

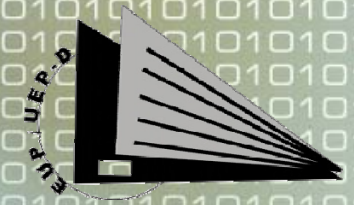


Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

ESCUELA UNIVERSITARIA POLITÉCNICA
UNIBERTSITATE ESKOLA POLITEKNIKOA

DONOSTIA – SAN SEBASTIÁN



Towards FPGA-based HIL emulation of sensorless first-order sliding-mode controller for doubly-fed induction generator

A. Susperregui^(), G. Tapia^(*), M. I. Martinez^(*), and H. Camblong^(*),^(**)*

^() UPV/EHU, ^(**) LIPSI-ESTIA*



Outline

- Introduction
 - Methodology
 - Motivation

- Real-time simulation case: Sensorless *1-SMC* of a *DFIG*
- Next step: Implementation of *1-SMC* on FPGA for HIL emulation
- Final considerations
 - Conclusions
 - Future work



Introduction

Methodology

- Aiming at speeding up off-line simulations as much as possible,
 - The models we build in *Simulink* are almost exclusively based on *S-function* blocks programmed in *C* language (C-MEX).
 - Concerning *power converters*, we used to
 - regard them as *ideal*, or
 - represent them in simplified form through *approximate models* considering mean values.
 - Those *approximations* may be *applicable to* power converters commanded via field oriented control (*FOC*).
 - Algorithms such as *DPC* or first-order sliding mode control (*1-SMC*) demand the use of *models considering the commutations* of the power converter's transistors.



Introduction

Motivation

- When incorporating converter models representing those commutations,
 - The execution of pure off-line simulations *slows down* considerably.
 - Due, above all, to *latency* and *'jitter' phenomena*, simulation *results* are not reliable any more, and may even become manifestly *incorrect*.
 - It is therefore necessary to have a *trustworthy tool* available, which allows not only facing the analysis and design tasks with *confidence*, but also *speeding up simulations*.

