Sound and Automated Synthesis of Digital Controllers for Continuous Plants

Alessandro Abate, Iury Bessa, Dario Cattaruzza, Lucas Cordeiro, Cristina David, Pascal Kesseli and Daniel Kroening

March 2017

Cyber-Physical Systems (CPS)

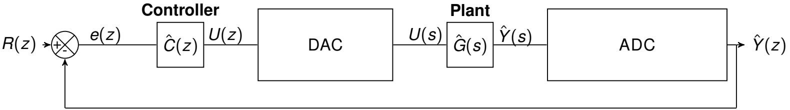
- modern controls are implemented with digital microcontrollers, embedded within dynamical plants representing physical components
- digital control literature: success and limitations

Cyber-Physical Systems (CPS)

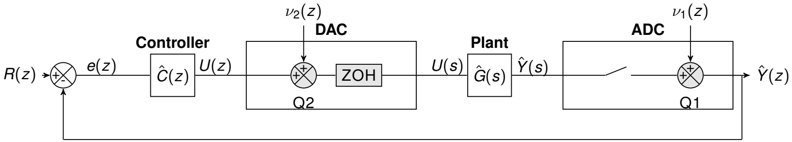
- modern controls are implemented with digital microcontrollers, embedded within dynamical plants representing physical components
- digital control literature: success and limitations

- use of counterexample guided inductive synthesis (CEGIS) to automate design of sound digital controllers

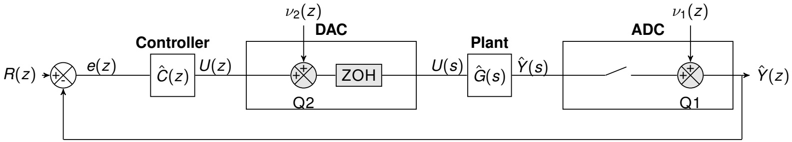
CPS Setup: Continuous Plant and Digital Controller

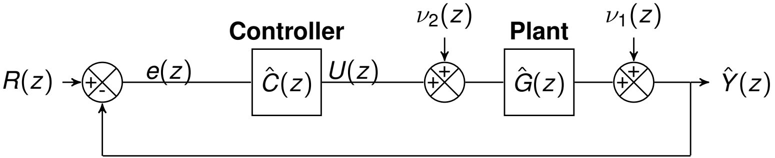


CPS Setup: Continuous Plant and Digital Controller

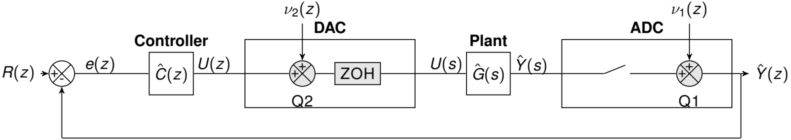


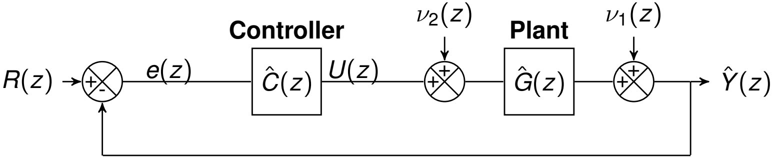
CPS Setup: Continuous Plant and Digital Controller





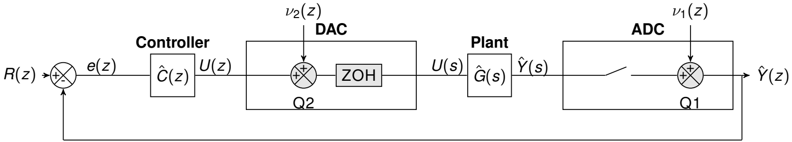
CPS Setup: Continuous Plant and Digital Controller

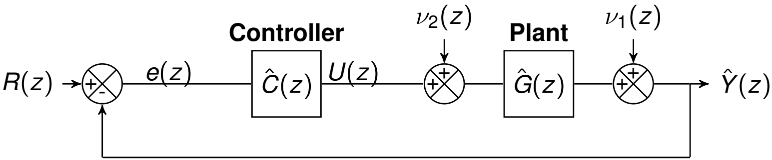




$$\hat{G}(z) = (1 - z^{-1})\mathcal{Z}\left\{\mathcal{L}^{-1}\left\{\frac{\hat{G}(s)}{s}\right\}_{t=kT}\right\}$$

