.gif,.jpg Greyscale Depth Image
PROGRAM_EXTENSION = .py Python Script
png = image-to-gcode
gif = image-to-gcode
jpg = image-to-gcode
py = python

[TASK]
TASK = milltask
CYCLE_TIME = 0.010

[RS274NGC]
PARAMETER_FILE = linuxcnc.var
USER_M_PATH = ./m_files

[EMCMOT]
EMCMOT = motmod
COMM_TIMEOUT = 1.0
COMM_WAIT = 0.010
BASE_PERIOD = 100000
SERVO_PERIOD = 1000000

[HAL]
HALFILE = ./machines/microindentor_simple/microindentor_simple.
    ↪ hal
HALFILE = ./machines/microindentor_simple/loadcell.hal
HALUI=halui
POSTGUI_HALFILE = ./machines/microindentor_simple/custom_postgui.
    ↪ hal
```

```
[HALUI]
# mdi command 00 - clear logfile
MDLCOMMAND = M125
# mdi command 01 - export logfile
MDLCOMMAND = M124
# mdi command 02 save original position
MDLCOMMAND = G30.1
# mdi command 03 restore original position button (originally G30
   ↪ )
MDLCOMMAND = o<microindentor_simple_retrieve_position> call
# mdi command 04 close camera
MDLCOMMAND = M128
# mdi command 05 one indent
MDLCOMMAND = o<one_indent_simple> call
# mdi command 06 show indent
MDLCOMMAND = o<show_indent> call
# mdi command 07 change relative coordinates to absolute
   ↪ coordinates when you press stop button
MDLCOMMAND = G90

[TRAJ]
AXES = 3
COORDINATES = X Y Z
MAX_ANGULAR_VELOCITY = 1000
DEFAULT_ANGULAR_VELOCITY = 6.00
LINEAR_UNITS = mm
ANGULAR_UNITS = degree
CYCLE_TIME = 0.010
DEFAULT_VELOCITY = 1
MAX_LINEAR_VELOCITY = 5

[EMCIO]
```

```
EMCIO = io
CYCLE_TIME = 0.100
TOOL_TABLE = tool.tbl

[AXIS_0]
TYPE = LINEAR
HOME = 0.0
MAX_VELOCITY = 5
MAX_ACCELERATION = 30.0
STEPGEN_MAXACCEL = 37.5
SCALE = 2000.0
FERROR = 1
MIN_FERROR = .25
MIN_LIMIT = -0.001
MAX_LIMIT = 133
HOME_SEQUENCE = 1
HOME_OFFSET = -0.5
HOME_SEARCH_VEL = -5
HOME_LATCH_VEL = 0.25
USE_INDEX = NO
HOME_IGNORE_LIMITS = YES

[AXIS_1]
TYPE = LINEAR
HOME = 0.0
MAX_VELOCITY = 5
MAX_ACCELERATION = 30.0
STEPGEN_MAXACCEL = 37.5
SCALE = 2000.0
FERROR = 1
MIN_FERROR = .25
MIN_LIMIT = -0.001
MAX_LIMIT = 30
```

```
HOME_SEQUENCE = 1
HOME_OFFSET = -0.5
HOME_SEARCH_VEL = -5
HOME_LATCH_VEL = 0.25
USE_INDEX = NO
HOME_IGNORE_LIMITS = YES

[AXIS_2]
TYPE = LINEAR
HOME = 0.0
MAX_VELOCITY = 2
MAX_ACCELERATION = 30
STEPGEN_MAXACCEL = 37.5
SCALE = -3333.333333
FERROR = 1
MIN_FERROR = .25
MIN_LIMIT = -0.001
MAX_LIMIT = 33
HOME_SEQUENCE = 0
HOME_OFFSET = -0.5
HOME_SEARCH_VEL = -2
HOME_LATCH_VEL = 0.25
USE_INDEX = NO
HOME_IGNORE_LIMITS = YES

[AXIS_3]
#TYPE = LINEAR
#HOME = 0.0
#MAX_VELOCITY = 100
#MAX_ACCELERATION = 5000
#STEPGEN_MAXACCEL = 5500
#SCALE = 100
#FERROR = 1
```

```
#MIN_FERROR = .25
#HOME_OFFSET = 22.0
#HOME_SEARCH_VEL=70
#HOME_LATCH_VEL=7
#USE_INDEX=NO
```

A.1.2 microindentor_instrumented.ini

```
# Generated by stepconf at Mon Jul  1 17:13:37 2013
# If you make changes to this file, they will be
# overwritten when you run stepconf again

[EMC]
MACHINE = microindentor_instrumented
DEBUG = 0

[DISPLAY]
DISPLAY = axis
EDITOR = gedit
POSITION_OFFSET = RELATIVE
POSITION_FEEDBACK = ACTUAL
MAX_FEED_OVERRIDE = 1.2
INTRO_GRAPHIC = linuxcnc.gif
INTRO_TIME = 0.5
PROGRAM_PREFIX = ./ngc_files
INCREMENTS = 5mm 1mm .5mm .1mm .05mm .01mm .005mm
GLADEVCP = ./machines/microindentor_instrumented/
    ↪ microindentor_instrumented.ui

[FILTER]
PROGRAM_EXTENSION = .png,.gif,.jpg Greyscale Depth Image
PROGRAM_EXTENSION = .py Python Script
png = image-to-gcode
```

```
gif = image-to-gcode
jpg = image-to-gcode
py = python

[TASK]
TASK = milltask
CYCLE_TIME = 0.010

[RS274NGC]
PARAMETER_FILE = linuxcnc.var
USER_M_PATH = ./m_files

[EMCMOT]
EMCMOT = motmod
COMM_TIMEOUT = 1.0
COMM_WAIT = 0.010
BASE_PERIOD = 100000
SERVO_PERIOD = 1000000

[HAL]
HALFILE = ./machines/microindentor_instrumented/
    ↪ microindentor_instrumented.hal
HALFILE = ./machines/microindentor_instrumented/loadcell.hal
HALUI=halui
POSTGUI_HALFILE = ./machines/microindentor_instrumented/
    ↪ custom_postgui.hal

[HALUI]
# mdi command 00 - clear logfile
MDI_COMMAND = M125
# mdi command 01 - export logfile
MDI_COMMAND = M124
# mdi command 02 save original position
```

```
MDLCOMMAND = G30.1
# mdi command 03 restore original position button (originally G30
    ↪ )
MDLCOMMAND = o<microindentor_simple_retrieve_position> call
# mdi command 04 testing indent
MDLCOMMAND = o<microindentor_instrumented_testing_indent> call
# mdi command 05 change relative coordinates to absolute
    ↪ coordinates when you press stop button
MDLCOMMAND = G90
# mdi command 06 show indent
MDLCOMMAND = o<show_indent> call
# mdi command 07 one indent
MDLCOMMAND = o<one_indent> call
# mdi command 08 close camera
MDLCOMMAND = M128

[TRAJ]
AXES = 3
COORDINATES = X Y Z
MAX_ANGULAR_VELOCITY = 1000
DEFAULT_ANGULAR_VELOCITY = 6.00
LINEAR_UNITS = mm
ANGULAR_UNITS = degree
CYCLE_TIME = 0.010
DEFAULT_VELOCITY = 1
MAX_LINEAR_VELOCITY = 5

[EMCIO]
EMCIO = io
CYCLE_TIME = 0.100
TOOL_TABLE = tool.tbl

[AXIS_0]
```

```
TYPE = LINEAR
HOME = 0.0
MAX_VELOCITY = 5
MAX_ACCELERATION = 30.0
STEPGEN_MAXACCEL = 37.5
SCALE = 2000.0
FERROR = 1
MIN_FERROR = .25
MIN_LIMIT = -0.001
MAX_LIMIT = 133
HOME_SEQUENCE = 1
HOME_OFFSET = -0.5
HOME_SEARCH_VEL = -5
HOME_LATCH_VEL = 0.25
USE_INDEX = NO
HOME_IGNORE_LIMITS = YES

[AXIS_1]
TYPE = LINEAR
HOME = 0.0
MAX_VELOCITY = 5
MAX_ACCELERATION = 30.0
STEPGEN_MAXACCEL = 37.5
SCALE = 2000.0
FERROR = 1
MIN_FERROR = .25
MIN_LIMIT = -0.001
MAX_LIMIT = 30
HOME_SEQUENCE = 1
HOME_OFFSET = -0.5
HOME_SEARCH_VEL = -5
HOME_LATCH_VEL = 0.25
USE_INDEX = NO
```

```
HOME_IGNORE_LIMITS = YES

[AXIS_2]
TYPE = LINEAR
HOME = 0.0
MAX_VELOCITY = 2
MAX_ACCELERATION = 30
STEPGEN_MAXACCEL = 37.5
SCALE = -3333.333333
FERROR = 1
MIN_FERROR = .25
MIN_LIMIT = -0.001
MAX_LIMIT = 33
HOME_SEQUENCE = 0
HOME_OFFSET = -0.5
HOME_SEARCH_VEL = -2
HOME_LATCH_VEL = 0.25
USE_INDEX = NO
HOME_IGNORE_LIMITS = YES

