



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta strojní

Ústav technologie obrábění, projektování a metrologie

Programování maker pro použití měřicí sondy na obráběcím stroji

Programming of macros for measuring probe utilization on a machine tool

Diplomová práce

Studijní program: Výrobní a materiálové inženýrství

Vedoucí práce: Ing. Jan Tomíček Ph.D.

Jan Rajal

Praha 2015

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Jana Tomíčka Ph.D., veškerou použitou literaturu uvádím v seznamu literatury na konci mé diplomové práce.

V Praze dne 18.6.2015

Jan Rajal

Poděkování

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Janu Tomíčkoví Ph.D., za ochotu odbornou pomoc a poskytnuté rady při zpracování diplomové práce.

Dále bych také rád poděkoval týmu společnosti technology-support s.r.o. za odborné rady v této problematice.

Abstrakt

Ve své diplomové práci se zabývám možností využití obrobkové sondy na CNC obráběcím stroji. Dále se snažím ukázat možnosti cyklů v řídicích systémech a CAD softwarech, ale vzhledem k nejrůznějším variantám od nejrůznějších výrobců zde uvádím pouze ty, které se vyskytují u většiny výrobců nejčastěji. V praktické části se věnuji možnosti programování maker v CAM softwaru GibbsCAM, pro generování a simulaci drah obrobkové sondy, což nám umožňuje snížit čas potřebný pro ustavení součásti do stroje a kontrolu kolizí sondy.

Abstract

In my thesis I deal with the possibility of using the spindle probe on the CNC machine. I also try to show the possibilities of cycles in the control systems and CAD software, but due to various variants from different manufactures frequently. The practical part is devoted to the possibility of creating the macros for the CAM software GibbsCAM to generate and simulate toolpath of spindle probe, which allows us to reduce the time required for setting up the workpiece to machine and probe collision check.

Klíčová slova

CAD, CAM, GibbsCAM, makra, CNC stroje, měření obrobku, obrobkové sondy, programování cyklů, upínání obrobku

Keywords

CAD, CAM, GibbsCAM, macros, CNC machines, workpiece measurement, spindle probe, programming cycles, workpiece clamping

Obsah

1.	Úvod	- 9 -
2.	Základní pojmy.....	- 10 -
2.1	CAD software.....	- 10 -
2.2	CAM a CAD/CAM software	- 10 -
2.2.1	Postprocesor.....	- 11 -
2.3	Makra	- 12 -
2.4	CNC stroje.....	- 12 -
2.4.1	NC programy	- 13 -
2.4.2	Použití měřicích sond na CNC obráběcích strojích	- 13 -
2.4.3	Přesnost měření na CNC obráběcích strojích	- 16 -
2.5	Souřadnicové měřicí stroje.....	- 17 -
2.6	Sondy pro ustavení a měření obrobku na obráběcím stroji.....	- 18 -
2.6.1	Sondy s optickým senzorem	- 19 -
2.6.2	Sondy s kinematickým odporovým senzorem	- 19 -
2.6.3	Tenzometrické sondy	- 20 -
3.	Měřicí cykly v řídicích systémech.....	- 20 -
3.1	Důležité body při frézování na CNC strojích.....	- 20 -
3.2	Tabulka nulových bodů.....	- 22 -
3.3	Měřicí cykly	- 23 -
3.4	Nejčastěji používané měřicí cykly pro zjišťování vztažných bodů.....	- 23 -
3.4.1	Měření vztažné roviny	- 23 -
3.4.2	Zjišťování šikmé polohy obrobku.....	- 24 -
3.4.3	Měření vztažného bodu ve středu drážky	- 24 -
3.4.4	Měření vztažného bodu ve středu výstupku	- 24 -
3.4.5	Měření vztažného bodu ve středu kapes na součásti	- 25 -
3.4.6	Měření vztažného bodu ve středu obdélníkové nebo kruhové geometrie	- 25 -
3.4.7	Měření vztažného bodu vnitřních rohů součásti	- 25 -
3.5	Nejčastěji používané cykly pro kontrolu obrobků	- 26 -
3.5.1	Proměřování díry	- 27 -
3.5.2	Proměřování kruhové geometrie na součásti	- 27 -
3.6	Plug-in EasyProbe	- 27 -
4.	Měřicí cykly v CAM	- 28 -
5.	Parametry, cíle, použité vybavení a software pro vypracování.....	- 30 -
5.1	Parametry pro vypracování	- 32 -
5.2	Popis stroje a řídicího systému.....	- 32 -
5.3	GibbsCAM	- 33 -

5.3.1	Možnosti programování maker v softwaru GibbsCAM	- 34 -
5.3.2	Machine definition document	- 36 -
5.3.3	Možnosti úpravy postprocesoru v GibbsCAM	- 37 -
6.	Vypravování	- 39 -
6.1	Požadavky na cykly pro upínání obrobku	- 40 -
6.1.1	Celkový vygenerovaný NC program pro automatické měření	- 40 -
6.1.2	Popis jednotlivých částí generovaného kódu:	- 41 -
6.2	Použité příkazy ve změně makra	- 42 -
6.2.1	Soubor DefineData.txt	- 42 -
6.2.2	Soubor CycleData.dlg	- 43 -
6.2.3	Soubor CycleData.mac	- 46 -
6.2.4	Soubor ToolPath	- 48 -
6.3	Makra pro ustavení počátku ve směru osy X a Y	- 49 -
6.3.1	NC program pro měření vztažných bodů ve směru osy X	- 52 -
6.3.2	NC program pro měření vztažných bodů ve směru osy Y	- 53 -
6.4	Makro pro ustavení počátku ve směru osy Z	- 55 -
6.4.1	NC program pro měření vztažných bodů ve směru osy Z	- 57 -
6.5	Makro pro ustavení počátku ve všech třech osách	- 59 -
6.5.1	NC program pro měření ve všech třech osách	- 64 -
6.6	Makro pro ustavení vztažného bodu ve středu kruhu	- 67 -
6.6.1	NC program pro měření vztažných bodů ve středu kruhu	- 70 -
6.7	Úpravy Postprocesoru	- 71 -
6.7.1	Použité příkazy	- 71 -
6.7.2	Vytvořené a upravené rutiny	- 74 -
7.	Praktická aplikace	- 79 -
8.	Závěr	- 80 -
9.	Seznam literatury	- 81 -

1. Úvod

Ve své práci se zaměřuji na oblast využití a ovládání měřících sond v procesu obrábění a to zejména přesné upínání obrobku, mezioperační kontrola obrobku, kontrola opotřebení nástroje a proměření a kontrola obrobení. V těchto jednotlivých okruzích se zaměřuji na popsání jejich celkových možností a výhod, které zde můžeme využívat. Příkladem zde může být popis jednotlivých měřících cyklů řídicích systémů CNC obráběcích strojů pro ustavování součástí a měření různých typů geometrie, časová úspora při upínání a snížení zmetků při produkci.

Dále se snažím o definici základních pojmů, které se v daném okruhu vyskytují. Příkladem může být definice obecně používané terminologie v této oblasti, jakými jsou pojmy CNC stroje, CAD/CAM systémy, post procesory, NC kódy a tak dále. Tyto jednotlivé body popisují, abych podal ucelený obraz k této problematice.

Také se ve své práci zaměřuji na vlastní obrobkové sondy, které se používají k ustavování obrobku a na jeho proměřování. Na možnosti, které nám nabízejí a na jejich princip jakým pracují.

Výstupem mé práce by mělo být zpracování problematiky v oblasti upínání součástí v CNC obráběcích strojích. Jedná se zejména o součásti, které je potřeba přesně upnout do stroje, nebo kvůli rozdílnému tvaru jednotlivých polotovarů nejsme schopni požadované přesnosti upnutí dosáhnout upínacím přípravkem, jako jsou například odlitky. Využití měřících cyklů pro přesné upínání obrobku do stroje nám také šetří čas potřebný k upínání obrobku a snižuje možnosti špatného ustavení.

Jak jsem již nastínil cílem této práce je doplnit CAM software GibbsCAM o makra pro CNC frézovací centrum na ustavování počátku součásti. Dále zpracovat informace týkající se možnosti programování maker, možnosti tvorby a úpravy post procesorů v tomto CAMu. Vytvoření těchto maker nám pomůže dále zkrátit čas, kdy stroj nepracuje, neboť takto budeme schopni vytvořit měřící cykly přímo při vytváření strategií pro obrábění v GibbsCAMu.

Výsledkem by tedy mělo být plně funkční rozšíření, vytvořené podle přání zadavatele, které bude umožňovat ustavovat součást podle zadaných nulových bodů, které s pomocí makra a měřící sondy najdeme přímo na součásti.

2. Základní pojmy

Cílem této kapitoly je popsání a případné vysvětlení základních pojmů v oblasti provádění měření pomocí dotykové sondy na CNC obráběcích strojích. Jedná se v zásadě o to udělat si ucelený obraz na celou problematiku a to i v tématech, kterých se dotýkáme pouze okrajově. Výstupem této kapitoly by měl být komplexní pohled na práci s obráběcími stroji od prvotního vytvoření modelu až po finální výstup v podobě ověření přesnosti obrobení na souřadnicovém měřicím stroji.

2.1 CAD software

Computer Aided Design neboli počítačem podporovaná konstrukce nebo design, je označení programů pro tvorbu, nebo úpravu geometrie. Jedná se o typ softwaru, ve kterém na základě zadaných parametrů můžeme vytvářet buď 2D nebo 3D geometrii. Tyto softwary se v dnešní době liší převážně v možnostech, jak komplexní geometrii jsme s jejich pomocí schopni navrhnout.

Ve strojírenské výrobě se dnes vyskytuje větší množství možných vstupů, jakými jsme schopni popsat nové či upravené výrobky, ať už formou papírových výkresů, počítačové 2D geometrie, nebo 3D modelu. Ačkoli se jedná o nejstarší možnost tak se i v dnešní době papírové výkresy stále používají, ale jejich role již není pro přípravu výroby tak zásadní, většinou se používají pouze při kontrole a popisu specifických částí výrobku popřípadě s jejich pomocí vytváříme 3D model, který následně převedeme do CAMu. Počítačová 2D geometrie se stále vyskytuje a to především při soustružení, kde nám u rotační součásti stačí pro obrábění pouze profil dané součásti.

Jak jsem se již zmínil výstupem z těchto programů je tvorba 2D nebo 3D geometrie, kterou následně převedeme do Softwaru typu CAM a vycházíme s ní pro generování drah nástroje.

2.2 CAM a CAD/CAM software

Computer Aided Manufacturing neboli počítačem podporovaná výroba je obecným pojmenováním typu všech softwarů, které se používají ve strojírenské výrobě pro generování drah na číslicově řízených obráběcích strojích. Tyto softwary ve své práci nejčastěji vychází ze zadaných nebo importovaných geometrických dat a to buď v podobě 2D geometrie, nebo 3D CAD modelu.

K ovládání číslicově řízeného obráběcího stroje můžeme samozřejmě využít i jiné metody a to například ruční psaní NC programu. To je v dnešní době již velmi neefektivní a často již nepoužitelná možnost z důvodu výroby komplikovaných dílců, pro které již nejsme schopni ručně napsat tak složité dráhy abychom je byli schopni obrobit. Další možností je využití programovacího dialogu vlastního řídicího systému CNC stroje, možnosti vytváření drah nástroje tímto způsobem jsou rozdílné v závislosti na výrobcí řídicího systému, ale v zásadě nám mnohdy nabízí i tvorbu komplexních drah nástroje pro tvarově složité součásti. Programování přímo na stroji má ale tu nevýhodu, že v době, kdy tvoříme nebo doplňujeme předem připravená NC data z CAM softwaru, nemůžeme obrábět. To nás vrací zpět k dnes již nutnosti využití CAM softwaru a to tak abychom na počítači vytvořili, pokud možno kompletní NC program a následně ho nahráli přímo do stroje a mohli rovnou obrábět tak abychom minimalizovali čas, kdy stroj neobrabí.

Dnes již existuje pouze velmi málo čistě CAM softwaru, a většina výrobců prezentuje své programy jako tak zvané CAD/CAM, kde se jedná o kombinaci těchto dvou typů softwaru. Takže se zde kromě možnosti generování drah vyskytuje i možnost tvorby vlastní geometrie. Ale právě v oblasti tvorby vlastní geometrie se většina softwarů velmi liší.

Jak jsem se již zmínil, v CAM softwarech vycházíme z vlastních nebo importovaných geometrických dat, na jejichž základě vytvoříme dráhy nástroje pro obrábění na číslicově řízených strojích. Tyto dráhy jsou uloženy v part programu, kde se mimo jiné nachází i například používané nástroje. Z těchto informací obsažených v part programu pomocí procesoru vytvoříme CL data, ze kterých nakonec pomocí post procesoru vygenerujeme NC kód. Tento NC kód nám bude po nahrání do obráběcího stroje řídit všechny pohyby nástroje a obrobku, tak jak jsme si je naprogramovali v CAM programu.

2.2.1 Postprocessor

Post processor je součástí CAM softwaru a slouží k převodu dat z CAD/CAM programů, přesněji řečeno CL dat, do datového jazyka konkrétního obráběcího stroje, jeho výstupem je NC program. Existuje mnoho odlišných formátů NC kódu v závislosti na různých parametrech, jako je druh řídicího systému na daném obráběcím stroji, také každý technolog nebo programátor má odlišný programovací styl. Správně vytvořený post processor by měl obsahovat informace o vlastnostech daného stroje, tak aby bylo co nejlépe využito všech jeho funkcí v souladu s CAD/CAM systémem. Ideální by bylo, kdyby existoval pouze jeden absolutně univerzální post processor, což bohužel není možné a každý stroj potřebuje

svůj specifický. Hlavní důvod proč nelze vytvořit univerzální post procesor je ten, že na světě zřejmě neexistují dva zcela totožné CNC obráběcí stroje. Nabídka moderních obráběcích strojů je velmi vysoká a každý obráběcí stroj se vyskytuje ještě v několika konfiguracích. (1)
(2)

Někteří výrobci se snaží vytvářet, alespoň částečně univerzální post procesory, které se dají využívat na stejném modelu stroje, ale jak jsem se již zmínil takřka každý stroj má jinou konfiguraci a proto bychom měli mít pro každý stroj specificky napsaný a otestovaný post procesor. Tímto způsobem minimalizujeme riziko chyby, možné kolize a tím i poškození často velmi drahých číslicově řízených strojů.

2.3 Makra

Makro je posloupnost příkazů, která je uložena pod určitým názvem a kterou lze spustit jedinou akcí a to například klávesovou zkratkou. Slouží ke zjednodušení nebo doplnění funkcí, které využíváme. Makro může být napsáno v některém z běžně využívaných programovacích jazyků jako je C++, nebo C#, popřípadě ve vlastním programovacím jazyku daného programu. (3)

S makry se můžeme setkat v mnoha programech, ale i operačních systémech, mohou plnit nejrůznější funkce a úlohy, například otevření programu pomocí klávesové zkratky a podobně. Také se může jednat o doplnění určitých programů v počítači o funkce, které neobsahují, jako v případě této práce, kde doplňujeme CAM software o funkce ustavení obrobku.

2.4 CNC stroje

Jsou počítačem řízené obráběcí stroje, zkratka CNC pochází z anglického výrazu Computer Numerical Control v překladu číslicové řízení počítačem. Tyto stroje využívají počítač k tomu, aby dokázali obrábět dle předem připravených NC programů. V základu se dají CNC obráběcí stroje rozdělit na soustružnické, frézovací a kombinované soustružnicko-frézovací. V poslední době se také můžeme setkat s označení tak zvané Generic machine, neboli obecné stroje. Tento pojem vznikl z důvodu, že v mnoha případech nejsme schopni přesně rozlišit, jestli se jedná o soustružnický nebo frézovací obráběcí stroj neboť míra splynutí těchto dvou druhů už je tak velká, že jsme schopni je použít v obou případech takřka bez omezení. (4)

2.4.1 NC programy

Jak již bylo řečeno, CNC obráběcí stroje jsou řízeny NC programem. Jedná se obvykle o řetězec příkazů začínající písmenem, po kterém následuje číselná hodnota. Jednotlivé kódy byly v minulosti normalizovány do normy ISO 6983. Tato norma zahrnuje pouze základní příkazy, proto si mnozí výrobci řídicích systémů tuto normu doplnili o své unikátní operace, jakými mohou být například speciální způsoby vrtací, měřicí, soustružnické a frézovací cykly. Někteří výrobci tuto normu nedodržují vůbec. (4)

Příkazy, které používají v NC programu, můžeme rozdělit do několika skupin, G-kódy, M-kódy a ostatní. G-kódy začínají vždy písmenem G a používají se k primárnímu řízení stroje z hlediska umístění nástroje do řezné polohy. Příkladem může být například standardní posuv a rychloposuv. M-kódy obsahují pomocné funkce, které se starají o ovládání obráběcího stroje. Příkladem může být například výměna nástroje, nebo ovládání chlazení na stroji. (4)

U NC programů také nesmíme zapomenout na pojmy absolutního a přírůstkového zadávání souřadnic pohybu. Tyto druhy pohybu nám určuje G-kód G90 a G91. Absolutní způsob zadávání souřadnic G1 G90 X15. Y 20. F200. nám v tomto případě přesune nástroj do místa určeného souřadnicemi X15 Y 20 mm od nulového bodu rychlostí 200 mm za minutu. Přírůstkový způsob zadávání souřadnic pohybu G1 G91 X15. Y 20. F200. se odvíjí od bodu, ve kterém se momentálně nachází nástroj, který se posune v osách X15 Y 20 mm od současné polohy rychlostí 200 mm za minutu. (4)

2.4.2 Použití měřících sond na CNC obráběcích strojích

V této části se věnuji možnosti použití měřících sond na CNC obráběcích strojích. Celkové náklady na pořízení sondy pro stroj jsou v celkové ceně tvořeny relativně malou sumou a jejich správným použitím se nám tato investice může rychle vrátit. Zejména jsme schopni snížit prostoje a to až o 30%. Seřízení je přesnější a takřka nezávislé na lidském faktoru což snižuje zmetkovitost. (5)

Na obráběcím stroji se nejčastěji provádějí druhy měření:

- 1) Měření korekce nástroje
- 2) Zarovnání součásti a měření vztažných bodů
- 3) Mezioperační kontrola součásti

4) Proměrování a kontrola obrobení součásti

2.4.2.1 Měření korekce nástroje

Pokud chceme obrábět s co největší přesností je nutné před zahájením vlastního obrábění nejdříve provést korekce všech používaných nástrojů. Nástroje se časem tupí a je nutné v pravidelných intervalech je kontrolovat, což nám umožňují nejrůznější druhy měřících sond od dotykových až po laserové.

Měření korekcí nástroje můžeme provádět přímo na stroji, což nám šetří čas z hlediska toho, že není nutné daný nástroj vyndávat a znovu upínat do stroje. Nevýhoda ovšem je, že v průběhu měření stroj nemůže obrábět a především u velkosériových výrob, kdy potřebujeme, aby stroje obráběli co nejmenšími prodlevami, nelze toto řešení použít.

2.4.2.2 Zarovnání součásti a měření vztažných bodů

Mnoho podniků si dosud pořád myslí, že měření na stroji je stojí drahý obráběcí čas a neuvědomují si naopak, že v mnoha ohledech, jim správné použití měřící sondy může pomoci naopak čas ušetřit. Zarovnávání součásti a měření vztažných bodů na součásti je jedno z nejpraktičtějších využití měřících sond na CNC obráběcím stroji. I v případě kdy máme dobře navrhnutý upínací přípravek, který umožňuje snadné upnutí polotovaru, se může stát, že se trochu pootočí, popřípadě posune, operátor si toho nevšimne a začne obrábět. Nejvýraznější úsporou může být v tomto případě právě odstranění drahých upínacích přípravků, které jsou speciálně navrhovány pro upínání součástí, které je potřeba přesně obrobit. S použitím měřících sond nemusí být obrobek ve stroji upnut tak přesně, stačí pouze zarovnat souřadný systém. (5)

Zejména závažná je tato situace při obrábění součásti, která už je částečně obrobená a my jí potřebujeme doobrobit z druhé, nebo jiné strany. Příkladem tohoto může být například součást pro hydraulické rozvody, kde na polotovaru ve tvaru kostky navrtáváme díry z různých stran, které se musí vzájemně setkat. (5)

2.4.2.3 Mezioperační kontrola součásti

Tento druh měření se provádí při vytváření prvku na obrobku, který pro svojí tvorbu potřebuje několik operací. Příkladem tohoto druhu operací mohou být malé závitové otvory. Zlomení vrtáku při vrtání několika otvorů může způsobit následné zničení závitníku, nebo i poškození obrobku. Tímto typem kontroly můžeme odhalit takto špatně provedené operace, zároveň nás to upozorní na poškození nástroje. (5)

Některé cykly v řídicím systému, popřípadě programy jako plug-in Productivity+™ umožňují okamžitou kontrolu obrobení z hlediska přesnosti obrobení. Tato relativně nová funkce nám přináší velké výhody a to zejména v podobě vyhodnocení. Máme zde totiž několik možných případů, které mohou nastat. V nejlepším případě kontrola ukáže, že obrobená část na součásti je v námi zadaných tolerancích, můžeme tedy pokračovat k další operaci.

V druhém případě je výsledek kontroly nedostačující, ale v té podobě, že je součást stále opravitelná. Jako příklad zde můžeme uvést obrábění kruhové kapsy o průměru 25 mm s tolerancí $\pm 0,1$ mm, výsledek kontroly je hodnota 24,8 mm. Řídicí systém v tomto případě neprovede další operaci, ale místo toho vygeneruje novou dráhu pro obrobení kapsy s korekcemi, tak abychom se dostali na námi požadovanou přesnost obrobení. Tato možnost přináší obrovské výhody ve formě kontroly přímo na stroji, nyní máme možnost eliminovat velkou část chyb již v počátku, kdy jsme schopni je snadno a rychle odstranit. Tato kontrola samozřejmě nikdy nebude tak přesná jako výstupní kontrola na souřadnicovém měřícím stroji, ale i tak je šance na odhalení chyb přímo na stroji, což je vždy výhodnější, neboť když obrobek vyndáme ze stroje, proměříme a následně chceme opravit, již nikdy ho do stroje neupneme stejně přesně.

V posledním případě je výsledek kontroly také nevyhovující a navíc je obrobek neopravitelný. V případě, kdy se ve výrobě vyskytne zmetek tak je to vždy špatně, ale nyní máme možnost ho identifikovat hned v počátcích jeho vzniku. Například zjistíme-li po druhé operaci, že jde o neopravitelný zmetek, zastavíme stroj a další operace již neprovádíme. To nám umožňuje drastické snížení nákladů i v případě, že daný obrobek musíme vyhodit.

2.4.2.4 Proměření a kontrola obrobení součástí

Tato možnost není v praxi tolik využívána jako předchozí možnosti, i přesto se s ní můžeme setkat. Pokud po odepnutí obrobku ze stroje zjistíme na souřadnicovém měřícím stroji, že něco není v pořádku, už nikdy obrobek do stroje neupneme stejným způsobem. Právě měření na CNC obráběcím stroji zvyšuje pravděpodobnost, že měření v měrovém středisku bude v pořádku. U měření rozměrných dílů navíc může nastat situace, že neexistuje, nebo si nemůžeme dovolit zařízení tak veliké, které by je bylo schopno změřit. (5)

Toto měření by ale nemělo být směrodatné pro přesnost, ale pouze pro kontrolu, neboť pokud došlo k nepřesnému obrábění vlivem nepřesnosti stroje, budeme tuto chybu dále kopírovat i při vlastním proměřování. Tudíž se tyto cykly hodí zejména pro kontrolu částí na

obrobku, kde mohla nepřesnost vzniknout například ohnutím nástroje při frézování hluboké kapsy.

2.4.3 Přesnost měření na CNC obráběcích strojích

Pokud chceme měřit na CNC obráběcím stroji, měli bychom si nejdříve zjistit, jakou má stroj přesnost a tedy s jakou chybou jsme schopni provádět měření. Při dodání nového stroje se provádí takzvané Přejímací zkoušky stroje, jsou uskutečněny nejdříve u výrobce a následně i u zákazníka. Tyto zkoušky nám ověřují základní vlastnosti stroje a to nejen přesnost ale i rozměry, zdvih v jednotlivých osách, hluk a mnoho dalšího. V této práci se ale budeme zabývat pouze přesností a to geometrickou přesností, přesností polohování, pracovní a výrobní přesností. (6)

Metodami pro zkoušky přesnosti obráběcích strojů se zabývá norma ISO 240, která specifikuje měření, zkušební předpisy a zkoušky geometrické přesnosti. Před samotnými zkouškami je nutné stroj upevnit na vhodnou základnu a vyrovnat jej do vodorovné polohy. Zkoušky geometrické přesnosti by se měly provádět výhradně na stroji, který je plně smontován. Výjimky mohou nastat pouze v případech, kdy nejsme schopni měření provést jinak, příkladem může být demontáž stolu, aby mohly být změřeny vodící plochy. Měření by dále mělo probíhat za teplotních podmínek, při kterých bude stroj pracovat. (7)

Norma ISO 240-1 specifikuje zásady zkoušek geometrických přesností stroje pracujících bez zatížení. Cílem je normalizovat zkušební přesnosti obráběcích strojů a jsou v ní obsaženy následující geometrické zkoušky: (7)

- 1) Přímost – Měříme úchylku polohy, lineární úchylku a úhlové úchylky.
- 2) Rovinnost – Měříme rovinnost stolů a desek u obráběcích strojů, pro správné dosednutí součástí.
- 3) Rovnoběžnost – Kontroluje se rovnoběžnost ploch s osami.
- 4) Kolmost – Měříme kolmost dvou ploch, například upínací plochy stolu k vedení stojanu (6)
- 5) Otáčení – Měříme nepřesnost rotačních os.

