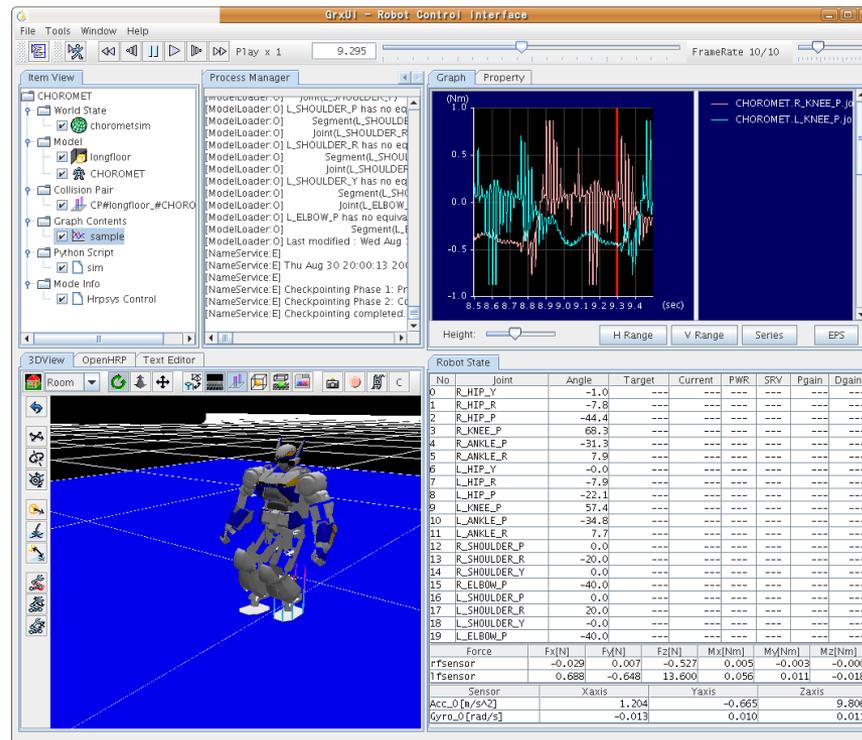


OpenHRP3講習会

GrxUIの利用方法について



ゼネラルロボティクス株式会社

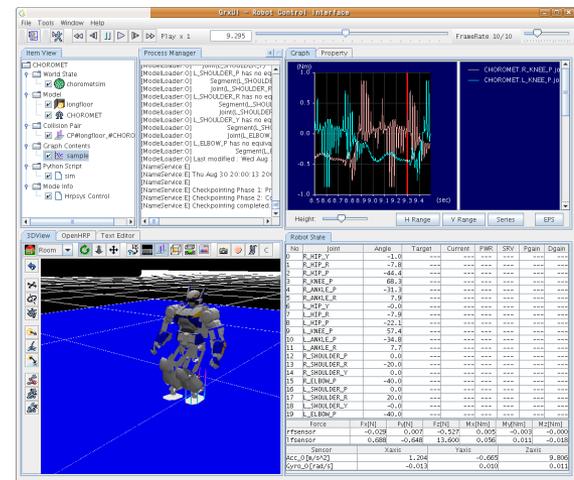
本日の内容

1. GrxUIの概要

2. サンプルプロジェクトの実行

3. プロジェクトの作成

4. その他の機能



1.GrxUIの概要

OpenHRP3実行に必要な基本機能

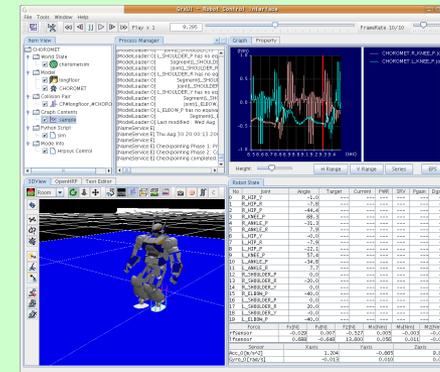
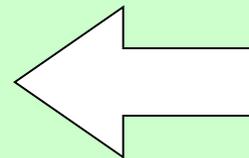
OpenHRP3

Model Loader

Collision Detector

Dynamics Simulator

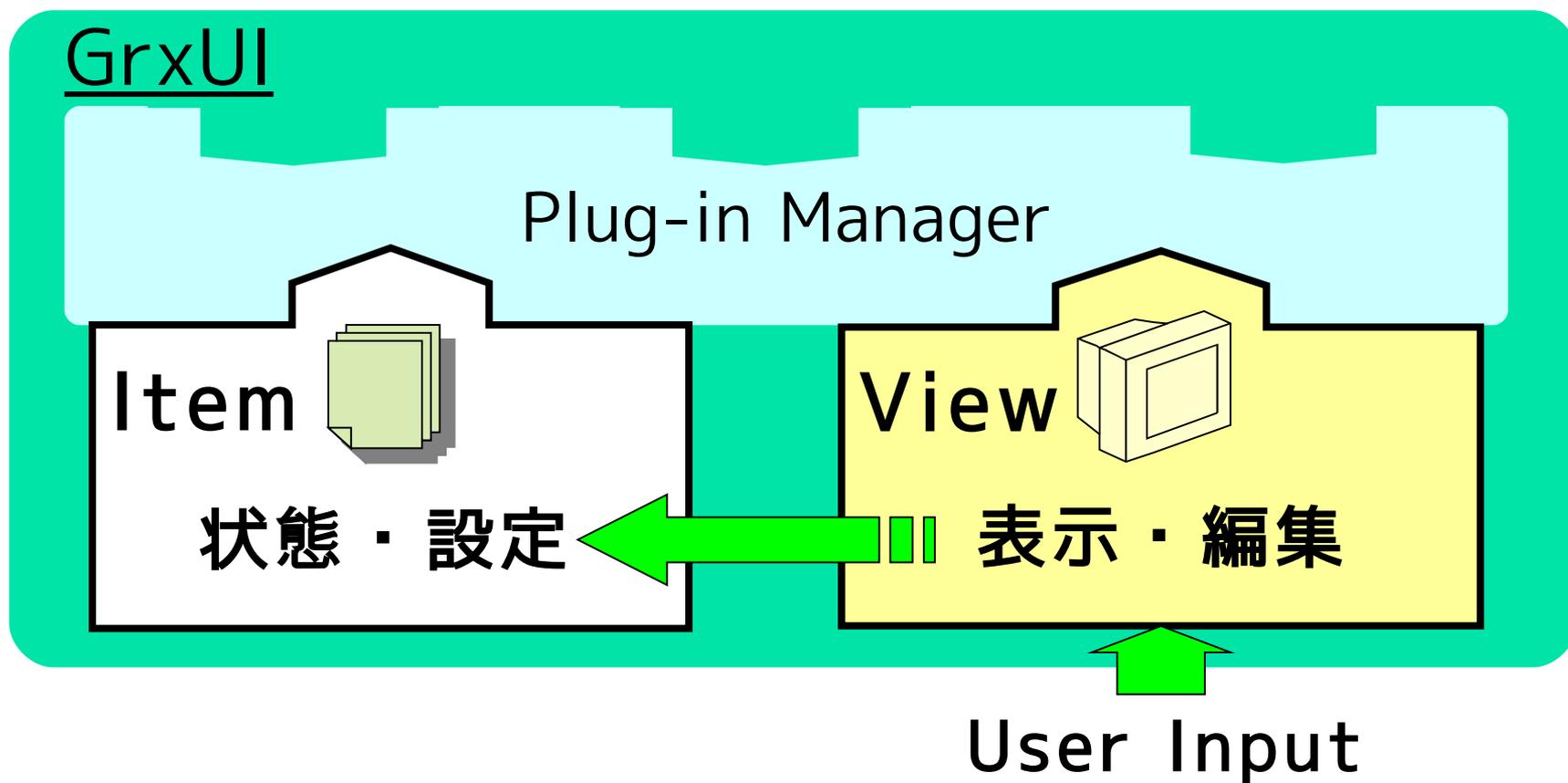
Robot Controller



- ・ 各サーバの起動・設定
- ・ スケジューリング
- ・ 結果の表示・保存

1.GruxUIの概要

必要な機能のPlug-in拡張



1.GrxUIの概要

Itemの例

World State



ロボットの状態量、干渉データのログ

Model

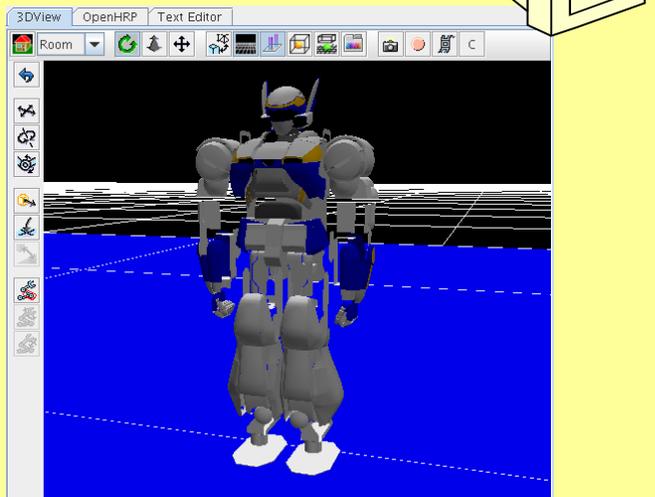
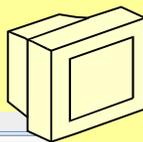


ロボット・環境の物理・表示モデル

1.GrxUIの概要

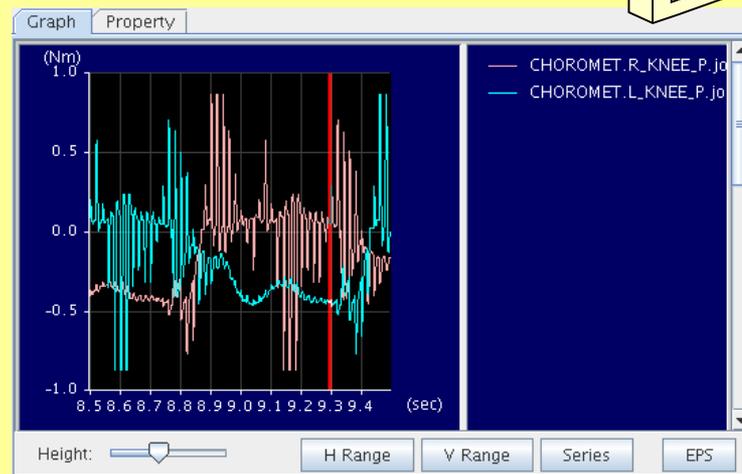
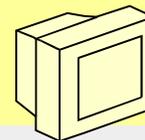
Viewの例

