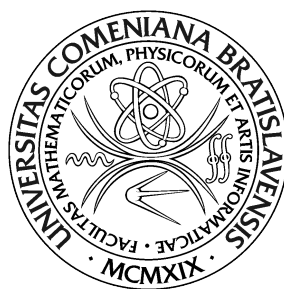


UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



RIADENIE MODELU KRÁČAJÚCEHO HMYZU  
DIPLOMOVÁ PRÁCA

2019  
Bc. ADRIÁN PAVČO

UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

RIADENIE MODELU KRÁČAJÚCEHO HMYZU  
DIPLOMOVÁ PRÁCA

Študijný program: Aplikovaná informatika  
Študijný odbor: 2511 Aplikovaná informatika  
Školiace pracovisko: Katedra aplikovanej informatiky  
Školiteľ: RNDr. Andrej Lúčny, PhD.

Bratislava, 2019  
Bc. Adrián Pavčo



Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky

## ZADANIE ZÁVEREČNEJ PRÁCE

**Meno a priezvisko študenta:** Bc. Adrián Pavčo  
**Študijný program:** aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)  
**Študijný odbor:** aplikovaná informatika  
**Typ záverečnej práce:** diplomová  
**Jazyk záverečnej práce:** slovenský  
**Sekundárny jazyk:** anglický

**Názov:** Riadenie modelu kráčajúceho hmyzu  
*Controlling Model of Walking Insect*

**Cieľ:** Cieľom práce je rozvíjať sčasti vyvinutý model kráčajúceho hmyzu. Model je vyvinutý v C# pod MRDS, ale je možné zvážiť jeho reimplementáciu v iných prostriedkoch, napr. v C++ a ODE. Každopádne treba ho najprv správne vyvážiť, aby ho príliš ťažký abdomen neprekacoval dozadu a doplniť o nejakú riadiacu architektúru (yarp alebo Agent-Space). A 'vytvoriť' v tom riadiaci systém, ktorý bude s modelom kráčať aspoň po rovine s možnosťou demonštrácie prechodu cez prekážku a krátku priepasť.

**Literatúra:** Cruse, H: Neural Networks as Cybernetic Systems (časti o kráčajúcom hmyze)  
dokumentácia k MRDS prípadne ODE  
články od Cruze-Schultz

**Kľúčové slová:** riadenie, kráčajúci hmyz, 3D modelovanie

**Vedúci:** RNDr. Andrej Lúčny, PhD.  
**Katedra:** FMFI.KAI - Katedra aplikovanej informatiky  
**Vedúci katedry:** prof. Ing. Igor Farkaš, Dr.  
**Dátum zadania:** 30.09.2016

**Dátum schválenia:** 14.10.2016  
prof. RNDr. Roman Ďurikovič, PhD.  
garant študijného programu

.....  
študent

.....  
vedúci práce

## Prehlásenie

Čestne prehlasujem, že som túto diplomovú prácu vypracoval samostatne a všetku použitú literatúru uvádzam v zozname.

V Bratislave, dňa xx.yy.2019

Bc. Adrián Pavčo: .....

## Podakovanie

Ďakujem vedúcemu práce RNDr. Andrejovi Lúčnemu, PhD. za odborné vedenie, ochotu a trpezlivosť.

## Abstrakt

Kľúčové slová: riadenie, kráčajúci hmyz, 3D modelovanie

# Abstract

**Keywords:** simulation, walking insect, 3D modeling

# Contents

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Ciele práce . . . . .	1
1.2	Členenie práce . . . . .	1
<b>2</b>	<b>Problematika</b>	<b>2</b>
2.1	Existujúce riešenia . . . . .	2
2.2	Východiskové riešenie . . . . .	2
<b>3</b>	<b>Nástroje</b>	<b>4</b>
3.1	Úvod . . . . .	4
3.2	Simulátor Gazebo . . . . .	4
3.2.1	Inštalácia . . . . .	5
3.2.2	Prostredie . . . . .	6
	Grafické rozhranie . . . . .	6
	Súbory . . . . .	8
	Štruktúra sdf modelov . . . . .	8
3.3	Potreba riadiacej architektúry . . . . .	10
3.4	ROS . . . . .	10
3.4.1	Inštalácia . . . . .	11
3.4.2	Architektúra TODO . . . . .	11
3.4.3	Prostredie TODO . . . . .	11
3.5	Použitie Gazebo+ROS TODO . . . . .	11
<b>4</b>	<b>Model mravca</b>	<b>12</b>
4.1	Anatómia mravca TODO . . . . .	12
4.2	Dedenie a úprava modelu . . . . .	12
4.2.1	Formáty modelov . . . . .	12
4.2.2	Konvertovanie modelu . . . . .	12
4.2.3	Skladanie modelu . . . . .	13
<b>5</b>	<b>Simulácia</b>	<b>19</b>





# Chapter 1

## Úvod

### 1.1 Ciele práce

### 1.2 Členenie práce

# Chapter 2

## Problematika

Preštylizovať: Vedci a výskumníci sa dlhodobo snažia pochopiť tajomstvá prírody. Východiskom tejto práce je spektrum pokusov a riešení snažiacich sa o modelovanie živých organizmov, predovšetkým oblasť nášho záujmu - hmyzu. Pôvodne sme mali pokračovať v riešení z roku 2011, model chodiaceho mravca vytvoreného ako diplomová práca. Tiež sme sa opierali o riešenia ilustrujúce a demonštrujúce použitie nami zvolených nástrojov.

### 2.1 Existujúce riešenia

TODO hexapod walking od Cruse, Schmitz, Schilling, Zollikofer

### 2.2 Východiskové riešenie

V roku 2011 vytvoril Andrej Riška virtuálny simulátor pre modelovanie správania mravca. Jeho simulátor mravca ako jeden z mála spájal model organizmu so skutočnými biologickými dátami organizmu. Jeho model bol získaný pomocou stereoskopie [ste]. Ako nástroj riešenia si vybral MRDS(Microsoft Robotics Developer Studio) [mrd], čo v tom čase bolo jedno z mála podporovaných voľne distribuovaných vývojových prostredí vhodných na prácu s reálnymi a simulovanými robotmi. MRDS používal programy napísané v C# a na zrealizovanie simulácií fyzikálny engine AGEIA PhysX.

Začal úpravami získaného 3D modelu mravca a priradil mu fyzikálny model (obr. 2.1), ktorý bolo potrebné vypočítať podľa jednotlivých anatomických častí. Následne implementoval rozhranie, ktoré dovoľovalo riadenie modelu. Na overenie správnosti do tohto riešenia zakomponoval multiagentový systém a spravil dva experimenty. Prvý dokazoval správnu spoluprácu fyzikálneho engine s vytvoreným modelom a druhý implementoval riadenie pomocou dvoch agentov komunikujúcich medzi sebou a rozhraním pomocou nepriamej komunikácie s prvkami subsumpčnej architektúry [Ris11]. V tom

čase bolo jeho riešenie pripravené na ďalšie rozvíjanie, napríklad ladenie správania alebo pridávanie senzorických vstupov. Riškové riešenie je dobrým priblížením problematiky, teórie a ako náčrt možného postupu. 3D model, s ktorým pracoval sa po rôznych úpravách použil aj v našej práci.