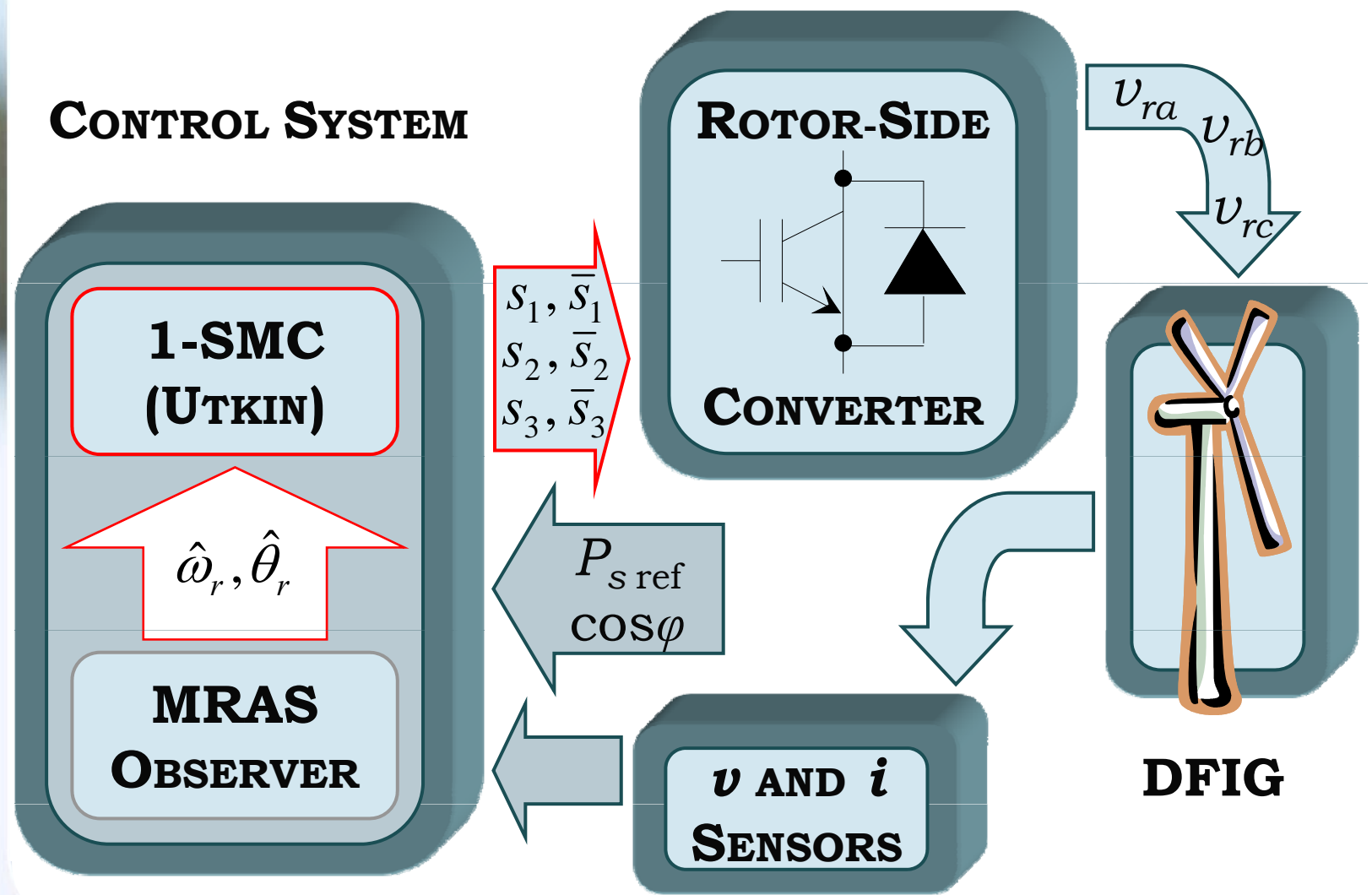


Real-time simulation case: Sensorless 1 -SMC of a *DFIG* (I)





Real-time simulation case: Sensorless 1-SMC of a DFIG (II)





Real-time simulation case: Sensorless *1*-SMC of a *DFIG* (III)

- For a *detailed description* of the control algorithm itself, please, refer to:

A. Susperregui, G. Tapia, I. Zubia, and J. X. Ostolaza, "Sliding-mode control of doubly-fed generator for optimum power curve tracking," *IET Electronics Letters*, vol. 46, no. 2, pp. 126–127, January 2010.