Norma ISO 240-2 určuje přesnost a opakovatelnost při nastavování polohy numericky řízených os na CNC obráběcích strojích. Testy jsou navrženy k měření relativních pohybů

mezi komponenty stroje, ke kterému je upnutý nástroj a ke kterému je upnutý polotovár. Měříme přímo jednotlivé osy na stroji. Tyto metody použijeme jak na lineární tak na rotační osy. Výrobce nebo prodejce je zároveň zodpovědný za dodání teplotních specifikací, při kterých je stroj schopen pracovat se specifikovanou přesností. Normou, která vymezuje postupy na určení tepelných vlivů na přesnost je ISO 230-3. (8)

Kruhové dráhy nástroje na CNC obráběcích strojích jsou vytvářeny pohybem dvou lineárních os. Tyto pohyby jsou ovlivněny geometrickými úchyly, úchyly číslíkového řízení a chyb pohonných strojů. Jako příklad nástroje pro měření kruhovitosti a odchylky kruhovitosti zde můžeme uvést například systém ballbar od společnosti Renishaw, který je schopný proměřit kruhové dráhy a nasnímaná data použít k výpočtu přesnosti v souladu s normou ISO 230-4. (7)

Pokud se chcete dozvědět další informace, nebo více podrobností o měření přesnosti obráběcích strojů můžete se podívat přímo na normu ISO 230, která má další části, ale ty pro nás nejsou v této práci tak zajímavé.

2.5 Souřadnicové měřicí stroje

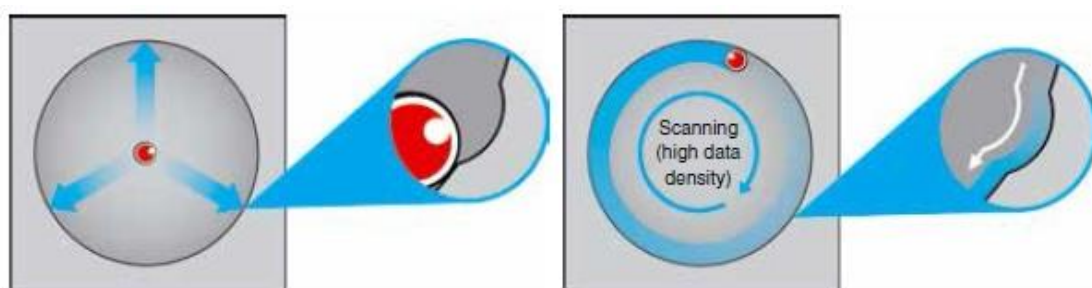
Souřadnicový měřicí stroj ve zkratce SMS, je zařízení, které slouží k měření rozměrových a geometrických charakteristik modelu. Princip souřadnicového měření spočívá ve stanovení základního bodu v prostoru a polohy dalších bodů na měřené součásti měříme formou souřadnicových rozměrů v osách X, Y a Z. Bod popisujeme v nepohyblivém například kartézském souřadnicovém systému, který je pevně spojen s rámem stroje. Z naměřených bodů vypočítáváme jednotlivé geometrické prvky. Z těchto geometrických prvků jsme schopni vyjádřit charakteristiky rozměrové a geometrické tolerance. Pro definování geometrických prvků využíváme modely konstruované pomocí CAD programů. Tyto data použijeme při přípravě řídicích programů pro měření na SMS. Pro vlastní měření můžeme využít buď dotykové snímací systémy, nebo bezdotykové. (9)

Existují různé konstrukce souřadnicových měřicích strojů, které jsou určeny převážně velikostí měřené součásti:

- Konzolový SMS
- Mostový SMS
- Portálový SMS

- Sloupový SMS s vodorovným ramenem

Kdybychom chtěli porovnat měření na souřadnicových měřicích strojích a CNC obráběcích strojích z hlediska přesnosti, dojdeme k závěru, že větší přesnosti takřka vždy dosáhne souřadnicový měřicí stroj a to díky několika faktorům: konstrukce SMS; větší tuhost používaných materiálů; menší teplotní roztažnost a také jsou umístěné ve specializovaných klimatizovaných pracovištích. Také už z vlastních používaných strategií měření na obráběcích strojích vidíme, že zde vzniká nepřesnost z důvodu malého počtu měřených bodů. Princip tohoto měření vidíme na Obrázku Obr. 1 - Porovnání strategií měření podle počtu měřených bodů. Při měření na obráběcím stroji se může na obrobku nacházet řezná kapalina popřípadě špony, což přispívá ke zhoršení kvality měření.



Obr. 1 - Porovnání strategií měření podle počtu měřených bodů (10)

2.6 Sondy pro ustavení a měření obrobku na obráběcím stroji

Dotyková sonda je ve své podstatě důmyslný spínač s hrotem, který se aktivuje při kontaktu s povrchem součásti a poskytne nám geometrická data. Geometrická data jsou ve formě polohy dotykového hrotu, která je přenesená do řídicího systému stroje, buď pomocí kabelu, optického přenosu, nebo rádiového přenosu. (11) (12)

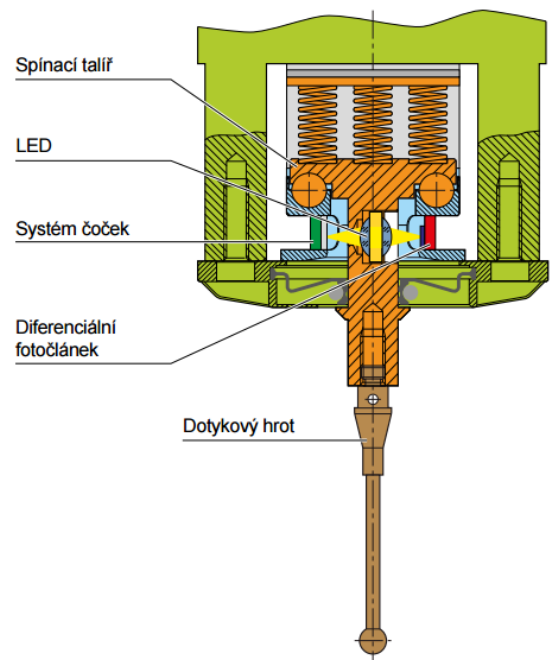
Optický přenos funguje na principu přenesení informací pomocí infračerveného světla do řídicí jednotky. Tyto systémy, ale vyžadují přímou viditelnost mezi sondou a přijímačem, jejich přenosová vzdálenost je 6 metrů, což je činí vhodné především pro menší až střední stroje bez složitých přípravků. (11)

Rádiový přenos signálu využívá rádiové vlny k přenesení informací ze sondy do přijímače. Přenos probíhá ve speciálně vyhraněném frekvenčním pásmu, aby nedocházelo k rušení, a také obsahuje specifické identifikátory, což umožňuje práci několika těchto systémů ve vzájemné blízkosti. Zde již není vyžadována přímá viditelnost mezi sondou a přijímačem a dosah se pohybuje okolo 15 metrů, proto tyto systémy můžeme využívat i na velkých portálových obráběcích strojích. (11)

V dnešní době se využívá velké množství nejrůznějších principů, na kterých fungují obrobkové sondy, a každá firma většinou používá vlastní. Některé základní principy jsou ale u všech výrobců velmi podobné například tříbodové uložení snímacích systémů, které vychází z matematického určení roviny právě třemi body. Nejčastěji se setkáme se sondami, které pracují s optickým, kinematicky odporovým, nebo tenzometrickým senzorem. (12)

2.6.1 Sondy s optickým senzorem

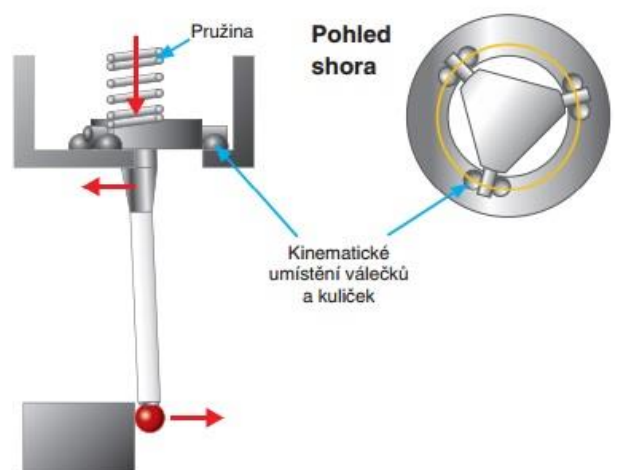
Tento typ sondy pracuje s jednou LED diodou, ze které vycházejí světelné paprsky následně zaostřeny pomocí soustavy čoček. Světelný paprsek následně dopadá jako světelný bod na diferenciální fotočlánek. Při vychýlení hrotu sondy vytvoří fotočlánek spínací signál. Dotykový hrot je pevně spojen se snímacím talířem a usazen v trojbodovém uložení. Senzor pracuje na bázi bezdotykového optického snímače, což zajišťuje nulové opotřebení, a tudíž mají dlouhou životnost. Princip této sondy je zobrazen na obrázku Obr. 2 - Sonda s optickým senzorem



Obr. 2 - Sonda s optickým senzorem (13)

2.6.2 Sondy s kinematickým odporovým senzorem

Hrot sondy je umístěn na tři rovnoměrně rozmístěné válečky, které sedí na šesti kuličkách zajišťující šestibodový kontakt v kinematickém uložení. Těmito kontakty prochází elektrický proud. Mechanismus je pružně fixován, což umožňuje vychýlení, jakmile se dotek sondy dotkne obrobku, a zároveň umožňuje sondě vrátit se do stejné polohy. Pružina tlačí na těleso trnu sondy, čímž vznikají styčné plochy mezi kuličkami a válečky, kterými prochází elektrický proud. Když dojde ke kontaktu doteku s obrobkem, válečky se vzdálí od

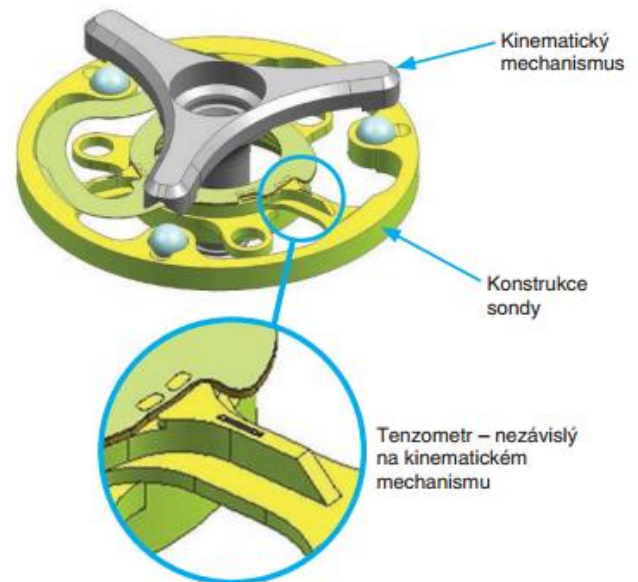


Obr. 3 - Sonda s kinematickým odporovým senzorem (14)

kuliček a zmenší se velikost styčných ploch, to zvýší elektrický odpor. Po dosažení prahové hodnoty se sonda rozeprve kontakt. Princip této sondy je zobrazen na obrázku Obr. 3 - Sonda s kinematickým odporovým senzorem

2.6.3 Tenzometrické sondy

Jedná se o velmi přesný druh snímače, tyto sondy jsou určeny pro náročné aplikace, kde je potřeba zajistit co největší přesnost. Využívají technologie křemíkových tenzometrů s přesnou mechanickou konstrukcí. Tenzometry jsou v tomto případě uloženy na vlastní konstrukci sondy oddělené od kinematického mechanismu. Tenzometry jsou umístěny tak, aby snímaly napětí ve všech osách. Po dosažení prahových



Obr. 4 - Tenzometrická sonda (14)

hodnot je vytvořen snímací signál. Díky tomu, že snímání je zcela nezávislé na vlastním mechanismu sondy, je zde velmi malá spínací síla, což zajišťuje vysokou opakovatelnost a konzistentní snímací charakteristiku. Princip této sondy je zobrazen na obrázku Obr. 4 - Tenzometrická sonda

3. Měřicí cykly v řídicích systémech

Dnes již takřka všechny moderní řídicí systémy, ale i některé starší v sobě mají obsaženy měřicí cykly pro ustavení a kontrolu obrobku. Stačí tedy stroj dovybavit pouze měřicí sondou a do určité míry můžeme používat CNC obráběcí stroj jako souřadnicový měřicí stroj.

V této kapitole popisují principy a možnosti, které nám tyto cykly nabízí, včetně základních informací o souřadných systémech vlastního CNC obráběcího stroje a základní body, od kterých se tyto souřadné systémy ustavují.

3.1 Důležité body při frézování na CNC strojích

Při obrábění se používá několik referenčních bodů. Jedná se o nulový bod stroje a nulový bod obrobku, od nich se odvíjí i souřadné systémy v závislosti na prováděném měření. Například je jednodušší ověřovat délku nástroje v souřadném systému stroje, ale rozměry

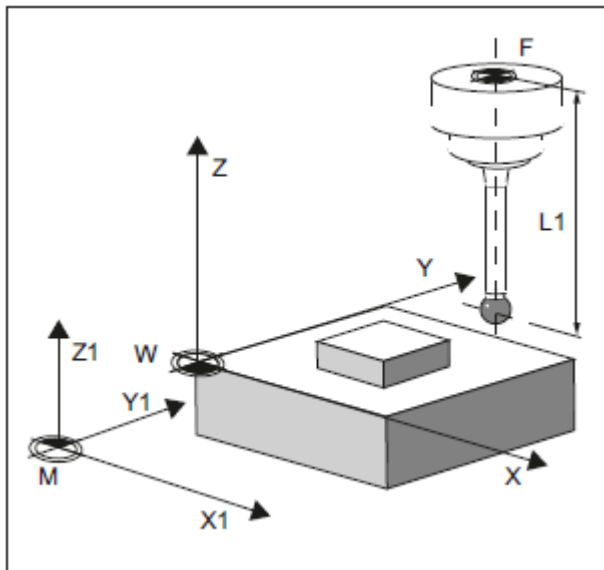
polotovaru jsou měřeny v souřadném systému obrobku. Na následujícím obrázku Obr. 5 - Referenční body při obrábění můžeme vidět rozmístění referenčních bodu při frézování, kde:

(15)

M = nula stroje v souřadném systému stroje

W = nula obrobku v souřadném systému obrobku

F = referenční bod nástroje



Obr. 5 - Referenční body při obrábění (15)

Nulový bod stroje je stanoven výrobcem. Tento bod je výchozí pro všechny další souřadnicové systémy a vztažné body na stroji. Jeho nejčastějším umístění u frézky bývá v krajních polohách stolu frézky, v osách X, Y, Z. Z pohledu obsluhy se bod může nacházet vlevo vpředu. (16)

Pozice referenčního bodu nástroje F je v souřadném systému stroje, jeho skutečná hodnota se odměřuje od nuly stroje M. Pozice špičky upnutého nástroje se, ale odvozuje od souřadného systému obrobku. V případě dotykové měřicí sondy se za špičku uvádí buď konec sondy, nebo střed kuličky sondy. Posunutí polohy počátku součásti (work offset) uvádí polohu nuly obrobku v závislosti na souřadnicovém systému stroje. (15)

Referenční bod stroje je určený výrobcem a nemůžeme ho změnit. Nachází se obvykle v koncových prostorách stroje. Jedná se o pevnou pozici referenčního bodu nástroje umístěného ve vřetenu frézky vzhledem k souřadnému systému stroje, do které stroj najede po

každém zapnutí. Pokud je ve vřetenu upnutý nástroj tak se do referenčního bodu přesune špička nástroje, jejíž hodnota je upravená o korekci daného nástroje. (16)

Bod špičky nástroje se nachází, jak napovídá název na špičce nástroje a použijeme ho pro stanovení délkové korekce nástroje. (16)

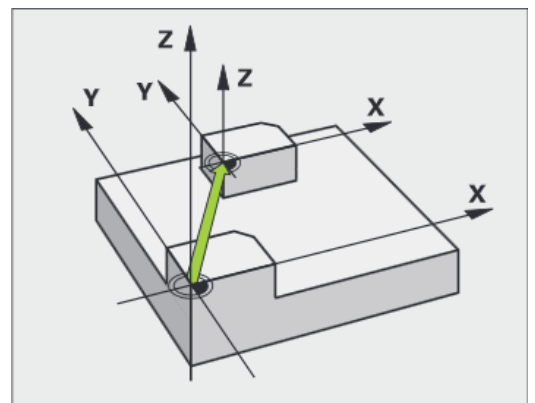
3.2 Tabulka nulových bodů

Nulové body při obrábění se používají jako reference, k níž se vztahují údaje o souřadnicích v obráběcím procesu. Nulový bod může být umístěn obsluhou nebo programátorem do libovolné polohy v obráběcím prostoru, nemusí být pouze jeden. Jak udává název této kapitoly, jedná se o tabulku nulových bodů uloženou ve stroji. (16)

Tabulka nulových bodů je přímo ve stroji a odkazujeme se na ní pomocí DIN/ISO kódů ve formě G54 až G59, nebo pokud jich potřebuji více, než 6 mohu je zadat ve formě G54.1P1 až G54.1P48. Tyto příkazy v sobě mají zadefinovanou přesnou polohu nulových bodů a lze se na ně odkazovat v NC programu. Takto vytvořené body jsou přesně definované pomocí pevného souřadného systému stroje, strojové nuly, která je definovaná v DIN/ISO kódu jako G53.

Před vlastním obráběním je vždy potřeba zjistit vztažný bod obrobku, aby se mohl vytvořit vztah k zadaným souřadnicím v obráběcím programu. Tento vztažný bod na součásti může být stejný bod, jaký jsme si nadefinovali, nebo mu můžeme přímo na stroji nadefinovat souřadnice, které potřebujeme v CAMu při výpočtu obráběcích drah, nebo mu můžeme přímo na stroji nadefinovat souřadnice, které potřebujeme. Toto první určení nulového bodu můžeme provádět buď pomocí nástroje takzvaným naškrábnutím, nebo pomocí dotykové sondy. Při použití nástroje místo sondy, může ale dojít k poškození povrchu, také to není tak přesné jako při použití sondy. (16)

Tyto nulové body jsou vhodné, pokud potřebujeme provádět na jedné součásti více stejných operací, tak nemusíme v CAM softwaru všechny definovat jednotlivě, ale definujeme obrábění pouze jednou a následně pomocí příkazu G52 (posunutí nulového bodu) můžeme opakovat obrábění na libovolném místě obrobku. Princip této



Obr. 6 - Princip posunutí počátku (17)

operace je znázorněn na obrázku Obr. 6 - Princip posunutí počátku(17)

3.3 Měřicí cykly

Měřicí cykly nejen v řídicích systémech, ale i obecně mají dvě základní funkce:

- 1) Ustavování počátku součásti
- 2) Měření a kontrola obrobene součásti

První bod jsem už částečně popsal v předchozí kapitole, jedná se o nalezení vztažného, nebo nulového bodu, podle kterého jsme schopni vytvořit souřadný systém, ke kterému se vztahují vygenerované dráhy obrábění. Vyhledávání vztažného bodu na součásti ale nemusí být pouze o nalezení rovin XY, XZ a YZ, může se také vztahovat k určitému prvku na součásti, jakým může být například kruhová, nebo obdélníková kapsa, díra popřípadě i jiné tvary.

Druhý bod se zaměřuje na měření přesnosti obrobene popřípadě na kontrolu, zdali bylo obrábění vůbec provedeno, například kvůli poškození nástroje atd.

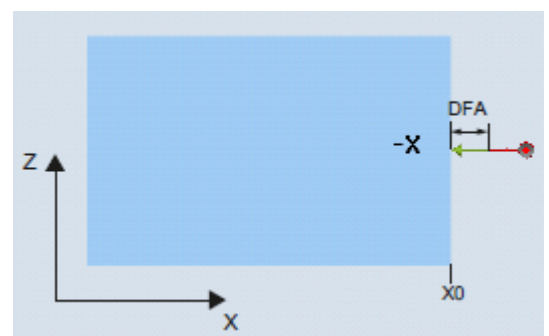
Pro obě funkce měření existuje v řídicích systémech mnoho různých strategií v závislosti na druhu řídicího systému, typu používaného stroje atd. Ačkoli u každého výrobce řídicího systému jsou obvykle jiná specifika definování těchto strategií jejich princip a výstup je většinou velmi podobný u některých typových strategií. Samozřejmě existují pro každého výrobce určité specifické strategie, ale ve většině případů jsou si měřicí cykly velmi podobné.

3.4 Nejčastěji používané měřicí cykly pro zjišťování vztažných bodů

V této kapitole jsem se zaměřil na uvedení těch nejčastěji používaných strategií ke zjištění vztažných bodů na součásti, se kterými se můžeme setkat ve většině řídicích systémů od různých výrobců.

3.4.1 Měření vztažné roviny

Jedná se o nejjednodušší prováděné měření. Tímto způsobem měříme roviny popřípadě hrany na obrobku rovnoběžné se základními rovinami XY, XZ, YZ. Předpokládáme zde tedy, že obrobek není nikterak natočený, neboť měření

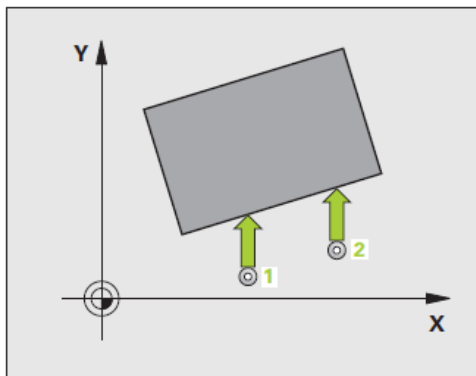


Obr. 7 - Měření vztažné roviny (15)

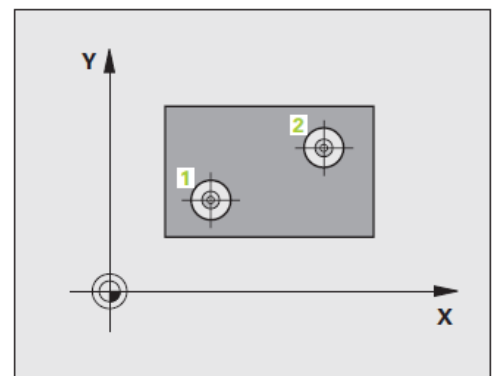
se provádí nejčastěji pouze jedním dotykem sondy ve směru X, Y, Z jak je znázorněno na obrázku Obr. 7 - Měření vztažné roviny (15)

3.4.2 Zjišťování šikmé polohy obrobku

Jak již napovídá název této kapitoly, zde uvádím několik strategií pro měření šikmo posazeného obrobku. Šikmost můžeme měřit pomocí dvou doteků sondy na jedné ploše obrobku, znázorněné na obrázku Obr. 9 - Měření šikmé plochy pomocí dvou bodů, nebo změřením dvou elementů na obrobku jakými mohou být například dvě díry popřípadě výstupky, znázorněné na obrázku Obr. 8 - Měření natočení součásti pomocí dvou elementů (17).



