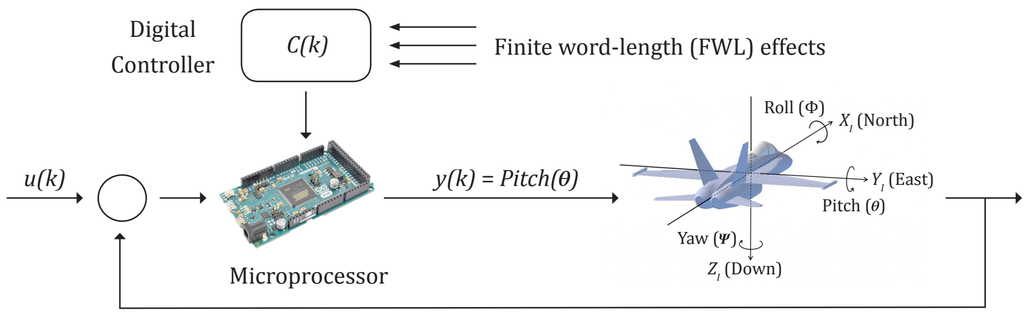
Verifying Digital Systems with MATLAB (Tool Demo)

Lennon Chaves[1], Iury Bessa[1], Lucas Cordeiro[1,2], Daniel Kroening[2] and Eddie Lima[1]

[1]Federal University of Amazonas, Brazil [2]University of Oxford, United Kingdom

lennonchaves@ufam.edu.br • iurybessa@ufam.edu.br • lucas.cordeiro@cs.ox.ac.uk • kroening@cs.ox.ac.uk • eddie_batista@yahoo.com.br

I Motivation



"...guaranteeing the correctness of cyber-physical systems (CPS) remains an a stounding challenge"

Xi Zheng *et al.*, 2014.

"Simulation alone is not sufficient to support verification and validation of CPS."

Sayan Mitra *et al.*, 2013.

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$ ← State-space model

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_m z^{-m}}{1 + a_1 z^{-1} + \ldots + b_n z^{-n}}$$ ← Transfer-function model

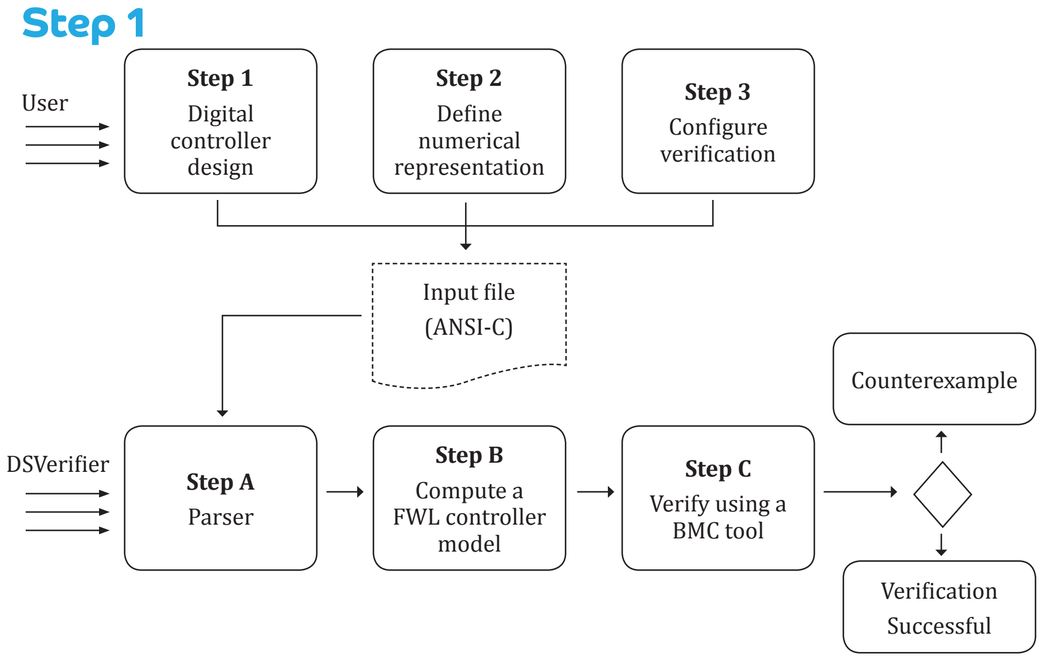
II Approach and Uniqueness

Bounded model checking

Translate the model into a VC ψ such that:

