

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

BAKALÁŘSKÁ PRÁCE

Plzeň, 2013

Michal Jirovský

PROHLÁŠENÍ

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne

.....

vlastoruční podpis

Poděkování

Touto cestou bych rád poděkoval vedoucímu bakalářské práce, panu Ing. Miroslavu Flídrovi, Ph.D., za výborné vedení práce a ochotu při poskytování cenných rad a věcných připomínek, které vedly k vypracování této práce. Rád bych také poděkoval své rodině za podporu během studia.

Abstrakt

Tato práce se zabývá řízením diferencially poháněného robotu. V první části je uveden popis zvoleného robotu a vybraných čidel, kterými je robot osazen. Ve druhé části je vytvořen matematický model robotu. Ten je poté použit při implementaci řídicího algoritmu, jehož úkolem je sledování náhodně generované trajektorie.

Klíčová slova

Diferencially řízený robot, Arduino, MATLAB, sledování trajektorie

Abstract

This work deals with control of differentially driven robot. In the first part, there is a description of the selected robot and sensors, which contains the robot. In the second part is created a mathematical model of robot. This is used to implement the control algorithm, whose task monitor randomly generated trajectories.

Keywords

Differential driven robot, Arduino, MATLAB, tracking trajectory

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 3 |
| 2 | Návrh hardwaru mobilního robotu | 4 |
| 2.1 | Platforma Arduino | 4 |
| 2.1.1 | Vývojový kit Arduino UNO | 5 |
| 2.1.2 | Vývojové prostředí | 5 |
| 2.2 | Vozítko Dagu Wild Thumper 4WD | 7 |
| 2.3 | Arduino pro řízení motorů | 8 |
| 2.4 | Sběrnice I2C | 9 |
| 2.4.1 | Co je I2C sběrnice | 10 |
| 2.4.2 | Princip komunikace | 10 |
| 2.5 | Čidla | 12 |
| 2.5.1 | Ultrazvukové čidlo | 12 |
| 2.5.2 | Servomotor | 14 |
| 2.5.3 | Optický enkodér | 15 |
| 2.5.4 | Inerciální měřicí jednotka | 16 |
| 2.5.5 | Magnetometr | 19 |
| 2.5.6 | Bezdrátový komunikační modul xBee | 21 |
| 2.5.7 | Logický úrovnňový převodník | 22 |
| 2.6 | Kompletace hardwaru | 23 |
| 3 | Matematický model a řízení mobilního robotu | 27 |
| 3.1 | Odvození kinematických rovnic robotu | 29 |
| 3.2 | Sledování referenční trajektorie | 31 |
| 3.3 | Simulace řízení | 36 |
| 4 | Závěr | 41 |

Seznam obrázků

| | | |
|------|--|----|
| 2.1 | Arduino UNO | 6 |
| 2.2 | Vývojové prostředí Arduino | 6 |
| 2.3 | Vývojové prostředí Arduino | 6 |
| 2.4 | Konstrukce Wild Thumper 4WD | 8 |
| 2.5 | Kontrolér pro řízení robotu | 9 |
| 2.6 | Schéma připojených zařízení přes I2C | 10 |
| 2.7 | Časový průběh přenosu dat na I2C | 12 |
| 2.8 | Použité ultrazvukové čidlo - Devantech SRF10 | 13 |
| 2.9 | Vyzařovací úhel pulzu | 13 |
| 2.10 | Připojení ultrazvukového čidla | 14 |
| 2.11 | Připojení servomotoru a použité servo | 15 |
| 2.12 | Optický enkodér - QTR-1RC | 15 |
| 2.13 | Princip inkrementálního enkodéru | 16 |
| 2.14 | Připojení enkodéru | 17 |
| 2.15 | Inerciální měřicí jednotka - MPU6050 | 17 |
| 2.16 | Princip měření akcelerometru | 18 |
| 2.17 | Princip měření gyroskopem | 19 |
| 2.18 | Připojení IMU | 20 |
| 2.19 | Magnetometr - HMC5883L | 20 |
| 2.20 | Připojení magnetometru | 21 |
| 2.21 | Moduly xBee s použitými shiely | 21 |
| 2.22 | Logický úroňový převodník | 22 |
| 2.23 | Připojení logického převodníku s čidly, které mají odlišné napájecí napětí | 23 |
| 2.24 | Umístění ultrazvukového čidla | 24 |
| 2.25 | Uchycení optického enkodéru | 24 |
| 2.26 | Inerciální měřicí jednotka, magnetometr a úroňový převodník | 25 |
| 2.27 | Kompletní schéma zapojení senzorů | 26 |
| 3.1 | Ilustrační obrázek čtyřkolového diferenciálně řízeného robotu | 28 |
| 3.2 | Blokové schéma pro sledování trajektorie robotu | 32 |
| 3.3 | Řízení na kružnici | 37 |

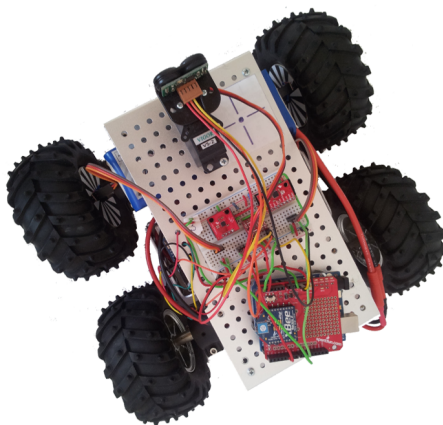
| | | |
|-----|---|----|
| 3.4 | Odchylka při řízení na kružnici | 37 |
| 3.5 | Řízení na elipse | 38 |
| 3.6 | Odchylka při řízení na elipse | 39 |
| 3.7 | Řízení na položené osmičce | 40 |
| 3.8 | Odchylka při řízení na položené osmičce | 40 |

1 Úvod

Cílem práce je vytvoření diferenciálně řízeného čtyřkolového robotu, který bude pomocí informací ze vhodně vybraných čidel schopen projet libovolně zadanou trajektorii.

První část práce je zaměřena na seznámení se s roboty řízenými diferenciálním pohonem a s konkrétním typem vybraného robotu. Samotný model je dodáván bez potřebné sensoriky a tak bylo nutné také navrhnout čidla, která bude nutné použít pro sledování robotu v prostoru. Dále bylo nutné zvolit také správný vývojový software s volně dostupnými knihovnami pro programování mikročipu a připojených čidel. Zvoleným programem byl software Arduino a programovací jazyk Wiring, který vychází z jazyka C++. Nakonec bylo nutné celý hardware zkompletovat a individuálně zakoupená čidla vhodně upevnit na konstrukci robotu.

Ve druhé části je detailně odvozen matematický model robotu, díky kterému jsme schopni pomocí vhodného softwaru odsimulovat jeho chování na počítači a dále je navrženo jednoduché řízení pomocí stavového regulátoru. Jak je uvedeno v dalších kapitolách, koordinace robotu je umožněna díky odlišným úhlovým rychlostem kol na každé straně nápravy. Z tohoto tvrzení se vycházelo i při návrhu řízení, které má za úkol sledovat zadanou trajektorii.



2 Návrh hardwaru mobilního robotu

Jak bylo uvedeno výše, práce je zaměřena na kompletní sestavení diferenciálně řízeného robotu od návrhu až po jeho programování. Prvním krokem ke splnění této práce bylo navrhnout, jaký typ konstrukce robotu zvolit, jakým způsobem bude řízen a jakými senzory bude osazen. Požadavky na návrh robotu byly následující:

- Výběr vhodného podvozku pro snadnou manipulaci v prostoru
- Detekce překážek, které se nachází v trase robotu
- Zjištění aktuální ujeté vzdálenosti
- Orientace robotu v prostoru
- Bezdrátová komunikace s vizualizačním prostředím

V této kapitole je popsán celkový hardware robotu od typu podvozku až po jednotlivá čidla, která jsou v práci použita.

2.1 Platforma Arduino

Srdcem navrhovaného robotu je platforma Arduino stejnojmenného italského výrobce. Je to otevřená elektronická platforma, založená na uživatelsky jednoduchém hardware a software. Desky této platformy jsou osazeny různými mikrokontroléry od firmy ATMEL a programují se pomocí speciálního programovacího jazyku, který je založený na jazyku Wiring¹ ve vlastním vývojovém prostředí. Projekty založené na Arduinu mohou jednoduše komunikovat se softwarem na stolním počítači nebo notebooku a zobrazovat naměřená data v prostředí Processing, které je použito v této práci. Hlavní výhody použité platformy jsou:

- jednoduché programování
- jednoduché připojení senzoriky
- nízká cena oproti jiným vývojovým kitům
- mnoho knihoven určených pro připojená čidla

¹Speciální programovací jazyk používaný pro programování mikrokontrolérů

- rozsáhlá uživatelská komunita

V následujících dvou kapitolkách je popsán použitý vývojový kit a stručný úvod do programovacího prostředí se základním kódem pro rozblíkní LED diody.

2.1.1 Vývojový kit Arduino UNO

Tento vývojový kit, obrázek (2.1), navržený italskými vývojáři, obsahuje jednočipový mikrokontrolér ATmega328. Uvedený čip vyrábí společnost Atmel z Norska a jedná se o procesor s architekturou AVR (osmibitový RISC, harvardská architektura). ATmega328 obsahuje 32kB FLASH paměti pro uložení programu, 2kB statické RAM pro data, 1kB EEPROM pro ukládání non-volatilních² dat a osmikanálový desetibitový A/D převodník. Dále obsahuje některé systémové periferie, jako jsou časovače či přerušovací subsystém. Tento mikrokontrolér dokáže při taktu 20 MHz provádět až 20MIPS³ – většina instrukcí totiž trvá právě jeden takt hodin.

Na samotné desce je vyvedeno 13 digitálních vstupně výstupních pinů, 6 analogových vstupů a 5V a 3,3V úrovně napětí. Komunikace s PC probíhá přes kabel USB typu A-B a pomocí toho kabelu je řešeno i napájení celé platformy. Další možností je napájení kitu pomocí jiného napět'ového zdroje, jako je například 9V baterie. To je vhodné zejména v případech, kdy není využívána komunikace s PC nebo je komunikace zprostředkována bezdrátově (například v práci použité moduly xBee).

2.1.2 Vývojové prostředí

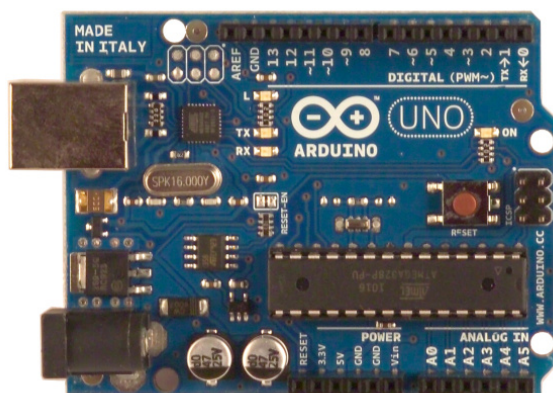
Integrované vývojové prostředí Arduino je tzv. open-source, takže je volně ke stažení na internetu. Neobsahuje žádné ladící prvky a jeho hlavní modul je napsán v jazyce Java. K programování v tomto editoru se používá již zmíněný jazyk Wiring.

Popis prostředí

Vývojové prostředí se skládá ze tří částí. První částí (obr 2.2) je nabídka menu a pod ní základní příkazy používané při práci v programu. Pro současnou verzi prostředí je zde 5 příkazů - kompilace programu, nahrání programu do čipu, načtení dříve vytvořených projektů, uložení aktuálního

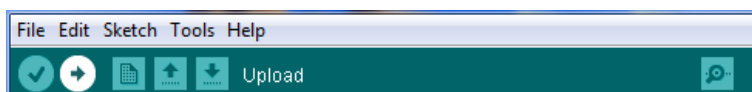
²Typ paměti u které obsah paměti není při odpojení napájení smazán

³Milion instrukcí za vteřinu - jednotka udávající výkonnost procesoru

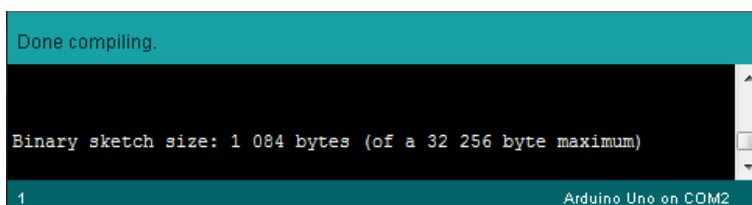


Obrázek 2.1: Arduino UNO

projektu a otevření nového projektu. Na levé straně je pak tlačítko pro otevření okna Serial Monitor, ve kterém se zobrazují naměřená data poskytnutá po sériové komunikaci. Uprostřed editoru je samotný textový blok, do kterého se píše kód a ve spodní je informační řádka, která informuje uživatele o případných chybách v napsaném programu nebo o úspěšném přeložení či nahrání programu. Na obrázku (2.3), kde je vyobrazen informační řádek, je vidět, že napsaný program byl úspěšně přeložen. V pravém dolním rohu je možno zjistit, jaký typ Arduina je připojen k PC a jaké číslo komunikačního portu využívá. V tomto případě se jedná o port COM2⁴.