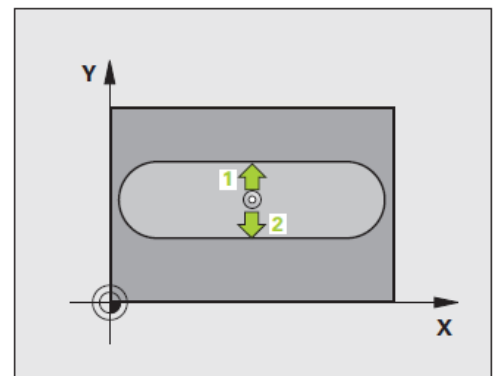
Obr. 9 - Měření šikmé plochy pomocí dvou bodů (17)



Obr. 8 - Měření natočení součásti pomocí dvou elementů (17)

3.4.3 Měření vztažného bodu ve středu drážky

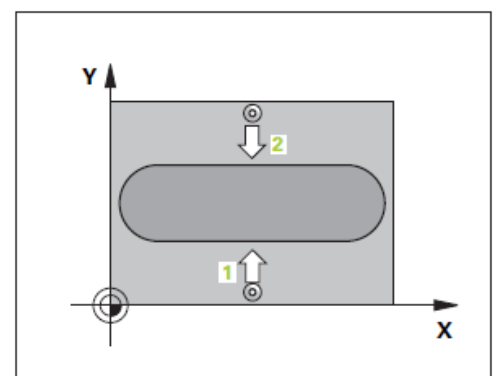
Tento cyklus dotykové sondy zjistí střed drážky na součásti a nastaví tento střed jako vztažný bod. Tomuto bodu můžeme také přiřadit přesné souřadnice. Jak ale vidíme z obrázku znázornění principu měření středu drážky tak změřená hodnota je pouze ve směru osy Y. Tuto strategii tedy většinou kombinujeme s dalšími pro zvýšení přesnosti. (17)



Obr. 10 - Určení vztažného bodu ve středu drážky (17)

3.4.4 Měření vztažného bodu ve středu výstupku

Tento cyklus je velmi podobný cyklu pro měření středu drážky ale s tím rozdílem, že zde měříme výstupek. Vztažný bod je v tomto případě nastaven do středu výstupku. Jak ale vidíme z obrázku znázornění principu měření středu drážky tak změřená hodnota je pouze ve směru osy Y. Tuto strategii tedy

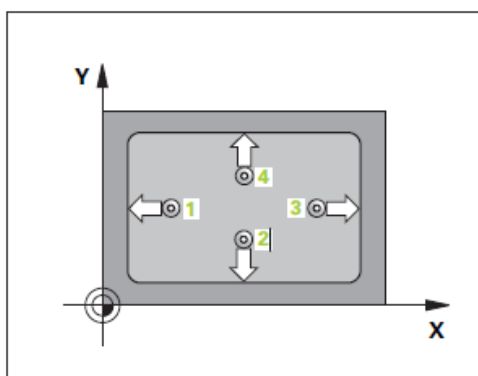


Obr. 11 - Princip měření středu výstupku (17)

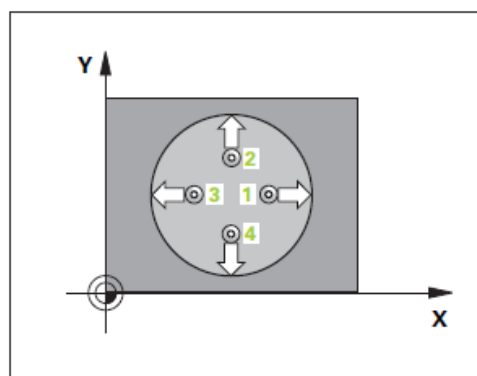
většinou kombinujeme s dalšími pro zvýšení přesnosti. (17)

3.4.5 Měření vztažného bodu ve středu kapes na součásti

Tyto cykly se používají pro určení vztažného bodu ve středu obdélníkové kapsy, nebo ve středu kruhové kapsy. Jak je patrné z obrázků Obr. 12 - Měření vztažného bodu ve středu obdélníkové kapsy Obr. 13 - Měření vztažného bodu ve středu kruhové kapsy, střed je měřen v ose X i v ose Y, což je velká výhoda oproti cyklům měření drážky.



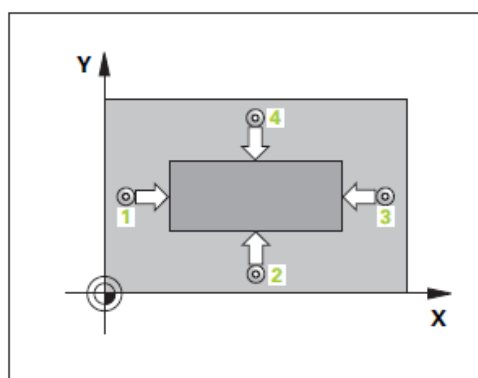
Obr. 12 - Měření vztažného bodu ve středu obdélníkové kapsy (17)



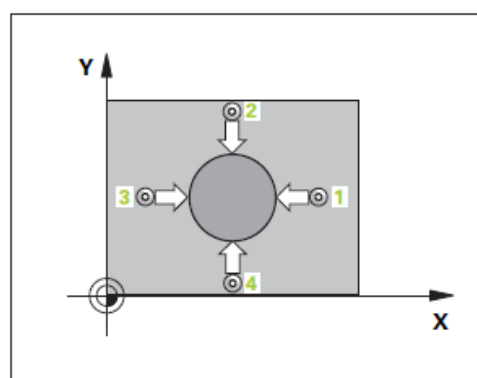
Obr. 13 - Měření vztažného bodu ve středu kruhové kapsy (17)

3.4.6 Měření vztažného bodu ve středu obdélníkové nebo kruhové geometrie

Tyto cykly se používají pro určení vztažného bodu ve středu obdélníkové geometrie, nebo ve středu kruhového výstupku na součásti. Jak je patrné z obrázků Obr. 15 - Měření vztažného bodu ve středu obdélníkové geometrie Obr. 14 - Měření vztažného bodu ve středu kruhové geometrie střed je měřen v ose X i v ose Y, což je velká výhoda oproti cyklům měření drážky.



Obr. 15 - Měření vztažného bodu ve středu obdélníkové geometrie (17)



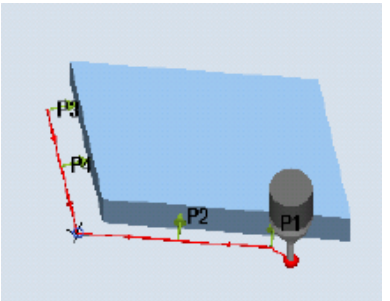
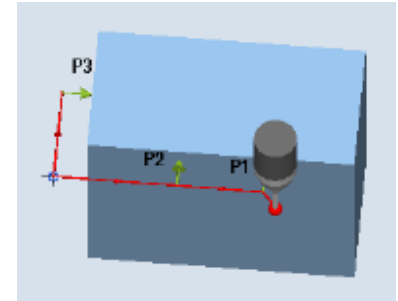
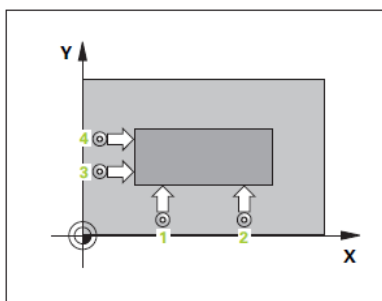
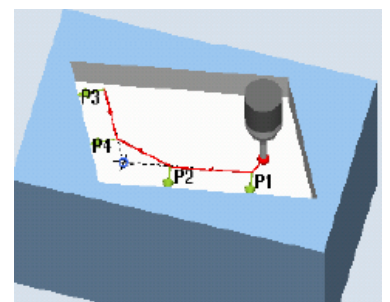
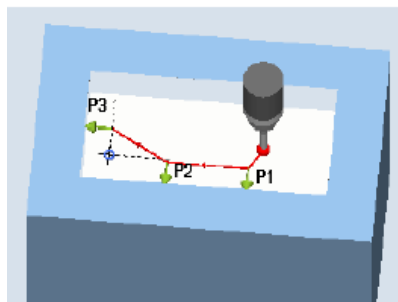
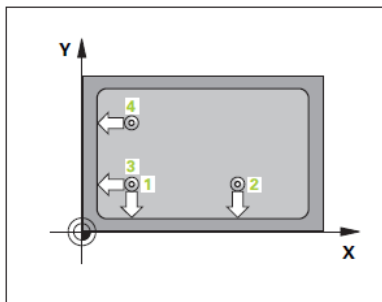
Obr. 14 - Měření vztažného bodu ve středu kruhové geometrie (17)

3.4.7 Měření vztažného bodu vnitřních rohů součásti

Pro proměření rohů existuje několik strategií v závislosti na požadované přesnosti a na tvaru součásti. Tyto cykly pracují na principu proměření buď dvou přímkou, nebo přímkou a

bodů. U některých výrobců jako je společnost Siemens máme možnost v těchto cyklech rovnou nastavit, zdali se jedná o měření vnějších či vnitřních rohů, naopak někteří výrobci jako společnost HeidenHain tyto cykly dělí do dvou. Princip a definice je ale natolik podobný, že je v tomto případě všechny zahrnuji do stejné skupiny.

- 1) Měření rohu dvou stran, které spolu svírají pravý úhel pomocí 3 bodů. Znázorněné na obrázku Obr. 18.
- 2) Měření rohu dvou stran, které spolu svírají pravý úhel pomocí 4 bodů. Znázorněno na obrázku Obr. 16.
- 3) Měření rohu dvou stran, které spolu svírají libovolný úhel. Znázorněno na obrázku Obr. 17



Obr. 16

Obr. 18

Obr. 17

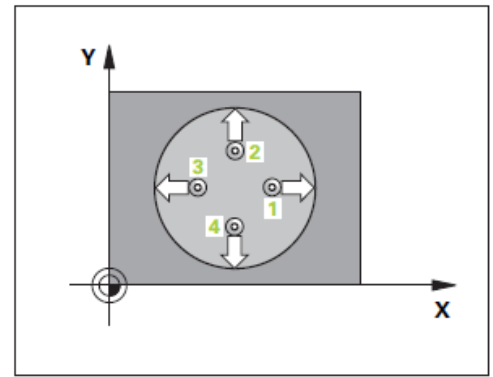
3.5 Nejčastěji používané cykly pro kontrolu obrobků

Měřicí cykly pro kontrolu obrobku jsou ve své podstatě velmi podobné cyklům pro vyrovnání součásti a proměrování vztažných bodů, mají ale jiný výstup. V tomto případě už neupravujeme podle naměřených hodnot nulové body a posléze souřadnicový systém, ale výstupem jsou kontrolní data, která nám udávají, do jaké míry se obrobek shoduje s požadavky.

3.5.1 Proměřování díry

Tento cyklus dotykové sondy provede měření a zjistí střed a průměr díry popřípadě kruhové kapsy. Pokud tomuto cyklu nadefinujeme i cílové parametry provede s nimi srovnání a uloží skutečné hodnoty a odchylky do systémových parametrů. (17)

Princip měření je znázorněn na obrázku Obr. 19 - Proměřování díry

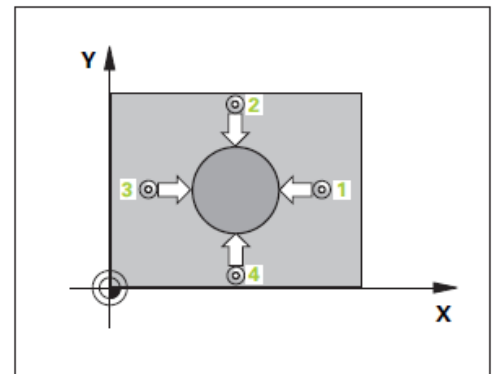


Obr. 19 - Proměřování díry (17)

3.5.2 Proměřování kruhové geometrie na součásti

Tento cyklus pracuje na podobném principu jako cyklus proměřování díry s tím rozdílem, že nyní měříme kruhový výstupek. Opět jsme schopni naměřené hodnoty průměru, středu a jejich odchylek uložit a porovnat je se zadanými hodnotami. (17)

Princip měření je znázorněn na obrázku Obr. 20 - Proměřování kruhové geometrie



Obr. 20 - Proměřování kruhové geometrie (17)

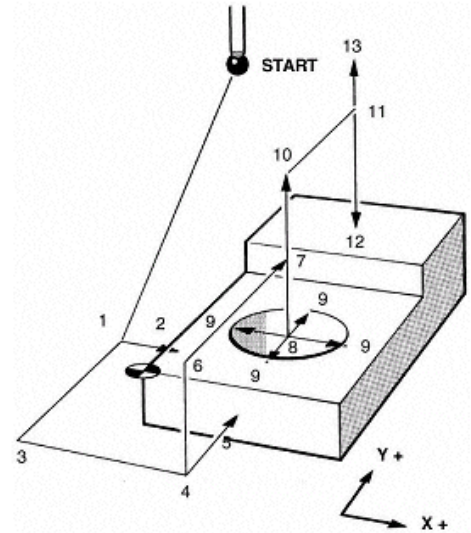
3.6 Plug-in EasyProbe

Pro stroje, jejichž řídicí systémy neobsahují měřicí cykly a cykly pro určování vztažných bodů přišla společnost Renishaw s možností rozšíření řídicího systému ve formě souboru měřících maker EasyProbe. Toto rozšíření nám umožní přímo na stroji programovat jednoduché měřicí cykly pro proměřování součástí a ustavení vztažných bodů, které slouží k posunutí souřadného systému. (18)

Tento soubor měřících maker nám ale také rozšíří celý řídicí systém o možnost použití obrobkových sond. Tento fakt je velmi důležitý, neboť i pokud máme například měřicí modul pro CAM software, na kterém generujeme NC program pro ovládání CNC obráběcího stroje, nelze provádět jakékoliv měření na stroji, jehož řídicí systém neumí pracovat s měřicí sondou, protože všechny tyto funkce jsou závislé na speciálních chráněných pohybech dotykové sondy. (18)

Výstupem tohoto programu, ale už nejsou vygenerované dráhy, které se pomocí postprocesoru převedou na NC program ale přímo NC program, který je vložen do námi vygenerovaného programu, který byl nahrán do obráběcího stroje.

Program EasyProbe umožňuje provádět měření i na strojích jejichž řídicí systémy tuto funkci samy o sobě obsahují. Jsme tedy schopni například používat: měření natočení šikmé plochy pomocí dvou bodů; měření povrchu jedním dotekem; měření kruhové kapsy, nebo výstupku; měření šířky drážky. Funkce měření šikmé plochy pomocí dvou bodů, může být také využita, ke kontrole 4té osy pokud jí stroj obsahuje. O kompletních funkcích tohoto systému na CNC frézkách vypovídá obrázek Obr. 21



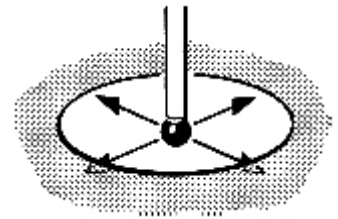
- Možnosti maker EasyProbe

Obr. 21 - Možnosti maker EasyProbe (18)

Příklad výstupu ze softwaru EasyProbe, může být nastavení vztažného bodu pro určení nulového bodu ve středu kruhové kapsy, jak vidíme na obrázku Obr. 22 - Střed kapsy

G56 P9023 D50. S56. I50. J50.

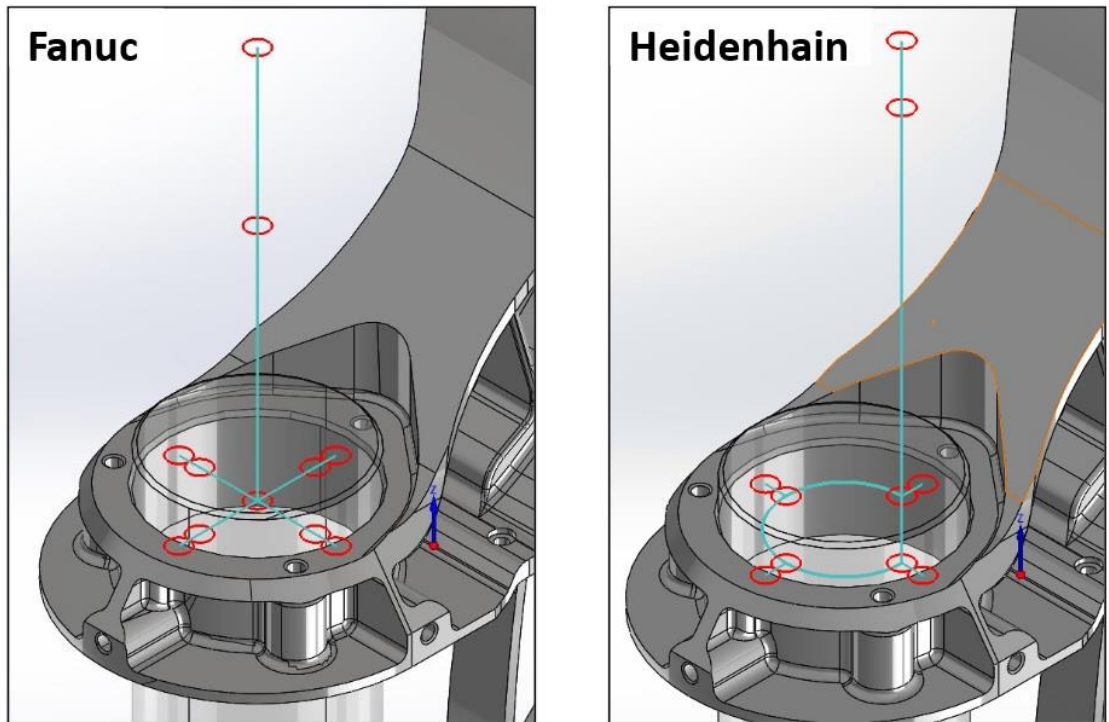
(19)



Obr. 22 - Střed kapsy (19)

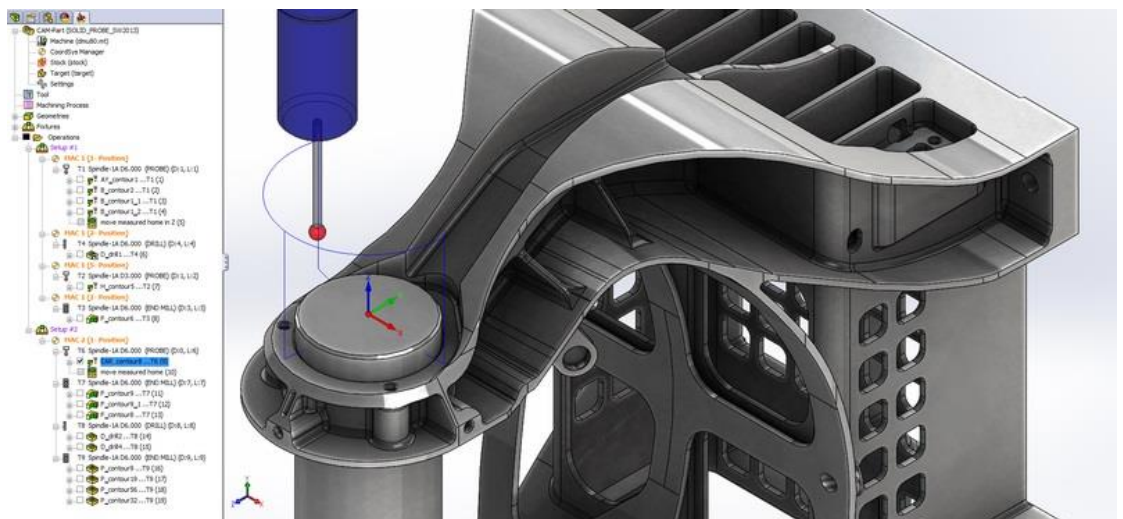
4. Měřicí cykly v CAM

U většiny softwarů se nesetkáme s přímou integrací měřících cyklů do CAD/CAM programů, ale pouze s doplňkovými plug-iny, nebo moduly často od vývojářů třetích stran. Všechny tyto měřicí doplňky pro CAM software jsou ale závislé na řídicím systému stroje, neboť pokud řídicí systém není vybaven, nebo doplněn makrem pro práci s měřicí sondou, měření nelze provést, protože stroj nebude schopen přečíst a vykonat vygenerovaný NC program. Dále se také liší měřicí dráhy podle řídicího systému, pro který jsou programované, neboť každý výrobce je má specifikované jinak.



Obr. 23 - Rozdílné dráhy sondy podle řídicího systému (20)

Příkladem těchto doplňků může být modul Solid Probe, jako doplněk SolidCAMu. Tento modul nám nabízí základní funkce práce s dotykovou obrobkovou sondou, například proměrování obrobku za využití předdefinovaných cyklů; změření nulového bodu obrobku; kontrola obrobení; kontrolu nástroje a další. Příklad tohoto doplňku si můžeme prohlédnout na obrázku Obr. 24 - Ustavení obrobku podle kruhové geometrie

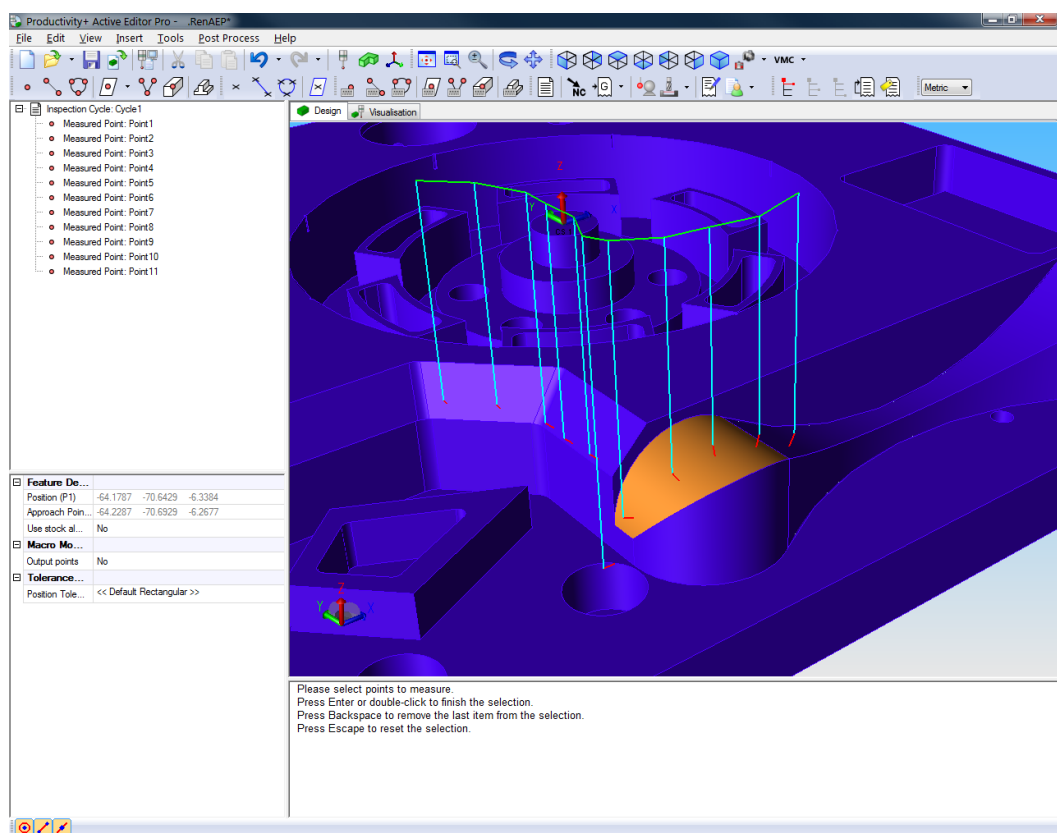


Obr. 24 - Ustavení obrobku podle kruhové geometrie (20)

Jednou z nejvýznamnější společností, která se zabývá problematikou výroby a ovládání sond pro měření na CNC obráběcích strojích je firma Renishaw. Tato firma přišla

s řešením v podobě softwaru Productivity+, které nabízí vložení jak jednoduchých měřících funkcí, tak i složitější měřící cykly přímo do výrobního programu. Tento program zjednodušuje celou řadu měřících a kontrolních operací, jako například kontrolu obrobení, nebo aktivní kontrolu, která nám umožňuje provádět změnu řezných parametrů v závislosti na výsledcích kontroly. (21)

Productivity+™ můžeme používat jako nezávislý program ve formě Active Editor Pro, kde můžeme definovat operace měřící sondy v prostředí, které se velmi podobá CAD/CAM softwarům. Nevýhoda tohoto řešení je zde ale s importem 3D modelu součásti. (21)



Obr. 25 - Ukázka pracovního prostředí programu Produktivity+ (21)

Dále můžeme Productivity+™ využívat ve formě plug-inu pro vybrané CAM softwary jako je například MasterCAM. (21)

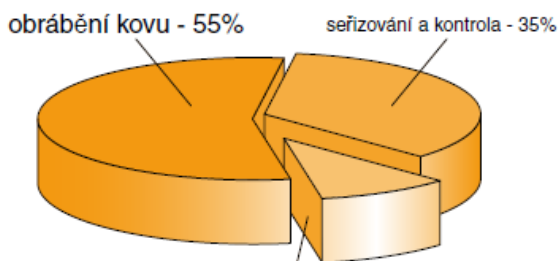
5. Parametry, cíle, použité vybavení a software pro vypracování

Jak jsem již nastínil, při obráběcím procesu můžeme sondy využít hned v několika oblastech: přípravy výroby, seřízení stroje a ustavení výrobku, mezioperační kontrola a výstupní celková kontrola.

