Distributed Robotic Architecture

Using Actin and DDS

Justin Keesling

Energid Technologies
www.energid.com
Overview

• Who is Energid?
• What is Actin?
• Actin with DDS
• Tasking with DDS
• Projects using DDS
• Live demo
Energid Technologies

- Develops software for simulation and control of any robotic system

- Actin software is at the core of our business
  - Actin SDK
  - Actin applications
  - Integration services

- Now a Teradyne company

- Cool videos: https://vimeo.com/energid
Actin Software

- General kinematics and dynamics model
  - Inverse and forward kinematics and dynamics
  - Kinematically redundant mechanisms
  - Fixed and mobile base manipulators
- General motion constraint and optimization framework
  - Dynamic collision avoidance
  - Joint limit and singularity avoidance
  - Strength optimization
  - Dynamic response to sensor data
- Adaptive tasking
  - Global path planning
  - Complex tool path motion control (EcScript)
  - Coordination of many robots and axes (Manipulation Director)
- General platform support
  - Easy integration with sensors and actuators
  - Kinematic model generation from CAD
  - Desktop applications for Windows, Linux, OS X
  - Real-time control on VxWorks, RT Linux, RTOS32, RTX64
Constraint Optimization

• End effector constraints
• Collision avoidance
• Joint limit avoidance
• Singularity avoidance
• Center of gravity

• Video
  • 3 HEBI Actuator Arms
  • 21 DOF Total
  • Frame end effector
Multi-Robot Coordination

• 5 Machines
  • Robot
  • Pipe Handler
  • Roughneck
  • Elevator/Lift
  • Slips

• Manipulation Director
  • Hierarchical tasking
  • Machine coordination
Collision Avoidance

- Collision exclusion maps
  - Self collision exclusion candidates
  - Static collision exclusion candidates
  - Dynamic collision exclusion candidates

- Video
  - UR5 (6-DOF)
  - Adaptive tasking
  - Dynamic collision avoidance
Kinematically Redundant Mechanisms

- Extra degrees of freedom allows optimizing for other constraints

- Video
  - 2 Theoretical 36-DOF “wedge actuator” arms
Global Path Planning

• Fanuc M-10iA/12 6 DOF arm

• Added turntable DOF

• Complex part inspection
  • 41 inspection points
  • Travelling salesman problem

• Additional constraints
  • Target has 36 LED emitters
  • Camera bar has 3 cameras
  • At least 4 LEDs visible to all 3 cameras
Real Time

• Video
  • HM Elfin 5 Robot (6-DOF)
  • Added linear rail
  • EcScript motion control
  • 1 ms Updates
Actin with DDS

• Started using DDS early 2013
  • Government customer required us to integrate Actin with their tools using DDS
  • Early RDS work indicated DDS would be a perfect fit

• Single-robot deployments
  • Increased demand for teleoperation

• Multi-robot deployments
  • One robot controller machine per robot
  • Communication between controllers
    • Commands, state synchronization, sensor feedback, hardware status
  • Scalable architecture
DDS Enables

• Flexible system architectures
  • Add new sensors, actuators, etc
  • Swap out components
  • Scalability
  • Redundancy

• Simplified communication between teams
  • Send the IDL, topic names, and QOS

• High degree of tunability through QOS

• Reduced development time!
Distributed Robot Control Architecture

Robot Controller
- DDS Common Layer
- DDS Tasking Layer
- DDS System State Layer
- DDS Sensor Layer
- Hardware Layer

Control Master
- DDS Common Layer
- DDS Tasking Layer
- DDS System State Layer
- DDS Sensor Layer
- Hardware Layer

Robot Controller
- DDS Common Layer
- DDS Tasking Layer
- DDS System State Layer
- DDS Sensor Layer
- Hardware Layer

tasks

robot states, task status, sensor values

pub

sub

© 2018 Energid Technologies Corporation. All rights reserved.
Tasking With DDS

- Manipulation Director

- DDS Tasking Implementation
  - Fundamentals
  - Topics
  - Sequence Diagram
  - Implementation Specifics