Obrázek 2.2: Vývojové prostředí Arduino



Obrázek 2.3: Vývojové prostředí Arduino

⁴Sériový port nebo také rozhraní RS232 sloužící ke komunikaci s externím zařízením

Psaní programu

Psaní programu v tomto editoru je velice snadné. Důležité je vědět, že program musí obsahovat dvě základní metody `setup()` a `loop()`. Metoda `setup()` je, jak název vypovídá, nastavovací. Zde se nastaví například rychlost sériové komunikace, nadefinují se jednotlivé digitální piny jako vstupy/výstupy a nebo se nastaví a kalibrují čidla používaná v projektu. Ve smyčce `loop()` se pak vyskytuje hlavní část programu, která se opakovaně vykonává.

Zde je uveden jednoduchý kód, pomocí kterého lze rozblikat LED diodu:

```
#define pinLED 13 ;           // nadefinovani pinu c.13 jako promenne

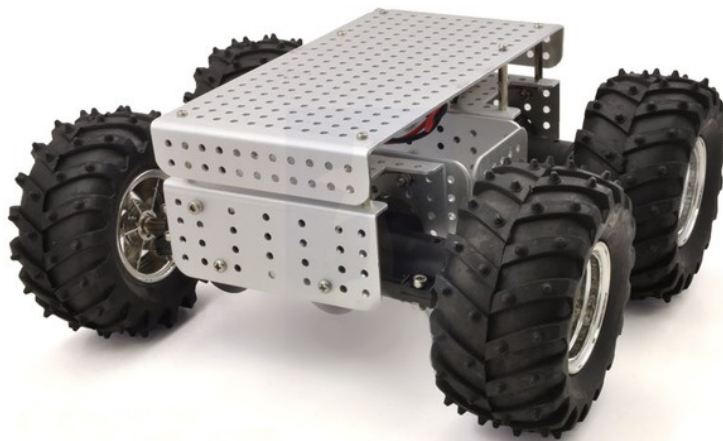
void setup() {
  pinMode(pinLED, OUTPUT);   // nastavi digitalni pin
                             // cislo 13 jako vysptuni
}

void loop() {
  digitalWrite(pinLED, HIGH); // rozsviti LED na pinu 13
  delay(1000);                // vycka 1 vterinu
  digitalWrite(pinLED, LOW);  // zhasne LED na pinu 13
  delay(1000);                // vycka 1 vterinu
}                             // cela smycka se opakuje
```

2.2 Vozítko Dagu Wild Thumper 4WD

Dalším bodem návrhu hardwaru bylo vybrání vhodného diferenciálně řízeného podvozku. V práci je zvolen robustní 4-kolový podvozek vyvinutý společností Dagu Electronics. Díky své robustnosti a použitým kolům je toto vozítko určeno do složitého terénu s výraznými nerovnostmi. To je jednoznačnou výhodou, protože v dnešní době jsou velmi náročné požadavky na pohyb robotu v obtížném terénu. Kola mají průměr 12,5 cm a mají šípovitý dezén s vystouplými hroty pro snadnější pohyb například v sypkém terénu. Každé kolo má nezávislé zavěšení a odpružení kola je řešeno pomocí pružiny, která by podle mého názoru mohla být měkčí, aby nebyl podvozek tak tvrdý. Rám robotu je vyroben z hliníkového materiálu s dírami o průměru 4 mm a vzdálenosti 1 cm od sebe. Díky tomu lze na rám bez problémů uchytit další potřebnou elektroniku. O pohon každého kola se stará stejnosměrný motor s převodem 75:1, který dokáže vyvinout rychlost

až 160 otáček za minutu. Typ podvozku je takzvaně diferenciální, což znamená, že otáčení robotu se uskuteční, jestliže otáčky motorů na každé straně jsou rozdílné. Na každé straně jsou dva motory, které jsou spojeny paralelně k sobě, aby otáčky kol byly stejné. Jmenovité provozní napětí je 7,2 V a je poskytováno baterií od firmy Turnigy s kapacitou 5000 mAh.



Obrázek 2.4: Konstrukce Wild Thumper 4WD

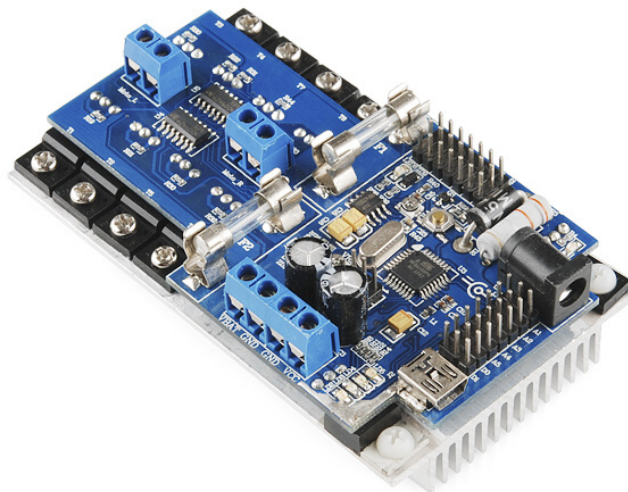
2.3 Arduino pro řízení motorů

Spolu s robotem je dodáván řídicí modul založený na platformě Arduino, který je již vybavený dvojitým H-můstkem⁵ pro řízení motorů robotů. Vzorový kód pro řízení motorů lze volně stáhnout na stránkách prodejce⁶. Tento kód je ovšem určen pro starší verze vývojového prostředí a tak bylo nutné sketch upravit pro současnou verzi Arduina. Tato platforma Arduina se stará o řízení motorů pomocí H-můstků a informaci o řízení může přijímat různými způsoby, například pomocí RC vysílače, sériové komunikace a nebo I2C sběrnice. V této práci jsou motory ovládány pomocí posledně zmíněné I2C komunikace, protože řídicí Arduino UNO komunikuje s tímto Arduinem pro řízení motorů právě pomocí této sběrnice. Co znamená tato sběrnice a jak funguje je popsáno v následující podkapitole. Samotný kontrolér je vyobrazen na obrázku (2.5) a obsahuje následující prvky:

⁵Zapojení umožňující přepínat polaritu napájecího napětí na svorkách motoru a tím měnit smysl otáčení

⁶Adresa výrobce - <http://robotsavy.com>

- Procesor ATmega168 s 16kB paměťí flash.
- Duální 15A H-můstky pro řízení motorů
- Detekci spálené pojistky H-můstku pomocí LED diody
- 7 vstupně/výstupních digitálních pinů
- 5 analogových vstupních pinů
- Komunikaci přes USB, TTL, I2C nebo RC přijímač
- Konektor pro nabíjení použitého napájecího akumulátoru
- Sledování kapacity baterie pomocí procesoru



Obrázek 2.5: Kontrolér pro řízení robotu

2.4 Sběrnice I2C

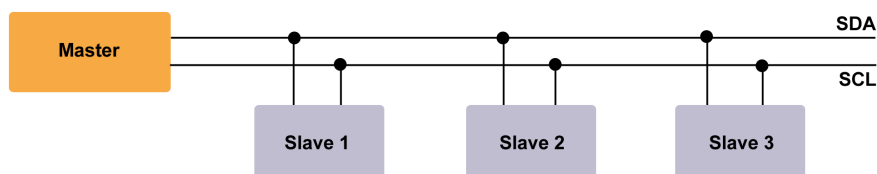
Další nedílnou součástí práce je využití komunikace použité senzorky s Arduinem přes komunikační sběrnici I2C. Vzhledem k tomu, že použité Arduino umožňuje komunikaci po této sběrnici, byla porížena čidla, která dokáží komunikovat také po této sběrnici.

2.4.1 Co je I2C sběrnice

I2C bus je zkratka která vznikla z IIC ⁷ bus. Jak již název napovídá, jedná se o interní datovou sběrnici sloužící pro komunikaci a přenos dat mezi jednotlivými integrovanými obvody většinou v rámci jednoho zařízení. Vyvinula ji firma Philips přibližně před 20 lety. V dnešní době tento způsob komunikace podporuje řada integrovaných obvodů. Hlavní výhodou této sběrnice je, že obousměrný přenos probíhá pouze po dvou vodičích. Prvním je datový vodič nazývaný SDA (serial data) a druhý vodič přenáší hodinový signál a nazývá se SCL (serial clock).

Na jednu sběrnici může být připojeno více integrovaných obvodů. V dnešní době jsou prvky sběrnice adresovány 7 nebo 10 bitově, což ve druhém případě umožňuje na jednu sběrnici připojit až 2^{10} modulů. V reálných případech je toto číslo podstatně menší, protože adresa modulu nelze určit všemi bity. V některých případech mívají moduly dokonce své pevné adresy a je nereálné připojení více prvků se stejnou adresou na jednu sběrnici.

Rychlost sběrnice může být v nejlepších případech až 1MHz, ale ne všechny integrované obvody tuto rychlost podporují a proto bývá nejčastěji 100kHz. Rychlost sběrnice se musí přizpůsobit nejpomalejšímu modulu. Pokud tedy všechny obvody budou moci komunikovat rychlostí 1MHz, ale jeden obvod bude schopen komunikovat pouze rychlostí 100kHz, tak rychlost komunikace celé sběrnice bude právě 100kHz.



Obrázek 2.6: Schéma připojených zařízení přes I2C

2.4.2 Princip komunikace

Pohledem na obrázek (2.6) je vidět, že jedno připojené zařízení komunikuje jako Master a ostatní komunikují v režimu Slave. Režim Master je nejčastěji zaváděn u řídicích kontrolérů, které pak sbírají data od ostatních čidel.

⁷Internal-Integrated-Circuit Bus - Vnitřní integrovaná sběrnice

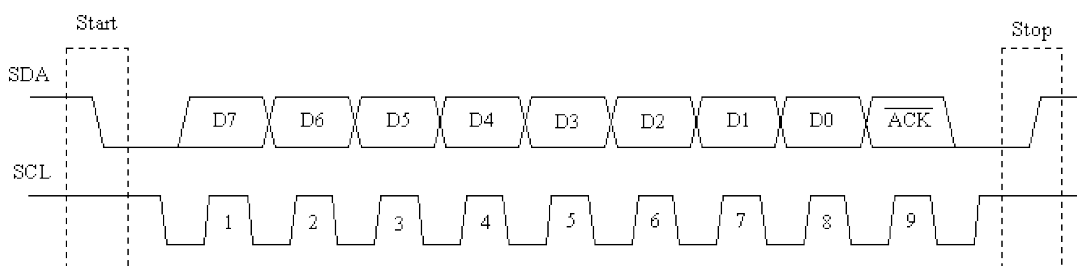
Zařízení v režimu Master může být více (režim multi-master), ale v této práci bude pouze jedno, Arduino UNO, které bude sbírat data vyslaná čidly.

