Dynaban, an Open-Source Alternative Firmware for Dynamixel Servo-Motors

Rémi Fabre, Quentin Rouxel, Grégoire Passault, Steve N'Guyen, Olivier Ly

Rhoban Football Club - LaBRI - University of Bordeaux - France

Symposium RoboCup 2016: Robot World Cup XX
July 4th, 2016

Dynamixel Servo-Motors

- Widely used in the Humanoid Leagues
- Closed firmware $\rightarrow$ slow evolution
- Hinders the full control of the low level
- Some features are unsatisfactory and poorly documented (current sensing)

Other Projects

OpenServo

Mature project.
Low cost, low computational power.

DDServo

Open hardware and software (RX28, RX64).
Advanced control methods simulated but not implemented.

Dynaban Features

RHOBAN HUMANOÏD ROBOTICS

Dynaban
An alternative firmware for Dynamixel servos

`https://github.com/RhobanProject/Dynaban`

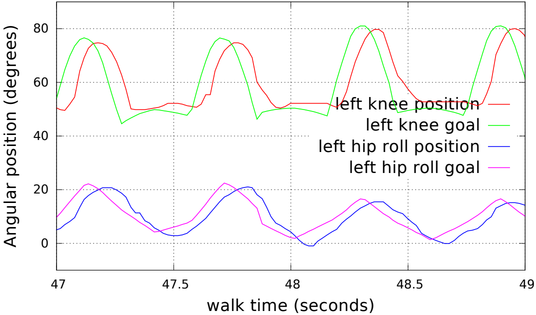
- Open Source
- Full retro compatibility
- Currently available for the MX-64 only but easily portable
- Heavily tested during the RoboCup

New features:

- High precision and high frequency measures
- Model based torque estimation
- Feed forward control (position and torque trajectories)

Feed Forward Control

- Default PID controller is not enough
- Feed forward solves many limitations of a purely reactive control



Same approach as:
Schwarz and Behnke: Compliant robot behaviour using servo actuator models identified by iterative learning control.
In: RoboCup International Symposium (2013)

→ Dynaban embeds it into the servo motor

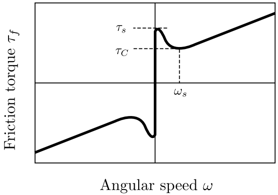
Models

Electric Model

$$U = \tau.\frac{R}{k_e} + k_e.\omega$$

U voltage
R resistance
k_e back-EMF
ω rotational speed

Classic friction model:



Friction Model

$$\tau_f = k_{vis}.\omega \;-\; sign(\omega).(\beta.\tau_s + (1-\beta)\tau_{cc})$$

$$\beta = e^{-\left|\frac{\omega}{\omega_{lin}}\right|^{\delta}}$$

ω_{lin} Stribeck effect limit
ω_{vis} viscous friction constant
τ_{cc} satisfies $\tau_f(\omega_{lin}) = \tau_c$

Parameters Optimization

Dynamical Model

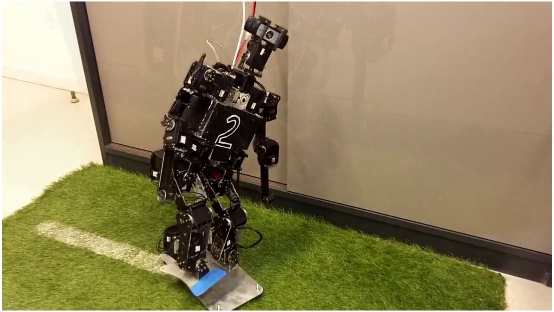
I_0 Shaft and gear box inertia

$\rightarrow$ The 7 model parameters are optimized from recorded data using the black box CMA-ES algorithm.
An order of magnitude better than manual tunning.

Note about Trajectory Representation:
Position, speed and torque trajectories are 4th order polynomials

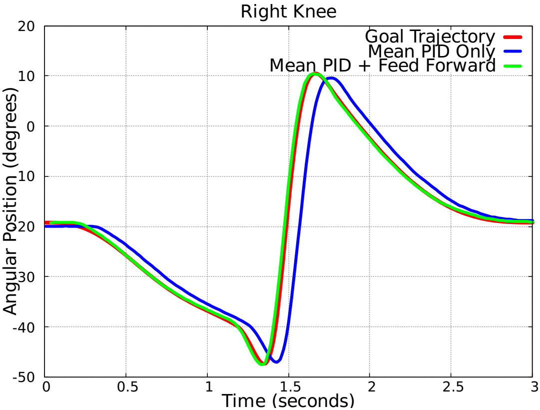
Experiments

- Highly dynamic kick movement
- Motion trajectory expressed in Cartesian space
- DOFs position, speed and torque are computed using kinematics and dynamics of the robot model.
- Polynomial splines are sent to Dynaban



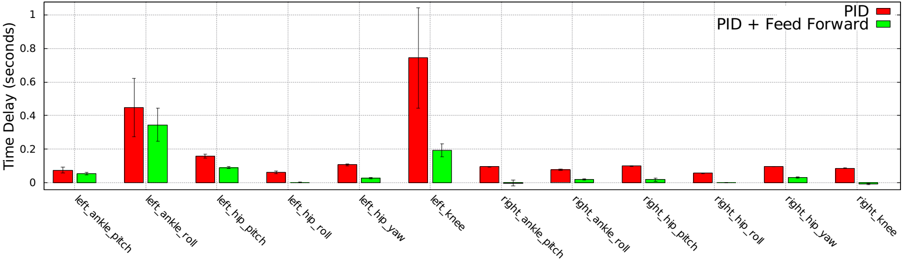
Preliminary Results

Average right knee trajectory:



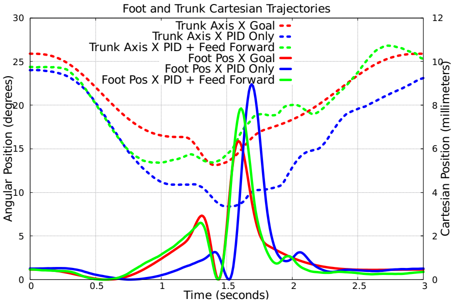
Detailed Results

High torques DOFs: $3\times$ better repeatability
Average delay on full motion: 80ms $\rightarrow$ 5ms
RMS error **after time shifting**: 46% higher with PID



Limitations

- Experiment done without ground truth
- Temperature influence observed but not modeled
- Hardware discrepancies between motors (+/- 10%) but model tunned only on one motor
- <u>Backslash not accounted for</u>



Conclusion

Thank for your attention