



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

KOMPLEXNÍ NÁVRH ZALÉVACÍHO MOBILNÍHO ROBOTU

COMPLEX DESIGN OF WATERING MOBILE ROBOT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Ondřej Podolinský

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Jiří Krejsa, Ph.D.

BRNO 2022

Zadání diplomové práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	Bc. Ondřej Podolinský
Studijní program:	Mechatronika
Studijní obor:	bez specializace
Vedoucí práce:	doc. Ing. Jiří Krejsa, Ph.D.
Akademický rok:	2021/22

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Komplexní návrh zalévacího mobilního robotu

Stručná charakteristika problematiky úkolu:

Jedním z projektů řešených na VUT FSI je projekt zalévacího autonomního mobilního robotu, jehož cílem je navrhnout zařízení, které se bude autonomně pohybovat ve vnitřním prostředí, a zde detekovat, klasifikovat a zalévat kytky. Na tomto projektu se podílí řada studentů FSI. Podstatou zadání této diplomové práce je především stanovení ideové koncepce řešení, dekompozice problému, komplexní návrh systému pro konkrétní platformu a integrace výsledků souvisejících prací.

Cíle diplomové práce:

- 1/ provedte rešerši stávajících řešení a analyzujte vhodnost použití robotu pro daný účel
- 2/ navrhnete ideovou koncepci řešení a provedte dekompozici řešené úlohy
- 3/ Vytvořte návrh systému pro konkrétní platformu
- 4/ Provedte integraci dílčích modulů třetích stran a simulačně ověřte

Seznam doporučené literatury:

KROB D (ed): Complex Systems Design & Management, Springer, 2021

NISE N.S.: Control Systems Engineering, Wiley, 2015

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2021/22

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Tato práce se zabývá návrhem zalévacího mobilního robotu. V práci je popsán aktuální stav a podoba závlahových systémů. Na základě toho je specifikována ideální oblast využití robotických závlah. Pro vybranou optimální oblast použití je provedena ideová dekompozice systému a je vyhotoven systémový návrh zalévacího robotu, který využívá dostupnou robotickou platformu. V neposlední řadě se práce zabývá jeho realizací formou integrace dílčích vyhotovených částí v rámci projektu a jeho testováním.

Summary

This master's thesis deals with a design of an irrigation mobile robot. It contains a review of modern irrigations systems and their characteristics. The thesis defines optimal fields of activity of robotics irrigation systems and presents a theoretical solution for one of them. The system draft of the irrigation mobile robot is designed on the foundation of the theoretical solution and it utilizes an available platform. The thesis contains details about the integration of partial sections of the project as well as testing.

Klíčová slova

robotika, ROS, závlahové systémy

Keywords

robotics, ROS, irrigation systems

PODOLINSKÝ, O. *Komplexní návrh zalévacího mobilního robotu*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2022. 93 s. Vedoucí doc. Ing. Jiří Krejsa, Ph.D.

Prohlašuji, že jsem tuto práci zpracoval samostatně s využitím zdrojů uvedených v seznamu na konci této práce.

Ondřej Podolinský

Děkuji Janu Bajerovi, Miroslavu Dosedělovi, Jiřímu Sladkému a Peteru Vizvárymu za příjemnou a inspirativní spolupráci na projektu. Děkuji též vedoucímu práce doc. Ing. Jiřímu Krejsovi, Ph.D. a doc. Ing. Stanislavu Věchetovi, Ph.D. za cenné rady, podněty a podporu při tvorbě této práce. Dále děkuji Ing. Miroslavu Čeplovi i dalším kolegům z firmy Bender Robotics za neocenitelnou pomoc a rady při tvorbě této práce. V neposlední řadě mnohokrát děkuji i své rodině a přátelům za možnost se touto prací i studiem zabývat.

Ondřej Podolinský

Obsah

1	Úvod	3
2	Problematika zavlažování rostlin	4
2.1	Závlahové systémy	5
2.1.1	Stacionární	6
2.1.2	Mobilní	9
2.1.3	Robotické	10
2.2	Porovnání vlastností závlahových systémů	12
2.3	Optimální využití robotických zavlažovacích systémů	14
2.3.1	Ilustrační příklad	14
3	Definice úlohy	18
3.1	Stanovení cíle práce	18
3.2	Vymezení úlohy a konkretizace zadání	18
4	Ideové řešení systému a dekompozice úlohy	20
4.1	Vymezení systému a jeho částí	21
4.2	Návrh funkcionality - use case	21
4.2.1	Hledání rostlin	22
4.2.2	Zalévání rostlin	23
4.3	Identifikace bloků v systému	24
4.3.1	Technické bloky	24
4.3.2	Organizační bloky	27
4.4	Shrnutí ideového návrhu	28
5	Návrh pro cílovou platformu	29
5.1	Základní technologie	30
5.1.1	Platforma Breach	30
5.1.2	ROS	30
5.2	Návrh funkcionality	32
5.2.1	Provozní fáze - start, provoz, vypínání	32
5.2.2	Zalévání rostlin	32
5.3	Identifikace procesů	35
5.4	Shrnutí konkrétního návrhu robotu	42
6	Integrace systému	43
6.1	Základní technologie	43
6.1.1	Git	43

6.2	Prototyp systému	44
6.3	Souhrn balíčků systému	47
7	Ověřování systému	48
7.1	Simulační ověřování	48
7.1.1	Provedení simulace	49
7.1.2	Provedení testování	50
7.1.3	Výsledky testování	50
7.2	Ověřování na robotu Leela	52
7.2.1	Popis robotu	52
7.2.2	Provedení testování	53
7.2.3	Výsledky testování	54
7.3	Ověřování prototypu na robotu Breach	56
7.3.1	Provedení testování	56
7.3.2	Výsledky testování	56
8	Závěr	59
9	Seznam použitých zdrojů	61
10	Seznam použitých zkratk a symbolů	65
11	Seznam příloh	66
12	Přílohy	67
12.1	Pracovní diagramy řídicího programu	67
12.2	Prototyp robotu	70
12.3	Testování systému	72
12.4	Tabulky měření	81

Úvod

Robotika prodělává v posledních letech bouřlivý vývoj. Člověk je svědkem neustálého posunu jejích možností v různých oblastech. Přes veškerý pokrok v těchto technologiích je však dobré mít stále na paměti jakýsi filozofický přesah tohoto druhu techniky - oproštění člověka od stereotypní, náročné a duševně nerozvíjející práce.

V posledních letech jsme svědky velkých změn, které se dějí ve všech koutech světa. Mezi největší patří zajisté dlouhodobá změna klimatu. Ta se výrazně projevuje v krajině. Každý domácí pěstitel rostlin, zahrádkář, lesník, či zemědělec i v našich zeměpisných šířkách v posledních letech pocituje výrazný tlak na péči o rostliny. Bez kontinuálního sledování aktuálního stavu krajiny a provádění vhodných zásahů - zalévání, hnojení, provádění postřiků atd., prakticky nelze dospět ke kvalitní sklizni plodin, či úspěšnému pěstování.

Tato práce si určitě neklade za cíl řešení těchto problémů v globálním měřítku. Pokládá si však otázku, zdali právě onen filozofický přesah robotiky nemůže alespoň lokálně pomoci s dopady těchto změn. Robot jakožto inteligentní stroj by mohl převzít tuto péči do vlastních aktuátorů.

Cílem této práce bude tedy návrh takového robotu, který by byl tuto péči schopen provádět. Jako nejvhodnější a nejlépe realizovatelná oblast zájmu na tomto poli byla vybrána problematika zalévání rostlin, která je součástí oněch klíčových činností během pěstování rostlin. Důležitým aspektem této práce je i odpověď na otázku, zdali je vůbec relevantní využívat schopností robotu na provádění této práce, popř. jestli je na tomto poli působnosti ještě pro něj volné místo.

Vzhledem ke komplexnosti návrhu i samotné realizace robotu, bylo již v prvotní fázi přistoupeno k řešení, které by bylo provedeno v rámci týmového projektu. Bližší informace o týmu i o rozdělení jednotlivých prací lze nalézt v kapitole o ideovém návrhu 4.3.2. Úkolem této práce je tedy i koordinace celého projektu, která by měla vést k jeho úspěšné realizaci.

Kap. 2

Problematika zavlažování rostlin

Se závlahou rostlin se setkává každý člověk téměř každý den ve svém životě, ať již přímo při péči o svou budoucí úrodu, či o své domácí rostlinstvo, nebo nepřímo při využívání výsledků závlahy - každodenním nákupem zeleniny počínaje a posezením pod vzrostlým stromem konče. Tyto příklady jsou krásným ukazatelem toho, jakou významnou roli umělá závlaha ve světě hraje.

Z pohledu zemědělské výroby patří závlaha do tzv. meliorací, to je soubor dosavadních vědecko-technických poznatků, které zajišťují zlepšení kvality půdy. Závlaha je definována jako činnost, jejímž výsledkem je zvlhčení půdy, porostu nebo přízemní vrstvy vzduchu, které zajišťuje optimální výsledky v rostlinné výrobě [1]. Tato definice je pouhým nepatrným zlomkem toho, co všechno tato komplexní problematika zahrnuje. V dnešní době ji lze už chápat jako celou soustavu ekologických, technických nebo ekonomických vlivů, které společně působí na půdu. Proto moderní zavlažové systémy je třeba zkoumat ze všech možných úhlů pohledu, neboť špatný jednostranný přístup může mít katastrofální důsledky ve všech zmíněných oblastech.

Závlahou rostlin se zabývá lidská civilizace od nepaměti. První zmínky o umělých zavlažovacích a odvodňovacích kanálech pocházejí už z dob starověkého Egypta, kdy byla v povodí řeky Nilu využívána tato technologie k zajištění stálosti úrody. Tyto vymoženosti našly své uplatnění i v ostatních vyspělých starověkých civilizacích - Čínou a Indií počínaje až po starověké Řecko a Řím. Známý jsou takové systémy i od indiánských civilizací ze Střední Ameriky. Pro tento účel lidstvo vytvořilo mnoho technických děl, jejichž následovníci jsou v širokém měřítku stále využívány - akvadukty, umělé nádrže nebo vodní kolo. [1, 2]

Prudký rozvoj závlah nastává s počátkem 19. století, kdy dochází k větší systemizaci biologických a ekologických poznatků a technika začíná umožňovat stavbu složitějších zavlažovacích děl. Nemalou roli hraje i rostoucí počet obyvatel Země, kdy je nutné zintenzivnit výnosy ze zemědělských ploch. Tento překotný vývoj ukazují data v tabulce 2.1, kdy je vidět růst zavlažovaných ploch v průběhu času.

Ve světě je zavlažováno zhruba 21% plochy obdělávané půdy, na této ploše je však vypěstována zhruba třetina všech plodin [2, str. 32]. Bez závlahových systémů by v některých státech světa byla silně omezena, nebo dokonce znemožněna produkce některých zemědělských produktů. Svou roli však hrají závlahy nejen na rozsáhlých zemědělských plochách, nýbrž i v malém měřítku v městských parcích, zahradách, sklenících nebo rodinných zahrádkách.

V současnosti dochází vzhledem k celosvětové změně klimatu k častějším extrémům počasí v jednotlivých oblastech světa. Je jasné, že závlahové potažmo meliorační systémy budou hrát stále významnější roli. Ukázkou tohoto směřování je stát Izrael, kdy v důsledku

Rok	Zavlažovaná plocha [mil. akr]	Lidská populace [mld. lidí]
1900	100	1,5
1950	235	2,5
1970	422	3,7
1990	598	5,3
1997	669	5,9
2017	806	7,5

Tab. 2.1: Růst zavlažované plochy a lidské populace od poč. 20. st [2, str. 27, převzato, přeloženo]

nedostatku vody je zemědělství nuceno ke zvyšování efektivity závlahových systémů. V dohledné době několika desetiletí bude nutné řešit podobné problémy i v oblastech, kde jsou dnes podmínky pro pěstování rostlin příznivé. Ztráty v zemědělské výrobě se, bohužel, mohou přetavit do socio-ekonomických problémů, které mohou ještě více oddálit řešení následků těchto změn.

Závlahové systémy jsou tedy starou, ale za to nesmírně důležitou technologií. Ideovým záměrem této práce je zkoumání možnosti rozšíření jejich rodiny o ty, které by byly založeny na robotických mobilních platformách. Ty by mohly za některých podmínek a při určitém využitím, které bude rozebráno dále, zaplnit prázdná místa ve využívání závlahových systémů, či jejich využívání zefektivnit.

2.1 Závlahové systémy

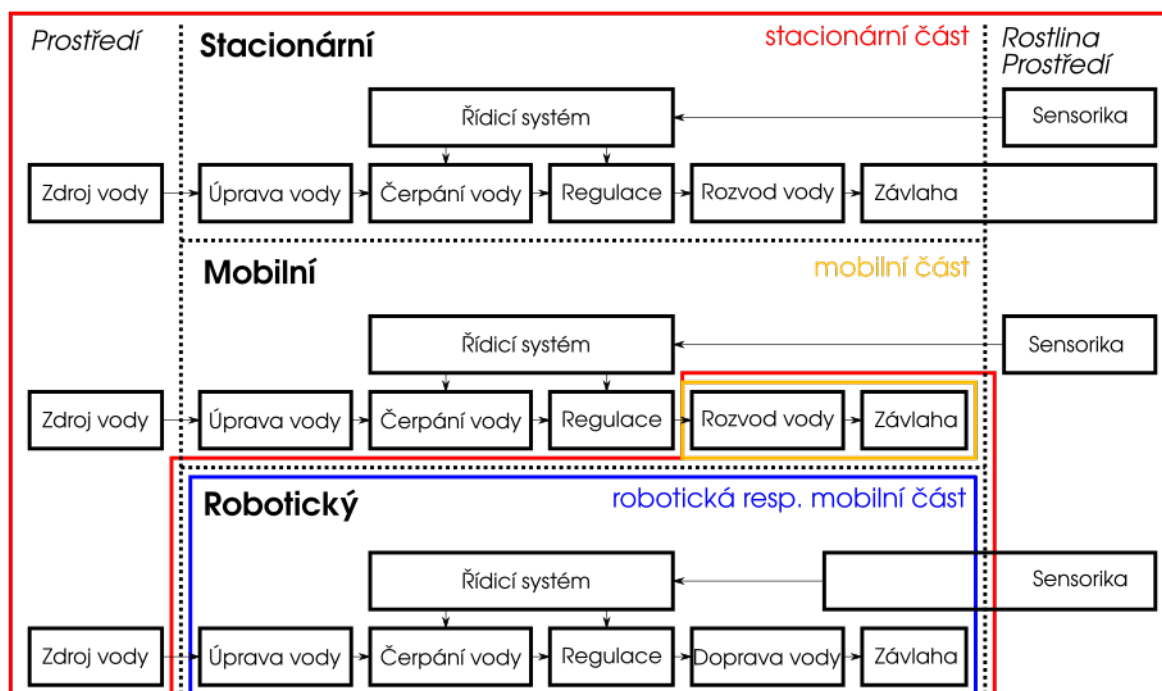
Hlavním úkolem závlahového systému je doprava vody k rostlině za podmínek, že rostlina nesmí být poškozena, musí být dodáno optimální množství vody, dodání vody po zavlažované ploše musí být rovnoměrné a mezi provedenými lokálními závlahami na jednotlivých úsecích plochy by měla být minimální časová prodleva. [1, str. 354] V dnešní době se v rámci závlahy řeší i hnojení či regulace mikroklimatu. Ve všech zavlažovacích systémech lze vysledovat několik základních stavebních prvků. Ty jsou naznačeny v diagramu 2.1.

Ze zdroje vody je odebírána voda, která je posléze upravena a dopravena pomocí zvolené metody k řídicímu členu, který reguluje množství vody dodané přímo k závlaze. V soustavě je přítomen kontrolní člen, který sleduje stav soustavy a předává dále tyto informace řídicímu členu.

Jednotlivé bloky mohou být provedeny různými způsoby - např. v oblasti transportu vody - od ohromných vodních děl, čerpacích stanic, kilometrových kanálů až po mikrozávlahu provedenou kapilárami. Jednotlivé funkce v závislosti na měřítku může vykonávat stroj i člověk.

Konkrétní varianty provedení bloků pak lze dělit podle různých více či méně objektivních kritérií - měřítko systému, prostředí (venkovní, vnitřní), množství dopravované vody, umístění (povrchové, podpovrchové), typu zavlažovaného objektu (plocha, bod, linie) atd..

Vzhledem k charakteru práce bylo zvoleno dělení na technické provedení jednotlivých funkčních bloků v již existujících široce rozšířených stacionárních, resp. mobilních závlahových systémech a na jejich provedení v experimentálních robotických.



Obr. 2.1: Systémový pohled na obecný stacionární, mobilní a robotický závlahový systém

2.1.1 Stacionární

Stacionární závlahové systémy dnes tvoří majoritní část závlah ve světě. Jejich realizace se objevuje v různých měřítkách a technických provedeních. Jejich celková koncepce je vždy v zásadě stejná a je naznačena na diagramu 2.1.

Ze zdroje vody je odebírána závlahová voda, která je dle úrovně znečištění a charakteru upravena. Poté je nejčastěji pomocí čerpadla dopravena potrubím ke koncové závlaze. Rozvod vody je rozdělen do jednotlivých potrubních sekcí, které jsou spínány regulačními ventily na pokyn řídicí jednotky.

Zdroj vody Zdroj vody tvoří klíčovou část závlahového systému. Jako zdroj jsou v zásadě využitelné čtyři prvky - sběr srážkové vody, odběr z lokální studny, odběr z vodovodu, nebo odběr vody z vodního toku.

Voda z různých zdrojů se liší v mnoha ohledech - v tvrdosti, koncentraci jiných organických a anorganických látek, množství a typu obsažených mikroorganismů, znečištění, teplotě, v množství, které lze odebrat, ceně, radioaktivitě atd.. Dle druhu odebírané vody je posléze nutné volit jednotlivé prvky závlahové soustavy. [3, str. 23]

Úprava vody Úpravu vody je nutné provádět dle požadavků jednotlivých zavlažovaných rostlin a požadavků technického řešení závlahy. Často je prováděna s intenzitou odpovídající jejímu znečištění. Můžeme ji rozdělit na mechanickou, chemickou a biologickou.

Mechanické metody spočívají zejména v odstranění nežádoucích pevných částic ve vodě. Používají se k tomuto účelu filtry různých typů a velikostí (hrubá, jemná) - od jednoduchých lamelových filtrů, hydrocyklonů, pískových filtrů až po česla a síta. Často se využívá fyzikálních jevů sedimentace atd.. [3, str. 38]

Chemická a biologická úprava vody má častěji za úkol odstranění nežádoucích chemických příměsí, či různých mikroorganismů.

Úprava vody vnáší do závlahové soustavy pozitivní prvek ve smyslu zlepšení kvality vody, ale též určité nežádoucí vlastnosti - snížení tlaku v soustavě v důsledku tlakové ztráty na filtračním prvku, zvýšení doby, po kterou čeká soustava na závlahovou vodu, další potřebu údržby atd..

Čerpání vody Čerpání vody je základním článkem transportu vody ve stacionárních systémech. Lze využít dvou způsobů - gravitačního a čerpání pomocí mechanických čerpadel.

Gravitačního způsobu je možné využít v situaci, kdy je k dispozici na místě závlahového systému lokální výškový rozdíl u zdroje vody. Často je využíván u velkých vodních děl.

Čerpadel je možno využívat prakticky v libovolném měřítku. Lze je rozdělit podle typů určení - na tlakovací, využívané pro dopravu vody v soustavě a cirkulační, využívané např. pro rozmíchávání hnojiva v nádrži při fergitaci¹.

Tlakovací čerpadla je pak možné dělit na ty pro čistou vodu a kalová. Dle použití na ponorná - uzpůsobená pro stálou práci pod vodou a povrchová. Dle způsobu konstrukce dále můžeme mít čerpadla odstředivá, membránová, peristaltická a zubová. Ty se často liší životností, max. průtokem i tlakem. Při využívání jednotlivých čerpadel je nutné zvažovat i možnost samonásávání a zajistit, aby čerpadlo nepracovalo po dlouhou dobu bez vody z důvodu přehřátí.

Pro potřeby tlakových systémů se často využívají tlakové stanice - tj. soustava čerpadla, systému ventilů, tlakové nádoby a manometru. Tyto stanice umí udržovat tlak v uzavřené tlakové soustavě v zadaných mezích.

Rozvod vody Rozvod vody je hlavním prostředkem dopravy vody v systému. Dle měřítka a určení závlahy se může jednat o soustavu potrubí, kanálů, tunelů, nebo jejich kombinací.

Při návrhu je potřeba uvažovat, zdali bude daný závlahový systém povrchový, nebo podpovrchový, uzavřený, či otevřený. V závislosti na určení je nutné volit správný průměr potrubí pro dostatečný průtok. V závislosti na tlaku v potrubí je nutné zvolit správnou tloušťku.

Z hlediska údržby je klíčové zejména zazimování potrubí, kdy je nutné tlakovým vzduchem dostat ven z potrubí zbylou vodu.

Závlahy Závlaha je koncovým prvkem závlahového systému. Na jejím technickém provedení je závislá celková účinnost soustavy. Volba závisí na typu zavlažovaných rostlin, velikosti zavlažované plochy a objemu potřebné vody. Základním dělením závlahy je dělení na povrchovou a podpovrchovou závlahu. Ty se liší v zásadních vlastnostech - zejména jde o efektivitu vsaku k rostlině, rychlost odpařování a erozní vliv. Dle způsobu provedení lze závlahu dělit do několika kategorií.

Závlaha výtopem je povrchovým řešením, které je často využíváno na rozsáhlých plochách. Půda se zaplaví několikacentimetrovou vrstvou vody, která je posléze vsáknuta do půdy. Závlaha může být dle pěstovaných rostlin přerušovaná, nebo kontinuální.

Závlaha přeronom je povrchovým řešením, které je využíváno na rozsáhlých zemědělských plochách. Dnes je její využívání na ústupu. Dochází k postupnému zalití mělkou vrstvou vody a k jejímu částečnému vsaku. Voda je poté odvedena kanálem.

¹tj. závlaze spojené s hnojením

Závlaha brázdivým podmokem je přechodný způsob závlahy využívající sítě zavlažovacích brázd. Do brázd je vpouštěna dle harmonogramu voda a dochází po celé délce ke vsaku ke kořenům rostlin.

Závlaha postřikem je progresivním typem povrchové závlahy, která je využívána na různých typech ploch. Na trhu existuje celá řada postřikovačů - výsuvné, rotační, bodové, jednoduché trysky nebo postřikovače úderové. Jejich významné využití je zejména v zavlažování velkých ploch.

Kapková závlaha je typ povrchové závlahy, kde je využíváno postupné vsakování malého objemu vody po dlouhou dobu. Do této kategorie je možné zařadit i tzv. mikrozávlahu, která je určena pro menší zejména vnitřní prostory. Závlaha je provedena formou trubice, která je buď perforovaná, nebo jsou na ni připojeny tzv. kapkovače. Kapkovače se dělí dle možnosti regulace průtoku, kdy je možné koupit jednoduché, které jsou regulovány pouze tlakem v systému nebo sofistikovanější, které umí regulovat množství vody pomocí mechanického nastavování. Vzhledem k malým průměrům jednotlivých rozváděcích trubic a kapilár je citlivá na kvalitní filtraci vody.

Mezi podpovrchové způsoby závlahy patří různé drenáže a kapkový potrubí. Často je realizováno individuálně za účelem potřeby dostat vodu blíže ke kořenům rostlin, např. zavlažovací trubice typu RWS.

Regulační členy Regulační členy ve stacionárních soustavách slouží k řízení množství vody dopravované závlahovým systémem. Jedná se zejména o elektromagnetické ventily a regulátory tlaku.

Elektromagnetické ventily se liší v materiálovém provedení (litina, mosaz, plast - PVC), v možném průtoku (až 90 m³/s u litinových) a též v maximálním přípustném tlaku, ten se pohybuje v rozmezí 1-16 barů. [4]

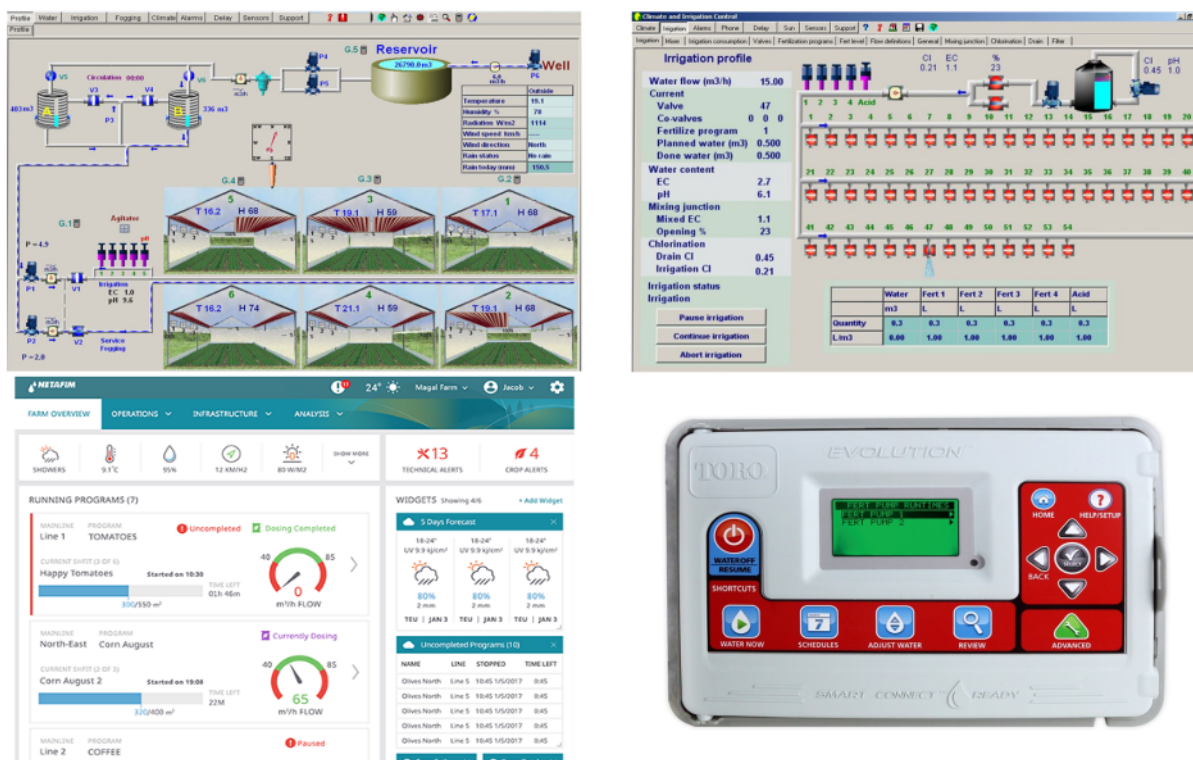
Regulátory tlaku (snižovače tlaku) jsou určeny k omezení maximálního tlaku na dané sekci potrubí. To je provedeno pomocí úbytku tlaku na mechanismu s pružinou a membránou. Přesnost této mechanické regulace často mírně kolísá.

Sensorika Sensorika v závlahových systémech slouží ke zjištění aktuálního stavu zavlažovaných plodin. Často jsou spojovány do komplexních bezdrátových sítí. Dnes jsou se závlahovou zmiňovány sensory - teploty, tlaku, průtokoměry, sensory zaplavení, pH-měry, tenziometry, měřiče vodivosti půdy.

Řídicí technika Řídicí technika v oblasti stacionárních závlahových systémů je využívána k spouštění jednotlivých sekcí v zadaných časových intervalech. Data ze sensoriky je možné využít k jemnější regulaci těchto procesů.

Řídicí jednotky lze dělit na jednoduché časovače a složitější kontroléry. Jejich napájení je často realizováno pomocí baterií, nebo solárního článku s akumulátorem. U složitějších systémů je dostupná i varianta pro napájení ze sítě.

Uživatelské rozhraní je provedeno nejčastěji pomocí panelu a několika tlačítek na řídicí jednotce. Ta umožňuje nastavení spínacích cyklů pro jednotlivé sekce. V závislosti na možnosti zapojení sensoriky umí konfigurovat i přímé množství vody, reakci na data z meteostanice atd.. Někteří výrobci dodávají k zařízením mobilní aplikace, které umožňují provést konfiguraci jednotky pomocí bezdrátového protokolu (často Bluetooth). Pro rozsáhlé zemědělské aplikace často výrobci nabízejí komplexní software, který umožňuje kromě konfigurace systému i některé agronomické výpočty a analýzy spojené se zavlažováním. Tyto analýzy jsou však už prováděny s komplexními sensorickými sítěmi, které



Obr. 2.2: Příklady provedení uživatelského rozhraní závlahových systémů - software Spirit Pro Irrigation [22, převzato] a Spirit Pro Hybrid firmy Jain Irrigation [21, převzato], software NetBeat [19, převzato] od firmy Netafim a jednoduchý kontrolér typu Evolution AG Series [20, převzato] od firmy Toro

zahrnují čidla v půdě, meteostanice, či dokonce letecké nebo družicové snímkování.

2.1.2 Mobilní

Mobilní zavlažovací systémy mohou být chápány jen jako část stacionárních systémů. Umí však zejména u rozsáhlejších ploch provádět rovnoměrnou závlahu rostlin, bez absurdního růstu počtu zavlažovacích prvků. Jejich principiální schéma lze vidět v diagramu 2.1. Jejich využívání se vzhledem k jednoduchému, ale účelnému provedení redukuje na rozsáhlé, dobře přístupné plochy.



Obr. 2.3: Valivé provedení závlahového stroje [23, převzato] a otáčivé provedení s obslužnou technikou firmy ABI-Irrigation [24, převzato]

V dnešní době jsou tyto závlahy představovány prakticky jen tzv. zavlažovacími stroji, které v závislosti na použité konfiguraci provádí jednorázový, nebo opakovaný pohyb na určené ploše. Tento pohyb je prováděn často jen hydromechanismy, které pohání tlaková voda. Ta je přivedena dlouhým potrubím, které se postupně odmotává z pomocného vozíku, nebo je stroj postupně přepojován k nejbližšímu hydrantu. Závlaha je prováděna výhradně roztříkovači.

2.1.3 Robotické

Pro základní rozvalu je nutné prozkoumat stav řešení na poli robotického zavlažování rostlin. Robotický zavlažovací systém musí tvořit zákonitě opět součást stacionární soustavy, ale na rozdíl od mobilního systému je schopen nahradit více jeho částí a tím podstatně redukovat jeho rozlohu a složitost. Jeho obecné schéma lze vidět na diagramu 2.1.

Aquarius (USA) - robot Aquarius, který vznikl jako projekt v americké firmě Dorhout R&D v roce 2013, je primárně určen pro skleníky.

Je tvořen mobilní platformou s diferenciálním podvozkem, na které je umístěna nádoba na vodu a zalévací zařízení. Zalévací zařízení je provedeno jako dvojitý vývod po stranách robotu, který je umístěn ve výšce určené využitými květináči. Pod vývody jsou umístěna taktilní čidla, která umožňují jednoduchou detekci květináčů.



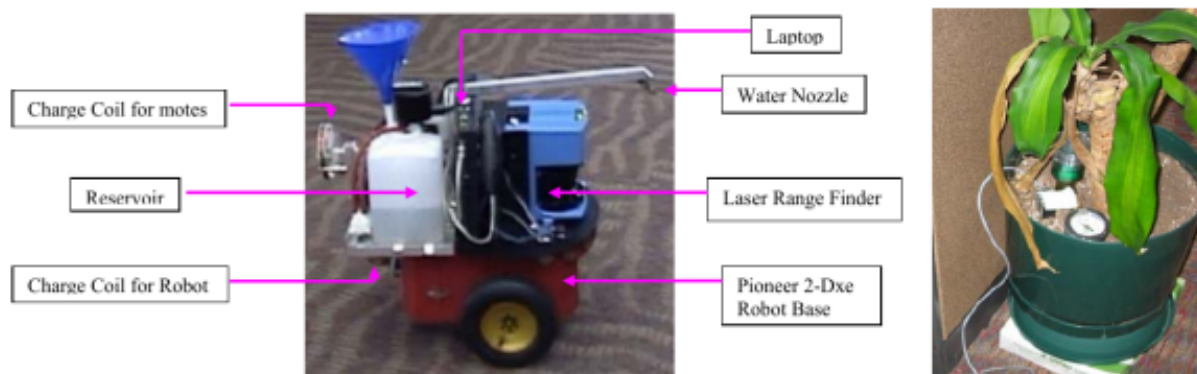
Obr. 2.4: Celkový pohled na robot Aquarius, detail zalévacího zařízení s taktilními čidly, rozmístění rostlin a naváděcí čáry [9, převzato]

Robotický systém se pohybuje v prostoru pomocí naváděcí čáry, která je sledována platformou. Květináče musejí být položeny v optimální vzdálenosti od čáry. Je možné do nich vložit sensor vlhkosti.

Tvůrci uvádějí, že platforma je plně automatizovaná, dokáže se samostatně nabít, doplnit vodu do nádrže a pomocí čáry se umí samostatně pohybovat v různých typech prostoru. [8, 9]

Robotic PlantCare system (USA) - tento komplexní systém pro vnitřní prostory vznikl ve společnosti Intel v roce 2002 za účelem demonstrování použitelnosti ideje chytré sítě. Je založen na open source robotickém frameworku BeeSoft.

Je složen ze zalévacího mobilního robotu na diferenciálním podvozku Pioneer 2-Dx, sensorické sítě a dokovací stanice, kde probíhá indukční nabíjení robotu a doplňování vody. Rostliny jsou zasazeny v květináčích o definovaném průměru a výšce.



Obr. 2.5: Robot systému PlantCare, detail rostliny se sensorikou [7, převzato]

Robot je pro zavlažování vybaven nádrží na vodu a pevně umístěným vývodem. Pro účely pohybu, lokalizace v prostoru a detekci květináčů je vybaven lidarem. Detekce květináčů funguje na základě Houghovy transformace, kdy je z mračna bodů z lidarů pomocí algoritmu identifikován hledaný tvar - kruh v prostoru.

Na každé rostlině jsou umístěny senzory teploty, vlhkosti a osvětlení. Všechny prvky spolu komunikují přes již zmíněnou síť. [7]

Robot pro údržbu trávníku (Srí Lanka) - tento zavlažovací robot určený pro údržbu travnatých ploch byl vyvinut v roce 2017 na University of Moratuwa na Srí Lance. Je založen na ROSu.



Obr. 2.6: Detail provedení zalévací části robotu, mech. návrh sondy vlhkosti a mech. návrh zalévací části [6, převzato]

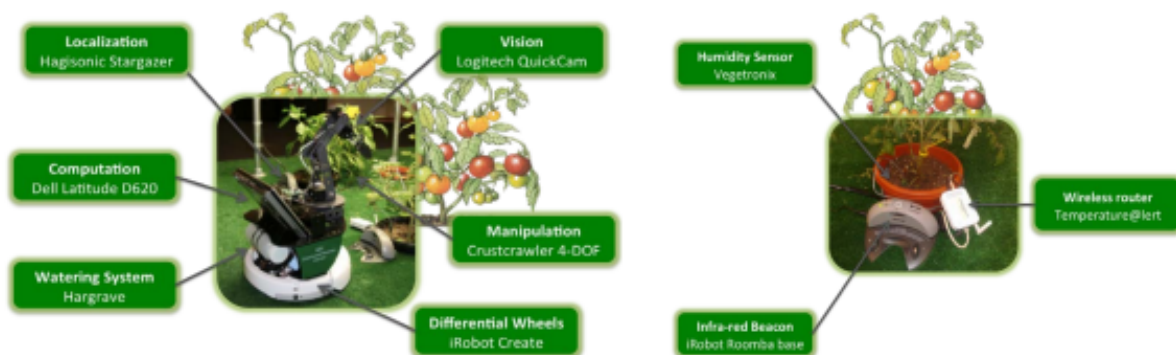
Robot je vybaven čtyřkolový podvozkem, jehož otáčení je prováděno smykem. Zajímavým řešením je závlahová část sestávající z rozstřikovače, elektromagnetického regulačního

ventilu a přípojky na vedenou hadici. Autoři zmiňují problémy s navíjením robota na hadici během jízdy. Mobilní platforma obsahuje též zabodávací mechanismus se senzorem vlhkosti typu FDR. Robot je tedy schopen při jízdě sledovat stav vlhkosti půdy pod ním.

