VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

SW PRO SIMULACI AUTONOMNÍCH ROBOTŮ

SOFTWARE FOR THE SIMULATION OF AUTONOMOUS ROBOTS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — MARTIN ŠŤASTNÝ
AUTHOR

VEDOUCÍ PRÁCE — ING. DANIEL ZUTH, PH.D.
SUPERVISOR

BRNO 2013

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky
Akademický rok: 2012/2013

ZADÁNÍ BAKALÁRSKÉ PRÁCE

student(ka): Martin Šťastný

který/která studuje v **bakalářském studijním programu**

obor: **Aplikovaná informatika a řízení (3902R001)**

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

SW pro simulaci autonomních robotu

v anglickém jazyce:

Software for the simulation of autonomous robot

Stručná charakteristika problematiky úkolu:

Práce bude obsahovat rešerši na téma SW pro simulaci autonomních robotů se zaměřením na opensource projekty. Poté si student vybere vhodný SW a předvede v něm ukázku robotické úlohy jako je například sledování čáry, vyhnutí se překážce a podobně. Práce je součástí řešení projektu IGA VUT Brno, FSI-S-11-31, Aplikace metod umělé inteligence.

Cíle bakalářské práce:

- Provést rešerši v oblasti SW pro simulaci autonomních robotů
- Popsat a porovnat jednotlivé SW
- Vybrat vhodný SW pro demonstraci ukázky
- Demonstrovat ukázku robotické úlohy ve vybraném SW

Seznam odborné literatury:
- NOVÁK, Petr. Mobilní roboty: pohony, senzory, řízení. Vyd. 1. Praha: BEN - technická literatura, 2004, 247 s. ISBN 80-730-0141-1.
- ZELINKA, Ivan. Umělá inteligence: v problémech globální optimalizace. Vyd. 1. Praha: BEN, 2002, 189 s. ISBN 80-730-0069-5.

Vedoucí bakalářské práce: Ing. Daniel Zuth, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2012/2013.

V Brně, dne 21.11.2012

.....	.....
Ing. Jan Roupec, Ph.D. Ředitel ústavu	prof. RNDr. Miroslav Doupovec, CSc., dr. h. c. Děkan fakulty

ABSTRAKT

Cílem této bakalářské práce je provést rešerši v oblasti SW pro simulaci autonomních robotů. V úvodu jsou vysvětleny obecné pojmy, týkající se robotické simulace. Dále jsou obecně vysvětleny výhody použití robotických simulátorů.

V první části rešerše jsou popsány vybrané robotické simulátory. Druhá část rešerše spočívá v porovnání jednotlivých robotických simulátorů z hlediska vhodnosti pro simulaci robotických úloh. Po porovnání je vybrán nejvhodnější robotický simulátor, ve kterém je následně demonstrována robotická ukázka.

ABSTRACT

Objective of this bachelor work is to make research of SW, which are used for simulation of autonomous robots. At the beginning I explain general concepts of robotic simulation. And I also explain advantages of usage of robotic simulators.

In the first part of my research are described some chosen robotic simulators. The second part of my research compares robotic simulators, which is more suitable for simulations of robotic operations. In the end of my research the most suitable simulator is taken and used for simple robotic demonstration.

KLÍČOVÁ SLOVA

Simulace, Robotický simulátor, SW pro simulaci, Sledování čáry, RoKiSim, RoboLogix, MobotSim, Breve, V-REP, Webots

KEY WORDS

Simulation, Robotic simulator, Software for the simulation, Line following, RoKiSim, RoboLogix, MobotSim, V-REP, Webots

PROHLÁŠENÍ O ORIGINALITĚ

Tímto prohlašuji, že tato předkládaná bakalářská práce je vypracována mnou samostatně a všechny zdroje, prameny a literaturu, které jsem při vypracování použil, nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

.....

Martin Šťastný, Brno, 2013

BIBLIOGRAFICKÁ CITACE

ŠŤASTNÝ, M. *SW pro simulaci autonomních robotů*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2013. 55 s. Vedoucí bakalářské práce Ing. Daniel Zuth, Ph.D.

PODĚKOVÁNÍ

Tímto bych chtěl poděkovat svému vedoucímu bakalářské práce Ing. Danielovi Zuthovi, Ph.D. za jeho cenné rady a za celkovou pomoc při realizaci práce.

Obsah

ABSTRAKT 5
KLÍČOVÁ SLOVA 5
PROHLÁŠENÍ O ORIGINALITĚ 7
BIBLIOGRAFICKÁ CITACE 7
PODĚKOVÁNÍ 9
1 ÚVOD 13
2 SIMULACE 15
2.1 Počítačová simulace 15
3 SW PRO SIMULACI 16
3.1 Robotické simulátory 16
3.2 Druhy robotických simulátorů 16
4 SW PRO ROBOTICKÉ RAMENO 16
4.1 RoKiSim - Robot Kinematics Simulator 17
4.1.1 Popis simulátoru RoKiSim 17
4.1.2 RoKiSim simulace 19
4.2 RoboLogix 20
4.2.1 Popis simulátoru RoboLogix 21
4.2.2 RoboLogix simulace 23
5 SW PRO MOBILNÍ ROBOTY 26
5.1 MobotSim 27
5.1.1 Popis simulátoru MobotSim 27
5.1.2 MobotSim simulace 28
5.2 Breve 30
5.2.1 Popis simulátoru Breve 30
5.2.2 Breve simulace 31
5.3 V-REP 32
5.3.1 Popis simulátoru V-REP 33
5.3.2 V-REP simulace 36
5.4 Webots 38

5.4.1 Popis simulátoru Webots 38
5.4.2 Webots simulace 40
6 POROVNÁNÍ SW 41
6.1 Vybrání vhodného SW pro demonstraci robotické úlohy 43
7 ROBOTICKÁ UKÁZKA VE V-REPU 44
7.1 Sledování vodící čáry 44
7.2 Demonstrace robotické úlohy: sledování čáry s vyhýbáním překážek 44
7.2.1 Vymodelování prostředí 45
7.2.2 Nastavení robota 48
7.2.3 Simulace robotické úlohy 51
8 ZÁVĚR 52
SEZNAM CITOVANÝCH ZDORJŮ 53
SEZNAM ODBORNÉ LITERATURY 56
SEZNAM PŘÍLOH NA CD 56

1 ÚVOD

Robotika je věda o robotech, jejich designu a vývoji jejich aplikací. Tato věda přímo souvisí s elektronikou, mechanikou a softwarem. Jelikož se v dnešní době robotika významně rozšiřuje, používá se takřka ve všech oborech od strojnictví, přes lékařství až k vojenství, vyvíjejí se i robotické simulátory, které jsou nedílnou součástí robotiky. Robotické simulátory mají za úkol simulovat reálné prostředí, ve kterém roboti pracují, simulovat pracovní úkony, které mají daní roboti na starost či krajní situace, které mohou nastat.

Použití robotických simulátorů je v praxi velmi důležité. Například u robotických manipulátorů je důležité si nejdříve v simulaci ukázat dráhu nástroje, dříve než reálně zničí výrobek/polotvar/nástroj. U autonomních robotů jsou simulátory důležité z hlediska optimalizace, je to tím, že v simulaci můžeme odzkoušet mnoho různých algoritmů a následně vybrat ten nejlepší (optimální). V simulaci můžeme testovat i zdroj robota, což vede k odstranění problému týkající se nedostatečné kapacity baterie.

V simulaci autonomních robotů zjistíme, jak se robot zachová v dané situaci a jak zvládne svojí práci, na kterou je naprogramován.

Autonomní roboti, jsou roboti, kteří mohou vykonávat požadované úkoly v neznámém prostředí bez průběžného vedení lidí. Mnoho druhů robotů má určitý stupeň autonomie. Vysoká míra autonomie je zvlášť žádoucí v oblastech, jako jsou výzkum vesmíru, čištění podlah, sekání trávníků, čištění odpadních vod, atd. Autonomní roboti stále vyžadují pravidelnou údržbu, stejně jako ostatní stroje.[1]

Ve své práci se budu zabývat robotickými simulátory, které slouží k otestování konstrukce a algoritmu těchto robotů. Dále umožňují odzkoušení určitých situací, které nemohou být "vytvořeny" v reálném světě kvůli nákladům, času nebo "jedinečnosti" daného zdroje. To je velmi užitečné především z hlediska úspory času a financí, jelikož předejdeme opakovanému předělávání konstrukce robota a upravování algoritmu.

Robotických simulátorů je na trhu velké množství, proto se ve své práci budu zabývat porovnáváním a popisováním jednotlivých robotických simulátorů a uvedu jejich výhody i nevýhody. Robotické simulátory budu porovnávat z několika hledisek. Po porovnání vybraných robotických simulátorů vyberu jeden, který je dle mého úsudku nejvhodnější pro vytváření robotických simulací.

Ve vybraném robotickém simulátoru budu demonstrovat jednoduchou robotickou úlohu, jako je sledování čáry, vyhýbání překážek, atd.

Pro lepší pochopení oblasti robotiky jsem čerpal znalosti z odborné literatury NOVÁK P. *Mobilní roboty: pohony, senzory, řízení* a ZELINKA I. *Umělá inteligence: v problémech globální optimalizace*.

2 SIMULACE

„Simulace je napodobení nějaké skutečné věci, stavu nebo procesu. Samotný akt simulace něčeho obecně znamená zobrazení některých klíčových vlastností nebo chování vybraných fyzikálních, nebo abstraktních systémů.“[2]

„Simulace se používá v mnoha souvislostech, zahrnujících modelování přírodních systémů nebo lidských systémů s cílem získat poznatky o jejich fungování. Jiné souvislosti zahrnují technologické simulace pro optimalizaci výkonu, bezpečnostní inženýrství, testování, školení a vzdělávání. Simulace může být použita pro zobrazení případných reálných dopadů alternativních podmínek a způsobů jednání.“[2]

„Klíčové otázky v simulaci zahrnují např. pořízení platných zdrojů informací o příslušném výběru klíčových charakteristik a chování, využití zjednodušujícího odhadu a předpokladů v rámci simulace a věrnost a platnost výsledků dané simulace.“[2]

2.1 Počítačová simulace

„Počítačová simulace je pokusem o vymodelování reálného světa či hypotetické situace za pomoci počítače tak, aby bylo možné studovat tento systém a vysledovat jak funguje. Chování tohoto systému může být pak předpovídáno měněním proměnných.“[3]

„Počítačové simulace se staly užitečným nástrojem při modelování mnoha přirozených systémů ve fyzice, chemii a biologii, ale také systémů z oblastí jako je ekonomie a společenské vědy (výpočetní sociologie), stejně jako napomohly proniknout do podstaty fungování systémů ve strojírenství. Simulace jsou důležitou součástí strojírenských systémů nebo jakýchkoliv systémů, které zahrnují více procesů. Například v elektrotechnickém strojírenství mohou systémy simulovat propagační čas a fázovou změnu vyvolanou pásovou výrobou. Většina strojírenských simulací ve výsledku znamená matematické výpočty a počítačový průzkum. Nicméně existuje spousta případů, kdy jsou matematické výpočty nespolehlivé. Simulace problematiky dynamiky tekutin často vyžaduje jak matematické propočty, tak fyzické simulace. V takové situaci je zapotřebí, aby fyzické modely měly jakousi dynamickou podobnost. Fyzikální a chemické simulace mají také často přímé užití, spíše než pouze výzkumný účel.“[3]

„Příkladem prospěšnosti využití počítačů pro simulace je oblast zabývající se simulacemi silničního provozu. V takovýchto simulacích se mění chování modelu v závislosti na nastavení původních parametrů převzatých z reálného prostředí.“[3]

„Tradičně bylo formální modelování systémů vytvářeno přes matematické modely, které se pokoušely nalézt analytická řešení, což umožňovalo předvídat chování systému dle souboru parametrů a počátečních podmínek. Počítačová simulace je často používána jako doplněk nebo náhrada modelování takových systémů, pro které není možné použit analytického řešení uzavřenými vzorci. Existuje mnoho různých druhů počítačových simulací, jejichž společnou vlastností je snaha o vytváření příkladů reprezentativních scénářů modelu, u nějž nelze vytvořit kompletní výčet všech možných stavů či tento výčet není dostupný.“[3]

„Možnost simulovat funkci výrobku nebo technického zařízení a také prověřit jeho chování v různých podmínkách bez nutnosti pořizovat fyzický model značně urychluje a také zlevňuje výzkum a vývoj.“[4]

3 SW PRO SIMULACI

„Simulační programy převádí matematický model (nebo transformovaný matematický model) do tvaru modelu simulačního, který je možno popsat výrazovými prostředky daného simulačního programu nebo pomocí programovacího jazyka.“[5]

V dnešní době existuje mnoho rozmanitých softwarových balíčků. Tyto balíčky jsou určené pro modelování počítačově založených simulací, které umožňují jeho relativně snadnou realizaci.[3]

3.1 Robotické simulátory

Robotické simulátory jsou softwary pro simulaci autonomních robotů. Jsou určeny k tomu, aby vytvořily vestavěné aplikace pro specifické, nebo jakékoli roboty, aniž by byly závislé na fyzickém robotovi. V některých případech je možné tyto aplikace přenést na fyzického robota bez jakýchkoliv modifikací.[6]

Jednou z nejpopulárnějších aplikací robotické simulace je 3D modelování a vykreslování robota a jeho prostředí. Tento typ softwaru má simulovat a vizualizovat pohyby robota ve skutečném světě. Některé simulátory jako např. *Robologix* používají fyzikální motory pro realističtější simulace pohybů robotů. Tyto simulace umožňují odzkoušet a doladit napsaný program (algoritmus) bez toho, aby byl fyzický robot zapnut, tzv. off-line programování. Úspěch simulace záleží na tom, jak moc se podobá reálné prostředí robota, simulovanému prostředí.[7]

3.2 Druhy robotických simulátorů

Robotické simulátory se dají rozdělit dle mnoha kritérií. Např. podle dostupnosti (komerční, opensourceové), podle platformy, pod kterou lze daný simulátor spustit (Windows, Linux, Mac OS), a některé simulátory mohou být pro určitý typ robotů, jako např. pro robotické rameno (koncový manipulátor), které je využíváno hlavně pro pásovou výrobu. Dále mohou být přímo pro specifické modely robotů, jako je např. robotický simulátor Webots NAO, který je určen pouze pro simulaci robotů NAO. V tomto případě lze přenášet simulační aplikace na opravdového robota bez jakékoli modifikace.

4 SW PRO ROBOTICKÉ RAMENO

„Robotická ramena se používají jako průmyslové manipulátory, které jsou dnes hojně využívány v mnoha odvětvích. V dnešní době je lze aplikovat do většiny výrobních procesů. Pro pohyb svého těla využívají většinou 6 os ale i více. Sedmá osa může být např. použita pro přesun po koleji, kdy robot popojíždí vedle výrobku, synchronizován s dopravníkem a po vykonání úlohy se vrací zpět nebo pro sevření kleští při bodovém sváření.“[8]

Použití robotických ramen (průmyslových manipulátorů) ve výrobě:

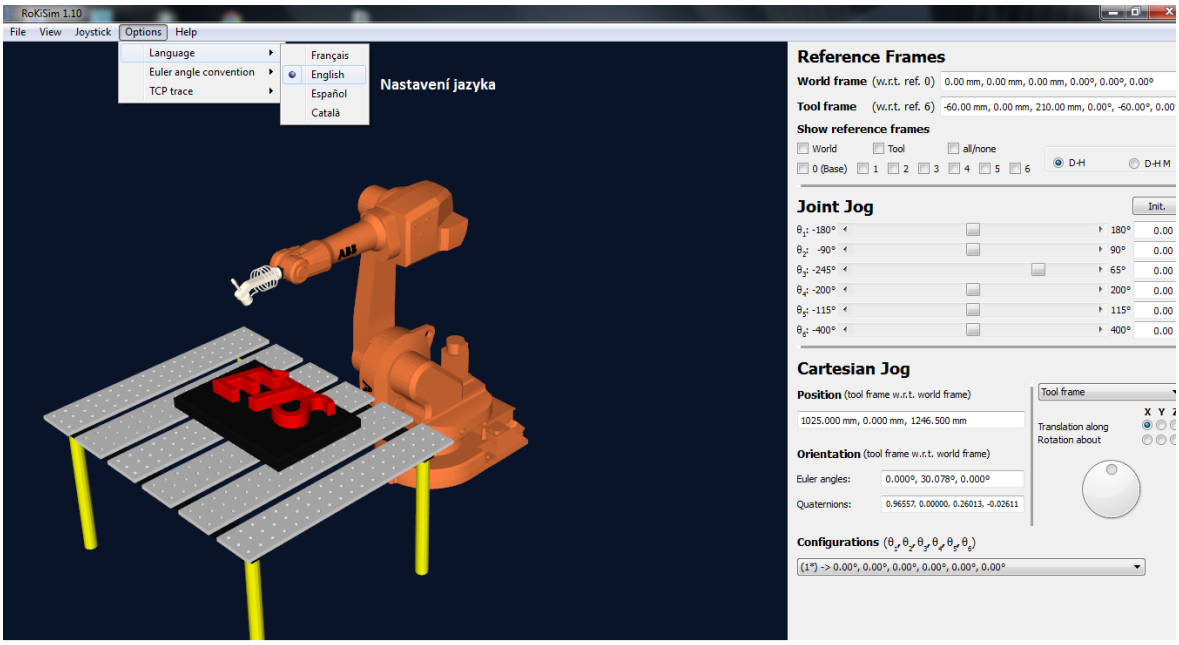
- bodové sváření - svařování karoserie automobilů;
- souvislé sváření;
- montáž - instalace a kompletace;
- aplikace - nanášení lepidel, těsniv;
- lakování;
- manipulace - překládání, nakládání a vykládání pro dopravníky;
- paletizace - skládání nebo vykládání výrobků na paletách;
- kontrola - měření pomocí kamer, laseru a čidel.[9]

4.1 RoKiSim - Robot Kinematics Simulator

RoKiSim je volný vzdělávací software pro 3D simulaci šestiosých robotů typu PUMA (Programmable Universal Manipulation Arm). Je dostupný pro Windows, Linux a Mac OS. Uživatel v něm může manipulovat virtuálním robotem buď pomocí šesti kloubů, nebo pomocí kartézského prostoru. Dokáže zobrazit různé referenční rámce (vzhledem k oběma konvencím) a dokáže promítnout všechny možné konfigurace robota (řešení inverzní kinematiky) pro danou polohu koncového efektoru.[10]

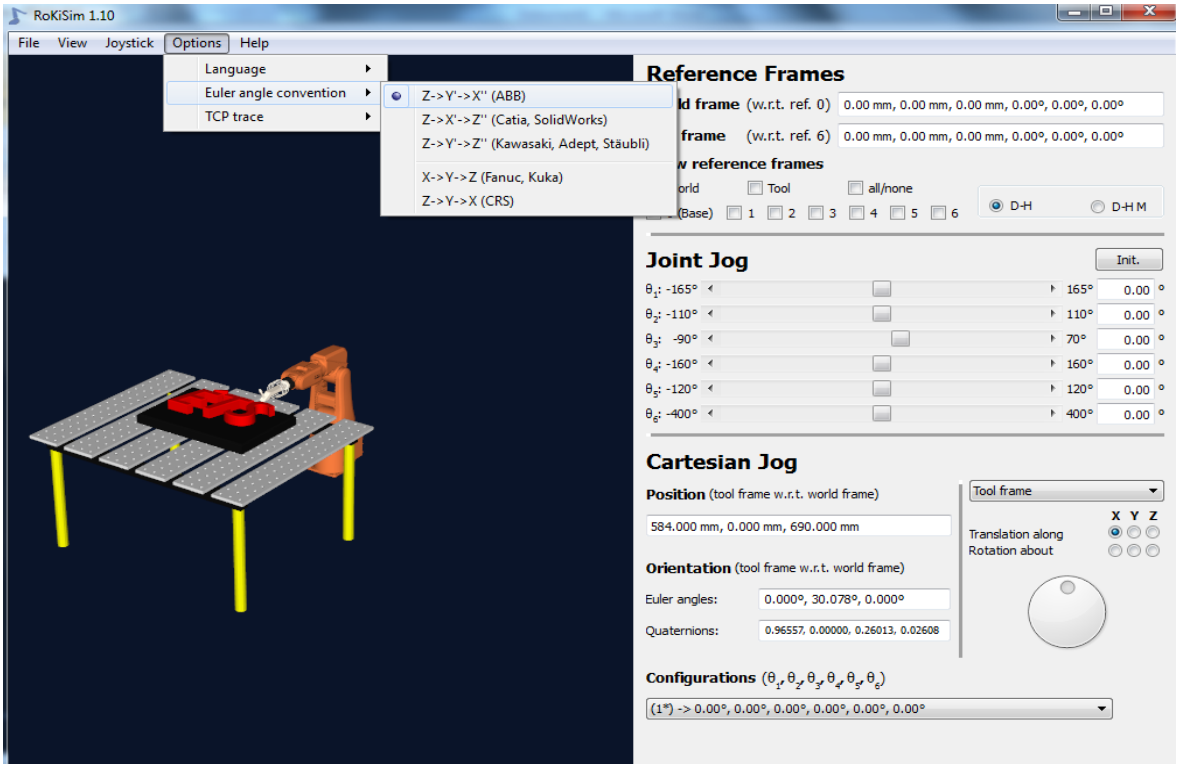
4.1.1 Popis simulátoru RoKiSim

RoKiSim menu může být zobrazeno ve čtyřech různých jazycích (angličtině, francouzštině, španělštině a katalánštině).



