

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽINÝRSTVÍ
ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY

FACULTY OF MECHANICAL ENGINEERING
DEPARTMENT OF DESIGN MACHINE AND EQUIPMENT

SOFTWAREOVÉ ROZHRANIE PRO UŽIVATELSKÉ APLIKACE LASER TRACKERU

A SOFTWARE INTERFACE FOR A LASER TRACKER USER APPLICATION

DIPLOMOVÁ PRÁCA

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Peter Kozáček

VEDÚCÍ PRÁCE

SUPERVISOR

Ing. Josef Knobloch, MSc.

BRNO 2015

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav výrobních strojů, systémů a robotiky

Akademický rok: 2014/15

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Bc. Peter Kozáček

který/která studuje v **magisterském studijním programu**

obor: **Výrobní stroje, systémy a roboty (2301T041)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce :

Softwarové rozhraní pro uživatelské aplikace laser trackeru

v anglickém jazyce:

A software interface for a laser tracker user application

Stručná charakteristika problematiky úkolu:

Mobilní souřadnicové přístroje laser trackery jsou velmi nákladná zařízení nacházející uplatnění v mnoha průmyslových oborech. Díky vývoji v posledních letech a související rostoucí přesnosti je možné trackery využívat i ve velmi náročných aplikacích jako je stavba a měření geometrie obráběcích strojů, pro které nejsou na trhu dostupné softwary. V této práci proto bude navrženo softwarové rozhraní pro laser tracker, jako první krok v projektu, jehož cílem je tvorba vlastních uživatelských aplikací upravených podle potřeb výrobců obráběcích strojů.

Cíle diplomové práce:

Provést rešerši možností tvorby uživatelských aplikací.

Ve studentem zvoleném přístupu a programovacím jazyce vytvořit základní návrh rozhraní s ohledem na další funkční rozšiřitelnost.

Vytvořit jednu variantu pro sbírání interferometrických měření délek, která bude moci být používána pro multilaterační algoritmus.

Vytvořit jednu variantu pro vyhodnocení základních geometrických tolerancí používaných při stavbě obráběcích strojů na základě 3D souřadnic.

Seznam odborné literatury:

<http://www.innovmetric.com/>

<http://www.leica-geosystems.com>

- WECK, Manfred, BRECHER, Christian. Werkzeugmaschinen : Konstruktion und

Berechnung. 2006. überarb. Auflage. Verlag Berlin Heidelberg : Springer, 2006. 701 s. ISBN

3-540-22502-1.

- Marek, Jiří, MM Průmyslové spektrum: Konstrukce CNC obráběcích strojů. 2006. Speciální vydání. Dostupný z WWW: . ISSN 1212-2572.

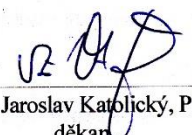
Vedoucí diplomové práce: Ing. Josef Knobloch, MSc

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2014/15.

V Brně, dne 17. 12. 2014




doc. Ing. Petr Blecha, Ph.D.
ředitel ústavu


doc. Ing. Jaroslav Katolický, Ph.D.
děkan

Anotácia

Súčasťou stavby strojov je vyhodnocovanie geometrických presností jednotlivých funkčných častí stroja. Tento proces je však veľmi zdĺhavý a náročný. Jedným z dôvodov spomínaných problémov je nedostatočné softvérové vybavenie(prispôsobenie) pre meracie zaradenia na získavanie geometrických presností. Diplomová práca sa zameriava na možnosti vývoja a tvorbu vlastnej aplikácie ,podľa požiadaviek zákazníka, pre meranie geometrických presností s Laser Trackerom AT901 od firmy Leica. Cieľom je vytvoriť základnú aplikáciu vhodnú na meranie geometrických presností, ktorá tento proces urýchli, bude užívateľsky prívetivá, jednoduchá a otvorená pre ďalšie rozšírenie.

Kľúčové slová

Laser Tracker, geometrická presnosť, AT901

Annotation

For assembly of product machine is necessary to evaluate a geometric accuracy of individual functional parts of the machine. This process is very demanding and time consuming. One of the reasons of mentioned problems is the lack of software (adaptation) for measuring of the accuracy of geometrical precision. This theses (diploma work) focuses at facilitating the development and creation of own application based on customer requirements and for measuring geometric accuracy with the Laser Tracker AT901 from Leica company. The aim is to create a basic application suitable for measurement of geometric accuracy, which would speed up this process, would be user-friendly, simple and open for further expansion.

Key words

Laser Tracker, geometric accuracy, AT901

Bibliografická citace:

KOZÁČEK, P. *Softwarové rozhraní pro uživatelské aplikace laser trackeru*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2015. 77 s. Vedoucí diplomové práce Ing. Josef Knobloch, MSc.

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Softwarové rozhraní pro uživatelské aplikace laser trackeru jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.“

V Brně dne: 25. května 2015

.....
podpis autora

Obsah

1	Úvod.....	13
2	Presnosť.....	14
2.1	Základné pojmy	14
2.2	Druhy presností.....	15
2.3	Volumetrická presnosť.....	15
3	Skúšky obrábacích strojov	17
3.1	ČSN ISO 230	19
3.2	Geometrická presnosť podľa ČSN ISO 230-1	19
3.2.1	Priamosť.....	20
3.2.2	Rovinnosť	20
3.2.3	Rovnobežnosť	20
3.2.4	Kolmosť priamok a rovín.....	21
3.2.5	Otáčanie	21
4	Chyby a neistoty merania	22
4.1	Chyby merania	22
4.2	Neistoty merania.....	23
4.2.1	Neistota typu A	23
4.2.2	Neistota typu B	24
4.2.3	Štandardná kombinovaná neistota.....	25
4.2.4	Rozšírená kombinovaná neistota.....	25
5	Trackovacie interferometry	26
5.1	IFM, ADM	26
5.2	Leica Absolute Tracker AT901	27

5.3	Leica Absolute Tracker AT402	29
5.4	Leica Absolute Tracker AT960.....	29
5.5	Tracker 3.....	30
5.6	RADIAN.....	30
5.7	Laser TRACER-NG	31
6	Softvér pre meranie s Laser Trackerom	33
6.1	emScon.....	33
6.2	InnovMetric PolyWorks.....	34
6.3	Trac-cal	36
7	Možnosti vývoja vlastnej aplikácie	37
8	Programovacie jazyky.....	39
8.1	Jazyk C	39
8.2	Jazyk C++	39
8.3	Visual Basic.....	40
8.4	C# (C sharp)	40
8.5	Microsoft Visual Studio	41
9	Návrh aplikácie	43
9.1	Testovacia aplikácia pre zisťovanie možností SDK.....	45
9.2	Postup merania	47
9.3	Návrh objektového modelu	47
9.4	Hlavné okno	48
9.5	Trieda LaserTracker.....	50
9.5.1	Nastavenia Laser Trackeru.....	51
9.6	Real-time zobrazenie	53
9.7	Meranie	54

9.8	Bodové meranie	58
9.9	Protokol.....	59
9.9.1	Vyhodnotenie pre rovinnosť.....	59
9.9.2	Výpočet neistôt meraní	60
9.10	Nastavenia.....	61
9.11	Export	63
10	Ďalší vývoj	64
11	Záver	65
12	Použitá literatúra, zdroje	67
13	Príloha A : Trieda Laser Tracker	69
14	Príloha B : Gnuplot.....	73
15	Príloha C : Namerané hodnoty bodu	74
16	Príloha D : Načítanie xlsx súboru pomocou Matlabu	75
17	Zoznam obrázkov	76
18	Prílohy	77

1 Úvod

Diplomová práce vznikla na popud potreby aktuálne riešeného projektu výskumného centra INTEMAC, Ústavu výrobných strojov, systémov a robotiky a štyroch českých výrobcov veľkých obrábacích strojov. Stavba veľkých strojov trvá niekoľko týždňov až mesiacov a je v mnohých prípadoch vykonávaná dvakrát – na montáži u výrobcu a po demontovaní znovu u zákazníka. Efektívne, spoľahlivé a užívateľsky prívetivé meracie postupy sú zásadné a môžu stavbu výrazne urýchliť, a preto aj zlacnieť.

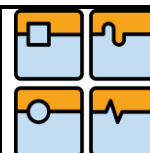
Práca je zameriavaná na zaisťovanie meranie a vyhodnocovania geometrickej presnosti stroja s využitím meracieho zariadenia Laser Tracker. Konkrétne sa bude jednať o typ AT901 od firmy Leica. K danému zariadeniu je štandardne dodávaný softvér emScon so základnými ovládacími funkciami.

Ďalším softvérom na meranie s trackerom je inšpekčný softvér Polyworks. V prípade Polyworks-u sa jedná o veľmi sofistikovaný, drahý systém, ktorý nie je primárne určený k meraniu presností na obrábacích strojoch.

Úlohou diplomovej práce bude vytvorenie vlastnej aplikácie určenú na meranie pomocou Laser Tracker AT901. Aplikácia bude vytváraná po jednotlivých etapách, od rozchodenia komunikácie, získavania dát až po vyhodnocovanie a vytváranie protokolu o meraní. Celá aplikácia bude koncipovaná tak, aby bolo možné kedykoľvek ju ďalej rozšíriť o potrebné funkcie.

Požiadavky na aplikáciu :

- Priame zobrazovanie hodnôt
- Meranie dĺžok, rovinnosti, rovnobežnosti
- Export dát z aplikácie
- Možnosť ďalšieho rozšírenia
- Čo najširšia možnosť používania (vzhľadom k OS)
- Nízka cena



2 Presnosť

Pod pojmom presnosť obrábania rozumieme stupeň zhody obrobenej súčiastky s jej geometrickým vzorom, ktorý je určený výkresom. Čím je táto zhoda väčšia, tým je presnosť obrábania vyššia. Na druhej strane príslušná odchýlka (chyba) predstavuje stupeň nezhody obrobenej súčiastky s príslušným geometrickým vzorom a nazývame ju nepresnosť. [1]

Nepresnosť [2] výrobných strojov je vyjadrená odchýlkou výsledkov jeho práce (obrobku) od jeho teoreticky bezchybného modelu. Z ekonomických dôvodov sa často presnosť výrobných strojov meria nepriamo, nie na obrobku ale na stroji.

2.1 Základné pojmy

Odchýlka tvaru Δt [2] je určená rozdielom tvaru skutočnej plochy vzhľadom na ideálny tvar súčiastky. Do odchýlky tvaru sa zahrňuje aj vlnitosť skutočnej plochy spôsobená hlavne kmitaním sústavy stroj-nástroj-obrobok .

Odchýlka polohy Δp je daná nepresnosťou vzájomného vzťahu dvoch prvkov. Jej veľkosť závisí od spôsobu obrábania, pracovných podmienok, presnosti a tuhosti obrábacieho stroja. Pri viacerých súčiastkach pôsobí aj vplyv montáže.

Odchýlka rozmeru Δr predstavuje rozmerovú nepresnosť skutočnej plochy vzhľadom na rozmer menovitej plochy. Matematicky ide o rozdiel medzi skutočným rozmerom a menovitým rozmerom.

Pri výrobe súčiastky nie je možné vyrobiť „dokonalú“ súčiastku (s absolútnou presnosťou). Rozmery obrobenej súčiastky sa vždy líšia od menovitej hodnoty(požadovanej). Daná odchýlka je závislá na mnohých činiteľoch, z ktorých je najdôležitejší je použitá operácia.

2.2 Druhy presností

Samotná presnosť je pojem, ktorý v sebe zahrňuje množstvo čiastočných presností ktoré sú súčasťou stroja. Všetky tieto presnosti sú kategorizované do štyroch skupín [2] :

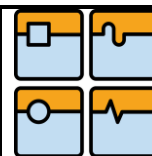
- a) Presnosť geometrická – vyjadruje odchýlku rozmerov, tvarov a vzájomnej polohy tých častí a skupín výrobného stroja, ktoré určujú polohu a relatívnu dráhu nástroja a obrobku. Geometrická presnosť sa zisťuje v nezaťaženom stave stroja a stanovujú ju obvykle platné normy.
- b) Presnosť kinematická – vyjadruje chyby vo väzbe pohybov v dôsledku výroby a montáže častí výrobných strojov, ovplyvňujúce kinematické reťazce pre pracovné pohyby jednotlivých uzlov výrobného stroja.
- c) Presnosť nastavenia nástroja
- d) Pracovná presnosť – ktorá sa skúša vyrobením vzorového kusu, u ktorého sa potom premeriava tvar, rozmery a vzájomná poloha plôch. Na presnosť práce výrobného stroja má vplyv tiež zaťaženie a teplota.

S rozvojom výpočtovej techniky sa tieto kategórie rozrástli o ďalšiu skupinu, a to volumetrická presnosť(ale v mnohých literatúrach sa uvádza oddelene).

2.3 Volumetrická presnosť

Veľký význam má v súčasnosti presnosť polohovania nie je meraná oddelene v jednotlivých osách, ale pri ich súčinnosti v priestore. Hovoríme o volumetrickej presnosti.

Volumetrické chyby zahrňujú chyby polohové, chyby rovinnosti, kolmosti a uhlové chyby. Volumetrická presnosť je reprezentovaná mapou pozície a orientácie chybových vektorov nástroja pri pohybe v priestore obrábacieho stroja. Existujú dva spôsoby merania



volumetrickej presnosti v smere priestorovej diagonály pracovného priestoru obrábacieho centra :

- súvislý spôsob merania diagonály
- sekvenční spôsob merania diagonály

Najvyšším prínosom volumetrie je jednoznačné zmapovanie geometrických chýb u veľkých obrábacích strojov a centier. U týchto typov strojov sa nečastejšie prejavuje chyba kolmosti pri pohybu v lineárnej ose. V prípade päťosých strojov sa táto chyba najviac prejavuje pri pohybu rotačných os v závislosti na zväčšujúcej sa vzdialenosti.

3 Skúšky obrábacích strojov

Aby sme mohli posúdiť presnosť jednotlivých častí strojov (a celého stroja), je potrebné tento stroj odtestovať. Tieto skúšky sú presne stanovené príslušnou normou. V nasledujúcej časti si uvedieme jednotlivé skúšky, ktoré sa týkajú obrábacích strojov.

Skúšanie obrábacích strojov [3] :

a) Preberacia skúška :

1. U výrobcu
2. U zákazníka

b) Štatistická prebierka – skúmanie indexov spôsobilosti procesov a obrábania

c) Prototypová skúška – posudzovanie zhody (CE)

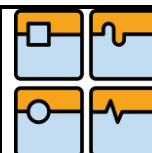
Preberacie skúšky obrábacích strojov :

a) Technické údaje a vlastnosti :

- Rozmery
- Zdvihy
- Otáčky a posuvy

b) Geometrická presnosť :

- Priamosť
- Kolmosť
- Rovnobežnosť
- Rovinnosť
- Súosovosť
- Obvodové hádzanie



c) Presnosť :

- Polohovania
- Pracovná presnosť
- Výrobná presnosť

d) Iné :

- Hluk
- Spoľahlivosť
- Chod naprázdno
- Dynamické vlastnosti
- Oteplenie
- Tuhosť

Počas skúšok musia byť stroje chránené pred prúdením vzduchu, tepelným žiarením a inými zdrojmi tepla. Ak by výsledky merania mohli v rozhodujúcej miere ovplyvniť teplo vznikajúce pri práci stroja, je potrebné tieto merania vykonať až po ukončení práce stroja pri chode naprázdno v súlade s predpísanými údajmi. Na teplotu pracovného prostredia má byť stabilizovaná aj teplota meracích prístrojov. Podľa potreby sa vykoná korekcia vplyvu teploty na výsledky merania. Celé postupy ako postupovať pri meraní chýb a aj ich vyhodnocovanie popisuje norma ČSN ISO 230.