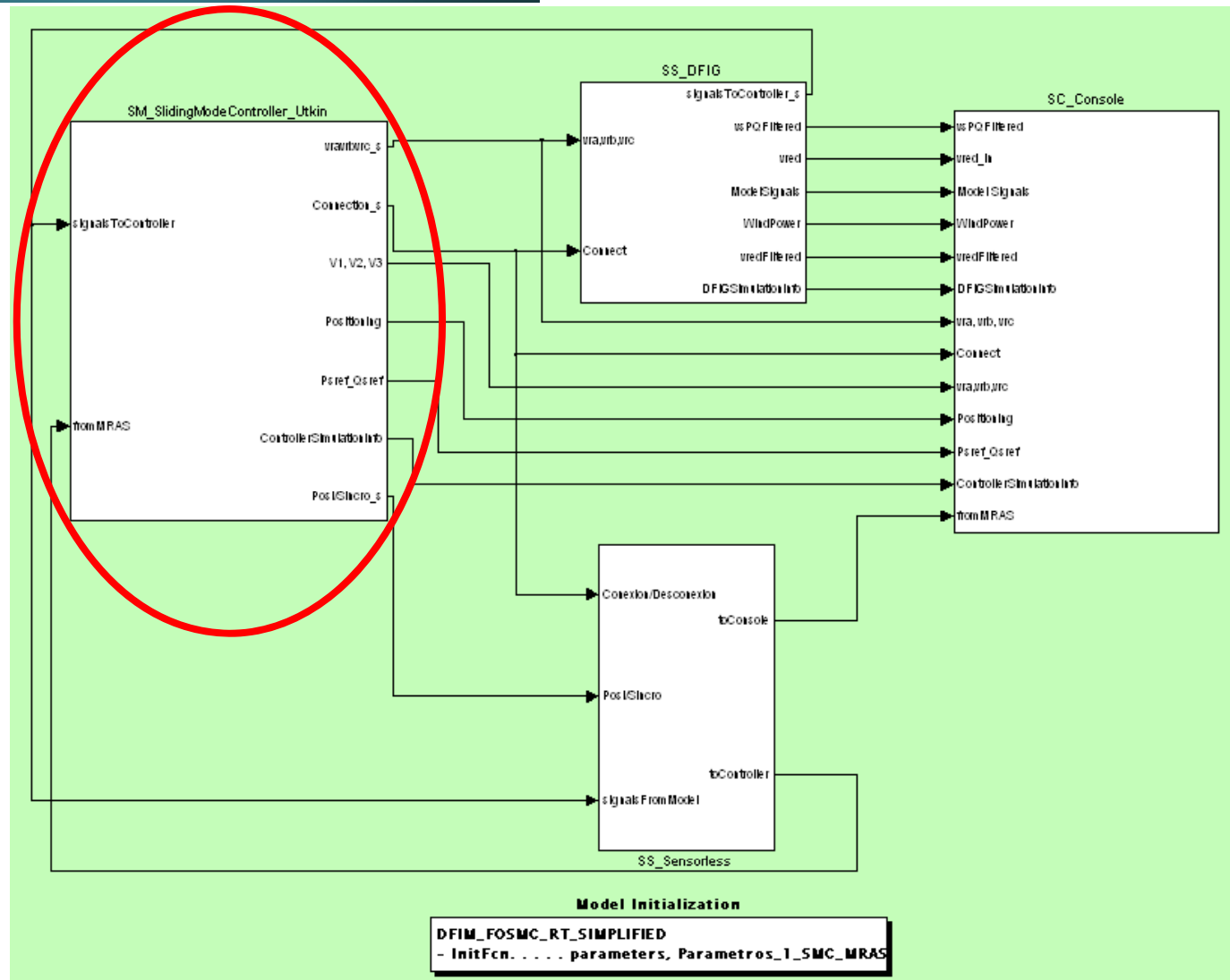


Real-time simulation case: Sensorless *1*-SMC of a *DFIG* (IV)

- It consists of *3 subsystems*, programmed in *C* via *3 S-functions*, each of which is run on a different CPU:
 - Doubly-fed induction generator (*DFIG*); *continuous*
 - Sliding-mode *controller*; *digital*; $f_s = 40 \text{ kHz}$
 - Position and speed *observer*; *digital*; $f_s = 1 \text{ kHz}$
- *Fixed* integration step; $25 \mu\text{s}$
- Mode: *single-task*
- Integration method: *ode5* (Dormand-Prince)



Real-time simulation case: Sensorless 1-SMC of a *DFIG* (V)





Real-time simulation case: Sensorless *1*-SMC of a *DFIG* (VII)

Event	Time instant (s)
Order of connection to the grid; start of synchronization process; initial convergence of the observer	0,474
End of synchronization process; connection to the grid	1,474
Start of power generation	1,974
Sudden increase of wind speed	7
Sudden decrease of wind speed	12



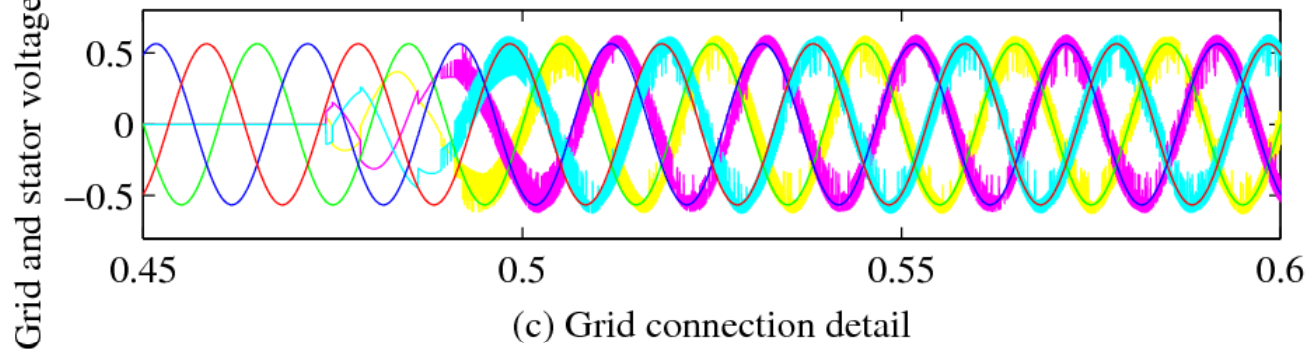


Real-time simulation case: Sensorless *1*-SMC of a *DFIG* (VIII)