[AXIS_3]
#TYPE = LINEAR
#HOME = 0.0
#MAX_VELOCITY = 100
#MAX_ACCELERATION = 5000
#STEPGEN_MAXACCEL = 5500
#SCALE = 100
#FERROR = 1
#MIN_FERROR = .25
#HOME_OFFSET = 22.0
#HOME_SEARCH_VEL=70
#HOME_LATCH_VEL=7
#USE_INDEX=NO
```

A.2 NGC soubory

A.2.1 microindentor_simple.ngc

```
#5399=1 (initial variable declaration)
M66 E1 L0 (read motion analog in 0 signal - x grid)
#1=#5399
M66 E2 L0 (read motion analog in 1 signal - y grid)
#2=#5399
M66 E3 L0 (read motion analog in 2 signal - x start coordinate)
#3=#5399
M66 E4 L0 (read motion analog in 3 signal - y start coordinate)
#4=#5399
M66 E5 L0 (read motion analog in 4 signal - x end coordinate)
#5=#5399
M66 E6 L0 (read motion analog in 5 signal - y end coordinate)
#6=#5399
M125 (clear logfile)
o101 if [#1 EQ 1] (if x grid equal 1)
        G0 Z0
        G0 X[#3] Y[#4]
        o102 if [#2 GT 1] (if y grid grater 1)
                #8=[[#6-#4]/[#2-1]] (calculate y increment)
                G91 (Incremental mode)
                o103 repeat [#2] (repeat according to y grid )
                o<indentation_simple> call (calling of o code for
                   ↪ indentation)
                G1 F50 Y[#8] (y increment move with speed 50)
                o103 endrepeat
                G90 (Absolute mode)
        o102 elseif [#2 EQ 1]   (if x grid=1 and y grid=1)
```

```
                o<indentation_simple> call (calling of o code for
                   ↪ indentation)
        o102 endif
G90
o101 elseif [#2 EQ 1] (else if y grid equal 1)
        G0 Z0
        G0 X[#3] Y[#4]  (fast move to x start and y start)
        o<indentation_simple> call      (calling of o code for
           ↪ indentation)
        o104 if [#1 GT 1] (if x grid grater 1)
                #7=[[#5-#3]/[#1-1]] (calculate x increment)
                G91 (Incremental mode)
                o105 repeat [#1-1]      (repeat number of x grid
                   ↪ -1)
                G1 F50 X[#7]  (x increment move with speed 50)
                o<indentation_simple> call   (calling of o code
                   ↪ for indentation)
                o105 endrepeat
                G90 (Absolute mode)
        o104 endif
G90
o101 else (else)
        #7=[[#5-#3]/[#1-1]] (calculate x increment)
        #8=[[#6-#4]/[#2-1]] (calculate y increment)
        G90     (absolute mode)
        (fast move to z=0)
        G0 Z0
        G0 X[#3] Y[#4]  (fast move to x start and y start)
        G91     (incremental mode)
        o106 repeat [#2]        (repat number of y grid)
        o<indentation_simple> call      (calling of o code for
           ↪ indentation)
```

```
                o107 repeat [#1-1]      (repat number of x grid
                   ↪ -1)
                G91                     (incremental mode)
                G1 F50 X[#7]            (x increment move with
                   ↪ speed 50)
                o<indentation_simple> call      (calling of o
                   ↪ code for indentation)
                o107 endrepeat
        G90             (absolute mode)
        G1 F50 X[#3]    (move to x start with speed 50)
        #9=#5421
        o108 if [#9 LT #6]
                G91             (incremental mode)
                G1 F50 Y[#8]    (y increment move with speed 50)
        o108 else
        G1 F50 Y[#4]
        o108 endif
        o106 endrepeat
G90             (absolute mode)
o101 endif
M124            (export logfile)
G0 Z0
G90
G0 X[#3-#5168] Y[#4+#5169]
#10=#5420       (current x)
#11=#5421       (current y)
o111 if [#1 EQ 1]
         o112 if [#2 GT 1]
                M126
                (make picture)
                o113 repeat [#2-1]
                        G91
                        G1 F50 Y[#36]
```

```
                        M126
                        (make picture)
                o113 endrepeat
                G90
        o112 elseif [#2 EQ 1]
                M126
                (make picture)
        o112 endif
G90
o111 elseif [#2 EQ 1]
        o114 if [#1 GT 1]
                M126
                (make picture)
                o115 repeat [#1-1]
                        G91
                        G1 F50 X[#35]
                        M126
                        (make picture)
                o115 endrepeat
        o114 endif
G90
o111 else
        o116 repeat [#2]
                M126
                (make picture)
                o117 repeat [#1-1]
                        G91
                        G1 F50 X[#35]
                        M126
                        (make picture)
                o117 endrepeat
                #12=#5421
                o118 if [#12 LT #6+#5169]
```

```
                        G90
                        G1 F50 X#10
                        G91                 (incremental mode)
                        G1 F50 Y[#36]       (y increment move with
                           ↪ speed 50)
                o118 endif
        o116 endrepeat
o111 endif
M2
```

A.2.2 indentation_simple.ngc

```
o<indentation_simple> sub
G90      (absolute mode)
(move with speed 25 until loss of contact)
G38.5 F25 Z32.9
(relative coordinates)
G91
(move with speed 25 2 more mm)
G1 F25 Z2
(dwell time for indentation 10s)
G4 P10
(absolute coordinates)
G90
(move with speed 25 and stop on contact)
G38.3 F25 Z0.1
(relative coordinates)
G91
(move with full speed -5mm)
G0 Z-5

o<indentation_simple> endsub
M2
```

A.2.3 microindentor_instrumented.ngc

```
#5399=1 (initial variable declaration)

M66 E1 L0 (read motion analog in 0 signal - x grid)
#1=#5399

M66 E2 L0 (read motion analog in 1 signal - y grid)
#2=#5399

M66 E3 L0 (read motion analog in 2 signal - x start coordinate)
#3=#5399

M66 E4 L0 (read motion analog in 3 signal - y start coordinate)
#4=#5399

M66 E5 L0 (read motion analog in 4 signal - x end coordinate)
#5=#5399

M66 E6 L0 (read motion analog in 5 signal - y end coordinate)
#6=#5399

M125 (clear logfile)

M123 (live plot)

o101 if [#1 EQ 1] (if x grid equal 1)
        G0 Z0

        G0 X[#3] Y[#4]
        o102 if [#2 GT 1] (if y grid grater 1)
```

```
                #36=[[#6-#4]/[#2-1]] (calculate y increment)
                G91 (Incremental mode)
                o103 repeat [#2] (repeat according to y grid )
                o<indentation_instrumented> call (calling of o
                   ↪ code for indentation)
                G91
                G1 F50 Y[#36] (y increment move with speed 50)
                o103 endrepeat
                G90 (Absolute mode)
        o102 elseif [#2 EQ 1]    (if x grid=1 and y grid=1)
                o<indentation_instrumented> call (calling of o
                   ↪ code for indentation)
        o102 endif
G90

o101 elseif [#2 EQ 1] (else if y grid equal 1)
        G0 Z0

        G0 X[#3] Y[#4]  (fast move to x start and y start)
        o<indentation_instrumented> call        (calling of o
           ↪ code for indentation)
        o104 if [#1 GT 1] (if x grid grater 1)
                #35=[[#5-#3]/[#1-1]] (calculate x increment)
                G91 (Incremental mode)
                o105 repeat [#1-1]      (repeat number of x grid
                   ↪ -1)
                G1 F50 X[#35]  (x increment move with speed 50)
                o<indentation_instrumented> call   (calling of o
                   ↪ code for indentation)
                o105 endrepeat
                G90 (Absolute mode)
        o104 endif
G90
```

```
o101 else (else)
        #35=[[#5-#3]/[#1-1]] (calculate x increment)
        #36=[[#6-#4]/[#2-1]] (calculate y increment)
        G90     (absolute mode)
        (fast move to z=0)
        G0 Z0
        G0 X[#3] Y[#4]  (fast move to x start and y start)
        G91     (incremental mode)
        o106 repeat [#2]        (repat number of y grid)
        o<indentation_instrumented> call        (calling of o
           ↪ code for indentation)
                o107 repeat [#1-1]      (repat number of x grid
                   ↪ -1)
                G91                     (incremental mode)
                G1 F50 X[#35]           (x increment move with
                   ↪ speed 50)
                o<indentation_instrumented> call        (calling
                   ↪ of o code for indentation)
                o107 endrepeat
        G90             (absolute mode)
        G1 F50 X[#3]    (move to x start with speed 50)
        #9=#5421
        o108 if [#9 LT #6]
                G91             (incremental mode)
                G1 F50 Y[#36]   (y increment move with speed 50)
        o108 else
        G90
        G1 F50 Y[#4]
        o108 endif
        o106 endrepeat
G90             (absolute mode)
```

```
o101 endif

M124            (export logfile)

G0 Z0

G90
G0 X[#3-#5168] Y[#4+#5169]
#10=#5420       (current x)
#11=#5421       (current y)

o111 if [#1 EQ 1]
         o112 if [#2 GT 1]
                 M126
                 (make picture)
                 o113 repeat [#2-1]
                         G91
                         G1 F50 Y[#36]
                         M126
                         (make picture)
                 o113 endrepeat
                 G90
        o112 elseif [#2 EQ 1]
                 M126
                 (make picture)
        o112 endif

G90
o111 elseif [#2 EQ 1]
        o114 if [#1 GT 1]
                 M126
                 (make picture)
                 o115 repeat [#1-1]
```

