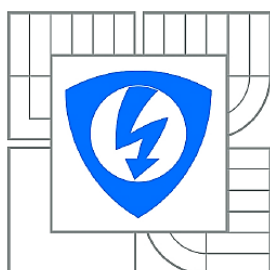


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

# URČOVÁNÍ 2D POLOHY POMOCÍ ULTRAZVUKOVÝCH SNÍMAČŮ

2D ULTRASONIC POSITIONING

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JIŘÍ JUNEK

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. ONDŘEJ HYNČICA

BRNO 2013



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Bakalářská práce

bakalářský studijní obor  
Automatizační a měřicí technika

**Student:** Jiří Junek  
**Ročník:** 3

**ID:** 146031  
**Akademický rok:** 2013/2014

## NÁZEV TÉMATU:

**Určování 2D polohy pomocí ultrazvukových snímačů**

## POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvořit systém pro lokalizaci objektu v rovině za pomoci ultrazvukových snímačů.

- 1) Seznamte se s principy a vlastnostmi měření vzdálenosti pomocí ultrazvukových snímačů. Využijte ultrazvukové snímače vzdálenosti a všesměrový převodník.
- 2) Realizujte připojení snímačů k mikrokontroléru a implementujte algoritmus měření vzdálenosti.
- 3) Měřením ověřte vlastnosti ultrazvukových snímačů a proveďte vyhodnocení. Vyhodnoťte možnosti použití pro lokalizaci v rovině.
- 4) Kombinací vysílače a dvou nebo více přijímačů vytvořte systém, který bude pomocí trilaterace určovat polohu vysílače v rovině.
- 5) Proveďte zpracování naměřených dat a implementujte přenos souřadnic do PC v reálném čase.

## DOPORUČENÁ LITERATURA:

- [1] SLOSS, Andrew N, Dominic SYMES a Chris WRIGHT. ARM system developer's guide: designing and optimizing system software. Amsterdam: Elsevier, 2004, 689 s. ISBN 15-586-0874-5.
- [2] Controller Area Network. CAN IN AUTOMATION (CIA). Controller Area Network [online]. [cit. 2013-02-01]. Dostupné z: <http://www.can-cia.org/index.php?id=can>

**Termín zadání:** 10.2.2014

**Termín odevzdání:** 26.5.2014

**Vedoucí práce:** Ing. Ondřej Hynčica

**Konzultanti bakalářské práce:**

**doc. Ing. Václav Jirsík, CSc.**

*Předseda oborové rady*

## UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být 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.

## **Abstrakt**

Práce se zabývá realizací a zhodnocením měření směrových i všesměrových ultrazvukových senzorů. Řízení ultrazvukových senzorů a vyhodnocování vzdáleností je realizováno mikrokontrolérem Arduino Duemilanove. První část práce se zabývá měřením směrových senzorů, jejich zapojením, ovládáním a následným zpracováním hodnot v Matlabu. Druhá část práce se zabývá řešením měření a zobrazování 2D polohy všesměrovými ultrazvuky, implementaci algoritmu trilaterace pro výpočet x-ových a y-ových souřadnic a srovnání použitých filtrů. Práce ke konci obsahuje také srovnání směrových a všesměrových ultrazvukových senzorů a zhodnocení jejich použitelnosti pro měření 2D polohy.

## **Klíčová slova**

arduino, lokalizace, měření 2D polohy, měření vzdálenosti, mikrokontrolér, ultrazvuk

## **Abstract**

This thesis deals with the implementation and evaluation of measuring directional and omni-directional ultrasonic sensors. Control of ultrasonic sensors and computation of a distance is realized by using a microcontroller Arduino Duemilanove. The first part deals with the measurement of directional sensors, their wiring and subsequent processing of the measurement is performed in Matlab. The second part deals with solving of measuring and visualization 2D position omni-directional ultrasonic sensors, implementation trilateration algorithm for calculating x and y coordinates and comparison the used filters. At the end of this thesis also includes a comparison of directional and omni-directional ultrasound and evaluate their applicability for measuring 2D position.

## **Keywords**

2D positioning, arduino, localization, measuring, microcontroller, ultrasonic, ultrasound

## **Bibliografická citace:**

JUNEK, J. *Určování 2D polohy pomocí ultrazvukových snímačů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 64 s. Vedoucí bakalářské práce byl Ing. Ondřej Hynčica.

## Prohlášení

Prohlašuji, že svou bakalářskou práci na téma „Určování 2D polohy pomocí ultrazvukových snímačů“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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: .....

.....  
podpis autora

## Poděkování

Děkuji vedoucímu bakalářské práce Ing. Ondřejovi Hynčicovi. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce, také za jeho vstřícnost a ochotu být kdykoliv nápomocen.

V Brně dne: .....

.....  
podpis autora

# Obsah

<b>Seznam obrázků .....</b>	<b>8</b>
<b>Seznam tabulek .....</b>	<b>10</b>
<b>Úvod .....</b>	<b>11</b>
<b>1 Ultrazvuk.....</b>	<b>12</b>
1.1 Vznik ultrazvuku .....	12
1.2 Rychlost šíření ultrazvuku .....	12
1.3 Využití ultrazvuku.....	13
<b>2 Arduino .....</b>	<b>14</b>
2.1 Arduino Duemilanove .....	14
2.2 Programové prostředí Arduino .....	14
<b>3 Použité ultrazvukové senzory .....</b>	<b>15</b>
3.1 US-100 .....	15
3.2 SRF08 .....	16
3.3 Válcový ultrazvukový převodník 80 kHz.....	17
3.3.1 Schéma vysílače .....	18
3.3.2 Schéma přijímače .....	19
<b>4 Používané metody měření ultrazvuků.....</b>	<b>21</b>
4.1 Měření doby letu (TOF) – 1. Metoda .....	21
4.2 Měření doby letu (TOF) – 2. Metoda .....	21
4.3 Ultrazvuková závora.....	22
4.4 Trilaterace .....	22
<b>5 Měření směrových ultrazvuků.....</b>	<b>25</b>
5.1 Postup měření.....	26
5.1.1 Měření senzorem US-100.....	26
5.1.2 Měření senzorem SRF08 .....	27
5.1.3 Měření vyzařovací charakteristiky.....	27
5.2 Získání a zpracování naměřených hodnot.....	27
5.2.1 Základní nastavení.....	27
5.2.2 Zpracování signálu ze senzoru US-100 .....	28
5.2.3 Zpracování signálu ze senzoru SRF08.....	29
5.3 Zpracování naměřených hodnot v MATLABu .....	29
5.4 Vlastní měření a grafy .....	30
5.4.1 Měření senzoru US-100.....	30

5.4.2	Měření vzdálenosti v $\pm 15^\circ$ od osy senzoru .....	32
5.4.3	Měření vyzařovací charakteristiky senzoru US-100 .....	34
5.4.4	Měření senzoru SRF08 .....	34
5.4.5	Měření vzdálenosti v $\pm 15^\circ$ od osy senzoru .....	37
5.4.6	Měření vyzařovací charakteristiky .....	38
<b>6</b>	<b>Měření všesměrových senzorů .....</b>	<b>39</b>
6.1	Měření senzorem PVDF 80 kHz .....	39
6.1.1	Grafy a závislosti .....	39
<b>7</b>	<b>Měření 2D polohy .....</b>	<b>42</b>
7.1	Obsluha ultrazvukových převodníků pomocí Arduina .....	43
7.1.1	Obsluha ultrazvukového vysílače .....	44
7.1.2	Obsluha ultrazvukových přijímačů .....	46
7.2	Realizace výpočtů 2D polohy v Arduinu .....	47
7.3	Realizace výpočtů 2D polohy v Matlabu .....	48
7.4	Real-time zobrazování v Matlabu .....	48
7.5	Filtrace hodnot .....	50
7.5.1	Klouzavý průměr .....	51
7.5.2	Čebyševův filtr .....	51
7.5.3	Butterworthův filtr .....	53
7.6	Dynamické vlastnosti, zhodnocení .....	54
<b>8</b>	<b>Zhodnocení měření .....</b>	<b>59</b>
8.1	Směrové ultrazvukové senzory .....	59
8.2	Všesměrové ultrazvukové senzory .....	60
<b>9</b>	<b>Závěr .....</b>	<b>62</b>
	<b>Literatura .....</b>	<b>64</b>
	<b>SEZNAM PŘÍLOH .....</b>	<b>66</b>

# SEZNAM OBRÁZKŮ

Obr. 1.1 : Sonar [12] .....	13
Obr. 2.1 : Arduino Duemilanove [7] .....	14
Obr. 3.1 : Ultrazvukový senzor US-100 [1] .....	15
Obr. 3.2 : Připojení senzoru US-100 k mikrokontroléru [1] .....	15
Obr. 3.3 : Časový diagram senzoru US-100 [1] .....	16
Obr. 3.4 : Ultrazvukový senzor SRF08 [13] .....	17
Obr. 3.5 : Schéma zapojení senzoru SRF08 [4] .....	17
Obr. 3.6 : Válcový 80 kHz ultrazvukový převodník [14] .....	18
Obr. 3.7 : Schéma 80kHz ultrazvukového PVDF vysílače [14] .....	18
Obr. 3.8 : Schéma 80kHz ultrazvukového PVDF přijímače [14] .....	19
Obr. 4.1 : Jednohlavý systém [8] .....	22
Obr. 4.2 : Dvojhavý systém (ultrazvuková závora) [8] .....	22
Obr. 4.3 : Trilaterace [9] .....	23
Obr. 5.1 : Měřicí stanoviště .....	26
Obr. 5.2 : Charakteristika senzoru US-100 .....	30
Obr. 5.3 : Charakteristika senzoru US-100 (s použitím mediánu) .....	31
Obr. 5.4 : Chyba měření v závislosti na vzdálenosti senzoru US-100 .....	31
Obr. 5.5 : Závislost nejistoty typu A v závislosti na vzdálenosti senzoru US-100 .....	32
Obr. 5.6 : Závislost chyby při $\pm 15^\circ$ a $0^\circ$ vychýlení od osy .....	32
Obr. 5.7 : Závislost nejistoty při $\pm 15^\circ$ a $0^\circ$ vychýlení od osy .....	33
Obr. 5.8 : Závislost rozptylu při $\pm 15^\circ$ a $0^\circ$ vychýlení od osy .....	33
Obr. 5.9 : Vyzařovací charakteristika senzoru US-100 .....	34
Obr. 5.10 : Charakteristika senzoru SRF08 .....	35
Obr. 5.11 : Charakteristika senzoru SRF08 (s použitím mediánu) .....	35
Obr. 5.12 : Chyba měření v závislosti na vzdálenosti senzoru SRF08 .....	36
Obr. 5.13 : Závislost nejistoty typu A v závislosti na vzdálenosti senzoru SRF08 .....	36
Obr. 5.14 : Závislost chyby při $\pm 15^\circ$ a $0^\circ$ vychýlení od osy .....	37
Obr. 5.15 : Závislost nejistoty při $\pm 15^\circ$ a $0^\circ$ vychýlení od osy .....	37
Obr. 5.16 : Závislost rozptylu při $\pm 15^\circ$ a $0^\circ$ vychýlení od osy .....	38
Obr. 5.17 : Vyzařovací charakteristika snímače SRF08 .....	38
Obr. 6.1 : Charakteristika senzoru PVDF 80kHz .....	40
Obr. 6.2 : Charakteristika senzoru PVDF 80 kHz s použitím mediánu .....	40
Obr. 6.3 : Chyba měření v závislosti na vzdálenosti senzoru PVDF 80 kHz .....	41



Obr. 6.4 : Závislost nejistoty typu A na vzdálenosti senzoru PVDF 80 kHz .....	41
Obr. 7.1 Schématické znázornění přípravku pro měření 2D polohy .....	43
Obr. 7.2 : Časový průběh ultrazvukového přijímače na osciloskopu .....	45
Obr. 7.3 : Statické měření bez filtrace hodnot .....	50
Obr. 7.4 : Statické měření s použitím filtru typu klouzavý průměr.....	51
Obr. 7.5 : Frekvenční charakteristiky filtrů typu dolní propust. (a) Butterworthův filtr; (b) I typ Čebyševova filtru; (c) II typ Čebyševova filtru; (d) eliptický filtr [14] .....	52
Obr. 7.6 : Statické měření s použitím Čebyševova filtru .....	53
Obr. 7.7 : Statické měření s použitím Butterworthova filtru.....	54
Obr. 7.8 : Vyregulování souboru hodnot jedné statické vzdálenost.....	55
Obr. 7.9 : Nakreslená střecha bez filtrace, se zobrazením skutečné předlohy .....	55
Obr. 7.10 : Nakreslená střecha po použití klouzavého průměru .....	56
Obr. 7.11 : Nakreslená střecha po vyfiltrování hodnot Butterworthovým filtrem.....	56
Obr. 7.12 : Nakreslená střecha po vyfiltrování hodnot Čebyševovým filtrem .....	57
Obr. 7.13 : Dlouhý nástup při špatně nastaveném filtru .....	57
Obr. 7.14 : Špatně nastavený filtr (Butterworthův) – nastavené velice malé frekvence .....	58
Obr. 7.15 : Srovnání použitých filtrů s předlohou .....	58

# SEZNAM TABULEK

Tab. 1 : Rychlosti ultrazvuku v různých prostředí [17] .....	12
Tab. 2 : Maximální hodnoty vypočítaných chyb, nejistot a rozptylu .....	60
Tab. 3 : Průměrné hodnoty vypočítaných chyb, nejistot a rozptylu .....	60
Tab. 4 : Tabulka vypočtených hodnot na senzoru US-100 .....	1
Tab. 5 : Tabulka vypočtených hodnot na senzoru US-100 při vychýlení +15° .....	2
Tab. 6 : Tabulka vypočtených hodnot na senzoru US-100 při vychýlení -15° .....	2
Tab. 7 : Tabulka vypočtených hodnot na senzoru SRF08 .....	3
Tab. 8 : Tabulka vypočtených hodnot na senzoru SRF08 při vychýlení +15° .....	4
Tab. 9 : Tabulka vypočtených hodnot na senzoru SRF08 při vychýlení -15° .....	4
Tab. 10 : Tabulka vypočtených hodnot na senzoru PVDF 80 kHz .....	5

# ÚVOD

Ultrazvuk se velmi hojně používá v průmyslových aplikacích pro zjišťování přítomnosti objektů, měření vzdálenosti objektů (překážek), nebo třeba také měření výšky hladiny kapalin. Dále se ultrazvuk např. používá v lékařství, kde dokážeme pomocí ultrazvuku nahlédnout do nitra lidského těla, ale také i např. tišit bolest lidského svalstva.

V této práci bude postupně vysvětleno měření se směrovými a všesměrovými ultrazvukovými senzory; a to měření metodou TOF se dvěma ultrazvukovými měniči; metodou ultrazvuková závora a také měření metodou trilaterace se dvěma přijímači a jedním vysílačem (což je základ pro měření 2D polohy se všesměrovými ultrazvukovými senzory). Práce obsahuje zapojení vysílače i přijímače pro válcový 80 kHz ultrazvukový převodník, který jsem v této práci použil, a samotné propojení senzorů s mikrokontrolérem. Tato práce se také zabývá ukázkou a vysvětlení postupu měření 2D polohy ultrazvukových snímačů, výpočtu x-ových a y-ových souřadnic trilaterací a zobrazení trajektorie ultrazvukového vysílače v PC pomocí Matlabu.

Jsou zde změřeny a zobrazeny statické charakteristiky v různých vzdálenostech a ověřeny dynamické vlastnosti nakreslením předlohy. Změřená podoba trajektorie srovnaná s předlohou a poté zhodnocena před a po použití digitálního filtru. Také je zde provedeno srovnání vlastností a použitelnosti směrových a všesměrových ultrazvukových senzorů.

# 1 ULTRAZVUK

Ultrazvuk je akustické (mechanické) vlnění částic s vyšší frekvencí než je slyšitelná hranice u lidí, tudíž větší jak 20 kHz. Ultrazvuk se šíří prostorem pomocí vibrací částic kolem svých rovnovážných poloh. Částice jsou vázány elastickými silami, tudíž vibrace částice se přenášejí jen na sousední částice a tím vzniká šíření ultrazvukových vln prostorem. Ultrazvuk se od slyšitelného zvuku fyzikálními vlastnostmi nijak neliší, jen v tom, že ho lidé nemohou slyšet. [17]

## 1.1 Vznik ultrazvuku

Ultrazvukovou vlnu získáme periodickým vybuzením vhodného materiálu (např. křemík, křišťál). Dochází zde ke vzniku tzv. obráceného piezoelektrického jevu. Obrácený piezoelektrický jev vytvoříme tak, že krystal vložíme do elektrického pole a přivádíme na stěny krystalu střídavé napětí, tím se krystal začne periodicky zkracovat a prodlužovat - rozkmitá se. Tyto změny rozměrů krystalu mají za následek vznik zvuku, resp. ultrazvuku. [18]

## 1.2 Rychlost šíření ultrazvuku

Rychlost šíření je závislá na více faktorech, závisí hlavně na typu prostředí ve kterém se ultrazvuk šíří (čím hustší prostředí, tím rychleji se ultrazvuk šíří) a závisí také na teplotě daného prostředí. V tab. 3.1 je zobrazeno porovnání šíření rychlostí ultrazvuku v různých prostředí.

Prostředí	v [m/s]
Vzduch (0°C)	331,4
Vzduch (20°C)	344
Měď	3500
Ocel	5000
Helium	927
Voda	1500

Tab. 1 : Rychlosti ultrazvuku v různých prostředí [17]

Vztah pro výpočet rychlosti šíření ultrazvuku ve vzduchu při teplotě  $\tau$  [°C] :

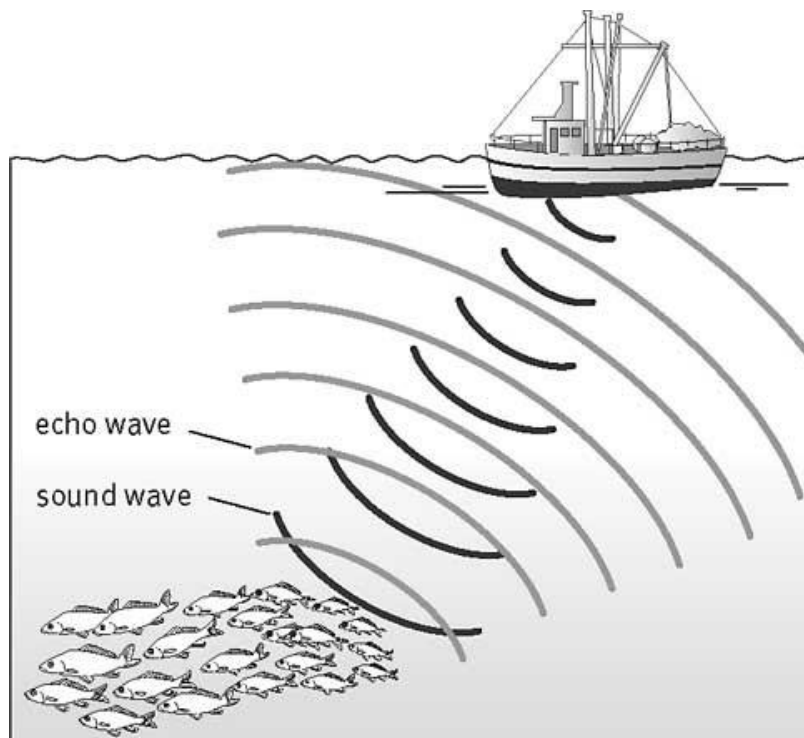
$$v = (331,4 + 0,6 * \tau) [m/s] \quad (1.1)$$

### 1.3 Využití ultrazvuku

Ultrazvuk používají netopýři k navigaci v prostoru (echolokace) - pomocí odrazů ultrazvuku od překážek se dokáží orientovat v prostoru a vyhýbat se překážek.

Ultrazvuk se používá v mnoha oborech :

- V technice – měření vzdálenosti, průtoku, rychlosti, mapování oblastí, hledání vad v materiálech (defektoskopie), čištění materiálu, měření tloušťky materiálu
  - polohování; pomocí více ultrazvukových senzorů dokážeme vypočítat souřadnice překážky nebo vysílače a jeho polohu zobrazit nebo s ní nadále pracovat
- V lékařství – ultrasonografie (ultrazvuk se odráží jinak od tkáně, než od kosti) – lze zobrazit vnitřek těla
  - hlubkový tepelný účinek, uvolnění svalového napětí, tišení bolesti
- V námořnictví – Sonar (zjištění předmětů ve vodě – např. ryb, lodních vraků)
  - Echolot (měření hloubky)

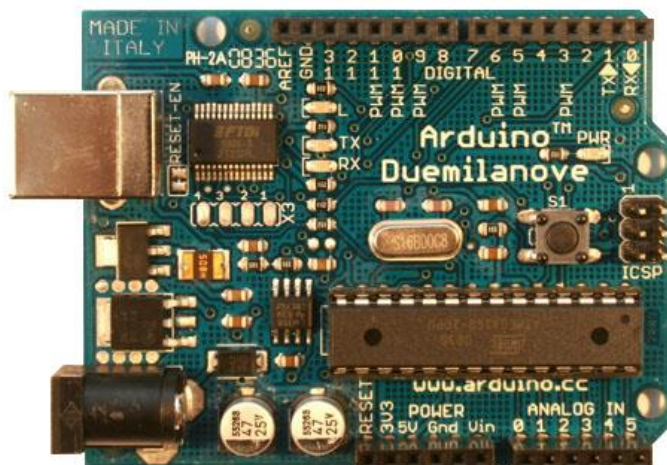


Obr. 1.1 : Sonar [12]

## 2 ARDUINO

### 2.1 Arduino Duemilanove

V této práci jsem použil mikrokontrolér Arduino Duemilanove (Obr. 2.1) s procesorem ATmega328 (datasheet dostupný na [6]) od firmy Atmel. Mikroprocesor pracuje na frekvenci 16 MHz a má 14 digitálních vstupních/výstupních pinů, z toho 6 jich může být použito jako softwarově řízené PWM výstupy a 6 analogových vstupů. Arduino lze propojit s počítačem pomocí sběrnice USB, jelikož deska obsahuje bootloader (který se postará o základní nastavení), lze mikrokontrolér programovat přímo přes USB (program je psaný v programovém prostředí Arduino popsany v kap. 2.2). I když je deska připojena pomocí USB sběrnice, komunikace je softwarově simulována jako komunikace přes RS-232. Více na [7].



Obr. 2.1 : Arduino Duemilanove [7]

### 2.2 Programové prostředí Arduino

Zdarma stažitelné Open-Source prostředí [5] umožňuje snadno psát a nahrávat program do mikroprocesoru na desce Arduino (v našem případě Arduino Duemilanove). Programovací jazyk Arduino je založen na jazyku C/C++.

Program je rozdělen do tří hlavních částí : struktury, proměnné a funkce. Struktury jsou *setup()* a *loop()*, před strukturu *setup()* deklarujeme proměnné, uvnitř *setup()* provádíme např. inicializaci proměnných, režim použitých pinů nebo nastavení sériové komunikace. Struktura *loop()* pracuje jako nekonečná smyčka, slouží tedy k psaní vlastního programu a pro aktivní řízení Arduino desky.

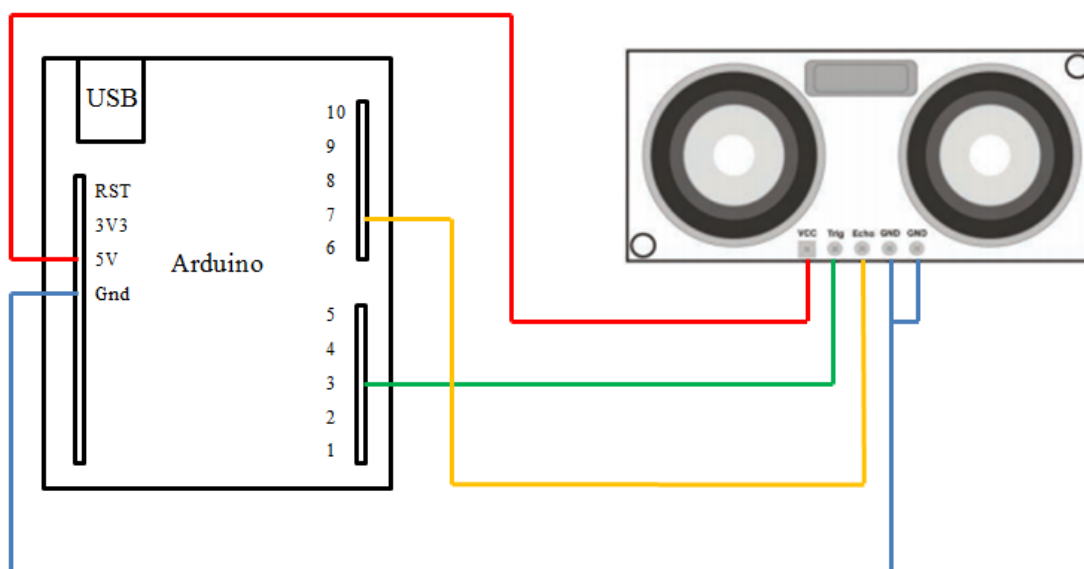
## 3 POUŽITÉ ULTRAZVUKOVÉ SENZORY

### 3.1 US-100

US-100 je ultrazvukový senzor používaný k měření krátkých vzdáleností, obsahuje obvody pro vyhodnocování změřené vzdálenosti. Senzor pracuje na frekvenci 40 kHz a pracovní oblast měření je od 2 cm do 350 cm (450 cm). Přesnost až 1 mm, zorný úhel 15° (změřená vyzařovací charakteristika na Obr. 6.5). Datasheet je k dispozici na [2].

