

Robotické simulační vozidlo

Laboratoře průmyslové elektorniky a senzorů

Semestrální projekt

Vojtěch Kozel
Katedra měření: B3B38LPE
Fakulta elektornická ČVUT
Praha, Česká republika
kozelho1@fel.cvut.cz

Abstract—Tento dokument pojednává o semestrálním projektu simulační úlohy robotického vozidla. Primárním cílem práce je realizovat bluetooth ovládání vozidla a jízdu stabilizovat za použití gyroskopu a regulátoru. Sekundárním cílem práce je realizovat autonomní mód vozidla pro sledování černé čáry. Terciární cíl je analogové řízení osvětlení vozidla.

Index Terms—bluetooth, PID regulátor, gyroskop, Arduino, H-bridge

I. ÚVOD

Účelem projektu je simulování reálné situace provozování vozidla se samostatnými náhony předních kol, tj. každé z kol má samostatný motor. Problém situace spočívá ve skutečnosti, že dva elektromotory při napájení stejným napětím nepodávají stejné otáčky (mechanické nepřesnosti, různá indukce). Zároveň těžiště vozidla není v jeho geometrickém středu a kola při obvodu nejsou kružnice (deformace pneumatik).

Z důvodu výše uvedených nepřesností vozidlo při plném napájení obou motorů (jízda vpřed) zahýbá doleva. Projekt realizuje regulování této chyby při manuálním ovládání vozidla pomocí rozhraní bluetooth a při autonomním módu sledování černé čáry.

II. MECHANICKÁ KONSTRUKCE

Mechanickou konstrukci řeším pomocí kovové stavebnice Merkur.¹ Viz obrázky 1, 2 a 3. Délka, šířka (rozchod kol) a maximální výška konstrukce jsou po řadě 22 cm, 13 cm a 10.5 cm.

A. Podvozek

Podvozek se skládá z jednoho otočného kola 360° (poloměr 2 cm) a dvou kol (poloměr 2.5 cm) s pohonem (každé se samostatným elektromotorem). Rozchod kol s náhonem je 13 cm.

B. Adaptéry

Na konstrukci jsou dva adaptéry tužkových baterií, každý po 4 bateriích.

¹MERKUR: <http://www.merkurtoys.cz>

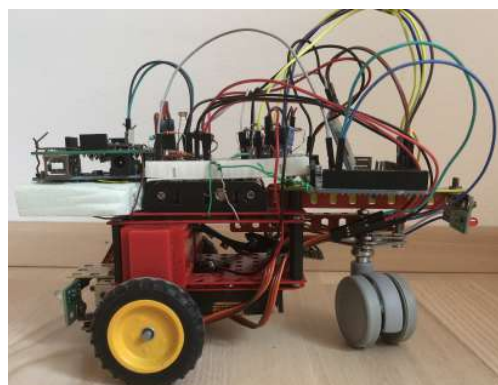


Fig. 1: Vozidlo: pohled z boku.

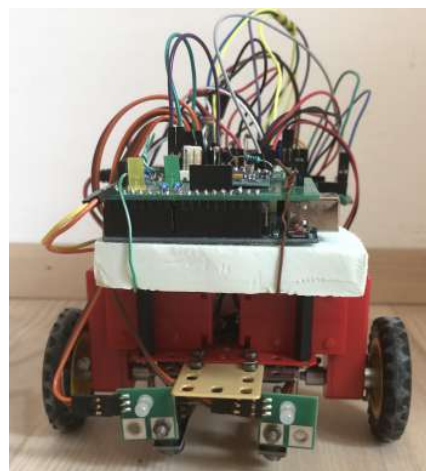


Fig. 2: Vozidlo: pohled z čela.

C. Řídící jednotka

Řídící jednotku pro gyroskop MPU6050 používám Arduino UNO, nachází se v přední části vozidla. V prostřední části je umístěné nepájivé pole pro zapojení obvodů a v zadní části je umístěna hlavní řídicí jednotka Arduino MEGA 2560.

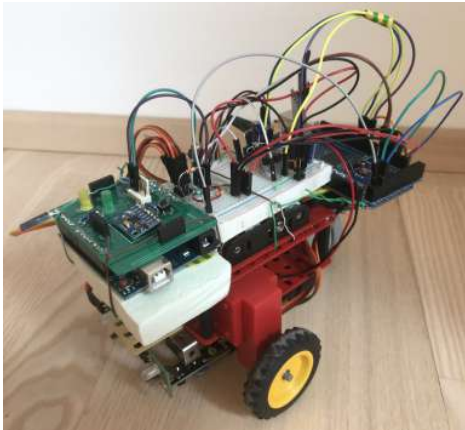


Fig. 3: Vozidlo: pohled z poloprofilu.

