Challenges in the software architecture design for autonomous legged robots

Marco Frigerio
Istituto Italiano di Tecnologia – Department of advanced robotics

May 6th, 2013
Workshop on Software Development and Integration in Robotics (SDIR VIII)
ICRA 2013 – Karlsruhe, Germany
Our HyQ robot

HyQ -
IIT's Hydraulic Quadruped Robot

C. Semini, J. Buchli, T. Boaventura,
M. Frigerio, M. Focchi, D. Caldwell

September 2011

C. Semini, J. Buchli, T. Boaventura, M. Frigerio, M. Focchi, D. G. Caldwell
Objectives of the project

- Build a *versatile* robot
  - Fast, dynamic motions
  - Careful navigation over rough terrain
- Build an *autonomous* robot
  - Power
  - Intelligence
- Extremely challenging
  - Hardware & Software
HyQ – Other behaviours

Software challenges

• The need for speed & guarantees
  – **Real-time** constraints at high frequencies (e.g. actuators control)
  – Hardware limitations
  – Limited execution times (e.g. online planning)
  – Simulations

• Reliability
  – 1ms control loop running for half an hour
  – Related to safety!

• Multi-domain problem
  – Control systems, rigid body dynamics, artificial vision, planning,...
Our control system

kinematics and dynamics

- User space software
- Motor control
  - sensors data
  - joints desired position/force
- Robot behavior
  - actuators command
- Robot Hardware I/O
- Operating system
  - RealTime OS
  - I/O boards driver
- Hardware
  - CPU board
  - Data I/O boards
  - sensors/actuators

- SDIR VIII @ ICRA, May 6th 2013
- Marco Frigerio - iit
Hardware I/O

- Hardware Abstraction Layer (HAL)
- Decouples hardware details from client code
  - e.g. changing hydraulic valves does not affect the controller
Motor control

- Autonomous process
- Low-level actuators control
- Hard real-time at “high” frequency (eg 1KHz)

Inverse dynamics control
Trajectory generation

- Autonomous process, hard real-time (e.g. 250Hz)
- Joints trajectories
- Robot kinematics
The SL software package

- An implementation of the motor control and the trajectory generation blocks
- Can control a real robot or run against a simulator

The SL software package

Limitations

- Single access point to sensors
- Deployment on a single machine
- “Trajectory generation” is reductive
Future developments

- A more flexible software architecture
  - Computational activities
  - Information sources (ie different sensors) and flows
  - Frequencies, complexities, etc.

- Locomotion
  - World model
  - Robot model
  - ...

♦ Bernini & Tisato, “Explaining Architectural Choices to Non-architects”, ECSA 2010
Example – activities

- Vision sensors sampling
- Inertial sensors sampling
- Position/force sensors sampling
- Actuators control
- Body-state estimation
- Feet-state estimation
- World mapping
- Foot-steps planner
- Joint trajectories generator
- World mapping
- Body-state estimation
- Feet-state estimation
- Actuators control
- Vision sensors sampling
- Position/force sensors sampling
Implementation technologies

- The architectural plan enables to catch technological opportunities
- Sound technologies are crucial
- For example
  - Code generation for kinematics and dynamics
Model-based code generation

- Kinematics and dynamics are fundamental
- Known in theory, hard to implement
- Code generation
  - Is flexible
  - Gives reliable and efficient code

- M. Frigerio et al., “A Domain Specific Language for kinematic models and fast implementations of robot dynamics algorithms”, DSLRob 2011
Websites

• HyQ
  - http://www.iit.it/hyq
  - Look also in youtube.com

• Robotics code generator
  - Follow the link from HyQ page, or: