Robot programming, Simulation and Environment using Choreonoid for Humanoid Beginner

2022/11/28

Yohei Kakiuchi

Toyohashi University of Technology

Outline

- Robot competitions using Simulation
- General information of Choreonoid
- Connecting to other system
- Development system on Choreonoid
- Learning Robot Programming

Outline

- Robot competitions using Simulation
- General information of Choreonoid
- Connecting to other system
- Development system on Choreonoid
- Learning Robot Programming

Robot competition using simulator

- RoboCup https://www.robocup.org/
  - Soccer(sim) developed multi agent simulator, Rescue(sim), @Home, Industrial
- DARPA Challenge
  - Grand(2005) / Urban(2007) Challenge
  - Robotics (Virtual2013, Trial2013, Final2015) Challenge
  - Subterranean Challenge (2017-2021) (Trial?)

  https://en.wikipedia.org/wiki/DARPA_Grand_Challenge
- JVRC (Japan Virtual Robotics Challenge) 2015
- WRS2020 (world robot summit)
  - Tunnel disaster challenge
- HVAC (Humanoid Virtual Athletics Challenge) << This WS

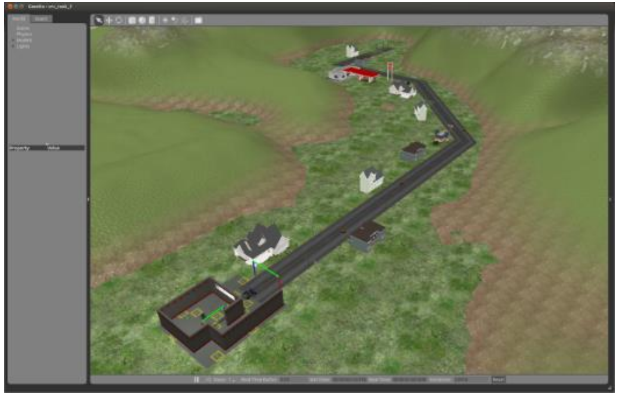
Robot competition using simulator

- Why simulator is used at competitions?
- Difficulties in real robots
  - Hardware (preparations robot, maintenance, environment)
- Wider variety of participants
  - Expert, Novice (experience)
  - Researcher, Developer and Hobbyist (profession)
  - Software, Hardware and Systems (speciality, interest)
- Targets (organizer, participants)
  - Boosting humanoid robots research
  - Testing new algorithms
  - Testing new hardware
  - Testing new integrate system

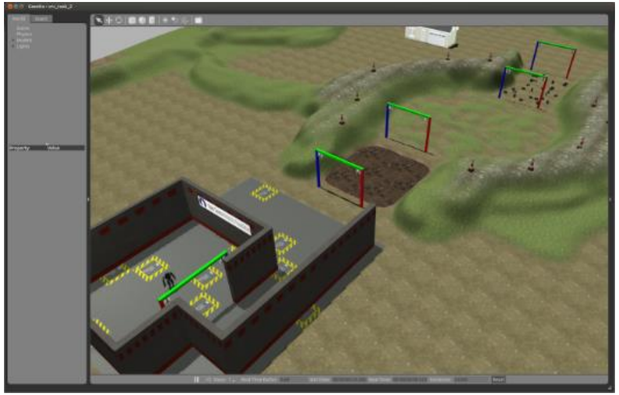
Virtual Robotics Challenge 2013

- Using Gazebo as simulator
  - Gazebo running in server, participants connecting from their site
- Target of competition
  - Required to solve real tasks
  - Control system
  - Novel robot interface
  - Share autonomous algorisms and operator's input

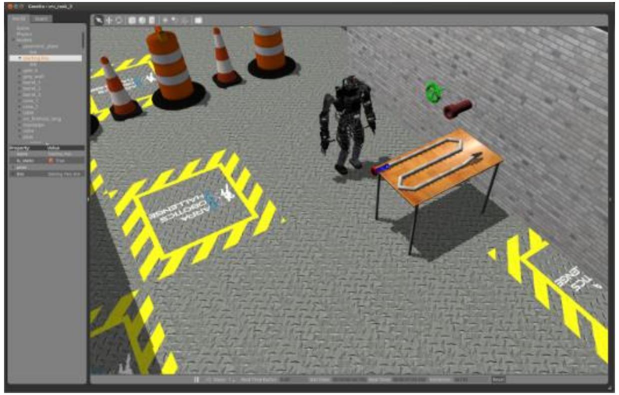
Virtual Robotics Task 1



Virtual Robotics Task 2



Virtual Robotics Task 3

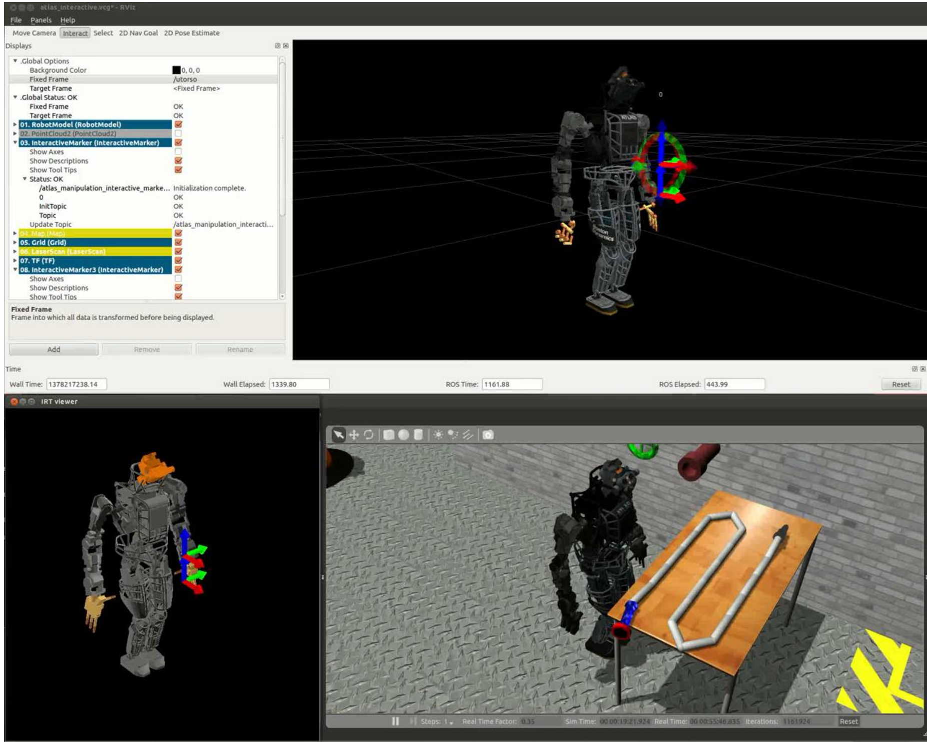


Robot control environment using Gazebo (VRC)

- Gazebo (http://gazebosim.org/)
  - Dynamics engine (ODE base)
  - Various environments (drcsim)
  - Highly compatible with ROS (gazebo-ros-pkgs)

<table>
<tr><td colspan="2">RVIZ (ROS visualization and controlling by marker)</td></tr>
<tr><td>Model expression on EusLisp</td><td>In Gazebo</td></tr>
</table>

x15



```
▼ .Global Options
    Background Color                          0, 0, 0
    Fixed Frame                               /utorso
    Target Frame                              <Fixed Frame>
▼ .Global Status: OK
    Fixed Frame                               OK
    Target Frame                              OK
▶ 01. RobotModel (RobotModel)                 ☑
▶ 02. PointCloud2 (PointCloud2)               □
▼ 03. InteractiveMarker (InteractiveMarker)   ☑
    Show Axes                                 □
    Show Descriptions                         ☑
    Show Tool Tips                            ☑
  ▼ Status: OK
      /atlas_manipulation_interactive_marke...  Initialization complete.
      0                                       OK
      InitTopic                               OK
      Topic                                   OK
    Update Topic                              /atlas_manipulation_interacti...
▶ 04. Map (Map)                               ☑
▶ 05. Grid (Grid)                             ☑
▶ 06. LaserScan (LaserScan)                   ☑
▶ 07. TF (TF)                                 ☑
▼ 08. InteractiveMarker3 (InteractiveMarker)  ☑
    Show Axes                                 □
    Show Descriptions                         ☑
    Show Tool Tips                            ☑
```

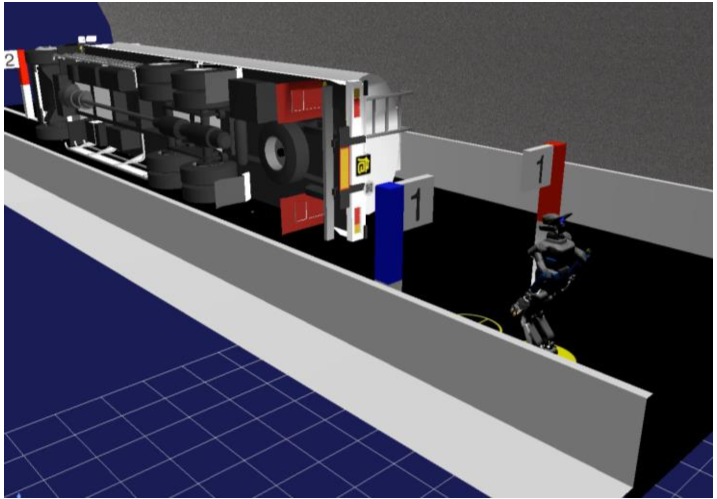
Fixed Frame
Frame into which all data is transformed before being displayed.

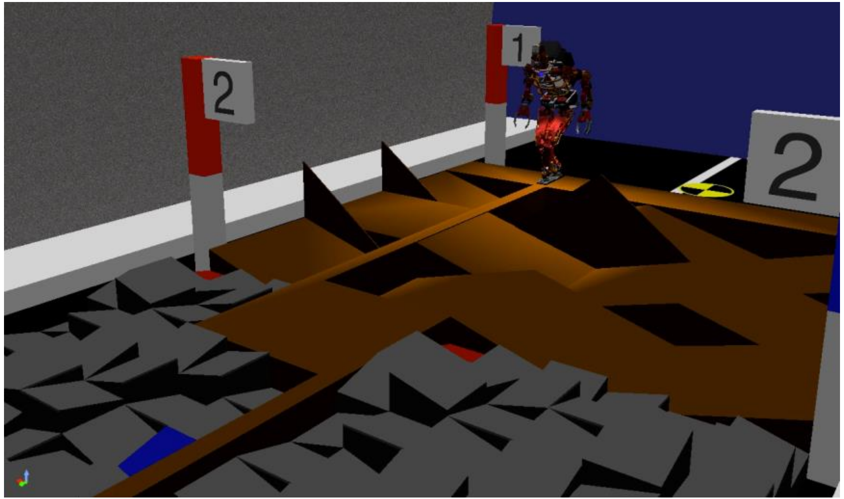
Time
Wall Time: 1378217238.14 Wall Elapsed: 1339.80 ROS Time: 1161.88 ROS Elapsed: 443.99

Steps: 1 Real Time Factor: 0.35 Sim Time: 00 00:19:21.924 Real Time: 00 00:55:46.835 Iterations: 1161924

JVRC(Japan Virtual Robotics Challenge)

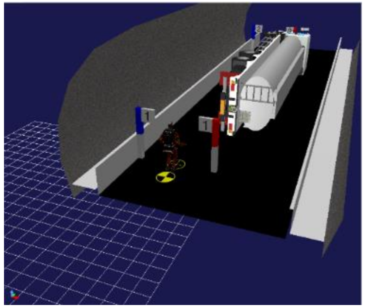
- Computer Simulation Competition for Disaster Response Robots
  - Target task: Disasters in tunnels
  - Oct 7～10, 2015
  - Simulator: Choreonoid
  - Participating Teams: 12

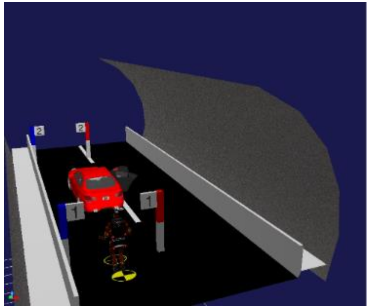


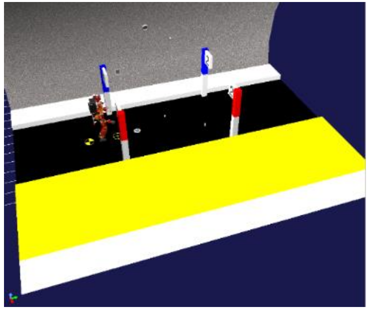


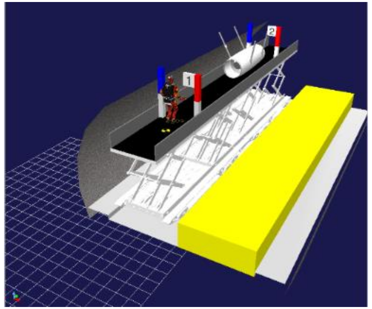
JVRC(Japan Virtual Robotics Challenge) Tasks

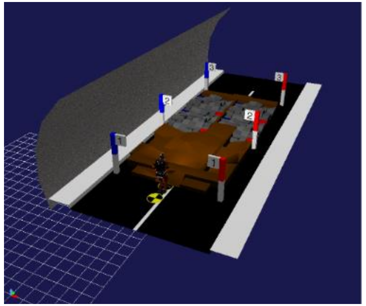
- Basic mobility in narrow areas
- Manipulation (heavy objects)
- Inspection
- Searching for missing persons

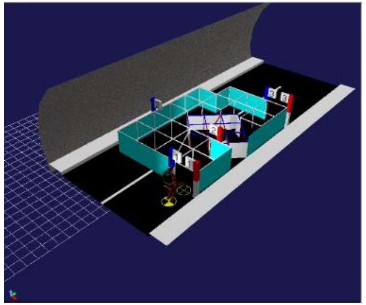


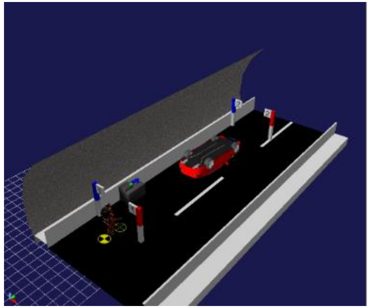


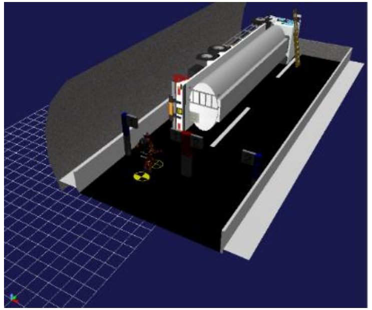


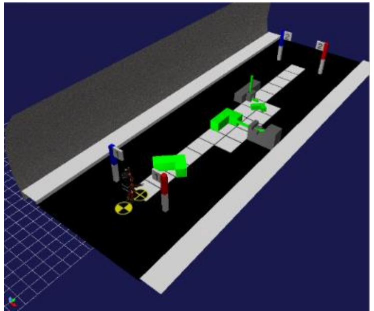


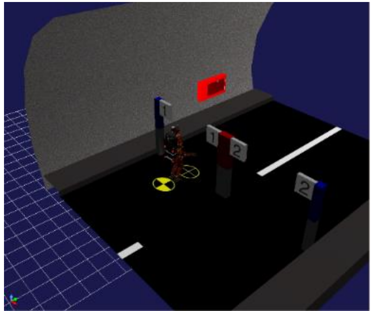






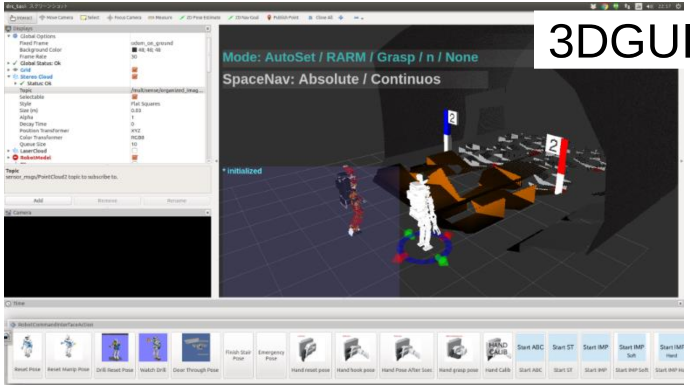


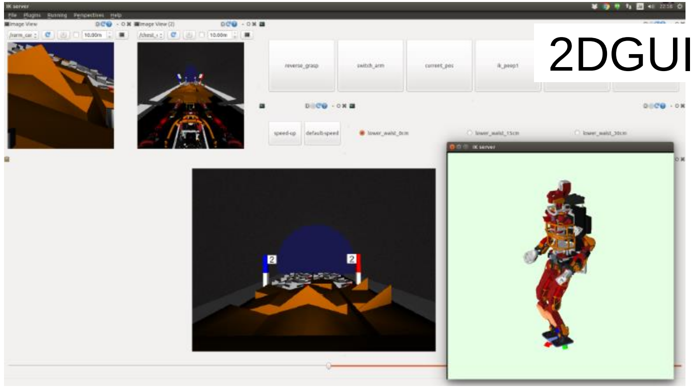


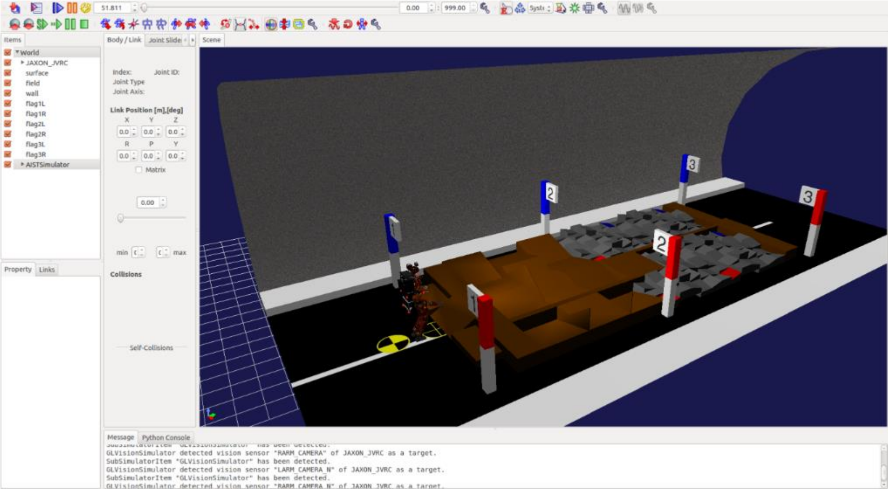


Robot control environment using Choreonoid (JSK lab.)