3DView



- ・ 3D表示・録画
- ・ 初期条件の設定

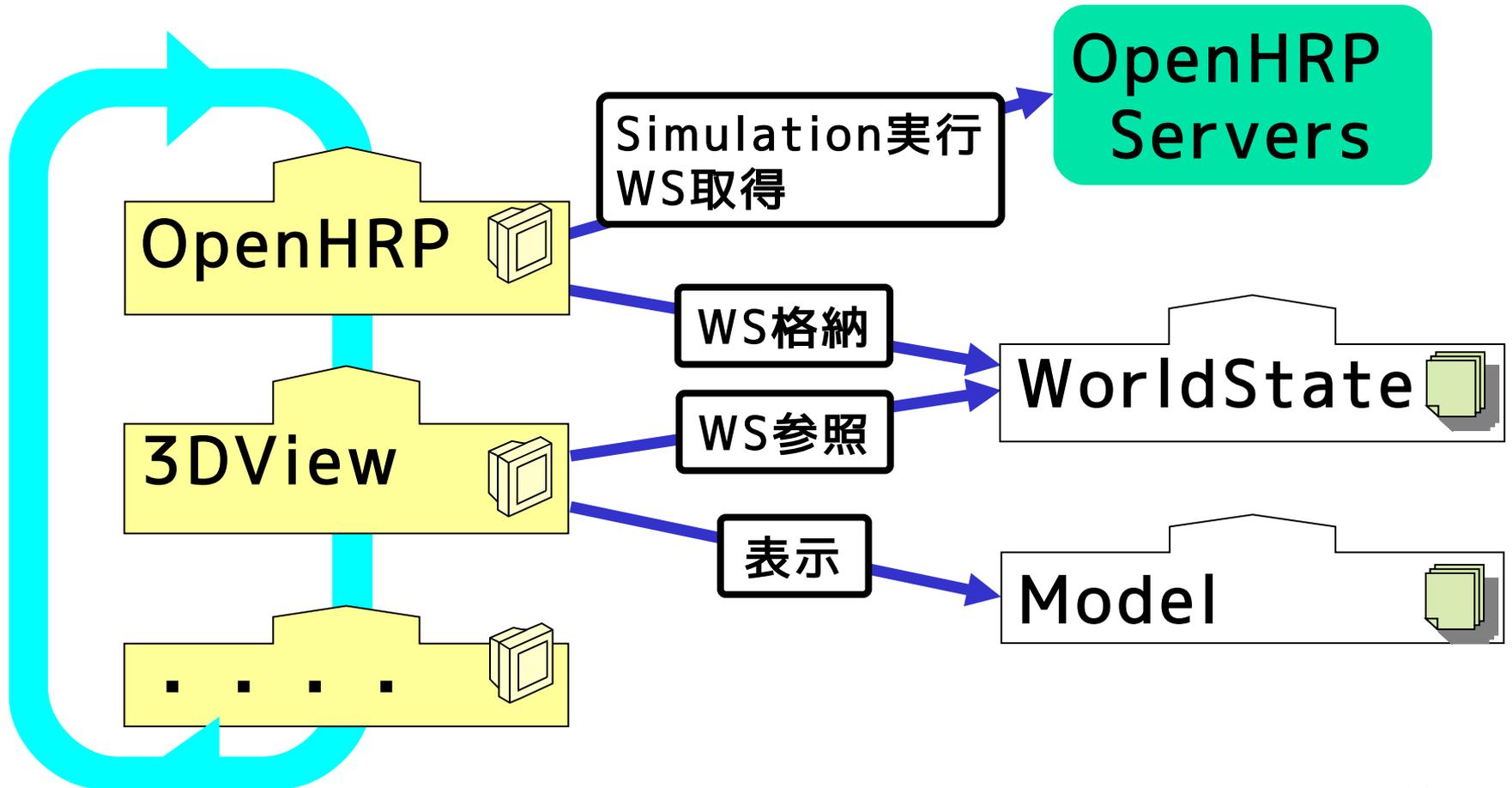
Graph



- ・ グラフ表示
- ・ 表示内容の設定

1.GrxUIの概要

OpenHRP3の実行

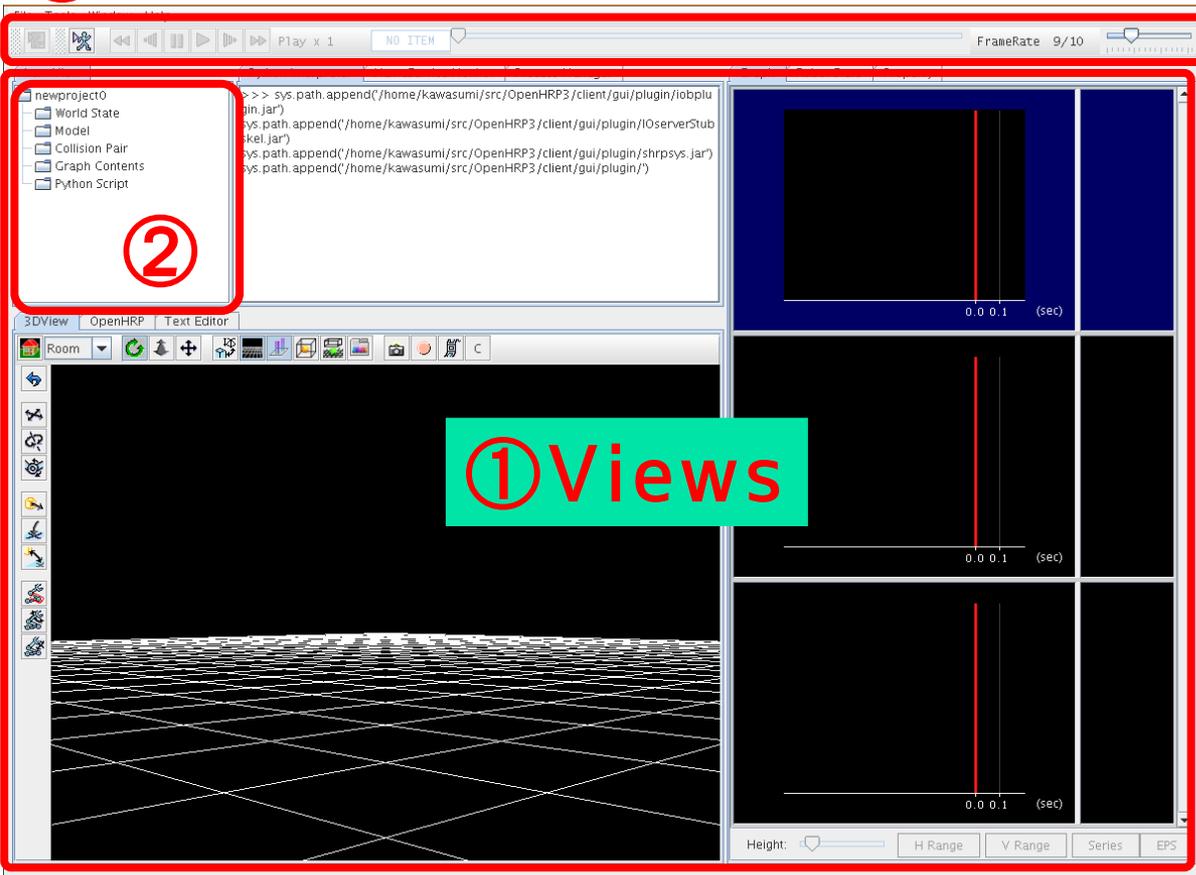


*WS ... Simulation世界の状態量

1.GrxUIの概要

画面構成

③



① Views
各ViewのGUIパネル

② Item View
Itemをツリー表示

③ Toolbar
よく使う機能を登録

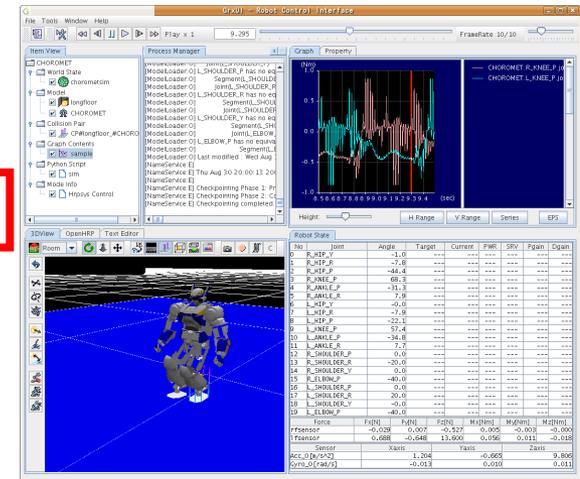
次は . . .

1. GrxUIの概要

2. サンプルプロジェクトの実行

3. プロジェクトの作成

4. その他の機能

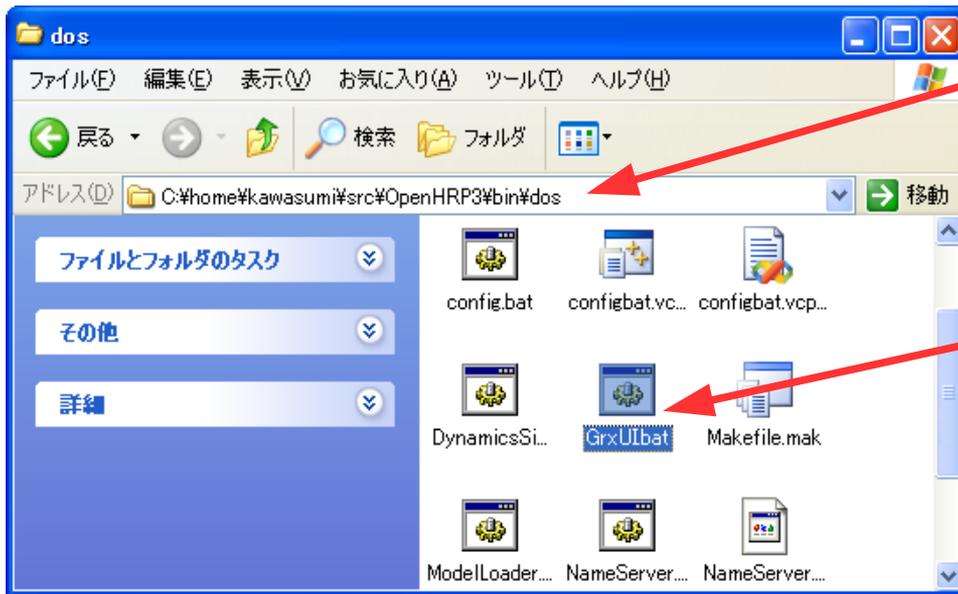


サンプルプロジェクトを利用して、GrxUIの基本的な操作方法について解説を行います。

2. サンプルプロジェクトの実行

GrxUIの起動

Windowsの場合



1. **OpenHRP3\bin\dos**
を開く

2. **GrxUI.bat**
をダブルクリック

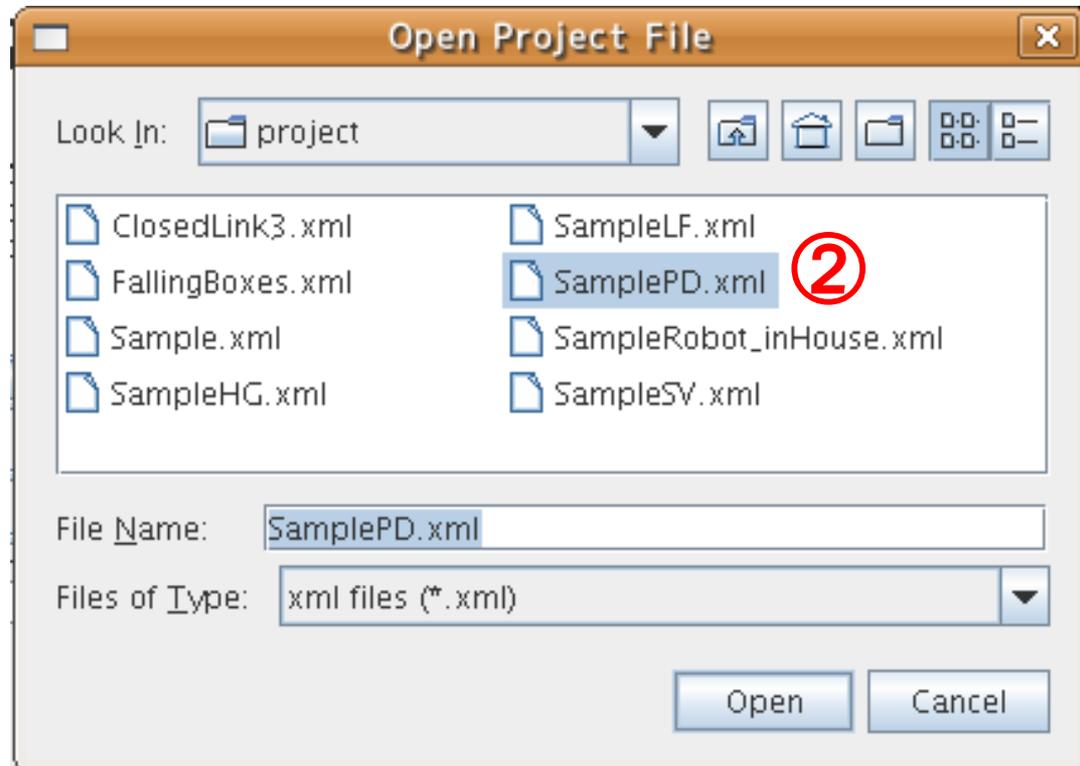
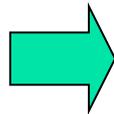
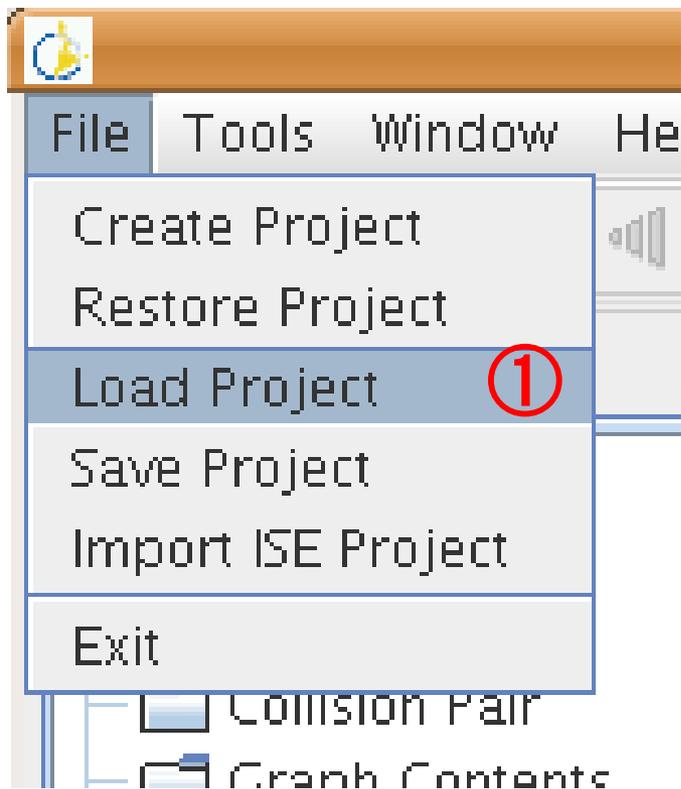
Linuxの場合

```
$ cd (OpenHRP3インストール先) /bin/unix  
$ ./GrxUI.sh
```

2. サンプルプロジェクトの実行

プロジェクトのロード

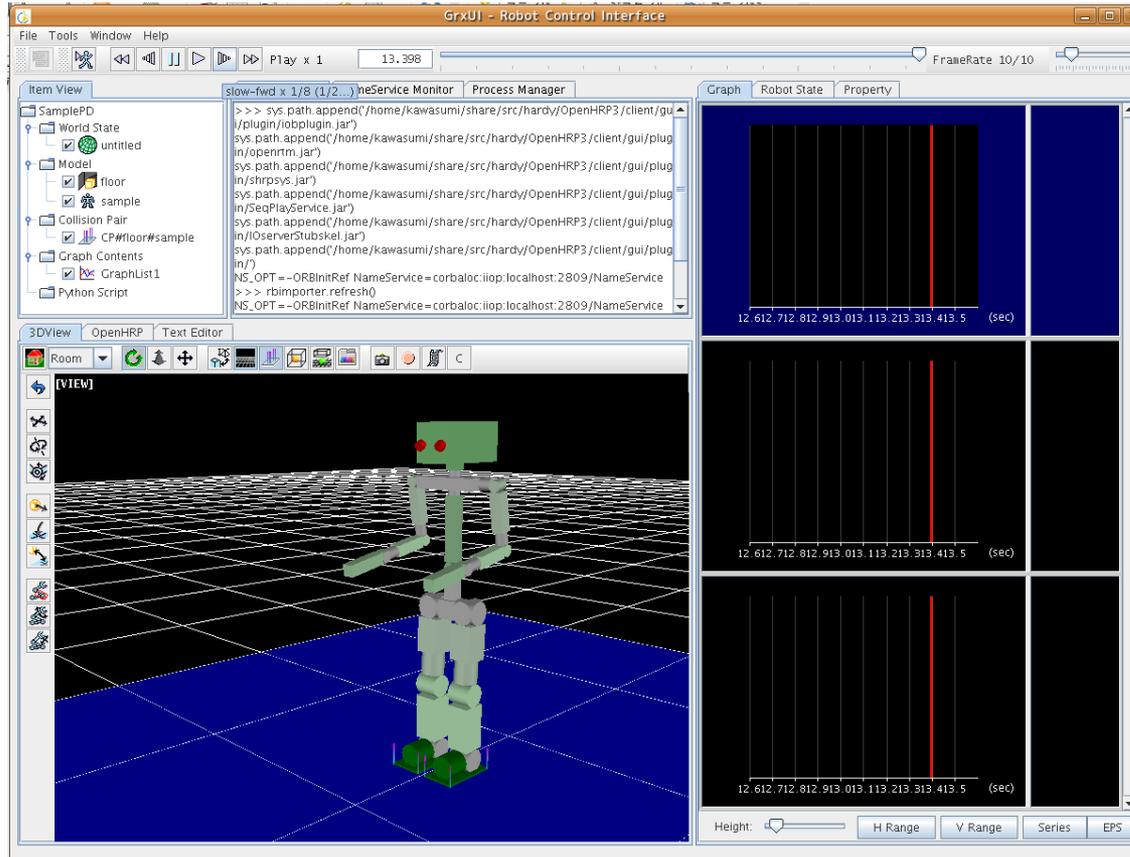
- ① File > Load Projectを選択
- ② SamplePD.xmlを開く