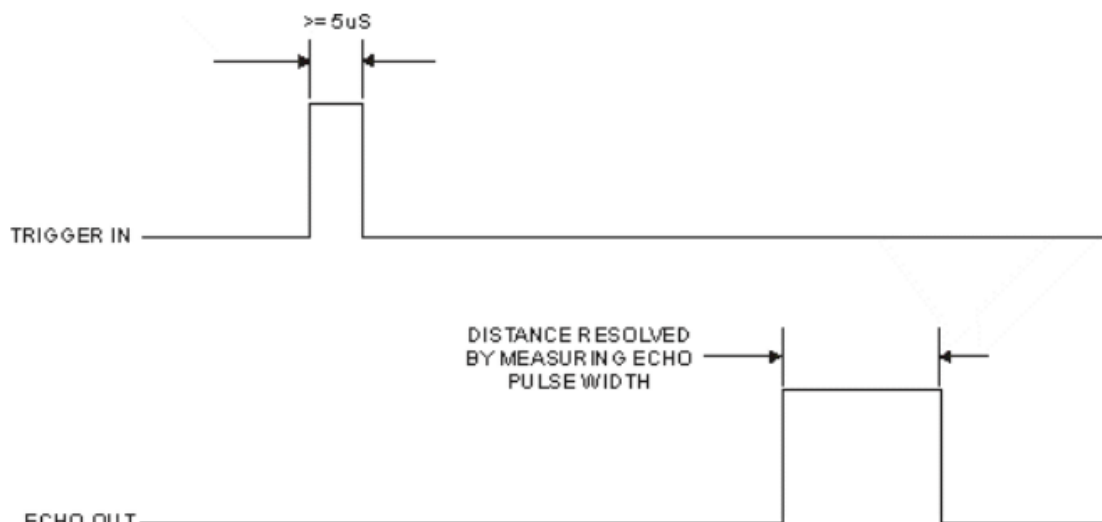


Obr. 3.1 : Ultrazvukový senzor US-100 [1]



Obr. 3.2 : Připojení senzoru US-100 k mikrokontroléru [1]

Senzor se připojuje k mikrokontroléru pěti konektory (viz. Obr. 3.2); VCC slouží k připojení napájení (3 V - 5 VDC), TRIGGER slouží k připojení řídicího pinu - po uvedení tohoto pinu do stavu *logic HIGH* senzor vyšle ultrazvukovou vlnu do prostoru, pokud senzor tuto vlnu po odrazu od některé překážky detekuje, senzor vyhodnotí čas od vyslání vlny do příchodu a šířka pulsu na výstupním konektoru ECHO odpovídá změřenému času (viz. Obr. 3.3). Dále pak dva konektory GND pro připojení země.



Obr. 3.3 : Časový diagram senzoru US-100 [1]

Vzdálenost lze tedy vypočítat následujícím způsobem :

$$D = \frac{d * v}{2} \quad (3.1)$$

Kde : D...vzdálenost

d...šířka pulsu [s]

v...rychlost zvuku (1.1)

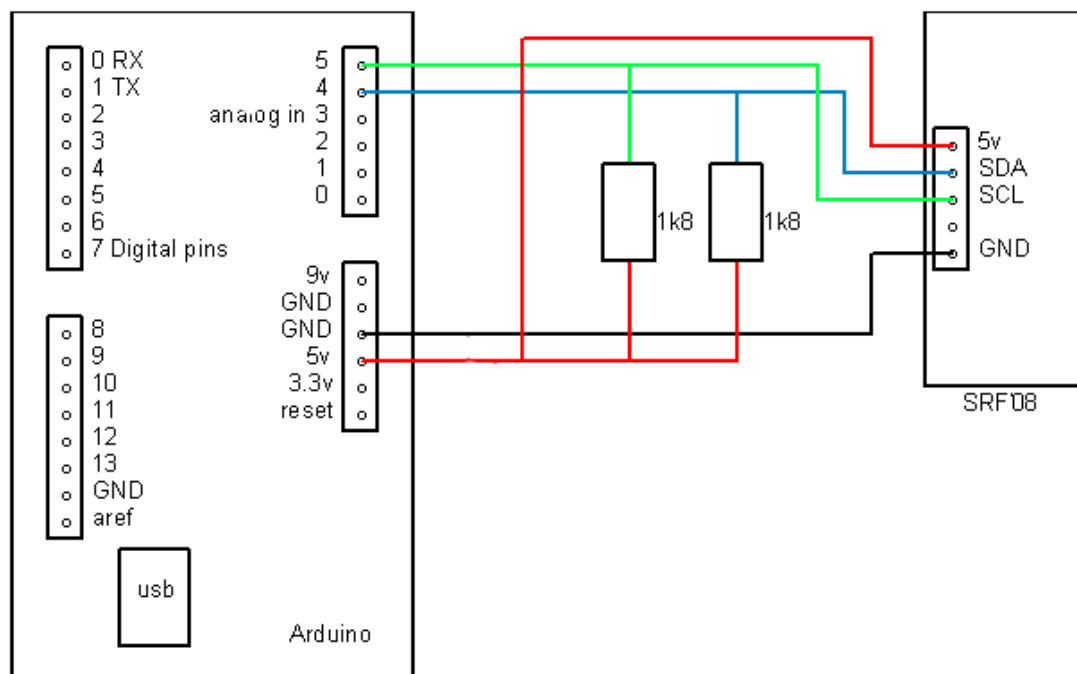
## 3.2 SRF08

Ultrazvukový senzor SRF08 lze použít pro měření vzdáleností od 3 cm do 11 m, pracuje na frekvenci 40 kHz a uživatelem může být nastaveno až 16 adres pro ovládání senzoru. Pomocí nastavitelných registrů a příkazů lze zvolit např. zobrazení v různých jednotkách (palce, cm, ms), nastavení módů měření (ANN nebo měření vzdálenosti) atd. Senzor také dokáže vyhodnotit vícenásobné echo v nastavitelných rozsazích. Výše zmíněný mód ANN (Artificial Neural Network) je určen pro získání a zpracování vícenásobného echa neuronovou sítí.





**Obr. 3.4 :** Ultrazvukový senzor SRF08 [13]



**Obr. 3.5 :** Schéma zapojení senzoru SRF08 [4]

### 3.3 Válcový ultrazvukový převodník 80 kHz

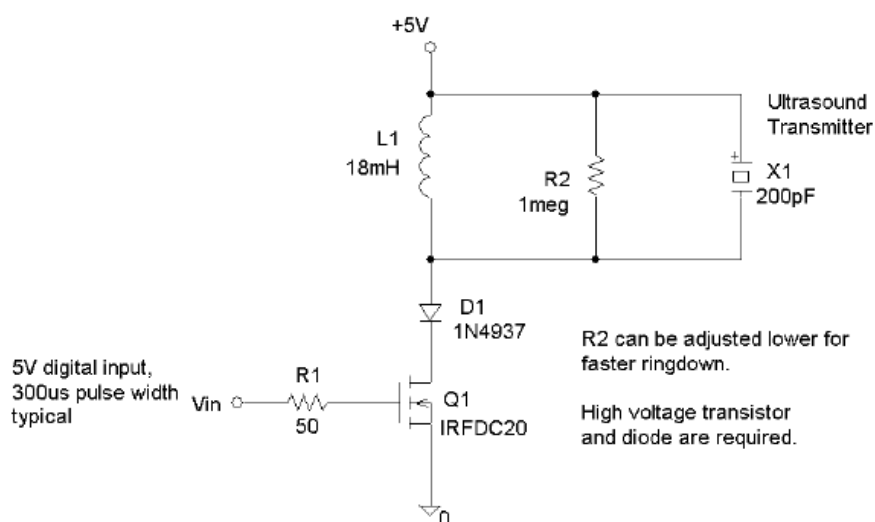
Válcový, všesměrový piezoelektrický ultrazvukový senzor pracující na frekvenci 80 kHz má využití v aplikacích kde je potřeba měnit vzájemný horizontální úhel mezi vysílačem a přijímačem. Výhodou je, jak už bylo řečeno horizontální směrovost, dále velká šířka pásma, nízká rezonanční „Q“ hodnota (nárůst a útlum signálu je mnohem kratší než u běžných keramických ultrazvukových měničů) [14]. Díky těmto vlastnostem se tento snímač dobře hodí pro polohovací aplikace, detekci objektů, digitizéry a další. Tento ultrazvukový převodník je tvořen pouze piezofilmem na nosníku a jeho vývody (Obr. 3.6), je tudíž potřeba k funkčnosti převodníku zapojit potřebnou elektroniku. Elektroniku jsem zapojil podle následujících schémat doporučených od výrobce tohoto převodníku.



Obr. 3.6 : Válcový 80 kHz ultrazvukový převodník [14]

### 3.3.1 Schéma vysílače

Na schématu (Obr. 3.7) lze vidět, že je zapotřebí použít vysokonapěťový tranzistor a diodu a to kvůli vysokému napětí, které je potřeba pro vybuzení piezoelektrického elementu. Funkce tohoto schématu spočívá v přivedení řídicího signálu na pin „Vin“ který přes odpor „R1“ řídí otevření tranzistoru „Q1“. Při otevření tohoto tranzistoru (typicky přivedením impulsu o velikosti 5V a šířce pulzu 300  $\mu$ s) je na induktoru „L1“ napětí a induktor se nabíjí po dobu otevření tranzistoru, po odpojení napětí, tudíž zavření tranzistoru „Q1“ vznikne na induktoru „L1“ a piezoelementu „X1“ napěťový ráz (o velikosti až 400 V), který mechanicky rozkmitá piezoelement a tím vznikne ultrazvuková vlna.



Obr. 3.7 : Schéma 80kHz ultrazvukového PVDF vysílače [14]



Ve schématu (Obr. 3.8) jsou tři operační zesilovače, kde první dva fungují jako zesilovače a poslední OZ jako komparátor. Tento komparátor udržuje na výstupu „TP1“ plné napájecí napětí do té doby, než od zesilovače přijde dostatečně velký zesílený signál od ultrazvukového přijímače. Pokud se tak stane, komparátor stáhne napětí na pinu „TP1“ k nule, a to po dobu odpovídající délce přijaté ultrazvukové půlvlny. Jelikož tento výstup je digitální, lze ho dobře zpracovat v mikrokontroléru. Nahrazením odporů „R7“ a „R9“ za odporové trimry docílíme možnosti nastavování citlivosti vyhodnocovacího obvodu.

## 4 POUŽÍVANÉ METODY MĚŘENÍ ULTRAZVUKŮ

Typ použité metody se odvíjí od potřebné přesnosti, typu materiálu sledovaného objektu a zda používáme Dvojhlavý nebo Jednohlavý systém (viz popsáno níže). Např. Dvojhlavý systém potřebuje další napájení pro vysílač, který je přidělán na sledovaném objektu (tudíž nezáleží na materiálu objektu), avšak přesnost je větší než u Jednohlavého systému. U Jednohlavého systému velice znehodnocuje přesnost měření nerovnost sledované plochy.

### 4.1 Měření doby letu (TOF) – 1. Metoda

Pro použití této metody je potřeba tzv. Dvojhlavý systém, tzn. že vysílač i přijímač má samostatný ultrazvukový měnič (jsou samostatně zapouzďeny) a zjišťujeme jejich vzdálenost [8]. Metoda měření je založena na měření času od doby kdy jsme vyslali ultrazvukovou vlnu (vybudili piezoelektrický element) z vysílače, do přijmutí této vlny přijímačem. Ze zjištěného časového rozdílu můžeme vypočítat vzdálenost objektů (s přesností, jakou jsme schopni měřit čas doby letu) jednoduchým přepočtem (4.1). Výhodou je větší přesnost a rozsah, nevýhodou je potřeba přimontovat ke sledovanému předmětu vysílač a s tím související potřeba další řídicí elektroniky a tudíž vyšší cena.

$$d = (331,4 + 0,6 * \tau) * t \quad [\text{m/s}] \quad (4.1)$$

### 4.2 Měření doby letu (TOF) – 2. Metoda

Tato metoda je založena na stejném principu jako 1. metoda, jediný rozdíl je v tom, že k tomuto měření využíváme tzv. Jednohlavý systém - vysílač i přijímač používají stejný ultrazvukový měnič a jsou zapouzďeny v jednom celku. Jak už název napovídá, jedná se o metodu, kde měříme čas mezi vysláním a přijmutím ultrazvukové vlny, ale získanou dobu letu musíme vydělit dvěma, protože dráha letu je dvojnásobná než reálná vzdálenost (vlna letí k odrazovému předmětu a zpátky k přijímači). Výhodou tohoto řešení je možnost měření vzdálenosti od objektů, které mají potřebné vlastnosti k odrazení zvukové vlny (neredukují odraz zvuku). Např. molitan, bavlněná ucpávka mají vysoké zvukově izolační vlastnosti a neodrážejí zvukovou vlnu.

### 4.3 Ultrazvuková závora

Principiálně je tato metoda stejná jako optická závora; světlo projde/neprojde. Používá se pro zjišťování přítomnosti předmětu. Tato metoda je nejjednodušší z hlediska náročnosti na vyhodnocování výstupu. Výstupem může být log 1 nebo log 0. Pokud používáme Dvojhlavý systém, je mezi vysílačem a přijímačem předem daná vzdálenost a ultrazvuková vlna vyslaná vysílačem je detekována přijímačem (log 1). Pokud je ale mezi přijímačem a vysílačem vložena překážka, ultrazvuková vlna se odrazí od překážky a nedoputuje k přijímači – vyhodnocena log 0, i když se vlna dostane k přijímači díky sekundárním odrazům (vícenásobné echo), je to také vyhodnoceno jako log 0.

U Jednohlavého systému tato metoda funguje na principu odrazu ultrazvukové vlny od překážky. Ultrazvuková vlna se odrazí od předmětu, pokud předmět přiblížíme do určité vzdálenosti od senzoru, nebo do osy vyzařování (viz. kap 5.1.3 Měření vyzařovací charakteristiky) – vyhodnocení log 1, jinak v normálním stavu log 0.



Obr. 4.1 : Jednohlavý systém [8]

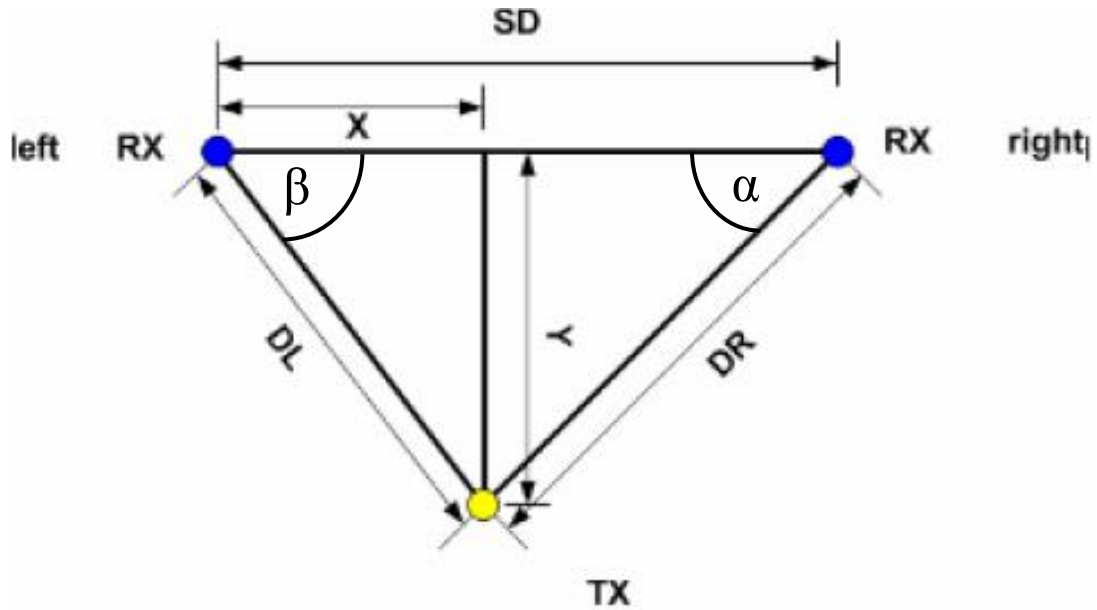


Obr. 4.2 : Dvojhlavý systém (ultrazvuková závora) [8]

### 4.4 Trilaterace

Pro tento typ měření jsou potřeba všesměrové ultrazvuky, důvod je ten, že při tomto měření se mění vzájemné úhly mezi vysílači a přijímačem, tato vlastnost se právě využívá pro výpočet souřadnic 2D polohy. K určení polohy ve dvourozměrném prostoru potřebujeme alespoň dva referenční body (statické přijímače) a jeden pohyblivý vysílač. Mezi přijímači musí být přesně stanovená vzdálenost (SD), tudíž pokud známe tuto vzdálenost a jsme schopni změřit vzdálenost od každého přijímače k vysílači (DR a

DL), dokážeme z těchto informací vypočítat pomocí Kosinovy a Pythagorovy věty x-ové a y-ové souřadnice (viz. obr. 4.3). Vztahy 4.2 a 4.3 jsou obecné vztahy Kosinovy věty pro výpočet úhlů u vrcholů RX. Vztahy pro výpočet x-ové a y-ové souřadnice jsou uvedeny ve vztazích 4.8, 4.9 a 4.10.



Obr. 4.3 : Trilaterace [9]

$$DR^2 = SD^2 + DL^2 - 2 * SD * DL * \cos(\beta) \quad (4.2)$$

$$DL^2 = DR^2 + SD^2 - 2 * DR * SD * \cos(\alpha) \quad (4.3)$$

$$\frac{-DR^2 + SD^2 + DL^2}{2 * SD * DL} = \cos(\beta) \quad (4.4)$$

$$\frac{-DL^2 + DR^2 + SD^2}{2 * DR * SD} = \cos(\alpha) \quad (4.5)$$

V tuto chvíli dokážeme vypočítat úhly jednoduše tím, že vynásobíme celou rovnici funkcí  $\arccos()$ ; na pravé straně dostaneme daný úhel  $\alpha$  nebo  $\beta$  ( $\cos()$  a  $\arccos()$  se vykrátí), na levou stranu použijeme také funkci  $\arccos()$  a výsledný úhel tedy bude rovný  $\arccos()$  levé strany. Viz. vztahy 4.6 a 4.7.

$$\cos^{-1}\left(\frac{-DR^2 + SD^2 + DL^2}{2 * SD * DL}\right) = \beta \quad (4.6)$$

$$\cos^{-1}\left(\frac{-DL^2 + DR^2 + SD^2}{2 * DR * SD}\right) = \alpha \quad (4.7)$$

V tuto chvíli již můžeme dopočítat x-ovou a y-ovou souřadnici:

$$Y = \sin(\beta) * DL \quad (4.8)$$

$$X = \sqrt{DR^2 - Y^2} \quad (4.9)$$

Jelikož jsem zvolil počátek souřadných os na pozici levého přijímače, je potřeba zajistit přepočet x-ové souřadnice, když se vysílač nachází nalevo od levého přijímače (záporná x-ová souřadnice), viz vztah (4.10). Y-ovou souřadnici počítáme stejně jako v předchozím případě.

$$X = SD - \sqrt{DL^2 - Y^2} \quad (4.10)$$



## 5 MĚŘENÍ SMĚROVÝCH ULTRAZVUKŮ

Při měření směrových ultrazvukových senzorů se hodí počítat standardně počítané chyby měření, a to jsou; rozptyl, standardní nejistota typu A a absolutní chyba měření. Pro sestavení charakteristik snímačů jsem zvolil výpočet středních hodnot; a to standardním průměrem a nebo mediánem, viz charakteristiky snímačů.

Rozptyl - udává míru odchylování veličiny od průměru. Označení pomocí písmene D pochází z anglického slova *dispersion* (rozptyl). DX veličiny X definujeme jako střední hodnotu čtverce odchylky veličiny X od své střední hodnoty EX [10]:

$$DX = E(X - EX)^2 \quad (5.1)$$

Standardní nejistota typu A - statistický údaj o přesnosti měření, nebo-li kvadratický průměr odchylek od aritmetického průměru ( $\bar{x}$ ), určuje se z opakovaných měření. Vypovídá o tom, v jakém intervalu by se měla nacházet skutečná hodnota. Mírou nejistoty typu A je výběrová směrodatná odchylka výběrového průměru [16].

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$
$$u(a) = \sqrt{\frac{1}{n(n-1)} \sum_{i=1}^n (x_i - \bar{x})^2}$$

Absolutní chyba - rozdíl průměru (resp. mediánu) souboru naměřených hodnot a nastavované vzdálenosti. Standardní chyba počítána v měření – rozdíl naměřené a skutečné hodnoty.

$$\delta_r = \frac{1}{n} \sum_{i=1}^n x_i - x_{skut} \quad (5.2)$$

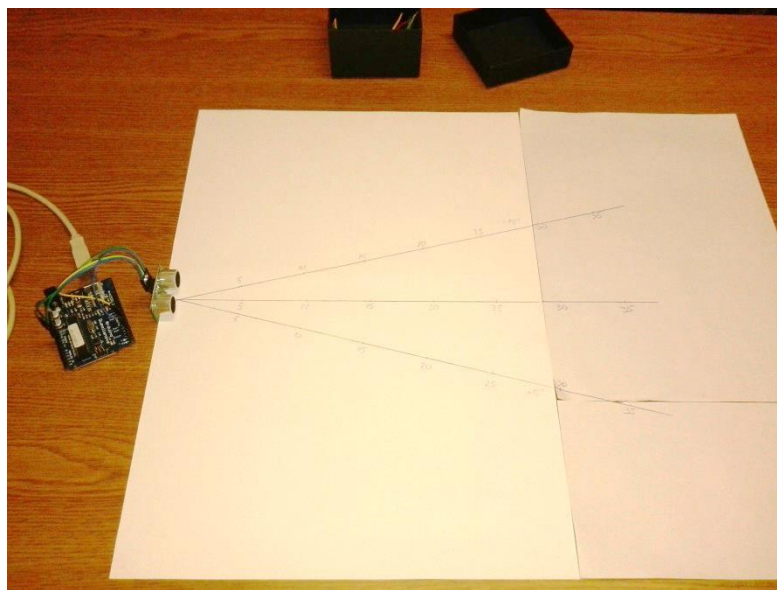
Medián - Medián je hodnota, kterou získáme tím, že seřadíme změřenou sérii vzestupně, a ta hodnota, která rozděluje tuto řadu na dvě stejné poloviny označujeme medián.

## 5.1 Postup měření

### 5.1.1 Měření senzorem US-100

Tento ultrazvukový musíme měřit 2. metodou TOF, protože tento senzor je typu „Jednohlavý systém“. Senzor připojíme k mikrokontroléru podle schématu (Obr. 3.2), mikrokontrolér je připojen pomocí sériové sběrnice USB s počítačem, abychom mohli naměřená data posílat do PC pro následné zpracování, popř. zobrazení v „Serial Monitoru“. Získané hodnoty jsou zpracovány v Matlabu (Kap. 5.2).

Měřicí stanoviště pro měření charakteristik senzoru je zobrazeno na Obr. 5.1. Do zvolené vzdálenosti od senzoru položíme překážku (měřený objekt), do konzole sériové linky zadáme skutečnou vzdálenost a spustíme měření příkazem „s“. Algoritmus nahraný v mikrokontroléru spustí sérii 1000 měření, naměřené hodnoty jsou posílány přes sériovou linku do PC. V počítači jsou naměřené hodnoty zapisovány do textového souboru pomocí programu RealTerm. Pro přehlednost, identifikaci a hlavně pro následné zpracování v Matlabu je každá série naměřených hodnot ohraničena hranatými závorkami a je jim přiřazen název, který udává skutečnou vzdálenost (hodnota, která byla zapsána do konzole při měření). Posouváním překážky v rozsahu od 3 cm (omezeno vlastnostmi snímače) až do 35 cm (zvolená pracovní oblast) a následným měřením získáme soubor hodnot ze kterých můžeme vynést závislost naměřené vzdálenosti na skutečné.



Obr. 5.1 : Měřicí stanoviště

### 5.1.2 Měření senzorem SRF08

Tento senzor se měří principiálně stejně jako US-100, liší se jen zapojením do mikrokontroléru (schéma na Obr. 3.5) a řídicím programem (algoritmem), viz předchozí kapitola.

### 5.1.3 Měření vyzařovací charakteristiky

V ultrazvukové senzorové technice je potřeba také změřit vyzařovací charakteristiku snímače, i když tyto senzory jsou směrové, mají nějaký mezní úhel (od osy snímače) při kterém jsou ještě schopny identifikovat překážku. To je způsobeno fyzikální podstatou šíření zvuku – zvuk se šíří jako vlna.