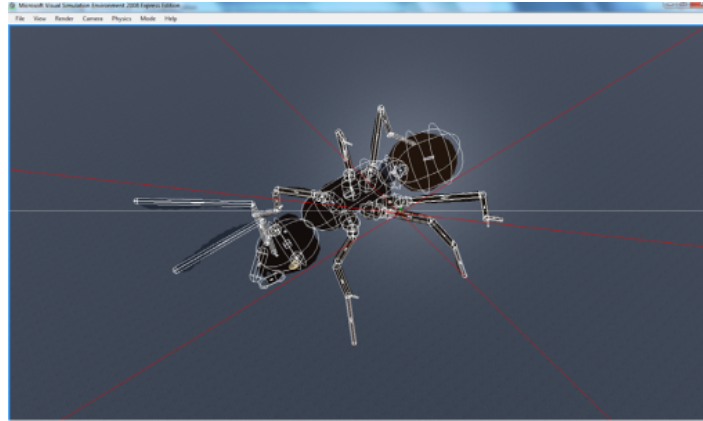


Figure 2.1: Spojenie fyzikálneho modelu s vizuálnym v zdedenom riešení. [Ris11].

# Chapter 3

## Nástroje

### 3.1 Úvod

Simulátory zohrávajú v robotike vitálnu úlohu ako nástroj na modelovanie reálneho sveta vo virtuálnom prostredí. Prinášajú hneď niekoľko výhod, práca s nimi je všeobecne rýchlejšia, lacnejšia, bezpečnejšia a menej frustrujúca. Nepotrebuje hardvér robota, zmeny jeho parametrov sú jednoduchšie, simulátor, naše telo a peňaženka zvláda experimentálne a nepredvídateľné správanie lepšie. Vďaka tomu sú ideálne na tvorbu a testovanie nových konceptov a algoritmov.

V zadaní práce sa predpokladalo rozvíjanie riešenia z roku 2011 spraveného v MRDS, prípadne jeho reimplementácia v modernejších nástrojoch. MRDS však od roku 2012 nebolo aktualizované a v roku 2014 Microsoft pozastavil činnosť jeho robotického skupiny [lay]. Projekt tak stratil podporu a následne aj komunitu. Preto sme si vybrali reimplementáciu v nástrojoch Gazebo [Gaza] a ROS [ROSa].

### 3.2 Simulátor Gazebo

Pri našom výbere simulátora sme hľadali moderný, rozšírený a voľne šíriteľný nástroj. Práve takýmto je Gazebo, realistický 3D simulátor robotov, ktorého začiatky siahajú do roku 2003. Bol vytvorený kvôli potrebe simulovania mobilných robotov v 3D prostredí, ktorí s prostredím môžu fyzikálne uspokojivo interagovať. Jeho 2D predchodcovia, napríklad Stage [Sta], sa zaoberali hlavne navigáciou kinematicky a geometricky jednoduchých objektov v statickom planárnom prostredí. Takéto simulátory boli rýchle a efektívne, no nedokázali simulovať robotov vo vzduchu či vo vode alebo manipulovanie robota s objektami prostredia.

Gazebo túto medzeru zaplnil, môže simulovať viacerých robotov naraz vo vnútornom aj vonkajšom 3D prostredí, interakciu robotov s prostredím, podporuje senzory a senzorickú odozvu.

Náčrt pôvodnej architektúry Gazebo (obr č. 3.1). World, čiže svet, reprezentuje množinu všetkých modelov a faktorov prostredia, ako sú napríklad gravitácia alebo svetlo. Každý model sa skladá aspoň z jedného tela (body, neskôr ho nahradil pojem link) a ľubovoľného počtu kĺbov a senzorov. Knižnice fyzikálnych enginov a grafiky komunikujú s Gazebo na najnižšej úrovni. Klient komunikuje s rozhraním cez zdieľanú pamäť [KH04].

Na modelovanie dynamiky tuhých telies využíval pôvodne len knižnicu ODE (Open Dynamics Engine) [ODE], neskôr boli pridané Bullet [Bul], Simbody [Sima] a DART [DAR]. Pri spustení Gazebo alebo vo world súbore môžeme simulátoru povedať, aký fyzikálny engine má byť použitý.

Na vizualizáciu simulácie a GUI boli v prvých verziách využívané knižnice OpenGL [Ope] a GLUT (OpenGL Utility Toolkit) [GLU]. V nových verziách simulátoru sa používa na účely renderovania engine OGRE [OGR] a na GUI framework Qt [Qt].

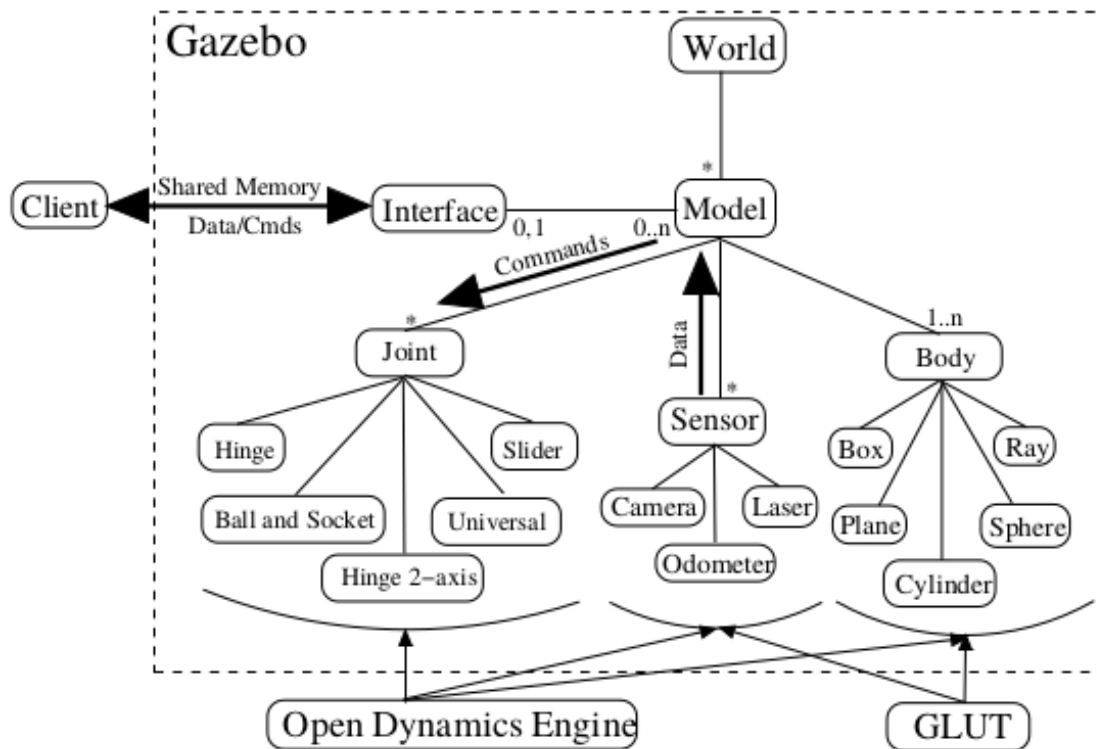


Figure 3.1: Pôvodná architektúra Gazebo [KH04].

### 3.2.1 Inštalácia

Simulátor Gazebo je otvorený voľne šíriteľný softvér. Má viac verzií, je dobré pracovať s najnovšou, aktuálne je to verzia 9. Vo väčšine prípadov sa inštaluje a používa na systémoch Linux distribúcie Ubuntu. My používame v tomto čase najaktuálnejšiu verziu 18, čitateľom odporúčame aby bola aspoň verzie 16. Je možné inštalovať a

používať ho aj na iných distribúciách Linuxu a tiež aj na Windowse alebo Macu, no nie je to vždy priamočiare a vyžaduje si to niekedy viac práce. Na webstránke projektu sú tutoriály aj pre tieto prípady [Gazc], my budeme popisovať a pracovať s Ubuntu.