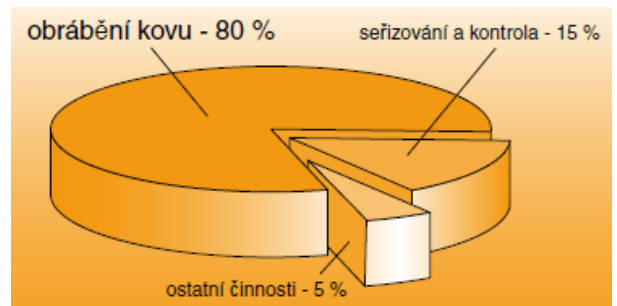
Ve své praktické části se zaměřuji na druhý bod tedy přesněji na ustavení obrobku při upnutí do stroje a převedení celé této činnosti z CNC obráběcího stroje, kde bychom ztráceli drahý výrobní čas do přípravy výroby, která probíhá v kanceláři u počítače, což nám také nabízí více času pro volbu správné strategie ustavení.

Pokud nepoužíváme sondy k ustavení obrobku a potřebujeme obrábět přesně, zbývá nám obvykle pouze jediná možnost, použít přesné a drahé upínače. K vyrovnání součásti se v tomto případě většinou používá číselníkový úchylkoměr. Zkušený operátor stroje je tímto způsobem schopen upnout obrobek relativně přesně, ale i tak to zabírá velké množství času a to hovoříme pouze o zkušeném operátorovi. Tato metoda je velmi náchylná právě k chybě operátora, tudíž upnutí trvá dlouho a stále nedosáhneme požadované přesnosti upnutí.

Použití dotykových sond pro ustavení obrobků je velkým přínosem v této oblasti, umožňuje obsluhu relativně velmi rychle a jednoduše nastavit požadovaný nulový bod s vysokou přesností. Tím jsme schopni velmi redukovat čas potřebný pro upnutí obrobku, jak ukazují i výsledky provedené studie od firmy Renishaw, která se zabývá právě výrobou sond pro obráběcí stroje. Jak můžeme vidět na obrázcích Obr. 27 - Struktura výrobního času bez použití sond a Obr. 26 - Struktura výrobního času s použitím sond došlo k výrazné úspoře ve výrobním čase ve formě seřizování.



Obr. 27 - Struktura výrobního času bez použití sond (25)



Obr. 26 - Struktura výrobního času s použitím sond (25)

Cílem mé práce je umožnit programování ustavovacích cyklů přímo v CAM softwaru GibbsCAM při přípravě obráběcího procesu. Společnost Renishaw sice nabízí příručku, jak lze snadno ručně napsat část NC programu určenou pro měření ručně, ale zde je velká šance výskytu chyb, také bychom takto museli doplňovat každý NC kód, který vygenerujeme. Dále máme možnost použít například software Productivity+™ pro programování drah sond a jejich simulaci, ale vzniká zde další práce spojená s přenosem modelu a propojováním dat z CAM softwaru.

Stroj, pro který jsou nová makra určena, je vybaven řídicím systémem Fanuc 0iMD. Tento řídicí systém ve své základní verzi neobsahuje měřicí ani ustavovací cykly, z tohoto

důvodu musel být nejdřív rozšířen o Plug-in EasyProbe od společnosti Renishaw, který nám umožňuje provádění základních měřících a ustavovacích cyklů.

5.1 Parametry pro vypracování

Požadavek na doplnění CAD/CAM softwaru GibbsCAM o možnost ustavení počátku součásti, které by bylo možno následně přímo generovat v NC kódu vzešlo od jednoho z našich zákazníků, společnosti Jaromír Biem – Kovovýroba. Byly stanoveny požadavky na typ a možnosti měřících cyklů:

1. Možnost měření v ose X
2. Možnost měření v ose Y
3. Možnost měření v ose Z
4. Možnost měření ve 3 osách
5. Možnost určení vztažného bodu do středu díry

5.2 Popis stroje a řídicího systému

Stroj, pro který byla určena daná makra, je tříosé vertikální obráběcí centrum od společnosti Feeler. Jedná se konkrétně o stroj Feeler VMP-45(A) znázorněné na obrázku Obr. 28 - Feeler VMP-45(A) s následujícími parametry:

- Rozměry stolu: 1 200 x 600 mm
- Rozsahy v jednotlivých osách: 1 100 x 610 x 600 mm
- Rychlost vřetena: 10 000 otáček za minutu
- Rozměry stroje: 3 180 x 2 750 x 2 900 mm



Obr. 28 - Feeler VMP-45(A)

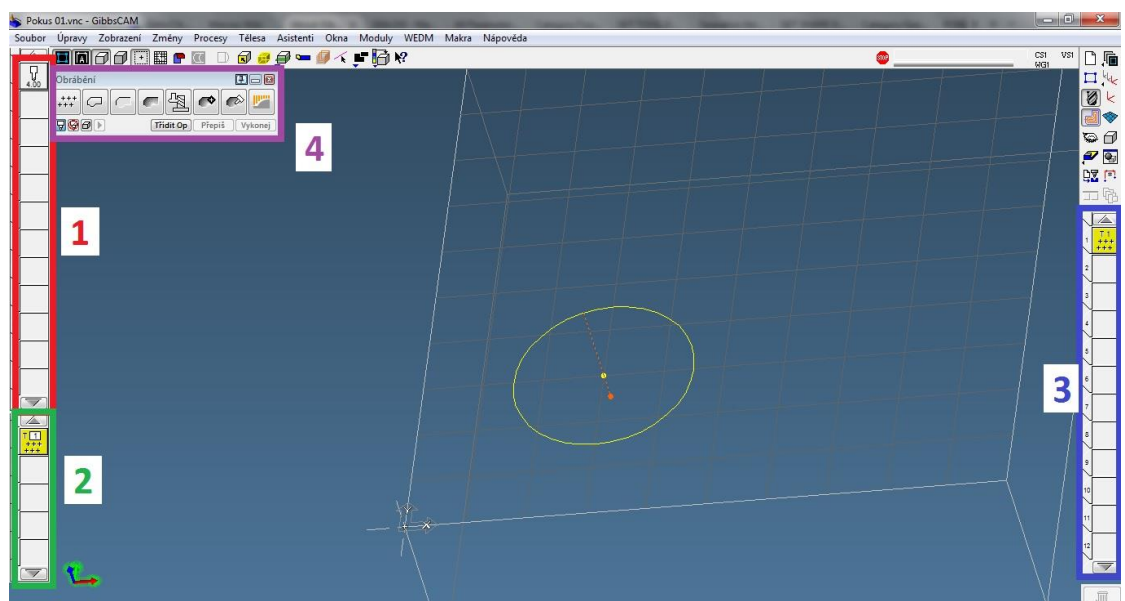
Řídicí systém tohoto stroje je Fanuc 0i-MD, který je určen pro řízení obráběcích strojů s maximálním počtem pěti os, nebo čtyř plynule řízených. Ovládací panel je znázorněn na Obr. 29 - Ovládací panel Fanuc 0i-MD. Tento systém neobsahuje žádné měřicí cykly, tudíž musí být o tuto možnost doplněn. Tento doplněk se nazývá EasyProbe a je od výrobce obrobkových sond Renishaw. Plug-in EasyProbe nám umožňuje provádět ustavování a měření součástí pomocí speciálních G-kódů, které generujeme v NC programu. Jako příklad zde můžeme uvést G-kód P9770, který zajišťuje bezpečnostní posuv, nebo P9023, který podle zadaných parametrů provádí měření.



Obr. 29 - Ovládací panel Fanuc 0i-MD

5.3 GibbsCAM

Pro programování měřících maker používám CAD/CAM software GibbsCAM od firmy 3D Systems. Jedná se o velmi dobře hodnocený software pro programování počítačem podporovanou výrobu, neboli CAM, který pracuje na platformě PC pro programování počítačem ovládaných obráběcích strojů neboli CNC. Jedná se o program, který se skládá z modulů, které nám umožňují si ho rozšiřovat podle potřeby. Tudíž lze tento software používat pro jednoduché obrábění například 2,5 D a následným rozšířením plynule přejít na komplexnější obrábění, popřípadě specializované funkce jako je souvislé obrábění na stroji se dvěma nástrojovými hlavami nebo vřeteny.



Obr. 30 - Ukázka uživatelského rozhraní GibbsCAMu

Rozhraní systému GibbsCAM, které je znázorněno na obrázku Obr. 30 - Ukázka uživatelského rozhraní GibbsCAMu, je tvořeno panely, které můžeme přizpůsobovat podle potřeby. Zde je pro ukázkou zobrazeno několik panelů, abychom si mohli udělat lepší představu o práci v tomto softwaru.

1. Panel nástrojů
2. Panel vytvářených operací
3. Panel vygenerovaných operací
4. Okno strategií obrábění

Podle obrázku *Uživatelské rozhraní GibbsCAMu* se může zdát, že oproti jiným programům jak může být PowerMILL od firmy Delcam, nebo NX od firmy Siemens PLM softwares, má poněkud neobvyklé ovládání. GibbsCAM je navrhován tak aby byl co nejjednodušší k použití právě pro technology obrábění. Toto na první pohled neobvyklé programovací rozhraní bylo tvořeno tak aby ovládání bylo co možná nejintuitivnější. Jednotlivé operace můžeme snadno vytvářet přetažením nástroje na příslušnou strategii obrábění. (22)

5.3.1 Možnosti programování maker v softwaru GibbsCAM

Společnost 3D Systems je poměrně vstřícná k možnosti uživatelskému rozšiřování softwaru GibbsCAM a komukoliv s platnou licencí dovoluje, aby si upravil a případně doplnil program o funkce, které potřebuje vytvářet prostřednictvím maker. Samozřejmě při použití upravených nebo vlastních maker na sebe bereme veškerou záruku jejich používání. Za takto vytvořené doplňky nám nikdo ze společnosti 3D Systems neručí.

Programovací jazyk, který používá GibbsCAM pro makra nám umožňuje vytvářet geometrii, nástroje a operace. Původně sloužil k vytváření nových dílců. Poskytoval uživateli metodu k vložení sady rozměrů pro tvorbu dílu. Následně byl ze zadaných rozměrů vytvořen tvar. (24)

Makra v softwaru GibbsCAM jsou složena z textových souborů, které jsou vykonávány přímo za chodu, tudíž nevyžadují žádnou kompilaci. Jazyk, který se v těchto textových souborech používá, je podobný programovacímu jazyku Basic. (24)

Jelikož se jedná o software vyvíjený ve Spojených státech, všechny soubory makra musí být napsány pouze znaky podle znakové sady ASCII, neboli American Standart Code for Information Interchange. K tvorbě a úpravě souborů můžeme použít textové editory. Námi

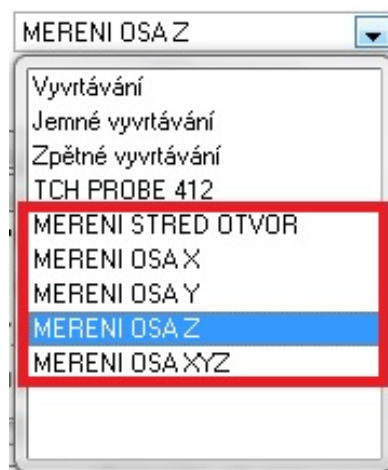
požadované vstupní hodnoty můžeme vložit buď přímo do makra, tato metoda nám ale neumožňuje provádět změny dle potřeby, nebo pomocí uživatelem definovaného dialogu. Tento dialog může obsahovat textové pole, pole pro zadávání hodnot, obrázky, rozklikávací menu a mnoho dalšího. (24)

Pro vlastní tvorbu těchto uživatelských maker bychom ale nejdříve měli mít alespoň základní znalosti programu GibbsCAM, programovací logiky s využitím IF/THEN a cyklů FOR/NEXT. (24)

Pomocí maker si můžeme v GibbsCAMu zjednodušit práci například při časté tvorbě tvarově stejné geometrie, která se liší pouze v určitých parametrech, si můžeme vytvořit makro, které nám bude generovat požadovanou konturu při zadání několika námi zvolených proměnných. Dále jsme schopni takto tvořit například tělesa, operace, nástroje, vlastní vrtací cykly a podobně. (25)

Pro tvorbu maker k ustavování součásti jsem volil makra vlastních vrtacích cyklů. Tento typ makra se pro daný účel nejlépe hodí, neboť vrtání ve své podstatě znamená pouze přesun nástroje do zadané polohy v ose X a Y, následně sjede z bezpečné roviny do požadované hodnoty v ose Z. Když najedeme takto nástrojem nebo v našem případě měřicí sondou do požadované výšky můžeme provést měřicí dotyk a to ve směru jedné, nebo dvou os.

Každá definice vrtacího cyklu, nebo v našem případě ustavovacího cyklu, musí obsahovat pojmenování, které se následně zobrazuje v menu při definici operace vrtání, ukázka s našimi makry je zobrazená na obrázku Obr. 31 - Možnosti definování ustavení. Aby tato funkce správně fungovala a mohli jsme makra správně používat, musíme také upravit soubor MDD. GibbsCAM používá MDD soubory k definování vlastností stroje a jsou v něm uložena veškerá data, která se týkají stroje, například při použití různých druhů chlazení, jako je vysokotlaké, nebo se zde také dají definovat vlastní vrtací cykly a v našem případě cykly pro určování vztažných a nulových bodů.



Obr. 31 - Možnosti definování ustavení

Složku s naším makrem musíme uložit do speciálního adresáře v počítači, odkud si ji načte GibbsCAM. Složka, která obsahuje soubory makra, musí mít stejný název jako ten co jsme definovali v souboru MDD, jinak nebude fungovat.

Složka vlastního vrtacího makra musí obsahovat následující soubory s těmito názvy:

1. DefineData.txt
2. CycleData.mac
3. CycleData.dlg
4. ToolPath.mac

První soubor obsahuje seznam všech proměnných a jejich předdefinovaných hodnot.

Druhý soubor se spustí po kliknutí na tlačítko Data cyklu uživatele v definici vrtací operace a otevře dialog makra.

Třetí soubor je vytvořené makro, obsahující dialog, do kterého se zadávají požadované hodnoty, pro definici cyklu.

Čtvrtý soubor slouží k definování dráhy pro simulaci vytvořeného cyklu v GibbsCAM. Tyto simulace nám umožňují ověření kolizí měřicí sondy s obrobkem.

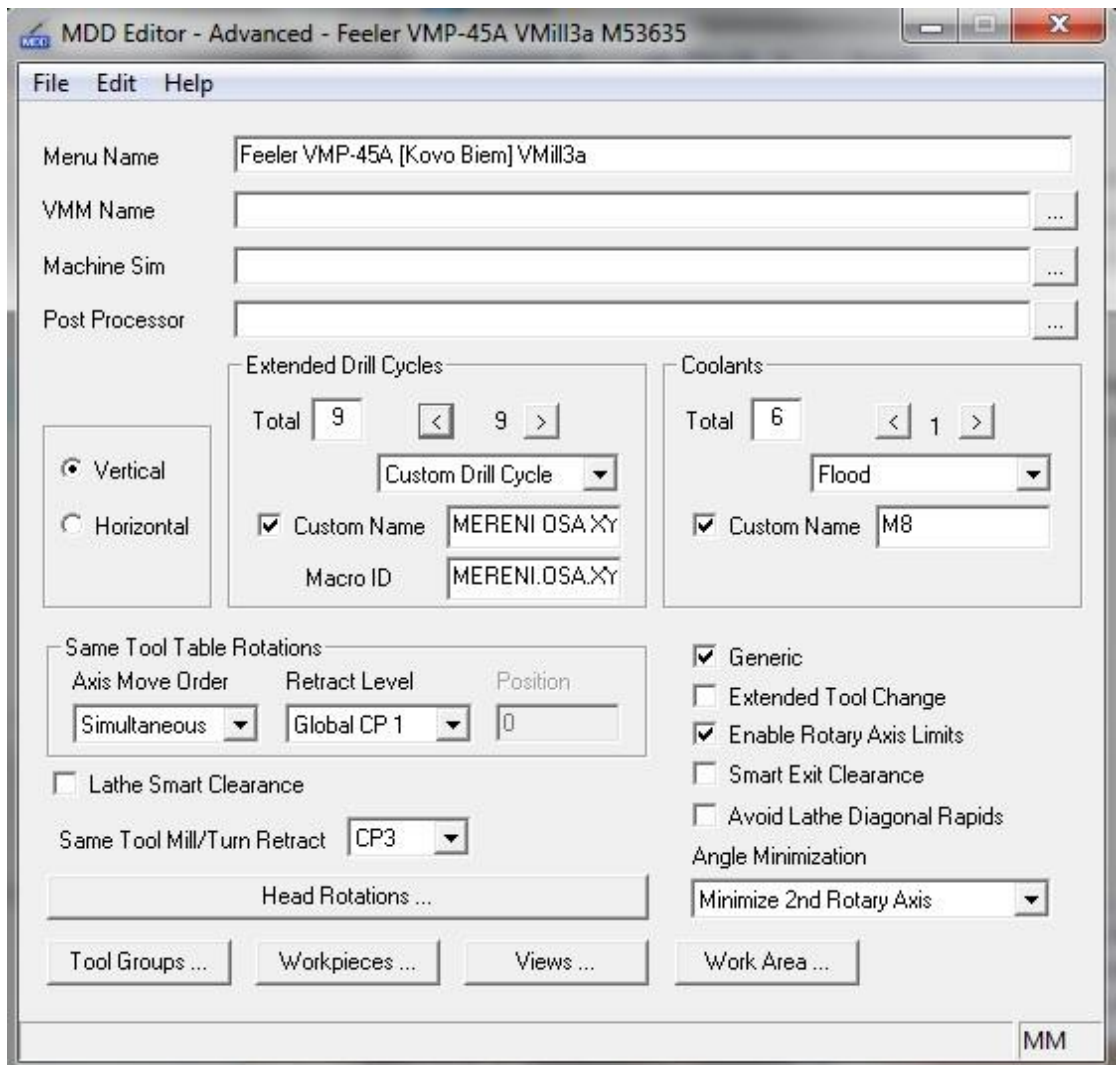
5.3.2 Machine definition document

Machine definition document neboli dokument definice stroje ve zkratce MDD je samostatný soubor s příponou *.mdd, který GibbsCAM využívá k formulování typu a vlastností stroje.

K úpravě tohoto souboru se používá program MDD Editor, v tomto programu jsme schopni definovat zcela nový stroj popřípadě upravit stávající. Dá se zde například vymežit typ a konstrukce stroje, zdali se jedná o frézku, nebo soustruh, popřípadě kolik os je daný stroj schopen využívat. Další věc, která se zde nachází je možnost přidání vlastních cyklů v podobě maker, a to například vrtací v našem případě měřicí cykly, nebo specifické možnosti chlazení stroje.

V našem případě jsme MDD souboru stroje přidali pět nových Vrtacích/Měřicích cyklů. Dále zde zadáme přesný název složky, ve které jsou uloženy soubory makra, a zvolíme jméno, které se zobrazí při definování měřicího cyklu v GibbsCAMu. Nestačí zde ale pouze zapsat název složky, kde je makro uloženo, danou složku musíme nahrát do správného

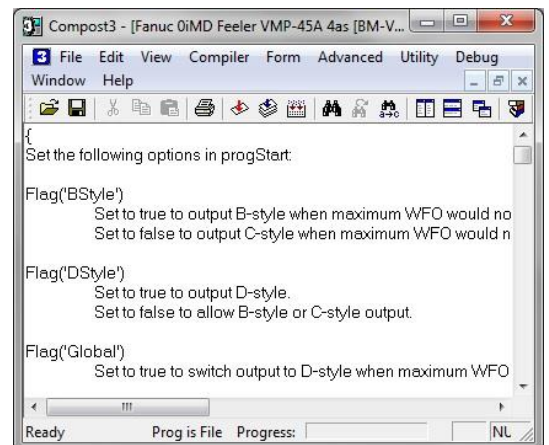
adresáře, odkud si ji načte GibbsCAM. Uživatelské prostředí si můžeme prohlédnout na obrázku Obr. 32 - Uživatelské rozhraní v MDD editoru.



Obr. 32 - Uživatelské rozhraní v MDD editoru

5.3.3 Možnosti úpravy postprocesoru v GibbsCAM

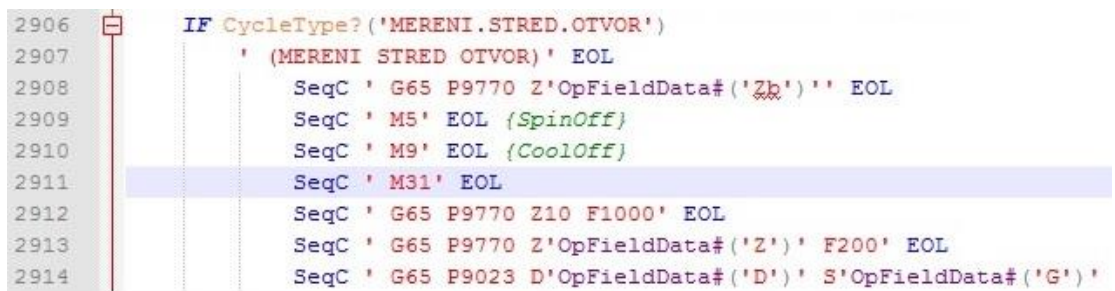
Podobně jako při tvorbě makra je software GibbsCAM velmi otevřený i z hlediska možnosti úpravy a tvorby vlastních postprocesorů. K úpravám a tvorbě nových postprocesorů v prostředí GibbsCAM se využívá aplikace ComPost znázorněna na obrázku Obr. 33 - Uživatelské rozhraní aplikace ComPost. Pomocí postprocesorů jsme schopni následně vytvořit



Obr. 33 - Uživatelské rozhraní aplikace ComPost 3

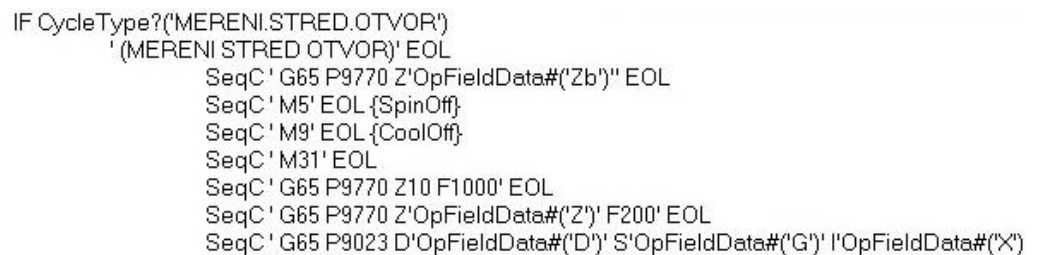
program pro ovládání CNC obráběcího stroje.

V GibbsCAMu se postprocesor dělí na dva typy souborů Form a Prog. Veškeré specifikace a parametry stroje se zadávají do souboru Form s příponou *.pst, vlastní napsaný program post procesoru je obsažen v textovém souboru Prog s příponou *.txt. Aplikace ComPost může být použita sama o sobě k úpravě nebo tvorbě post procesoru, ale většinou se používá pouze k úpravě souboru Form, zde jsme schopni definovat G-kody k některým funkcím jako je zapínání a vypínání chlazení, roztáčení vřetene po a proti směru hodinových ručiček. Dále jsme zde schopni definovat výstupní jednotky některých proměnných. K úpravě souboru Prog využíváme nejčastěji některého textového editoru v tomto případě softwaru Notepad++. Jedná se o volně dostupný textový editor zdrojového kódu s podporou několika programovacích jazyků. Pro tento program byl také vyvinut speciální plug-in, který umožňuje barevné rozlišení jednotlivých funkcí a cyklů, to nám umožňuje mnohem snadněji se orientovat v kódu postprocesoru. I změny v textovém souboru Prog jsme schopni provádět v aplikaci ComPost, ale ta nám neumožňuje toto barevné rozlišení a zároveň nikdy nebyla vytvářena jako textový editor, což velmi výrazně omezuje možnosti práce se samotným textem. Z následujících obrázků jsou na první pohled patrné rozdíly v zobrazení, ze kterých vidíme, že program Notepad++ znázorněný na obrázku Obr. 34 - Postprocesor otevřený v programu Notepad++, je mnohem přehlednější než program ComPost 3 znázorněný na obrázku Obr. 35 - Postprocesor otevřený v programu ComPost



```
2906 IF CycleType?('MERENI.STRED.OTVOR')
2907 ' (MERENI STRED OTVOR)' EOL
2908   SeqC ' G65 P9770 Z'OpFieldData#('Zb')'' EOL
2909   SeqC ' M5' EOL {SpinOff}
2910   SeqC ' M9' EOL {CoolOff}
2911   SeqC ' M31' EOL
2912   SeqC ' G65 P9770 Z10 F1000' EOL
2913   SeqC ' G65 P9770 Z'OpFieldData#('Z')' F200' EOL
2914   SeqC ' G65 P9023 D'OpFieldData#('D')' S'OpFieldData#('G')'
```

Obr. 34 - Postprocesor otevřený v programu Notepad++



```
IF CycleType?('MERENI.STRED.OTVOR')
  '(MERENI STRED OTVOR)' EOL
  SeqC ' G65 P9770 Z'OpFieldData#('Zb')'' EOL
  SeqC ' M5' EOL {SpinOff}
  SeqC ' M9' EOL {CoolOff}
  SeqC ' M31' EOL
  SeqC ' G65 P9770 Z10 F1000' EOL
  SeqC ' G65 P9770 Z'OpFieldData#('Z')' F200' EOL
  SeqC ' G65 P9023 D'OpFieldData#('D')' S'OpFieldData#('G')' 'OpFieldData#('X')
```

Obr. 35 - Postprocesor otevřený v programu ComPost 3

Dnes už se takřka nikdy nevytváří zcela nový postprocessor, ale pouze se upravují stávající. Abychom mohli upravovat již existující postprocessor, potřebujeme jak soubor Form tak Prog. Tyto soubory mohou být libovolně pojmenovány, ale vzhledem k tomu, že pro GibbsCAM existují tisíce postprocesorů a bylo by zcela nemožné se v nich orientovat, kdyby si každý pojmenovával svůj postprocessor dle libosti, byl vytvořen systém značení postprocesorů, který umožňuje všem vývojářům snadné a rychle rozpoznání. Po provedení všech úprav otevřeme soubor typu Form v programu ComPost 3 a zkompilujeme, tím se změny provedené v souboru typu Prog zobrazí i ve Formu a vytvoří se fungující Post Processor.