2. サンプルプロジェクトの実行

SamplePD.xml

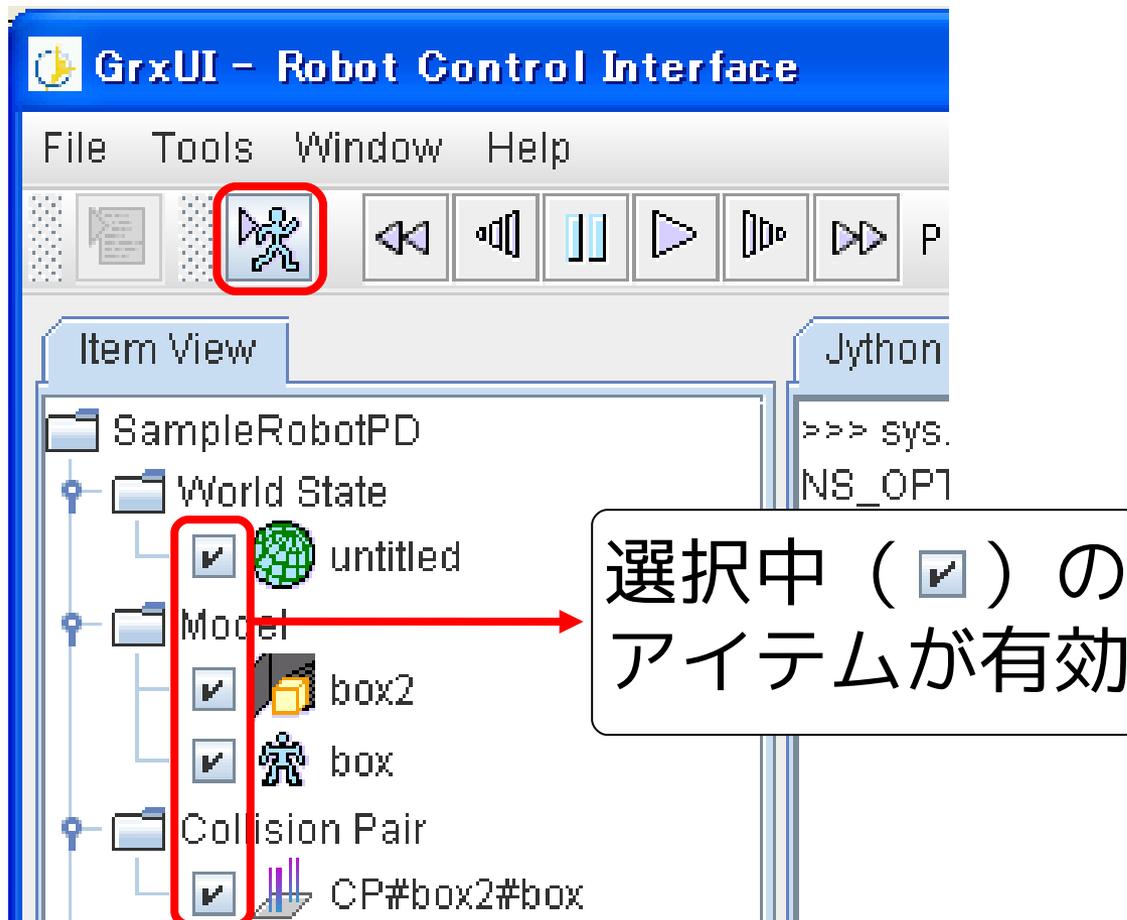
サンプルロボットの歩行シミュレーション



2. サンプルプロジェクトの実行

シミュレーションの開始

シミュレーション開始ボタン  を押す

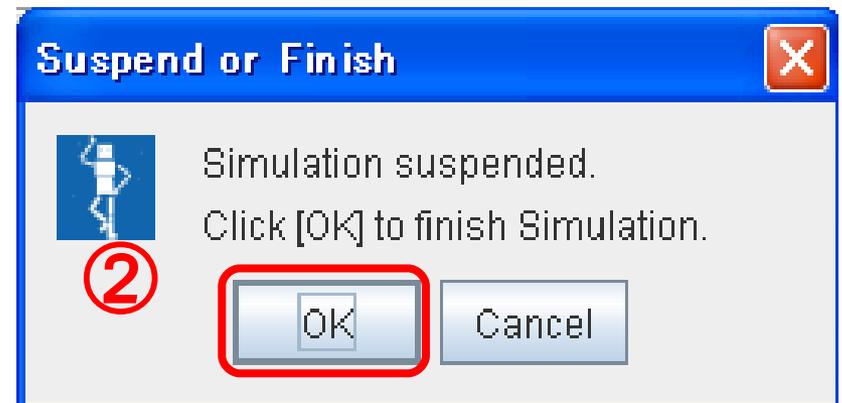
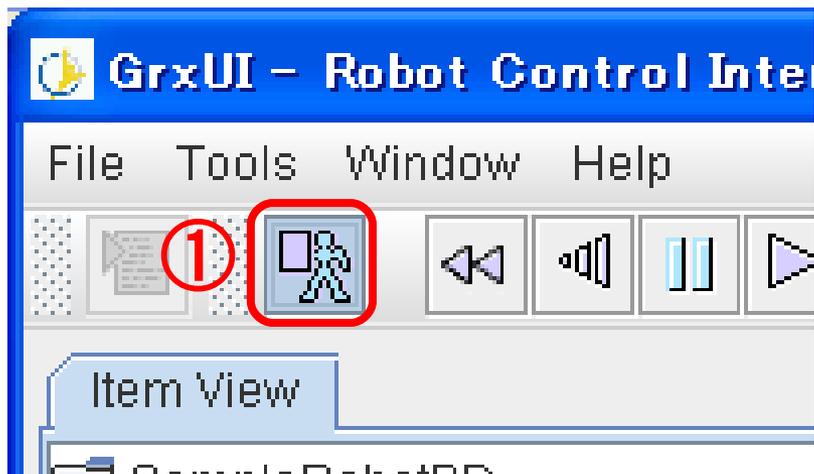


2. サンプルプロジェクトの実行

シミュレーションの中断・終了

設定時間が来る前に終了したい場合は . . .

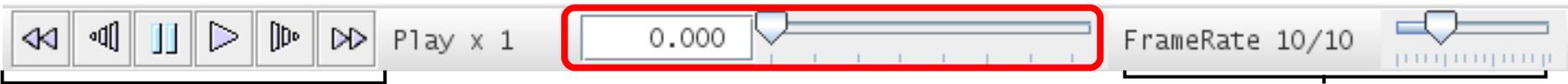
- ① シミュレーション終了ボタンを押す
- ② [OK]ボタン: 終了 [Cancel]: 続行



2. サンプルプロジェクトの実行

Playback機能

再生ボタン*やSlider操作でシミュレーション結果を確認



PlaybackToolbar

TimeSlider

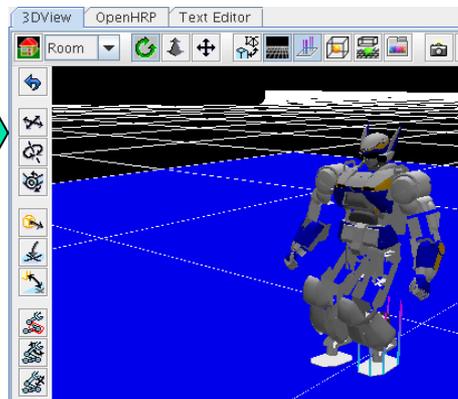
FrameRateスライダ

Item View

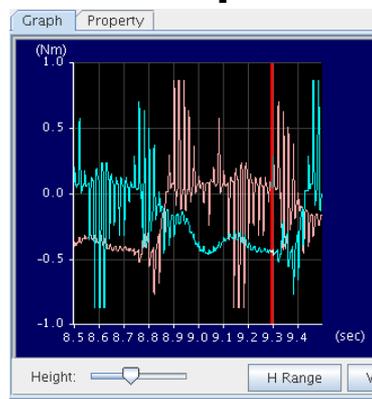
- CHOROMET
 - World State
 - chorometsim
 - Model
 - longfloor

World Stateに結果を格納

3DView



Graph



Robot Stat

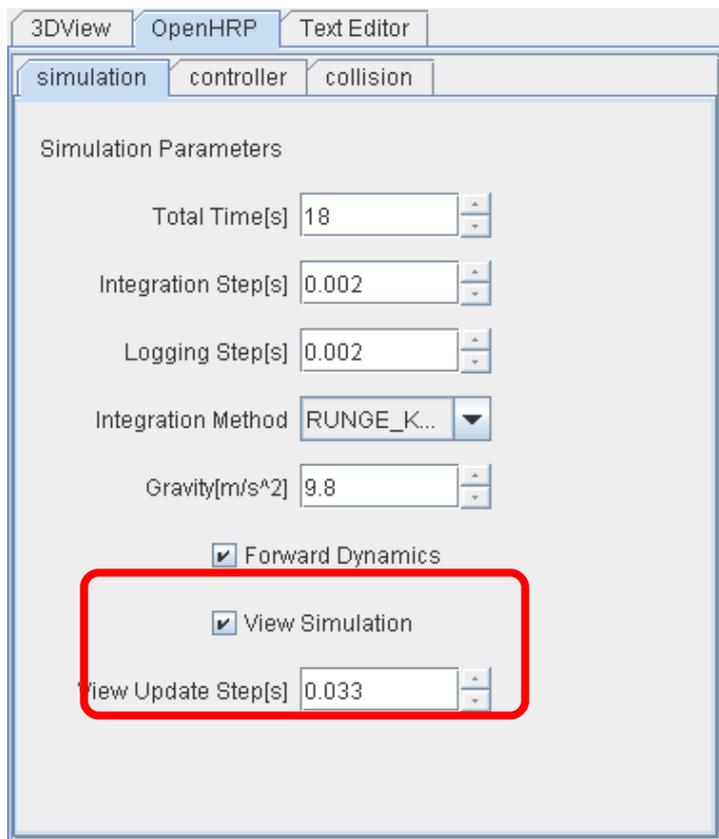
No.	Joint	Angle	Target	Current	PWR	SRV	Pgain	Dgain
0	R_HIP_Y	-1.0	---	---	---	---	---	---
1	R_HIP_R	-7.8	---	---	---	---	---	---
2	R_HIP_P	-44.4	---	---	---	---	---	---
3	R_KNEE_P	68.3	---	---	---	---	---	---
4	R_ANKLE_P	-31.3	---	---	---	---	---	---
5	R_ANKLE_R	7.9	---	---	---	---	---	---
6	L_HIP_Y	-0.0	---	---	---	---	---	---
7	L_HIP_R	-7.9	---	---	---	---	---	---
8	L_HIP_P	-22.1	---	---	---	---	---	---
9	L_KNEE_P	57.4	---	---	---	---	---	---
10	L_ANKLE_P	-34.8	---	---	---	---	---	---
11	L_ANKLE_R	7.7	---	---	---	---	---	---
12	R_SHOULDER_P	0.0	---	---	---	---	---	---
13	R_SHOULDER_R	-20.0	---	---	---	---	---	---
14	R_SHOULDER_Y	0.0	---	---	---	---	---	---
15	R_ELBOW_P	-40.0	---	---	---	---	---	---
16	L_SHOULDER_P	0.0	---	---	---	---	---	---
17	L_SHOULDER_R	20.0	---	---	---	---	---	---
18	L_SHOULDER_Y	-0.0	---	---	---	---	---	---
19	L_ELBOW_P	-40.0	---	---	---	---	---	---
Force		Fx[N]	Fy[N]	Fz[N]	Mx[Nm]	My[Nm]	Mz[Nm]	
rfsensor		-0.029	0.007	-0.527	0.005	-0.003	-0.000	
tsensor		0.688	-0.648	13.600	0.056	0.011	-0.018	
Sensor		Xaxis		Yaxis		Zaxis		
Acc_0 [m/s^2]		1.204		-0.665		9.806		
Gyro_0 [rad/s]		-0.013		0.010		0.011		