Nyní bude popsáno, jakým způsobem I2C sběrnice funguje. Nejprve je nutné, aby oba vodiče před zahájením komunikace byly v logické jedničce, což je zajištěno pull-up rezistory připojenými mezi napájecím napětím a vodičem SDA případně SCL. Poté je vyslán start bit a Master začne generovat hodinový signál na vodiči SCL. Když jeden čip vysílá, všechny ostatní přijímají a pouze podle adresy určují, zda jsou vysílaná data určena jim. Čip, který chce vyslat/přijmout data musí nejprve definovat adresu čipu, s kterým chce komunikovat a zda půjde o příjem nebo vysílání - tedy o čtení nebo zápis. To určuje R/W (read/write) bit, který je součástí adresy. Zařízení, které bylo požádáno o vysílání/přijímání dat se ozve cílovému zařízení potvrzujícím ACK bitem. Dále je zahájen přenos mezi dvěma zařízeními. Přenos je realizován vždy po 8 bitech a posledním potvrzovacím ACK bitem. Po skončení přenosu se generuje stop bit, který ukončí komunikaci. Vše je podrobněji vysvětleno v následujících bodech v pořadí, jak probíhá přenos.

- **Stav klidu** - Oba vodiče jsou v logické jedničce, Master tedy neregeneruje hodinový signál a neprobíhá žádný přenos.
- **Start bit** - Zahajuje přenos nebo jeho další část. Je vygenerován tak, že se změní úroveň SDA z logické 1 na logickou 0. Vodič SCL je stále ve stejné logické úrovni jako v klidovém stavu.
- **Přenos dat** - Data jsou přenášena po 1 bytu tedy 8 po sobě jdoucích bitů. Přenáší se od nejvyššího po nejnižší (7 bitů) a poslední bit určuje, zdali se jedná o čtení nebo zápis. Při přenosu dat se může logická úroveň na SDA měnit pouze pokud je SCL v logické 0. Při každém pulzu na SCL je přenesen jeden bit, který je vodiči SDA.
- **Potvrzující bit Ack** - Význam zkratky tohoto bitu je acknowledge, což v překladu znamená potvrzení, a slouží k potvrzení správného přijetí dat. Ack bit se odesílá stejným způsobem jako by se odesílal devátý bit dat, ale opačným směrem, aby zařízení, které data odesílalo, bylo informováno o přijetí dat. Pokud přenos proběhl v pořádku, tak odešle logickou 0. Logická 0 potvrzujícího bitu znamená rovněž to, že je přijímač připraven na příjem dalšího bytu, který následuje okamžitě při dalším pulzu na SCL. Pokud přenos selhal odešle logickou 1.

- **Condition stop** - Bit, který ukončuje přenos. Je generován stejně jako start bit, ale logická úroveň SDA se změní z 0 na 1 zatímco SCL zůstává v logické 1.

Celý proces komunikace je znázorněn na obrázku (2.7). Komunikace je zahájena Start bitem, následuje 7 bitů (D7 - D1) s přenášenou informací, 8. bit (D0) určuje směr komunikace a zakončujícím bitem Ack. Celý proces komunikace pak ukončí Stop bit.



Obrázek 2.7: Časový průběh přenosu dat na I2C

2.5 Čidla

V této sekci jsou uvedeny základní informace o použitých čidlech a u každého čidla je uvedeno schéma připojení k Arduino. Tato schémata byla vytvořena pomocí programu Fritzing⁸. Jelikož samotný robot neobsahuje žádné senzory, které by získávaly informace o jeho pohybu v prostoru, bylo nutné robot vybavit patřičnými čidly, aby byl dostatek informací k monitorování jeho pohybu.

2.5.1 Ultrazvukové čidlo

Jedním z požadavků na návrh bylo, aby robot dokázal detekovat překážku a mohl určit její vzdálenost. Pro splnění tohoto požadavku bylo použito ultrazvukové čidlo, které je popsáno na následujících řádkách.

Princip ultrazvukové detekce

Ultrazvukové čidlo vysílá cyklicky vysokofrekvenční impuls, který se šíří prostorem rychlostí zvuku. Pokud narazí na nějaký předmět, odrazí se od

⁸zdarma ke stažení na <http://fritzing.org>

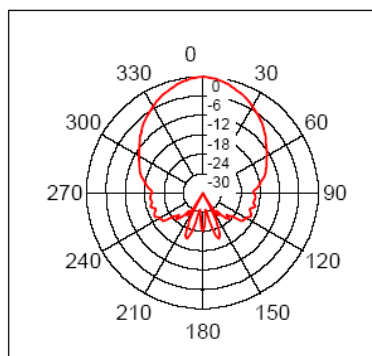
něj a vrací se zpět k čidlu jako ozvěna. Z časového intervalu mezi vysláním impulsu a návratem ozvěny odvodí čidlo vzdálenost k předmětu. Vzhledem k tomu, že ultrazvuková čidla měří vzdálenost na principu časové odezvy na vyslaný zvukový impuls při konstantní rychlosti šíření zvuku a ne pomocí měření intenzity, mají vynikající schopnost potlačení pozadí. Téměř všechny materiály, které odrážejí zvuk, lze detekovat bez ohledu na jejich barvu. Lze je použít bez problému na průhledné materiály a tenké fólie. Tato čidla umožňují měřit vzdálenosti od 30 mm do 10 m s přesností 1 mm. Některá čidla mají rozlišení až 0,18 mm.



Obrázek 2.8: Použité ultrazvukové čidlo - Devantech SRF10

Parametry použitého čidla Devantech SRF10

Použité ultrazvukové čidlo (obr 2.8) má rozsah měření od 6 cm do 6 m. Vyzařovací úhel, kterým lze detekovat překážku, je 72° . Vyzařovací úhel ultrazvukového pulzu je vidět na obrázku (2.9).

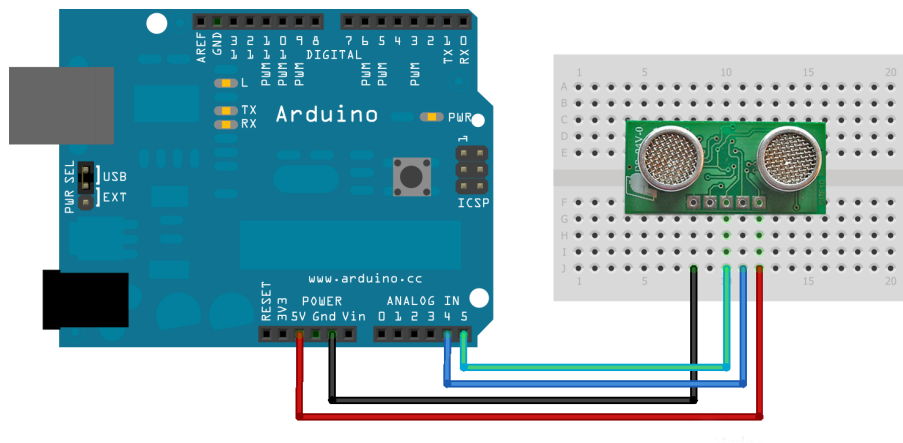


Obrázek 2.9: Vyzařovací úhel pulzu

Využití čidla

V této práci je čidlo využito pro detekci překážek v trajektorii pohybu robotu. Je umístěno na servo motoru a překážky tak lze detekovat ve více

směrech v závislosti na natočení serva. Čidlo je napájeno napětím 5V a komunikace s Arduinem probíhá pomocí již probrané I2C sběrnice a v samotném programu se zvolí možnost, v jakých jednotkách je vzdálenost měřena (zvoleny cm). Připojení je zobrazeno na obrázku (2.10)



Obrázek 2.10: Připojení ultrazvukového čidla

2.5.2 Servomotor

Vyzařovací úhel ultrazvukového čidla je poměrně velký, tak je čidlo umístěno na servomotoru, pro případ, že by bylo nutné měřit vzdálenost překážky po boku robotu, kde by v případě pevné montáže čidla nebylo možné vzdálenost měřit.

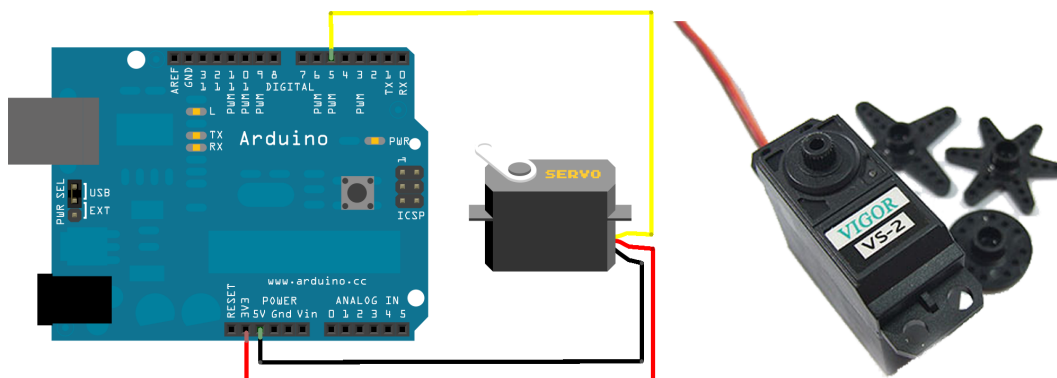
Servomotor se používá pro takové projekty, kde je třeba využít různého natočení, například čidla, v jedné ose. Pomocí serv lze ovládat například uzavírání ventilů v regulovaných soustavách. Časté využití serv je také v modelářské oblasti. Servomotory jsou z důvodu tepelných ztrát řízeny prakticky výhradně tranzistorovými měniči s pulzně šířkovou modulací.

Polohování servomotorů:

- plynulé, jako zarážky slouží koncové polohy
- ve dvou či více diskretních polohách

Polohu servomotorů lze určit podle počtu vyslaných impulzů a nebo podle času, po který byl pohon zapnutý. Určení polohy servomotoru lze vylepšit

použitím snímačů pro odečet polohy (například využití optických enkodérů). Signál těchto čidel lze využít k dalšímu řízení pohonu, například vypnout motor po dojezdu do krajní polohy. Zavedením lineární záporné zpětné vazby lze servomotorem řídit polohu zařízení v celém rozsahu jeho pracovní dráhy.



Obrázek 2.11: Připojení servomotoru a použité servo

2.5.3 Optický enkodér

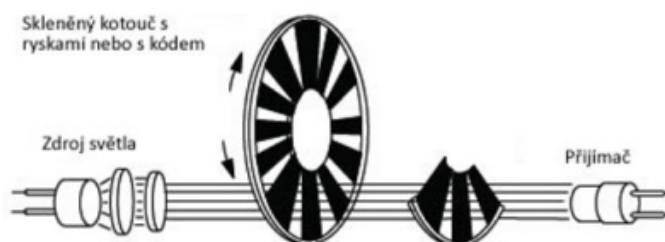
Optické enkodéry jsou dnes velmi rozšířené, hlavně proto, že jsou digitální a digitální systémy postupně nahrazují systémy analogové. Optický enkodér tvoří zdroj impulzů, které vytváří zdroj světla, jenž prochází přes kotouč s kódem, a snímače, které snímají přijaté impulzy. Enkodéry se dělí na inkrementální a absolutní nebo rotační a lineární.



Obrázek 2.12: Optický enkodér - QTR-1RC

Princip optických enkodérů

Optický kotouč je spojen s otočnou spojovací hřídelí, která spojuje motor a kolo. Kotouč se tedy otáčí současně s kolem. Světlo vysílané LED diodou, která je umístěna v dostatečné vzdálenosti před kotoučem, se odráží/prochází pouze od bílých částí kotouče, zbývající černé úseky disku světlo pohlcují. Světelné impulsy odražené od optického kotouče aktivují fototranzistor. Elektrické impulsy fototranzistoru poté generují obdélníkový signál. Aby bylo u snímače zajištěno generování obdélníkových impulsů bez rušení, musí být elektrický signál zesílen a elektronicky zpracován.