Inštalácia sa skladá z niekoľko príkazov. Prvým sa pridá adresa Gazebo repozitárov do súboru `sources.list`, druhým sa zabezpečí správnosť zdroja. Tretím aktualizujeme všetky naše repozitáre, vrátane novopridaného. Až posledný príkaz vykoná samotnú inštaláciu:

```
sudo sh -c 'echo "deb
http://packages.osrfoundation.org/gazebo/ubuntu-stable
'lsb_release -cs' main" >
/etc/apt/sources.list.d/gazebo-stable.list'
wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key
add -
sudo apt-get update
sudo apt-get install gazebo9
```

Systém sa nás spýta na používateľské heslo a potvrdenie inštalácie. Správnosť inštalácie overíme príkazom, ktorým spustíme simulátor:

```
gazebo
```

Týmto príkazom sa v skutočnosti spustia dva navzájom komunikujúce programy:

- **gzserver** - server, hlavná časť simulátora, používa fyzikálny engine na simulovanie, generuje senzorické dáta
- **gzclient** - klient, pripája sa na server, zobrazuje GUI, vizualizuje simuláciu, môžeme cez neho interagovať so simuláciou, pracovať s modelmi

### 3.2.2 Prostredie

#### Grafické rozhranie

Spustíme Gazebo s hotovým demo súborom `gripper.world`:

```
gazebo worlds/gripper.world
```

Na tomto jednoduchom modeli sa zoznámime s grafickým rozhraním simulátora, prípadne môžeme spustiť a testovať iný predpripravený model zo zoznamu:

```
ls /usr/share/gazebo-9/worlds
```

Hlavnou časťou rozhrania (obr. č. 3.2) je scéna. Vľavo od scény sa nachádza ľavý panel, vpravo nájdeme pravý, ten je však po spustení simulátora skrytý, musíme ho najskôr rozbaľiť. Nad a pod scénou sa nachádzajú horný, respektíve dolný panel nástrojov. Na úplnom vrchu rozhrania sa tradične nachádza menu aplikácie.

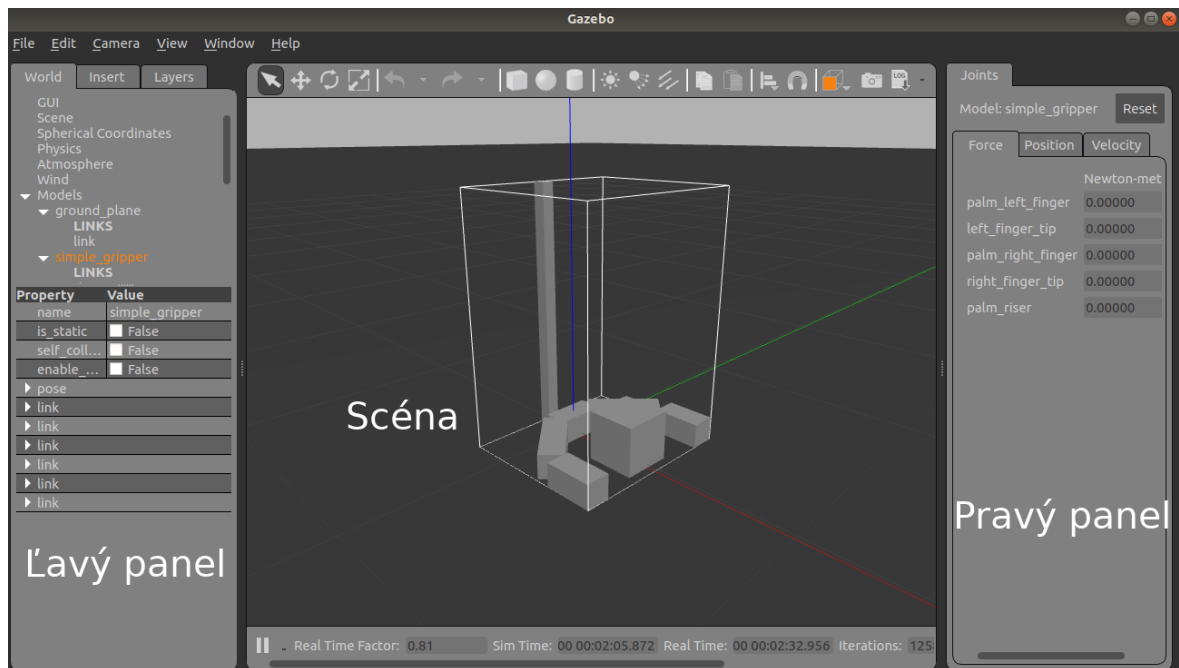


Figure 3.2: Gazebo GUI.

Na scéne sa vykresľuje svet simulácie, myšou s ňou môžeme interaktívne pracovať. Ľavý panel obsahuje tri karty:

- World - tu sú zobrazené entity a parametre nášho sveta, modely a ich štruktúra, svetlá, rôzne nastavenia zobrazenia scény, fyziky
- Insert - do scény môžeme pridávať modely z rôznych databáz, či už vlastné alebo z komunity
- Layers - modely môžeme zaradzovať do rozdielnych skupín viditeľnosti

Po nakliknutí modelu sa zobrazia jeho kĺby v pravom paneli. S jeho pohyblivými časťami môžeme manipulovať tak, že im zadávame rôzne fyzikálne parametre a pozorujeme zmenu v simulácii.

Cez spodný panel nástrojov pozastavujeme alebo spúšťame simuláciu. Tiež tu môžeme sledovať informácie o čase a dĺžke simulácie a FPS (frames per second) kvalite vykresľovania.

Horný panel nástrojov obsahuje najviac používané nástroje simulácie. Transláciu, rotáciu, škálovanie modelu, pridávanie jednoduchých objektov či svetiel. Nechýba tlačidlo dozadu alebo snímka obrazovky.

V aplikačnom menu sa nachádzajú nástroje na prácu so súborami, úpravu modelov, nastavenia simulácie a zobrazenia a pomoc. Grafické rozhranie je podrobnejšie popísané na webstránke projektu [Gazb].



## Súbory

Pri práci s Gazebo sa stretneme s dvoma typmi XML súborov popisujúcich simuláciu alebo jej časti:

- **.sdf** súbory - popisujú štruktúru a vlastnosti konkrétneho modelu robota
- **.world** súbory - popisujú celý svet simulácie vrátane nastavení fyziky, svetiel, scény, kamier, pluginov, modely sa do nich pridávajú buď referenciou na sdf súbor cez include tag, alebo sú popísané tak ako v konkrétnom sdf súbore

Gazebo je cez terminál spustiteľné len s world súbormi. V spustenej aplikácii môžeme importovať sdf modely do scény cez Insert, vytvárať, upravovať a exportovať ich cez Model Editor. Celú scénu je možné uložiť len ako súbor typu world. Do novovzniknutého súboru sa zapíšu nastavenia scény, ktoré vidíme vo World tabe ľavého panelu.

Databáza modelov obsahuje priečinky jednotlivých modelov (obr č. 3.3 vľavo), každý z nich obsahuje tieto položky (obr č. 3.3 vpravo) [Gazd]:

- **model.sdf** - povinný súbor popisujúci štruktúru a vlastnosti modelu
- **model.config** - povinný súbor obsahujúci meta dáta modelu - verzia sdf, autor a slovný popis modelu
- meshes, nepovinný priečinok s 3D súbormi modelu
- materials, nepovinný priečinok obsahujúci textúry, resp. materiály
- plugins, nepovinný priečinok obsahujúci súbory na riadenie modelu

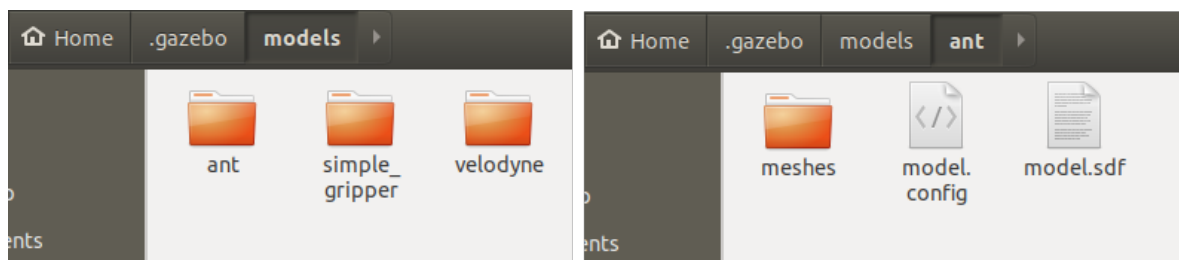


Figure 3.3: Štruktúra priečinkov databázy modelov Gazebo.

