Dokumentace projektu BPC-PRP

Vojtěch Vařecha
Fakulta informačních technologií
Vysoké učení technické v Brně
Brno, Česká Republika
xvarec06@vutbr.cz

Vojtěch Růžička
Fakulta informačních technologií
Vysoké učení technické v Brně
Brno, Česká Republika
xruzic56@vutbr.cz

***Abstrakt*—V rámci kurzu BPC-PRP se na připraveném robotovi implementují 3 hlavní úkoly – jízda po čáře, jízda koridorem a únik z bludiště. Implementace probíhá na vyšší úrovni abstrakce za využití systému ROS2 a jazyka C++. Robot disponuje mj. diferenciálním podvozkem, IR snímači čáry, LiDARem, ultrazvukovými senzory vzdálenosti a kamerou.**

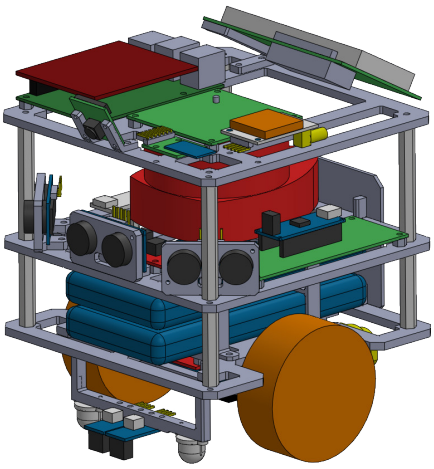
***Klíčová slova*—Robotika, projekt, BPC-PRP, ROS2, labyrint**

I. Úvod

Podstatou projektu v předmětu BPC-PRP je využít již sestaveného robota a implementovat postupně řešení tří úkolů (jízdy po čáře, jízdy uprostřed koridoru a úniku z bludiště).

Připravený robot (viz Obrázek 1) se pohybuje s využitím diferenciálního podvozku, kde obě kola pohání motory propojené pro zajištění zpětné vazby s rotačními enkodéry. Pro sledování čáry je vybaven dvěma infračervenými senzory, pro jízdu v koridoru a vyhýbání se překážkám pak může využít rotační LiDAR a tři ultrazvukové senzory vzdálenosti. Dále může využít IMU, kameru, tři integrovaná tlačítka a čtyři RGB LED (především pro signalizaci svého stavu), viz také návod k sestavení robota [2].

Na robotovi se nachází počítač Raspberry Pi, na kterém běží systém ROS2 [5]. S tímto počítačem komunikuje přes WiFi uživatelský počítač, kde je (opět s pomocí ROS2) spouštěn program ovládající činnost robota.



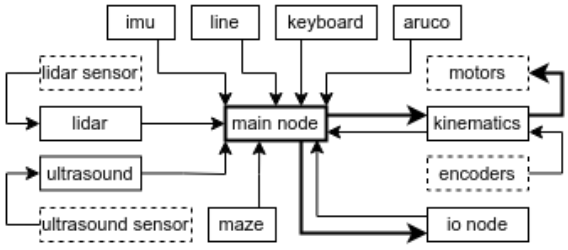
Obrázek 1: Model robota (převzato z [4])

Prvním úkolem robota je za pomoci infračervených senzorů sledovat čáru (tj. jet podél ní). Čára je v jednom případě rovná, v druhém případě tvoří kruh a v posledním případě tvoří obrys čísla 8. Druhým úkolem je, s využitím LiDARu, ultrazvukových senzorů nebo kombinace obojího, projet koridorem tak, aby se robot držel uvnitř a nedotýkal se stěn. Chodby mají stejné tvary jako čáry, tedy linie, kruh a osmička. Posledním úkolem je (zjednodušeně) s využitím libovolných senzorů robota a libovolné strategie projet bludištěm. Bludiště je tvořeno již známými koridory, kde se před každou křižovatkou vedoucí k cíli nachází ArUco kód napovídající správný směr odbočení. Kompletní pravidla jsou k dispozici v kapitolách 2.8, 2.12 a 2.13 zadání laboratoří a zkoušek [3].

II. Řešení projektu

Implementace jednotlivých úkolů a dílčích částí řešení probíhala postupně. Nejdříve bude představen globální pohled na architekturu programu, následně chronologicky vytvořené části.

Program řídící činnost robota byl implementován v jazyce C++ a ve své architektuře s drží zvyklosti systému ROS2 [5] používat uzly (*ROS nodes*), které mají různé zodpovědnosti. V našem řešení existuje jeden hlavní uzel (`main node`), který sbírá informace ze svých vstupních uzlů a publikuje pokyny pro uzly, které sledují jeho výstupy.



Obrázek 2: Program z celkového pohledu

Celkový pohled na komunikaci námi vytvořených uzlů je ukázán na Obrázku 2. Uzly přímo napojené na hlavní uzel jsou nakresleny plnou čarou a komunikace přímo ovlivňující vnější stav robota je vyznačena tlustšími šipkami. Je vhodné zde také zmínit, že většina uzlů na obrázku dále interaguje se systémovými uzly robota, které již byly pro naši implementaci předpřipraveny.

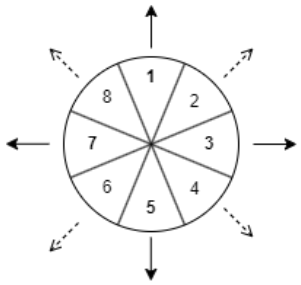
Vlastnost	Hodnota
Poloměr kola	35 mm
Obvod kola	22 mm
Rozchod kol	128 mm
Pulzy enkodéru	576 za rotaci

Tabulka I: Vlastnosti robota

Aby byl robot schopen řízeného pohybu, byl implementován uzel `kinematics`, který se stará o kinematiku diferenciálního podvozku a využívá pomocné uzly `motors` a `encoders`. Uvnitř tohoto uzlu jsou mj. zadány vlastnosti robota vypsané v Tabulce I.