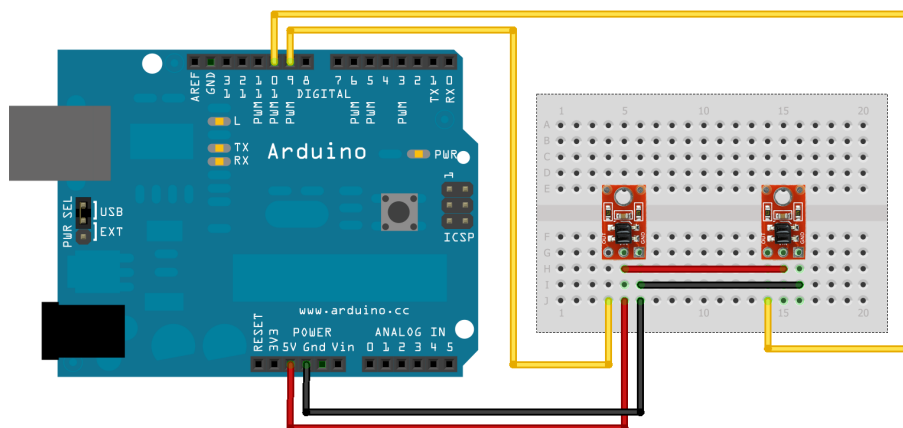
Obrázek 2.13: Princip inkrementálního enkodéru

Použitý enkodér - QTR-1RC

Tento malý modul zobrazený na obrázku (2.13) obsahuje na jedné desce zároveň vysílač a přijímač signálu. Jako vysílač je použita LED dioda vyzařující IR světlo. Jelikož je zdroj světla i jeho přijímač na jedné desce, světlo optickým kotoučem neprochází, ale odráží se od něj zpět. Odražené světlo od disku detekuje fototranzistor, který slouží jako přijímač. Světelné signály dokáže fototranzistor detekovat až do vzdálenosti 9,5mm. Enkodér je napájen napětím 5V. Vodič, kterým probíhá komunikace mezi čidlem a mikrokontrolérem je připojen na digitální výstup Arduina. Na kolech vozítka jsou použity rotační optické enkodéry s 24 přechody. Vzhledem k tomu, že na každém kole je použit pouze jeden tento enkodér, není možné měřit směr otáčení kol. Pokud bychom požadovali měřit i směr otáčení, bylo by nutné použít kvadrurní enkodér.

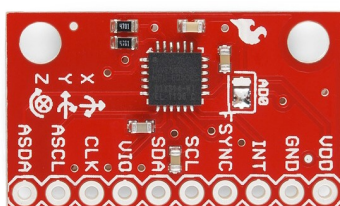
2.5.4 Inerciální měřicí jednotka

Inerciální měřicí jednotka, neboli IMU, vychází z anglického spojení Inertial measurement unit a slouží k měření zrychlení, rychlosti a polohy v prostoru. Má velmi široké uplatnění, například u automobilového průmyslu se měřicí



Obrázek 2.14: Připojení enkodéru

jednotka používá zejména pro vyhodnocování jízdních manévřů vozidla, zpravidla při testování jízdních vlastností, nebo k záznamu trasy v kombinaci s GPS.



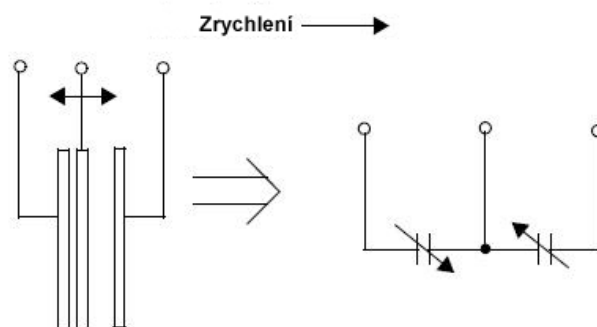
Obrázek 2.15: Inerciální měřící jednotka - MPU6050

IMU měří data prostřednictvím dvou senzorů uspořádaných do tří na sebe kolmých os. Prvním z nich jsou gyroskopy měřící úhlovou rychlost, které dodávají informace o orientaci (natočení) objektu. Druhý typ použitého senzoru jsou akcelerometry informující o zrychlení objektu. Dále použitá jednotka obsahuje také potřebné AD převodníky, digitální zpracování signálu pomocí filtrů a komunikační sběrnici I2C, po které jednotka komunikuje s mikrokontrolérem. Princip měření jednotky je založen na technologii MEMS⁹, která spočívá v proměnlivosti kapacity tříelektrodového vzduchového kondenzátoru. Prostřední elektroda se pohybuje v závislosti na velikosti zrychlení a mění tak kapacitu mezi oběma krajními elektrodami.

⁹Micro-Electro-Mechanical Systems - technologie používaná pro pohybové senzory

Akcelerometr

Jak již bylo výše zmíněno, akcelerometry mohou měřit zrychlení nebo vibrace. Změřenou hodnotu poté převádějí na elektrický signál, který je dále zpracováván a vyhodnocován například mikrokontrolérem. Tyto senzory měří dva druhy zrychlení. Jedním je dynamické zrychlení, které se projevuje při změně rychlosti pohybujícího se objektu, na kterém je čidlo umístěno, a druhým je statické zrychlení, které je dáno zemskou gravitací. Princip měření akcelerometru je znázorněn na obrázku (2.16). Je zde vidět, že pokud působí zrychlení doprava, elektroda kondenzátoru se přesune vlivem síly doleva a porovnáním kapacit mezi oběma elektrodami získáme příslušnou hodnotu zrychlení v dané ose.



Obrázek 2.16: Princip měření akcelerometru

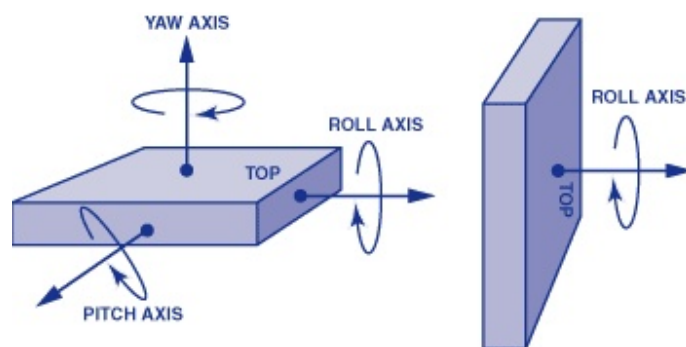
Použití akcelerometrů v praxi:

- Automobilový průmysl - Airbag, ESP¹⁰, zabezpečení vozidel
- Navigační systémy
- Detekce rychlosti a pohybu
- Měření vibrací
- Měření natočení a detekce otáčení
- Měření akcelerace

¹⁰Elektronický stabilizační program - omezení výkonu vozidla, zamezení prokluzu hnacích kol

Gyroskop

Gyroskopy jsou obecně určeny pro měření úhlové rychlosti, tzn. údaj o tom, jak se měřený objekt rychle otáčí, v jednotkách stupňů za sekundu ($^{\circ}/s$). Rotaci je možné typicky měřit vzhledem k jedné ze tří os x, y, z, někdy označované jako podélná osa (roll axis), příčná osa (pitch axis) a svislá (kolmá) osa (yaw axis) viz obrázek (2.17). Rozdíl mezi akcelerometrem a gyroskopem je pouze jeden, ovšem velice důležitý. Akcelerometr měří hodnotu vychýlení vnitřní struktury jako důsledek působícího zrychlení zatímco gyroskop zjišťuje rotační pohyb, přesněji řečeno rychlost ustáleného rotačního pohybu. U akcelerometru se struktura pohybuje jen při konání pohybu. U gyroskopu je však strukturou pohybováno vlivem mechanické rezonance úmyslně, aby se projevil efekt Coriolisovy síly. Coriolisova síla je tzv. virtuální síla, která působí na libovolný hmotný předmět či objekt, který se pohybuje rychlostí v v soustavě rotující kolem osy rotace úhlovou rychlostí ω . Vzhledem k tomu, že gyroskopy se používají společně s akcelerometry, tak jejich použití je také stejné.



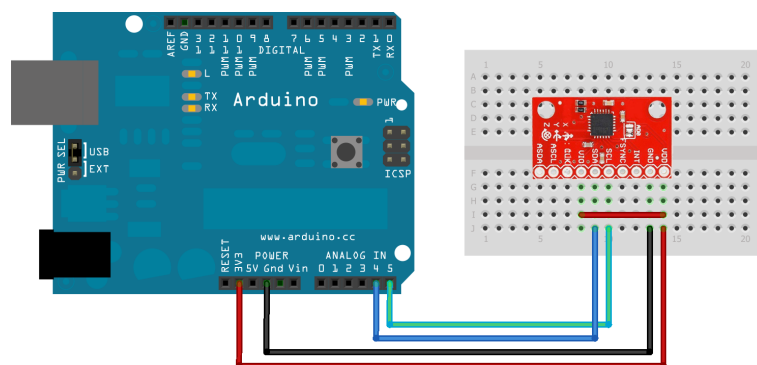
Obrázek 2.17: Princip měření gyroskopem

Připojení IMU k Arduinu

Měřící jednotka komunikuje s mikroprocesorem také pomocí I2C sběrnice. Napájení je 3,3V a je nutné připojit pin VIO na inerciální měřící jednotce přímo k napájení 3,3V. Tato verze IMU obsahuje i piny pro připojení externího zařízení (ASDA a ASCL).

2.5.5 Magnetometr

Magnetometr slouží k měření magnetického pole Země a nebo ostatních magnetických polí, jejichž velikost je v rozsahu možností použité součástky.



Obrázek 2.18: Připojení IMU

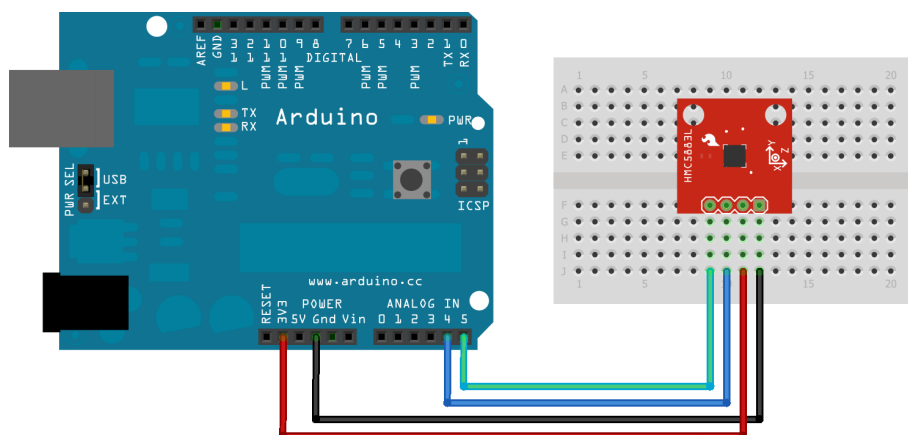
Ve většině případů dokáží pole měřit ve třech osách - tzv. 3-osé magnetometry. Své uplatnění magnetometry dnes naleznou například v mobilních zařízeních, v leteckém provozu, v astronomii a nebo v GPS navigacích. Většinou se používají společně s akcelerometry a gyroskopy a společně tak lze zjistit úplné informace o pohybu a orientaci daného předmětu, kde jsou tyto tři senzory použity.



Obrázek 2.19: Magnetometr - HMC5883L

Princip měření magnetometru

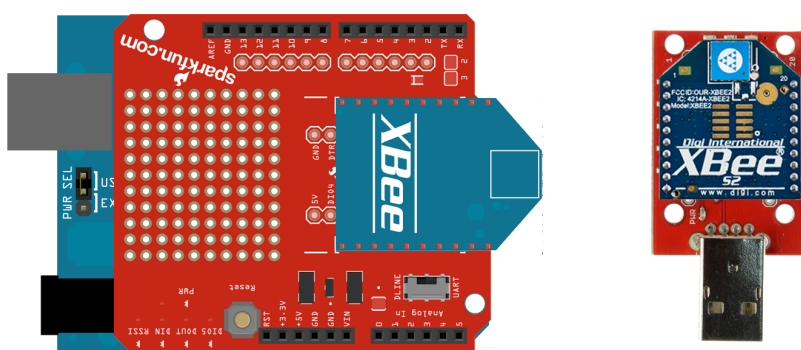
Princip měření magnetického pole je založen na magnetorezistivním jevu. Tento jev je založen na změně odporu magnetického materiálu při změně působícího magnetického pole. Struktura 3-osého magnetometru obsahuje tři snímače magnetického pole ve formě Wheatstonových můstků složených z rezistorů z magnetorezistivního materiálu. Každý můstek je orientován ve směru jedné měřicí osy. Po připojení napájení je mezi výstupy můstku napětí úměrné směru a velikosti magnetického pole.



