

BPC-PRP Projekt

Filip Botlo

Fakulta informačních technologií
Vysoké učení technické v Brně
Brno, Česká republika
filip.botlo@gmail.com

Jakub Kolář

Fakulta elektrotechniky a komunikačních technologií
Vysoké učení technické v Brně
Brno, Česká republika
247395@vutbr.cz

Abstrakt—

Klíčová slova—robotika, BPC-PRP, lidar, IMU, PID

I. ÚVOD

Článek se zaměřuje na programové vybavení autonomního robota, jehož úkolem je vyjet z bludiště sestaveného z buněk 40 na 40 cm o maximální rozloze 8 na 8 buněk. K řízení robota byl využit framework ROS 2, který usnadňuje integraci komponent, jako jsou senzory a řízení pohybu. ROS 2 podporuje vícejádrové procesory a komunikaci přes middleware DDS, což ho činí vhodným pro real-time a distribuované aplikace.

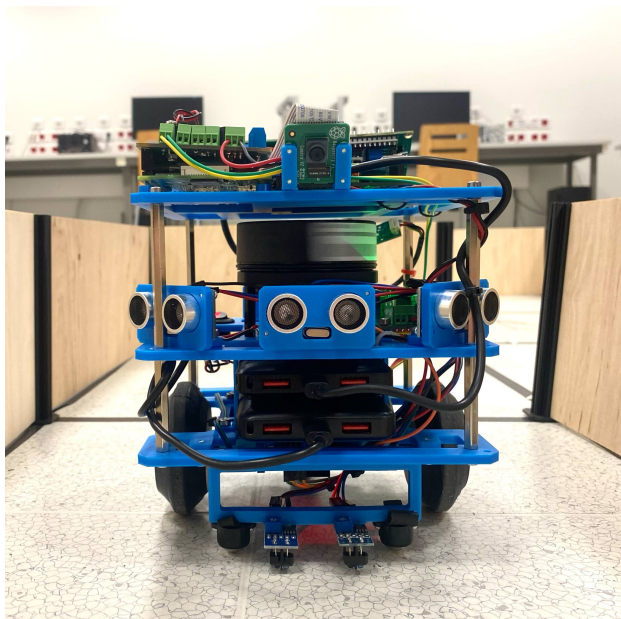


Fig. 1. Robot

Z konstrukce robota vyplývá, že je neholonomní, což znamená, že stupeň volnosti pohybu robota (pohyb v jedné ose a rotace) je menší než stupeň volnosti prostoru (3 DOF). O pohyb robota se starají kola poháněná motory na stranách.

Pro řízení robota používáme vybrané komponenty, které umožňují detekovat okolní prostor a orientaci robota v něm. LiDAR je používán k detekci překážek (okolních stěn), díky kterým je možné řízení pohybu robota v prostoru. IMU v projektu slouží k řízení otáčení robota v křižovatkách. O

detekci Aruco tagů, které poskytují informaci o nejkratší cestě z bludiště, se stará kamera.

II. ŘÍZENÍ ROBOTA V KORIDORU

V počátečním stavu se robot nachází ve stabilní pozici, kdy probíhá kalibraci IMU. Teprve až po stisknutí prostředního tlačítka začne robot vykonávat řízený pohyb. V rámci řízení pohybu v node nazvaný *pid* robot přepíná mezi 2 stavy, a to *DRIVE_FORWARD* a *TURNING*.

Řízení robota ve stavu *DRIVE_FORWARD* spočívá v nastavování rychlostí otáčení do levého a pravého kola. Tyto rychlosti jsou nastavovány PD regulátorem. Na základě pozice robota v prostoru se mění relativní odchylka robota od ideální pozice (*error*), pomocí níž se vypočítá akční zásah do levého a pravého kola. Odchylka je pro filtraci brána jako průměr ze dvou po sobě jdoucích hodnot (*error_avg*) a ve výpočtu akčního zásahu (*speed_diff*) násobena proporcionální konstantou *Kp*. Akční zásah se dále skládá z derivační složky určené rozdílem dvou posledních odchylek *d_error* násobené derivační konstantou *Kd* (rovnice 1).

$$speed_diff = Kp * error_avg + Kd * d_error \quad (1)$$

Na levé kolo je pak nastavena základní rychlost (142) mínus hodnota *speed_diff*, na pravém kole je tato hodnota k základní rychlosti přičtena. Hodnota *speed_diff* je programem satureována na interval $\langle -6, 6 \rangle$. Relativní odchylka se počítá podle aktuální specifické pozice robota v prostoru. Tato data jsou získána z určených úhlů na lidar.