## Štruktúra sdf modelov

Model definovaný v sdf je kolekciou týchto častí [sdfb]:

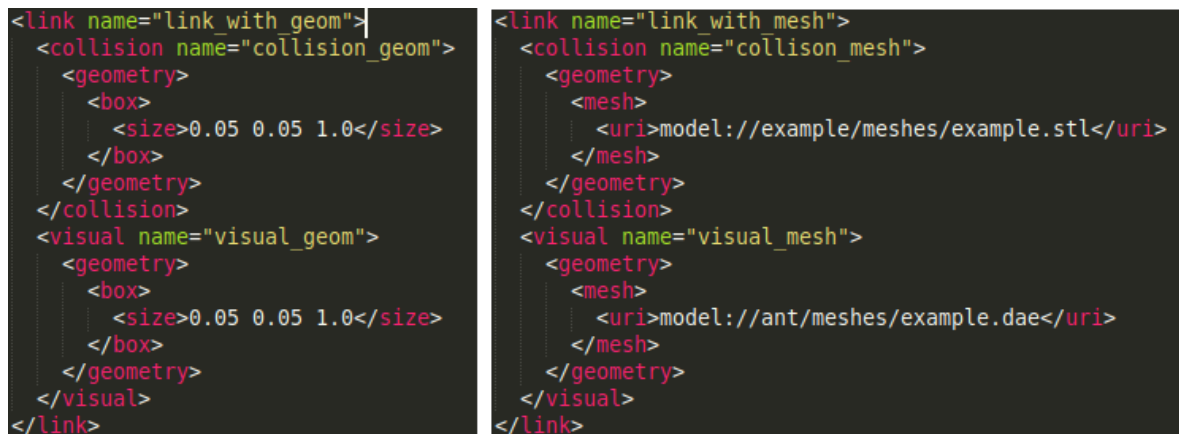
- **linky**(články) - fyzické časti modelu, obsahujú vizuálne, kolízne a inerciálne dáta

- **jointy**(klby) - rôzne spoje medzi článkami modelu
- **pluginy** - knižnice slúžiace na riadenie modelu

Všeobecný postup pri tvorbe sdf modelu je nasledovný [sdfb]:

1. pridanie článku
2. pridanie kolízneho elementu danému článku
3. pridanie vizuálneho elementu danému článku
4. pridanie inerciálnych vlastností danému článku
5. opakovanie tohto postupu pre všetky články modelu
6. pridanie klbov, ak existujú
7. pridanie pluginov, ak existujú

Vizuálne a kolízne elementy sú definované buď jednoduchými geometrickými tvarmi alebo sa odvolávajú na 3D súbory (obr č. 3.4) v meshes podpriechinku modelu. Podporované sú súbory OBJ (.obj) [obj], COLLADA (.dae) - COLLABorative Design Activity [dae], STL (.stl) - skratka stereolithography [stl].



```
<link name="link_with_geom">
  <collision name="collision_geom">
    <geometry>
      <box>
        <size>0.05 0.05 1.0</size>
      </box>
    </geometry>
  </collision>
  <visual name="visual_geom">
    <geometry>
      <box>
        <size>0.05 0.05 1.0</size>
      </box>
    </geometry>
  </visual>
</link>
```

```
<link name="link_with_mesh">
  <collision name="collision_mesh">
    <geometry>
      <mesh>
        <uri>model://example/meshes/example.stl</uri>
      </mesh>
    </geometry>
  </collision>
  <visual name="visual_mesh">
    <geometry>
      <mesh>
        <uri>model://ant/meshes/example.dae</uri>
      </mesh>
    </geometry>
  </visual>
</link>
```

Figure 3.4: Vľavo použitie geometrického tvaru, vpravo referencia na 3D súbor pri vizuálnom a kolíznom elemente sdf.

V sdf môžeme spájať články týmito typmi klbov [sdff]:

- **fixed** - pevný spoj, spája dva linky do nemennej vzájomnej polohy
- **revolute** - kĺb v otáčajúci sa na jednej osi
- **revolute2** - dva revolute kĺby v sérii
- **prismatic** - posuvný kĺb na jednej osi

- ball - guľový kĺb
- universal - guľový kĺb obmedzujúci pohyb na jednej osi
- screw - závitový kĺb posúvajúci sa a otáčajúci sa na jednej osi
- gearbox - sprevodované revolúte jointy

### 3.3 Potreba riadiacej architektúry

V začiatkoch našej práce sme sa zoznámili a pracovali len s Gazebo. Po čase sme prišli nato, že samotný simulátor nám stačiť nebude. Naš model je verne poskladaný a cez grafické rozhranie simulátora s ním môžeme aj pohybovať, no predmetom práce je jeho riadenie, konkrétne schopnosť chôdze. Aby to náš model mravca dokázal, potrebujeme simulátor doplniť riadiacou architektúrou. Vtedy prišlo vysvetlenie, prečo sa často v zdrojoch a na fórach objavuje spoločne s Gazebom pojem ROS, znamenajúci Robot Operating System.

### 3.4 ROS

ROS je otvorený framework pre prácu s robotmi. Je to kolekcia nástrojov a knižníc, umožňujúca tvorbu komplexných a robustných robotických systémov naprieč rozličnými platformami.

Podobne ako pri našom simulátore, jeho začiatky sa datujú do polky 2000's [QGS15]. V tej dobe sa veľa jednotlivcov a inštitúcií venujúcich sa robotike hralo chtiac či nechtiac na svojom pieskovisku. Hlavným dôvodom jeho vzniku bola potreba vytvoriť nástroj na riadenie robotov, ktorý by bol pre komunitu a z komunity. Časom sa ukázalo, že postoj a postup zvolený pri ROS robotike nesmierne pomohol a posunul ju správnym smerom. Dôkazom sú tisíce používateľov, od gárážových nadšencov, vzdelávacie a priemyselné inštitúcie, až po NASA [BGE<sup>+</sup>16].

Jeho charakteristické črty sú [QCG<sup>+</sup>09]:

- **rovný s rovným** - riadenie je v ňom distribuované, tvorené mnohými navzájom komunikujúcimi programami
- **nástrojovo založený** - skladá sa z veľa komponentov s úzkym zameraním namiesto jedného vývojového a runtime prostredia
- **viacjazyčný** - programy môžeme písať v akomkoľvek jazyku, pre ktorý existuje klientská knižnica [rosl]

- **štíhly (thin)** - vytvárané knižnice sú nezávislé na ROS, ľahšie testovanie, opakovanosť kódu
- **otvorený a voľne šíriteľný** - pod BSD licenciou, komerčné aj nekomerčné použitie

### 3.4.1 Inštalácia

ROS, podobne ako samotné Ubuntu, vydáva nové verzie softvéru ako distribúcie. V čase našej práce je najnovšia a zároveň odporúčaná verzia Melodic Morenia. Popisujeme a pracujeme s touto distribúciou.