TimeSliderの指す時刻のWorld Stateを表示

2. サンプルプロジェクトの実行

ViewSimulation機能

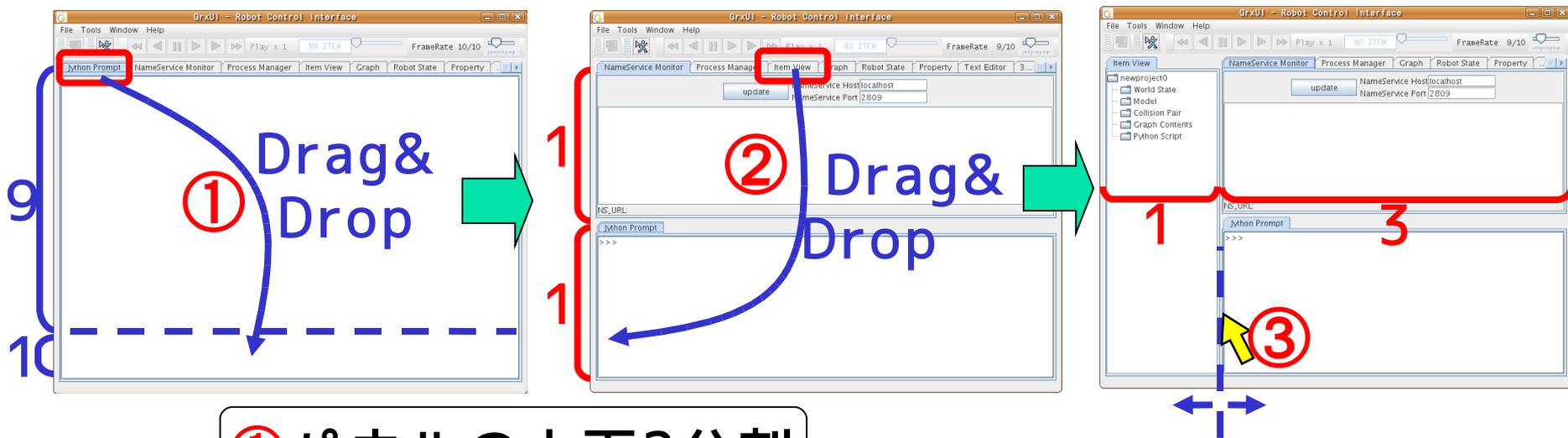
- View Simulationを有効にしてからSimulation開始
- CORBA経由で画像や距離情報を取得



2. サンプルプロジェクトの実行

画面構成変更 1

② 端部へのDragで1:3分割
上下の場合も同様

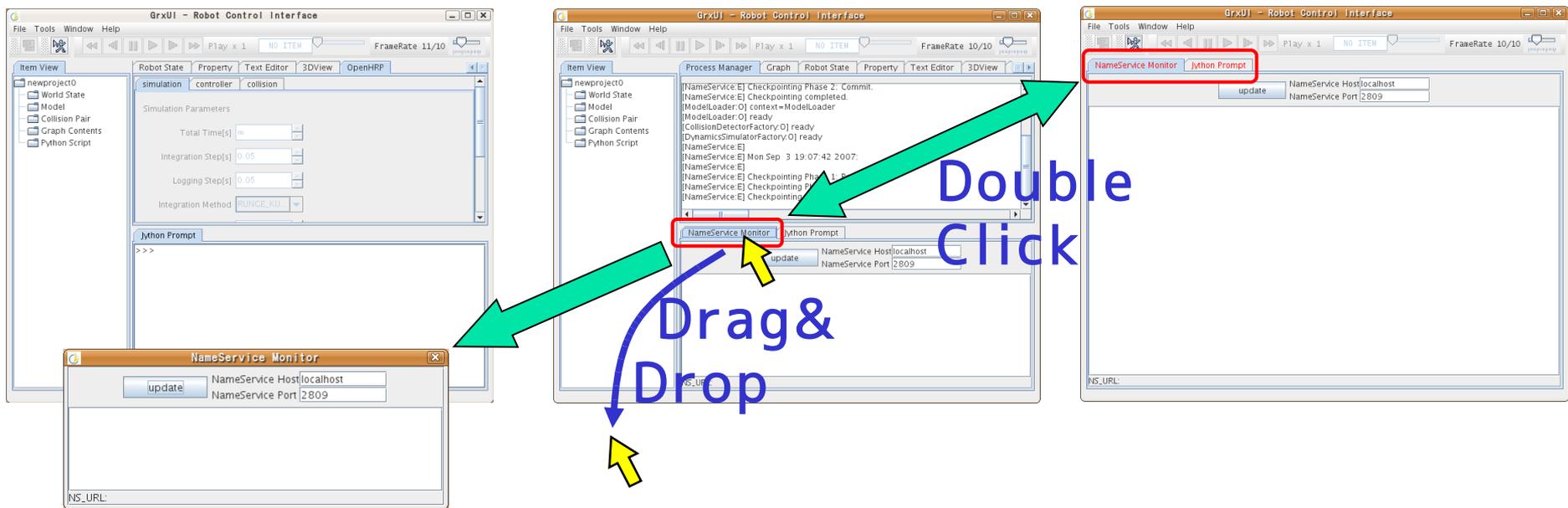


① パネルの上下2分割
左右の場合も同様

③ 区切り部のDragで微調整

2. サンプルプロジェクトの実行 画面構成変更 2

Double Clickで最大化

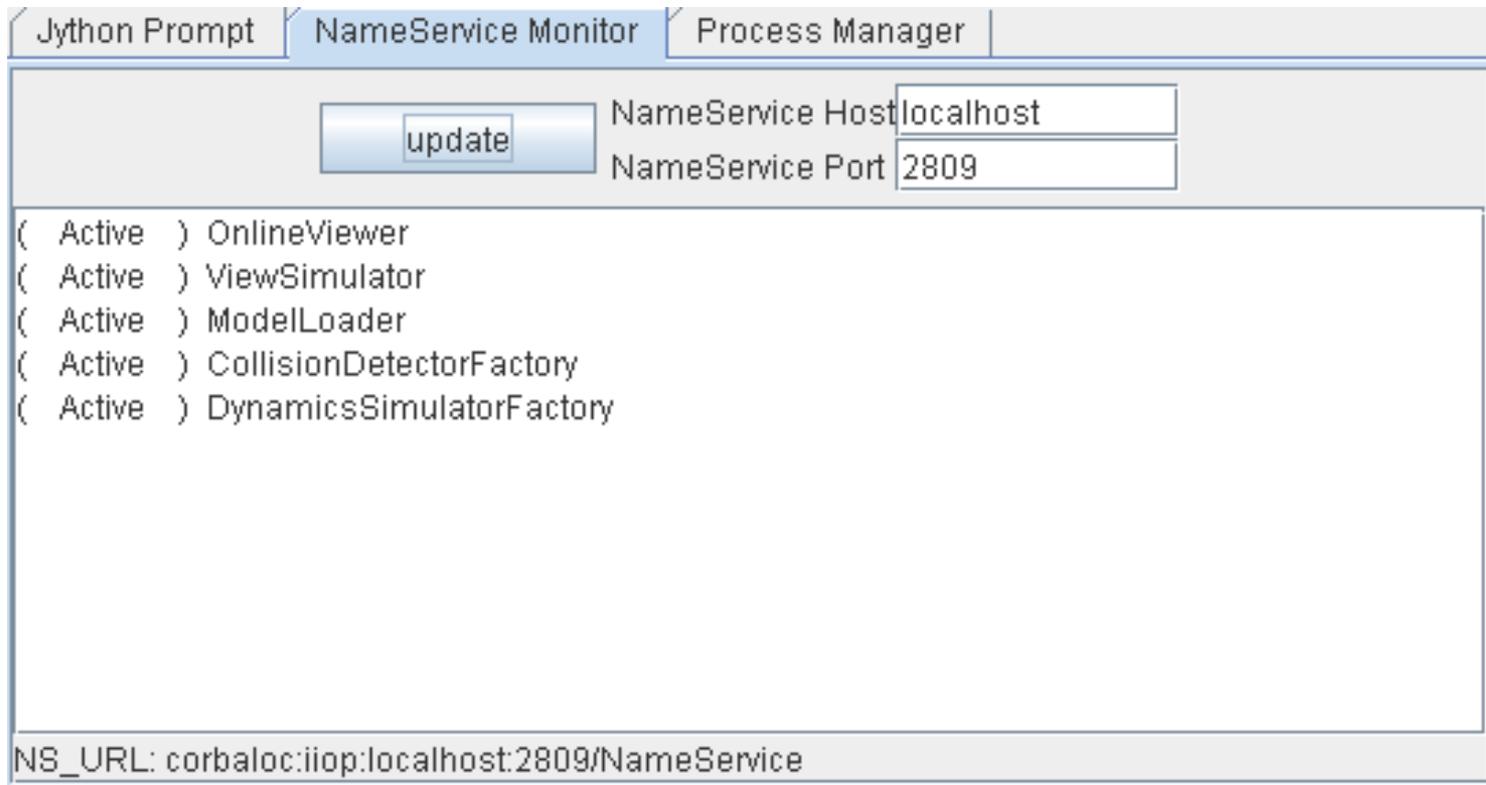


Window外への
Drag&DropでWindow化

2. サンプルプロジェクトの実行

NameService Monitor View

CORBA NameServiceに登録された名前リストを表示



The screenshot shows a software interface with three tabs: "Jython Prompt", "NameService Monitor", and "Process Manager". The "NameService Monitor" tab is active. It features an "update" button and two input fields: "NameService Host" with the value "localhost" and "NameService Port" with the value "2809". Below these fields is a list of active objects:

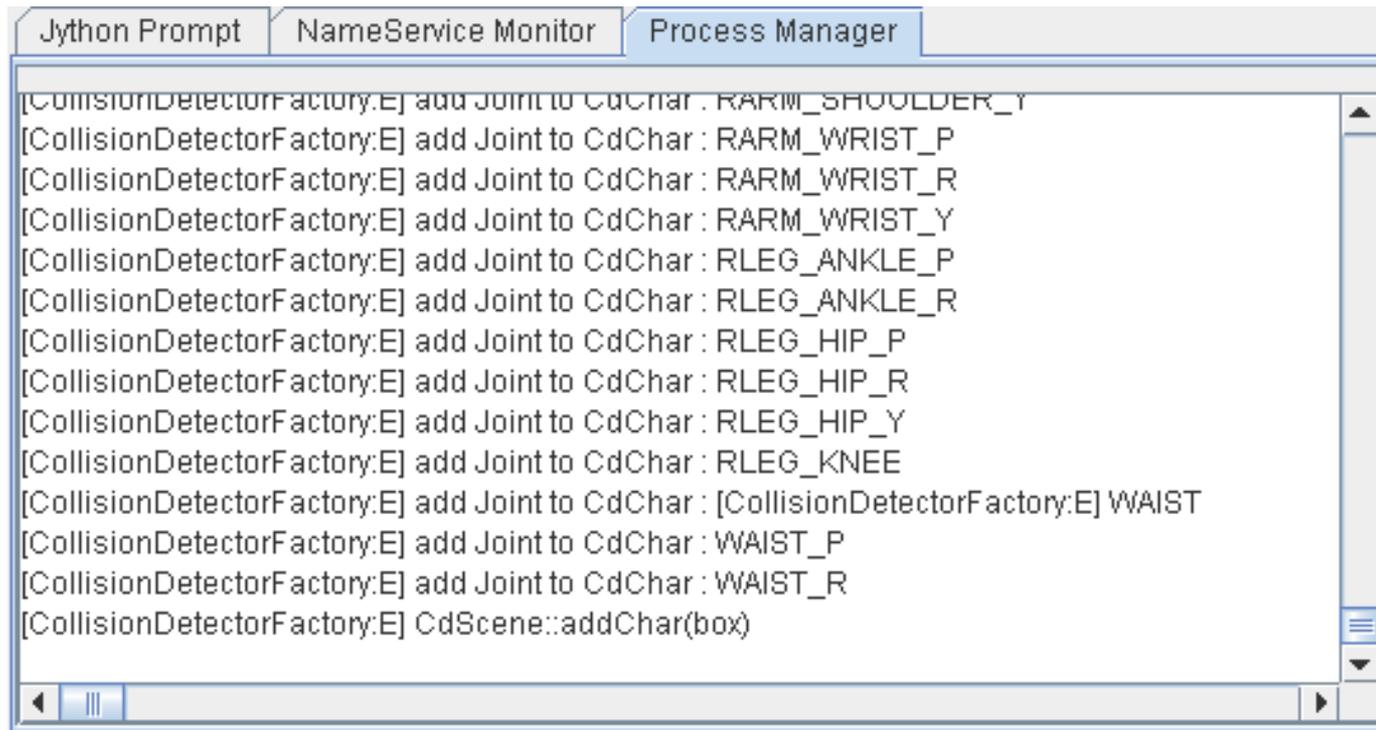
- (Active) OnlineViewer
- (Active) ViewSimulator
- (Active) ModelLoader
- (Active) CollisionDetectorFactory
- (Active) DynamicsSimulatorFactory

At the bottom of the interface, the NS_URL is displayed as: NS_URL: corbaloc:iiop:localhost:2809/NameService

2. サンプルプロジェクトの実行

Process Manager View

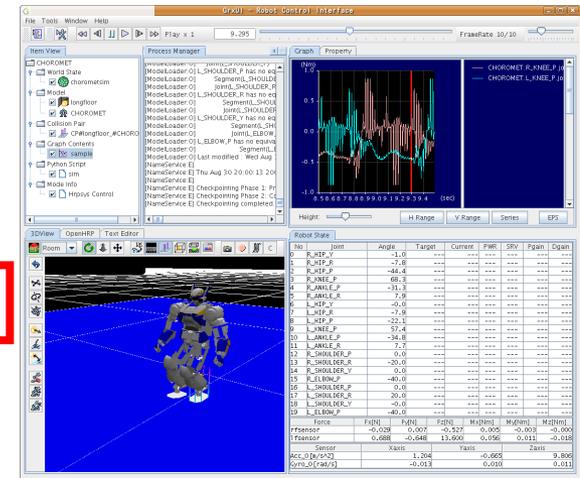
- ・ GrxUIから起動した各プロセスの出力を表示
- ・ 右クリックメニューで個別の表示/非表示設定
- ・ Tools->Process Managerで起動・終了



```
Jython Prompt | NameService Monitor | Process Manager  
[CollisionDetectorFactory:E] add Joint to CdChar : RARM_SHOULDER_Y  
[CollisionDetectorFactory:E] add Joint to CdChar : RARM_WRIST_P  
[CollisionDetectorFactory:E] add Joint to CdChar : RARM_WRIST_R  
[CollisionDetectorFactory:E] add Joint to CdChar : RARM_WRIST_Y  
[CollisionDetectorFactory:E] add Joint to CdChar : RLEG_ANKLE_P  
[CollisionDetectorFactory:E] add Joint to CdChar : RLEG_ANKLE_R  
[CollisionDetectorFactory:E] add Joint to CdChar : RLEG_HIP_P  
[CollisionDetectorFactory:E] add Joint to CdChar : RLEG_HIP_R  
[CollisionDetectorFactory:E] add Joint to CdChar : RLEG_HIP_Y  
[CollisionDetectorFactory:E] add Joint to CdChar : RLEG_KNEE  
[CollisionDetectorFactory:E] add Joint to CdChar : [CollisionDetectorFactory:E] WAIST  
[CollisionDetectorFactory:E] add Joint to CdChar : WAIST_P  
[CollisionDetectorFactory:E] add Joint to CdChar : WAIST_R  
[CollisionDetectorFactory:E] CdScene::addChar(box)
```

次は . . .