Nejdříve zvolíme několik bodů s různou vzdáleností od snímače. Měření provádíme takovým způsobem, že spustíme měření v otevřené smyčce (tzn. že měříme bez limitu, nebo-li do té doby než ručně ukončíme sekvenci měření) a naměřené hodnoty zobrazujeme na obrazovce počítače pomocí „Serial Monitor“. Zvolené body pro oblast měření posouváme od osy snímače a kontrolujeme, zda je překážka ještě detekována (měřené hodnoty se mění v malém rozmezí), pokud překročíme mezní úhel, měřené hodnoty se mnohonásobně zvětší (záleží na další nejbližší překážce v cestě ultrazvukové vlny). Souřadnice tohoto bodu si zapíšeme a pokračujeme stejným způsobem pro všechny body. Ze získaných bodů sestojíme vyzařovací charakteristiku (Obr. 6.5 a 6.13).

## 5.2 Získání a zpracování naměřených hodnot

### 5.2.1 Základní nastavení

V této práci jsem použil vývojový kit Arduino Duemilanove s procesorem ATmega 328. Senzor připojíme k mikrokontroléru podle schématu (Obr. 3.2). Piny „trigger“ a „echo“ musíme nadefinovat v programu následujícím způsobem: „*int trigger = 3*“; tato definice znamená, že jsme trigger připojili na desce do pinu číslo 3. Aby tato definice byla úplná, musíme do bloku „*setup()*“ specifikovat, zda je nadefinovaný pin výstupní nebo vstupní: např. „*pinMode(trigger,OUTPUT)*“.

Po nejdůležitějším kroku, což je nadefinování pinů můžeme postoupit k psaní vlastního programu, který píšeme do bloku „*loop()*“, potřebné proměnné definujeme ve stejné oblasti jako definování pinů (před blokem „*setup()*“). Program napsaný do bloku „*loop()*“ se provádí v „nekonečné smyčce“, pokaždé když program dojde ke konci bloku „*loop()*“, začne se tento blok znovu provádět (skočí na začátek). Tato část

programu se vykonává nekonečně mnohokrát, nebo-li do té doby než přijde nějaká externí událost (např. odpojení napětí, nebo ukončení sériové komunikace).

### 5.2.2 Zpracování signálu ze senzoru US-100

Po nastavení správných pinů v programu můžeme přejít k psaní kódu, který bude obsluhovat ultrazvukový snímač a vyhodnocovat naměřené hodnoty. Z Obr. 3.3 je jasné, že pro zajištění vyslání ultrazvukové vlny musí být výstupní pin „trigger“ zapnutý, nebo-li ve stavu HIGH alespoň po dobu 5  $\mu$ s. Vyslání vlny je realizováno pomocí příkazy „*delayMicroseconds()*“ a „*digitalWrite()*“; nejdříve nastavíme výstup „trigger“ na log 1 („*digitalWrite(trigger,HIGH);*“) a následným zpožděním o délce 5  $\mu$ s zajistíme vyslání ultrazvukové vlny, nakonec uvedeme výstup trigger zpět na hodnotu log 0 příkazem „*digitalWrite(trigger,LOW)*“.

Změřenou vzdálenost zjistíme ze šířky přijaté vlny (jak lze vidět na Obr. 3.3) a vynásobíme ji s rychlostí šíření zvuku. Šířku impulsu lze jednoduše zjistit pomocí příkazu „*pulseIn()*“. Hodnotu šířky impulsu tedy uložíme do proměnné „*cas*“ příkazem „*pulseIn(echo,HIGH)*“. Hodnota uložená v proměnné „*cas*“ je v  $\mu$ s, a převodem přes rychlost (5.2) vypočítáme vzdálenost v metrech (popř. centimetrech, jak je zvoleno v této práci). Výslednou vzdálenost pošleme (vytiskneme na obrazovku) přes sériovou linku příkazem *Serial.println(vzdalenost,10);*.

Do programu jsou implementovány funkce, které zautomatizují a zjednoduší měření; např. zadáním dvou-číselné hodnoty do příkazové konzole v „Serial Monitoru“ zapíšeme hodnotu měřené vzdálenosti, poté zadáním písmene „s“ do konzole sériové linky spustíme sérii měření (nastaveno na 1000). Série měření je ohraničena hranatými závorkami a před těmito závorkami je u písmene „x“ zapsaná hodnota, kterou jsme zadali (výsledný formát je „x00 = [...]“), tento zápis usnadňuje následnou práci s hodnotami, a zpracování v Matlabu.

$$rychlost = (331,4 + 0,6 * tepl) \quad (5.3)$$

$$vzdalenost = \frac{rychlost * cas}{10^4 * 2} \quad (5.4)$$

Kde  $10^4$  je konstanta pro převod na cm, výslednou vzdálenost vydělíme 2, protože vlna putuje dvojnásobnou vzdálenost

### 5.2.3 Zpracování signálu ze senzoru SRF08

Pro měření tohoto senzoru jsem použil knihovny z [4], které zajišťují obsluhu senzoru (vysílání ultrazvukové vlny a nastavování registrů). Funkce pro obsluhu měření a formátování výstupních hodnot jsou stejné jako při měření senzoru US-100.

Pomocí příkazů (commands) [4] lze měnit zobrazování výsledků ve třech jednotkách, a to v centimetrech, milisekundách a palcích. Měření v centimetrech je pro tuto úlohu bezpředmětné a nepřesné, proto jsem zvolil zobrazování v milisekundách a následným přepočítáním na centimetry pomocí stejného přepočtu, jako při měření senzoru US-100 (5.3) jsem získal přesnější hodnoty (v závislosti na přesnosti interního čítače).

## 5.3 Zpracování naměřených hodnot v MATLABu

Změřené hodnoty jsou již upravené pro práci v Matlabu (viz. kap. 5.2), můžeme tedy textový soubor, v kterém máme uložené hodnoty měření, přímo otevřít v Matlabu. Soubor znovu uložíme, ale v této chvíli už jako script (\*.m), a můžeme ho spustit. Pokud jsme postupovali dobře, ve „Workspace“ se zobrazí proměnné, kde každá proměnná představuje sérii měření. S těmito proměnnými můžeme nadále provádět potřebné výpočty a zobrazovat jejich závislosti.

Vytvořil jsem funkci, ve které jsem nadefinoval výpočet statistických údajů, které se při měření obvykle zjišťují, jsou to : standardní nejistota typu A, rozptyl a absolutní chyba [10]. Tato funkce ( $[rozptyl,chyba,nejistota] = fce(a,b)$ ) má dva vstupy a tři výstupy. Vstup „a“ představuje proměnnou, v níž je uložena jedna série měření, vstup „b“ je referenční hodnota (skutečná nastavená vzdálenost), tuto hodnotu lehce zjistíme z jména proměnné. Výstupy jsou vypočítané hodnoty.

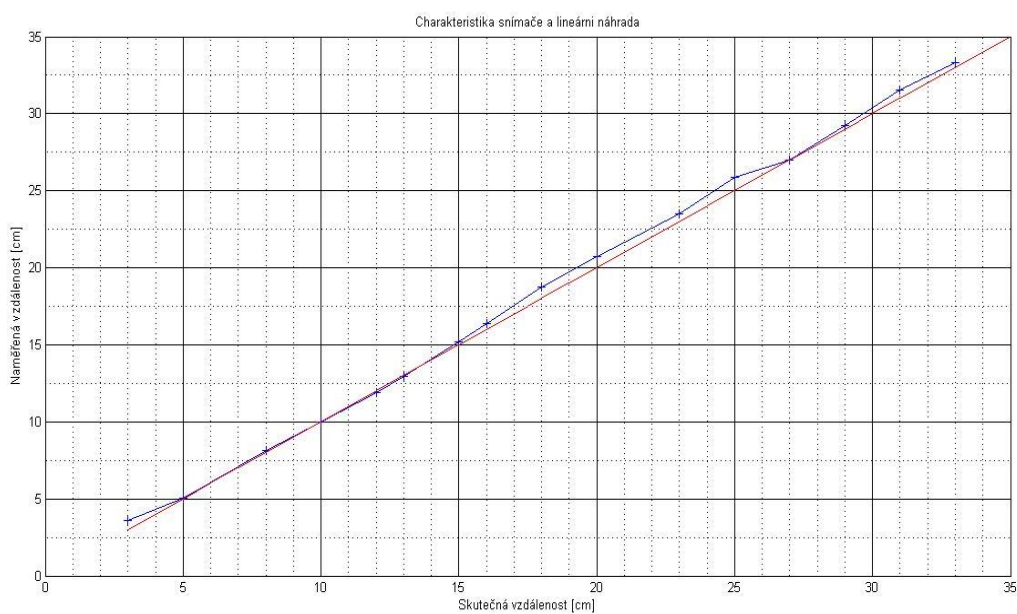
V hlavním skriptu postupně necháme touto funkcí vypočítat všechny proměnné, které jsme získali měřením. Vypočtené hodnoty jsou ukládány do nových proměnných (vektorů), tak aby v každém vektoru byly jen nejistoty, chyby atd. Dále vytvoříme vektor, ve kterém budou hodnoty skutečných vzdáleností, tento vektor použijeme pro vytváření grafů, a to hlavně pro vytvoření charakteristik senzorů.

## 5.4 Vlastní měření a grafy

### 5.4.1 Měření senzoru US-100

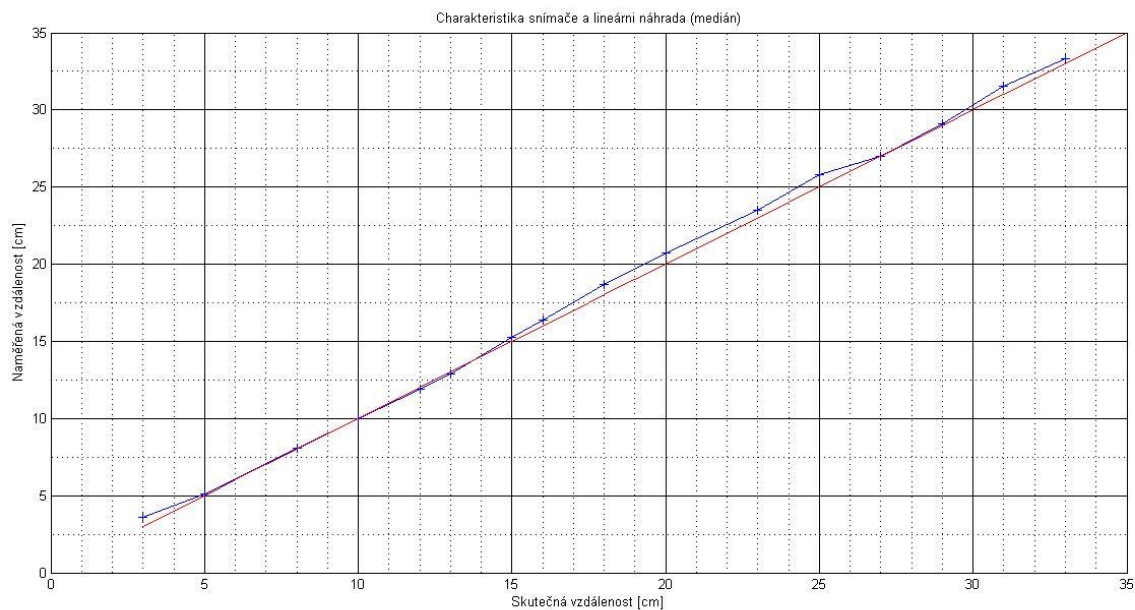
Měření vzdálenosti a porovnání s lineární náhradou. Při tomto grafu jsem použil jako změřenou vzdálenost průměr z naměřené série hodnot.

- modrý průběh - body představují průměry naměřených hodnot
- červený průběh - lineární náhrada (hodnoty které bych měl získat)



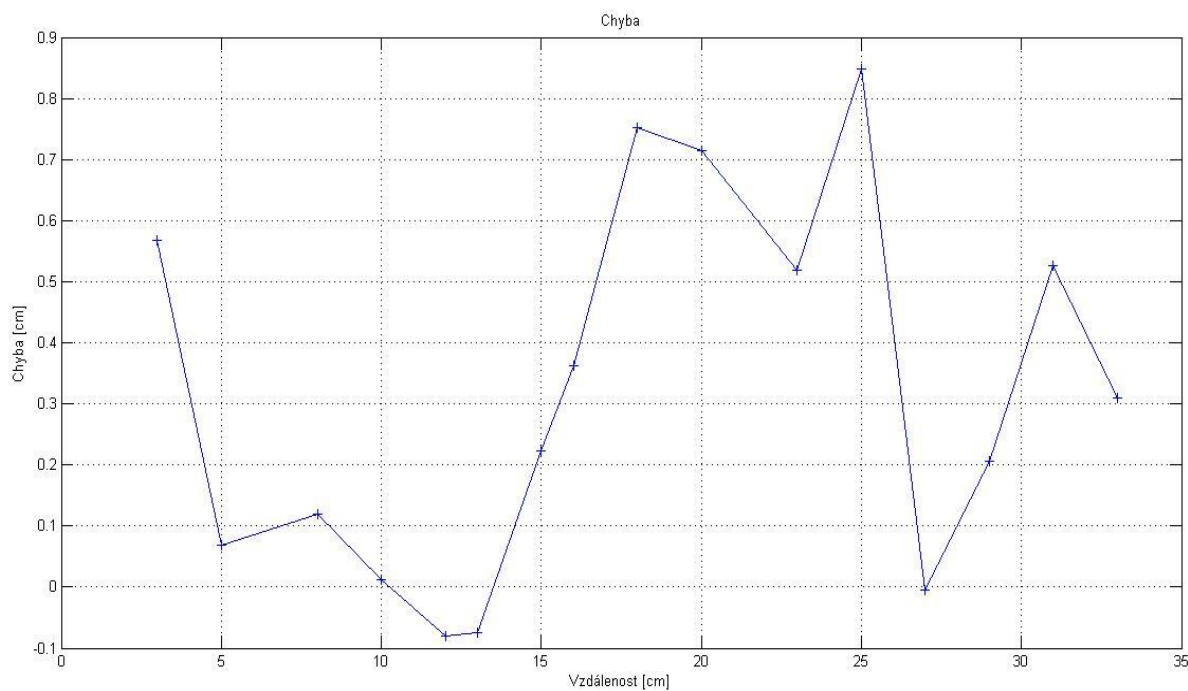
**Obr. 5.2 :** Charakteristika senzoru US-100

V dalším grafu (Obr. 5.3) jsem použil namísto průměru medián změřené série.

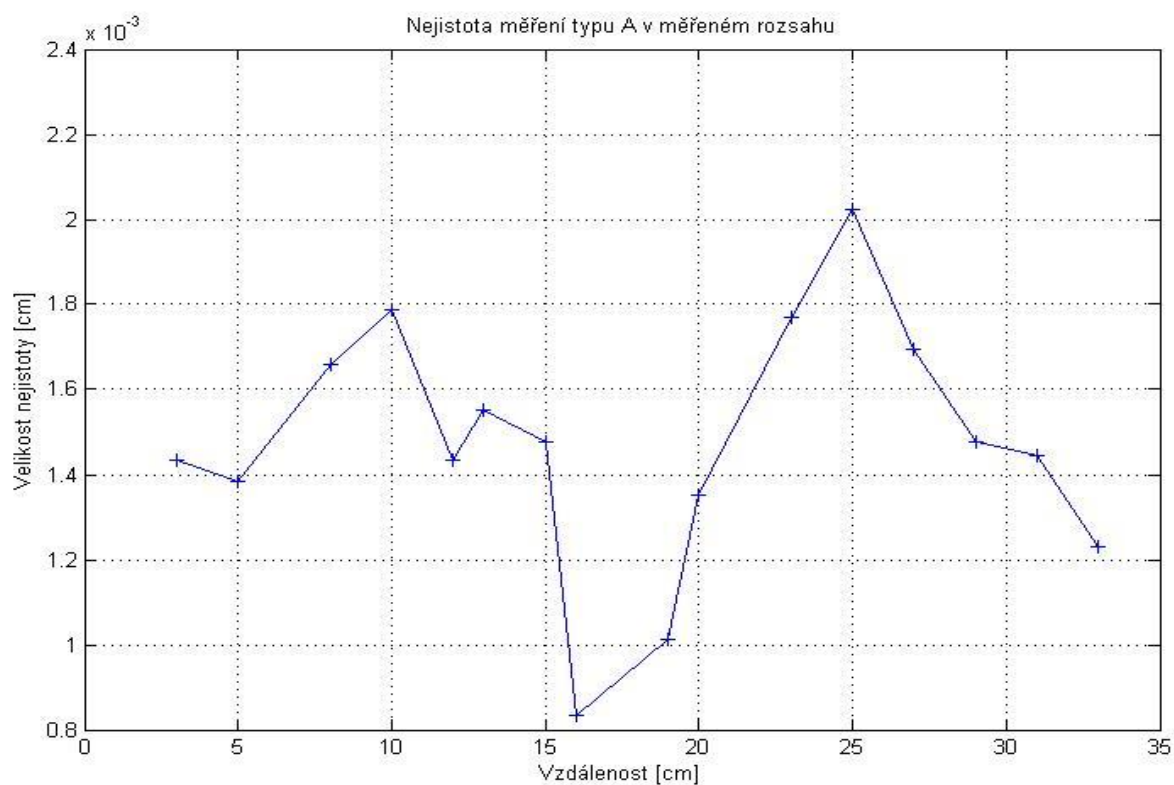


**Obr. 5.3 :** Charakteristika senzoru US-100 (s použitím mediánu)

Vypočítaná chyba v celém rozsahu měření. Chyba měření je rozdíl průměru naměřených hodnot a nastavované vzdálenosti (standardní chyba počítaná v měření).



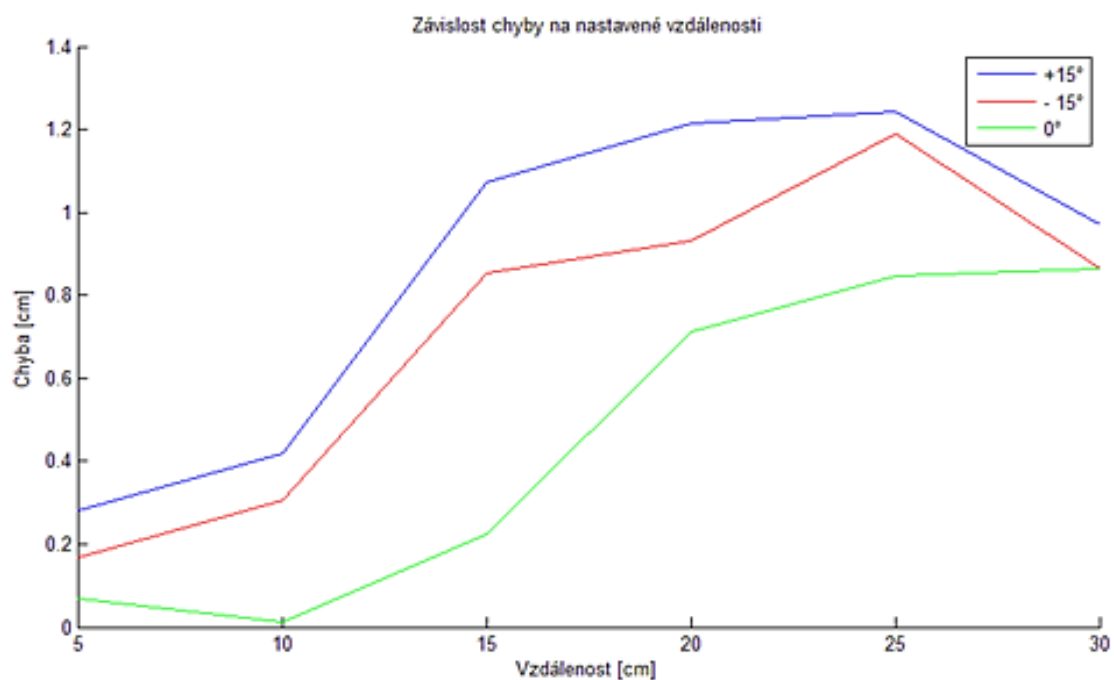
**Obr. 5.4 :** Chyba měření v závislosti na vzdálenosti senzoru US-100



Obr. 5.5 : Závislost nejistoty typu A v závislosti na vzdálenosti senzoru US-100

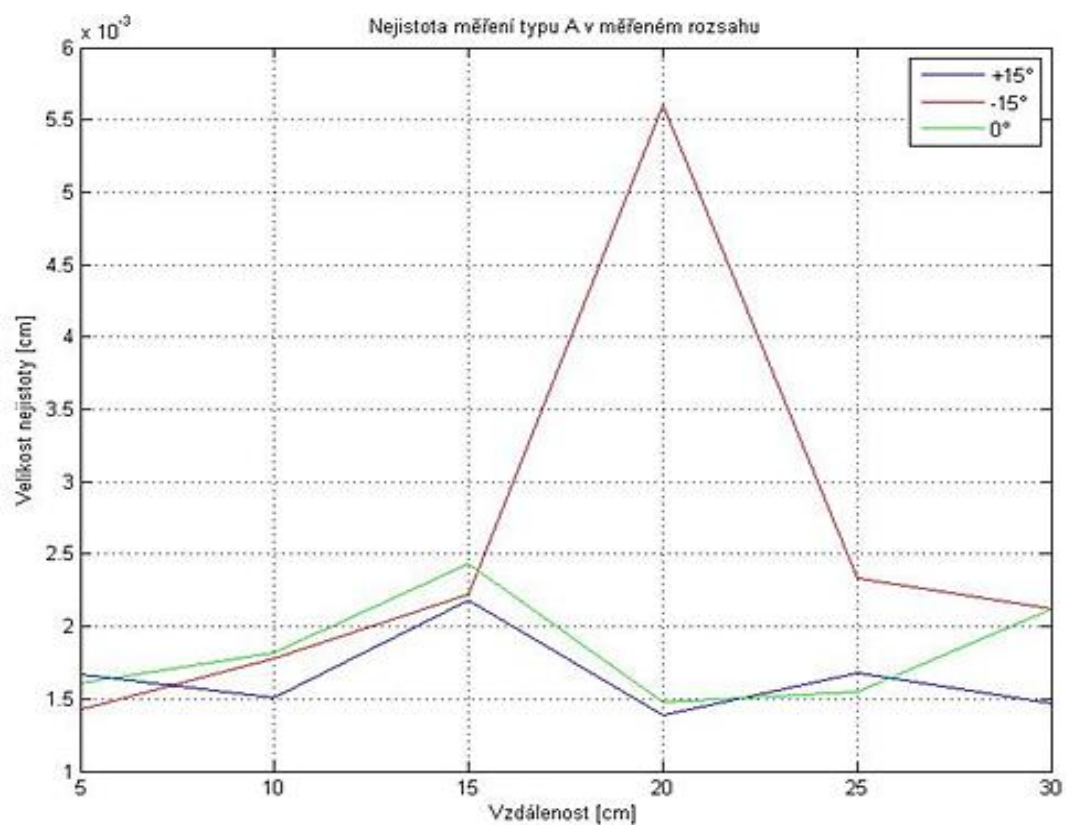
## 5.4.2 Měření vzdálenosti v $\pm 15^\circ$ od osy senzoru

Vychýlení od osy v  $-15^\circ$  je ke straně vysílače.