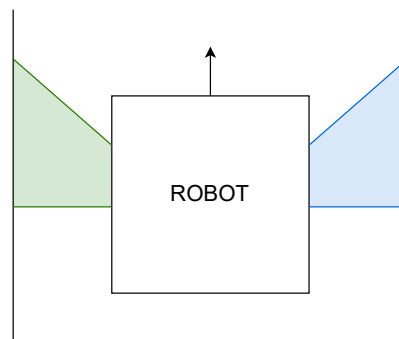


Fig. 2. Relativní pozice robota v koridoru při řízení na základě obou stěn

Nejjednodušší a nejspolehlivější typ řízení, které robot využívá, je řízení v koridoru. V tomto případě robota obklopují z obou stran stěny (obr. 2), takže je relativní odchylka určena rozdílem výšečí ve výhledu lidarů vlevo a vpravo. Robot se při tomto řízení snaží vycentrovat svou pozici vzhledem k okolním stěnám. Pokud právě robot vyjel ze zatáčky a nachází se blízko u jedné ze zdí (vzdálenost ode zdi je pod stanovenou úrovní), k akčnímu zásahu se přičte/odečte hodnota 2 (podle znaménka *speed_diff*), aby se akční zásah zvětšil a došlo k rychlejšímu vyregulování odchylky.

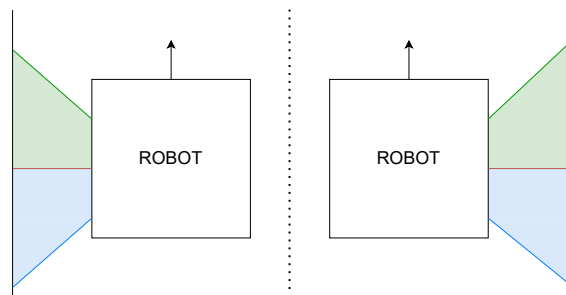


Fig. 3. Relativní pozice robota v koridoru při řízení na základě jedné stěny

Pokud robot kolem sebe nemá obě stěny, orientuje se za pomoci pouze jedné stěny. Konkrétně pouze té, která je blíž. To demonstruje obr. 3. Odchylka, která je vstupem pro výpočet rychlosti kol, se pak počítá z rozdílu hodnoty v jednotlivých vyznačených výšečích. Této regulace se využívá i ve stavu stavu po otočení robota, který narazil do stěny. Při rotaci doleva se následně robot řídí pravou stěnou, při rotaci doprava levou stěnou. Tento typ regulace je pevně nastaven po 3 vteřiny po otočce.

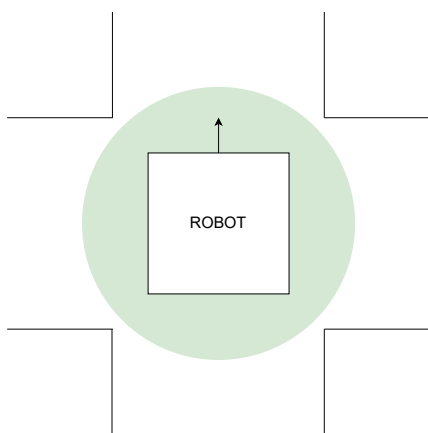


Fig. 4. Relativní pozice robota v křižovatce +

Poslední situace, ve které se robot může nacházet, je křižovatka typu „+“ (obr. 4). V této fázi robot při vjezdu spoléhá na vyregulování pozice v předchozím koridoru a dále se již spoléhá pouze na jízdu rovně, protože nevidí záchytný bod, kterého by se mohl řídit.

III. CHOVÁNÍ ROBOTA PŘI ZMĚNĚ SMĚRU JÍZDY

Pohyb robota v bludišti krom regulace jízdy rovně dále zahrnuje otáčení. Robot se při rotaci řídí na základě dat z inerciály. Při počátečním postavení se provede kalibrace, kdy se zaznamená úhel v této pozici. Při požadované rotaci robota v bludišti se pak každá tato rotace vztahuje k počátečnímu nakalibrovanému úhlu. Povolené rotace jsou pouze v násobcích $k \cdot 90^\circ$ od kalibrovaného úhlu.