Obr. 1 Prostředí RoKiSim, nastavení jazyka.

Orientace robota může být prezentována několika konvencemi Eulerových úhlů jako například jednotkou čtveřic (používanou ABB robotics).[10]

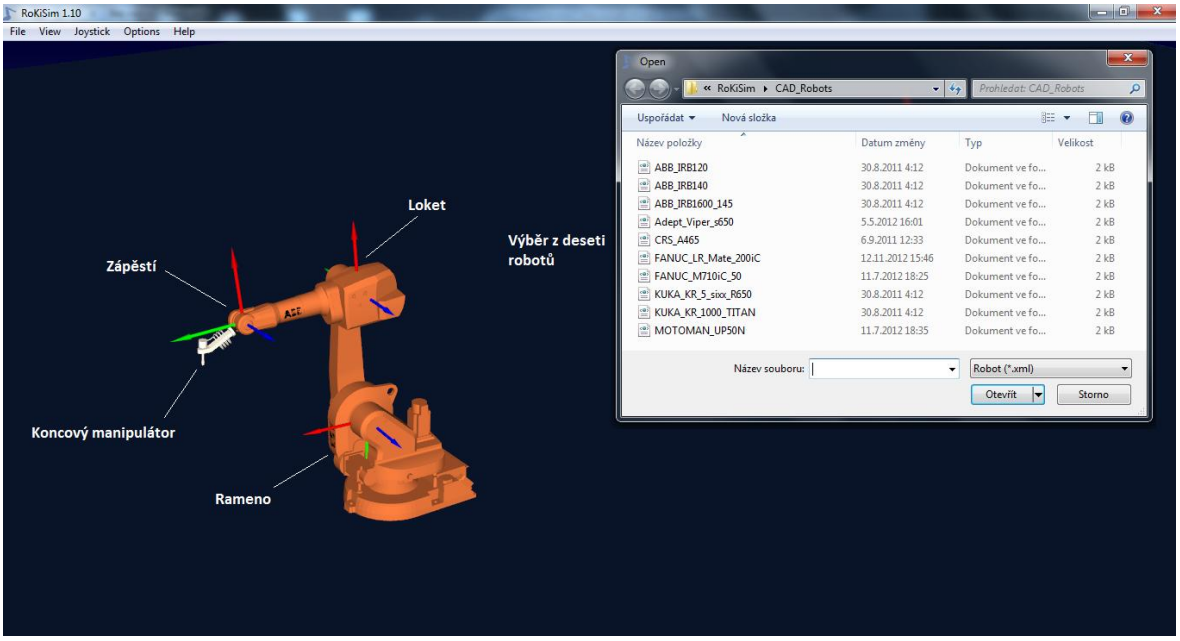


Obr. 2 Nastavení Eulerových úhlů.

Balení RoKiSim přichází s deseti oblíbenými průmyslovými modely robotů (ABB IRB 120, ABB IRB 140, ABB IRB 1600/1.45, Adept Viper S650, CRC A465, FANUC LR Mate 200iC, FANUC M-710iC/50, KUKA KR 5 sixx R650, KUKA KR 1000 TITAN and Motoman UP50N) stejně tak jako s šesti koncovými manipulátory. Poměrně jednoduše se dá přidat nový model robota nebo nový model koncového manipulátoru (v ASCII STL formátu). Balíček také přichází se simulací tří částí robota (rameno, loket, zápěstí). Simulace obsahuje soubor ASCII s koncovkou *.sim obsahující sekvence číselných hodnot pro šest kloubních proměnných a další soubor ASCI, který specifikuje robota a jeho pomůcky.[10]

Tento software pro simulaci robotů také přichází se schopností importovat geometrii objektů a vložit je do prostředí robota. Objekty musejí být v souboru ASCII STL nebo SPL.[10]

Pro načtení jakéhokoli robota klikneme na políčko *File* a dále na políčko *Load a robot*, či *Load a tool* pro načtení koncového manipulátoru. Pomocí *Load object* můžeme do prostředí RoKiSim načíst různé objekty. Také můžeme načíst některou z předdefinovaných jednoduchých simulací, které RoKiSim nabízí.

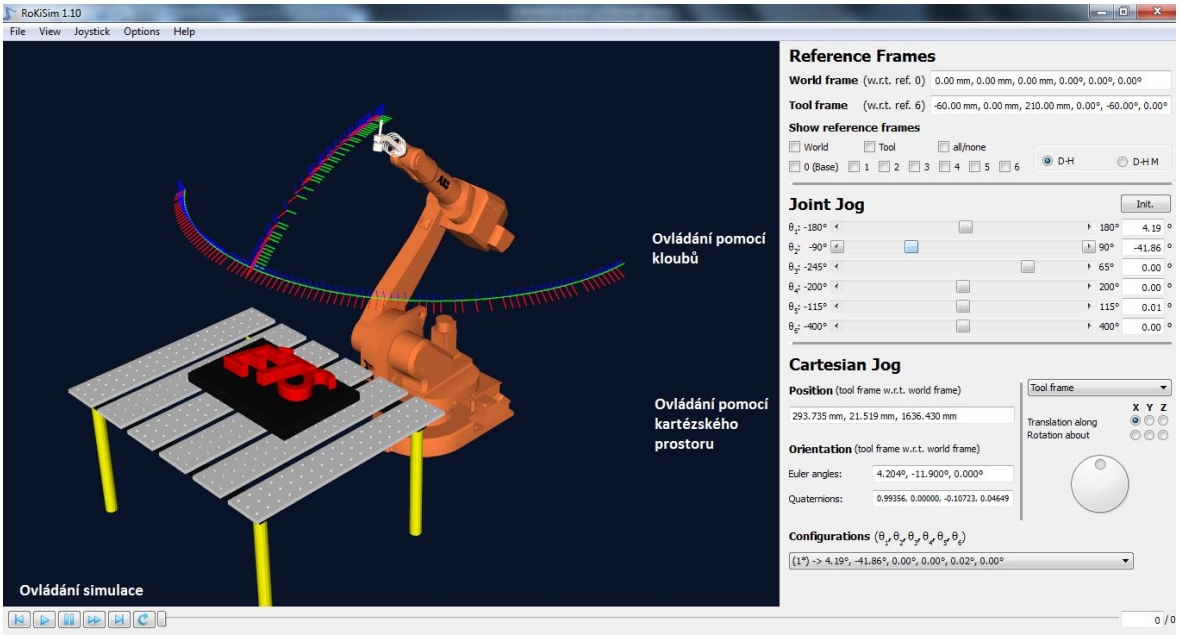


Obr. 3 Ukázka robotů.

RoKiSim neumožňuje zachytávání obrázků přímo z obrazovky, ale když maximalizujeme jeho okno a minimalizujeme nepotřebné panely, stisknutím Alt+PrtScn, můžeme vložit obrázek do oblíbeného grafického programu nebo přímo do programů jako Microsoft Word nebo PowerPoint. Ve Windows 7 můžeme použít Snipping tools pro vytvoření screenshotu, vyjmutí obdélníkové oblasti nebo oblasti vlastní specifikace.[11]

4.1.2 RoKiSim simulace

Nejdříve si nastavíme model robota, konvenci Eulerových úhlů, jazyk, koncový manipulátor, objekty a uložíme vše jako výchozí nastavení v souboru menu.



Obr. 4 Ovládání simulace.

Pro načtení simulace stiskneme Ctrl + D. Potom použijeme simulační panel v dolní části okna RoKiSim nebo následující zkratky:

- ↓ Spustí nahranou simulaci.
- ↑ Zastaví simulaci a nastaví robota do výchozího nastavení.
- → Zastaví simulaci a krokuje vpřed.
- ← Zastaví simulaci a krokuje vzad.

4.2 RoboLogix

RoboLogix je simulační software, který je vytvořen pro napodobování skutečného světa. S RoboLogixem se učíte, zkoušíte, vytváříte programy, které jste napsaly pro pěti-osé průmyslové roboty se širokou škálou praktických využití. Je dostupný pro MAC OS, Linux a Windows. Primární zaměření programu jsou automatizované výrobní procesy, které zahrnují paletizaci, svařování, lakování, provoz "vezmi a polož" a dovolují přizpůsobení prostředí, takže si můžete vytvořit své vlastní aplikace. S RoboLogixem může uživatel rozběhnout simulátor na testování a vyzkoušení programu robota a kontrolu algoritmů, zatímco se vykreslují okamžitá zrychlení, rychlosti a polohy.[12]

Tento robotický simulátor je vhodný pro studenty stejně jako pro robotické designéry a inženýry. Simulační software umožňuje ověření dosažitelnosti cestovních rozsahů a kolizí. To umožňuje zvýšení spolehlivosti procesu plánování, rozvoje programu a snížení celkového času na dokončení úkonu. RoboLogix umožňuje programátorům psát své vlastní programy, upravovat prostředí a používat dostupné senzory. Tyto senzory zahrnují videokamery, které se používají pro získání požadované pozice robotického ramena.

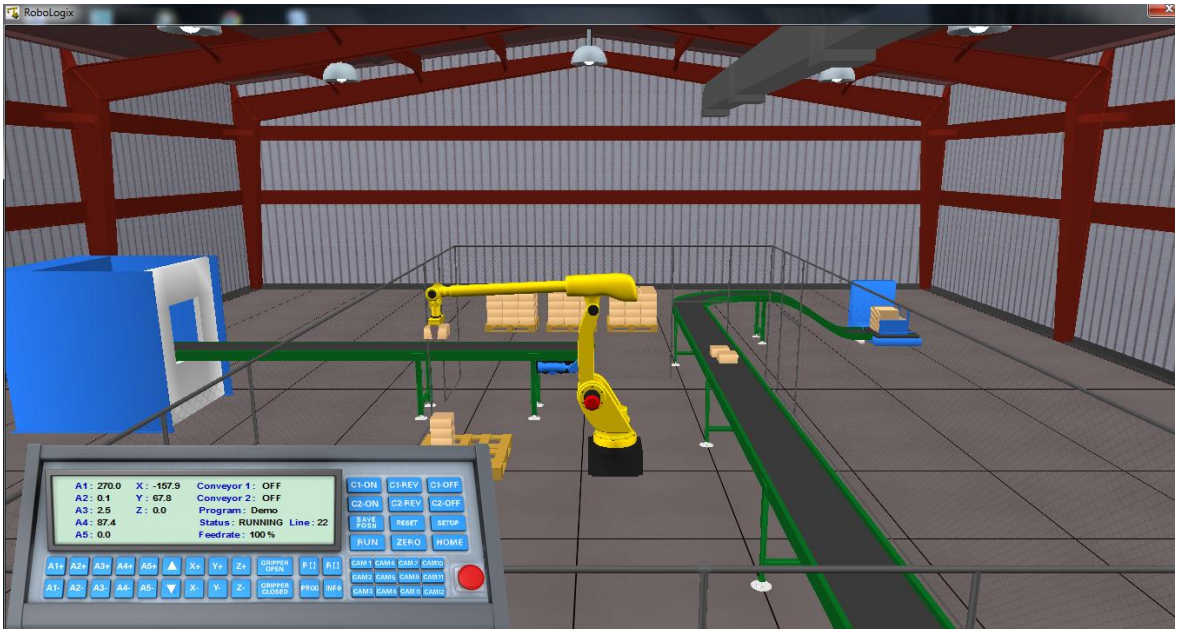
Kromě toho, RoboLogix obsahuje ruční ovládací panel, který umožňuje uživateli dávat robotovi příkazy a potom jej vrátit do výchozí pozice.[12]

Ve virtuálním prostředí RoboLogix jsou zobrazeny principy 3D simulace jak modelování geometrie, tak modelování kinematiky.[13]

Obr. 5 Prostředí RoboLogix.

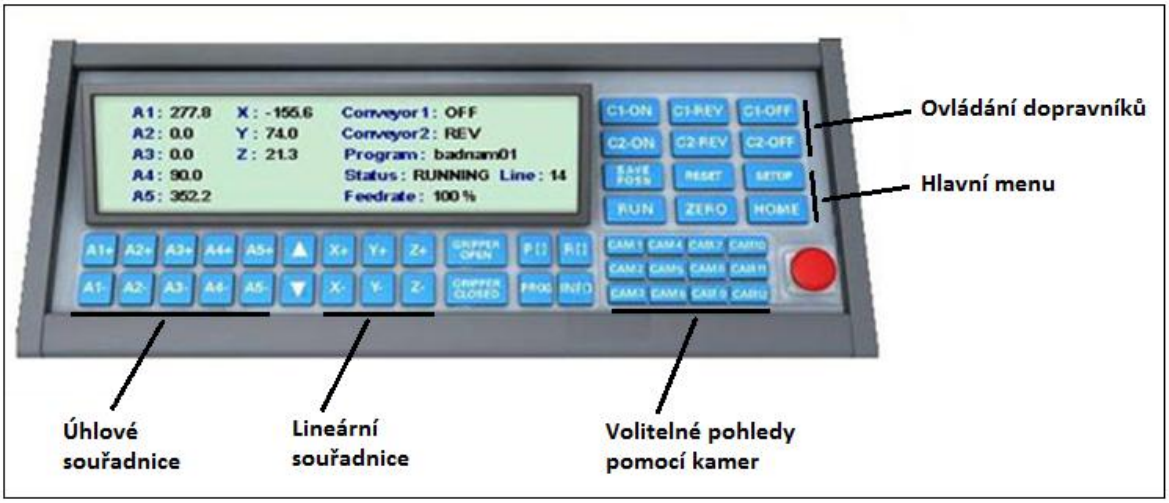
4.2.1 Popis simulátoru RoboLogix

Výchozí pracovní prostor se skládá ze dvou dopravníků, obrobků (krabicí), palet, ovládacího panelu a robotického ramena.



Obr. 6 Výchozí pracovní prostor.

RoboLogix poskytuje 12 pohledů na daný pracovní prostor robota. Tyto pohledy jsou přístupné pomocí dvanácti tlačítek *CAM* a umožňují sledování z různých úhlů a perspektiv. Pomocí těchto kamer se může uživatel pohybovat v 3D animovaném prostředí v podstatě stejným způsobem jako v reálném prostředí. Při programování robota vám kamerové pohledy umožní provádět jemné úpravy robotického ramena nebo koncového manipulátoru. Tyto pohledy umožňují v reálném čase vizualizovat, jak se robotické rameno přibližuje k objektům.[13]

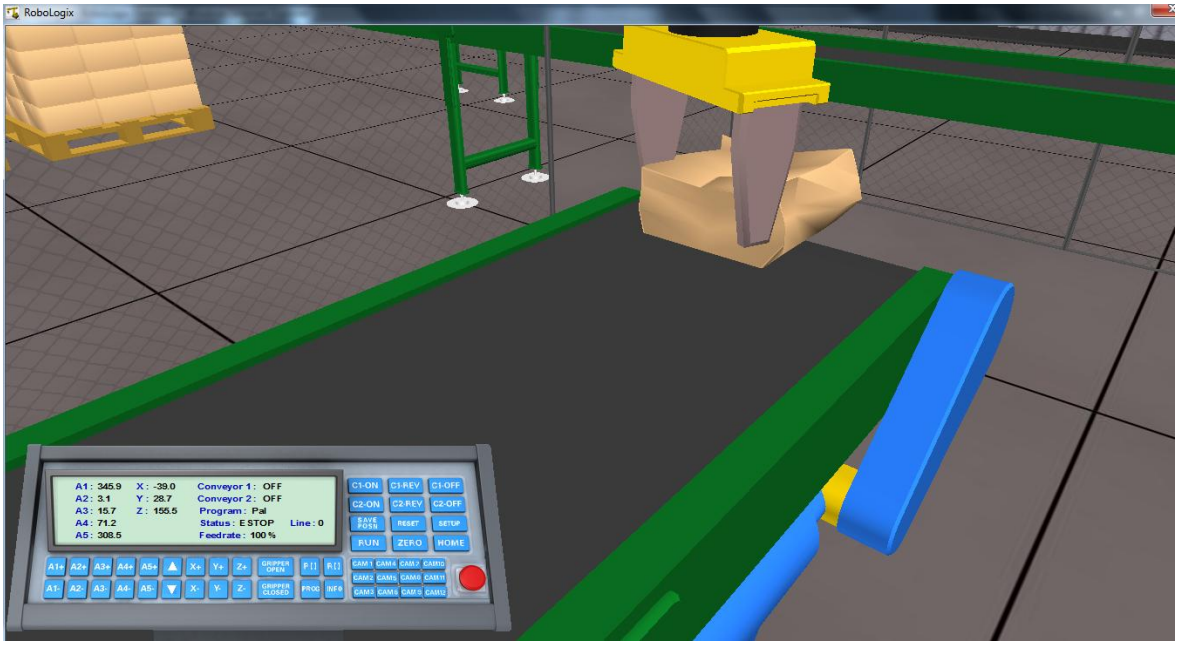


Obr. 7 Ovládací panel RoboLogix.

Ovládací panel se skládá ze dvou řídících funkcí, z funkce kontroly prostředí a z funkce kontroly dopravního systému, která může pohybovat dopravníkem dopředu nebo dozadu. Pohyb robota zajišťují úhlové souřadnice *A1-A5* a lineární souřadnice *X,Y,Z*. Úhlové souřadnice jsou používány pro velké pohyby (přesuny) a lineární souřadnice jsou používány pro menší (jemné) kroky.[14]

V hlavním menu ovládacího panelu se nachází příkazy *RESET*, *HOME*, *SETUP* a *ZERO*, které se používají pro automatické nastavení robota. Funkce *RESET* se používá k obnovení systému. Funkce *HOME* vrátí robota do základní polohy a funkce *ZERO* vynuluje nastavené souřadnice robotického ramena. Funkce *SETUP* umožňuje přístup k dialogovému oknu, ve kterém se nastavují rychlosti dopravníků, úhlových posuvů (deg/sec) a lineárních posuvů (cm/sec).[14]

Programovací prostředí RoboLogix je zcela bezpečné, ale poskytuje velmi realistickou simulaci řídících systémů s využitím robotického zařízení. Jakmile se obrobek poškodí vlivem nárazu, pádu, nebo vlivem nesprávné pozice koncového manipulátoru, bude obrobek označen jako poškozený. Takový obrobek může být vymazán nebo odstraněn z pracovního prostoru stisknutím tlačítka *RESET*.[13]



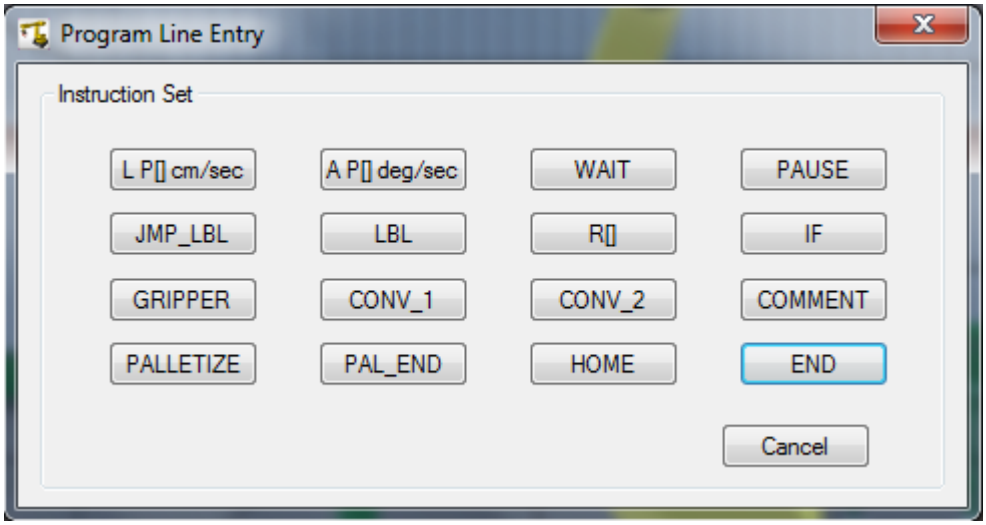
Obr. 8 Poškozený obrobek.

4.2.2 RoboLogix simulace

RoboLogix používá skriptovací programovací jazyk k ovládání softwarových aplikací a využívá instrukční sadu, která je běžná u velkých výrobců robotů. RoboLogix programy se skládají z datových objektů a toku programu. Datové objekty jsou umístěny v registrech a tok programu představuje seznam instrukcí k programování robota.[14]

Programovací jazyky jsou obecně určeny pro vytváření datových struktur a algoritmů od základů, zatímco skriptovací jazyky jsou určeny spíše pro spojení komponent a instrukcí dohromady, proto má RoboLogix instrukční sadu, kde je seznam programových příkazů, které se používají pro zjednodušení procesu programování a poskytnou tak rychlý vývoj aplikací. [14]

Instrukční sada obsahuje 16 příkazů, kterými se udávají robotům pokyny pro plnění úkolů jako je přechod do konkrétní polohy, přesouvání objektů, provádění podprogramu, čekání, atd.[14]



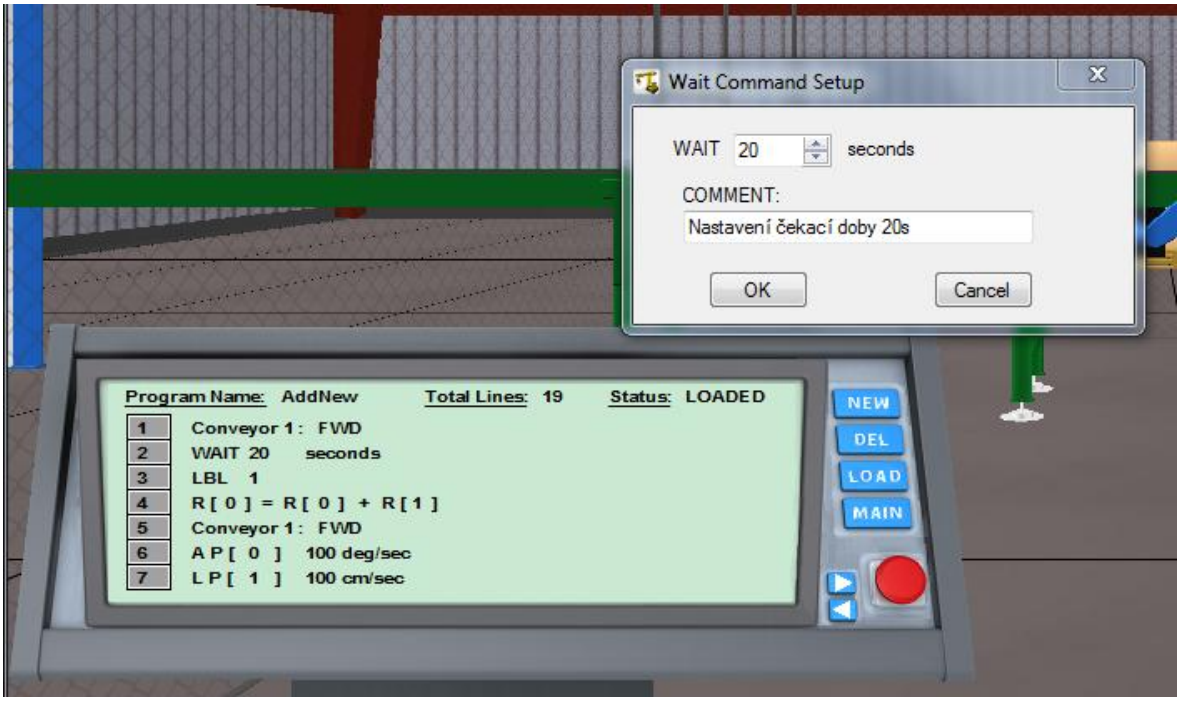
Obr. 9 Instrukční sada RoboLogix.

Instrukční sada Robologix obsahuje následující příkazy:

LP	Instrukce přímočarého pohybu který pohybuje středovým bodem nástroje (TCP) po přímce do nové polohy dané osami (X,Y,Z).
AP	Instrukce úhlového pohybu, který hýbe TCP pomocí úhlového posunutí do nové polohy (A1-A5).
R	Vykonává početní operace ve dvou registrech a do třetího registru ukládá výsledky. Výpočetní operace zahrnují : + - * / .
GRIPPER	Řízení koncového manipulátoru (otevřeno/zavřeno).
IF	Porovnává hodnoty umístěny ve dvou registrech pomocí : <, >, <=, >=, =. Pokud jsou podmínky splněny, provede další řádek programu, pokud ne, přeskočí na další řádek.
WAIT	Provádí zpoždění, kde vyčká určenou dobu, než se aktivuje. Tato instrukce vyčká na podmínky nebo časový úsek před tím než se bude pokračovat.

LBL	Umístění v programu, kam proces skáče popořadě, aby mohl být program vykonáván. LBL může být jakýmkoliv místem v programu, které bylo identifikováno (označeno) jako místo určení.
JMP LABEL	Instrukce, která skočí z jistých částí programu a provede začátek programu v místech označených značkou LBL.
PALLETIZE	Poskytuje přístup k dialogovému boxu instrukcí paletizace s daty náležící do sloupců řádků a vrstev (RCL) objektů které jsou paletizovány. Paletizační a depaletizační operace jsou vybrány použitím této instrukce.
PAL_END	Instrukce ukončení paletizace je příkaz vzniklý dokončením paletizační operace.
PAUSE	Při zahájení instrukce se dojde k zastavení programu. Pokud je PAUSE neaktivní, program pokračuje z místa, kde byl pozastaven.
HOME	Vrátí koncový manipulátor do výchozí pozice, do defaultního nebo startovního bodu robota.
END	Ukončí program.
CONV1	Ovládání prvního dopravníku (řídícího). Umožní ovládání dopravníku vpřed a vzad.
CONV2	Ovládání druhého dopravníku. Poskytuje FWD/ REV / OFF. [15]

Při programování umožňuje RoboLogix psát ke každému řádku instrukce komentář, který ulehčí orientování se v kódu a usnadní jeho upravování. Psaní komentářů vede i k lepšímu pochopení fungování programu.



Obr. 10 Napsaný program pomocí instrukční sady.

5 SW PRO MOBILNÍ ROBOTY

SW pro mobilní roboty se zabývá autonomními i vzdáleně řízenými roboty, kteří se mohou pohybovat (na kolech, pásech, vzduchem, atd.). Autonomní robot se sám rozhoduje co udělá (bez vnějšího zásahu), na rozdíl od vzdáleně řízeného robota, který je řízen programem, nebo přímo člověkem).[16]

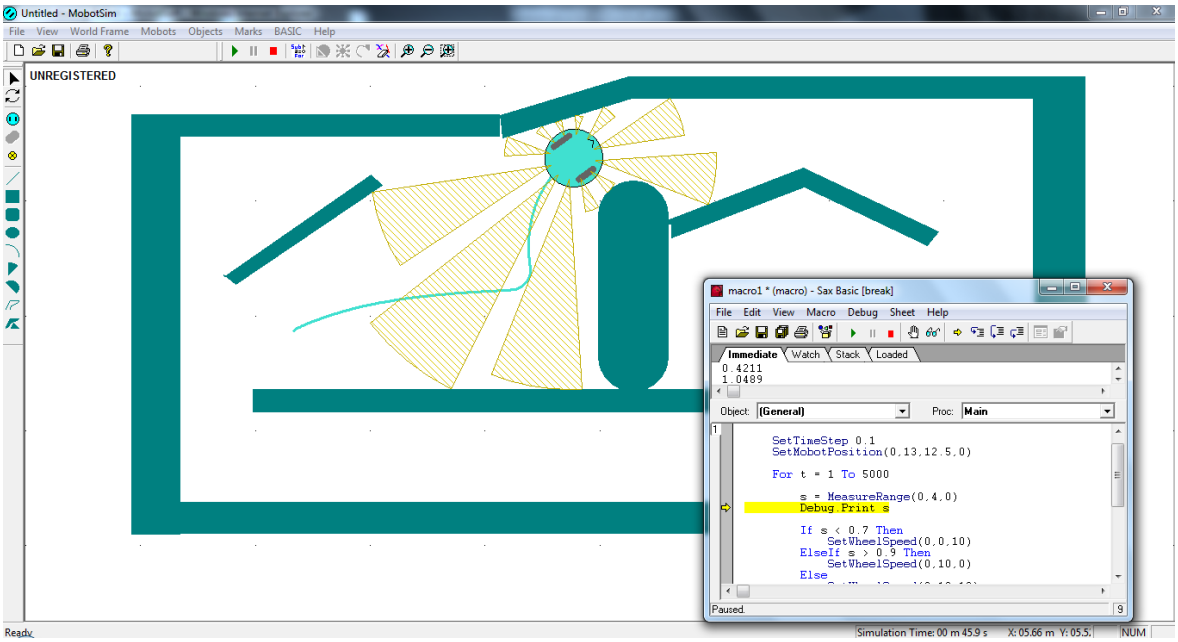
Způsoby navigace mobilních robotů:

- Dead reckoning (odometrie) - matematická procedura pro určování současné pozice vozidla pomocí postupného přičítání díky kurzu a rychlosti v průběhu času.
- Sledování vodící čáry - robot opticky či pomocí magnetometrů (či Hallových sond) sleduje vodící čáry. Je to nejpoužívanější metoda.
- Inerciální navigace - využívá gyroskopů a akcelerometrů pro měření zrychlení a následně tak určuje výslednou pozici.
- GPS
- Navigace pomocí taktilních (dotykových) a proximitních (bezdotykových) senzorů. [17]

5.1 MobotSim

MobotSim je komerční software pro 2D simulaci diferenciálních pohonů mobilních robotů. Poskytuje grafické rozhraní, které představuje prostředí, ve kterém se dají snadno vytvořit, nastavit a upravit roboti a objekty. Simulátor MobotSim je dostupný pro Windows.[18]

MobotSim byl vytvořen pro vědce, studenty a robotické nadšence, kteří chtějí navrhovat, testovat a simulovat mobilní roboty, zkoušet nové techniky jako jsou autonomní navigace, vyhýbání překážek, umělá inteligence, integrace datových senzorů, atd.[18]

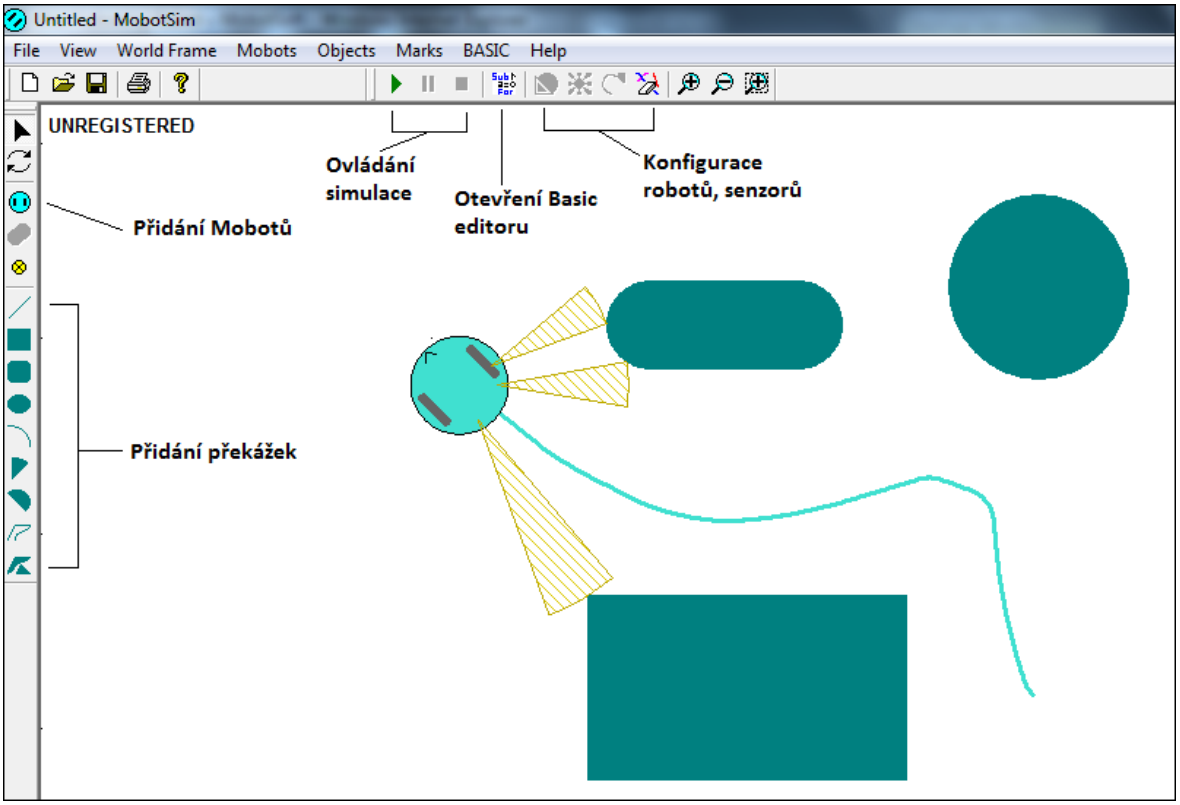


Obr. 11 Prostředí MobotSim.

5.1.1 Popis simulátoru MobotSim

MobotSim má tyto hlavní znaky:

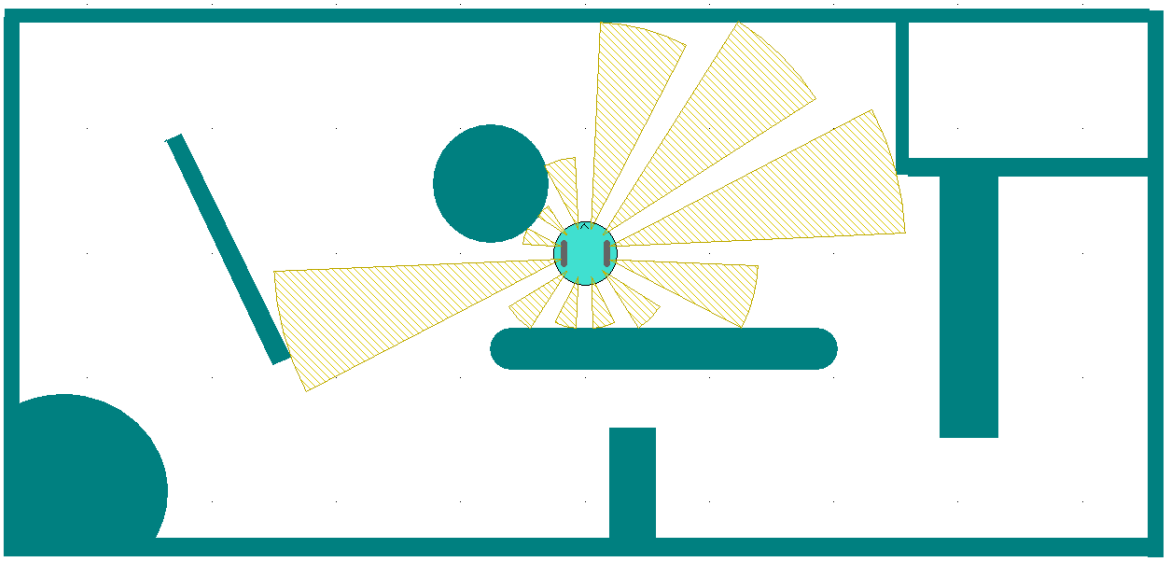
- Simulátor používá pouze svoje roboty (*Moboty*), kteří mají diferenciální pohon.
- Do prostředí můžeme přidat nekonečně robotů a překážek, které mohou být předdefinované nebo je můžeme vytvořit jednoduchým kreslením.
- Flexibilní konfigurace robotů nastavením parametrů (průměr kola, vzdálenost mezi koly, počet senzorů, nastavení úhlů mezi senzory, atd.).
- Konfigurace rozsahu senzorů.
- Rychlý a snadný vývoj simulace díky jazyku Basic.