Obrázek 2.20: Připojení magnetometru

2.5.6 Bezdrátový komunikační modul xBee

Tento bezdrátový modul využívá jednoduchý komunikační standard nazývaný ZigBee, který umožňuje komunikaci s různými zařízeními ve vzdálenosti desítek až stovek metrů a tím lze ušetřit náklady, které by obnášelo použití komunikace pomocí vodičů. Tyto moduly mají velmi nízkou spotřebu a díky tomu je lze uplatnit v bateriemi napájených aplikacích. Obecné využití modulu bývá v monitorování a řízení systémů. Další nespornou výhodou je adresace několika tisíců dalších komunikačních modulů do jedné komunikační sítě, spolehlivost a v neposlední řadě nízká pořizovací cena.



Obrázek 2.21: Moduly xBee s použitými shiedly

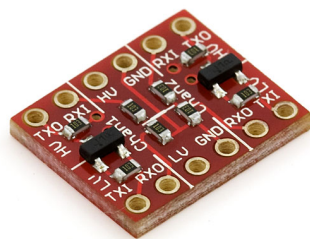
2.5.7 Logický úroňový převodník

Poslední v této práci použitá součástka není senzor, ale slouží k oddělenému zpracování signálů senzorů, které mají různé napájecí napětí a jejichž komunikace s procesorem probíhá pomocí I2C, SPI a jiných digitálních sběrnic. V tomto případě bude tedy oddělovat signály ze sběrnice I2C. Pokud by byla čidla připojena přímo bez využití převodníku, mohla by být poškozena. Dalším důvodem použití této součástky byla nutnost galvanického oddělení logických úrovní, protože jejich hodnoty se pro různá napájecí napětí také liší. Tento použitý převodník neumí pracovat s analogovými signály.

Popis připojení

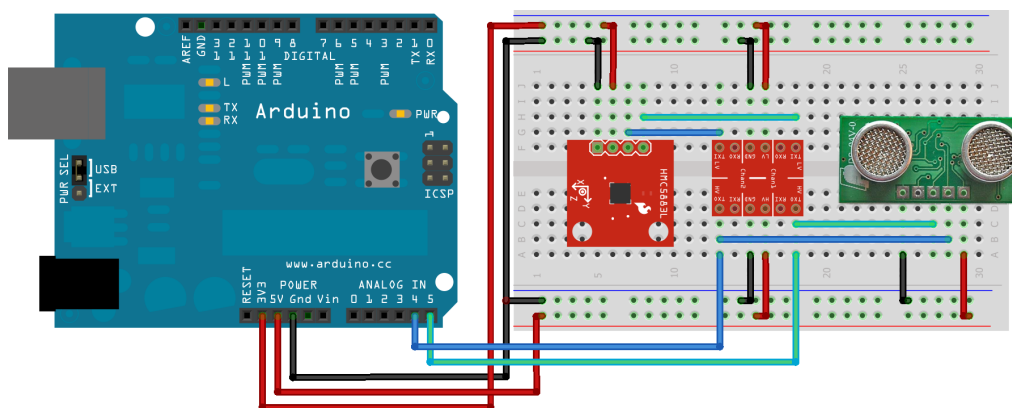
Připojení převodníku k mikropočítači vypadá následovně.

- HV - připojení vyšší napět'ové úrovně (5V)
- LV - připojení nižší napět'ové úrovně (3,3V)
- GND - připojení uzemnění
- TXO - připojení signálových vodičů z arduina
- TXI - připojení signálových vodičů ze senzorů



Obrázek 2.22: Logický úroňový převodník

Jelikož je připojení úroňového převodníku poněkud komplikovanější, tak je na obrázku (2.23) znázorněno zapojení s magnetometrem a ultrazvukovým čidlem. Každé čidlo je napájeno jinou úrovní napětí a je zde tak lépe vidět oddělení komunikace čidel napájených jiným napětím. Prvky napájené vyšším napětím komunikují s čipem přímo a u senzorů, které jsou napájeny nižší hodnotou napětí, prochází signály převodníkem.



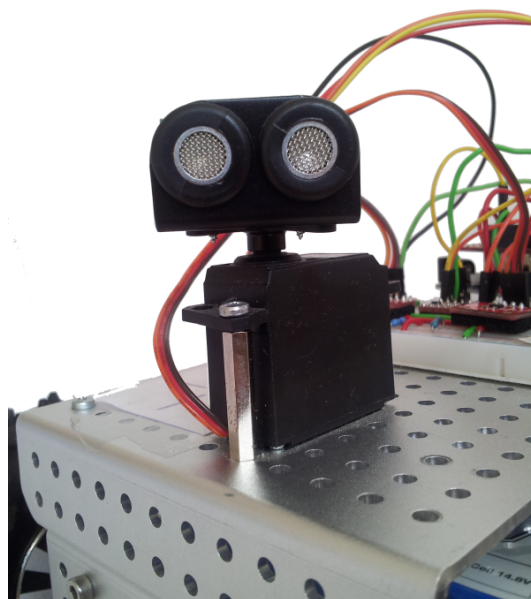
Obrázek 2.23: Připojení logického převodníku s čidly, které mají odlišné napájecí napětí

2.6 Kompletace hardwaru

Po výběru příslušných čidel byla na řadě realizace kompletního robotu s upevněním veškeré dostupné sensoriky. Montáž čidel bude popsána v pořadí, v jakém byly dané senzory uvedeny výše.

Ultrazvukové čidlo bylo připevněno na otočnou hřídel servomotoru a celý tento komplet byl pomocí distančních sloupků připevněn ke konstrukci podvozku. Důležité zde byl výběr vhodného místa připevnění, aby nebyly ultrazvukové paprsky odraženy od samotné konstrukce vozítka. Vzhledem k tomu, že robot detekuje překážky v cestě, je čidlo umístěno na přídi robotu, viz obrázek (2.24).

Nejvíce komplikací nastalo při montáži optických enkodérů. Musely být připevněny na motory a v maximální vzdálenosti 9,5mm od optického kotouče. Ovšem pokud byly senzory připevněny přímo k motorům, tak vzdálenost k optickému kotouči byla přes 2cm, což bylo nevyhovující. Problém byl vyřešen distančními sloupky, které zmenšily vzdálenost na 5mm. Použití distančních sloupků ale znemožnilo otáčení daného kola, protože šroub, který upevňuje kolo ke hřídeli motoru byl příliš dlouhý - viz obrázek (2.25) vpravo dole. Bylo nutné tedy tyto šrouby nahradit kratšími, které nevyčnívaly ze závitů - viz obrázek (2.25) vpravo nahoře. Poslední nedostatek nastal při testování čidel. Rychlost otáčejících se kol byla příliš vysoká a enkodéry nestíhaly zaznamenávat všechny změny v odrazech signálu. Vzdálenost čidla od osy otáčení hřídele byla proto pomocí plastového proužku nastavena, aby čidlo měřilo v části, kde je úhlová



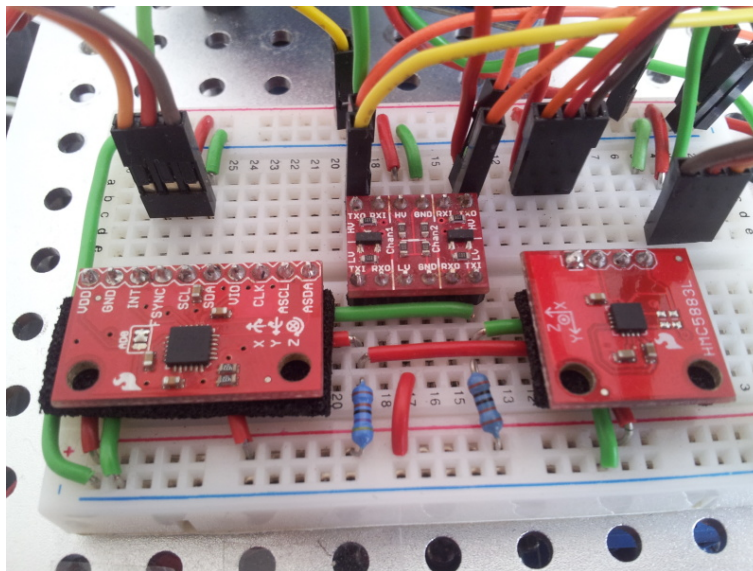
Obrázek 2.24: Umístění ultrazvukového čidla

rychlost kola nejmenší, tedy proti vnějšímu okraji optického kotouče. Připevnění jednoho z enkodérů je vyfoceno na obrázku (2.25) nalevo.



Obrázek 2.25: Uchycení optického enkodéru

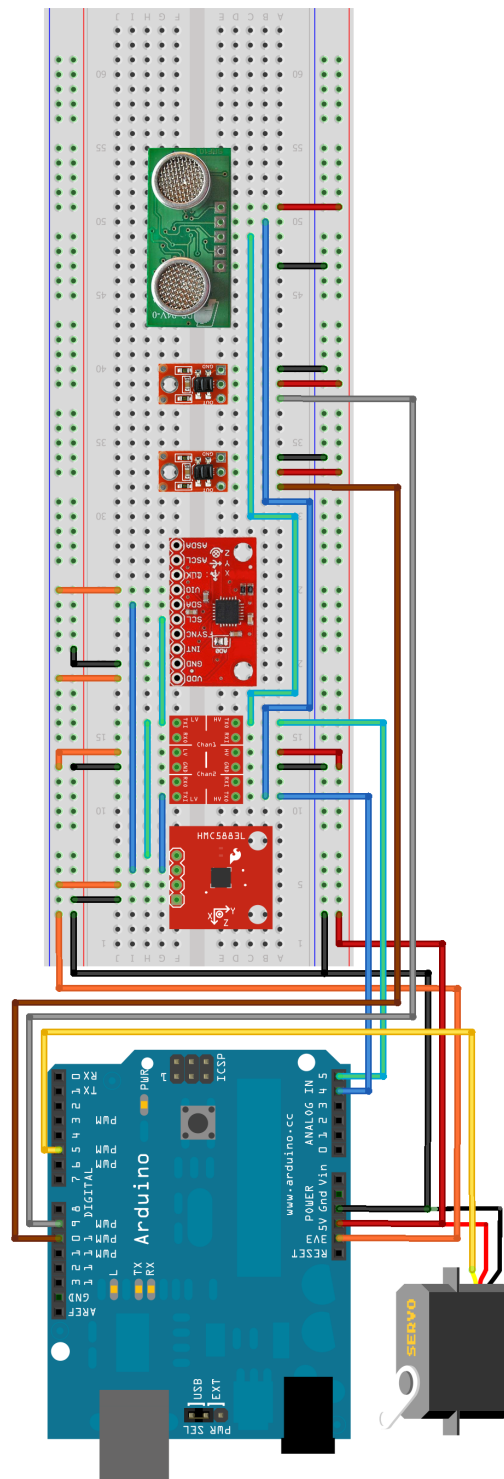
Inerciální měřicí jednotka, magnetometr a úroňový převodník se nachází na horní části robotu v nepájivém poli. Zde bylo nutné si dát pozor pouze na správné zapojení vodičů. Umístění těchto čidel je vidět na obrázku (2.26)



Obrázek 2.26: Inerciální měřicí jednotka, magnetometr a úroňový převodník

Dále by bylo nevhodné, aby komunikace mezi vozítkem a počítačem byla zprostředkována pomocí USB kabelu a tak byly použity bezdrátové komunikační moduly xBee, které dokáží velmi snadno komunikovat s počítačem, který zobrazí poskytnutá data.

Na obrázku (2.27) je vyobrazeno kompletní schéma připojení čidel k Arduino. Schéma bylo vytvořeno opět pomocí programu Fritzing.



Obrázek 2.27: Kompletní schéma zapojení senzorů

3 Matematický model a řízení mobilního robotu

Tato sekce se zabývá odvozením obecného matematického modelu robotu, který lze následně aplikovat jako popis použitého robotu. Pro odvozený matematický model bylo v prostředí v prostředí Simulink¹, které je součástí programu MATLAB², navrženo řízení pro sledování požadované trajektorie.

Co je diferenciálně poháněný robot?

