Case for Energy Efficient Robots

![Graph showing the relationship between estimated fraction of total energy spent in compute and rated mission length in minutes. The graph includes points for various robots such as Turtlebot 2, Slocum Glider, PR2, Aldebaran NAO, Clearpath Pelican, 3DR IRIS, Asce Tec Pelican, and 3DR Aero.]
Computation on a Robot

![Bar chart showing computation load on a robot, with categories like nodelet, self_filter, python, collider_node, environment_server, 83-lan1, pthread, XnSensorServer, move_arm_head_monitor, pr2_marker_control, libbz2, and Uncategorized. The chart contrasts simulated PR2 with physical PR2.]
• HCDC Workloads
  • What can we run on the HCDC chip, now and future

• HCDC Programming Language
  • Accessing the HCDC through annotated C code

• HCDC Compiler
  • How HCDC configurations are generated

• HCDC API Library
  • Lower level interface to HCDC functions
HCDC Workloads

- HCDC & computational robotics
  - Speeding up modelling forward dynamics

- HCDC & smooth optimization
  - Solving constrained and unconstrained quadratic programming with gradient descent

- HCDC & solving linear equations
  - Inverting matrices using gradient descent on analog circuits

- HCDC & solving ordinary differential equations
  - Simulating nonlinear ODEs of up to fourth order
int main (int argc, char** argv) {

    // Outputs
    double a, b;

    #pragma HCDC_begin y' = y*y + y, y(0)=5.0, y(10)=-?
    #pragma HCDC_map y a
    #pragma HCDC_map y' b
    #pragma HCDC_scale 10.0

    //--- Optional digital code here ---/
    #pragma HCDC_end

    return 0;
}

Job of HCDC Compiler

$$\Theta = -0.5 * \Theta - 5 * \sin \Theta, \Theta(0) = 1$$
Job of HCDC Compiler

- DAC
- ADC
- Analog Input
- Analog Output
- \( \sin(x) \)
```c
34  hcdcInit();
35
36  // Set initial integrator values
37  float initial_y0 = 5.000000;
38
39  // Call HCDC wiring instructions
40  setSimpleConn ( {fans[0], out0, muls[0], in0} );
41  setSimpleConn ( {fans[0], out1, muls[0], in1} );
42  setSimpleConn ( {fans[0], out2, ints[0], in0} );
43
44  setSimpleConn ( {muls[0], out0, ints[0], in1} );
45
46  setIntInitial ( {ints[0], initial_y0} );
47  setSimpleConn ( {ints[0], out0, fans[0], in0} );
```
Some Advance Warning!

• We have a working example of converting damped oscillator differential equation -> complete HCDC code

• Realistically, user of HCDC would write a mix of compliable C code & HCDC instructions
  • Most controllers are not system of differential equations
  • Getting accurate results requires trial & error
  • Working with the limited dynamic range of HCDC may be unfamiliar