Obr. 12 Popsání prostředí MobotSim.

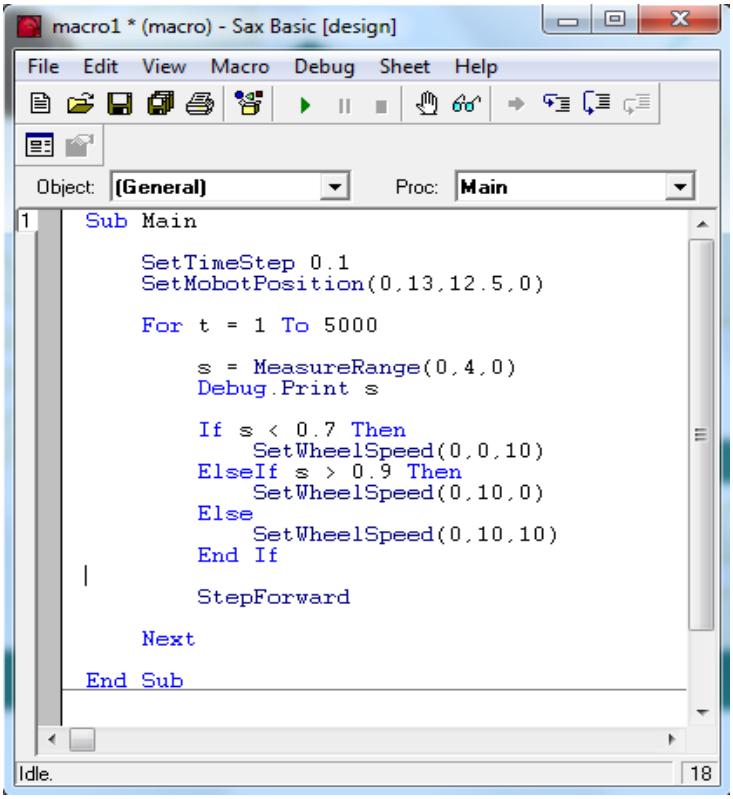
5.1.2 MobotSim simulace

Simulaci zahájíme vytvořením nového projektu kliknutím na záložku *File* a poté na záložku *New*. MobotSim vždy začíná s prázdným prostředím, které obsahuje pouze jednoho robota s defaultním nastavením, proto musíme vytvořit požadované prostředí pomocí různých překážek.



Obr. 13 Vytvoření prostředí pomocí překážek.

Pro nastavení pohybu robotů, používá MobotSim Basic editor, v němž si uživatel může vytvářet makra, které využívají specifické funkce jazyka Basic, k získání informací o souřadnicích robotů, k získání dat ze senzorů a jízdních údajů.[18]

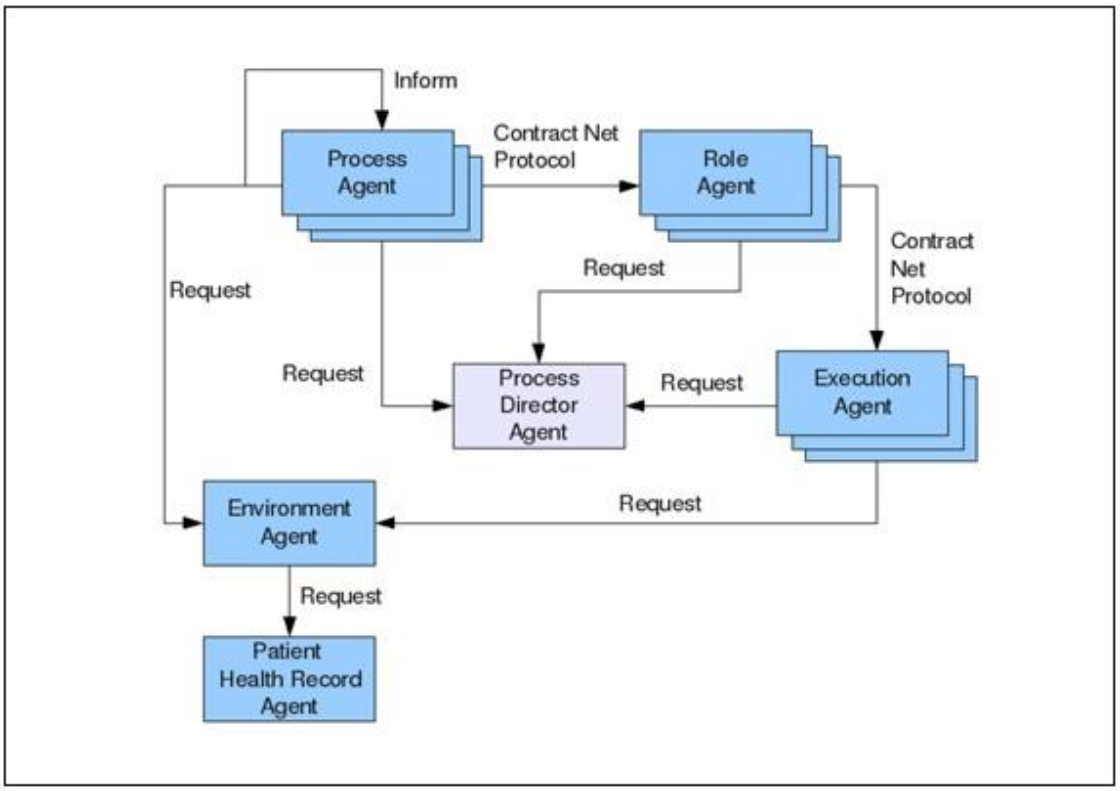


Obr. 14 Jednoduchý kód "sledování zdi" v Basic editoru.

5.2 Breve

Breve je volný opensourceový balíček, pro jednoduché vytváření 3D simulací multiagentních systémů a umělého života. Použitím skriptovacího jazyka Python můžeme nadefinovat chování agentů v 3D světě a sledovat, jak se vzájemně ovlivňují. Breve zahrnuje fyzikální simulace a detekce kolizí, takže můžeme simulovat realistické bytosti a zobrazit svůj nasimulovaný svět. Robotický simulátor Breve je dostupný pro MAC OS, Linux a Windows.[19]

„Multi-Agentní Systém (MAS) je systém skládající se z několika softwarových agentů, kteří jsou schopni vzájemné spolupráce, která vede k řešení daného problému.“[20]



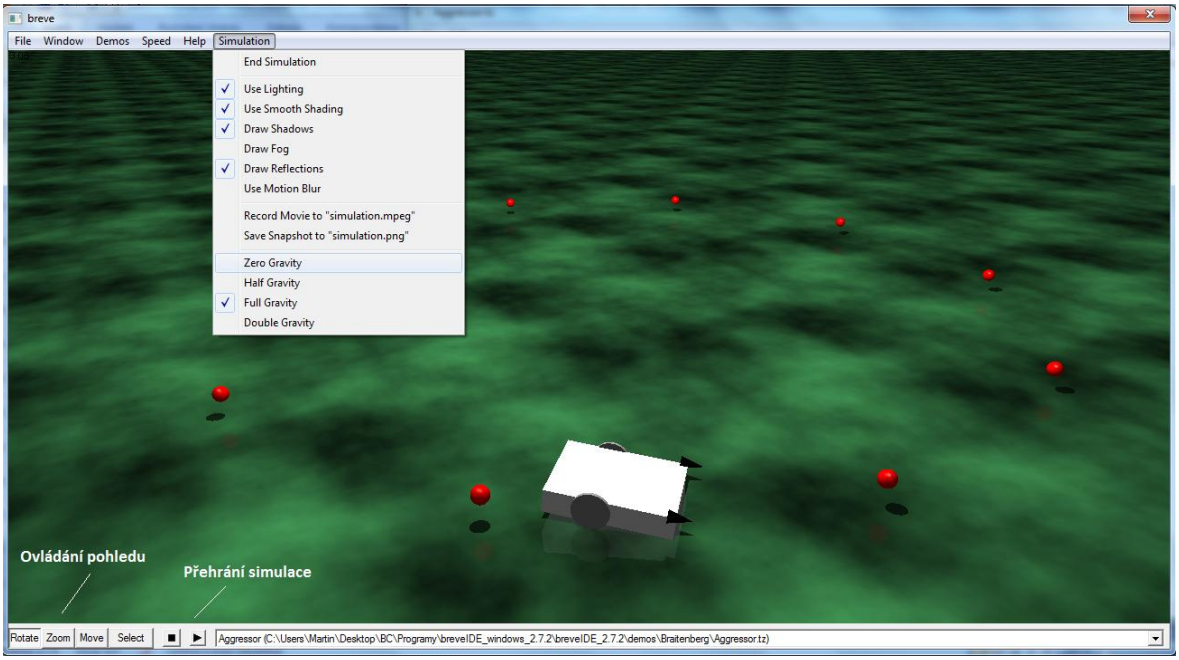
Obr. 15 Schéma multiagentního systému.[21]

5.2.1 Popis simulátoru Breve

Simulace v Breve jsou psány pomocí známého jazyka Python nebo pomocí jazyka Steve, který je objektově orientovaný a využívá mnoho vlastností jazyků jako například jazyka C.

Kromě vlastní simulace, můžeme v Breve spustit simulační dema, od nejjednodušších po složitější. Spuštěním dema se nám zobrazí i jeho vlastní kód, který můžeme použít do své simulace. Kliknutím na záložku *Simulation* můžeme přidávat různé simulace prostředí, např. přidat mlhu, nastavit poloviční či úplně vypnout gravitaci.

Také můžeme ovládat rychlost simulace, kliknutím na záložku *Speed*. Ovládání pohledu a samotné přehrávání simulace se nachází v levém dolním rohu.

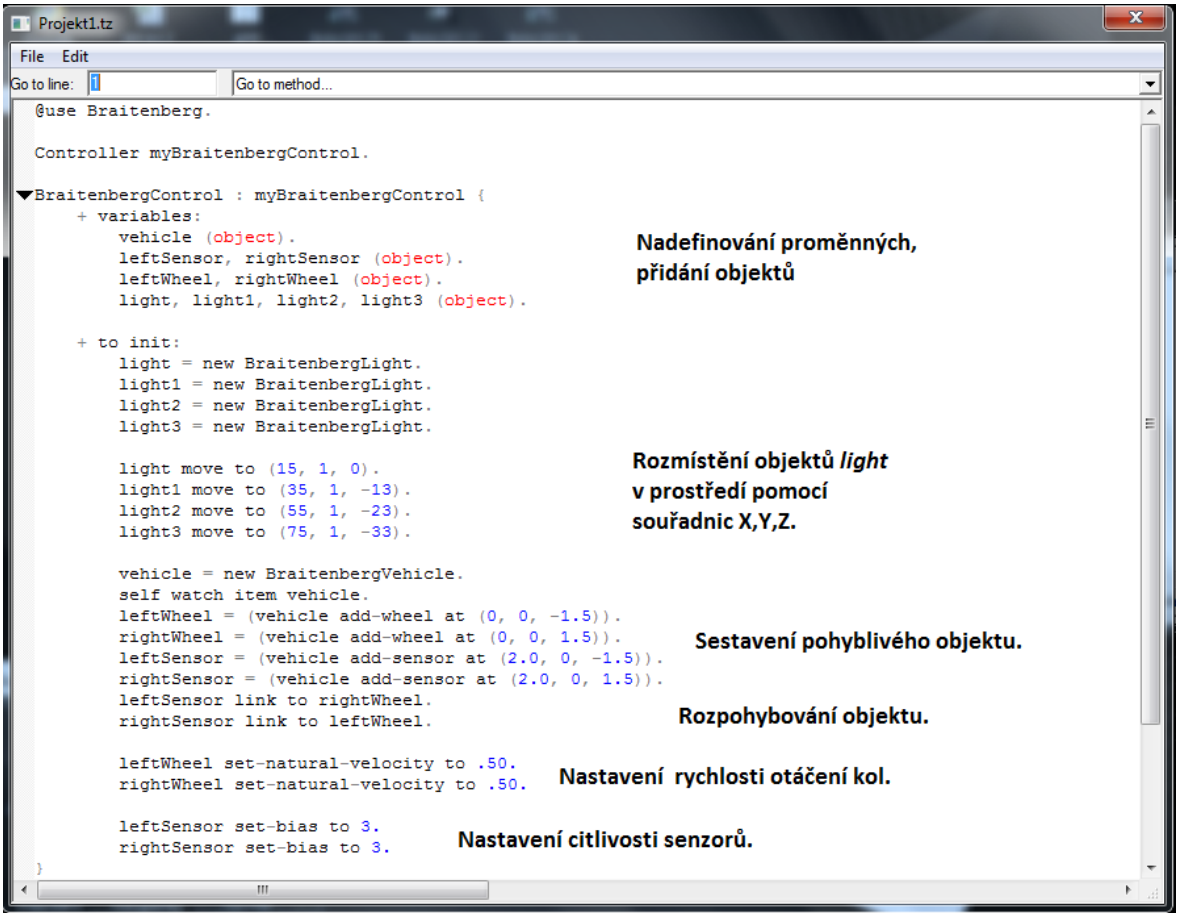


Obr. 16 Spuštění dema a nastavení simulace prostředí.

5.2.2 Breve simulace

Ovládat robotický simulátor Breve je velice jednoduché. Jestliže chceme začít programovat simulaci, v nabídce menu klikneme na záložku *File*, dále na záložku *New* a otevře se nám nové dialogové okno pro zadávání kódu.

Vlastní simulace se píše pomocí dynamického objektově orientovaného skriptovacího jazyka Python v samostatném okně. Pomocí programovacího jazyka můžeme do simulace přidávat různé objekty, které následně můžeme rozpohybovat.



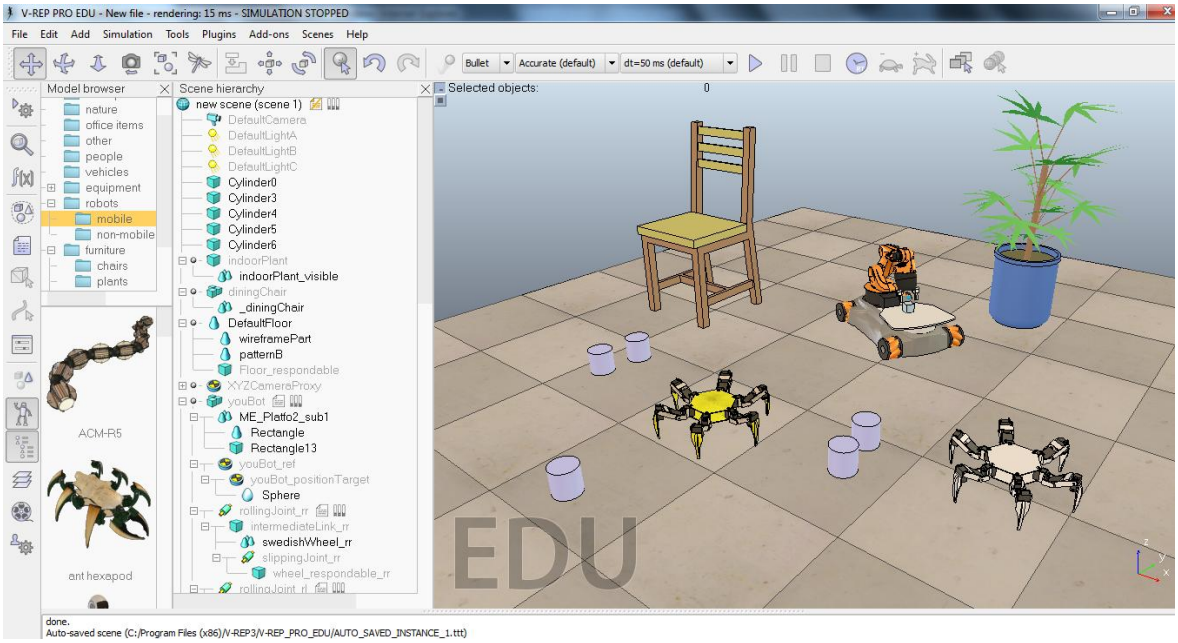
Obr. 17 Naprogramování jednoduché simulace.

Robotický simulátor Breve je vhodný pro jednoduché 3D simulace, jeho ovládání je velice jednoduché, a proto je vhodný spíše pro začátečníky.

5.3 V-REP

V-REP je švýcarský robotický simulátor vybavený mnoha funkcemi a propracovaným rozhraním API (Application Programming Interface).

Robotický simulátor V-REP, s integrovaným vývojovým prostředím, je založen na distribuované řídící architektuře: Každý objekt/model lze individuálně ovládat pomocí vloženého skriptu. To dělá V-REP velmi univerzálním a ideálním pro multi-robotické aplikace.[22]



Obr. 18 Prostředí V-REP.