1. GrxUIの概要
2. サンプルプロジェクトの実行
3. プロジェクトの作成
4. その他の機能

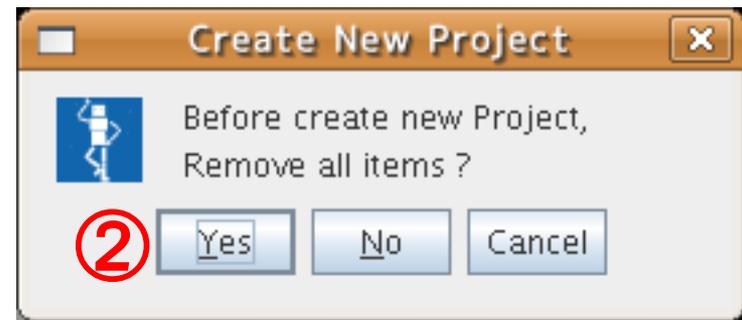
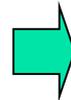
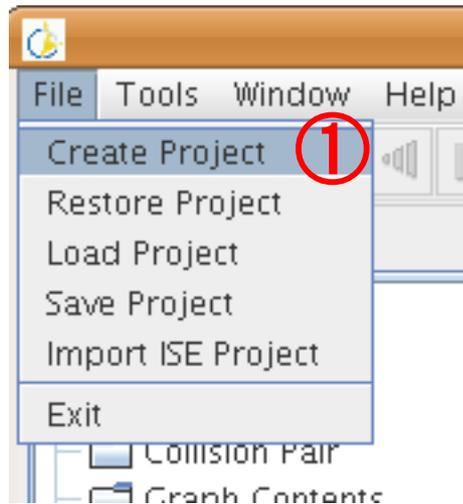


OpenHRP3/client/gui/project/SamplePD.xml
と同様のプロジェクトを作成を行います

3.プロジェクトの作成

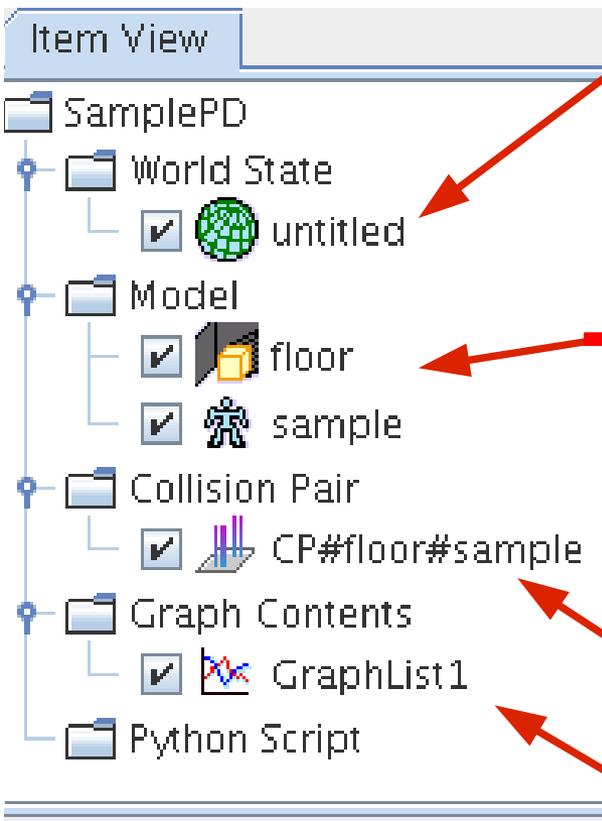
プロジェクトの生成

- ① File->Create Projectを選択
- ② 現れたダイアログでYesを選択し全アイテムを消去



3.プロジェクトの作成

手順



1.実行条件の設定

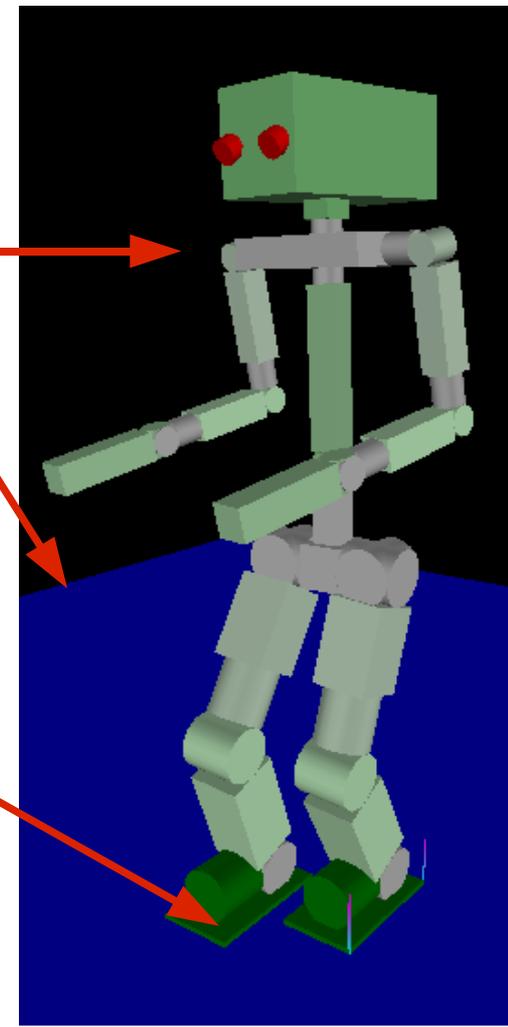
2.モデルのロード

3.初期姿勢設定

4.Controllerの設定

5.干渉チェック設定

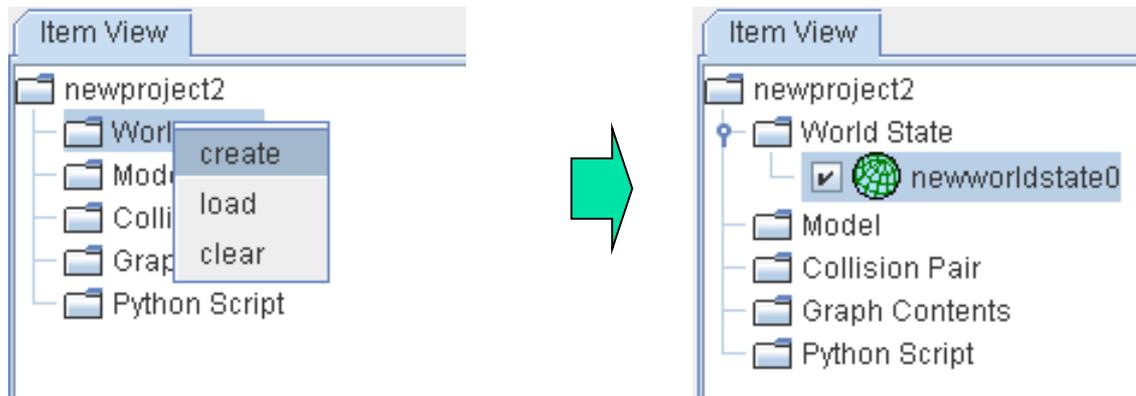
6.グラフの設定



3.プロジェクトの作成

実行条件の設定 1

World Stateの右クリックメニューからcreateを選択



WorldStateアイテムはシミュレーション実行条件をプロパティとして保持する

3.プロジェクトの作成

実行条件の設定 2

Simulation タブで設定

シミュレーション時間:13.4

積分タイムステップ: 0.001

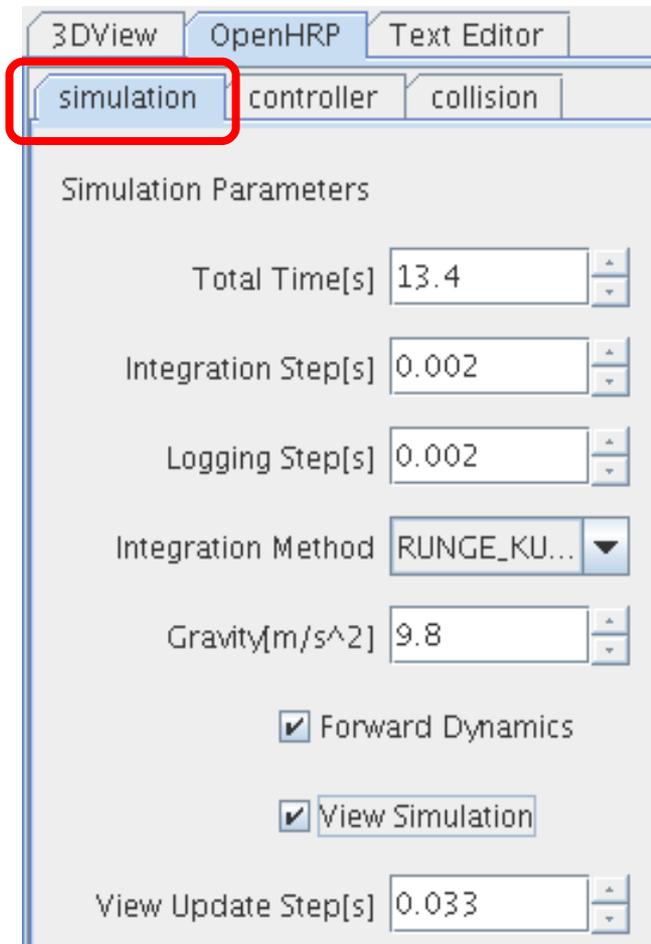
Logging周期 : 0.001

積分方法 : RUNGE_KUTTA

重力加速度 : 9.8

順動力学計算の有無: ON

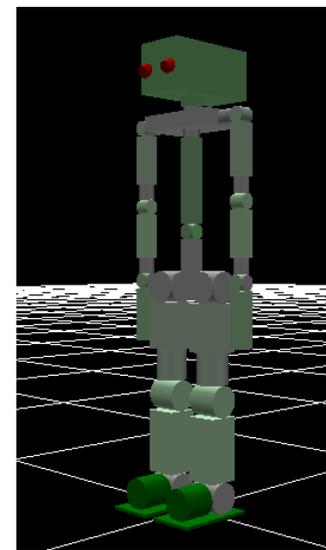
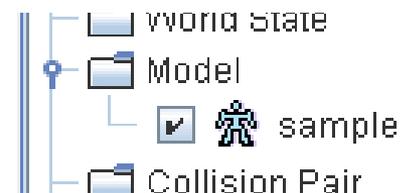
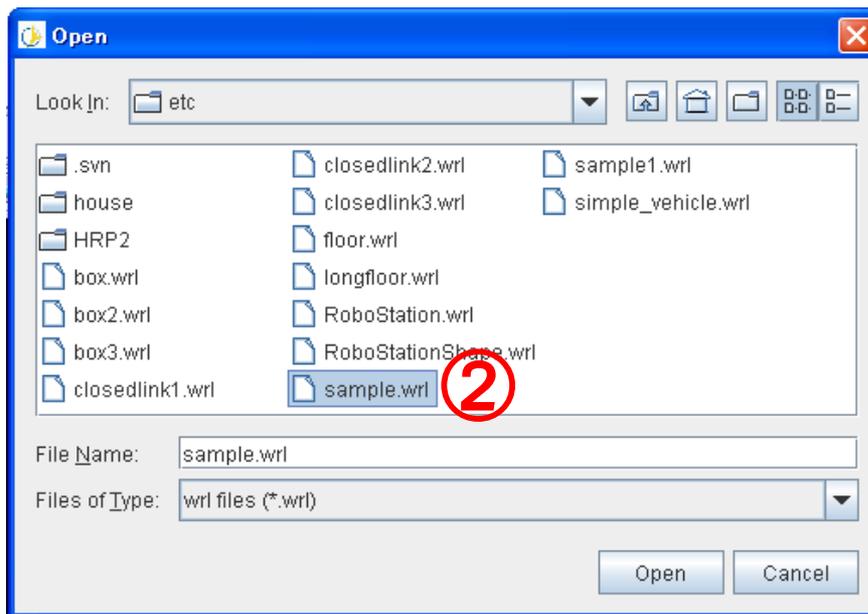
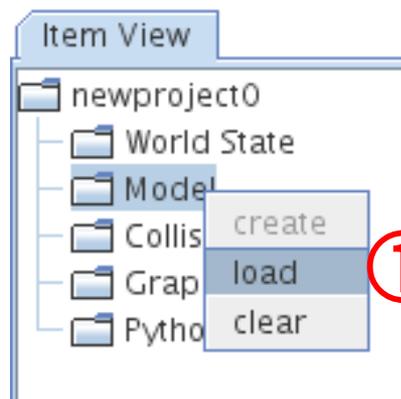
ViewSimulationの有無: ON