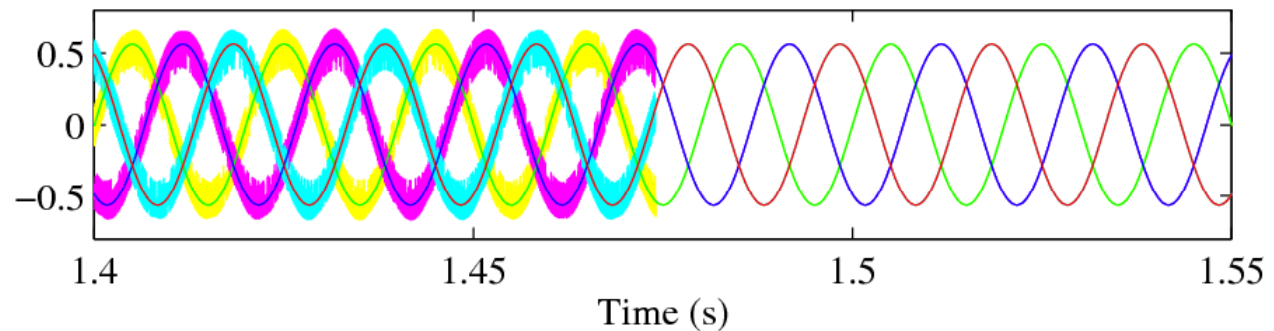
(a) Grid and stator voltages at the synchronization stage



(b) Detail at the synchronization stage

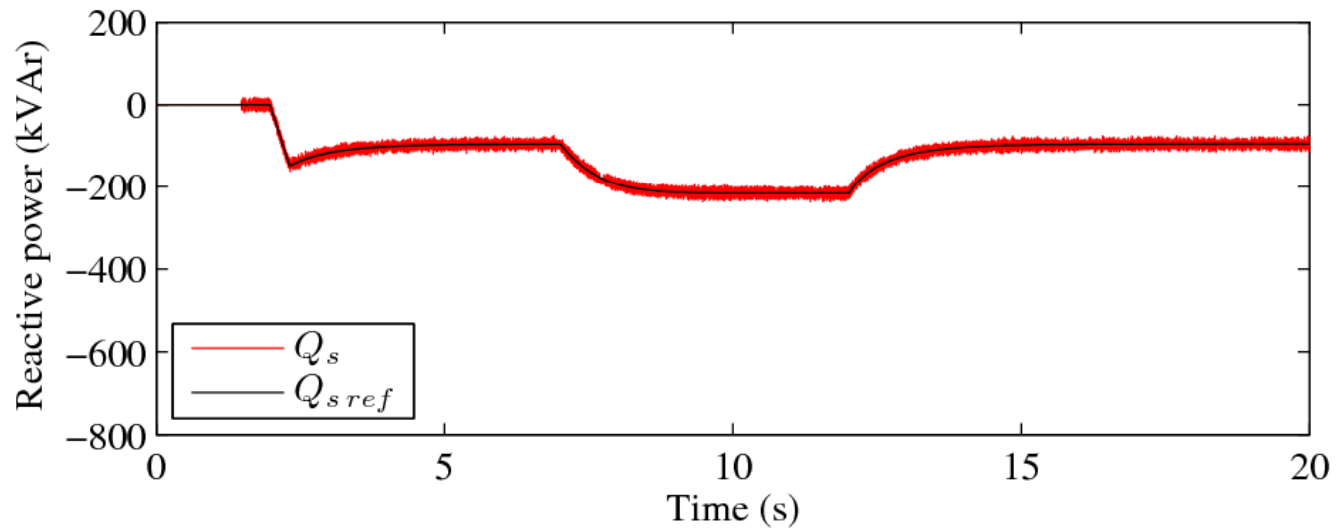
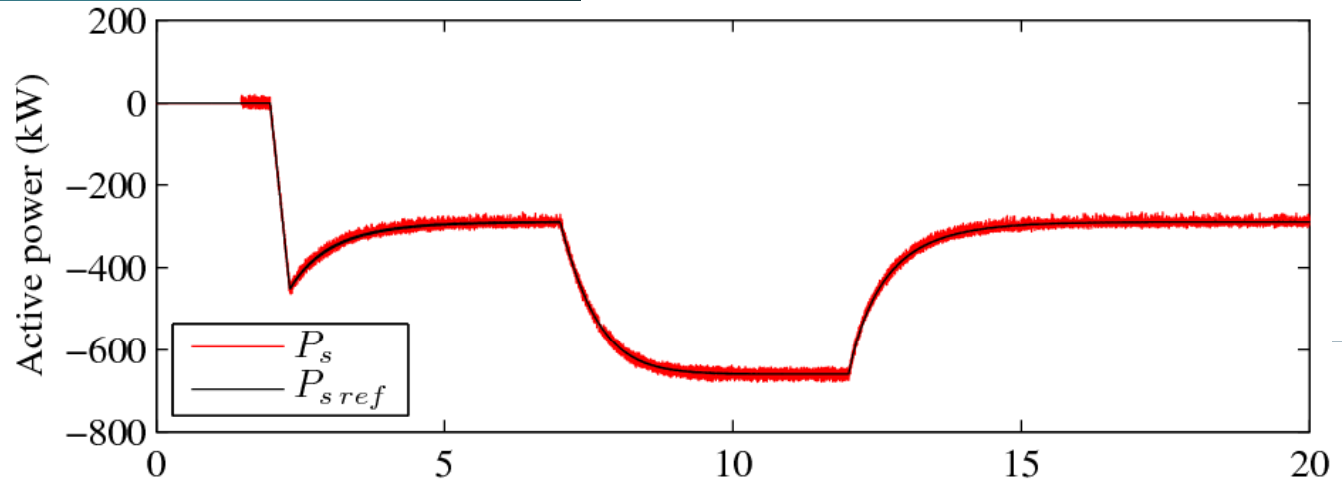