Lokalizace je založena na fúzi dat z GPS, odometrie, akcelerometrů a gyroskopu. V práci je kladen důraz na odzkoušení vlastního algoritmu na plné pokrytí zavlažované oblasti. [6]

Robotický zahradník (USA) - tento systém vznikl v kolektivu pracovníků Univerzity of Colorado a Massachusetts Institute of Technology v roce 2019 za účelem zkoumání možnosti stavby automatické robotického zahradníka pohybujícího se ve vnitřním prostředí. Založen je na systému ROS.

Systém sestává z mobilního robota, na kterém je umístěn manipulátor a zavlažovací systém a rostlin, které jsou vybaveny dokovací stanicí u květináče a sensorikou propojenou do sítě.



Obr. 2.7: Robot a jeho části, rostlina s dokovací stanicí [5, převzato]

Robot je postaven na platformě iRobot s diferenciálním podvozkem. Robot dále obsahuje kameru, určenou pro detekci rostlin, plodů atd.. Pro potřebu lokalizace je využíván Hagisonic StarGazer. To je čidlo, které je schopné, na základě odrazu IR záření od speciálního štítků, tento štítek lokalizovat. Robot je k dokovací stanici - rostlině, naveden pomocí IR signálu. [5]

Jiné systémy Pro úplnost lze uvést příklady dalších prací, které ukazují různé více, či méně komplexní pokusy o stavbu robotických závlahových systémů. Některé z nich obsahují zajímavé konstrukční detaily a nápady. [10, 11, 12, 13, 14, 15, 16, 17, 18]

2.2 Porovnání vlastností závlahových systémů

V předchozí kapitole byly nastíněny dosavadní řešení klíčových uzlů závlahových systémů. Na základě toho je možné porovnat některé vlastnosti stacionárních, resp. mobilních systému a systémů robotických.

Zdroj vody - zdroj vody může mít v obou případech stejnou podobu. Vzhledem k rozsáhlosti využití stacionárních závlah se však může velmi lišit potřebnou *vydatností*.

Úprava vody - úprava vody je prováděna ve stacionárních systémech vždy před jejím čerpáním. V robotickém systému v závislosti na potřebě, může být provedena už v rámci zdroje vody, nebo takové zařízení může být součástí robotu.

Čerpání vody - čerpání vody se v obou variantách opět liší pouze v předpokládaném potřebném průtoku a tlaku. Potřeba výkonu u čerpání též úzce souvisí s provedením rozvodu vody, kdy tlakové ztráty rostou úměrně délce rozvodu, jeho typu, či průtoku.

Regulace - v regulaci se projevuje výrazný rozdíl v potřebě regulačních členů pro provedení co nejindividuálnější závlahy ve stacionárních systémech. Počet takových prvků v nich roste úměrně velikosti systému. Pro každý prvek samozřejmě musí být provedeno spojení s řídicím systémem a napájení.

V robotických systémech naopak při růstu rozsahu závlahy zůstává počet regulačních členů konstantní a je tím dokonce zaručena i opravdu individuální možnost konfigurovatelné závlahy. Vzhledem k umístění prvků na robotické platformě též mizí problémy se spojením s řídicí jednotkou a provedením napájení.

Rozvod / Doprava vody - rozvod vody ve stacionárních systémech může dosahovat prakticky libovolných rozměrů. Průtoky nejsou omezeny. Konfigurace průtoku je prováděna v rámci jednotlivých sekcí. V nich může být provedena i sekční fergitace. S růstem délky rozvodu rostou i ztráty na něm. Rozvody též pro svou práci potřebují minimální množství vody, které zajistí jejich funkčnost. Pokud jsou rozvody otevřené, může docházet k masivnímu odpařování vody.

Doprava vody je u robotických systémů principiálně omezena množstvím energie a maximálním možným dopravovaným objemem vody. Nepotřebuje však žádné stavební, či technické úpravy prostředí. Fergitace může být provedena individuálně pro každou rostlinu. Minimální množství vody pro zajištění funkčnosti je zanedbatelné.

Řídicí systém - provedení řídicího systému se ve stacionárních systémech řídí principiálně množstvím spouštěných sekcí a množstvím, resp. komplexností použité sensoriky. Vzhledem k provedení regulace a rozvodu vody je nutné u větších systémů provádět stavbu řídicí sítě, která spojuje mnoho subjednotek přímo u sekcí.

U robotických závlah je možné sloučit řídicí systém robotu s řídicím systémem závlahy. V základu je tedy možné volně rozšiřovat množství použité sensoriky, či reg. prvků, jejichž počet je oproti stacionárnímu systému zanedbatelný. Při menším počtu prvků je tedy možné provedení sofistikovanějšího řízení závlahy. Podstatnou nevýhodou je však nutné uzpůsobení robotu pro autonomní práci v pracovním prostředí. To je často spojeno s vyšší složitostí systému v různých oblastech - plánování práce, komunikace s uživatelem atd..

Sensorika - množství použité sensoriky ve stacionárních systémech úzce souvisí s jeho určením. Pokud je sensorika využita pro individuální kontrolu stavu u každé pěstované rostliny, pak je náročnost provedení takové sensorické sítě enormní. Není nutné dodávat i fakt, že je potřeba zajistit komunikaci a jejich napájení. Robotické systémy takovým problémem v oblasti sensoriky trpět nemusí díky individuálnímu přístupu ke každé rostlině, tj. robot může být vybaven pouze jednou takovou sensorickou sadou.

Druhou možností je využití sensoriky pro kontrolu stavu větších oblastí, či sekcí. Tato možnost je využitelná v obou variantách, jak ve statickém a tak v mobilních provedení na robotu.

Závlaha - může být řešena u stacionárních systémů libovolným způsobem, který je omezen pouze vydatností zdroje vody. Robotický systém může využít vzhledem k omezenému množství vezené vody pouze nějakou formu postřiku, či kapkové závlahy, která však může být velmi dobře cíleně aplikována.

2.3 Optimální využití robotických zavlažovacích systémů

Na základě průzkumu dnešních možností závlahové techniky a analýzy jejich vlastností lze pozorovat, že existují určité situace, kde roboty mohou najít v oblasti zavlažování své uplatnění. To je principiálně omezeno pouze zásobou vezené vody a energie.

Hlavní devizou robotického závlahového systému je vyšší míra možné sofistikovanosti, jednoduché rozšiřitelnosti, vysoká míra adaptibility a hlavně principiální individuální přístup ke každé zálivce.

Na základě těchto vlastností byly identifikovány 2 hlavní potencionálně optimální oblasti nasazení robotických zavlažovacích systémů.

Závlahové roboty určené pro vnitřní prostory, které nejsou určeny k pěstování rostlin - pod takovými prostory lze vnímat domácnosti, kanceláře, chodby atd.. Tyto prostory primárně nejsou adaptovány pro pěstování rostlin. Tím je myšlen rozvod závlahové vody a umístění regulačních, resp. řídicích prvků. Principiálně to není nemožné, avšak v dnešní době budovy s takovým systémem nejsou rozšířené.

Rostliny v těchto místech často nejsou příliš závlahově náročné a jsou rozptýleny v různých místnostech a umístěny v různých polohách. Při vhodně zvoleném systému rozpoznávání rostlin je možné pro každou rostlinu nastavit individuální péči. Robot nemusí provádět pouze zálivku, ale též může být uzpůsoben pro inspekci rostlin, hnojení, odebrání zvadlých listů atd..

Závlahové roboty určené pro rozsáhlé venkovní prostory, ve kterých se nachází individuálně vysázená zeleň - pod těmito prostory se rozumí v podstatě jakýkoliv veřejný prostor, který není primárně určen pro umístění zeleně - ulice, okolí chodníků atd.. Tato místa jsou často charakterizována dobrou dopravní přístupností, ale v podstatě není ve finančních a možná i ve vodohospodářských silách k nim dovést závlahový systém. Ten má kritický význam zejména u čerstvě vysázené zeleně, kdy je nutná pravidelná zálivka.

2.3.1 Ilustrační příklad

Předpoklad - řekněme, že existuje robotický závlahový systém, který je určen pro vnitřní prostory. Robot je vybaven řídicím systémem a sensorikou do té míry, že dokáže autonomně pracovat ve vnitřních prostorách budov. Robot je schopen měřit objem dodané vody, je vybaven dále teploměrem, senzorem tlaku, osvětlení a měřičem půdní vlhkosti. Robot dokáže rozpoznávat jednotlivé rostliny a tak upravovat individuální zálivku.

Zadání úlohy - mějme administrativní budovu, u které je kladen požadavek, aby byla dodatečně vybavena stacionárním zavlažovacím systémem, který má stejné vlastnosti jako ten robotický.

Obecný návrh - budova má p pater, na každém z nich je s místností ve kterých se mohou nacházet rostliny. Předpokládejme, že vnitřní schéma budovy odpovídá deskovému objektu. Budova je napojena na vodovodní řad, který má na výstupu tlak 3 bary. Místnosti mají šířku a a délku b , šířku chodby nazvěme a_c .

Řekněme, že celý systém navrhne jako stromovitou strukturu, kdy kmen bude tvořit centrální rozvod vody a jednotlivé větve pak rozvody do jednotlivých místností. Stanovme, že jednotlivé místnosti budou tvořit samostatné sekce, které budou napojeny na hlavní rozvod pomocí elektromagnetického ventilu. Každé patro bude mít s sekcí. V závislosti na potřebě zavlažování zatím stanovme, že vybereme takovou řídicí jednotku, která bude zvládat dostatečný počet sekcí. Za elektromagnetickým ventilem bude napojen filtr s regulátorem tlaku. Zalévání jednotlivých rostlin v květináčích bude provedeno pomocí mikrozávlahy, která bude rozvedena hadičkami z rozbočovače za filtrem pro každý kapkovač rostliny zvlášť.

Po stanovení celkové koncepce řešení lze přistoupit k návrhu a výběru jednotlivých komponent. Charakter závlahy bude nepravidelný a průtok nebude velký. Kapkovač byl zvolen s průtokem q a cenou C_k . Vzhledem k tomu, že systém konkuruje robotu, můžeme říci, že vždy bude v jednom čase v provozu pouze jedna sekce závlahy. V místnosti bude průměrně r rostlin. Tomu odpovídá okamžitý minimální průtok:

$$Q = r \cdot q \quad (2.1)$$

Při zvoleném průměru rozvodného potrubí D o ceně/m C_D , resp. závlahového d o ceně/m C_d , vychází střední rychlost proudění:

$$v = \frac{Q}{\pi \cdot \frac{D^2}{4}} \quad (2.2)$$

Pro proveditelnost takové řešení je třeba sledovat tlakové ztráty v systému. Při znalosti průtoků v potrubí lze vyčíst z nomogramů pro polypropylenové potrubí ztráty na 1 m $\Delta p/l$ a tvarové součinitele ζ pro přechodky C_p , T-kusy a kolena o ceně C_t . Tlaková ztráta se pak na prvku dá vypočítat dle [25]:

$$\Delta p = \zeta \frac{\rho}{2} v^2 \quad (2.3)$$

Tlakové ztráty jednotlivých ventilů o ceně C_v , filtrů o ceně C_f atd. uvádí výrobci v katalogích pro předpokládané rozsahy průtoků. Celková tlaková ztráta nesmí překročit nejvyšší dovolenou mez, resp. tlak rozvodem mikrozávlahy nesmí poklesnout pod limit, jinak dojde k nepravidelnosti závlahy, nebo k jejímu selhání. V tomto případě by neměl klesnout pod 2, 4 baru.

Celkovou tlakovou ztrátu pro nejvzdálenější sekci od zdroje vody na patře, lze vypočítat dle vztahu:

$$\Delta p_c = (s - 1) \cdot \Delta p_T + \Delta p_{kol.} + \left(\left(\frac{s}{2} \cdot a \right) + a_c \right) \cdot \Delta p_{potr.}/m + \Delta p_{vent.} + \Delta p_{filtr} \quad (2.4)$$

Zatím byl popsán rozvod vody v budově. Nyní je potřeba doplnit to, že pro splnění podmínek je nutné každou rostlinu doplnit o sensor vlhkosti půdy o ceně C_{sv} a každou sekci (místnost) o sensor tlaku o ceně C_{sp} , teploty o ceně C_{st} a osvětlení o ceně C_{so} . Pro každou sekci je zvolena řídicí jednotka o ceně C_c . Předpokládejme, že řídicí jednotka je schopná všechna tato data zpracovat a případně využít pro regulaci.

Celkovou cenu takového systému (bez započítání lid. práce, staveb. úprav atd., vodičů) lze stanovit pomocí vztahu:

$$C = p \cdot s \cdot \left(\left(a + \frac{a_c}{2} \right) \cdot C_D + C_t + C_v + C_f + C_{so} + C_{st} + C_{sp} + C_c + r \cdot (b \cdot C_d + C_k + C_{sv}) \right) \quad (2.5)$$

Konkrétní návrh - pro příklad výpočtu uvádím návrh pro administrativní budovu s parametry uvedenými v tabulce.

Parametr	Hodnota
p	6
s	30
r	5
a [m]	3
a_c [m]	3
b [m]	5

Tab. 2.2: Parametry budovy

Při použití kapkovačů 8 l/h, vychází malý okamžitý průtok v soustavě:

$$Q_s = r \cdot q = 6 \cdot \frac{8}{60 \cdot 60} = 0,0132 \text{ l/s} \quad (2.6)$$

Očekávaný průtok předpokládejme zhruba dvakrát vyšší. Pro ten lze vyčíst z tabulek jednotlivé tlakové ztráty a odporové součinitele². Další tabulka ukazuje příklady konkrétních využitých komponent³ pro jedno patro budovy.

Položka	Typ	Počet / délka	Cena [Kč] kus / m	ζ	Δp za Q_s	Poznámka
Rozváděcí potrubí	PPR D32	135 m	31	-	0,003 kPa/m	
Závlahové potrubí	DRIPLINE XF 16 mm	900 m	7,4	-	-	ztráty nepodstatné
T-kusy	T-kus PP-R D25	29	8,2	1,1	0,0055 kPa	
Koleno	Koleno PP-R D25 FF 90°	1	9,4	1,5	0,0075 kPa	
Elektromag. ventil	Rain Bird 100-HV-MM	30	529	-	27 kPa	
Reg. tlaku s filtrem	Rain Bird PRF-075-RBY	30	705	-	25 kPa	
Kapkovač	PCJ DRP 8 l/h, NIP	180	6	-	-	
Redukce	D32 x 1"	30	100	0,55	0,0028 kPa	
Redukce	1" x 3/4"	30	30	0,55	0,0087 kPa	
Redukce	16mm x 3/4"	30	12	0,55	0,0175 kPa	
Sensor půdní vlhkosti	Půdní vlhkoměr + elektronika	180	100	-	-	cena odhad
Měřicí stanice	Sensorika + elektronika	30	100	-	-	cena odhad
Řídicí jednotka	Hunter HCC-800-PL 8 až 38 sekci	1	16 656	-	-	pro celé patro

Tab. 2.3: Tabulka zvolených komponent systému pro patro a jejich množství a charakteristiky

Maximální tlakové ztráty na nejdále sekci za regulátorem tlaku lze vyčíslit na:

$$\Delta p_c = (30 - 1) \cdot 0,0055 + 0,0075 + ((30/2 \cdot 3) + 1,5) \cdot 0,003 + 25 + 27 \cong 52,3 \text{ kPa} \quad (2.7)$$

Což je stále nad dolní limitem maximálního tlakového úbytku před regulátorem tlaku. Na této ztrátě mají podíl z 99% ventil a filtr. Jde vidět, že při malých průtocích mikrozávlahy, je tlaková ztráta na potrubí a prvcích přijatelná, systém je tedy realizovatelný za těchto průtoků i pro rozsáhlejší komplexy. Celkovou cenu komponent lze odhadnout na:

²dle [29, 26]

³dostupné komponenty dle [28, 27]

$$C = 6 \cdot (16656 + 30 \cdot ((3+1, 5) \cdot 31 + 8, 2 + 529 + 805 + 142 + 6 \cdot (5 \cdot 7, 4 + 6 + 100))) = 546642 \text{ korun} \quad (2.8)$$

Interpretace výsledků - příklad ukazuje, že cena za stacionární systém v závislosti na charakteristikách budovy může dle definované specifikace dosahovat celkem absurdních částek. Zde je nutné podotknout, že byly uvážovány pouze maloobchodní ceny bez započtení množstevních slev. Odhadovaná cena sensoriky je pouze orientační, maloobchodní ceny pro komerční závlahy při počtu sensorů, bohužel, naprosto převyšují cenu zbylého systému. V kalkulaci nejsou uvažovány částky za lidskou práci a prováděné stavební úpravy.

Zajímavým faktem je i to, že i přes množství použitých prvků tento systém stále není vybaven stejnými vlastnostmi, jako ten robotický - zmínit jde třeba jen individuální přístup ke každé rostlině v místnosti, versus sekční přístup stacionární soustavy. Diskutabilní je i přímé napojení závlahového systému na vodovodní řad - z důvodu teploty, obsahu chloru atd.. Detailnější rozbor provedení by si zasloužil i zjednodušený pohled na rozvody vody po místnosti i budově. Závěrem lze pro ilustraci doplnit celkové množství vody v systému - zhruba 1750 litrů.

Robotický systém má tedy i přes mnohonásobně méně užitých prvků stále v některých ohledech lepší vlastnosti. Kromě závlahy má však přidanou hodnotu další možnosti rozšiřování úkolů - vnitřní doprava, inspekce atd..

Kap. 3

Definice úlohy

Před samotným započítím jakékoliv další práce na projektu, je nejprve nutné vyjasnit, čemu se vlastně bude věnovat. Cílem této kapitoly je tedy definování záměru a cíle práce, který bude posléze konkretizován v podobě bližší specifikace.

Tato fáze projektu úzce souvisí s jeho plánováním. Na základě zvolených cílů a požadavků, je možné prvně odhadnout rozsah projektu a jeho náročnost. Je nutné zde poznamenat, vzhledem k charakteru projektu jako závěrečné práce, že specifikace systému byly provedeny tak, aby je bylo možné s omezeným počtem lidí, prostředky a ve vymezeném čase splnit.

3.1 Stanovení cíle práce

Po provedení rešerše problematiky zalévání lze konstatovat, že existují oblasti, kde zalévací robot může nalézt uplatnění. Hlavní cíl práce tedy může sledovat prvotní ideu - bude provedena stavba autonomního mobilního zalévacího robotu pro optimální oblast jeho použití.

Snahou by tedy mělo být provedení robotického systému, který bude v maximální možné míře **adaptabilní** - bude schopen reagovat na různé změny a podmínky ve svém okolí, **rozšiřitelný** - systém bude zpracován tak, aby bylo možné mu jednoduše přidávat nové funkce, moduly, ale i práci a **inteligentní** - systém bude sám přijímat důležitá rozhodnutí, která se budou týkat jeho práce.

Vzhledem k demonstrativnímu charakteru této práce byly jako vedlejší cíle stanoveny - zejména snaha o maximalizování výhod takového řešení a minimalizování jeho nevýhod. Neoddělitelnou součástí hlavního a vedlejších cílů je do určité míry zkoumání vlastností, či obecných rysů takového robotického systému.

Dobrým návrhem by měla být redukována zejména **uživatelská náročnost ovládání** - která u robotického řešení může představovat vážnou překážku v jeho nasazení.

3.2 Vymezení úlohy a konkretizace zadání

S přihlednutím k obecným cílům práce, lze vyspecifikovat konkrétnější zadání celého projektu.

Prostředí - jaké bude optimální pracovní prostředí systému?

Robot se bude pohybovat ve vnitřních prostorech budov - např. kancelářské budovy, domácnosti atd..

Koncepce řešení - jak bude celý systém řešen?

- provedení stavby robotického systému a robotu na dostupné mobilní platformě s využitím dostupných hardwarových nebo softwarových komponent.

Úkoly systému - jaké budou hlavní úkoly robotu?

Autonomní pohyb ve vnitřních prostorách budovy - robot se musí spolehlivě pohybovat prostorem bez kolizí s překážkami. Musí umět adekvátně reagovat na změny v tomto prostředí a nečekané události.

Rozpoznávání rostlin - robot musí umět rozlišit jednotlivé rostliny v prostoru za účelem správného provedení zalití. V tomto požadavku se skrývá nejen fyzické rozpoznání rostliny v prostoru, nýbrž i uchovávání znalosti o rostlině v systému a identifikace této konkrétní rostliny v prostoru.

Zalévání rostlin - robot musí provést spolehlivé autonomní zalití rostliny, která se nachází v prostoru v různých výškách a pozicích, optimálním množstvím vody. Tento úkol úzce souvisí se schopností robotu dopravovat vodu a uchovávat znalosti, či z nich vyvozovat závěry o správném zalití.

Komunikace s uživatelem - rob. systém musí účelně a jednoduše komunikovat s jeho uživatelem. Uživatel musí mít do určité míry právo upravovat chod systému a mít přístup k jeho datům, či sledovat jeho stav.

Kap. 4

Ideové řešení systému a dekompozice úlohy

Po definování záměru práce a stanovení jednotlivých dílčích cílů lze přejít k prvotní rozvaze o tom, jak bude vyvíjený robotický systém fungovat. Cílem této kapitoly je definování základního minimálního - tzv. *use case*, což je v podstatě první komplexnější představa o zamýšlených schopnostech a tedy i užitých vlastnostech finálního systému.

V rámci tohoto prvotního popisu musejí být ve vší obecnosti identifikovány jednotlivé hlavní bloky budoucího systému a musí být jasně daná jejich role a vzájemná interakce, která zajistí splnění požadovaných cílů. Po rozboru a definování, co systém je a co by měl dělat, lze posléze přistoupit k otázce, jak by se dané činnosti měly provádět. To plynule přechází v kapitolu definování systémových bloků, které řeší celou tuto problematiku v projektu ze dvou zásadních pohledů - možného technického provedení a pohledu organizace práce.

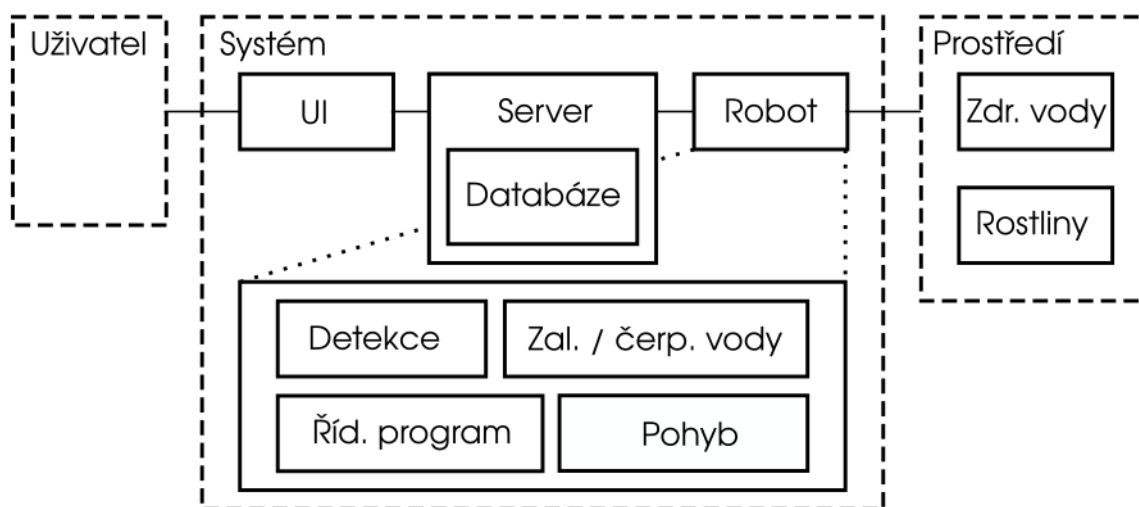
Do této chvíle se v rámci této práce silněji neprojevil její projektový charakter, který je však přítomem od prvotní ideje, která byla stanovena již v první kapitole. Dle [32] lze projektové fáze dělit do zhruba čtyř fází - *Plánování*, *Analýzy*, *Návrhu* a *Implementace*.

Ve fázi *Plánování* dochází k prvotnímu zhodnocení ideového návrhu projektu z různých úhlů pohledu, který se týká nákladů, příp. realizačních možností a předpokládaného přineseného užitku, či praktická využitelnost. Ve fázi *Analýzy* se provádí prvotní obecný návrh *use case*, který slouží k definování obecných principů funkce nejen navrhovaného systému, ale též i procesu jeho návrhu. Ve fázi *Návrhu* dochází ke konkrétnímu rozpracování systémového návrhu, který musí dekomponovat celou soustavu na jednotlivé hardwarové, či softwarové části, které je možné jasně pojmenovat a definovat jejich funkci a spojení. Poslední fází je *Implementace*, ve které probíhá na základě již vypracovaného návrhu provedení jednotlivých částí systému, jejich integrace a testování. Nedílnou součástí všech fází je i průběžné vyhotovování projektové dokumentace, která slouží jak k interní komunikaci mezi členy týmu, tak i pro samotnou prezentaci výsledků práce.

Projekt zalévacího robotu prochází v určité redukované podobě všemi těmito fázemi. Dílčím úkolem této práce je tedy nejen samotný návrh robotického zalévacího systému, ale i jeho úspěšná realizace, jejíž úspěch nemalou měrou závisí i na řízeném provádění celého projektu jednotlivými projektovými kroky.

4.1 Vymezení systému a jeho částí

Vymezení jednotlivých bloků je znázorněno v diagramu 4.1. Diagram ukazuje dělení projektu v několika v rovinách.



Obr. 4.1: Dekompozice systému na obecné a technické bloky

Nad řešeným systémem stojí uživatel - ten má za úkol zejména přípravu prostředí pro jeho správnou funkci¹, přípravu systému pro správné spuštění, správu systému v průběhu jeho fungování a v neposlední řadě provádí kontrolu při jeho práci.

Za řešený systém lze v této práci považovat mobilní robot a všechny obslužné moduly, které slouží k interakci s ním, nebo k jeho správě. Interakce by měla probíhat pomocí uživatelského rozhraní (zkráceně UI), které by mělo komunikovat se serverem. Robot by tak měl být spojen s UI pomocí mezipřímky, který by měl zajišťovat centrální uchovávání znalostí systému a určitým způsobem tuto komunikaci moderovat.

Všechny tyto části interagují s okolním prostředím, ve kterém jsou rozeznávány v zásadě dva typy objektů - zdroje vody a rostliny. Ty se mohou nacházet v libovolných polohách v prostoru a můžou mít různou podobu.

4.2 Návrh funkcionality - use case

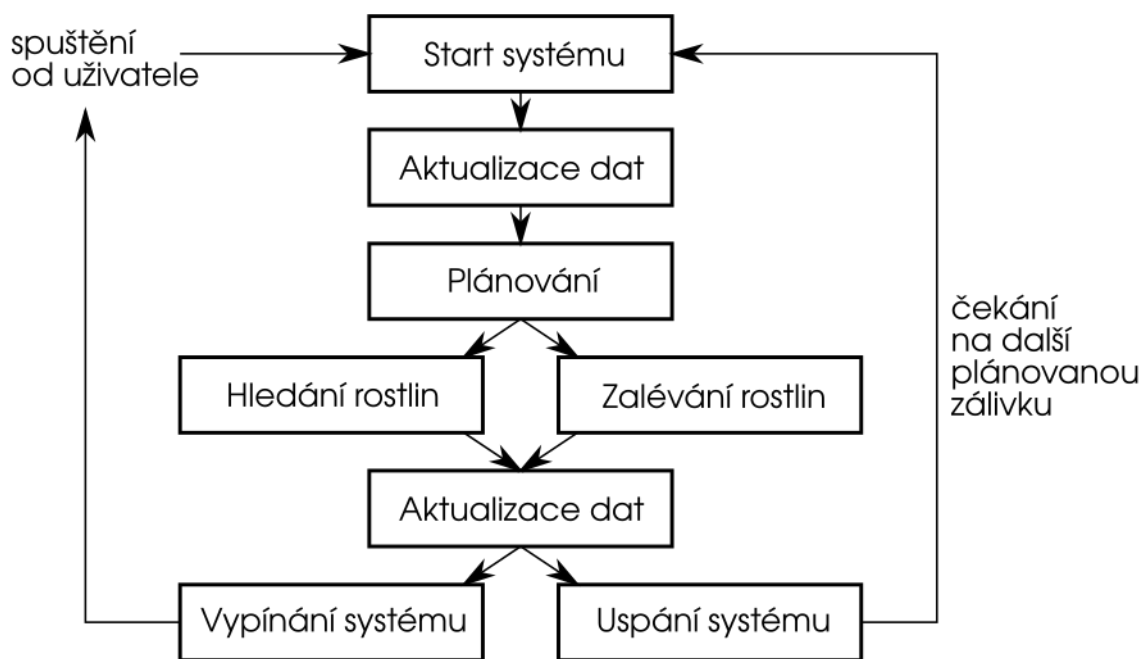
Funkcionalitu systému lze definovat jako sled činností, které jsou vykonávány za účelem splnění cílů práce. Use case v této práci je tedy brán jako chronologický přehled 4.2 všech činností systému, který zajistí jeho funkčnost a určitým způsobem jasně naznačí jeho možnosti, či omezení.

Mějme nadšeného uživatele nového robotického zalévacího systému, ten pro uvedení systému v činnost předpřipraví dle sys. specifikací prostředí - rostliny, zdroje vody atd.². Po úspěšném uvedení systémového serveru v činnost může uživatel pomocí provedeného UI provést první konfiguraci systému. Po této konfiguraci může dojít k zapnutí robotu.

Po zapnutí robotu a úspěšné inicializaci všech systémových částí, které uživatel může sledovat pomocí UI, robot dle zadaných, nebo zjištěných systémových dat může přejít k výběru vhodné funkcionality.

¹Je jasné, že čím méně takových příprav musí být, tím je systém sofistikovanější.

²Ty budou popsány v dalších kapitolách.



Obr. 4.2: Pracovní diagram systému

Systém pracuje se dvěma hlavními funkcionalitami - zalévání rostlin a hledání rostlin³. Jejich názvy do určité míry naznačují jejich účel.

Po provedení funkcionality dochází k aktualizaci dat v systému a přechodu do režimu spánku, či k vypnutí robotu. Poté se sled činností vrací zpět k zapínání systému.

4.2.1 Hledání rostlin

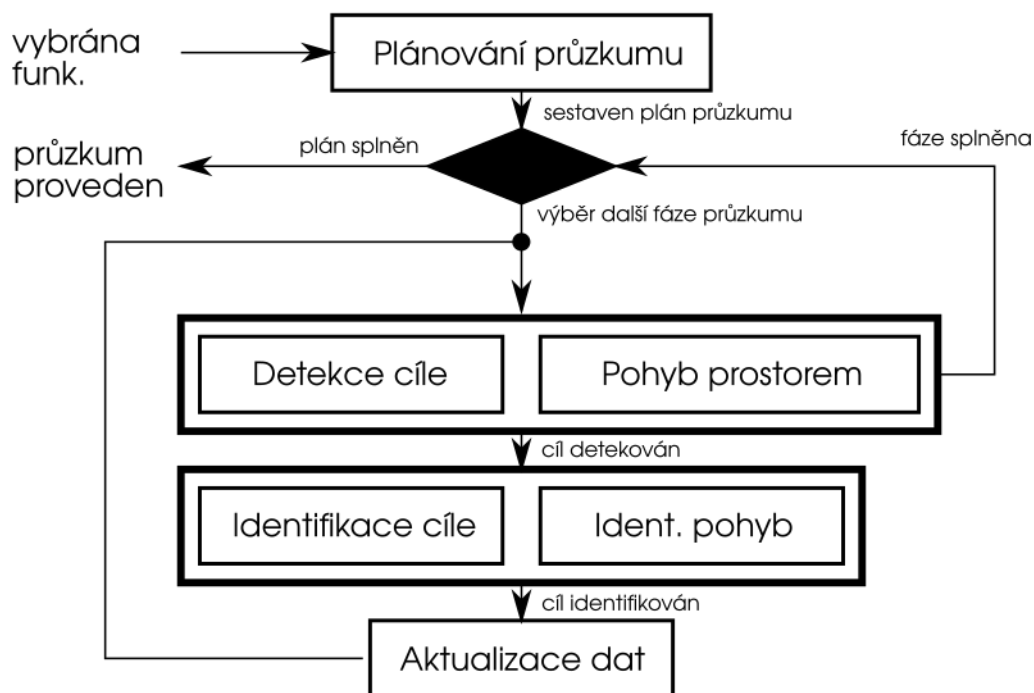
Při prvotním spuštění systému musí být provedeno hledání rostlin. To by svým způsobem mělo zajišťovat automatický sběr informací o prostředí, ve kterém se robot bude pohybovat. Řekněme, že uživatel při prvotní konfiguraci nedá robotu v podstatě žádné informace. Systém tedy nemá k dispozici mapu prostředí, seznam rostlin, jejich umístění, potřebu vody atd..

Robot tedy musí provést inteligentní průzkum prostoru za účelem sběru maximálního počtu dat. Mělo by tedy docházet k simultánnímu mapování prostoru a detekci objektů v prostoru a k jejich klasifikaci, či identifikaci. Celou funkcionalitu lze rozdělit do 2 kroků.

Pohyb prostorem - v této fázi je prováděn robotem průzkum prostředí. V závislosti na množství znalostí systému může být plánovaný, či víceméně náhodný. Probíhá do chvíle, kdy dojde k detekci potenciálního systémového objektu. V systému musí být přítomen rozhodovací proces, který na základě detekce objektu může stanovit, zdali je známý, nebo neznámý.

Identifikace objektu - při detekci požadovaného objektu musí být provedena jeho identifikace, či klasifikace. V tomto kroku musí být zjištěny všechny potřebné údaje o

³Toto dělení bylo provedeno pro potřebu zjednodušení vývoje systému, v ideálním případě by měly tyto funkcionality splynout v jednu.



Obr. 4.3: Pracovní diagram funkcionality - *Hledání rostlin*

detekovaném objektu a jeho charakteristiky. V závislosti na úspěšnosti či neúspěšnosti identifikace jsou pak aktualizovány data v systému.

Po provedení několika takových hledání systém už využívá tuto funkcionalitu jen pro aktualizaci systémových dat - přidávání rostlin, úprava prostředí atd..