```
                        G91
                        G1 F50 X[#35]
                        M126
                        (make picture)
                o115 endrepeat
        o114 endif

G90
o111 else

        o116 repeat [#2]
                M126
                (make picture)
                o117 repeat [#1-1]
                        G91
                        G1 F50 X[#35]
                        M126
                        (make picture)
                o117 endrepeat
                #12=#5421
                o118 if [#12 LT #6+#5169]
                        G90
                        G1 F50 X#10
                        G91             (incremental mode)
                        G1 F50 Y[#36]   (y increment move with
                           ↪ speed 50)
                o118 endif
        o116 endrepeat
o111 endif

M2
```

A.2.4 indentation_instrumented.ngc

```
o<indentation_instrumented> sub

(indentation code- indentation with given speed if it's known,
   ↪ otherwise with speed F0.1)

M66 E0 L0 (read the force value from motion analog pin 0)
#1=0
#2=#5399

M66 E7 L0 (read the maximum value of force which we want to make
   ↪ indentation with)
#3=#5399

o100 sub
G90 (absolute coordinate system)
o110 if [#31 GT 0.5] (if the value of contact indenter --
   ↪ specimen surface is known, move to that position minus 0.3)
G1 F50 Z[#31-0.3]

G38.3 F0.5 Z32 (move until force value 0.5 N is reached with
   ↪ lower speed)
(DEBUG, #5063)
G4 P5 (wait 5 sec)

G91 (relative coordinates)

o103 repeat [1]

        o101 while [#1 LT 1]
                o104 if [#33 GT 0] (if the indentation speed is
                   ↪ known move one step)
```

```
                G1 F#33 Z0.0003
            o104 else (move one step with low speed)
                G1 F0.1 Z0.0003
            o104 endif
            #1=[#1+0.0003]
            M66 E0 L0 (read the force value on load cell)

            #2=#5399
            M66 E7 L0 (read the requested indentation force)
            #3=#5399

            o102 if [#2 GE #3] (if the load on load cell is
               ↪ greater stop)

                o101 break

            o102 else

                o101 continue

            o102 endif

        o101 endwhile

o103 endrepeat

o110 else
        G90
        G38.3 F0.5 Z32 (move spowly until reach force 0.5N)
        G91

        o106 repeat [1]
```

```
        o107 while [#1 LT 1]

                o111 if [#33 GT 0] (if the indentation speed is
                   ↪ known move one step)
                        G1 F#33 Z0.0003
                o111 else (move one step with low speed)
                        G1 F0.1 Z0.0003
                o111 endif
                #1=[#1+0.0003]

                M66 E0 L0 (read the force value on load cell)
                #2=#5399
                M66 E7 L0 (read the requested indentation force)
                #3=#5399

                o108 if [#2 GT #3] (if the load on load cell is
                   ↪ greater stop)

                        o107 break

                o108 else

                        o107 continue

                o108 endif

        o107 endwhile

        o106 endrepeat

o110 endif

G90 (absolute coordinates)
```

```
G4 P15 (wait 15 sec)

o109 if [#33 GT 0]
        G38.5 F#33 Z0 (if indentation speed is known move until
           ↪ force of 0.5 N is reached)
o109 else
        G38.5 F0.1 Z0 (else move with low speed until force value
           ↪  0.5 N is reached)
o109 endif
g91 (relative coordinates)

g1 F5 Z-0.5 (move indenter to save position)

o100 endsub

o100 call
G90

o<indentation_instrumented> endsub
M2 (end program)
```

A.2.5 calibration_manta.ngc

```
o<calibration_manta> sub

M66 E3 L0 (read motion analog in 2 signal - x start coordinate)
#1=#5399

M66 E4 L0 (read motion analog in 3 signal - y start coordinate)
#2=#5399

#3=#5420 (current x coordinate)
```

```
#4=#5421 (current y coordinate)

#5=ABS[#1-#3] (x offset)
#6=ABS[#2-#4] (y offset)

#5168=#5          (remember offset x)
#5169=#6          (remember offset y)

o<calibration_manta> endsub
```

A.2.6 testing_indent.ngc

```
o<microindentor_instrumented_testing_indent> sub
M66 E0 L0 (read the force value from motion analog pin 0)

#1=0

#2=#5399

M66 E7 L0 (read the maximum value of force which we want to make
   ↪ indentation with)
#3=#5399

G38.3 F0.5 Z32 (move until force value 0.5 N is reached with
   ↪ lower speed)
#31=#5063 (save current position when the contact indenter --
   ↪ surface is reached)

M125 (clear logfile)

G4 P5 (wait 5 sec)
G91 (relative coordinates)
```

```
o103 repeat [1]

        o101 while [#1 LT 1]

                G1 F0.1 Z0.0003 (move slowly one step)

                #1=[#1+0.0003]

                M66 E0 L0 (read actual force value from load cell
                   ↪ )

                #2=#5399


                o102 if [#2 GT #3] (if actual force value is
                   ↪ bigger than indentation force stop)

                        o101 break

                o102 else

                        o101 continue

                o102 endif

        o101 endwhile

o103 endrepeat

#32=#5422 (save position of the indenter when it is fully loaded)
#33=[[#32-#31]*6] (calculate and save indentation speed)

G90 (absolute coordinates}
```

```
G4 P15 (wait 15 sec)
G38.5 F0.1 Z0 (move until force value on load cell is 0.5N)

M124 (export logfile)

G90 (absolute coordinates)
g0 Z0 (move home with indenter)



o<microindentor_instrumented_testing_indent> endsub
```

A.2.7 retrieve_position.ngc

```
o<microindentor_simple_retrieve_position> sub
G0 Z0            (first move fast to Z0)
G0 X#5181 Y#5182        (than move to retrieve position with X
   ↪ and Y)
G0 Z#5183               (first than move with indentor)
o<microindentor_simple_retrieve_position> endsub
```

A.2.8 one_indent_instrumented.ngc

```
o<one_indent> sub

#1=#5420 (current x position)
#2=#5421 (current y position)

G90 (absolute coordinates)
G0 X[#1+#5168] Y[#2-#5169] (move under the indenter)

o101 if [#33 EQ 0] (unknown speed of indentation)
        o<microindentor_instrumented_testing_indent> call
```

```
o101 else
        M125 (clear logfile)
        o<indentation_instrumented> call
        M124 (export logfile)
o101 endif

G90 (absolute coordinates)
G0 Z0 (move home with indenter)

#3=#5420 (current x position)
#4=#5421 (current y position)
G0 X[#3-#5168] Y[#4+#5169] (move under camera)

M126 (make picture of the indenter)
o<one_indent> endsub
```

A.2.9 one_indent_simple.ngc

```
o<one_indent_simple> sub

#1=#5420 (current x position)
#2=#5421 (current y position)

G90 (absolute coordinates)
G0 X[#1+#5168] Y[#2-#5169] (move under the indenter)

M125 (clear logfile)
o<indentation_simple> call
M124 (export logfile)

G90 (absolute coordinates)
G0 Z0 (move home with indenter)
```

```
#3=#5420 (current x position)
#4=#5421 (current y position)
G0 X[#3-#5168] Y[#4+#5169] (move under camera)

M126 (make picture of the indenter)
o<one_indent> endsub
```

A.2.10 show_indent.ngc

```
o<show_indent> sub

#1=#5420
#2=#5421

G90
G0 X[#1-#5168] Y[#2+#5169]
M127
o<show_indent> endsub
```

A.3 HAL soubory indentoru se zatížením konstantní silou

A.3.1 microindentor_simple.hal

```
# Generated by stepconf at Mon Jul  1 17:13:37 2013
# If you make changes to this file, they will be
# overwritten when you run stepconf again
loadrt trivkins
loadrt [EMCMOT]EMCMOT base_period_nsec=[EMCMOT]BASE_PERIOD
   ↪ servo_period_nsec=[EMCMOT]SERVO_PERIOD num_joints=[TRAJ]
   ↪ AXES num_aio=7
#loadrt probe_parport
loadrt hal_parport cfg="0 out  "
setp parport.0.reset-time 1000
loadrt stepgen step_type=0,0,0,0
loadrt pwmgen output_type=1

addf parport.0.read base-thread
addf stepgen.make-pulses base-thread
```

```
addf pwmgen.make-pulses base-thread
addf parport.0.write base-thread
addf parport.0.reset base-thread

addf stepgen.capture-position servo-thread
addf motion-command-handler servo-thread
addf motion-controller servo-thread
addf stepgen.update-freq servo-thread
addf pwmgen.update servo-thread

#net spindle-cmd <= motion.spindle-speed-out => pwmgen.0.value
#net spindle-on <= motion.spindle-on => pwmgen.0.enable
#net spindle-pwm <= pwmgen.0.pwm
#setp pwmgen.0.pwm-freq 100.0
#setp pwmgen.0.scale 1166.66666667
#setp pwmgen.0.offset 0.114285714286
#setp pwmgen.0.dither-pwm true
#net spindle-cw <= motion.spindle-forward

net estop-out => parport.0.pin-01-out
net xstep => parport.0.pin-02-out
setp parport.0.pin-02-out-reset 1
net xdir => parport.0.pin-03-out
net ystep => parport.0.pin-04-out
setp parport.0.pin-04-out-reset 1
net ydir => parport.0.pin-05-out
net zstep => parport.0.pin-06-out
setp parport.0.pin-06-out-reset 1
net zdir => parport.0.pin-07-out
#net astep => parport.0.pin-08-out
setp parport.0.pin-08-out-reset 1
#net adir => parport.0.pin-09-out
#net spindle-cw => parport.0.pin-14-out
```