III. ZAPOJENÍ OBVODŮ

Schéma kompletního zapojení je v příloze na obrázku 7.

A. H-bridge

Motory ovládám prostřednictvím PWM. Jejich spínání řeším pomocí Hbridge. Můstek má oproti běžnému spínání jedním tranzistorem výhodu změny polaroty na spotřebiči. Lze tedy ovládat otáčení motoru oběma směry. Schéma H-bridge je v příloze na obrázku 5. Jelikož je elektromotor principálně indukční zátěží, umístil jsem do zapojení z důvodu eliminace napěťových špiček diody D1 až D4.

B. Řídící obvod osvětlení

Pro dynamické řízení osvětlení jsem sestavil obvod založený na fotorezistoru. Čím více světla (fotonů) na fotorezistor dopadá, tím více elektronů se uvolňuje z valenční sféry a zvyšuje se tak elektrická vodivost přechodu (snižuje se elektrický odpor fotorezistoru).

Regulátor osvětlení tak funguje na principu proměnlivého odporového děliče. Fotorezistor ve tmě má vysoký odpor a vzhledem k zapojení na obrázku 6 dochází k sepnutí NPN tranzistoru a rozsvícení LED diody. Výsledkem je plynulé rozsvícení LED diod v závislosti na klesání intenzity okolního světla. Rovněž je možné spínat osvětlení přes mikroprocesor.

C. Senzor černé barvy

Pro rozpoznání černé čáry na vozovce jsem využil komerční řešení v podobě senzoru TCRT5000 od firmy Vishay Semiconductors². Senzor pracuje na principu infračervené diody a fototranzistoru. Dioda vysílá záření, které se od stínítka odráží, v případě, že stínítko je tmavé barvy (resp. černá čára), záření se odráží méně (resp. je téměř pohlceno). Schéma zapojení je v příloze na obrázku 6. V závislosti na odrazu záření se otevírá resp. zavírá fotorezistor a výstupní signál je nastaven na logickou nulu nebo jedničku.

²Vishay Semiconductors:
<https://www.vishay.com/docs/83760/tcrt5000.pdf>

D. Gyroskop

Gyro MPU6050 je připevněno na shieldu na desce Arduino UNO. Informace o úhlu natočení vozidla okolo osy Z posílám z desky UNO do desky MEGA 2560 prostřednictvím sériové komunikace.

IV. BLUETOOTH OVLÁDÁNÍ

Ovládací aplikaci na PC jsem implementoval v jazyce Python. Vlastnosti ovládací aplikace:

- Volba módu vozidla (manuální ovládání / autonomní sledování černé čáry),
- Sepnutí světel vozidla (nezávisle na okolním osvětlení).

V případě manuálního ovládání jízdy aplikace snímá klávesy (šipky) a podle toho odesílá instrukce do vozidla. Pro přeposílání informací mezi PC a vozítkem volím sériovou komunikaci prostřednictvím bluetooth čipu HC-06. Zatáčení vozidla při manuálním ovládání řeším způsobem opačného směru otáčení motorů (např. otočení vpravo: levý motor se otáčí směrem vpřed a pravý motor směrem vzad).

V. REGULÁTOR JÍZDY VPŘED

Jak jsem psal již v úvodu, při stejném napětí (stejná frekvence PWM) podávají elektromotory různé otáčky a vozidlo tak při jízdě vpřed zatáčí doleva. Z toho důvodu jsem musel navrhnout regulátor. Vzhledem ke skutečnosti, že neznám stavový popis systému (sestavení soustavy rovnic, která by vozidlo popisovala, by bylo značně náročné), jsem využil experimentální metodu návržení konstant PID regulátoru (vzhledem ke vzorkování gyroskopu 100 ms se jedná o diskrétní regulátor). Rovnice diskrétního absolutního PID regulátoru má tvar:

$$y[k] = K_P E[k] + \sum_i K_I E[k] \Delta t + K_D \frac{E[k] - E[k-1]}{\Delta t}, \quad (1)$$

kde K_P je zesílení proporcionální složky, K_I zesílení integrační složky, K_D zesílení diferenční složky, $E[k]$ je informace o chybě a $y[k]$ je výstup regulátoru.