3.プロジェクトの作成

ロボットモデルのロード

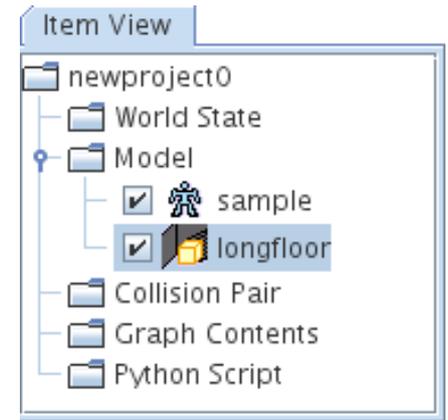
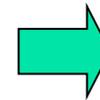
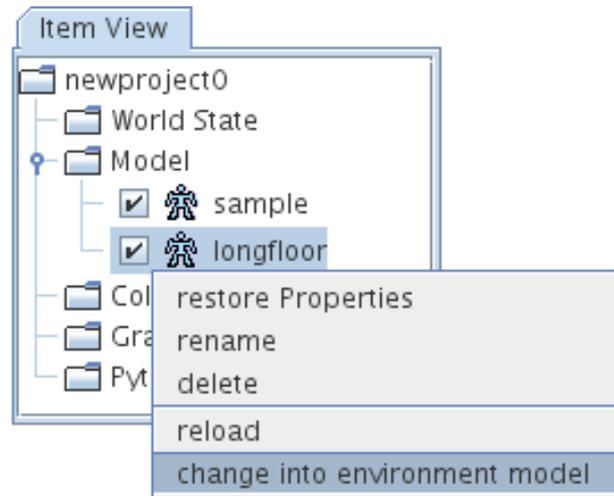
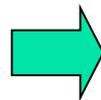
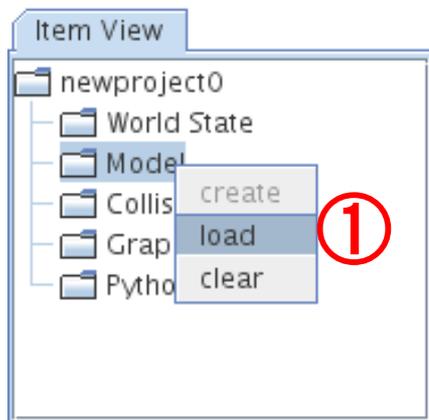
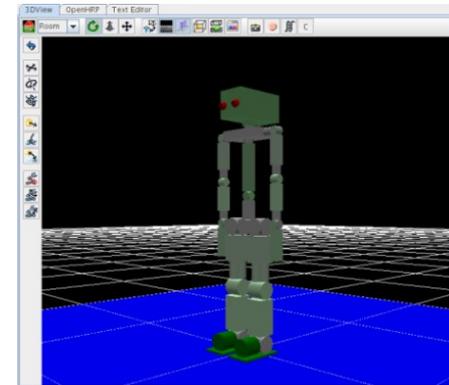
- ① “Model”の右クリックメニューのloadを選択
- ② OpenHRP3/etc/sample.wrlを読み込む



3.プロジェクトの作成

環境モデルのロード

- ① ロボットモデルと同様にlongfloor.wrlを読み込む
- ② “longfloor”の右クリックメニューから、**change into ... (環境モデルに設定)**を選択

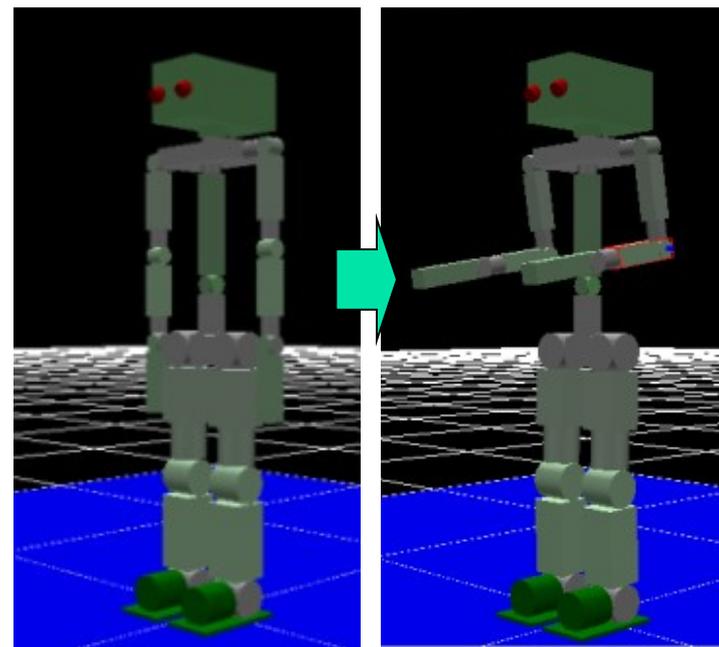


3.プロジェクトの作成

モデルの初期条件設定例

Property View

Name ▾	Value
isRobot	true
controller	PDcontroller
controlTime	0.0020
WAIST_R.angle	0.0
WAIST_P.angle	0.0
WAIST.translation	0.0 0.0 0.714



RARM_SHOULDER_P.angle	0.173	
RARM_ELBOW.angle	-1.57	(Radian)
LEFT_KNEE.angle	0.0705	

3.プロジェクトの作成

モデルの初期条件設定方法 1

Property Viewによる数値入力

WAIST.translation
モデル位置を設定
x[m] y[m] z[m]

WAIST.rotation
姿勢を以下4値で設定
x y z

θ [rad]
(回転軸ベクトル) (回転角度)

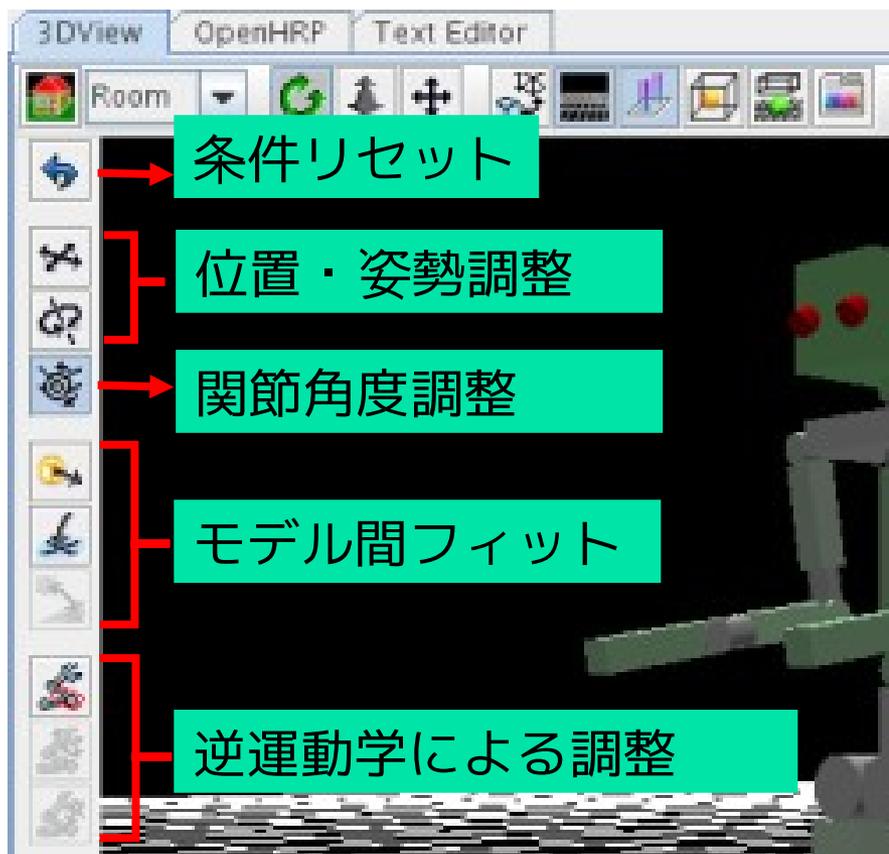
関節軸名.angle
関節角度[rad]を設定

Graph		Robot State		Property	
		Name ▾			
isRobot		true			
imageProcessor					
imageProcessTime		5			
controller		CHOROMETCont			
controlTime		0.001			
WAIST.translation		0.0 0.0 0.18995			
WAIST.rotation		0.0 1.0 0.0 0.0			
R_SHOULDER_Y.angle		0.0			
R_SHOULDER_R.angle		0.0			
R_SHOULDER_P.angle		0.0			
R_KNEE_P.angle		0.0			
R_HIP_Y.angle		0.0			
R_HIP_R.angle		0.0			
R_HIP_P.angle		0.0			
R_ELBOU_P.angle		0.0			
R_ANKLE_R.angle		0.0			
R_ANKLE_P.angle		0.0			

3.プロジェクトの作成

モデルの初期条件設定方法 2

3DView



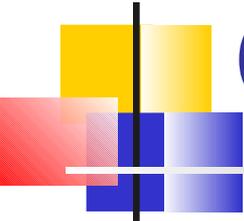
Robot State View

No	Joint	Angle	Target	Current	PWR	SRV	Pgain	Dgain
0	RLEG_HIP_R	0.0	---	---	---	---	---	---
1	RLEG_HIP_P	-2.1	---	---	---	---	---	---
2	RLEG_HIP_Y	0.0	---	---	---	---	---	---
3	RLEG_KNEE	4.5	---	---	---	---	---	---
4	RLEG_ANKLE_P	-2.4	---	---	---	---	---	---
5	RLEG_ANKLE_R	0.0	---	---	---	---	---	---
6	RARM_SHOULDER_P	10.0	---	---	---	---	---	---
7	RARM_SHOULDER_R	-0.2	---	---	---	---	---	---
8	RARM_SHOULDER_Y	0.0	---	---	---	---	---	---
9	RARM_ELBOW	-90.0	---	---	---	---	---	---
10								
11								
12								
13								
14	LLEG_HIP_P	-2.1	---	---	---	---	---	---
15	LLEG_HIP_Y	0.0	---	---	---	---	---	---
16	LLEG_KNEE	4.5	---	---	---	---	---	---
17	LLEG_ANKLE_P	-2.4	---	---	---	---	---	---
18	LLEG_ANKLE_R	0.0	---	---	---	---	---	---
19	LARM_SHOULDER_P	10.0	---	---	---	---	---	---
20	LARM_SHOULDER_R	-0.2	---	---	---	---	---	---
21	LARM_SHOULDER_Y	0.0	---	---	---	---	---	---
22	LARM_ELBOW	-90.0	---	---	---	---	---	---
23	LARM_WRIST_Y	0.0	---	---	---	---	---	---
24	LARM_WRIST_P	0.0	---	---	---	---	---	---
25	LARM_WRIST_R	0.0	---	---	---	---	---	---
26	WAIST_P	0.0	---	---	---	---	---	---
27	WAIST_R	0.0	---	---	---	---	---	---
28	CHEST	0.0	---	---	---	---	---	---

現在の関節角度を表示

3.プロジェクトの作成

Controllerについて



Dynamics Simulator

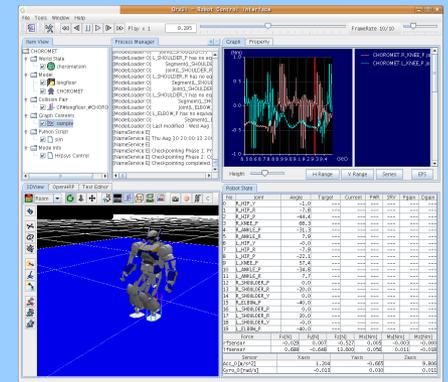
起動・停止
スケジューリング
可視化

GrxUI

仮想ロボットの
制御

Controller

起動・停止
スケジューリング



プロジェクトSamplePD.xmlの場合

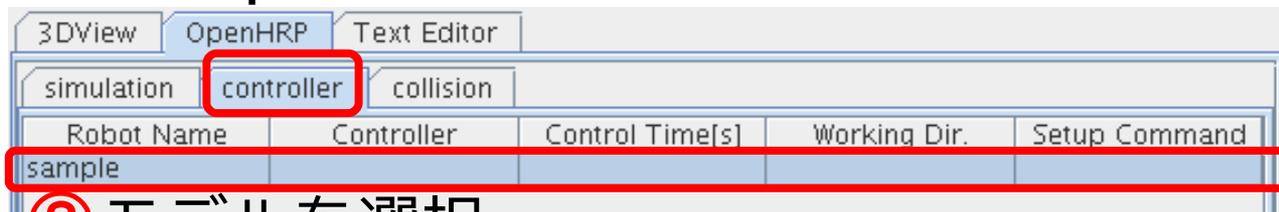
Path: OpenHRP/Controller/rtc/SamplePD/SamplePD.sh

CORBA ID: SamplePDController

3.プロジェクトの作成

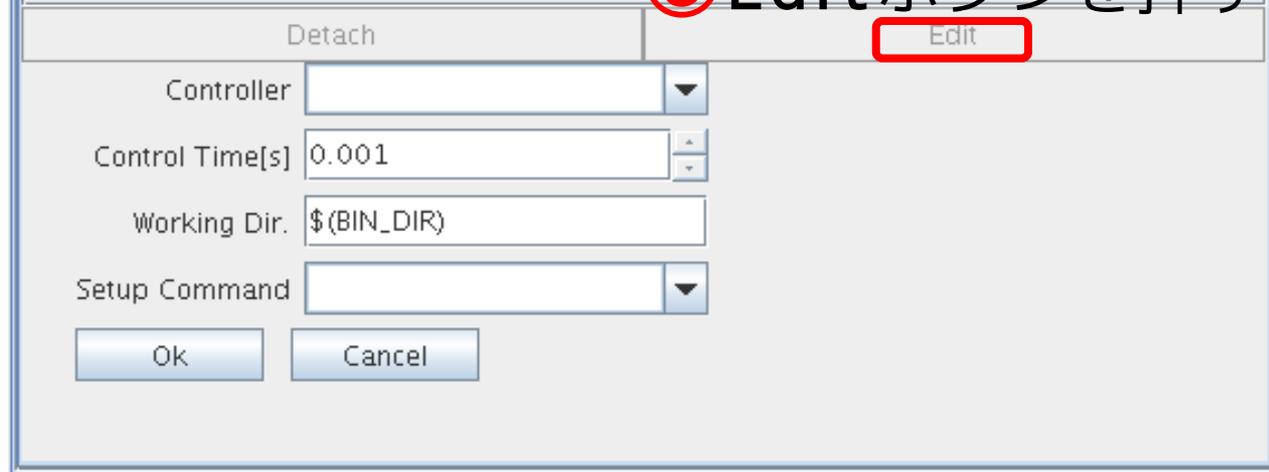
Controllerの設定 1

① OpenHRP->Controllerタブを選択



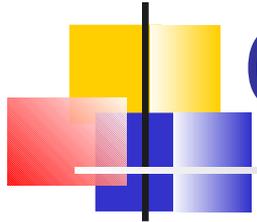
② モデルを選択

③ Editボタンを押す



3.プロジェクトの作成

Controllerの設定 2



Detach	
Controller	<input type="text" value=""/> ④
Control Time[s]	<input type="text" value="0.001"/> ⑤
Working Dir.	<input type="text" value="\$ (BIN_DIR)"/> ⑥
Setup Command	<input type="text" value=""/> ⑦