- Configured to use the same interface as the real robot
- Almost same GUI using DRC

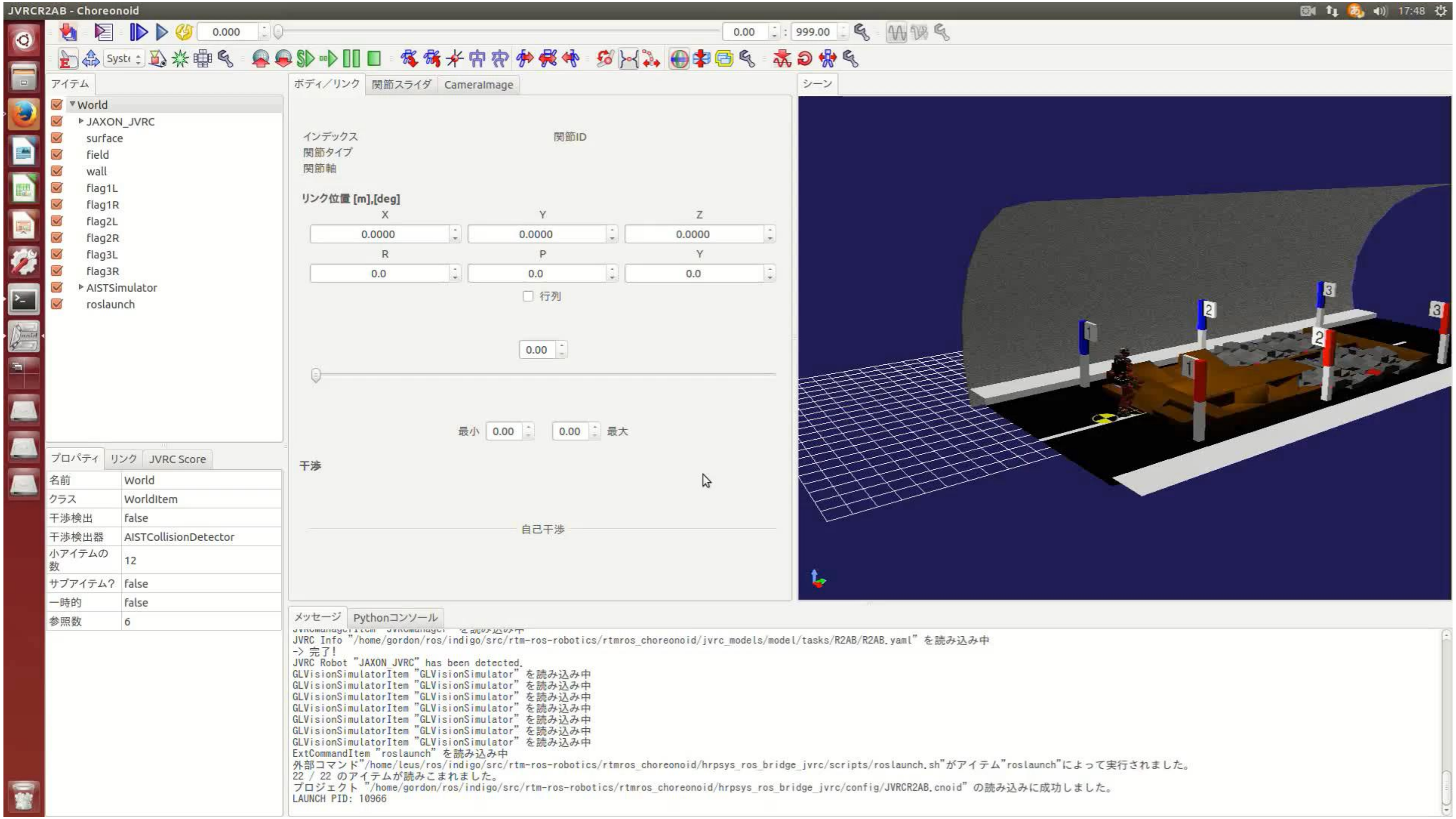

3DGUI


2DGUI



```
GLVisionSimulator detected vision sensor "RARM_CAMERA" of JAXON_JVRC as a target.
SubSimulatorItem "GLVisionSimulator" has been detected.
GLVisionSimulator detected vision sensor "LARM_CAMERA_N" of JAXON_JVRC as a target.
SubSimulatorItem "GLVisionSimulator" has been detected.
GLVisionSimulator detected vision sensor "RARM_CAMERA_N" of JAXON_JVRC as a target.
```

Robot control environment using Choreonoid



アイテム

- ☑ World
  - ☑ JAXON_JVRC
  - ☑ surface
  - ☑ field
  - ☑ wall
  - ☑ flag1L
  - ☑ flag1R
  - ☑ flag2L
  - ☑ flag2R
  - ☑ flag3L
  - ☑ flag3R
  - ☑ AISTSimulator
  - ☑ roslaunch

インデックス
関節ID
関節タイプ
関節軸

リンク位置 [m],[deg]

X	Y	Z
0.0000	0.0000	0.0000
R	**P**	**Y**
0.0	0.0	0.0

□ 行列

0.00

最小 0.00 0.00 最大

干渉

自己干渉

プロパティ　リンク　JVRC Score

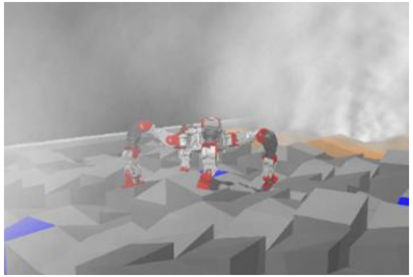
名前	World
クラス	WorldItem
干渉検出	false
干渉検出器	AISTCollisionDetector
小アイテムの数	12
サブアイテム?	false
一時的	false
参照数	6

```
JVRCManagerItem "JVRCManager" を読み込み中
JVRC Info "/home/gordon/ros/indigo/src/rtm-ros-robotics/rtmros_choreonoid/jvrc_models/model/tasks/R2AB/R2AB.yaml" を読み込み中
-> 完了!
JVRC Robot "JAXON_JVRC" has been detected.
GLVisionSimulatorItem "GLVisionSimulator" を読み込み中
GLVisionSimulatorItem "GLVisionSimulator" を読み込み中
GLVisionSimulatorItem "GLVisionSimulator" を読み込み中
GLVisionSimulatorItem "GLVisionSimulator" を読み込み中
GLVisionSimulatorItem "GLVisionSimulator" を読み込み中
GLVisionSimulatorItem "GLVisionSimulator" を読み込み中
GLVisionSimulatorItem "GLVisionSimulator" を読み込み中
ExtCommandItem "roslaunch" を読み込み中
外部コマンド"/home/leus/ros/indigo/src/rtm-ros-robotics/rtmros_choreonoid/hrpsys_ros_bridge_jvrc/scripts/roslaunch.sh"がアイテム"roslaunch"によって実行されました。
22 / 22 のアイテムが読みこまれました。
プロジェクト "/home/gordon/ros/indigo/src/rtm-ros-robotics/rtmros_choreonoid/hrpsys_ros_bridge_jvrc/config/JVRCR2AB.cnoid" の読み込みに成功しました。
LAUNCH PID: 10966
```

WRS(World Robot Summit)

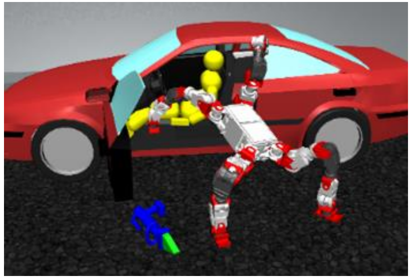
World Robot Summit

Task T1
Traversing Obstacles

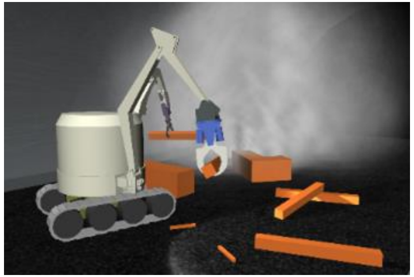


Task T2
Vehicle Inspection

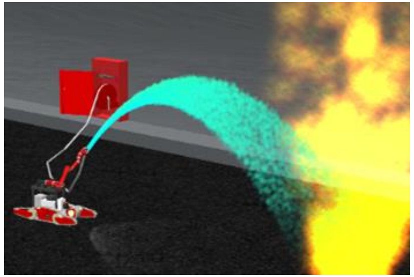
Task T3
Rescue using Tools



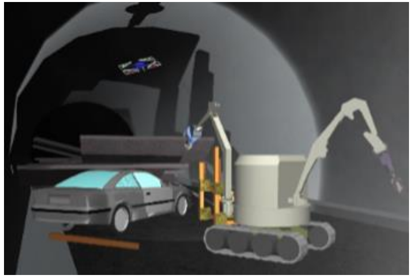
Task T4
Secure the Route



Task T5
Fire Extinguish

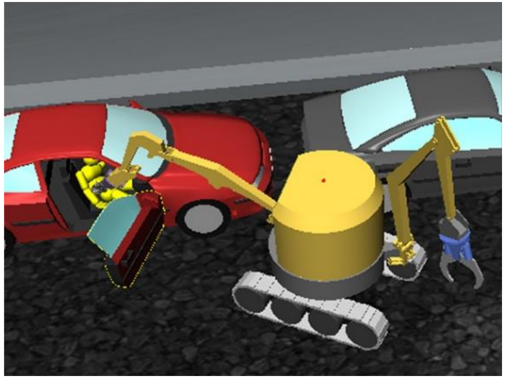


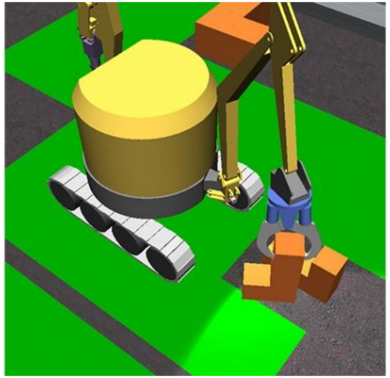
Task T6
Shoring and Breaching

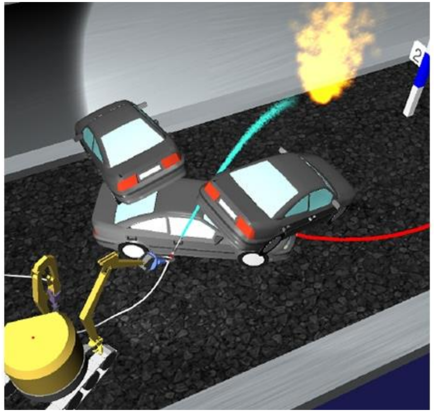


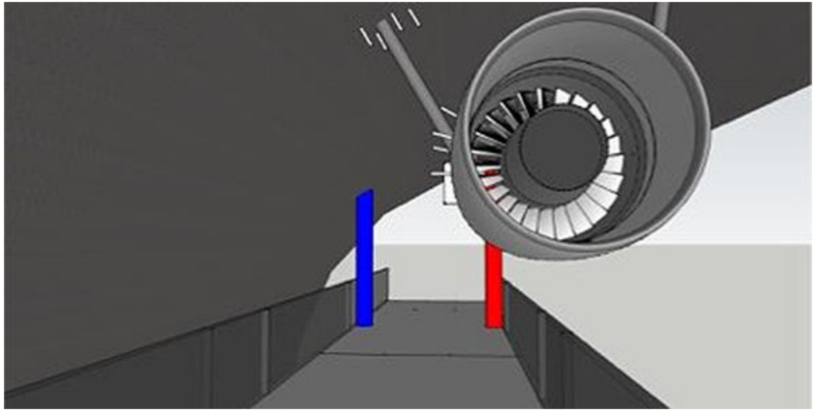
WRS2020 (Tunnel disaster challenge)

- Tunnel disaster challenge
  - https://worldrobotsummit.org/wrs2020/challenge/disaster/tunnel.html



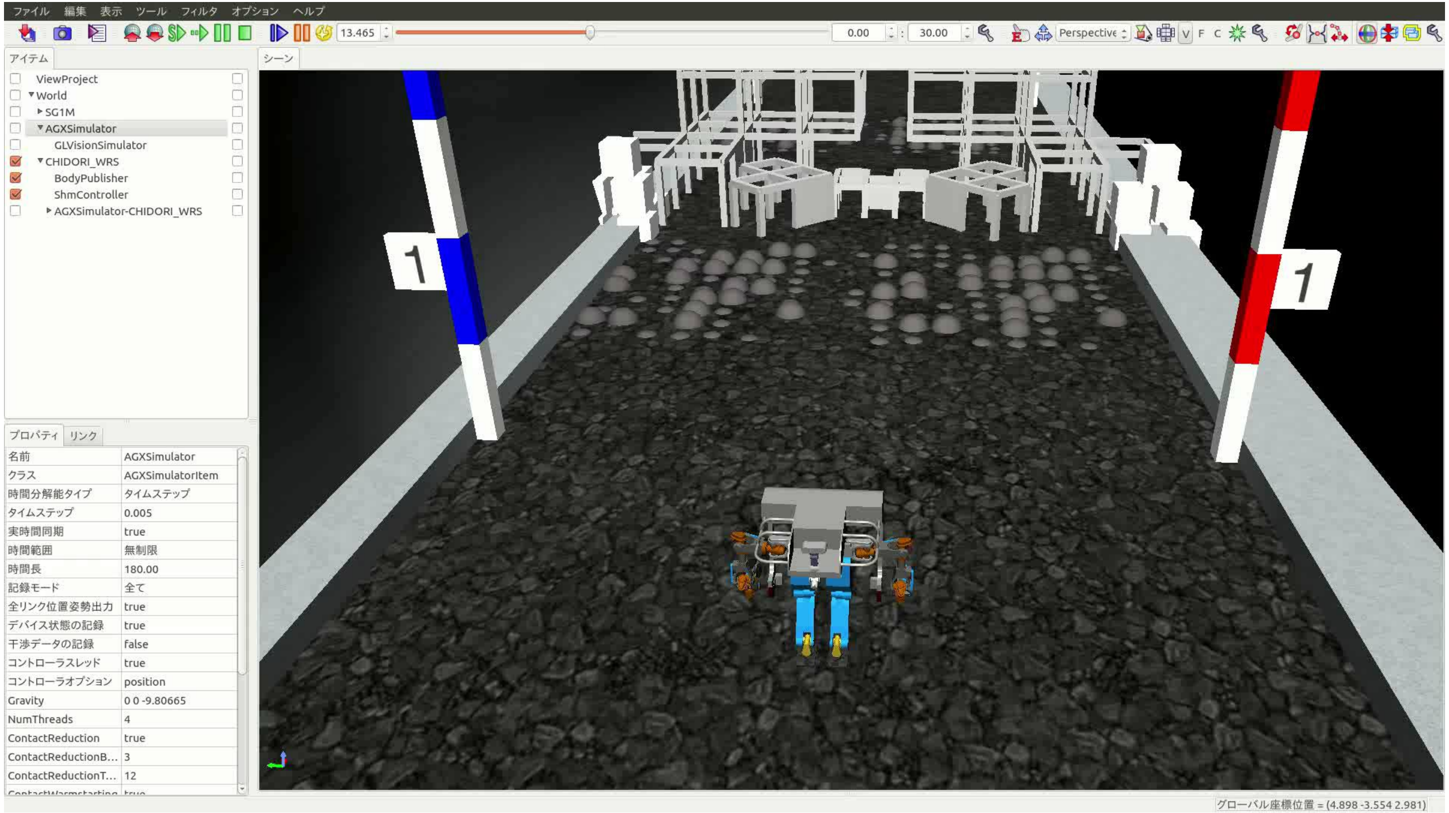






WRS2020 (Tunnel disaster challenge)

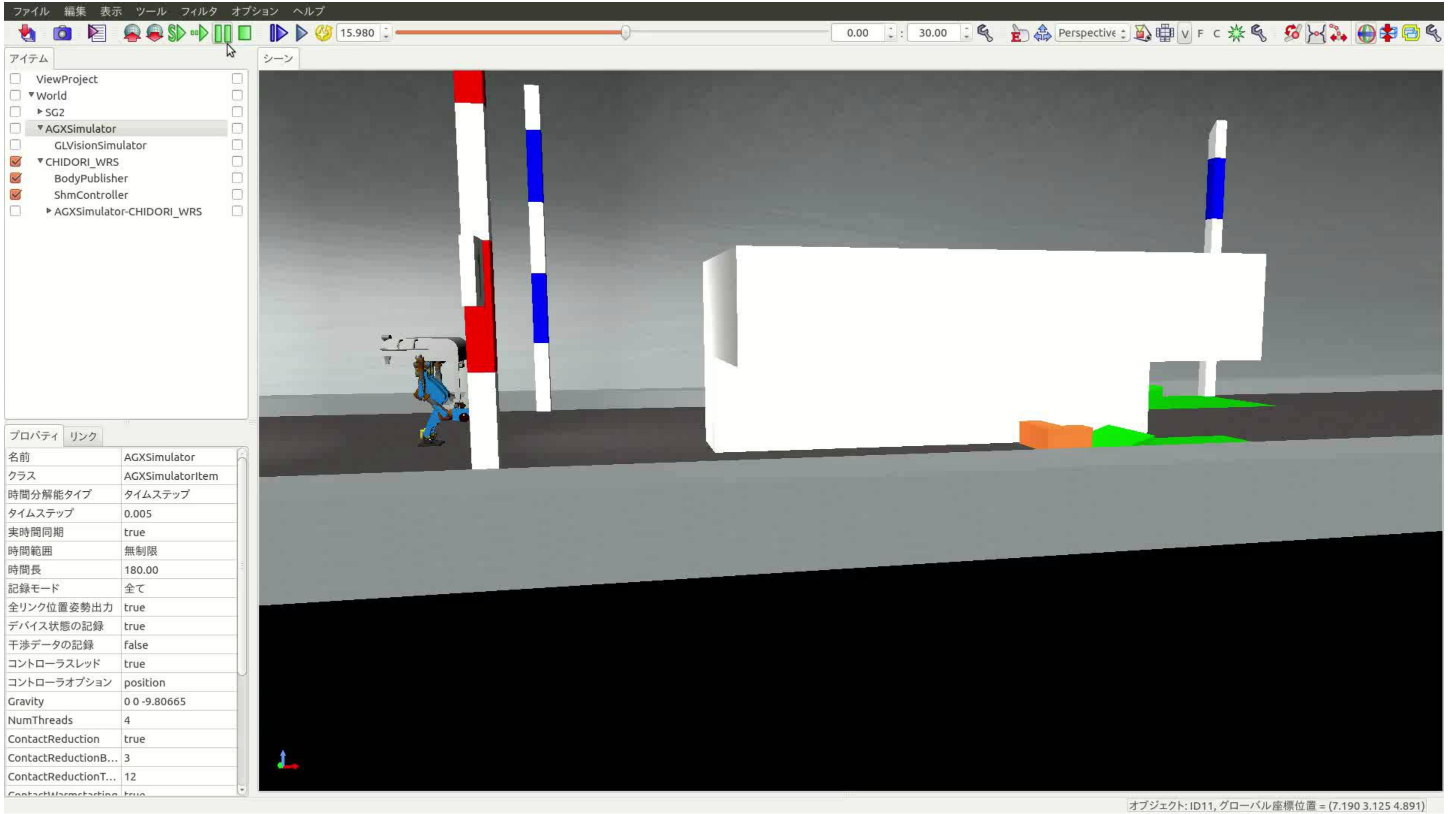
- Stage Gate (Locomotion through rough terrain)