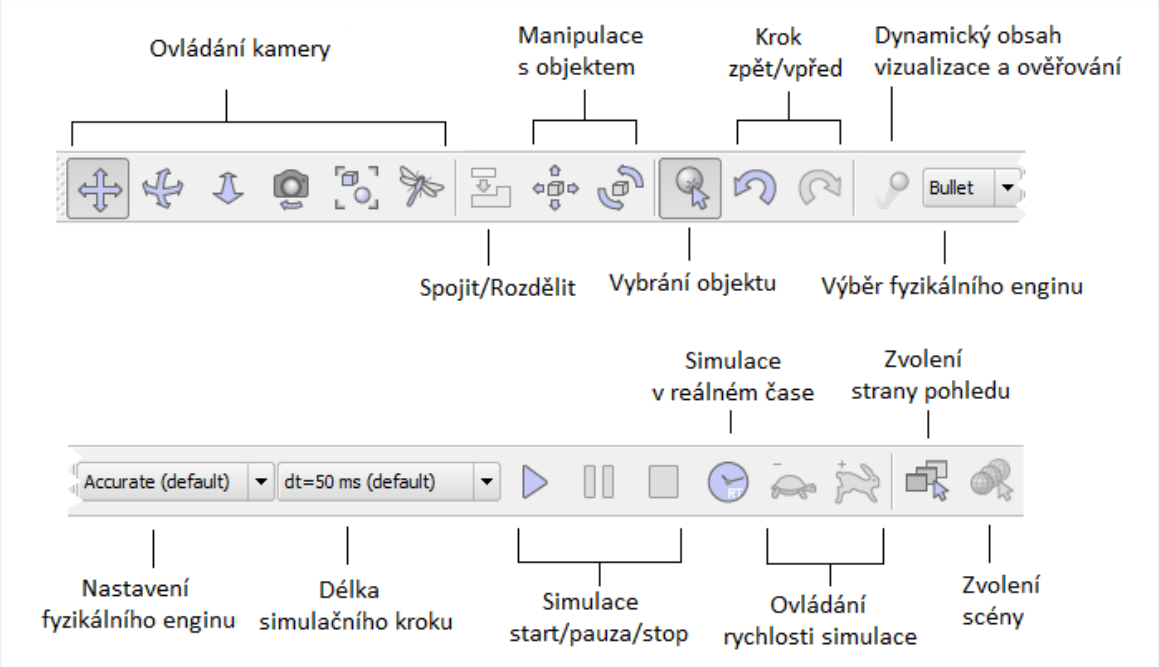
5.3.1 Popis simulátoru V-REP

Hlavní využití robotického simulátoru V-REP:

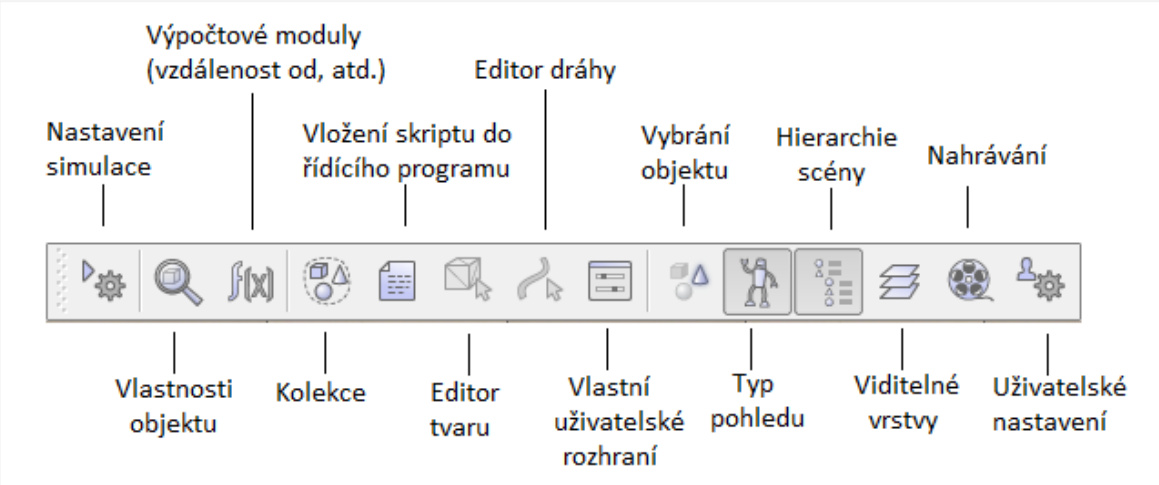
- rychlé prototypování a ověřování;
- rychlý vývoj algoritmů;
- robotika spojená se vzděláním;
- vzdálený monitoring;
- simulace systémů pro automatizaci výroby;
- bezpečnostní monitoring;
- prezentace produktu. [23]

K vytvoření reálného světa používá simulátor V-REP dva fyzikální enginy (Bullet, ODE), ty zajišťují rychlé dynamické výpočty, které slouží k reálným fyzikálním interakcím jako je tření, tuhost pružin, atd. V-REP podporuje přizpůsobitelné dynamické částice, které mohou být použity k simulaci vzduchové či vodní trysky, tryskového motoru, atd.

K ovládání simulátoru se používají dva hlavní panely, které se skládají ze simulační části, která slouží k ovládání simulace a nástrojové části, která slouží k ovládání scény.

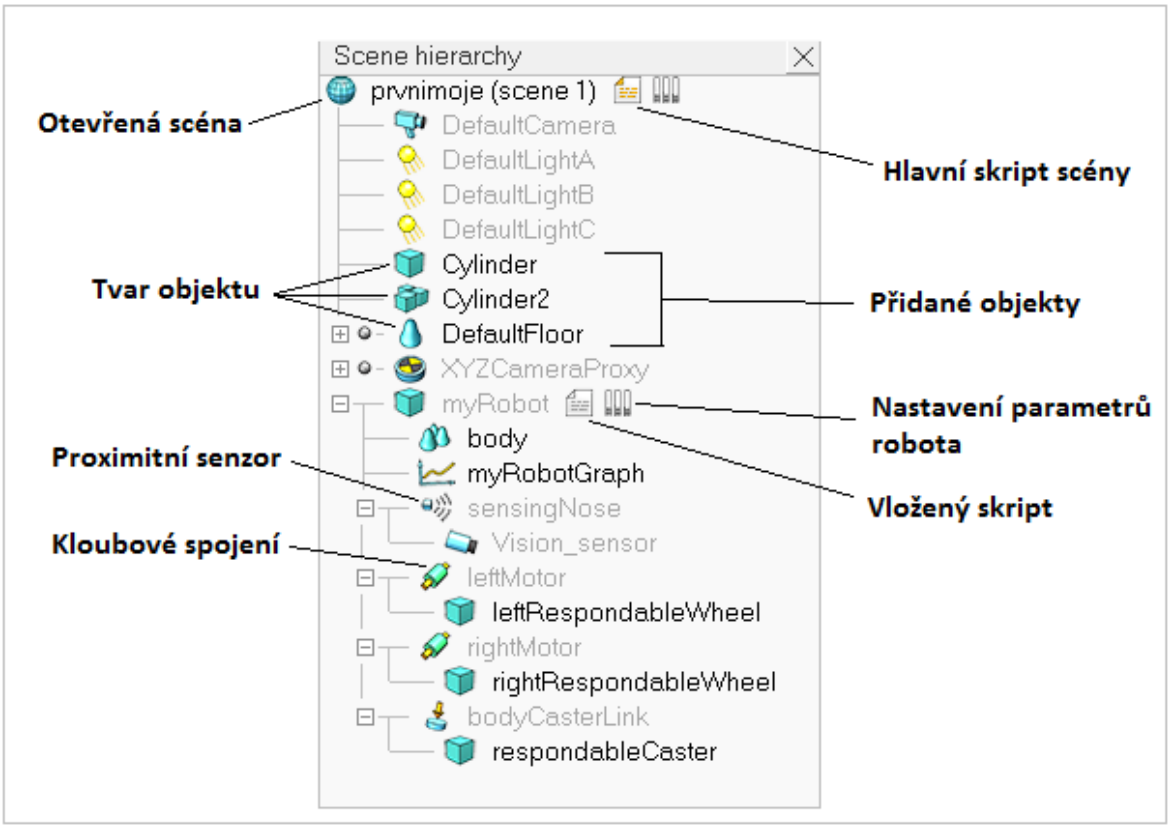


Obr. 19 Ovládací panel 1 (Toolbar1).



Obr. 20 Ovládací panel 2 (Toolbar 2)

Důležitým prvkem v robotickém simulátoru V-REP je rozvržení scény, kde lze vidět všechny objekty ve virtuálním světě a jejich vlastnosti. Pro související objekty a kontrolní skripty lze v hierarchii scény, přetáhnutím na jiný objekt, vytvořit vztah nadřazený/podřízený.



Obr. 21 Hierarchie scény V-REP.

Robotický simulátor V-REP nabízí velkou škálu modelů robotů a různých objektů. Nabídka modelů zobrazuje v horní části strukturu V-REP modelů a ve spodní části miniatury modelů obsažených ve vybrané složce. Modely lze přidávat do scény jednoduchým přetažením i během probíhající simulace.

Obr. 22 Nabídka modelů (objektů).

V-REP vychází ve třech verzích:

- V-REP PRO EDU je zdarma, je to verze určena pro studenty, učitele a profesory vysokých škol. Je bez omezení (tj. plně funkční).
- V-REP PRO je komerční verze pro společnosti, výzkumné instituce a neziskové organizace. Je bez omezení (tj. plně funkční).
- V-REP PLAYER je volně distribuovaná verze, má omezenou schopnost editování a nemá schopnost ukládání scény. [24]

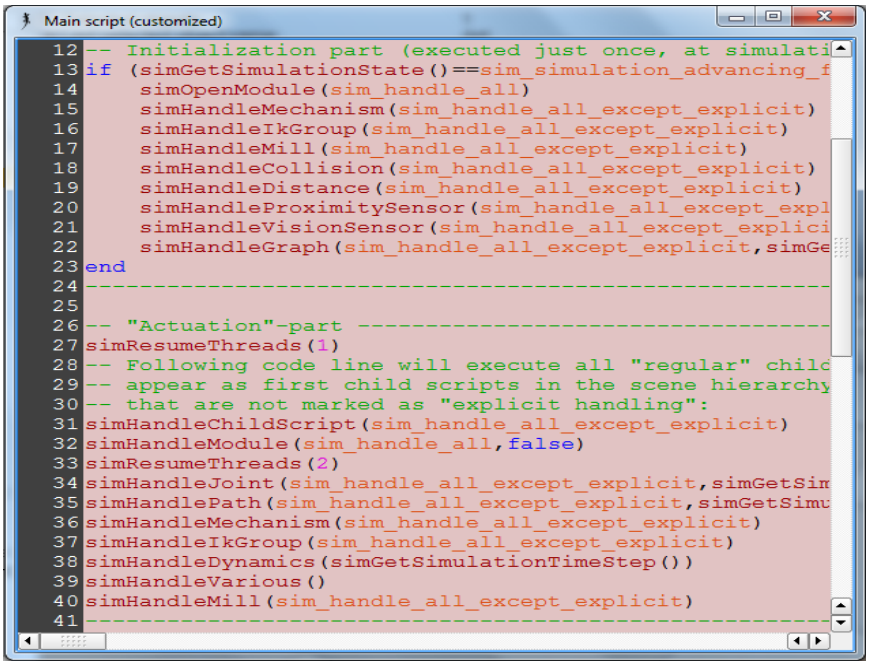
5.3.2 V-REP simulace

V-REP nabízí vzdálené rozhraní API (Application Programming Interface), které umožňuje ovládat simulace z externí aplikace nebo ze vzdáleného hardwaru (např. skutečný robot, vzdálený počítač, atd.). Vzdálené rozhraní API se skládá zhruba ze sto funkcí, které mohou být volány pomocí C/C++ aplikací, skriptu Python, Java aplikací, programu Matlab, nebo pomocí skriptu Urbi.[25]

Simulaci můžeme upravovat i pomocí vkládaných skriptů. Tyto skripty jsou psány pomocí skriptovacího jazyka Lua. Psaním Lua skriptů můžeme upravovat konkrétní simulace, ne však simulátor jako takový.[26]

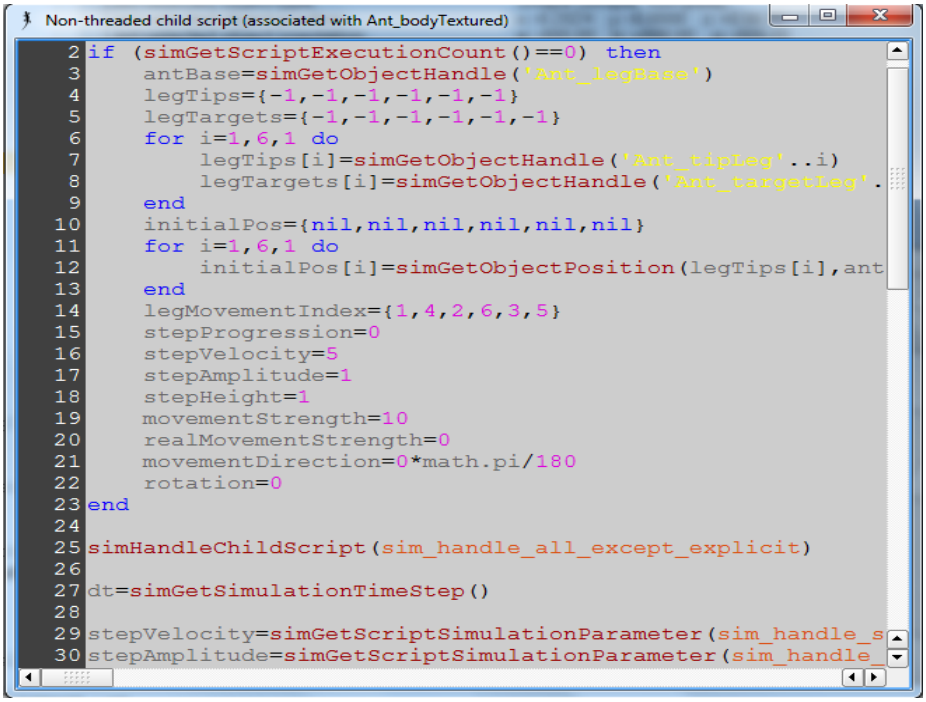
V-REP podporuje dva hlavní typy vložených skriptů:

- *Jeden hlavní skript*, který je ve výchozím nastavení. Každá scéna obsahuje hlavní skript, který zpracovává všechny funkce. Bez hlavního skriptu je nemožné simulaci spustit.



Obr. 23 Hlavní skript scény.

- *Neomezený počet podřízených skriptů*. Každá scéna obsahuje podřízené skripty, které mají na starost určitou část simulace. Nejčastěji se používají pro ovládání modelů.



Obr. 24 Podřízený skript.

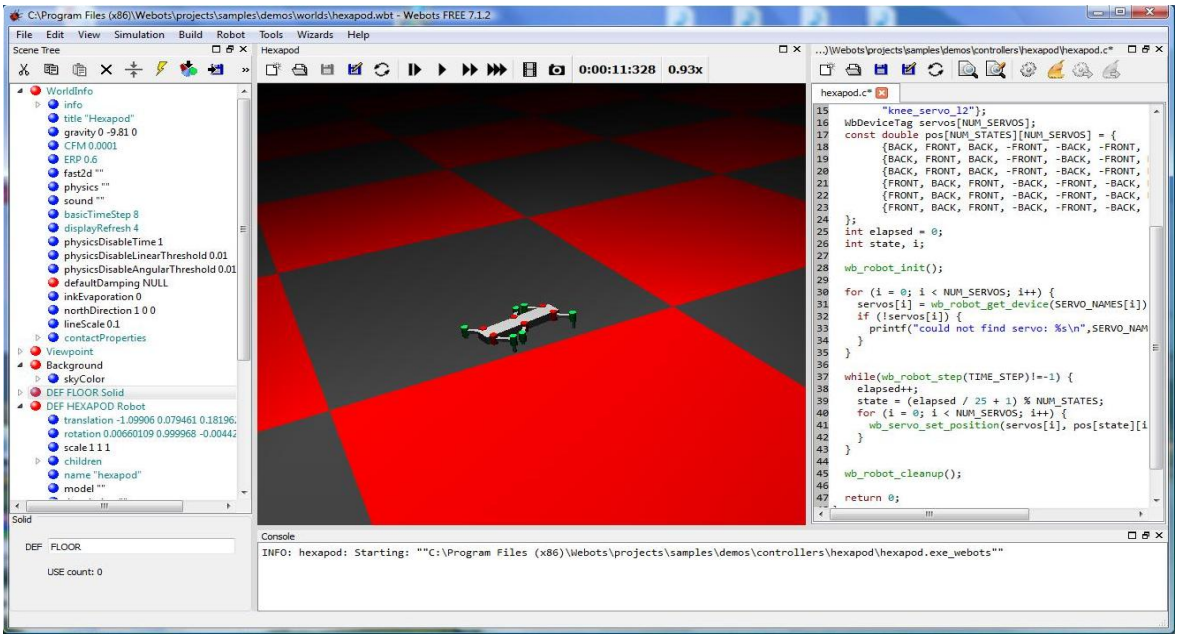
Rozdíly mezi *hlavním* a *podřízeným* skriptem:

- Může existovat pouze jeden hlavní skript, který může obsahovat neomezený počet podřízených skriptů.
- Hlavní skript je nezávislý a neměl by být upravován. Podřízené skripty jsou spojeny s objekty a jsou přizpůsobitelné.
- Hlavní skript nelze duplikovat. Podřízené skripty mohou být duplikovány v jedné scéně i mezi scénami.[27]

5.4 Webots

Webost je profesionální robotický simulační software. Nabízí rychlé vyvíjení prostředí, umožňuje uživateli vytvořit 3D virtuální světy s fyzikálními vlastnostmi jako jsou hmotnost, součinitel tření, atd. Uživatel může přidat jednoduché pasivní nebo aktivní objekty. Aktivními objekty se rozumí mobilní roboti, kterých je několik druhů např. kolové roboty, chodící roboty, či létající roboty. Roboti mohou být vybaveni mnoha senzory např. dotykovými senzory, senzory vzdálenosti, atd. V simulátoru Webots se také dají využít už hotové příklady programů, které mohou uživatelé použít do svých simulací.[28]

Webots také obsahuje řadu rozhraní pro skutečné mobilní roboty, takže jakmile se váš simulovaný robot chová podle očekávání, můžete přenést svůj kontrolní program do skutečného robota, jako je Nao, Darwin-OP, atd.[28]



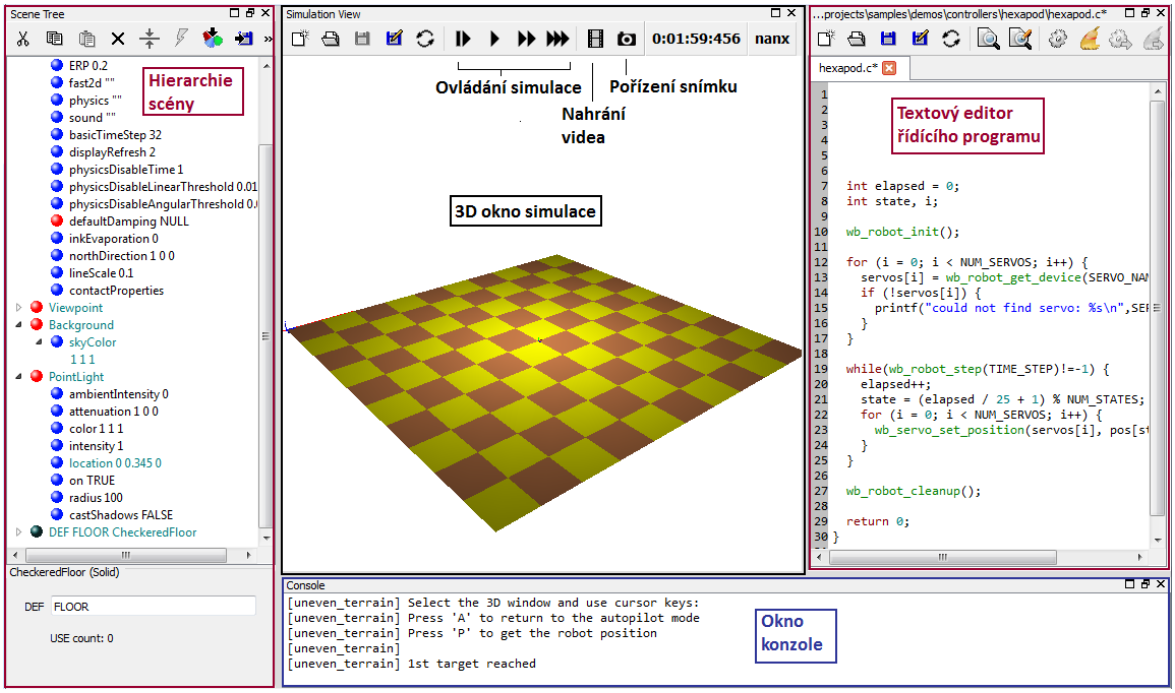
Obr. 25 Prostředí Webots.