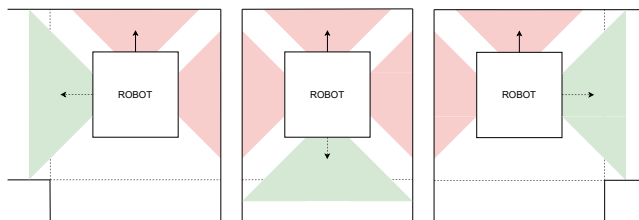


Fig. 5. Robot v povinné zatáčce

A. Povinné zatáčky

Prvním typem otáček je takzvané povinné otáčení, které nastává, jestliže robot před sebou lidarem detekuje překážku a má možnost se otočit pouze na jednu ze stran. V tomto případě se robot rozhoduje o otočce na základě dat z bočních paprsků lidarů. V případě, že robot detekuje místo ve vedlejší buňce na jedné ze stran a na druhé ne, zatočí právě tam, kde je volno o 90° . Pro slepé místo v bludišti robot nedetekuje prostor ani v jedné z vedlejších buněk, a tak rovnou provede otočku o 180° .

B. Zpracování ArUco značek

Pro zpracování křižovek využíváme kameru, která detekuje ArUco značky a používá je pro rozhodnutí o směru. Konkrétně jsou zpracovány a uloženy ve funkci **aruco_callback**, která nám umožňuje výběr mezi značkami s **jednocifernými** ID (vedoucími k východu) a **dvoucifernými** ID (vedoucími k pokladu). Následně se ID ukládá do proměnné **last_aruco_id**, jejíž hodnotu program uchovává, dokud není použita na některé z křižovek.

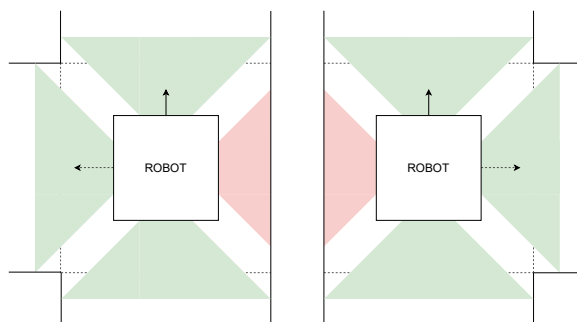


Fig. 6. Robot v křižovatce s volitelným výběrem směru

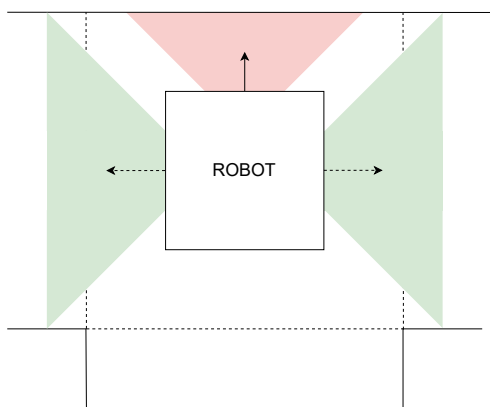


Fig. 7. Robot v křižovatce s volitelným výběrem směru

C. Zatáčky s výběrem

Druhým typem otáček jsou **otáčky s výběrem**, kde se robot řídí předem nasnímanými ArUco značkami. V první možnosti se robot *zastaví před stěnou* a má na výběr mezi **pravou** a **levou** stranou, přičemž si stranu zvolí podle výše zmíněné proměnné. Ve druhé možnosti robot *jede v koridoru* a když mine volnou stěnu, také si podle proměnné vyhodnotí, zda bude pokračovat rovně, nebo odbočí.

IV. SHRUTÍ PROJEKTU

Autíčko má pomocí IMU, LIDARu a kamery implementovaný systém, který mu umožňuje navigovat se a úspěšně uniknout z bludiště ve 3D prostoru.

ACKNOWLEDGMENT

Vestibulum erat nulla, ullamcorper nec, rutrum non, non-ummy ac, erat. Aliquam erat volutpat. Etiam sapien elit, consequat eget, tristique non, venenatis quis, ante. Maecenas sollicitudin.