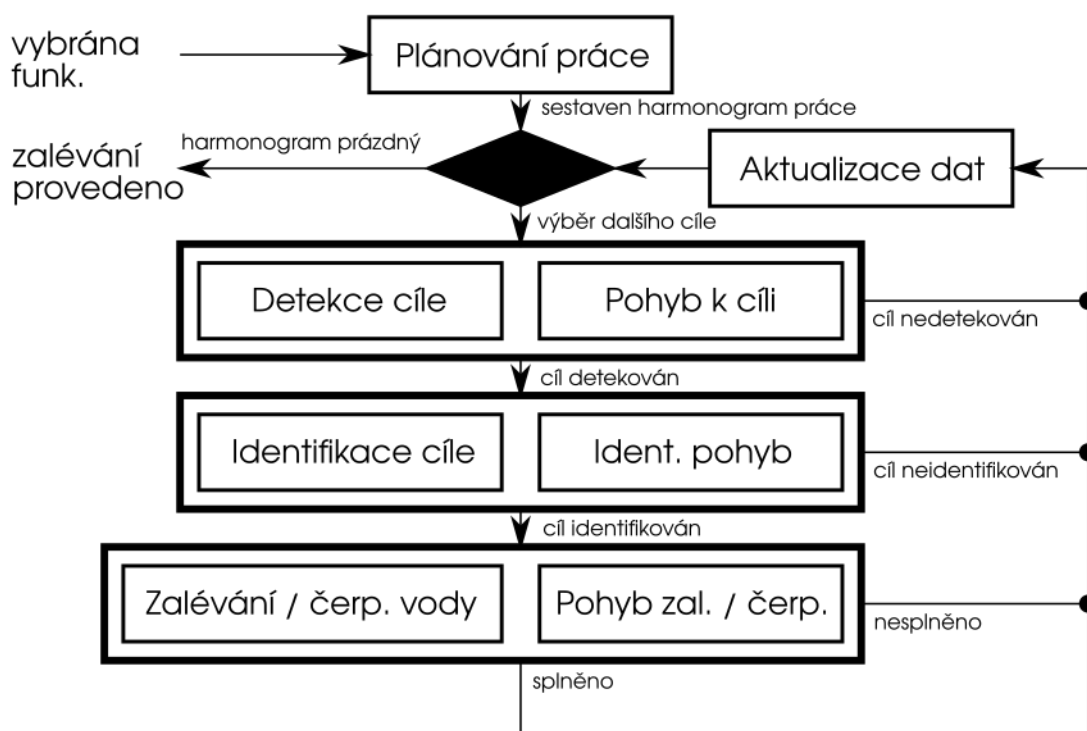
4.2.2 Zalévání rostlin

Po úspěšném sběru dat při hledání rostlin, či pečlivém zadání od uživatele, může systém přejít k jeho hlavní úloze - zalévání rostlin. Při přechodu do této fáze již má všechny potřebné informace o požadované práci - plán prostředí, seznam objektů - rostlin, zdrojů vody, jejich charakteristiky, potřeby a pozice.

Po aktualizaci všech dat dochází k rozplánování práce. Pod tímto pojmem si lze představit tvorbu harmonogramu zalití jednotlivých rostlin proloženou potřebou načerpání vody. Harmonogram je zpracován na základě aktuální potřeb a stavu jednotlivých objektů.

Po zpracování harmonogramu práce je přistoupeno k samotnému zalévání, to je provedeno pomocí smyčky, kdy jsou pro každou položku v harmonogramu naplánovány 3 fáze.

Pohyb k cíli - je fáze, kdy se robot pohybuje k dané položce v harmonogramu. V této fázi je vedle pohybu přítomna i detekce cíle - robot musí být schopen v prostoru daný objekt rozeznat. Tato fáze je ukončena po úspěšném, či neúspěšném dojetí k cíli, nebo úspěšné, či neúspěšné detekci objektu.



Obr. 4.4: Pracovní diagram funkcionality - *Zalévání rostlin*

Identifikace cíle - po detekci musí dojít k identifikaci objektu. Identifikace rostliny před zalitím je klíčovou kontrolní funkcí pro správnou funkci systému. Díky ní je možné jasně ztotožnit daný objekt s uchovávanými informacemi v systému. Po úspěšném provedení identifikace se přechází do dalšího stavu.

Zalévání / čerpání vody - v závislosti na typu objektu v této fázi dochází k čerpání vody ze zdroje, nebo k zalévání rostliny. Po úspěšném, či neúspěšném zalití dochází k aktualizaci dat v systému.

Po provedení práce - úspěšném, či neúspěšném zalití rostlin, či načerpání vody v harmonogramu je základní činnost systému ukončena.

4.3 Identifikace bloků v systému

Na základě výše popsané funkcionality, lze definovat systémové bloky. Ty lze chápat jako skupiny, které obsahují úkoly podobného určení, či provádějí jasně definovanou činnost. Jednotlivé bloky lze vidět v diagramu 4.1.

4.3.1 Technické bloky

Uživatelské rozhraní (UI) - z ideového řešení systému vyplývá, že uživatelské rozhraní je základním vstupní branou jakékoliv interakce s uživatelem. Rešerše ukázala, že provedení rozhraní závlahových systémů je podobné a umožňuje uživateli v základu nastavení závlahy - spouštění jednotlivých sekcí v definovaném čase. Z principu je provedeno

velice jednoduše a přívětivě.

UI v tomto projektu musí umožnit uživateli totéž - základní nastavení zálivky pro každou rostlinu, v zásadě však musí, vzhledem k větší míře potřebě informací v systému o prostředí, umožnit uživateli správu a kontrolu i těchto dat. Přitom se však musí držet stejného cíle - musí být provedeno přehledně a jednoduše.

Uživatel by měl mít v základu přehled o všech objektech v systému a jejich parametrech. Měl by mít možnost jejich kontroly, úpravy, či úplného odstranění. Tyto objekty by měly být strukturovány a případně přiřazovány pro jednotlivé roboty, oblasti.

UI by mělo být schopno zobrazovat stav systému, přehled plánovaných akcí a přehled akcí vykonaných. Součástí toho by měla být i možnost zasahování do automaticky generovaného harmonogramu práce.

Mělo též splňovat požadavek jednoduché rozšiřitelnosti funkcí i spravovaného systému. V neposlední řadě by mělo obsahovat i některé funkce, které by měly umožňovat uživateli i jednoduché řízení systému.

Jako řešení se nabízí softwarová verze - mobilní, či webová aplikace s požadovanými funkcemi. Verze rozhraní ve verzi návrhu vlastního elektronického ovladače s displejem⁴ se jeví jako těžkopádná a příliš konkrétní, ikdyž je v některých případech nutná.⁵

Server s databází - by měl tvořit jakési centrální úložiště dat v systému. Tyto data by měly být přehledně strukturovány a přístup k nim by měl být jednoduchý, jak ze strany robotu, tak ze strany UI. Server musí svým způsobem komunikaci mezi robotem a UI i moderovat. Zvolená architektura se serverovým uzlem je zamýšlena též jako určitý krok k rozšiřování systému - více robotů, více oblastí atd..

Server v závislosti na rozsahu využití takového robotického systému může pracovat na samostatném zařízení, či přímo na robotu. Není tedy nutné žádné specifické vybavení, stačí jen základní síťové prvky.

Řídicí program - pod tímto blokem si lze představit všechny „vysokoúrovňové“ řídicí programy robotu provádějící spouštění, vypínání systému, či provedení jednotlivých algoritmů funkcionality. Takový program tvoří jakousi kostru celého programového vybavení, která posléze volá jednotlivé další procesy v systému.

Detekce - tento blok by měl zajišťovat veškeré činnosti spojené s rozpoznáváním objektů v prostředí a jejich identifikací. Blok má za úkol vybavit robot potřebnou sensorikou.

V systému jsou rozpoznávány 2 typy objektů - rostliny a zdroje vody. Tyto objekty se mohou lišit v mnoha ohledech - tvaru, velikosti, barvy atd.. Provedení jejich detekce je ve vnitřním prostoru možné na základě obecné struktury, která se je naučí rozeznávat - např. z detekce z obrazu umělou neuronovou sítí, nebo na základě nějakého specifického znaku, či zařízení, buď přirozeného, nebo uživatelem na ni umístěného⁶. Vzhledem k různým možným umístěním objektů v prostoru je pak nutná i vhodná orientace takového znaku. Řešerše ukázala několik prací, které takový znak využívaly.

Z pohledu provedení není používání umělého znaku nejvhodnější. Uživateli přibude další úkon, na kterém bude ve velké míře závislá úspěšnost provádění funkcionality. V zásadě může umístění takového znaku znamenat i určitý problém prostorový, či estetický

⁴Mobilního, či statického na robotu.

⁵Např. prosté Stop tlačítko, či signalizace jeho stavu atd., to je řešeno už v určité míře v rámci využití platformy.

⁶Typicky využití barvy, tvaru, či např. vysíláče, ident. čipu.

- takový znak musí být systémem snadno detekovatelný. U obecné struktury je hlavní problém ve zvládnutí spolehlivého natrénování rozpoznávací struktury.

Podobným problémem je identifikace objektu. Na rozdíl od detekce nejde o obecné rozeznání daného typu objektu - rostliny, či zdroje vody, ale o konkrétní určení položky v systému. To je proveditelné opět pomocí speciálního přirozeného znaku objektu, nebo pomocí umělého identifikátoru.

Užití identifikátoru má stejné vlastnosti, jako použití umělého znaku u detekce rostlin. V určitých situacích může jejich činnost splynout. Narozdíl od umělého znaku však identifikátor nemusí být snadno detekovatelný. Systém může být informován, kde takový identifikátor na detekovaném objektu hledat. Ten pak obsahuje všechny důležité informace o objektu, nebo jen odkazy na ně. Bez využití identifikátoru si lze představit strukturu, která by např. odhadovala potřebu vody na základě velikosti, či druhu rostliny. Špatně provedený odhad by mohl mít pro daný objekt někdy katastrofální důsledky - např. přelití.

Pohyb - tento blok zajišťuje veškerou funkcionalitu spojenou s pohybem v prostoru. Jedná se zejména o plánování trasy, navigaci, lokalizaci v prostoru, či mikropolohování robotu u rostliny. Stará se též o sensoriku spojenou s těmito částmi. Nezanedbatelnou částí této funkcionality je i provedení reakce na překážky v prostoru, či implementace některých rozhodovacích procesů, úzce souvisejících pouze s pohybem robotu.

S těmito hlavními úkoly jsou spojeny také veškeré záležitosti, které souvisí s popisem prostoru - výpočty vzdálenosti mezi objekty, či optimalizace při plánování práce robotu.

Zalévání / čerpání vody - je blok, který má na starost provedení finálního zalévání rostliny, či čerpání vody. Jedná se o komplexní problém zajištění dopravy a uchování vody na robotu, dopravy vody k rostlině, provedení čerpání vody atd., které v sobě zahrnuje jak mechanické provedení, tak el. zapojení i softwarovou část. Cílem tohoto bloku je tedy provedení zalévací platformy na robotu.

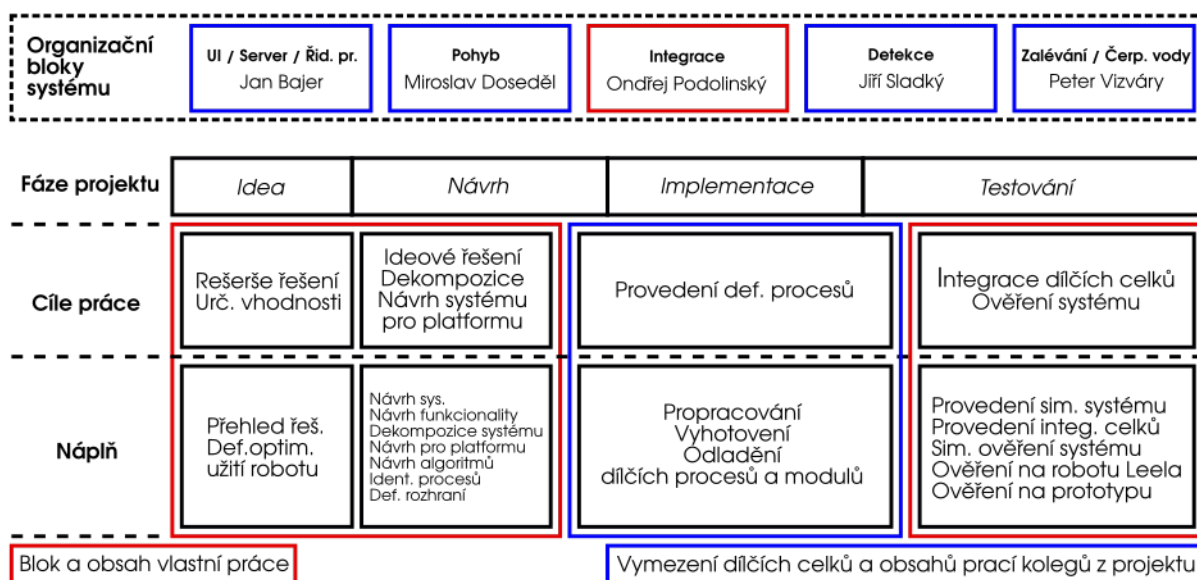
Provedení jednotlivých částí je závislé na přístupu k řešení dané problematiky. Dopravu vody lze zařídit v vnitřních prostorech prakticky jen umístěním nádrže na vodu na robotu. Některé experimentální varianty viděné v rešerši - např. tažení hadice za robotem v tomto případě nejsou moc proveditelné. Součástí řešení musí být i systém čerpání vody s regulačními prvky.

Co se týká zalévacího mechanismu nabízí se základní řešení pomocí zalévacího ramena, které bude schopno přizpůsobovat svou konfiguraci aktuální poloze rostliny. Použitelný je i pevný vývod na robotu, ale ten jistě snižuje požadovanou adaptabilitu systému. Jiné exotické varianty typu provedení stříkání vody k rostlině ani při ideovém návrhu nenabízí vizi spolehlivě použitelného řešení.

Nezbytnou součástí řešení zalévací platformy musí být i její vybavení sensorikou, která zajistí, nejlépe samostatnou, funkci zalévání rostlin a čerpání vody.

4.3.2 Organizační bloky

Systém bylo nutné rozdělit nejen podle technických celků, ale též podle jednotlivých řešitelů. Celý projekt je řešen v pětičlenném kolektivu, kdy má každý jeho člen na starost zde vymezenou část. Vzhledem k charakteru této práce bylo nutné jednotlivé bloky co nejvíce osamostatnit, ale zároveň myslet na to, že jejich spojením má vzniknout funkční celek.



Obr. 4.5: Dekompozice systému na organizační bloky s uvedenými řešiteli a jejich působnost v jednotlivých fázích projektu

Zde je nutné poznamenat, že z pohledu růstu členů týmu roste i potřeba řešení problematiky komunikace v týmu a je kladen důraz na zvolení správné metodiky práce. Metodiku práce na tomto projektu nejlépe vystihuje tzv. *waterfall* model, paralelního typu [32, str. 52]. Ten umožňuje po provedení společného komplexního návrhu samostatnou práci na jednotlivých jeho částech. Tento model práce vykrytalizoval z požadavku na co největší oddělení jednotlivých prací a provedení testování až po integraci dílčích částí.

Na jednotlivé bloky se kladou následující požadavky - rozbor stanovené úlohy a jednotlivých určených procesů, zvolení optimálního řešení, konkrétní návrh jeho provedení, konečná implementace dílčí části s uvažovaným def. rozhraním, debugování a jeho co nejkomplexnější možné otestování před integrací do celku.

UI / Server / Řídicí program - provedení tohoto bloku tvoří jakýsi kompaktní celek obsluhy systému uživatelem a řízení jednotlivých procesů. Jedná se veskrze o provedení softwarového řešení. Spojením těchto třech částí byl elegantně vyřešen problém definování rozhraní mezi bloky, které nesouvisí přímo s robotem. Stal se tak vnitroblokovým problémem. Detailní rozbor této části lze nalézt v [34].

Detekce - provedení této části plně odpovídá technickému bloku detekce. Práce je veskrze softwarového ražení s důrazem na sensoriku. Týká se pouze robotu. Detailní rozbor této části lze nalézt v [33].

Pohyb - tvoří kompaktní celek spojený s pohybem robotu. Jedná se o část, která se zabývá zejména softwarovou částí s prvky sensoriky. Detailní rozbor této části lze nalézt v [35].

Zalévání / čerpání vody - tvoří komplexní celek, který má za úkol rozbor výše definované úlohy. Tento blok obsahuje mechanické, elektronické, tak softwarové řešení pouze na robotu. Detailní rozbor této části lze nalézt v [36].

Integrace - integrační blok je popsán v celé této práci. Po fázi ideového návrhu má za úkol zejména provedení konkretizace návrhu, posléze integraci dílčích celků a konečné provedení ověřování a testování prototypu. V neposlední řadě též provádí i implementaci některých procesů na robotu, které se už nevměstnaly do výše zmíněných funkčních bloků.⁷

4.4 Shrnutí ideového návrhu

V této kapitole byla nastíněna výsledná podoba navrhovaného systému. Na základě ideového návrhu bylo možné zběžně posoudit náročnost jeho řešení, či rozsah práce. Z tohoto důvodu tato fáze probíhala v několika iteracích, po kterých se návrh stabilizoval na popsané podobě.

Ideový návrh měl důsledky v technickém návrhu zejména na oddělení hlavní funkcionality systému - zalévání rostlin od hledání rostlin. Diskuze i bližší rozbor naznačily, že by práce na jedné funkcionalitě při zahrnutí obou činností vysoce převyšovala možnosti našeho malého, ale o to více nadšeného kolektivu. V následující práci byl tedy kladen důraz pouze na hlavní funkcionalitu, která bude rozpracována v konkrétním návrhu.

Tato fáze byla klíčová z hlediska organizace práce uvnitř projektu. Lze říci, že po provedení ideového návrhu, měl každý řešitel jasné povědomí o tom, za co bude v projektu zodpovědný a kterým směrem se jeho práce bude ubírat.

⁷Jsou tedy provedeny nad rámec této práce.

Kap. 5

Návrh pro cílovou platformu

Po provedení základního - ideového konceptu řešení zadání, lze přejít k bližší specifikaci jednotlivých bloků. Projekt se v této fázi dostává z fáze *Plánování / Analýzy* do *Návrhu*, která musí jednoznačně definovat konkrétní podobu systému.

Již v předchozí fázi projektu byl položen základ systémového návrhu, tj. souboru bloků v systému a vztahů, či interakcí mezi nimi. Vzhledem k obecnému pojetí, co takový systémový návrh znamená, je možné zkoumat v závislosti na jeho komplexnosti jeho různé aspekty - jednotlivé funkční bloky, rozhraní mezi nimi, ale též metodiku práce, časový harmonogram vývoje, způsoby komunikace v týmu atd.. Význam systémového návrhu a jeho vliv na podobu výsledného celku roste úměrně rozsahu projektu.

Dobrý systémový návrh lze charakterizovat dle [31] některými obecnými rysy. Mezi ně patří například nutnost modelování celého systému - pod tím si lze představit libovolný model, který dokáže popsat danou řešenou problematiku tak, že z něho jde vyvodit některé závěry pro řešení problému. Návrh by měl být prováděn „od shora dolů“ - tj. od obecného konceptu, ke konkrétnímu řešení. Musí být kladen důraz na prioritizaci - nutnost uvažování o průběhu práce, do ní lze zařadit i práci na nejsložitějších částech systému v první řadě. Klíčová je i nutnost dekompozice systému na menší moduly a definování rozhraní mezi nimi - tj. provedení dělby práce, která mezi sebou má jasně nastavený způsob komunikace (technický, ale i lidský). Je nutné dbát i na oddělení funkcí jednotlivých modulů. Celý projekt by měl vést k co nejrychlejší stavbě prototypu, na kterém pak mohou být v co nejkratších intervalech okamžitě vyzkoušeny jednotlivé nově přidané funkce - tzv. *rapid prototyping*. Nesmí se zapomenout i na provedení projektové dokumentace. Dalších podobných pouček je celá řada.

Je zde nutné poznamenat, že systémový návrh, či související návrh systémové architektury je dynamický proces, při kterém jsou neustále promýšleny, či přehodnocovány jednotlivé jeho části. Jakoukoliv podobu systémového návrhu běžícího projektu lze vnímat jako stav okamžitého řešení.

V této práci je systém vnímán jako soubor bloků (technických / organizačních), které provádějí definované procesy. Proces lze chápat jako funkci, nebo soubor subprocessů, který zajišťuje provedení určeného úkolu. Jednotlivé procesy mezi sebou komunikují pomocí definovaného rozhraní. Systémový návrh je pak proveden jako pseudokód algoritmů, který obsahuje návaznosti a reakce jednotlivých procesů. Procesy jsou posléze přiřazeny jednotlivým blokům, které zajišťují jejich zpracování, návrh a implementaci. Je nutné poznamenat, že proces v závislosti na konkrétním bloku může zahrnovat jak softwarové řešení, tak mechanické, či elektronické. V návrhu je kladen důraz na dodržení zásad dobrého systémového návrhu.

Cílem této kapitoly je tedy provedení systémového návrhu - návrhu algoritmů a iden-

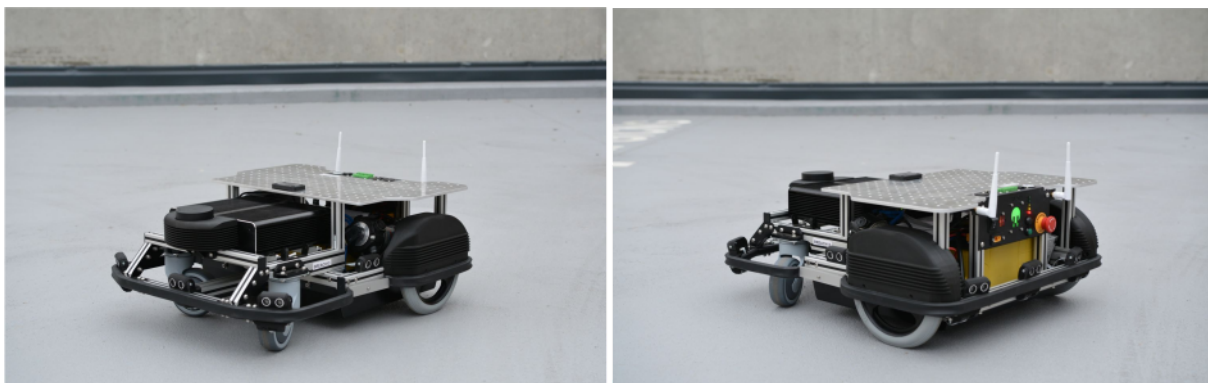
tifikace jednotlivých procesů, které zajišťují funkcionalitu z předchozí kapitoly.

5.1 Základní technologie

Základním odrazovým můstkem pro konkrétní návrh systému je předpokládaná cílová platforma, která již zpočátku jasně definuje některé zejména fyzické, ale i hardwarové, či softwarové parametry.

5.1.1 Platforma Breach

Jako cílová platforma celého systému byl vybrán robot Breach firmy Bender Robotics, který byl i touto firmou pro realizaci projektu zapůjčen. Přehled jeho základních parametrů je uveden v tabulce 5.1. Jedná se o mobilní robot pro vnitřní prostory s neodpruženým diferenciálním typem podvozku se dvěma pojezdovými koly. Robot je určen pro provedení různých nástaveb, které mohou být umístěny na konzoli v jeho zadní části. Tyto nástavby mohou být spojeny s platformou pomocí uživatelského panelu, který se nachází pod konzolí. Uživatelský panel umožňuje připojení napájení na baterii 5V/12V, připojení k počítači nebo k routeru robotu, signalizuje stav robotu a v neposlední řadě též obsahuje nouzové Stop tlačítko.



Obr. 5.1: Mobilní robotická platforma Breach [37, převzato]

Robot je vybaven dvěma EC motory s enkodéry, které jsou ke kolům připojeny přes šnekové převody. Ty jsou řízeny pomocí regulátoru, který komunikuje pomocí sériové sběrnice UART přes USB s řídicím počítačem. Celý řídicí systém robotu je postaven na frameworku ROS, takže řízení posléze přebírá uzel, který ovládá jednotlivé motory pomocí zprávy typu *Twist cmd_vel*. Tento uzel též posílá data z odometrie. K regulátoru je doplněn modul, ke kterému jsou připojeny pomocné sonary a ochranný gumový pás s tlakovým čidlem pro detekci kolize s objekty. Robot je dále vybaven lidarem typu RPlidar A1. Do systému je pak připojen pomocí svého uzlu.

5.1.2 ROS

Jako softwarový základ byl zvolen systém ROS verze Noetic. ROS (*Robot Operation System*) je opensource nástroj, který je určen pro vývoj robotických systémů. Je tvořen řadou knihoven i užitečných programů. Obsahuje obecné rámce pro vyhotovení vlastních robotických funkcionalit, ale též i hotová řešení některých částí - lokalizace, navigace atd..

Parametr	Provedení
Celkové rozměry (D × Š × V)	620 × 540 × 255 mm
Typ podvozku	Diferenciální s pohonem dvou kol
Závěs kol	Pevný
Převodovka	Šneková 1:50 nebo 1:25. Zabraňuje pohybu robotu ve vypnutém stavu.
Hmotnost platformy	34 kg
Max. únosnost	20 kg
Max. rychlost	1,9 km/h nebo 3,8 km/h
Provozní rychlost	1,5 km/h nebo 3,0 km/h
Nájezd na jedno nabití	Min 7 km
Typ baterie	LiFePO4
Kapacita, napětí baterie	720 Wh, 12 V
LIDAR (laserový skener)	Rozsah 240 ° před robotem, dosah 6 m, přesnost 0,5-2 °
Pneumatické nárazníky	Celý obvod robotu
Ultrazvukové senzory	10×, dosah 4 m, přesnost 4 mm
Řízení	Pasivně chlazené PC s procesorem Intel i5 s předkonfigurovaným systémem ROS

Tab. 5.1: Základní parametry robotu [37, převzato, upraveno]

Pro potřeby systému byla přijata balíčková verze integrace systému, kterou ROS využívá. Jednotlivé procesy tedy budou po vyhotovení přiřazeny ke svému balíčku, který bude možné jednoduše vřadit do systému. Balíček obsahuje všechny potřebné soubory, programy atd., které slouží k provedení softwarové části procesu.

Softwarový základ každého procesu bude tvořit tzv. *uzel*. Systém založený na ROSu je vždy síťový a distribuovaný. Komunikaci mezi jednotlivými uzly řídí tzv. *master*, který pracuje na počítači, na kterém běží *roscore*. Ten navazuje spojení mezi jednotlivými uzly na základě def. odebíraných zpráv a imple. služeb v rámci sítě, na které běží. Po navázání komunikace už uzly přeposílají data přímo mezi sebou s využitím protokolů TCP, či UDP.

V práci je pro potřebu rozhraní mezi procesy používán standard systému ROS - od *akcí* - dlouhodobých úkolů s def. zpětnou vazbou, při kterých může být komunikován progres řešení, přes *služby* - jednorázový rychle proveditelný úkol s def. zpětnou vazbou až po *zprávy* - strukturované informace.

Zpráva - tvoří základní mechanismus pro posílání informací v ROSu. Existuje velké množství předdefinovaných zpráv, které se týkají např. popisu pozice v prostoru, navigace, sensoriky atd.. Pracuje na principu publisher-subscriber. *Publisher* posílá do systému zprávy s jednoznačně určeným typem a názvem. Ty jsou odebírány jednotlivými *subscriber* [38, 40].

Služba - pracuje na principu server-klient. Server představuje program, který čeká na zavolání klienta a vykonává takovou činnost, která je jednorázová a časově omezená. Klient aktivně čeká na výsledek [39, 40].

Akce - pracuje na principu server-klient. Server je reprezentován programem, který čeká na zavolání akce klientem s definovaným cílem. Po jeho získání přejde do stavu, kdy je daná akce vykonávána. V průběhu vykonávání může server posílat klientovi definovanou zpětnou vazbu. Klient nemusí aktivně čekat na výsledek, to je provedeno pomocí *callbacků*, které se vyvolávají při přechodu do různých stavů. Může též vykonávání akce přerušit. Po provedení akce posílá server klientovi def. výsledek [41, 40].

Server může obsluhovat několik akcí najednou, v této práci je však přijat požadavek u každého serveru obsluhovat pouze jeden aktivní cíl za účelem jednodušší správy aktivních procesů. V rámci výsledku je vždy posílám i tzv. **exitcode**, tj. číselná reprezentace

výsledného řešení akce, či služby - úspěšně provedeno, přerušeno, problém atd..

5.2 Návrh funkcionality

V předchozí kapitole byla vypracováno ideové řešení funkcionality systému, které bylo rozčleněno do několika fází. Nyní budou fáze podrobněji popsány a identifikovány jednotlivé procesy a jejich návaznosti.

Fáze mají jednoznačně danou posloupnost - po *spuštění systému* dochází k *provozu*, ve kterém je vybrána aktuální funkcionalita systému - *zalévání rostlin*, nebo *hledání rostlin*. Posléze je systém buď uspán, nebo dojde k *vypnutí systému*.

5.2.1 Provozní fáze - start, provoz, vypínání

Start systému je zahájen uživatelským spuštěním systému robotu a serveru. Po rozběhu operačního systému robotu dochází k automatickému spuštění ROSu. Po tomto může dojít ke spuštění řídicího programu, který posléze kontrolovaně spouští ostatní uzly v systému. Řídicí program poté může přejít k řízení procesů.

Provoz systému tvoří krátké mezidobí, které slouží k přechodu k jednotlivým funkcionalitám. Při přechodu do provozu dochází ke spouštění procesů, které mají výrazný asynchronní, či paralelní charakter a slouží jako podpůrné bloky. Spuštěny jsou též veškeré uzly systému.

Základním procesem je komunikace robotu se serverem, potažmo s UI. Tento proces zajišťuje přístup do systémové databáze a umožňuje uživateli konfiguraci a sledování stavu systému. Mezi další podpůrné procesy patří ty, které jsou spojeny s fyzickým popisem prostředí a robotu, a ty, které běží neustále - např. lokalizace, procesy spojené se sensorikou, či ovládáním pohonů.

Robot se po startu nachází v předem známé pozici - u základny. Po aktualizaci dat, robot přejde ke konkrétní funkcionalitě. Vzhledem ke stavu řešení je vybráno vždy zalévání rostlin.

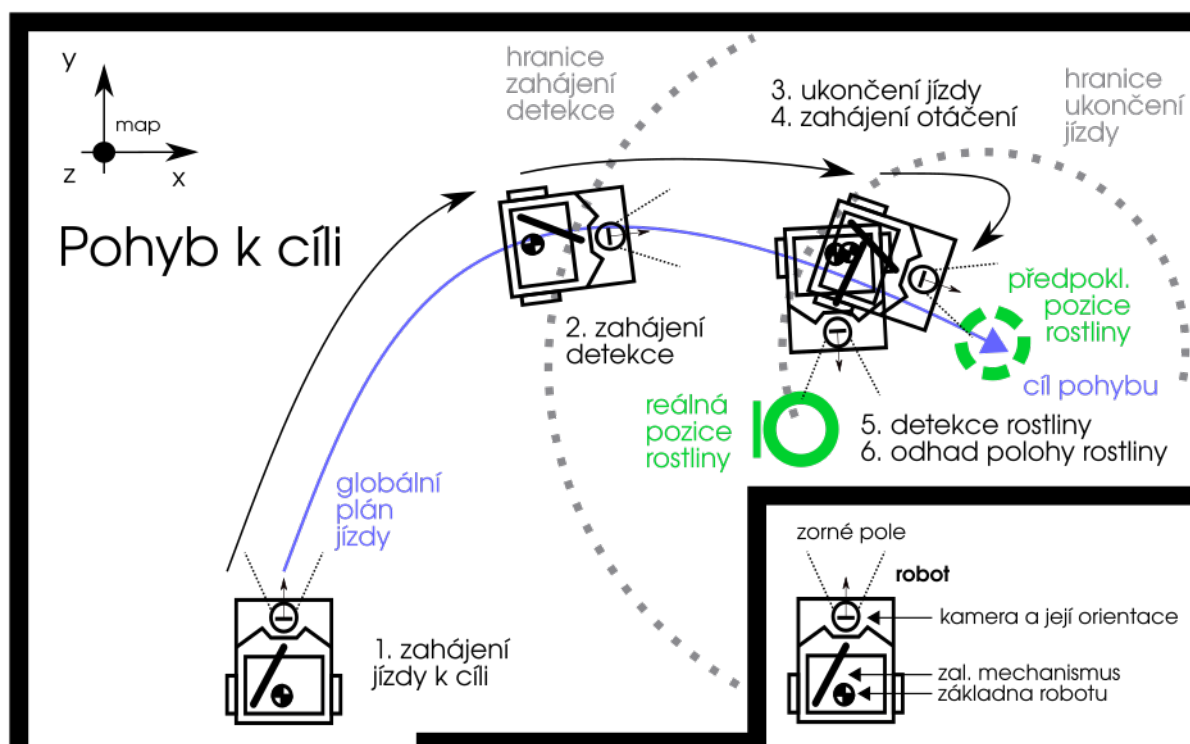
Po provedení funkcionality se robot vrací do výchozí pozice. V závislosti na stavu pak přechází buď do režimu spánku, nebo dochází k vypnutí celého systému robotu. V režimu spánku by měly být vypnuty všechny procesy vyjma komunikace s databází a řídicího programu. Systém by měl přejít do aktivního stavu v čase další plánované zálivky. Při vypnutí systému dochází při potřebě zálevání opět k nové inicializaci všech procesů.

5.2.2 Zalévání rostlin

Funkcionalita zalévání rostlin je ideově členěna do 4 fází - *plánování práce*, *pohyb k cíli*, *identifikace cíle* a *zalévání / čerpání vody*.

Plánování práce - po výběru funkcionality zalévání rostlin přechází řídicí program k plánování práce. Z databáze jsou vybrány ty rostliny, které mají plánovanou zálivku v daný čas. U těchto rostlin je zjištěna i jejich odhadovaná poloha a množství vody, potřebné k zálivce, spolu s těmi jsou zjištěny informace i o dostupných zdrojích vody. Na základě těchto údajů, může být posléze přistoupeno k rozplánování práce - tj. tvorby harmonogramu, který zajišťuje optimální sled zalévání rostlin a jednotlivých čerpání. Při tomto plánování může být optimalizován čas zalévání, či ujetá vzdálenost. Výsledkem této fáze je tedy harmonogram, který je vykonáván v cyklu řídicím programem.

Pohyb k cíli - je fáze, při které robot má za úkol dojet k místu, kde se podle dat v databázi nachází požadovaný objekt - rostlina, nebo zdroj vody. Robot se tedy autonomně pohybuje předem známým prostorem. Při provádění pohybu je sledována poloha robotu, kdy je v určité vzdálenosti před cílem zahájena detekce objektu.



Obr. 5.2: Fáze pohybu k cíli

Při úspěšné detekci rostliny dochází k přerušení příjezdu a je zjištěna její přibližná pozice. Na základě této informace je pak možné v rozhodovacím procesu přibližně určit, zdali se jedná o požadovanou rostlinu, či nikoliv¹.

Pokud během jízdy k detekci objektu nedojde, robot provede otočení na místě. Pokud při něm dojde k detekci, proběhne obdobný proces jako v předchozí stati.

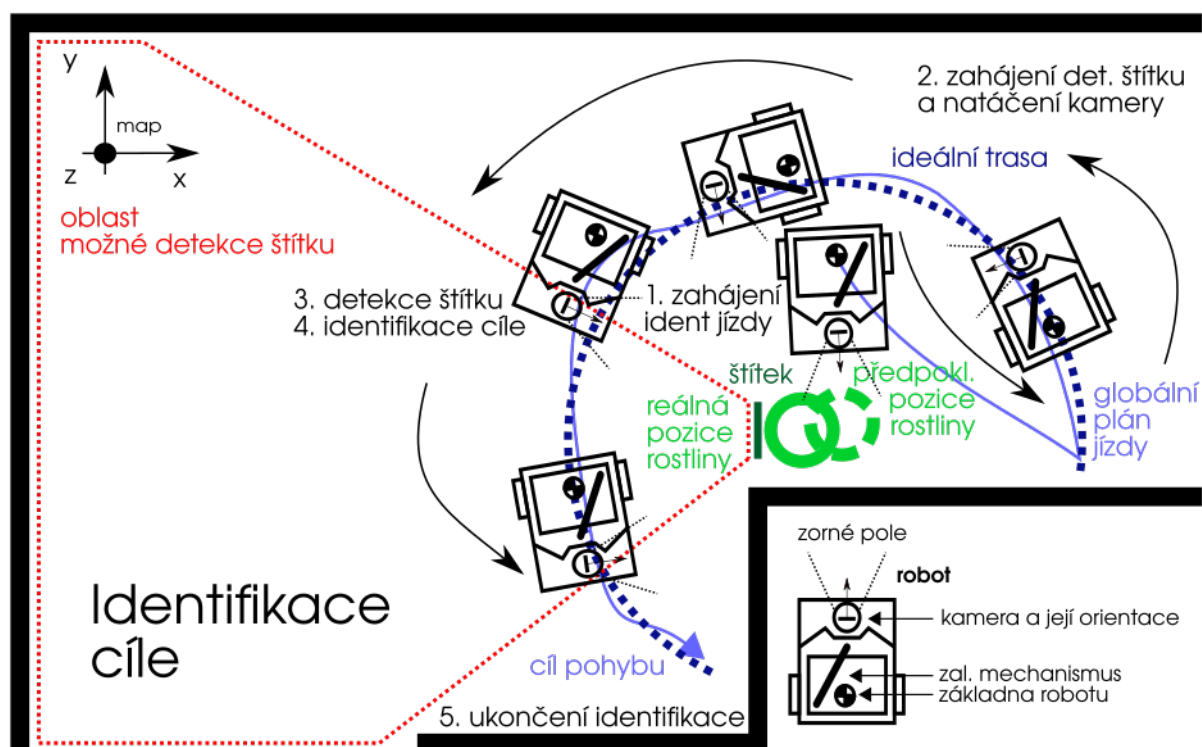
Pokud je tato fáze neúspěšná, robot aktualizuje informace o neproveditelné zálivce dané rostliny v plánu práce a přejde plynule k další položce v harmonogramu.

Identifikace cíle - po provedení detekce objektu může dojít k jeho identifikaci. Ta je provedena pomocí detekce a dekodování identifikačního štítku, který je umístěn na květináči, či zdroji vody. Ke každému objektu je přiřazen unikátní kód, který je systémem automaticky vygenerován.

Robot má za úkol na základě odhadu pozice objektu z předchozí fáze provést její objetí. Současně dochází k natáčení kamery směrem k detekované rostlině.

Po úspěšné detekci štítku dochází k přerušení identifikačního pohybu. Řídicí program poté může usoudit dle identifikačního kódu, zdali se jedná o hledaný objekt. Pokud k detekci štítku nedojde, k přerušení identifikace dochází po provedení plného, nebo maximálně možného objetí objektu, o tom je rozhodnuto rozhodovací strukturou, která to provádí na základě rozplánování pohybu a aktuálního plnění objetí objektu.

¹Např. v situaci, kdy dojde k detekci více rostlin.



Obr. 5.3: Fáze identifikace cíle

Pokud k detekci správného štítku nedojde, řídicí program opět provede aktualizaci dat v systému a přejde k další položce v harmonogramu.

Zalévání / čerpání vody - pokud je objekt úspěšně identifikován, může dojít v závislosti na jeho typu - k zalití, nebo načerpání vody. To je provedeno v několika fázích. K přesnému navedení robotu na cíl dochází pomocí detekce a lokalizace identifikačního štítku.

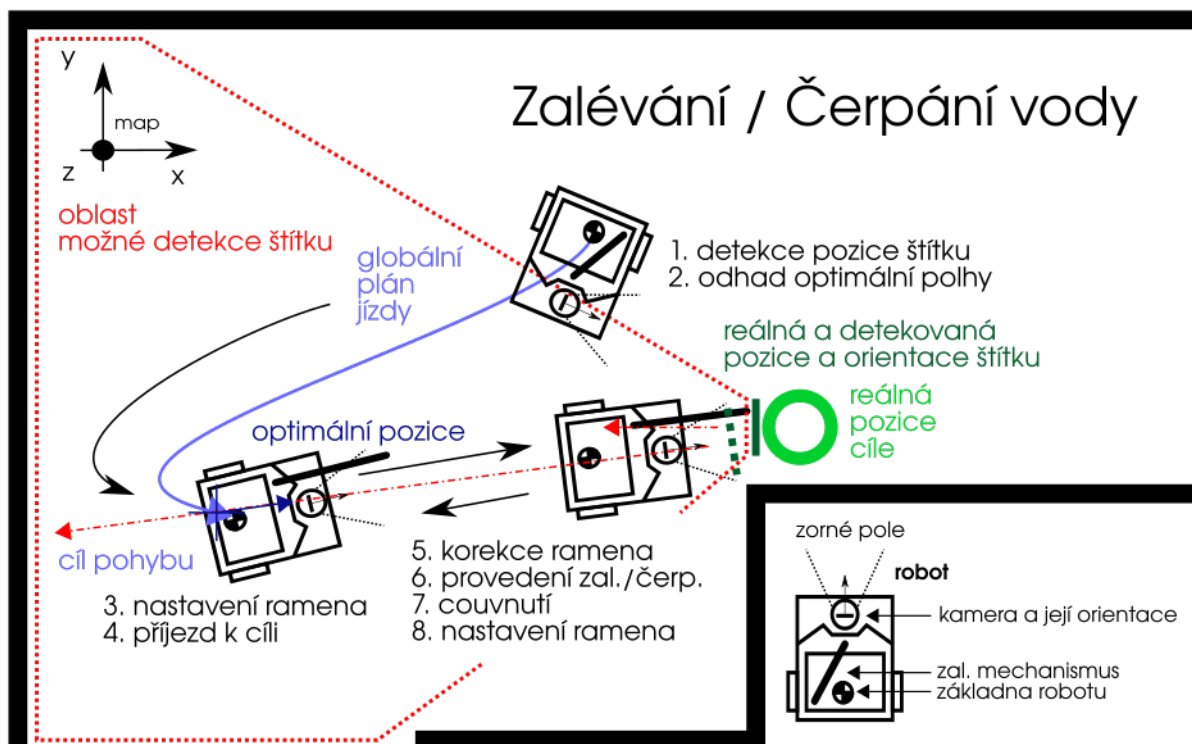
V této fázi dochází ke spuštění průběžné detekce štítku, kdy je periodicky zjišťována jeho pozice vůči kameře. Po přepočtení této polohy do absolutního souřadnicového systému mapy dochází ke zjištění tzv. optimální pozice, tj. pozice, která je přesně před štítkem v určité vzdálenosti a je orientována směrem k objektu. Robot je poté k této pozici naveden.

Po navedení do optimální pozice dochází k přenastavení zalévacího ramena do pracovní polohy, tj. úhlovému a výškovému nastavení dle informací o výšce květináče z databáze a popř. i výšky umístění objektu určené z detekce štítku, či detekce objektu.

Poté dochází k příjezdu k rostlině, což je pohyb, který je regulován pomocí dat z detekce štítku. Přerušen je ve chvíli, kdy se robot dostane do definované vzdálenosti od štítku, resp. objektu, nebo dojde k detekci objektu sensory umístěnými na zalévací platformě.

Po přiblížení je provedena kontrola správné pozice ramena nad objektem a její dodatečná korekce. Poté dojde k zalití rostliny množstvím vody, které je převzato z databáze, nebo k načerpání vody. Poté je provedeno couvnutí robotu a nastavení ramena do přepravní pozice.

Po ukončení této fáze je provedena aktualizace systémových dat, kdy je na server poslán výsledek pro aktuální objekt zájmu a je aktualizována jeho poloha dle detekce.



Obr. 5.4: Fáze zalévání / čerpání vody

5.3 Identifikace procesů

Po nastínění jednotlivých fází provedení celé navrhované funkcionality lze identifikovat jednotlivé procesy, které ji budou zajišťovat. Výčet je proveden od řídicích procesů k procesům pomocným.

Řídicí program - provádí řízení jednotlivých procesů. Jejich sled je patrný v příložených diagramech 12.1, 12.2 a 12.3. Sledují navrženou funkcionality z předchozí stati.

Zodpovědný org. blok	UI / Server / Řídicí program
Část systému	Robot

Navržený pseudokód provedený v jazyce UML obsahuje požadované hlavní kroky řešení v chronologickém pořadí pro každou fázi funkcionality. Procesy stojící mimo funkcionality jsou popsány dále. V diagramech je patrná určitá paralelnost.

Server - hlavním úkolem systémového serveru je uchovávání informací a moderování komunikace mezi UI a robotem. Součástí tohoto procesu je i vyhotovení komunikačního mostu UI/Server a Server/Robot.

Zodpovědný org. blok	UI / Server / Řídicí program
Část systému	Server

Server v základu musí uchovávat 3 typy informací - o rostlinách, zdrojích vody a prostředí. Pro potřeby uživatelského rozhraní je nutné zaznamenávat i aktuální stavové informace o systému.

Každému objektu je přiřazeno unikátní identifikační číslo - ID, na základě kterého je jednoznačně identifikovatelný. K tomu je uvedena přibližná poloha objektu v prostoru. K rostlině je přiřazeno datum poslední zálivky, frekvence zalévání a potřebné množství vody. U zdroje vody je uchováváno množství dostupné vody. U prostředí je v základu uchovávána jeho mapa a informace o ní.

Tato základní struktura posléze může být rozšířena o zadávání spec. oblastí, či konkrétního robotu. Systém může uchovávat též některá personalizovaná data pro lepší přehlednost při obsluze systému.

Mezi stavové informace patří zejména aktuální poloha robotu, stav jednotlivých modulů, či procesů, logování jeho pracovní činnosti nebo pracovního harmonogramu.

Uživatelské rozhraní - hlavním úkolem uživatelského rozhraní je provedení interakce systému s uživatelem.

Zodpovědný org. blok	UI / Server / Řídicí program
Část systému	UI

Systémová aplikace by měla být základní vstupní branou uživatele do systému. Měla by tedy obsahovat jednoduchý a názorný manuál pro jeho prvotní, či průběžnou konfiguraci. To zahrnuje např. opatření všech objektů patřičnými identifikátory, či důrazné upozorňování uživatele na vlastnosti a omezení systému.

Uživateli musí být umožněn přístup ke kontrole, či změně uchovávaných dat o systému v databázi. Musí být proveden individuální přístup ke každému objektu.

Dalším úkolem aplikace je umožnit uživateli řízení systému. Ať již pracovní - spuštění zalévacího cyklu, okamžité ukončení práce, příkaz pro návrat do výchozí pozice, tak prosté manuální řízení robotu - v podstatě provedení dálkového ovládání.

V neposlední řadě aplikace musí být schopna v rozumné míře spravovat a kontrolovat aktuální a plánovanou práci, či stav systému. Pod tím je myšlena poloha robotu, stav baterie, množství aktuálně vezené vody, výpis závad, či problémů.

Popis prostředí a robotu - je proces, který zahrnuje několik subprocessů, které se zabývají popisem prostředí a robotu v běžícím systému ROS.

Zodpovědný blok	Pohyb
Část systému	Robot

Základní fyzický popis robotu i prostředí je proveden v systému ROS pomocí souřadnicových systémů, které tvoří stromovitou strukturu a transformací mezi nimi, které jsou spravovány balíčkem *tf2*. Ten umožňuje sdílení informací o transformacích mezi tělesy v systému.

Základním s.s. robotu je **base_link**, který je umístěn v základně robotu. Ten je obvykle přímým, či nepřímým potomkem s.s. mapy **map**. Robot i prostředí obsahují další s.s., která jsou spojeny buď s jednotlivými tělesy na robotu, nebo jsou vytvářeny v rámci specifických procesů - např. s.s. odometrie **odom**.

O posílání s.s. robotu a mapy se stará obvykle větší množství uzlů - i procesů. Základní posílání je provedeno obvykle tzv. statickými publishery, kteří posílají do systému ty transformace, které jsou časově neproměnné.

Společně s mapovým s.s. musí být do systému vysílána zpráva typu *nav_msgs/OccupancyGrid*, která obsahuje mapu prostředí. Tato mapa pak slouží jako podklad pro práci dalších procesů - lokalizace, či plánování pohybu.

Lokalizace - je proces, který zajišťuje robotu odhad jeho polohy v prostoru.

Zodpovědný blok	Pohyb
Část systému	Robot

Lokalizace probíhá na základě dat ze sensorů robotu a dostupné mapy prostředí. Základním lokalizačním prostředkem cílové platformy je lidar společně s odometrií. Lokalizace obvykle pracuje s prvotním odhadem polohy. Výsledkem lokalizace je provedení transformace z s.s. **odom** do s.s. **base_link**.

Vzhledem k rozsahu projektu bylo zvoleno řešení lokalizace s pomocí hotového balíčku v systému ROS - AMCL.

Sensorika a řízení pohonů - je skupina subprocessů, která zajišťuje získávání dat a předávání řídicích povelů do regulačních prvků. Zahrnuje sensory a pohony primárně určené pro pohyb.

Zodpovědný blok	Integrace
Část systému	Robot

Jedná se zejména o provedení komunikace s lidarem a pohony robotu. Všechny funkce spojené s tímto účelem jsou provedeny pomocí převzatých balíčků.

Do systému pak jdou data z lidarů ve standardní formě zprávy typu *nav_msgs/Scan* a řízení pohonů je provedeno pomocí zprávy typu *geometry_msgs/Twist*. Odometrie je do systému posílána pomocí transformací z s.s. **odom** do s.s. **base_link**.

Sestavení tabulky vzdáleností - je proces, který úzce souvisí s plánováním práce. Jedná se o sestavení tabulky vzájemných vzdáleností objektů v prostoru na základě mapy prostředí.

Zodpovědný blok	Pohyb
Část systému	Robot

Sestavení plánu práce - sestavení plánu práce má za cíl definování harmonogramu zalévání jednotlivých rostlin, který bude proložen čerpáním vody.

Zodpovědný blok	UI / Server / Řídicí program
Část systému	Robot

Sestavení takového harmonogramu je v podstatě optimalizační proces, který pracuje s tabulkou vzdáleností jednotlivých objektů v prostoru. Plánování pak má za úkol minimalizaci ujeté vzdálenosti, či vhodné načasování jednotlivých záливоk.

Pohyb k cíli - je proces, který zajišťuje plánování pohybu robotu ve fázi pohybu k zadanému cíli.

Zodpovědný blok	Pohyb
Část systému	Robot

Proces si na základě převzatého ID cíle v databázi zjistí předpokládanou polohu cíle. Poté musí naplánovat optimální globální trasu k cíli, kterou pak na základě lokálního plánování musí provést. Během cesty musí adekvátně reagovat na možné překážky na trase. Pokud se cíl cesty nachází v detekovatelné překážce musí robot provést alespoň přiblížení na definovanou vzdálenost. V rámci zpětné vazby vrací hodnotu procentuálního plnění projetí plánované trasy.

Proces je ukončen na základě informace o dojetí do blízkosti def. cíle, nebo je přerušen nadřazeným řídicím programem. Pokud je v cestě nalezena nepřekonatelná překážka, či nastane jiný poruchový stav, přeruší proces činnost a dá řídicímu programu zpětnou vazbu ve formě chybového exitkódu.

Proces vyžaduje implementaci vlastního komplexního plánovače, kterého by posléze bylo možné užít i v některých dalších procesech. Důvodem k požadavku na vypracování vlastního plánovače byl zejména vyvarování se některých nechtěných vlastností u již hotových řešení v ROSu, či snaha mít plánování pohybu více pod kontrolou.

Přepočty v souřadnicových systémech - je proces, který zajišťuje přepočítávání souřadnic mezi souřadnicovými systémy.

Zodpovědný blok	Integrace
Část systému	Robot
Rozhraní	Služba
Název	<i>calculatePosition</i>

Tento proces dovede z def. souřadnicového systému ve kterém je umístěn bod, přepočítat jeho souřadnice do zvoleného cílového systému. To je možné díky stromovité struktuře transformací v ROSu. Cílový i aktuální s.s. musí být součástí jednoho transformačního stromu.

Z pohledu systému ROS se jedná o vcelku jednoduše uskutečnitelnou funkci, kterou je možno provést pomocí knihovny *tf2_ros*. Centrální přepočet byl navrhnut za účelem zjednodušení provedení jednotlivých dílčích procesů a snadnější integraci a ověřování celkové práce systému.

Snímání kamerou - proces zajišťuje přebírání obrazových dat z kamery. Obraz je do systému posílán kontinuálně.

Zodpovědný blok	Detekce
Část systému	Robot
Rozhraní	Zpráva
Název	<i>videoFrames</i>

Pro potřeby systému jsou plánovány 2 kamery na robotu. Hlavní kamera tzv. pohybová by měla být umístěna na podvozku robotu v jeho přední části před pojezdovými koly na dvouosém pohyblivém mechanismu. Měla by sloužit primárně k detekci rostlin při příjezdu k cíli a prvotní detekci štítku. Druhá kamera tzv. detekční by měla být umístěna pevně na pohyblivé části zalévacího ramena. Sloužit by měla k přesnějšímu navedení robotu k rostlině ve fázi zalévání.

Uzel by měl být schopen s def. frekvencí posílat snímky do systému ve formátu *sensor_msgs/Image* s požadovaným rozlišením.

Detekce rostliny / zdroje vody - je proces který má za úkol rozpoznat objekt v prostoru.

Zodpovědný blok	Detekce
Část systému	Robot
Rozhraní	Akce
Název	<i>detectGoal</i>

Proces si na základě přijatého ID zvolí požadovaný typ detekce - rostlina, či zdroj vody. Detekce bude probíhat na základě obrazových dat z kamery robotu. Výsledkem detekce je zpětná vazba pro řídicí program o úspěšné detekci požadovaného objektu.

Určení pozice rostliny - je proces, který zajišťuje lokalizaci detekovaných objektů v prostoru.

Zodpovědný blok	Detekce
Část systému	Robot
Rozhraní	Akce
Název	<i>detectPositionGoal</i>

Proces je volán po úspěšné detekci cíle. Na základě zvolené metody musí určit relativní souřadnice objektu vůči kameře v souřadnicovém systému **detection_link**. Tato poloha je posléze přepočítána do absolutního souřadnicového systému prostředí **map** pomocí procesu přepočtů souřadnic a je poslána řídicímu programu.

Orientace kamery - je proces, který zajišťuje natáčení kamery ve dvou osách. Jeho úkolem je návrh kamerového mechanismu, jeho řízení a provedení interakce se zbývajícím systémem.

Zodpovědný blok	Integrace
Část systému	Robot
Rozhraní	Akce, Služba
Název	<i>setAutoOrientation, setPitchOrientation, setYawOrientation</i>

Principiálně musí být kamerový modul vybaven natáčecím mechanismem, pohony ve dvou osách, řídicí jednotkou, která posléze bude komunikovat s nadřazeným programem, který bude komunikovat s ROsem. Na mechanismu bude umístěna pohybová kamera.

Mechanismus musí do systému posílat transformace z jeho def. souřadnicových systému, jedná se zejména o s.s. **camera_link**, který je pevně spojen s otočnou částí mechanismu a **detection_link**, který se nachází na stejné pozici, ale je vhodně natočen za účelem jednoduché konverze relativních souřadnic z detekce.

Proces zajišťuje kontinuální natáčení kamery vzhledem k zadanému cíli v s.s. **map**. Doplňkové služby jsou provedeny za účelem jednorázového natočení ve zvolené ose.

Identifikační pohyb - je proces, který zajišťuje objetí detekovaného objektu v identifikační fázi funkcionality hledání rostlin.

Proces převezme jako vstup bod v s.s. mapy, který je považován za umístění objektu v prostoru. Na základě tohoto bodu provede plánování pohybu, který má za úkol v ideálním

Zodpovědný blok	Pohyb
Část systému	Robot
Rozhraní	Akce
Název	<i>moveToIdentify</i>

případě objektů rostliny po kružnici s def. poloměrem. Proces musí zajistit vhodným natočením robotu v průběhu jeho činnosti možnou detekci štítků pomocí pohybové kamery.

V závislosti na množství překážek, ať již statických - zanesených v mapě, nebo dynamických, musí provádět přeplánování trasy. Proces musí též provést rozhodnutí o ukončení identifikačního pohybu, které musí přijmout po provedení plného objektu. Řídícímu programu musí být umožněno okamžité přerušení pohybu.

Detekce štítků - je proces, který zajišťuje detekci identifikačních štítků ve fázi identifikace a zalévání/čerpání vody. Úkolem procesu je i volba vhodného identifikátoru. Detekce štítku musí být provedena s jistotou a vhodnou přesností.

Zodpovědný blok	Detekce
Část systému	Robot
Rozhraní	Akce
Název	<i>detectMarker, detectMarkerContinuous</i>

Proces je zahájen převzetím specifického ID, které má detekovat. V fázi identifikace je provedena jednorázová detekce štítku, jejíž výsledkem je potvrzení o detekci def. ID. Ve fázi zalévání je prováděna průběžná detekce, která vrací detekovanou rel. pozici štítku v rámci zpětné vazby akce. Ta je zjišťována v rámci s.s. **detection_link**.

Přepočet dat ze štítku - je proces, který zajišťuje vhodnou konverzi dat z detekce štítku pro ostatní procesy systému.

Zodpovědný blok	Integrace
Část systému	Robot
Typ procesu	Akce
Název	<i>tf_marker_calculation</i>

Proces zajišťuje s pomocí přepočtů souřadnic v prostoru odesílání - polohy identifikačního štítku v s.s. **map**, polohu tzv. optimální pozice v s.s. **map**, tj. pozice v určité vzdálenosti před štítkem orientovaná směrem k němu, úhlu natočení osy robotu vůči štítku a v poslední řadě vzdálenost kamery od štítku.

Konverze dat je určena zejména pro procesy spojené s finálním pohybem při zalévání/čerpání vody a samotným zaléváním/čerpáním vody.

Pohyb zalévací / čerpací - je skupina subprocessů, které zajišťují pohyb robotu ve fázi zalévání/čerpání vody ve funkcionalitě zalévání rostlin.

Prvním procesem je pohyb do tzv. optimální pozice. Ten je proveden na základě odebrané zprávy o její poloze v prostoru, kterou převezme od procesu přepočtů dat ze štítku. Robot posléze musí provést pokus o dosažení této pozice s danou orientací. Měl by obsahovat též určitý rozhodovací proces, který dokáže vyhodnotit úspěšnost dosažení optimální pozice.

Zodpovědný blok	Pohyb
Část systému	Robot
Rozhraní	Akce
Název	<i>moveToOptimalPosition, moveCloserToPlantpot, moveBackward</i>

Druhým pohybem je pohyb robotu k cílovému objektu. Vzhledem k umístění štítku na objektu, bylo zvoleno schéma jednoduchého regulovaného pohybu řízeného pomocí dat z procesu přepočtu dat ze štítku - úhlu natočení robotu a vzdálenosti od štítku. Do regulace, či řízení by měla být zahrnuta i data ze zalévací platformy, která by měla zajišťovat přesnější detekci objektu. Proces musí vyhodnocovat sensorická data z několika zdrojů.

Posledním pohybem je prosté couvání robotu od objektu pro zajištění bezproblémového odjezdu od rostliny a složení ramena do přepravní pozice.

Zalévání / čerpání vody - je proces, který má za úkol finální zalití rostliny, či načerpání vody ze zdroje. Cílem tohoto procesu je analýza, návrh a provedení mech. řešení zalévací platformy, jejího řízení a konečné zapojení do systému.

Zodpovědný blok	Zalévání/ čerpání vody
Část systému	Robot
Rozhraní	Akce, Služba
Název	<i>doWP, setWorkPosition, setBasePosition</i>

Zalévací platforma se musí skládat ze tří částí - nádrže na vodu, rozvodu vody a zalévacího ramena. Zásoba vody se odvíjí od nosnosti využití platformy, u cílové platformy se pohybuje v řádu 10-15 l, to umožňuje zalití přibližně 50 rostlin na jedno čerpání².

Rozvod vody by měl splňovat funkci jak čerpání vody, tak zalévání. V rámci procesu se musí definovat i charakteristiky, omezení, či provedení zdroje vody nebo podoba rostlin.

Varianta koncového členu s pohyblivým ramene byla zvolena prakticky jako jediná proveditelná varianta, která umožňuje možné adaptabilní, rozšířitelné a spolehlivé řešení spojení zalévací a čerpací funkce. Na základě předpokládaného využití systému ve vnitřních prostorách administrativních budov byly stanoveny základní požadavky na rozsahy pohybu ramena. Rameno dle návrhu musí být vybaveno přiměřeným množstvím sensoriky, které umožní jeho spolehlivou funkci.

Celý proces je do systému vřazen hlavní akcí - provedením zalévání/čerpání a dvěma službami, které převádějí rameno do pracovní a cestovní pozice. Doplnkové služby slouží kromě přenastavení pozic ramena k včasné konfiguraci celé platformy před, nebo po provedení hlavní akce.

V rámci hlavní akce robot musí provést konečnou korekci ramena na daný objekt a rozhodnout na základě sensorických a uchovávaných dat, zdali je možné provést daný úkon. Posléze může s využitím převzatého ID cíle stanovit potřebné množství vody, kterou má vyčerpat, či načerpat. Výsledek akce předá řídicímu programu.

²Dle průzkumu průměrné potřeby závlivky rostliny v domácnosti - 250 ml.

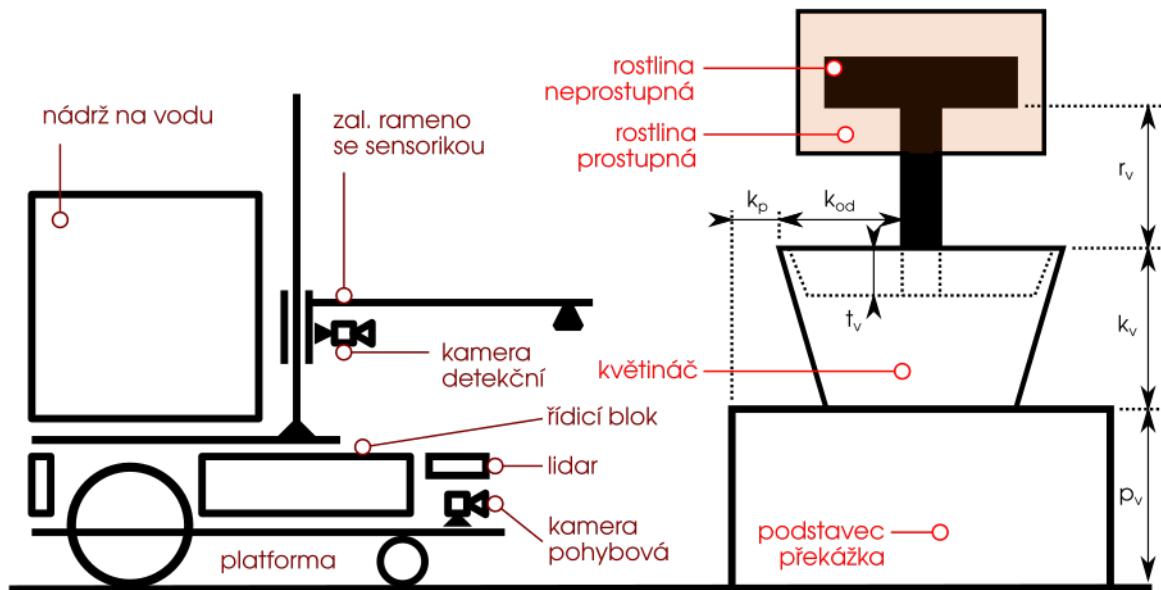
5.4 Shrnutí konkrétního návrhu robotu

Cílem této kapitoly bylo rozpracování ideového návrhu pro cílovou platformu. Nákres výsledné předpokládané podoby robotu lze vidět na přiloženém obrázku. Přiložená tabulka následně ukazuje klíčové parametry systému, na které je v průběhu dalšího detailnějšího návrhu kladen důraz.

Parametry překážky byly odvozeny od průměrných rozměrů pracovního stolu, př. nízkého parapetu. Pracovní prostor představují minimální potřebné rozměry u objektu pro provedení spolehlivé práce zalévací platformy. Rozměry objektu definují minimální možnosti systému.

Parametr	Popis	Cíl
k_p	hloubka překážky	max. 0.4 m
k_{od}	hloubka pracovního prostoru	minimalizace
k_v	výška objektu	max. 0.3 m
t_v	dosažitelná hloubka objektu	maximalizace
r_v	výška pracovního prostoru	minimalizace
p_v	výška překážky	max. 0.75 m

Tab. 5.2: Tabulka cílových parametrů systému



Obr. 5.5: Předpokládaná podoba robotu a parametry objektu, resp. rostliny

Kap. 6

Integrace systému

Po provedení konkrétního návrhu byly důkladně rozpracovány jednotlivé definované procesy určenými bloky. Po jejich implementaci bylo možné přistoupit k jejich integraci do jednoho celku.

Je nutné podotknout, že úspěšnost provedení integrace je závislá už na fázi ideového návrhu, který je posléze vyspecifikován do konkrétnější podoby. Pro její co největší jednoduchost a efektivitu je nezbytné provedení popisu rozhraní v systému, které musí být přehledné, jasně určené a každý blok v systému k němu musí mít přístup.

Integrace systému v čase probíhala postupně dle vyhotovení jednotlivých procesů. Do integrace samozřejmě patří nejen softwarové řešení projektu, ale též i hardwarová část. Hardwarová integrace silně závisí na fázi ověřování a vyvrcholí v podstatě jen na půli cesty - stavbou prototypu. Softwarová část se naproti tomu liší pouze v konkrétních detailech aktuální testované platformy.

V práci byl využit pro integraci softwarové části velice výhodný způsob balíčkování ROSu jednotlivých dílčích celků. Díky jednoduše použitelnému způsobu komunikace bylo rozhraní nadefinováno přesně - jeho typ i předávané informace, ale zároveň nebylo nutné určit, jak daná funkcionalita bude provedena, což jednotlivým řešitelům v rámci projektu nesvazovalo ruce.

V této kapitole bude popsána výsledná podoba integrovaného systému s důrazem na odlišnosti oproti předpokládanému řešení.

6.1 Základní technologie

6.1.1 Git

Jako hlavní integrační nástroj byl využit Git. Je to verzovací systém, který umožňuje správu a zálohování projektů.

Pro účely projektu byl pro každý blok vytvořen tzv. *branch*, což je v podstatě určitý obraz projektu. Tyto *branch* pak mají možnost ukládání změn v čase vytvářením tzv. *commitů*, jejichž historie je zaznamenávána a systém se tak může při potřebě vrátit zpět do určitého uloženého stavu bez ztráty dat.

Git je distribuovaný. Každý řešitel měl k dispozici lokální adresář, který posléze synchronizoval se serverem. Tím byla provedena jak záloha práce, tak i možnost jednoduchého dálkového přístupu integrátora k dílčím celkům. Jednotlivé *branch* pak lze spojit do jednoho celku - provést tzv. *merge* a vytvořit tak společný výsledek.

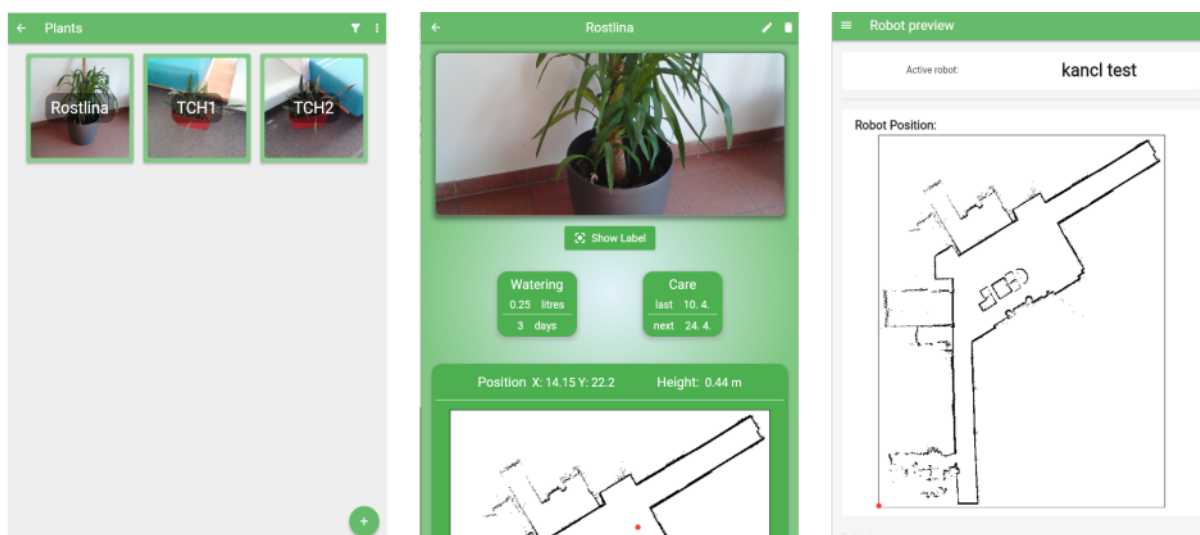
V projektu každý *branch* obsahuje jednotlivé ROS balíčky, na kterých daný blok pracuje.

6.2 Prototyp systému

Výsledná podoba systému po implementaci se díky systémovému návrhu velmi blíží očekávanému výsledku. Po hlubší analýze dílčích bloků v navazujících pracích však došlo k některým menším změnám, které poněkud změnilly některé vlastnosti systému. Ty budou shrnuty v této kapitole. Detaily provedení, či vlastnosti jednotlivých procesů jsou podrobněji popsány v kapitole a příloze o ověřování a testování systému.

Výsledný prototyp se skládá ze 3 částí - z uživatelského rozhraní, systémového serveru a prototypu zalévacího robotu.

UI - poskytuje plnou požadovanou funkcionalitu formou primárně mobilní, ale i webové aplikace provedené pomocí nástroje Flutter. Uživatel má možnost správy celého systému, exportu/importu potřebných dat, či štítku a je schopen sledovat pomocí výpisů stav systému. Aplikace je nad rámec základních požadavků vybavena možností správy více zařízení v několika oblastech. Jediným omezením je menší možnost přímého zásahu do harmonogramu práce.



Obr. 6.1: Podoba prototypu uživatelského rozhraní systému - přehled rostlin, informace o rostlině, poloha robotu

Server - pro komunikaci se serverem bylo provedeno vhodné rozhraní mezi systémem ROS na robotu a UI a serveru pomocí protokolu *MQTT*. Všechny data jsou přehledně strukturována a je možné je libovolným způsobem rozšiřovat.

Zalévací robot - prototyp robotu je postaven na platformě Breach, která byla zapůjčena firmou Bender Robotics. Seznam použitých komponent je uveden v tabulce 6.1. Robot je řízen z důvodu urychlení testování externě umístěným počítačem s operačním systémem Ubuntu 20.04 s ROS Noetic, ke kterému je připojen router. Interní počítač i wifi robotu byly tedy vyřazeny z činnosti a veškerý ovládací software byl přenesen na externí. K platformě je připojena zalévací platforma a kamerový modul, které byly vytvořeny v rámci projektu. Základní rozměrové parametry robotu jsou uvedeny v nákresu 12.5 a tabulce 12.1 v příloze. Příloha též obsahuje blokové schéma komponent robotu 12.6.