5.4.1 Popis simulátoru Webots

Robotický simulátor Webots je vhodný i pro výzkumné a vzdělávací projekty v oblasti mobilní robotiky zejména v následujících oblastech:

- Vývoj prototypů mobilních robotů (akademický výzkum, automobilní průmysl, letectví, hračkářský průmysl, atd.).
- Výzkum pohybu robotů (humanoidní, čtyřnozí, atd.).
- Multiagentní výzkum (inteligence roje, spolupracující skupiny robotů, atd.).
- Výzkum adaptivního chování (genetické algoritmy, neuronové sítě, atd.).
- Výuka robotiky (lekce robotiky, C/C++/Java/Python programovací lekce, atd.). [29]

Webots GUI (grafické uživatelské rozhraní) se skládá ze čtyř hlavních oken: *3D okno*, které zobrazuje a umožňuje interakci s 3D simulací, okno *hierarchie scény*, které reprezentuje hierarchii vytvořeného světa, okno *textového editoru*, ve kterém můžeme upravovat zdrojový kód a okno *konzole*, které zobrazuje kompilaci a výstupy regulátorů.



Obr. 26 Webots GUI.

Webots licencí je několik verzí, a to Webots PRO, Webots EDU, Webots NAO a Webots FREE. Tyto licence se liší funkcemi, cenou a postupem instalace.

- Webots PRO je nejvýkonnější verze. Je určena pro výzkumné a vývojové projekty.
- Webots EDU je určena pro studentské účely.
- Webots NAO je speciální verze omezena na simulaci robotů NAO. Tato verze neumožňuje vytvoření nových robotů, avšak je vybavena různými modely robotů NAO.
- Webbots Free je volně dostupná omezená verze.

Webots je přeložen do francouzštiny, němčiny, španělštiny, čínštiny a japonštiny (a částečně do italštiny).

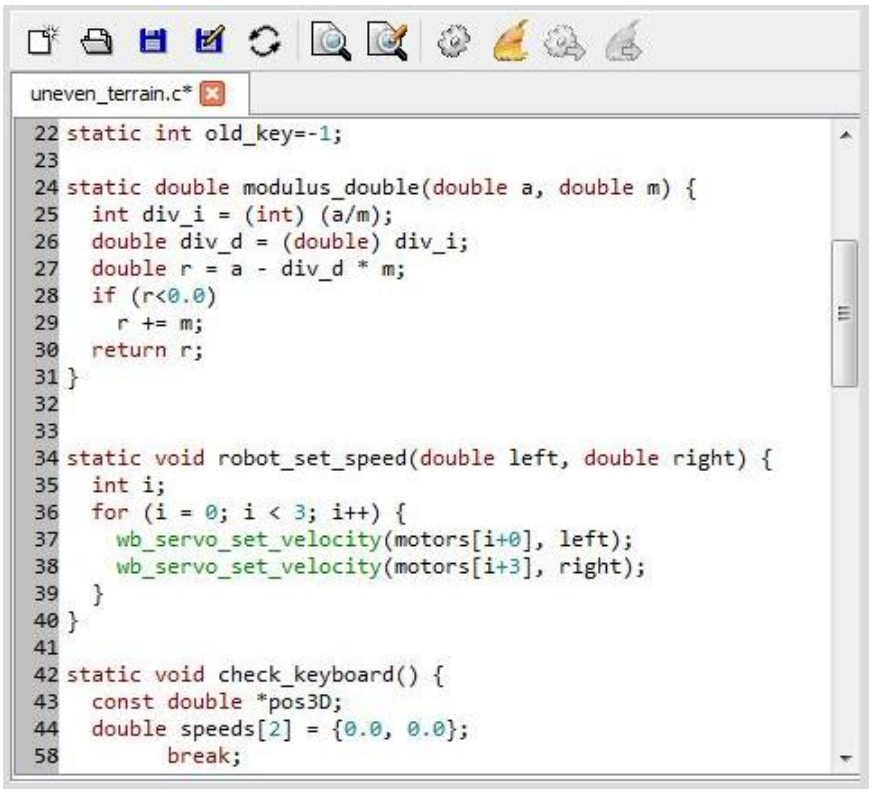
5.4.2 Webots simulace

Vlastní simulace se skládá ze tří položek:

1. Soubor *Webots World* (.wbt), který definuje jednoho nebo více robotů a jejich prostředí.
2. Jeden nebo více řídících programů (v C / C++ / Java / Python / Matlab).
3. Volitelný fyzický plugin.

Soubor *Webots World* obsahuje popis vlastností robota a jeho 3D prostředí. Dále obsahuje popis všech objektů např. pozici, geometrii, vzhled (barvu nebo jas), fyzické vlastnosti, typ objektu a jiné. Prostředí je uspořádáno hierarchicky, kde objekty mohou obsahovat další předměty. Například robot může obsahovat dvě kola, čidlo pro určení vzdálenosti a servo, které samo o sobě obsahuje kameru a jiné. Soubor *Webots World* neobsahuje řídící kód robota, pouze specifikuje název kontroloru, který je vyžadován všemi roboty. Prostředí je ukládáno jako *soubor.wbt*, tyto soubory jsou ukládány v podadresáři prostředí v každém projektu Webots.[30]

Řídící program je počítačový program, který řídí robota v určitém prostředí (*Webots World*). Řídící programy mohou být psány v mnoha programovacích jazycích podporovaných ve Webots (C, C++, Java, Python, Matlab). Při spuštění simulace, Webots spustí specifické řídící programy, každý jako samostatný proces. Je možné použít stejný řídící kód pro více robotů, nicméně proces bude spuštěn pro každého robota.[31]



Obr. 27 Náhled řídícího programu.

Robotický simulátor Webots používá více než 1033 univerzit a výzkumných center po celém světě.

6 POROVNÁNÍ SW

V této kapitole se zabývám porovnáváním jednotlivých robotických simulátorů. Simulátory budu srovnávat z několika hledisek:

Dostupné platformy: Windows, Linux, Mac OS

Nabízené licence:

- *Open source* - „tato licence označuje software s otevřeným zdrojovým kódem. Otevřenost zde znamená jak technickou dostupnost kódu, tak legální dostupnost - licenci software, která umožňuje, při dodržení jistých podmínek, uživatelům zdrojový kód využívat, například prohlížet a upravovat.“[32]
- *Freeware* - „je proprietární software, který je distribuován bezplatně. Podmínky bezplatného používání a šíření jsou definovány v licenční smlouvě, která je často specifická pro každý freeware. Autor si u freeware licence zpravidla ponechává autorská práva, například nedovoluje program upravovat nebo omezuje použití zdarma jen pro specifické účely.“[33]
- *Komerční* - „je takový software, který je šířen za úplatu. To znamená, že pokud produkt chcete používat, musíte za to tvůrci zaplatit. Takový software je obvykle možné používat jen dle omezení dané jeho licencí. Často je tak omezen počet instalací software současně, přenositelnost licence či právo modifikace produktu.“[34]
- *Studentská* - je licence pro vzdělávací počítačový software, jejichž primárním účelem je výuka nebo sebevzdělávání.[35]

2D/3D simulace:

- *2D* (dvojdimenzionální) označuje svět, který je popsán dvěma rozměry, předměty ve dvourozměrném světě mají obsah a např. délku a šířku, avšak nemají objem.
- *3D* (trojdimenzionální) označuje svět, který je popsán třemi rozměry, předměty ve trojrozměrném světě mají objem.

Druh simulace:

- *Simulace průmyslových manipulátorů* - je vhodná pro výrobní procesy, pásovou výrobu, atd.
- *Simulace mobilních robotů* - je vhodná pro procesy v nebezpečných a těžko dostupných místech.

Funkce:

- Každý simulátor má své hlavní funkce. Některé simulátory poskytují více funkcí a některé jsou specializovány na určité procesy (výrobní procesy, mapování prostředí, atd.).

Přehled robotických simulátorů a srovnávacích hledisek uvádím v porovnávací tabulce:

SW	Platforma	Licence	2D/3D	Simulace	Funkce
RoKiSim	Windows Linux Mac OS	Open source	3D	Robotických ramen (průmyslové manipulátory)	Simulace programovatelné univerzální manipulační ruky
RoboLogix	Windows Linux Mac OS	Komerční	3D	Robotických ramen (průmyslové manipulátory)	Automatizace výroby (paletizace, lakování, svařování, atd.)
MobotSim	Windows	Komerční	2D	Mobilních robotů	Mapování prostředí, vybrání optimální cesty
Breve	Windows Linux Mac OS	Open source	3D	Mobilních robotů	Simulace multiagentních systémů a umělého života
V-REP	Windows Linux Mac OS	Komerční Studentská Open Source	3D	Mobilních i stacionárních robotů	Rychlý vývoj algoritmů, rychlé prototypování robotů, velmi reálný svět
Webots	Windows Linux Mac OS	Komerční Studentská Freeware NAO	3D	Mobilních i stacionárních robotů	Návrh komplexního robotického nastavení s jedním nebo více roboty ve sdíleném prostředí

Tab. 1 Porovnání SW.

Závěrečné shrnutí robotických simulátorů:

RoKiSim - je Open source simulátor robotických ramen. Je vhodný pro jednoduché modelování a jednoduchou simulaci výrobních procesů.

RoboLogix - poskytuje simulaci výrobních procesů, zejména pásové výroby. Je vhodný i pro složitější simulace pásové výroby s kontrolou detekcí kolizí.

MobotSim - poskytuje simulaci robotů s diferenciálním pohonem. Je omezen 2D simulací, proto slouží hlavně pro robotické úlohy, jako jsou: zmapování prostředí, vybrání optimální cesty, atd.

Breve - je snadno ovladatelný Open source simulátor. Slouží k vytváření jednoduchých simulací mobilních robotů.

V-REP - je velmi výkonný simulátor, obsahuje nejvíce funkcí ze všech uvedených. Má velmi propracované GUI (grafické uživatelské rozhraní). Poskytuje *V-REP Forum*, které slouží pro technickou podporu.

Webots - obsahuje nejvíce licencí, je podobný simulátoru V-REP, avšak nemá tak propracované GUI. Jeho výhodou je jeho verze *Webots NAO*, ve které může přenášet simulované aplikace na roboty NAO bez jakýchkoliv modifikací.

6.1 Vybrání vhodného SW pro demonstraci robotické úlohy

Vybrání vhodného SW (robotického simulátoru) pro demonstrování robotické úlohy jsem prováděl z těchto hledisek:

- reálnost simulace,
- jednoduchost ovládání,
- přehlednost,
- dostupné funkce,
- dostupné modely robotů.

Nejreálnější simulaci, díky dvou fyzikálním enginům a dynamickým částicím, předvedl robotický simulátor V-REP.

V poměru s jeho mnoha funkcemi je snadno ovladatelný a to i díky přehlednosti, kterou zajišťuje jeho hierarchie scény *viz Obr. 5.*

V-REP obsahuje širokou škálu modelů robotů a různých objektů, se kterými se snadno manipuluje, a to díky modelovému prohlížeči, který obsahuje náhledy miniatur a podporuje přidání modelů do vytvořené scény jednoduchým přetažením.

Za velkou výhodu robotického simulátoru V-REP považuji jeho technickou podporu, kterou zprostředkovává *V-REP Forum*, ve kterém můžete vznášet své dotazy ohledně simulátoru, které vám administrátor obratem zodpoví a snaží se co nejvíce pomoct.

7 ROBOTICKÁ UKÁZKA VE V-REPU

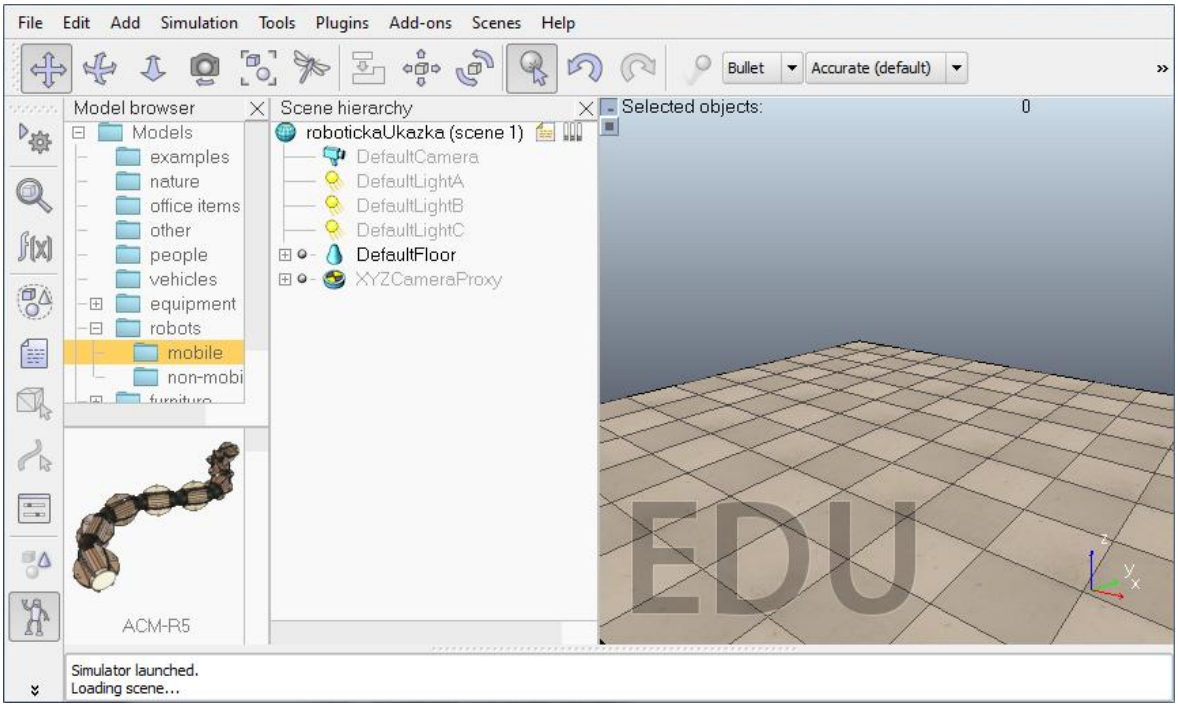
Jako robotickou úlohu, pro demonstraci ukázky, jsem vybral: *Sledování čáry s vyhýbáním překážek*, kterou budu simulovat ve vybraném robotickém simulátoru V-REP. Jedná se o úlohu, kdy vybraný robot bude sledovat předdefinovanou trasu určenou vodící čárou a zároveň se bude vyhýbat překážkám, které budou na této trase umístěny.

7.1 Sledování vodící čáry

„Jde o jednu z nejpoužívanějších metod pro navigaci mobilních robotů. Princip činnosti lze obecně popsat tak, že robot reaguje na snímanou pozici nějaké externí trasy. Je zřejmé, že metoda má hned dvě značná omezení: je nutná modifikace prostředí (instalace vodících čar) a není možná navigace na libovolné místo - robot může být navigován pouze tam, kde se nachází vodící čáry. Při použití této metody je možno použít prakticky neomezené množství strojů pracujících současně bez nebezpečí vzájemného rušení.“[36]

7.2 Demonstrace robotické úlohy: sledování čáry s vyhýbáním překážek

Demonstraci ukázky začneme vytvořením nové scény kliknutím na políčko *File* a dále na *New scene*. Tímto se nám vytvoří nová scéna defaultního nastavení. Tuto scénu uložíme pod názvem "robotickaUkazka " kliknutím na *File → Save scene as*.

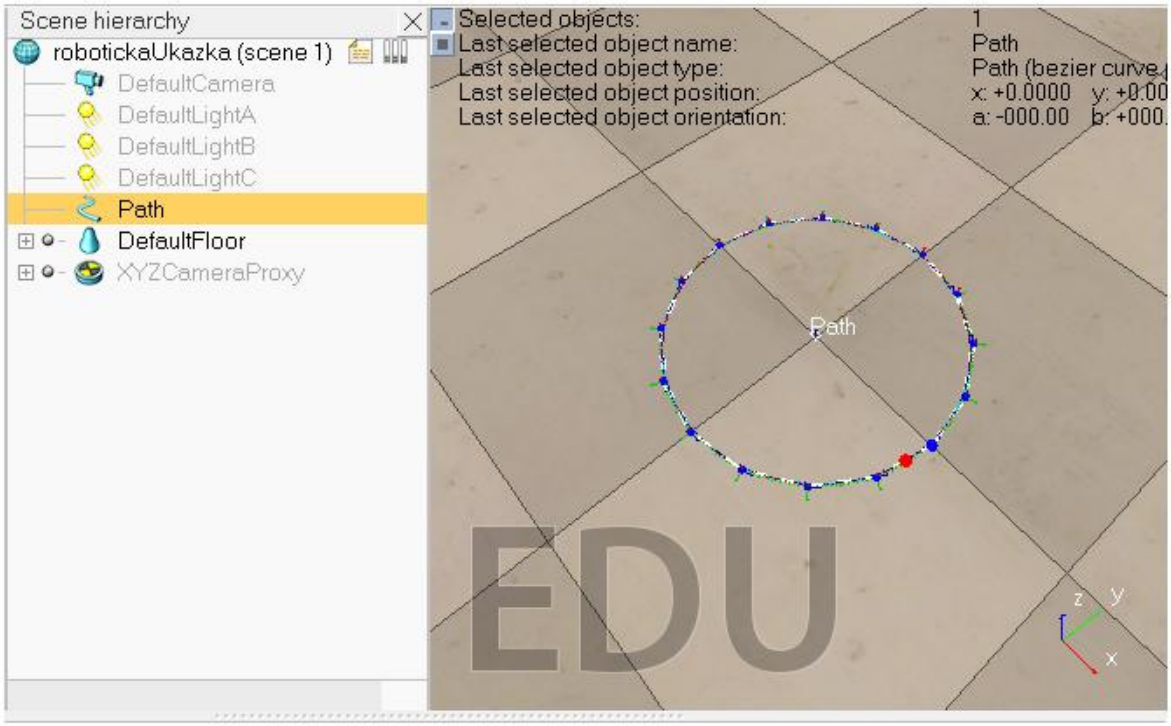


Obr. 28 Defaultní scéna.