Ok Cancel

④ CORBA Idを指定
例：SamplePDController

⑤ 制御周期を指定
例：0.002

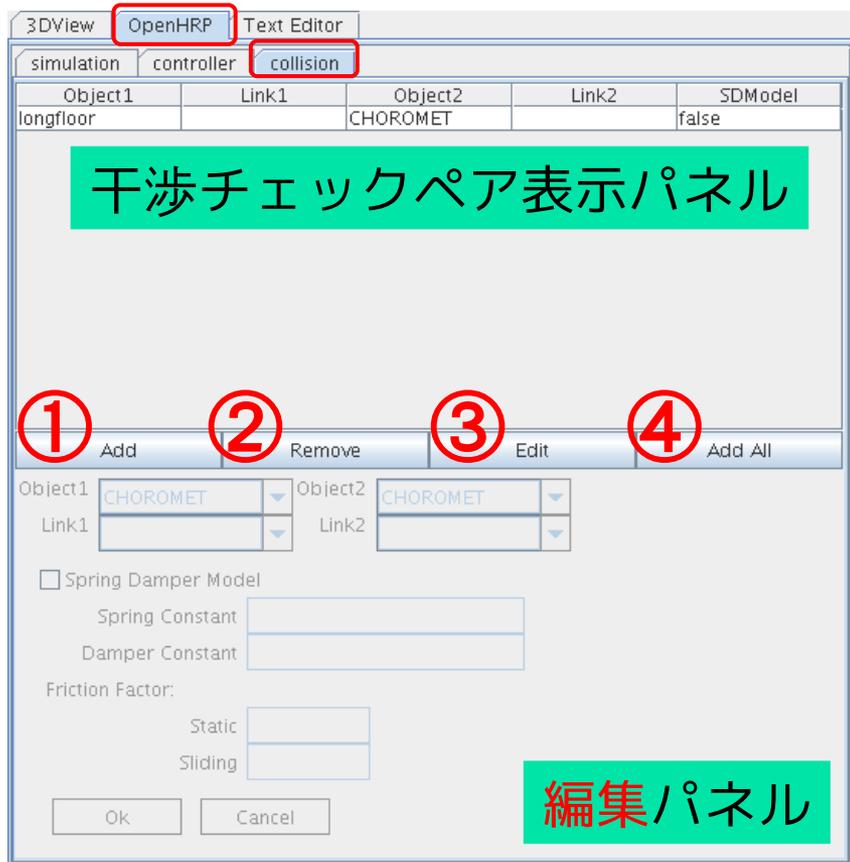
⑥ ディレクトリを指定
例：\$(OPENHRPHOME)/
Controller/rtc/SamplePD/

⑦ 起動コマンドを指定
例：SamplePD.sh

3.プロジェクトの作成

干渉チェックペア設定 1

OpenHRP->collisionタブ



- ① Add
干渉チェックペア追加
- ② Remove
選択中のペアを削除
- ③ Edit
干渉チェックペアを編集
- ④ Add All
選択中モデルで取りうる全ペアを追加

3.プロジェクトの作成

干渉チェックペア設定 2

干渉チェックペアの編集方法

① Object1/2

- ・モデルを選択
- ・Addボタンで新規作成した場合のみ変更可能

② Link1/2

- ・Linkを選択
- ・省くと全リンクが対象

③ Friction Factor

- ・摩擦係数を設定

Object1: longfloor ①

Object2: sample

Link1: ②

Link2:

Spring Damper Model

Spring Constant: 0 0 0 0 0 0

Damper Constant: 0 0 0 0 0 0

Friction Factor:

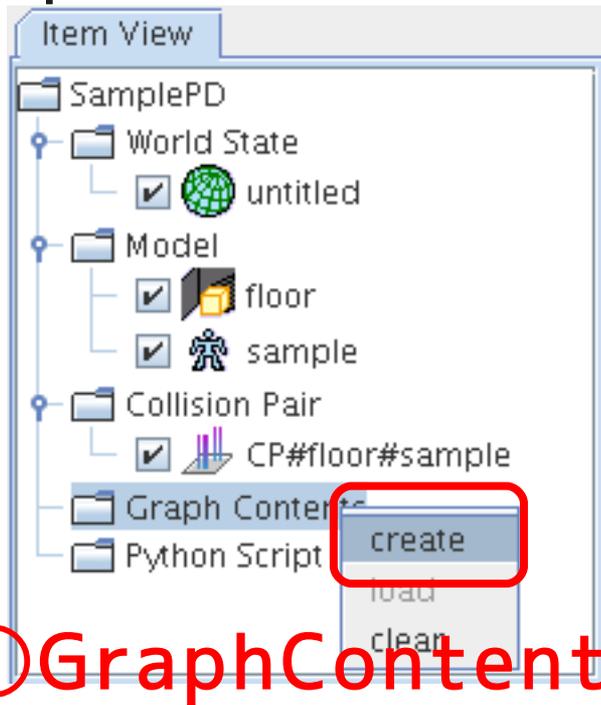
Static: 0.5 ③

Sliding: 0.5

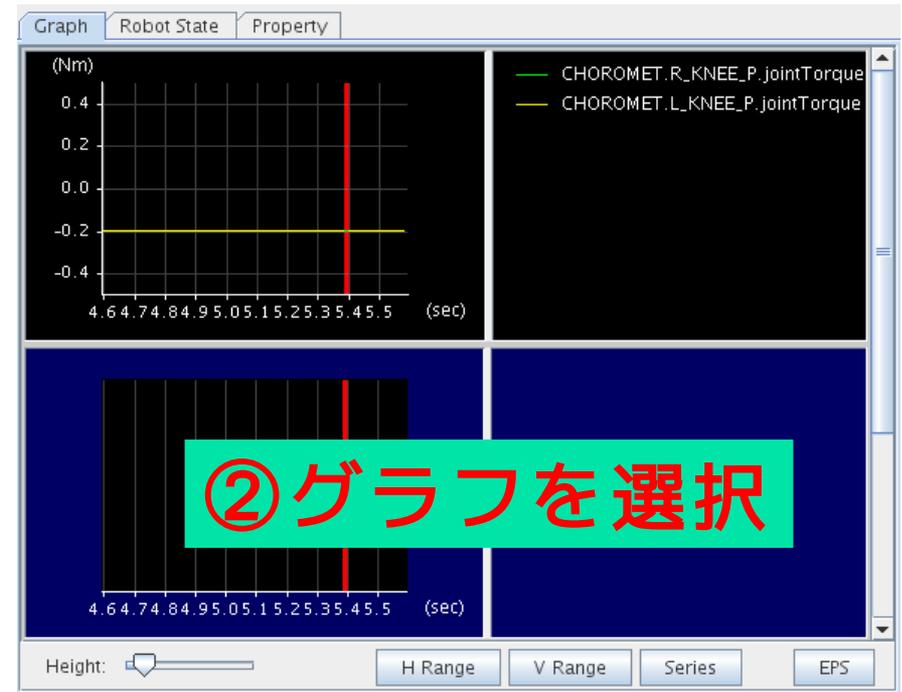
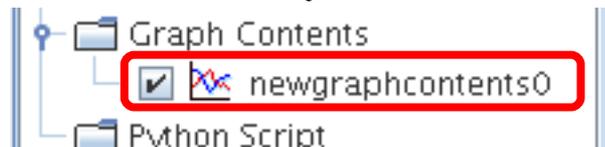
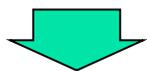
編集パネル

3.プロジェクトの作成

Graphコンテンツ設定 1



① GraphContents
を生成

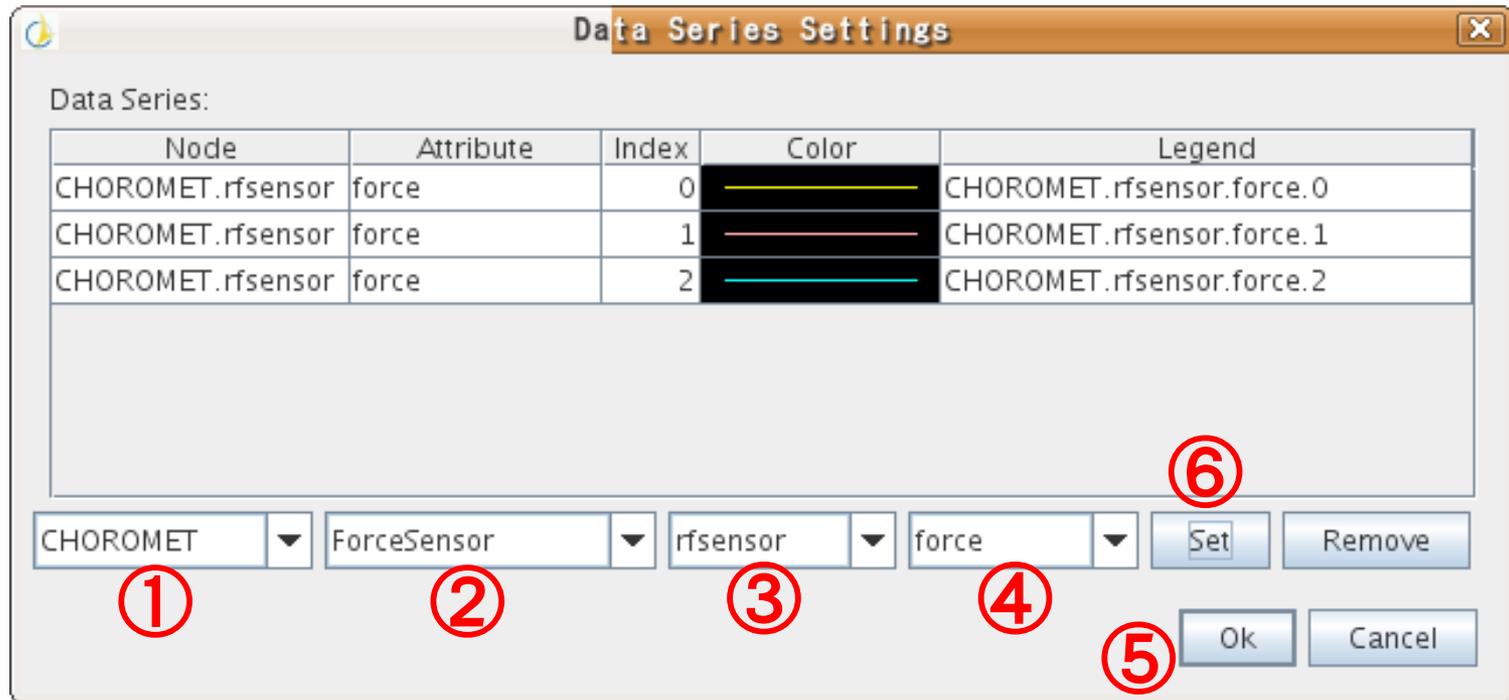


③ Seriesボタンを押す

3.プロジェクトの作成

Graphコンテンツ設定 2

- ① ロボット名選択
- ② センサ種類選択
- ③ ノード名選択
- ④ 項目選択
- ⑤ Setボタンを押す
- ⑥ OKボタンを押す



3.プロジェクトの作成

Graphコンテンツ設定 3

- ① グラフの高さ調整
- ② 定義域の設定
- ③ 赤線マーカ位置の設定
- ④ 値域の設定
- ⑤ 画像ファイル出力



Horizontal Range

Horizontal Range: (sec) ②

Marker Position: ③

Ok Cancel

Vertical Range

From: ④

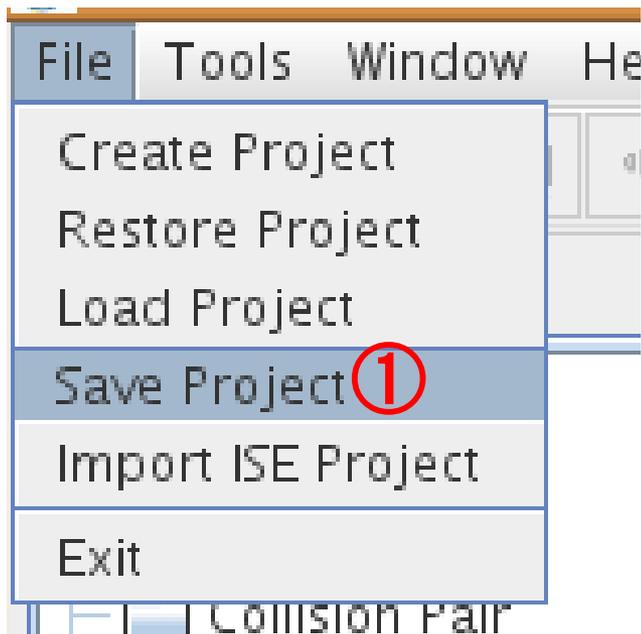
To:

Ok Cancel

3.プロジェクトの作成

プロジェクトの保存

- ① File->Save Project を選択し“SampleTest”として保存
- ② Save Window Config.ダイアログ
Yes : 画面構成も保存する No : 画面構成は保存しない
Cancel : プロジェクト自体保存をしない



3.プロジェクトの作成

プロジェクトファイルの構成

```
<mode name="OpenHRP3">
```

```
  <item class="クラス名" name="表示名" url="URL" select="true">
    <property name="項目名" value="値"/>
  </item>
```

...

Itemの設定

```
  <view class="クラス名" name="表示名">
    <property name="項目名" value="値"/>
  </view>
```

...

Viewの設定

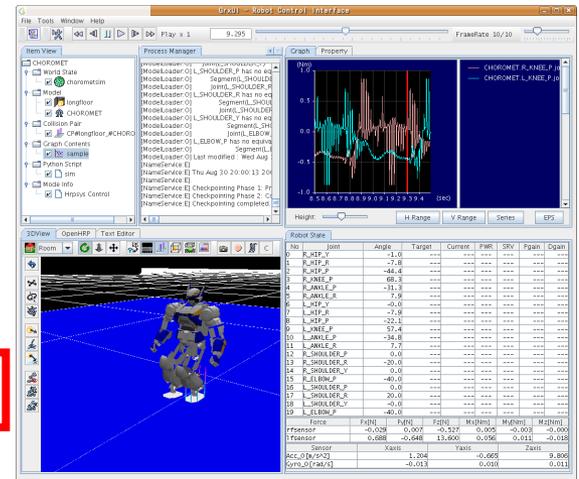
```
  <windowconfig>
    . . .
  </windowconfig>
```

画面の構成

```
</mode>
```

次は . . .