```
#net spindle-pwm => parport.0.pin-16-out
net xenable => parport.0.pin-17-out

setp stepgen.0.position-scale [AXIS_0]SCALE
setp stepgen.0.steplen 1
setp stepgen.0.stepspace 0
setp stepgen.0.dirhold 15200
setp stepgen.0.dirsetup 15200
setp stepgen.0.maxaccel [AXIS_0]STEPGEN_MAXACCEL
net xpos-cmd axis.0.motor-pos-cmd => stepgen.0.position-cmd
net xpos-fb stepgen.0.position-fb => axis.0.motor-pos-fb
net xstep <= stepgen.0.step
net xdir <= stepgen.0.dir
net xenable axis.0.amp-enable-out => stepgen.0.enable

setp stepgen.1.position-scale [AXIS_1]SCALE
setp stepgen.1.steplen 1
setp stepgen.1.stepspace 0
setp stepgen.1.dirhold 15200
setp stepgen.1.dirsetup 15200
setp stepgen.1.maxaccel [AXIS_1]STEPGEN_MAXACCEL
net ypos-cmd axis.1.motor-pos-cmd => stepgen.1.position-cmd
net ypos-fb stepgen.1.position-fb => axis.1.motor-pos-fb
net ystep <= stepgen.1.step
net ydir <= stepgen.1.dir
net yenable axis.1.amp-enable-out => stepgen.1.enable

setp stepgen.2.position-scale [AXIS_2]SCALE
setp stepgen.2.steplen 1
setp stepgen.2.stepspace 0
setp stepgen.2.dirhold 15200
setp stepgen.2.dirsetup 15200
setp stepgen.2.maxaccel [AXIS_2]STEPGEN_MAXACCEL
```

```
net zpos-cmd axis.2.motor-pos-cmd => stepgen.2.position-cmd
net zpos-fb stepgen.2.position-fb => axis.2.motor-pos-fb
net zstep <= stepgen.2.step
net zdir <= stepgen.2.dir
net zenable axis.2.amp-enable-out => stepgen.2.enable

#setp stepgen.3.position-scale [AXIS_3]SCALE
#setp stepgen.3.steplen 1
#setp stepgen.3.stepspace 0
#setp stepgen.3.dirhold 15200
#setp stepgen.3.dirsetup 15200
#setp stepgen.3.maxaccel [AXIS_3]STEPGEN_MAXACCEL
#net apos-cmd axis.3.motor-pos-cmd => stepgen.3.position-cmd
#net apos-fb stepgen.3.position-fb => axis.3.motor-pos-fb
#net astep <= stepgen.3.step
#net adir <= stepgen.3.dir
#net aenable axis.3.amp-enable-out => stepgen.3.enable

net estop-out <= iocontrol.0.user-enable-out
net estop-out => iocontrol.0.emc-enable-in

#loadusr -W hal_manualtoolchange
#net tool-change iocontrol.0.tool-change => hal_manualtoolchange.
   ↪ change
#net tool-changed iocontrol.0.tool-changed <=
   ↪ hal_manualtoolchange.changed
#net tool-number iocontrol.0.tool-prep-number =>
   ↪ hal_manualtoolchange.number
#net tool-prepare-loopback iocontrol.0.tool-prepare => iocontrol
   ↪ .0.tool-prepared
```

A.3.2 custom_postgui_simple.hal

```
# Include your customized HAL commands here
# The commands in this file are run after the AXIS GUI (including
    ↪ PyVCP panel) starts

### LOADCELL
# instead of using halsampler from cmd, load it realtime
# each second           = 1000000000 nS
# each 0.1 second       = 100000000 nS
# each 0.01 second      = 10000000 nS
# each 0.001 second     = 1000000 nS
# each 0.02 second      = 20000000 nS   (orbit merret 50 Hz
    ↪ sampling rate - set the box to 100 Hz rate!)

loadrt threads name1=sampler_thread period1=20000000

# sampler (cmd: halsampler >> logfile.txt) "FFFFF" means 5 pins!
loadrt sampler depth=10 cfg="FFUUU"
addf sampler.0 sampler_thread

# name the force readout "force" and add it to sampler (pin 1)
net force loadcell.forcevalue => sampler.0.pin.1
net fread loadcell.forceread

# add force value to motion analog pin 0
linksp force motion.analog-in-00

# link force value to the pyvcp value displayed in GUI
net force => gladevcp.force_label

# name the systemtime readouts and add them to sampler
net date1 systime.date1 => sampler.0.pin.2
```

```
net date2 systime.date2 => sampler.0.pin.3
net date3 systime.date3 => sampler.0.pin.4

# add signal that force value has been read to motion digital pin
   ↪ 0
#linksp fread motion.digital-in-00
#newsig sig1 bit
#linksp sig1 motion.digital-out-00

# add the z-position to the sampler and ositions seenable in hbar
net alog halui.axis.2.pos-feedback => sampler.0.pin.0 gladevcp.
   ↪ indent_hbar

# load halsampler to user-space and make it log into a file
loadusr halsampler -t ./logfile/logfile.txt

#load function and 3 times
loadrt and2 count=3
addf and2.0 servo-thread
addf and2.1 servo-thread
addf and2.2 servo-thread

#Estop led resp Power led connection with estop resp power
net external-estop halui.estop.activate <= parport.0.pin-15-in
net estop-signal gladevcp.estop_led <= halui.estop.is-activated
net power-signal gladevcp.power_led and2.0.in0 <= halui.machine.
   ↪ is-on

#Home Button
net home-action gladevcp.home_button => halui.home-all     #halui.
   ↪ joint.0.home halui.joint.1.home halui.joint.2.home

#Homed led
```

```
net home-signal-x gladevcp.x_homed_led and2.0.in1 <= halui.joint
    ↪ .0.is-homed
net home-signal-y gladevcp.y_homed_led and2.1.in0 <= halui.joint
    ↪ .1.is-homed
net home-signal-z gladevcp.ind_homed_led and2.1.in1 <= halui.
    ↪ joint.2.is-homed

#Enable gray table if power is on and axis are homed
net and1 and2.2.in0 <= and2.0.out
net and2 and2.2.in1 <= and2.1.out
net jog-enable and2.2.out => gladevcp.gray_table

#Positions seenable in hbar
net x-axis-feedback gladevcp.xposition_hbar <= halui.axis.0.pos-
    ↪ feedback
net y-axis-feedback gladevcp.yposition_hbar <= halui.axis.1.pos-
    ↪ feedback
#net ind-axis-feedback gladevcp.indent_hbar <= halui.axis.2.pos-
    ↪ feedback

#Speedslider
net jog-speed-slider gladevcp.jog_speed => halui.jog-speed

#Axis jog
net x-axis-jog-plus gladevcp.x+_button => halui.jog.0.plus
net x-axis-jog-minus gladevcp.x-_button => halui.jog.0.minus

net y-axis-jog-plus gladevcp.y+_button => halui.jog.1.plus
net y-axis-jog-minus gladevcp.y-_button => halui.jog.1.minus

net ind-axis-jog-plus gladevcp.indent+_button => halui.jog.2.plus
net ind-axis-jog-minus gladevcp.indent-_button => halui.jog.2.
    ↪ minus
```

```
#homing and switch
net home-stop-switch-x axis.0.pos-lim-sw-in axis.0.home-sw-in <=
    ↪ parport.0.pin-10-in
net home-stop-switch-y axis.1.pos-lim-sw-in axis.1.home-sw-in <=
    ↪ parport.0.pin-11-in
net home-stop-switch-z axis.2.pos-lim-sw-in axis.2.home-sw-in <=
    ↪ parport.0.pin-12-in

#indentation contact
net motion-probe motion.probe-input gladevcp.contact_led <=
    ↪ parport.0.pin-13-in

#program led diode
net program-led gladevcp.program_led <= halui.program.is-running

#program run, pause, stop
net program-run gladevcp.start_button => halui.program.run halui.
    ↪ mode.auto
net program-stop gladevcp.stop_button => halui.program.stop halui
    ↪ .mdi-command-07
net program-pause gladevcp.pause_button => halui.program.pause

# NET SPINBOXES TO MOTION ANALOG
net number-x motion.analog-in-01 <= gladevcp.xgrid_spinbutton-f
net number-y motion.analog-in-02 <= gladevcp.ygrid_spinbutton-f
net coordinates-xstart motion.analog-in-03 <= gladevcp.
    ↪ xstart_spinbutton-f
net coordinates-ystart motion.analog-in-04 <= gladevcp.
    ↪ ystart_spinbutton-f
net coordinates-xend motion.analog-in-05 <= gladevcp.
    ↪ xend_spinbutton-f
```

```
net coordinates-yend motion.analog-in-06 <= gladevcp.
    ↪ yend_spinbutton-f

# NET LOGFILE OPERATION BUTTONS
net clear_logging_file halui.mdi-command-00 <= gladevcp.
    ↪ clear_button
net export_logfile halui.mdi-command-01 <= gladevcp.export_button
net save-position halui.mdi-command-02 <= gladevcp.
    ↪ save_position_button
net retrieve-position halui.mdi-command-03 <= gladevcp.
    ↪ ret_position_button