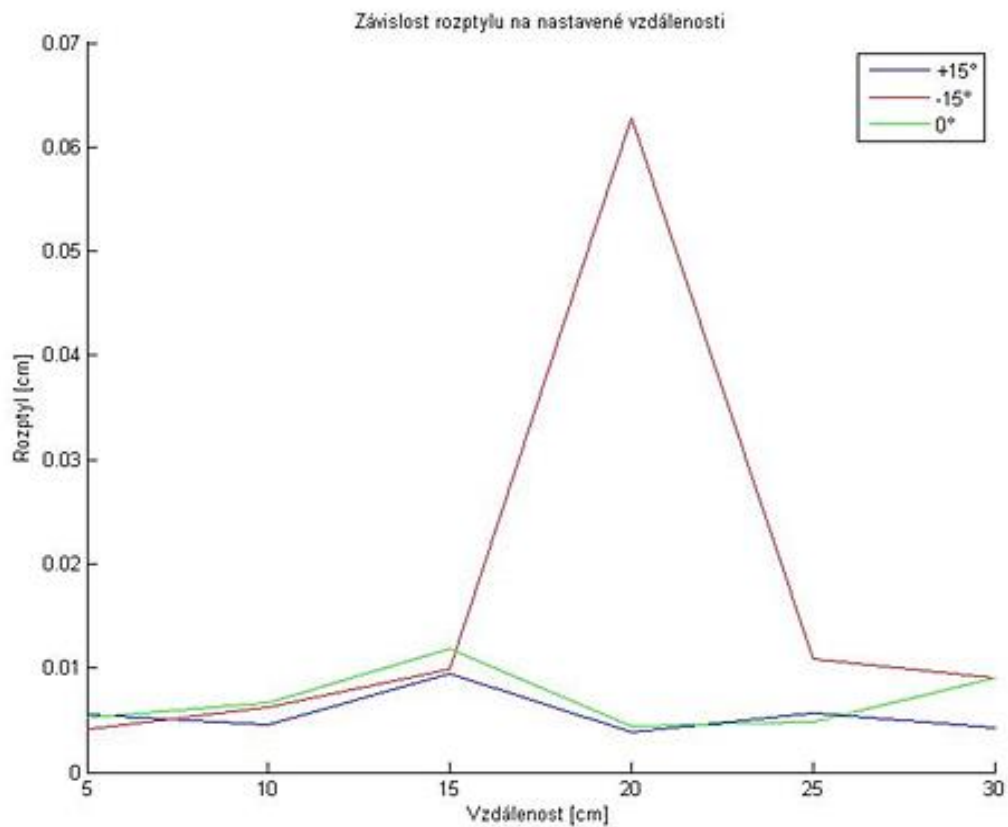


Obr. 5.6 : Závislost chyby při  $\pm 15^\circ$  a  $0^\circ$  vychýlení od osy





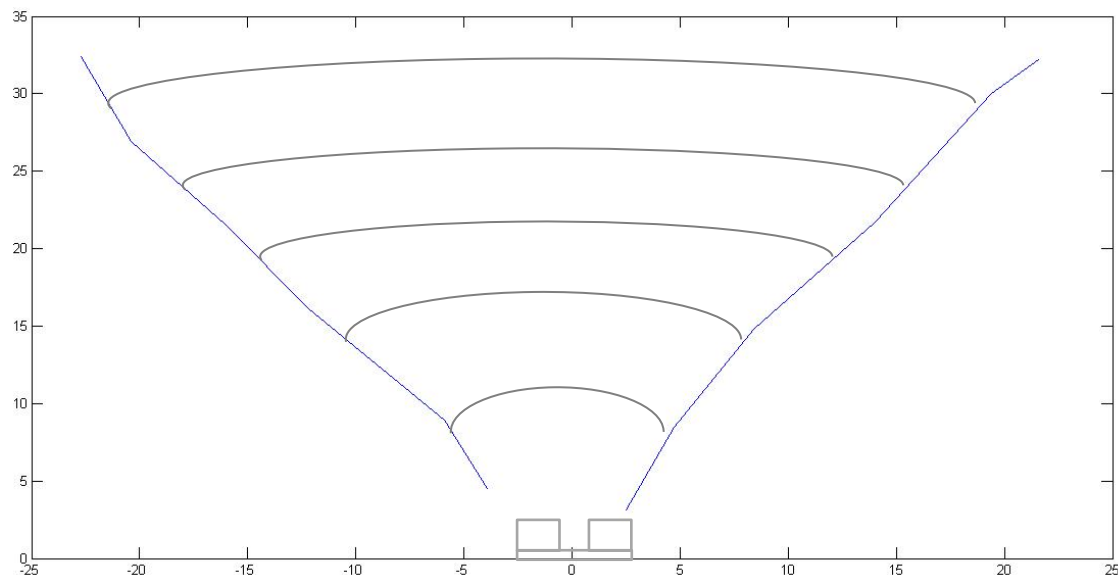
Obr. 5.7 : Závislost nejistoty při  $\pm 15^\circ$  a  $0^\circ$  vychýlení od osy



Obr. 5.8 : Závislost rozptylu při  $\pm 15^\circ$  a  $0^\circ$  vychýlení od osy

### 5.4.3 Měření vyzařovací charakteristiky senzoru US-100

Na následujícím grafu (Obr. 5.9) jsou zobrazeny hranice, kdy senzor ještě dokázal detekovat překážku. Mezi křivkami je prostor (vyšrafovaná část), kde lze změřit vzdálenost od překážky, avšak s přibližováním k těmto hranicím velice klesá přesnost. Záporná hodnota na x-ové ose je levá strana ultrazvukového senzoru (strana vysílače).

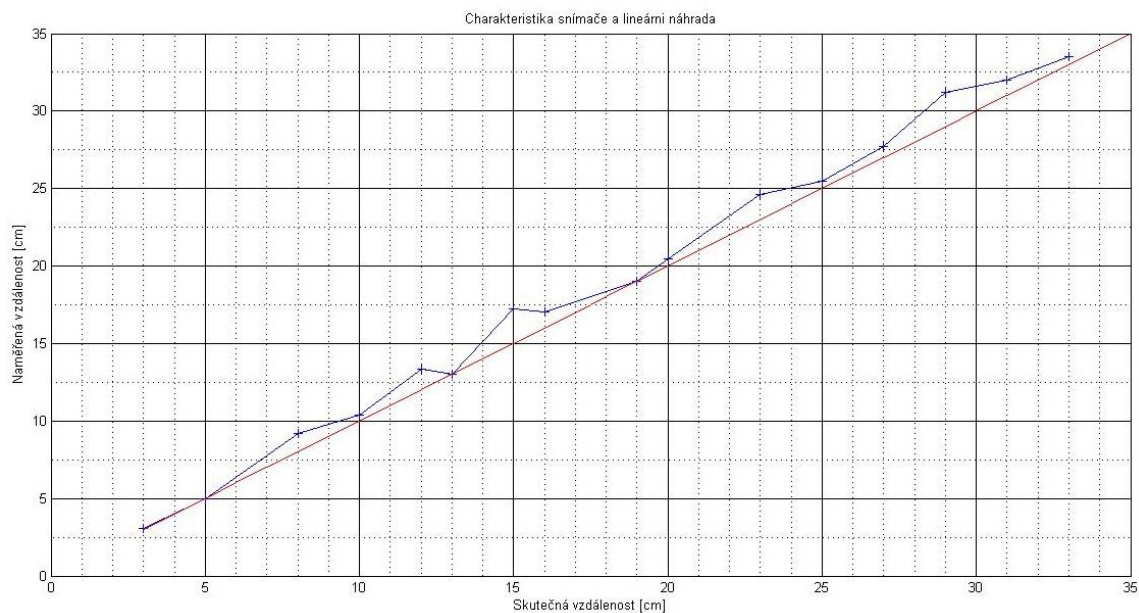


Obr. 5.9 : Vyzařovací charakteristika senzoru US-100

### 5.4.4 Měření senzoru SRF08

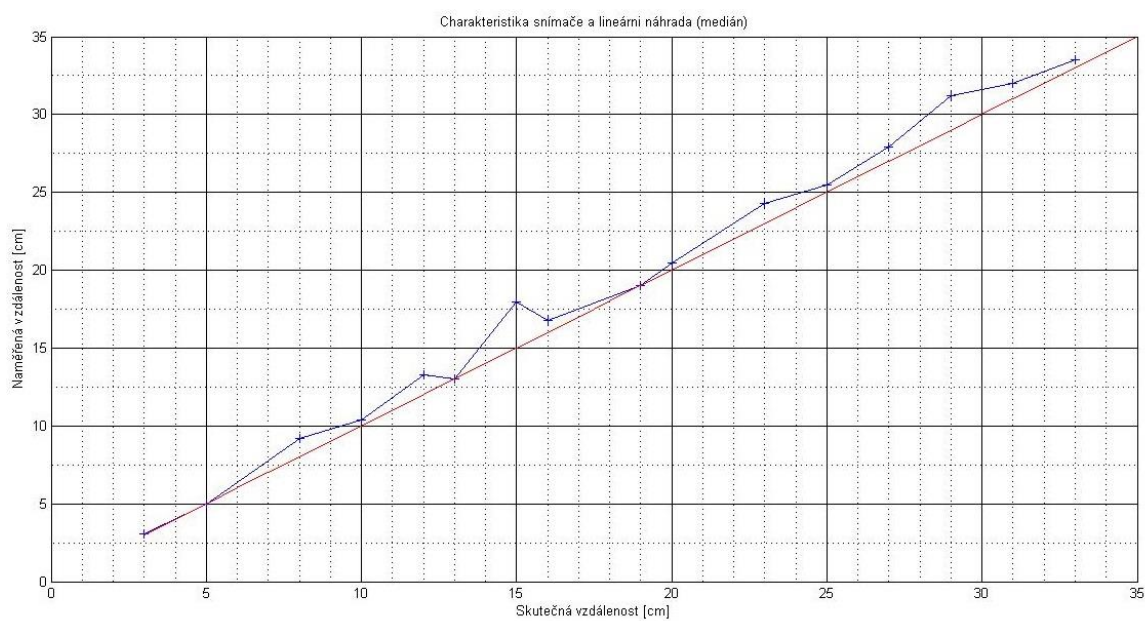
Měření vzdálenosti a porovnání s lineární náhradou. Při tomto grafu jsem použil jako změřenou vzdálenost průměr z naměřené série hodnot.

- modrý průběh - body představují průměry naměřených hodnot
- červený průběh - lineární náhrada (hodnoty které bych měl získat)



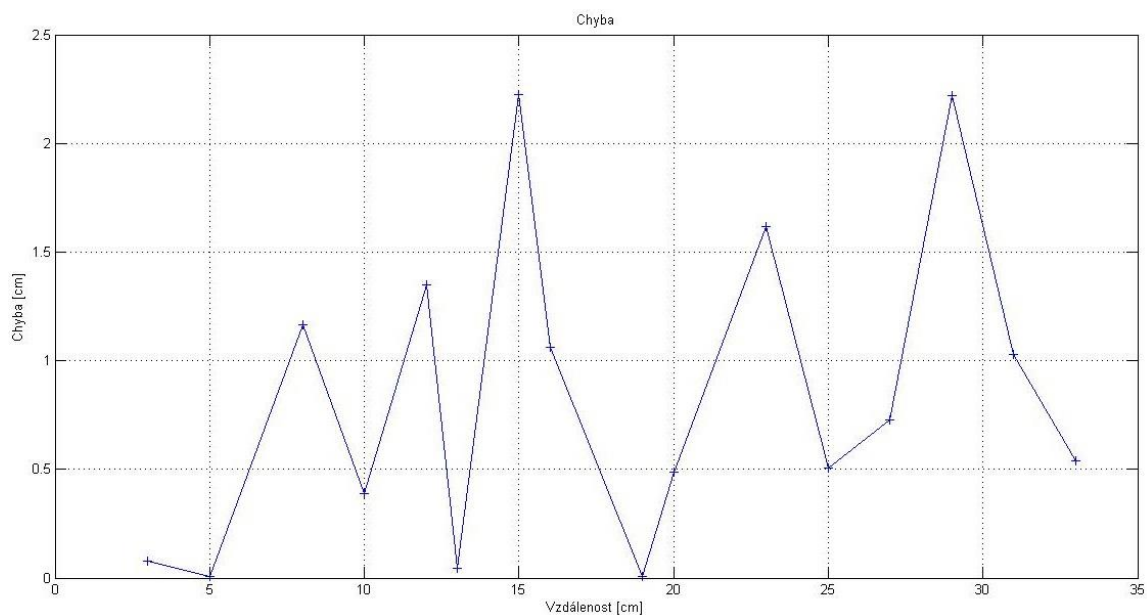
**Obr. 5.10 :** Charakteristika senzoru SRF08

V dalším grafu (Obr. 5.11) jsem použil namísto průměru medián změřené série.

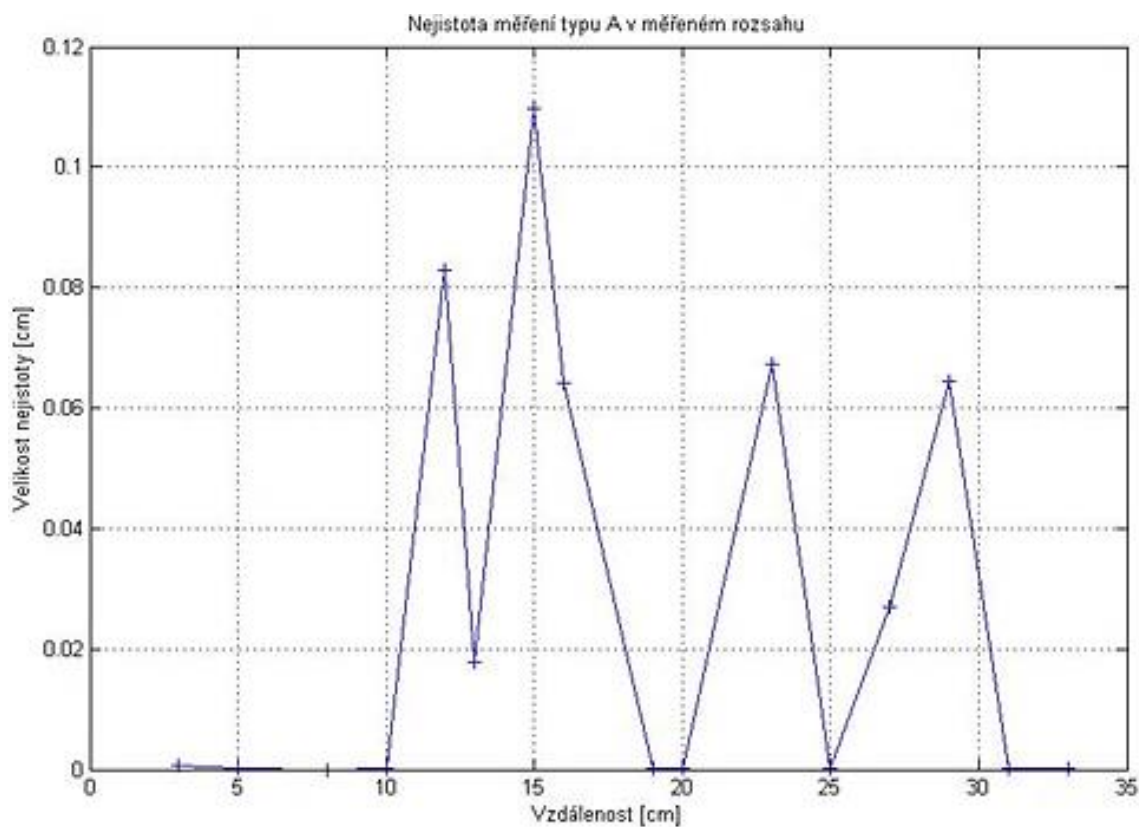


**Obr. 5.11 :** Charakteristika senzoru SRF08 (s použitím mediánu)

Vypočítaná chyba v celém rozsahu. Chyba měření je rozdíl průměru naměřených hodnot a nastavované vzdálenosti (standardní chyba počítaná v měření).



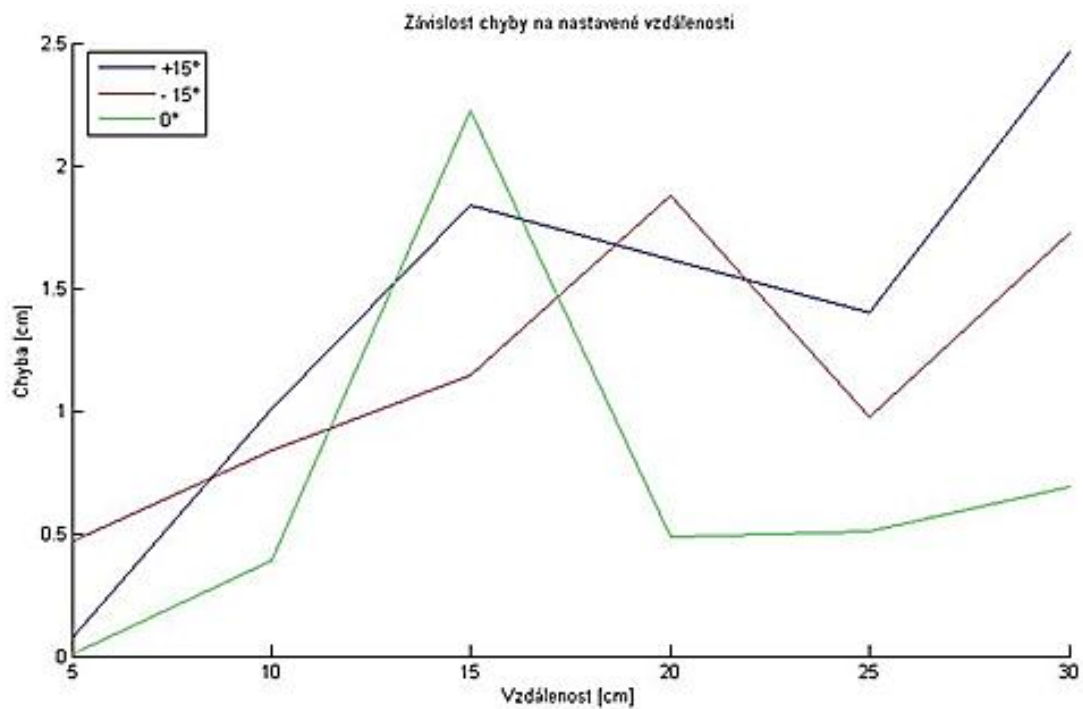
**Obr. 5.12 :** Chyba měření v závislosti na vzdálenosti senzoru SRF08



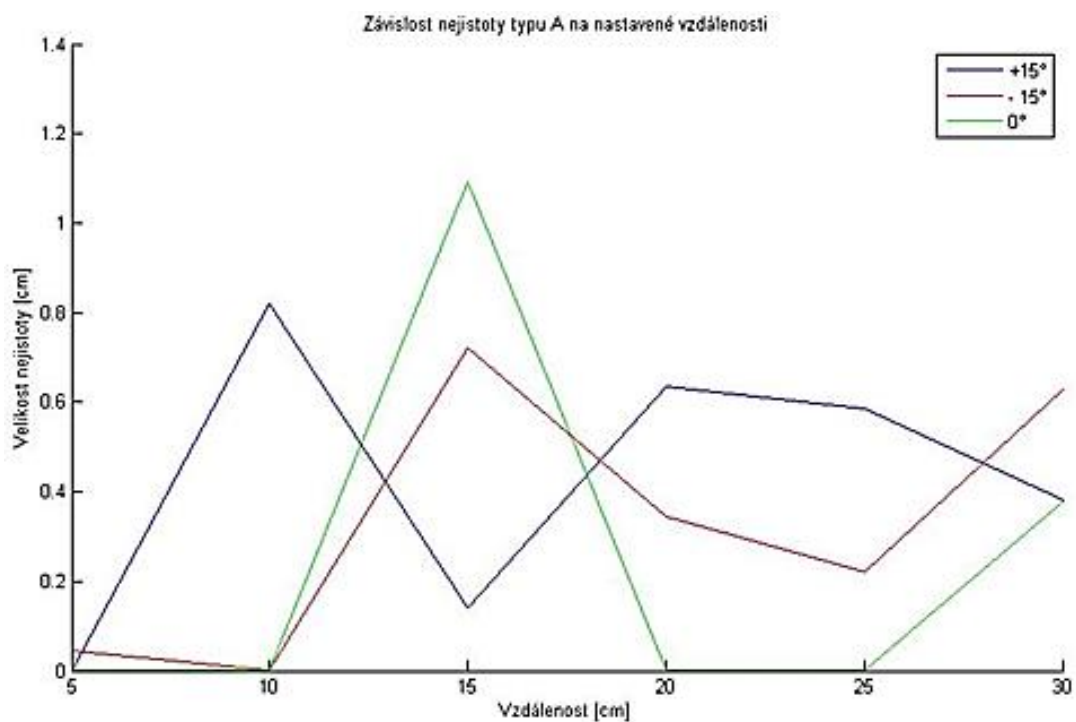
**Obr. 5.13 :** Závislost nejistoty typu A v závislosti na vzdálenosti senzoru SRF08

### 5.4.5 Měření vzdálenosti v $\pm 15^\circ$ od osy senzoru

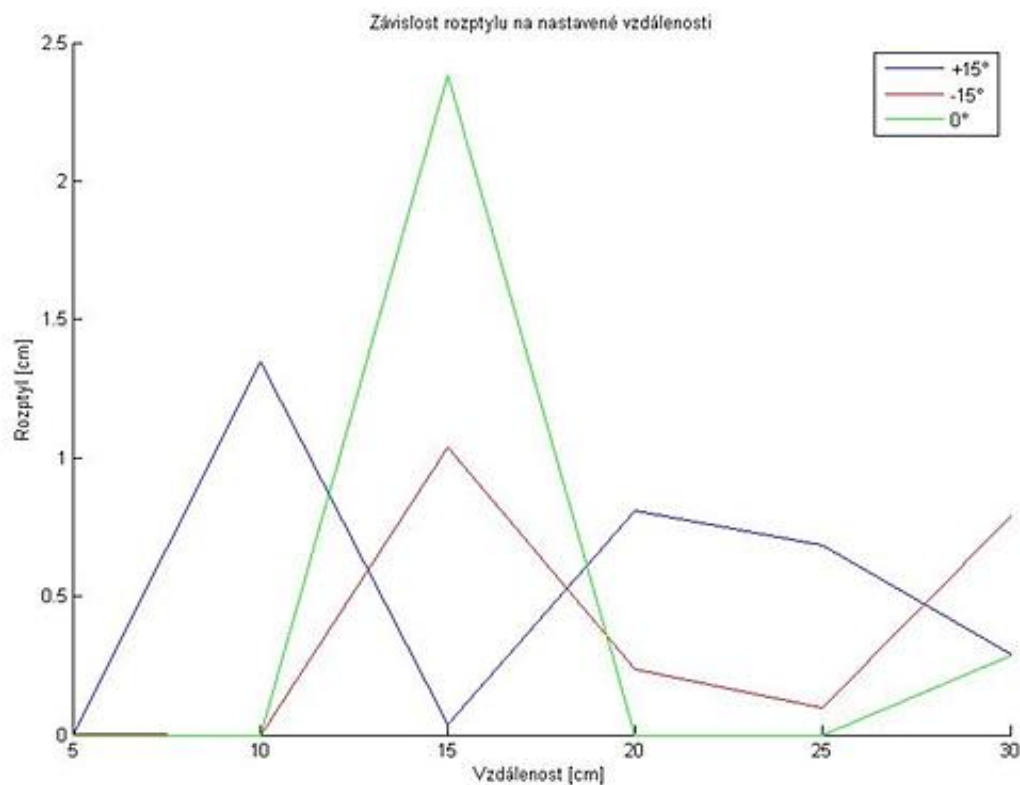
Vychýlení od osy v  $-15^\circ$  je ke straně vysílače.



Obr. 5.14 : Závislost chyby při  $\pm 15^\circ$  a  $0^\circ$  vychýlení od osy



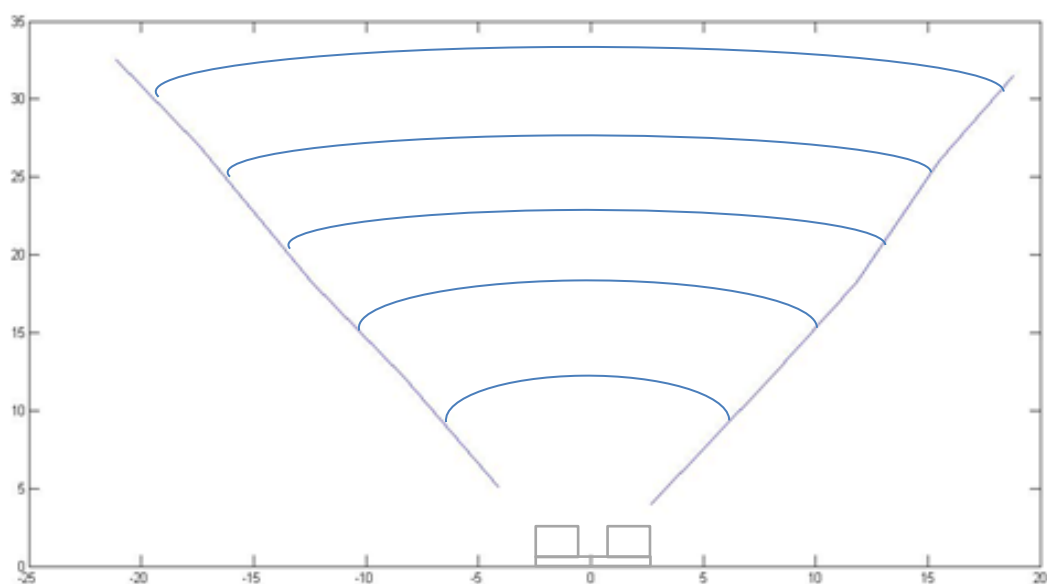
Obr. 5.15 : Závislost nejistoty při  $\pm 15^\circ$  a  $0^\circ$  vychýlení od osy



Obr. 5.16 : Závislost rozptylu při  $\pm 15^\circ$  a  $0^\circ$  vychýlení od osy

#### 5.4.6 Měření vyzařovací charakteristiky

Na následujícím grafu (Obr. 5.17) jsou zobrazeny hranice, kdy senzor ještě dokázal detekovat překážku. Mezi křivkami je prostor (vyšrafovaná část), kde lze změřit vzdálenost od překážky, avšak s přibližováním k těmto hranicím velice klesá přesnost. Záporná hodnota na x-ové ose je levá strana ultrazvukového senzoru (strana vysílače).