ψ is satisfiable iff φ has counterexample of max. depth *k*

Step 1



$$\begin{cases} x(k+1) = \begin{bmatrix} 0.9969 & 0.05649 & 0 \\ -0.0001 & 0.99570 & \\ 0 & 0.5658 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0.0024 \\ 0.0002 \\ 0.0001 \end{bmatrix} u(k) \\ y = (k) = [0 \quad 0 \quad 1]x(k) + [0]u(k) \end{cases}$$

Step 2

Numerical representation < *I* , *F* >:

- ***I*** is the integer part and
- ***F*** is the fractional part

```
implementation <3,4>
states = 3;
inputs = 1;
outputs = 1;
A = [...]
B = [...]
C = [...]
D = [...]
```

Step 3

Setup verification:

- choose a property Φ;
- a maximum verification time;
- a bound k;
- a BMC tool.

Properties:

- stability;
- quantization error;
- controllability;
- observability;

Step A

DSVerifier builds an ANSI-C code representation of the digital system based on the specification.

C

Step B

DSVerifier formulates a FWL model based on fixedpoint arithmetic:

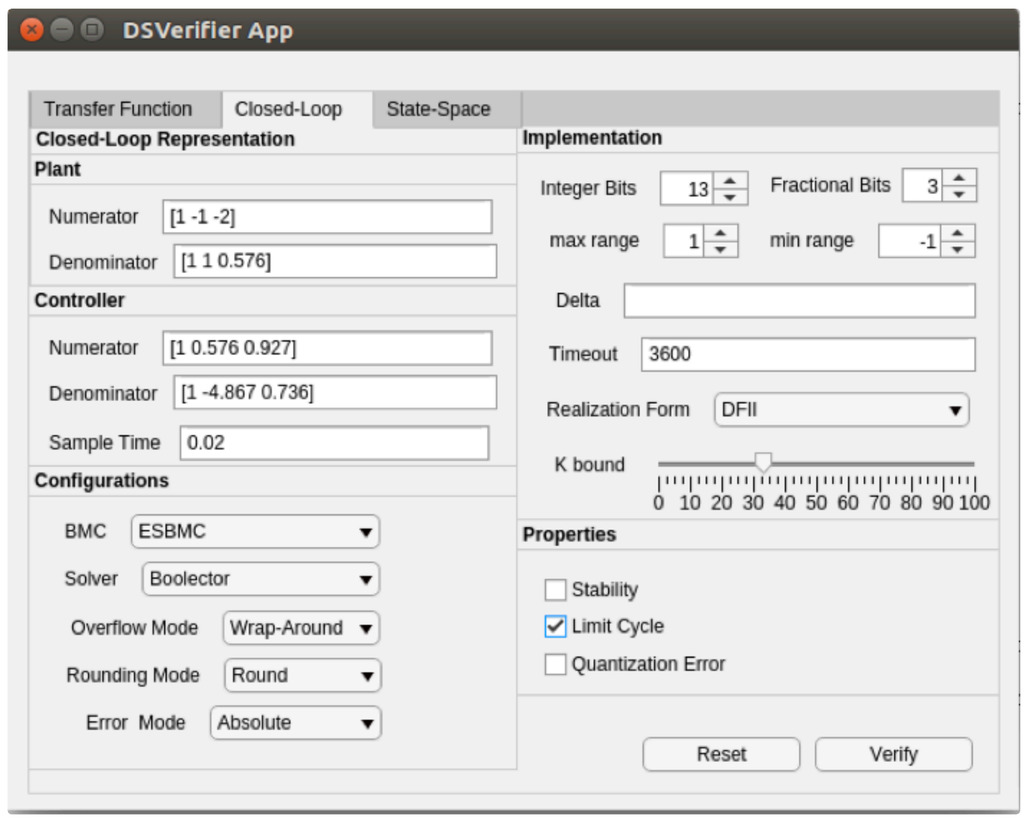
$$\text{FWL}\,[\bullet]:\Re \longrightarrow Q[\Re]$$

Step C

DSVerifier checks a property Φ up to a bound *k*:

Φ	*Bits < I, F >*	Result
Quantization error	32-bits <15,16>	**>1%**
	16-bits <7,8>	**>1%**
	8-bits <3,4>	**<1%**
Stability	8-bits <3,4>	**Unstable**

III DSVerifier Toolbox



IV Contributions

i. support for transfer-function and state-space representations in open- and closed-loop form;

ii. verify different numerical representations and realization forms of digital systems;

iii. provide a MATLAB toolbox to check specific properties of digital systems while taking into account FWL;

As future work:

- verify uncertainties in digital systems represented by state-space;
- integrate counterexample reproducibility for digital systems

Sponsors:

For further information, publications, and downloads, see: http://www.dsverifier.org/