#NET CLOSE CAMERA WINDOW FUNCTION WITH GUI
net close-camera halui.mdi-command-04 <= gladevcp.
    ↪ close_camera_button

# NET BUTTON ONE INDENT IN GUI WITH OPERATION SHOW SURFACE -->
    ↪ MAKE INDENT AT SELECTED SPACE --> MAKE PICTURE OF THAT
    ↪ INDENT
net one-indent halui.mdi-command-05 <= gladevcp.one_indent_button

# NET BUTTON SHOW INDENT IN GUI WITH OPERATION SHOW INDENT
net show-indent halui.mdi-command-06 <= gladevcp.
    ↪ show_indent_button
```

A.3.3 loadcell_simple.hal

```
# Include your customized HAL commands here
# This file will not be overwritten when you run stepconf again

loadusr ./plugins/loadcell.py
loadusr ./plugins/system_time.py
```

A.4 HAL soubory indentoru řízeného silou

A.4.1 microindentor_instrumented.hal

```
# Generated by stepconf at Mon Jul  1 17:13:37 2013
# If you make changes to this file, they will be
# overwritten when you run stepconf again
loadrt trivkins
loadrt [EMCMOT]EMCMOT base_period_nsec=[EMCMOT]BASE_PERIOD
   ↪ servo_period_nsec=[EMCMOT]SERVO_PERIOD num_joints=[TRAJ]
   ↪ AXES num_aio=8
#loadrt probe_parport
loadrt hal_parport cfg="0 out  "
setp parport.0.reset-time 1000
loadrt stepgen step_type=0,0,0,0
loadrt pwmgen output_type=1

addf parport.0.read base-thread
addf stepgen.make-pulses base-thread
addf pwmgen.make-pulses base-thread
addf parport.0.write base-thread
addf parport.0.reset base-thread

addf stepgen.capture-position servo-thread
addf motion-command-handler servo-thread
addf motion-controller servo-thread
addf stepgen.update-freq servo-thread
addf pwmgen.update servo-thread

#net spindle-cmd <= motion.spindle-speed-out => pwmgen.0.value
#net spindle-on <= motion.spindle-on => pwmgen.0.enable
#net spindle-pwm <= pwmgen.0.pwm
#setp pwmgen.0.pwm-freq 100.0
```

```
#setp pwmgen.0.scale 1166.66666667
#setp pwmgen.0.offset 0.114285714286
#setp pwmgen.0.dither-pwm true
#net spindle-cw <= motion.spindle-forward

net estop-out => parport.0.pin-01-out
net xstep => parport.0.pin-02-out
setp parport.0.pin-02-out-reset 1
net xdir => parport.0.pin-03-out
net ystep => parport.0.pin-04-out
setp parport.0.pin-04-out-reset 1
net ydir => parport.0.pin-05-out
net zstep => parport.0.pin-06-out
setp parport.0.pin-06-out-reset 1
net zdir => parport.0.pin-07-out
#net astep => parport.0.pin-08-out
setp parport.0.pin-08-out-reset 1
#net adir => parport.0.pin-09-out
#net spindle-cw => parport.0.pin-14-out
#net spindle-pwm => parport.0.pin-16-out
net xenable => parport.0.pin-17-out

setp stepgen.0.position-scale [AXIS_0]SCALE
setp stepgen.0.steplen 1
setp stepgen.0.stepspace 0
setp stepgen.0.dirhold 15200
setp stepgen.0.dirsetup 15200
setp stepgen.0.maxaccel [AXIS_0]STEPGEN_MAXACCEL
net xpos-cmd axis.0.motor-pos-cmd => stepgen.0.position-cmd
net xpos-fb stepgen.0.position-fb => axis.0.motor-pos-fb
net xstep <= stepgen.0.step
net xdir <= stepgen.0.dir
net xenable axis.0.amp-enable-out => stepgen.0.enable
```

```
setp stepgen.1.position-scale [AXIS_1]SCALE
setp stepgen.1.steplen 1
setp stepgen.1.stepspace 0
setp stepgen.1.dirhold 15200
setp stepgen.1.dirsetup 15200
setp stepgen.1.maxaccel [AXIS_1]STEPGEN_MAXACCEL
net ypos-cmd axis.1.motor-pos-cmd => stepgen.1.position-cmd
net ypos-fb stepgen.1.position-fb => axis.1.motor-pos-fb
net ystep <= stepgen.1.step
net ydir <= stepgen.1.dir
net yenable axis.1.amp-enable-out => stepgen.1.enable

setp stepgen.2.position-scale [AXIS_2]SCALE
setp stepgen.2.steplen 1
setp stepgen.2.stepspace 0
setp stepgen.2.dirhold 15200
setp stepgen.2.dirsetup 15200
setp stepgen.2.maxaccel [AXIS_2]STEPGEN_MAXACCEL
net zpos-cmd axis.2.motor-pos-cmd => stepgen.2.position-cmd
net zpos-fb stepgen.2.position-fb => axis.2.motor-pos-fb
net zstep <= stepgen.2.step
net zdir <= stepgen.2.dir
net zenable axis.2.amp-enable-out => stepgen.2.enable

#setp stepgen.3.position-scale [AXIS_3]SCALE
#setp stepgen.3.steplen 1
#setp stepgen.3.stepspace 0
#setp stepgen.3.dirhold 15200
#setp stepgen.3.dirsetup 15200
#setp stepgen.3.maxaccel [AXIS_3]STEPGEN_MAXACCEL
#net apos-cmd axis.3.motor-pos-cmd => stepgen.3.position-cmd
#net apos-fb stepgen.3.position-fb => axis.3.motor-pos-fb
```

```
#net astep <= stepgen.3.step
#net adir <= stepgen.3.dir
#net aenable axis.3.amp-enable-out => stepgen.3.enable

net estop-out <= iocontrol.0.user-enable-out
net estop-out => iocontrol.0.emc-enable-in

#loadusr -W hal_manualtoolchange
#net tool-change iocontrol.0.tool-change => hal_manualtoolchange.
   ↪ change
#net tool-changed iocontrol.0.tool-changed <=
   ↪ hal_manualtoolchange.changed
#net tool-number iocontrol.0.tool-prep-number =>
   ↪ hal_manualtoolchange.number
#net tool-prepare-loopback iocontrol.0.tool-prepare => iocontrol
   ↪ .0.tool-prepared
```

A.4.2 custom_postgui_instrumented.hal

```
# Include your customized HAL commands here
# The commands in this file are run after the AXIS GUI (including
   ↪  PyVCP panel) starts

### LOADCELL
# instead of using halsampler from cmd, load it realtime
# each second           = 1000000000 nS
# each 0.1 second       = 100000000 nS
# each 0.01 second      = 10000000 nS
# each 0.001 second     = 1000000 nS
# each 0.02 second      = 20000000 nS   (orbit merret 50 Hz
   ↪ sampling rate - set the box to 100 Hz rate!)

loadrt threads name1=sampler_thread period1=20000000
```

```
loadrt wcomp count=1
addf wcomp.0 servo-thread

# sampler (cmd: halsampler >> logfile.txt) "FFFFF" means 5 pins!
loadrt sampler depth=10 cfg="FFUUU"
addf sampler.0 sampler_thread

# name the force readout "force" and add it to sampler (pin 1)
net force loadcell.forcevalue => sampler.0.pin.1 wcomp.0.in
net fread loadcell.forceread

# add force value to motion analog pin 0
linksp force motion.analog-in-00

# link force value to the pyvcp value displayed in GUI
net force => gladevcp.force_label

# name the systemtime readouts and add them to sampler
net date1 systime.date1 => sampler.0.pin.2
net date2 systime.date2 => sampler.0.pin.3
net date3 systime.date3 => sampler.0.pin.4

# add signal that force value has been read to motion digital pin
   ↪ 0
#linksp fread motion.digital-in-00
#newsig sig1 bit
#linksp sig1 motion.digital-out-00

# add the z-position to the sampler and ositions seenable in hbar
net alog halui.axis.2.pos-feedback => sampler.0.pin.0 gladevcp.
   ↪ indent_hbar
```

```
# load halsampler to user-space and make it log into a file
loadusr halsampler -t ./logfile/logfile.txt

#load function and 3 times
loadrt and2 count=3
addf and2.0 servo-thread
addf and2.1 servo-thread
addf and2.2 servo-thread

loadrt or2 count=1
addf or2.0 servo-thread

setp wcomp.0.min                    0.5
setp wcomp.0.max                    60

#Estop led resp Power led connection with estop resp power
net first-or-estop parport.0.pin-15-in => or2.0.in0
net second-or-estop wcomp.0.over => or2.0.in1
net external-estop halui.estop.activate <= or2.0.out
net estop-signal gladevcp.estop_led <= halui.estop.is-activated
net power-signal gladevcp.power_led and2.0.in0 <= halui.machine.
   ↪ is-on

#Home Button
net home-action gladevcp.home_button => halui.home-all     #halui.
   ↪ joint.0.home halui.joint.1.home halui.joint.2.home

#Homed led
net home-signal-x gladevcp.x_homed_led and2.0.in1 <= halui.joint
   ↪ .0.is-homed
net home-signal-y gladevcp.y_homed_led and2.1.in0 <= halui.joint
   ↪ .1.is-homed
```