Obr. 5.17 : Vyzařovací charakteristika snímače SRF08

## 6 MĚŘENÍ VŠESMĚROVÝCH SENZORŮ

### 6.1 Měření senzorem PVDF 80 kHz

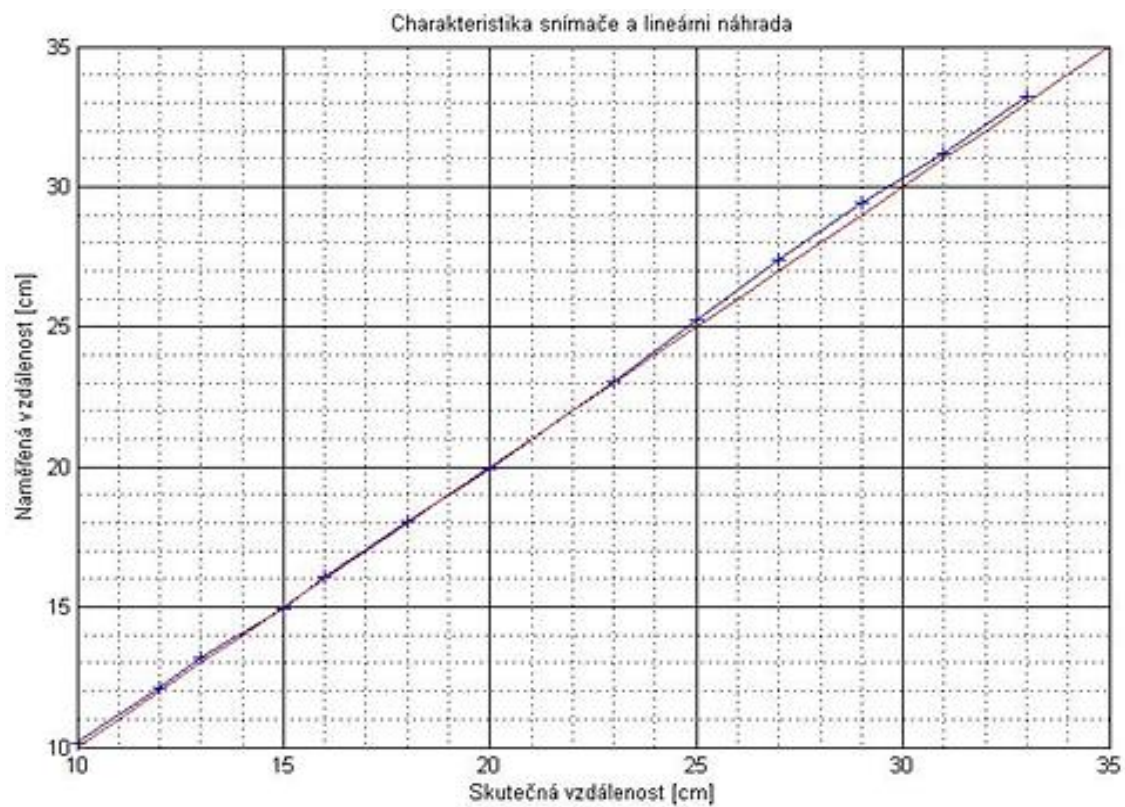
Tento senzor jsem měřil 1. metodou TOF, protože jsem měl k dispozici dva tyto ultrazvukové senzory, jeden jsem použil ve funkci přijímače a druhý senzor ve funkci vysílače. Výpočet vzdáleností, komunikace po sériové lince, princip ukládání dat atd. je stejné jako při měření směrových senzorů (kap. 5). Vysílač měříme v předem zvolených vzdálenostech, po umístění vysílače na danou vzdálenost a změření souboru tisíce hodnot, uložíme tyto hodnoty pro další zpracování. Proměříme stejnou pracovní oblast jako u předešlých směrových ultrazvukových senzorů (pro následné zhodnocení), jen s tím rozdílem, že pracovní oblast musí začínat až v 10 cm, důvodem je vznik šumové elektrické rozkmitání v obvodu přijímače (více v kap. 6.1.1, popř. Obr. 6.2). Po změření všech hodnot můžeme přejít k vyhodnocení měření, k tomu použijeme stejný algoritmus jako u směrových senzorů, výsledkem zhodnocení budou tedy závislosti; nejistoty měření typu A, absolutní chyby měření a rozptylu v závislosti na nastavované vzdálenosti. Vyzařovací charakteristiku tohoto senzoru není potřeba měřit, protože je tento senzor všesměrový; tudíž jeho zorný úhel je  $360^\circ$ , tento fakt jsem experimentálně ověřil – nebyla nalezena žádná oblast (úhel), ze které by senzor nedokázal nezachytit ultrazvukovou vlnu.

#### 6.1.1 Grafy a závislosti

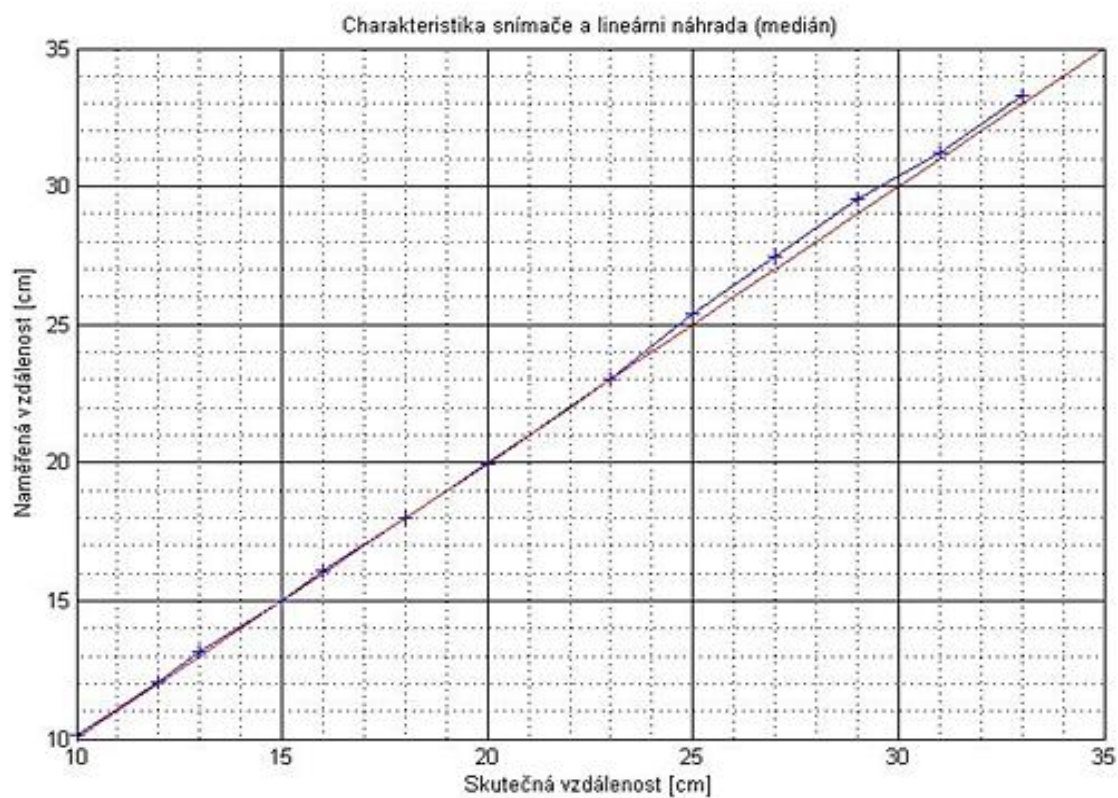
Měření vzdálenosti a porovnání s lineární náhradou. Při tomto grafu jsem použil jako změřenou vzdálenost průměr z naměřené série hodnot.

- modrý průběh - body představují průměry naměřených hodnot
- červený průběh - lineární náhrada (hodnoty které bych měl získat)





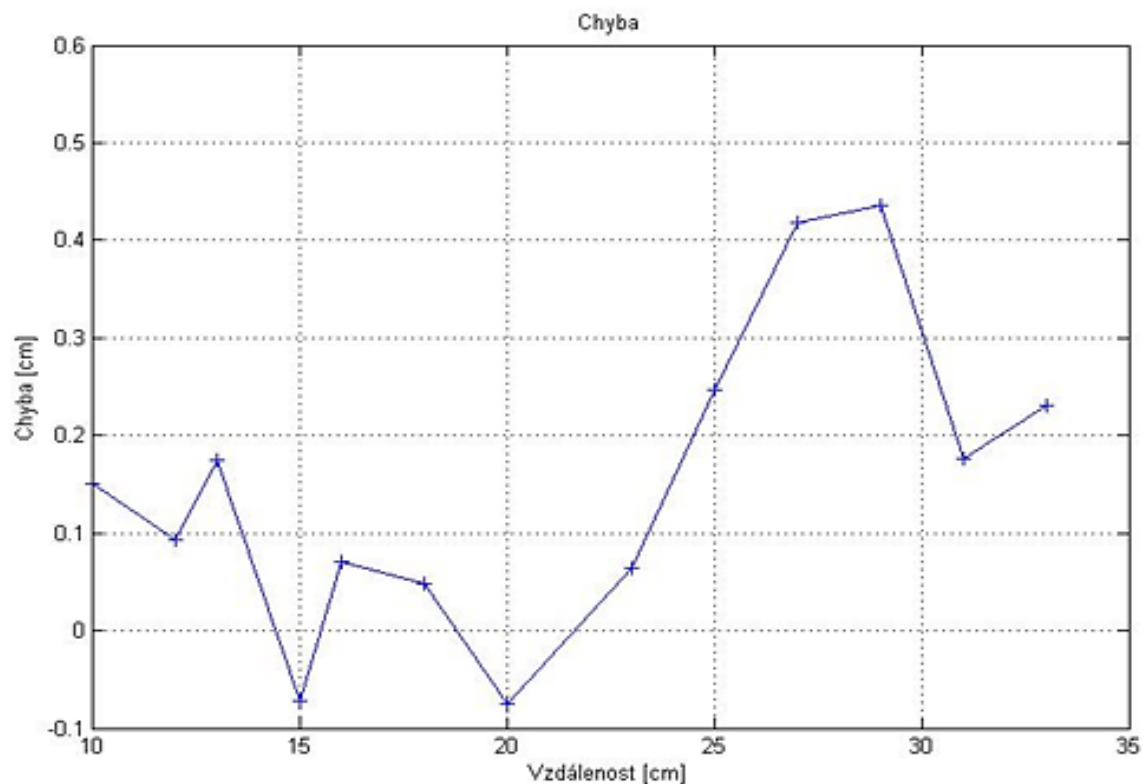
Obr. 6.1 : Charakteristika senzoru PVDF 80kHz



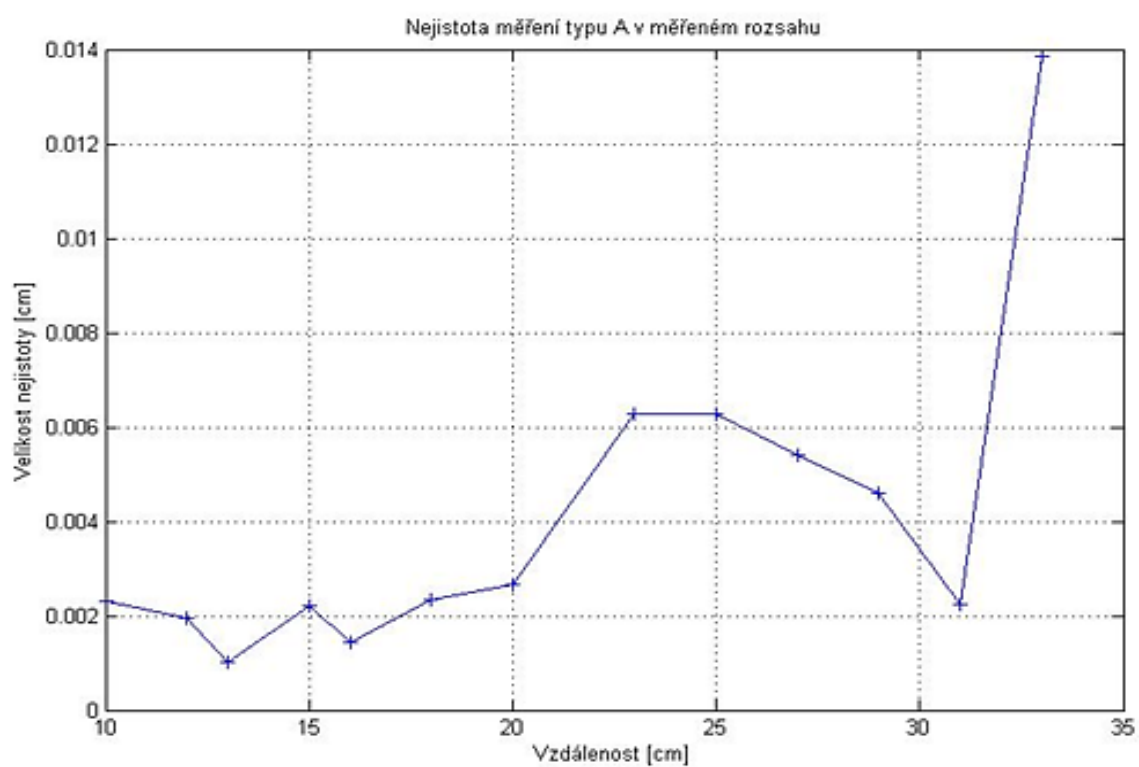
Obr. 6.2 : Charakteristika senzoru PVDF 80 kHz s použitím mediánu



Vypočítaná chyba v celém rozsahu měření. Chyba měření je rozdíl průměru naměřených hodnot a nastavované vzdálenosti (standardní chyba počítaná v měření).



**Obr. 6.3 :** Chyba měření v závislosti na vzdálenosti senzoru PVDF 80 kHz



**Obr. 6.4 :** Závislost nejistoty typu A na vzdálenosti senzoru PVDF 80 kHz

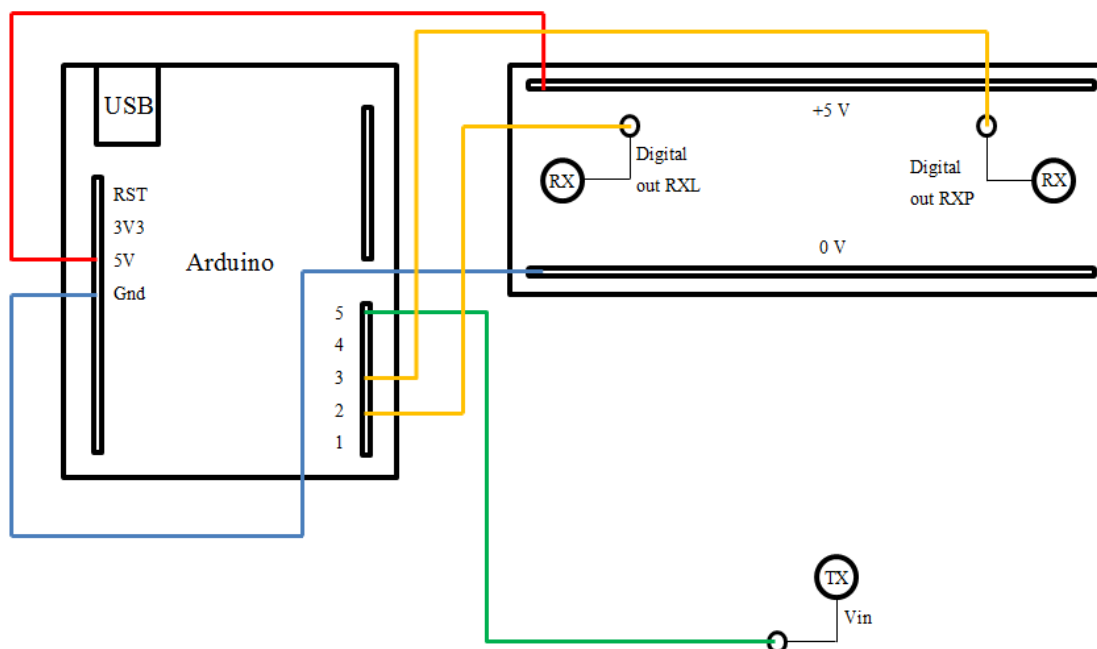
## 7 MĚŘENÍ 2D POLOHY

Pro získání 2D polohy, nebo-li získání x-ových a y-ových souřadnic určitého předmětu (překážky), nebo jak je v této práci zvoleno mobilního vysílače potřebujeme znát, vypočítat, popřípadě změřit potřebné informace. Nejjednodušší možnost jak získat x-ové a y-ové souřadnice je dokázat vytvořit obecný trojúhelník složený z přijímačů a vysílače (popř. překážky), kde pomocí goniometrických funkcí dopočítáme potřebné informace. V této práci jsem použil dva ultrazvukové přijímače a jeden ultrazvukový vysílač které představují zmíněný obecný trojúhelník, tyto přijímače jsou statické a mají mezi sebou stanovenou vzdálenost (14 cm), vysílač je mobilní a přimontovaný na dřevěným hranolku s tlačítkem pro lepší mobilitu a ovladatelnost. Tlačítko slouží k ovládání vysílání; při stisknutí tlačítka vysílač vysílá ultrazvukové vlny, přijímač tyto ultrazvukové vlny detekuje, a poté se dopočítává vzdálenost vysílače a jeho souřadnice. Po dostatečném množství dat se trajektorie vysílače vykresluje na PC. V opačném případě, pokud je vysílač bez napájení můžeme se s vysílačem přesunout na potřebné místo a pokračovat v psaní, tato vlastnost je vhodná pro tvorbu obrázků, obrazců, psaní písma a podobně. Postup výpočtu souřadnic je vyřešen v Kap 4.4.

Přesnost x-ových a y-ových souřadnic se odvíjí od toho, s jakou přesností dokážeme změřit dobu letu ultrazvukové vlny od vysílače k přijímačům, tedy jejich vzdálenost, samozřejmě na přesnost mají vliv i další podněty, např. okolní rušení, nepřesně nastavené vzdálenosti mezi přijímači a další. Doba letu ultrazvukové vlny je měřeno vnitřním čítačem v mikrokontroléru Arduino, a to s rozlišením 4  $\mu$ s, přepočtem na vzdálenost pomocí rychlosti zvuku dostaneme rozlišení 1,324 mm (6.1).

$$d = 331,4 * 4 * 10^{-6} = 0,001324 \text{ m} = 1,324 \text{ mm} \quad (6.1)$$

Přesnost může být také způsobena rychlým kolísáním teploty, nebo špatně (nepřesně) nastavené teploty používané při výpočtech, jak je totiž známo, rychlost šíření zvuku je přímo závislá na teplotě (4.1). Přesnost je také dána okolním šumem a rušením.



Obr. 7.1 Schématické znázornění přípravku pro měření 2D polohy

## 7.1 Obsluha ultrazvukových převodníků pomocí Arduino

Ultrazvuky připojíme k mikrokontroléru, názorné schématické zapojení je na Obr. 6.1. Po připojení napájení a zemí zapojíme ultrazvukový pin „Vin“ ve funkci „transmitter“ (vysílač) na pin č. 5, ultrazvukové přijímače ve funkci „reciever“ (přijímač), resp. jejich piny „Digital out“ zapojíme na digitální piny 2 a 3 mikrokontroléru Arduino. Vysílač můžeme zapojit do jakéhokoliv pinu typu „digital output“, pokud v řídicím programu změníme číslo na řádku „*const int trigger = 5;*“ na příslušné číslo pinu. Přijímače musíme ponechat na těchto pinech, protože použitý typ mikrokontroléru podporuje přerušování jen na těchto pinech. Přerušování pro tuto úlohu je důležité, protože pomocí nesynchronního hardwarového přerušování dokážeme rychle reagovat na vnější události, právě například zjištění přijetí ultrazvukové vlny už při prvním zákmitu. Jiným softwarovým řešením by tato úloha byla dosti nepřesná, protože by nebylo zajištěné odchytnutí prvního rozkmitu ultrazvukové vlny, nebo dokonce by se mohlo stát, že bychom nezachytili celou ultrazvukovou vlnu.

Program se spustí automaticky ihned po spuštění komunikace mezi počítačem a mikrokontrolérem, toto musí být zajištěno, aby při spuštění programu přes Matlab mohlo ihned začít vykreslování trajektorie. Program vykresluje nekonečně dlouho, to může být změněno jak v programu pro obsluhu ultrazvuku, tak v Matlabu přidáním cyklu „for“ a jeho podmínkou pro ukončení. Nejdříve musíme nastavit komunikaci po sériové lince, módy pinů ve funkci „*setup()*“, popř. konstanty pro výpočty a další. Na začátku hlavního programu (ve funkci „*loop()*“) nejdříve pro jistotu vypneme přerušování na oba

piny přijímačů, a to z toho důvodu, aby při vykonávání programu pro obsluhu vysílače nenastalo přerušení od přijímačů, které by v této chvíli nemělo nastat.

### 7.1.1 Obsluha ultrazvukového vysílače

Následující blok programu je určen pro ovládání vysílání ultrazvukového vysílače. V proměnných „*CurrentTimeL*“ a „*CurrentTimeP*“ zachycují aktuální hodnotu časovače při přerušení, které nastane při přijmutí ultrazvukové vlny přijímačem, tyto proměnné musejí být na začátku každého měření vynulovány.

```
CurrentTimeL = 0;
CurrentTimeP = 0;
digitalWrite(trigger, LOW);
delayMicroseconds(5);

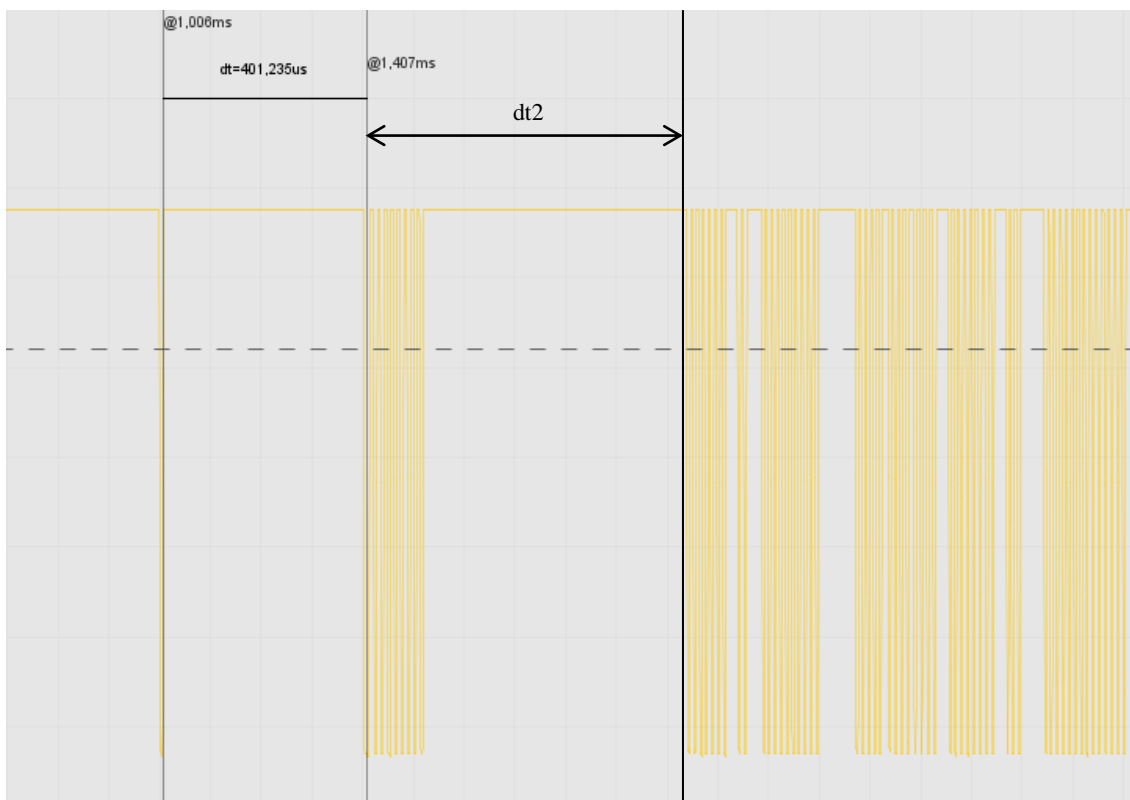
digitalWrite(trigger, HIGH);
delayMicroseconds(400);

StartTime = micros ();
digitalWrite(trigger, LOW);

delayMicroseconds(300);
attachInterrupt(0, prijemL, LOW);
attachInterrupt(1, prijemP, LOW);
```

Aktuální hodnota interního časovače obsahuje čas v mikrosekundách od spuštění mikrokontroléru, pomocí příkazu „*micros()*“ tento čas vyčteme a můžeme ho uložit do nějaké proměnné a nadále s ním pracovat. Tento časovač se spustí při každém spuštění mikrokontroléru, rozlišení tohoto časovače je 4  $\mu$ s a po přibližně 70 minut se časovač vynuluje [5]. Příkazem „*digitalWrite(pin, hodnota)*“ (parametr **pin** – číslo pinu, který chceme ovládat; parametr **hodnota** – na jakou hodnotu chceme nastavit daný pin (HIGH, LOW)) nastavujeme pin vysílače do klidového stavu (LOW) nebo ho uvedeme do stavu k vysílání (HIGH). Pro jistotu nejdříve nastavíme pin na hodnotu LOW (mohlo by se stát, že pin nebyl nastaven na hodnotu LOW správně) a poté následuje zpoždění 5  $\mu$ s (klidový čas). V datasheetu pro použitý ultrazvukový senzor [13] doporučují budit tento senzor ve vysílacím módu impulsem se šířkou alespoň 300  $\mu$ s, to zajistíme příkazem („*delayMicroseconds(400);*“), poté je potřeba uložit čas, kdy jsme vyslali ultrazvukovou vlnu (obvod vysílače vyšle ultrazvukovou vlnu až po sestupné hraně

impulzu). Již tedy zmíněným příkazem „*micros()*“ uložíme čas při vyslání ultrazvukové vlny a uvedeme pin do stavu LOW. Následující zpoždění (*delayMicroseconds(300);*) je určeno pro softwarové odfiltrování elektrického rozkmitání při vzniklém přepětí způsobené induktorem v obvodu vysílače (viz schéma v kap 3.3.1). Na Obr. 6.2 je ukázán průběh na osciloskopu ultrazvukového přijímače při přijímání ultrazvukové vlny, časové zpoždění označené „*dt*“ je doba, kdy ultrazvukový vysílač je ve stavu HIGH, tento stav trvá 400  $\mu$ s (viz. výše), po tomto zpoždění je vlna vyslána a nastává elektrické rozkmitání, které se, jak je vidno projeví i na přijímači. Časové zpoždění „*dt2*“ je doba letu ultrazvukové vlny k přijímači. Elektrické rozkmitání má trvání cca 100-200  $\mu$ s, ale pro jistotu je filtrovací čas zvolen 300  $\mu$ s, touto příčinou je pracovní oblast tohoto přípravku posunuta asi od 9 cm. Příkazem „*attachInterrupt(pin,fce,stav);*“ (parametr **pin** – číslo pinu na kterém detekujeme přerušení; parametr **fce** – funkce/metoda, která se vyvolá a provede po zjištění tohoto přerušení; parametr **stav** – určuje, při jaké stavu na pinu se má vykonat přerušení, možné nastavitelné stavy jsou LOW, HIGH, FALLING, RISING, CHANGE) zapneme hardwarové přerušení na daných pinech.