Jak jsem se již zmínil, název post procesoru je tedy složen z názvu a typu řídicího systému, značky a druhu stroje a nakonec několika číslic a písmen, které definují další parametry stroje jako například počet os stroje a zdali se jedná o metrický či palcový post procesor.

6. Vypravování

Toto zadání vzniklo ve spolupráci s pražskou firmou technology-support s.r.o., která se zabývá technologickou podporou pro strojírenské firmy. Zákazník této firmy, firma Jaromír Biem – Kovovýroba sídlící v Ciboušově u Klášterce nad Ohří vznesla požadavek na doplnění produkčního CAM softwaru GibbsCAM o možnost programování cyklů měřicí sondy pro upínání obrobku.

Konkrétní případ a stroj, na kterém tuto problematiku ukazují, je pouze pro nastínění postupu. Tyto makra můžeme využít v mnohem širší oblasti nejrůznějších obráběcích strojů s řídicím systémem Fanuc, který byl rozšířen o plug-in EasyProbe, neboť po správné úpravě postprocesoru je můžeme použít bez větších změn. Pokud jde o další řídicí systémy tak už budou muset proběhnout větší úpravy, ale stále zde budeme pracovat na tom samém principu vytváření maker pro systém GibbsCAM. Dalo by se tedy říct, že v případě měřících a ustavovacích cyklech jsem limitován pouze parametry a schopnostmi řídicího systému stroje.

Moje řešení takto vytvořených maker má další velmi výraznou výhodu a to, že je můžeme upravit přímo na přání každého zákazníka. Nejen generovaný výstup ale i uživatelské rozhraní zadávání je navrženo tak, aby odpovídalo potřebám konkrétního klienta a jeho požadavkům.

6.1 Požadavky na cykly pro upínání obrobku

Firma Jaromír Biem – Kovovýroba, vznesla požadavek na doplnění CAM programu GibbsCAM o pět nových cyklů, které by se dali použít pro upínání obrobku.

1. Měření vztažného (nulového) bodu ve směru osy X
2. Měření vztažného (nulového) bodu ve směru osy Y
3. Měření vztažného (nulového) bodu ve směru osy Z
4. Měření vztažného (nulového) bodu ve směru os XYZ
5. Měření vztažného bodu ve středu kruhové geometrie

Zároveň byl zaslán i požadovaný výstupní NC kód, který by měl být výstupem daného cyklu pro upnutí součásti. Níže můžeme vidět celý vygenerovaný NC program, který obsahuje pouze jednu operaci a to právě cyklus pro ustavení součásti v ose X. V různých cyklech se mění pouze část, která je oddělena od ostatní části odřádkami. Ve zbytku kódu se nachází ostatní operace, například výměna nástroje a na konci programu odjetí do bezpečné vzdálenosti. Před každým řádkem je v našem případě číselné označení, to se ale vztahuje k vysvětlivkám jednotlivých částí kódu a stejně jako červené ilustrační odřádky (=) nejsou součástí generovaného NC programu.

6.1.1 Celkový vygenerovaný NC program pro automatické měření

1. O20
2. N10(SEZNAM NASTROJU)
3. N20(T1)
4. N10 G0 G17 G40 G80 G90
(*****)
5. N20 G91 G30 G0 Z0.
6. N30 G91 G30 G0 X0. Y0.
7. N40 T1
8. N50 M6
(*****)
9. N60 G54 G17 G90 G0 G94
10. N70 G0 X-10. Y10.

11. N80 G43 H1 D1

(MERENI OSA X) =====

12. N90 G65 P9770 Z50.

13. N100 M5

14. N110 M9

15. M31

16. N120 G65 P9770 Z5 F1000

17. N130 G65 P9770 Z-5. F200

18. N140 G65 P9023 S56. X10. I0

19. N150 #550=#138

20. N160 G0 Z50. F5000

=====

21. N170 M32

22. N180 G91 G30 G0 Z0.

23. N190 G91 G30 G0 X0. Y0.

24. N200 M30

6.1.2 Popis jednotlivých částí generovaného kódu:

1. Řádek jedna až tři nám pouze uvádí základní informace o počtu nástrojů.
4. G17 načte rovinu obrábění v našem případě XY; G40 zruší poloměrové korekce; G80 vypne vrtací cykly; G90 zvolí absolutní programování.
5. G91 zvolí přírůstkové programování; G30 G0 Z0. nájezd do reference v ose Z.
6. G91 zvolí přírůstkové programování; G30 G0 X0. Y0. nájezd do reference v osách X a Y.
7. T1 načte do systému nový nástroj.
8. M6 dojde k reálné výměně nástroje.
9. G54 určí nulový bod; G17 načte rovinu obrábění v našem případě XY; G90 zvolí absolutní programování; G94 určí jednotky posuvu v [mm/min].

10. G0 X-10. Y10. najetí rychloposuvem v osách X a Y do bodu, který jsme si definovali v softwaru GibbsCAM jako výchozí pro měření.
11. Zapnutí délkových korekcí
12. Řádky 12. až 19. se týkají vlastních měřících cyklů a budu je rozvádět samostatně v dalších bodech.
20. M32 vypne sondu.
21. Najedeme rychloposuvem do reference ve směru osy Z.
22. Najedeme rychloposuvem do reference ve směru os X a Y.
23. M30 ukončí NC program.

6.2 Použité příkazy ve změně makra

Vlastní vrtací makra, kterých využívám ve své práci a přizpůsobuji je na využití pro ustavení součásti, se skládají ze 4 souborů. Každý soubor plní vlastní funkci a je napsán pomocí příkazů ve specifickém programovacím jazyce GibbsCAM. Všechny soubory mnou vytvořených maker si můžeme prohlédnout v příloze číslo 1.

V této kapitole se věnuji přesnému popisu obsahu jednotlivých souborů, včetně jednotlivých příkazů, které obsahují. Některé znaky a příkazy jsou stejné pro všechny soubory makra například řádky, před kterými je znak “!“, jsou neaktivní a tudíž ať je za nimi jakýkoliv text nebo příkaz tyto řádky do makra nikterak nezasahují. Tato vlastnost se nám může hodit například při vkládání popisků do vlastního kódu pro lepší orientaci.

6.2.1 Soubor DefineData.txt

Tento soubor slouží k definici všech proměnných, které jsou v daném makru využité. Zároveň je zde také uvedena počáteční (předvyplněná) hodnota přiřazená jednotlivým proměnným.

Obsah tohoto souboru se tedy skládá pouze z proměnných a jejich hodnot. Například pracujeme-li s parametry X, Y, Z, pak soubor DefineData.txt má následující obsah:

X = 10

Y = 10

Z = 10

6.2.2 Soubor CycleData.dlg

Soubor CycleData.dlg slouží pro vytvoření grafického rozhraní, do kterého zadáváme hodnoty proměnných, se kterými pracujeme. Oproti souboru DefineData.txt, kde nám hodnoty slouží předdefinování vybraných hodnot, které můžeme měnit pouze přímo po otevření v textovém editoru, zde máme možnost měnit jednotlivé hodnoty přímo v dialogovém okně dle potřeby přímo v programu GibbsCAM.

Toto dialogové okno vytváříme podobně jako v jiných programovacích jazycích i zde pomocí námi definovaných příkazů. Těchto příkazů je velké množství a proto zde popisují pouze ty, které jsem použil. Pro více informací o dalších možnostech se můžete po registraci podívat na stránkách <https://macros.gibbscam.com>.

1. DIALOG

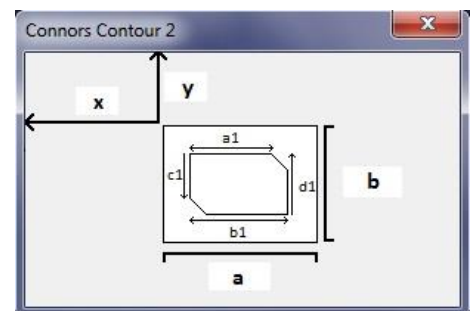
Tento příkaz slouží k tvorbě vlastního dialogového okna pro zadávání proměnných, které se otevře při zmáčknutí tlačítka Data cyklu uživatele, při definování vrtací (v našem případě ustavovací) operace.

DIALOG “**Název okna**“, <**umístění okna na obrazovce**>, <**rozměry okna**>

- Název okna udává jméno, které se bude zobrazovat v horním rámečku okna.
- Umístění okna na obrazovce určuje polohu, kde se dané okno zobrazí na monitoru, vzhledem k hornímu levému rohu. Tato hodnota se udává ve formě bodu [x, y], jednotky jsou v pixelech.
- Rozměry okna udávají velikost (šířka, výška) dialogového okna v pixelech.
- Příklad příkazu:
 - DIALOG “**Ustavení roviny XY**”, 100, 100, 300, 500

2. IMAGE

Tento příkaz slouží k vložení obrázku do dialogového okna. V názvu obrázku, který chceme vložit, se nesmí vyskytovat diakritika a mezery. Dále musí být ve formátu rastrového obrázku *.bmp.



Obr. 36 - Definice příkazu IMAGE

IMAGE <Umístění obrázku>, <Rozměry obrázku>, “Název obrázku“

- Umístění obrázku určuje polohu obrázku vzhledem k levému hornímu rohu. Na obrázku Obr. 36 - Definice příkazu IMAGE je znázorněná poloha zleva (x) a shora (y). Rozměry jsou v pixelech.
- Rozměry obrázku určují šířku a výšku obrázku v pixelech. Na obrázku Obr. 36 - Definice příkazu IMAGE je znázorněná pod označením šířka (a) a výška (b).
- Název obrázku nesmí obsahovat diakritiku ani mezery, uvádíme včetně formátu *.bmp. V případě že se obrázek nachází mimo složku, kde jsou uloženy ostatní soubory makra tak je nutné uvést zde i cestu k obrázku. Například: C:\Obrázky\Obrazek.bmp.
- Příklad příkazu:
 - IMAGE 25, 25, 1000, 300, “Obrázek.bmp”

3. LABEL

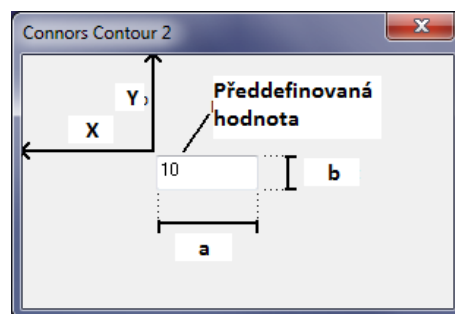
Jedná se o popisek, který vkládáme na libovolné místo dialogového okna, a slouží k popisu jednotlivých vstupů ve formě příkazu INPUT.

LABEL <Umístění obrázku>, <Rozměry obrázku>, “Text popisku“

- Definice polohy a velikosti popisku má stejné parametry jako příkaz IMAGE (x, y, a, b), které vidíme na obrázku Obr. 36 - Definice příkazu IMAGE.
- Text popisku je vlastní text, který se bude zobrazovat v dialogovém okně.
- Příklad příkazu:
 - IMAGE 25, 25, 120, 24, “Popisek 1”

4. INPUT

Pomocí příkazu INPUT vkládáme číselné hodnoty, které budou přiděleny jednotlivým proměnným.



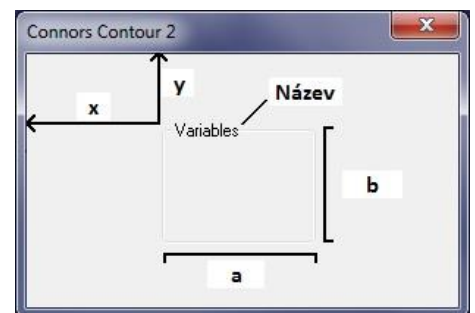
Obr. 37 - Definice příkazu INPUT

INPUT <Umístění obrázku>, <Rozměry obrázku>, <Proměnná>, <Základní hodnota>

- Definice polohy a velikosti vstupu má stejné parametry jako příkaz IMAGE, nebo LABEL (x, y, a, b), zde jsou znázorněné na obrázku Obr. 37 - Definice příkazu INPUT.
- Proměnná určuje, ke které proměnné bude přiřazena číselná hodnota.
- Základní hodnotou můžeme předdefinovat počáteční hodnotu pro danou proměnnou, podobně jako v souboru marka DefineData.txt, ale právě proto, že již v tomto souboru máme tyto hodnoty určené tak zde pouze opět napíšeme proměnnou.
- Příklad příkazu:
 - INPUT 25, 25, 120, 24, X, X

5. FRAME

Tímto příkazem vložíme do dialogového okna ohraničení, které může sloužit například k oddělení určitých skupin vstupů (INPUT), což zlepšuje orientaci při definování jednotlivých parametrů.



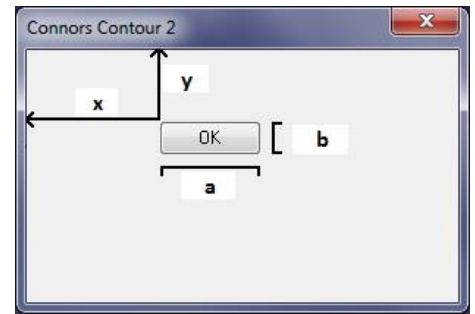
Obr. 38 - Definice příkazu FRAME

FRAME <Umístění obrázku>, <Rozměry obrázku>, <Název ohraničení>

- Definice polohy a velikosti ohraničení má stejné parametry jako příkaz IMAGE, LABEL, nebo INPUT (x, y, a, b), zde je vidíme na obrázku Obr. 38 - Definice příkazu FRAME
- Název ohraničení je nadpis pro ohraničenou skupinu.
- Příklad příkazu:
 - FRAME 25, 25, 600, 100, Skupina A

6. Tlačítko OK a CANCEL

Tlačítko OK a CANCEL musí být v každém dialogovém okně. OK uloží zadané hodnoty a zavře okno, tlačítko CANCEL zavře okno bez uložení zadaných hodnot.



OK <Umístění obrázku>, <Rozměry obrázku>

Obr. 39 - Definice zobrazení tlačítek

CANCEL <Umístění obrázku>, <Rozměry obrázku>

- Definice polohy a velikosti tlačítek má stejné parametry jako příkaz IMAGE, LABEL, nebo INPUT (x, y, a, b), zde je vidíme na obrázku Obr. 39 - Definice zobrazení tlačítek
- Příklad příkazu:
 - OK 25, 25, 70, 24
 - CANCEL 25, 25, 70, 24

6.2.3 Soubor CycleData.mac

Tento soubor makra se spustí po kliknutí na tlačítko Data cyklu uživatele v okně definice vrtacích cyklů a slouží k zobrazení dialogového okna a pro kontrolu zadaných údajů. Jsme zde schopni pomocí základních programovacích příkazů jako je IF, THEN a dalších zkontrolovat údaje, které jsme zadali do dialogového okna. Tato kontrola může sloužit například k zabránění kolize sondy, při zadání malého nájezdu pro měření.

Před napsáním vlastních podmínek kontroly zadaných hodnot je většinou potřeba načíst několik dalších parametrů jakými může být například číslo dané operace a poloměr sondy pro danou operaci pomocí příkazů GET OP DATA, nebo GET TOOL DATA. Toto jsou pouze dva příklady, kterých jsem využil ve své práci, další najdete po registraci na stránkách <https://macros.gibbscam.com>.

CALLBACK_PROC načte číslo procesu pro zjednodušení a přehlednost můžeme toto číslo operace uložit pod jiným označením ve formě: `iprocc = CALLBACK_PROC`.

GET_PROC_DATA <Číslo procesu>, <Parametr>, <Proměnná>

- Číslo procesu můžeme napsat ručně, ale ne vždy je operace ustavení jako první takže je nutné tuto hodnotu generovat pomocí příkazu `CALLBACK_PROC`.
- Parametr, který chceme z daného procesu načíst. Může se jednat o velké množství veličin, v našem případě načteme číslo nástroje, pod kterým je uložena v programu sonda.
- Číslo daného nástroje je uloženo pod proměnnou, kterou zde napíšeme.
- Příklad příkazu pro načtení čísla nástroje z programu GibbsCAM:
 - `GET_PROC_DATA iproc, TOOL_NUM, toolnum`

`GET_TOOL_DATA <Číslo nástroje>, <Parametr>, <Proměnná>`

- Číslo nástroje zde můžeme buď trvale přiřadit pokud máme sondu neustále ve stejné pozici na stroji, pokud tomu tak není, načteme jí pomocí příkazu `GET_PROC_DATA`.
- Parametrů zde můžeme načíst opět velké množství, v našem případě načteme poloměr nástroje.
- Hodnotu poloměru přiřadíme zde napsané proměnné.
- Příklad příkazu pro načtení poloměru nástroje:
 - `GET_TOOL_DATA toolnum, MTOOL_RADIUS, toolrad`

`CALLBACK_RETURNCODE = 0` tato část kódu kontroluje, jestli jsou všechny nastavené parametry v pořádku. V případě, kdy se vyskytne chyba a hodnota se nerovná nule, zobrazí se hlášení o chybě.

`:show_dialog`

Dialog "CycleData.dlg"

- Tato část kódu otevře námi definované dialogové okno v souboru makra `CycleData.dlg`.

Příkaz `IF/THEN`

- `IF <hodnota1> <podmínka> <hodnota2> THEN`

- Podmínka může mít několik podob například <, >, =
- Příklad příkazu pro zobrazení správy o nesplněné podmínce:
 - $X = 2$
 - IF $X < 4$ THEN Message “Prilis mala hodnota X”

6.2.4 Soubor ToolPath

Tento soubor makra slouží k definování dráhy nástroje sondy, při měření. Tyto dráhy nástroje slouží pouze pro simulaci pohybu nástroje v softwaru GibbsCAM a s jejich pomocí se můžeme podívat na přesnou simulaci pohybu měřicí sondy a ověřit tak, že nedochází k žádné kolizi.

Pro generování dráhy nástroje nejdříve potřebujeme načíst některé údaje ze softwaru GibbsCAM jako číslo operace, číslo nástroje sondy a poloměr sondy. Těmito příkazy jsem se zabýval v kapitole 6.2.3 Soubor CycleData.mac.

Makra pro ustavování počátku v programu GibbsCAM vychází z maker pro Vlastní vrtací cykly, a tyto cykly se orientují podle bodu, který jsme vytvořili v prostoru, kde má začít vlastní měření (Výchozí bod měření). Tento výchozí bod měření je nulovým bodem pro definování drah pohybu nástroje v osách X a Y, nulová poloha v ose Z je daná hlavní bezpečnostní rovinou.

GET_CUSTOM_DRILL_CYCLE_DATA

- Tento příkaz slouží pro načtení dat z vlastního ustavovacího (vrtacího) cyklu. Jeho definice je velmi podobná jako u příkazu GET_PROC_DATA, nebo GET_OP_DATA.
 - GET_CUSTOM_DRILL_CYCLE_DATA <Parametr, který chci načíst>, <Proměnná, které přiřadím načtenou hodnotu>
- Pomocí tohoto příkazu mohu načíst velké množství údajů týkajících se vlastního cyklu. Ve své práci ho využívám k načtení polohy výchozího bodu v souřadném systému součásti nebo hodnoty hlavní bezpečnostní roviny.
- Příklad pro načtení hlavní bezpečnostní roviny:
 - GET_CUSTOM_DRILL_CYCLE_DATA retract_plane, retract

ADD_CUSTOM_DRILL_CYCLE_PATH feed

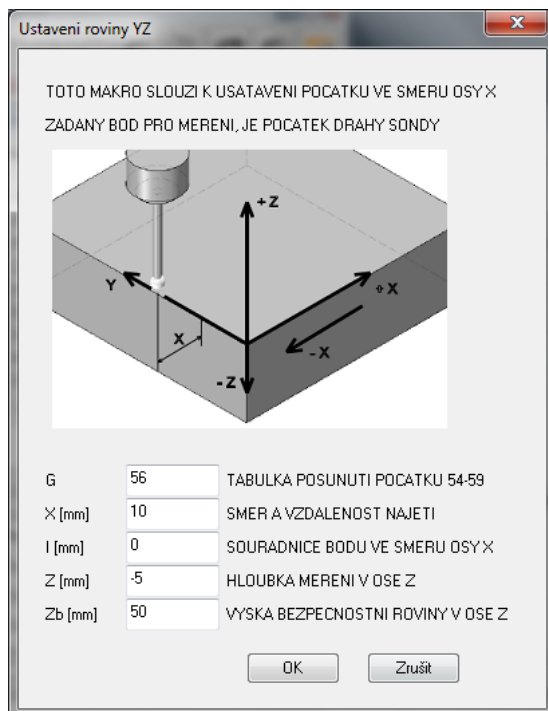
- Tento příkaz slouží k pohybu nástroje při simulaci. Tento pohyb může být ve všech třech osách nebo pouze ve směru jedné osy.
- Příklad pohybu nástroje ve třech osách o 10 mm:
 - ADD_CUSTOM_DRILL_CYCLE_PATH feed, 10, 10, 10
- Příklad pohybu nástroje v ose X o 10 mm:
 - ADD_CUSTOM_DRILL_CYCLE_PATH feed_x, 10

6.3 Makra pro ustavení počátku ve směru osy X a Y

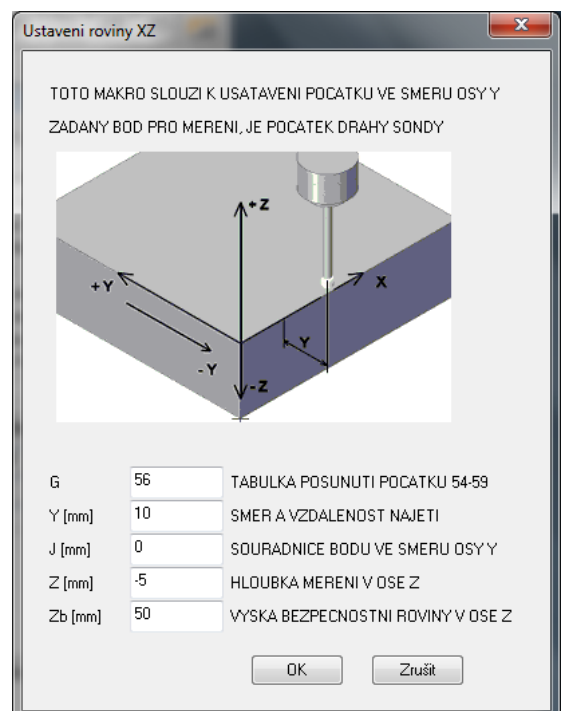
Jde o dva na sobě nezávislé cykly, ale definicí a použitými příkazy jsou takřka identické, pouze s rozdílem směru najetí k měřené rovině, proto je popisují v jedné kapitole.

Tyto cykly slouží k ustavení počátku v rovinách na součásti, které jsou rovnoběžné buď s rovinou YZ, nebo XZ. Podle námi definovaných parametrů naměřené rovině přiřadíme danou hodnotu, od které bude určen nulový bod.

Takto vypadají vytvořená dialogová okna, do kterých doplňujeme požadované údaje.



Obr. 41 - Dialogové okno pro definici ustavení v rovině YZ



Obr. 40 - Dialogové okno pro definici ustavení v rovině XZ

Jak jsem již zmínil makra pro ustavení rovin YZ a XZ jsou takřka totožná, proto zde uvádím pouze makro pro měření v ose X a barevně zvýrazňuji pouze jednotlivé rozdíly. Modrá barva náleží cyklu pro pohyb sondy ve směru osy X a červená pro pohyb sondy ve směru osy Y.