- Hierarchical Tasking
Manipulation Director

- Library of manipulation tasks
  - Low-level blocks used to build higher-level blocks
  - All blocks are reusable
  - Write to and read from XML
  - Extensible architecture

- Theater terminology
  - Director - directs the execution of a script
  - Cast - Assigns actors (robots) to play roles in the script
  - Scripts - Composed of scenes
  - Scenes - Composed of directions for the actors
  - Directions
    - Stage Directions - Non “speaking” instructions to actors
    - Manipulation Directions - The “speaking” lines … For a robot, this means “movement”
  - Poses
    - Reusable transformations
    - Can be fixed or relative
    - Used inside directions for defining positions and orientations
  - Cues

At some point, we lost the “plot” (of theater terminology)
Manipulation Director GUI
Tasking Fundamentals

• Tasking provider “provides” a resource to be tasked
• Tasking requester “requests” control of a resource
• Only one requester can have control of provider at a time
• Control is granted or revoked by the provider
• Control takes into account
  • Machine state (Manual vs. Auto)
  • Continuity of control
• Purpose
  • Manage objects in the manipulation director library
  • Manage task execution of manipulation director objects
• Not so “secret” sauce
  • Multiple keys
  • Content filtered topics
Tasking Topics

- TaskingControlRequest
  - Request control of a particular role
  - Keyed by role and UUID of the requester
- TaskingControlSessionStart
  - Informs a provider when control has been granted
  - When the instance is no longer alive, the requester no longer has control
  - Keyed by role and UUID of the requester
- TaskingControlCommand
  - Send a command to the role
  - Commands manage manipulation director library and execution
  - Keyed by role and UUID of the requester
- TaskingControlResponse
  - Send a response from the role to the requester
  - Responses indicate the success or failure of commands
  - Keyed by role and UUID of the requester
Tasking Sequence Diagram
Tasking Implementation Specifics

- **Tasking Requester**
  - Generates UUID
  - Publishes
    - TaskingControlRequest uuid=UUID
    - TaskingControlCommand uuid=UUID
  - Subscribes
    - TaskingControlSessionStart (role = 'roleName' AND uuid = 'requesterUuid')
    - TaskingControlResponse (role = 'roleName' AND uuid = 'requesterUuid')

- **Tasking Provider**
  - Starts tasking control session to grant control
  - Stops tasking control session to revoke control
  - Subscribes
    - TaskingControlRequest (role = 'roleName')

- **Tasking Control Session**
  - Direct connection to requester UUID
  - Publishes
    - TaskingControlSessionStart uuid=UUID
    - TaskingControlResponse uuid=UUID
  - Subscribes
    - TaskingControlCommand (role = 'roleName' AND uuid = 'requesterUuid')
Hierarchical Tasking

- Lower-level tasking provider
  - Provides lower-level manipulation director library
  - Single machine tasks
  - Single machine does not mean single role

- Higher-level tasking provider
  - Requests control of lower-level tasking providers
  - Provides higher-level manipulation director library
  - Direction sends tasking commands to lower-level providers
  - Multi-machine tasks

- How many levels are needed?
Projects Using DDS

- CANRIG Robotics
- URCaps Integration
CANRIG Robotics

- Formerly Robotic Drilling Systems
  - Acquired by NABORS in 2017
- Manual operations dominate global rig fleet
  - Remove people from the rig floor
  - Save lives
  - Save time
- Tripping is the biggest time consumer
- CANRIG Robots
  - Next-generation
  - Fully electric
  - Maintenance free
  - Design life = 10 years
  - Beautiful design!
URCaps Integration

- 6 DOF UR3, UR5, or UR10
- Compute node separate from robot controller
  - Runs Actin
  - Provides 3D rendered images
Live Demo

- User interface process (Actin Viewer)
- Robot simulation process (Actin Viewer)
- User interface process 2 (Actin Viewer)