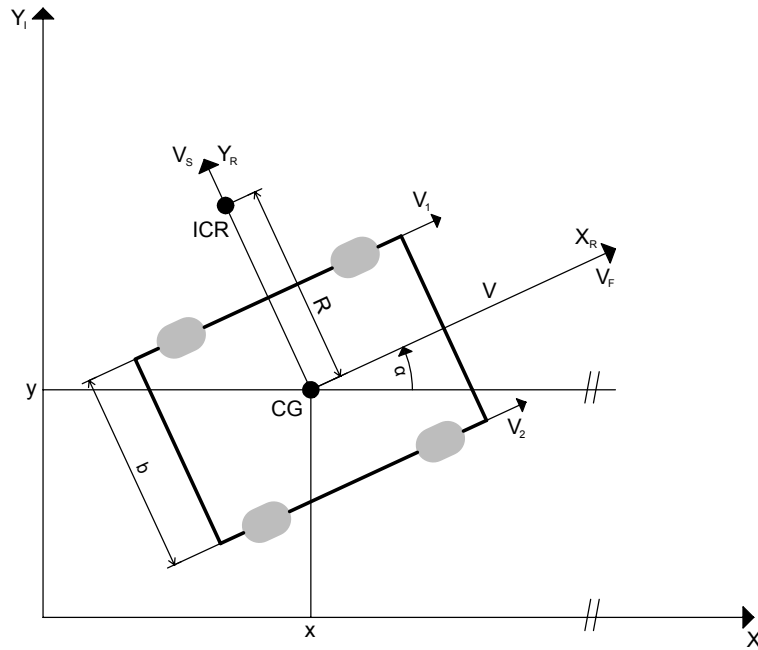
Diferenciálně řízená vozidla jsou kolová nebo pásová vozidla, která se pohybují v prostředí pomocí nezávislých pohonů na obou stranách podvozku. V této práci se použitý podvozek skládá ze dvou nezávisle poháněných dvojic kol. Takto poháněná vozidla jsou v praxi hojně využívána zejména pro práci v náročném prostředí, protože jednou z jejich výhod je velmi dobrá průchodnost obtížným terénem a dobrá manévrovatelnost v takovém terénu. Další jejich nespornou výhodou je také robustní a velmi jednoduchá konstrukce. V pásové podobě se používají téměř výhradně pro vojenské, zemědělské nebo stavební účely. Jedním z nejrozšířenějších pásových vozidel je rolba používaná v horských oblastech, kde se vozidlo pohybuje v měkkém a velmi obtížném prostředí. Druhý typ reprezentují pohony pomocí kol. Tato vozidla jsou často využívána i v méně náročném průmyslovém, nebo ve školním a laboratorním prostředí.

Pohyb robotu a jeho souřadné systémy

Před samotným odvozením rovnic charakterizujících model robotu bude na následujících řádcích uvedeno jak se vlastně robot pohybuje a co bude nutné vědět pro samotné odvození modelu. Pro sestavení matematického modelu robotu bylo nutné učinit několik zjednodušení. Především bylo předpokládáno, že se robot bude pohybovat pouze v rovném terénu a jeho kola budou v neustálém v kontaktu se zemí. Dále budou zanedbány síly vzniklé třením kol a ostatní pasivní síly. Modelovaným robotem je systém sestavující se ze čtyř kol upevněných ke konstrukci podvozku. Ilustrační obrázek podvozku a veličiny, pomocí kterých jsou rovnice modelovány, jsou znázorněny na obrázku (3.1).

¹Nadstavba programu MATLAB pro simulaci a modelování dynamických systémů

²Interaktivní matematické prostředí a skriptovací programovací jazyk vyvíjený společností MathWorks



Obrázek 3.1: Ilustrační obrázek čtyřkolového diferencially řízeného robotu

Z obrázku (3.1) je vidět, že jednou z možností, jak pohyb robotu reprezentovat je pomocí souřadnic v osách x , y a úhlu natočení α . Další možností, jak vyjádřit pohyb robotu jsou rychlosti v a zrychlení a pro každou osu. Mezi veličinami x , v a a platí následující vztahy (3.2). Budeme uvažovat například vztahy pro složky v ose x čili poloha x , rychlost v_x a zrychlení a_x .

$$\begin{aligned} v_x &= \frac{dx}{dt} \\ a_x &= \frac{dv_x}{dt} = \frac{d^2x}{dt^2} \end{aligned} \quad (3.1)$$

V inerciální soustavě značené I může být poloha popsána také vektorem ξ_I , který se skládá z výše uvedených složek x , y a α , viz rovnice (3.3).

$$\xi_I = \begin{bmatrix} x \\ y \\ \alpha \end{bmatrix} \quad (3.2)$$

Pokud bude zapotřebí znát rychlosti v jednotlivých osách, tak se složky vektoru ξ_I zderivují podle času a výsledkem bude vektor $\dot{\xi}_I$ s rychlostními parametry.

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} \quad (3.3)$$

Samotný stav robotu je ovšem reprezentován v souřadném systému, který bude značen písmenem R . Rychlosti ve směrech od tohoto souřadného systému \dot{x}_R a \dot{y}_R jsou podle obrázku (3.1) rovny rychlostem v_f a v_s a ω je úhlová rychlost otáčení kolem ICR³. Tento souřadný systém se používá pro stanovení polohy kol a výpočtu omezujících podmínek. Vektor $\dot{\xi}_R$ bude vypadat takto (3.4):

$$\dot{\xi}_R = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \omega \end{bmatrix} \quad (3.4)$$

Aby bylo možné převádět vztahy mezi oběma souřadnými systémy, bude nutné zavést rotační matici. Pro převod ze souřadného systému R do systému I bude využita rotační matice s označením R_{RI} , viz (3.5).

$$R_{RI} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

Tato matice ovšem slouží pouze k převodu v oblasti rychlostí. Pro převod v oblasti spočtených poloh by musela být použita takzvaná homogenní transformace. Pro tento případ ale bude stačit použití převodu s rotační maticí podle rovnice (3.6).

$$\dot{\xi}_I = R_{RI} \cdot \dot{\xi}_R \quad (3.6)$$

3.1 Odvození kinematických rovnic robotu

Vstupní veličiny systému jsou rychlosti v_1 a v_2 , které udávají rychlost otáčení kol na každé straně robotu. Výstupními veličinami bude poloha těžiště robotu v inerciální soustavě (souřadnice x a y) a úhel natočení

³anglická zkratka "coordinate of instantaneous center of rotation" s významem souřadnice okamžitého středu rotace

robotu (α), viz obrázek (3.1). Za předpokladu, že se robot může pohybovat jen dopředu nebo dozadu a pohyb do boku není umožněn, lze stanovit, že rychlost v_s bude rovna nule a rychlost v_f tak bude rovna rychlosti

$$v = \frac{v_1 + v_2}{2}.$$

Pro rychlosti levých kol v_1 a pro rychlosti pravých kol v_2 platí následující vztahy (3.7), kde r je poloměr kol robotu a $\dot{\phi}_i$ je úhlová rychlost kol.

$$\begin{aligned} v_1 &= r \cdot \dot{\phi}_1 \\ v_2 &= r \cdot \dot{\phi}_2 \end{aligned} \quad (3.7)$$

Pokud jsou rychlosti kol na každé straně rozdílné a zároveň konstantní, robot se pohybuje kolem středu otáčení (ICR), s poloměrem zatáčení. Pro tuto vlastnost platí rovnice (3.8)

$$\begin{aligned} v &= R \cdot \omega \\ v_1 &= \left(R + \frac{b}{2}\right) \cdot \omega \\ v_2 &= \left(R - \frac{b}{2}\right) \cdot \omega \end{aligned} \quad (3.8)$$

kde v je rychlost robotu v jeho těžišti, R je poloměr otáčení od těžiště robotu, ω je úhlová rychlost otáčení robotu a b je rozchod kol (vzdálenost kola jedné strany od kola druhé strany).

Nyní z obou předchozích rovnic (3.7 a 3.8) lze vyjádřit rychlost robotu v v jeho těžišti a úhlovou rychlost ω pomocí úhlových rychlostí kol $\dot{\phi}_i$ (tedy $\dot{\phi}_1$ a $\dot{\phi}_2$), viz rovnice (3.9).

$$\begin{aligned} v &= r \cdot \frac{(\dot{\phi}_2 + \dot{\phi}_1)}{2} \\ \omega &= r \cdot \frac{(\dot{\phi}_2 - \dot{\phi}_1)}{b} \end{aligned} \quad (3.9)$$

Tyto rovnice (3.9) nyní určují rychlostní parametry robotu a můžeme je zapsat do vektoru (3.10), který reprezentuje souřadný systém robotu.

$$\dot{\xi}_R = \begin{bmatrix} v \\ 0 \\ \omega \end{bmatrix} \quad (3.10)$$

Ze znalosti konstrukce podvozku víme, že rychlosti $v = v_f = \dot{x}_R$ a $v_s = \dot{y}_R = 0$ a pokud se porovnájí vektory (3.10) a (3.4) je vidět, že odvození souhlasí.

Pokud se nyní porovnájí vztahy uvedené ve vektorech (3.10) je vidět, že tyto vztahy si odpovídají, protože rychlost $v = v_f = \dot{x}_R$ a $0 = v_s = \dot{y}_R$.

Přenásobením prvního vektoru ze vztahů (3.10) rotační maticí (3.5) se získá diferenciální rovnice popisující časový vývoj stavu odvozeného systému popsaného v inerciální soustavě I pomocí polohy a úhlu.

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ 0 \\ \omega \end{bmatrix} = \begin{bmatrix} v \cdot \cos \alpha \\ v \cdot \sin \alpha \\ \omega \end{bmatrix} \quad (3.11)$$

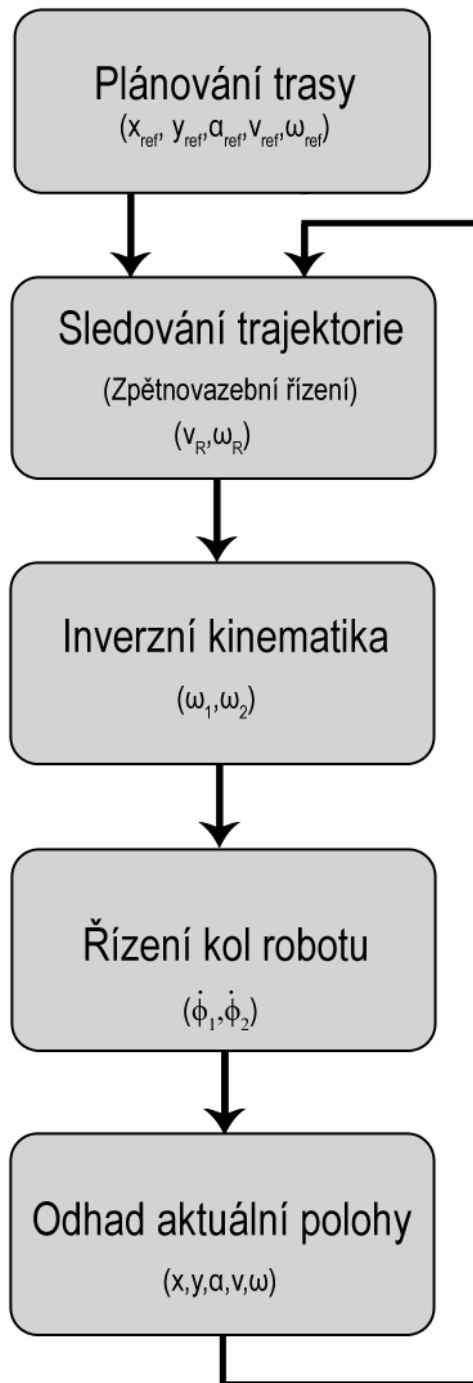
Zpětným dosazením vztahů (3.9) pro rychlost a úhlovou rychlost za příslušné stavové proměnné budou stavové rovnice vypadat následovně (3.12):

$$\begin{aligned} \dot{x} &= r \cdot \frac{\phi_1 + \phi_2}{2} \cdot \cos \alpha \\ \dot{y} &= r \cdot \frac{\phi_1 + \phi_2}{2} \cdot \sin \alpha \\ \dot{\alpha} &= r \cdot \frac{\phi_1 - \phi_2}{b} \end{aligned} \quad (3.12)$$

3.2 Sledování referenční trajektorie

V běžné praxi bývá nutným požadavkem, aby robot dokázal projet naplánovanou trasu. Jak se robot řídí pomocí výpočtu veličin, kterými se ovládá pohyb robotu (v tomto případě úhlové rychlosti kol), je popsáno na následujících řádcích.

Na obrázku (3.2) je vidět základní blokové schéma pro sledování trajektorie. Principem sledování je znalost aktuálních souřadnic a úhlu natočení skutečného robotu a referenčního robotu, který se pohybuje po generované trajektorii. Jsou-li tyto hodnoty k dispozici, lze se na základě zpětnovazebního řízení určit pro každý následující krok úhlové rychlosti kol tak, aby pohyb skutečného robotu souhlasil s pohybem referenčního robotu.



Obrázek 3.2: Blokové schéma pro sledování trajektorie robotu

Aby robot mohl sledovat námi určenou trajektorii, je nutné znát její parametry. Vzhledem k tomu, že matematický model byl odvozen pro inerciální systém dvou souřadnic, je vhodné jako parametry použít souřadnice X , Y a úhel natočení robota α . Díky akcelerometru, gyroskopu a optickým enkodérům máme hned několik možností, jak zjistit i současnou polohu a natočení reálného robota v inerciální soustavě.

Referenční trajektorii, kterou má pohyb robota kopírovat, vytváří blok generátor trajektorie. Tento blok může být tvořen libovolnými časově závislými funkcemi, jako jsou například sinus a cosinus. Důležité je, aby v náhodně vybraném čase z intervalu $(0, T)$, kde T je perioda, byla známa informace o aktuální poloze bodu referenční trajektorie. Pokud jsou známy rychlosti referenčního robota, lze podle nich získat také úhel natočení referenčního robota a to díky matematické funkci $ATAN2^4$, která má dva parametry, v tomto případě právě zmíněné rychlosti v osách X a Y .

$$\alpha_{ref} = ATAN2(\dot{y}_{ref}, \dot{x}_{ref}) + k\pi \quad (3.13)$$

Druhým blokem schématu je sledování trajektorie, které lze implementovat mnoha způsoby. V této práci je robot řízen pomocí stavového regulátoru.

U tohoto stavového regulátoru je nejprve nutné určit odchylku aktuálních souřadnic referenční a skutečné trajektorie. Je zde také implementován zpětnovazební zákon řízení, který stanovuje potřebnou rychlost jízdy a úhlovou rychlost otáčení, díky které robot dosáhne referenční trajektorie.

Odchylka souřadnic se spočítá jako rozdíl aktuálních a referenčních souřadnic robota:

$$\begin{aligned} e_x &= x_{ref} - x \\ e_y &= y_{ref} - y \\ e_\alpha &= \alpha_{ref} - \alpha \end{aligned} \quad (3.14)$$

Jak bylo řečeno v kapitole týkající se matematického modelu, robot má svůj souřadný systém a pro převod ze souřadného systému robota do inerciálního souřadného systému byla použita rotační matice. Nyní je nutné udělat opačnou transformaci - převést souřadnice do souřadného systému robota a k tomu se využije zpětné rotační matice, která má tvar:

⁴Tato funkce vrátí úhel, jehož tangens je podíl dvěma zadanými čísly

$$R_{IR} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$

Převedením do souřadného systému robotu získáme transformované souřadnicové odchylky ve tvaru:

$$\begin{bmatrix} e_{xt} \\ e_{yt} \\ e_{\alpha t} \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} e_x \\ e_y \\ e_\alpha \end{bmatrix} \quad (3.16)$$

Dále je nutné na základě zákona zpětnovazebního řízení stanovit rychlost v_r a úhlovou rychlost otáčení ω_r v těžišti robotu. Pro výpočet těchto rychlostí ovšem musí být známy také obě rychlosti referenčního robotu, tedy rychlost v_{ref} a úhlovou rychlost ω_{ref} , které se spočítají:

$$v_{ref} = \sqrt{\dot{x}_{ref}^2 + \dot{y}_{ref}^2} \quad (3.17)$$

$$\omega_{ref} = \frac{\ddot{y}_{ref}\dot{x}_{ref} + \ddot{x}_{ref}\dot{y}_{ref}}{\dot{x}_{ref}^2 + \dot{y}_{ref}^2}$$

Za použití nelineární transformace dostaneme vztah (3.18) pro výpočet řídicích rychlostí v_r a ω_r

$$v_r = v_{ref} \cos(e_{\alpha t}) - u_1 \quad (3.18)$$

$$\omega_r = \omega_{ref} - u_2$$

kde u_1 a u_2 jsou řídicími vstupy dynamického modelu odchylek ve tvaru

$$\dot{e} = \begin{bmatrix} 0 & \omega_{ref} & 0 \\ -\omega_{ref} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ \sin(e_3) \\ 0 \end{bmatrix} v_{ref} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.19)$$

Podle definice lineárního zpětnovazebního zákona se nadefinují řídicí vstupy dynamického modelu odchylek (3.19) u_1 a u_2 , které reprezentují nastavení stavového regulátoru v dalším kroku řízení a budou ovlivňovány regulačními odchylkami:

$$\begin{aligned} u_1 &= -k_1 e_x \\ u_2 &= -k_2 \text{sign}(v_{ref}) e_y - k_3 e_\alpha \end{aligned} \quad (3.20)$$

Ve vztazích (3.20) figurují parametry zesílení k_1 , k_2 a k_3 , které doposud nebyly zmíněny. Tyto parametry se určí z charakteristické rovnice v uzavřené smyčce, která má obecný tvar

$$(\lambda + 2\xi a)(\lambda^2 + 2\xi a\lambda + a^2) = 0 \quad (3.21)$$

kde parametr $\xi \in (0,1)$ je koeficient tlumení, parametr $a > 0$ je vlastní frekvence. Aby byl celý polynom stabilní, tak vlastní čísla tohoto polynomu musí mít záporné reálné části.

Ze znalosti charakteristického polynomu určíme koeficienty $k_1 - k_3$, které se rovnají

$$k_1 = k_3 = 2\xi a, \quad k_2 = \frac{a^2 - \omega_{ref}^2}{|v_{ref}|} \quad (3.22)$$

Za předpokladu, že k_2 a v_{ref} se budou blížit nekonečnu, lze vyjádřit vlastní frekvenci a jako $a = \sqrt{\omega_{ref}^2 + bv_{ref}^2}$ a po dosazení a vyjádření tohoto vztahu do (3.22) vzniknou rovnice

$$\begin{aligned} k_1 = k_3 &= 2\xi \sqrt{\omega_{ref}^2 + bv_{ref}^2} \\ k_2 &= bv_{ref} \end{aligned} \quad (3.23)$$

kde b je počet stupňů volnosti. Čím větší bude b tím přesnější bude celý výpočet.

Výsledný a nejdůležitější vztah, ke kterému bylo zapotřebí se dopracovat a ze kterého jsou určeny řídicí rychlosti v_r a ω_r , je tedy

$$\begin{aligned} v_r &= v_{ref} \cos(\alpha_{ref} - \alpha) + k_1 [\cos\alpha(x_{ref} - x) + \sin\alpha(y_{ref} - y)] \\ \omega_r &= \omega_{ref} + k_2 \text{sign}(v_{ref}) [\cos\alpha(x_{ref} - x) - \sin\alpha(y_{ref} - y)] + k_3(\alpha_{ref} - \alpha) \end{aligned} \quad (3.24)$$

Odvození těchto rovnic pro řízení robotu bylo použito z materiálu uvedeného ve zdrojích pod číslem [8].

Další blok je nazván inverzní kinematika. Při odvozování kinematického modelu se na základě úhlových rychlostí kol vypočítaly rychlosti robotu v a ω . Zde je ovšem nutné udělat opačný výpočet. Z předchozího bloku je známa rychlost v_r a úhlová rychlost otáčení ω_r a z těchto rychlostí se musí stanovit úhlová rychlost kol. Budou zde platit stejné vztahy jako pro kinematický model, ovšem v obráceném sledu, proto také název inverzní. K jejich určení postačí rovnice (3.7) a (3.8). Dosazením rovnice (3.7) za (3.8) a vyjádřením ω_1 a ω_2 získáme potřebné úhlové rychlosti:

$$\begin{aligned}\omega_1 &= \frac{(R + \frac{b}{2}) \cdot \omega_r}{r} \\ \omega_2 &= \frac{(R - \frac{b}{2}) \cdot \omega_r}{r}\end{aligned}\tag{3.25}$$

Tuto rovnici lze upravit využitím prvního vztahu z (3.8).

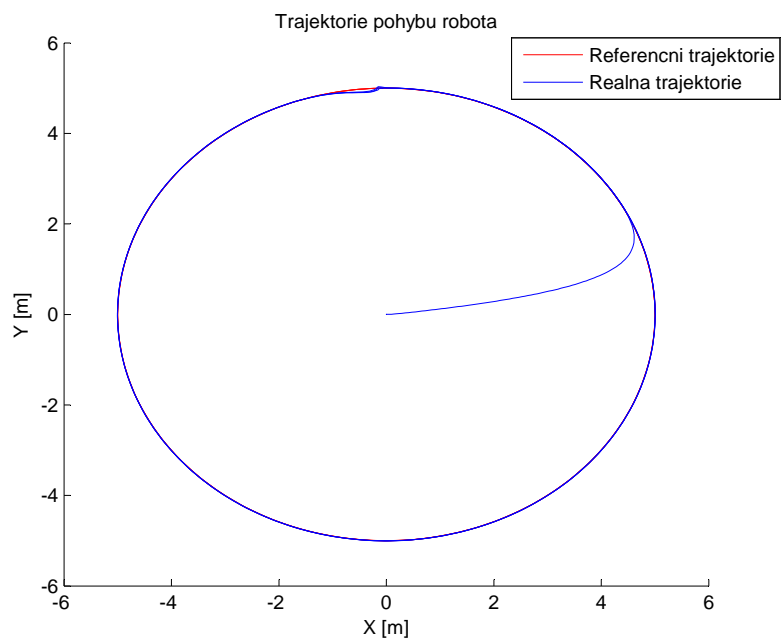
$$\begin{aligned}\omega_1 &= \frac{v_r + \frac{b}{2} \cdot \omega_r}{r} \\ \omega_2 &= \frac{v_r - \frac{b}{2} \cdot \omega_r}{r}\end{aligned}\tag{3.26}$$

Blok regulátoru otáček kol má za úkol sledovat, jestli požadované úhlové rychlosti kol, které byly vypočítány předchozími bloky, odpovídají skutečným otáčkám. Rovnice pro odhad aktuální polohy robotu z rychlostí v a ω jsou uvedeny v kapitole odvození matematického modelu (3.1).

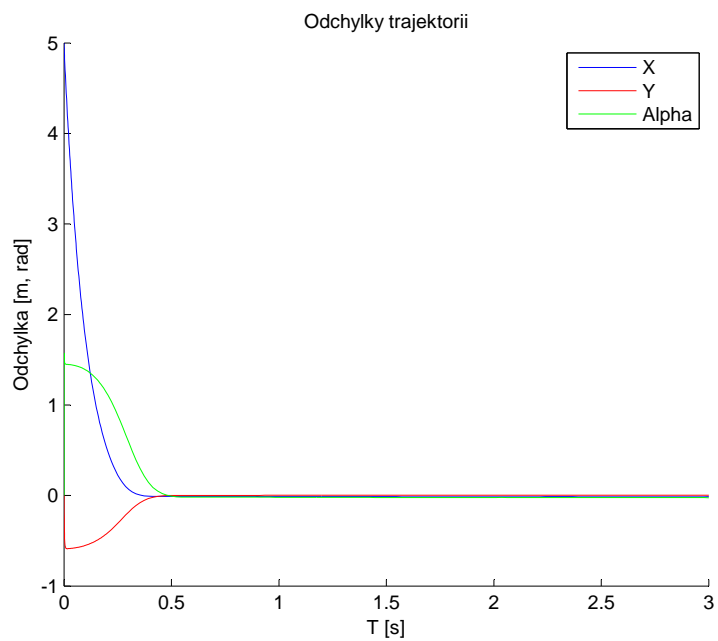
3.3 Simulace řízení

Navržený řídicí algoritmus byl implementován a ověřen v prostředí Matlab/Simulink. Schéma zapojení jednotlivých bloků odpovídá blokovému schématu (3.2). Algoritmus byl ověřen na třech různých trajektoriích s různými počátečními souřadnicemi referenčního robotu.