Postup inštalácie je podobný ako pri Gazebo:

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu trusty main"
> /etc/apt/sources.list.d/ros-latest.list'
wget http://packages.ros.org/ros.key -O - | sudo apt-key add -
sudo apt-get update
sudo apt-get install ros-melodic-desktop-full python-rosinstall
```

Aby ROS fungoval, musíme ešte inicializovať rosdep a takzvané enviroment variables:

```
sudo rosdep init
rosdep update
echo "source /opt/ros/melodic/setup.bash" » ~/.bashrc
source ~/.bashrc
```

### 3.4.2 Architektúra TODO

### 3.4.3 Prostredie TODO

## 3.5 Použitie Gazebo+ROS TODO

# Chapter 4

## Model mravca

V tejto kapitole opíšeme anatómiu mravca a ukážeme ako sme postupovali v úpravách zdedeného modelu, aby bol pripravený na simuláciu.

### 4.1 Anatómia mravca TODO

### 4.2 Dedenie a úprava modelu

#### 4.2.1 Formáty modelov

Model mravca sme zdedili v dvoch formátoch, OBJ a VRML2 [vrml]. Tento model bol získaný metódou stereoskopie. V oboch prípadoch šlo spolu o 52 súborov jednotlivých častí mravca. Nato aby sme súbory mohli použiť na simuláciu v Gazebo, je potrebné aby sdf model dostal súbory v niektorom z formátov obj, dae alebo stl.

Pri otvorení obj súborov sme ale hneď zistili, že všetky sú vycentrované na začiatok globálnych súradníc a tým pádom z nich nevieme zistiť informáciu o pozícii údov vzhľadom k telu mravca a samotné skladanie modelu by bola nepomerne náročnejšie. Pri modeli vo VRML2 sme sa najskôr presvedčili pomocou nástroja FreeWRL [fre], že zdedený model je korektný. Pri importovaní hlavného súboru ant.wrl sa naozaj zobrazil model mravca. Začali sme sa tak pokúšať skonvertovať nejakým spôsobom model vo formáte VRML2 do nami použiteľného formátu.

#### 4.2.2 Konvertovanie modelu

Model mravca vo formáte VRML2 sa skladal z 52 wrl súborov údov a hlavného súboru ant.wrl, ktorý sa na tieto súbory odvolával a spájal ich do hierarchie (obr č. 4.1). Pokúšali sme sa konvertovať tieto súbory cez rôzne online nástroje, meshconv [mesc], simtrans [simb]. Tiež sme sa pokúšali o import do Blenderu [ble], Meshlabu [mesd], CAD nástrojov. Nič zo spomenutého nefungovalo podľa našich predstáv. Neskôr sme

```

27 EXTERNPProto ant-head[] "ant-head.wrl"
28 EXTERNPProto ant-head-dots[] "ant-head-dots.wrl"
29 EXTERNPProto ant-eye-left[] "ant-eye-left.wrl"
30 EXTERNPProto ant-eye-right[] "ant-eye-right.wrl"
31 EXTERNPProto ant-scape-left[] "ant-scape-left.wrl"
32 EXTERNPProto ant-scape-right[] "ant-scape-right.wrl"
33 EXTERNPProto ant-flagellum-left[] "ant-flagellum-left.wrl"
34 EXTERNPProto ant-flagellum-right[] "ant-flagellum-right.wrl"
35 EXTERNPProto ant-scape-bed-left[] "ant-scape-bed-left.wrl"
36 EXTERNPProto ant-scape-bed-right[] "ant-scape-bed-right.wrl"
37 EXTERNPProto ant-mandible-left[] "ant-mandible-left.wrl"
38 EXTERNPProto ant-mandible-right[] "ant-mandible-right.wrl"
39 EXTERNPProto ant-thorax[] "ant-thorax.wrl"
40 EXTERNPProto ant-gaster[] "ant-gaster.wrl"
41 EXTERNPProto ant-petiole[] "ant-petiole.wrl"
42 EXTERNPProto ant-node[] "ant-node.wrl"
43 EXTERNPProto ant-cox-front-left[] "ant-cox-front-left.wrl"
44 EXTERNPProto ant-cox-front-right[] "ant-cox-front-right.wrl"
45 EXTERNPProto ant-femur-front-left[] "ant-femur-front-left.wrl"
46 EXTERNPProto ant-femur-front-right[] "ant-femur-front-right.wrl"
47 EXTERNPProto ant-femur-trchanter-front-left[] "ant-femur-trchanter-front-left.wrl"
48 EXTERNPProto ant-femur-trchanter-front-right[] "ant-femur-trchanter-front-right.wrl"
49 EXTERNPProto ant-tibia-front-left[] "ant-tibia-front-left.wrl"
50 EXTERNPProto ant-tibia-front-right[] "ant-tibia-front-right.wrl"
51 EXTERNPProto ant-tibia-strip-front-left[] "ant-tibia-strip-front-left.wrl"
52 EXTERNPProto ant-tibia-strip-front-right[] "ant-tibia-strip-front-right.wrl"
53 EXTERNPProto ant-tarsus-front-left[] "ant-tarsus-front-left.wrl"
54 EXTERNPProto ant-tarsus-front-right[] "ant-tarsus-front-right.wrl"
55 EXTERNPProto ant-cox-middle-left[] "ant-cox-middle-left.wrl"
56 EXTERNPProto ant-cox-middle-right[] "ant-cox-middle-right.wrl"
57 EXTERNPProto ant-femur-middle-left[] "ant-femur-middle-left.wrl"
58 EXTERNPProto ant-femur-middle-right[] "ant-femur-middle-right.wrl"
59 EXTERNPProto ant-femur-trchanter-middle-left[] "ant-femur-trchanter-middle-left.wrl"
60 EXTERNPProto ant-femur-trchanter-middle-right[] "ant-femur-trchanter-middle-right.wrl"
61 EXTERNPProto ant-tibia-middle-left[] "ant-tibia-middle-left.wrl"
62 EXTERNPProto ant-tibia-middle-right[] "ant-tibia-middle-right.wrl"
63 EXTERNPProto ant-tibiole[] "ant-tibiole.wrl"

```

Figure 4.1: Odvolávanie sa na wrl súbory údov(vľavo) a hierarchia údov(vpravo) v hlavnom ant.wrl súbore. [Ris11]

narazili na nástroj VrmMerge [vrmb], za pomoci ktorého bolo možné jednotlivé údy skonvertovať do formátu X3D [x3d]. Tie nebolo problém importovať do Blenderu a následne ich exportovať ako COLLADA alebo STL súbory. Dôležité pri tom bolo, že zostali zachované údaje o ich globálnej pozícii. Údaje o hierarchii údov z hlavného wrl súboru boli ale stratené. Keď sme všetky výsledné súbory importovali do Blendru, vznikol nám korektný model mravca (obr č. 4.2).

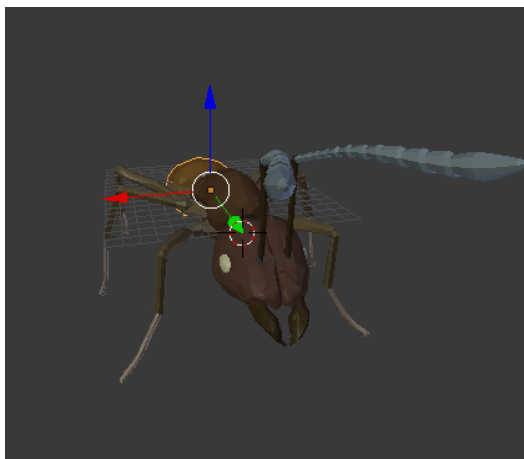


Figure 4.2: Poskladaný ant.blend z COLLADA súborov

### 4.2.3 Składanie modelu

Model mravca vytvárame podľa postupu popísaného v časti o použitých nástrojoch. Postupne pridávame články údov mravca, dávame im referencie na vizuálne a kolízne COLLADA súbory. Začínáme od trupu mravca (obr č. 4.3), správnosť postupu kontrolujeme importovaním modelu do Gazebo GUI (obr č. 4.4). Do adresáru modelu