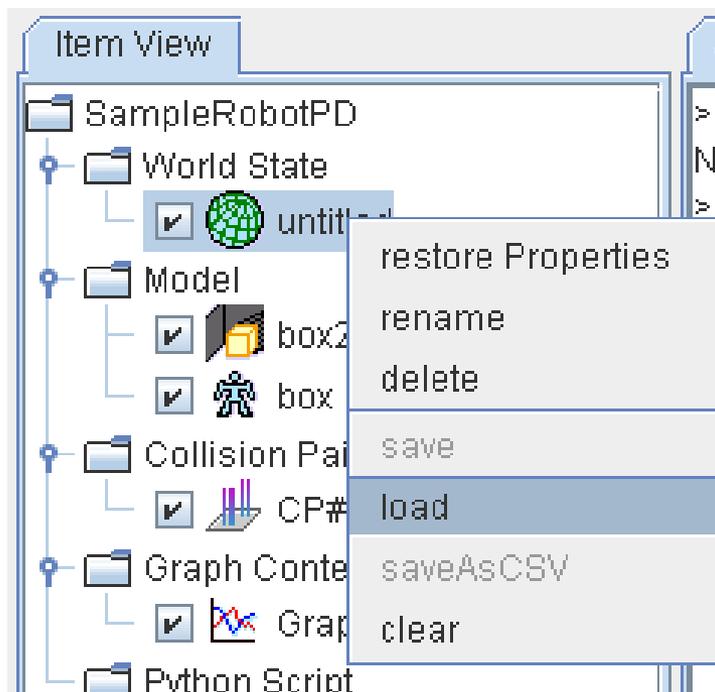
1. GrxUIの概要
2. サンプルプロジェクトの実行
3. プロジェクトの作成
4. その他の機能



4.その他の機能

実行結果のSave/Load機能

World State->untitledの右クリックメニューから
結果のSave/Loadを行う



client/gui/log/untitled.logとして保存

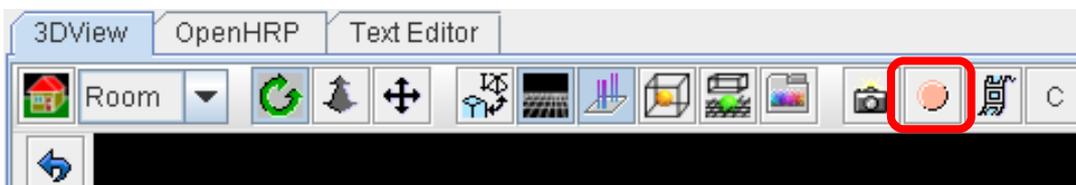
4. その他の機能

動画の保存機能

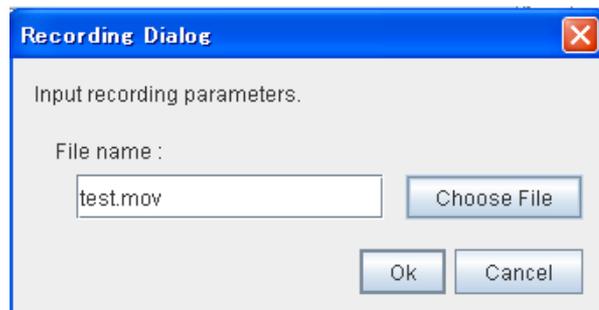
① 33に設定



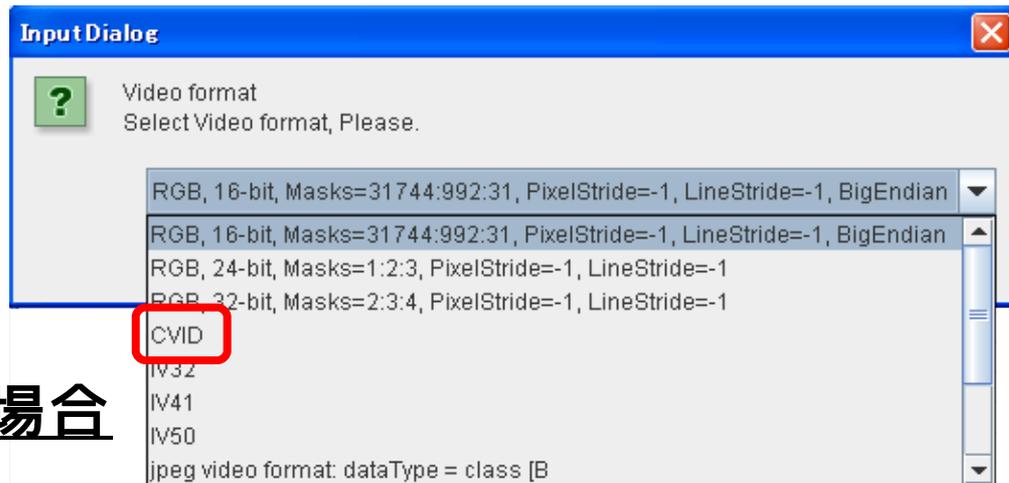
② 録画ボタンを押す



③ 保存先の設定



④ 保存形式の設定



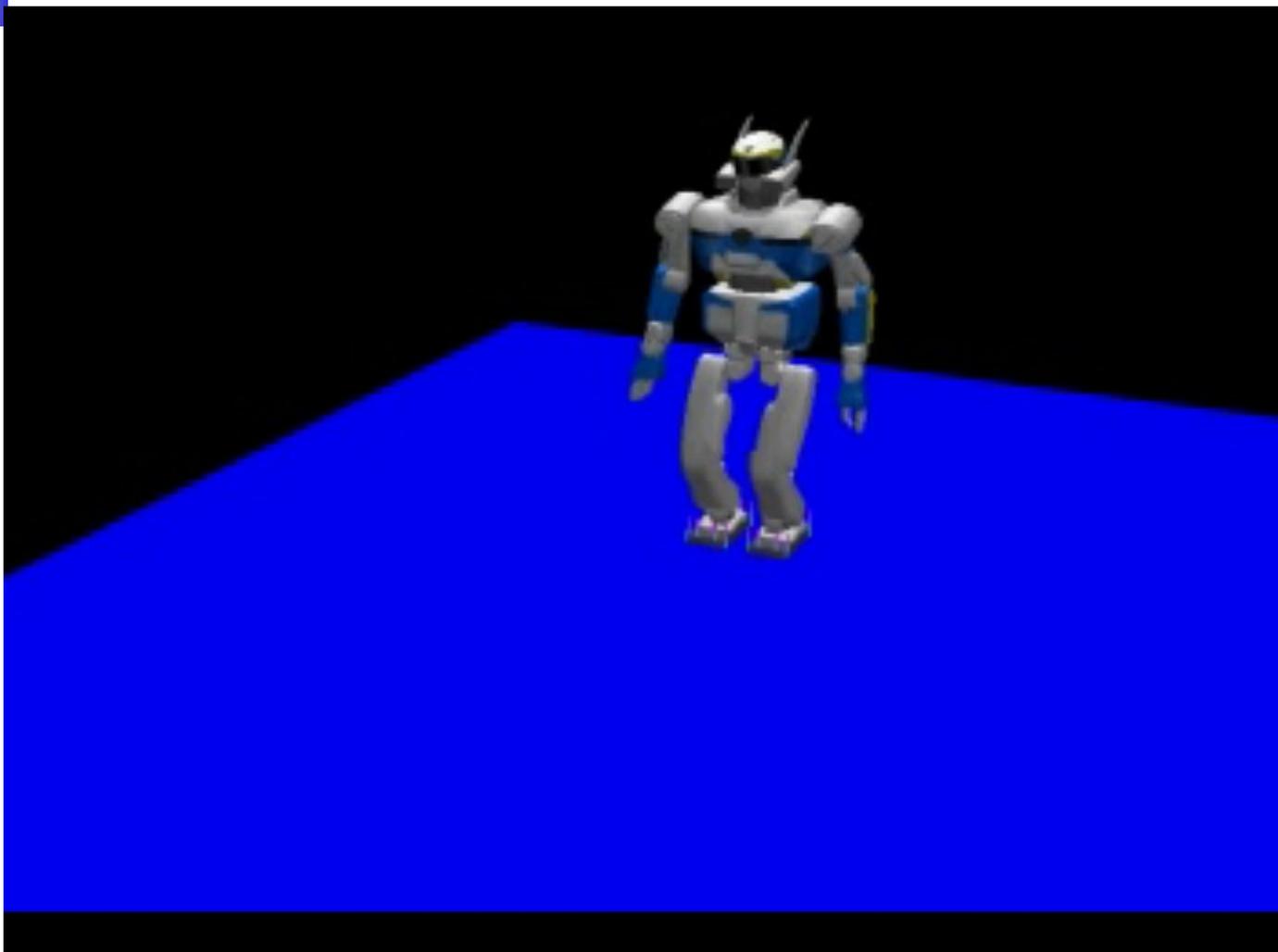
PowerPointに貼りつける場合

Windows : CVIDを選択

Linux : デフォルト設定で、TMpegEnc+QTReaderでmpegに変換

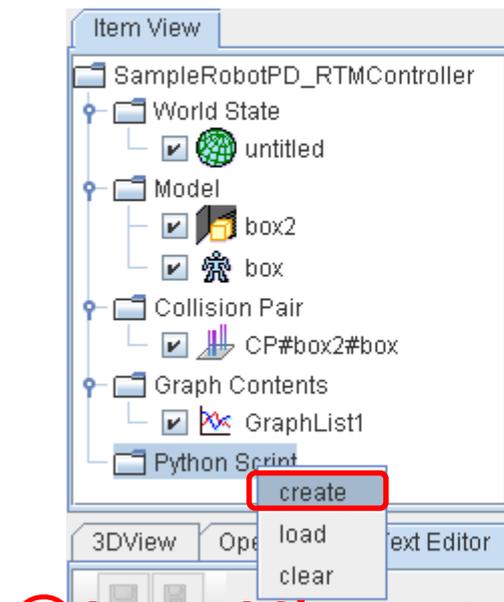
4. その他の機能

動画保存のサンプル

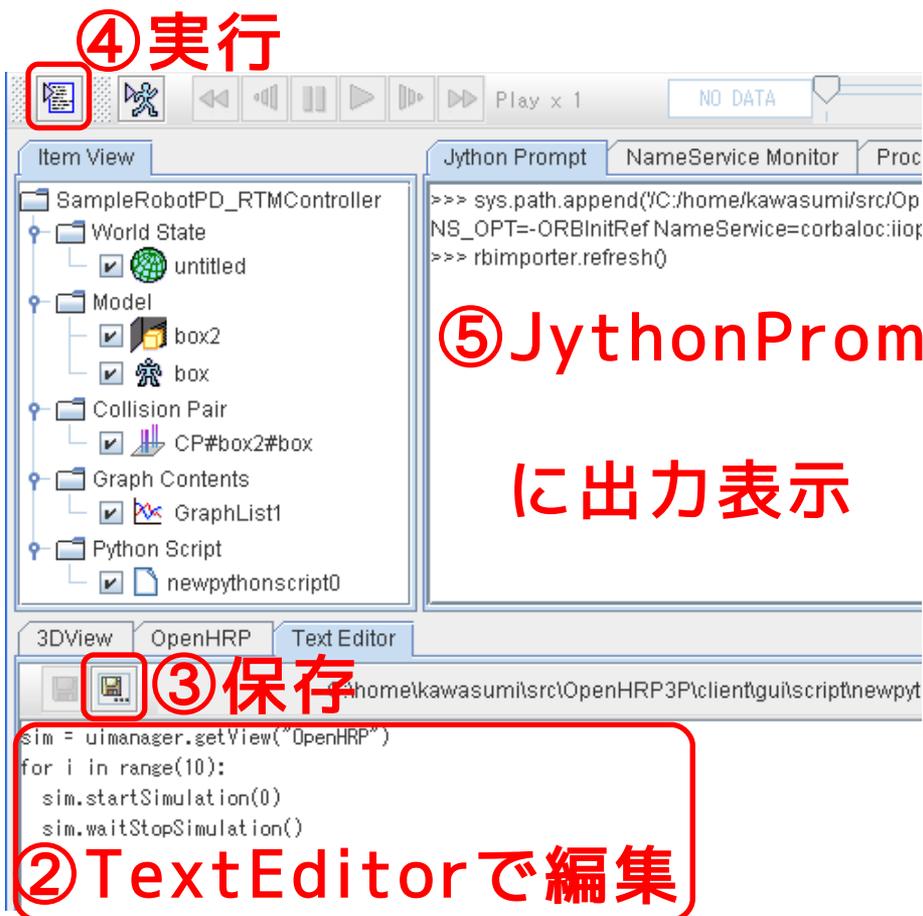


4. その他の機能

スクリプトの実行方法



① ItemViewで
スクリプト生成



④ 実行

⑤ JythonPrompt
に出力表示

③ 保存

② TextEditorで編集

4. その他の機能

スクリプト例

スクリプトから全てのItemとViewにアクセス可能

<例> シミュレーションの繰り返し実行

```
sim = uimanager.getView("OpenHRP")
for i in range(10):
    ...設定変更...
    sim.startSimulation(0)
    sim.waitStopSimulation()
```

i=0 : interactive
i=1 : not interactive

4. その他の機能

スクリプト例(Swingの利用)

```
from javax.swing import *
```

```
def act(evt):  
    print "punch!!!"
```

```
f = JFrame()
```

```
c = f.getContentPane()
```

```
c.add( JButton("Punch", actionPerformed=act) )
```

```
f.setSize(200,200)
```

```
f.setVisible(1)
```



4. その他の機能

スクリプトによるロボット駆動例1

```
Def punch():  
    seq.sendMsg(":joint-angles 全関節角度 時間")  
    seq.sendMsg(":joint-angles 全関節角度 時間")  
    . . .  
punch(seq)
```