**Obr. 7.2 :** Časový průběh ultrazvukového přijímače na osciloskopu

### 7.1.2 Obsluha ultrazvukových přijímačů

Za blokem pro obsluhu vysílače musí následovat program pro zachycení ultrazvukové vlny a následné výpočty pro další práci. Jak je napsáno v předchozí kapitole, po vyslání ultrazvukové vlny jsou zapnuty hardwarové přerušování typu LOW na piny přijímačů a uložen čas vyslání vlny. Po této části programu se program dostane do smyčky, ze které nevystoupí, dokud nebude splněna jedna z následujících podmínek;

- 1) První podmínkou je přijetí ultrazvukové vlny na obou přijímačích, toto zajišťují metody „*prijemL*“ a „*prijemP*“, které se vykonají při přerušování na daných pinech. Dále se v těchto funkcích inkrementují proměnné „*kolikL*“ a „*kolikP*“, ve kterých je uloženo kolikrát nastalo přerušování (pokud chceme zachytávat jiný než první zážitek ultrazvuku, můžeme k tomu lehce přizpůsobit program tím, že přepíšeme proměnnou „*kolikatej*“, která je implicitně nastavena na 1). Touto podmínkou jsou naplněny proměnné, ve kterých je uložen čas, kdy byla vlna přijata, a to jsou „*CurrentTimeL*“ a „*CurrentTimeP*“. Aktuální hodnotu času při přijetí vlny vyčítám z interního čítače pomocí již zmíněného příkazu „*micros()*“.
- 2) Druhou podmínkou pro vystoupení z této smyčky je ztracení (nepřijetí) vlny. Pokud nepříjde do určitého předdefinovaného času (nastaven na 100 ms) přerušování na obou pinech (k obou přijímačům nepříjde ultrazvuková vlna), program z této smyčky vystoupí a může začít od začátku, resp. program zase začne vykonávat obsluhu vysílače (kap. 7.1.1). Tato čekací smyčka je vytvořena pomocí příkazu „*micros()*“, a to tak, že každým průchodem smyčkou porovnává program aktuální dobu, kterou čeká s nastavenou hodnotou (maximální doba čekání), pokud se tedy tyto hodnoty rovnají (ultrazvuková vlna nebyla přijata na obou přijímačích), program vypíše po sériové lince „ztraceno“.

Pokud tedy program vyhodnotil přijetí ultrazvukové vlny na obou přijímačích a jsou uloženy časy přijetí, můžeme s nimi nadále pracovat, důležitým krokem je konečný výpočet uraženého času ultrazvukové vlny – odečtením času, kdy byla vyslána vlna od aktuálního času dostaneme dobu letu ultrazvukové vlny (viz následující úryvek programu). Dále můžeme aplikovat přepočet tohoto času na vzdálenost, a to v námi zvolených jednotkách (zde centimetry).

```
ElapsedTime_L = CurrentTimeL-StartTime;  
ElapsedTime_P = CurrentTimeP-StartTime;  
distanceL = ((ElapsedTime_L*0.000001)*rychl)*100;  
distanceP = ((ElapsedTime_P*0.000001)*rychl)*100;
```

## 7.2 Realizace výpočtů 2D polohy v Arduinu

Teoretický postup a odvození vzorců pro výpočet 2D pozice je rozebrán v Kap 4.4. V této kapitole je popsána implementace vzorců do jazyka Wiring (jazyk používaný softwarovým prostředím Arduino). Prvním krokem, pokud tedy máme vypočtené vzdálenosti senzorů, je výpočet úhlů u ultrazvukových přijímačů („*alfa*“ a „*beta*“), viz následující kód. Při výpočtech 2D polohy jsem předem určil začátek souřadných os na pozici levého přijímače, všechny další výpočty se tedy budou vztahovat k tomuto bodu. Před výpočtem x-ové a y-ové souřadnice si musíme uvědomit, že může nastat stav, kdy úhel při vrcholu levého přijímače bude větší než 90° (vysílač se nachází nalevo od levého přijímače). V tomto případě musíme použít jiné vzorce, protože při tomto úhlu se předmět pohybuje v záporných x-ových souřadnicích a to předchozími vzorci nemůžeme vypočítat, musíme tedy tento stav ošetřit. Nejjednodušší možností je kontrolovat velikost úhlu *beta*, použijeme tedy vzorce v bloku začínajícím řádkem „*if(beta) > 1.5*“, a pokud je tato podmínka splněna, máme jistotu, že vysílač se nachází v záporných x-ových hodnotách. Pomocí proměnné „*zakl*“ přepočítáme kladnou x-ovou souřadnici na zápornou x-ovou tak, abychom zanechali počátek souřadných os na levém přijímači, a přitom zajistili správné vykreslování trajektorie.

```
aalfa = ((-(distanceL*distanceL)+(distanceP*distanceP)+
          +(zakl*zakl))/(2*distanceP*zakl));
abeta = ((-(distanceP*distanceP)+(distanceL*distanceL)+
          +(zakl*zakl)) / (2*distanceL*zakl));
alfa = acos(alfa);
beta = acos(abeta);

if((beta) > 1.5)
{
y = distanceL*sin(beta);
x = zakl - (sqrt((distanceP*distanceP)-(y*y)));
}
else
{
y = distanceL*sin(beta);
x = sqrt((distanceL*distanceL)-(y*y));
}
Serial.println('x');
Serial.println(distanceL);
```

```

Serial.println(distanceP);

// Serial.println(x);
// Serial.println(y);

```

Na posledních řádcích je zajištěno posílání „x“ a „y“ souřadnic přes sériovou linku, před těmito souřadnicemi nejdříve posíláme písmeno x, toto písmeno zajišťuje při dalším zpracování v Matlabu, aby posílané data byly zpracovávány ve správném pořadí, a to; nejdříve písmeno „x“, souřadnice „x“ a naposled souřadnice „y“. Samozřejmě můžeme zvolit jiný sled posílaných hodnot, nebo i také můžeme namísto souřadnic posílat vzdálenosti „distanceL“ a „distanceP“ a následně počítat souřadnice v Matlabu (viz. Kap. 7.3), stačí jen zvolit sled posílaných hodnot a poté nastavit v Matlabu stejný sled přijímání.

### 7.3 Realizace výpočtů 2D polohy v Matlabu

Tento způsob výpočtu jsem zvolil, abych zvýšil přesnost výpočtu souřadnic, ve výpočtu se totiž vyskytuje funkce „sin()“, tudíž právě přesnost vypočtených souřadnic se také odvíjí od toho jak přesně je tato funkce „sin()“ aproximována. Analogie výpočtů je stejná jako při počítání souřadnic v Arduinu (viz. Kap. 7.2). Dále v této práci budu používat tento způsob výpočtu souřadnic.

### 7.4 Real-time zobrazování v Matlabu

Nejdříve musíme nastavit parametry sériové komunikace; jako jsou port, baud rate, parita a další. V Matlabu pro zprovoznění sériové komunikace existuje metoda „serial“. Aby nastala komunikace mezi mikrokontrolérem a Matlabem, musejí se shodovat parametry nastavení sériové komunikace v Arduinu s hodnoty nastavenými v Matlabu, parametry komunikace nastavíme v Matlabu tímto příkazem „*s1 = serial('COM21','BaudRate',9600,'Parity','none');*“. Při nastavení sériové komunikace vytvoříme objekt „s1“, v tomto objektu jsou uloženy parametry komunikace, a pomocí tohoto objektu můžeme ovládat komunikaci. Pomocí příkazy „*fopen()*“ otevřeme sériový port pro komunikaci a „*fclose()*“ ukončíme komunikaci. Důležité je mít na paměti, že při otevření kanálu musíme pokaždé tento kanál ukončit a v jednu chvíli může být využíván v počítači jen jedním programem.

Důležitým krokem, jak je zmíněno výše, je zajištění správného sledu načítání hodnot. Přicházející řádky po sériové lince lze číst příkazem „*fgets()*“, jelikož prvním



znakem, který by měl přijít je písmeno ‚x‘, nejjednodušším způsobem jak zjistit jestli právě přijatý znak je písmeno ‚x‘ je pomocí hodnoty ASCII pro písmeno x. Zjištění hodnoty ASCII přijatého znaku jsem zajistil přetypováním metodou „double()“, touto metodou vznikne nový vektor, kde na prvním místě tohoto vektoru je ASCII hodnota přijatého znaku. Tuto získanou hodnotu porovnáme s přednastavenou hodnotou, představující hodnotu ASCII písmena ‚x‘ v podmínce (`if(prijato(1) == 120)`). Pokud je podmínka splněna, je zřejmé, že po tomto přijatém řádku následuje vypočtená vzdálenost od levého přijímače „distanceL“ a hned poté vzdálenost od pravého přijímače „distanceP“. Program tedy pokračuje tím, že tyto přijaté hodnoty nejdříve přetypuje příkazem „str2double()“ a poté je uloží do proměnných, nejlépe se stejným jménem.

V tuto chvíli, kdy známe vzdálenosti od obou senzorů, vypočítáme x-ové a y-ové souřadnice, jak je popsáno v Kap 7.3. Po vypočítání souřadnic musíme odstranit hodnoty, které by velice ovlivnili statistické hodnoty. Musíme tedy zjistit, jestli výsledek je vůbec číslo metodou „`isnan(p) ~= 1`“ (pokud je proměnná „p“ napsaná do závorek číslo, vrátí metoda hodnotu 1). Dále může nastat, že výsledek vyjde nekonečno, to lze ošetřit metodou „`isinf(p) ~= 1`“ – analogicky jako při metodě „`isnan()`“; pokud je proměnná „p“ nekonečno, metoda vrátí hodnotu 1. Následujícím úryvkem programu (podmínkou) kontrolujeme vhodnost výsledku (výsledek musí být konečné číslo), pokud tedy obě hodnoty souřadnic jsou konečná čísla, podmínka je splněna a program může pokračovat.

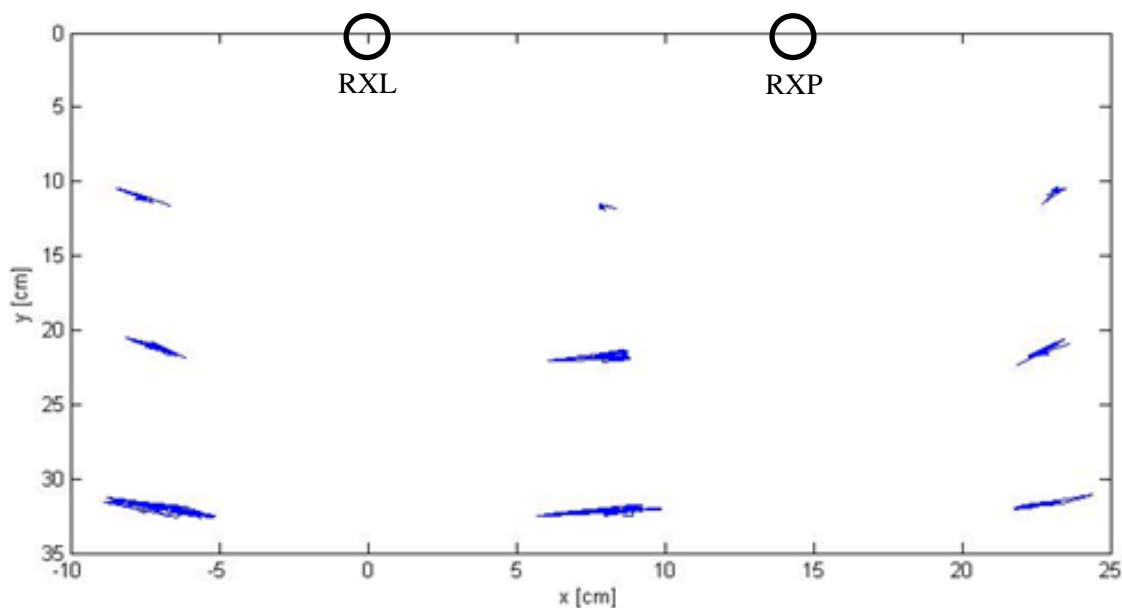
```
if((isinf(x(i)) ~= 1) || (isnan(x(i)) ~= 1) || (isinf(y(i))
    ~= 1) || (isnan(y(i)) ~= 1))
```

Pokud by se stalo, že některá souřadnice není konečné číslo, program by tyto hodnoty zahodil a začal číst nové (resp. pokračoval na začátku cyklu, u přečtení prvního řádku na sériové lince a kontroly písmene ‚x‘). Pokud vypočtené souřadnice vyhovují, program je příkazem „`plot(x,y)`“ vykreslí do grafu. Jelikož ale Matlab nevykresluje grafy okamžitě (čeká až přijdou všechny hodnoty – resp. nastane konec cyklu, skriptu), musíme zajistit vykreslení těchto hodnot po každém výpočtu; za příkaz „`plot(x,y)`“ napíšeme příkaz „`drawnow`“; tímto příkazem jsme zajistili real-time vykreslování. Poté program posune index vektorů souřadnic na další pozici a program může pokračovat ve zpracování další pozice. Samotné vykreslování, komunikace po sériové lince, výpočty a samotná obsluha Matlabu zabere nějaký čas, proto je potřeba do programu v Arduino, na konci algoritmu zajistit vyčkání, a to cca 30 ms, pokud tak nebude uskutečněno, bude samotné vykreslování viditelně zpožděno.

## 7.5 Filtrace hodnot

Pro tento typ aplikace se filtrace hodnot velice hodí, spíše tento typ aplikace filtraci vyžaduje pro zajištění smysluplné přesnosti. Cílem filtru je totiž odstranění signálu (šumu), který nesouvisí s přenášenou informací, ale naopak zhoršuje jeho kvalitu a přesnost. V okolí ultrazvukových senzorů vniká ve vzduchu mnoho rušivých vlivů, např. lokální zvýšení teploty změní rychlost zvuku, také změna vlhkosti vzduchu má vliv na vlastnosti ultrazvuku nebo dokonce i vítr – změní trajektorii vyslané ultrazvukové vlny a vlna putuje déle než je skutečná vzdálenost. Rušení také může vznikat i ve vedení a logických obvodech přijímače ultrazvuku. Vhodným výběrem filtru dokážeme velice zlepšit vlastnosti přípravku i jeho relativní přesnost. Kmitočtové selektivní (výběrové) filtry se rozdělují do čtyř kategorií; a to dolní propust, horní propust, pásmová propust a pásmová zadrž. Typově se pro tuto práci hodí použít filtr typu dolní propust (filtr, který nepropustí vyšší frekvence než je zvoleno), odfiltruje šumový signál o vyšších frekvencích a ideálně zůstane jen signál, který obsahuje pro nás důležitou informaci. Existuje mnoho filtrů tohoto typu, avšak není možné v této práci zhodnotit všechny možné filtry, proto byly zvoleny Čebyševův a Butterworthův filtr. Dále je v této práci ještě zvolen jednoduchý filtr typu klouzavý průměr, a to klouzavý průměr ze čtyř hodnot včetně současné.

Na následujícím grafu je zobrazeno statické měření devíti bodů, a to bez filtrace hodnot. Vodorovné osy jsou zvoleny v 10, 20 a 30 cm, svislé osy v -6, 7.5 a 25 cm. V dalších kapitolách bude zobrazeno jak se tato statická charakteristika mění (vyhlazuje) s použitím určitého filtru.

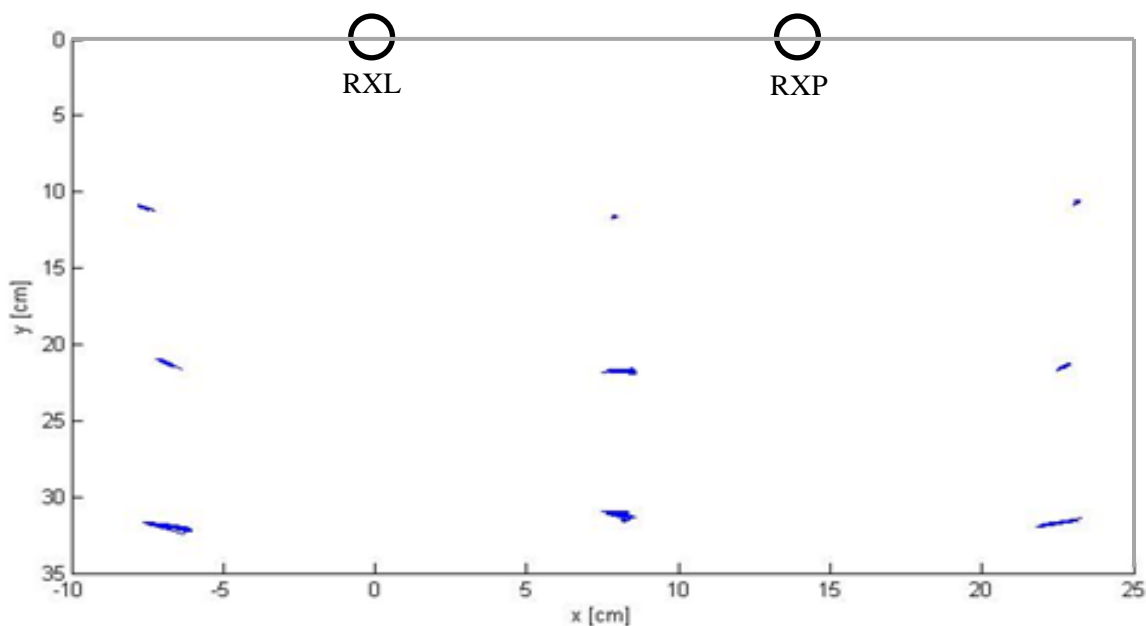


Obr. 7.3 : Statické měření bez filtrace hodnot

### 7.5.1 Klouzavý průměr

Jednoduchý filtr, který zohledňuje historii předchozích hodnot a současnou (aktuální naměřenou) hodnotu. Tento filtr obecně můžeme rozdělit do dvou kategorií, a to dle typu používaného průměru; buď použijeme standardní aritmetický průměr, nebo vážený průměr, u kterého máme možnost zohlednit stáří dat a tím jim zvětšit nebo zmenšit váhu (důležitost), kterou budou ovlivňovat výsledek. V této práci jsem zvolil klasický klouzavý průměr ze čtyř hodnot včetně současné, tudíž aktuální hodnota je sečtená se třemi předešlými hodnotami a vydělena čtyřmi (nebo-li počtem vzorků), viz vzorec 6.2.

$$x_{kp}(t) = \frac{1}{4} * (x(t) + x(t - 1) + x(t - 2) + x(t - 3)) \quad (6.2)$$



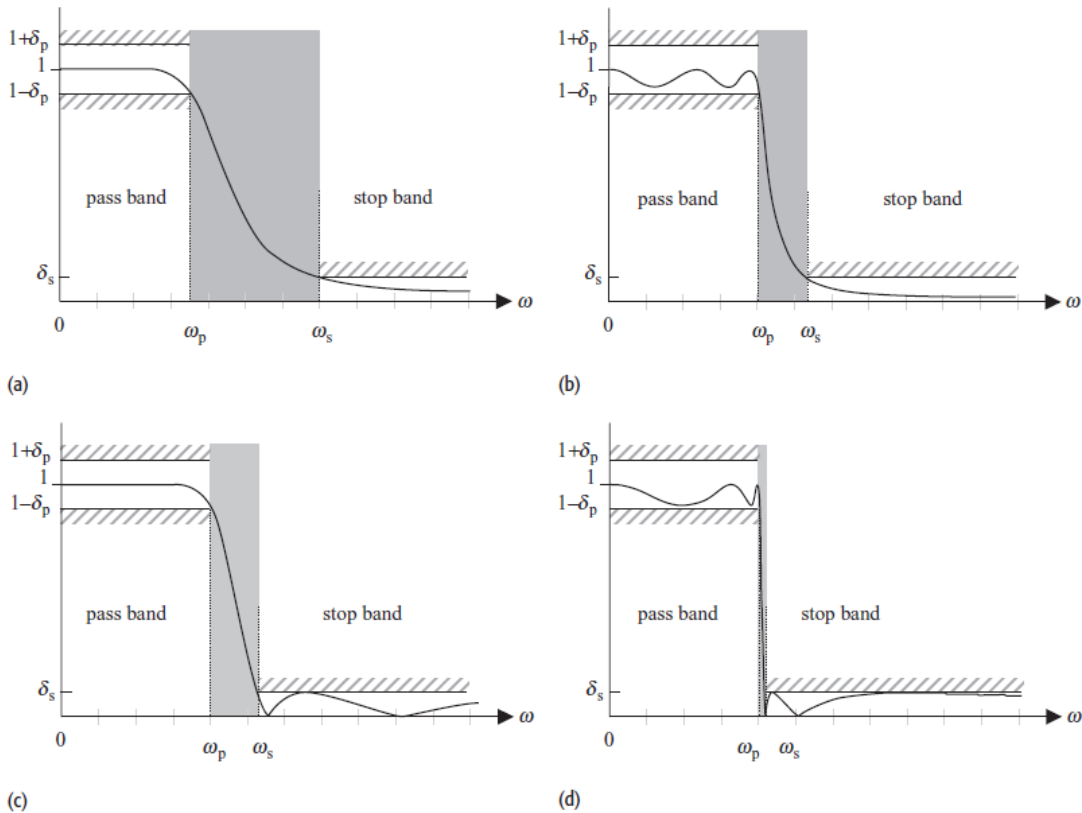
Obr. 7.4 : Statické měření s použitím filtru typu klouzavý průměr

### 7.5.2 Čebyševův filtr

Čebyševův filtr typu I omezuje šířku propustného pásma pomocí Čebyševovy aproximační funkce. Nevýhodou této aproximační funkce je výrazná velikost zvlnění v pásmu propusti, ale za to má velkou strmost mezi přechodem propustného pásma a pásmem potlačení. Frekvenční charakteristiky filtru typu doplní propusti jsou zobrazeny na Obr. 6.4. Důležité jsou mezní frekvence  $\omega_p$  a  $\omega_s$  – tyto frekvence ohraničují kde by signál měl být ještě propuštěn (ideálně bez útlumu, na Obr. 6.4 vyznačen jako  $\delta_p$ ) a kde by měl být signál ideálně úplně potlačen (úroveň šumového signálu je zde označeno  $\delta_s$ ). Prostor mezi těmito body mezních frekvencí je tzv. přechodová oblast. Přechodová charakteristika ideálního filtru typu dolní propust: vztah 6.3. [14]

$$H_{lp}(\omega) = \begin{cases} A, & |\omega| \leq \omega_c \\ 0, & |\omega| > \omega_c \end{cases} \quad (6.3)$$