Obr. 6.2: Prototyp zalévacího robotu

Komponenta	Provedení	Poznámka
Řídicí počítač	Jetson Nano	GPU 128jádř. Maxwell, CPU 4jádř. ARM A57, Paměť 4 GB 64-bit LPDDR4
Router	GLinet GL-AR300M	
Lidar	RPlidar A1	
Kamera	Microsoft LifeCam HD-3000	
Kamerový mech.	Servomechanismus	jednoosý, servo SG92R
Kamerový mech.	Řízení	Arduino Uno
Zalévací mech.	Stavitelné rameno	posuvná vazba s pohyb. šroubem, otočná vazba zpřevodovaná, krok. motory
Zalévací mech.	Sensorika	Ultrazvuk. sensor HY-SRF05, koncové dotykové spínače
Zalévací mech.	Řízení	Arduino Uno
Čerpadlo	Membránové	DC 6-12V R385
Nádrž	Plastový box	objem 4 l

Tab. 6.1: Seznam komponent robotu

Spouštění, Provoz a Vypínání - spouštění systému probíhá ve dvou krocích - část je spuštěna hlavním řídicím programem a zejména převzaté funkcionality a některé parametry jsou do systému vřazeny pomocí *launch* souboru. Pro komunikaci se serverem, potažmo s UI slouží samostatný uzel.

Program automaticky přechází do fáze Zalévání rostlin. Po jeho dokončení aktualizuje data v databázi a přejde do režimu čekání na další plánovanou závlivku. Přerušování jednotlivých procesů zatím není provedeno. Při potřebě vypnutí systému řídicí program přeruší jednotlivé jím spuštěné uzly.

Pohyb k cíli - v rámci pohybu k cíli není zatím prováděna prvotní optimalizace plánování práce. Rostliny jsou zalévány dle jejich pořadí v databázi. V řídicím programu není implementován rozhodovací proces, který by rozlišil při detekci více rostlin tu potenciálně správnou. To úzce souvisí s konečným požadavkem na prostředí, ve kterém se v rámci

Proces	Stav	Poznámka
Řídicí program	provedeno	-
Server	provedeno	spojení serveru s robotem
Setup prostředí	provedeno	-
Lokalizace	provedeno	-
Sensorika a řízení pohonů	provedeno	-

Tab. 6.2: Procesy fáze - *Spouštění, Provoz a Vypínání*

předpokládaného prostoru může nacházet pouze jedna rostlina v květináči umístěném na zemi.

Proces	Stav	Poznámka
Pohyb k cíli	provedeno	-
Snímání kamerou	provedeno	pouze pohybová kamera
Detekce rostliny / zdroje vody	provedeno	pouze det. rostliny s květináčem
Určení polohy rostliny / zdr. vody	provedeno	pouze det. rostliny s květináčem
Přepočty v s.s.	provedeno	-

Tab. 6.3: Procesy fáze - *Pohyb k cíli*

Identifikace cíle - po dobu průběhu integrace nebyl dodán proces, který provádí identifikační pohyb. To má za následek silný požadavek na viditelnost identifikačního štítku při příjezdu k cíli, resp. přechodu k jeho identifikaci.

Robot byl dále vybaven pouze pohybovou kamerou, která je umístěna na jednoosém polohovacím mechanismu. To má za následek potřebu umístění štítku v přiměřené výšce na objektu.

Proces	Stav	Poznámka
Identifikační pohyb	neprovedeno	-
Detekce štítku	provedeno	jednorázová, identifikátor Aruco
Orientace kamery	provedeno	pouze jednoosý mech.

Tab. 6.4: Procesy fáze - *Identifikace cíle*

Zalévání / Čerpání vody - v průběhu rozpracovávání procesu Zalévání / čerpání vody došlo k přehodnocení požadavku na samostatné čerpání vody ze zdroje. Bylo stanoveno, že to bude provedeno pomocí dokovací stanice, která však nebude v rámci nynějšího projektu řešena. Tím pádem se objektem v celém systému stává pouze rostlina.

Výsledná podoba ramena a jeho konfigurace jasně určují výsledné rozměrové parametry rostliny. Vzhledem k prototypovému provedení ramene jsou poněkud sníženy. Zalévací platforma též neprovádí v širokém rozsahu korekci po přiblížení k rostlině, což má za následek přenesení této korekce na Pohyb zalévací / čerpací.

Proces	Stav	Poznámka
Pohyb zalévací / čerpací	provedeno	-
Zalévání / čerpání vody	provedeno	pouze čerpání vody
Detekce štítku	provedeno	průběžná, identifikátor Aruco
Přepočet dat ze štítku	provedeno	-

Tab. 6.5: Procesy fáze - *Zalévání / Čerpání vody*

6.3 Souhrn balíčků systému

Souhrn balíčků lze rozdělit na vlastní vyhotovené a převzaté balíčky. Jejich přehled je uveden v tabulkách. U každého z nich je uveden přebírající blok a imp. proces. Balíček *breach* byl převzat společně s platformou z firmy Bender Robotics.

Balíček	Blok	Proces
zaleela_main	UI / Server / Říd. program	Řídicí program
zaleela_mqtt_api	UI / Server / Říd. program	Server
yolov5	Detekce	Detekce cíle, Určení polohy cíle
aruco	Detekce	Detekce štítku, Snímání kamerou
planning	Pohyb	Veškeré pohyby
distance_matrix	Pohyb	Sestavení tabulky vzdáleností
wp_motion	Zalévání / čerp. vody	Zalévání / čerpání vody
tf_calculation	Integrace	Přepočty v s.s., Přep. dat ze štítku
camera_motion	Integrace	Orientace kamery

Tab. 6.6: Vlastní balíčky systému

Balíček	Blok	Proces
amcl [47]	Pohyb	Lokalizace
move_base [47]	Pohyb	Pohyb k cíli, Pohyb do optimální pozice
robot_state_publisher [44]	Pohyb	Popis prostředí a robotu
joint_state_publisher [43]	Pohyb	Popis prostředí a robotu
map_server [47]	Pohyb	Popis prostředí a robotu
roboclaw_node [45]	Pohyb	Sensorika a řízení pohonů
breach	Pohyb	Sensorika a řízení pohonů
rplidar_ros [42]	Pohyb	Sensorika a řízení pohonů
gmapping [46]	Integrace	Popis prostředí a robotu

Tab. 6.7: Převzaté balíčky systému

Kap. 7

Ověřování systému

Po provedení postupné implementace jednotlivých dílčích částí a jejich integrace do jednoho celku bylo možné přistoupit k ověřování funkčnosti systému, potažmo k jeho testování.

Vzhledem k rozsáhlosti řešeného projektu bylo zvoleno schéma několikařázkového ověřování, které se skládá ze simulace, ověřování na menším robotu a ověřování na prototypu. Tyto fáze se v čase poněkud překrývaly, z tohoto důvodu, i důvodu časového už proto nebylo nutné, či možné provádět plné otestování některých částí zejména v simulaci. Důraz byl v konečné fázi kladen zejména na prototyp systému.

Postup ověřování v jednotlivých fázích probíhal obvyklým způsobem - tj. v počátku byla ověřována dílčí část samostatně, posléze k ní byly připojeny další bloky, které s ní nějakým způsobem interagují, po tomto bylo možné přistoupit k zapojení celé funkcionality do systému.

Byl proveden soupis všech testovacích scénářů 12.2, které byly postupně procházeny. Jejich důkladnější popis a průběh je uveden v příloze práce 12.3. Tyto scénáře byly odvozeny na základě výsledné implementace dílčích celků - tj. i výsledných možností systému. Cíle ověřování a testování lze rozdělit na několik druhů - (F) ověřování funkčnosti a správné reakce v systému, (P) ověřování přesnosti, či rychlosti a (S) stanovení spolehlivosti.

Ověřování funkčnosti do jisté míry souvisí s integrací. Bylo provedeno prakticky u každého vřazeného procesu. Provádění ověřování přesnosti a úspěšnosti naopak úzce souvisí s fází implementační. Na základě zjištěných nedokonalostí, či problémů, bylo možné díky ověřování přeformulovat jednotlivé požadavky na procesy. Z toho je zřejmé, tak jako u jiných fází, že celý proces testování lze chápat jako soubor velkého počtu iterací, které mají za úkol zvýšit přesnost a spolehlivost jednotlivých procesů.

7.1 Simulační ověřování

Pro potřebu prvotního ověřování systému byla v rámci této práce provedena simulace systému za využití nástroje Gazebo Classic verze 11.

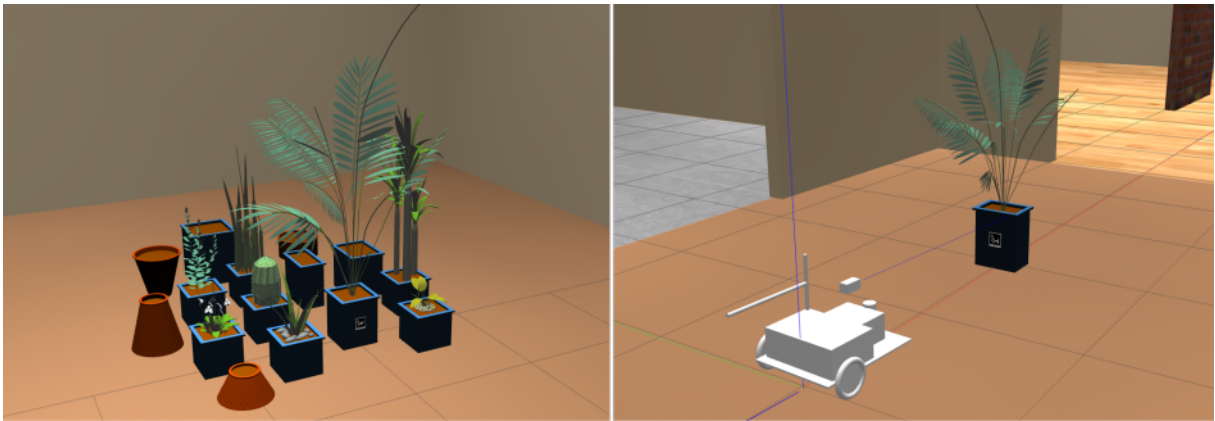
Gazebo je opensource simulační nástroj. Jeho historie sahá až do roku 2002, kdy byl zahájen jeho vývoj na University of Southern California Andrew Howardem a Nate Koenigem. Po pár letech se tento software dostal pod správu Open Source Robotics Foundation, kde jeho vývoj probíhá dodnes [49]. Gazebo umožňuje simulaci rob. systémů i okolního prostředí. Využívá grafický engine OGRE pro vizualizaci. Podporuje fyzikální enginy ODE, Bullet, DART, či Simbody. Uživatel má možnost jednoduché správy simulace pomocí konfiguračních souborů, či pomocných pluginů. Vzhledem k rozšířenosti a

oficiální podpoře ze strany ROSu je v něm k dispozici velké množství hotových pluginů, užívaných k simulaci sensoriky, či řízení apod..

Hlavním cílem simulace je ověřování funkčnosti jednotlivých dílčích celků a účelnosti jejich rozhraní. Vzhledem k podobě simulace bylo možné ověřit reakce prakticky všech modulů na příchozí simulovaná data ze sensorů a ověřit i správnost řídicích pokynů. V neposlední řadě bylo možné ověřit funkčnost spojení celku řídicím programem.

7.1.1 Provedení simulace

V rámci simulace byly namodelovány všechny části prostředí i systému. Simulace tak zahrnuje zjednodušený model robotu Breach, rostlinu i s jejími náležitostmi, sim. zdroj vody i namodelované vnitřní prostředí. Jako fyzikální engine byl použit ODE.



Obr. 7.1: Podoba objektů v simulaci - rostliny, květináče, resp. zdroje vody, model robotu s rostlinou se štítkem

Parametrický model robotu je definován pomocí maker formátu URDF. Makra obsahují jednotlivé části robotu. Robot je modelovaný jako diferenciální typ s jedním pojezdovým kolem. Do vazeb kol jsou umístěny simulované pohony, které jsou ovládány pluginem *libgazebo_ros_diff_drive*, který simuluje i odometrii, která je odesílána přímo v transformační zprávě. Na robotu je namodelováno též jednoduché dvouosé řízení zalévacího ramena, které je ovládáno pluginem *ros_control*. Model robotu též obsahuje simulovaný lidar, který je převzat z pluginu *libgazebo_ros_laser*, sonary z *libgazebo_ros_range* a kameru.

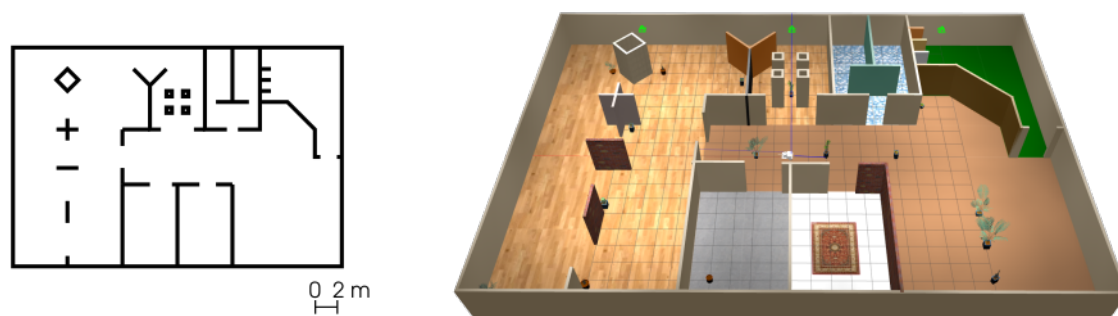
Prostředí simulace je definováno v XML souboru typu SDF. Objekty prostředí - rostliny, zdroje vody a světla jsou automaticky generovány dle konfigurovatelného tabulkového souboru *csv* pomocí skriptu v Pythonu. Výsledkem tohoto skriptu je část souboru SDF, kde jsou nadefinovány jednotlivé požadované položky. To bylo provedeno pro potřebu jednoduché správy celé simulace.

Model rostlin se skládá ze 3 částí - modelu květináče, štítku a meshe rostliny. Byly provedeny 2 modely květináče - kulatý a obdelníkovitý, které jsou parametrizovány a pomocí maker v programu FreeCad automaticky generovány pro požadované velikosti. Model štítku je automaticky spojen s texturou, na které je vyobrazen daný Aruco identifikátor. Meshe rostlin byly staženy z volně dostupného zdroje [48]. V simulaci je tedy namodelováno celkem 8 typů rostlin. Měřítka meshe rostliny je dle zvoleného květináče automaticky přizpůsobeno. Všechny tyto tři části jsou automaticky spojeny do jednoho objektu v rámci SDF souboru prostředí. Model zdroje vody vychází z modelu rostliny, neobsahuje jen její mesh.

Simulační prostor je modelován jako vnitřní prostředí, které by mělo obsahovat variaci všech možných konfigurací stěn, či vchodů.

7.1.2 Provedení testování

Pro potřeby testování byla vytvořena simulovaná kancelář o rozměrech $30 \times 20 \text{ m}$, která obsahuje 28 rostlin a 2 zdroje vody se všemi jejich náležitostmi. Její obrázek i mapu lze vidět na obrázku 7.2. Všechny tyto rostliny jsou pro potřebu testování zaneseny i v databázi systému. Díky komplexnímu modelu tak bylo možné ověřit v určitém rozsahu prakticky všechny procesy v systému.



Obr. 7.2: Mapa s grafickým měřítkem a podoba simulované kanceláře

Nejdůležitější se stala simulace pro blok **Pohybu**, který díky ní mohl prvotně ověřit a konfigurovat, jak převzaté moduly lokalizace, či plánování, tak vlastní vyhotovený plánovač, či jednotlivé pohyby. Provádění testování pohybu vedlo k vytvoření specifických *launch* souborů pro každou fázi testování. Ty obsahují konkrétní parametry pro jednotlivé moduly a spouští některé, zejména převzaté funkcionality.

U bloku **Detekce** byly testovány díky simulované kalibrované kameře prakticky všechny požadované procesy. Pro účely detekce rostliny byl vytvořen tréninkový dataset, který posléze sloužil jak podklad pro trénování neuronové sítě yolov5. Díky namodelovaným štítkům bylo možné ověřit funkčnost jejich detekce. V neposlední řadě by bylo možné díky namodelovanému lidarů ověřit i funkčnost lokalizace det. rostlin, ale to už nebylo z časových důvodů provedeno.

U bloku **Zálévání / čerpání vody** byla ověřena pouze reakce procesu na pokyny od řídicího programu.

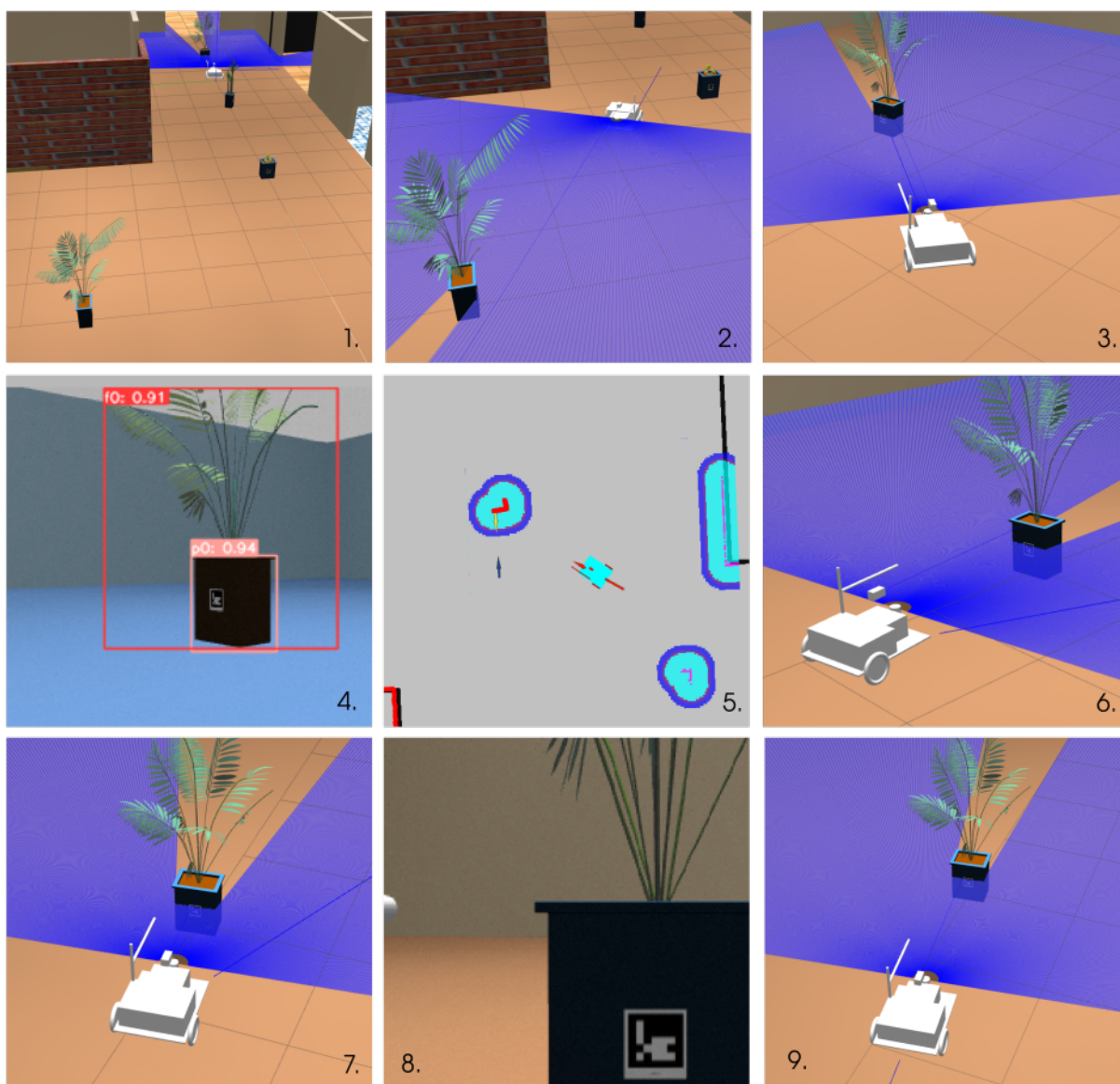
Integrační blok provedl testování všech požadovaných procesů. Díky simulovanému kamerovému mechanismu bylo možné testovat jeho reakce na pokyny od řídicího programu i na aktuální stav systému - polohu robotu atd..

U **Řídicího programu** bylo testováno správné provedení všech požadovaných kroků a jeho reakce na různé kombinace odpovědí řízených procesů. V neposlední řadě byla ověřena i komunikace se systémovým serverem a UI.

Na příložených obrázcích 7.3 lze vidět ukázkou testovací jízdy robotu k rostlině. Zobrazuje jednotlivé fáze řídicího programu.

7.1.3 Výsledky testování

Výsledný stav řešení před započítím dalšího testování lze shrnout tak, že bylo zjištěno, že všechny připravené simulovatelné moduly reagují adekvátním, či téměř požadovaným

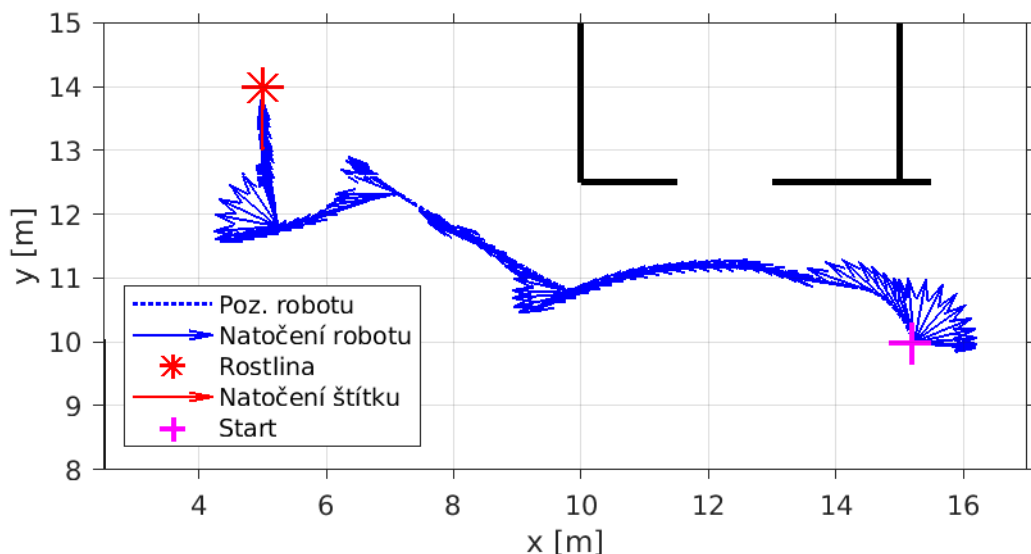


Obr. 7.3: Příklad testovací jízdy v simulaci - 1. zahájení pohybu, 2. zahájení detekce rostliny, 3. detekce rostliny, 4. záznam z detekce, párování rostlina-květináč, 5. detekce pozice a orientace štítku (žlutá šipka) a odhad optimální pozice (modrá šipka), 6. dosažení optimální pozice, 7. dojezd k rostlině, 8. záznam z kamery po příjezdu, 9. pozice robotu po couvnutí

způsobem.

Pro další fáze testování bylo stěžejní zjištění stavu pohybu, zejména těch částí, které byly nuceny z důvodu implementování pro integraci využívat vestavěný plánovač *move_base* systému ROS. Ukázalo se, že pro účely projektu má některé špatné vlastnosti, které se týkají zejména globálního plánování pohybu - kdy robot neuskuteční ani pokus o jízdu do určené polohy, ve které detekuje překážku nebo též špatně proveditelné kontroly výsledku prováděné jízdy - např. kontrola dojezdu, správného natočení. Tyto vlastnosti pohybů bohužel provázejí systém při všech fázích testování.

V rámci řídicího programu též není implementována část, která se týká rozhodovacího procesu při detekci rostlin. To úzce souvisí s detekcí rostlin, která vrací při více detekovaných objektech pouze jeden výsledek. To má za následek požadavek pouze jedné



Obr. 7.4: Záznam jízdy robotu v prostoru

detekovatelné rostliny v pracovním prostoru robotu, tj. v blízkosti při dojezdu k cíli.

7.2 Ověřování na robotu Leela

Po provedení simulačního ověřování alespoň základních procesů mohlo dojít k další fázi testování na reálném menším robotu.

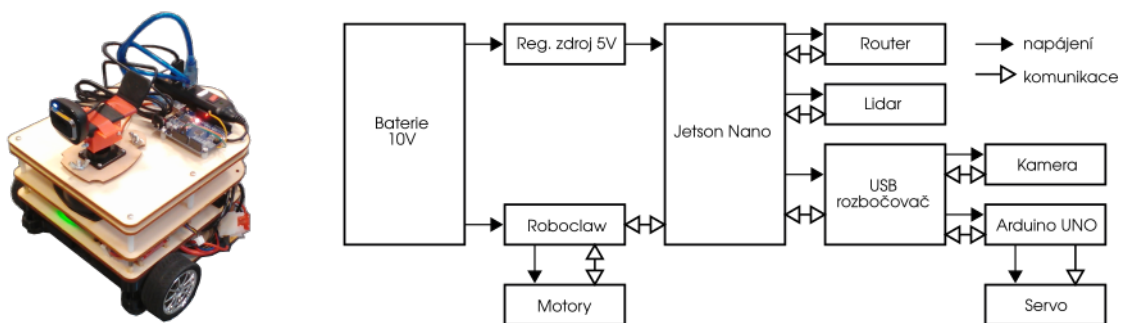
Cílem této fáze je potvrzení funkčnosti již testovaných modulů v reálném prostředí, konfigurace jednotlivých procesů pro reálný svět, potvrzení funkčnosti zatím netestovaných modulů, ověřování životaschopnosti navrhovaných algoritmů v reálném světě a v neposlední řadě i testování celkového chování.

7.2.1 Popis robotu

Pro testování byl využit menší robot Leela, který se svou stavbou velice blíží k platformě Breach. Jeho podobu lze vidět na přiloženém obrázku.

Jedná o robot s diferenciálním typem podvozku, který je poháněn dvěma EC motory s mag. enkodéry. Řízení robotu provádí počítač typu Jetson Nano. K tomu je připojen malý wifi router typu GLinet N300. Robot je vybaven lidarem typu RPLiDAR A1. Řízení motorů probíhá pomocí ovladače Roboclaw, se kterým systém komunikuje přes vlastní uzel. Motory jsou vybaveny enkodéry, jejichž signál Roboclaw převádí přímo na data z odometrie. Pro tu však bylo nutné vyhotovit vlastní script, který ji převádí do vhodného transformačního tvaru. Pohony jsou napájeny přímo z baterie, řídicí počítač je k ní připojen přes regulátor napětí.

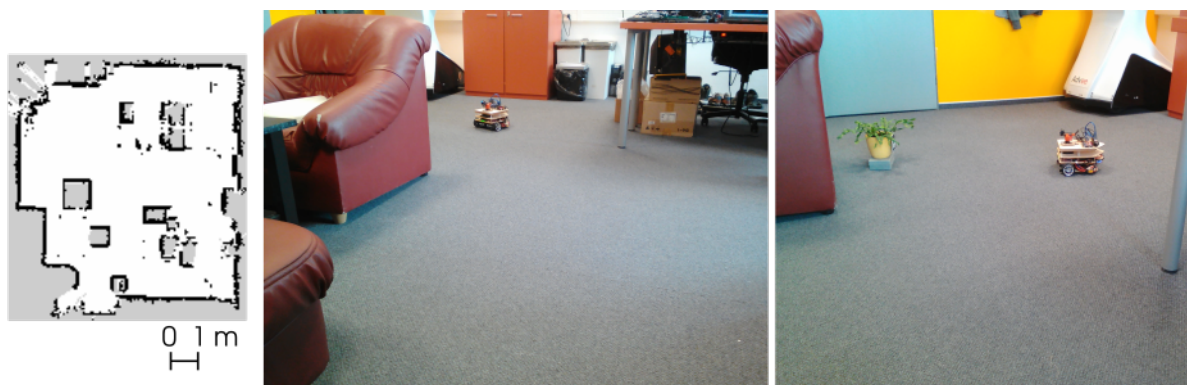
Pro potřeby systému byl robot dovybaven rozbočovačem USB a kamerovým modulem, který má podobu otočného mechanismu s modelářským servem. Kamera je typu Microsoft LifeCam HD-3000.



Obr. 7.5: Robot Leela a jeho blokové schéma komponent

7.2.2 Provedení testování

Testování probíhalo v kanceláři, jejíž mapu a podobu lze vidět na přiloženém obrázku 7.6. Před zahájením testování musel být prostor namapován a být zjištěna a zaznamenána poloha testovací rostliny s identifikačním štítkem.



Obr. 7.6: Mapa prostoru s grafickým měřítkem a jeho podoba

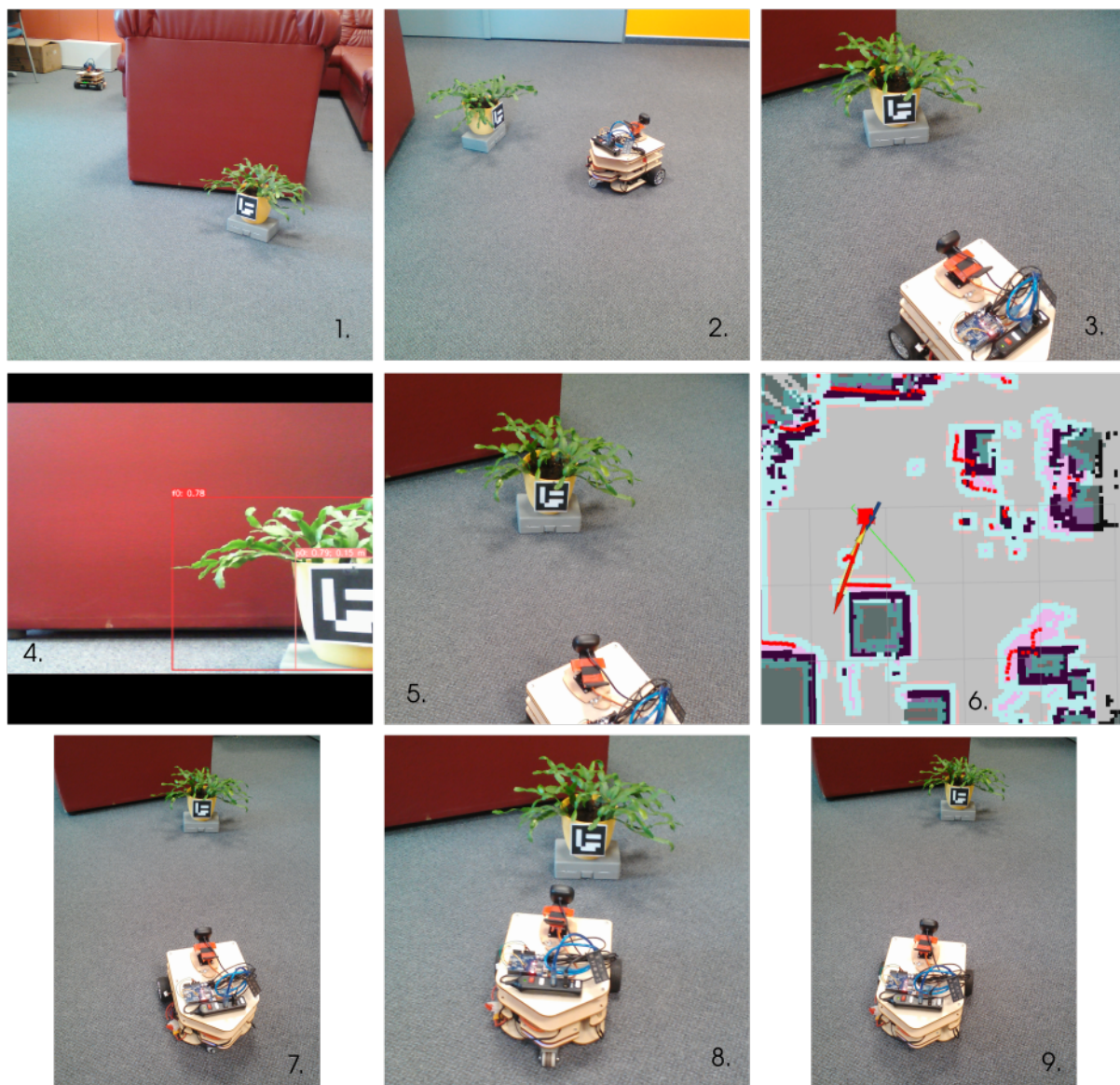
Testování začalo konfigurací a odlaďováním reálných modulů robotu - lidarů, řízení motorů, kamerového modulu a kamery. Potom mohlo být přistoupeno ke konfiguraci prostředí, testování převzatých modulů pohybu - lokalizace a plánování pohybu a v neposlední řadě zprovoznění vlastních dílčích procesů.

Hlavním cílem bylo otestování navržených algoritmů pro 3 hlavní fáze zalévání rostlin. Záznam z testovací jízdy lze vidět na přiložených obrázcích 7.7. Fáze detekce rostliny byla testována bez větších problémů vyjma pohybu, jehož nedostatky byly zjištěny v předchozí fázi.

Fáze identifikace cíle byla otestována pouze z části. Z důvodu nedodání implementace identifikačního pohybu, bylo nutné natáčet štítek na rostlině vždy ve směru viditelném ze směru příjezdu robotu. To bohužel přetrvává i v další fázi testování. Zbylé podpůrné procesy této fáze, zahrnující obsluhu kamerového modulu, detekci štítku, či přepočty souřadnic byly otestovány bez větších problémů.

V poslední fázi Zalévání / čerpání vody byly otestovány veškeré pohyby a jejich reakce na příchozí data z detekce. Příjezd do optimální pozice je prováděn s nastavenou přesností vestavěného plánovače. Zkoušení průběžné detekce štítku ukázala nutnost filtrace polohy a orientace štítku. Příjezd k rostlině je nakonec řešen prostým dvoustavovým regulátorem, který reguluje pomocí úhlu natočení robotu od štítku. Ukončován je vždy v určité

vzdálenosti štítku od robotu. V této fázi tedy musel být štítek po celou dobu příjezdu viditelný. Zbylé couvání robotu probíhalo bez problémů.

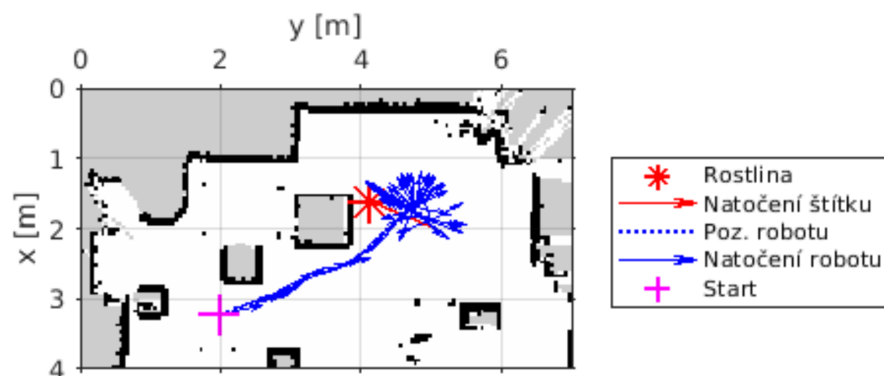


Obr. 7.7: Příklad testovací jízdy s robotem Leela - 1. zahájení pohybu k rostlině, 2. dojezd k cíli a zahájení otáčení, 3. detekce rostliny, 4. záznam z detekce rostliny, 5. natočení kamery k detekovanému cíli, 6. detekce pozice a orientace štítku (žlutá šipka) a odhad optimální pozice (modrá šipka), 7. pozice robotu po dosažení opt. pozice, 8. příjezd k rostlině, 9. pozice robotu po couvnutí

7.2.3 Výsledky testování

Pro provedení této fáze testování je možné říci, že navržený přístup k řešení požadované funkcionality systému lze považovat za správný. Odladění jednotlivých procesů na reálném robotu ukázalo životaschopnost navržených algoritmů.