Jízda po čáře je pak implementována v uzlu `line`, a to s řízením typu *bang-bang* i s řízením pomocí PID regulátoru (jehož implementaci využíváme i v dalších uzlech). Konkrétní hodnoty jednotlivých složek regulátoru jsou $K_p = 1$; $K_i = 0,001$ a $K_d = 0,1$. Za zmínku také stojí, že po spuštění naší implementace sledování čáry provede robot na počátku kalibrační pohyb, při kterém určí aktuální hodnoty dosažitelné na čáře a mimo ni. Tím je schopen se přizpůsobit různým světelným podmínkám i různým odstínům povrchu a čáry.

Obdobně je implementován i pohyb koridorem s pomocí ultrazvukových senzorů, tedy existuje i *bang-bang* verze i PID verze řízení průjezdu. Složky PID (resp. PD) regulátoru zde mají hodnoty $K_p = 1$; $K_i = 0$ a $K_d = 0,4$. Ultrazvukové senzory jsou znovu využity i u implementace jízdy v koridoru s pomocí LiDARU, kde se aktivují v případě přílišného přiblížení ke stěně chodby (v naší implementaci vždy, když LiDAR přestává fungovat spolehlivě a vrací neplatné výsledky).



Obrázek 3: Rozdělení laloků LiDARu

U průjezdu chodbou s pomocí LiDARu jsme opět implementovali i *bang-bang* i PID řízení, pro pozdější pohyb bludištěm ale využíváme pouze PID regulátor (lépe řečeno PD, hodnoty složek jsou totožné s těmi u ultrazvukových senzorů). Jelikož LiDARová jednotka robota rotuje a snímá celé jeho okolí, pomyslný kruh je v naší implementaci rozdělen na 8 výsečí, jak je znázorněno na Obrázku 3. Při běžném pohybu chodbou, kde je jeho cílem držet se v jejím středu, využívá robot laloků označených sudými čísly a čárkovanými šipkami. Ve chvíli, kdy v chodbě detekuje nepravidelnost (potenciální křižovatku), využije laloky označené lichými čísly a plnými šipkami pro určení jejího typu a následně své další trasy.

Detekovanou vzdálenost, resp. přítomnost nepravidelnosti, robot v naší implementaci signalizuje pomocí změny barvy světla zabudovaných LED. Změnou barvy světla jedné ze 4 diod také signalizuje vybraný směr další cesty.

V uzlu `aruco` se nachází implementace poslední dílčí části projektu, a to detekce a získání hodnoty ArUco kódů pomocí robotovy kamery. Největší část implementace obstarává knihovna OpenCV [1], kterou pro detekci a přečtení kódu naše implementace využívá. Uzel přečtenou hodnotu ArUco kódu pouze předává dál, aby ji měla k dispozici část programu zabývající se řízením trasy robota.

Konečným úkolem robota je projet bludištěm, což je v naší implementaci řešeno dvěma strategiemi. První z nich pouze určuje, že robot bude sledovat pravou stěnu bludiště (což časem vede k nalezení cíle, jelikož se v bludišti nenachází žádné smyčky) a druhá potom využívá přítomnosti ArUco kódů, které napovídají nejkratší cestu k cíli. V případě, že druhá strategie selže, např. z důvodu „přehlédnutí“ kódu, aplikuje se druhá strategie – robot by měl tedy v každém případě dorazit do cíle.

V průběhu řešení projektu byla také navíc implementována možnost ovládat činnosti a jízdu robota s pomocí klávesnice počítače (na čemž spolupracují uzly `keyboard` a `main node`). Různým činnostem byla přiřazena písmena, např. `L` spouští jízdu s pomocí LiDARu a `U` s pomocí ultrazvukových senzorů. Navíc je možné v případě potřeby ovládat klávesami `W`, `A`, `S` a `D` jízdu s pomocí ultrazvuku ohyby vpřed, vlevo, vpravo a vzad.

III. Závěr

S výslednou implementací našeho projektu je robot schopen sledování čáry, jízdy v koridoru s použitím ultrazvukových senzorů i LiDARu, načítání ArUco kódů a průjezdu bludištěm s využitím dříve popsaného. Navíc signalizuje aktuální směr jízdy, typ detekované křižovatky a jeho pohyb lze ovládat z klávesnice. Hlavní limitací tak zůstává propustnost WiFi sítě, po níž spolu robot a počítač, který robota ovládá, komunikují.

Odkazy

[1] G. Bradski. “The OpenCV Library”. In: *Dr. Dobb's Journal of Software Tools* (2000).

[2] A. Ligocki a J. Minařík. *Fenrir Project - Universal Robotic Platform*. 1. vyd. Brno, březen 2025. URL: https://robotics-but.github.io/fenrir-project/.

[3] A. Ligocki, P. Šopák a J. Minařík. *BPC-PRP - Practical Robotics and Computer Vision*. 1. vyd. Brno, květ. 2025. URL: https://robotics-but.github.io/BPC-PRP/.

[4] A. Ligocky. “Course Introduction. Robotics and Computer Vision (BPC-PRP)”. online. Podklady k přednášce. Brno, ún. 2025. URL: https://github.com/Robotics-BUT/BPC-PRP/blob/master/keynotes/bpc-prp-1-introduction.pdf.

[5] S. Macenski et al. “Robot Operating System 2: Design, architecture, and uses in the wild”. In: *Science Robotics* 7.66 (2022), eabm6074. DOI: 10.1126/scirobotics.abm6074. URL: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074.