```
net home-signal-z gladevcp.ind_homed_led and2.1.in1 <= halui.
    ↪ joint.2.is-homed

#Enable gray table if power is on and axis are homed
net and1 and2.2.in0 <= and2.0.out
net and2 and2.2.in1 <= and2.1.out
net jog-enable and2.2.out => gladevcp.gray_table

#Positions seenable in hbar
net x-axis-feedback gladevcp.xposition_hbar <= halui.axis.0.pos-
    ↪ feedback
net y-axis-feedback gladevcp.yposition_hbar <= halui.axis.1.pos-
    ↪ feedback
#net ind-axis-feedback gladevcp.indent_hbar <= halui.axis.2.pos-
    ↪ feedback

#Speedslider
net jog-speed-slider gladevcp.jog_speed => halui.jog-speed

#Axis jog
net x-axis-jog-plus gladevcp.x+_button => halui.jog.0.plus
net x-axis-jog-minus gladevcp.x-_button => halui.jog.0.minus

net y-axis-jog-plus gladevcp.y+_button => halui.jog.1.plus
net y-axis-jog-minus gladevcp.y-_button => halui.jog.1.minus

net ind-axis-jog-plus gladevcp.indent+_button => halui.jog.2.plus
net ind-axis-jog-minus gladevcp.indent-_button => halui.jog.2.
    ↪ minus

#homing and switch
net home-stop-switch-x axis.0.pos-lim-sw-in axis.0.home-sw-in <=
    ↪ parport.0.pin-10-in
```

```
net home-stop-switch-y axis.1.pos-lim-sw-in axis.1.home-sw-in <=
    ↪ parport.0.pin-11-in
net home-stop-switch-z axis.2.pos-lim-sw-in axis.2.home-sw-in <=
    ↪ parport.0.pin-12-in

#indentation contact
#net motion-probe motion.probe-input gladevcp.contact_led <=
    ↪ parport.0.pin-13-in
net probe-input motion.probe-input gladevcp.contact_led <= wcomp
    ↪ .0.out

#program led diode
net program-led gladevcp.program_led <= halui.program.is-running

#program run, pause, stop- when zou press stop it also change to
    ↪ absolute coordinates
net program-run gladevcp.start_button => halui.program.run halui.
    ↪ mode.auto
net program-stop gladevcp.stop_button => halui.program.stop halui
    ↪ .mdi-command-05
net program-pause gladevcp.pause_button => halui.program.pause

# NET SPINBOXES TO MOTION ANALOG
net number-x motion.analog-in-01 <= gladevcp.xgrid_spinbutton-f
net number-y motion.analog-in-02 <= gladevcp.ygrid_spinbutton-f
net coordinates-xstart motion.analog-in-03 <= gladevcp.
    ↪ xstart_spinbutton-f
net coordinates-ystart motion.analog-in-04 <= gladevcp.
    ↪ ystart_spinbutton-f
net coordinates-xend motion.analog-in-05 <= gladevcp.
    ↪ xend_spinbutton-f
net coordinates-yend motion.analog-in-06 <= gladevcp.
    ↪ yend_spinbutton-f
```

```
net indentation-force motion.analog-in-07 <= gladevcp.
    ↪ force_spinbutton-f

# NET LOGFILE OPERATION BUTTONS
net clear_logging_file halui.mdi-command-00 <= gladevcp.
    ↪ clear_button
net export_logfile halui.mdi-command-01 <= gladevcp.export_button

# NET SAVE POSITION AND RETRIEVE POSITION
net save-position halui.mdi-command-02 <= gladevcp.
    ↪ save_position_button
net retrieve-position halui.mdi-command-03 <= gladevcp.
    ↪ ret_position_button

# NET TESTING INDENT OPERATION WITH BUTTON IN GUI
net testing-indent halui.mdi-command-04 <= gladevcp.
    ↪ test_indent_button

# NET BUTTON SHOW INDENT IN GUI WITH OPERATION SHOW INDENT
net show-indent halui.mdi-command-06 <= gladevcp.
    ↪ show_indent_button

# NET BUTTON ONE INDENT IN GUI WITH OPERATION SHOW SURFACE -->
    ↪ MAKE INDENT AT SELECTED SPACE --> MAKE PICTURE OF THAT
    ↪ INDENT
net one-indent halui.mdi-command-07 <= gladevcp.one_indent_button

#NET CLOSE CAMERA WINDOW FUNCTION WITH GUI
net close-camera halui.mdi-command-08 <= gladevcp.
    ↪ close_camera_button
```

A.4.3 loadcell_instrumented.hal

```
# Include your customized HAL commands here
# This file will not be overwritten when you run stepconf again

loadusr ./plugins/loadcell.py
loadusr ./plugins/system_time.py
```

A.5 M soubory

A.5.1 M123

```
#!/bin/sh
#
# show LIVEPLOT window with graph when used in NGC file
#
# to be used with NGC files
# usage: M123
#
# Tomas Fila; ITAM AS CR; Feb 2015

sleep 3
xterm -e /home/biomech/linuxcnc_nela/m_files/liveplot &

exit 0
```

A.5.2 M124

```
#!/bin/sh
#
# renames the LOGFILE to UNIQUE NAME (date and time)
# which prevents overwriting the old measurement files
```

```
#
# to be used with veprohrnek NGC files
# usage: M124
#
# Tomas Fila, Petr Koudelka; ITAM AS CR; Aug 2012
# based on M106 by Ondrej Jirousek

# set destination directory
DIR=~/laboratory/force_logs

# rename logfile according to time and date
#FILE=`date +%Y%m%d-%H%M%S-$(($(date +%N) / 1000000))`
FILE=`date +%Y%m%d-%H%M%S`
FILE="num_force_displ_${FILE}"

# stop halsampler and wait until truly stopped
sleep 1
killall -STOP halsampler
sleep 1

# copy logfile to destination directory and rename it
cp ./logfile/logfile.txt $DIR/$FILE

# clear the logfile
> ./logfile/logfile.txt

# delete the first and the last line from the logfile
sed -i 1d $DIR/$FILE
sed -i '$d' $DIR/$FILE
# delete lines where overruns occured
sed -i '/overrun/d' $DIR/$FILE

# let halsampler continue
```

```
killall -CONT halsampler

exit 0
```

A.5.3 M125

```
#!/bin/sh
#
# tool to delete LOGFILE.TXT before measurement
# to be used with linux NGC files
#
# usage: M125
#
# Tomas Fila, Petr Koudelka; ITAM AS CR; Aug 2012

# terminate halsampler - otherwise severe errors occure!
killall -TERM halsampler

# remove logfile
rm ./logfile/logfile.txt

# restart halsampler using HAL
halcmd loadusr halsampler -t ./logfile/logfile.txt

exit 0
```

A.5.4 M126

```
#!/bin/sh
#
# make picture with manta when used in NGC file
#
```

```
# to be used with NGC files
# usage: M126
#
# Nela Fenclova; ITAM AS CR; Feb 2015

cd /home/biomech/laboratory_software/manta/bioMantaSnapShot_v0.33
python bioMantaSnapShot_v0.33_light.py -t ./snap -p matice

exit 0
```

A.5.5 M127

```
#!/bin/sh
#
# show view of manta when used in NGC file
#
# to be used with NGC files
# usage: M127
# ukonceni: killall python
# Nela Fenclova; ITAM AS CR; Feb 2015

cd /home/biomech/laboratory_software/manta/bioManta_v0.33_light
python bioManta_v0.33_light.py -s 25 &



exit 0
```

A.5.6 M128

```
#!/bin/sh
#
# close manta window
```

```
#
# to be used with NGC files
# usage: M128
# ukonceni: killall python
# Nela Fenclova; ITAM AS CR; Mar 2015

killall python



exit 0
```

A.6 Pluginy

A.6.1 loadcell.py

```
#!/usr/bin/python
import hal, time, serial
import datetime
import os
from os.path import expanduser

#DEBUG
DEBUG = 0

# Open file where the force values will be written:
##measurement_file=open("/home/biomech/single_trab/software/
    ↪ silomer/force_output.txt","w")

# Read the limits from a file
# default file is: /home/biomech/single_trab/software/silomer/
    ↪ limits.txt
home=expanduser("~")
```

```
file_path=home+"/linuxcnc2/plugins/limits.txt"
limits_file=open(file_path,"r")
limits=limits_file.readline()
limits=limits.strip().split()
limits_file.close()
limits_file_time_last=os.path.getmtime(file_path)

lower = float(limits[0])
upper = float(limits[1])
force_limit_reached = 0

# Create a new HAL pin which can be hooked to pyvcp
h = hal.component("loadcell")
h.newpin("forcevalue", hal.HAL_FLOAT, hal.HAL_OUT)
h.newpin("forceread", hal.HAL_BIT, hal.HAL_OUT)
h.newpin("force_limit_reached", hal.HAL_BIT, hal.HAL_OUT)
h.ready()

def sendDemand():        #send #99 in hexa
        ser.write(chr(0x23))
        ser.write(chr(0x39))
        ser.write(chr(0x39))
        ser.write(chr(0x0D))