První ověření proběhlo na kruhové trajektorii. Tato trajektorie se vygeneruje pomocí funkcí sinus a cosinus za předpokladu, že obě funkce mají stejně velkou amplitudu. Na obrázku (3.3) je vidět, že řízení po kružnici je téměř ideální, protože skutečná trajektorie překrývá trajektorii referenční. Na obrázku (3.4) jsou vykresleny odchylky jednotlivých souřadnic. Je vidět, že díky počátečním podmínkám, které byly u obou souřadnic odlišné, je na začátku odlišná i odchylka, ale postupně se odchylka minimalizuje a po zbytek simulace se její hodnota blíží k nule.

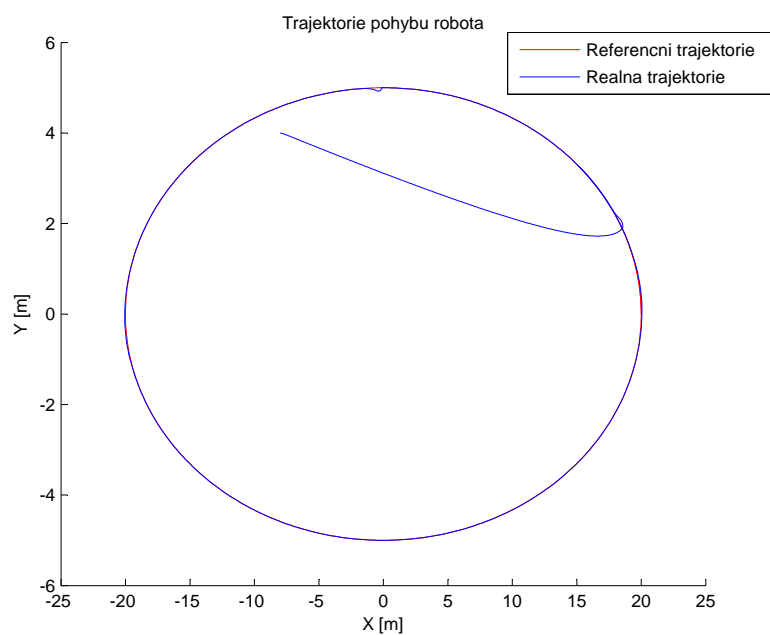


Obrázek 3.3: Řízení na kružnici

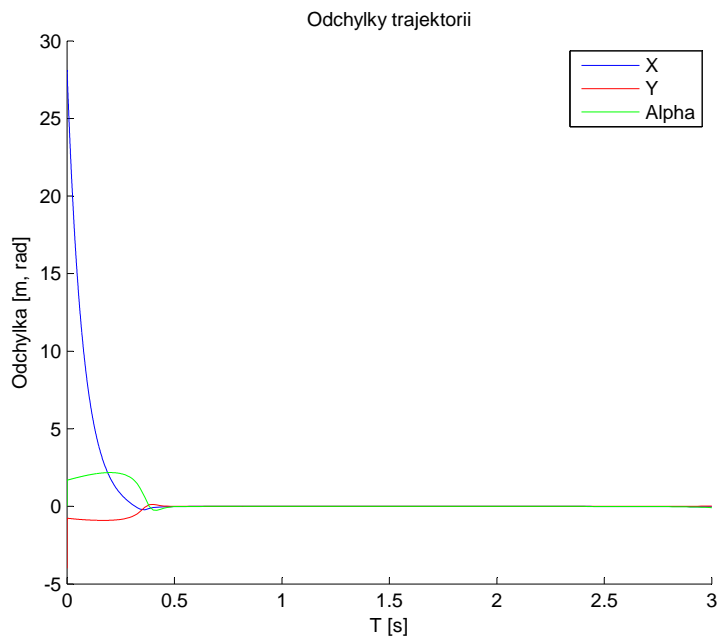


Obrázek 3.4: Odchylka při řízení na kružnici

Další trajektorií, na které bylo řízení vyzkoušeno, je elipsa. Modifikace elipsy je odlišná od kružnice pouze v rozdílné amplitudě. Zatímco funkce cosinus měla v tomto případě amplitudu stejnou, jako v minulé simulaci (5m), u funkce sinus byla amplituda zvětšena na hodnotu 20m. Na obrázku (3.5) je sice na první pohled kružnice, při pohledu na osy je ale vidět, že každá má svou stupnici nastaveno jinak. Po dosažení reálného robotu referenční trajektorie je opět vidět, že robot elipsu kopíruje téměř dokonale. Na obrázku (3.6) jsou vidět, stejně jako v prvním případě, odchylky, které se po necelé půl vteřině opět blíží nulové hodnotě.

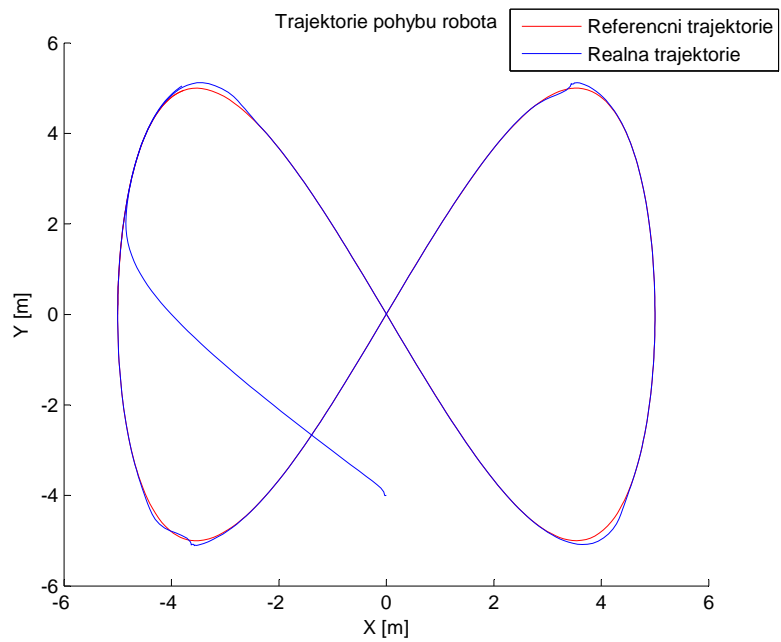


Obrázek 3.5: Řízení na elipse

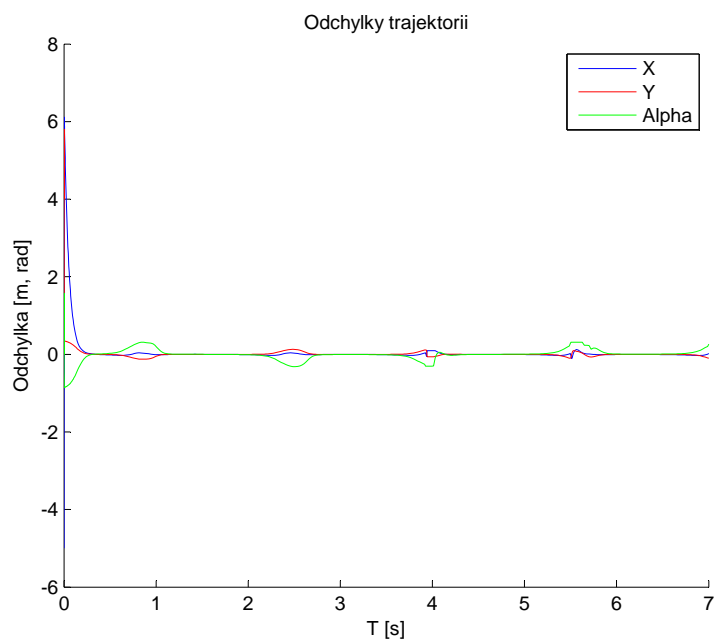


Obrázek 3.6: Odchylka při řízení na elipse

Poslední simulace řízení byla navržena pro průběh "položené osmičky". Modifikace byla opět pomocí stejných funkcí, akorát se neměnila amplituda ale frekvence funkce sinus, která se zdvojnásobila. Z obrázku (3.7) je vidět, že při rychlejších změnách natočení nastávaly drobné odchylky skutečné trajektorie od referenční. Po chvíli se ale robot opět dostal na plánovanou trasu. Odchylky jsou v tomto případě zaznamenány i na obrázku (3.8), kde je, na rozdíl od předchozích případů, vidět drobné odchýlení od požadované trasy. Toto odchýlení je způsobeno nastavením regulátoru. Pokud by parametry charakteristického polynomu (3.21) b a ξ byly nastaveny na vyšší hodnotu, robot by dokázal sledovat trasu přesněji, ale mohlo by to být na úkor stability.



Obrázek 3.7: Řízení na položené osmičce



Obrázek 3.8: Odchylyka při řízení na položené osmičce

4 Závěr

Hlavním cílem této práce byl návrh a kompletace hardwarové části a matematický popis chování diferenciálně řízeného robotu s návrhem řízení pro sledování požadované trajektorie.

V první fázi proběhl výběr vhodných senzorů, které jsou schopny poskytnout data, která se budou dále zpracovávat. Po zakoupení navržených senzorů probíhalo jejich testování za použití softwaru Arduino a volně dostupných knihoven k jednotlivým čidlům. Při testování bylo zjištěno, že některá data poskytnutá z některých čidel vykazují zkreslené informace a že bude zapotřebí využít filtrů, které budou eliminovat chyby měření. Tyto chyby nastaly u akcelerometru a gyroskopu, kde se projevoval značný drift. V navazující práci bude nutné chyby eliminovat pomocí Kalmanova nebo komplementárního filtru. Navržené a použité senzory jsou prozatím ve finální podobě, ale určitě by nebylo špatné je v další fázi projektu rozšířit například o kamerku, která by poskytovala více informací o monitorovaném prostředí, než poskytuje samotné ultrazvukové čidlo.

Druhá fáze byla teoretická a simulační. Podle parametrů diferenciálního robotu byl sestaven jeho kinematický popis, díky kterému jsme schopni chování robotu nejprve odsimulovat a poté daný algoritmus aplikovat na fyzický model. Po odvození kinematických rovnic bylo navrženo jednoduché řízení pomocí stavového regulátoru, které má za úkol kopírovat trajektorii referenčního robotu. Toto řízení bylo ověřeno v simulačním prostředí Simulink. Ze simulací je patrné, že pokud se robot pohybuje s konstantní úhlovou rychlostí, tak dokáže trajektorii kopírovat s minimálními odchylkami. Čím větší změny v úhlové rychlosti nastávají, tím více se pak obě trajektorie liší. To je dáno nastavením robustnosti regulátoru.

Dalším postupem v diplomové práci bude realizace softwarové části robotu, kde bude třeba napsat vhodný algoritmus, který dokáže sledovat pohyb robotu a navrhnout vhodné vizualizační prostředí v programu Processing, kde lze zobrazit naměřená data. V této práci byl předpoklad, že se v dané trase pohybu nebudou nacházet žádné překážky. V navazující části budou v trase robotu uvažovány i náhodně umístěné překážky a úkolem robota bude monitorovat prostředí tak, aby byl schopen vypočítat nejkratší možnou cestu, kterou bude nutné urazit z počátečního bodu do konečného.

Literatura

- [1] Stránky projektu Arduino Obecné informace, parametry použitého kitu
<http://arduino.cc/>
- [2] Robot - Dagu Wild Thumper 4WD Pack Arduino
Použité informace z: <http://robosavvy.com/>.
- [3] I2C sběrnice
Použité informace z:
<http://www.hw.cz/navrh-obvodu/strucny-popis-sbornice-i2c-a-jeji-prakticke-vyuziti-k-pripojeni-externi-EEPROM-24LC256>.
- [4] Sparkfun Informace o použitých čidlech
<https://www.sparkfun.com/>.
- [5] Sypták Michal, Modelování a řízení kolových robotů, 2009
http://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=15185
- [6] Havlát Petr, Simulátor mobilních robotů v prostředí Simulink, 2009
http://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=14874
- [7] Nițulescu, M., Mobile Robot Tracking Control Experiments along Memorized Planed Trajectories, 2006
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4022971
- [8] Alessandro De Luca, Giuseppe Oriolo, Marilena Vendittelli, Control of Wheeled Mobile Robots: An Experimental Overview, 2004
<http://www.dis.uniroma1.it/~labrob/pub/papers/Ramsete01.pdf>
- [9] Mircea Nițulescu, Mobile Robot Tracking Control Experiments along Memorized Planed Trajectories, 2006
http://www.ugr.es/~dblanc/FFI_antiguo/Aplicaciones_Tema2.pdf