Strategies for Using a Simulation in the Development of the Bremen Autonomous Wheelchair

Thomas Röfer

Bremen Institute for Safe and Secure Systems Center for Computing Technology

University of Bremen
Contents

**SimRobot**
- Platforms
- Simulated Objects
- Sensors
- Creating a Simulation

**Architecture**
- Synchronous / Asynchronous
- C++ Derivation Hierarchy
- Flow of Information

**Methodology**
- Evolution of Simulation
- Example

**Outlook**
SimRobot - Simulated Objects

**Bodies**
- polygons

**Emitters**
- radial
- spot

**Actuators**
- rotational joints
- translational joints
- objects w. 6 DOF
- vehicles w. steering+driving axle
SimRobot - Sensors

Intensities of Radiation
- Tactile
- Camera
- Facette

Camera ↔ Facette
SimRobot - Sensors

Intensities of Radiation
- Tactile
- Camera
- Facette

Distances
- Whisker
- Whiskerfield
- Camera
- Facette
- Ultrasonic

Noise

Collision-Free Execution
SimRobot - Creating a Simulation

- **GUI**
- **SimRobot**
- **Controller.cpp**
- **Scene.scn**
- **C++ compiler/linker**
- **SimRobXX.exe**
- **Simulation**
Architecture - Synchronous / Asynchronous
Thread 1
- SimRobot
- Controller

Thread 2
- CanBus
- Wheelchair

Thread 3
- FrameGrabber
- Marks

SimChair
- Scanner
- MarkMap
- RouteNavigation
Architecture - Flow of Information

**Notification**
- SimChair
- Odometry
- StopSensor
- AdaptiveSteering
- BasicBehaviors
- Scanner
- MarkMap
- RouteNavigation

**Modification**
- SimChair
- Odometry
- StopSensor
- AdaptiveSteering
- BasicBehaviors
- Scanner
- MarkMap
- RouteNavigation

(setPosition) -> setSpeed
Evolution of the Simulation

Simulation V 1.0
Controller V 1.0

Simulation V 1.1
Controller V 1.1

Simulation V 1.1
Controller V 2.0

Reality
Controller V 1.0

Reality
Controller V 1.1

Reality
Controller V 2.0
Driving through a Door - Simulation

1.42 m

∞ m
Outlook

New Wheelchair „Rolland“
- Synchronous simulation of 32 ms steps