```
1 <?xml version="1.0" ?>
2 <sdf version="1.6">
3   <model name="ant">
4     <static>false</static>
5     <pose>0 0 4 0 0 0</pose>
6
7     <link name="ant_thorax">
8       <gravity>0</gravity>
9       <collision name="ant_thorax_collision">
10        <geometry>
11          <mesh>
12            <uri>model://ant/meshes/ant-thorax.dae</uri>
13          </mesh>
14        </geometry>
15      </collision>
16      <visual name="ant_thorax_visual">
17        <geometry>
18          <mesh>
19            <uri>model://ant/meshes/ant-thorax.dae</uri>
20          </mesh>
21        </geometry>
22      </visual>
23    </link>
24
25  </model>
26 </sdf>
27
```

Figure 4.3: Pridanie thorax článku do sdf.

na úrovni sdf súboru modelu pridáme aj config súbor, ktorý opisuje model, autora a použitú verziu sdf.

Týmto spôsobom postupne pridáme všetky články údov mravca. Vznikne niekoľko storiadkový súbor popisujúci všetky údy mravca. Zatiaľ sú to ale žiadnym spôsobom nespojené články, ktorým sme priradili vizuálne a kolízne súbory. Taktiež im chýbajú inerciálne dáta.

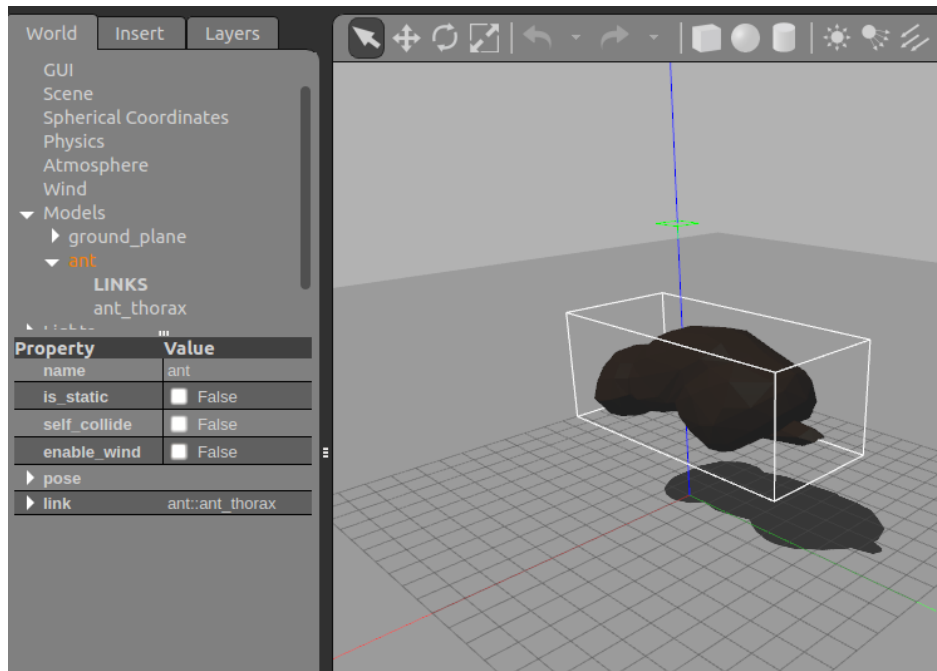


Figure 4.4: Iportovanie thorax modelu do Gazebo GUI.

Ďalej je potrebné vytvoriť hierarchiu medzi článkami. Medzi všetky články okrem trupu(thorax) a 6 počiatočných častí nôh(cox) použijeme fixný kĺb. Tak medzi nimi spravíme pevný spoj. Trup a nožičky spojíme revolúte kĺbom (obr č. 4.5).

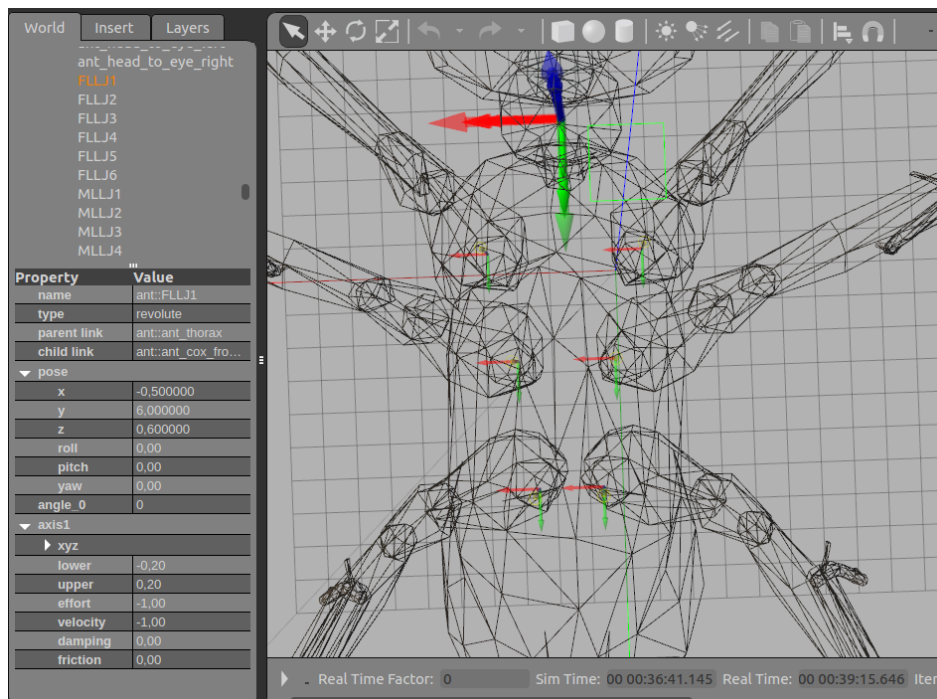


Figure 4.5: Viditeľné revolúte kĺby spájajúce trup s nohami.

Články tak máme pospájané, zostáva nám priradiť im inertial dáta. Sú to tieto: hmotnosť, pozícia ťažiska a matica zotrvačnosti [sdfa]. K exaktným dátam o hmot-



nosti jednotlivých údov sme sa nedostali, spravíme odhad. Pozíciu ťažiska a maticu zotrvačnosti získame cez online nástroj Mesh Cleaner [mesb] (obr č. 4.6), ktorému odovzdáme súbor údu. Toto spravíme pre všetkých 52 súborov.

Mesh Cleaner akceptuje len vodotesné 3D súbory. Náš model obsahoval niekoľko súborov, ktoré také neboli. Museli sme ich preto opraviť za pomoci Blenderu a Meshlabu [mesa]. Takým bol napríklad súbor tykadla (obr č. 4.7). Na zaplátanie diery v 3D súbore sme použili voľne dostupné rozšírenie Blenderu zvané 3D Print Toolbox [3dp], konkrétne jeho funkciu make manifold. Aj tu sa ale vyskytli ďalšie problémy. Niektoré zo súborov obsahovali vnútorné plochy – interior faces. Tieto sme museli manuálne odstrániť v Meshlabe (obr č. 4.8).

Po pridaní inerciálnych dát všetkým článkom je model pripravený na simuláciu (obr č. 4.9).

## Mesh Cleaner

In case everything went well, you should see the volume of the mesh, its center of mass and inertia matrix.



Figure 4.6: Nástroj Mesh Cleaner počíta inerciálne dáta konkrétneho 3D súboru [mesb].

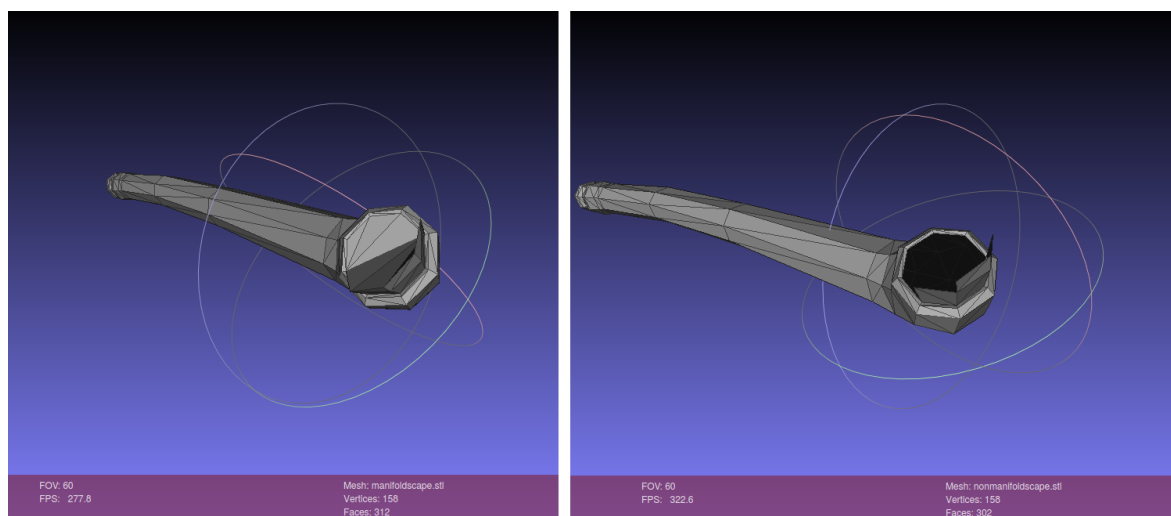


Figure 4.7: Nevodotesné tykadlo pred(vľavo) a po zaplátaní(vpravo) v Meshlabe.

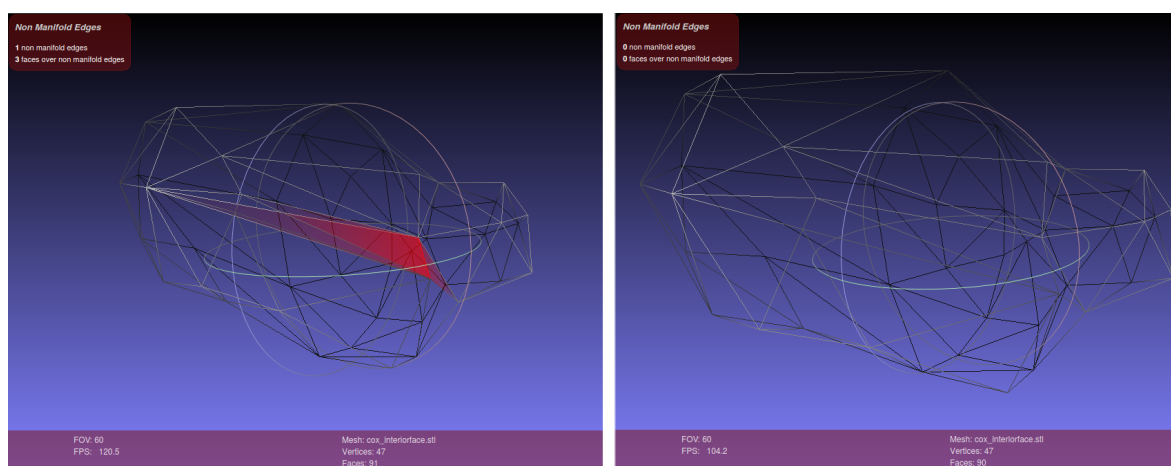


Figure 4.8: Odstránenie vnútornej tváre v 3D súbore v Meshlabe, pred(vľavo) a po(vpravo).

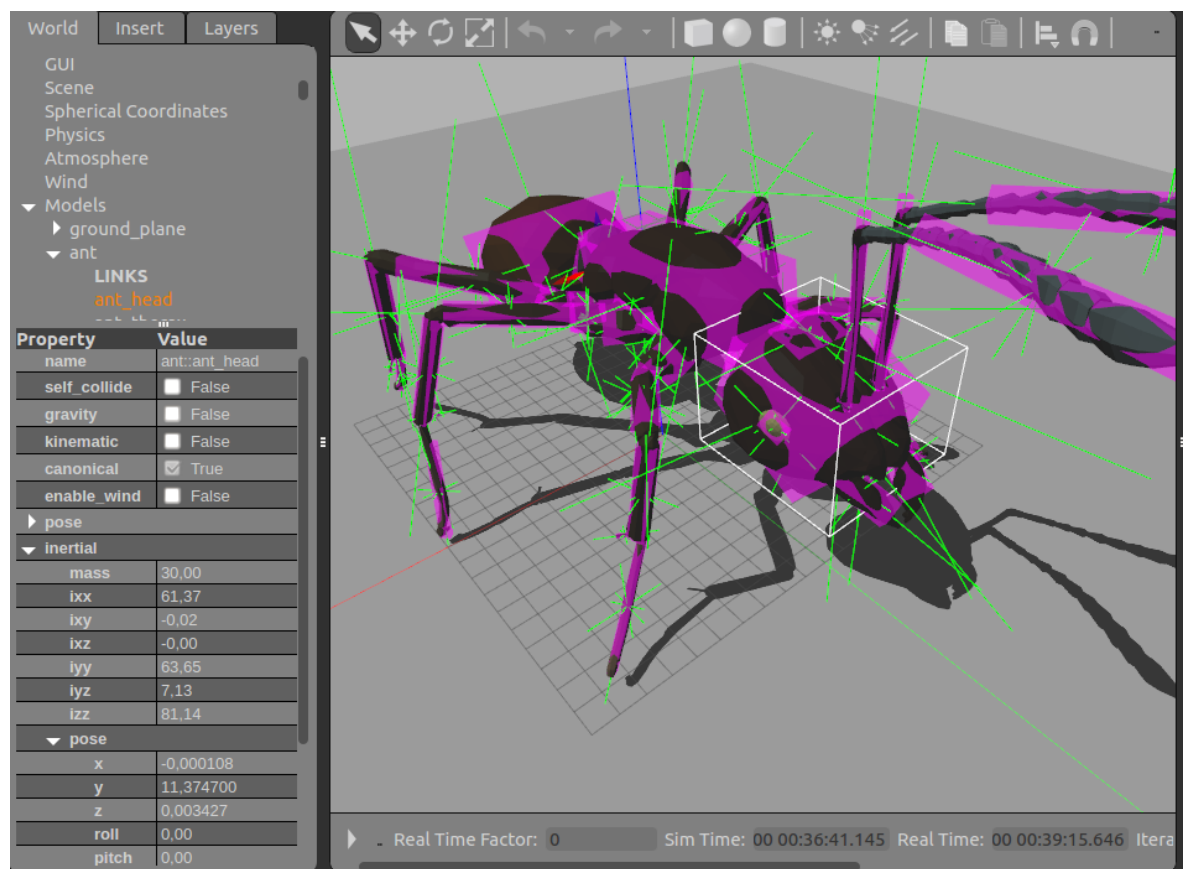


Figure 4.9: Model s kompletnými dátami zobrazený v Gazebo je pripravený na simuláciu. Sú zobrazené inerciálne dáta a ťažiská článkov.