Hlavním přínosem této fáze testování bylo výrazné zjednodušení prvotní konfigurace systému pro reálný svět. Užité reálné moduly byly posléze převzaty a umístěny přímo na prototyp robotu, tj. včetně řídicího počítače. Tím bylo výrazně zjednodušeno též testování



Obr. 7.8: Záznam jízdy robotu v prostoru

systému.

Testování ukázalo některé specifické vlastnosti systému, které nejsou pro navržené řešení úplně vhodné - pomineme-li vlastnosti pohybu, pak je to zejména obnovovací frekvence systémových transformací, která je poněkud nízká. To má za následek pomalé provádění přepočtů souřadnic. Příčina tohoto problému byla identifikována až v další fázi testování. Byla jím zejména obnovovací frekvence dat z lidarů a odometrie.

Dalším problémem bylo i zjištění omezených možností využitého počítače Jetson Nano, kdy zejména při současně spuštěné detekci rostlin a určování jejich polohy byla jeho operační paměť prakticky plně vytížena.

7.3 Ověřování prototypu na robotu Breach

Po ověření životaschopnosti systému v reálných podmínkách bylo možné přistoupit ke konečnému testování prototypu systému postavenému na platformě Breach, který byl detailněji popsán v kapitole Integrace systému.

Hlavním cílem této fáze testování byla konečná konfigurace systému, provedení testování funkčnosti zbylých modulů a samozřejmě ověření splnění cíle práce - provedení úspěšného zalévání rostlin.

7.3.1 Provedení testování

Testování probíhalo na chodbě, jejíž podobu lze vidět na obrázku 7.9. Pro potřeby systému byly do databáze zaneseny 3 rostliny, které byly na základě předchozích závěrů testování vhodně umístěny.



Obr. 7.9: Mapa prostoru s grafickým měřítkem a jeho podoba

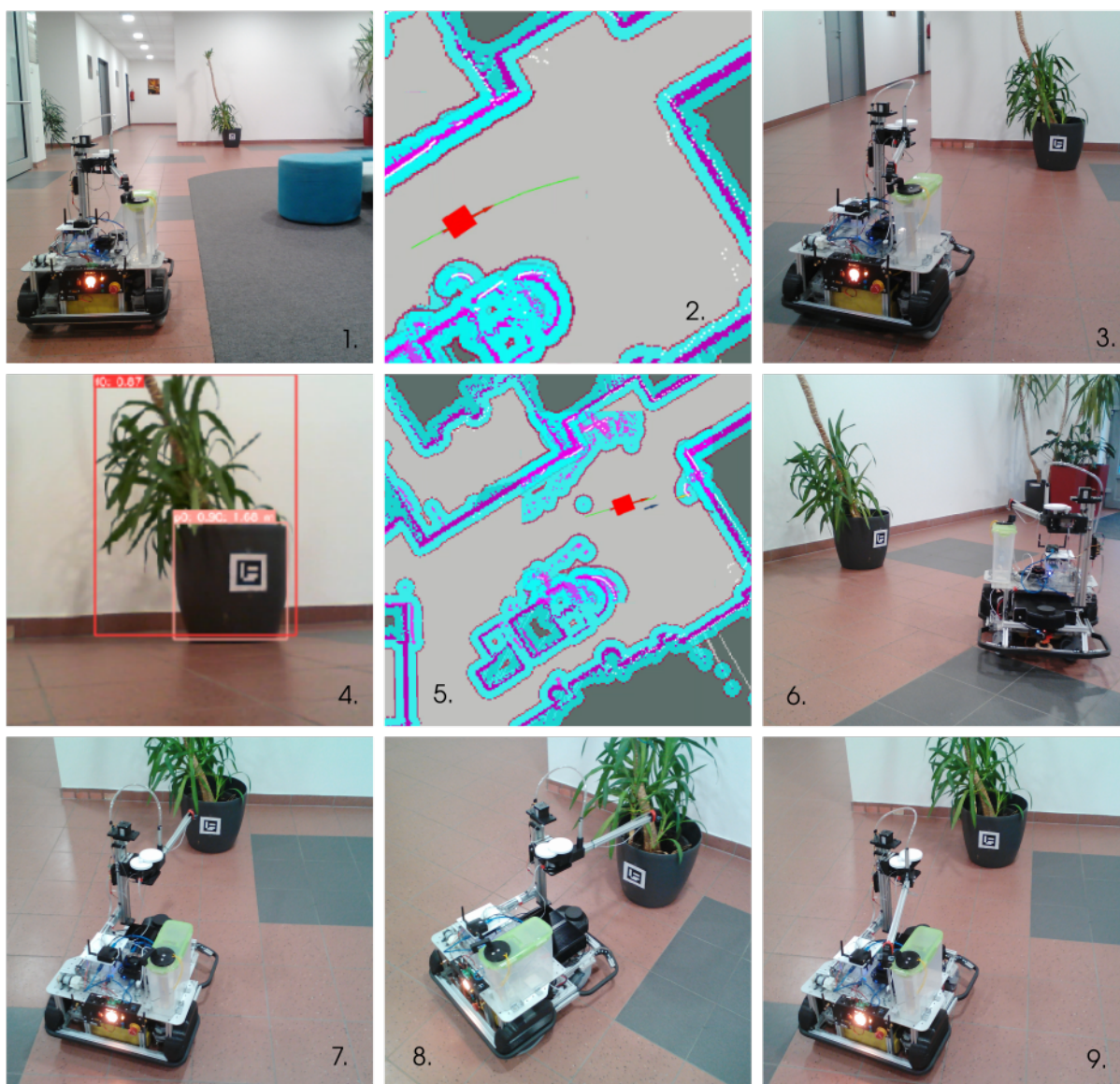
Hlavním odladovaným procesem bylo Zalévání / čerpání vody. Na prototyp byla umístěna vyhotovená zalévací platforma, která byla propojena s řídicím počítačem. Po provedení otestování její funkčnosti bylo možné přistoupit opět ke konfiguraci prostředí, řízení pohonů a úpravy dat z lidarů. Na základě zjištěných problémů s transformacemi v systému byla uměle zvýšena obnovovací frekvence odometrie a lidarů.

Po odladění dílčích modulů bylo přistoupeno k závěrečným jízdám, u kterých byla stanovována u vybraných procesů přesnost a celková úspěšnost provedení funkcionality. Záznam z provedené jízdy lze vidět na příložených obrázcích 7.10.

7.3.2 Výsledky testování

Tato fáze testování prokázala životaschopnost celého systému. Byly v ní otestovány všechny moduly, se kterými jednotlivé procesy v systému pracují. Přes drobné nedostatky, či nedodělky v některých modulech a výrazně prototypovou podobu pohybu, systém v testovacím prostředí pracuje s poměrně dobrou spolehlivostí. Tu lze přibližně odhadnout s pomocí tabulky provedených závěrečných testů 7.1.

Prototypové provedení zalévací platformy ukázalo některé výhody i nedořešené nedostatky. Z pohledu systémového návrhu se jedná zejména o nedostatečné provedení vlastní korekce zalévacího ramena v procesu zalévání. Z tohoto bloku, vzhledem k provedení mechanismu ramena, vyvstal požadavek na provedení korekce s pomocí procesu pohybu zalévacího / čerpacího ve větší míře, než bylo původně předpokládáno. To má zatím za



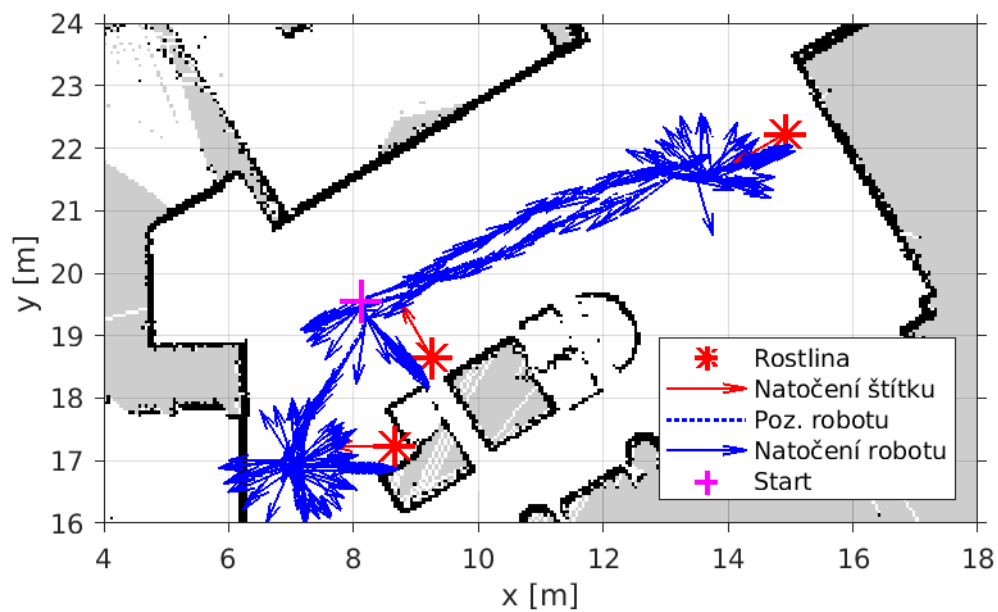
Obr. 7.10: Příklad testovací jízdy s prototypem - 1. zahájení pohybu k cíli, 2. plánování pohybu k cíli, 3. detekce rostliny, 4. záznam z detekce rostliny, párování rostlina-květináč, 5. detekce pozice a orientace štítku (žlutá šipka) a odhad optimální pozice (modrá šipka), 6. přesun do optimální pozice, 7. dosažení optimální pozice a nastavení ramena do pracovní pozice, 8. příjezd k rostlině a provedení její zálivky, 9. couvnutí robotu a nastavení ramena do přepravní pozice.

Test	21.3.2022	24.3.2022	27.3.2022	29.3.2022	12.4.2022	27.4.2022	Celkově
Celkově	8	13	10	5	15	6	57
Úspěšný	2	10	9	5	12	4	42
Neúspěšný	6	3	1	0	3	2	15
Úspěšnost %	25	76.9	90	100	80	66.7	73.7

Tab. 7.1: Tabulka úspěšnosti provedených testů celku

následek nedostatečně robustně provedenou korekci. To úzce souvisí s malým množstvím sensoriky, která je umístěna na rameni.

Systém též neumí v dostatečné míře reagovat na poruchy, které se mohou projevit v



Obr. 7.11: Záznam jízdy robotu v prostoru

různých procesech. V dalších fázích je tedy nutné jej vybavit reakcemi na různé příchozí exitkódy z dílčích procesů, které nejsou funkční.

Závěr

Tato práce se věnovala návrhu a realizaci mobilního zalévacího robotu.

Prvotním úkolem této práce bylo zhodnocení jeho možného uplatnění. Na základě rešerše stavu moderních závlahových systémů bylo možné identifikovat jejich obecnou strukturu. Byly zhodnoceny charakteristické vlastnosti a silné, či slabé stránky jejich technického provedení. Rešerše ukázala, že existují na tomto poli i některé experimentální robotické systémy. Na základě toho bylo možné stanovit eventuálně výhodné oblasti využití.

Byly identifikovány dvě oblasti možného nasazení robotického závlahového systému. Jedná se o využití ve vnitřních prostorech, které nejsou určeny primárně pro pěstování rostlin a ve venkovních prostorech, kde je potřeba provedení závlahy individuální zeleně. Vnitřní výhodné určení bylo podpořeno jednoduchým ilustračním příkladem, který ukazuje těžkopádnost řešení závlahy rostlin uvnitř kancelářské budovy klasickým stacionárním systémem.

Na základě stanovení existence optimální oblasti využití bylo přistoupeno k úkolu hlavnímu - návrhu zalévacího robotu. Na základě dostupných komponent bylo rozhodnuto, že cílovým prostředím budou vnitřní prostory a robot bude využívat dostupnou robotickou platformu Breach od firmy Bender Robotics. Dle závěru rešerše byly stanoveny obecné cíle směřování návrhu a posléze i konkrétní specifikace.

Na základě jejich rozpracování bylo vyhotoveno ideové řešení systému, ve kterém byly sledovány obecné cíle práce. Díky navržené funkcionalitě - *use case*, bylo možné provést rozdělení systému na bloky, které zahrnují hlavní technické části. Celý systém byl řešen projektově v malém kolektivu, který zahrnoval celkem 5 lidí. V této fázi bylo tedy nutné dekomponovat systém i organizačně a stanovit pole působnosti jednotlivých řešitelů.

Ideový návrh bylo poté možné důkladněji propracovat ve formě konkrétního návrhu pro platformu Breach, který specifikuje systémový návrh vnitřní struktury, která je rozdělena na procesy, mezi kterými je jasně vydefinováno rozhraní. Jeho hlavní část byla vypracována na bázi systému ROS, který je k tomuto účelu více než vhodný. Na základě tohoto návrhu bylo možné zahájit analýzu, návrh a konečnou implementaci dílčích procesů, které už byly prováděny jednotlivými organizačními bloky.

Díky specifikování jasného systémového návrhu bylo možné přistoupit k integraci dílčích vyhotovených částí - tj. spojení jednotlivých vyhotovených procesů. Výsledkem integrace je prototyp systému, který se skládá z uživatelského rozhraní, systémového serveru a prototypu zalévacího robotu.

V průběhu integrace byl prováděn i poslední vytyčený úkol této práce - celkové otestování systému a zhodnocení jeho možností, či vlastností. Testování probíhalo z důvodu obsáhlosti projektu několikafázově. Pro potřeby prvotního ověření byla v rámci této práce

vyhotovena simulace, která umožnila komplexní otestování dílčích procesů systému. Poté bylo přistoupeno k testování na menším robotu Leela. Posledním krokem ověřování systému bylo testování prototypu zalévacího robotu na cílové platformě Breach, který již obsahoval všechny vyhotovené moduly a procesy.

Testování ukázalo, že navržený a implementovaný systém dokáže splnit, byť v zatím omezené míře stanovený úkol - dokáže zalévat rostliny ve vnitřních prostorách budovy. Prototyp bohužel nedisponuje některými potřebnými funkcionalitami, ale díky modulárnímu návrhu, není problém v další fázi vývoje systém dále vylepšovat.

Vylepšení se musí týkat hlavně oblasti rozhodování robotu, kdy by měl být promyšlen rozhodovací proces, kterým by systém robustně vyhodnotil např. cílovou rostlinu. To silně souvisí i s potřebnou implementací fúze dat, která by umožnila zvýšit i při nynějším vybavením spolehlivost systému v některých fázích činnosti.

Závěrem lze tedy říci, že cíl práce byl i přes některá úskalí splněn. Byl nejen navrhnut, ale i realizován komplexní robotický systém, který plní zadaný úkol - zalévání rostlin ve vnitřních prostorách budov.

Část 9

Seznam použitých zdrojů

- [1] BENETÍN, Ján, Jiří FÍDLER, Anton HRABAL a Štefan RAUČINA. *Závlahy*. Bratislava: Příroda, 1979.
- [2] EISENHAUER, Dean E. Eisenhauer, Derrel L. MARTIN a Glenn J. Hoffman HOFFMAN, HEEREN, Derek M. Heeren, ed. *Irrigation Systems Management* [online]. American Society of Agricultural and Biological Engineers, 2021 [cit. 2022-04-01]. ISBN 978-1-940956-42-8. Dostupné z: <https://elibrary.asabe.org/textbook.asp?confid=ism2021>
- [3] KOZLOVSKY DUFKOVÁ, Jana. *Závlahy a odvodnění: teoretické základy a praktická cvičení*. V Brně: Mendelova zemědělská a lesnická univerzita, 2009. ISBN 978-80-7375-335-1.
- [4] MAROÚŠEK, Jan. *Zavlažování*. Brno: ERA, 2008. Stavíme. ISBN 978-80-7366-119-9.
- [5] CORRELL, Nikolaus, Nikos ARECHIGA, Adrienne BOLGER, et al. Indoor robot gardening: design and implementation. *Intelligent service robotics* [online]. Berlin/Heidelberg: Springer-Verlag, 2010, **3**(4), 219-232 [cit. 2022-04-01]. ISSN 1861-2776. Dostupné z: doi:10.1007/s11370-010-0076-1
- [6] MECHSY, L. S. R., M.U.B. DIAS, W. PRAGITHMUKAR a A.L. KULASEKERA. A Mobile Robot Based Watering System for Smart Lawn Maintenance. In: *2017 17th International Conference on Control, Automation and Systems (ICCAS)*. IEEE, 2017, s. 7. ISBN 978-89-93215-14-4. Dostupné z: doi:10.23919/ICCAS.2017.8204233
- [7] SIKORSKI, Kevin. *A Robotic PlantCare System*. 2003. Dostupné také z: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.4.2980&rep=rep1&type=pdf>. University of Washington.
- [8] Tech-Savvy Inventor Builds Robotic Waterer. *Farm Show magazine*. 2013, **2013**(37), 1. Dostupné také z: https://www.farmshow.com/a_article.php?submit_search=1&keywords1=aquarius&aid=27210&page_num=1
- [9] Aquarius robot. *Dorhout R&D LLC* [online]. 2021 [cit. 2022-04-01]. Dostupné z: http://dorhoutrd.com/aquarius_robot
- [10] LOGHIN, Dumi. The Irrigator: AI-powered irrigation robot. *Hackster.io* [online]. 2020 [cit. 2022-04-03]. Dostupné z: <https://www.hackster.io/dumiloghlin/the-irrigator-ai-powered-irrigation-robot-80f1d5>

- [11] ARAVIND, R, M DAMAN a B S KARIYAPPA. Design and development of automatic weed detection and smart herbicide sprayer robot. In: *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*. IEEE, 2015, 2015, s. 257-261. ISBN 978-1-4673-6670-0. Dostupné z: doi:10.1109/RAICS.2015.7488424
- [12] Plant Irrigation Water Sprinkler Agriculture Robot. *NevonProject* [online]. 2022 [cit. 2022-04-03]. Dostupné z: <https://nevonprojects.com/plant-irrigation-water-sprinkler-robot/>
- [13] CHEN, Minghan, Yilong SUN, Boyi LIU, et al. Design and Implementation of A Novel Internet of Things Irrigation System With A Precision Irrigation Robot. In: *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, 2021, 2021-7-15, s. 1008-1015. ISBN 978-1-6654-3678-6. Dostupné z: doi:10.1109/RCAR52367.2021.9517587
- [14] HASSAN, Ahmed, Rao M. ASIF, Ateeq Ur REHMAN, Zuhaib NISHTAR, Mohammed K. A. KAABAR a Khan AFSAR. Design and development of an Irrigation Mobile Robot. *IAES International Journal of Robotics and Automation (IJRA)*. 2021, **10**(2), 75-90. ISSN 2722-2586. Dostupné z: doi:10.11591/ijra.v10i2.pp75-90
- [15] V, Nandeesh T a H. M KALPANA. Smart Multipurpose Agricultural Robot. In: *2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*. IEEE, 2021, 2021-7-9, s. 1-6. ISBN 978-1-6654-2849-1. Dostupné z: doi:10.1109/CONECCT52877.2021.9622632
- [16] SOWJANYA, K Durga, R SINDHU, M PARIJATHAM, K SRIKANTH a P BHARGAV. Multipurpose autonomous agricultural robot. In: *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, 2017, 2017, s. 696-699. ISBN 978-1-5090-5685-9. Dostupné z: doi:10.1109/ICECA.2017.8212756
- [17] RAFI, Rabiul Hossen, Shuva DAS, Nawsher AHMED, Iftekhar HOSSAIN a S. M TASLIM REZA. Design & implementation of a line following robot for irrigation based application. In: *2016 19th International Conference on Computer and Information Technology (ICCIT)* [online]. IEEE, 2016, s. 480-483 [cit. 2022-04-01]. Dostupné z: doi:10.1109/ICCITECHN.2016.7860245
- [18] NAGARAJA, Hema, Reema ASWANI a Monisha MALIK. Plant Watering Autonomous Mobile Robot. *IAES international journal of robotics and automation* [online]. 2012, **1**(3) [cit. 2022-04-01]. ISSN 2089-4856. Dostupné z: doi:10.11591/ijra.v1i3.1305
- [19] NetBeat. *Netcraft* [online]. Netcraft, 2022 [cit. 2022-05-03]. Dostupné z: <https://netcraft.co.il/project/project-4/>
- [20] Evolution AG Series Controller: Irrigation and Fertigation Controller For Agriculture & Horticulture. *TORO* [online]. 2022 [cit. 2022-05-03]. Dostupné z: <https://www.toro.com/en/agriculture/irrigation-controllers/evolution-ag-series-controller>
- [21] Spirit Pro Hybrid. *NAANDANJAIN* [online]. [cit. 2022-05-03]. Dostupné z: <https://naandanjain.com/products/spirit-pro-hybrid/>

- [22] Spirit Pro Irrigation. *NAANDANJAIN: A Jain Irrigation Company* [online]. [cit. 2022-05-03]. Dostupné z: <https://naandanjain.com/products/spirit-pro-irrigation/>
- [23] KOMEN, Jeroen. *Irrigation in Idaho, USA* [online]. 2010-7-16 [cit. 2022-05-03]. Dostupné z: [https://commons.wikimedia.org/wiki/File:Irrigation_in_Idaho,_USA_\(9429674404\).jpg](https://commons.wikimedia.org/wiki/File:Irrigation_in_Idaho,_USA_(9429674404).jpg)
- [24] Falcon boom. *ABI Irrigation* [online]. [cit. 2022-05-03]. Dostupné z: <https://abi-irrigation.com/product/falcon-irrigation-booms/>
- [25] VRÁNA, Jakub. Vliv místních odporů na tlakové ztráty v potrubí. *Tzb info* [online]. 2012-4-23 [cit. 2022-04-05]. ISSN 1801-4399. Dostupné z: <https://voda.tzb-info.cz/teorie-voda-kanalizace/8514-vliv-mistnich-odporu-na-tlakove-ztraty-v-potrubi>
- [26] *Mezinárodní katalog výrobků pro zavlažovací systémy*. Rain Bird Corporation, 2020. Dostupné také z: https://www.rainbird.com/sites/default/files/media/documents/2020-05/2020_INTL_Turf_Catalog_CZK_web.pdf
- [27] *Doltak eshop* [online]. Doltak, 2020 [cit. 2022-04-05]. Dostupné z: <https://www.doltak.cz>
- [28] *DEK eshop* [online]. DEK, 2022 [cit. 2022-04-05]. Dostupné z: <https://www.dek.cz/>
- [29] *Tabulky tlakových ztrát*. WAVIN Ekoplastik, 2015. Dostupné také z: <https://www.wavinekoplastik.com/cz/tabulky-tlakovych-ztrat>
- [30] MILES, Russ a Kim HAMILTON. *Learning UML 2.0*. Sebastopol: O'Reilly, 2006, 269 s. ISBN 0-596-00982-8.
- [31] BAHILL, A. Terry a Rick BOTTA. Fundamental Principles of Good System Design. *Engineering Management Journal*. 2015, **20**(4), 9-17. ISSN 1042-9247. Dostupné z: doi:10.1080/10429247.2008.11431783
- [32] DENNIS, Alan, Barbara Haley WIXOM a Roberta M. ROTH. *Systems Analysis and Design*. Fifth edition. Hoboken: Wiley, 2012. ISBN 978-1-118-05762-9.
- [33] SLADKÝ, Jiří. Detekce a klasifikace objektů zájmu zalévacího robotu zpracováním obrazu [online]. Brno, 2022 [cit. 2022-05-05]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/140181>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojíního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Jiří Krejsa.
- [34] BAJER, Jan. Návrh a realizace komunikačního rozhraní autonomního mobilního robotu [online]. Brno, 2022 [cit. 2022-05-05]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/140227>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojíního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Stanislav Věchet.
- [35] DOSEDĚL, Miroslav. Návrh a realizace strategie plánování pohybu mobilního robotu [online]. Brno, 2022 [cit. 2022-05-04]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/140228>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojíního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Stanislav Věchet.

- [36] VIZVÁRY, Peter. Návrh zalévacího modulu pro mobilní robot [online]. Brno, 2022 [cit. 2022-05-05]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/140184>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Jiří Krejsa.
- [37] Breach: mobilní robotická platforma. *Bender Robotics* [online]. [cit. 2022-05-05]. Dostupné z: <https://www.benderrobotics.cz/breach.html>
- [38] HENDRIX, Austin. Msg. *Ros.org* [online]. [cit. 2022-05-05]. Dostupné z: <http://wiki.ros.org/msg>
- [39] Services. *Ros.org* [online]. 2019-07-18 [cit. 2022-05-05]. Dostupné z: <http://wiki.ros.org/Services>
- [40] QUIGLEY, Morgan, Brian GERKEY a William D. SMART. *Programming robots with ROS*. Sebastopol: O'Reilly & Associates Incorporated, [2015]. ISBN 978-1-449-32389-9.
- [41] LUO, Huiwu. Actionlib: Package Summary. *Ros.org* [online]. 2019-07-22 [cit. 2022-05-05]. Dostupné z: <http://wiki.ros.org/actionlib/DetailedDescription>
- [42] Slamtec: RPlidar. *GitHub* [online]. 2021-10-27 [cit. 2022-02-11]. Dostupné z: https://github.com/Slamtec/rplidar_ros
- [43] V. LU, David a Jackie KAY. Joint state publisher. *GitHub* [online]. [cit. 2022-05-13]. Dostupné z: https://github.com/ros/joint_state_publisher
- [44] SUCAN, Ioan, Jackie KAY a Wim MEEUSSEN. Robot state publisher. *GitHub* [online]. [cit. 2022-05-13]. Dostupné z: https://github.com/ros/robot_state_publisher
- [45] BAZEMORE, Brad. Roboclaw_ros. *GitHub* [online]. 2017-2-10 [cit. 2022-02-13]. Dostupné z: https://github.com/sonyccd/roboclaw_ros
- [46] GERKEY, Brian. ROS perception: slam_gmapping. *GitHub* [online]. 2020-10-02 [cit. 2022-09-03]. Dostupné z: https://github.com/ros-perception/slam_gmapping
- [47] MARDER-EPPSTEIN, Eitan. ROS Planning: Navigation. *GitHub* [online]. 2021-11-18 [cit. 2021-09-01]. Dostupné z: <https://github.com/ros-planning/navigation>
- [48] Indoor plants 3D model. *TurboSquid* [online]. 2019-1-28 [cit. 2021-09-02]. Dostupné z: <https://www.turbosquid.com/3d-models/indoor-plants-3d-model-1372214>
- [49] *Gazebo: History* [online]. Open Source Robotics Foundation, 2014 [cit. 2022-05-09]. Dostupné z: <https://classic.gazebosim.org/>

Část 10

Seznam použitých zkratek a symbolů

ROS	<i>Robot Operating System</i>
UI	<i>User interface</i>
URDF	<i>Unified Robot Description Format</i>
XML	<i>Extensible Markup Language</i>
SDF	<i>Spatial Data File</i>
S.S.	<i>Souřadnicový systém</i>

Část 11

Seznam příloh

K práci jsou přiloženy rozšiřující kapitoly a složky obsahující ROS balíčky se zdrojovým kódem k vlastním vyhotoveným procesům a simulaci.

Kapitoly

Pracovní diagramy řídicího programu
Prototyp robotu
Testování systému
Tabulky měření

Složky

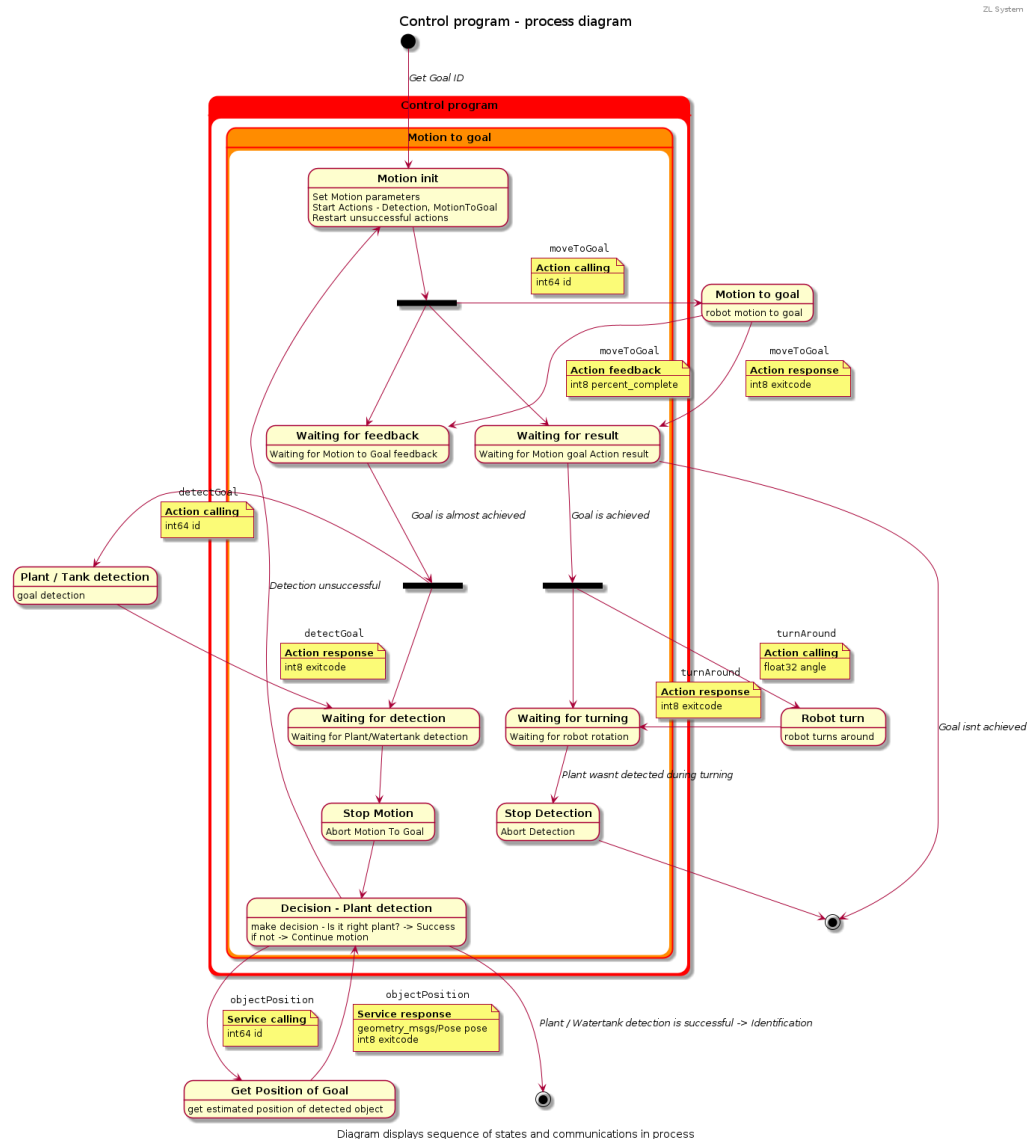
<i>\tf_calculation</i>	<i>\action</i>	def. akcí
	<i>\driver</i>	program pro Arduino
	<i>\msg</i>	def. zpráv
	<i>\script</i>	zdr. kód procesů
	<i>\srv</i>	def. služeb
	<i>\...</i>	auto. gen. soubory
<i>\camera_motion</i>	<i>\action</i>	def. akcí
	<i>\launch</i>	pomocné <i>launch</i> soubory
	<i>\msg</i>	def. zpráv
	<i>\rviz</i>	pomocné konfigurační soubory
	<i>\script</i>	zdr. kód procesů
	<i>\srv</i>	def. služeb
	<i>\...</i>	auto. gen. soubory, pomocné složky a soubory
<i>\sim_zl</i>	<i>\launch</i>	pomocné <i>launch</i> soubory
	<i>\model</i>	složka s konf. modelu robotu
	<i>\rviz</i>	pomocné konfigurační soubory
	<i>\script</i>	pomocné konfigurační programy
	<i>\world</i>	složka s def. objekty a prostředím
	<i>\...</i>	auto. gen. soubory, pomocné složky a soubory

Část 12

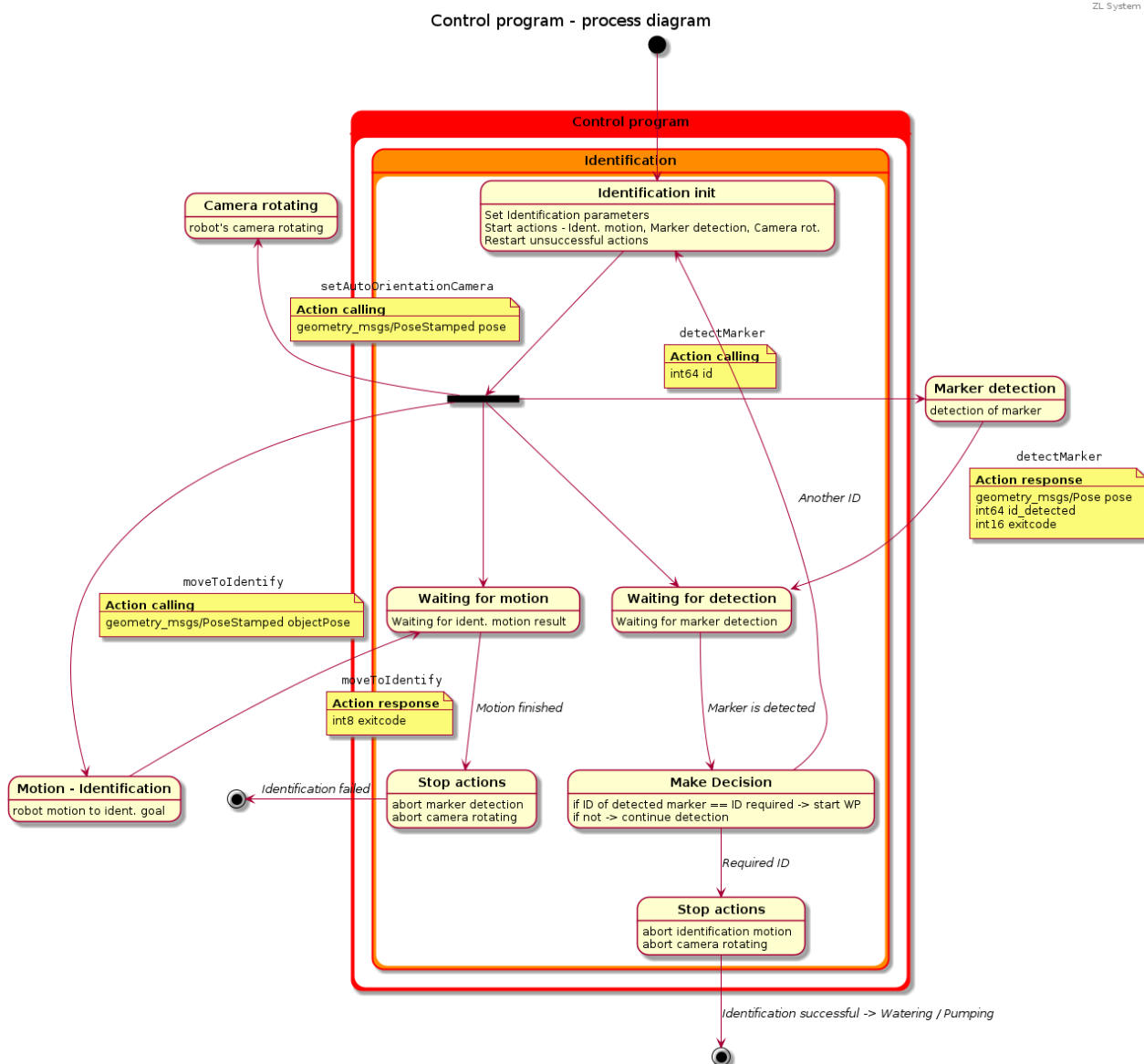
Přílohy

12.1 Pracovní diagramy řídicího programu

V této příloze jsou přiloženy pracovní diagramy řídicího programu, které byly propracovány v rámci konkretizace návrhu systému.