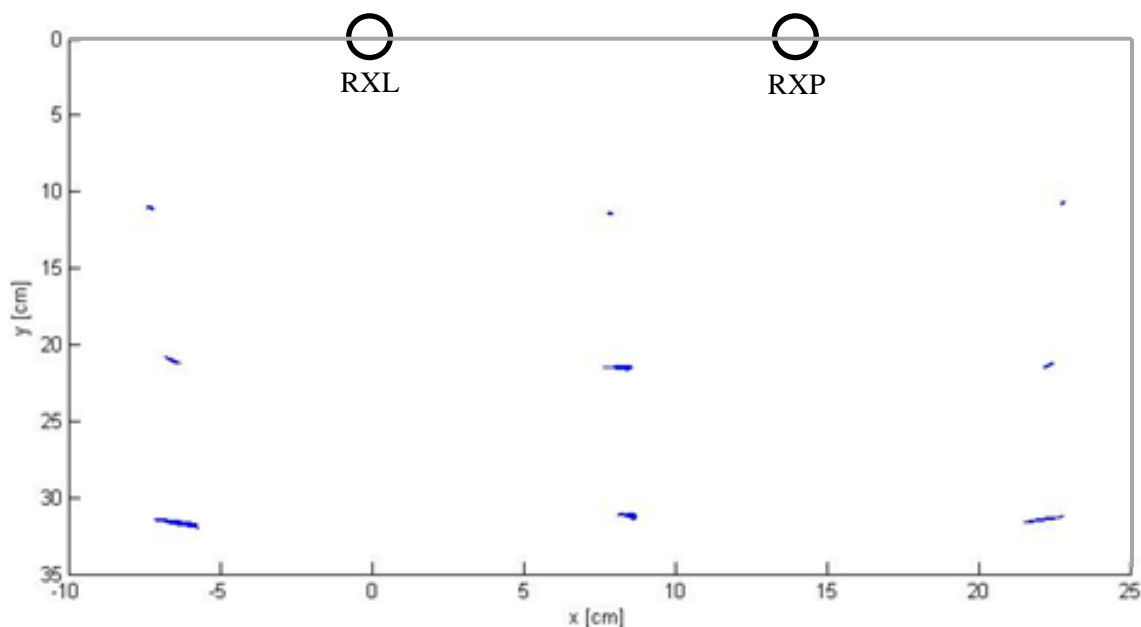
kde  $\omega_c$  je mezní frekvence filtru, pásmo propusti určuje interval  $|\omega| \leq \omega_c$ , přičemž pásmo zádrže je dáno intervalem  $\omega_c < |\omega| < \infty$ . [14]



**Obr. 7.5 :** Frekvenční charakteristiky filtrů typu dolní propust. (a) Butterworthův filtr; (b) I typ Čebyševova filtru; (c) II typ Čebyševova filtru; (d) eliptický filtr [14]

Na následujícím grafu (Obr. 6.5) je znázorněno statické měření z Obr. 6.2, a to při použití tohoto filtru. Nastavení parametrů :  $\omega_p = 0.03$ ,  $\omega_s = 0.12$ ,  $R_p = 0.2$ ,  $R_s = 50$ . Použité příkazy v Matlabu:

```
[n,Wp] = cheblord(Wp,Ws,Rp,Rs)
[b,a] = cheby1(n,Rp,Wp);
```



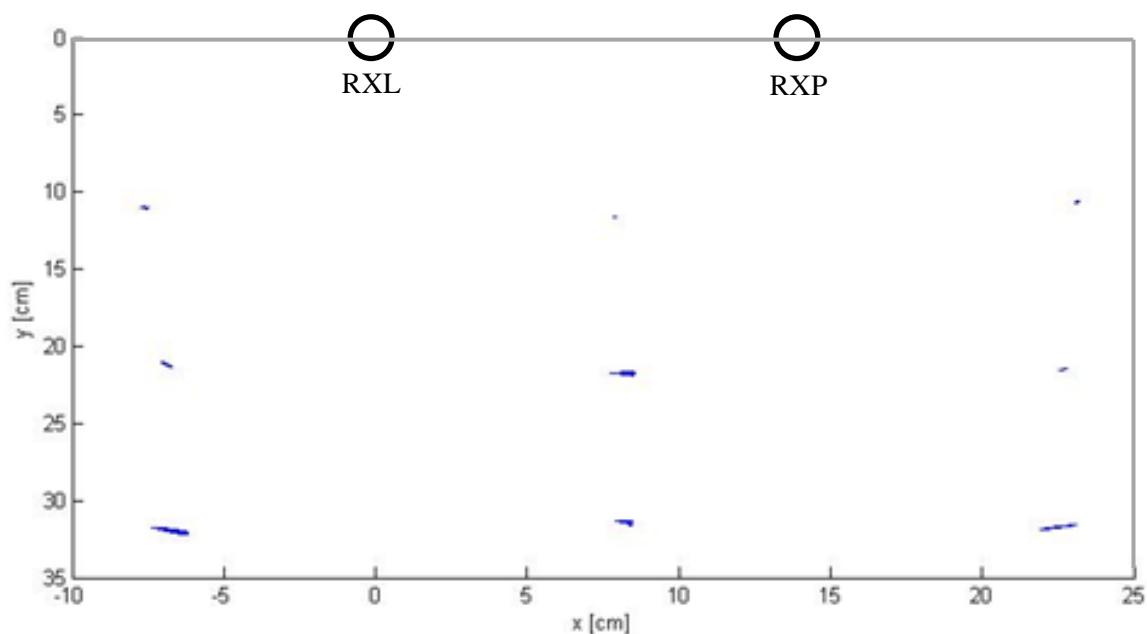
Obr. 7.6 : Statické měření s použitím Čebyševova filtru

### 7.5.3 Butterworthův filtr

Butterworthův filtr se vyznačuje oproti Čebyševovým filtrům menší strmostí přechodové oblasti, ale oproti tomu to kompenzuje viditelně plošším průběhem v propustné oblasti, to lze vidět na Obr. 6.4. [14]

Na následujícím grafu (Obr. 6.6) je znázorněno statické měření z Obr. 6.2, a to při použití tohoto filtru. Nastavení parametrů :  $\omega_p = 0.001$ ,  $\omega_s = 0.14$ ,  $R_p = 0.01$ ,  $R_s = 50$ . Použité příkazy v Matlabu:

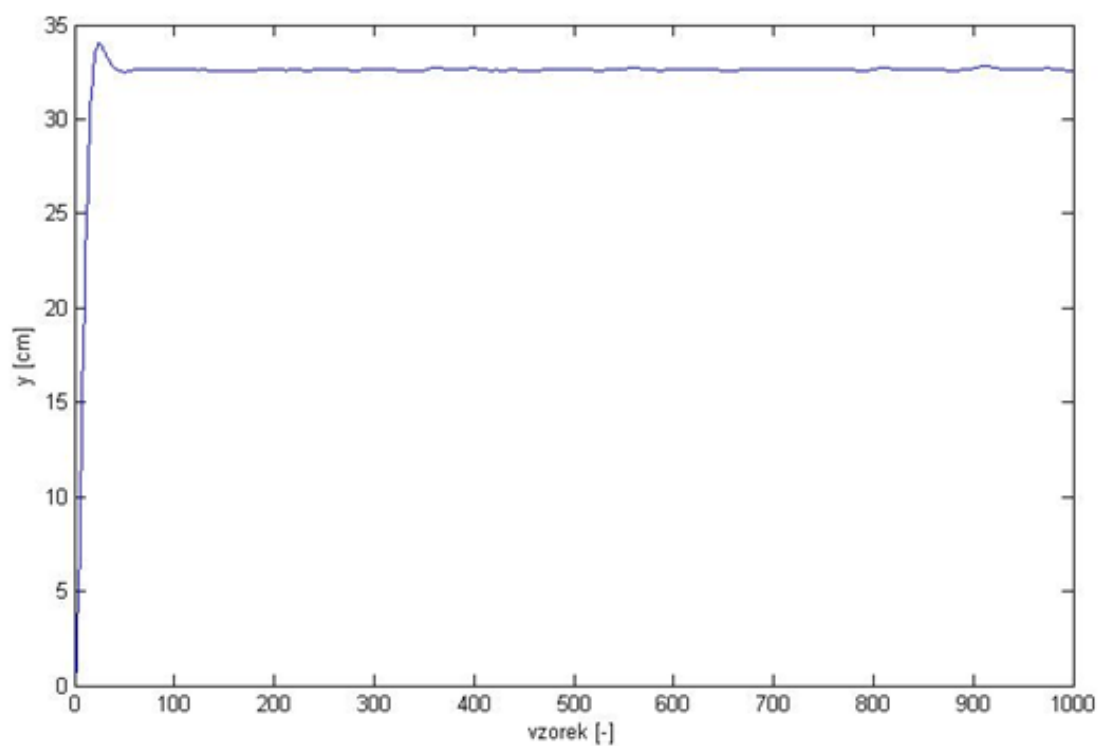
```
[N,wc] = buttord(Wp,Ws,Rp,Rs);  
[b,a] = butter(N,wc);
```



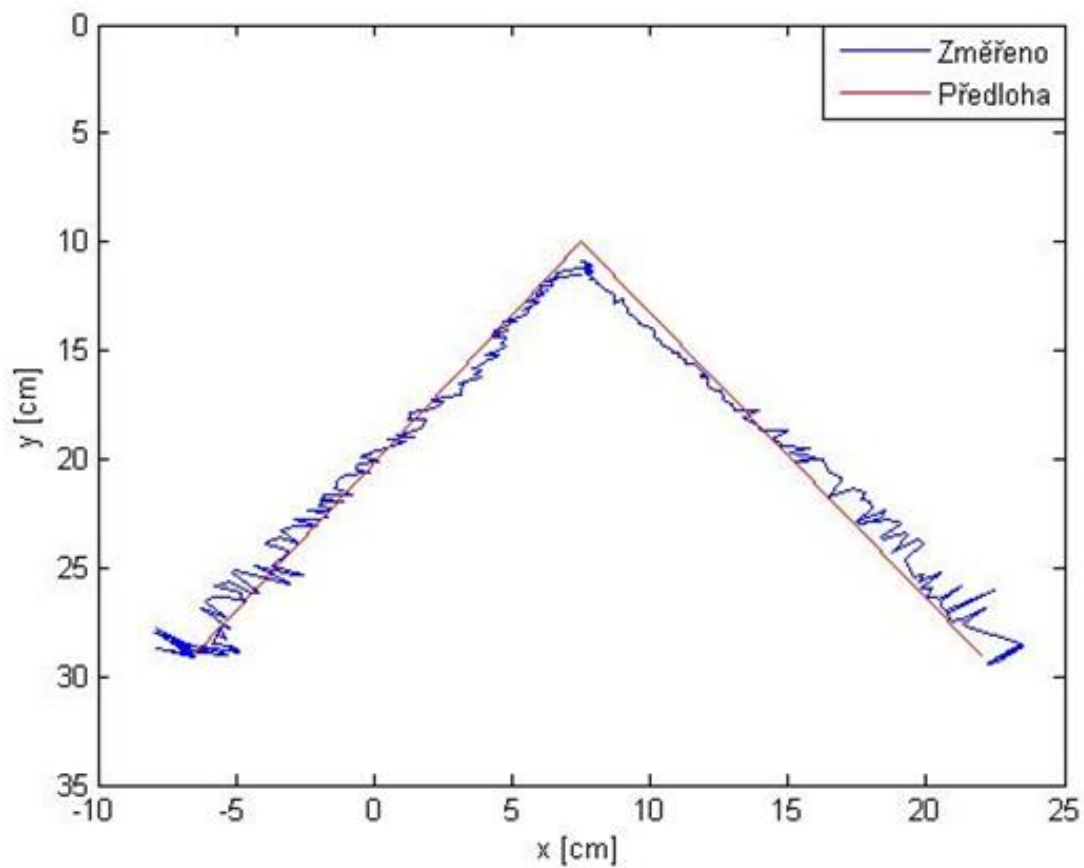
Obr. 7.7 : Statické měření s použitím Butterworthova filtru

## 7.6 Dynamické vlastnosti, zhodnocení

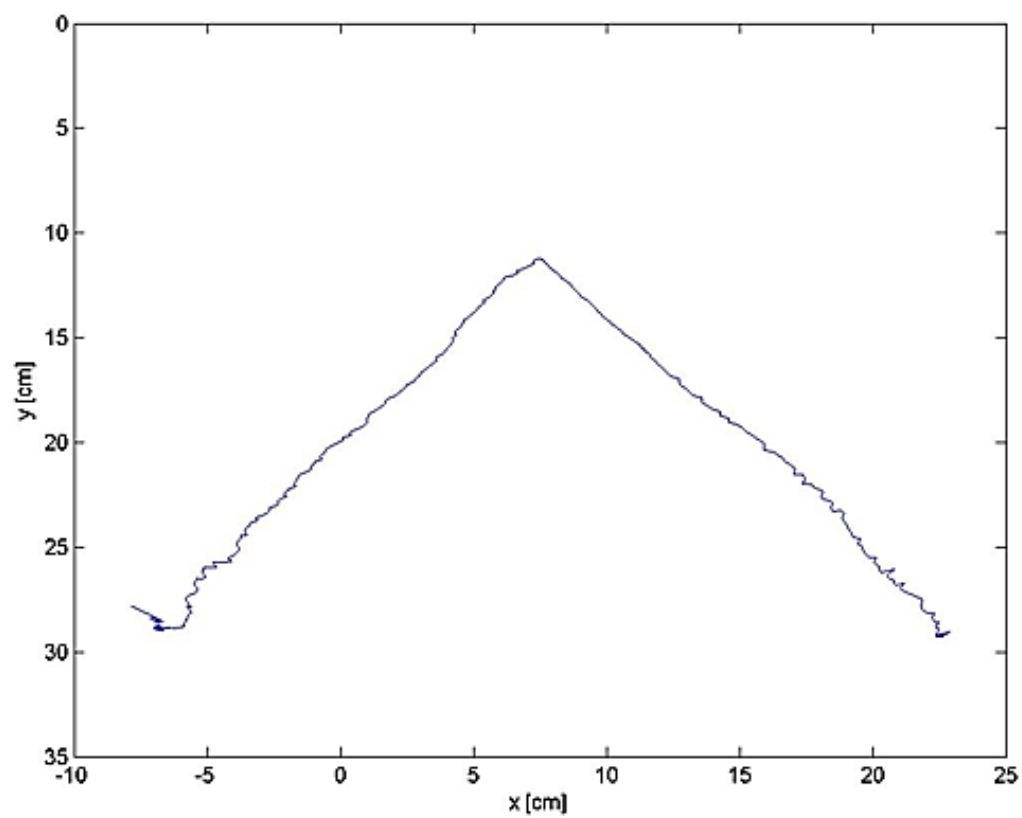
Jako předlohu pro měření dynamických vlastností jsem zvolil „stříšku“, kterou jsem nakreslil pomocí vytvořeného přípravku v této práci. Abych předešel ovlivněním hodnot vlastním pohybem vysílače (třes rukou, trochu jiný nebo pomalejší pohyb atd.), jsou používány vždy ty samé hodnoty. Přičemž hodnoty souřadnic jsem dopočítával v Matlabu, před výpočtem těchto souřadnic jsem aplikoval již zmíněné filtry a na následujících grafech lze vidět rozdíly při použití a typu použitého filtru. Při použití Čebyševova nebo Butterworthova filtru velice záleží na nastavení parametrů; při nastavení velmi malých hodnot frekvencí se data „zdeformují“ – bod, který ve skutečnosti odpovídá bodu  $[-5;30]$ , je po vyfiltrování hodnot vyhodnocen v bodě  $[0;20]$ , v opačném případě, tedy při nastavení vysokých frekvencí nebudou souřadnice, resp. trajektorie dostatečně vyhlazena. Dalším problémem při nastavení velmi malých hodnot frekvencí je velice dlouhý nástup vyregulování vzdálenosti (viz. Obr. 6.12). Při každém použití těchto filtrů se musí zajistit, pokud to tedy aplikace vyžaduje, odstranění prvních pár hodnot (počet záleží na nastavení filtru). Při neodstranění těchto hodnot by velice zkreslili data, důvodem je dynamický náběh filtru, toto lze vidět na začátku grafu (Obr. 6.7.). V této práci by neodstranění těchto hodnot velice zkreslilo výslednou trajektorii.



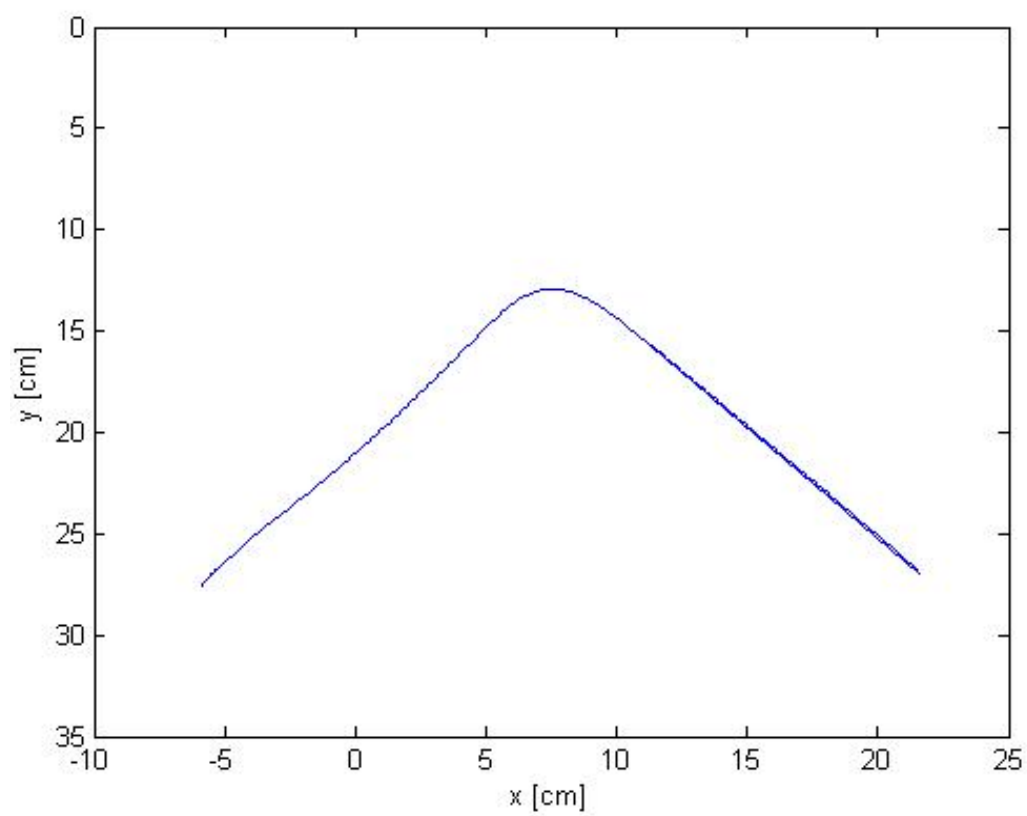
**Obr. 7.8 :** Vyregulování souboru hodnot jedné statické vzdálenost