名前	AGXSimulator
クラス	AGXSimulatorItem
時間分解能タイプ	タイムステップ
タイムステップ	0.005
実時間同期	true
時間範囲	無制限
時間長	180.00
記録モード	全て
全リンク位置姿勢出力	true
デバイス状態の記録	true
干渉データの記録	false
コントローラスレッド	true
コントローラオプション	position
Gravity	0 0 -9.80665
NumThreads	4
ContactReduction	true
ContactReductionB...	3
ContactReductionT...	12
ContactWarmstarting	true

WRS2020 (Tunnel disaster challenge)

- Stage Gate (Manipulation – Heavy Object / Tools)

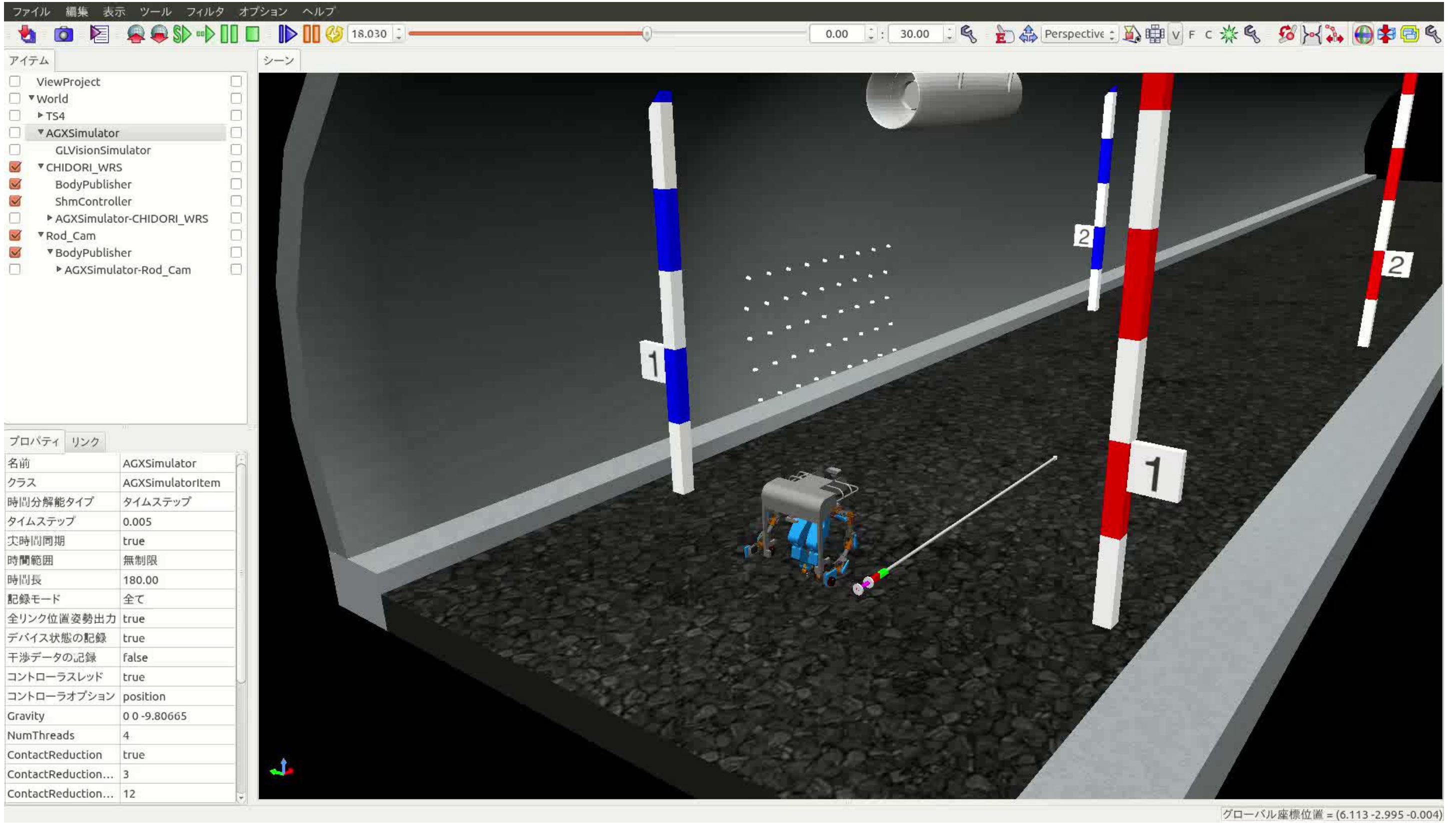


名前	AGXSimulator
クラス	AGXSimulatorItem
時間分解能タイプ	タイムステップ
タイムステップ	0.005
実時間同期	true
時間範囲	無制限
時間長	180.00
記録モード	全て
全リンク位置姿勢出力	true
デバイス状態の記録	true
干渉データの記録	false
コントローラスレッド	true
コントローラオプション	position
Gravity	0 0 -9.80665
NumThreads	4
ContactReduction	true
ContactReductionB...	3
ContactReductionT...	12
ContactWarmstarting	true

オブジェクト: ID11, グローバル座標位置 = (7.190 3.125 4.891)

WRS2020

- Stage Gate (Investigation using camera)



- ☐ ViewProject
- ☐ ▼World
  - ☐ ▶TS4
  - ☐ ▼AGXSimulator
    - ☐ GLVisionSimulator
  - ☑ ▼CHIDORI_WRS
    - ☑ BodyPublisher
    - ☑ ShmController
    - ☐ ▶AGXSimulator-CHIDORI_WRS
  - ☑ ▼Rod_Cam
    - ☑ ▼BodyPublisher
      - ☐ ▶AGXSimulator-Rod_Cam

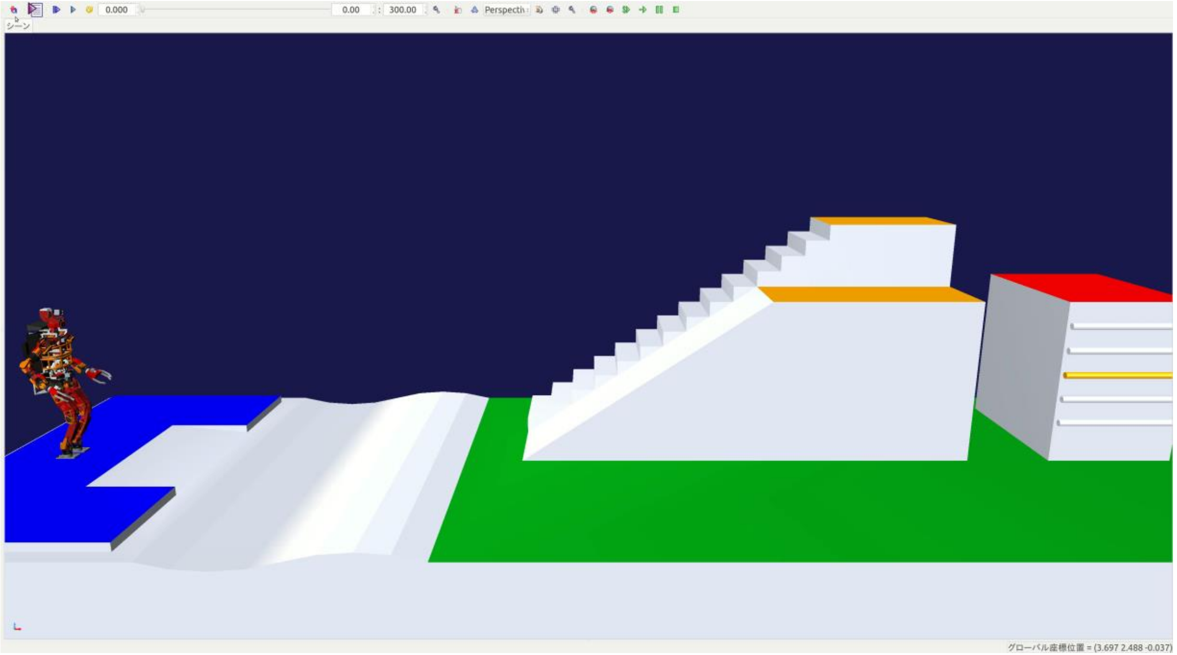
名前	AGXSimulator
クラス	AGXSimulatorItem
時間分解能タイプ	タイムステップ
タイムステップ	0.005
実時間同期	true
時間範囲	無制限
時間長	180.00
記録モード	全て
全リンク位置姿勢出力	true
デバイス状態の記録	true
干渉データの記録	false
コントローラスレッド	true
コントローラオプション	position
Gravity	0 0 -9.80665
NumThreads	4
ContactReduction	true
ContactReduction...	3
ContactReduction...	12

グローバル座標位置 = (6.113 -2.995 -0.004)

HVAC (Humanoid Virtual Athletics Challenge)

- Whole body control for difficult environment
- Operator set the target

Team Jaxon(2021)



グローバル座標位置 = (3.697 2.488 -0.037)

x5

HVAC (Humanoid Virtual Athletics Challenge) (https://ytazz.github.io/vnoid/)

Roundup of robot competitions using simulation

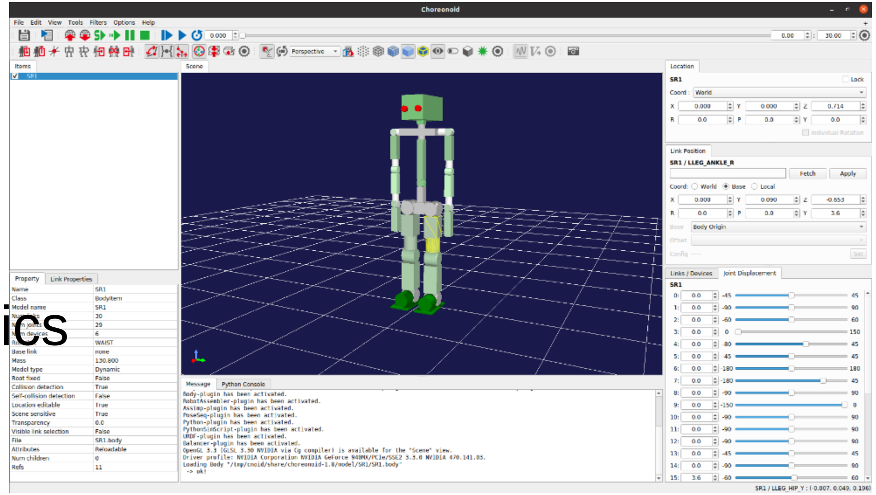
- Simulator should be easy to use
  - Most important performance is speed (second is accuracy)
  - Well documented and many samples
- How to increase participants
  - Good samples (correspondence with a thesis)
  - Allow for variety of purpose (few restrictions)
  - Increased complexity of task is a trade-off for the entry barriers.
- My personal hope is that a real robot is a familiar target

Outline

- Robot competitions and Simulation
- General information of Choreonoid
- Connecting to other system
- Development system on Choreonoid
- Learning Robot Programming

General Information (Choreonoid)

- Choreonoid
  - Open source
    - https://choreonoid.org/
    - https://github.com/choreonoid
  - Integrated GUI platform for Robotics
    - Choreography (for dancing)
    - Simulation
    - Visualization of sensors
    - Remote control
  - Plugin system



Location: SR1 □ Lock
Coord : World
X 0.000 Y 0.000 Z 0.714
R 0.0 P 0.0 Y 0.0
□ Individual Rotation

Link Position: SR1 / LLEG_ANKLE_R
Coord: ○ World ● Base ○ Local
X 0.000 Y 0.090 Z -0.653
R 0.0 P 0.0 Y 3.6
Base Body Origin
Offset
Config ----- Set

Property	Value
Name	SR1
Class	BodyItem
Model name	SR1
Num links	30
Num joints	29
Num devices	6
Root link	WAIST
Base link	none
Mass	130.800
Model type	Dynamic
Root fixed	False
Collision detection	True
Self-collision detection	False
Location editable	True
Scene sensitive	True
Transparency	0.0
Visible link selection	False
File	SR1.body
Attributes	Reloadable
Num children	0
Refs	11

```
Body-plugin has been activated.
RobotAssembler-plugin has been activated.
Assimp-plugin has been activated.
PoseSeq-plugin has been activated.
Python-plugin has been activated.
PythonSimScript-plugin has been activated.
URDF-plugin has been activated.
Balancer-plugin has been activated.
OpenGL 3.3 (GLSL 3.30 NVIDIA via Cg compiler) is available for the "Scene" view.
Driver profile: NVIDIA Corporation NVIDIA GeForce 940MX/PCIe/SSE2 3.3.0 NVIDIA 470.141.03.
Loading Body "/tmp/cnoid/share/choreonoid-1.8/model/SR1/SR1.body"
 -> ok!
```

Joint Displacement: SR1

0:	0.0	-45	45
1:	0.0	-90	90
2:	0.0	-60	60
3:	0.0	0	150
4:	0.0	-80	45
5:	0.0	-45	45
6:	0.0	-180	180
7:	0.0	-180	45
8:	0.0	-90	90
9:	0.0	-150	0
10:	0.0	-90	90
11:	0.0	-90	90
12:	0.0	-90	90
13:	0.0	-45	45
14:	0.0	-90	90
15:	3.6	-60	60

SR1 / LLEG_HIP_Y : (-0.007, 0.049, 0.196)

- From 2019, Start of business activities for commercial use

- 「統合ロボットシミュレータChoreonoidの最新機能」計測と制御2018年57巻10号p.700-705
- https://www.jstage.jst.go.jp/article/sicejl/57/10/57_700/_pdf/-char/ja

Choreonoid

- Integrated GUI platform
  - Can be used for various purposes
  - GUI can be customized through the use of extensions
  - Using Libraries for robotic programming

1. Robot simulator

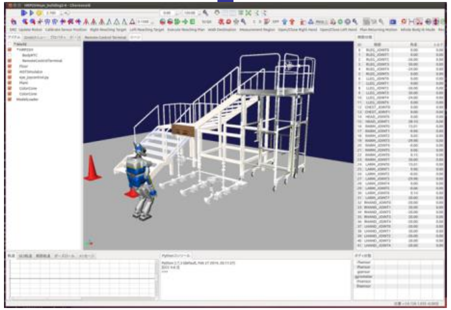
Used in JVRC (Japan virtual robotics challenge)

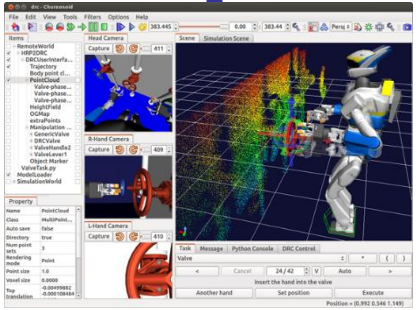
2. Remote Operation Interface

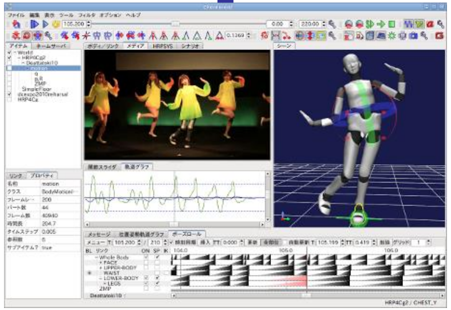
Remote Operation Interface
in DRC Finals

3. Choreography

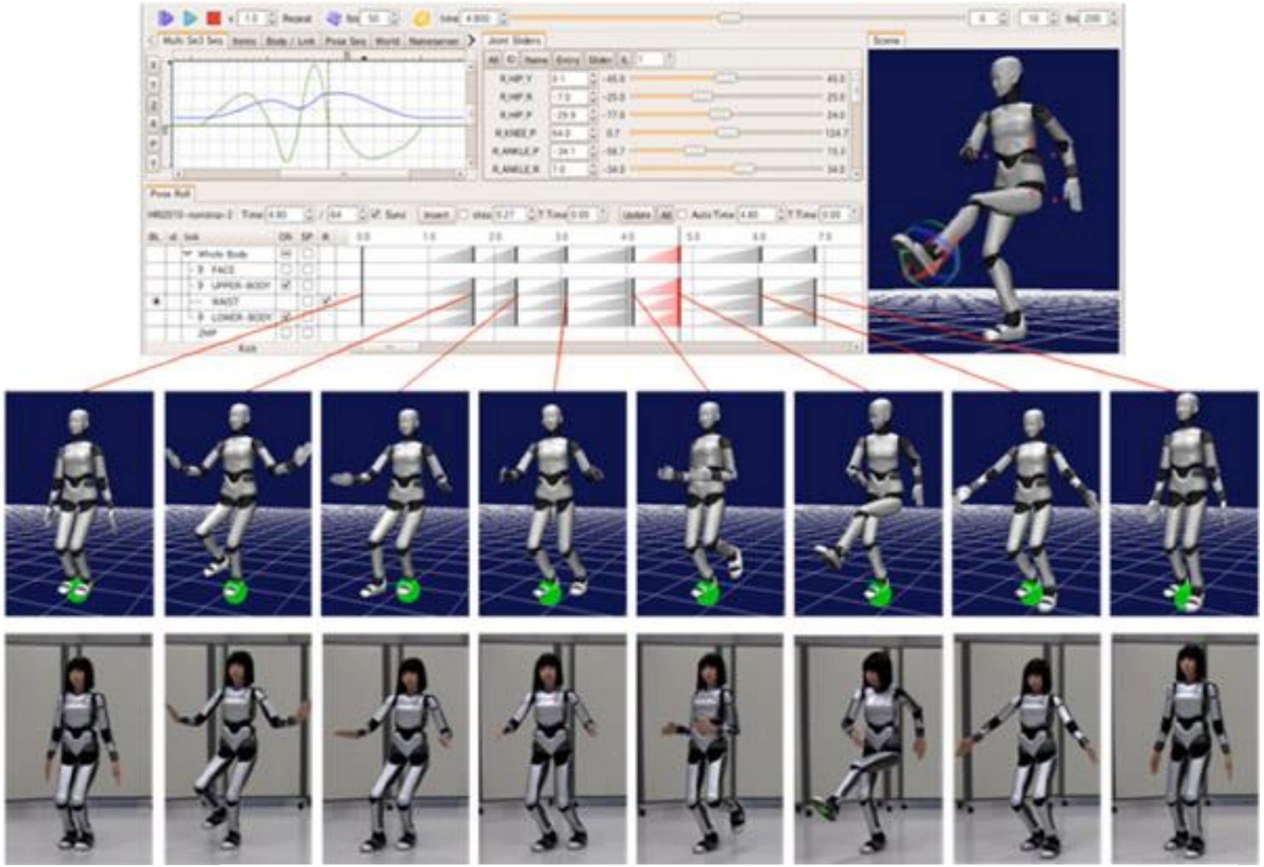
Automatic balance adjustment function allows choreography with the appearance of a CG character.





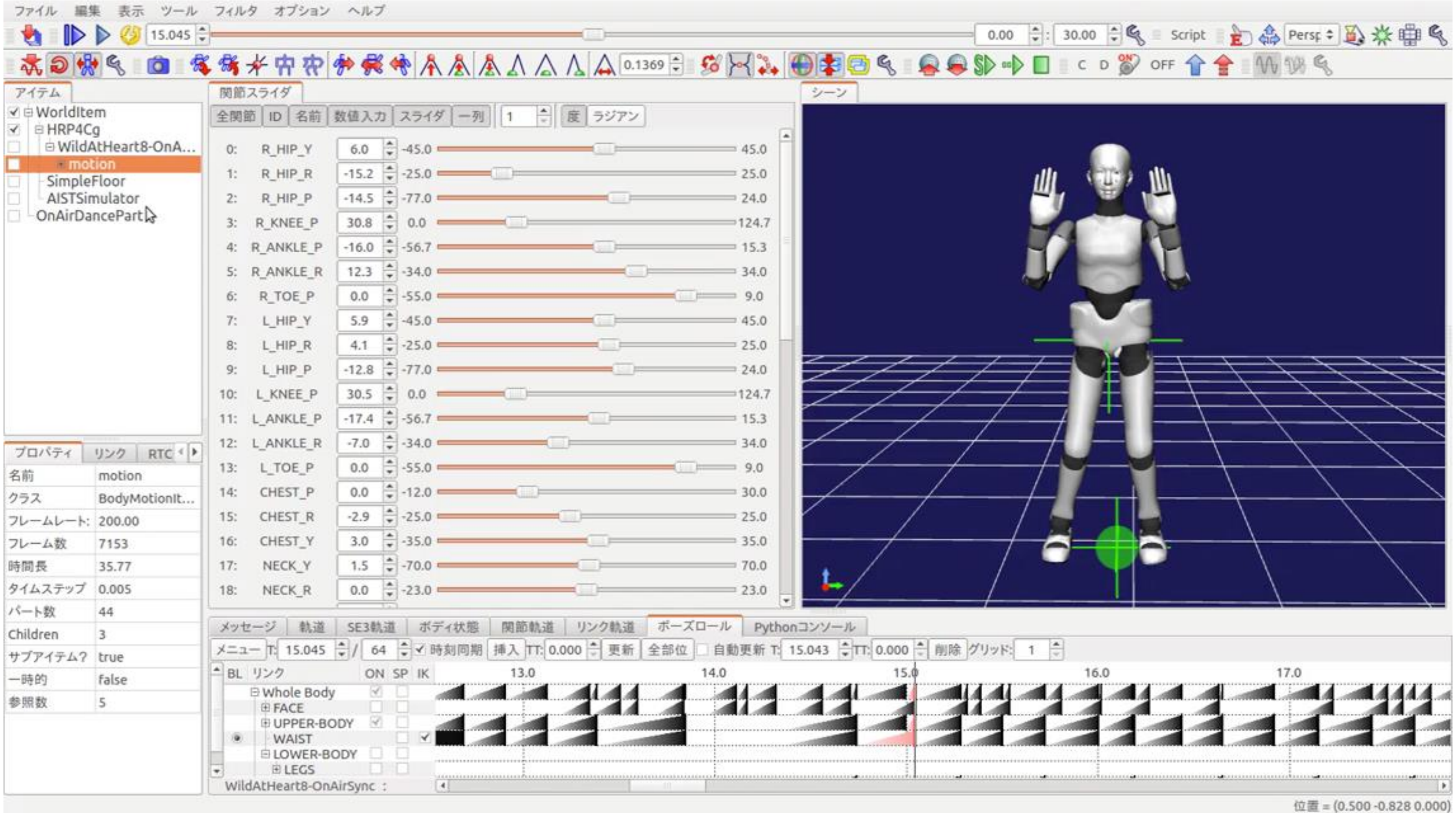


Choreonoid (Choreography)



ChoreonoidとChoreonoidを用いて作成したヒューマノイドロボットHRP-4Cの動作例
https://choreonoid.org/ja/about.html

Choreonoid (Choreography and Whole body dynamics)

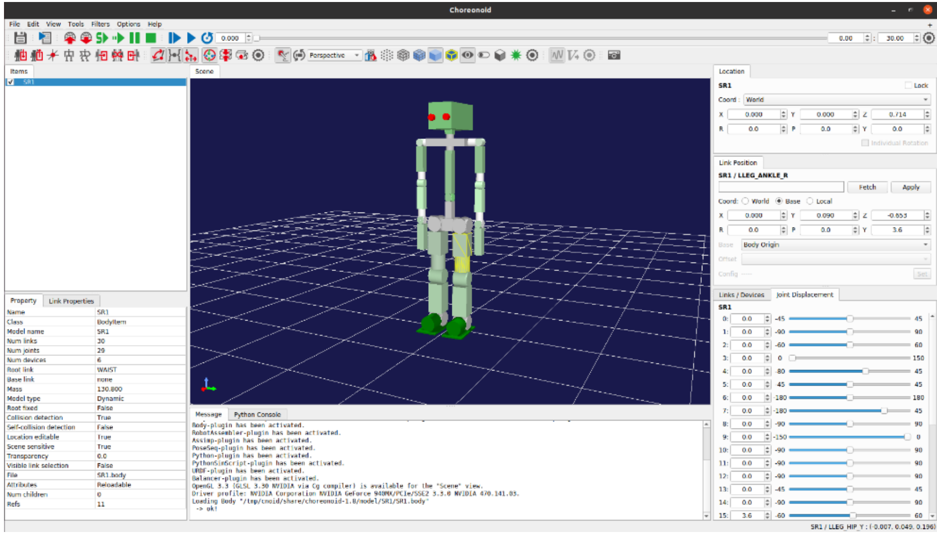


0:	R_HIP_Y	6.0	-45.0	45.0
1:	R_HIP_R	-15.2	-25.0	25.0
2:	R_HIP_P	-14.5	-77.0	24.0
3:	R_KNEE_P	30.8	0.0	124.7
4:	R_ANKLE_P	-16.0	-56.7	15.3
5:	R_ANKLE_R	12.3	-34.0	34.0
6:	R_TOE_P	0.0	-55.0	9.0
7:	L_HIP_Y	5.9	-45.0	45.0
8:	L_HIP_R	4.1	-25.0	25.0
9:	L_HIP_P	-12.8	-77.0	24.0
10:	L_KNEE_P	30.5	0.0	124.7
11:	L_ANKLE_P	-17.4	-56.7	15.3
12:	L_ANKLE_R	-7.0	-34.0	34.0
13:	L_TOE_P	0.0	-55.0	9.0
14:	CHEST_P	0.0	-12.0	30.0
15:	CHEST_R	-2.9	-25.0	25.0
16:	CHEST_Y	3.0	-35.0	35.0
17:	NECK_Y	1.5	-70.0	70.0
18:	NECK_R	0.0	-23.0	23.0

名前	motion
クラス	BodyMotionIt...
フレームレート:	200.00
フレーム数	7153
時間長	35.77
タイムステップ	0.005
パート数	44
Children	3
サブアイテム?	true
一時的	false
参照数	5

General Information (Choreonoid)

- Software structure of Choreonoid (3 main libraries)
  - Body (Robot model) library
    - Robot structure – Body, Link, Sensor
    - Kinematics
    - Dynamics
  - Base system (GUI) library
    - View (3D visualize view, etc.)
    - Panel for Body, Link, Sensor
    - Tool bar
  - Utility library
    - Robot model loader (using yaml)
    - Matrix (Eigen), Algorisms
  - Python bindings



Property	
Name	SR1
Class	BodyItem
Model name	SR1
Num links	30
Num joints	29
Num devices	6
Root link	WAIST
Base link	none
Mass	130.800
Model type	Dynamic
Root fixed	False
Collision detection	True
Self-collision detection	False
Location editable	True
Scene sensitive	True
Transparency	0.0
Visible link selection	False
File	SR1.body
Attributes	Reloadable
Num children	0
Refs	11

```
Body-plugin has been activated.
RobotAssembler-plugin has been activated.
Assimp-plugin has been activated.
PoseSeq-plugin has been activated.
Python-plugin has been activated.
PythonSimScript-plugin has been activated.
URDF-plugin has been activated.
Balancer-plugin has been activated.
OpenGL 3.3 (GLSL 3.30 NVIDIA via Cg compiler) is available for the "Scene" view.
Driver profile: NVIDIA Corporation NVIDIA GeForce 940MX/PCIe/SSE2 3.3.0 NVIDIA 470.141.03.
Loading Body "/tmp/cnoid/share/choreonoid-1.8/model/SR1/SR1.body"
 -> ok!
```

0.00 : 30.00

Location

SR1 □ Lock

Coord : World

X 0.000 Y 0.000 Z 0.714

R 0.0 P 0.0 Y 0.0

□ Individual Rotation

Link Position

SR1 / LLEG_ANKLE_R

Coord: ○ World ● Base ○ Local

X 0.000 Y 0.090 Z -0.653

R 0.0 P 0.0 Y 3.6

Base Body Origin

Offset

Config -----

Joint Displacement

SR1

0:	0.0	-45	45
1:	0.0	-90	90
2:	0.0	-60	60
3:	0.0	0	150
4:	0.0	-80	45
5:	0.0	-45	45
6:	0.0	-180	180
7:	0.0	-180	45
8:	0.0	-90	90
9:	0.0	-150	0
10:	0.0	-90	90
11:	0.0	-90	90
12:	0.0	-90	90
13:	0.0	-45	45
14:	0.0	-90	90
15:	3.6	-60	60

SR1 / LLEG_HIP_Y : (-0.007, 0.049, 0.196)

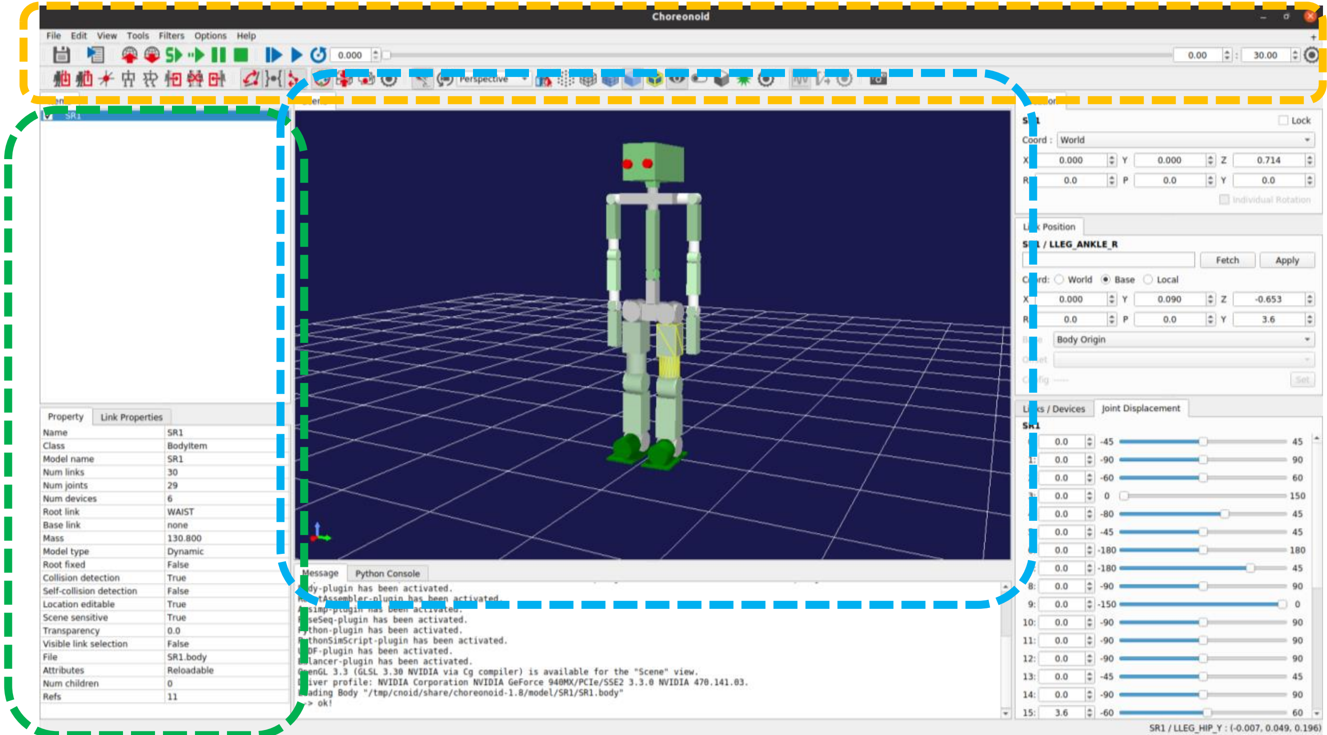
General Information (Choreonoid)

- Software structure of Choreonoid (3 main libraries)
  - Robot model library
  - Base system (GUI) library
  - Utility library
  - Plugin system

Tool Bar

Panel

Model View



Property	Link Properties
Name	SR1
Class	BodyItem
Model name	SR1
Num links	30
Num joints	29
Num devices	6
Root link	WAIST
Base link	none
Mass	130.800
Model type	Dynamic
Root fixed	False
Collision detection	True
Self-collision detection	False
Location editable	True
Scene sensitive	True
Transparency	0.0
Visible link selection	False
File	SR1.body
Attributes	Reloadable
Num children	0
Refs	11

SR1

Coord : World

X 0.000 Y 0.000 Z 0.714

R 0.0 P 0.0 Y 0.0

Link Position

SR1 / LLEG_ANKLE_R

Coord: ○ World ● Base ○ Local

X 0.000 Y 0.090 Z -0.653

R 0.0 P 0.0 Y 3.6

Base Body Origin

Offset

Config -----

Links / Devices | Joint Displacement

SR1

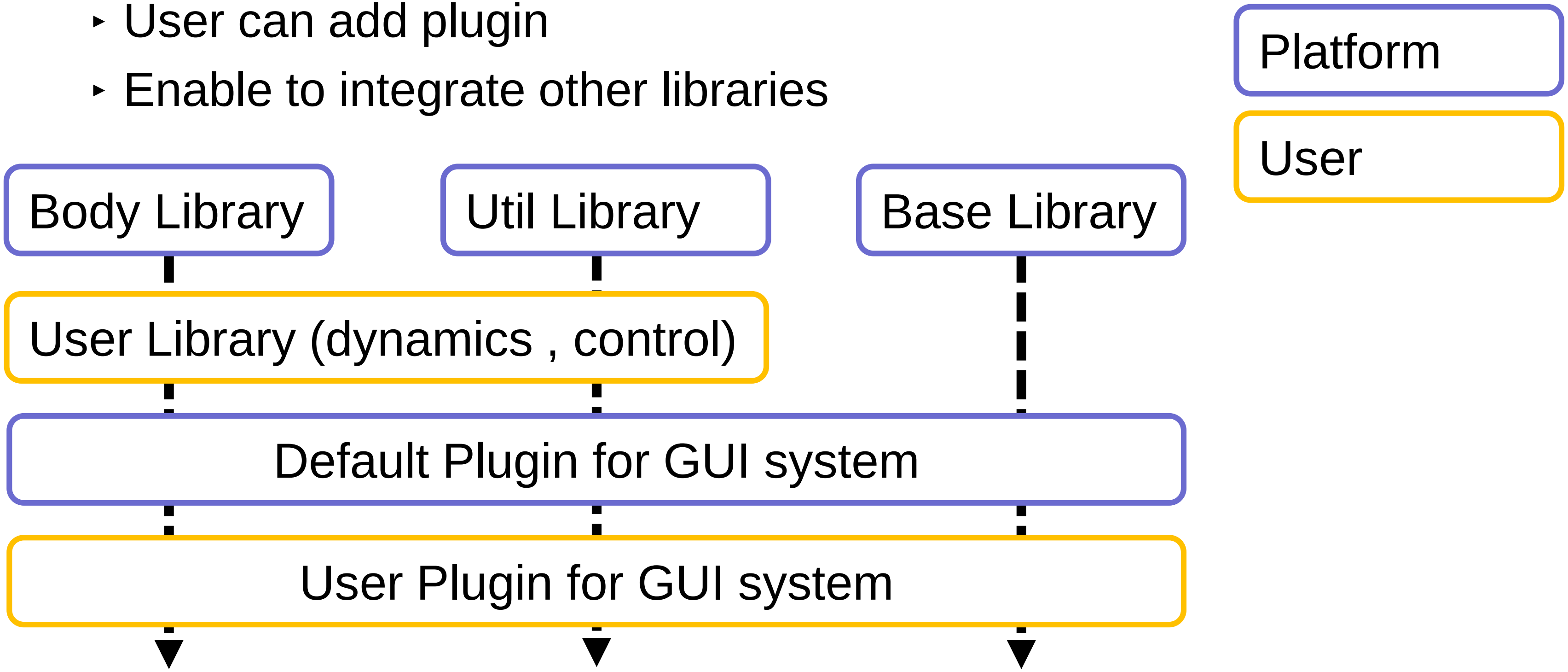
0:	0.0	-45	45
1:	0.0	-90	90
2:	0.0	-60	60
3:	0.0	0	150
4:	0.0	-80	45
5:	0.0	-45	45
6:	0.0	-180	180
7:	0.0	-180	45
8:	0.0	-90	90
9:	0.0	-150	0
10:	0.0	-90	90
11:	0.0	-90	90
12:	0.0	-90	90
13:	0.0	-45	45
14:	0.0	-90	90
15:	3.6	-60	60

```
Body-plugin has been activated.
RobotAssembler-plugin has been activated.
Assimp-plugin has been activated.
PoseSeq-plugin has been activated.
Python-plugin has been activated.
PythonSimScript-plugin has been activated.
URDF-plugin has been activated.
Balancer-plugin has been activated.
OpenGL 3.3 (GLSL 3.30 NVIDIA via Cg compiler) is available for the "Scene" view.
Driver profile: NVIDIA Corporation NVIDIA GeForce 940MX/PCIe/SSE2 3.3.0 NVIDIA 470.141.03.
Loading Body "/tmp/cnoid/share/choreonoid-1.8/model/SR1/SR1.body"
-> ok!
```