# Chapter 5

## Simulácia

# Záver

# Bibliography

- [3dp] 3D Print Toolbox nástroj. <https://github.com/caretdashcaret/MeshRepairFor3DPrinting>. Navštívené: december 2018.
- [anaa] Ant anatomy. <https://flrec.ifas.ufl.edu/media/flrecifasufledu/pdfs/pestants/AntAnatomy.pdf>. Navštívené: december 2018.
- [anab] Ant anatomy, ask a biologist. <https://askabiologist.asu.edu/explore/ant-anatomy>. Navštívené: december 2018.
- [BGE<sup>+</sup>16] Julia Badger, Dustin Gooding, Kody Ensley, Kimberly Hambuchen, and Allison Thackston. *ROS in space: A case study on robonaut 2*, volume 625, pages 343–373. 02 2016.
- [ble] Blender nástroj. <https://www.blender.org/>. Navštívené: december 2018.
- [Bul] Bullet. <https://github.com/bulletphysics/bullet3>. Navštívené: január 2019.
- [CS05] Holk Cruse and Malte Schilling. *First order and second order embodiment – robots with the ability to plan ahead*. Biefeld University, 2005.
- [dae] COLLADA formát. <https://en.wikipedia.org/wiki/COLLADA>. Navštívené: december 2018.
- [DAR] DART. <https://dartsim.github.io/>. Navštívené: január 2019.
- [enc] Formica cinerea Mayr, 1853 - Encyclopedia of Life. <http://www.eol.org/pages/400890>. Navštívené: december 2018.
- [fre] FreeWRL nástroj. <http://freewrl.sourceforge.net/>. Navštívené: december 2018.
- [Gaza] Gazebo. <http://gazebo-sim.org/>. Navštívené: január 2019.
- [Gazb] GazGUI. <http://gazebo-sim.org/tutorials?tut=guided%5Fb2>. Navštívené: január 2019.

- [Gazc] GazInstall. <http://gazebo-sim.org/tutorials?cat=install>. Navštívené: január 2019.
- [Gazd] GazStruct. [http://gazebo-sim.org/tutorials?tut=model\\_structure](http://gazebo-sim.org/tutorials?tut=model_structure). Navštívené: január 2019.
- [GLU] GLUT. <https://www.opengl.org/resources/libraries/glut/>. Navštívené: január 2019.
- [KH04] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3, Sep. 2004.
- [lay] Microsoft lays off Robotics team. [https://www.roboticsbusinessreview.com/research/microsoft\\_lays\\_off\\_robotics\\_team/](https://www.roboticsbusinessreview.com/research/microsoft_lays_off_robotics_team/). Navštívené: december 2018.
- [mesa] Let's talk mesh repair. <https://caret-dash-caret.com/2014/12/04/lets-talk-mesh-repair/>. Navštívené: december 2018.
- [mesb] Mesh Cleaner nástroj. [https://www.hamzamerzic.info/mesh\\_cleaner/](https://www.hamzamerzic.info/mesh_cleaner/). Navštívené: december 2018.
- [mesc] Meshconv nástroj. <https://www.patrickmin.com/meshconv/>. Navštívené: december 2018.
- [mesd] Meshlab nástroj. <http://www.meshlab.net/>. Navštívené: december 2018.
- [mrd] MRDS. [https://docs.microsoft.com/en-us/previous-versions/microsoft-robotics/bb483024\(v=msdn.10\)](https://docs.microsoft.com/en-us/previous-versions/microsoft-robotics/bb483024(v=msdn.10)). Navštívené: december 2018.
- [MSFM16] Anil Mahtani, Luis Sanchez, Enrique Fernandez, and Aaron Martinez. *Effective Robotics Programming with ROS - Third Edition*. Packt Publishing, 3rd edition, 2016.
- [obj] Wavefront obj formát. [https://en.wikipedia.org/wiki/Wavefront\\_obj\\_file](https://en.wikipedia.org/wiki/Wavefront_obj_file). Navštívené: december 2018.
- [ODE] ODE. <https://www.ode.org/>. Navštívené: január 2019.
- [OGR] OGRE. <https://www.ogre3d.org/>. Navštívené: január 2019.
- [Ope] OpenGL. <https://www.opengl.org/>. Navštívené: január 2019.

