EMBEDDED IMPLEMENTATION OF A REAL-TIME SWITCHING CONTROLLER ON A ROBOTIC ARM

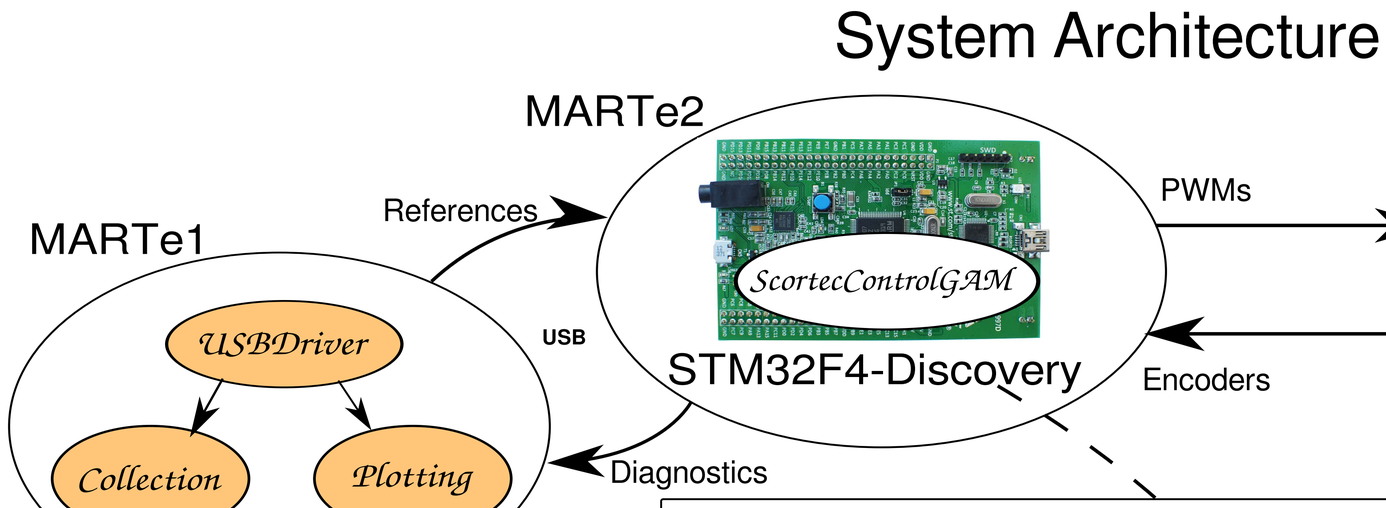
G. Ferrò[1,6], A. C. Neto[2], F. Sartori[2], L. Boncagni[3], D. Carnevale[1], Mateusz Gospodarczyk[1], A. Monti[1], A. Moretti[1], R.Vitelli[2], L. Capellà[4], I. Herrero[5]

[1]Dipartimento di Ingegneria Civile e Ingeneria Informatica, Università di Roma Tor Vergata, Via del Politecnico 1, 00133 Rome, Italy
[2]Fusion for Energy, Josep Pla 2 - 08019 Barcelona, Spain
[3]ENEA Unità Tecnica Fusione, C.R. Frascati, Via E.Fermi 45, 00044-Frascati, Rome, Italy
[4]Vitrociset Belgium, Rue Devant les Hetres 2 B-6890 Transinne. Belgium
[5]GTD Sistemas de Informaci´on, Paseo Garc´ıa Faria 17 E-08005 Barcelona, Spain
[6]CREATE Consortium Via Claudio, 21 80125 Napoli, Italy

The work leading to this publication has been based on a project performed in Fusion for Energy under the Contracts F4E-OFC-361-06 and F4E-OFC-620-01. This publication reflects the views only of the author, and Fusion for Energy cannot be held responsible for any use which may be made of the information contained therein

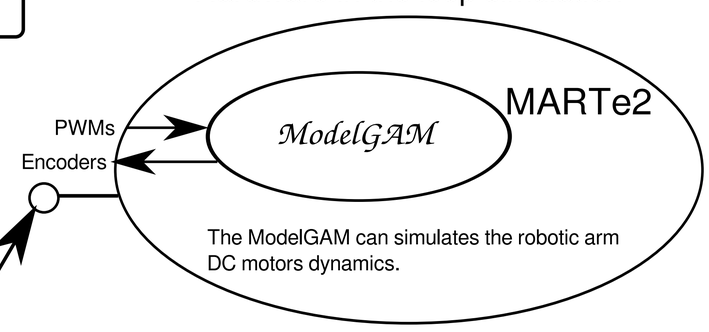
- New version of MARTe C++ framework has been developed with a software architecture aiming at enabling the execution of the same code across different bare-metal systems.
- The control algorithm to drive the DC motors of the robotic arm is based on a switching PID theory.
- Hardware in the loop control system simulation has been implemented.
- This work presents and compares the performance of the control algorithm implementation on a bare-metal and on a FreeRTOS deployment.