SR1 / LLEG_HIP_Y : (-0.007, 0.049, 0.196)

General Information (Choreonoid)

- Software structure of Choreonoid
  - Plugin system
    - User can add plugin
    - Enable to integrate other libraries



Platform
User
Body Library
Util Library
Base Library
User Library (dynamics , control)
Default Plugin for GUI system
User Plugin for GUI system

Outline

- Robot competitions and Simulation
- General information of Choreonoid
- Connecting to other system
- Development system on Choreonoid
- Learning Robot Programming

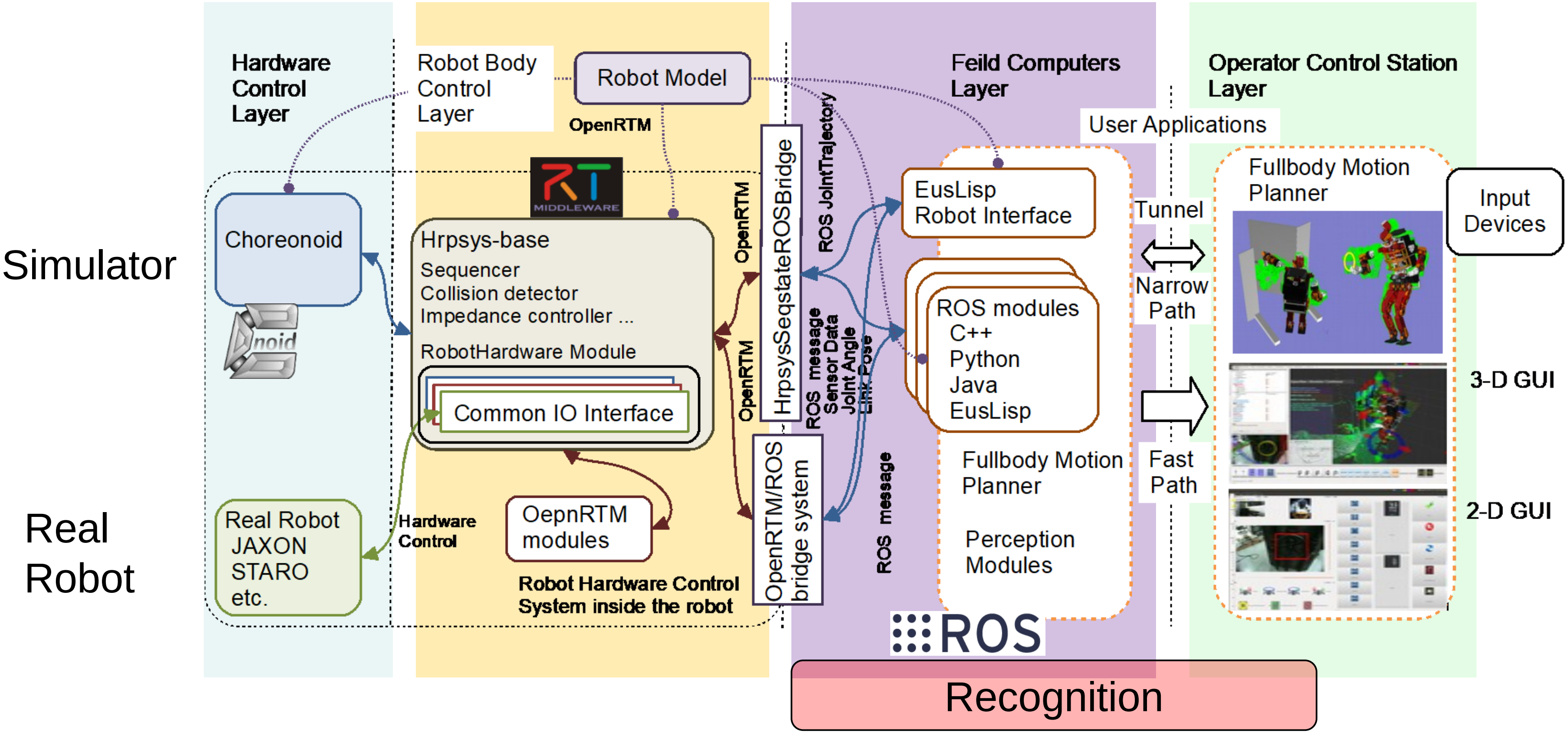
Layered software configuration for simulator and real robot

Hardware control

Real-time Body control

Non real-time Behavior control

User operation (UI)



Humanoid robot control system in simulation

- Humanoid robot control system
  - – Body control
  - – Behavior control
  - – Recognition
  - – Operator (UI)

Left image

Right image (Binarization)

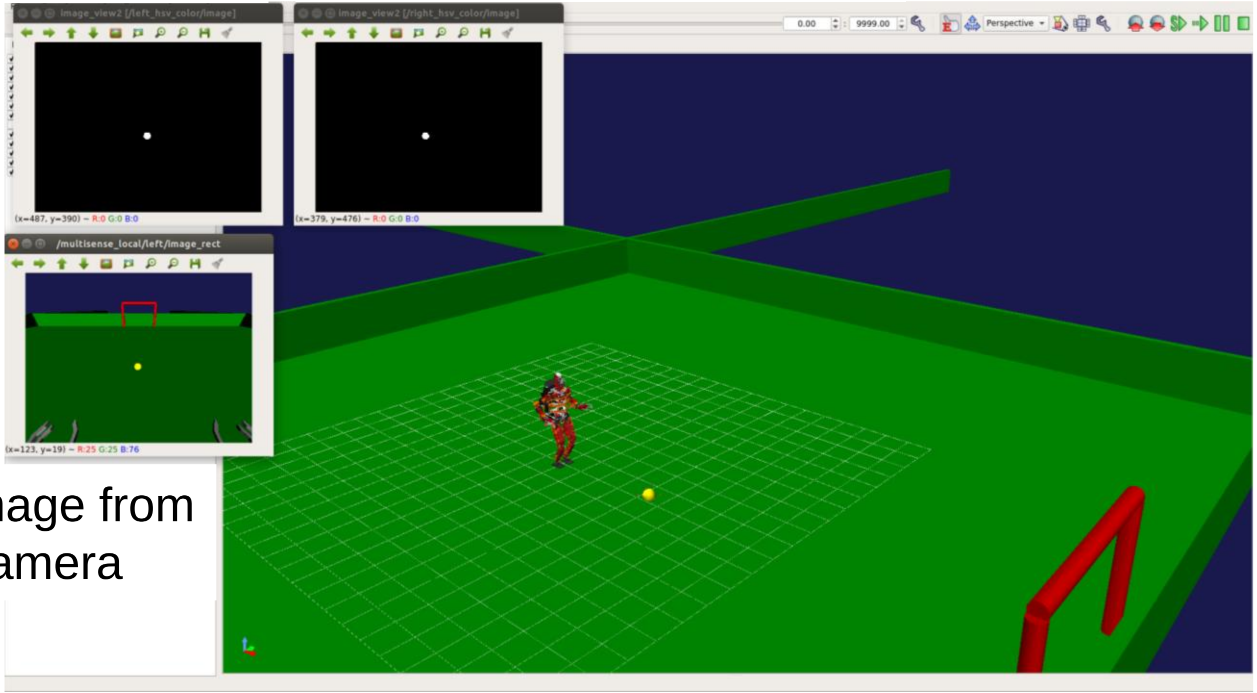
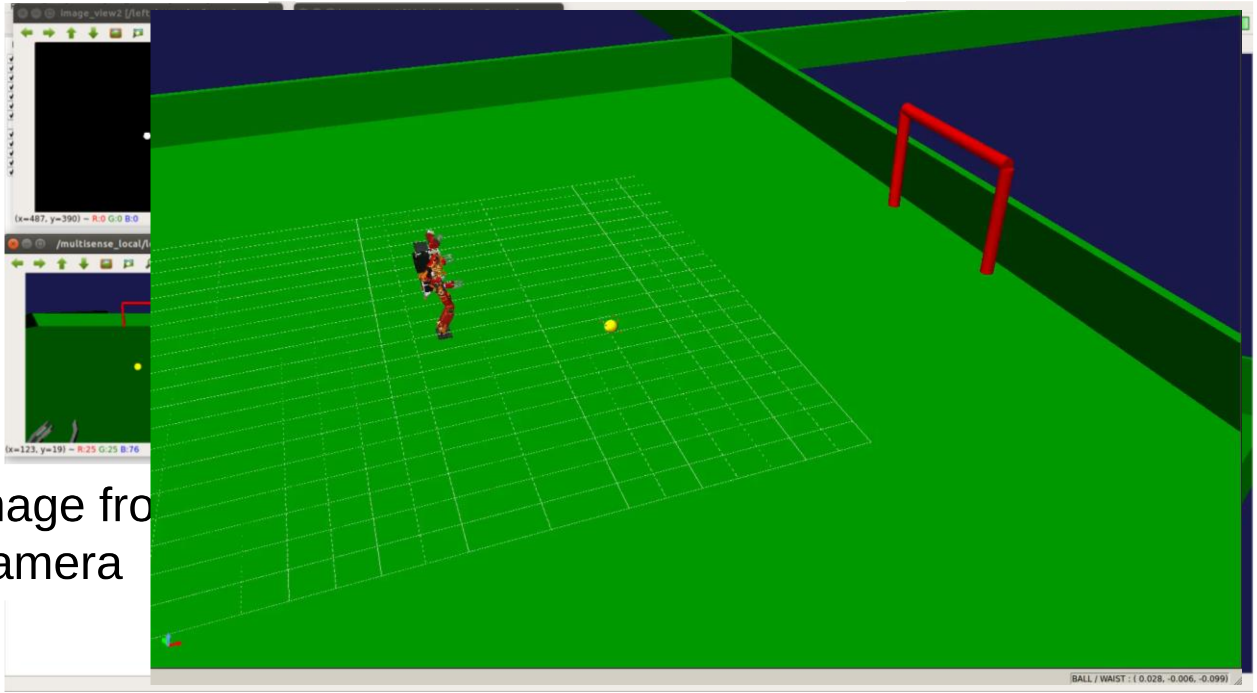

image_view2 [/left_hsv_color/image]
(x=487, y=390) ~ R:0 G:0 B:0
image_view2 [/right_hsv_color/image]
(x=379, y=476) ~ R:0 G:0 B:0
/multisense_local/left/image_rect
(x=123, y=19) ~ R:25 G:25 B:76
0.00
9999.00
Perspective

Image from Camera

Humanoid robot control system in simulation

- Humanoid robot control system
  - Body control
  - Behavior control
  - Recognition
  - Operator (UI)

Left image Right image (Binarization)



(x=487, y=390) ~ R:0 G:0 B:0

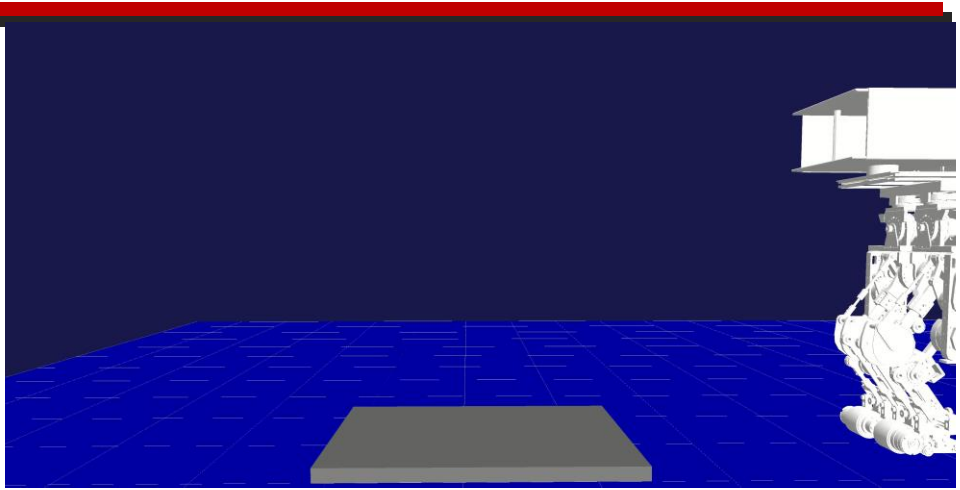
(x=123, y=19) ~ R:25 G:25 B:76

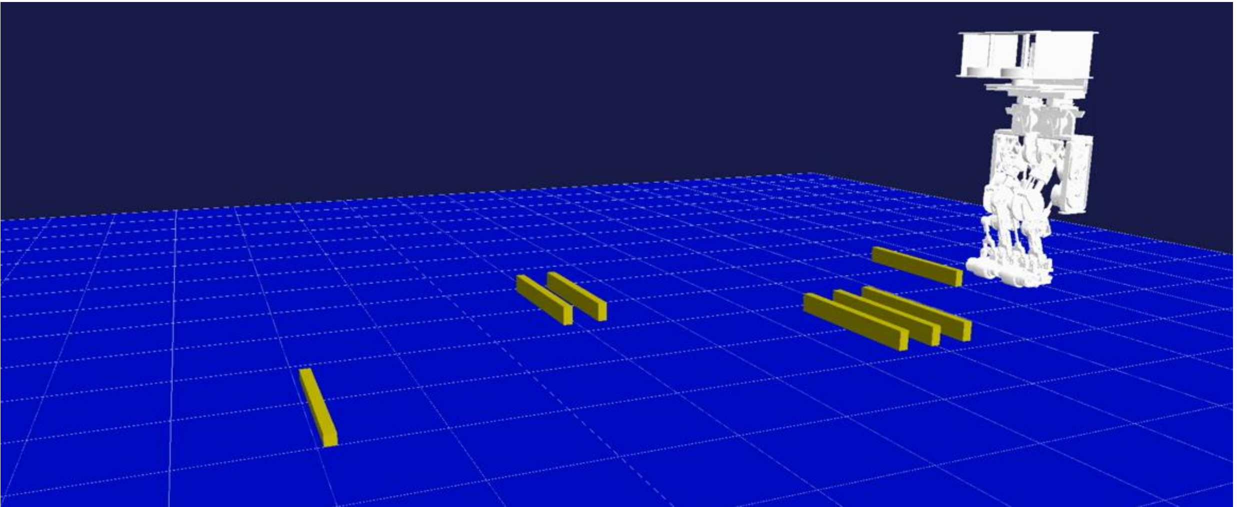
BALL / WAIST : (0.028, -0.006, -0.099)

Image fro
Camera

Hybrid locomotion (In simulator)

Moving over steps





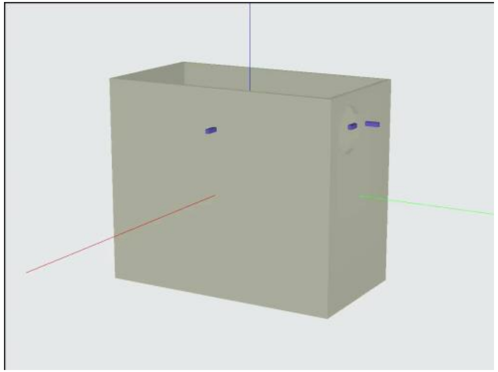
Obstacle Avoidance Leg Wheel Movement

Outline

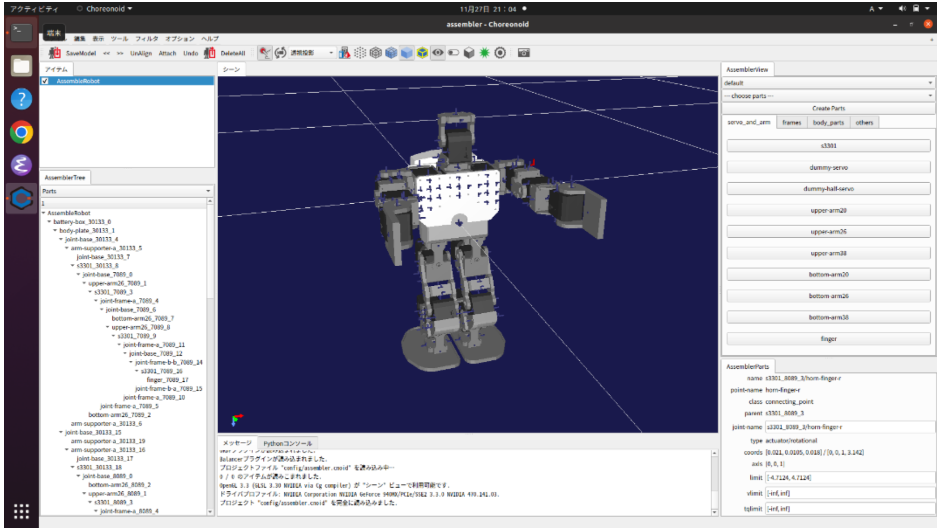
- Robot competitions and Simulation
- General information of Choreonoid
- Connecting to other system
- Development system on Choreonoid
- Learning Robot Programming

Robot assemble system on Choreonoid

- Platform to configure a robot using actuator module
- Support various series of actuator modules
  - By writing definitions file



KXR (kondo kagaku)