3.1 ČSN ISO 230

Norma ČSN ISO 230 je rozsiahly súbor dokumentov zaoberajúcich sa vnútornými mechanickými vlastnosťami stroja, vplyv teplôt na chod stroja a tiež vplyvu stroja na samotné okolie.

ČSN ISO 230 [4]:

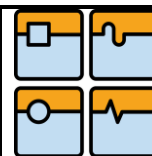
- 230-1 → meranie geometrickej presnosti
- 230-2 → Stanovenie presnosti nastavenia polohy v číslicovo riadených osách
- 230-3 → Meranie tepelného chovania a teplôt
- 230-4 → Stanovenie presnosti kruhovej interpolácie
- 230-5 → Meranie akustického tlaku a výkonu
- 230-6 → Stanovenie presnosti nastavenia na diagonálach pracovného priestoru
- 230-7 → Stanovenie presnosti chodu vretena za rotácie
- 230-8 → Vibrácie
- 230-9 → Nepresnosti merania
- 230-10 → Meranie na samotných obrábacích strojoch

V našej práci sa konkrétne budeme zameriavať na meranie a vyhodnocovanie geometrickej presnosti (teda ISO ČSN230-1) V nasledujúcich podkapitolách si priblížime jednotlivé pojmy.

3.2 Geometrická presnosť podľa ČSN ISO 230-1

Geometrická presnosť zahŕňa odchýlky tvaru a polohy funkčných častí stroja, vzájomnú polohou funkčných častí alebo plôch. Meraní geometrickej presnosti sa skladá z jednotlivých meraní :

- Priamosti
- Kolmosti, pravouhlosti
- Rovnobežnosti
- Rovinnosť
- Otáčania (obvodové hádzanie)



3.2.1 Priamost'

Geometrické skúšky týkajúceho sa priamosti :

- Priamost' čiary v rovine
- Priamost' súčiastky
- Priamost' pohybu

Skúška priamosti sa vykonáva predovšetkým u vodiacich plôch, zväčša vodováhou (libelou).

3.2.2 Rovinnosť

Plocha je považovaná za rovinnou v danom rozsahu merania, keď všetky jej body ležia medzi dvoma rovinami vzájomne vzdialenými o určenú hodnotu a rovnobežnými s hlavným smerom plochy.

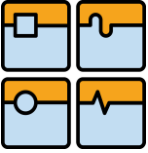
Rovinnosť sa skúša napríklad u upínacích plôch frézok. Používa sa k tomu primeraných dosiek (prípadne pravítok a vodováh). Pre náročnejších meraní sa používajú mosty a dosky s citlivým úchylkomermi.

3.2.3 Rovnobežnosť

Čiara je považovaná za rovnobežnú s rovinou, pokiaľ pri meraní vzdialenosti tejto čiary od reprezentatívnej priamky priesečnice tejto roviny s rovinou na tuto rovinu kolmou a prechádzajúcou niekoľkými bodmi danej čiary neprekročil najväčší rozdiel zistených hodnôt v danom rozsahu meraní predom určenú hodnotu.

Meranie zahrňuje :

- Rovnobežnosť čiar a rovín
- Rovnobežnosť pohybu
- Zhodnosť vzdialeností
- Zhodnosť os, súosovosť

	Ústav výrobných strojů, systémů a robotiky	Str. 21
	DIPLOMOVÁ PRÁCE	

Rovnoběžnosť vodiacich plôch sa skúša napr. priečne kladenými vodovákami (mikrometrickými odpichy). Rovnoběžnosť plôch s osami alebo os navzájom sa kontroluje pomocou meracích trŕňov.

3.2.4 Kolmost' priamok a rovín

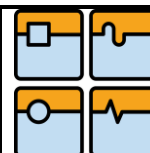
Dve roviny, dve priamky alebo priamka a rovina sú považované za kolmé, ak úchylka rovnobežnosti so vzorovým uholníkom neprekročí predpísanú hodnotu. Vzorový uholník môže byť metrologicky uholník alebo rámová vodováha, alebo sa môže skladať z kinematických rovín alebo čiar.

Kolmost' dvoch plôch sa obvykle kontroluje meracími valcami d číselníkovými úchylkomermi alebo meracími hranolmi. Optickou cestou môžeme merať kolmost' napríklad laserovým interferometrom.

3.2.5 Otáčanie

Meranie sa týka :

- Obvodového hádzania
- Periodického axiálneho pohybu
- Čelné hádzanie



4 Chyby a neistoty merania

Meraním rozumieme súbor činností, ktorých je cieľom stanoviť hodnotu danej veličiny. Do merania zahrňame experimentálne postupy a rôzne výpočty.

Hodnotu meranej veličiny získanej pri meraní nemôžeme určiť s absolútnou presnosťou. Nameraná hodnota je buď menšia alebo väčšia, ako skutočná hodnota - hovoríme o chybe merania.

4.1 Chyby merania

Podľa spôsobu výskytu rozdeľujeme chyby [5] na :

- Systematické (Sústavné)chyby - sú zapríčinené nepresnosťou meracieho prístroja a použitej meracej metódy. Ich veľkosť je zvyčajne známa. Preto sa dajú účinne eliminovať korekciou výsledku.
- Náhodné chyby - vznikajú z náhodných príčin, občas - nepravidelne. Sú zvyčajne zapríčinené vonkajšími vplyvmi, napr. vplyvom zmeny teploty a pod.
- Hrubé chyby - vznikajú nepozornosťou - nesprávnym odčítaním z prístroja, zlým zápisom do tabuľky hodnôt a pod

Podľa spôsobu vyjadrenia sa chyby rozdeľujú na:

- Absolútne chyby
- Relatívne chyby

Absolútna chyba merania – je algebrický rozdiel medzi výsledkom merania a konvenčne pravou hodnotou meranej veličiny. Má rovnakú jednotku ako meraná veličina, môže mať aj zápornú hodnotu.

Relativná chyba merania - je podiel absolútnej chyby merania a konvenčne pravej hodnoty meranej veličiny. Je to len číslo resp. hodnota v percentách, má kladnú hodnotu.

4.2 Neistoty merania

Neistotu merania [6] definujeme ako parameter, ktorý súvisí bezprostredne s výsledkom merania a ktorý charakterizuje rozpätie hodnôt v ktorých sa nachádza výsledná nameraná hodnota voči hodnote pravej.

Zdroje neistôt :

- neúplná definícia meranej veličiny
- nedokonalá realizácia definície meranej veličiny
- nevhodný výber prístroja
- nevhodný postup pri meraní
- nedostatočne známe účinky podmienok prostredia (ich nedokonalé meranie),
- zmeny pri opakovaných meraniach meranej veličiny pri očividne rovnakých podmienkach.

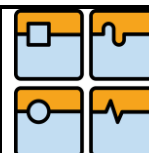
4.2.1 Neistota typu A

Neistota typu A sa určuje štatistickým spracovaním nameraných hodnôt. Daná neistota je reprezentovaná výberovým rozptylom opakovane meranej hodnoty. (aspoň 10x)

$$\bar{x} = \frac{1}{n} \sum_{j=1}^n x_j \quad (1.0)$$

$$u_a(x) = s(x) = \sqrt{\frac{\sum_{j=1}^n (x_j - \bar{x})^2}{n(n-1)}} \quad (1.1)$$

n – počet meraní. Ak je n menšie ako 10, treba vziať do úvahy spoľahlivosť daného vyhodnotenia štandardnej neistoty typu A. Prípadne smerodajnú odchýlku vynásobiť koeficientom k (pre $n > 10$ je $k = 1$, vid' tabuľka), tiež označovaný ako bezpečnostný faktor [7].



<i>n</i>	2	3	4	5	6	7	8	9	10 a viac
<i>k</i>	7,0	2,3	1,7	1,4	1,3	1,3	1,2	1,2	1

Tabuľka 1: Hodnoty bezpečnostných faktorov [7]

4.2.2 Neistota typu B

Neistoty typu B sú také, ktoré sa musia zisťovať inými než štatistickými metódami z opakovaných meraní. Ich veľkosť možno odhadnúť na základe doplňujúcich informácií (o presnosti prístroja) od výrobcu. Ak tieto informácie nie sú k dispozícii, môžeme ich získať porovnaním údajov prístroja s iným, presnejším prístrojom.

$$u_B(x_i) = \frac{\pm\Delta}{k} \quad (1.3)$$

$\pm\Delta$ – maximálna dovolená odchýlka daného prístroja

k- koeficient rozloženia (normálne rozloženie $\chi=2$)

Pri viacerých zdrojoch chýb neistoty typu B je výsledná neistota :

$$u_B(x) = \sqrt{\sum_{i=0}^m u_{Bi}^2(x)} \quad (1.4)$$

Neistoty typu B sa dajú vypočítať [7]:

1. Z už známej rozšírenej neistoty prepočtom na neistotu kombinovanú
2. Odhadom z variability zdroja a štatistického rozdelenia
3. Zo známych údajov o kombinovanej neistote (certifikáty, tabuľky)

Typy rozložení :

- Normálne (Gausovo)
- Rovnomerné (Pravouhlé)
- Trojuholníkové (Simpsonovo)
- bimodálne (Diracovo)
- lichobežníkové
- kvadratické
- kosínové a ďalšie.

4.2.3 Štandardná kombinovaná neistota

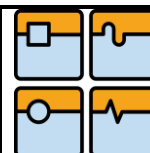
Málokedy si vystačíme v praxi s jedným alebo druhým typom neistoty. Je potrebné stanoviť výslednú neistotu z predchádzajúcich typov. Výslednú neistotu (kombinovanú neistotu) vypočítame ako odmocninu zo súčtu kvadrátov neistôt typu A a B.

$$u_c(x) = \sqrt{u_A^2 + u_B^2} \quad (1.5)$$

4.2.4 Rozšírená kombinovaná neistota

Aby bolo dosiahnuté lepšie pokrytie [8], ktorého pravdepodobnosť sa bude blížiť k 100 % (ale v skutočnosti ho nikdy nedosiahneme), je nutné štandardnú neistotu rozšíriť koeficientom rozšírenia k_r .

$$U(x) = k_r u_c(x) \quad (1.6)$$



5 Trackovacie interferometry

V testovaní geometrický presností sa čím ďalej presadzujú zariadenia na princípe využitia laserového signálu (optického signálu). Medzi tieto zariadenia patria laserový interferometer, Laser Tracker, Laser Tracer a pod.

Laser Tracker je zariadenie ktoré slúži na presné meranie rôznorodých objektov na veľké vzdialenosti (rádovo metre) za pomoci optického signálu. Ich presnosť sa začína v jednotkách μm .

Merajú sa dva uhly(θ, ϕ) a vzdialenosť(d). Tracker pošle laserový lúč na reflexný cieľ ktorý je súčasťou meraného objektu. Svetlo odrazené od cieľa je spätne vystopované. Reflexná cieľ sa líšia, ale najobľúbenejší je sféricky namontovaný odrazka (SMR). Ako svetlo znovu vstupuje do trackeru, ktorým je meraná vzdialenosť od trackeru do SMR.

Prepočet na súradnice :

$$X = d * \cos \theta * \sin \phi \quad (1.7)$$

$$Y = d * \sin \theta * \sin \phi \quad (1.8)$$

$$Z = d * \cos \phi \quad (1.9)$$

5.1 IFM, ADM

Meranie vzdialenosti [9], dôležitá funkcia laserového sledovača, môže byť inkrementálna alebo absolútna. Meranie prírastkovej vzdialenosti sa vykonáva pomocou interferometra (IFM) a frekvenčne stabilizovaného, hélium-neónového laseru. Laserové svetlo sa rozdeľuje do dvoch lúčov. Jeden putuje priamo do interferometra. Druhý lúč cestuje z laserového trackera, odráža od sféricky pripojeného reflektora (SMR) a na spätočnej ceste, prechádza do interferometra. Vnútri interferometra sa tieto dva lúče svetla zmiešavajú(interferujú), čo má za následok cyklické zmeny zakaždým keď sa SMR presunie bližšie alebo ďalej od laserovej sledovača o vzdialenosť rovnajúcu sa jednej štvrtine vlnovej dĺžky svetla ($\sim 0,0158$ mikróv). Elektronika počíta cyklické zmeny (známe ako "okrajové počty") pre určenie prejdenej vzdialenosti.

ADM (Absolute distance measurement) - absolútne meranie vzdialenosti. V posledných niekoľkých rokoch však ADM systémy prešli dramatickým zlepšením, ktoré ponúkajú presnosť porovnateľnú s interferometrami. Výhodou meranie ADM v priebehu merania kumulatívnej vzdialenosti je schopnosť jednoducho nasmerovať lúč na cieľ a

„vystreliť“. Infračervené svetlo z polovodičového lasera sa odráža od SMR a znovu vstupuje do trackeru, kde je premenený na elektrický signál. Elektronický obvod analyzuje signál na určenie jeho doby letu, násobením hodnoty s hodnotou rýchlosti svetla vo vzduchu pre určenie vzdialenosti lasera tracker a SMR.

V súčasnej dobe absolútnu diaľkomery s rozvojom elektroniky, dostatočne výkonné a rýchle, aby umožnili vysokorýchlostné skenovanie bez zanedbateľné strát presnosti.

5.2 Leica Absolute Tracker AT901

Jedná sa Laser Tracker pre ktorý bude vyvíjaná aplikácia. Je vybavený funkciou PowerLock. Automaticky sa uzamkne na akýkoľvek pohyblivý cieľ bez zásahu užívateľa. Rozsah merania je až do 160m. Svojím vybavením a funkčnosťou je predurčený na merania pre obrábacie stroje.

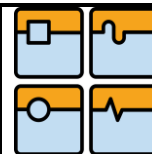
Rozsah : 80 m

Presnosť 3D (AIDM): $\pm(15 \mu\text{m} + 6 \mu\text{m}/\text{m})$

Presnosť (IFM): $\pm 0.5 \mu\text{m}/\text{m}$



Obrázok 1: Laser Tracker AT901



Celá meracia stanica sa skladá z viacerých častí :

- Laser Tracker
- Kabeláž
- Napájací AT kontroler



Obrázok 2: Pracovisko s Laser Trackerom

Spomenutý Laser Tracker nie je však jediným typom na trhu. Firma Leica poskytuje Laser Tracker AT401. Následne sú na trhu aj ďalšie firmy napr. API, FARO. Ich produkty budú stručne spomenuté v nasledujúcich podkapitolách.



5.3 Leica Absolute Tracker AT402

Jedná sa o ďalší produkt od firmy Leica. Je schopný byť napájaný vlastnou internou batériou a pracovať v náročných prostrediach.

Má zvýšenú odolnosť a dlhšiu životnosť batérie. AT402 má unikátny "All in One" návrh systému, ktorý zahŕňa tiež potrebné príslušenstvo, ako úroveň gravitácie, monitorovanie životného prostredia a integrovaný IR pre diaľkové ovládanie.



Obrázok 3 : Laser Tracker AT401

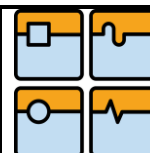
5.4 Leica Absolute Tracker AT960

Najnovšia generácia prenosných CMM systémov založeného na osvedčenej technológii, AT960 spája funkčnosť 6DOF na Leica Absolute Tracker AT901 s T-Cam a použiteľnosťou kompaktného riešenia Leica Absolute Tracker AT402. Váži menej ako 14 kg.

AT960 laser tracker má možnosť merania až do 120 m. Kompatibilný s reflektormi ako je Leica T-Probe, Leica T-Scan a Leica T-Mac, ponúka mnoho rôznych kontrolných aplikácií.