- Okno bylo vytvořeno následujícími příkazy v souboru makra CycleData.dlg.
 1. dialog "Ustaveni roviny YZ", 30, 50, 400, 520
 2. label 20, 25, 350, 24, "TOTO MAKRO SLOUZI K USATAVENI POCATKU VE SMERU OSY X/Y"
 3. label 20, 50, 350, 24, "ZADANY BOD PRO MERENI, JE POCATEK DRAHY SONDY"
 4. image 25, 75, 320, 233, "obr.probe. X/Y.bmp"
 5. label 20, 315, 120, 24, "G"
 6. label 20, 340, 120, 24, " X/Y [mm]"
 7. label 20, 365, 120, 24, " I/J [mm]"
 8. label 20, 390, 120, 24, "Z [mm]"
 9. label 20, 415, 120, 24, "Zb [mm]"
 10. input 80, 310, 70, 24, G, G
 11. input 80, 335, 70, 24, a/b, a/b
 12. input 80, 360, 70, 24, I/J, I/J
 13. input 80, 385, 70, 24, Z, Z
 14. input 80, 410, 70, 24, Zb, Zb
 15. label 155, 315, 200, 24, "TABULKA POSUNUTI POCATKU 54-59"
 16. label 155, 340, 160, 24, "SMER A VZDALENOST NAJETI"
 17. label 155, 365, 200, 24, "SOURADNICE BODU VE SMERU OSY X/Y "
 18. label 155, 390, 230, 24, "HLOUBKA MERENI V OSE Z"
 19. label 155, 415, 230, 24, "VYSKA BEZPECNOSTNI ROVINY V OSE Z"
 20. cancel 260, 450, 70, 24
 21. ok 170, 450, 70, 24
- Námi předdefinované hodnoty v souboru DefineData.txt jsou:
 1. a/b = 10

2. $I/J = 0$
3. $G = 56$
4. $Zb = 50$
5. $Z = -5$

- Parametry pro kontrolu zadaných hodnot obsažené v souboru makra CycleData.mac jsou:

1. `iprocc = callback_proc`
2. `callback_returncode = 0`
3. `get_proc_data iproc, TOOL_NUM, toolnum`
4. `get_tool_data toolnum, MTOOL_RADIUS, toolrad`
5. `:show_dialog`
6. `dialog "CycleData.dlg" ! show the custom dialog`
7. `icode = 0`
8. `if G < 54 then icode = 1`
9. `if G > 59 then icode = 1`
10. `if abs(a) < toolrad then icode = 2`
11. `if icode = 1 then message "NEPLATNA HODNOTA G MOZNO POUZE V ROZMEZI 54-59"`
12. `if icode = 2 then message "PRILIS MALA HODNOTA NAJEZDU X"`
13. `if icode > 0 then goto show_dialog`

Pro možnost simulování daných cyklů jsme vytvořili následující program, který je uložen v souboru makra ToolPath.mac.

1. `iop = callback_op`
 2. `callback_returncode = 0`
 3. `get_custom_drill_cycle_data retract_plane, retract`
 4. `get_op_data iop, tool_num, itool`
 5. `get_tool_data itool, mtool_radius, toolrad`
- !MERENI VE SMERU OSY X**
6. `add_custom_drill_cycle_path feed, 0, 0, Zb-retract`
 7. `add_custom_drill_cycle_path feed_z, Z-retract`
 - 8.1 `IF a/b > 0, THEN add_custom_drill_cycle_path feed_ x/y, a/b-toolrad`

8.2 IF $a/b < 0$, THEN add_custom_drill_cycle_path feed_ x/y , a/b +toolrad

9. add_custom_drill_cycle_path feed_ x/y , 0

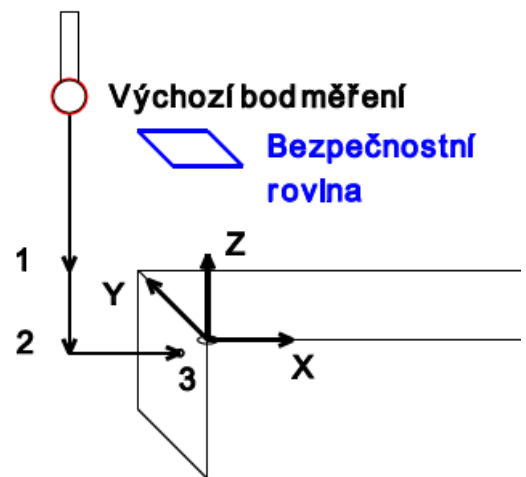
10. add_custom_drill_cycle_path feed_ z , Z_b -retract

6.3.1 NC program pro měření vztažných bodů ve směru osy X

V této části popisují pouze tu část NC kódu, která řídí vlastní pohyby obrobkové sondy. Jedná se o část programu mezi body 12. až 20., která se nachází v bodu Celkový vygenerovaný NC program pro automatické měření. Jednotlivé pohyby obrobkové sondy jsou znázorněné na obrázku Obr. 42 - Zobrazení dráhy sondy pro ustavení v rovině YZ a číselně označené body 1, 2 a 3. Části programu, před kterými je označení a až g, jsou funkce, které se netýkají pohybu vlastního měření. Kurzívou označené části textu se odkazují na jednotlivé námi definované hodnoty, které zadáváme do dialogového okna, znázorněného na Obr. 41 - Dialogové okno pro definici ustavení v rovině YZ.

Vygenerovaná část NC programu pro měření vztažných bodů v ose X

- a. N90 G65 P9770 Z50.
- b. N100 M5
- c. N110 M9
- d. M31
1. N120 G65 P9770 Z5 F1000
2. N130 G65 P9770 Z-5. F200
3. N140 G65 P9023 S56. X10. I0
- e. N150 #550=#138
- f. N160 G0 Z50. F5000



Obr. 42 - Zobrazení dráhy sondy pro ustavení v rovině YZ

Popsání jednotlivých bodů vygenerovaného NC programu

- a. Na řádku 10 v NC kódu Celkový vygenerovaný NC program pro automatické měření jsem se v popisu jednotlivých bodů zmínil, že sonda najede do námi vybraného bodu ze softwaru GibbsCAM, ale pouze v ose X a Y. Nyní v tomto bodě sjedeme níže ve směru osy Z a to na námi definovanou hodnotu *Výška bezpečnostní roviny v ose Z (Zb)*. Námi zvolená hodnota je v tomto případě Z50. K posunu zde použijeme speciální funkci P9770 bezpečnostní posuv, který nám okamžitě stroj zastaví, pokud se sonda čehokoliv dotkne.

- b. M5 vypne otáčky vřetena.
 - c. M9 vypne chlazení
 - d. M31 zapne dotykovou sondu.
1. Sonda sjede z *Výška bezpečnostní roviny v ose Z (Zb)* v našem případě $Z=50\text{mm}$ do hodnoty, která definuje hlavní Bezpečnostní rovinu v ose Z v našem případě $Z=5\text{mm}$. Opět bude posuv vykonán bezpečnostním posuvem a rychlostí $f = 1\ 000$ [mm/min].
 2. V tomto bodě nám sonda sjede do námi zadané hladiny *Hloubka měření v ose Z*, kde bude probíhat samotné měření. V našem případě $Z = -5\ \text{mm}$. Vidíme, že hodnota posuvu se zde snížila na $f = 200$ [mm/min].
 3. V bodě tři dojde k provedení vlastního měření, pomocí funkce P9023 a námi zadaných parametrů. Hodnota *S56 určuje nulový bod z tabulky*, pro který bude nastaven měřený bod; X nám určuje *Směr a vzdálenost najetí*, pokud je hodnota kladná pohybuje se sonda v kladném směru osy X ; I neboli *Souřadnice bodu ve směru osy X* určuje skutečnou hodnotu, kterou má mít naměřený bod. Po ukončení měření se sonda vrátí do bodu 2. Naměřené hodnoty se uloží do #138.
- e. Dojde k provedení posunutí nulového bodu o naměřené hodnoty uložené v #138.
 - f. Odjedeme rychloposuvem ve směru osy Z do hodnoty *Výška bezpečnostní roviny v ose Z (Zb)*.

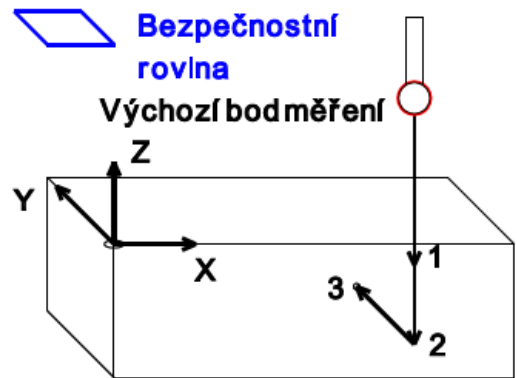
6.3.2 NC program pro měření vztažných bodů ve směru osy Y

V této části popisují pouze tu část NC kódu, která řídí vlastní pohyby obrobkové sondy. Jedná se o část programu mezi body 12. až 20., která se nachází v bodu Celkový vygenerovaný NC program pro automatické měření. Jednotlivé pohyby obrobkové sondy jsou znázorněné na obrázku Obr. 43 - Zobrazení dráhy sondy pro ustavení v rovině XZ a číselně označené body 1, 2 a 3. Části programu, před kterými je označení a až g, jsou funkce, které se netýkají pohybu vlastního měření. Text, který je psán kurzívou označuje námi definované hodnoty zadávané do dialogového okna na obrázku Obr. 40 - Dialogové okno pro definici ustavení v rovině XZ.

Dráhy tohoto měření jsou velmi podobné drahám měření v ose X, s tím rozdílem, že nyní je měřicí pohyb prováděn v ose Y.

Vygenerovaná část NC programu pro měření vztažných bodů v ose Y

- a. N90 G65 P9770 Z50.
- b. N100 M5
- c. N110 M9
- d. M31
- 1. N120 G65 P9770 Z5 F1000
- 2. N130 G65 P9770 Z-5. F200
- 3. N140 G65 P9023 S56. Y10. J0
- e. N150 #550=#138
- f. N160 G0 Z50. F5000



Obr. 43 - Zobrazení dráhy sondy pro ustavení v rovině XZ

Popsání jednotlivých bodů vygenerovaného NC programu

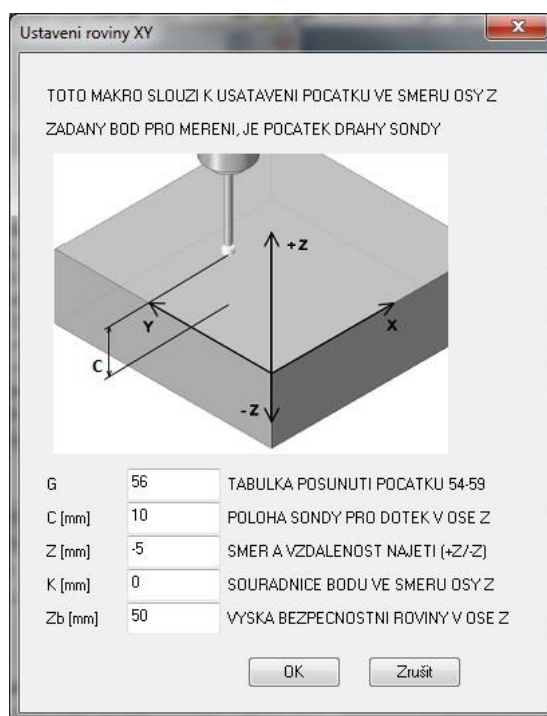
- a. Na řádku 10 v NC kódu *Celkový vygenerovaný NC program pro automatické měření* jsem se v popisu jednotlivých bodů zmínil, že sonda najede do námi vybraného bodu *Výchozí bod měření* ze softwaru GibbsCAM, ale pouze v ose X a Y. Nyní v tomto bodě sjedeme níže ve směru osy Z a to na námi definovanou hodnotu *Výška bezpečnostní roviny v ose Z (Zb)*. Námi zvolená hodnota je v tomto případě Z50. K posunu zde použijeme speciální funkci P9770 bezpečnostní posuv, který nám okamžitě stroj zastaví, pokud se sonda čehokoliv dotkne.
- b. M5 vypne otáčky vřetena.
- c. M9 vypne chlazení
- d. M31 zapne dotykovou sondu.
- 1. Sonda sjede z *Výška bezpečnostní roviny v ose Z (Zb)* v našem případě Z=50mm do hodnoty, která definuje hlavní *Bezpečnostní roviny v ose Z* v našem případě Z=5mm. Opět bude posuv vykonán bezpečnostním posuvem a rychlostí $f = 1\ 000$ [mm/min].

2. V tomto bodě nám sonda sjede do námi zadané hladiny *Hloubka měření v ose Z*, kde bude probíhat samotné měření. V našem případě $Z = -5$ mm. Vidíme, že hodnota posuvu se zde snížila na $f = 200$ [mm/min].
3. V bodě tři dojde k provedení vlastního měření, pomocí funkce P9023 a námi zadaných parametrů. Hodnota *S56 určuje nulový bod z tabulky*, pro který bude nastaven měřený bod; *Y* nám určuje *Směr a vzdálenost najetí*, pokud je hodnota kladná pohybuje se sonda v kladném směru osy *Y*; *J Souřadnice bodu ve směru osy Y* určuje skutečnou hodnotu, kterou má mít naměřený bod. Po ukončení měření se sonda vrátí do bodu 2. Naměřené hodnoty se uloží do #138.
- e. Dojde k provedení posunutí nulového bodu o naměřené hodnoty uložené v #138.
- f. Odjedeme rychloposuvem ve směru osy *Z* do hodnoty *Výška bezpečnostní roviny v ose Z (Zb)*.

6.4 Makro pro ustavení počátku ve směru osy Z

Tento měřicí cyklus se velmi podobá cyklu pro ustavení ve směru osy X a Y. Rozdíl je zde ve směru nájezdu shora, což vyžaduje jiný tvar vygenerovaného NC programu a jiné zadané hodnoty.

Dialogové okno pro definici cyklu ustavení ve směru osy Z vypadá následovně.



Obr. 44 - Dialogové okno pro definici ustavení v rovině XY

- Okno bylo vytvořeno následujícími příkazy v souboru makra CycleData.dlg
 1. dialog "Ustaveni roviny XY", 30, 50, 400, 520
 2. label 20, 25, 350, 24, "TOTO MAKRO SLOUZI K USATAVENI POCATKU VE SMERU OSY Z"
 3. label 20, 50, 350, 24, "ZADANY BOD PRO MERENI, JE POCATEK DRAHY SONDY"
 4. image 25, 75, 320, 249, "obr.probe.Z.bmp"
 5. label 20, 315, 120, 24, "G"
 6. label 20, 340, 120, 24, "C [mm]"
 7. label 20, 365, 120, 24, "Z [mm]"
 8. label 20, 390, 120, 24, "K [mm]"
 9. label 20, 415, 120, 24, "Zb [mm]"
 10. input 80, 310, 70, 24, G, G
 11. input 80, 335, 70, 24, C, C
 12. input 80, 360, 70, 24, d, d
 13. input 80, 385, 70, 24, K, K
 14. input 80, 410, 70, 24, Zb, Zb
 15. label 155, 315, 200, 24, "TABULKA POSUNUTI POCATKU 54-59"
 16. label 155, 340, 200, 24, "POLOHA SONDY PRO DOTEK V OSE Z"
 17. label 155, 365, 200, 24, "SMER A VZDALENOST NAJETI (+Z/-Z)"
 18. label 155, 390, 200, 24, "SOURADNICE BODU VE SMERU OSY Z"
 19. label 155, 415, 230, 24, "VYSKA BEZPECNOSTNI ROVINY V OSE Z"
 20. cancel 260, 450, 70, 24
 21. ok 170, 450, 70, 24
- Námi předdefinované hodnoty v souboru DefineData.txt jsou:
 1. $d = -10$
 2. $C = 10$
 3. $Zb = 50$
 4. $K = 0$
 5. $G = 56$

- Parametry pro kontrolu zadaných hodnot obsažené v souboru makra CycleData.mac jsou:
 1. `iprocc = callback_proc`
 2. `callback_returncode = 0` ! no errors so far, set to non-zero if an error
 3. `get_proc_data iproc, TOOL_NUM, toolnum`
 4. `get_tool_data toolnum, MTOOL_RADIUS, toolrad`
 5. `:show_dialog`
 6. `dialog "CycleData.dlg"` ! show the custom dialog
 7. `icode = 0`
 8. `if G < 54 then icode = 1`
 9. `if G > 59 then icode = 1`
 10. `if abs(d) < toolrad then icode = 2`
 11. `if icode = 1 then message "NEPLATNA HODNOTA G MOZNO POUZE V ROZMEZI 54-59"`
 12. `if icode = 2 then message "PRILIS MALA HODNOTA NAJETI"`
 13. `if icode > 0 then goto show_dialog`
- Pro možnost simulování daných cyklů jsme vytvořili následující program, který je uložen v souboru makra ToolPath.mac.
 1. `iop = callback_op`
 2. `callback_returncode = 0`
 3. `get_op_data iop, tool_num, itool`
 4. `get_tool_data itool, mtool_radius, toolrad`
 5. `get_custom_drill_cycle_data retract_plane, retract`
`!MERENI VE SMERU OSY Z`
 6. `add_custom_drill_cycle_path feed, 0, 0, Zb-retract`
 7. `add_custom_drill_cycle_path feed_z, C-retract`
 8. `add_custom_drill_cycle_path feed_z, C+d-retract`
 9. `add_custom_drill_cycle_path feed_z, Zb-retract`

6.4.1 NC program pro měření vztažných bodů ve směru osy Z

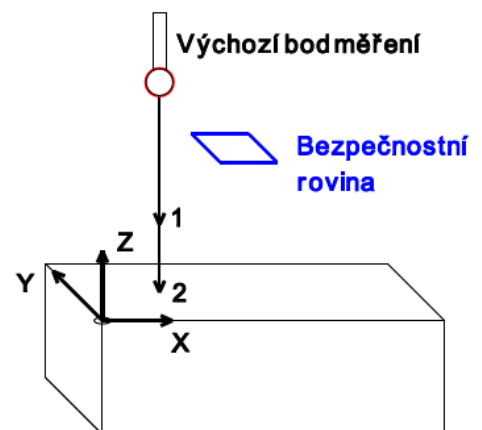
V této části popisují pouze tu část NC kódu, která řídí vlastní pohyby obrobkové sondy. Jedná se o část programu mezi body 12. až 20., která se nachází v bodu Celkový

vygenerovaný NC program pro automatické měření. Jednotlivé pohyby obrobkové sondy jsou znázorněné na obrázku Obr. 45 - Zobrazení dráhy sondy pro ustavení v rovině XZ a číselně označené body 1 a 2. Části programu, před kterými je označení a až f, jsou funkce, které se netýkají pohybu vlastního měření. Text, který je psaný kurzívou znázorňuje námi doplňované hodnoty do dialogového okna na obrázku Obr. 44 - Dialogové okno pro definici ustavení v rovině XY.

Nyní provádíme měření v ose Z, takže již není třeba nejdříve najíždět do hlavní Bezpečnostní roviny a pod touto rovinou se pohybovat velmi pomalým posuvem, stačí zde pouze sjet do požadované hodnoty v ose Z, ze které budeme provádět měření.

Vygenerovaná část NC programu pro měření vztažných bodů v ose Z

- a. N90 G65 P9770 Z50.
- b. N100 M5
- c. N110 M9
- d. M31
- 1. N120 G65 P9770 Z10 F1000
- 2. N140 G65 P9023 S56. Z-10. K0
- e. N150 #550=#138
- f. N160 G0 Z50. F5000



Obr. 45 - Zobrazení dráhy sondy pro ustavení v rovině XZ

Popsání jednotlivých bodů vygenerovaného NC programu

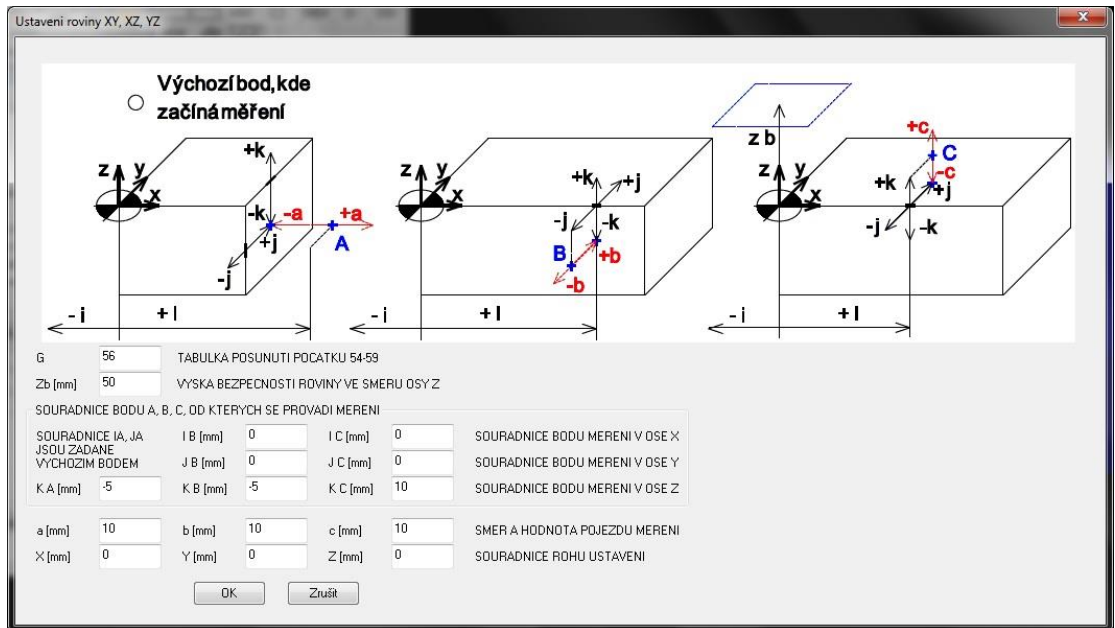
- a. Na řádce 10 v NC kódu *Celkový vygenerovaný NC program pro automatické měření* jsem se v popisu jednotlivých bodů zmínil, že sonda najede do námi vybraného bodu *Výchozí bod měření* ze softwaru GibbsCAM, ale pouze v ose X a Y. Nyní v tomto bodě sjedeme níže ve směru osy Z a to na námi definovanou hodnotu *Výška bezpečnostní roviny v ose Z (Zb)*. Námi zvolená hodnota je v tomto případě Z50. K posunu zde použijeme speciální funkci P9770 bezpečnostní posuv, který nám okamžitě stroj zastaví, pokud se sonda čehokoliv dotkne.
- b. M5 vypne otáčky vřetena.
- c. M9 vypne chlazení

- d. M31 zapne dotykovou sondu.
1. Sonda sjede z *Výška bezpečnostní roviny v ose Z (Zb)* v našem případě $Z=50\text{mm}$ do Polohy sondy pro dotek v ose Z v našem případě $Z=10\text{mm}$. Opět bude posuv vykonán bezpečnostním posuvem a rychlostí $f = 1\ 000$ [mm/min].
 2. V tomto bodě dojde k provedení vlastního měření, pomocí funkce P9023 a námi zadaných parametrů. Hodnota *S56 určuje nulový bod z tabulky*, pro který bude nastaven měřený bod; Y nám určuje *Směr a vzdálenost najetí*, pokud je hodnota kladná pohybuje se sonda v kladném směru osy Z; K neboli *Souřadnice bodu ve směru osy Z*, určuje skutečnou hodnotu, kterou má mít naměřený bod. Po ukončení měření se sonda vrátí do bodu 1. Naměřené hodnoty se uloží do #138.
- e. Dojde k provedení posunutí nulového bodu o naměřené hodnoty uložené v #138
- f. Odjedeme rychloposuvem ve směru osy Z do hodnoty *Výška bezpečnostní roviny v ose Z (Zb)*.

6.5 Makro pro ustavení počátku ve všech třech osách

V tomto měřicím cyklu už neměříme pouze jednotlivé osy samostatně, ale dojde ke spojení tří cyklů pro měření osy X, Y a Z do jednoho. Toto nás ovšem staví před problém, ve své podstatě tento cyklus měří bod v prostoru, který se nachází v rohu tvořeným třemi naměřenými plochami, ovšem na součásti se mohou nacházet rohy vnitřní a vnější a také na každé ploše se může nacházet měřený bod v jiné hladině, abychom dosáhli univerzálně použitého cyklu, nemůžeme zde už tedy vycházet pouze z jednoho bodu. Vycházíme tedy ze tří definovaných bodů A, B a C, od kterých se bude provádět měřicí pohyb v jednotlivých osách. Přičemž souřadnice X a Y bodu A vycházejí pro zjednodušení z výchozího (počátečního) bodu měření, tudíž následně stačí zadat pouze hloubku měření v ose Z. Body B a C musíme následně zadat sami do dialogového okna vytvořeného makra.

Dialogové okno pro definici cyklu ustavení ve směru všech os vypadá následovně.