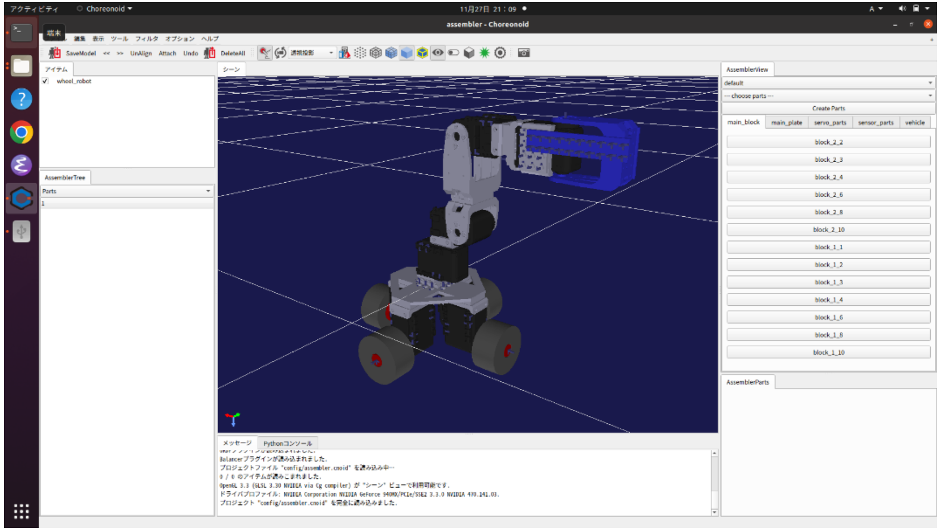


```
1
▾ AssembleRobot
  ▾ battery-box_30133_0
    ▾ body-plate_30133_1
      ▾ joint-base_30133_4
        ▾ arm-supporter-a_30133_5
            joint-base_30133_7
          ▾ s3301_30133_8
            ▾ joint-base_7089_0
              ▾ upper-arm26_7089_1
                ▾ s3301_7089_3
                  ▾ joint-frame-a_7089_4
                    ▾ joint-base_7089_6
                        bottom-arm26_7089_7
                      ▾ upper-arm26_7089_8
                        ▾ s3301_7089_9
                          ▾ joint-frame-a_7089_11
                            ▾ joint-base_7089_12
                              ▾ joint-frame-b-b_7089_14
                                ▾ s3301_7089_16
                                    finger_7089_17
                                joint-frame-b-a_7089_15
                            joint-frame-a_7089_10
                    joint-frame-a_7089_5
                bottom-arm26_7089_2
        arm-supporter-a_30133_6
      ▾ joint-base_30133_15
          arm-supporter-a_30133_19
        ▾ arm-supporter-a_30133_16
            joint-base_30133_17
          ▾ s3301_30133_18
            ▾ joint-base_8089_0
                bottom-arm26_8089_2
              ▾ upper-arm26_8089_1
                ▾ s3301_8089_3
                  ▾ joint-frame-a_8089_4
```

```
Balancerプラグインが読み込まれました.
プロジェクトファイル "config/assembler.cnoid" を読み込み中…
0 / 0 のアイテムが読みこまれました.
OpenGL 3.3 (GLSL 3.30 NVIDIA via Cg compiler) が "シーン" ビューで利用可能です.
ドライバプロファイル: NVIDIA Corporation NVIDIA GeForce 940MX/PCIe/SSE2 3.3.0 NVIDIA 470.141.03.
プロジェクト "config/assembler.cnoid" を完全に読み込みました.
```

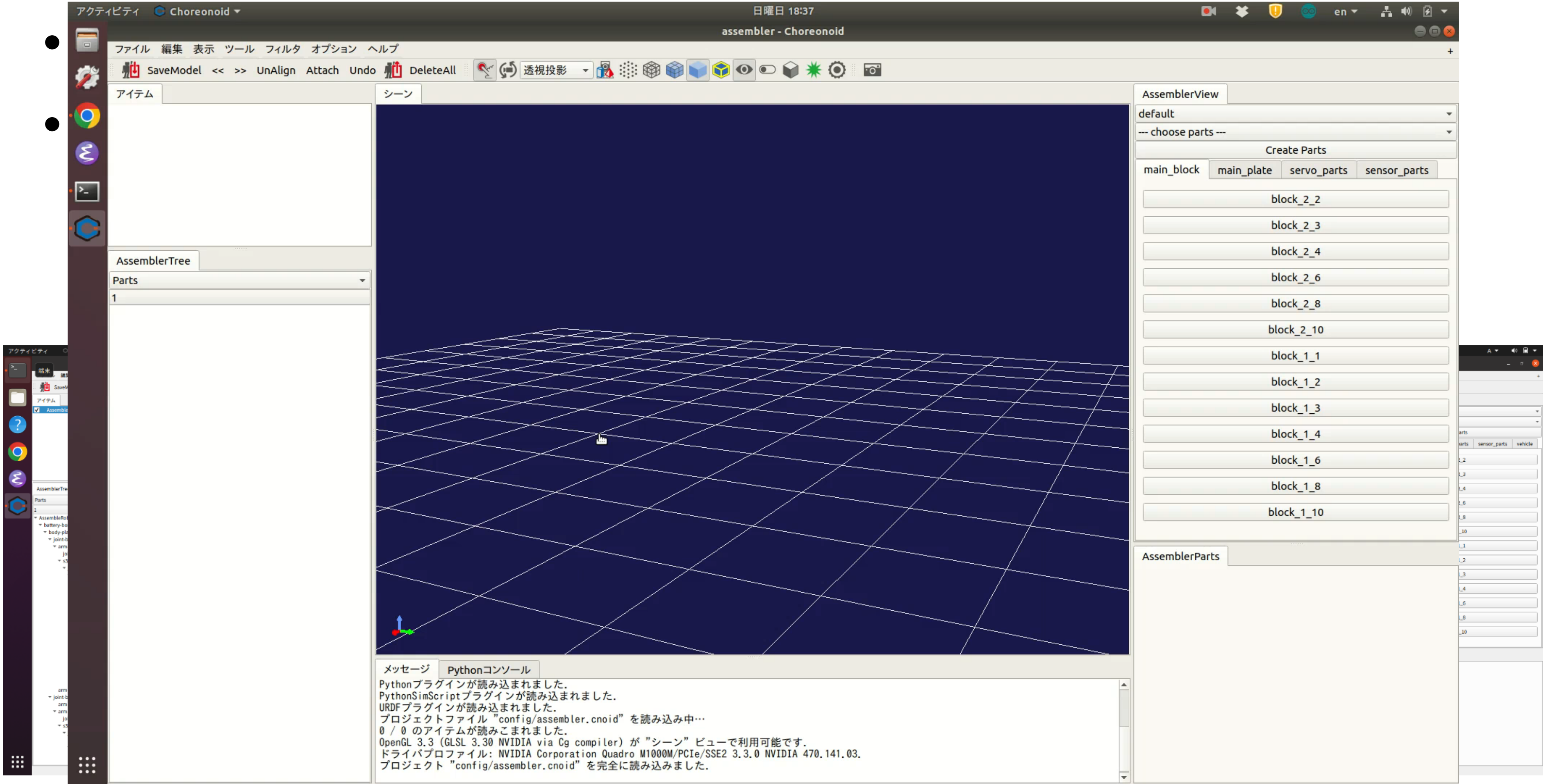
name s3301_8089_3/horn-finger-r
point-name horn-finger-r
class connecting_point
parent s3301_8089_3
joint-name s3301_8089_3/horn-finger-r
type actuator/rotational
coords [0.021, 0.0105, 0.018] / [0, 0, 1, 3.142]
axis [0, 0, 1]
limit [-4.7124, 4.7124]
vlimit [-inf, inf]
tqlimit [-inf, inf]

Dynamixel and Lego block



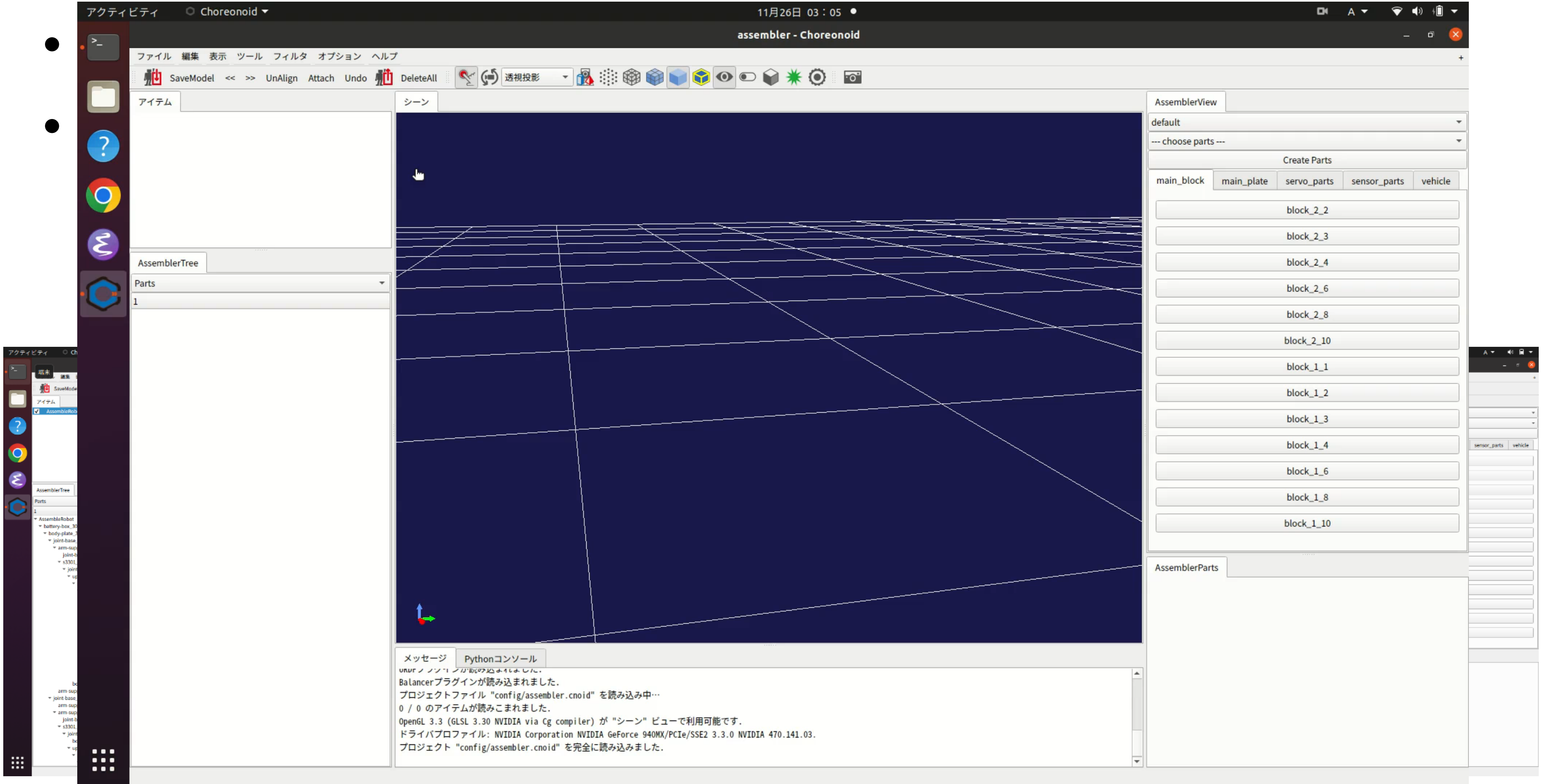
```
Balancerプラグインが読み込まれました.
プロジェクトファイル "config/assembler.cnoid" を読み込み中…
0 / 0 のアイテムが読みこまれました.
OpenGL 3.3 (GLSL 3.30 NVIDIA via Cg compiler) が "シーン" ビューで利用可能です.
ドライバプロファイル: NVIDIA Corporation NVIDIA GeForce 940MX/PCIe/SSE2 3.3.0 NVIDIA 470.141.03.
プロジェクト "config/assembler.cnoid" を完全に読み込みました.
```

Robot assemble system on Choreonoid



```
Pythonプラグインが読み込まれました.
PythonSimScriptプラグインが読み込まれました.
URDFプラグインが読み込まれました.
プロジェクトファイル "config/assembler.cnoid" を読み込み中…
0 / 0 のアイテムが読みこまれました.
OpenGL 3.3 (GLSL 3.30 NVIDIA via Cg compiler) が "シーン" ビューで利用可能です.
ドライバプロファイル: NVIDIA Corporation Quadro M1000M/PCIe/SSE2 3.3.0 NVIDIA 470.141.03.
プロジェクト "config/assembler.cnoid" を完全に読み込みました.
```

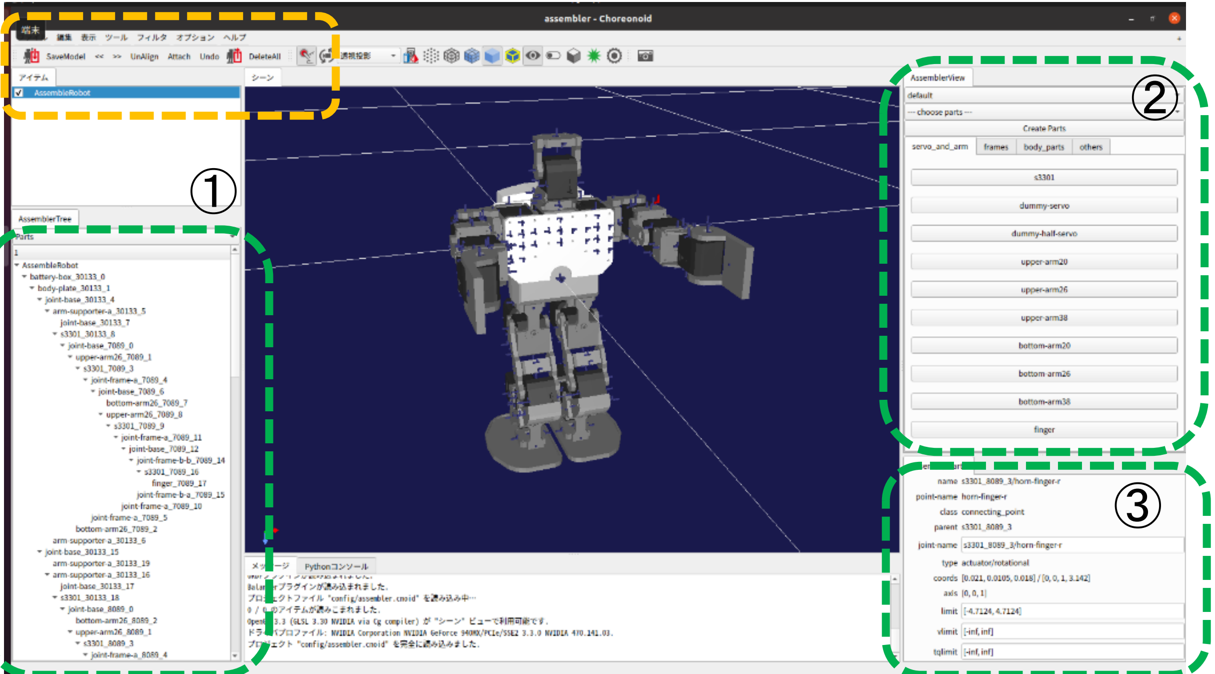
Robot assemble system on Choreonoid



```
URDFプラグインが読み込まれました.
Balancerプラグインが読み込まれました.
プロジェクトファイル "config/assembler.cnoid" を読み込み中…
0 / 0 のアイテムが読みこまれました.
OpenGL 3.3 (GLSL 3.30 NVIDIA via Cg compiler) が "シーン" ビューで利用可能です.
ドライバプロファイル: NVIDIA Corporation NVIDIA GeForce 940MX/PCIe/SSE2 3.3.0 NVIDIA 470.141.03.
プロジェクト "config/assembler.cnoid" を完全に読み込みました.
```

Robot assemble system on Choreonoid

- Platform to configure a robot using actuator module
- Written as a Choreonoid plugin
  - – https://github.com/IRSL-tut/robot_assembler_plugin



```
1
▾ AssembleRobot
  ▾ battery-box_30133_0
    ▾ body-plate_30133_1
      ▾ joint-base_30133_4
        ▾ arm-supporter-a_30133_5
            joint-base_30133_7
          ▾ s3301_30133_8
            ▾ joint-base_7089_0
              ▾ upper-arm26_7089_1
                ▾ s3301_7089_3
                  ▾ joint-frame-a_7089_4
                    ▾ joint-base_7089_6
                        bottom-arm26_7089_7
                      ▾ upper-arm26_7089_8
                        ▾ s3301_7089_9
                          ▾ joint-frame-a_7089_11
                            ▾ joint-base_7089_12
                              ▾ joint-frame-b-b_7089_14
                                ▾ s3301_7089_16
                                    finger_7089_17
                                joint-frame-b-a_7089_15
                            joint-frame-a_7089_10
                    joint-frame-a_7089_5
                bottom-arm26_7089_2
          arm-supporter-a_30133_6
      ▾ joint-base_30133_15
          arm-supporter-a_30133_19
        ▾ arm-supporter-a_30133_16
            joint-base_30133_17
          ▾ s3301_30133_18
            ▾ joint-base_8089_0
                bottom-arm26_8089_2
              ▾ upper-arm26_8089_1
                ▾ s3301_8089_3
                  ▾ joint-frame-a_8089_4
```

name s3301_8089_3/horn-finger-r
point-name horn-finger-r
class connecting_point
parent s3301_8089_3
joint-name s3301_8089_3/horn-finger-r
type actuator/rotational
coords [0.021, 0.0105, 0.018] / [0, 0, 1, 3.142]
axis [0, 0, 1]
limit [-4.7124, 4.7124]
vlimit [-inf, inf]
tqlimit [-inf, inf]

```
Balancerプラグインが読み込まれました。
プロジェクトファイル "config/assembler.cnoid" を読み込み中…
0 / 0 のアイテムが読みこまれました。
OpenGL 3.3 (GLSL 3.30 NVIDIA via Cg compiler) が "シーン" ビューで利用可能です。
ドライバプロファイル: NVIDIA Corporation NVIDIA GeForce 940MX/PCIe/SSE2 3.3.0 NVIDIA 470.141.03.
プロジェクト "config/assembler.cnoid" を完全に読み込みました。
```

Customized Tool Bar

①

Panel : parts tree

②

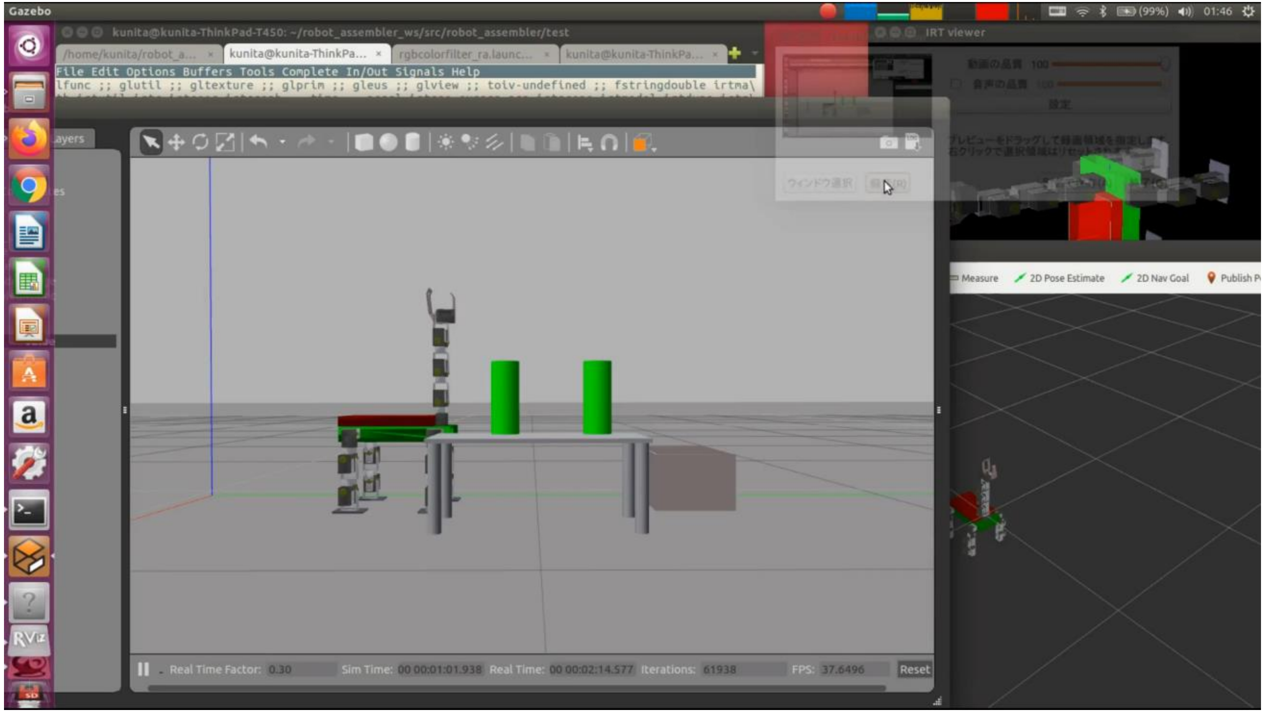
Panel : parts selection list for building robot

