Choreonoid as a Software Framework for Implementing Graphical Robotics Applications

Shin’ichiro Nakaoka
AIST
Demonstration(2)
Choreonoid Framework Overview

A framework which covers GUIs and visualizations of robotics applications

High-performance monolithic (single process) structure all written in C++

Additional functions and their cooperation with existing functions can be flexibly implemented as plugins
Basic Design of the framework

Items and item tree structure
Views and toolbars with flexible layout system
MVC-like architecture with signal-slot mechanism
class SamplePlugin : public Plugin {
public:
    SamplePlugin( ): Plugin(”Sample”) { depend(”Body”); }
    virtual bool initialize() {
        ToolBar* bar = new ToolBar(”Sample1”);
        ToolBar::Increment -> addButton(”Increment“)
        -> sigClicked().connect(bind(&onButtonClicked, +0.04));
        ToolBar::Decrement -> addButton(”Decrement“)
        -> sigClicked().connect(bind(&onButtonClicked, -0.04));
        addToolBar(bar);
        return true;
    }
};

void onButtonClicked(double dq){
    ItemList<BodyItem> bodyItems =
    ItemTree::mainInstance()->selectedItems<BodyItem>();
    for(size_t i=0; i < bodyItems.size(); ++i){
        BodyPtr body = bodyItems[i]->body();
        for(int j=0; j < body->numJoints(); ++j){
            body->joint(j)->q += dq;
        }
        bodyItems[i]->notifyKinematicStateChange(true);
    }
}
Possible Applications

Simulators
Motion planning tools
Scenario scripting tools
Model design tools
Robot operation tools
Future Plan

We will soon open the official website www.coreonoid.org

We will release source / binary packages with an open-source license