Obrázok 4: Leica AT960



5.5 Tracker 3

Jedná sa o zariadenie od firmy API [10]. Vyhotovuje sa v troch variantoch. Záleží na tom do akej vzdialenosti chceme merať. (30,80,120 m) Vďaka pevnej a kompaktnej konštrukcii, tento laser tracker patrí medzi najmenšie, presné IFM / ADM laser tracker k dispozícii na trhu.

Rozlíšenie(IFM) : 0.08 μm

Rozlíšenie(ADM) : 0.1 μm

Uhlové rozlíšenie : $\pm 0.018^\circ$

Vlastnosti :

- Jednoduchá prevádzka
- Rýchle jednobodové meranie
- Kontinuálne meranie bodu



Obrázok 5 : Tracker 3

5.6 RADIANT

Ďalší z produktov od firmy Automated Precision Inc [11]. (API)

Lineárny rozsah : 100 m

Uhlové rozlíšenie : $\pm 0.018^\circ$

Rozlíšenie(IFM) : 0.08 μm

Rozlíšenie(ADM) : 0.1 μm

Vlastnosti :

- Jednoduchá prevádzka
- Pokročilé príslušenstvo
- Všestranná montáž
- Viacnásobná SMR identifikácia



Obrázok 6 : Radian



5.7 Laser TRACER-NG

Nejedná sa síce o zariadenie Laser Tracker, ale laser tracer, ktorý je však veľmi rozšírený. Jedná sa o produkt od firmy Etalon [12], vyvinutý pre kalibráciu, kontrolu a zlepšenie presnosti meracích strojov a obrábacích strojov. Jedná sa o sledovanie laserový interferometrom, ktorý automaticky sleduje reflektor, a teda umožňuje identifikáciu geometrických odchýlok s najvyššou presnosťou.

Na rozdiel od bežných meracích prístrojov, LaserTracer ponúka bezprecedentnú presnosť stredú otáčania vďaka patentovanej meracej techniky: guľa s odchýlkou iba 50 nanometrov sa používa ako referenčný bod pre optický interferometer. V dôsledku toho, mechanické chyby otáčania a otočnej osi sú plne kompenzované.

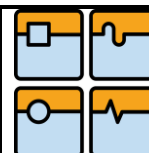
Laser TRACER však nedokáže merať len vzdialenosti, na priemerovanie obrobku alebo na stavbu stroju použiteľný nie je.

Technické špecifikácie :

Rozlíšenie :	0.001 μ m
Merací rozsah :	0,2-15m
Uhlový rozsah :	-225° - 225°



Obrázok 7: Laser Tracer [12]



Predchádzajúcej časti boli zmienené známe trackovacie zariadenia, ktoré sa využívajú pri meraní presností v obrábacích strojoch. Každé zariadenie má svoje výhody a nevýhody a voľba závisí na konkrétnej aplikácii. V nasledujúcej tabuľke je prehľad zmienených zariadení.

	AT901	Tracker 3	RADIAN	Laser Tracer	AT960
Rozsah [m]	160	120	100	15	120
Rozlíšenie(IFM) [μm]	±0.5	0.08	0.08	0.001	±0.5
Rozlíšenie(ADM) [μm]	±(15 + 6 μm/m)	0.1	0.1	-	±(15 + 6 μm/m)

Tabuľka 2: Porovnanie zariadení

6 Softvér pre meranie s Laser Trackerom

Na to aby sme mohli Laser Tracker využívať na merania potrebuje softvér, ktorý zvláda komunikáciu s Laser Tracker a zároveň zvládal vyhodnocovanie nameraných dát. Existuje veľmi obmedzené množstvo daných softvérov, konkrétne sa jedná o Polyworks, emScon a Trac-cal. Nevýhodou týchto softvérov je uzavretosť a finančná náročnosť. V nasledujúcich podkapitolách si jednotlivé softvérové možnosti.

6.1 emScon

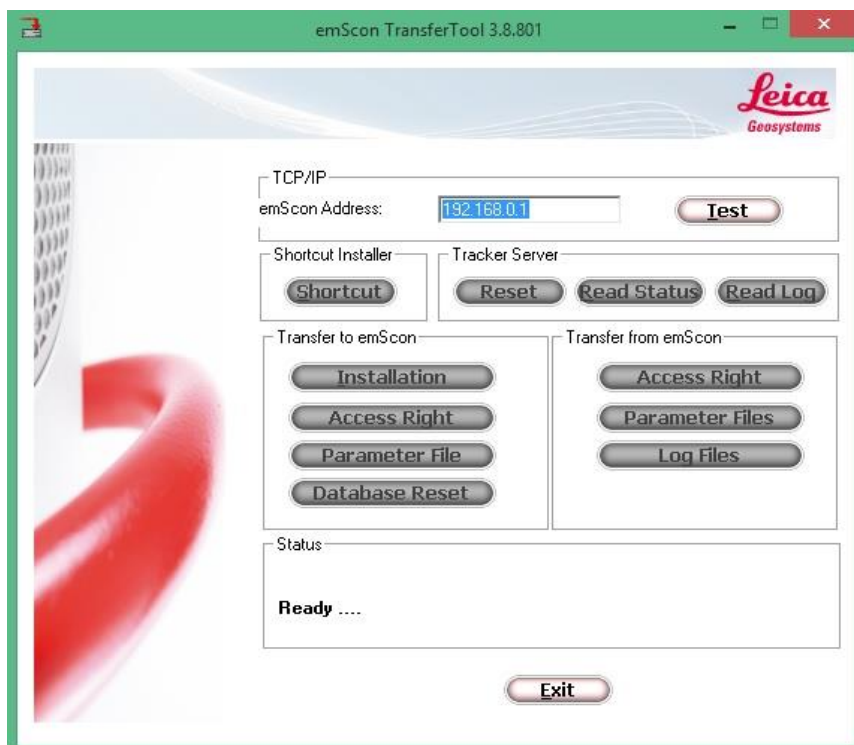
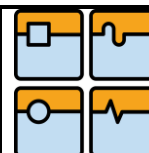
Je softvér ktorý dodáva spoločnosť Leica pre svoje trackeri. Tiež je označovaný ako Tracker Programming Interface (Programovateľné rozhranie pre tracker) . Je vhodný pre použitie s celou Leica Geosystems Laser Tracker rodiny. Od najstaršieho Leica LT300 a Leica LT (D) 500 serie na najnovšie absolútne tracker rodiny Leica AT 901-LR, Leica AT 901-MR a Leica AT 901-B.

S emScon môže tracker komunikovať súbežne s akýmkoľvek iným softvérom, na ľubovoľnej platforme, od Windows[®] na UNIX[®], Linux[®] a ďalšie. To vám umožní ovládať svoje trackeri pomocou štandardného softvéru, ako je Excel, Word, Access a Visual Basic alebo pomocou vlastného vyvinutého softvéru napísaného v C/C++ alebo C sharp.

emScon so svojim programovacím rozhraním, je vhodný pre použitie s celou Leica Geosystems Laser Tracker rodiny. Od najstaršieho Leica LT300 a Leica LT (D) 500 serie na najnovšie absolútny tracker rodiny Leica AT 901-LR, Leica AT 901-MR a Leica AT 901-B.

Tracker Programming Interface je dodávaný s plne otvoreným zdrojovým kódom, ktorý obsahuje niekoľko ukážkových aplikácií pre jednoduchú integráciu.

Cena : 0 Kč (dodávaný k zariadeniu)



Obrázok 8:emScon

6.2 InnovMetric PolyWorks

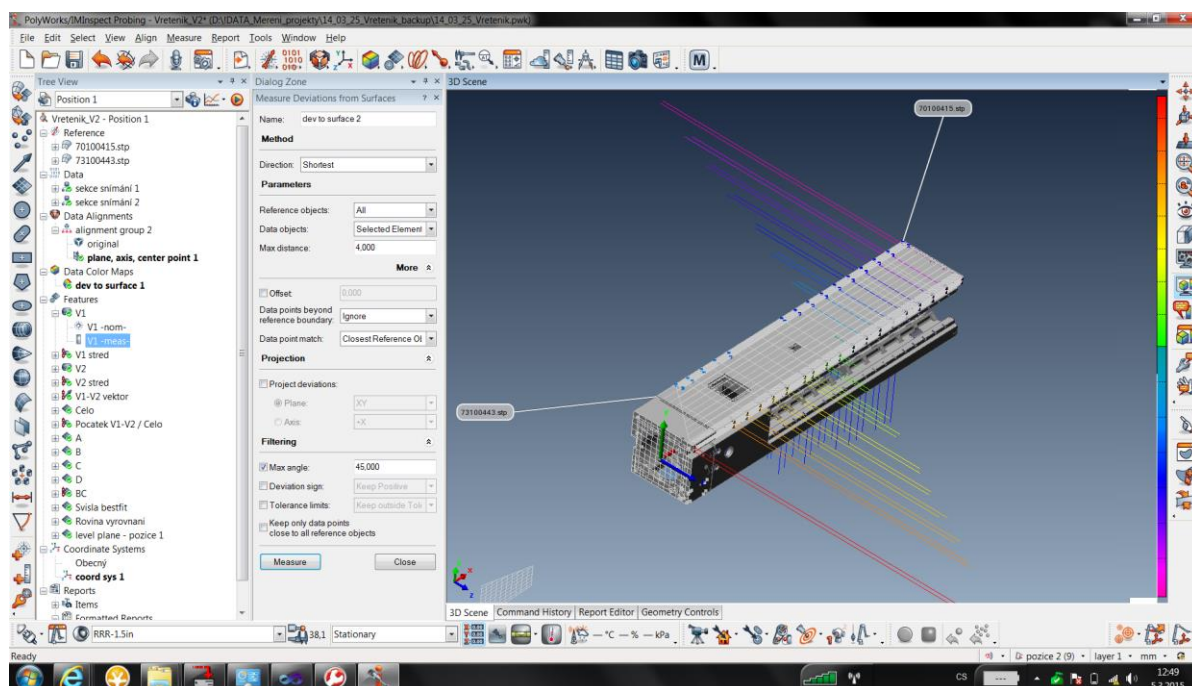
Jedná sa o metrologický softvér [13], ktorý na základe softvérovej architektúry dokáže efektívne spracovávať obrovské objemy dát z 3D skenerov v reálnom čase. Vďaka modulárnej architektúre softvér pokrýva veľké množstvo aplikácií – od jednoduchej inšpekcie pomocou dotykových meracích systémov, cez spracovanie skenovaných dát kamerou a ich porovnanie s CAD modelmi.

Obsahuje dva moduly :

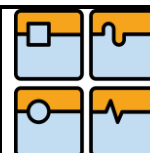
- PolyWorks/Inspector je plnohodnotný 3D softvér. Užívateľovi ponúka možnosť priamej komunikácie s meracími systémami od rôznych výrobcov a umožniť tak priame načítavanie dát určených pre spracovanie. Merania môžu byť vyhodnotené voči CAD modelu alebo môže byť porovnané s iným meraním.

- PolyWorks/Modeler je plnohodnotný 3D softvér pre kompletné riešenie reverzného inžinierstva. Naskenované dáta môžu byť upravované, filtrované, redukované a podobne. (pre dosiahnutie optimálneho výsledku) Automatickým procesom triangulácie, ktorý je plne parametricky kontrolovaný operátorom, ponúka softvér PolyWorks výrobu kvalitných polygonálnych modelov.

Cena : $\approx$ 500 000 až 1 000 000 Kč



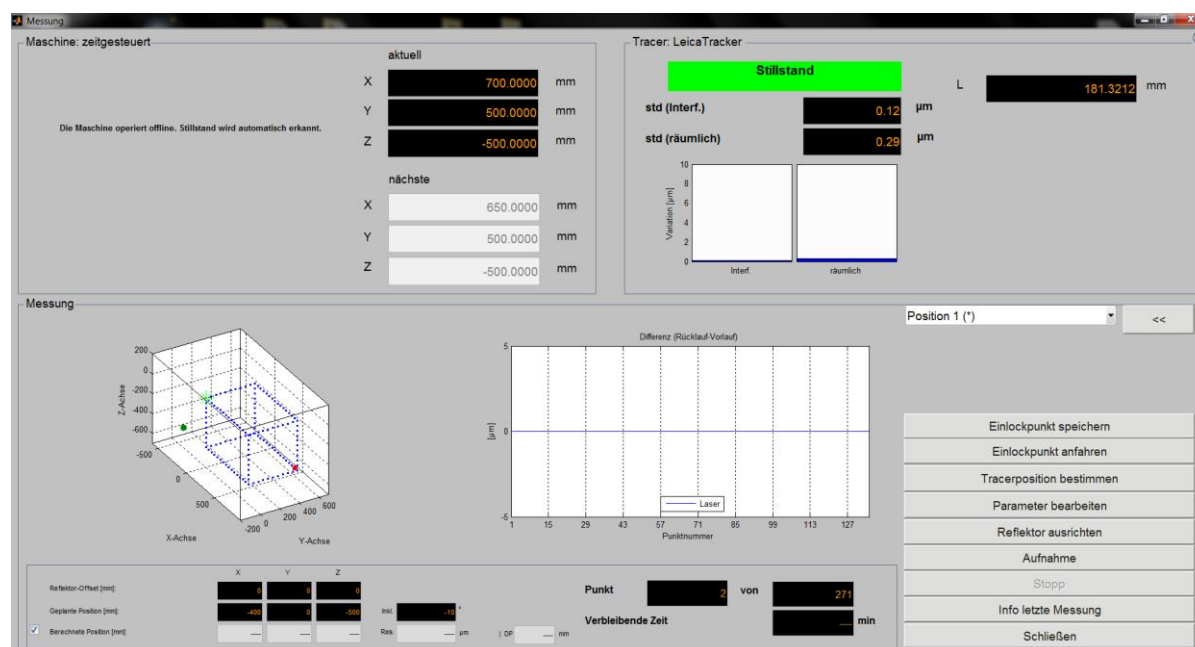
Obrázok 9: Polyworks



6.3 Trac-cal

Jedná sa o ďalší merací systém vytvorený firmou Etalon. Slúži k privedeniu kalibrácie okrem zadania korekcií priamo do riadenia kalibrovaného obrábacieho stroje. Po zadaní potrebných informácií software sám navrhne postup kalibrácie, vygeneruje kalibračnú mapu a zanesie korekcie do riadenia obrábacieho stroje. Primárne je určená pre Laser Tracer. Avšak je možnosť dokúpiť modul, ktorý podporuje aj merania s ďalšími zariadeniami ako je Laser Tracker.

Cena : $\approx 200\,000$ Kč (základná cena)



Obrázok 10: Rozhranie Trac-cal

7 Možnosti vývoja vlastnej aplikácie

Nasledujúca časť je zameraná na možnosti vývoja vlastnej aplikácie a možnosti použitia programovacích jazykov. V predchádzajúcej kapitole bol zmienení softvér emScon (emScon), ktorý slúži ako programovateľné rozhranie – emScon SDK (software development kit). Samotné SDK má v podstate 3 možnosti ako jeho prostredníctvom vytvárať vlastné aplikácie.

1. Prvou možnosťou je využitie hlavičkových súborov *ES_C_API_Def* a *Enum*, ktoré obsahujú patričné funkcie a premenné pre prácu s Laser Tracker. Na to aby sme ich mohli využívať, aplikácia musí byť vytváraná v jazyku C (daný jazyk bude popísaný nižšie). Avšak táto možnosť nesie zo sebou aj veľa nevýhod. Jednou z nich je náročnosť programovania, „obmedzené možnosti programovania“. Preto je pre nás táto možnosť nie je príliš vhodná.
2. Druhou možnosťou je využitie knihovne *ES_MCPP_API_Wrapper*. V podstate sa jedná o kompiláciu hlavičkových súbor *ES_C_API_Def* a *Enum*. Výhodou je, že danú knihovňu môžeme využiť pre programovanie vo vyšších jazykoch. Nevýhodou je, že všetky informácie pre komunikáciu s meracou jednotkou sú primané a odosielané cez TCP/IP ktorý je potrebné si vytvoriť.
3. Poslednou možnosťou je využitie knihovne *LTControl*. Ako v predchádzajúcej možnosti, tak aj v tomto prípade využíva vyššie programovacie jazyky. Knihovňa obsahuje metódy pre naviazania spojenia s jednotkou, metódy pre prácu pre synchronný a asynchronný režim. Aby sme ju mohli používať je potrebné ju zaregistrovať.(pomocou služby Regsvr32). Nevýhoda tohto riešenie je obmedzenosť len na systémoch Microsoft Windows (programovanie aplikácie).