System Architecture

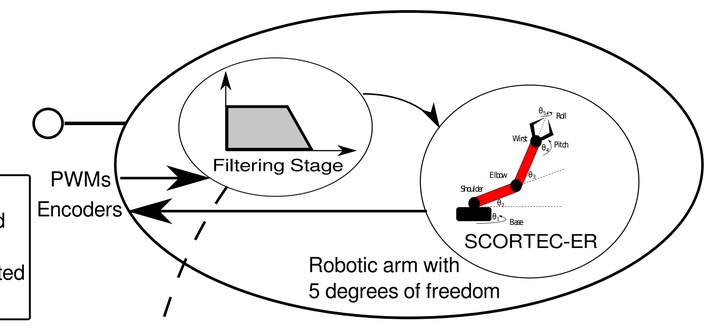


Two applications have been developed:
-**Application-1**: The ScortecControlGAM is directly linked to the hardware encoder counters and PWM signal sources.
-**Application-2**: Performs a hardware-in-the-loop simulation including a ModelGAM interconnected with the ScortecControlGAM to simulate the robotic arm.

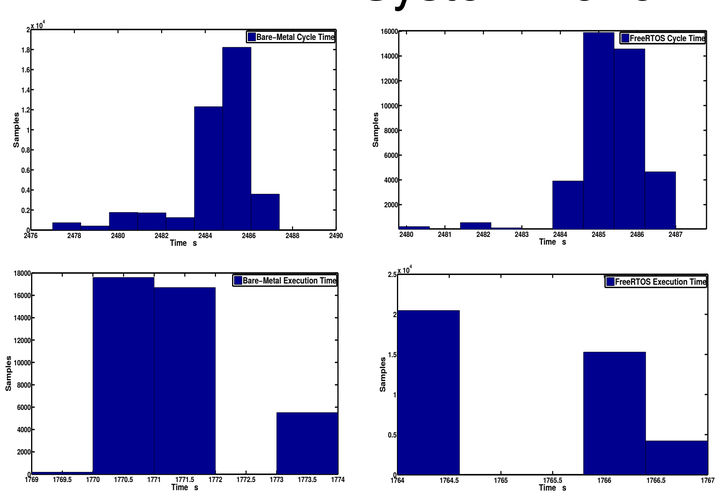
Hardware in the loop simulation



Robot phisically connected



System Performance



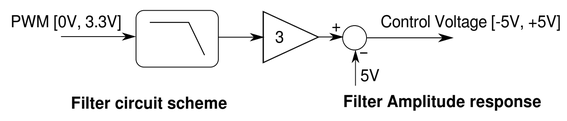
Cycle Time synchronised on 2.5ms

Bare-Metal Mean:2.484ms St.Deviation:2us
FreeRTOS Mean:2.485ms St.Deviation:1us

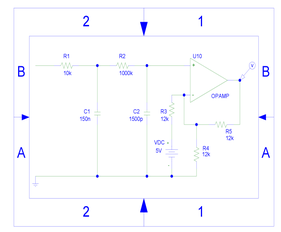
Cycle Execution Time

Bare-Metal Mean:1.772ms St.Deviation:1us
FreeRTOS Mean:1.765ms St.Deviation:1us

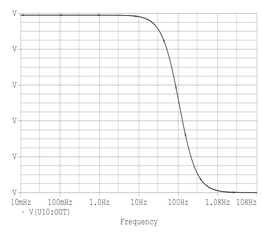
Filtering Stage



Filter circuit scheme



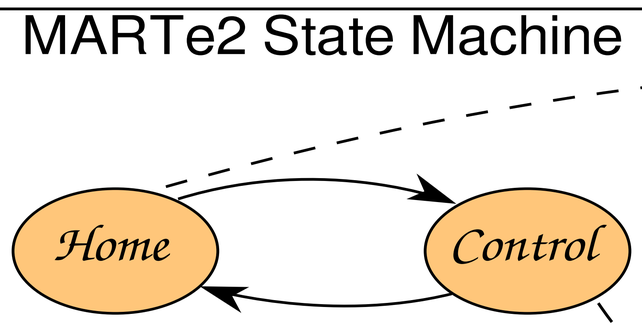
Filter Amplitude response



It is highly recommended to choose a PWM frequency greater than 1KHz.