#ser = serial.Serial('/dev/ttyS0',9600,8)
ser = serial.Serial('/dev/ttyUSB0',9600,8)
ser.open()

try:
        timeStart=time.time()
        sendDemand()
        while 1:
```

```
if (limits_file_time_last!=os.path.getmtime(
    ↪ file_path)):
        limits_file=open(file_path,"r")
        limits=limits_file.readline()
        limits=limits.strip().split()
        lower = float(limits[0])
        upper = float(limits[1])
        limits_file_time_last=os.path.getmtime(
            ↪ file_path)
#time.sleep(0.001)
h.forceread = 0
a=ser.inWaiting()
if a:
        message=""
        end_message=0
        while end_message!=1:
                char=ser.read(1)        # read one
                    ↪ char from buffer
                #print char.encode("hex")
                message=message+char
                if char.encode("hex")=="0d":
                    ↪          #end of message
                        message=message[3:]
                        try:
                                message=float(
                                    ↪ message)
                                now = datetime.
                                    ↪ datetime.
                                    ↪ now()
                                timestamp=now.
                                    ↪ strftime("%
                                    ↪ Y%m%d-%H%M%
                                    ↪ S-")+
```

```
                    ↪ unicode(now
                    ↪ .
                    ↪ microsecond
                    ↪ /1000)
                measureTime=
                    ↪ unicode(
                    ↪ time.time()
                    ↪ -timeStart)
                ##measurement_
                    ↪ file.write(
                    ↪ unicode(
                    ↪ timestamp)
                    ↪ +"\t"+
                    ↪ unicode(
                    ↪ measureTime
                    ↪ )+"\t"+
                    ↪ unicode(
                    ↪ message)+"\
                    ↪ n")
                ##measurement_
                    ↪ file.flush
                    ↪ ()
            except:
                #print message
                print "wrong␣
                    ↪ recieve␣
                    ↪ message"
                end_message=1
                sendDemand()
                continue
            #print message
            h.forcevalue = message
            h.forceread = 1
```

```
                                        end_message=1
                                        #time.sleep(0.2)
                                        sendDemand()
                                        if h.forcevalue > upper
                                            ↪ or h.forcevalue <
                                            ↪ lower:
                                                h.force_limit_
                                                    ↪ reached = 1
                                        else:
                                                h.force_limit_
                                                    ↪ reached = 0

                    time.sleep(0.01)
except KeyboardInterrupt:
    raise SystemExit

##measurement_file.close()
```

A.6.2 system_time.py

```
#!/usr/bin/python
#**************************
#read system time and save to variable 'timestamp' in coriander
    ↪ format YYYYMMDD-HHMMSS-microsecond (e.g.
    ↪ 20110513-172857-236)
#**************************
import hal, time
import datetime

DEBUG = 0

# create a new HAL pin which can be hooked to pyvcp
h = hal.component("systime")
```

```
h.newpin("date1", hal.HAL_U32, hal.HAL_OUT)
h.newpin("date2", hal.HAL_U32, hal.HAL_OUT)
h.newpin("date3", hal.HAL_U32, hal.HAL_OUT)
h.newpin("dateread", hal.HAL_BIT, hal.HAL_OUT)
h.ready()

h.dateread = 0

try:
        while 1:
                time.sleep(0.0001)
                h.dateread = 1
                now = datetime.datetime.now()
                h.date1=int(now.strftime("%Y%m%d"))
                h.date2=int(now.strftime("%H%M%S"))
                h.date3=int(now.microsecond/1000)

except KeyboardInterrupt:
        raise SystemExit

#       timestamp=now.strftime("%Y%m%d-%H%M%S-")+unicode(now.
    ↪ microsecond/1000)
#       print date1
#       print date2
#       print date3
#       print timestamp                 #print variable with
    ↪ system time
```

Příloha B

Zdrojové kódy pro vyhodnocení pomocí software Matlab

B.1 Skripty pro vyhodnocení tvrdosti podle Vickerse

B.1.1 vickers_indentor.m

```
disp 'run...'

clear all
close all

F = 30;                                          %set indentation
    ↪ force
maxDev = 5;                                      %maximal
    ↪ angle deviation
Aireg = 10;                                      %maximal
    ↪ triangle iregularity
alfa=68;                                 %1/2 angle of Vickers
    ↪ indentor

%%input values dialog  (2011/10/27)
```

```
prompt = {'Force [N]','Angle deviation tolerance [%]','Areas
    ↪ iregularity tolerance [%]'};                    %input values
dlg_title = 'Input values';                           %dialog
    ↪ title
num_lines = 1;                                        %number
    ↪ of lines for each user-entered value
def = {num2str(F), num2str(maxDev), num2str(Aireg)};  %default
    ↪ values
answer = inputdlg(prompt,dlg_title,num_lines,def);    %answer
    ↪ cell

F = str2num(answer{1});                               %cell
    ↪ string to number
maxDev = str2num(answer{2})/100;
Aireg = str2num(answer{3})/100;

%%dialog for reading specified image files (2011/10/20 v0.3)
[ImData, CoreName, FileName, PathName] = ImReadDialog('show');

hold on                                               %retain
    ↪ current graph in figur

%%find optimal size of graphical window for indent vertex picking
    ↪ (2011/10/18 doktor@itam.cas.cz)
title('Pick a diagonal of one indent.')               %figure
    ↪ title
[xa,ya]=ginput(2);                                    %
    ↪ graphical input from mouse
la=1.2*ceil(sqrt((xa(2)-xa(1))^2+(ya(2)-ya(1))^2));   %
    ↪ subwindow size

%%manual picking of indent vertices & Delaunay triangulation
    ↪ (2011/10/17 kytyr@itam.cas.cz)
```

```
%%3D matrix of vertices (x,y,indent number) and triangle mesh
    ↪ metrix based on grafical imput
%% dialog box (2011/10/19)
prompt = 'Number of indents to assess:';                    %question
dlg_title = 'Indents selection';                            %dialog
    ↪ title
num_lines = 1;                                              %number
    ↪ of lines for each user-entered value
def = {'1'};                                                %default
    ↪ value
answer = inputdlg(prompt,dlg_title,num_lines,def);          %answer
    ↪ cell

if isempty(answer)                                          %if you
    ↪ press "cancel"
        answer{1,1} = '-1';
end

nind = str2num(answer{1,1});                                %number
    ↪ of indents

j=1;
while j~=nind+1
        %% work only with subwindow containing selected indent to
            ↪ avoid slow response of MatLab (2011/10/18
            ↪ doktor@itam.cas.cz)
        title('Select an indent.')
        [x0,y0]=ginput(1);                                  %
            ↪ select an indent
        yL=ceil(max(1,y0-la));                              %
            ↪ subwindow definition
        xL=ceil(max(1,x0-la));
        yH=floor(min(y0+la,size(ImData,1)));
```

```
xH=floor(min(x0+la,size(ImData,2)));

%%open a new figure for the subwindow containing selected
    ↪ indent
hold off
figure
imshow(ImData(yL:yH,xL:xH));                                    %
    ↪ show selected subwindow

repick = 1;
while repick == 1
        [x,y] = ginput(5);
            ↪          %pick 5 verteces of the indent
        Tri = delaunay(x,y);
            ↪          %create triangle mesh

        %% check picked triangles          (2011/10/26
            ↪ doktor@itam.cas.cz)
        for i=size(Tri,1):-1:1
            %TeX: $|UV| = \sqrt{ (u_x- v_x)^2 + (u_y-
                ↪ v_y)^2  }$
            u=[x(Tri(i,1))-x(Tri(i,2)) y(Tri(i,1))-y(
                ↪ Tri(i,2))];       %points distance
            v=[x(Tri(i,3))-x(Tri(i,2)) y(Tri(i,3))-y(
                ↪ Tri(i,2))];
            w=[x(Tri(i,1))-x(Tri(i,3)) y(Tri(i,1))-y(
                ↪ Tri(i,3))];
            %TeX: $\varphi = arccos \frac{|{\bf u} \
                ↪ cdot {\bf v}|}{|{\bf u}| \cdot |{\bf
                ↪ v}|}$
            phi(i,1)=acos((abs(dot(u,v))/(norm(u)*norm(
                ↪ v))));
```

```
            phi(i,2)=acos((abs(dot(w,v))/(norm(w)*norm(
                ↪ v))));
            phi(i,3)=acos((abs(dot(u,w))/(norm(u)*norm(
                ↪ w))));
        end

        phi=max(phi,[],2);
        phiDev=abs(phi-pi/2)/(pi/2);

        maxDev = 0.05;
            ↪          %maximal deviation centrol
            ↪ (2011/10/27)
        if max(phiDev) > maxDev
            warmsg = {'Indent nonsquareness: [%]'
                ↪ num2str(max(phiDev)*100) 'Try it
                ↪ again?'};          %cell warning
                ↪ message
            dlg_title = 'Pick carefully';
                ↪                    %dialog title
            def = 'Yes';
                ↪                    %default value
            reply3 = questdlg(warmsg,dlg_title,'Yes',
                ↪ 'No',def);        %answer
            switch reply3
            case 'Yes'
                repick = 1;
            case 'No'
                repick = -1;
            end
        else
            repick = -1;
        end
end     %while repick == 1
```

```
    %% show picked indent in the working subwindow
       ↪ (2011/10/18)
    hold on
    triplot(Tri,x,y,'Color',[1 0 1]);                    %
       ↪ plot the mesh