Obr. 12.1: Pracovní diagram pro fázi *Pohyb k cíli*

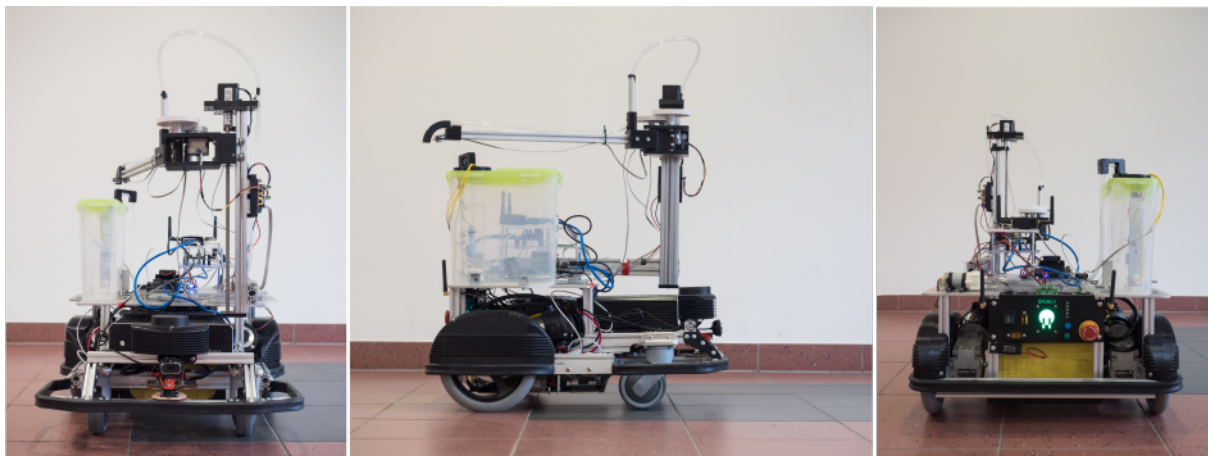
Obr. 12.2: Pracovní diagram pro fázi *Identifikace cíle*



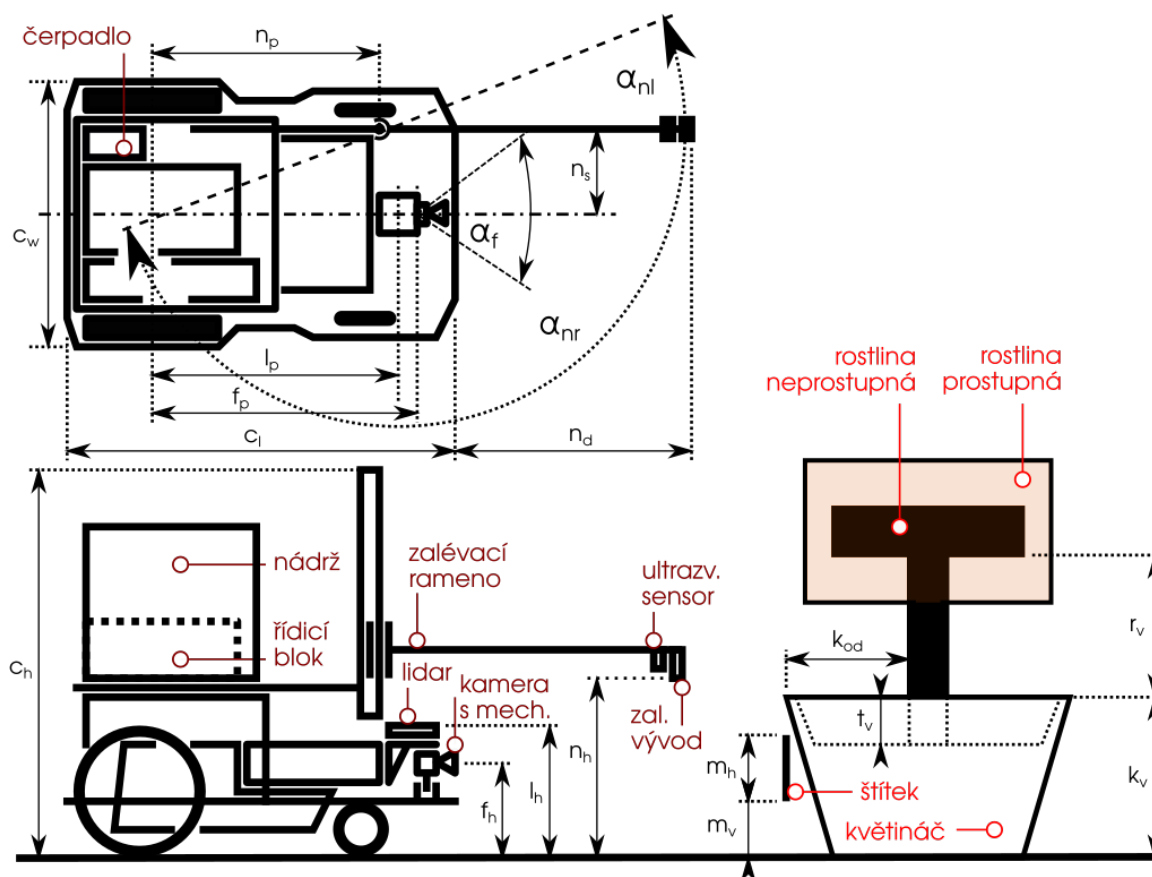
69

12.2 Prototyp robotu

V této příloze je uveden náčrtek prototypu a výsledné parametry robotu i rostliny. Doplněno je i blokové schéma zapojení komponent.



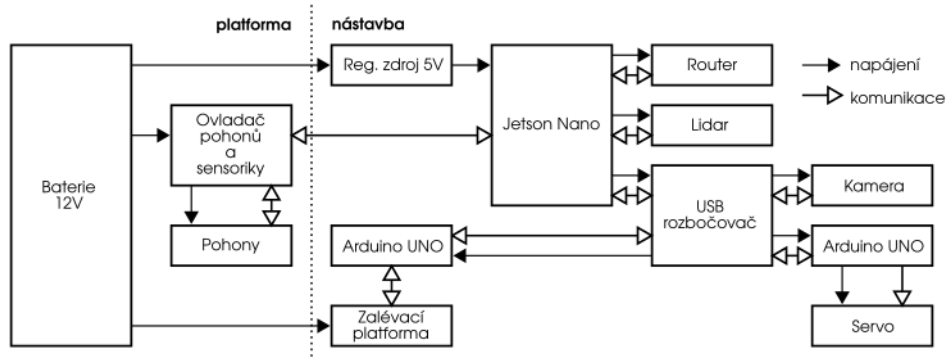
Obr. 12.4: Prototyp zalévacího robotu



Obr. 12.5: Náčrtek prototypu s parametry

Parametr	Význam	Hodnota
c_l	délka robotu	620 mm
c_w	šířka robotu	540 mm
c_h	výška robotu	710 mm
k_v	výška objektu	min. 250 mm
r_v	výška pracovního prostoru	min. 100 mm
k_{od}	hloubka pracovního prostoru	min. 70 mm
t_v	dosažitelná hloubka objektu	0 mm
m_v	výška pozice štítu	min. 50 mm, max. 250 mm
m_h	rozměr štítu	77 mm
n_p	pozice ramena	352 mm
n_s	vyosení ramena	30 mm
n_d	přesah ramena	400 mm
l_p	pozice lidar	435 mm
l_h	výška pozice lidar	245 mm
f_p	pozice kam. mech.	470 mm
f_h	výška pozice kam. mech.	165 mm
α_{nr}	úhlové natočení ramena	165°
α_{nl}	úhlové natočení ramena	25°
α_f	úhlové natočení kam. mech.	90°

Tab. 12.1: Tabulka parametrů prototypu



Obr. 12.6: Blokové schéma zapojení komponent prototypu

12.3 Testování systému

V následující kapitole budou podrobněji popsány jednotlivé testovací scénáře. Jejich přehled je v tabulce 12.2. Pokud není uvedeno jinak, jejich testování dopadlo úspěšně. U některých scénářů bylo možné provést měření přesnosti, či spolehlivosti.

Fáze	Scénář	Simulace	Leela	Breach
Detekce	<i>Snímání kamerou</i>	N	F	F, P
	<i>Detekce rostliny</i>	F	F	F
	<i>Lokalizace rostliny</i>	N	F	F, P
	<i>Detekce štítku jed.</i>	F	F	F
	<i>Detekce štítku průb.</i>	F	F	F, P
Pohyb	<i>Pohyb cíli</i>	F	F	F
	<i>Pohyb identifik.</i>	NI	NI	NI
	<i>Pohyb do opt. poz.</i>	F	F	F
	<i>Pohyb k rostlině</i>	F	F	F, P, S
	<i>Couvnutí</i>	F	F	F
	<i>Otáčení</i>	F	F	F
	<i>Plán. vlastní</i>	NI	NI	NI
	<i>Plán. převzatý konfigur.</i>	F	F	F
	<i>Lokalizace</i>	F	F	F
	<i>Řízení pohonů a sensorika</i>	N	F	F
	<i>Mapování</i>	F	F	F
	<i>Setup prostředí</i>	F	F	F
	<i>Test zal. platforma</i>	N	N	F
	<i>Přesun. do prac. pozí.</i>	N	N	F
	<i>Přesun. do přeprav. pozí.</i>	N	N	F
	<i>Provedení zalití</i>	N	N	F
	<i>Posílání sensor. dat</i>	N	N	F
Integrace	<i>Test kam. modulu</i>	N	F	F, P
	<i>Auto. orientace kamery</i>	F	F	F
	<i>Natáčení kamery</i>	F	F	F
	<i>Přepočty souřadnic</i>	F	F	F
	<i>Přepočty dat z det. štítků</i>	F	F	F, P
	<i>Odometrie pro Leelu</i>	N	F	N
Říd. prog.	<i>Spouštění systému</i>	F	F	F
	<i>Komunikace s databází</i>	F	F	F
	<i>Komunikace s UI</i>	F	F	F
	<i>Řízení procesů</i>	F	F	F
	<i>Celek - vše provedeno</i>	F	F	F
	<i>Celek - bez dojetí k rostlině</i>	F	F	F
	<i>Celek - bez detekce rostliny</i>	F	F	F
	<i>Celek - bez detekce štítku</i>	F	F	F
	<i>Celek - bez ident. štítku</i>	F	F	F
	<i>Celek - bez úsp. zalití</i>	F	F	F
	<i>Vypínání systému</i>	F	F	F
	<i>Update databáze</i>	F	F	F

– pokračování na další straně –

– pokračování z předchozí strany –				
Fáze	Scénář	Simulace	Leela	Breach
UI	<i>Update databáze</i>	F	F	F
	<i>Funkčnost rozhraní</i>	F	F	F
	<i>Personalizace prostředí</i>	F	F	F
Celý systém	<i>Rostl. v předpokládané pozici</i>	F	F	F, S
	<i>Rostl. v nepředpokládané blízké pozici</i>	F	F	F, S
	<i>Rostl. v nepředpokládané daleké pozici</i>	F	F	F, S
	<i>Rostl. neexistuje</i>	F	F	F, S
	<i>2 rostliny v obec. poloze</i>	F	F	F, S
	<i>3 rostliny v obec. poloze</i>	F	F	F, S

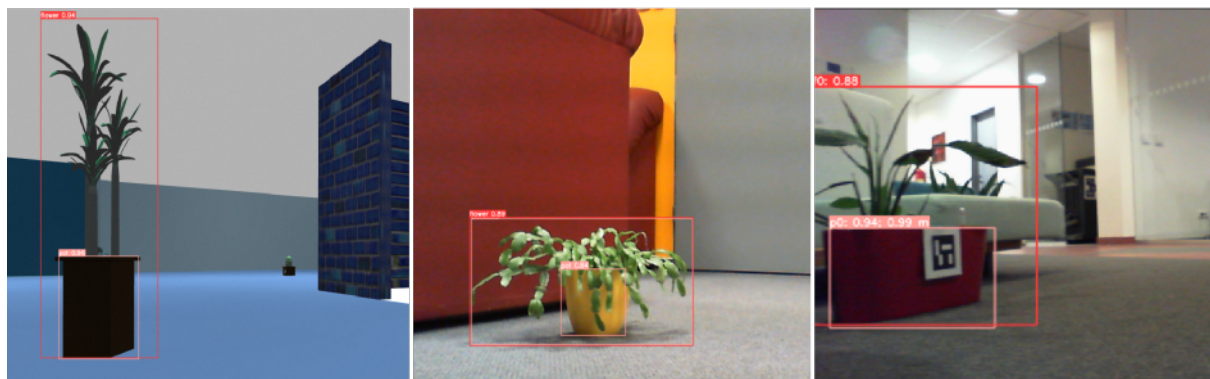
Tab. 12.2: Testovací scénáře a jejich provedení v jednotlivých fázích - (F) testování funkčnosti, (P) testování přesnosti, (S) testování spolehlivosti, (N) neprovádí se, (NI) neprovedeno, neintegrováno

Přebírání dat z kamery - je testována funkčnost a charakteristika přebírání dat z kamery. Script je proveden v rámci balíčku *aruco* v jazyce Python s využitím knihovny *openCV*. Podrobnější popis lze nalézt v [33].

Simulace - v simulaci není prováděno.

Leela / Prototyp - testováno bylo přebírání obrazových dat v rozlišení 424×240 px, 640×360 px a 1280×800 px. V rámci kompromisu mezi náročností zpracovávání dat ostatními procesy v systému a přesností závislé na rozlišení snímků bylo zvoleno rozlišení 640×360 px s frekvencí snímkování 10 Hz.

Detekce rostliny - je testována funkčnost procesu a jeho interakce s řídicím programem, popř. s kamerovým uzlem. Script je proveden v rámci balíčku *yolov5* v jazyce Python s využitím knihovny *yolov5*. Podrobnější popis lze nalézt v [33].



Obr. 12.7: Příklady provedené detekce ze simulace, testování na robotu Leela a prototypu s již implementovanou lokalizací rostliny

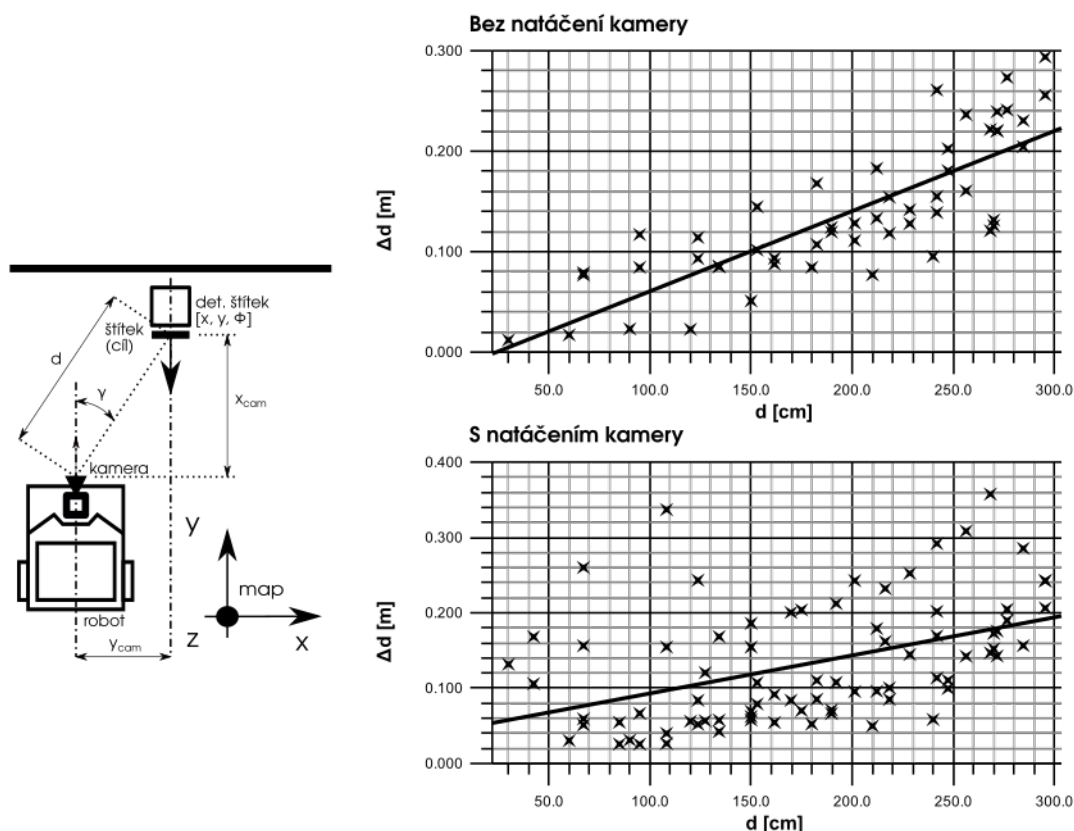
Simulace - testování je úspěšně provedeno s natrénovanou neuronovou sítí na datasetu vytvořeném v simulaci.

Leela / Prototyp - testování je úspěšně provedeno se strukturou natrénovanou na zpracovaném datasetu pro tento projekt. Proces detekuje kombinaci rostliny a květináče. V testovacím prostředí se osvědčil práh věrohodnosti detekce 0.7, při nižším docházelo často k falešným detekcím nábytku apod.. Proces průměrně spotřebovává 2533 MB, což představuje zhruba 63% veškeré operační paměti.

Průběžná detekce štítku - je testována funkčnost procesu a jeho komunikace s řídicím programem. Dále je ověřována přesnost a rychlost detekce štítků typu Aruco. Proces je proveden pomocí scriptu v rámci balíčku *aruco* v jazyce Python s využitím knihovny *openCV*. Podrobnější popis lze nalézt v [33].

Simulace - funkčnost úspěšně otestována.

Leela / Breach - byla zjišťována přesnost a rychlost detekce v součinnosti s procesy Přepočítávání dat ze štítku, Přepočtů v s.s. a Lokalizací. Proces je schopen s využitým rozlišením předávat informaci o detekci s frekvencí zhruba 10 Hz při využití 1775 MB RAM.



Obr. 12.8: Schéma měření a grafy závislosti chyby detekce na vzdálenosti kamery od štítku

Přesnost detekce byla zjišťována při dvou režimech - kamery v klidu a při jejím natáčení k cíli. Nákres měření a zavedené parametry lze vidět na obrázku. Výsledky měření

jsou zaznamenány v tabulkách 12.3 a 12.4. Z nich je patrné, že přesnost detekce je ovlivněna vzdáleností od detekovaného štítku, svou roli hraje i nepřesnost mech. kamerového modulu, či lokalizace. V rámci přibližně dvoumetrového maximálního odstupu od štítku, je přesnost i spolehlivost detekce však při zvoleném rozlišení kamery přijatelná a pohybuje se zhruba do 15 cm.

Lokalizace rostliny - je testována funkčnost procesu a jeho komunikace s řídicím programem. Dále je ověřována přesnost lokalizace. Proces je proveden pomocí scriptu v rámci balíčku *yolov5* v jazyce Python s využitím knihovny *yolov5* a knihovny *midas*. Podrobnější popis lze nalézt v [33].

Simulace / Leela - z časových důvodů netestováno. Funkcionalita nahrazena pevně stanovenou pozicí, kterou je pozice rostliny v systémové databázi.

Breach - byla zjišťována přesnost detekce v součinnosti s procesy Přepočtů v s.s. a Lokalizací. Výsledky několika měření lze vidět v tabulce 12.5. Proces v testovacím zjednodušeném prostředí vykazuje průměrnou chybu zhruba 30 cm.



Obr. 12.9: Schéma měření a snímek detekce z měření

Pohyb k cíli - je testována funkčnost procesu a jeho komunikace s řídicím programem. Proces je proveden pomocí scriptu v jazyce Python v rámci balíčku *planning*. V integrovaném provedení je využit balíček *move_base*. V rámci testování byla provedena i kontrola jeho konfigurace pro jednotlivé fáze. *Move_base* je spouštěn mimo hlavní řídicí program pomocí *launch* souboru. Podrobnější popis lze nalézt v [35].

Simulace - simulace ukázala některé neideální rysy chování plánovače - neprovedení pokusu o jízdu k pozici, na které je detekován objekt, či nespolehlivou interakci s *move_base* s pomocí akčního rozhraní. Při dojezdu na volnou pozici je proces funkční.

Leela / Breach - proces je funkční při výše zmíněných podmínkách.

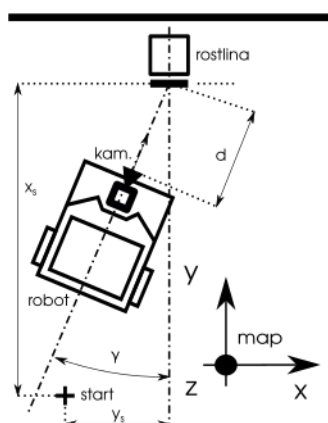
Pohyb identifikační - nebyl dodán pro provedení integrace, netestován.

Pohyb do optimální pozice - je testována funkčnost procesu a jeho komunikace s řídicím programem. Proces je proveden pomocí scriptu v jazyce Python v rámci balíčku *planning*. V integrovaném provedení je využit balíček *move_base*. Podrobnější popis lze nalézt v [35].

Simulace / Leela / Breach - v procesu není implementována kontrola volného prostoru kolem předpokládané optimální pozice. Při předpokladu vhodně umístěného štítu uživatelem, však lze tento požadavek zařadit do přípravy prostředí systému. Dále není provedena reakce při nedosažitelné pozici. Před zahájením jízdy je uskutečněno 10 s čekání na ustálení filtrovaných hodnot z detekce štítu.

Pohyb k rostlině - je testována funkčnost procesu a jeho komunikace s řídicím programem a jinými procesy. Proces je proveden pomocí scriptu v rámci balíčku *planning*. Podrobnější popis lze nalézt v [35].

Simulace / Leela - funkčnost byla úspěšně otestována. V integrovaném provedení má příjezd podobu zpětnovazebně regulovaného pohybu s pomocí dvoustavového regulátoru. Ten reguluje otáčení na základě odchylky robotu od osy štítu. Regulátor je zastavován při překročení minimální vzdálenosti od štítu.



Obr. 12.10: Schéma měření a snímek z testování

Breach - regulátor je už zastavován při překročení vzdálenosti od štítu - 30 cm, nebo signálu o detekci květináče od procesu Zálévání / Čerpání vody. U procesu byla testována přesnost příjezdu z různých úhlů odklonu od štítu - tj. vzdálenost robotu od štítu při zastavení a spolehlivost příjezdu. Naměřená data lze vidět v tabulce 12.6. Úspěšnost dosahuje 93% i přes poměrně jednoduché provedení.

Couvnutí - je testována funkčnost procesu a jeho komunikace s řídicím programem. Proces je proveden pomocí scriptu v rámci balíčku *planning*. Podrobnější popis lze nalézt v [35].

Simulace / Leela / Breach - funkčnost byla úspěšně otestována. Proces má podobu couvnutí o jasně stanovenou vzdálenost, která je kontrolována přepočty souřadnic polohy robotu. Kontrola kolize není provedena.

Lokalizace - je testována funkčnost. Je provedena v rámci převzatého balíčku *amcl*. Lokalizace je spouštěna mimo hlavní řídicí program pomocí *launch* souboru.

Simulace / Leela / Breach - funkčnost byla úspěšně otestována. Frekvence obnovování transformací v systému závisí na rychlosti posílání dat z lidarů. Proto byla na prototypu uměle zvýšena frekvence odesílání.

Řízení pohonů - je testována funkčnost. Je provedena v rámci převzatých balíčků *roboclaw* a *breach*. Script odometrie pro Leelu je proveden v rámci balíčku *tf_calculation*. Programy jsou spouštěny mimo hlavní řídicí program pomocí *launch* souboru.

Simulace - netestováno, řízení a odometrie simulovány pomocí pluginu.

Leela - funkčnost byla úspěšně otestována. Pro úpravu dat bylo nutné vytvořit script, který převádí zprávu *odom* do transformačních zpráv.

Breach - funkčnost byla úspěšně otestována. Pro potřebu zvýšení rychlosti obnovování transformačních dat byla navýšena frekvence posílání transformačních zpráv do s.s. *odom*.

Setup prostředí - je provedena konfigurace prostředí formou posílání mapy testovacího prostoru a aktualizace dat v rámci systémové databáze. Posílání mapy je provedeno pomocí převzatého balíčku *map_server*, který přebírá soubory obsahující namapovaný prostor v rámci balíčku *planning*. Konfigurace je provedena mimo řídicí program v rámci *launch* souboru.

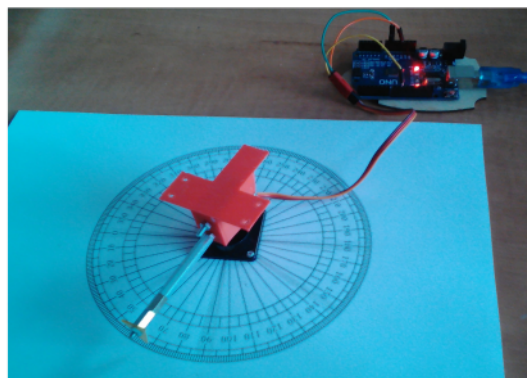
Testování zalévací platformy - je testována funkčnost zal. platformy a její komunikace se systémem. Podrobnější popis lze nalézt v [36].

Simulace / Leela - prováděno je jen testování na simulaci komunikace s řídicím programem.

Breach - v první fázi byla prováděna konfigurace zal. platformy, která byla přes svůj uzel zapojena do systému. Po přesunutí zal. ramena do pracovní pozice dochází ke spuštění detekce květináče, která posílá informaci o jeho přítomnosti do systému zhruba s frekvencí 5 Hz. To je přerušeno při zavolání hlavní akce Zalévání. Při něm nedochází ve větší míře ke korekci ramena nad květináčem. Je prováděna pouze jeho kontrola. Poté dochází k bezproblémovému přesunu ramena do přepravní pozice.

Testování kamerového mechanismu a orientování kamery - je testována funkčnost modulu a jeho úhlová přesnost natáčení v jedné ose. Proces je implementován v rámci balíčku *camera_motion*.

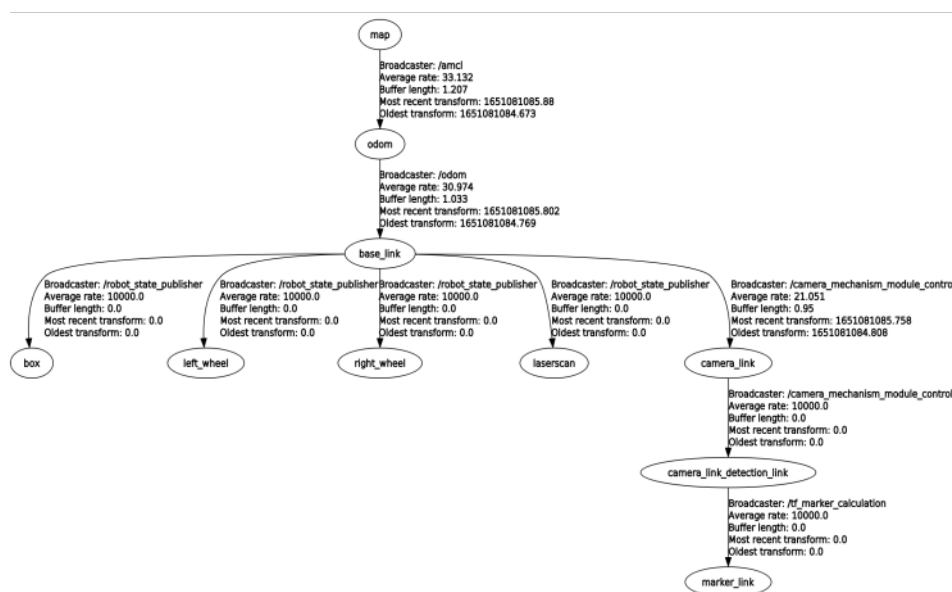
Simulace - je prováděno testování funkčnosti a komunikace s ostatními procesy v systému.



Obr. 12.11: Testovací přípravek na měření přesnosti mechanismu

Leela / Breach - bylo provedeno odladění reálného mechanismu a jeho připojení pomocí mikrokontroléru do systému. Byla testována přesnost mechanismu s využitím úhlooměru. Záznamy z měření jsou uvedeny v tabulce 12.9. Z nich je patrné, že mechanismus reaguje na řídicí povel s chybou zhruba 2° při vůli 2° . To je posléze zohledněno v procesech spojených s detekcí.

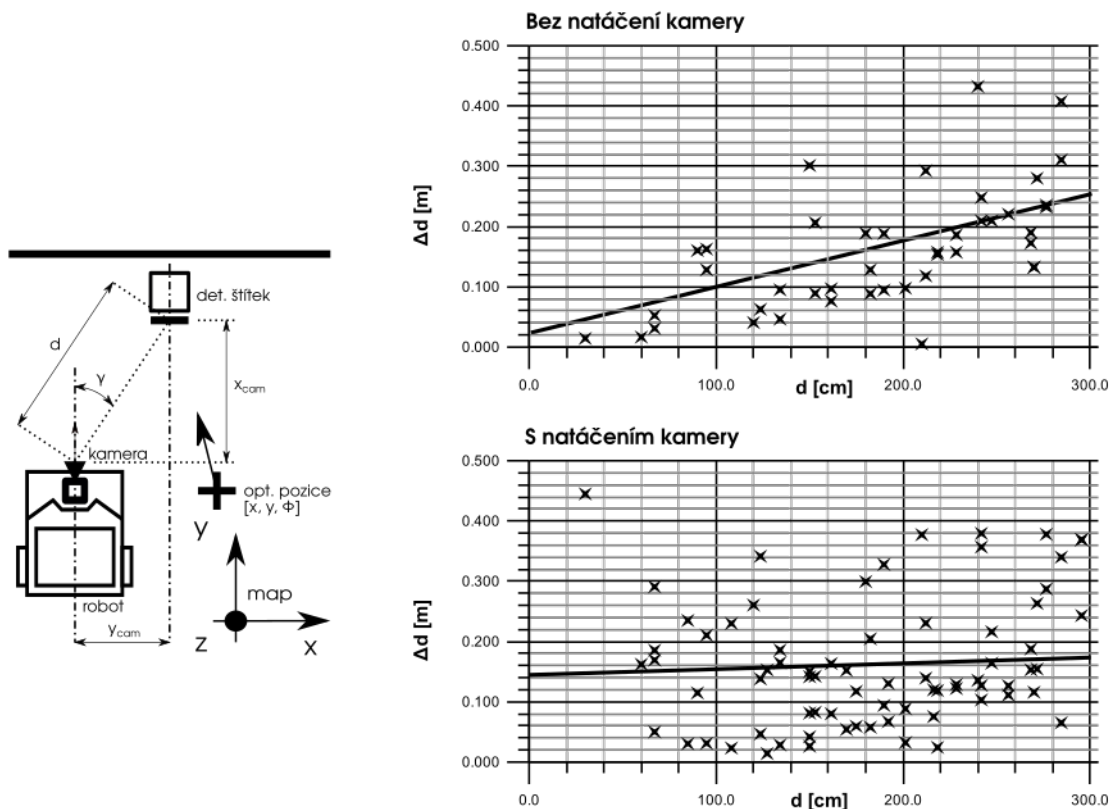
Přepočítávání souřadnic - je testována funkčnost a rychlost procesu. Proces je implementován v rámci balíčku *tf_calculation*.



Obr. 12.12: Záznam souřadnicových systémů na prototypu při detekování štítku a jejich frekvence odesílání

Přepočítávání dat ze štítku - je testována funkčnost a rychlost procesu. Proces je implementován v rámci balíčku *tf_calculation*.

Simulace - je prováděno testování funkčnosti a komunikace s ostatními procesy v systému.



Obr. 12.13: Schéma měření a grafy závislosti chyby odhadu polohy opt. pozice na vzdálenosti kamery od štítku

Leela / Breach - na základě testování detekce štítku byla zjištěna potřeba dodatečné filtrace příchozích dat. Po implementaci mediánového filtru s oknem 30 vzorků byl problém šumu zejména při detekci orientace štítku v testovacím prostředí odstraněn. Zprávy se posílají se stejnou frekvencí rovnou 7.5 Hz. Byla testována též přesnost odhadu optimální pozice v režimu s natáčenou kamerou a kamerou v klidu. Výsledky měření lze vidět v tabulkách 12.7 a 12.8. Chyba odhadu se pohybuje průměrně do zhruba 20 cm.

Testování řídicího programu - je testována jeho funkčnost a reakce na různé odpovědi od řízených procesů, či stavy systému. V rámci toho byla ověřena i funkčnost provedení komunikace s databází. Proces je implementován v rámci balíčku *zaleela_main* a *zaleela_mqtt*. Podrobnější popis lze nalézt v [34].

Simulace - bylo provedeno ověření celkového chování řídicího programu, tj. sled jednotlivých fází, spouštění def. procesů a komunikace s ostatními částmi systému - aktualizace dat v databázi, či posílání některých systémových informací. V neposlední řadě bylo odzkoušeno chování programu v cyklu.

Leela / Breach - testováním byla potvrzena funkčnost programu. V programu chybí implementace reakce na některé vstupy např. od uživatele, či havarijní výstupy z procesů. To je však možné považovat za součást dalšího vylepšování systému.

Uživatelské rozhraní - bylo provedené otestování základních funkcí uživatelského rozhraní. Jeho finální podobu lze vidět na přiložených obrázcích 6.1. Podrobnější popis lze nalézt v [34].

Jednalo se zejména o přidávání testovacích rostlin a jejich konfiguraci. Byla otestována i zpětná vazba pro uživatele ve formě výpisu stavu ze systému, či zobrazení dat v mapě. V neposlední řadě byl proveden i test možnosti ručního ovládání robotu. Personalizace prostředí také probíhá bez problémů.

Celý systém - v této fázi bylo ověřováno chování celku na prototypu. V zásadě se tyto testy týkaly variace úkolu zalití definované rostliny, nebo rostlin, které byly zaneseny do systému.

V prvním scénáři se rostlina nacházela v předpokládané pozici dle databáze v systému. Vzhledem k dříve zmíněným problémům pohybu k cíli však tento scénář nemohl být úspěšně vykonán. V dalších scénářích se již rostlina nacházela mimo přesnou pozici, popř. se nenacházela v okolí předpokládané pozice, což umožnilo provádění úspěšného testování.