Při začátku jízdy vpřed (resp. vzad) se v programu uloží referenční úhel natočení vozidla (setpoint). Od něj se se vzorkovací frekvencí 100 ms odečítá aktuální úhel gyroskopu a vypočítává se tak aktuální odchylka od referenčního úhlu (error). Vstupem PID regulátoru je informace o chybě (error) a výstupem je údaj (0 až 255), který se odečítá od PWM hodnoty (0 až 255) ovládání pravého motoru. Schéma regulátoru je na obrázku 4.

V principu regulace funguje následovně: levý motor má nastavené maximální napětí a hodnota pravého motoru rozdíl "255 - údaj PID regulátoru", čímž se eliminuje zatáčení vlevo. Při návržení konstant PID regulátoru jsem postupoval podle následujících bodů: [1]

- 1) Nastavil jsem K_P , K_I (zesílení integrační složky), K_D (zesílení derivační složky) na nulu.

- 2) Po malých přírůstcích jsem zvyšoval K_P až do dosažení přijatelného chování systému (rychlost odezvy a jejich kmitavost), tj. vozidlo se při jízdě vpřed na dráze přibližně třech metrů odchýlilo od referenčního úhlu o 10°
- 3) Po malých přírůstcích jsem zvyšoval K_I až do přijatelného snížení regulační odchylky. Se zesilováním integračního členu se sice snižuje regulační odchylka a systém se stává stabilnějším, ale nevýhodou je snížení rychlosti odezvy.
- 4) Po malých přírůstcích jsem zvyšoval K_D , čímž se zvýšila rychlost odezvy systému na chybu. Obecně nevýhoda derivátoru je zesílení šumu, což se při vyšším zesílení může promítat na výstup systému v podobě oscilace.

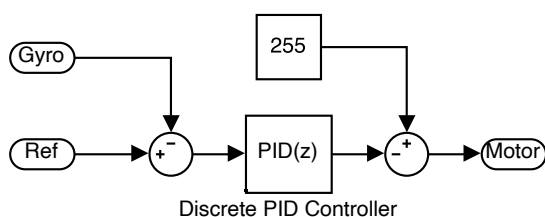


Fig. 4: Schéma regulátoru pro jízdu vpřed.

VI. AUTONOMNÍ SLEDOVÁNÍ ČÁRY

A. Úvod

Pro detekci černé čáry využívám dva senzory TCRT5000 s roztečí 4 cm ve výšce 0.7 cm nad vozovkou. Pro tuto konstrukci postačuje jako vozovka hladký, světlý dřevěný povrch a jako černá vodící čára 1 cm široká elektrická páska. Princip algoritmu: vozidlo je ve startovní pozici, kdy se čára nachází mezi senzory. V případě, že jeden ze senzorů snímá černou čáru a druhý ne, algoritmus zaznamená chybu a spustí regulaci zatáčení. Například situace, kdy levý senzor je nad černou čárou a pravý ne: je nutné zatočit doleva — pravý motor se nastaví na maximální napětí a hodnota na levém se snižuje podle výstupu regulátoru.

B. Regulátor

Při návrhu regulátoru jsem vycházel z modelu PID. Vstupem regulátoru je pouze logická nula, resp. logická jedna (senzor je, resp. není nad černou čárou). Z toho důvodu jsem musel nahradit diferenční složku regulátoru novým členem (v následujícím algoritmu "differential value"). Tento člen se výrazně projevuje na začátku regulace (zvýrazní šum, resp. první reakci na kontakt s vodící čárou) a po jedné sekundě od kontaktu se vyruší.

Maximální výstupní hodnota regulátoru je 200. Jedná se o ochranu, aby nedošlo k úplnému zablokování kola.

Algorithm 1 Algorithm for line follower controller

Input: forward, // indicates if a change has occurred
: K_P , K_I , K_D // controller constants

Output (global variable): regulation difference

Initialisation :

1: differential value \leftarrow 10

2: integral \leftarrow 0

3: forward \leftarrow actual order

LOOP Process

4: **while** not forward **do**

5: integral \leftarrow integral + 1

6: **if** (differential value > 0) **then**

7: differential value \leftarrow differential value - 1

8: **end if**

9: **end while**

10: computed value \leftarrow $K_P + K_I \cdot$ integral + $K_D \cdot$
differential value

11: **if** (computed value > 200) **then**

12: computed value \leftarrow 200

13: **end if**

Vzhledem k podobnosti vytvořeného regulátoru s PID regulátorem jsem při nalezení konstant zesílení složek postupoval Ziegler-Nicholsonovou metodou. [2]