7.2.1 Vymodelování prostředí

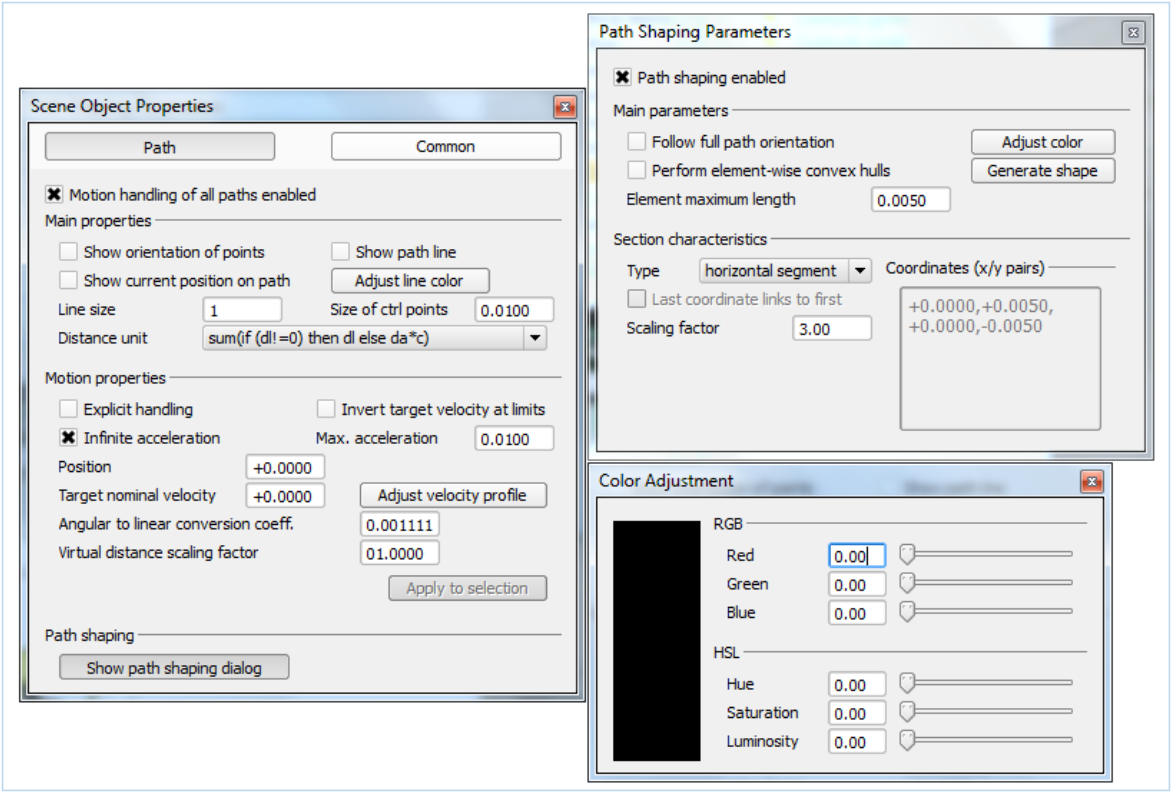
Vytvoříme vhodné prostředí, ve kterém chceme, aby se robot pohyboval. Jako první vytvoříme trasu, kterou robot bude následovat.

Trasu vytvoříme kliknutím na záložku *Add*, která se nachází v liště menu. Dále klikneme ne záložku *Path* a pro trasu kruhové typu vybereme *Circle type*. Do prostředí se nám přidá trasa kruhového typu, která se zobrazí i v hierarchii scény.



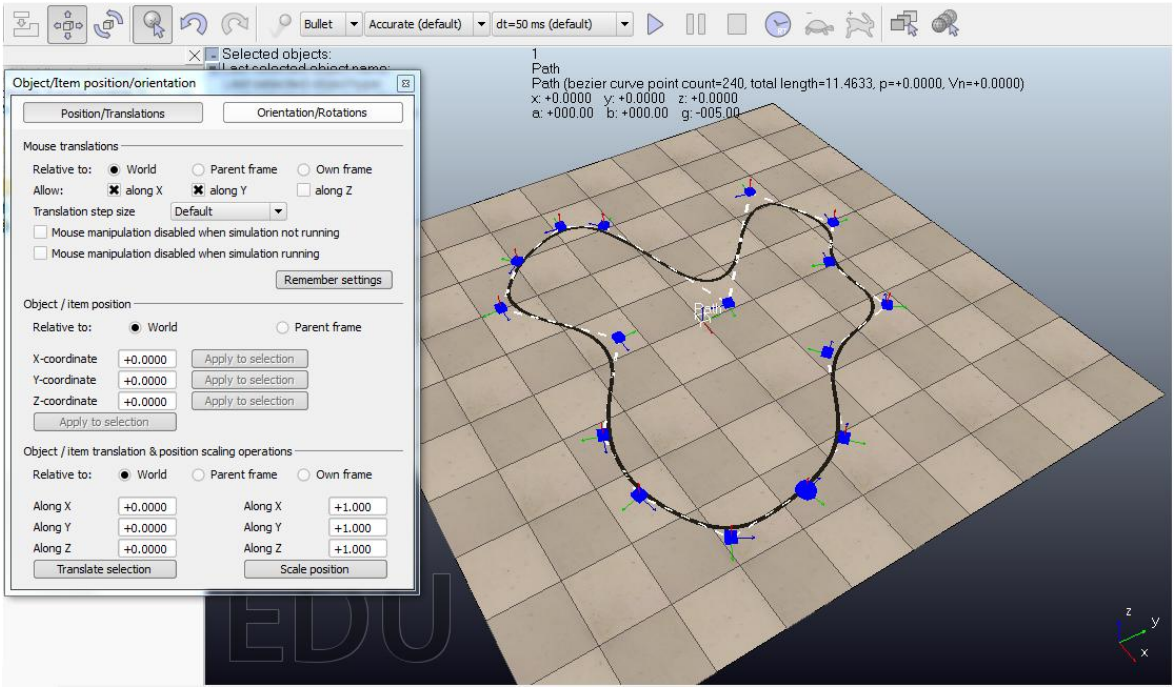
Obr. 29 Přidání trasy.

Nastavení parametrů trasy provedeme dvojitým klinutím na ikonu *Path* v hierarchii scény, která nám otevře *Scene path properties*. Zde zrušíme zaškrtnutí *Show orientation of points*, *Show path line* and *Show current position*, aby se nám při simulaci zobrazil pouze čistý tvar trasy. Dále v tomto okně klikneme na *Show path shaping dialog*, pro nastavení parametrů trasy. Zaškrtneme *Path shaping enable* a nastavíme *Scaling factor* na hodnotu 3.00 (trasa se třikrát zvětší). Barvu trasy nastavíme na černou, provedeme to klinutím na *Adjust color*.



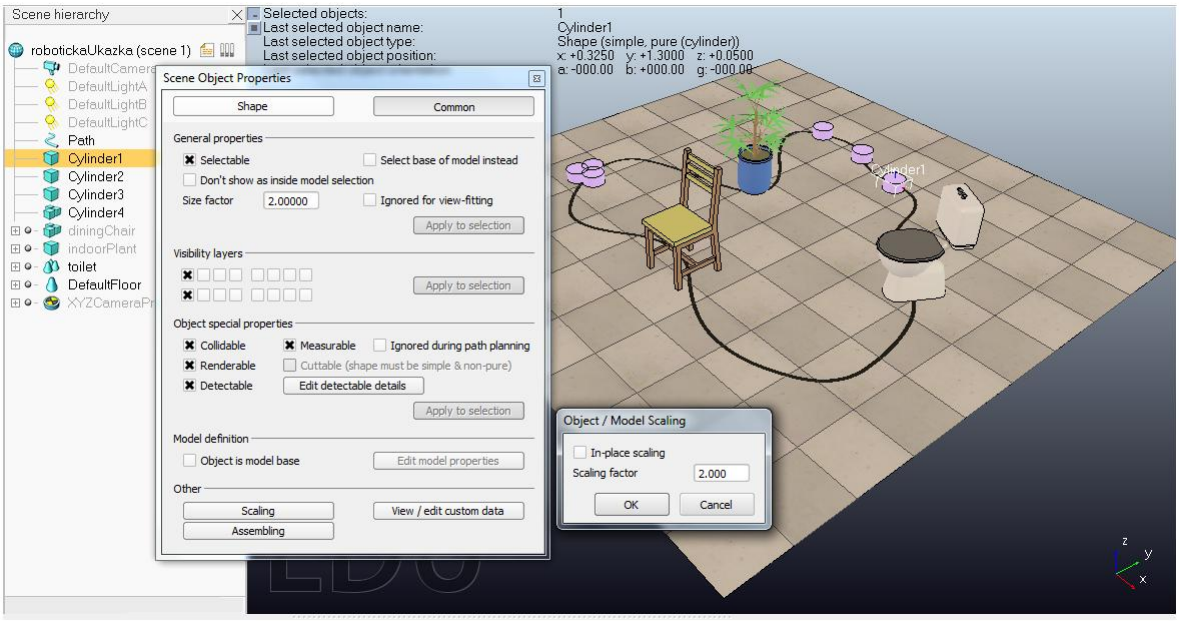
Obr. 30 Nastavení parametrů trasy.

Jako poslední nastavíme tvar trasy. V hierarchii klikneme na ikonu *Path*, tím se nám na trase vyznačí jednotlivé body, které můžeme v módu *Position/Translations* libovolně posouvat.



Obr. 31 Nastavení tvaru trasy.

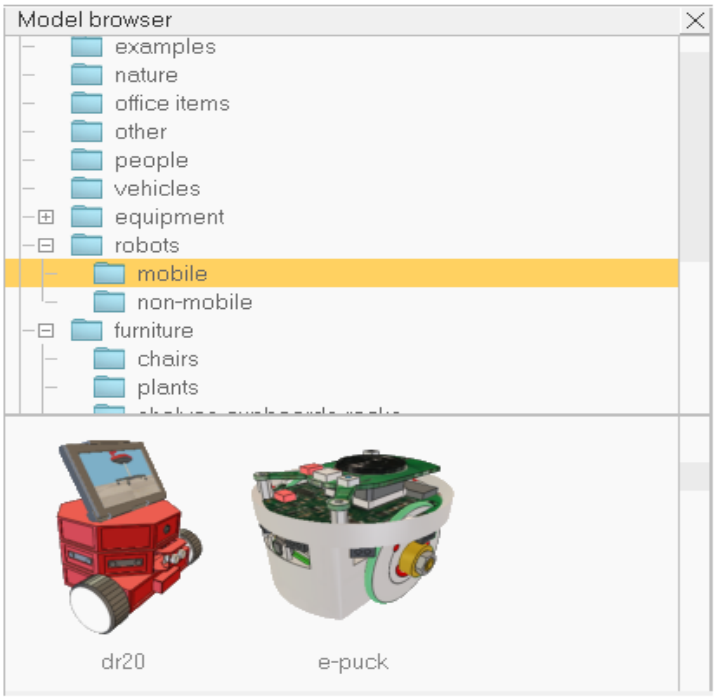
Následně přidáme do prostředí různé objekty, které umístíme na vytvořenou trasu robota. Můžeme přidávat objekty dvou druhů, buď modely objektů poskytované simulátorem, nebo vlastní vytvořené objekty. Modely objektů přidáme do prostředí jednoduchým přetažením jejich miniatur z modelového prohlížeče. Vlastní objekty vytvoříme následovně: *Add* → *Primitive shape* → *Cylinder*. Do prostředí se nám přidá válec, který můžeme libovolně kopírovat. Velikost přidaných objektů nastavíme v okně *Scene Object Properties*, které otevřeme dvojitým kliknutím na jednotlivé ikony objektů v hierarchii scény. Dále klineme na záložku *Common* a v dolní části otevřeme záložku *Scaling*, kde nastavíme *Scaling factor*.



Obr. 32 Přidání objektů.

7.2.2 Nastavení robota

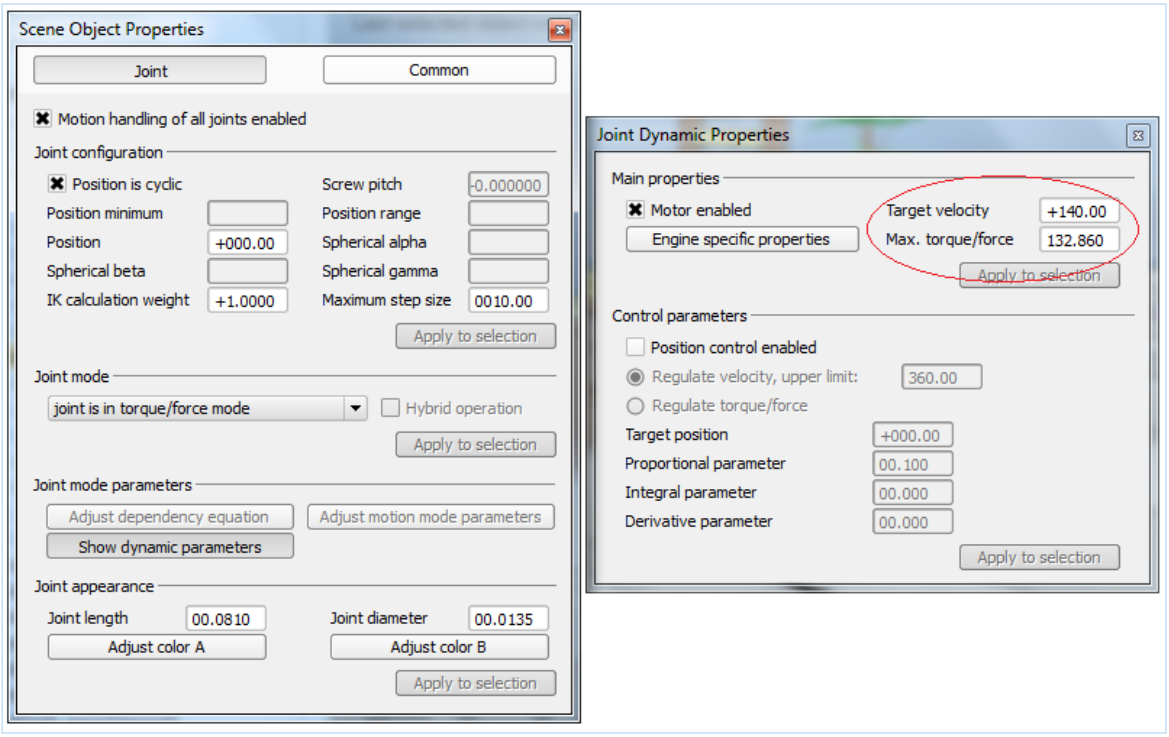
Pro tuto robotickou ukázku jsem vybral model robota e-puck, kterého přidáme do prostředí jednoduchým přetažením miniatury z modelového prohlížeče.



Obr. 33 Vybrání robota e-puck.

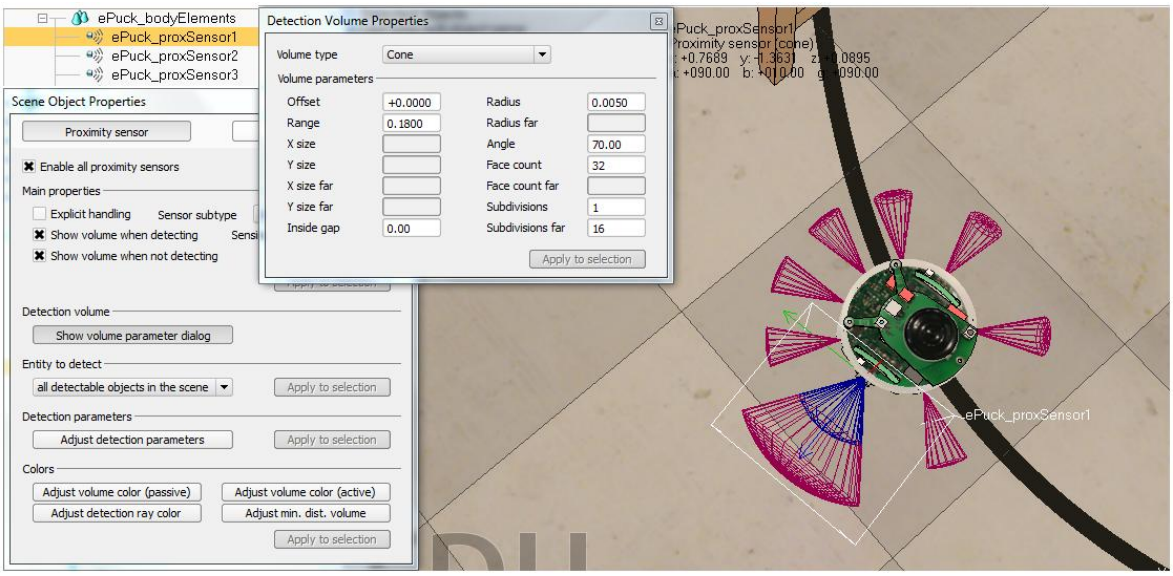
E-puck je mini mobilní robot, který byl původně vyvinut pro výukové účely. E-puck hardware a software je plně opensource, poskytuje nízkoúrovňový přístup ke každému elektronickému zařízení a nabízí neomezené možnosti rozšíření.[37]

Jako první nastavíme velikost robota, stejně jako u předchozího příkladu. Dále nastavíme rychlost, kterou se robot bude pohybovat. To provedeme dvojitým kliknutím na ikony *ePuck_rightJoint* a *ePuck_leftJoint*. Otevře se nám okno *Scene Object Properties*, kde zvolíme *Show dynamic parameters* a v okně *Joint Dynamic Properties* nastavíme *Target velocity* a *Max. torque/force.*



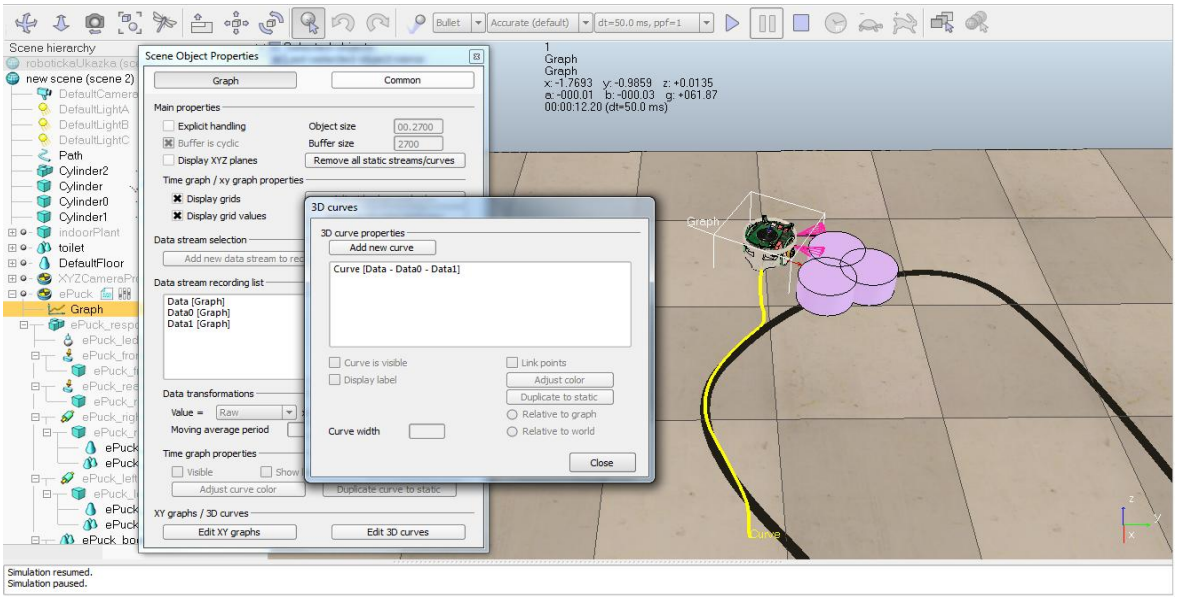
Obr. 33 Nastavení rychlosti robota.