(c) Grid connection detail



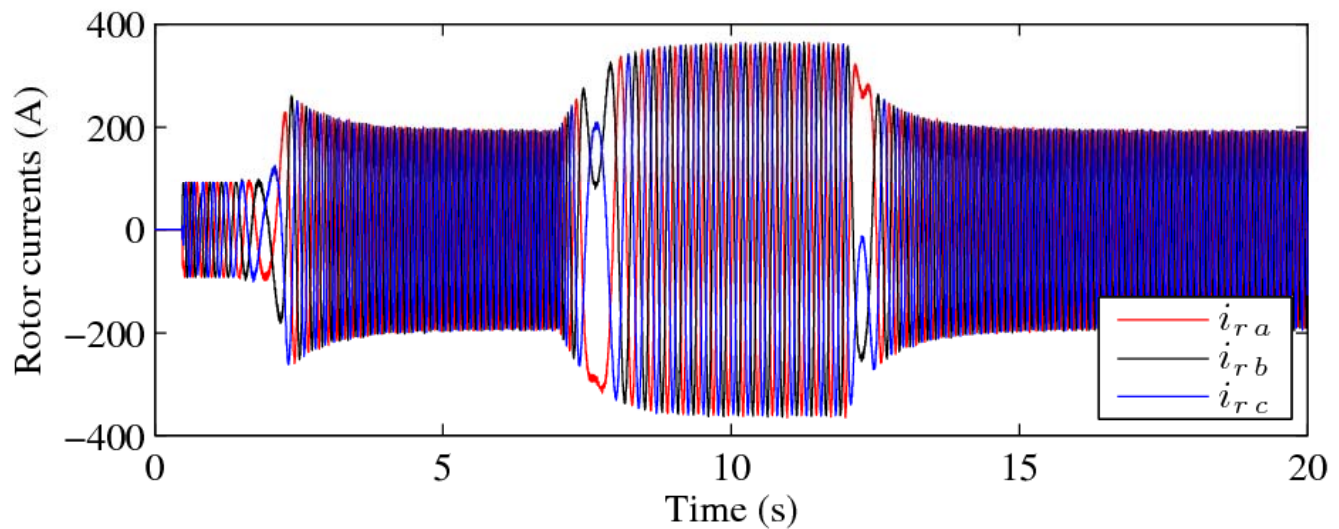
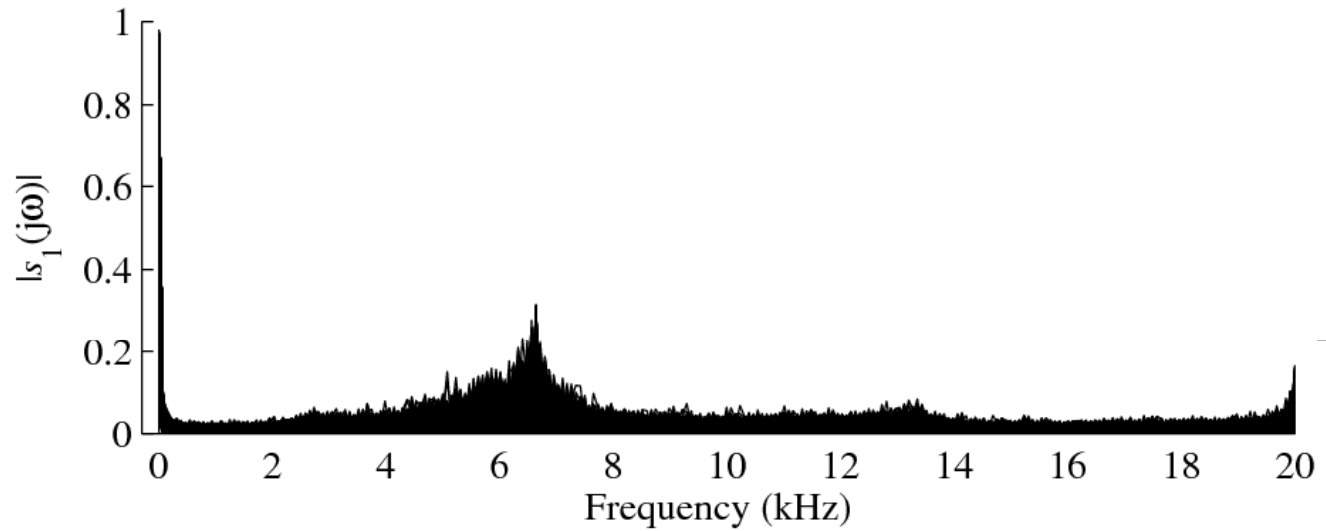