    %% shift obtained coordinates into correct position in
       ↪ the original image
    x=x+xL-1;
    y=y+yL-1;

    %% store obtained coordinates and conectivities
    vrtx(:,1,j) = x;
    vrtx(:,2,j) = y;
    TRI(:,:,j) = Tri;
stredy(j,1)=x(1);
stredy(j,2)=y(1);


    %% close the subwindow using dialog and switch
       ↪ (2011/10/20)
    if nind == -1
            prompt = 'Do you want pick another indent?';
               ↪          %question
            dlg_title = 'next indent';
               ↪          %dialog title
            def = 'Yes';
               ↪          %default value
            reply2 = questdlg(prompt,dlg_title,'Yes','No',def
               ↪ );      %answer
            switch reply2
            case 'Yes'
```

```
                    stp = 1;
                case 'No'
                    stp = -1;
                end
            else
                title('Press a key to close this window and
                    ↪ continue.')
                pause
            end
            close

            %% add picked indent into the original image (2011/10/18)
            hold on
            triplot(Tri,x,y,'Color',[1 0 1])
            if (nind == -1) && (stp == -1)
                j = nind+1;
            else
                j = j+1;
            end

end                                                    %while j
    ↪ ~=nind+1

%%indents numbering
for i=1:1:size(vrtx,3)
            xt = floor(sum(vrtx(:,1,i))/size(vrtx,1));          %
                ↪ X-coodr indent center
            yt = floor(sum(vrtx(:,2,i))/size(vrtx,1));          %
                ↪ X-coodr indent center
            text(xt+100,yt,num2str(i),'Color',[1 0 1],'FontSize',18)
                ↪          %text in figure
        stredy(i,3)=xt;
        stredy(i,4)=yt;
```

```
end

pause(0.1)

%%make results folder
DirName = strcat(CoreName,'_res');
mkdir(DirName);
cd(DirName);

%%save image dialog (2011/10/18)
Title = 'Save image with indents';                          %dialog
    ↪ title
SaveName = strcat(CoreName,'_ind');                         %default
    ↪ filename
FilterSpec = {'*.png'};                                     %image
    ↪ types list
[SaveIm] = uiputfile(FilterSpec,Title,SaveName);            %save
    ↪ image dialog
if SaveIm == 0
        disp('nothing to save');
else
        saveas(gcf,SaveIm);                                 %
            ↪ save current figure as...
end

close

pxsize = load('..\..\vars\pxsize.txt', '-ascii')

vrtx = vrtx*(pxsize/1000);                                  %px to mm
    ↪ conversion

%%same matrix of vertices and triangle mesh (2011/10/20)
```

```
Title = 'Save matrix of vertices';
SaveName = strcat(CoreName, '_vtrx');
[SaveVrtx] = uiputfile('*mat', Title, SaveName);
pause(0.1)
Title = 'Save mesh matrix';
SaveName = strcat(CoreName, '_TRI');
[SaveTRI] = uiputfile('*mat', Title, SaveName);

if size(SaveVrtx,2) < 2 && size(SaveTRI,2) < 2                    %
   ↪ if strings are shorter...
        errordlg('The matrix is empty','Warning');                %
           ↪ warning message
else
        save([SaveVrtx,'.mat'],'vrtx');                           %
           ↪ save matrix
        save([SaveTRI,'.mat'],'TRI');
end

cd ..;

AREA = TriSurfArea(vrtx,TRI);

%%indent regularity control (2011/11/05)
j = 1;
iregN = [];
for i=1:1:size(AREA,3)
Amean = mean(AREA(:,:,i));
Astd = std(AREA(:,:,i));
if Astd/Amean > Aireg
iregN(j,1) = i;
iregN(j,2) = Astd/Amean*100;
j = j + 1;
end
```

```
end

%if iregN ~= []
%warmsg = {'triangle regularity [%]:' num2str(iregN) 'Do you want
    ↪  ignore them?'}; %cell warning message
%dlg_title = 'triangle regularity control';
%def = 'Yes'; %default value
%reply4 = questdlg(warmsg, dlg_title ,'Yes','No',def); %answer
%switch reply4
%case 'Yes'
%AREA(:,:,iregN(:,1)) = []; %delete bad indents
%case 'No'
%end
%end

%%Vickers hardness (2011/10/30)
%area of the indent consist of 4 triangles
indA=sum(AREA,2);
%for i=1:1:size(indA,3)
        %TeX: $HV = \frac{F}{A}$ $A \approx \frac{d^2}{1.8544}$
  %  HV(i)=F/indA(:,:,i)*(2/0.1891)*0.9;
    %stredy(i,5)=HV(i);
%end

%% vypocet podle uhlopricek
for i=1:1:size(vrtx,3)
uhlopricka1(i)=sqrt((vrtx(2,1,i)-vrtx(4,1,i))^2+(vrtx(2,2,i)-vrtx
    ↪ (4,2,i))^2);
uhlopricka2(i)=sqrt((vrtx(3,1,i)-vrtx(5,1,i))^2+(vrtx(3,2,i)-vrtx
    ↪ (5,2,i))^2);
u2(i)=((uhlopricka1(i)+uhlopricka2(i))/2)^2;
HV2(i)=0.1891*(F/u2(i))
stredy(i,5)=HV2(i);
```

```
end

format long
cd(DirName);
%Title = 'Save Vickers Hardness';                                    %save
    ↪ hardness map data
%SaveName = strcat(CoreName,'_HV');
%[SaveHV] = uiputfile('*mat',Title,SaveName);
%save([SaveHV,'.mat'],'HV','-ASCII');
Title = 'Save Vickers Hardness 2';                                   %save
    ↪ hardness map data 2
SaveName = strcat(CoreName,'_HV2');
[SaveHV2] = uiputfile('*mat',Title,SaveName);
save([SaveHV2,'.mat'],'HV2','-ASCII');

Title = 'Save Vickers Hardness and coodrs indent center';
    ↪                    %save hardness map data
SaveName = strcat(CoreName,'_stredy');
[Savestredy] = uiputfile('*mat',Title,SaveName);
save([Savestredy,'.mat'],'stredy','-ASCII');

Title = 'Save avarage length of diagonal';                                %
    ↪ save hardness map data 2
SaveName = strcat(CoreName,'_d');
[Saved] = uiputfile('*mat',Title,SaveName);
save([Saved,'.mat'],'u2','-ASCII');

Title = 'Save length of diagonal 1';                                 %save
    ↪ hardness map data 2
SaveName = strcat(CoreName,'_u1');
[Saved] = uiputfile('*mat',Title,SaveName);
save([Saved,'.mat'],'uhlopricka1','-ASCII');
```

```
Title = 'Save length of diagonal 2';                    %save
    ↪ hardness map data 2
SaveName = strcat(CoreName,'_u2');
[Saved] = uiputfile('*mat',Title,SaveName);
save([Saved,'.mat'],'uhlopricka2','-ASCII');

cd ..;




disp('EoF');
```

B.1.2 ImReadDialog.m

```
%ImReadDialog.m v0.3, 2011/10/20 kytyr@itam.cas.cz
%
%function [out, CoreName, FileName, PathName] = ImReadDialog(show
    ↪ )
%dialog for reading specified image files with "show" optional
    ↪ parameter"
%in case of cancelation all output values will be zero
%FilterSpec = {'*.bmp;*.tif;*.png;*.jpg'};
%Input:
%image file selection by dialog
%Output:
%image file and its core name, full name and path

function [ImData, CoreName, FileName, PathName] = ImReadDialog(
    ↪ show)

FilterSpec = {'*.bmp;*.tif;*.png;*.jpg'};               %image
    ↪ types list
```

```
Title = 'Select the image file';                    %dialog
    ↪ title
[FileName,PathName] = uigetfile(FilterSpec,Title);  %open
    ↪ standard dialog box

if nargin == 0
        show = 0;
else
        show = 1;
end

if FileName == 0
        errordlg('Loading canceled','Warning');
            ↪          %warning message
        CoreName = 0;
        ImData = 0;
        show = 0;
else
        ImData = imread([PathName FileName]);       %
            ↪ file loading
        dot = strfind(FileName, '.');               %
            ↪ "dot" position in filename
        CoreName = FileName(1:dot-1);               %
            ↪ core name
end

if show == 1
        imshow(ImData);                             %
            ↪ show image
end
```

B.2 Funkce pro výpočet velikosti pixelu

B.2.1 pixelsize.m

```
function [sc] = pxsize(filename)
% calibration of three point bending setup
%       calibration of ccd camera magnification
%
% usage [sc] = pxsize(filename)
%
%   'filename.ext' picture of the ruler
%   pick 0.5mm distance (corresponds to 5 longer marks)
%
% result saved in './vars/pxsize.txt', sc in [um/px]
%
% this version detects color depth and decides whether to convert
   ↪ into grayscale

projsc=imread(filename); %read the calibration image

image_info=imfinfo(filename);

if image_info.ColorType=='truecolor'

projsc=rgb2gray(projsc);

end

imshow(imadjust(projsc))      % show the image of the the setup

[x y]=ginput(2);      % record picked points

close                 % close the graphical window
```

```
d=sqrt(abs(y(2)-y(1))^2+abs(x(2)-x(1))^2); % distance between
    ↪ ruler marks (real distance 500um)

sc=500/d;

save ../vars/pxsize.txt sc -ascii

end
```