Str. 38	Ústav výrobných strojů, systémů a robotiky	
	DIPLOMOVÁ PRÁCE	

Využitelnost knihovny *LTControl* je rozsiahla. Jej pomocou je možné tvoriť „aplikácie¹“ v softvéri Matlab, LabView alebo aj Microsoft Excel. Prvých dvoch zmienených softvérov je nevýhodou ich vysoká cena. Microsoft Excel je cenovo prijateľný ale nie je vhodný na vytváranie tak komplexnej aplikácie ako bude potrebné

Ďalšou možnosťou vývoja vlastných aplikácií je využitie softvéru Polyworks. Ten umožňuje vytvárať makrá vo svojom vlastnom prostredí a tiež možnosť použitia SDK. Nevýhodou spomínaného riešenia je nutnosť vlastniť daný softvér. Tým nespĺňame zadané požiadavky, keďže k aplikácii by bolo potrebné mať zakúpený softvér Polyworks (tým pádom by využitie aplikácie bolo obmedzené len systémy, ktorí ho majú) , ktorý je zároveň veľmi drahý(InnovMetric PolyWorks). Z týchto dôvodom sa preto daná možnosť nebudeme využívať.

Všetky zmienené možnosti majú svoje výhody a aj nevýhody. Vzhľadom k tomu, že naša aplikácia bude hlavne určená na meranie hodnôt a ďalších parametrov, súčasne ju budeme vytvárať vo Windows (aj samotná aplikácia je primárne určená na operačné systémy Microsoft Windows), tak najvhodnejšia varianta pre nás je využiť knihovňu *LTControl*. Programovacie štúdio bude použité Microsoft Visual Studio 2013 vzhľadom k jeho širokému spektru programovacích jazykov a možností(Microsoft Visual Studio).

¹ Nejedná sa o plnohodnotné aplikácie ako je bežne brané

8 Programovacie jazyky

V dnešnej dobe existuje množstvo jazykov. Každý má svoje silné aj slabé stránky. Celú túto škálu by sme mohli rozdeliť do dvoch kategórií² :

- Programovacie jazyky nižšej úrovne – jedná sa hlavne o programovacie jazyky pre mikrokontroléry. Medzi najznámejšie patrí ANSI C a Asembler.
- Programovacie jazyky vyššej úrovne – jedná sa o väčšinou novších jazykov, ktoré už majú v sebe zakomponované aj grafické knihovne, komunikačné knihovne a pod. Medzi tieto jazyky radíme napr. Visual Basic, C#.

Každý programovací jazyk má svoje prednosti a jeho voľba závisí na úlohe a znalostiach programátora. V dnešnej dobe medzi najpoužívanejšie patria C, C++, C#, Python, javascript. Javascript je najviac využívaný pri tvorbe webových stránok a jazyk Python je uprednostňovaný pri vývoji aplikácií pod systémom Linux. Následné tri jazyky budú v ďalších podkapitolách bližšie popísané.

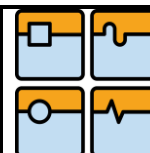
8.1 Jazyk C

Je univerzálny programovací jazyk nízkej úrovne (low level language) [14]. Má veľmi úsporné vyjadrovanie a je štruktúrovaný. Obsahuje veľký súbor operátorov a moderné dátové štruktúry. Jeho veľkou výhodou je väčšia efektívnosť a rýchlosť oproti iným jazykom. Jedná sa o veľmi populárny jazyk, hlavne pre užívateľov Linuxu a Unixu. Samotný operačný systém Unix bol celý napísaný v jazyku C.

8.2 Jazyk C++

Jedná sa viacparadigmaticý [15] programovací jazyk vyššej úrovne, ktorý umožňuje pracovať aj s prostriedkami nízkej úrovne. Vychádza z jazyka C (spätná kompatibilita), ktorý je obohatený prácou s objektami. Prináša nové vlastnosti, vrátane deklarácií, pretypovanie so syntaxou funkcií, new/delete (alokovanie a dealokovanie pamäte), typ bool (v C nie je), referencie, inline funkcie, implicitné argumenty, preťaženie funkcií, priestory mien, triedy

² Nejedná sa však o jediné rozdelenie



(class), interface a pod. Ide o jeden z najpoužívanějších a najefektívnejších programovacích jazykov.

8.3 Visual Basic

Patrí medzi programovacie jazyky vyššej úrovne, objektovo orientované a udalosťami riadené(event - udalosťami) [16]. V posledných rokoch sa stal hlavným programovacím prostriedkom na platforme Microsoft. Rôznorodé varianty tohto jazyka sa využívajú k programovaniu „klasických“ a sieťových aplikácií, ale aj na tvorbu makier v balíku Microsoft Office (Visual Basic for Application). Tiež je využívaný pri programovaní internetových alebo mobilných aplikácií(Android, Windows 8). Je hlavne preferovaný kvôli svojej jednoduchšej syntaxi, veľkej variabilite a flexibilita .

8.4 C# (C sharp)

Jedná sa o jazyk vyššej úrovne, ktorý je objektovo-orientovaný programovací jazyk vyvinutý spoločnosťou Microsoft ako časť ich iniciatívy .NET [17]. Prvá verzia uzrela svetlo v roku 2002. Od tej doby prešla mnohými zmenami až k verzii C# 5.0 .(už je prichystaná aj verzia C# 6.0, ktorá by mala byť predstavená v tomto roku) Je vhodný na programovanie webových stránok (s využitím ASP.NET), sieťových aplikácií ako na strane serveru tak aj klienta a samozrejme bežných aplikácií. S novými verziami Microsoft Visual štúdia (od 2013) je možné vytvárať aplikácie aj pre Android, iOS a Windows 8.1 RT(a tiež Windows Phone). Samotný jazyk je rozšírený aj na Linuxové platformy(Mono project). Samotná syntax jazyk vychádza z jazyka C/C++.

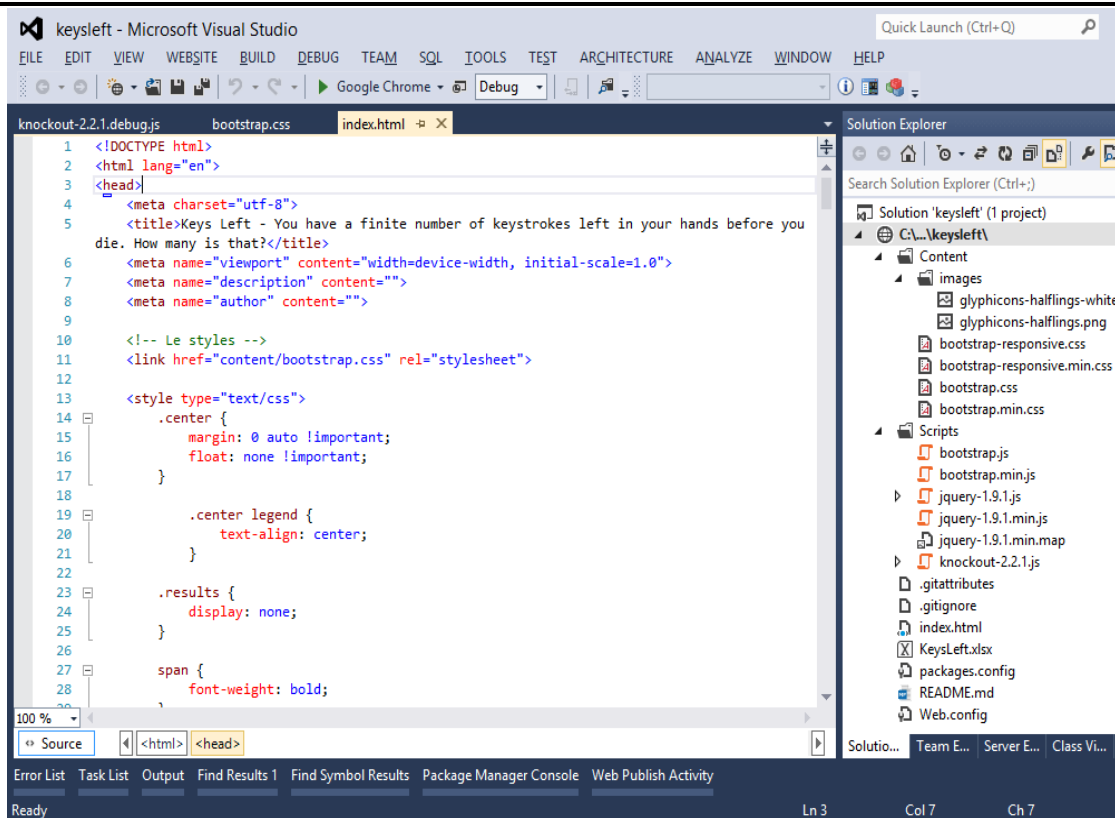
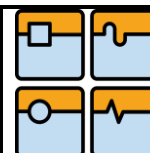
Vzhľadom k zmieneným výhodám a nevýhodám daných jazykov, bude aplikácia vytváraná v jazyku C#. Hlavný dôvod výberu spočíva v jeho rozšírenosti podpory a použiteľnosti. Samotné štúdium pre vývoj aplikácie bude použité Visual Studio 2013 .

8.5 Microsoft Visual Studio

Microsoft Visual Studio [18] je integrované vývojové prostredie (IDE) od spoločnosti Microsoft. Používa sa k vývoju počítačových programov pre systém Microsoft Windows, rovnako ako webové stránky, webové aplikácie a webové služby. Visual Studio využíva softvérové produkty Microsoft vývojovej platformy, ako je Windows API, Windows Forms, Windows Presentation Foundation, Windows Store a Microsoft Silverlight.

Visual Studio obsahuje editor kódu podporujúci IntelliSense (dokončenie komponentov), rovnako ako refaktorovanie. Integrovaný debugger funguje aj ako source-level debugger a machine-level debugger. Ďalšie vstavané nástroje sú Forms Designer pre stavebné GUI aplikácie, web designer, trieda dizajnéra a návrhára schém databáz. Akceptovanie pluginov, ktoré zvyšujú funkčnosť takmer na každej úrovni, vrátane pridania podpory pre zdroje, riadiace systémy (ako Subversion) a pridáva nové sady nástrojov, ako sú editory a vizuálneho návrhára pre jazyky konkrétnou doménou alebo sady nástrojov pre iné aspekty životného cyklu vývoja softvéru (rovnako ako klient Team Foundation Server: Team Explorer).

Podporuje rôzne programovacie jazyky a umožňuje editovanie kódu a debugger na podporu (v rôznej miere) takmer akýkoľvek programovací jazykov. Vstavaný jazyky obsahujú C, C++ a C++ / CLI (cez Visual C++), VB.NET (via Visual Basic .NET), C# (cez Visual C#), a F# (Visual Studio 2010). Podpora pre iné jazyky, ako je M, Python, Ruby a okrem iného je k dispozícii prostredníctvom jazykových služieb inštalovaných samostatne. To tiež podporuje XML / XSLT, HTML / XHTML, JavaScript a CSS.



Obrázok 11: Rozhranie Visual Studia

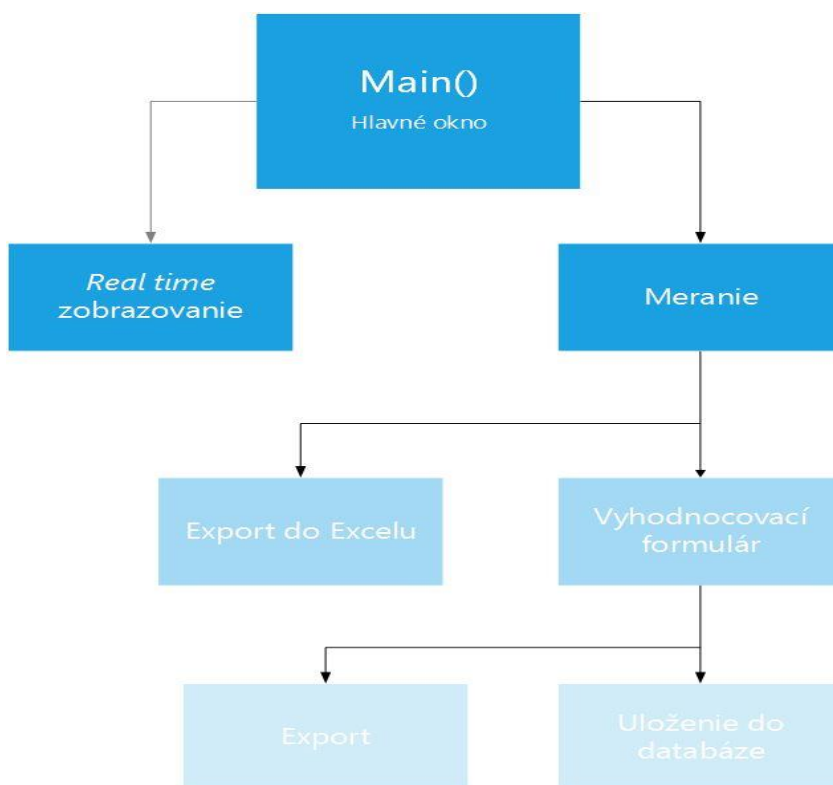
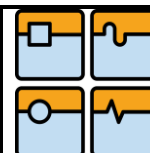
9 Návrh aplikácie

V predchádzajúcej kapitole boli rozobraté jednotlivé výhody a nevýhody rôznych programovacích jazykov, ktoré by bolo možné využiť. Po prejdení jednotlivých možností bolo rozhodnuté pre programovací jazyk C # s využitím knihovne *LTControl.dll*. Ako vývojové štúdio pre naše potreby bude použité Microsoft Visual Studio 2013.

Pred samotným programovaním bude predbežne navrhnutá architektúra aplikácie z funkčného hľadiska. Pri návrhu je braný ohľad na požiadavky, ktoré boli stanovené a budú priblížené.