Real-time simulation case: Sensorless 1 -SMC of a *DFIG* (IX)



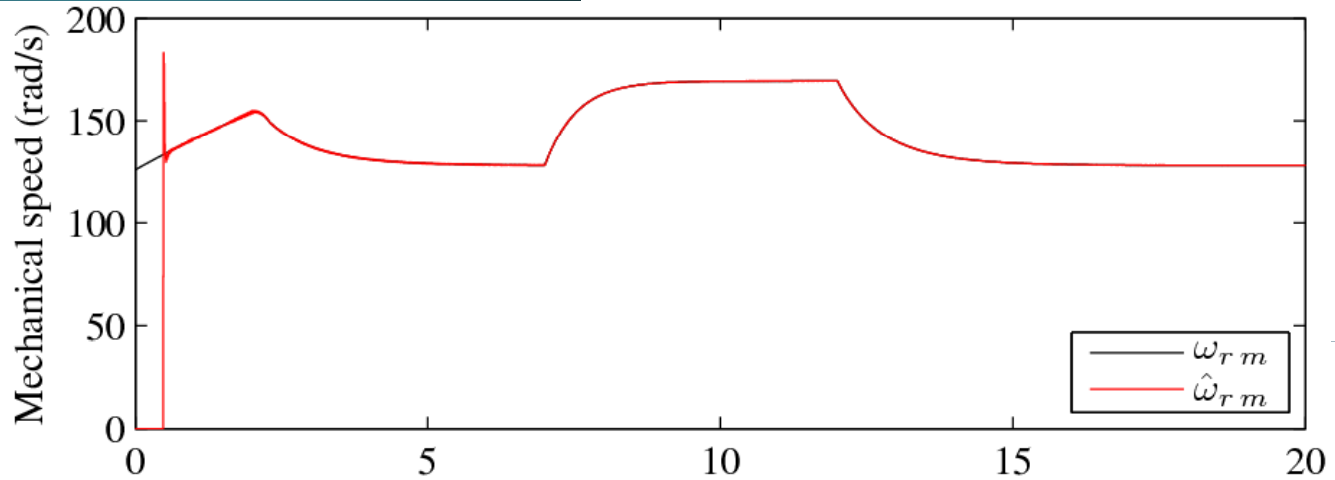


Real-time simulation case: Sensorless 1 -SMC of a *DFIG* (X)