③

Panel : parts information

Robot assemble system on Choreonoid

- Platform to configure a robot using actuator module
- Verify configured robot in simulation and in real world



kunita@kunita-ThinkPad-T450: ~/robot_assembler_ws/src/robot_assembler/test

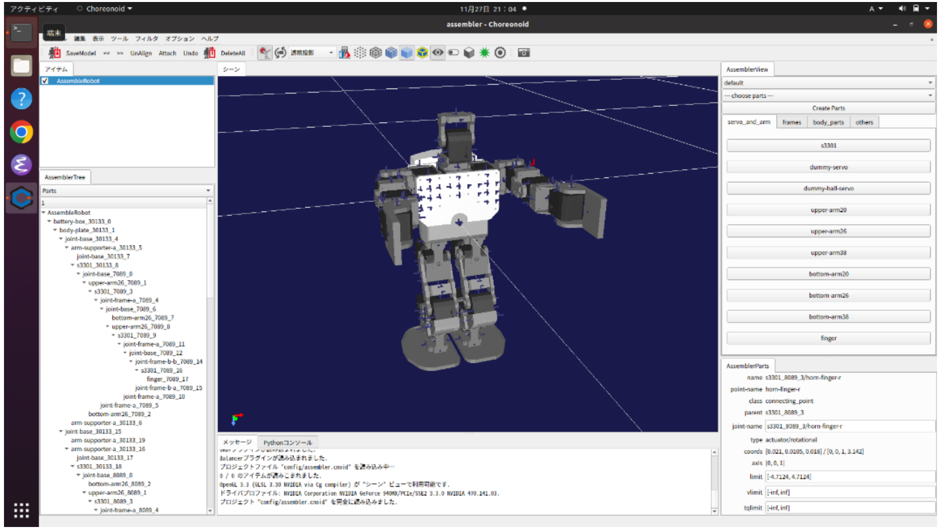
```
lfunc ;; glutil ;; gltexture ;; glprim ;; gleus ;; glview ;; toiv-undefined ;; fstringdouble irtma\
```

Real Time Factor: 0.30 Sim Time: 00 00:01:01.938 Real Time: 00 00:02:14.577 Iterations: 61938 FPS: 37.6496

Robot assemble system on Choreonoid

- Platform to configure a robot using actuator module
- Verify configured robot in simulation and in real world

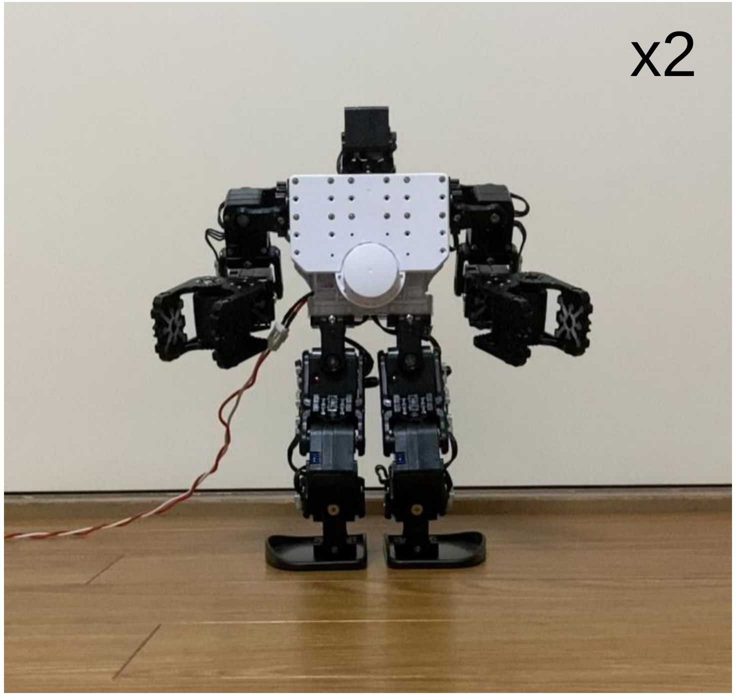
KXR (kondo kagaku)



```
1
▾ AssembleRobot
  ▾ battery-box_30133_0
    ▾ body-plate_30133_1
      ▾ joint-base_30133_4
        ▾ arm-supporter-a_30133_5
            joint-base_30133_7
          ▾ s3301_30133_8
            ▾ joint-base_7089_0
              ▾ upper-arm26_7089_1
                ▾ s3301_7089_3
                  ▾ joint-frame-a_7089_4
                    ▾ joint-base_7089_6
                        bottom-arm26_7089_7
                      ▾ upper-arm26_7089_8
                        ▾ s3301_7089_9
                          ▾ joint-frame-a_7089_11
                            ▾ joint-base_7089_12
                              ▾ joint-frame-b-b_7089_14
                                ▾ s3301_7089_16
                                    finger_7089_17
                                joint-frame-b-a_7089_15
                            joint-frame-a_7089_10
                    joint-frame-a_7089_5
                  bottom-arm26_7089_2
          arm-supporter-a_30133_6
      ▾ joint-base_30133_15
          arm-supporter-a_30133_19
        ▾ arm-supporter-a_30133_16
            joint-base_30133_17
          ▾ s3301_30133_18
            ▾ joint-base_8089_0
                bottom-arm26_8089_2
              ▾ upper-arm26_8089_1
                ▾ s3301_8089_3
                  ▾ joint-frame-a_8089_4
```

```
Balancerプラグインが読み込まれました.
プロジェクトファイル "config/assembler.cnoid" を読み込み中…
0 / 0 のアイテムが読みこまれました.
OpenGL 3.3 (GLSL 3.30 NVIDIA via Cg compiler) が "シーン" ビューで利用可能です.
ドライバプロファイル: NVIDIA Corporation NVIDIA GeForce 940MX/PCIe/SSE2 3.3.0 NVIDIA 470.141.03.
プロジェクト "config/assembler.cnoid" を完全に読み込みました.
```

name s3301_8089_3/horn-finger-r
point-name horn-finger-r
class connecting_point
parent s3301_8089_3
joint-name s3301_8089_3/horn-finger-r
type actuator/rotational
coords [0.021, 0.0105, 0.018] / [0, 0, 1, 3.142]
axis [0, 0, 1]
limit [-4.7124, 4.7124]
vlimit [-inf, inf]
tqlimit [-inf, inf]


x2

Outline

- Robot competitions and Simulation
- General information of Choreonoid
- Connecting to other system
- Development system on Choreonoid
- Learning Robot Programming

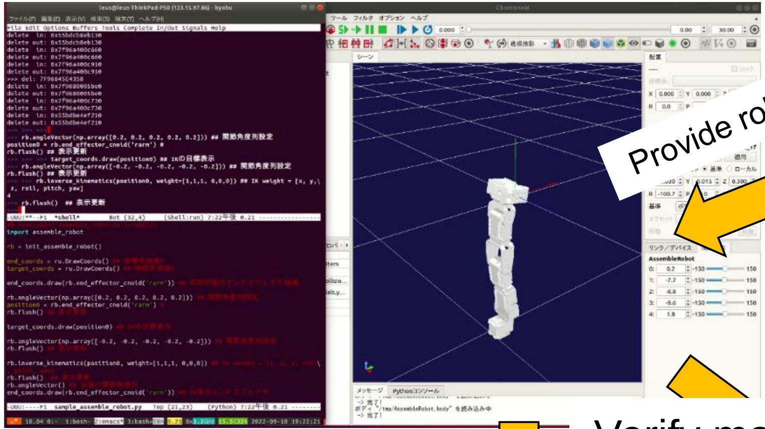
Education of Robot System using Choreonoid

Interactive Robot Programming

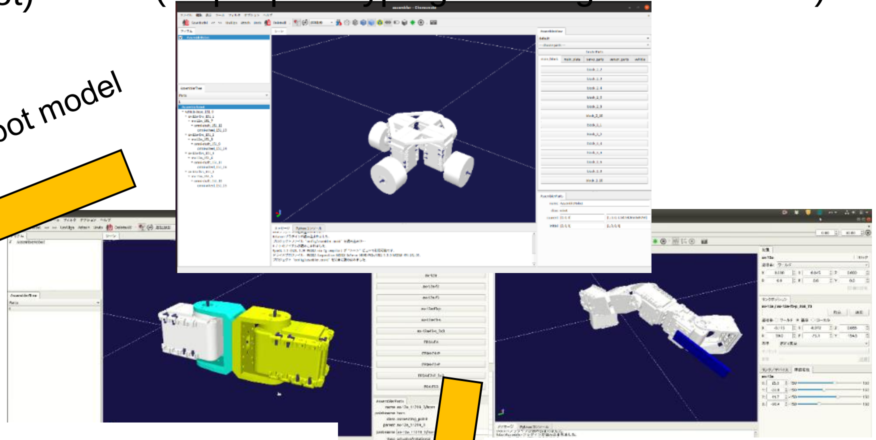
(Building behavior while verifying the motion of the robot)

Robot assembler

(Rapid prototyping of configurable robots)



```
File Edit Options Buffers Tools Complete In/Out Signals Help
delete  in: 0x55bdcb8eb130
delete out: 0x55bdcb8eb130
delete  in: 0x7f96a400c660
delete out: 0x7f96a400c660
delete  in: 0x7f96a400c910
delete out: 0x7f96a400c910
>>> del: 7F96645E4358
delete  in: 0x7f9688005be0
delete out: 0x7f9688005be0
delete  in: 0x7f96a400c730
delete out: 0x7f96a400c730
delete  in: 0x55bdbe4ef210
delete out: 0x55bdbe4ef210
>>> >>> >>>
>>> rb.angleVector(np.array([0.2, 0.2, 0.2, 0.2, 0.2])) ## 関節角度列設定
position0 = rb.end_effector_cnoid('rarm') #
rb.flush() ## 表示更新
>>> >>> >>> target_coords.draw(position0) ## IKの目標表示
>>> rb.angleVector(np.array([-0.2, -0.2, -0.2, -0.2, -0.2])) ## 関節角度列設定
rb.flush() ## 表示更新
>>> >>> rb.inverse_kinematics(position0, weight=[1,1,1, 0,0,0]) ## IK weight = [x, y,\
 z, roll, pitch, yaw]
4
>>> rb.flush()  ## 表示更新
>>>
-UUU:**--F1  *shell*        Bot (32,4)     (Shell:run) 7:22午後 0.21 ----------------
import assemble_robot

rb = init_assemble_robot()

end_coords = ru.DrawCoords()
target_coords = ru.DrawCoords()

end_coords.draw(rb.end_effector_cnoid('rarm'))

rb.angleVector(np.array([0.2, 0.2, 0.2, 0.2, 0.2]))
position0 = rb.end_effector_cnoid('rarm')
rb.flush()

target_coords.draw(position0)

rb.angleVector(np.array([-0.2, -0.2, -0.2, -0.2, -0.2]))
rb.flush()

rb.inverse_kinematics(position0, weight=[1,1,1, 0,0,0])
rb.flush()
rb.angleVector()
end_coords.draw(rb.end_effector_cnoid('rarm'))

-UUU:----F1  sample_assemble_robot.py   Top (21,23)    (Python) 7:22午後 0.21 --------
 18.04 0:-  1:bash- 2:emacs* 3:bash  0.21  8x3.2GHz 15.5G32% 2022-09-18 19:22:21
```

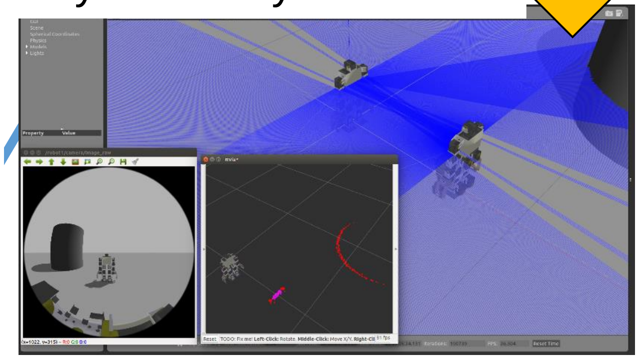


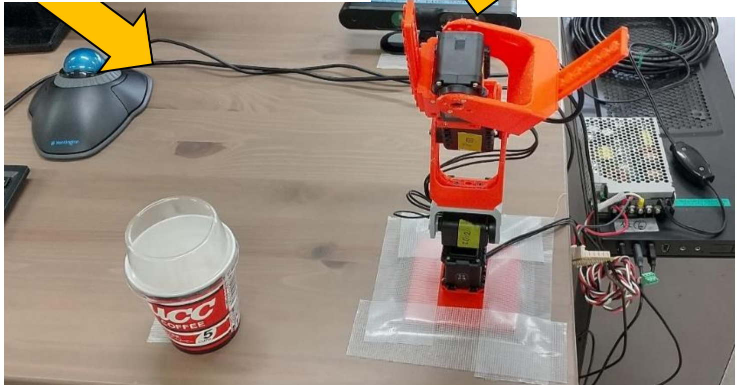
Provide robot model

Verify motion by real robot

Assemble real robot

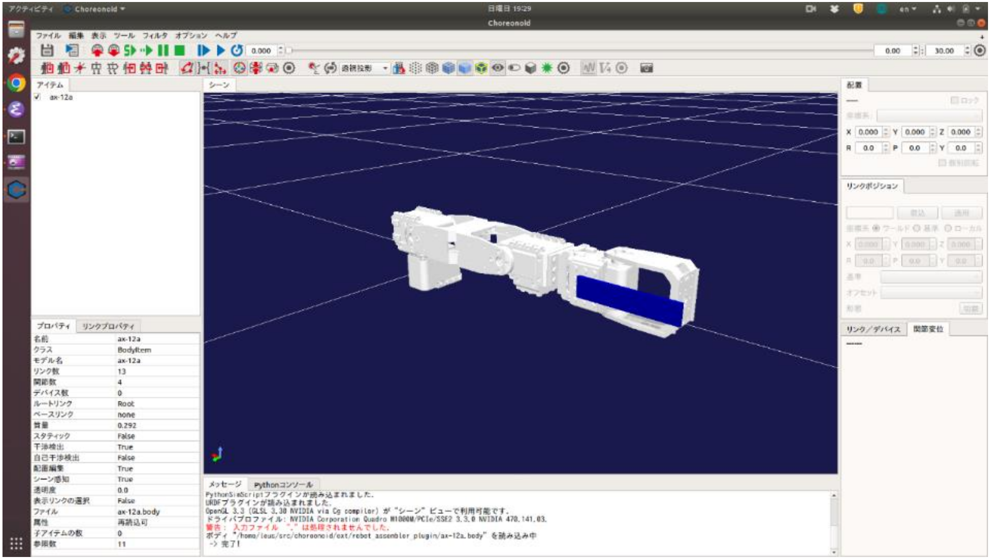
Verify motion by simulator





Education of Robot System using Choreonoid

- 1 week experiential learning
- For 3rd year undergraduate student
  - No familiarity with robot programming



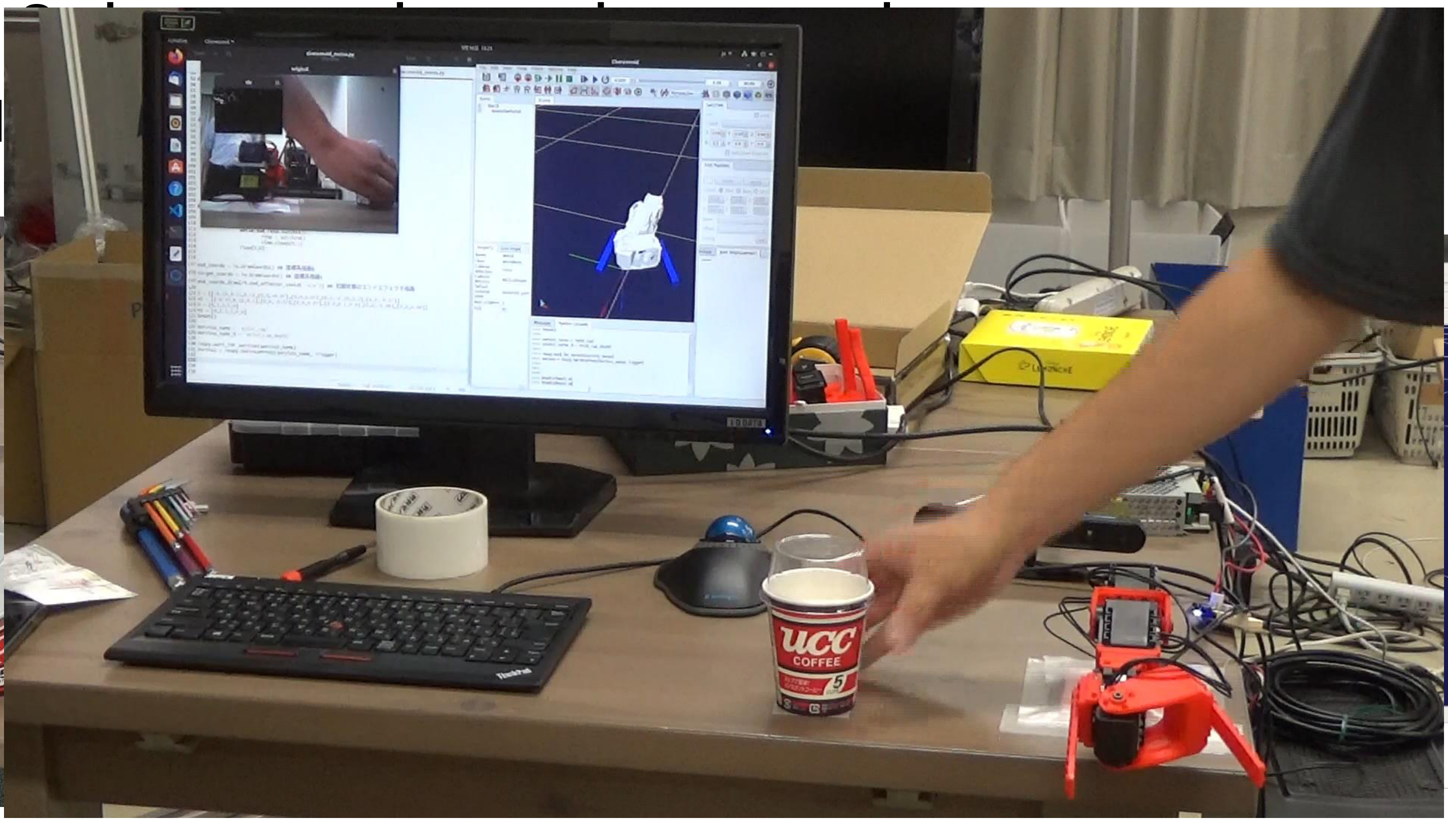
名前	ax-12a
クラス	BodyItem
モデル名	ax-12a
リンク数	13
関節数	4
デバイス数	0
ルートリンク	Root
ベースリンク	none
質量	0.292
スタティック	False
干渉検出	True
自己干渉検出	False
配置編集	True
シーン感知	True
透明度	0.0
表示リンクの選択	False
ファイル	ax-12a.body
属性	再読込可
子アイテムの数	0
参照数	11

```
PythonSimScriptプラグインが読み込まれました.
URDFプラグインが読み込まれました.
OpenGL 3.3 (GLSL 3.30 NVIDIA via Cg compiler) が "シーン" ビューで利用可能です.
ドライバプロファイル: NVIDIA Corporation Quadro M1000M/PCIe/SSE2 3.3.0 NVIDIA 470.141.03.
警告: 入力ファイル "." は処理されませんでした.
ボディ "/home/leus/src/choreonoid/ext/robot_assembler_plugin/ax-12a.body" を読み込み中
 -> 完了!
```