Požiadavky na aplikáciu :

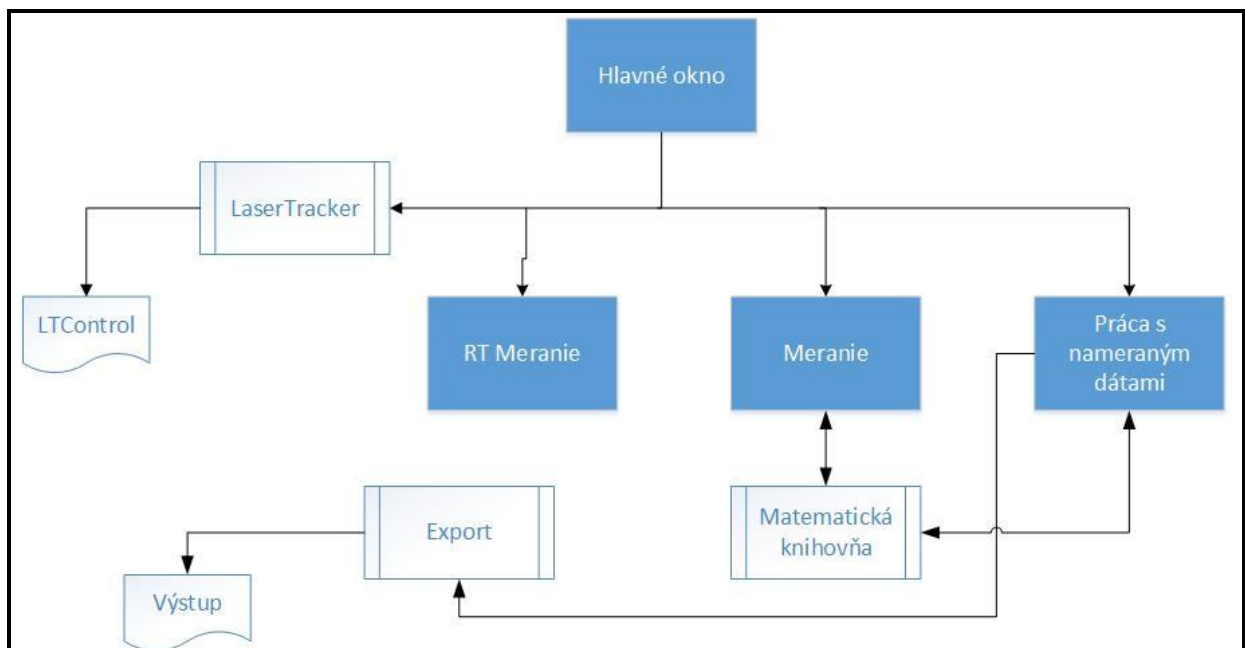
- 1) Priame zobrazovanie hodnôt – Na základe požiadavku bude potrebné vytvorenie samostatnej funkcie, ktorá bude zabezpečovať spustenie meranie a vrátenia nameraných hodnôt. Musí byť prispôsobená na cyklické volanie aby bolo dosiahnuté stáleho zobrazovania hodnôt.
- 2) Meranie dĺžok – samotné zariadenie obsahuje možnosť nastavenia niekoľkých súradných systémov. Meranie dĺžok dosiahneme využitým sférického sur. Systému. Preto bude potrebné aby aplikácia obsahovala možnosť zmeny súradného systému.
- 3) Export dát z aplikácie - Dáta budú musieť byť uložené vo vhodnej podobe, aby bol možný export. Základný export bude do excelu (*.xls), kde je možnosť ďalšieho spracovania dát, prípadne ďalší export napr. do Matlabu.
- 4) Možnosť ďalšieho rozšírenia – Aplikácie musí byť napísaná tak, aby bolo možné sa v nej následne vyznať a rozšíriť o ďalšie funkcie.
- 5) Čo najširšia možnosť používania (vzhľadom k OS) – zabezpečené použitým Microsoft Visual Studio 2013
- 6) Nízka cena



Obrázok 12: Blokový diagram návrhu

Pri prvotnom návrhu pomyselné rozdeľujeme aplikáciu na dve časti. Jedna časť bude slúžiť na Real-time zobrazovanie hodnôt s možnosťou ukladania nameraných hodnôt. Druhá časť bude slúžiť na klasické meranie, napr. dĺžok, rovinnosti. Ďalšou požiadavkou je možnosť namerané dáta ďalej používať aj mimo danej aplikácie, preto bude doplnená o možnosť exportu do excelu.

Následne navrhujeme funkčne-objektový diagram, ktorý bude reprezentovať beh a funkciu programu. Bude potrebné triedu na zastrešenie práce s knihovňou *LTControl*. Vytvoríme ďalšie triedy ktoré budú slúžiť na zastrešovanie Matematických operácii a exportu do Excelu. Grafické znázornenie diagramu je na obrázku nižšie.



Obrázok 13: Funkčný diagram návrhu aplikácie

Hlavné okno – súbor jednotlivých grafických rozhraní aplikácie

Výstup – výstupný súbor (excel, prípadne txt)

Matematická knihovňa – súbor všetkých matematických funkcií , ktoré budú potrebné v našej aplikácii

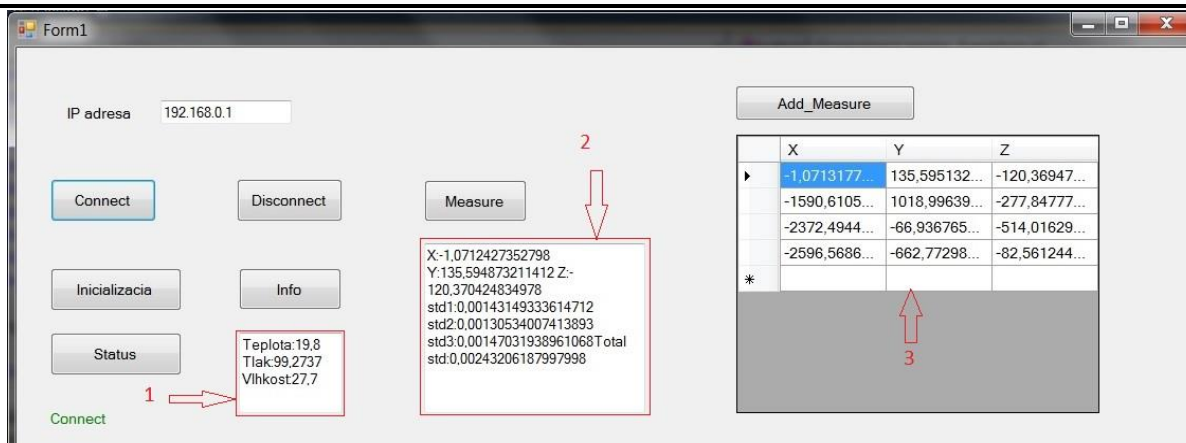
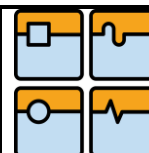
LaserTracker – jedná sa triedy (objekt), ktorý obsahuje metódy na obsluhovanie hardvéru

Real-time meranie (RT meranie) – modul, ktorý nám bude priamo ukazovať hodnoty v priestore (zobrazované hodnoty sa budú odvíjať od súradnicového systému)

Meranie – modul pre „klasické“ meranie (bude vysvetlené neskôr)

9.1 Testovacia aplikácia pre zisťovanie možností SDK

Pre prvotné samotné testovanie bude vytvorená demonštračná aplikácia. Bude slúžiť len na testovanie metód a komunikácie s Laser Trackerom. Rozhranie danej aplikácie je vidieť na obrázku nižšie.



Obrázok 14: Rozhranie testovacej aplikácie

Ako je vidieť aplikácia obsahuje tlačidlá pre pripojenie a odpojenie, inicializácie, získanie statusu a informácii o jednotke a ďalšie. Na obrázku je vidieť jednotlivé okná s výpismi. Prvé okno (označené 1) obsahuje informácie o prostredí, druhé okno obsahuje informácie o zmeranej pozícii s odchýlkami (*std*). Ako je vidieť, vrátená hodnota je príliš dlhá (2) a bude potrebné ju skrátiť. To isté platí pre tabuľku nameraných hodnôt (3). Tiež by bolo vhodné doplniť stĺpec, ktorý bude poukazovať na postupnosť nameraných bodov, aby sa pri väčšom množstve dát nestala neprehľadná.

Na základe predchádzajúcich poznatkov rozvrhneme následnú aplikáciu. Celý vývoj si rozdelíme do následných častí :

1. Návrh objektového modelu aplikácie
2. Návrh a vytvorenie triedy *LaserTracker*
3. Naprogramovanie modulu *Real-time meranie*
4. Naprogramovanie modulu *Meranie*
5. Naprogramovanie modulu *Spracovávanie dát*
6. Návrh designu a naprogramovanie *Hlavného okna*
7. *Doplňovanie a rozširovanie aplikácie*
8. Kompletizácia, testovanie aplikácie

9.2 Postup merania

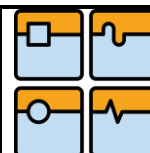
Pred samotným vytváraním aplikácie je potrebné si stanoviť jednotlivé postupy merania. Na základe postupu merania budú vytvorené jednotlivé funkcie a možnosti nastavenia celého merania.

Proces merania :

1. Spustenie trackeru, aplikácie
2. Voľba typu merania
3. Nastavenie súradného systému → *SetCoordinateSystemType()*
4. Nastavenie stacionárneho merania → *SetMeasurementMode()*
5. Nastavenie jednotlivých parametrov merania → *SetStationaryModeParam()*
6. Umiestnite reflektor do meranej pozície (bodu)
7. Zmeranie pozície → *StaticMeasurement()*
8. Uloženie nameranej hodnoty
9. Opakovanie bodov 6,7,8
10. Vyhodnotenie meranie

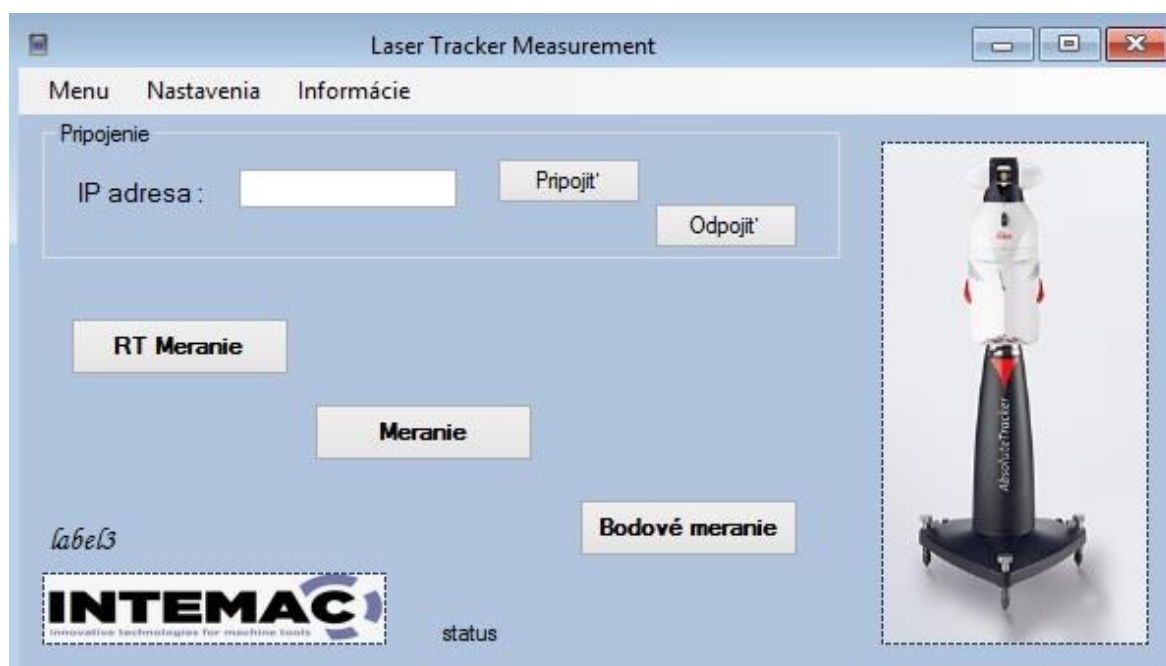
9.3 Návrh objektového modelu

Na základe požiadavkou a navrhnutého postupu merania sa bude objektový návrh odvíjať od prvotného návrhu v predchádzajúcej časti. Súčasťou aplikácie budú 3 vzájomne prepojené „moduly“ (RT meranie, Meranie, Bodové meranie). Pre každý typ meranie je potrebné nastavovať zadané parametre – to bude zabezpečovať trieda *Settings*. Na obrázku nižšie (Obrázok 16: Objektový model aplikácie Obrázok 14: Rozhranie testovacej aplikácie) je vidieť kompletný objektový model. Čierna čiara predstavuje prepojenie, červená dátový prístup (posielanie príkazov) a zelená namerané hodnoty (dáta). V ďalších častiach bude postupne rozobraný jednotlivý návrh a vývoj časti aplikácie. Ako prvá časť na ktorá bude rozobraný návrh hlavného okna a objektovej triedy pre Laser Tracker.

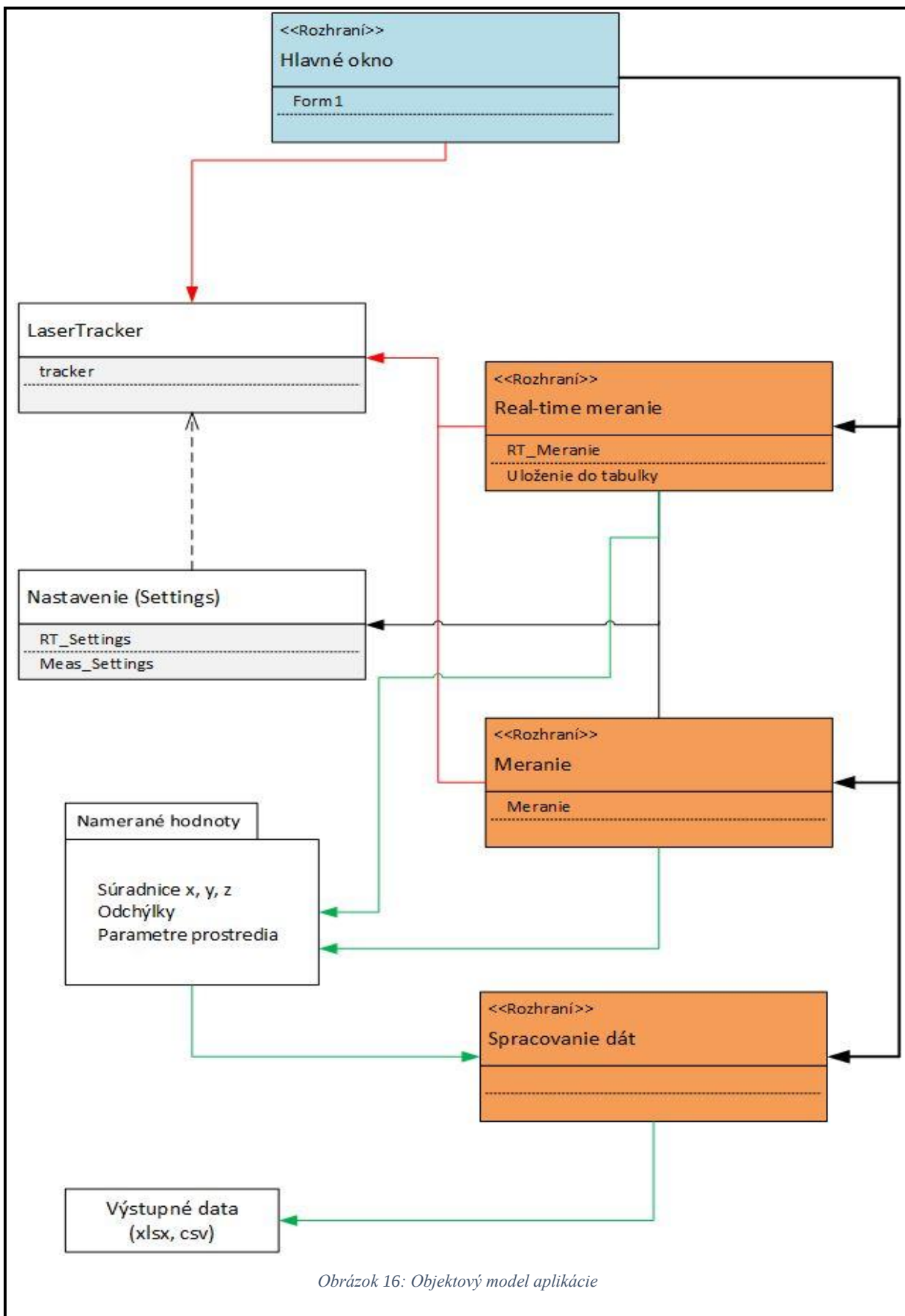


9.4 Hlavné okno

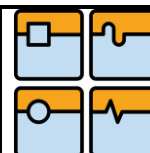
Hlavné okno, tiež označované ako *Main form* alebo *Form1*, je základom celej aplikácie. Vytvára prvotný dojem a slúži ako prepojenie k ďalším funkčným častiam. Musí preto obsahovať jednotlivé prepojenie, jednoduchú orientáciu a tiež je vhodné aby aplikácia obsahoval stručný návod pre pomoc s aplikáciou. Kvôli jednoduchej orientácii je hlavné okno pomyselné rozdelené. Prvá časť je blok *Pripojenie*, ktorý obsahuje IP adresu Laser Trackeru a tlačítka pripojiť a odpojiť. Až po následnom úspešnom pripojení sa užívateľovi objavia ďalšie tri tlačidlá, ktoré slúžia na prepnutie do jednotlivých módov.