Obr. 46 - Dialogové okno pro definici ustavení ve všech rovinách

- Okno bylo vytvořeno následujícími příkazy v souboru makra CycleData.dlg
 1. dialog "Ustaveni roviny XY, XZ, YZ", 200, 100, 1050, 590
 2. image 25, 25, 1026, 292, "obr.probe.XYZ.bmp"
 3. label 20, 300, 120, 24, "G"
 4. label 20, 325, 120, 24, "Zb [mm]"
 5. FRAME 10, 350, 635, 100, "SOURADNICE BODU A, B, C, OD KTERÝCH SE PROVÁDÍ MĚŘENÍ"
 6. label 20, 375, 120, 50, "SOURADNICE IA, JA JSOU ZADÁNE VÝCHOZÍM BODEM"
 7. label 20, 425, 120, 24, "K A [mm]"
 8. label 160, 375, 120, 24, "I B [mm]"
 9. label 160, 400, 120, 24, "J B [mm]"
 10. label 160, 425, 120, 24, "K B [mm]"
 11. label 300, 375, 120, 24, "I C [mm]"
 12. label 300, 400, 120, 24, "J C [mm]"
 13. label 300, 425, 120, 24, "K C [mm]"
 14. label 20, 465, 120, 24, "a [mm]"
 15. label 20, 490, 120, 24, "X [mm]"

16. label 160, 465, 120, 24, "b [mm]"
17. label 160, 490, 120, 24, "Y [mm]"
18. label 300, 465, 120, 24, "c [mm]"
19. label 300, 490, 120, 24, "Z [mm]"
20. input 80, 295, 60, 24, G, G
21. input 80, 320, 60, 24, Zb, Zb
22. input 80, 420, 60, 24, KA, KA
23. input 80, 460, 60, 24, a, a
24. input 80, 485, 60, 24, X, X
25. input 220, 370, 60, 24, IB, IB
26. input 220, 395, 60, 24, JB, JB
27. input 220, 420, 60, 24, KB, KB
28. input 220, 460, 60, 24, b, b
29. input 220, 485, 60, 24, Y, Y
30. input 360, 370, 60, 24, IC, IC
31. input 360, 395, 60, 24, JC, JC
32. input 360, 420, 60, 24, KC, KC
33. input 360, 460, 60, 24, c, c
34. input 360, 485, 60, 24, Z, Z
35. label 155, 300, 200, 24, "TABULKA POSUNUTI POCATKU 54-59"
36. label 155, 325, 270, 24, "VYSKA BEZPECNOSTI ROVINY VE SMERU OSY
Z"
37. label 440, 375, 200, 24, "SOURADNICE BODU MERENI V OSE X"
38. label 440, 400, 200, 24, "SOURADNICE BODU MERENI V OSE Y"
39. label 440, 425, 200, 24, "SOURADNICE BODU MERENI V OSE Z"
40. label 440, 465, 200, 24, "SMER A HODNOTA POJEZDU MERENI"
41. label 440, 490, 200, 24, "SOURADNICE ROHU USTAVENI"
42. cancel 260, 520, 70, 24
43. ok 170, 520, 70, 24

- Námi předdefinované hodnoty v souboru DefineData.txt jsou:
 1. Zb = 50
 2. X = 0
 3. Y = 0
 4. Z = 0
 5. KA = -5
 6. G = 56
 7. IB = 0
 8. JB = 0
 9. KB = -5
 10. IC = 0
 11. JC = 0
 12. KC = 10
 13. a = 10
 14. b = 10
 15. c = -10

- Parametry pro kontrolu zadaných hodnot obsažené v souboru makra CycleData.mac jsou:
 1. iproc = callback_proc
 2. callback_returncode = 0 ! no errors so far, set to non-zero if an error
 3. get_proc_data iproc, TOOL_NUM, toolnum
 4. get_tool_data toolnum, MTOOL_RADIUS, toolrad
 5. :show_dialog
 6. dialog "CycleData.dlg" ! show the custom dialog
 7. icode = 0
 8. if G < 54 then icode = 1
 9. if G > 59 then icode = 1
 10. if abs(a) <= toolrad then icode = 2
 11. if abs(b) <= toolrad then icode = 3
 12. if icode = 1 then message "NEPLATNA HODNOTA G MOZNO POUZE V ROZMEZI 54-59"
 13. if icode = 2 then message "PRILIS MALA HODNOTA NAJEZDU a"
 14. if icode = 3 then message "PRILIS MALA HODNOTA NAJEZDU b"
 15. if icode > 0 then goto show_dialog

- Pro možnost simulování daných cyklů jsme vytvořili následující program, který je uložen v souboru makra ToolPath.mac:
 1. iop = callback_op
 2. callback_returncode = 0
 3. get_custom_drill_cycle_data x, X1
 4. get_custom_drill_cycle_data y, Y1
 5. get_custom_drill_cycle_data retract_plane, retract
 6. get_op_data iop, tool_num, itool
 7. get_tool_data itool, mtool_radius, toolrad
 - !MERENI VE SMERU OSY X
 8. add_custom_drill_cycle_path feed, 0, 0, Zb-retract
 9. add_custom_drill_cycle_path feed_z, KA-retract
 - 11.1 IF a>0 then add_custom_drill_cycle_path feed_x, a-toolrad
 - 11.2 IF a<0 then add_custom_drill_cycle_path feed_x, a+toolrad
 12. add_custom_drill_cycle_path feed_x, 0
 13. add_custom_drill_cycle_path feed_z, Zb-retract
 - !MERENI VE SMERU OSY Y
 14. add_custom_drill_cycle_path feed, -X1+IB, -Y1+JB, Zb-retract
 15. add_custom_drill_cycle_path feed_z, KB-retract
 - 16.1 IF b>0 then add_custom_drill_cycle_path feed_y, -Y1+JB+(b-toolrad)
 - 16.2 IF b<0 then add_custom_drill_cycle_path feed_y, -Y1+JB+(b+toolrad)
 17. add_custom_drill_cycle_path feed_y, -Y1+JB
 18. add_custom_drill_cycle_path feed_z, Zb-retract
 - !MERENI VE SMERU OSY Z
 19. add_custom_drill_cycle_path feed, -X1+IC, -Y1+JC, Zb-retract
 20. add_custom_drill_cycle_path feed_z, KC-retract
 21. add_custom_drill_cycle_path feed_z, KC-retract+c
 22. add_custom_drill_cycle_path feed_z, Zb-retract
 - !ODJETI DO VYCHOZIHO BODU
 23. add_custom_drill_cycle_path feed, 0, 0, Zb-retract

6.5.1 NC program pro měření ve všech třech osách

V této části popisují pouze tu část NC kódu, která řídí vlastní pohyby obrobkové sondy. Jedná se o část programu mezi body 12. až 20., která se nachází v bodu Celkový vygenerovaný NC program pro automatické měření. Jelikož se zde jedná o měření, které vyžaduje více pohybů sondy je zde i větší počet řádků NC programu, takže výsledný kód je v tomto případě delší. Jednotlivé pohyby obrobkové sondy jsou znázorněné na obrázku Obr. 47 - Zobrazení dráhy sondy pro ustavení ve všech rovinách a číselně označené body 1 a 12. Části programu, před kterými je označení a až f, jsou funkce, které se netýkají pohybu vlastního měření.

a. N90 G65 P9770 Z50.

b. N100 M5

c. N110 M9

d. N120 M31

(OSA X)

1. N130 G65 P9770 Z5. F1000

2. N140 G65 P9770 Z-5. F200

3. N150 G65 P9023 S56. X10. I0

4. N160 G0 Z50. F5000

(OSA Y)

5. N170 G65 P9770 X10. Y-10.

6. N180 G65 P9770 Z5. F1000

7. N190 G65 P9770 Z-5. F200

8. N200 G65 P9023 S56. Y10. J0

9. N210 G0 Z50. F5000

(OSA Z)

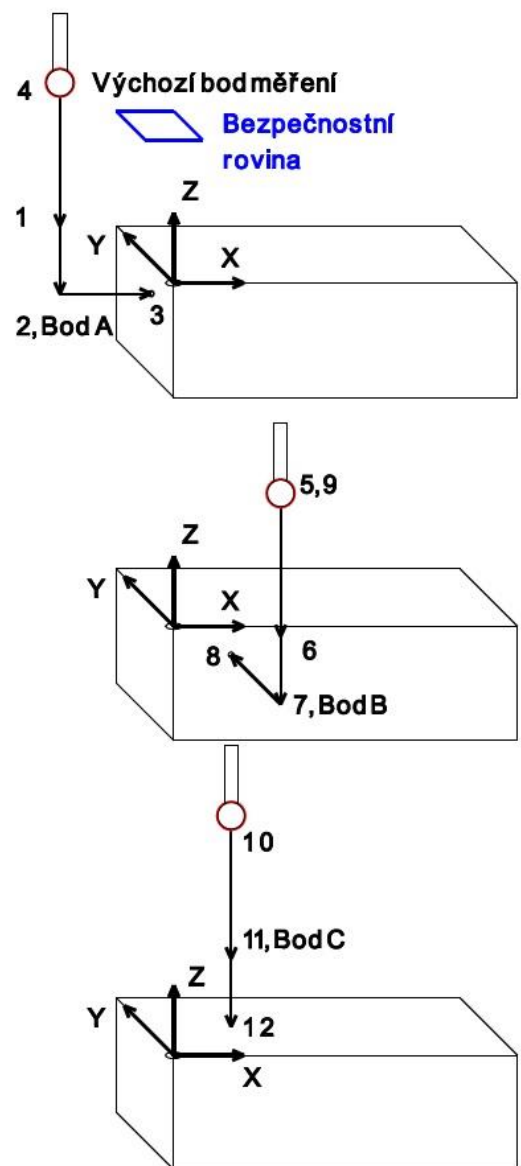
10. N220 G65 P9770 X10. Y10.

11. N230 G65 P9770 Z10. F1000

12. N240 G65 P9023 S56. Z-10. K0

e. N250 #550=#138

f. N260 G0 Z50. F5000



Obr. 47 - Zobrazení dráhy sondy pro ustavení ve všech rovinách

Popsání jednotlivých bodů vygenerovaného NC programu

- a. Na řádce 10 v NC kódu *Celkový vygenerovaný NC program pro automatické měření* jsem se v popisu jednotlivých bodů zmínil, že sonda najede do námi vybraného bodu Výchozí bod měření ze softwaru GibbsCAM, v tomto případě do bodu A pro provedení prvního měření, ale pouze v ose X a Y. Nyní v tomto bodě sjedeme níže ve směru osy Z a to na námi definovanou hodnotu *Výška bezpečnostní roviny v ose Z (Zb)*. Námi zvolená hodnota je v tomto případě Z50. K posunu zde použijeme speciální funkci P9770 bezpečnostní posuv, který nám okamžitě stroj zastaví, pokud se sonda čehokoliv dotkne.
 - b. M5 vypne otáčky vřetena.
 - c. M9 vypne chlazení
 - d. M31 zapne dotykovou sondu.
1. Sonda sjede z *Výška bezpečnostní roviny v ose Z (Zb)* v našem případě Z=50mm do hodnoty, která definuje hlavní Bezpečnostní rovinu v ose Z v našem případě Z=5mm. Opět bude posuv vykonán bezpečnostním posuvem a rychlostí $f = 1\ 000$ [mm/min].
 2. V tomto bodě nám sonda sjede do námi zadané hladiny *Hloubka měření v ose Z*, kde bude probíhat samotné měření. V našem případě $Z = -5$ mm. Vidíme, že hodnota posuvu se zde snížila na $f = 200$ [mm/min].
 3. V bodě tři dojde k provedení vlastního měření, pomocí funkce P9023 a námi zadaných parametrů. Hodnota *S56 určuje nulový bod z tabulky*, pro který bude nastaven měřený bod; X nám určuje *Směr a vzdálenost najetí*, pokud je hodnota kladná pohybuje se sonda v kladném směru osy X; I neboli *Souřadnice bodu ve směru osy X* určuje skutečnou hodnotu, kterou má mít naměřený bod. Po ukončení měření se sonda vrátí do bodu A.
 4. Odjedeme rychloposuvem ve směru osy Z do hodnoty *Výška bezpečnostní roviny v ose Z (Zb)*. Tento pohyb ale není poslední jak je tomu u měření v jednotlivých osách, zde pouze přejedeme do bezpečnostní hladiny, abychom se mohli bez kolizí posunout do dalšího bodu měření.

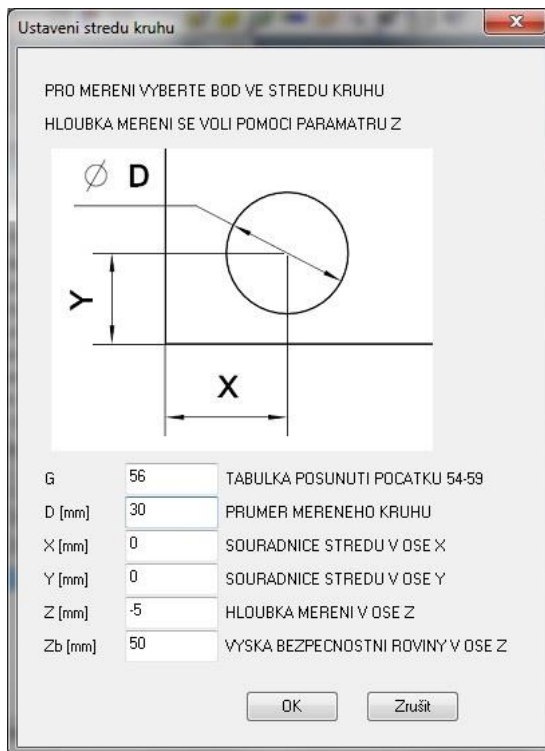
5. Sonda se přemístí v bezpečné rovině do bodu B, ale pouze v osách X a Y.
6. Sonda sjede z *Výška bezpečnostní roviny v ose Z (Zb)* v našem případě $Z=50\text{mm}$ do hodnoty, která definuje hlavní Bezpečnostní rovinu v ose Z v našem případě $Z=5\text{mm}$. Opět bude posuv vykonán bezpečnostním posuvem a rychlostí $f = 1\ 000$ [mm/min].
7. V tomto bodě nám sonda sjede do námi zadané hladiny *Hloubka měření v ose Z*, kde bude probíhat samotné měření. V našem případě $Z = -5$ mm. Vidíme, že hodnota posuvu se zde snížila na $f = 200$ [mm/min].
8. V tomto bodě dojde k provedení měření stejně jako v řádku tři, ale není ve směru osy Y. Rozdíl je pouze v parametrech Y, který opět určuje směr a vzdálenost najetí a J neboli *Souřadnice bodu ve směru osy Y*. Po ukončení měření se sonda vrátí do bodu B.
9. Odjedeme rychloposuvem ve směru osy Z do hodnoty *Výška bezpečnostní roviny v ose Z (Zb)*. Tento pohyb ale není poslední jak je tomu u měření v jednotlivých osách, zde pouze přejedeme do bezpečnostní hladiny, abychom se mohli bez kolizí posunout do dalšího bodu měření.
10. Sonda se přemístí v bezpečné rovině nad bod C, ale pouze v osách X a Y.
11. Sonda sjede z *Výška bezpečnostní roviny v ose Z (Zb)* v našem případě $Z=50\text{mm}$ do hodnoty, která definuje hlavní Bezpečnostní rovinu v ose Z v našem případě $Z=10\text{mm}$. Opět bude posuv vykonán bezpečnostním posuvem a rychlostí $f = 1\ 000$ [mm/min].
12. Stejně jako v řádcích tři a osm, zde dojde k měření v ose Z. Tomu jsou opět uzpůsobené parametry Z a K. Po ukončení měření se sonda vrátí do bodu C. Naměřené hodnoty jsou uloženy do #138.
- e. Dojde k provedení posunutí nulového bodu o naměřené hodnoty uložené v #138.
- f. Odjedeme rychloposuvem ve směru osy Z do hodnoty *Výška bezpečnostní roviny v ose Z (Zb)*.

6.6 Makro pro ustavení vztažného bodu ve středu kruhu

Tento cyklus provede proměření středu kruhu a dle něj nastaví nulový bod v zadaných souřadnicích. Nulový bod tedy nemusí ležet přímo ve středu kruhu, ale může se nacházet tam, kde ho definujeme parametry X a Y.

Tento cyklus slouží k definici vztažného, nebo nulového bodu na střed kruhu, díry nebo kruhové kapsy. Zde už se neměří pouze jeden dotyk, ale k výpočtu středu kruhu se použijí 4 body, změřené postupně ve směru osy X a Y. Pro vypočtený střed kruhu jsou následně zadány hodnoty v ose X a Y, které nám přesně definují vztažný bod, ok kterého se odvíjí souřadný systém. Můžeme také nastavit hodnoty na [0,0] a vytvořit takto ve středu kruhu přímo nulový bod.

Dialogové okno pro definici cyklu ustavení ve směru osy Z vypadá následovně.



Obr. 48 - Dialogové okno pro definici ustavení podle středu kruhové kapsy

- Okno bylo vytvořeno následujícími příkazy v souboru makra CycleData.dlg
 1. dialog "Ustaveni stredu kruhu", 30, 50, 400, 550
 2. image 25, 75, 320, 275, "obr.probe.kruh.bmp"
 3. label 20, 25, 300, 24, "PRO MERENI VYBERTE BOD VE STREDU KRUHU"

4. label 20, 50, 300, 24, "HLOUBKA MERENI SE VOLI POMOCI PARAMATRU Z"
5. label 20, 315, 120, 24, "G"
6. label 20, 340, 120, 24, "D [mm]"
7. label 20, 365, 120, 24, "X [mm]"
8. label 20, 390, 120, 24, "Y [mm]"
9. label 20, 415, 120, 24, "Z [mm]"
10. label 20, 440, 120, 24, "Zb [mm]"
11. input 80, 310, 70, 24, G, G
12. input 80, 335, 70, 24, D, D
13. input 80, 360, 70, 24, X, X
14. input 80, 385, 70, 24, Y, Y
15. input 80, 410, 70, 24, Z, Z
16. input 80, 435, 70, 24, Zb, Zb
17. label 155, 315, 200, 24, "TABULKA POSUNUTI POCATKU 54-59"
18. label 155, 340, 160, 24, "PRUMER MERENEHO KRUHU"
19. label 155, 365, 200, 24, "SOURADNICE STREDU V OSE X"
20. label 155, 390, 200, 24, "SOURADNICE STREDU V OSE Y"
21. label 155, 415, 200, 24, "HLOUBKA MERENI V OSE Z"
22. label 155, 440, 230, 24, "VYSKA BEZPECNOSTNI ROVINY V OSE Z"
23. cancel 260, 480, 70, 24
24. ok 170, 480, 70, 24

- Nými předdefinované hodnoty v souboru DefineData.txt jsou:

1. $D = 0$
2. $X = 0$
3. $Y = 0$
4. $G = 56$
5. $Z = -5$
6. $Zb = 5$

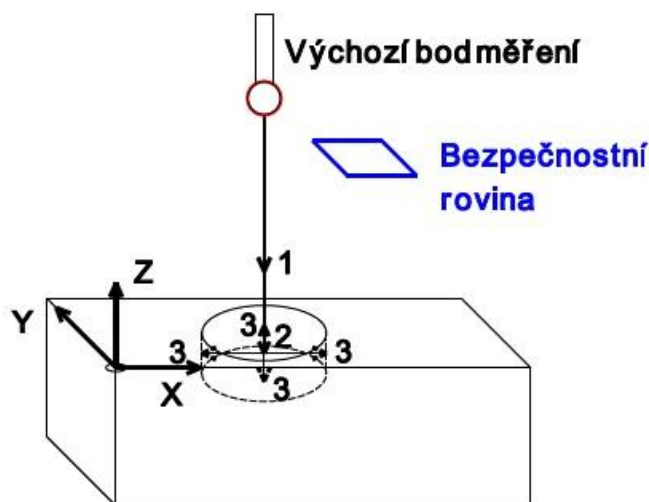
- Parametry pro kontrolu zadaných hodnot obsažené v souboru makra CycleData.mac jsou:
 1. `iprocc = callback_proc`
 2. `callback_returncode = 0` ! no errors so far, set to non-zero if an error
 3. `get_proc_data iproc, TOOL_NUM, toolnum`
 4. `get_tool_data toolnum, MTOOL_RADIUS, toolrad`
 5. `:show_dialog`
 6. `dialog "CycleData.dlg"` ! show the custom dialog
 7. `icode = 0`
 8. `if G < 54 then icode = 1`
 9. `if G > 59 then icode = 1`
 10. `if D < (2*toolrad)+1 then icode = 2`
 11. `if D > 100000 then icode = 2`
 12. `if icode = 1 then message "NEPLATNA HODNOTA G MOZNO POUZE V ROZMEZI 54-59"`
 13. `if icode = 2 then message "PRILIS MALA DIRA NA TENTO NASTROJ"`
 14. `if icode > 0 then goto show_dialog`
- Pro možnost simulování daných cyklů jsme vytvořili následující program, který je uložen v souboru makra ToolPath.mac.
 1. `iop = callback_op`
 2. `callback_returncode = 0`
 3. `get_custom_drill_cycle_data retract_plane, retract`
 4. `get_op_data iop, tool_num, itool`
 5. `get_tool_data itool, mtool_radius, toolrad`
 6. `add_custom_drill_cycle_path feed_z, Z-retract`
 7. `add_custom_drill_cycle_path feed_x, D/2-toolrad`
 8. `add_custom_drill_cycle_path feed_x, -(D/2-toolrad)`
 9. `add_custom_drill_cycle_path feed_x, 0`
 10. `add_custom_drill_cycle_path feed_y, D/2-toolrad`
 11. `add_custom_drill_cycle_path feed_y, -(D/2-toolrad)`
 12. `add_custom_drill_cycle_path feed, 0, 0, Z-retract`

13. add_custom_drill_cycle_path feed, 0, 0, Zb-retract

6.6.1 NC program pro měření vztažných bodů ve středu kruhu

V této části popisují pouze tu část NC kódu, která řídí vlastní pohyby obrobkové sondy. Jedná se o část programu mezi body 12. až 20., která se nachází v bodu Celkový vygenerovaný NC program pro automatické měření. Jednotlivé pohyby obrobkové sondy jsou znázorněné na obrázku Obr. 49 - Zobrazení dráhy sondy pro ustavení podle kruhové kapsy a číselně označené body 1 a 2. Části programu, před kterými je označení a až f, jsou funkce, které se netýkají pohybu vlastního měření.

- a. N90 G65 P9770 Z50.
- b. N100 M5
- c. N110 M9
- d. N120 M31
- 1. N130 G65 P9770 Z5. F1000
- 2. N140 G65 P9770 Z-5. F200
- 3. N150 G65 P9023 D30. S56. I30. J30.
- e. N160 #550=#138
- f. N170 G0 Z50 F5000



Obr. 49 - Zobrazení dráhy sondy pro ustavení podle kruhové kapsy

Popsání jednotlivých bodů vygenerovaného NC programu

- a. Na řádce 10 v NC kódu *Celkový vygenerovaný NC program pro automatické měření* jsem se v popisu jednotlivých bodů zmínil, že sonda najede do námi vybraného bodu *Výchozí bod měření* ze softwaru GibbsCAM, ale pouze v ose X a Y. Nyní v tomto bodě sjedeme níže ve směru osy Z a to na námi definovanou hodnotu *Výška bezpečnostní roviny v ose Z (Zb)*. Námi zvolená hodnota je v tomto případě Z50. K posunu zde použijeme speciální funkci P9770 bezpečnostní posuv, který nám okamžitě stroj zastaví, pokud se sonda čehokoliv dotkne.
- b. M5 vypne otáčky vřetena.
- c. M9 vypne chlazení
- d. M31 zapne dotykovou sondu.

1. Sonda sjede z *Výška bezpečnostní roviny v ose Z (Zb)* v našem případě $Z=50\text{mm}$ do hodnoty, která definuje hlavní Bezpečnostní rovinu v ose Z v našem případě $Z=5\text{mm}$. Opět bude posuv vykonán bezpečnostním posuvem a rychlostí $f = 1\ 000$ [mm/min].
2. V tomto bodě nám sonda sjede do námi zadané hladiny *Hloubka měření v ose Z*, kde bude probíhat samotné měření. V našem případě $Z = -5$ mm. Vidíme, že hodnota posuvu se zde snížila na $f = 200$ [mm/min].
3. V bodě tři dojde k provedení vlastního měření, pomocí funkce P9023 a námi zadaných parametrů. Hodnota *S56 určuje nulový bod z tabulky*, pro který bude nastaven měřený bod; *D* nám určuje *Průměr měřeného kruhu*; parametry *I* a *J* určují skutečnou hodnotu, kterou má vypočítaný střed kruhu. Po ukončení měření se sonda vrátí do bodu 2.
- e. Dojde k uložení naměřených hodnot do Řídicího systému.
- f. Odjedeme rychloposuvem ve směru osy Z do hodnoty *Výška bezpečnostní roviny v ose Z (Zb)*.

6.7 Úpravy Postprocesoru

V této kapitole se zaměřuji na úpravy, které jsem udělal v postprocesoru, abych umožnil správnou funkci ustavovacích cyklu. Základní informace co je to postprocesor a k čemu slouží, jsem se zaměřil již kapitole 2.2.1 Postprocesor.