Testování vyvrcholilo provedením zalévání 2 a 3 rostlin v cyklu. Celková úspěšnost provádění zalévání je zřejmá z tabulky 7.1. Vzhledem k velkému počtu opakování byla většina testů prováděna bez zalití rostliny. Průměrná doba zalití rostliny i se započtením jízdy se pohybovala okolo 3 minut.

12.4 Tabulky měření

i	x_{cam} [cm]	y_{cam} [cm]	d_{cam} [cm]	γ [°]	x [m]	y [m]	ϕ [rad]	$ \Delta x $ [m]	$ \Delta y $ [m]	$ \Delta d $ [m]	$ \Delta \phi $ [°]
1	270	0	270.0	0.0	14.829	22.129	-2.741	0.079	0.098	0.126	7.029
2	270	0	270.0	0.0	14.820	22.129	-2.741	0.088	0.098	0.132	7.029
3	240	0	240.0	0.0	14.853	22.149	-2.943	0.054	0.078	0.095	18.602
4	210	0	210.0	0.0	14.850	22.176	-2.632	0.057	0.051	0.077	0.786
5	180	0	180.0	0.0	14.842	22.174	-2.506	0.066	0.053	0.084	6.399
6	150	0	150.0	0.0	14.876	22.187	-2.606	0.031	0.040	0.051	0.662
7	120	0	120.0	0.0	14.896	22.208	-2.604	0.012	0.019	0.023	0.796
8	90	0	90.0	0.0	14.904	22.204	-2.741	0.003	0.023	0.023	7.053
9	60	0	60.0	0.0	14.908	22.210	-2.643	0.000	0.017	0.017	1.430
10	30	0	30.0	0.0	14.916	22.219	-2.621	0.009	0.008	0.012	0.186
11	270	60	276.6	12.5	15.149	22.224	-2.636	0.241	0.003	0.241	1.007
12	240	60	247.4	14.0	15.087	22.209	-2.874	0.180	0.019	0.181	14.667
13	210	60	218.4	15.9	15.059	22.198	-2.623	0.152	0.029	0.154	0.281
14	180	60	189.7	18.4	15.023	22.195	-2.651	0.115	0.032	0.119	1.882
15	150	60	161.6	21.8	14.986	22.187	-2.638	0.078	0.040	0.087	1.128
16	120	60	134.2	26.6	14.982	22.186	-2.673	0.075	0.042	0.085	3.125
17	270	120	295.5	24.0	15.202	22.221	2.817	0.294	0.006	0.294	48.618
18	240	120	268.3	26.6	15.127	22.192	-2.739	0.219	0.035	0.222	6.931
19	270	90	284.6	18.4	15.130	22.168	-2.485	0.223	0.059	0.230	7.610
20	240	90	256.3	20.6	15.141	22.190	2.993	0.234	0.037	0.236	38.503
21	210	90	228.5	23.2	15.044	22.189	-2.610	0.136	0.038	0.142	0.467
22	180	90	201.2	26.6	15.029	22.184	-2.654	0.121	0.043	0.128	2.075
23	270	30	271.7	6.3	15.107	22.095	-2.850	0.199	0.132	0.239	13.296
24	240	30	241.9	7.1	15.141	22.111	-2.674	0.234	0.116	0.261	3.184
25	210	30	212.1	8.1	15.038	22.098	-2.756	0.130	0.129	0.183	7.889
26	180	30	182.5	9.5	15.041	22.124	-2.740	0.133	0.103	0.168	7.000
27	150	30	153.0	11.3	15.009	22.124	-2.582	0.101	0.103	0.145	2.072
28	120	30	123.7	14.0	14.985	22.143	-3.113	0.077	0.084	0.114	28.353
29	90	30	94.9	18.4	14.992	22.146	-2.624	0.085	0.081	0.117	0.339
30	60	30	67.1	26.6	14.967	22.178	-2.666	0.059	0.049	0.077	2.747
31	270	-30	271.7	6.3	15.120	22.285	-2.358	0.212	0.058	0.220	14.885
32	270	-30	271.7	6.3	15.120	22.285	-2.358	0.212	0.058	0.220	14.885
33	240	-30	241.9	7.1	15.044	22.202	-2.862	0.137	0.025	0.139	13.965
34	210	-30	212.1	8.1	15.040	22.208	-2.503	0.132	0.019	0.133	6.601
35	180	-30	182.5	9.5	15.011	22.197	-2.695	0.103	0.030	0.107	4.394
36	150	-30	153.0	11.3	15.004	22.194	-2.634	0.096	0.033	0.102	0.912
37	120	-30	123.7	14.0	15.000	22.213	-2.643	0.092	0.014	0.093	1.425
38	90	-30	94.9	18.4	14.987	22.198	-2.588	0.079	0.029	0.084	1.702
39	60	-30	67.1	26.6	14.981	22.199	-2.657	0.074	0.028	0.079	2.247
40	270	-60	276.6	12.5	15.159	22.336	-2.612	0.251	0.109	0.273	0.325
41	240	-60	247.4	14.0	15.102	22.285	-2.644	0.194	0.058	0.202	1.510
42	210	-60	218.4	15.9	15.021	22.193	-2.763	0.113	0.034	0.118	8.282
43	180	-60	189.7	18.4	15.032	22.236	-2.651	0.124	0.009	0.124	1.908
44	150	-60	161.6	21.8	15.000	22.210	-2.621	0.092	0.017	0.093	0.192
45	120	-60	134.2	26.6	14.989	22.204	-2.677	0.082	0.023	0.085	3.355
46	270	-90	284.6	18.4	15.088	22.324	-2.772	0.180	0.097	0.204	8.802
47	240	-90	256.3	20.6	15.059	22.282	-2.736	0.151	0.055	0.161	6.736
48	210	-90	228.5	23.2	15.031	22.261	-2.622	0.123	0.034	0.128	0.235
49	180	-90	201.2	26.6	15.017	22.248	-1.652	0.109	0.021	0.111	55.357
50	270	-120	295.5	24.0	15.119	22.372	-1.748	0.211	0.145	0.256	49.836
51	240	-120	268.3	26.6	15.023	22.263	-2.719	0.115	0.036	0.121	5.780
52	210	-120	241.9	29.7	15.048	22.293	-2.719	0.140	0.066	0.155	5.801
$\bar{z}$								0.125	0.051	0.140	8.659
$\sigma(z)$								0.070	0.036	0.070	12.893

Tab. 12.3: Tabulka měření přesnosti průběžné detekce štítků kamerou s neměnným natočením u štítku s pozicí $x = 14.91$ m, $y = 22.23$ m a natočením $\Phi = 0.52$ rad

i	x_{cam} [cm]	y_{cam} [cm]	d_{cam} [cm]	γ [°]	x [m]	y [m]	ϕ [rad]	Δx [m]	Δy [m]	Δd [m]	$\Delta \phi$ [°]
1	270	120	295.5	24.0	15.003	22.450	-2.730	0.095	0.223	0.243	6.427
2	270	120	295.5	24.0	15.003	22.450	-2.730	0.095	0.223	0.243	6.427
3	240	120	268.3	26.6	14.736	22.540	-2.508	0.172	0.313	0.358	6.307
4	210	120	241.9	29.7	14.808	22.501	-2.432	0.100	0.274	0.292	10.641
5	180	120	216.3	33.7	14.740	22.388	-2.451	0.168	0.161	0.232	9.545
6	150	120	192.1	38.7	14.707	22.296	-2.519	0.201	0.069	0.212	5.676
7	120	120	169.7	45.0	14.710	22.259	-2.550	0.198	0.032	0.200	3.920
8	90	120	150.0	53.1	14.723	22.247	-2.518	0.185	0.020	0.186	5.734
9	60	120	134.2	63.4	14.741	22.202	-2.538	0.167	0.025	0.168	4.596
10	30	120	123.7	76.0	14.885	22.469	-2.598	0.023	0.242	0.243	1.125
11	270	90	284.6	18.4	14.793	22.489	-2.446	0.115	0.262	0.286	9.829
12	240	90	256.3	20.6	14.744	22.489	-2.503	0.163	0.262	0.309	6.567
13	210	90	228.5	23.2	14.711	22.385	-2.494	0.197	0.158	0.252	7.113
14	180	90	201.2	26.6	14.746	22.408	-2.469	0.162	0.181	0.243	8.545
15	150	90	174.9	31.0	14.736	22.337	-2.477	0.172	0.110	0.204	8.083
16	120	90	150.0	36.9	14.770	22.297	-2.481	0.138	0.070	0.154	7.842
17	90	90	127.3	45.0	14.788	22.238	-2.491	0.120	0.011	0.120	7.255
18	60	90	108.2	56.3	14.847	22.369	-2.141	0.061	0.142	0.155	27.354
19	270	60	276.6	12.5	14.953	22.411	-2.381	0.046	0.184	0.190	13.605
20	240	60	247.4	14.0	14.849	22.320	-2.693	0.059	0.093	0.110	4.278
21	210	60	218.4	15.9	14.841	22.302	-2.571	0.066	0.075	0.100	2.701
22	180	60	189.7	18.4	14.884	22.294	-2.343	0.024	0.067	0.071	15.774
23	150	60	161.6	21.8	14.877	22.272	-2.477	0.031	0.045	0.054	8.088
24	120	60	134.2	26.6	14.867	22.239	-2.476	0.041	0.012	0.042	8.132
25	90	60	108.2	33.7	14.891	22.207	-2.470	0.017	0.020	0.026	8.494
26	60	60	84.9	45.0	14.926	22.176	-2.501	0.018	0.052	0.055	6.699
27	30	60	67.1	63.4	15.007	22.348	-2.653	0.099	0.121	0.156	2.030
28	270	30	271.7	6.3	14.964	22.394	-2.407	0.057	0.167	0.176	12.061
29	240	30	241.9	7.1	14.921	22.429	-2.409	0.013	0.201	0.202	12.000
30	210	30	212.1	8.1	14.780	22.352	-2.594	0.128	0.125	0.179	1.348
31	180	30	182.5	9.5	14.840	22.279	-2.535	0.068	0.052	0.085	4.753
32	150	30	153.0	11.3	14.824	22.294	-2.499	0.084	0.067	0.107	6.838
33	120	30	123.7	14.0	14.863	22.253	-2.506	0.045	0.026	0.052	6.444
34	90	30	94.9	18.4	14.884	22.218	-2.491	0.024	0.009	0.026	7.283
35	60	30	67.1	26.6	14.861	22.190	-2.508	0.046	0.037	0.059	6.319
36	30	30	42.4	45.0	14.934	22.393	-2.111	0.026	0.166	0.168	29.045
37	270	0	270.0	0.0	14.940	22.397	-2.111	0.032	0.170	0.173	29.045
38	270	0	270.0	0.0	14.946	22.374	-2.569	0.038	0.147	0.152	2.810
39	240	0	240.0	0.0	14.958	22.256	-2.545	0.051	0.029	0.058	4.156
40	210	0	210.0	0.0	14.908	22.277	-2.344	0.000	0.050	0.050	15.690
41	180	0	180.0	0.0	14.895	22.278	-2.728	0.013	0.051	0.052	6.323
42	150	0	150.0	0.0	14.903	22.284	-2.537	0.005	0.057	0.058	4.613
43	120	0	120.0	0.0	14.881	22.276	-2.411	0.026	0.049	0.056	11.871
44	90	0	90.0	0.0	14.903	22.258	-2.510	0.004	0.031	0.031	6.214
45	60	0	60.0	0.0	14.923	22.253	-2.507	0.015	0.026	0.030	6.386
46	30	0	30.0	0.0	15.039	22.240	-2.380	0.131	0.013	0.132	13.642
47	270	-30	271.7	6.3	15.048	22.202	-2.744	0.140	0.025	0.143	7.209
48	240	-30	241.9	7.1	15.070	22.179	-3.011	0.162	0.048	0.169	22.505
49	210	-30	212.1	8.1	14.990	22.178	-2.741	0.083	0.049	0.096	7.033
50	180	-30	182.5	9.5	15.000	22.167	-2.814	0.093	0.060	0.110	11.218
51	150	-30	153.0	11.3	14.979	22.192	-2.797	0.071	0.035	0.079	10.255
52	120	-30	123.7	14.0	14.977	22.179	-2.684	0.069	0.048	0.084	3.797
53	90	-30	94.9	18.4	14.963	22.190	-2.669	0.055	0.037	0.066	2.934
54	60	-30	67.1	26.6	14.956	22.211	-2.670	0.049	0.016	0.051	2.981
55	30	-30	42.4	45.0	14.948	22.325	2.984	0.041	0.098	0.106	39.012
56	270	-60	276.6	12.5	15.091	22.318	-2.826	0.183	0.091	0.205	11.945
57	240	-60	247.4	14.0	15.007	22.231	-2.562	0.099	0.004	0.099	3.211
58	210	-60	218.4	15.9	14.965	22.165	-2.631	0.058	0.062	0.085	0.750
59	180	-60	189.7	18.4	14.944	22.172	-2.604	0.036	0.055	0.066	0.822
60	150	-60	161.6	21.8	14.957	22.305	-2.592	0.049	0.078	0.092	1.505
61	120	-60	134.2	26.6	14.949	22.187	-2.668	0.042	0.040	0.057	2.859
62	90	-60	108.2	33.7	14.934	22.197	-2.663	0.027	0.030	0.040	2.554
63	60	-60	84.9	45.0	14.928	22.211	-2.657	0.020	0.016	0.026	2.248
64	30	-60	67.1	63.4	15.158	22.155	-2.595	0.250	0.072	0.260	1.303
65	270	-90	284.6	18.4	15.047	22.155	-2.578	0.139	0.072	0.156	2.318
66	240	-90	256.3	20.6	15.042	22.181	-2.700	0.135	0.046	0.142	4.673
67	210	-90	228.5	23.2	15.043	22.177	-2.651	0.136	0.050	0.145	1.875
68	180	-90	201.2	26.6	14.977	22.161	-2.627	0.069	0.066	0.095	0.524
69	150	-90	174.9	31.0	14.964	22.185	-2.646	0.056	0.042	0.070	1.621
70	120	-90	150.0	36.9	14.955	22.177	-2.655	0.047	0.050	0.069	2.140
71	90	-90	127.3	45.0	14.933	22.176	-2.651	0.025	0.051	0.057	1.876
72	60	-90	108.2	56.3	15.234	22.142	2.940	0.326	0.085	0.337	41.562
73	270	-120	295.5	24.0	15.087	22.124	-2.595	0.179	0.103	0.206	1.335
74	240	-120	268.3	26.6	15.015	22.127	-2.671	0.107	0.100	0.147	3.042
75	210	-120	241.9	29.7	14.997	22.156	-2.645	0.089	0.071	0.114	1.531
76	180	-120	216.3	33.7	15.043	22.137	-2.621	0.135	0.090	0.162	0.163
77	150	-120	192.1	38.7	14.972	22.141	-2.637	0.065	0.086	0.108	1.109
78	120	-120	169.7	45.0	14.927	22.145	-2.643	0.019	0.082	0.084	1.427
79	90	-120	150.0	53.1	14.898	22.166	-2.643	0.010	0.061	0.062	1.408
$\bar{z}$								0.086	0.087	0.133	7.600
$\sigma(z)$								0.066	0.071	0.080	8.046

Tab. 12.4: Tabulka měření přesnosti průběžné detekce štítků s aktivním natáčením kamery u štítku s pozicí $x = 14.91$ m, $y = 22.23$ m a natočením $\Phi = 0.52$ rad

i	x_{cam} [m]	y_{cam} [m]	d [m]	x [m]	y [m]	$ \Delta x $ [m]	$ \Delta y $ [m]	$ \Delta d $ [m]
1	12.927	21.862	2.219	15.182	22.643	0.181	0.006	0.181
2	12.986	21.814	2.181	14.755	22.451	0.247	0.197	0.316
3	12.986	21.814	2.181	14.811	22.472	0.190	0.177	0.260
4	13.687	21.458	1.774	14.685	22.354	0.317	0.295	0.433
5	13.687	21.458	1.774	14.699	22.364	0.302	0.285	0.415
6	14.399	21.400	1.386	14.790	22.183	0.211	0.466	0.512
7	14.930	21.470	1.181	14.998	22.304	0.003	0.345	0.345
8	14.930	21.470	1.181	14.995	22.265	0.006	0.384	0.384
9	14.837	21.494	1.166	14.974	22.331	0.028	0.318	0.319
10	13.155	22.369	1.867	15.017	22.718	0.016	0.069	0.071
11	13.716	22.698	1.286	14.823	22.684	0.179	0.035	0.182
12	13.735	22.518	1.273	14.936	22.695	0.065	0.046	0.080
13	13.093	21.217	2.386	15.227	23.004	0.225	0.355	0.420
14	13.163	21.293	2.284	15.208	22.958	0.206	0.309	0.371
15	13.673	20.938	2.166	14.678	22.025	0.324	0.623	0.702
16	13.759	21.269	1.857	14.922	22.541	0.079	0.108	0.134
17	12.083	21.798	3.039	14.911	22.762	0.091	0.113	0.145
18	12.417	21.487	2.834	15.151	22.342	0.150	0.307	0.342
19	12.257	20.851	3.281	14.649	22.726	0.353	0.078	0.361
20	12.367	20.384	3.474	14.835	22.594	0.166	0.055	0.175
21	11.504	21.494	3.683	15.410	22.779	0.409	0.130	0.429
22	11.842	20.689	3.718	15.113	22.922	0.112	0.273	0.295
$\bar{z}$						0.175	0.226	0.312
$\sigma(z)$						0.118	0.160	0.152

Tab. 12.5: Tabulka měření přesnosti odhadování polohy rostliny v prostoru se známou pozicí $x = 15.00$ m a $y = 22.65$ m

i	x_s [mm]	y_s [mm]	d_{ROS} [mm]	d [mm]	γ [°]	Výsledek
1	750	0	253	245,0	0,0	úspěšný
2	750	0	252	240,0	0,0	úspěšný
3	750	0	264	240,0	0,0	úspěšný
4	750	0	264	235,0	0,0	úspěšný
5	750	0	266	245,0	0,0	úspěšný
6	750	300	251	193,2	11,5	úspěšný
7	750	300	262	243,3	11,7	úspěšný
8	750	300	265	237,2	16,6	úspěšný
9	750	300	287	226,3	21,1	úspěšný
10	750	300	281	248,6	14,0	úspěšný
11	1000	0	268	205,0	0,0	úspěšný
12	1000	300	260	232,6	11,1	úspěšný
13	1000	450	279	237,7	20,1	úspěšný
14	1000	300	265	127,5	27,3	úspěšný
15	1000	450	258	235,8	18,4	úspěšný
16	1000	300	278	237,7	12,8	úspěšný
17	1000	450	282	245,2	18,6	úspěšný
18	1250	0	265	255,0	0,0	úspěšný
19	1250	300	275	212,9	9,7	úspěšný
20	1250	300	269	256,2	14,2	úspěšný
21	1250	450	302	248,8	19,3	neúspěšný
22	1250	450	265	256,3	18,4	úspěšný
23	1250	600	264	257,4	20,2	úspěšný
24	1250	600	259	245,2	19,6	úspěšný
25	1250	600	254	239,3	20,6	úspěšný
26	1500	300	282	264,1	14,5	úspěšný
27	1500	300	257	245,8	6,4	úspěšný
28	1500	300	250	228,0	13,0	úspěšný
29	1500	450	310	243,8	14,8	neúspěšný
30	1500	450	284	272,0	14,5	úspěšný
31	1500	450	258	249,9	16,9	úspěšný
32	1500	600	264	251,4	17,3	úspěšný
33	1500	600	283	212,1	11,0	úspěšný
34	1500	600	279	208,1	20,6	úspěšný
35	1500	750	277	203,5	24,7	úspěšný
36	1500	750	273	243,8	22,3	úspěšný
37	1500	750	260	253,9	27,5	úspěšný
38	1500	0	254	240,0	0,0	úspěšný
Úspěšnost %						93,3

Tab. 12.6: Tabulka měření testování dojezdu

i	x_{cam} [cm]	y_{cam} [cm]	d_{cam} [cm]	γ [°]	x [m]	y [m]	ϕ [rad]	$ \Delta x $ [m]	$ \Delta y $ [m]	$ \Delta d $ [m]	$ \Delta \phi $ [°]
1	270	0	270.0	0.0	13.505	21.567	0.401	0.101	0.086	0.132	6.823
2	270	0	270.0	0.0	13.505	21.567	0.401	0.101	0.086	0.132	6.823
3	240	0	240.0	0.0	13.383	21.852	0.199	0.223	0.370	0.432	18.396
4	210	0	210.0	0.0	13.601	21.480	0.510	0.005	0.002	0.005	0.580
5	180	0	180.0	0.0	13.657	21.300	0.635	0.051	0.182	0.189	6.605
6	150	0	150.0	0.0	13.446	21.737	0.535	0.160	0.255	0.301	0.868
7	120	0	120.0	0.0	13.612	21.442	0.537	0.006	0.040	0.041	1.002
8	90	0	90.0	0.0	13.523	21.619	0.401	0.083	0.137	0.160	6.847
9	60	0	60.0	0.0	13.595	21.494	0.499	0.011	0.013	0.017	1.224
10	30	0	30.0	0.0	13.618	21.474	0.520	0.012	0.008	0.014	0.020
11	270	60	276.6	12.5	13.838	21.496	0.506	0.232	0.015	0.232	0.801
12	240	60	247.4	14.0	13.623	22.109	0.268	0.017	0.627	0.627	14.461
13	210	60	218.4	15.9	13.757	21.454	0.519	0.151	0.028	0.154	0.075
14	180	60	189.7	18.4	13.700	21.488	0.491	0.094	0.006	0.094	1.676
15	150	60	161.6	21.8	13.694	21.441	0.504	0.088	0.041	0.097	0.921
16	120	60	134.2	26.6	13.645	21.507	0.469	0.039	0.025	0.046	2.919
17	270	120	295.5	24.0	13.788	22.697	-0.325	0.182	1.216	1.229	48.412
18	240	120	268.3	26.6	13.750	21.605	0.403	0.144	0.123	0.190	6.725
19	270	90	284.6	18.4	13.943	21.252	0.656	0.337	0.230	0.408	7.816
20	240	90	256.3	20.6	13.659	22.412	-0.148	0.053	0.930	0.932	38.297
21	210	90	228.5	23.2	13.754	21.429	0.532	0.148	0.053	0.157	0.673
22	180	90	201.2	26.6	13.704	21.481	0.487	0.098	0.000	0.098	1.868
23	270	30	271.7	6.3	13.745	21.725	0.292	0.138	0.244	0.280	13.090
24	240	30	241.9	7.1	13.834	21.383	0.468	0.228	0.099	0.248	2.978
25	210	30	212.1	8.1	13.700	21.554	0.386	0.093	0.072	0.118	7.683
26	180	30	182.5	9.5	13.718	21.418	0.401	0.111	0.064	0.128	6.794
27	150	30	153.0	11.3	13.750	21.334	0.560	0.144	0.148	0.206	2.278
28	120	30	123.7	14.0	13.485	22.100	0.029	0.121	0.618	0.629	28.147
29	90	30	94.9	18.4	13.697	21.392	0.518	0.091	0.090	0.128	0.132
30	60	30	67.1	26.6	13.635	21.491	0.476	0.029	0.009	0.031	2.541
31	270	-30	271.7	6.3	14.066	21.234	0.783	0.460	0.247	0.523	15.091
32	270	-30	271.7	6.3	14.066	21.234	0.783	0.460	0.247	0.523	15.091
33	240	-30	241.9	7.1	14.180	20.995	0.280	0.574	0.487	0.753	13.758
34	210	-30	212.1	8.1	13.854	21.326	0.639	0.248	0.156	0.293	6.807
35	180	-30	182.5	9.5	13.662	21.550	0.447	0.056	0.068	0.088	4.187
36	150	-30	153.0	11.3	13.693	21.464	0.508	0.087	0.018	0.089	0.706
37	120	-30	123.7	14.0	13.666	21.501	0.499	0.059	0.020	0.063	1.219
38	90	-30	94.9	18.4	13.737	21.385	0.553	0.131	0.097	0.163	1.908
39	60	-30	67.1	26.6	13.655	21.500	0.484	0.049	0.018	0.052	2.041
40	270	-60	276.6	12.5	13.833	21.544	0.529	0.227	0.062	0.236	0.531
41	240	-60	247.4	14.0	13.793	21.576	0.497	0.187	0.094	0.210	1.304
42	210	-60	218.4	15.9	13.628	21.638	0.379	0.022	0.156	0.157	8.075
43	180	-60	189.7	18.4	13.783	21.417	0.490	0.177	0.065	0.188	1.701
44	150	-60	161.6	21.8	13.664	21.532	0.520	0.058	0.050	0.076	0.014
45	120	-60	134.2	26.6	13.629	21.574	0.465	0.023	0.092	0.095	3.149
46	270	-90	284.6	18.4	13.690	21.781	0.370	0.084	0.299	0.311	8.595
47	240	-90	256.3	20.6	13.681	21.689	0.406	0.075	0.207	0.221	6.530
48	210	-90	228.5	23.2	13.781	21.546	0.519	0.175	0.064	0.186	0.029
49	180	-90	201.2	26.6	14.898	20.762	1.490	1.291	0.720	1.478	55.563
50	270	-120	295.5	24.0	14.858	20.908	1.393	1.252	0.574	1.377	50.042
51	240	-120	268.3	26.6	13.655	21.647	0.423	0.049	0.165	0.173	5.574
52	210	-120	241.9	29.7	13.680	21.677	0.422	0.074	0.196	0.209	5.595
z								0.175	0.191	0.287	8.557
$\sigma(z)$								0.250	0.250	0.331	12.924

Tab. 12.7: Tabulka měření přesnosti odhadu optimální pozice kamerou s neměnným natočením u štítu s pozicí $x = 14.91$ m, $y = 22.23$ m a natočením $\Phi = 0.52$ rad

i	x_{cam} [cm]	y_{cam} [cm]	d_{cam} [cm]	γ [°]	x [m]	y [m]	ϕ [rad]	$ \Delta x $ [m]	$ \Delta y $ [m]	$ \Delta d $ [m]	$ \Delta \phi $ [°]
1	270	120	295.5	24.0	13.628	21.850	0.411	0.022	0.368	0.369	6.221
2	270	120	295.5	24.0	13.628	21.850	0.411	0.022	0.368	0.369	6.221
3	240	120	268.3	26.6	13.529	21.653	0.634	0.077	0.171	0.188	6.514
4	210	120	241.9	29.7	13.614	21.609	0.709	0.008	0.127	0.127	10.847
5	180	120	216.3	33.7	13.664	21.433	0.690	0.058	0.049	0.075	9.751
6	150	120	192.1	38.7	13.491	21.421	0.623	0.116	0.060	0.130	5.882
7	120	120	169.7	45.0	13.466	21.422	0.592	0.140	0.060	0.152	4.126
8	90	120	150.0	53.1	13.506	21.370	0.624	0.100	0.112	0.150	5.940
9	60	120	134.2	63.4	13.507	21.350	0.604	0.099	0.132	0.165	4.803
10	30	120	123.7	76.0	13.538	21.816	0.543	0.068	0.335	0.342	1.331
11	270	90	284.6	18.4	13.648	21.532	0.695	0.042	0.050	0.065	10.035
12	240	90	256.3	20.6	13.516	21.547	0.638	0.090	0.065	0.111	6.773
13	210	90	228.5	23.2	13.484	21.523	0.648	0.122	0.042	0.129	7.319
14	180	90	201.2	26.6	13.587	21.456	0.673	0.019	0.026	0.032	8.751
15	150	90	174.9	31.0	13.595	21.365	0.665	0.011	0.117	0.117	8.289
16	120	90	150.0	36.9	13.581	21.342	0.660	0.025	0.139	0.142	8.048
17	90	90	127.3	45.0	13.594	21.329	0.650	0.012	0.152	0.153	7.462
18	60	90	108.2	56.3	13.408	21.947	1.001	0.198	0.465	0.505	27.560
19	270	60	276.6	12.5	13.876	21.384	0.761	0.270	0.098	0.287	13.811
20	240	60	247.4	14.0	13.498	21.669	0.449	0.108	0.187	0.216	4.072
21	210	60	218.4	15.9	13.586	21.495	0.571	0.020	0.014	0.024	2.908
22	180	60	189.7	18.4	13.826	21.238	0.799	0.220	0.244	0.328	15.980
23	150	60	161.6	21.8	13.697	21.346	0.665	0.091	0.136	0.163	8.294
24	120	60	134.2	26.6	13.697	21.320	0.666	0.091	0.162	0.186	8.338
25	90	60	108.2	33.7	13.715	21.279	0.672	0.109	0.203	0.230	8.701
26	60	60	84.9	45.0	13.724	21.278	0.641	0.118	0.203	0.235	6.905
27	30	60	67.1	63.4	13.689	21.647	0.488	0.083	0.166	0.186	1.824
28	270	30	271.7	6.3	13.868	21.458	0.734	0.262	0.024	0.263	12.267
29	240	30	241.9	7.1	13.939	21.300	0.733	0.333	0.182	0.380	12.206
30	210	30	212.1	8.1	13.500	21.572	0.547	0.106	0.090	0.139	1.554
31	180	30	182.5	9.5	13.612	21.424	0.607	0.006	0.057	0.058	4.959
32	150	30	153.0	11.3	13.659	21.349	0.643	0.053	0.132	0.142	7.044
33	120	30	123.7	14.0	13.635	21.347	0.636	0.029	0.135	0.138	6.650
34	90	30	94.9	18.4	13.702	21.295	0.651	0.096	0.187	0.211	7.489
35	60	30	67.1	26.6	13.647	21.318	0.634	0.041	0.164	0.169	6.525
36	30	30	42.4	45.0	14.164	21.108	1.031	0.558	0.374	0.672	29.251
37	270	0	270.0	0.0	14.164	21.108	1.031	0.558	0.374	0.672	29.251
38	270	0	270.0	0.0	13.694	21.557	0.573	0.088	0.075	0.116	3.017
39	240	0	240.0	0.0	13.727	21.419	0.596	0.120	0.062	0.136	4.363
40	210	0	210.0	0.0	13.872	21.214	0.797	0.266	0.267	0.377	15.896
41	180	0	180.0	0.0	13.489	21.757	0.413	0.117	0.276	0.300	6.116
42	150	0	150.0	0.0	13.672	21.434	0.604	0.066	0.048	0.081	4.819
43	120	0	120.0	0.0	13.775	21.283	0.731	0.168	0.199	0.261	12.078
44	90	0	90.0	0.0	13.680	21.394	0.632	0.074	0.087	0.115	6.421
45	60	0	60.0	0.0	13.717	21.363	0.635	0.111	0.118	0.162	6.593
46	30	0	30.0	0.0	13.954	21.204	0.762	0.348	0.277	0.445	13.849
47	270	-30	271.7	6.3	13.670	21.622	0.398	0.064	0.140	0.154	7.003
48	240	-30	241.9	7.1	13.604	21.839	0.131	0.002	0.357	0.357	22.299
49	210	-30	212.1	8.1	13.765	21.314	0.401	0.159	0.167	0.231	6.827
50	180	-30	182.5	9.5	13.583	21.685	0.328	0.023	0.203	0.205	11.012
51	150	-30	153.0	11.3	13.622	21.563	0.345	0.016	0.081	0.083	10.049
52	120	-30	123.7	14.0	13.623	21.525	0.457	0.017	0.044	0.047	3.590
53	90	-30	94.9	18.4	13.634	21.495	0.472	0.028	0.013	0.031	2.728
54	60	-30	67.1	26.6	13.622	21.529	0.472	0.016	0.048	0.050	2.775
55	30	-30	42.4	45.0	13.467	22.560	-0.157	0.139	1.078	1.087	38.806
56	270	-60	276.6	12.5	13.671	21.854	0.315	0.065	0.372	0.378	11.739
57	240	-60	247.4	14.0	13.753	21.409	0.580	0.147	0.073	0.164	3.417
58	210	-60	218.4	15.9	13.691	21.399	0.511	0.085	0.083	0.119	0.544
59	180	-60	189.7	18.4	13.657	21.403	0.538	0.051	0.079	0.094	1.028
60	150	-60	161.6	21.8	13.671	21.529	0.550	0.064	0.048	0.080	1.711
61	120	-60	134.2	26.6	13.612	21.510	0.474	0.006	0.028	0.028	2.653
62	90	-60	108.2	33.7	13.596	21.503	0.479	0.010	0.021	0.023	2.348
63	60	-60	84.9	45.0	13.599	21.511	0.484	0.007	0.029	0.030	2.042
64	30	-60	67.1	63.4	13.877	21.375	0.546	0.271	0.106	0.291	1.509
65	270	-90	284.6	18.4	13.853	21.248	0.564	0.247	0.234	0.340	2.524
66	240	-90	256.3	20.6	13.712	21.551	0.442	0.106	0.069	0.127	4.467
67	210	-90	228.5	23.2	13.726	21.460	0.491	0.120	0.022	0.122	1.669
68	180	-90	201.2	26.6	13.672	21.423	0.514	0.066	0.059	0.088	0.318
69	150	-90	174.9	31.0	13.655	21.515	0.495	0.049	0.033	0.059	1.415
70	120	-90	150.0	36.9	13.632	21.476	0.486	0.026	0.006	0.026	1.934
71	90	-90	127.3	45.0	13.609	21.468	0.491	0.003	0.014	0.014	1.670
72	60	-90	108.2	56.3	13.773	22.441	-0.202	0.167	0.960	0.974	41.356
73	270	-120	295.5	24.0	13.806	21.343	0.547	0.200	0.138	0.243	1.541
74	240	-120	268.3	26.6	13.755	21.446	0.471	0.149	0.036	0.153	2.836
75	210	-120	241.9	29.7	13.707	21.502	0.497	0.101	0.020	0.103	1.325
76	180	-120	216.3	33.7	13.692	21.398	0.521	0.086	0.084	0.120	0.044
77	150	-120	192.1	38.7	13.630	21.419	0.504	0.024	0.063	0.067	0.902
78	120	-120	169.7	45.0	13.611	21.428	0.499	0.005	0.054	0.054	1.221
79	90	-120	150.0	53.1	13.582	21.447	0.499	0.024	0.034	0.042	1.201
$\bar{z}$								0.105	0.154	0.203	7.640
$\sigma(z)$								0.109	0.177	0.192	8.066

Tab. 12.8: Tabulka měření přesnosti odhadu optimální pozice s aktivním natáčením kamery u štítu s pozicí $x = 14.91$ m, $y = 22.23$ m a natočením $\Phi = 0.52$ rad

i	$\phi = 0^\circ$	$\phi = 45^\circ$	$\phi = 68^\circ$	$\phi = 90^\circ$	$\phi = 120^\circ$	$\phi = 145^\circ$	$\phi = 170^\circ$
1	1	43	68	90	117	143	168
2	0	43	68	87	118	142	171
3	0	43	68	88	118	144	172
4	0	43	69	88	117	145	167
5	0	43	69	87	118	143	170
6	0	43	69	88	117	143	167
7	0	44	68	89	118	145	168
8	0	43	69	88	119	145	172
9	0	43	69	89	118	144	170
10	0	43	68	89	120	144	167
11	0	43	68	88	118	145	168
12	0	43	68	88	117	145	167
13	0	44	68	88	119	144	168
14	1	44	68	87	117	143	172
15	0	44	68	86	117	145	167
ϕ	0,1	43,3	68,3	88,0	117,9	144,0	168,9
$\sigma(\phi)$	0,35	0,46	0,49	1,00	0,92	1,00	2,02
$ \Delta(\phi) $	0,1	1,7	1,7	2,0	2,1	1,0	1,1

Tab. 12.9: Tabulka měření přesnosti dosažení natočení serva do různých poloh ϕ