Obrázok 15: Hlavné okno



Obrázok 16: Objektový model aplikácie



Ďalej hlavné okno obsahuje aj vrchnú lištu so základnou ponukou. Medzi základnú ponuku sú vložené možnosti Inicializácie (Menu), nastavenia a tiež manuál k aplikácii (informácie). Pre oživenie sú vložené obrázky Laser Trackeru a firmy Intemac. Štítok *status* vypisuje informácie o stave Laser Trackeru (Pripojený/Odpojený/Pripravený).

Po pripojení k Laser Trackeru, sa zobrazia tlačítka pre voľbu modulu merania. Po kliknutí sa načíta okno zvoleného modulu. Hlavné okno zostáva stále aktívne v pozadí.

9.5 Trieda LaserTracker

Trieda je vytvorená tak, aby k nej mohli pristupovať všetky moduly a zároveň je trieda stále pripojená k Laser Trackeru (hardvéru).

Metódy triedy :

- *public void Connect(string ipaddress)* – slúži pre pripojenie k zariadeniu
- *public void Disconnect()* – slúži pre odpojenie zariadenia
- *public void Initialize()* – inicializácia Laser Trackeru
- *public string Information()* – vracia informácie o zariadení
- *public string StatusTracker()* – vracia status Laser Trackeru
- *public string ParamProstredia()* – vracia hodnoty prostredia (teplota, tlak, vlhkosť)
- *public string MeranieBoduSta()* – zmeranie bodu v priestore (Stacionárne)
- *public string MeranieBoduSta2()* – zmeranie bodu v priestore (Stacionárne)
- *public void NastavenieADM(double targetStabilityTolerance, int retryTimeFrame, int NumberOfRetry)* – nastavenie ADM merania
- *public void ContinuesMeasurement(int index)* – spustenie kontinuálneho merania
- *public void SetMeasurement()* – nastavenie merania (mód merania, súradnicový systém, čas priemerovania a pod. – viď nižšie)

Kód triedy LaserTracker s vybranými metódami a popisom je v prílohe. (Príloha A : Trieda Laser Tracker).

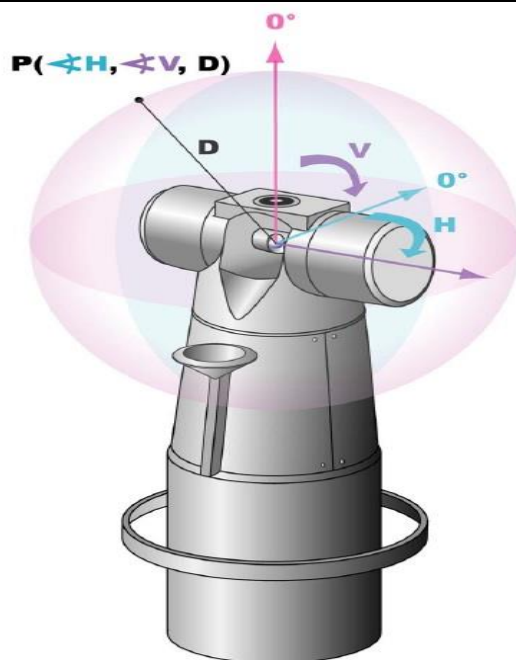
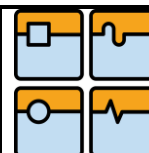
9.5.1 Nastavenia Laser Trackeru

V predchádzajúcej časti bolo spomenuté možné nastavenia Laser Trackeru, ktoré má na starosti metóda *SetMeasurement*. V tejto časti budú zmienené možnosti nastavenia súradnicového systému a meracie módy.

Súradnicové systémy

1. Right-Handed Rectangular – pravotočivý (pravoruký) kartézsky systém
2. Left-Handed Rectangular - ľavotočivý (ľavoruký) kartézsky systém. Dosiahnuté zmenou znamienka osi X.
3. Left-Handed Rectangular - ľavotočivý (ľavoruký) kartézsky systém. Dosiahnuté zmenou znamienka osi Y.
4. Left-Handed Rectangular - ľavotočivý (ľavoruký) kartézsky systém. Dosiahnuté zmenou znamienka osi Z.
5. Cylindrical Clockwise system – cylindrický systém (v smere hodinových ručičiek)
6. Cylindrical Counter-Clockwise system – cylindrický systém (v proti-smere hodinových ručičiek)
7. Spherical Clockwise system - sférický systém (v smere hodinových ručičiek)
8. Spherical Counter-Clockwise system - sférický systém (v proti-smere hodinových ručičiek)

Z vyššie uvedených možnosti budeme využívať kartézsky a sférický súradnicový systém. Kartézsky je potrebný kvôli, potrebe merania presnosti v jednotlivých osách. Voľba sférického systému je na základe požiadavku merania dĺžok (vracia dva uhly a polomer).



Obrázok 17: Sférický súradný systém trackeru

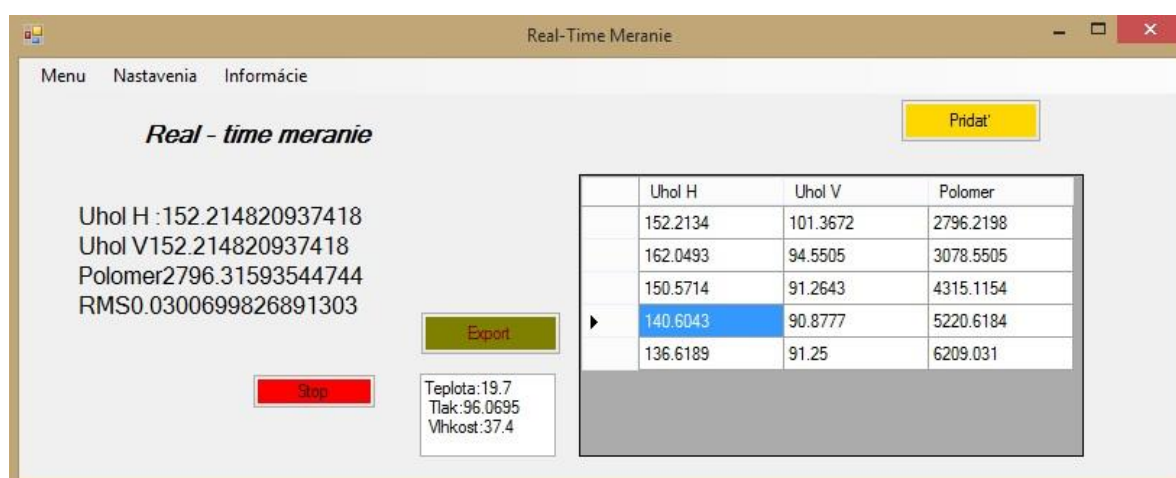
Meracie módy

1. Stacionárny mód merania – jedná sa o meranie jedného bodu, kde cieľ (objekt) merania je stacionárny. Meraný bod je priemerom niekoľkých meraných bodov.
2. Kontinuálny meranie – jedná sa o kontinuálne meranie, kde je reflektor stále snímaný na základe časového intervalu.
3. Kontinuálne snímanie dĺžok
4. Grid
5. SphereCenter
6. CircleCenter
7. Stacionárny mód merania 6D
8. Kontinuálny meranie 6D
9. Kontinuálne snímanie dĺžok 6D
10. Grid 6D
11. SphereCenter 6D
12. CircleCenter 6D

Zo vypísaných meracích metód bude hlavne využitá prvá metóda (Stacionárny mód merania). Dôvodom je jednoduchšia obsluha a menšia náročnosť z hľadiska výkonu samotného počítača a riadiacej jednotky Laser Trackeru. Aplikácia bude ešte obsahovať možnosť kontinuálneho režimu a režim kontinuálneho snímania dĺžok pre špecifické prípady.

9.6 Real-time zobrazenie

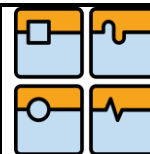
Jednou z hlavných častí aplikácie je zobrazovanie v reálnom čase. Je potrebné navrhnuť danú časť tak aby bolo možné sledovať hodnoty súradníc a zároveň nezahltili systém. Preto je zvolená možnosť stacionárneho meracieho módu, ktorá bude cyklicky volaná. (*MeranieBoduSta()*) Samozrejme bude tiež obsahovať tabuľku do ktorej bude možné vkladať zobrazované hodnoty. Aby metóda nebola stále volaná, bude vhodné doplniť aplikáciu možnosťou spustenia a ukončenia zobrazovania. Daný návrh je zobrazený na obrázku nižšie.



Obrázok 18: RT meranie

Samotné vypisovanie hodnôt je zabezpečené pomocou objektu Timer, ktorý obsluhuje náležité metódy. Hodnotu času pre timer je možné meniť v nastavení. Pre neznalých užívateľov sa doporučuje ponechať východzie nastavenie. (prípadne doporučené na základe požiadavky) Kód obsahujúci daný objekt :

```
private void timer1_Tick(object sender, EventArgs e)
{
    // Stavový automat, ktorý slúži na rozlíšenie merania podľa sur. systému
    switch (SetParametre.Parametre.SurSystem)
```



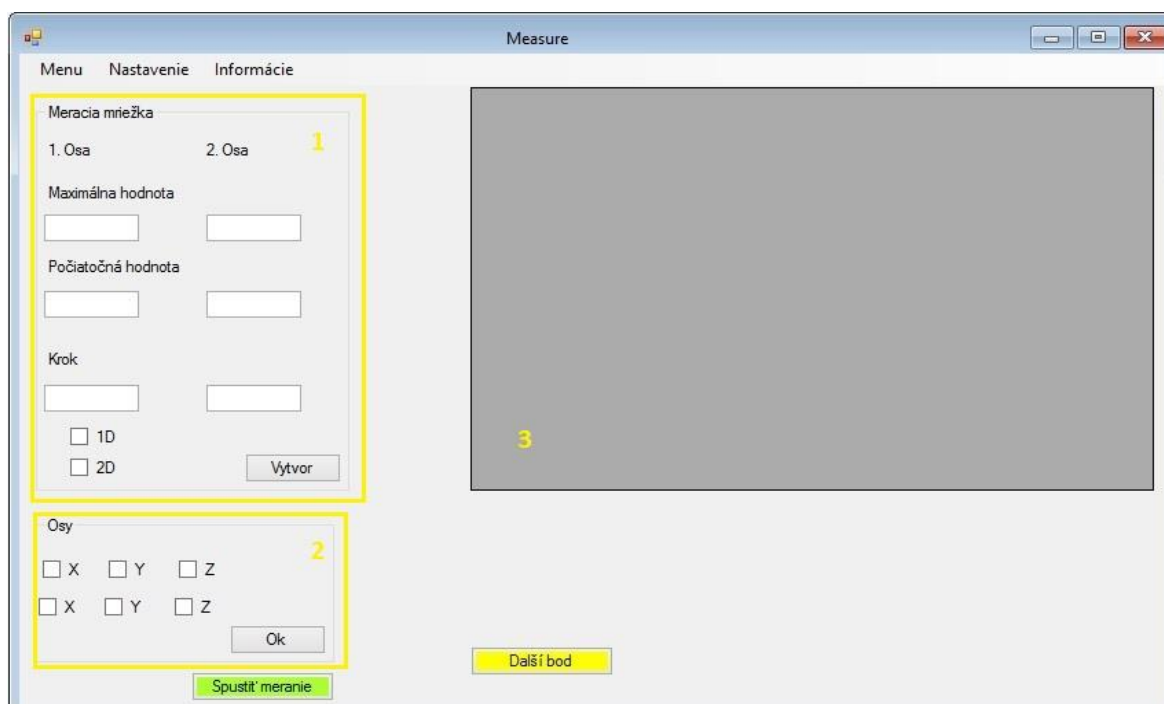
```
{  
    case "Kartézsky súradnicový systém":  
        label1.Text = LaserTracker.Tracker.MeranieBoduSta();  
        break;  
  
    case "Sférický súradnicový systém":  
        label1.Text = LaserTracker.Tracker.MeranieBoduSta2();  
        break;  
}  
    i++;  
if (i==10)  
    {  
        //aktualizácia parametrov prostredia  
        textBox1.Text = LaserTracker.Tracker.ParamProstredia();  
        i=0;  
    }  
}
```

Okno ďalej obsahuje údaje o hodnotách prostredia (teplota, tlak..) a tiež možnosť exportu do excelu uložených hodnôt v tabuľke. Zobrazované hodnoty je možné vložiť do tabuľky pomocou tlačítka *Pridať*, stlačením klávesy *p* alebo použitím klávesy *Ok* na diaľkovom. Samotný modul obsahuje možnosť nastavenia súradnicového systému kartézkeho aj sférického (Nastavenia → Meranie). Pri zmene nastavenia musí byť zobrazovanie vypnuté.

9.7 Meranie

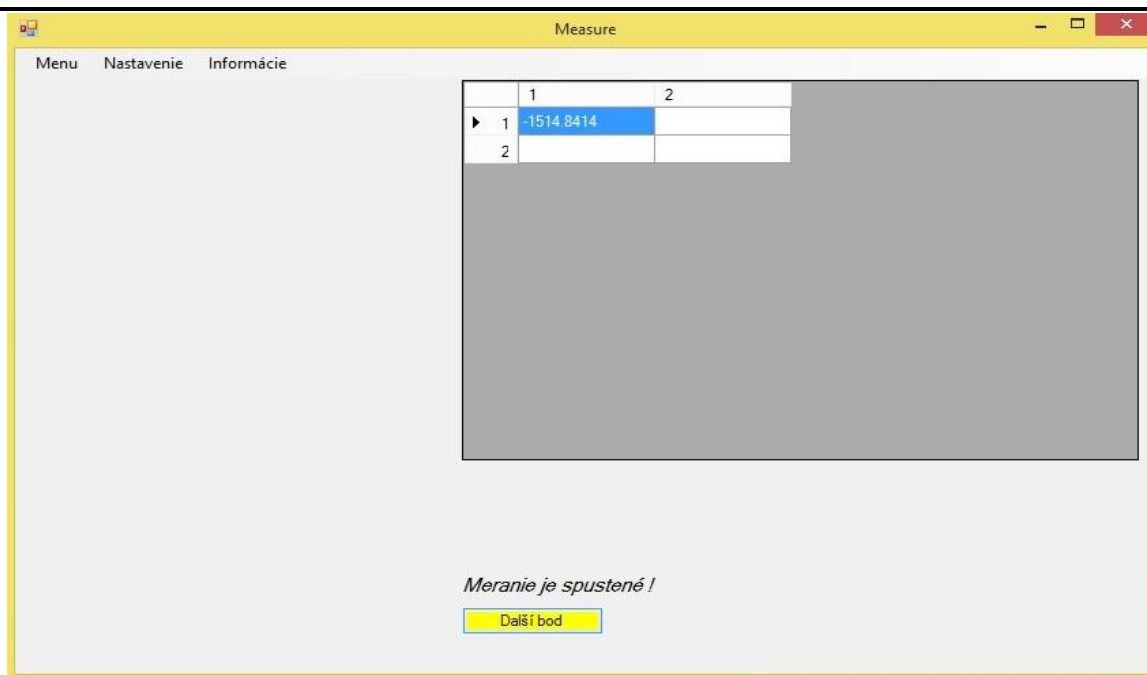
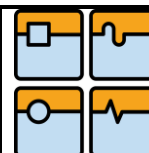
Ďalším modulom aplikácie je meranie. Jedná sa o meranie buď jednej osi (1D), napríklad rovinnosť, alebo dvoch (2D), napríklad kolmosť. Výber typu merania závisí od toho čo bude merané. Celé meranie bude vytvorené ako postupný (krok za krokom) mechanizmus. Na začiatku si užívateľ zvolí typ a počet bodov merania (na základe kroku a minima/maxima). Kvôli jednoduchšej orientácii a možnosti spracovania, budú merania

robené v postupnej tabuľke. Tabuľka sa vytvorí na základe vložených dát. Následne sa do nej budú zapisovať namerané hodnoty.