**Obr. 7.9 :** Nakreslená střecha bez filtrace, se zobrazením skutečné předlohy

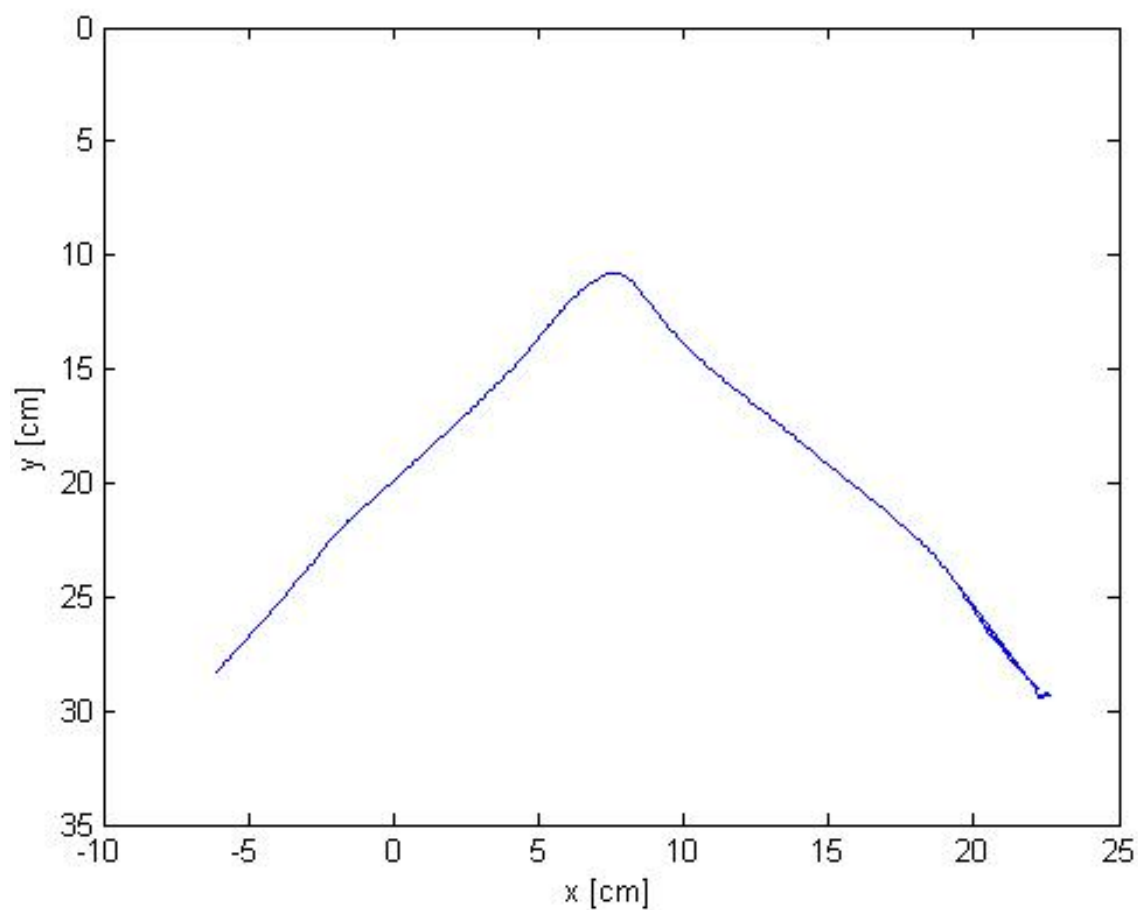


**Obr. 7.10 :** Nakreslená střecha po použití klouzavého průměru

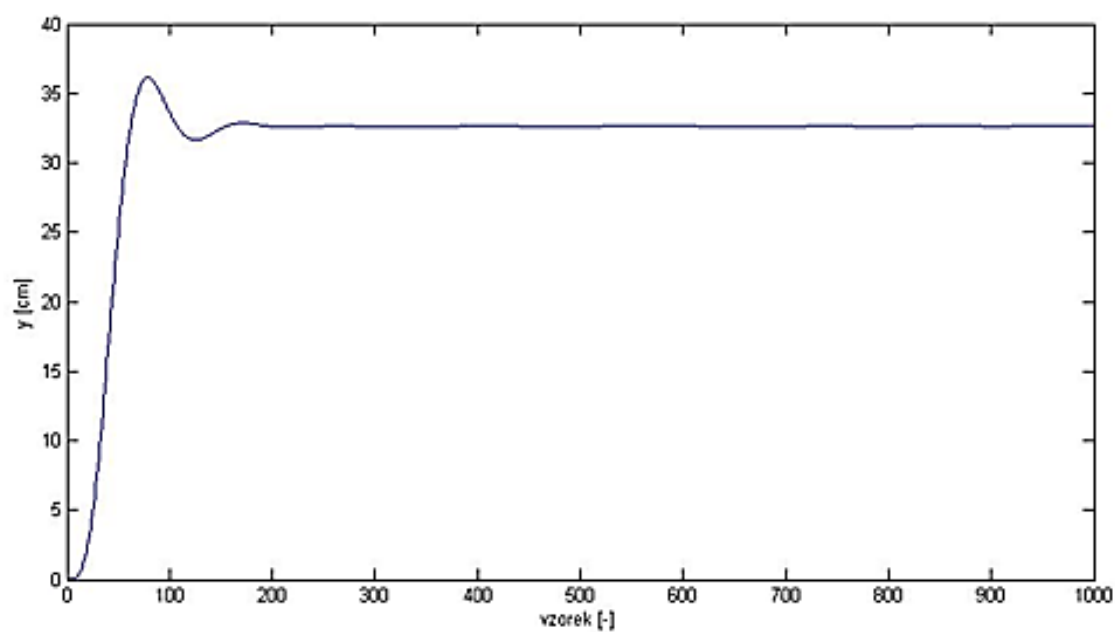


**Obr. 7.11 :** Nakreslená střecha po vyfiltrování hodnot Butterworthovým filtrem

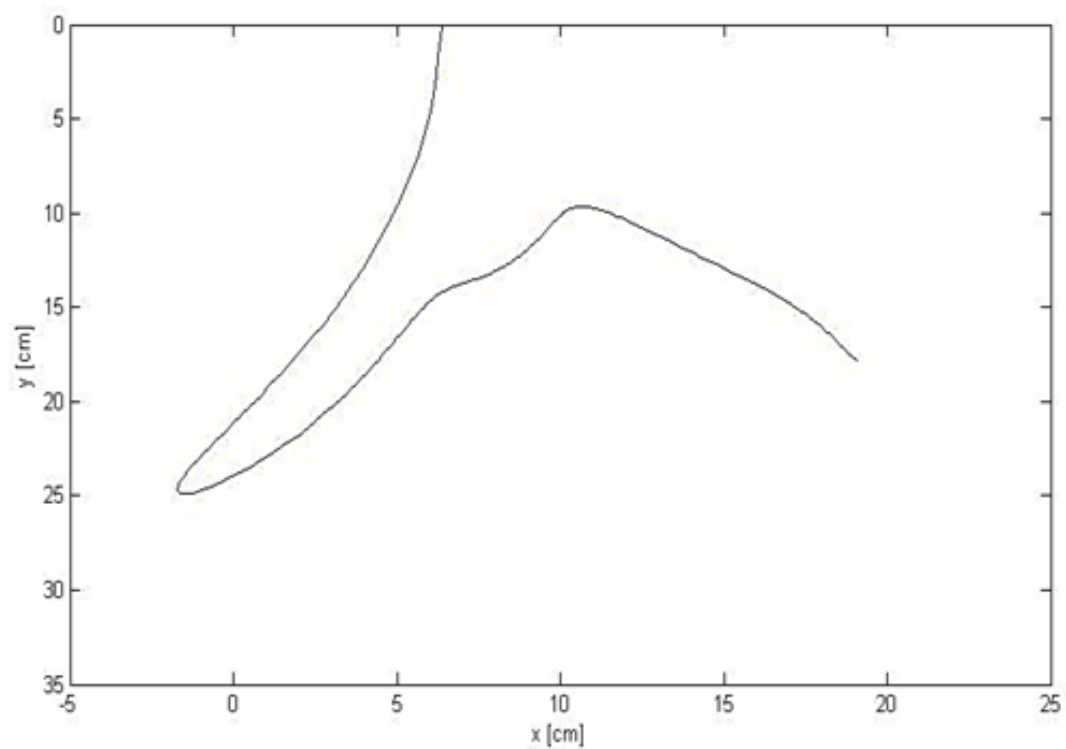




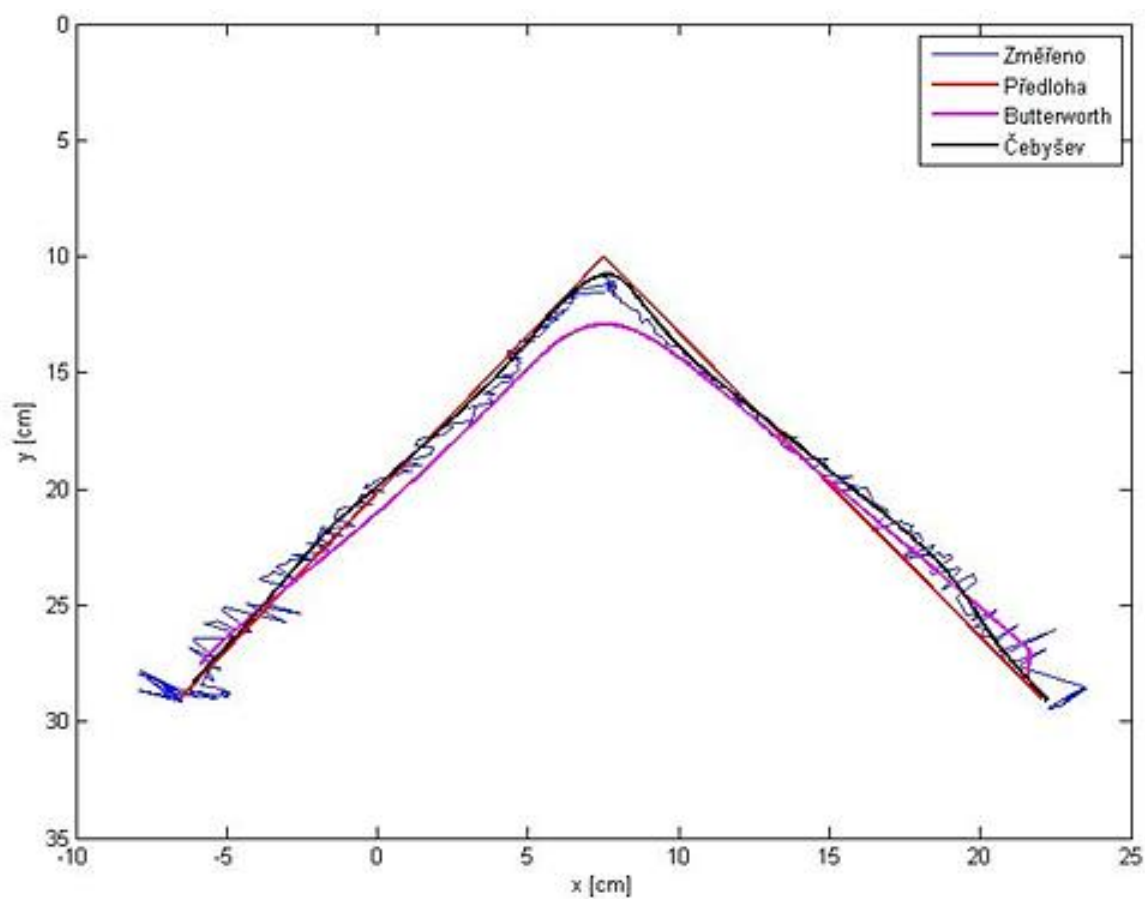
**Obr. 7.12 :** Nakreslená střecha po vyfiltrování hodnot Čebyševovým filtrem



**Obr. 7.13 :** Dlouhý nástup při špatně nastaveném filtru



**Obr. 7.14 :** Špatně nastavený filtr (Butterworthův) – nastavené velice malé frekvence



**Obr. 7.15 :** Srovnání použitých filtrů s předlohou

## 8 ZHODNOCENÍ MĚŘENÍ

### 8.1 Směrové ultrazvukové senzory

V této práci byly k měření použity ultrazvukové senzory US-100 a SRF08, na grafech (v kap. 5.4) lze vidět, že ultrazvukový senzor US-100 byl jednoznačně přesnější než SRF08, i když při zvyšování vzdálenosti se chyba měření u senzoru US-100 zvětšovala, maximální naměřená chyba byla 0,8475 cm. Chyba měření ultrazvukového senzoru SRF08 dosahovala v některých místech více než 2 cm. S podobnými výsledky dopadlo také měření při vychýlení překážky v  $\pm 15^\circ$  od osy senzoru; při měření senzoru US-100 byla maximální naměřená chyba v  $\pm 15^\circ$  1,3 cm, oproti tomu při měření senzoru SRF08 byla maximální chyba 2,2258 cm, srovnání závislostí chyb při vychýlení překážky  $\pm 15^\circ$  a  $0^\circ$  od osy je na grafech Obr. 5.6 a Obr. 5.14. Z těchto naměřených hodnot lze konstatovat, že senzor SRF08 měří vzdálenost s dvojnásobnou velikostí chyby, než senzor US-100.

V datasheetu pro senzor US-100 [2] uvádějí, že zorný úhel je menší než  $15^\circ$ . Při měření vyzařovací charakteristiky jsem zjistil, že tento úhel se pohybuje okolo  $32^\circ$  na straně vysílače a asi  $26^\circ$  na straně přijímače, celkový zorný úhel je tedy cca  $60^\circ$ . Avšak při takovémto velkém vychýlení se mnohonásobně zmenší přesnost. Minimální detekující vzdálenost tohoto senzoru odpovídá hodnotě uváděné v datasheetu (2 cm). Maximální chyba při měření opakovatelnosti byla 0,0906 cm, průměrná 0,0891 cm. Při vychýlení překážky o  $\pm 15^\circ$  při vzdálenosti 5 cm začínala chyba na cca 0,3 cm na obou stranách, přitom ve stejné vzdálenosti při nevychýlené překážce začínala chyba v 0,1 cm (ve vzdálenosti 10 cm dokonce desetiny mm), se zvyšující se vzdáleností chyba mírně rostla. Přitom při zvyšování vzdálenosti v  $\pm 15^\circ$  byla vysoká chyba už při vzdálenosti 15 cm (a to až 1 cm), při vzdálenosti 30 cm byly velikosti chyb srovnatelné s nevychýlenou překážkou. Velikost nejistoty typu A, jak lze vidět na grafu Obr. 5.7, je nezávislá na vychýlení.

Oproti menší přesnosti vůči senzoru US-100 se senzor SRF08 vyznačuje větším zorným úhlem, při měření vyzařovací charakteristiky se zorný úhel pohyboval okolo  $35^\circ$  na straně vysílače a asi  $31^\circ$  na straně přijímače, celkový zorný úhel je tedy cca  $65-70^\circ$ . Chyba měření senzoru SRF08 velice kolísala v celém měřeném rozsahu, ve vzdálenosti 15 cm dosáhla dokonce chyba měření velikosti 2,4 cm. Nejistota měření typu A měla podobný trend, jako chyba měření a dokonce o řád vyšší než senzor US-100. Velikosti chyb při vychýlení v  $\pm 15^\circ$  a srovnání s nevychýlenou překážkou je tento rozdíl viditelně méně znatelný (velikost chyby je u tohoto senzoru méně závislá na vychýlení) než u senzoru US-100; velikosti chyby při vzdálenosti 15 cm je srovnatelná s vychýlením v  $+15^\circ$  (při vychýlení v  $-15^\circ$  je poloviční chyba než u nevychýlené

překážky). Pokud přesáhneme vychýlení více jak  $15^\circ$ , přesnost senzoru se rapidně snižuje u obou senzorů. Minimální velikost detekovaného předmětu senzorem US-100 jsem zjistil 4 mm, senzorem SRF08 je minimální detekovatelná velikost předmětu 3 mm.

**Tab. 2 :** Maximální hodnoty vypočítaných chyb, nejistot a rozptylu

	Chyba [cm]	Nejistota [cm]	Rozptyl [cm]
US-100	0.8475	0.0049	0.0478
SRF08	2.2258	0.1097	2.3826
PVDF 80 kHz	0.4365	0.0138	0.3829

**Tab. 3 :** Průměrné hodnoty vypočítaných chyb, nejistot a rozptylu

	Chyba [cm]	Nejistota [cm]	Rozptyl [cm]
US-100	0.3165	0.0023	0.0132
SRF08	0.8407	0.0271	0.4055
PVDF 80 kHz	0.1512	0.0040	0.0547

## 8.2 Všesměrové ultrazvukové senzory

Jejich velká a zásadní výhoda spočívá v tom, že nezáleží na vzájemném úhlu mezi přijímačem a vysílačem, popř. mezi ultrazvukovým senzorem a překážkou. Jak je dokázáno na grafech směrových senzorů Obr. 5.6 a 5.14, velikost chyby se zvyšuje s vychylováním od osy snímače. Toto však u všesměrových snímačů neplatí, překážku (popř. vysílač) dokážeme detekovat v jakémkoliv úhlu. Omezení detekce překážky tedy závisí jen na maximální změřitelné vzdálenosti. V kap. 6 je provedeno měření použitého všesměrového válcového ultrazvukového převodníku PVDF 80kHz.

Na grafu charakteristiky snímače (Obr. 6.1) lze vidět, že oproti předchozím senzorům US-100 a SRF08 tento senzor velmi dobře kopíruje nastavenou hodnotu (měří relativně přesně nastavenou vzdálenost), a nevznikají zde žádné větší výkyvy v chybovosti jako v předchozích případech. Přesnost se samozřejmě zmenšuje se zvyšující se vzdáleností, toto viditelně nastává v hodnotě 25 cm, v této vzdálenosti je chyba 0,25 cm (viz. Obr. 6.3), maximální změřená chyba v pracovní oblasti je 0,4365 cm (pro senzor SRF08 byla 2,2258 cm), do vzdálenosti 25 cm je chyba skoro konstantní a menší než 0,2 cm. Podobně tomu bylo i u nejistoty typu A (Obr. 6.4), ale ve vzdálenosti 33 cm tato nejistota rapidně vzrostla, a to až na 0,014 cm.

Jednoznačně je tento ultrazvukový senzor PVDF 80 kHz přesnější, spolehlivější a využitelnější díky svému neomezenému zornému úhlu než ostatní ultrazvukové senzory použité v této práci. V měřené pracovní oblasti je tento senzor velice přesný, dle mého názoru, při vzdálenosti 30 cm s absolutní chybou o velikosti 0,43 cm

a průměrnou chybou 0,1512 je u ultrazvukových snímačů velmi nízká. Díky své vyzařovací charakteristice, resp. neomezenému zornému úhlu a dobré přesnosti se tento senzor hodí na podobné úlohy jako je popsáno v této práci - 2D polohování, popř. identifikace v plošném prostoru, ultrazvukové pero, interaktivní tabule a další.

Použité směrové senzory se hodí spíše ke zjišťování orientační vzdálenosti, popřípadě k detekci přítomnosti objektů. Jejich výhoda spočívá v jednoduché programové ovladatelnosti, jednoduchosti, kompaktnosti a také např. ceně.

## 9 ZÁVĚR

Cílem první části této bakalářské práce bylo seznámit se s měřením směrových a všesměrových ultrazvukových senzorů, s jejich vlastnostmi a přesností při měření vzdálenosti.

Principy používané k měření vzdáleností pomocí ultrazvuků jsou rozebrány v kap. 4. Popis zapojení senzorů k mikrokontroléru a následná implementace algoritmu pro měření vzdálenosti je popsána v kap. 5, v této kapitole je také popsáno zpracování naměřených hodnot v Matlabu. Použité směrové ultrazvuky US-100 a SRF08 byly proměřeny ve zvolené pracovní oblasti; v této pracovní oblasti byly změřeny absolutní chyby, nejistoty typu A a velikost rozptylů v závislosti na nastavené vzdálenosti nebo vychýlení od osy senzoru. Senzor SRF08 lze nastavit do tří režimů zobrazování změřené vzdálenosti, při zvolení zobrazování v centimetrech se měřená hodnota zaokrouhlovala na celé centimetry. Tato funkce pro toto měření nemělo význam (zaokrouhlování znehodnocuje přesnost senzoru), v algoritmu pro obsluhu tohoto snímače bylo zvoleno zobrazování hodnot času (milisekund), tyto hodnoty času byly následně vynásobeny rychlostí šíření zvuku a tím získání přesnějších hodnoty. Dále byl proměřen všesměrový ultrazvukový převodník PVDF 80 kHz použitý v druhé části této práce. Zhodnocení vlastností a přesnosti těchto ultrazvukových senzorů je provedeno v kap. 8.

Druhá část této práce spočívá ve vytvoření a zprovoznění přípravku pro lokalizaci ultrazvukového vysílače v rovině, schématické znázornění přípravku je na Obr. 7.1. Kombinací jednoho vysílače a dvou ultrazvukových přijímačů a pomocí implementace algoritmu trilaterace jsem dokázal získat souřadnice vysílače a následným spojením těchto souřadnic vykreslit jeho trajektorii v reálném čase na počítači. Trajektorie je vykreslována pomocí programu Matlab. Obsluha všesměrových ultrazvukových senzorů, je jako v první části práce, pomocí mikrokontroléru Arduino. V této práci jsou popsány dvě možnosti výpočtu souřadnic, a to pomocí Arduina a Matlabu, jako vhodnější jsem zvolil provádění výpočtů souřadnic Matlabem, a to kvůli lepší přesnosti. Dalším krokem bylo zajištění filtrace hodnot. Byly zvoleny filtry typu Klouzavý průměr, Čebyšev a Butterworth. Srovnání bez použití a při použití těchto filtrů je provedeno v kap. 7.5 a tomu odpovídající obrázky. Nejlepší průběh byl získán při použití Čebyševova filtru, avšak i při použití jednoduchého filtru typu Klouzavý průměr se daná trajektorie velice zlepšila. Použité ultrazvukové senzory v první části se pro tuto aplikaci nehodí a to kvůli špatné přesnosti a malému zornému úhlu.

Dalšími kroky pro zajištění lepší přesnosti a zlepšení vlastností bych navrhoval hlavně zavedení stínění přijímačů proti elektromagnetickému a elektrickému poli. Oddělení elektrického obvodu vysílače od přijímačů, aby nevznikal elektrický rozkmit,

kterým je způsobena posunutá pracovní oblast. S tím související zavedení synchronizačního impulsu pro zjištění času vyslání ultrazvukové vlny (nebo-li indikace pro začátek počítání času od vyslání vlny), a to například pomocí laserové nebo infračervené diody; rychlost světla je několikanásobně vyšší než rychlost zvuku, můžeme tedy časové zpoždění mezi vysláním a přijmutím světelného paprsku zanedbat a považovat ho jako skutečný čas vyslání ultrazvukové vlny. Dále použití výkonnějšího mikrokontroléru s přesnějším interním čítačem a zapouzdření obou přijímačů pro zajištění odstranění druhotných odrazů ultrazvukových vln od předmětů.

# LITERATURA

- [1] Ultrasonic Sonar Module: Hardware Manual Rev 1r0. E-GIZMO MECHATRONIX. [online]. 2011 [cit. 2013-12-30]. Dostupné z: <http://www.e-gizmo.com/KIT/images/ultrasonicsonar/ultrasonic%20sonar%20module%201r0.pdf>
- [2] US-100 ultrasonic sensor module temperature compensated. EBAY INC. [online]. [cit. 2013-12-30]. Dostupné z: <http://www.ebay.com/itm/US-100-ultrasonic-sensor-module-temperature-compensated-/251076829817>
- [3] SRF08 Ultra sonic range finder: Technical Specification. DEVANTECH LTD (ROBOT ELECTRONICS). [online]. [cit. 2013-12-30]. Dostupné z: <http://www.robot-electronics.co.uk/htm/srf08tech.shtml>
- [4] SRF08 Ultra Sonic Range Finder. [online]. Duben 2009, Květen 2012 [cit. 2013-12-30]. Dostupné z: <http://playground.arduino.cc/Main/SonarSrf08>
- [5] Arduino. [online]. [cit. 2013-12-30]. Dostupné z: <http://arduino.cc/>
- [6] ATmega328: Key parameters for ATmega328. ATMEL CORPORATION. [online]. Leden 2013 [cit. 2013-12-30]. Dostupné z: <http://www.atmel.com/devices/atmega328.aspx?tab=parameters>
- [7] Arduino Duemilanove. [online]. [cit. 2013-12-30]. Dostupné z: <http://arduino.cc/en/Main/arduinoBoardDuemilanove>
- [8] ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY. *Měření fyzikálních veličin: návody do laboratorních cvičení*. Brno, 19. února 2013. [cit. 2013-12-30].
- [9] J. Lee, S. H. Hong, Y. Chung, Robust Tracking of Ultrasonic Pen for Handwriting Digitizer Applications. [online]. Srpen 2007. [cit. 2013-12-30]. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4341591>
- [10] ÚSTAV MATEMATIKY. *Matematika 3*. 2005. vyd. Brno, 2002, 2005 [cit. 2013-12-30].
- [11] KANČO, V. Ultrazvukový měřič vzdálenosti: bakalářská práce. Brno: FEKT VUT v Brně, 2008. 45 s., 8 příl. [cit. 2013-12-30].
- [12] Telecomunicacions: Sonar. In: [online]. 16.02.2014 [cit. 2014-05-26]. Dostupné z: <http://telecombar.blogspot.cz/2014/02/sonar.html>
- [13] SOLARBOTICS LTD. *Solarbotics* [online]. 2001, 17.11.2008 [cit. 2014-05-26]. Dostupné z: <https://solarbotics.com/product/40320/>



- [14] MEASUREMENT SPECIALTIES, Inc. *80 kHz Cylindrical Ultrasound Transducer*. 05.12.2009 [cit. 2014-05-26]. Dostupné z: [http://www.meas-spec.com/product/t\\_product.aspx?id=2492](http://www.meas-spec.com/product/t_product.aspx?id=2492)
- [15] MANDAL, Mrinal Kr a Amir ASIF. *Continuous and discrete time signals and systems*. New York: Cambridge University Press, 2007, xiv, 865 p. ISBN 05-218-5455-5 [cit. 2014-05-26]
- [16] BEJČEK, L., ČEJKA, M., REZ, J., GESCHEIDTOVÁ, E., STEINBAUER, M. *Měření v elektrotechnice. Měření v elektrotechnice*. VUT- FEKT, 2002. [cit. 2014-05-26]
- [17] [Www.ultrazvuk.cz](http://www.ultrazvuk.cz): stránky o ultrazvuku [online]. 2007. vyd. [cit. 2014-05-26]. Dostupné z: <http://www.ultrazvuk.cz/index.php>
- [18] Piezoelektrický jev. In: [online]. 23.08.2008 [cit. 2014-05-26]. Dostupné z: <http://fyzmatik.pise.cz/573-piezoelektricky-jev.html>

## **SEZNAM PŘÍLOH**

**Příloha A** – Tabulky vypočtených hodnot

**Příloha B** – CD-ROM

# A TABULKY VYPOČTENÝCH HODNOT

**Tab. 4 :** Tabulka vypočtených hodnot na senzoru US-100

Vzdálenost [cm]	Chyba [cm]	Nejistota [cm]	Rozptyl [cm]
3	0.5668	0.0013	0.0036
5	0.0676	0.0016	0.0052
8	0.1186	0.0016	0.0050
10	0.0124	0.0018	0.0066
12	-0.0801	0.0015	0.0042
13	-0.0743	0.0015	0.0044
15	0.2230	0.0024	0.0118
16	0.3616	0.0009	0.0017
18	0.7519	0.0022	0.0099
20	0.7136	0.0015	0.0044
23	0.5194	0.0049	0.0478
25	0.8475	0.0016	0.0048
27	-0.0055	0.0032	0.0202
29	0.2055	0.0047	0.0433
31	0.5260	0.0043	0.0366
33	0.3098	0.0010	0.0022

**Tab. 5 :** Tabulka vypočtených hodnot na senzoru US-100 při vychýlení +15°

<b>Vzdálenost [cm]</b>	<b>Chyba [cm]</b>	<b>Nejistota [cm]</b>	<b>Rozptyl [cm]</b>
5	0.2795	0.0017	0.0055
10	0.4177	0.0015	0.0046
15	1.0722	0.0022	0.0095
20	1.2132	0.0014	0.0038
25	1.2426	0.0017	0.0056
30	0.9700	0.0015	0.0043

**Tab. 6 :** Tabulka vypočtených hodnot na senzoru US-100 při vychýlení -15°

<b>Vzdálenost [cm]</b>	<b>Chyba [cm]</b>	<b>Nejistota [cm]</b>	<b>Rozptyl [cm]</b>
5	0.1693	0.0014	0.0041
10	0.3075	0.0018	0.0063
15	0.8521	0.0022	0.0099
20	0.9307	0.0056	0.0628
25	1.1913	0.0023	0.0109
30	0.8641	0.0021	0.0090

**Tab. 7 :** Tabulka vypočtených hodnot na senzoru SRF08

<b>Vzdálenost [cm]</b>	<b>Chyba [cm]</b>	<b>Nejistota [cm]</b>	<b>Rozptyl [cm]</b>
3	0.0771	0.0008	0.0001
5	0.0052	0.0000	0.0000
8	1.1676	0.0000	0.0000
10	0.3888	0.0000	0.0000
12	1.3494	0.0829	1.3607
13	0.0459	0.0179	0.0635
15	2.2258	0.1097	2.3826
16	1.0605	0.0642	0.8155
18	0.0060	0.0000	0.0000
20	0.4852	0.0000	0.0000
23	1.6168	0.0674	0.8991
25	0.5076	0.0000	0.0000
27	0.7264	0.0269	0.1432
29	2.2218	0.0645	0.8231
31	1.0264	0.0000	0.0000
33	0.5400	0.0000	0.0000

**Tab. 8 :** Tabulka vypočtených hodnot na senzoru SRF08 při vychýlení +15°

<b>Vzdálenost [cm]</b>	<b>Chyba [cm]</b>	<b>Nejistota [cm]</b>	<b>Rozptyl [cm]</b>
5	0.0740	0.0000	0.0000
10	1.0090	0.0826	1.3495
15	1.8364	0.0141	0.0392
20	1.6159	0.0639	0.8079
25	1.4006	0.0587	0.6828
30	2.4628	0.3830	0.2908

**Tab. 9 :** Tabulka vypočtených hodnot na senzoru SRF08 při vychýlení -15°

<b>Vzdálenost [cm]</b>	<b>Chyba [cm]</b>	<b>Nejistota [cm]</b>	<b>Rozptyl [cm]</b>
5	0.4650	0.0046	0.0043
10	0.8360	0.0000	0.0000
15	1.1451	0.0724	1.0373
20	1.8770	0.0347	0.2385
25	0.9763	0.0221	0.0969
30	1.7235	0.0631	0.7894

**Tab. 10 :** Tabulka vypočtených hodnot na senzoru PVDF 80 kHz

<b>Vzdálenost [cm]</b>	<b>Chyba [cm]</b>	<b>Nejistota [cm]</b>	<b>Rozptyl [cm]</b>
10	0.1514	0.0023	0.0106
12	0.0932	0.0020	0.0077
13	0.1755	0.0010	0.0020
15	-0.0714	0.0022	0.0097
16	0.0709	0.0014	0.0042
18	0.0479	0.0023	0.0110
20	-0.0750	0.0027	0.0142
23	0.0635	0.0063	0.0788
25	0.2465	0.0063	0.0790
27	0.4188	0.0054	0.0582
29	0.4365	0.0046	0.0425
31	0.1762	0.0022	0.0100
33	0.2311	0.0138	0.3829