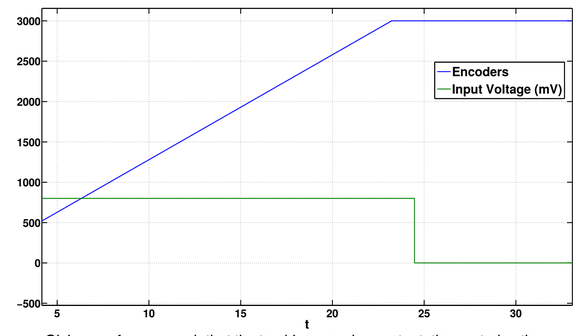
The **ScortecControlGAM** has a different behavior depending on the MARTe2 state.
State-1: the robot is driven to its **home** position (each joint to its limit switch)
State-switch: triggered by the user pressing STM32F4 user button. Encoders are set to zero for each robot joint
State-2: the robot can be controlled by the user.

MARTe2 State Machine



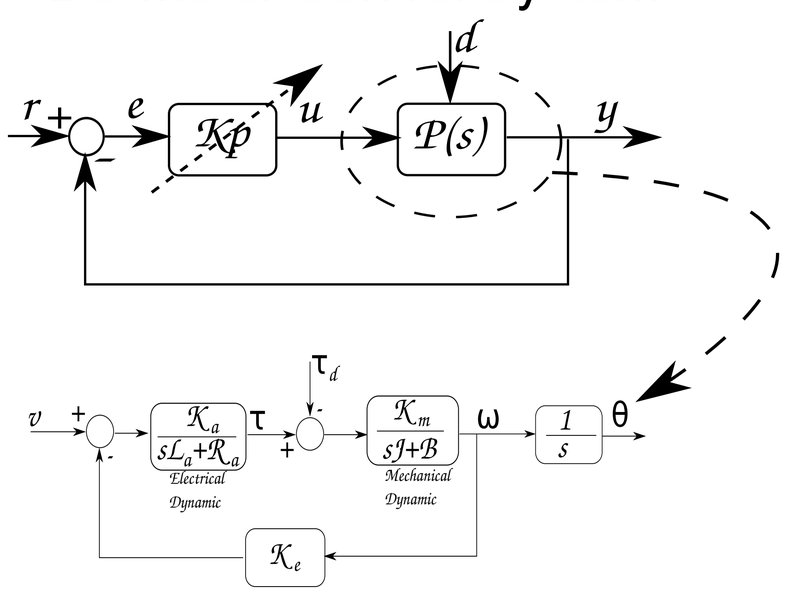
Hardware in the loop simulation

Home Procedure



Giving a reference such that the tracking error is constant, the control action brings the joint to its limit switch and then becomes zero.

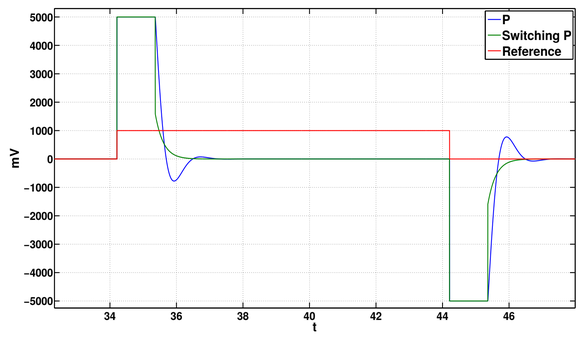
DC Motor Control System



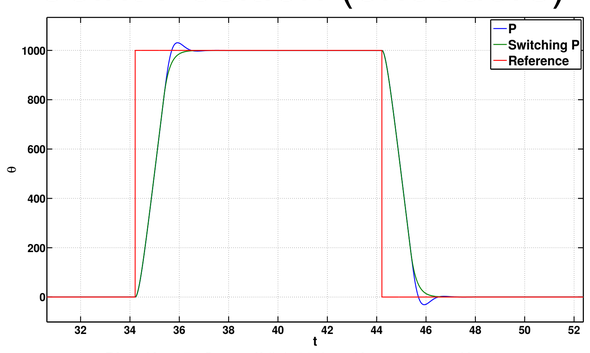
L_a, R_a = Motor armature intuctance and resistance
J = Momentum of Inertia
B = Viscosus friction
τ_d = Disturbance torque

K_e = Motor speed constant
K_a = Motor torque constant
K_m= Motor mechanical constant
K_p = Proportional controller gain

Input Voltage (mV)



Joint Position (encoders)



Blue: Kp=25 **Green**: if (|e|>200) --> Kp=25; else --> Kp =8

- FreeRTOS performance figures similar to Bare-Metal.
- Switching proportional controller can improve the transient responce performance (less overshoot, higher convergence speed).
- The same infrastructure can be used to develop similar real-time applications using this ecosystem.