- 1) Nalezl jsem kritické zesílení K_r proporcionální složky při kterém se systém dostane na mez stability. Toto zesílení jsem experimentálně určil jako 42.
- 2) Změřil jsem periodu kmitů (přibližně $T_r = 3$ s).
- 3) Zesílení jednotlivých složek regulátoru jsem vypočetl jako:

$$\begin{aligned}
 K_P &= 0.6K_r = 25.2, \\
 K_I &= \frac{1.2K_r}{T_r} = 16.8, \\
 K_D &= \frac{3K_r T_r}{40} = 9.45.
 \end{aligned} \tag{2}$$

Na základě výsledků experimentálních testů jsem zvýraznil okamžitou reakci na kontakt s čárou nastavením K_D na hodnotu 14.

VII. ZÁVĚR

Modelovou úlohu považuji za splněnou, jelikož se mi podařilo realizovat všechny tři cíle. Ukázková videa manuálního ovládání a autonomní jízdy jsou dostupná na <https://youtu.be/90INTfowXT4>. Zdrojové kódy včetně podoby ovládací aplikace jsou dostupné na <https://gitlab.fel.cvut.cz/kozelvo1/robotic-simulation-vehicle/-/tree/master>.

REFERENCES

- [1] KLÁN, Petr. Moderní metody nastavení PID regulátorů: Nastavení metodou pokusů [online]. Ústav informatiky AV ČR Praha [cit. 2020-05-04]. Dostupné z: https://valter.byl.cz/sites/default/files/soubory/moderni_nastav_pid.pdf
- [2] Ziegler-Nichols Tuning Rules for PID: Tuning Rule. Microstar Laboratories, Inc. [online]. [cit. 2020-05-04]. Dostupné z: <http://www.mstarlabs.com/control/znrule.html#Ref4>

PŘÍLOHA

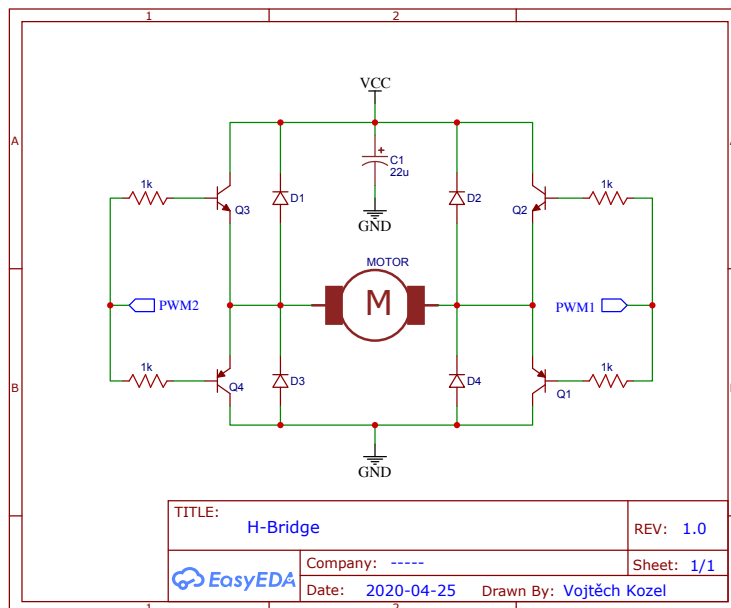
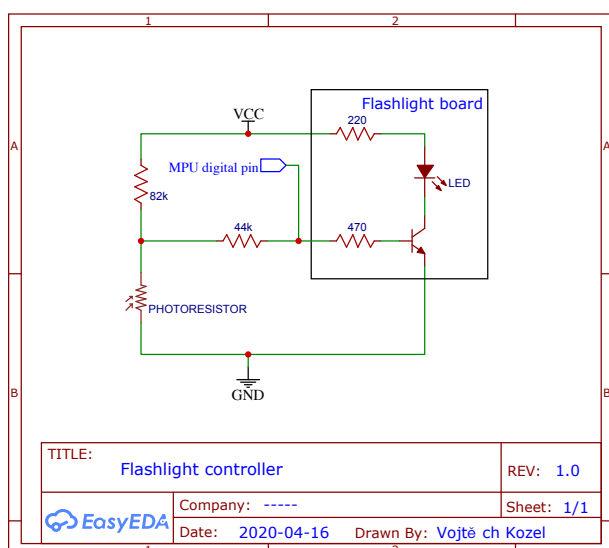
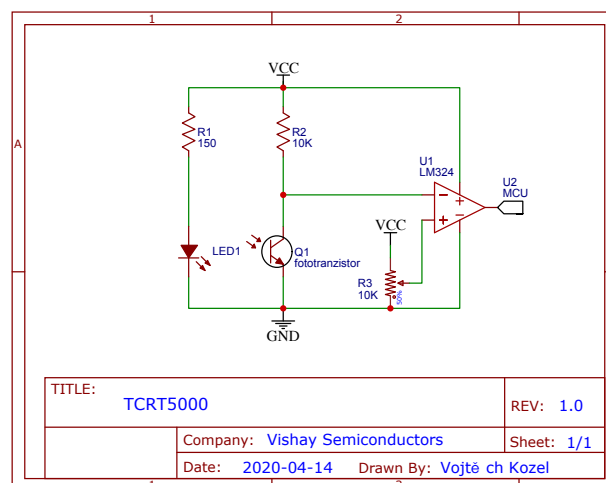


Fig. 5: Schéma zapojení: H-bridge



(a) Flashlight controller



(b) TCRT5000

Fig. 6: Schéma zapojení

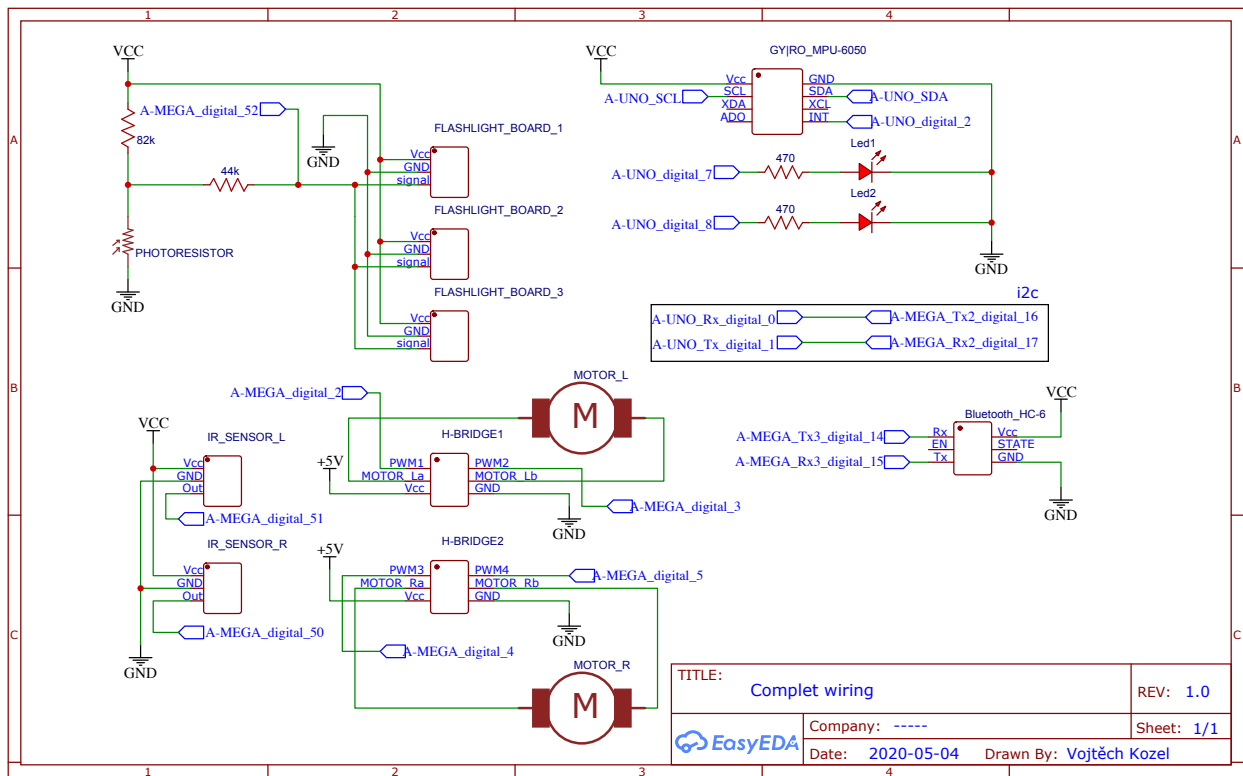


Fig. 7: Kompletní schéma zapojení.