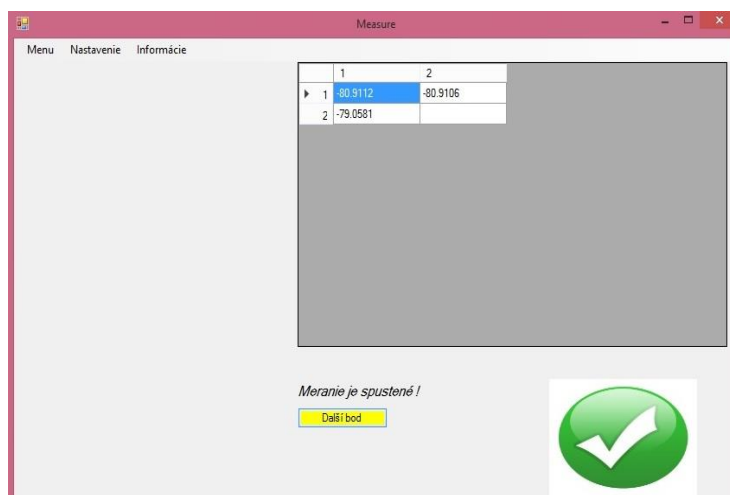
Obrázok 19: Design merania

Ako je vidieť z obrázku, postupuje sa z vrchu na dol. V prvej časti sú zadávané informácie na vytvorenie tabuľky (1)-zadáваме počiatočnú a koncovú hodnotu a krok akým budeme po nej postupovať(napr. začíname od 50 mm do 100mm po 10mm → 50-100-10). Tým bude vytvorená pomyselná mriežka bodov v ktorej sa bude meranie pohybovať. V druhej časti sa vyberú osy(2). Následne sa nám vytvorí tabuľka(3). Hodnoty sú pridávané pomocou tlačidla *Další bod*. Zároveň je v tabuľke zvýraznené, ktorý bod bude meraný(vid' obrázok nižšie). Pridávanie hodnôt je možné pomocou tlačítka *Další bod* alebo stlačením klávesy Ok na diaľkovom.



Obrázok 20: Ukážka z merania

Celý modul bude je následne doplnený o ďalšiu možnosť. Doplníme samotný proces merania o možnosť porovnania nameranej odchýlky voči požadovanej a na základe nastavenia. Na základe splnenia hranice bude užívateľ informovaný či je nameraná hodnota správna alebo nesprávna.



Obrázok 21: Správne meranie

Všetky namerané hodnoty sú posielané do triedy Protocol. Tam sú následné hodnoty spracované, upravené a podľa potreby aj vykreslené. Výpis vybraných metód objektu Meranie :

```
public Measure()
```

```
private void Form_Start()
```

private void ZlaHodnota() – upozornenie na zle nameranú hodnotu
private void SpravnaHodnota() - upozornenie na správne nameranú hodnotu
private void Form_Create()
public void CreateTable_1D() – vytvorenie tabuľky pre 1D meranie
public void CreateTable_2D() - vytvorenie tabuľky pre 2D meranie
private void GetSuradnice() – načítanie súradného systému
private void Meranie_1D(int Riadok, int Stlpec) – 1-krokové meranie
private void Meranie_1D_Step() – Celé meranie krok za krokom
private void Meranie_2D_Step()– Celé meranie krok za krokom
private double[,] VytvoreniePola(DataGridView table) – vytvorenie poľa z tabuľky nameraných hodnôt

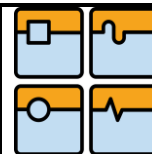
Metóda, ktorá zabezpečuje predanie hodnôt triede Protocol :

```

private void vyhodnotenieToolStripMenuItem_Click(object sender, EventArgs e)
{
    Protocol Protokol=new Protocol();    //vytvorenie instance triedy
    Protokol.Nacitanie(dataGridView1); //Načítanie dát z tabuľky
    Protokol.Form_Measure();
    Protokol.Show();    //Zobrazenie
}

```

Celý modul Meranie zvláda meranie a ukladanie nameraných dát. Obsahuje zároveň základné možnosti nastavenia v rámci merania. Vzhľadom k tomu, že problematika vytvárania takto obiahlej aplikácie je veľmi veľká. Preto je modul postupne stále doplnovaný a testovaný aby sa dosiahlo čo najväčších možností.



9.8 Bodové meranie

Meranie, ktoré je určené na meranie jedného, totožného bodu viackrát. Nie je určené pre veľké množstvo bodov. V rámci merania budú vyhodnocované neistoty merania.

ID	Súradnica X	Súradnica Y	Súradnica Z	RMS
0	-372.8686	-804.3592	39.7564	0.000570589906...
1	-372.8686	-804.3592	39.7562	0.000523899692...
2	-372.8686	-804.3594	39.756	0.001428300426...
	-372.8686	-804.3592	39.7563	0.000547244799...

Obrázok 22: Bodové meranie

Všetky opakované merania bodu sa zobrazujú v tabulke. Po ukončení merania bodu je potrebné kliknúť na tlačidlo *Ulož*. Po kliknutí sa body spriemerujú a uložia. Pokiaľ bude meraný ďalší bod stlačíme *Ďalší bod*. Všetky namerané hodnoty sa ukládajú do objektových Listov(List<T>) :

```
List<List<Bod>> MeraneBody = new List<List<Bod>>();
```

```
List<Bod> FinalPoint = new List<Bod>(); - spriemerované body
```

```
List<Bod> Meranie = new List<Bod>(); - meraný bod(niekoľkokrát)
```

Samotná trieda Bod :

```
public class Bod
{
    //objekt bod
    public double Value1;
    public double Value2;
    public double Value3;
    public double Val_RMS;
}
```

Všetky namerané hodnoty sa predávajú do triedy Protocol. V danej triedy tiež prebieha výpočet neistôt merania.

Daný modul je vhodný pre určovanie neistôt, chýb merania alebo hľadania závislostí od okolia na danom meraní. V rámci testovania bolo spracované meranie. Vybraný bod bol meraný 30x, aby sme mohli vyjadriť chybu, ktorá nastáva pri konkte reflektora s plochou. Zpriemerovaná hodnota RMS je rovná $0.00293 \text{ mm} = 2.93 \mu\text{m}$. Všetky namerané hodnoty sú priložené v prílohe : Príloha C : Namerané hodnoty bodu.

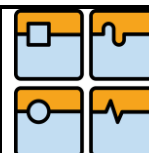
9.9 Protokol

Jedná sa o časť programu, ktorá zabezpečuje spracovanie nameraných dát, ich vykreslenie a export. Celé vykresľovanie prebieha v samostatnom programe GnuPlot (viac informácii viď. Príloha B : GnuPlot). Namerané dáta sa predávajú pomocou triedy GnuPlot priamo programu s nastaveniami pre softvér GnuPlot. Daný prístup bol zvolený, kvôli problematike nájst vhodnú knihovňu na vykresľovanie v 3D (väčšina knihovien je platená). Metódy pre vyhodnocovanie sú vytvorené špeciálne pre naše jednotlivé účely, keďže vytvorenie univerzálnej metódy by bolo veľmi náročné a samotná funkcia by bola veľmi neprehľadná. V rámci práce bolo vyhodnocovanie zamerané na vyhodnocovanie rovinnosti.

9.9.1 Vyhodnotenie pre rovinnosť

Po úspešnom zmeraní hodnôt (Modul Meranie-1D), v záložke Menu-Vyhodnotenie bude spustená časť Protokol. Z načítaných dát sa vypočíta priemerná hodnota, ktorá sa zobrazí v prvom okne. Aby sme sa „zbavili“ kartézsky súradníc, celá tabuľka prejde korekciou. Tá je vykonaná pomocou rozdielu všetkých nameraných hodnôt od priemernej hodnoty. Kód výpočtu :

```
private double Korekcia_1D()
{
    var tuple=MinMax(dataGridView1);
    double Priemer = System.Convert.ToDouble(textBox1.Text);
    int CountPoint = SetParametre.Parametre.CountNumber;
    foreach (DataGridViewRow row in dataGridView1.Rows)
```



```
{  
    foreach (DataGridViewCell cell in row.Cells)  
    {  
        double Val = System.Convert.ToDouble(cell.Value);  
  
        if (Val>Priemer)  
        {  
            cell.Value = Math.Round(Math.Abs(Math.Abs(Priemer) - Math.Abs(Val)), CountPoint);  
        }  
        else  
        {  
            cell.Value = Math.Round(-1*Math.Abs((Math.Abs(Priemer)- Math.Abs(Val)) ),CountPoint);  
        }  
    }  
}  
  
double delta = Math.Abs(Minimum) - Math.Abs(Maximum);  
delta = Math.Abs(delta);  
return delta;  
}
```

Celá funkcia vracia hodnotu delta, ktorá predstavuje najväčšiu odchýlku v rovinne (rozdiel hodnôt minima a maxima v danej ose). Takto spracované data môžu byť následne vykreslené. Takto realizované vyhodnocovanie nie je vhodné pre naklonené roviny.

9.9.2 Výpočet neistôt meraní

Ako bolo v predchádzajúcich častiach zmienené, výpočet neistôt sa vzťahuje v časti programu Bodové meranie. Všetky body musia mať rovnaký počet premeraní (opakovaných meraní). Z opakovaných meraní sa vypočíta priemerná hodnota, ktorá sa uloží do pamäte. (konkrétne do Listu FinalPoint) Celý výpočet neistôt merania sa vzťahuje k nameraným dĺžkam. Pokiaľ nie je pri meraní použitý potrebný súradnicový systém, je potrebné hodnoty prepočítať. Výpočet neistôt bude postupne popísaný.

1. Neistota typu A

Je počítaná podľa vzťahu 1.1 . Hodnoty sú brané z listov *MeraneBody*, ktorý obsahuje všetky namerané hodnoty daného bodu. Vypočítaná neistota sa ukladá opäť do pamäte pre ďalšie využitie. (List U_a)

2. Neistota typu B

Neistota typu je tvorená viacerými zdrojmi. Prvým zdrojom je samotná chyba Laser Trackeru, ktorá je uvedená v katolígu. $u_B = \pm(7 \mu\text{m} + 3 \mu\text{m}/\text{m})$. Táto neistota sa v berie úvahu ak využívame ADM meranie. V prípade IFM je hodnota $u_B = \pm 0.5 \mu\text{m}/\text{m}$ (Nastavenie → Meranie → Použitie ADM). Ďalším zdrojom neistôt je proces meranie. Pre stanovenie tejto neistoty bude využitý parameter RMS, ktorý vracia odchýlku. Tá bude následne prepočítaná podľa vzťahu 1.3 .

$$u_B(x_i) = \frac{RMS}{\chi} \quad (2.0)$$

Celková neistota je rovná kvadratickému súčtu jednotlivých neistôt. Vypočítaná neistota je opäť uložená(list U_b).

3. Štandardná a rozšírená kombinovaná neistota

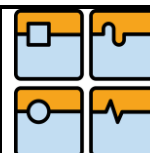
Dané neistoty sú dopočítané podľa vzťahov 1.5 a 1.6, pričom koeficient rozšírenia $k=2$. Následne sú hodnoty uložené. (U_c, U_x)

9.10 Nastavenia

Každý modul má svoje nastavenia a tiež aj Laser Tracker. Aby bolo možné k týmto nastaveniam pristupovať, bude vytvorené okno „Nastavenia“ ,pre daný účel. Je vytvorené univerzálne okno pre všetky nastavenia. Aby bolo zamedzené, užívateľovi „omylom“ meniť iné hodnoty ako tie, ktoré su požadované, sú nepotrebné okno „schované“ .

Všetky hodnoty sú uložené v triede *SetParametre*, ktorá sa vytvára iba raz. Obsahuje všetky východzie nastavenia. Z tejto triedy sú načítavané hodnoty pre všetky merania a rozhrania aplikácie. Časť kódu triedy *SetParametre* :

```
class SetParametre
```



```
{ ....  
private SetParametre()  
{  
    //východzie nastavenia  
    SignUser = false;  
    TimeMeas = 800;  
    TimeTimer = 1700;  
    iDialogoveOkno = false;  
    StatusLabel = true;  
    SignIn = false;  
    IDCoordinateSystem = 0;  
    useADM = false;  
    untilRMS = 0.01;  
    CountNumber = 4;  
    LimitRMS = 0.01;  
    SuradnicovySystem = "Kartézsky súradnicový systém 1";  
}  
}
```

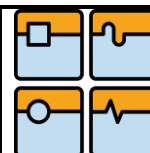
Obrázok 23: Okno Nastavenia

9.11 Export

Jedná sa o objektovú triedu, ktorá zabezpečuje export do súborov *.xlsx a *.tcd. Súbory *.tcd sú určené pre softvér Trac-cal. Dôvodom voľby pre export do daného súboru je využitie vyhodnocovania a porovnávania merania v softvéri Trec-cal. (Trac-cal) Súbory *.xlsx (Microsoft office excel) sú vhodné na ďalšie spracovávanie údajov. Ich výhodou je tiež možnosťou ich načítania v ďalších softvéroch ako napríklad Matlab.

class Export

```
{
    //trieda obsahujúce metódy na export do rôznych súborov
    public void ExportToExcel(DataGridView MeasureData)
    public void EportToTcd(DataGridView MeasData)
}
```



10 Další vývoj

Takto vytvorená aplikácia je základným kameňom pre ďalšie možnosti vývoja aplikácie (softvéru) pre meranie s Laser Trackerom. Vytvorená aplikácia dokáže merať ako jednotlivé súradnice tak aj dĺžiek. Všetky namerané hodnoty sú možné exportovať do excelovského súboru, kde môžu byť ďalej spracované, prípadne môžu byť určité vyhodnotenia vykonané priamo v aplikácii. Aplikácia nie je konečná, jej vývoj je otázka desiatok mesiacov (postupne sa bude ďalej dopĺňať a ladiť).

Ďalšími možnosťami rozšírenia aplikácie je rozšírenie o ďalšie metódy vyhodnocovania a spracovania neistôt merania, ovládanie na diaľku pomocou mobilných zariadení (zatiaľ je možnosť ovládania pomocou „remote controler“), vytvorenie vlastnej knihovne na vykresľovanie v 3D, možnosť práce s datami na mobilných telefónoch a tabletoch. V následných krokoch by mohla byť aplikácia rozšírená na spracovávanie chýb v číslícovo riadených osách.

11 Záver

Na začiatku samotnej práce bol spracovaný prehľad dostupných softvérových riešení (Softvér pre meranie s Laser Trackerom). Jednotlivé možnosti boli popísané a porovnané. Ich hlavnou nevýhodou je cena a obmedzenosť použitia.