Dále provedeme nastavení senzorů robota. Robot e-puck má celkem osm senzorů. Senzory nastavíme v *Scene Object Properties*, dvojitým kliknutím na ikonu jednotlivých senzorů. Zde vybereme *Show volume parameter dialog* a v nově otevřeném okně nastavíme *Range, Radius, Angle, Face count* (rozsah, poloměr, úhel a počet snímaných ploch).



Obr. 34 Nastavení senzorů.

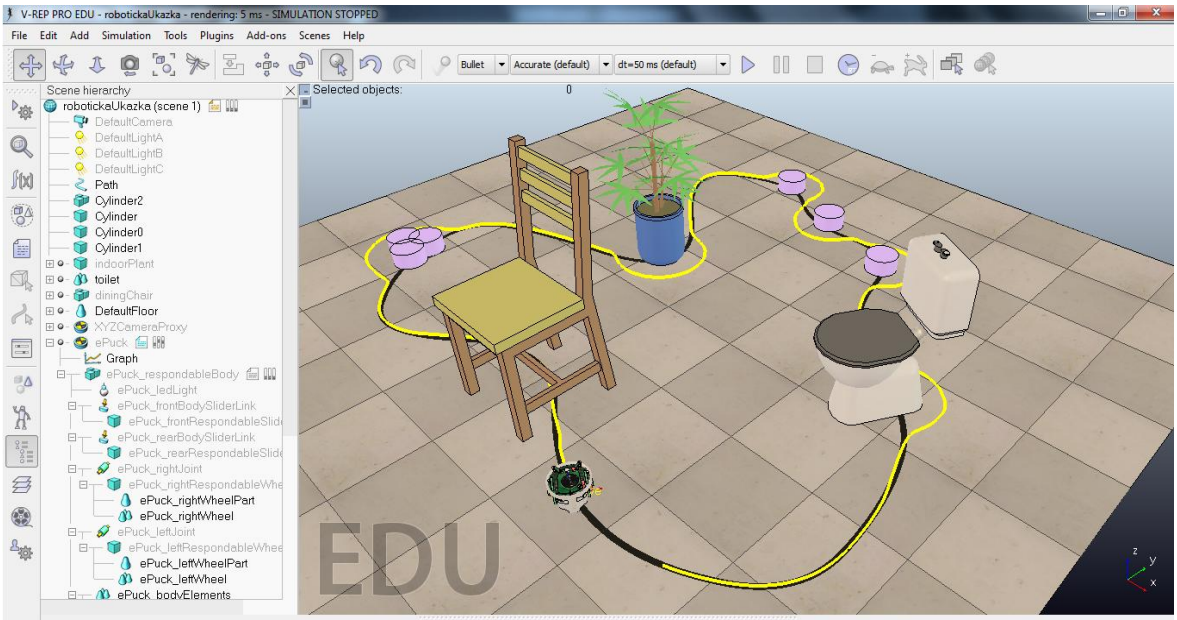
Nakonec robotovi přidáme graf, který bude zaznamenávat a zobrazovat projetou dráhu. Klikneme na: *Add → Graph.* Vytvořený graf přidáme v hierarchii pod robota e-puck. Označíme vytvořený graf, poté označíme robota e-puck a klikneme na pravé tlačítko myši. Zobrazí se nám menu, ve kterém vybereme: *Edit → Make last selected object parent.* Dále otevřeme *Scene Graph Properties* a do *Data stream recording list* přidáme tři nová data kliknutím na *Add new data stream to record*. Postupně přiřadíme tři data: *absolute x-position, absolute y-position* a *absolute z-position* k objektu *Graph.* Dále klikneme na *Edit 3D curves* pro nastavení křivky projeté trasy. Přidáme novou křivku (*Add new curve*) a k hodnotám *x,y,z-value* přiřadíme *Data, Data0, Data1*. Jako poslední nastavíme šířku křivky (*Curve width*) na hodnotu 4.



Obr. 35 Zobrazení projeté dráhy a detekce překážky.

7.2.3 Simulace robotické úlohy

Po spuštění simulace bude robot e-puck následovat vytvořenou trasu a vyhýbat se překážkám umístěným na trase. Díky přidanému grafu můžeme vidět dráhu, kterou robot skutečně projel.



Obr. 36 Demonstrace robotické úlohy.

8 ZÁVĚR

V této práci jsou popsány vybrané robotické simulátory, které jsou různého typu. V práci jsem se zaměřil jak na jednodušší robotické simulátory, které poskytují pouze některé robotické simulace buď vlivem menšího počtu funkcí, vlivem méně propracovaného prostředí nebo menším počtem modelů reálných robotů, tak na robotické simulátory vysoké úrovně, mezi které patří robotické simulátory Webots a V-REP. Tyto robotické simulátory převyšovaly ostatní simulátory ve všech směrech. Jsou vhodné jak pro simulaci mobilních robotů, tak i pro simulaci robotů stacionárních. Mají velmi propracované uživatelské prostředí, oba simulátory obsahují přehledné uspořádání scény prostřednictvím hierarchie scény. Mají širokou škálu modelů reálných robotů, což je výhodné hlavně z hlediska přenášení vytvořených aplikací na skutečného robota.

Robotické simulátory Webots a V-REP mají velmi dobře propracovanou dynamiku, která je velmi důležitá pro reálnost simulace. V této disciplíně vede simulátor V-REP, který má oproti simulátoru Webots, dva fyzikální enginy (Ode, Bullet) a podporuje přizpůsobitelné dynamické částice. K tomu má i propracovanější grafické rozhraní.

V první části práce jsem měl za úkol popsat a porovnat robotické simulátory, což jsem dle mého úsudku splnil. Dostatečně jsem jednotlivé robotické simulátory popsal. Popisoval jsem jak jejich prostředí, tak vytvoření samotné simulace. Poté jsem uvedl srovnání všech robotických simulátorů prostřednictvím srovnávací tabulky *viz Tab. 1*. Na závěr jsem provedl shrnutí jednotlivých simulátorů, uvedl jsem, k čemu jsou vhodné a jejich hlavní výhody.

V druhé části práce jsem se zabýval výběrem vhodného robotického simulátoru pro demonstraci robotické ukázky. K výběru jsem, na prvním místě, přistupoval z hlediska poskytnutí co nejreálnější simulace. Dále jsem hodnotil jednoduchost ovládání (vzhledem k množství dostupných funkcí) a orientování se v programu. Jako poslední jsem hodnotil i výběr programovatelných robotů.

Jako nejvhodnější SW pro simulaci autonomních robotů jsem vybral robotický simulátor V-REP. Tento simulátor byl optimální ze všech porovnávacích hledisek. Také poskytuje velmi dobrou technickou podporu v podobě mnoha tutoriálů a hlavně v podobě V-REP fóra, kde administrátor odpoví na každou otázku ohledně V-REP simulátoru.

V poslední části mé práce jsem pomocí vybraného robotického simulátoru V-REP demonstroval robotickou ukázku. Jako robotickou ukázku jsem vybral úlohu "sledování čáry s vyhýbáním překážek". Tato úloha je velice známá a používaná, pořádají se na ní různé robotické soutěže. Ukázku jsem nejdříve demonstroval rozebráním úlohy, dále jsem demonstroval vymodelování prostředí, ve kterém se robot bude pohybovat, a vymodeloval jsem trasu, kterou robot bude následovat. Provedl jsem nastavení vybraného robota a demonstroval simulaci robotické ukázky "sledování čáry s vyhýbáním překážek".

SEZNAM CITOVANÝCH ZDORJŮ

[1] WEIHU, Y. *Autonomous Robots Research Advances, Construction Robotics and Automation in Autonomous Robots Research Advances*. Vyd.1. Nova Science Publishers, Inc. 2008, 368s. ISBN-13: 978-1604561852.

[2] Wikipedie, Otevřená encyklopedie. *Simulace* [online]. 2008, 16.5.2013 [cit. 2013-05-17]. Dostupné z: <http://cs.wikipedia.org/wiki/Simulace>

[3] Wikipedie, Otevřená encyklopedie. *Počítačová Simulace* [online]. 2008, 16.5.2013 [cit. 2013-05-17]. Dostupné z: <http://cs.wikipedia.org/wiki/Simulace>

[4] VAVŘÍN, Jan. *Optimalizace počtu stanovišť vybraného autobusového nádraží s využitím simulace, Přínosy simulace* [PDF dokument]. Pardubice: Univerzita Pardubice, Fakulta elektrotechniky a informatiky, 2009. Dostupné z: <http://dspace.upce.cz/bitstream/10195/34787/5/VavrinJ_Optimalizace%20poctu_MB_1%20cast_2009.pdf >

[5] JANČÍKOVÁ, Zora. *Modelování a simulace, Simulační programy* [PDF dokument]. Ostrava: Technická Univerzita Ostrava, Fakulta Metalurgie a Materiálového inženýrství, 2008. Dostupné z: <http://www.fmmi.vsb.cz/miranda2/export/sites-root/fmmi/cs/okruhy/urceno-pro/studenty/podklady-ke-studiu/studijni-opory/638-Jancikova-Modelovani-a-simulace.pdf>

[6] Wikipedie, Otevřená encyklopedie. *Robotické simulátory* [online]. 2008, 16.5.2013 [cit. 2013-05-17]. Dostupné z: <http://cs.wikipedia.org/wiki/Simulace>

[7] Logic Design Inc. *RoboLogix User guide*, *Introduction* [online]. 2010 [cit. 2013-05-10]. Dostupné z: <https://www.robologix.com/docs/pdf/RoboLogix%20User%20Guide%20V7.pdf >

[8] INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ. *Talent pro vědu, Výukový model robota s pěti stupni volnosti* [PDF dokument]. Str. 15. [cit. 2013-05-15]. Dostupné z: <http://www.talenty-pro-vedu.cz/resources/files/exponaty-talenty-pro-vedu.pdf>

[9] HENZELÝ, Jakub. *Výuková pomůcka pro předmět mikropočítače – robot, Stacionární roboty* [PDF dokument]. Zlín: Univerzita Tomáše Bati ve Zlíně, Ústav aplikované informatiky. 2009. Dostupné z: <http://dspace.k.utb.cz/bitstream/handle/10563/8682/henzel%C3%BD_2009_bp.pdf?sequence=1>

[10] École de technologie supérieure. *RoKiSim - Robot Kinematics Simulator* [online]. 2013 [cit. 2013-04-17]. Dostupné z: <http://www.parallemic.org/RoKiSim.html>

[11] École de technologie supérieure. *Taking a screencast* [online]. 2013 [cit. 2013-04-18]. Dostupné z: <http://www.parallemic.org/RoKiSim.html>

[12] Logic Design Inc. *Welcome to RoboLogix* [online]. 2013 [cit. 2013-04-18]. Dostupné z: <http://www.robologix.com/>

[13] Logic Design Inc. *RoboLogix Overview* [online]. 2013 [cit. 2013-04-19]. Dostupné z: <http://www.robologix.com/robologix_overview.php>

[14] Logic Design Inc. *Programming With RoboLogix* [online]. 2013 [cit. 2013-04-19]. Dostupné z: <http://www.robologix.com/programming_robologix.php>

[15] Logic Design Inc. *Control panel* [online]. 2010 [cit. 2013-04-20]. Dostupné z: <https://www.robologix.com/docs/pdf/RoboLogix%20User%20Guide%20V7.pdf>

[16] KOŠNAR, Karel. *Mobilní robotika, Kinemtaika* [PDF dokument]. Praha: České Vysoké Učení Technické v Praze, Fakulta elektrotechnická. Dostupné z: <http://www.roznovskastredni.cz/dwnl/pel2007/06/Kosnar.pdf >

[17] WINKLER, Zbyněk, DLOUHÝ, Martin . *Odometrie, Bug algoritmy, Lokalizace, GPS, Exaktní plánování* [online]. 2003-2006, [cit. 2013-04-21]. Dostupné z: <http://robotika.cz/guide/cs>

[18] Best Shareware. *MobotSim - Mobile Robot Simulator* [online]. 2013 [cit. 2013-04-21]. Dostupné z: <http://www.bestshareware.net/download/mobotsim-mobile-robot-simulator.htm>

[19] Spiderland. *The breve Simulation Environment* [online]. 2006 [cit. 2013-04-21]. Dostupné z: <http://www.spiderland.org/>

[20] BEDNÁŘ, Michal. *Multiagentní systémy* [online]. 2007, 23.4.2008 [cit. 2013-05-15]. Dostupné z: <http://www.multiagent.tym.cz/nicoolas/Home.php>

[21] BOŠANSKÝ, B. *Agent technology center, Multiagentní architektura založena na procesech* [online] [cit. 2013-05-15]. Dostupné z: <http://www.ejbi.org/en/ejbi/article/13-cs-multiagentni-architektura-zalozena-na-procesech-aplikovana-na-domenu-domaci-pece.html>

[22] Coppelia Robotics. *V-REP home* [online]. 2013 [cit. 2013-05-15]. Dostupné z: <http://www.coppeliarobotics.com/index.html>

[23] Coppelia Robotics. *Main Features* [online]. 2013 [cit. 2013-05-12]. Dostupné z: <http://www.coppeliarobotics.com/v-repSpecifications.pdf>

[24] Coppelia Robotics. *V-REP Flavours* [online]. 2013 [cit. 2013-05-12]. Dostupné z: <http://www.coppeliarobotics.com/v-repOverviewPresentation.pdf>

[25] Coppelia Robotics. *Remote Interfaces* [online]. 2013 [cit. 2013-05-12]. Dostupné z: <http://www.coppeliarobotics.com/v-repOverviewPresentation.pdf>

[26] Coppelia Robotics. *Local Interfaces* [online]. 2013 [cit. 2013-05-12]. Dostupné z: <http://www.coppeliarobotics.com/v-repOverviewPresentation.pdf>

[27] Coppelia Robotics. *Embedded scripts* [online]. 2013 [cit. 2013-05-12]. Dostupné z: <http://www.coppeliarobotics.com/helpFiles/index.html>

[28] Cyber Botics. *Introduction toWebots* [online]. 2013, 22nd of May 2013 [cit. 2013-05-23]. Dostupné z: <http://www.cyberbotics.com/guide.pdf>

[29] Cyber Botics. *What can I do with Webots* [online]. 2013, 22nd of May 2013 [cit. 2013-05-23]. Dostupné z: <http://www.cyberbotics.com/guide.pdf>

[30] Cyber Botics. *What is a world?* [online]. 2013, 22nd of May 2013 [cit. 2013-05-23]. Dostupné z: <http://www.cyberbotics.com/guide.pdf>

[31] Cyber Botics. *What is a controller?* [online]. 2013, 22nd of May 2013 [cit. 2013-05-23]. Dostupné z: <http://www.cyberbotics.com/guide.pdf>

[32] Wikipedie, Otevřená encyklopedie. *Otevřený software* [online]. 2004, 26.3.2013 [cit. 2013-05-10]. Dostupné z: <http://cs.wikipedia.org/wiki/Otev%C5%99en%C3%BD_software>

[33] Wikipedie, Otevřená encyklopedie. *Freeware* [online]. 2005, 8.3.2013 [cit. 2013-05-10]. Dostupné z: < http://cs.wikipedia.org/wiki/Freeware>

[34] Wikipedie, Otevřená encyklopedie. *Komerční software* [online]. 2006, 10.3.2013 [cit. 2013-05-10]. Dostupné z: <http://cs.wikipedia.org/wiki/Komer%C4%8Dn%C3%AD_software>

[35] Wikipedia, The free encyclopedia. *Educational software* [online]. 2006, 13rd of May 2013 [cit. 2013-05-15]. Dostupné z: <http://en.wikipedia.org/wiki/Educational_software>

[36] ŠOLC, František, ŽALUD, Luděk. *ROBOTIKA, Sledování vodící čáry* [PDF dokument]. Brno: Vysoké Učení Technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2002. Dostupné z: <http://matescb.skvorsmalt.cz/robotika_kybernetika/VUT_Brno_Robotika.pdf>

[37] Cyber Botics. *e-puck Overview* [online]. 2013, 22nd of May 2013 [cit. 2013-05-01]. Dostupné z: < http://www.cyberbotics.com/e-puck>

SEZNAM ODBORNÉ LITERATURY

NOVÁK, Petr. *Mobilní roboty: pohony, senzory, řízení*. Vyd. 1. Praha: BEN - technická literatura, 2004, 247 s. ISBN 80-730-0141-1.

ZELINKA, Ivan. *Umělá inteligence: v problémech globální optimalizace*. Vyd. 1. Praha: BEN, 2002, 189 s. ISBN 80-730-0069-5.

SEZNAM PŘÍLOH NA CD

- Hlavní dokument: BP, 2013, Šťastný Martin, SW pro simulaci autonomních robotů.pdf.
- Animace simulované robotické ukázky.
- Skripty psané pomocí jazyka LUA (skripty.zip)