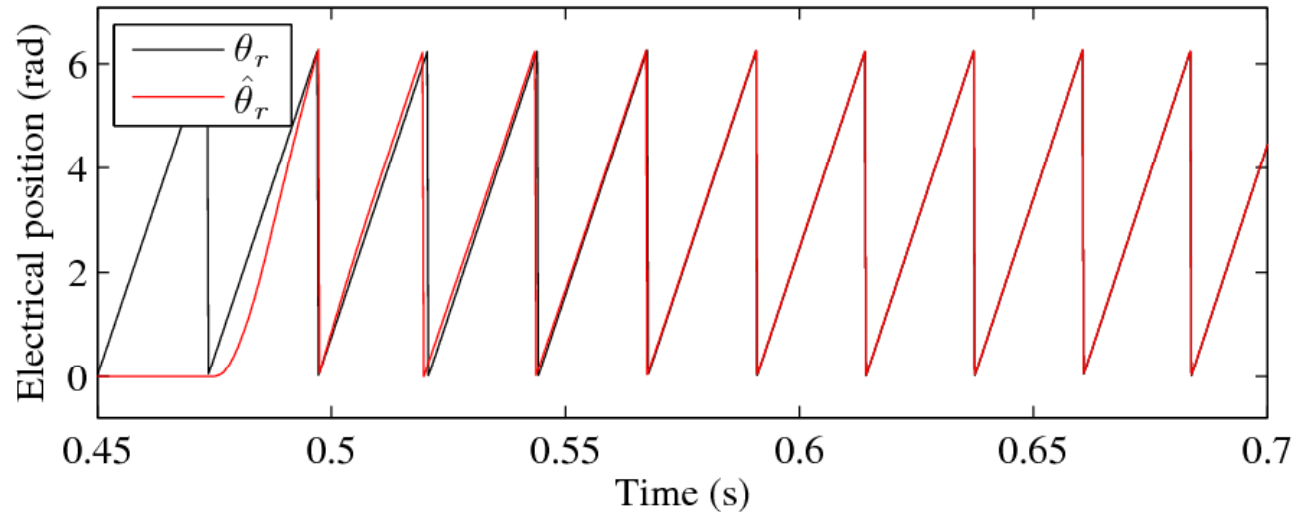




Real-time simulation case: Sensorless 1-SMC of a *DFIG* (XI)