- [QCG<sup>+</sup>09] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.
- [QGS15] Morgan Quigley, Brian Gerkey, and William D. Smart. *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*. O'Reilly Media, Inc., 1st edition, 2015.
- [Qt] Qt. <https://www.qt.io/>. Navštívené: január 2019.
- [Ris11] Andrej Riska. Modelovanie správania živých systémov, FMFI UK, diplomová práca, 2011.
- [ROSa] ROS. <http://wiki.ros.org/>. Navštívené: január 2019.
- [rosb] rosmultiling. <http://wiki.ros.org/Client%20Libraries>. Navštívené: január 2019.
- [Sch07] Josef Schmitz. *Neurobiological foundations of hexapod locomotion in insects and robots*. Biefeld University, 2007.
- [sdfa] SDF inertial tutoriál. <http://gazebo-sim.org/tutorials?tut=inertia>. Navštívené: december 2018.
- [sdfb] SDF model tutoriál. [http://gazebo-sim.org/tutorials?tut=build\\_model](http://gazebo-sim.org/tutorials?tut=build_model). Navštívené: december 2018.
- [sdfe] SDF format. <http://sdformat.org/>. Navštívené: december 2018.
- [sdfe] SDF format Joints. <http://sdformat.org/spec?elem=joint>. Navštívené: december 2018.
- [SHSC13] Malte Schilling, Thierry Hoinville, Josef Schmitz, and Holk Cruse. Walknet, a bio-inspired controller for hexapod walking. *Biological Cybernetics*, 107(4):397–419, 2013.
- [Sima] Simbody. <https://github.com/simbody/simbody>. Navštívené: január 2019.
- [simb] Simtrans nástroj. <http://fkanehiro.github.io/simtrans/html/simtrans.html>. Navštívené: december 2018.
- [SPSS14] Axel Schneider, Jan Paskarbeit, Malte Schilling, and Josef Schmitz. Hector, a bio-inspired and compliant hexapod robot. In Armin Duff, Nathan F. Lepora, Anna Mura, Tony J. Prescott, and Paul F. M. J. Verschure, editors,



*Living Machines*, volume 8608 of *Lecture Notes in Computer Science*, pages 427–429. Springer, 2014.

- [Sta] Stage. <http://wiki.ros.org/stage>. Navštívené: január 2019.
- [ste] Stereoscopy, wikipedia. <https://en.wikipedia.org/wiki/Stereoscopy>. Navštívené: december 2018.
- [stl] STL formát. [https://en.wikipedia.org/wiki/STL\\_\(file\\_format\)](https://en.wikipedia.org/wiki/STL_(file_format)). Navštívené: december 2018.
- [urd] URDF formát. <http://wiki.ros.org/urdf>. Navštívené: december 2018.
- [vrma] VRML formát. <https://en.wikipedia.org/wiki/VRML>. Navštívené: december 2018.
- [vrmb] VrmMerge nástroj. <http://www.deem7.com/vrmlmerge.php>. Navštívené: december 2018.
- [x3d] X3D formát. <https://en.wikipedia.org/wiki/X3D>. Navštívené: december 2018.
- [Zol94] Christoph Zollikofer. Stepping patterns in ants. *J Exp Biol*, 192(1):95–106, July 1994.

# List of Figures

2.1	Riřkov model . . . . .	3
3.1	Architektúra Gazebo . . . . .	5
3.2	Gazebo GUI . . . . .	7
3.3	Databáza modelov Gazebo . . . . .	8
3.4	Vizuálny a kolízny element v sdf . . . . .	9
4.1	Ant VRML súbor . . . . .	13
4.2	Poskladaný Ant blend . . . . .	13
4.3	Pridanie thorax článku do SDF . . . . .	14
4.4	Import thorax modelu do Gazebo . . . . .	15
4.5	Revolute kĺby spájajúce trup s nohami . . . . .	15
4.6	Nástroj Mesh Cleaner počíta inerciálne dáta . . . . .	16
4.7	Zaplátanie 3D súboru v Meshlabe . . . . .	17
4.8	Odstránenie vnútornej tváre v 3D súbore v Meshlabe . . . . .	17
4.9	Model s kompletnými dátami zobrazený v Gazebo . . . . .	18