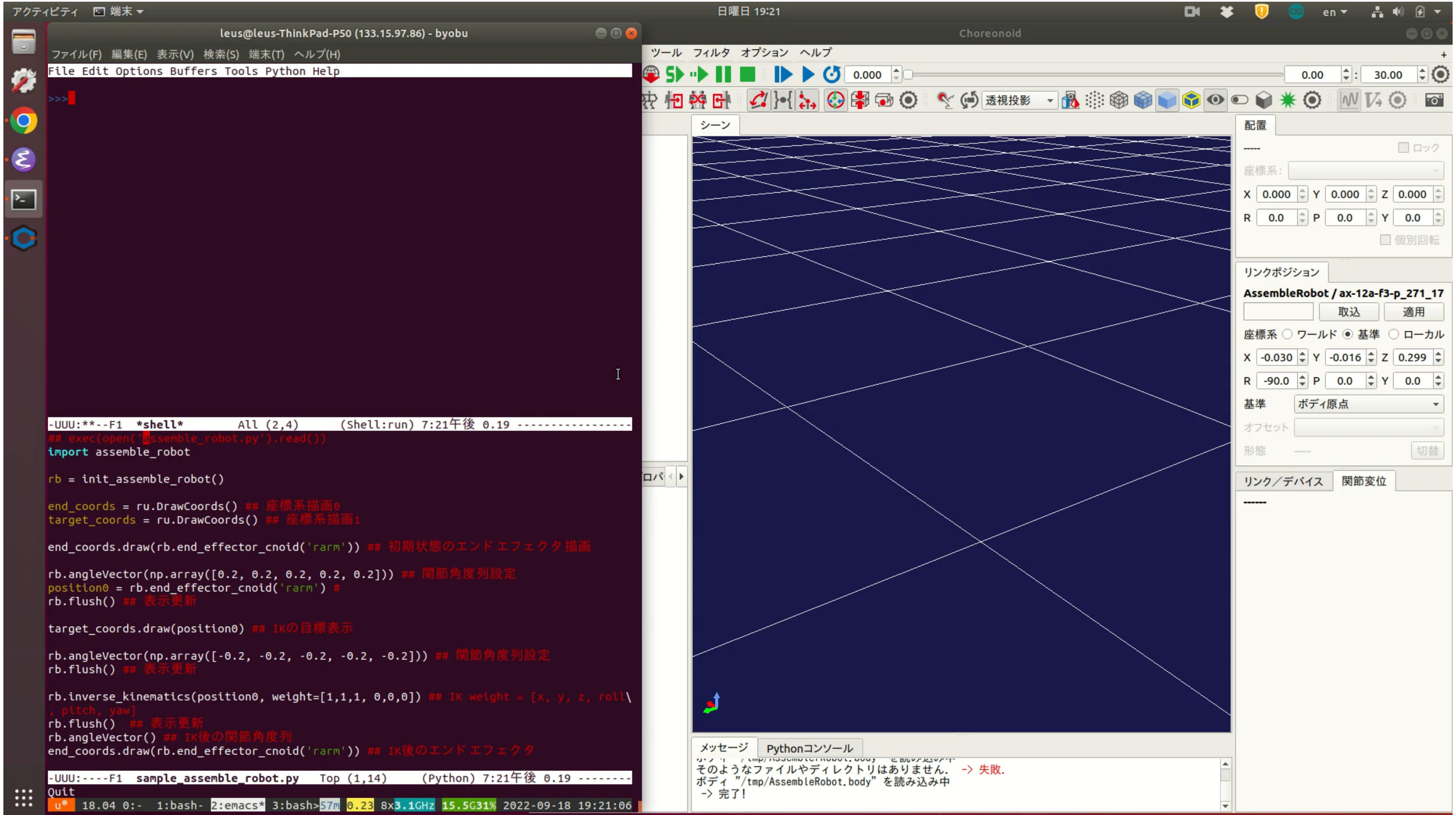
Education of Robot System using Choreonoid

- 1 week experiential learning
- For
  - N

x4



Interactive Robot Programming


アクティビティ
端末
日曜日 19:21
en
leus@leus-ThinkPad-P50 (133.15.97.86) - byobu
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
File Edit Options Buffers Tools Python Help
Choreonoid
ツール フィルタ オプション ヘルプ
透視投影
シーン
配置
ロック
座標系:
個別回転
リンクポジション
取込
適用
リンク／デバイス
関節変位
メッセージ
Pythonコンソール

```
>>> 
```

```
-UUU:**--F1  *shell*        All (2,4)      (Shell:run) 7:21午後 0.19 -----------------
## exec(open('assemble_robot.py').read())
import assemble_robot

rb = init_assemble_robot()

end_coords = ru.DrawCoords() ## 座標系描画0
target_coords = ru.DrawCoords() ## 座標系描画1

end_coords.draw(rb.end_effector_cnoid('rarm')) ## 初期状態のエンドエフェクタ描画

rb.angleVector(np.array([0.2, 0.2, 0.2, 0.2, 0.2])) ## 関節角度列設定
position0 = rb.end_effector_cnoid('rarm') #
rb.flush() ## 表示更新

target_coords.draw(position0) ## IKの目標表示

rb.angleVector(np.array([-0.2, -0.2, -0.2, -0.2, -0.2])) ## 関節角度列設定
rb.flush() ## 表示更新

rb.inverse_kinematics(position0, weight=[1,1,1, 0,0,0]) ## IK weight = [x, y, z, roll\
, pitch, yaw]
rb.flush()  ## 表示更新
rb.angleVector() ## IK後の関節角度列
end_coords.draw(rb.end_effector_cnoid('rarm')) ## IK後のエンドエフェクタ

-UUU:----F1  sample_assemble_robot.py   Top (1,14)    (Python) 7:21午後 0.19 --------
Quit
 u®  18.04 0:-  1:bash-  2:emacs*  3:bash>57m  0.23  8x3.1GHz  15.5G31%  2022-09-18 19:21:06
```

Time: 0.000 | 0.00 : 30.00

配置: -----

X	0.000	Y	0.000	Z	0.000
R	0.0	P	0.0	Y	0.0

AssembleRobot / ax-12a-f3-p_271_17

座標系 ○ ワールド ● 基準 ○ ローカル

X	-0.030	Y	-0.016	Z	0.299
R	-90.0	P	0.0	Y	0.0

基準: ボディ原点

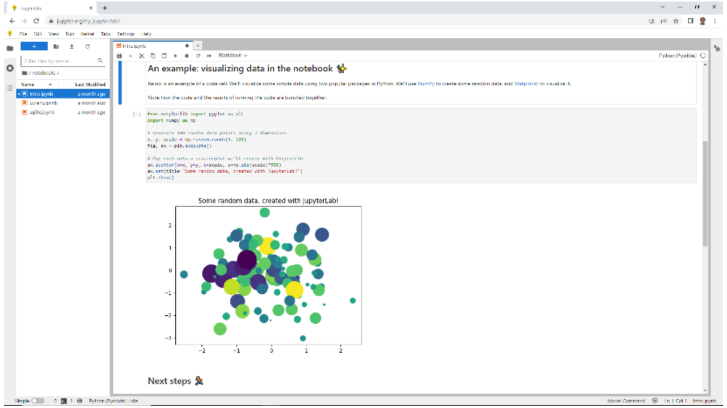
オフセット

形態 ----- 切替

```
そのようなファイルやディレクトリはありません.  -> 失敗.
ボディ "/tmp/AssembleRobot.body" を読み込み中
 -> 完了!
```

Learning programming using Jupyter Notebook

- Jupyter notebook
  - – Interactive computing
  - – Program execution and take a note
  - – Graphs and display views are also recorded
  - – Providing the notebooks you made
  - – Browser-based and enable to run in any environment
  - – Available in a various languages



An example: visualizing data in the notebook ✨

Below is an example of a code cell. We'll visualize some simple data using two popular packages in Python. We'll use NumPy to create some random data, and Matplotlib to visualize it.

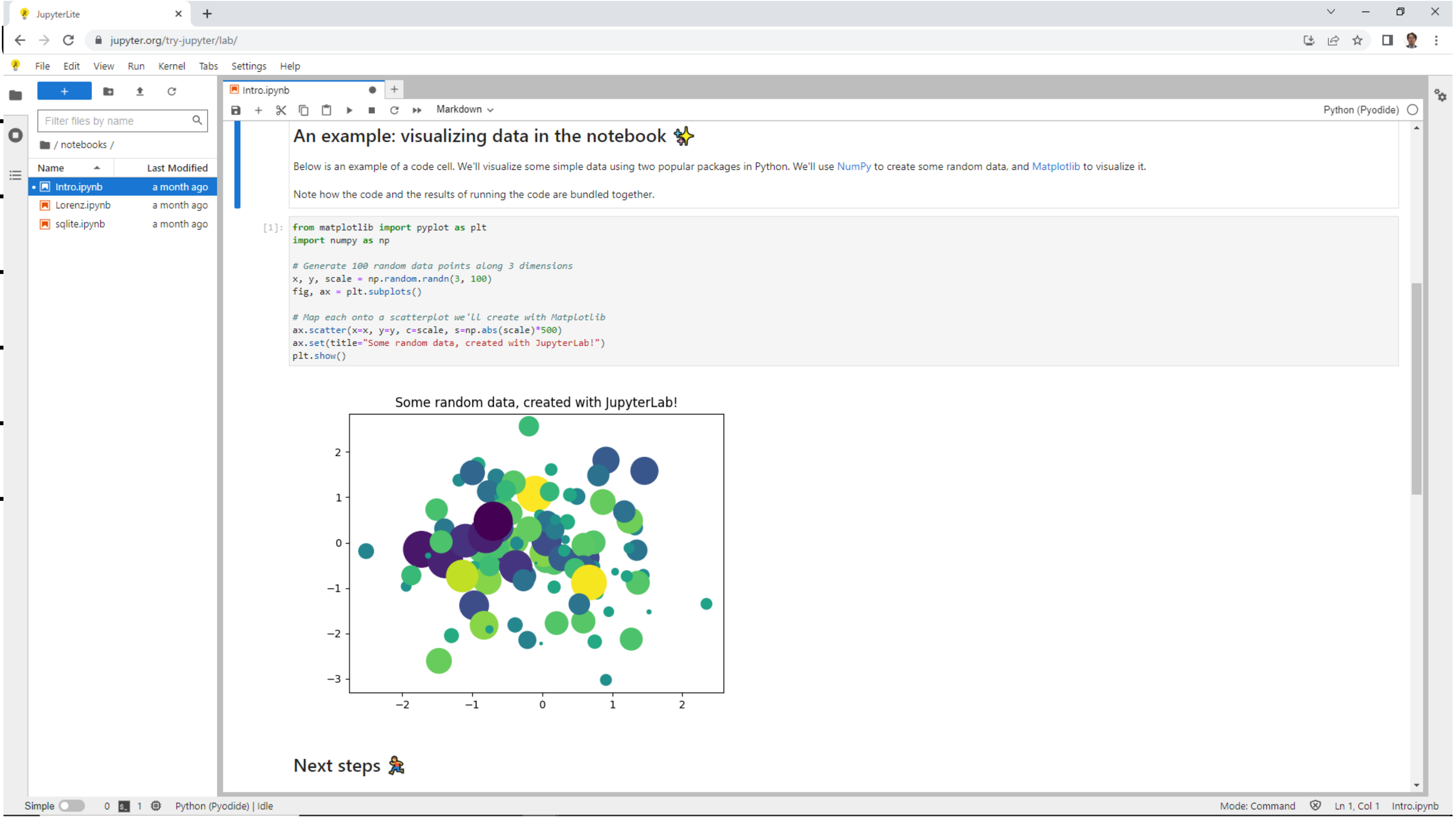
Note how the code and the results of running the code are bundled together.

```
from matplotlib import pyplot as plt
import numpy as np

# Generate 100 random data points along 3 dimensions
x, y, scale = np.random.randn(3, 100)
fig, ax = plt.subplots()

# Map each onto a scatterplot we'll create with Matplotlib
ax.scatter(x=x, y=y, c=scale, s=np.abs(scale)*500)
ax.set(title="Some random data, created with JupyterLab!")
plt.show()
```

Next steps 🏃

Learning programming using Jupyter Notebook

- J



An example: visualizing data in the notebook ✨

Below is an example of a code cell. We'll visualize some simple data using two popular packages in Python. We'll use NumPy to create some random data, and Matplotlib to visualize it.

Note how the code and the results of running the code are bundled together.

[1]:

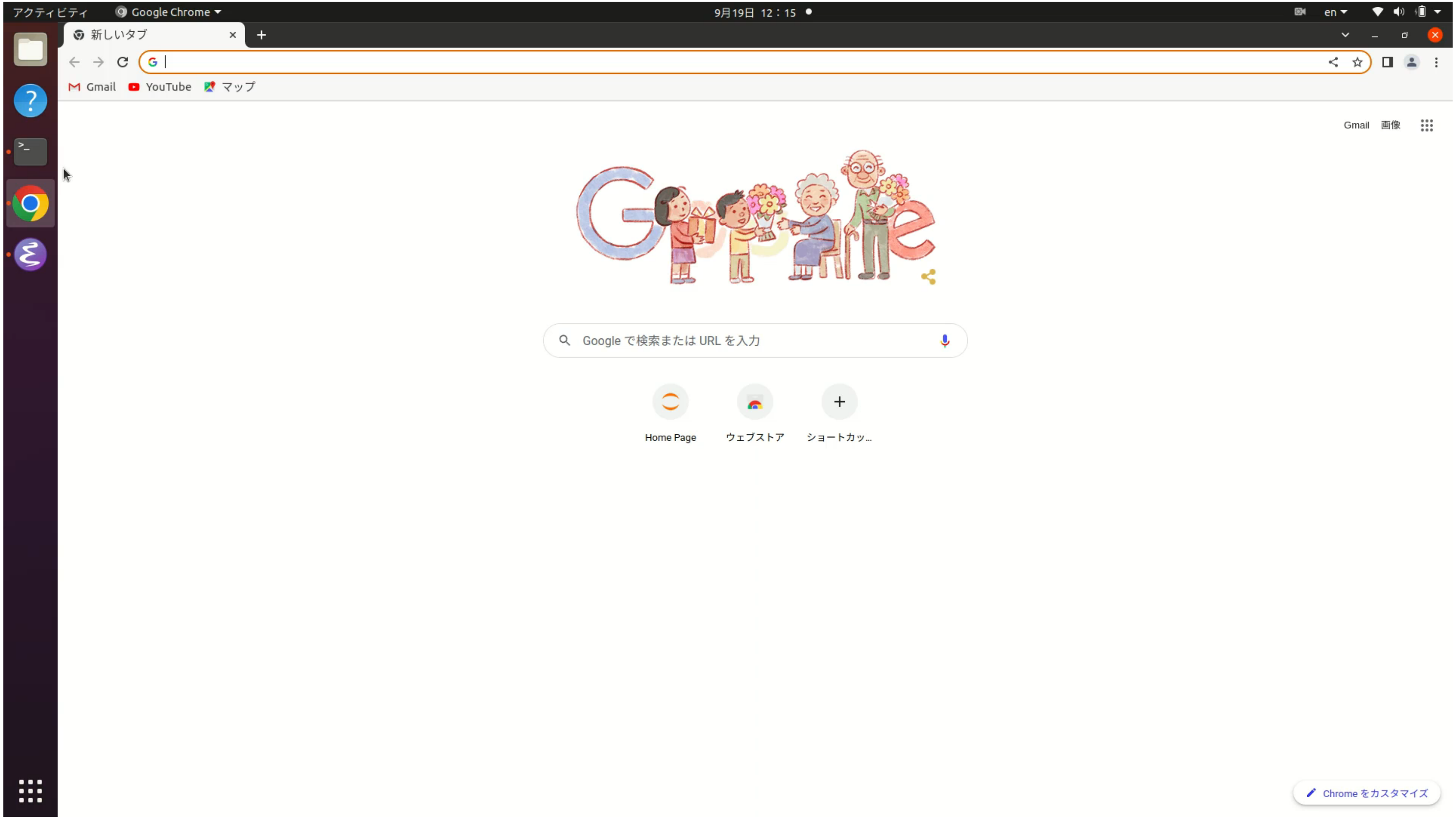
```
from matplotlib import pyplot as plt
import numpy as np

# Generate 100 random data points along 3 dimensions
x, y, scale = np.random.randn(3, 100)
fig, ax = plt.subplots()

# Map each onto a scatterplot we'll create with Matplotlib
ax.scatter(x=x, y=y, c=scale, s=np.abs(scale)*500)
ax.set(title="Some random data, created with JupyterLab!")
plt.show()
```

Next steps 🏃

Using Jupyter Notebook with Choreonoid

- Implement Jupyter kernel using xeus
  - https://github.com/jupyter-xeus/xeus
  - C++ interface library
- Implement Choreonoid Plugin
  - https://github.com/IRSL-tut/jupyter_plugin
  - Learning interactive robot programming

Using Jupyter Notebook with Choreonoid



x2