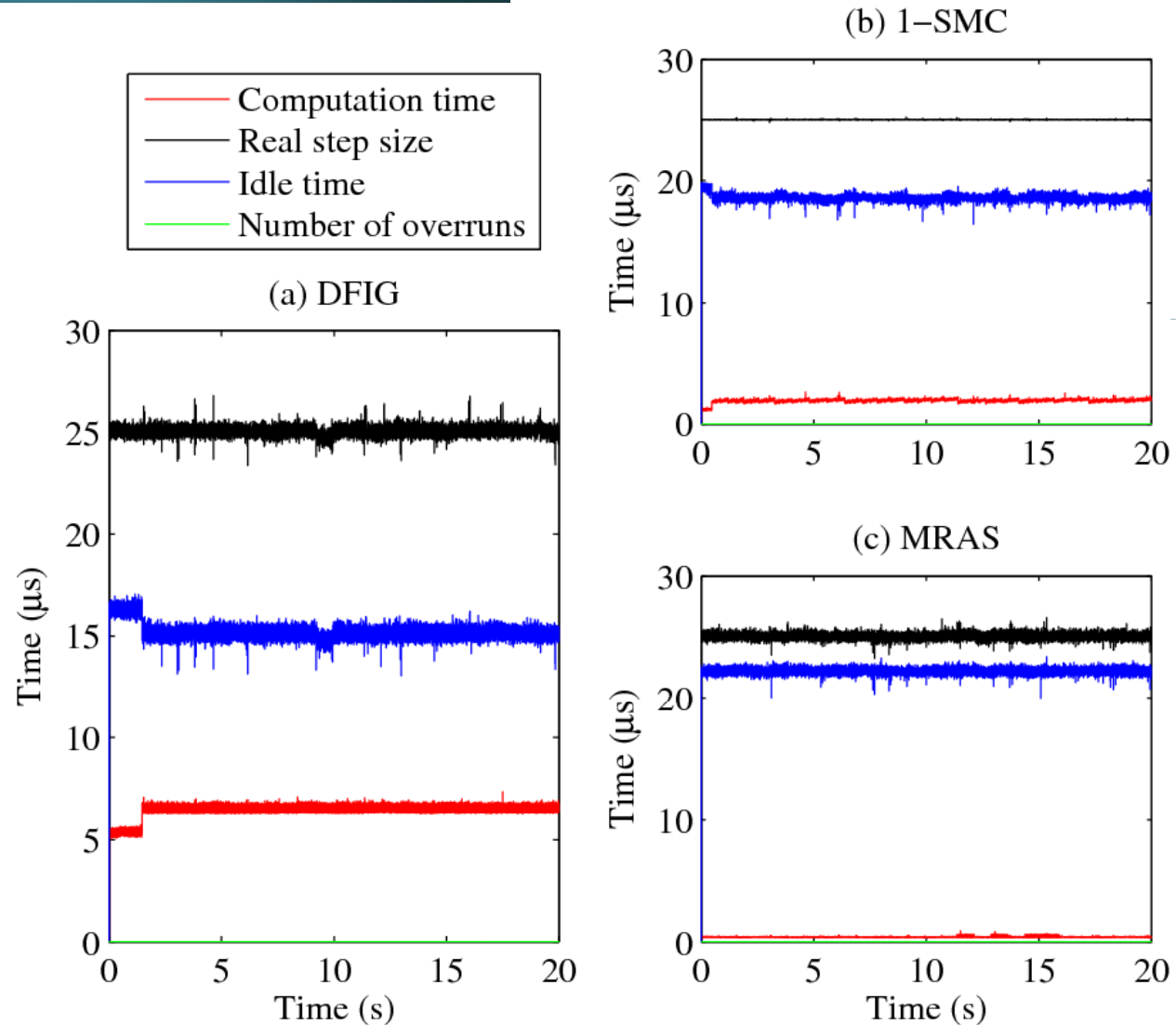


Detail at the beginning of the position estimation





Real-time simulation case: Sensorless 1-SMC of a *DFIG* (and XII)





Next step: Implementation of *1-SMC* on FPGA for HIL emulation (I)

- Given that the *1-SMC* algorithm *operates at* a particularly high sample rate of *40 kHz*, a *FPGA* might be considered as a *natural means to implement it* if a final product to be adopted by industry was required.
- Currently, the authors of this presentation and engineers from *Opal-RT* collaborate in the *XSG-based realization of the entire 1-SMC algorithm on a Virtex-II Pro series FPGA by Xilinx embedded in the eMEGAsim platform*.
- The objective consists in performing *HIL emulation* by replacing the *1-SMC* algorithm running on a CPU by that programmed in the FPGA.



Next step: Implementation of *1-SMC* on FPGA for HIL emulation (and II)

- Documentation provided to *Opal-RT* for training course preparation:
 - *List of parameters and constants* of the *1-SMC* algorithm
 - *List of inputs to and outputs* from the algorithm, specifying their sample rates and which of them are external to the *eMEGAsim* platform
 - *List of all* the input, output and intermediate *variables* included *in the algorithm*, together with their range of variation and fixed-point format
 - Full-detail *pseudo-code* of the *1-SMC* algorithm
 - *Original Simulink model* with the *1-SMC* programmed as a *C-MEX S-function* running on a CPU of *eMEGAsim*



Final considerations

Conclusions

- We believe that *the training course* arranged by *Opal-RT on the particular XSG implementation of the 1-SMC algorithm*
 - *will* definitively *be* a valuable help for us,
 - as well as *a shuttle for implementations of other control algorithms* we may synthesize in the future.
- Through *XSG implementation of the proposed 1-SMC*, it would be proved that it *is* not just a theoretical conception of difficult materialization, but *a practical algorithm that could be industrially implemented on a commercially available FPGA*.



Final considerations

Future work

- Once the *1-SMC* algorithm implemented on the FPGA is validated through HIL emulation, it is planned to test, via *rapid control prototyping*, the global control system consisting of
 - a *1-SMC* algorithm programmed on the FPGA embedded in eMEGAsim, and
 - a *MRAS* observer, implemented as a *C-MEX S-function* running on a CPU,
 on a 7-kW lab-scale *DFIG test bench*.