V ďalšej časti diplomovej práce boli poukázané možnosti vývoja vlastnej aplikácie pre meranie so zariadeným Laser Tracker od firmy Leica. Boli rozobraté ich výhody a nevýhody a na základe odtestovania jednotlivých možností bola zvolená možnosť vytvorenia aplikácie pomocou knihovne *LTControl*, ktorá je dodávaná k základnému softvéru emScon, v jazyku C#(Sharp) s využitím prostredia Microsoft Visual Studio 2013.

Prí navrhovaní aplikácie bolo prvým krokom odtestovanie možností knihovne *LTControl*. Pre tento účel bola vytvorená testovacia aplikácia (Testovacia aplikácia pre zisťovanie možností SDK) pri ktorej boli zisťované možnosti a obmedzenia danej knihovne. Na základe získaných poznatkov bolo prístupné k návrhu danej aplikácie.

Na základe zadaných požiadaviek bola aplikácia rozdelená do piatich častí: RT meranie, Meranie, Bodové meranie, Nastavenia, Vyhodnocovanie. V práci sú postupne jednotlivé časti rozobrané a popísané bližšie.

Vývoj aplikácie bol sprevádzaný množstvom problémom, ktoré bolo potrebné postupne riešiť. Aplikácie bola priebežne testovaná (boli vykonávané meranie) a ladená. Do testovania boli zapojení aj ďalší ľudia, aby boli odhalené ďalšie nedostatky, prípadne nevhodné volené rozloženie funkcií.

Samotnou aplikáciou je možné merať ako v kartézskom tak aj sférickom súradnicovom systéme. Namerané dáta je možné ukladať a v prípade potreby exportovať do excelovského súboru alebo súboru *.tcd (súbory systému Trac-cal). Aplikácia tiež ponúka množstvo nastavení od voľby zobrazovania až po nastavenie meraní (priemerovací čas, použitie ADM a pod.). Po úspešnom nameraní je možné v aplikácii spraviť vyhodnotenie meranie a vykreslenie. Pre vykreslenie využívame externého programu GnuPlot (jedná sa o opensource). Meracia časť je doplnená o možnosť diaľkového ovládania, čo uľahčuje pri meraní veľkých vzdialeností prácu.

Str. 66	Ústav výrobných strojů, systémů a robotiky	
	DIPLOMOVÁ PRÁCE	

Samotná časť protokol bola zameraná na spracovanie a vyhodnocovanie nameraných dát. V rámci tejto kapitoly boli stanovené a algoritmizované výpočty neistôt merania.

Problematika neistôt merania pre laser tracker je veľmi obsiahla a problematická a preto sa postupne táto časť bude v ďalších krokoch rozvíjať a ľadiť.

Výhodou vytvorenej aplikácie je jej jednoduchosť, prívetivosť a možnosť stáleho roširovania. Aplikácia zatiaľ obsahuje základné veci, ale v budúcnosti bude ďalej rozšírená o možnosť ovládania pomocou mobilných zariadení, metódami na spracovanie a podobne.

12 Použitá literatura, zdroje

- [1] P. Diemeč, Presnosť obrábacích strojov a jej matematické modelovanie, Košice: SjF TU v Košiciach, 2001, p. 145.
- [2] J. Ženíšek a M. Jenkut, Výrobní stroje a zařízení, Praha: Alfa, 1986, p. 276.
- [3] J. Marek, Konstrukce obráběcích strojů, Brno: MM publishing, 2010, p. 420.
- [4] Jiří Marek a kol., Konstrukce CNC obráběcích strojů III, Brno: MM publishing, 2014.
- [5] V. Antošovský, Elektrické merania I., Alfa, 1989, p. 456.
- [6] Ing. František Doleček, doc. Ing. Rudolf Palenčár, doc. Ing. Jiří Pernikář, „Chyby a nejistoty v měření,“ Brno, 1999.
- [7] d. D. I. P. Němeček, Nejistoty měření, Praha: Česká společnost pro jakost, 2008, p. 98.
- [8] K. I. t. p. (KLTP), „Neistoty,“ 2009. [Online]. Available: <http://www.senzorika.leteckafakulta.sk/?q=node/222>. [Cit. 4 Marec 2015].
- [9] „Laser Trackers – IFM vs ADM Technology,“ FARO Technologies, 22 September 2009. [Online]. Available: <http://farotechnologies.blogspot.cz/2009/09/laser-trackers-ifm-vs-adm-technology.html>. [Cit. 10 Apríl 2015].
- [10] „API Tracker3 Laser Tracker,“ [Online]. Available: <http://www.apitechnical.com/tracker3-laser-tracker/>. [Cit. 15 Február 2015].
- [11] „Radian Laser Tracker – an API Laser Tracker,“ [Online]. Available: <http://www.apitechnical.com/metrology-equipment/api-laser-tracker-radian/>. [Cit. 10 December 2014].
- [12] „LaserTRACER-NG,“ [Online]. Available: <http://www.etalon-ag.com/products/lasertracer/?lang=en>. [Cit. 14 Marec 2015].
- [13] [Online]. Available: <http://www.nms.sk/produkty/polyworks/>. [Cit. 18 November 2014].
- [14] P. Herout, Učebnica jazyka C, České Budějovice: Kopp, 1997.
- [15] „C++,“ 19 Január 2015. [Online]. Available: <http://sk.wikipedia.org/wiki/C%2B%2B>. [Cit. 21 Február 2015].

Str. 68	Ústav výrobních strojů, systémů a robotiky	
	DIPLOMOVÁ PRÁCE	

[16] D. Morkes, Učebnice Visual Basic 6.0, Praha: Computer Press, 2000.

[17] „C Sharp,“ 12 Júl 2014. [Online]. Available: http://sk.wikipedia.org/wiki/C_Sharp. [Cit. 20 Január 2015].

[18] „Microsoft Visual Studio,“ [Online]. Available: http://en.wikipedia.org/wiki/Microsoft_Visual_Studio. [Cit. 10 November 2014].

13 Príloha A : Trieda Laser Tracker

```

public class LaserTracker
{
    private static LaserTracker tracker;

    private LaserTracker()
    { }

    public static LaserTracker Tracker
    {
        get
        {
            if (tracker == null)
            {
                tracker = new LaserTracker();
            }
            return tracker;
        }
    }

    private LTConnect conn; //vytvorenie prepojenia na riadiacu jednotku
    private LTCommandSync sync; //objekt na synchronnu komunikáciu

    public double Measure_1;
    public double Measure_2;
    public double Measure_3;
    public double RMS;

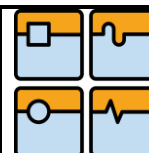
    public void Connect(string ipaddress)
    {
        conn = new LTConnect(); //vytvorenie prepojenia

        this.sync = (LTCommandSync) conn.ILTCommandSync;
        conn.SelectNotificationMethod(LTC_NotifyMethod.LTC_NM_Event, 0, 0);
        conn.ConnectEmbeddedSystem(ipaddress, 700);
    }

    public void Disconnect()
    {
        conn.DisconnectEmbeddedSystem();
    }

    public void Initialize()

```



```
{
    try
    {
        sync.Initialize();
        //natocenie do "zachytavacej polohy"
        GoHome();
    }
    catch (Exception except)
    {
        MessageBox.Show(except.Message);
    }
}

public void GoHome()
{
    try
    {
        //Natocenie do vychodzej polohy
        sync.GoBirdBath();
    }
    catch (Exception except)
    {
        MessageBox.Show(except.Message);
    }
}

public string StatusTracker()
{
    //status
    ES_TrackerStatus status = new ES_TrackerStatus();
    sync.GetTrackerStatus(out status);
    String msg = "Tracker Status: " + status.ToString();
    return (msg);
}

public string ParamProstredia()
{
    //vypis parametrov prostredia
    double teplota;
    double tlak;
    double Vlhkost;
    string Params = "";

    try
    {
```

```

sync.GetEnvironmentParams(out teplota, out tlak, out Vlhkost);

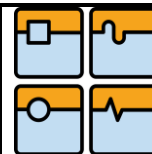
Params = "Teplota:" + teplota.ToString() + "\r\n Tlak:" + tlak.ToString() + "\r\n
Vlhkost:" + Vlhkost.ToString();

    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message);
    }
    return (Params);
}

public string MeranieBoduSta()
{
    //Meranie
    double X;
    double Y;
    double Z;
    double std1, std2, std3;
    double stdTotal;
    double Err1, Err2, Err3;
    double astd1, astd2, astd3;
    double astdTotal;
    double teplota1, tlak1, vlhkost1;
    bool TryMode;
    string Suradnice=" ";

    try
    {
        sync.MeasureStationaryPoint(out X, out Y, out Z, out std1, out std2, out std3, out
stdTotal, out Err1, out Err2, out Err3, out astd1, out astd2, out astd3, out astdTotal, out
teplota1, out tlak1, out vlhkost1, out TryMode);
        Suradnice = "X: " + X.ToString() + "\r\n Y: " + Y.ToString() + "\r\n Z: " + Z.ToString() +
Environment.NewLine + "RMS: " + stdTotal.ToString();
        Measure_1 = Math.Round(X, SetParametre.Parametre.CountNumber);
        Measure_2 = Math.Round(Y, SetParametre.Parametre.CountNumber);
        Measure_3 = Math.Round(Z, SetParametre.Parametre.CountNumber);
        RMS = stdTotal;
    }
    catch(Exception e)
    {
        MessageBox.Show(e.Message);
    }
}

```



```
    return (Suradnice);
}

public void NastavenieADM(double targetStabilityTolerance,int retryTimeFrame,int
NumberOfRetry)
{
    //Nastavenie ADM merania
    sync.SetAdmParams(targetStabilityTolerance, retryTimeFrame, NumberOfRetry);
}

public void SetMeasurement()
{
    int time_meas = SetParametre.Parametre.TimeMeas;
    string SurSys = SetParametre.Parametre.SurSystem;
    bool ADM = SetParametre.Parametre.useADM;
    switch (SurSys)
    {
        case "Kartézsky súradnicový systém":
            sync.SetCoordinateSystemType(ES_CoordinateSystemType.ES_CS_RHR);
            break;

        case "Sférický súradnicový systém":
            sync.SetCoordinateSystemType(ES_CoordinateSystemType.ES_CS_SCW);
            break;
    }

    sync.SetMeasurementMode(ES_MeasMode.ES_MM_Stationary);
    sync.SetStationaryModeParams(time_meas, ADM);
}
}
```

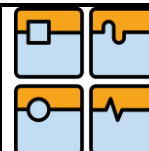
14 Příloha B : Gnuplot

Gnuplot³ je prenosný, príkazovým riadkom riadený grafický nástroj pre Linux, OS / 2, MS Windows, OS X, VMS, a mnoho ďalších platforiem. Zdrojový kód je chránený autorskými právami, ale voľne distribuovaný . Pôvodne bol vytvorený pre vedcov a študentov aby mohli interaktívne prezentovať matematické funkcie a dáta, a tiež aby rozrástli podporovanosť ne-interaktívne skriptovania, ako je napríklad webové skriptovanie. Gnuplot bol podporený a aktívne vyvíjaný od roku 1986.

Výstupné formáty :

- Postscript (vrátane eps)
- Pdf
- Png
- gif
- Jpeg
- LaTeX,

³ <http://www.gnuplot.info/>



15 Příloha C : Namerané hodnoty bodu

ID	Súradnica X	Súradnica Y	Súradnica Z	RMS
0	-0.036	96.3387	2473.9172	0.002554003
1	-0.036	96.339	2473.9169	0.002817062
2	-0.0361	96.3385	2473.9166	0.004759367
3	-0.0362	96.339	2473.916	0.003884812
4	-0.0363	96.339	2473.9156	0.002219864
5	-0.0361	96.3391	2473.9156	0.003065434
6	-0.0362	96.3388	2473.9156	0.003294641
7	-0.0362	96.339	2473.9153	0.007775787
8	-0.0361	96.3388	2473.915	0.001081098
9	-0.0363	96.3388	2473.9141	0.002807016
10	-0.0363	96.3387	2473.9169	0.001518262
11	-0.0362	96.3389	2473.9172	0.001494268
12	-0.0363	96.339	2473.9172	0.002578603
13	-0.0364	96.339	2473.9175	0.005972733
14	-0.0363	96.3389	2473.9182	0.002000894
15	-0.0363	96.3388	2473.9182	0.001101102
16	-0.0363	96.3389	2473.9179	0.00132465
17	-0.0364	96.3388	2473.9179	0.003191652
18	-0.0364	96.3388	2473.9179	0.003191652
19	-0.0363	96.3387	2473.9182	0.002098766
20	-0.0362	96.3388	2473.9179	0.003502776
21	-0.0364	96.3388	2473.9179	0.001611189
22	-0.0363	96.3387	2473.9182	0.003064719
23	-0.0363	96.3388	2473.9185	0.001081447
24	-0.0363	96.3387	2473.9185	0.003654359
25	-0.0362	96.3388	2473.9185	0.004242159
26	-0.0364	96.3389	2473.9185	0.004665556
27	-0.0362	96.3388	2473.91	0.003500758
28	-0.0361	96.3388	2473.9096	0.001866559
29	-0.0362	96.3388	2473.9096	0.00182512

Tlak : 96.7 kPa

Teplota : 21.4 °C

Vlhkosť: 46.6 %

16 Příloha D : Načítanie xlsx súboru pomocou Matlabu

% Scrip ktorý slúži ako ukážka načítavania dát zo súboru

% ktorý je exportovaný vytvorenou aplikáciou⁴.

clear all

close all

clc

filename = 'MeraneData.xlsx';

A = xlsread(filename, '1:31');

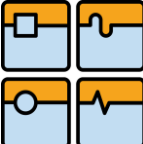
T = array2table(A, 'VariableNames', {'ID' 'X' 'Y' 'Z' 'RMS'});

Otestovaný príklad je priložený v prílohách k diplomovej práci.

⁴ <http://www.mathworks.com/help/matlab/ref/xlsread.html>

17 Zoznam obrázkov

Obrázok 1: Laser Tracker AT901	27
Obrázok 2: Pracovisko s Laser Trackerom	28
Obrázok 3 : Laser Tracker AT401	29
Obrázok 4: Leica AT960	29
Obrázok 5 : Tracker 3.....	30
Obrázok 6 : Radian.....	30
Obrázok 7: Laser Tracer [12].....	31
Obrázok 8:emScon.....	34
Obrázok 9:Polyworks	35
Obrázok 10: Rozhranie Trac-cal	36
Obrázok 11: Rozhranie Visual Studia	42
Obrázok 12: Blokový diagram návrhu.....	44
Obrázok 13: Funkčný diagram návrhu aplikácie.....	45
Obrázok 14: Rozhranie testovacej aplikácie	46
Obrázok 15: Hlavné okno.....	48
Obrázok 16: Objektový model aplikácie	49
Obrázok 17: Sférický súradný systém trackeru	52
Obrázok 18: RT meranie	53
Obrázok 19: Design merania.....	55
Obrázok 20: Ukážka z merania	56
Obrázok 21: Správne meranie	56
Obrázok 22: Bodové meranie	58
Obrázok 23: Okno Nastavenia	62

	Ústav výrobných strojů, systémů a robotiky	Str. 77
	DIPLOMOVÁ PRÁCE	

18 Přílohy

- Video – priečinok obsahujúci videá, ktoré slúžia na predstavenie jednotlivých častí vytvorenej aplikácie
- LaserTracker_App – zložka obsahujúca riešenie (Solutions), ktorá v sebe zahŕňa projekt vytváranej aplikácie a projekt na vytvorenia inštalačného balíčka danej aplikácie
- Setup – priečinok obsahujúci inštalačné balíky pre nainštalovanie aplikácie
- Laser Tracker for measurement – samotná spustiteľná aplikácia
- Readxlsx – ukázkový príklad na načítanie excelovského súboru v sw matlabe
- Poster_DP.pdf – plagát o diplomovej práci
- Originálna práca vo formáte docx a pdf