Daný postprocesor, se kterým pracuji, má přes čtyři tisíce řádků, a popisovat zde veškeré příkazy pro nás nemá takový přínos. Celý postprocesor se nachází v Příloze číslo 2 a zde popisují pouze úpravy, které se přímo týkají správného fungování sond.

Postprocesor se skládá z rutin a příkazů, které nám definují jednotlivé úkony. Rutina se skládá z jednotlivých příkazů a je tvořena tak, aby usnadnila orientaci v jednotlivých částech postprocesoru. Následně je voláme v příslušném pořadí tak, aby došlo ke správnému vygenerování NC kódu.

6.7.1 Použité příkazy

Jak jsem se již zmínil, v této kapitole se nacházejí pouze popsané jednotlivé změny, které jsem udělal v postprocesoru, pro správné fungování měřicí sondy. Více o úpravách

postprocesorů v softwaru GibbsCAM naleznete po registraci na stránkách <https://compost.gibbscam.com>.

6.7.1.1 IF/ELSE/END

V každém postprocesoru používaném v programu GibbsCAM se setkáme s klasickou programovací logikou velmi podobnou jiným programovacím jazykům. Za příkazem IF následuje podmínka, pokud je podmínka splněna, pokračujeme dále a zastavíme se až u příkazu ELSE, kde skočíme na konec k příkazu END. Pokud není podmínka za IF splněna, přeskočíme k příkazu ELSE a do něj pokračujeme až na konec k příkazu END. V postprocesoru je velké množství takto definovaných podmínek.

IF AbsValue? – ptáme se, zdali je programování absolutní G90.

IF CycleType?('MERENI.STRED.OTVOR') ptáme se jestli se jedná o cyklus s názvem MERENI.STRED.OTVOR, pokud ano vykoněj následující příkazy.

IF Flag? ('X') – Ptáme se podle Booleovy logiky, zdali je proměnná X nastavená pravdivě nebo nepravdivě. Pokud je proměnná pravdivá provedou se příkazy za IF.

6.7.1.2 Rutina/RETURN

Aby rutina fungovala správně, musíme jí správně definovat. V prvním řádku se nachází název dané rutiny, podle kterého jí mohu na správném místě v postprocesoru zavolat a nechat proběhnout, tak aby vytvořila mnou požadovaný NC kód. Na dalších řádcích může rutina obsahovat jakýkoliv námi zvolený příkaz. Rutina musí být na posledním řádku ukončená příkazem RETURN.

6.7.1.3 EOL

Tímto příkazem z anglické zkratky End Of Line ukončujeme jednotlivé řádky (věty) v NC kódu.

6.7.1.4 Příkaz OpFieldData#('X')

Veškeré příkazy, které končí znakem mřížka # neboli anglicky sharp, nám v postprocesoru generují pouze číselnou hodnotu. Tato hodnota odpovídá proměnné, která se nachází v závorce v našem případě X. Tato proměnná je definovaná podle námi zadaných údajů do dialogových oken jednotlivých ustavovacích cyklů.

6.7.1.5 Části textu ohraničené apostrofy

Části textu, které jsou ohraničené znaky apostrofů, jsou přímo generovány do NC kódu. Tuto vlastnost nejčastěji používáme při vkládání popisků pro snadnější orientaci v programu, můžou nám ale také pomoci generovat speciální G kód, pro který neexistuje speciální příkaz. V našem případě to může být bezpečnostní posuv 'P9770', tímto způsobem ho můžeme vložit přímo do programu na požadované místo.

6.7.1.6 Příkaz SeqC

Tento příkaz nám na začátku každého řádku (věty) NC kódu generuje označení řádku, nejčastěji ve formě písmene N a následně číslo řádku.

6.7.1.7 Formáty generovaných čísel

V postprocesorech softwaru GibbsCAM můžeme mít definovaný takřka neomezený počet číselných formátů, které následně použijeme při generování čísel. Tímto způsobem můžeme upravovat čísla dle potřeby, například pokud chceme dané číslo zaokrouhlit na námi daný počet desetinných míst, nebo v případě, že proměnná není definovaná, vygenerujeme 0. Abychom každému příkazu nemuseli definovat vlastní formát, vytvoříme si pouze takové, které používáme a přiřadíme jim číslo, které napíšeme k příkazu generující číselnou hodnotu. Definici jednotlivých formátů si můžete prohlédnout v tabulce na **Chyba! Nenalezen zdroj odkazů..** Následně veškerá čísla vygenerovaná tímto příkazem budou mít námi určenou hodnotu. Pro příklad si zde můžeme uvést příkaz OpFieldData#, jenž má přiřazený formát výstupu číslo 10:

FORMAT(OpFieldData#,10)

#10 = '###.##;0'

	1.2	-1.2	120.	.0012	0
###.###	= 1.2	-1.2	120.	.0012	
###.###; 0	= 1.2	-1.2	120.	.0012	0
###.###; 0.	= 1.2	-1.2	120.	.0012	0.
###.###0	= 1.2000	-1.2000	120.0000	.0012	.0000
0##^####	= 0012	-0012	120	0000012	0
##^0000	= 12000	-12000	1200000	0012	0000
##^0000; 0:5	= 12000	-12000	1200000	0010	0
##^##0	= 12000	-12000	1200000	12	0
bbb^0000	= 0012000	-0012000	1200000	0000012	0000000
+###.###	= 1.2	1.2			
-###.###	= 01.2	-1.2			
±###.###	= +1.2	-1.2			
*###.###	= -1.2	1.2			
*+###.###	= -1.2	-1.2			
+*###.###	= 1.2	1.2			
*±###.###	= -1.2	+1.2			
*-###.###	= -1.2	01.2			
3±+*###.###	= -3.6	-3.6			

Obr. 50 - Definice číselných formátů

6.7.1.8 *Jednotlivé příkazy*

SpinOn – Zapne otáčky vřetena.

SpinOff – Vypne otáčky vřetena M5.

Speed – Roztočí vřeteno na danou rychlost například na 3 000 [ot/min] S3000.

CoolOff – Vypne chlazení M9.

SetFlag/SetFlagF – těmito příkazy nastavujeme Booleovu logiku pro daný parametr na pravdivou nebo nepravdivou.

feedIPM – rutina odkazující nás na posuv za minutu G94.

feedIPR – rutina odkazující nás na posuv na otáčku G95.

ZCP1 – najetí do hlavní bezpečnostní roviny ve směru osy Z.

ToolOffset# - opět jde o vygenerovanou hodnotu délky popřípadě průměru nástroje včetně korekcí nástroje, popřípadě sondy.

6.7.2 **Vytvořené a upravené rutiny**

6.7.2.1 *Rutina defProbeCycle*

Tato rutina je ze všech úprav postprocesoru asi nejdůležitější, neboť mi generuje vlastní NC kód ustavovacích cyklů. Rutina obsahuje pět IF cyklů pro všechny naše ustavovací cykly a vygeneruje mi program podle proměnných, které jsem definoval do dialogových oken jednotlivých cyklů.

Tuto rutinu volám v postprocesoru u vrtacích cyklů, neboť jsme i při vytváření jednotlivých maker vycházeli právě z vlastních vrtacích cyklů.

defProbeCycle:

```
IF CycleType?('MERENI.STRED.OTVOR')
  '(MERENI STRED OTVOR)' EOL
  SeqC ' G65 P9770 Z'OpFieldData#('Zb')" EOL
  SeqC SpinOff EOL
  SeqC CoolOff EOL
  SeqC ' M31' EOL
  SeqC ' G65 P9770' ZCP1 ' F1000' EOL
  SeqC ' G65 P9770 Z'OpFieldData#('Z')' F200' EOL
```

```

    SeqC ' G65 P9023 D'OpFieldData#('D') S'OpFieldData#('G') I'OpFieldData#('X')
J'OpFieldData#('Y')" EOL
    SeqC ' #550=#138' EOL
    SeqC ' G0 Z50 F5000' EOL
END
IF CycleType?('MERENI.OSA.X')
    ' (MERENI OSA X)' EOL
    SeqC ' G65 P9770 Z'OpFieldData#('Zb')" EOL
    SeqC SpinOff EOL
    SeqC CoolOff EOL
    SeqC ' M31' EOL
    SeqC ' G65 P9770' ZCP1 ' F1000' EOL
    SeqC ' G65 P9770 Z'OpFieldData#('Z')' F200' EOL
    SeqC ' G65 P9023 S'OpFieldData#('G') X'OpFieldData#('X') I'OpFieldData#('I')" EOL
    SeqC ' #550=#138' EOL
    SeqC ' G0 Z'OpFieldData#('Zb')' F5000' EOL
END
IF CycleType?('MERENI.OSA.Y')
    ' (MERENI OSA Y)' EOL
    SeqC ' G65 P9770 Z'OpFieldData#('Zb')" EOL
    SeqC SpinOff EOL
    SeqC CoolOff EOL
    SeqC ' M31' EOL
    SeqC ' G65 P9770' ZCP1 ' F1000' EOL
    SeqC ' G65 P9770 Z'OpFieldData#('Z')' F200' EOL
    SeqC ' G65 P9023 S'OpFieldData#('G') Y'OpFieldData#('Y') J'OpFieldData#('J')" EOL
    SeqC ' #550=#138' EOL
    SeqC ' G0 Z'OpFieldData#('Zb')' F5000' EOL
END
IF CycleType?('MERENI.OSA.Z')
    ' (MERENI OSA Z)' EOL
    SeqC ' G65 P9770 Z'OpFieldData#('Zb')" EOL
    SeqC ' M5' EOL
    SeqC ' M9' EOL
    {SeqC ' M31' EOL}
    SeqC ' G65 P9770 Z'OpFieldData#('C')' F1000' EOL
    SeqC ' G65 P9023 S'OpFieldData#('G') Z'OpFieldData#('Z') K'OpFieldData#('K')" EOL
    SeqC ' #550=#138' EOL

```

```

SeqC ' G0 Z'OpFieldData#('Zb')' F5000' EOL
END
IF CycleType?('MERENI.OSA.XYZ')
  ' (MERENI OSA XYZ)' EOL
  SeqC ' G65 P9770 Z'OpFieldData#('Zb')" EOL
  SeqC ' M5' EOL
  SeqC ' M9' EOL
  ' (OSA X)' EOL
  SeqC ' G65 P9770' ZCP1 ' F1000' EOL
  SeqC ' G65 P9770 Z'OpFieldData#('KA')' F200' EOL
  SeqC ' G65 P9023 S'OpFieldData#('G')' X'OpFieldData#('a')' I'OpFieldData#('X')" EOL
  SeqC ' G0 Z'OpFieldData#('Zb')' F5000' EOL
  ' (OSA Y)' EOL
  SeqC ' G65 P9770 X'OpFieldData#('IB')' Y'OpFieldData#('JB')" EOL
  SeqC ' G65 P9770' ZCP1 ' F1000' EOL
  SeqC ' G65 P9770 Z'OpFieldData#('KB')' F200' EOL
  SeqC ' G65 P9023 S'OpFieldData#('G')' Y'OpFieldData#('b')' J'OpFieldData#('Y')" EOL
  SeqC ' G0 Z'OpFieldData#('Zb')' F5000' EOL
  ' (OSA Z)' EOL
  SeqC ' G65 P9770 X'OpFieldData#('IC')' Y'OpFieldData#('JC')" EOL
  SeqC ' G65 P9770 Z'OpFieldData#('KC')' F1000' EOL
  SeqC ' G65 P9023 S'OpFieldData#('G')' Z'OpFieldData#('c')' K'OpFieldData#('Z')" EOL
  SeqC ' #550=#138' EOL
  SeqC ' G0 Z'OpFieldData#('Zb')' F5000' EOL
RETURN

```

6.7.2.2 Rutina *ckMacroDrilling*

Tato rutina se nachází v postprocesoru právě z důvodu použití Booleovské logiky. Vytvořil jsem si proměnnou *MacroDrilling*, u které následně definuji, jestli je pravdivá nebo, což probíhá pomocí IF cyklů, kde se ptám, jestli je daná operace jeden z mnou vytvořených cyklů. V případě že ano proměnná *MacroDrilling* je nastavena jako pravdivá.

Takto nastavenou proměnnou následně použiji v různých částech postprocesoru, kde mi slouží v jednotlivých IF cyklech a upravuje generovaný NC program.

```

ckMacroDrilling: {27/Feb/2015}
  SetFlagF('MacroDrilling')
  IF CycleType?('MERENI.STRED.OTVOR')
    SetFlag('MacroDrilling')

```

```

END
IF CycleType?('MERENI.OSA.X')
    SetFlag('MacroDrilling')
END
IF CycleType?('MERENI.OSA.Y')
    SetFlag('MacroDrilling')
END
IF CycleType?('MERENI.OSA.Z')
    SetFlag('MacroDrilling')
END
IF CycleType?('MERENI.OSA.XYZ')
    SetFlag('MacroDrilling')
END
RETURN

```

6.7.2.3 Rutina *getSpeed*

Tato rutina mi nastavuje požadované otáčky pro právě prováděnou operaci, samozřejmě při použití sondy není otáčení vřetene zapotřebí, tudíž při pravdivé hodnotě proměnné MacroDrilling se nevygeneruje žádný výstup (no output).

```

getSpeed:
    IF Flag?('MacroDrilling')
        {no output}
    ELSE
        IF Flag?('UseDrillCycle') AND RigidTap?
            {no output}
        ELSE
            Speed
        END
    END
RETURN

```

6.7.2.4 Rutina *getSpeedSpinOn*

Zde vidíme velmi podobnou rutinu jako v případě *getSpeed*, ale s tím rozdílem, že zde dochází ještě k roztočení vřetena na rozdíl od rutiny *getSpeed*, kde se vřeteno otáčelo a pouze jsme upravovali otáčky.

```

getSpeedSpinOn:
    IF Flag?('MacroDrilling')

```

```

        {no output}
    ELSE
        IF Flag?('UseDrillCycle') AND RigidTap?
            {no output}
        ELSE
            Speed SpinOn
        END
    END
RETURN

```

6.7.2.5 Rutina doOfstOn

Tuto rutinu použijeme k načtení délkových a průměrových korekcí nástroje v našem případě sondy.

```

doOfstOn:
    IF Flag?('MacroDrilling')
        ' G43' ' H' ToolOffset# ' D' ToolOffset#
    ELSE
        IF AnyRotaryMoves? AND ABCToolPath? AND Flag?('SupportTCP')
            ' G43.4' ZCP1 ToolOfst
            SetOutputPartSpace SetFlag('TCPon')
        ELSE
            ' G43' ' H' ToolOffset# ' D' ToolOffset# ZCP1
        END
    END
RETURN

```

6.7.2.6 Rutina doSpinOff

U všech operací obrábění musíme na konci operace popřípadě při výměně nástroje vypnout otáčení vřetena (M5) a rozdílem nejsou ani vrtací cykly, ze který vycházíme. Sonda se ale samozřejmě netočí, takže zde není ani nutné vypínat otáčky. Je zde ale nutné ukončit cyklus sond právě to nám zde dělá G kód M32, který zde generuji do programu pomocí apostrofů.

```

doSpinOff:
    IF Flag? ('MacroDrilling')
        SeqLabC' M32' EOL
    ELSE
        SpinOff

```

```
END  
RETURN
```

6.7.2.7 Rutina doFeedMode

Veškeré vrtací operace v tomto postprocesoru jsou nastaveny na posuv na otáčku, ale v našem případě, kdy se sonda netočí, musíme přepnout posuv na minutový.

```
doFeedMode:
```

```
  IF Drilling?  
    IF Flag?('MacroDrilling')  
      feedIPM  
    ELSE  
      feedIPR  
    END  
  ELSE  
    feedIPM  
  END  
RETURN
```

7. Praktická aplikace

Výstupem praktické části mé práce jsou makra, která fungují podle zadaných parametrů vypracování zákazníka, jež je v tomto případě společnost Jaromír Biem – Kovovýroba sídlící v Ciboušově u Klášterce nad Ohří. V tomto podniku je využijeme na stroji Feeler VMP-45(A) s řídicím systémem Fanuc 0i-MD. I když zatím veškeré simulace proběhly bez problémů a generovaný výstup souhlasí se zadaným, tak úplnou jistotu o funkčnosti budeme mít až po dlouhodobějším užívání přímo v praxi, které momentálně probíhá.

Praktická část mé diplomové práce je určena pro specifický podnik a stroj, ale to neznamená, že může být použita pouze tam. Tyto měřicí makra budou fungovat po úpravě postprocesoru na jakémkoliv stroji s řídicím systémem Fanuc 0i. Pokud jde o použití na jiných řídicích systémech, tam už budeme muset provést úpravy podle daného systému a jeho možností z hlediska ustavovacích nebo měřících cyklů, ale princip funkce makra bude pořád stejný.

Přesnou funkci práce jednotlivých maker pro ustavení můžeme vidět v Příloze číslo 3, která obsahuje manuály pro definování jednotlivých cyklů.

8. Závěr

Cílem této práce bylo popsání a úvod do problematiky měření s pomocí obrobkových sond na CNC obráběcích strojích. Jedná se o velmi široké téma, které je dnes často zmiňované, neboť s narůstajícími možnostmi strojů roste i požadavek právě na využívání sond k ustavení a kontrole výrobku. Pro mnoho firem je velmi atraktivní zejména proto, že se v mnoha případech správného využití sondy docílí k drastickému snížení času pro upnutí součásti a zlepší se kontrola zmetkovitosti přímo na stroji. Zároveň je méně náchylné k chybě operátora stroje, což je v dnešní době také velká výhoda.

Dále jsem ve své práci popsal základní cykly, které se dnes využívají při měření na CNC obráběcích strojích. Také jsem se zaměřil na zařízení a vybavení, s jehož pomocí se dnes provádí měření v průmyslových podnicích, jako jsou například obrobkové sondy, u kterých jsem se také snažil nastínit principi jejich fungování.

V praktické části své práce jsem řešil problematiku právě ustavovacích cyklů a jejich implementaci do CAM softwaru GibbsCAM. Toto řešení nám šetří čas přípravy výroby a čas potřebný pro přesné ustavení obrobku do stroje. Zároveň je zde také ukázán postup přenosu ovládání obrobkových sond do CAM softwaru, což je asi největším přínosem této práce, neboť klasický postup by byl ruční ovládání sondy, popřípadě vyplňování dialogových oken jednotlivých cyklů a jejich vkládání do NC kódu až na stroji, to nám ale zároveň zvyšuje čas, kdy stroj nepracuje. Při programování ustavovacích cyklů společně s dalšími operacemi přímo v CAMu mám rovněž možnost provést simulaci, což mi snižuje riziko kolize sondy.

Jak jsem již zmínil část implementace do softwaru GibbsCAM, zde ukazuji na ustavovacích cyklech, je to ale více méně pouze možný příklad daného využití, který má za úkol ukázat postup a možnosti pro další využití. Neboť takřka veškeré měřicí a ustavovací cykly, které jsou obsaženy v řídicím systému stroje, jsem tímto způsobem schopen implementovat i do CAMu. Jediné omezení v tomto případě je tedy sám řídicí systém, ale i tento problém jsem zde řešil a to s pomocí doplňkového plug-inu EasyProbe.

9. Seznam literatury

1. **Ing. Staněk, Vlastimil.** jak se stát spokojeným uživatelem? *t-support.cz*. [Online] [Citace: 10. květen 2015.] <http://www.t-support.cz/kat/jak-se-stat-spokojenym-uzivatelem>.
2. **Miroslav, Sadílek.** Postprocesor - slabé místo CAM systémů? *mmspektrum.com*. [Online] [Citace: 10. květen 2015.] <http://www.mmspektrum.com/clanek/postprocesor-slabe-misto-cam-systemu.html>.
3. **Šrůtková, Vendula.** Úvod do programování (VBA) . *Gymnázium, Česká Třebová*. [Online] [Citace: 31. květen 2015.] <http://www.gymnct.cz/vesr/vb.pdf>.
4. **technology-support s.r.o.** Základní technologie. *t-support.cz*. [Online] [Citace: 16. Duben 2015.] <http://www.t-support.cz/?rubrika=1120>.
5. **Ing. Staněk, Vlastimil.** proč měřit na CNC obráběcím stroji? *t-support.cz*. [Online] [Citace: 18. Duben 2015.] <http://www.t-support.cz/kat/proc-merit-na-cnc-obrabecim-stroji-3>.
6. **Ing. Michalíček, Michal.** Predikace pracovní přesností CNC obráběcích strojů. *Dizertační práce*. Brno : Vysoké učení technické v Brně, 2013.
7. **Žák, František.** Kontrola a kalibrace obráběcích a souřadnicových měřicích strojů. *Bakalářská práce*. Brno : Vysoké učení technické v Brně, 2014.
8. **ISO.** ISO 230-2:2014(en). *ISO.org*. [Online] [Citace: 25. Duben 2014.] <https://www.iso.org/obp/ui/#iso:std:iso:230:-2:ed-4:v1:en>.
9. *Souřadnicové měřicí stroje*. **Katedra výrobních systémů - Technická univerzita v Liberci**.
10. **Renishaw.** SPRINT™: měření prizmatických dílců. *Renishaw*. [Online] [Citace: 13. červen 2015.] <http://resources.renishaw.com/en/details/brochure-sprint-high-speed-scanning-system--56249>.
11. —. Slovníček práce se sondami a terminologie. *Renishaw*. [Online] [Citace: 11. červen 2015.] <http://www.renishaw.cz/cs/slovnicek-prace-se-sondami-a-terminologie--12479>.
12. **Bc. Kohút, Josef.** Použití měřicích sond (nástrojové a obrobkové) pro stroje z produkce TOS Kuřim. Brno : Fakulta strojního inženýrství.

13. **Heidenhain.** 3-D dotykové sondy pro obráběcí stroje. *Heidenhain*. [Online] [Citace: 11. červen 2015.] [http://www.heidenhain.cz/cs_CZ/php/dokumentace-informace/prospekty/popup/media/media/file/view/file-0475/file.pdf#page=.](http://www.heidenhain.cz/cs_CZ/php/dokumentace-informace/prospekty/popup/media/media/file/view/file-0475/file.pdf#page=)
14. **Renishaw.** Proč si vybrat RENGAGE™? *Renishaw*. [Online] 11. červen 2015. [http://resources.renishaw.com/cs/details/brozura-proc-si-vybrat-rengage--61430.](http://resources.renishaw.com/cs/details/brozura-proc-si-vybrat-rengage--61430)
15. **Siemens.** Siemens - Industry Online Support. *Program manual - SINUMERIK 840D sl / 828D - Measuring Cycles*. [Online] [Citace: 5. duben 2015.] [https://support.industry.siemens.com/cs/document/56950548?lc=en-CZ.](https://support.industry.siemens.com/cs/document/56950548?lc=en-CZ)
16. **Štulpa, Miroslav.** *Programování obráběcích strojů*. Praha : Grada Publishing a.s., 2015. 978-80-247-5269-3.
17. **HeidenHain.** www.heidenhain.de. *Příručka uživatele - Programování cyklů - iTNC 530*. [Online] [Citace: 6. duben 2015.] [http://content.heidenhain.de/doku/tnc_guide/pdf_files/iTNC530/34049x-05/zyklen/670_388-C0.pdf.](http://content.heidenhain.de/doku/tnc_guide/pdf_files/iTNC530/34049x-05/zyklen/670_388-C0.pdf)
18. **Renishaw.** Inspection Plus - software for machining centres. *Renishaw*. [Online] [Citace: 17. květen 2015.]
19. —. EasyProbe - probe software for machining centres. <http://www.renishaw.com>. [Online] 3. květen 2015. [http://www.renishaw.com/en/easyprobe-probe-software-for-machining-centres--6092.](http://www.renishaw.com/en/easyprobe-probe-software-for-machining-centres--6092)
20. **SolidCAM.** SolidCAM Solid Probe module. *SolidCAM*. [Online] [Citace: 5. květen 2015.] [http://www.solidcam.com/us/cam-solutions/solid-probe/.](http://www.solidcam.com/us/cam-solutions/solid-probe/)
21. **Renishaw.** Productivity+™ - PC based probe software for machining centres. *Renishaw*. [Online] [Citace: 17. květen 2015.] [http://www.renishaw.com/en/productivity-pc-based-probe-software-for-machining-centres--6252.](http://www.renishaw.com/en/productivity-pc-based-probe-software-for-machining-centres--6252)
22. —. Průvodce užití sond RENISHAW na strojích VASON CZ s.r.o.
23. **technology-support s.r.o.** proč zvolit zrovna GibbsCAM? *t-support.cz*. [Online] [Citace: 3. květen 2015.] [http://www.t-support.cz/kat/proc-volit-zrovna-gibbscam.](http://www.t-support.cz/kat/proc-volit-zrovna-gibbscam)

24. **3D Systems.** About GibbsCAM Macros. *Gibbs Makros Wiki*. [Online] [Citace: 3. květen 2015.]

25. —. Makros Examples. *Gibbs Makros Wiki*. [Online] [Citace: 3. květen 2015.]
<https://macros.gibbscam.com/index.php/Category:Examples>.