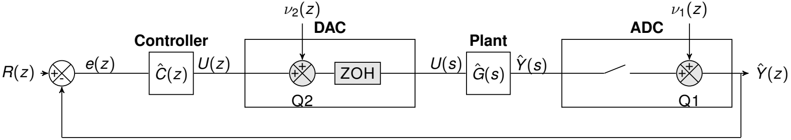
CPS Setup: Continuous Plant and Digital Controller

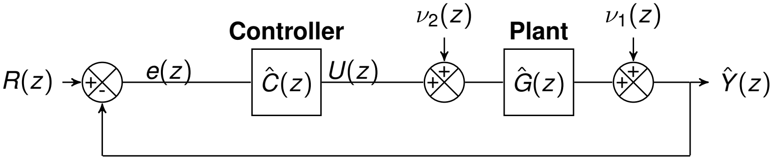




$$\hat{Y}(z) = \frac{\hat{G}(z)\hat{C}(z)}{1+\hat{G}(z)\hat{C}(z)}R(z) + \frac{1}{1+\hat{G}(z)\hat{C}(z)}\nu_1(z) + \frac{\hat{G}(z)}{1+\hat{G}(z)\hat{C}(z)}\nu_2(z)$$

CPS Setup: Continuous Plant and Digital Controller





Common denominator:

$$\hat{G}_n(z)\hat{C}_n(z) + \hat{G}_d(z)\hat{C}_d(z)$$

Stability Analysis via Jury's Criterion

$$S(z) = \hat{G}_n(z)\hat{C}_n(z) + \hat{G}_d(z)\hat{C}_d(z) = \sum_{i=0}^{N_S} c_i z^{N_s - i}, c_0 \neq 0$$

$$M = \begin{pmatrix} v_1^{(0)} & \cdots & v_{N_S}^{(0)} \\ v_1^{(1)} & \cdots & v_{N_S}^{(1)} \\ \vdots & \vdots & \vdots \\ v_1^{(N_S-1)} & \cdots & v_{N_S}^{(N_S-1)} \end{pmatrix},$$

$$v_j^{(k)} = \begin{cases} c_{j-1}, & k = 0 \\ 0, & k > 0 \wedge \text{if } j > N_S - k \\ v_1^{(k-1)} - v_{N_S-j}^{(k-1)} \cdot \frac{v_1^{(k-1)}}{v_{N_S-k}^{(k-1)}}, & k > 0 \wedge \text{if } j \leq N_S - k \end{cases}$$

$$S(z) \text{ is stable } \Leftrightarrow$$

$S(1) > 0$

Stability Analysis via Jury's Criterion

$$S(z) = \hat{G}_n(z)\hat{C}_n(z) + \hat{G}_d(z)\hat{C}_d(z) = \sum_{i=0}^{N_S} c_i z^{N_s - i}, c_0 \neq 0$$

$$M = \begin{pmatrix} v_1^{(0)} & \cdots & v_{N_S}^{(0)} \\ v_1^{(1)} & \cdots & v_{N_S}^{(1)} \\ \vdots & \vdots & \vdots \\ v_1^{(N_S-1)} & \cdots & v_{N_S}^{(N_S-1)} \end{pmatrix},$$

$$v_j^{(k)} = \begin{cases} c_{j-1}, & k = 0 \\ 0, & k > 0 \wedge \text{if } j > N_S - k \\ v_1^{(k-1)} - v_{N_S-j}^{(k-1)} \cdot \frac{v_1^{(k-1)}}{v_{N_S-k}^{(k-1)}}, & k > 0 \wedge \text{if } j \leq N_S - k \end{cases}$$

$S(z)$ is stable $\Leftrightarrow$

$S(1) > 0 \wedge (-1)^{N_S} S(-1) > 0$

Stability Analysis via Jury's Criterion

$$S(z) = \hat{G}_n(z)\hat{C}_n(z) + \hat{G}_d(z)\hat{C}_d(z) = \sum_{i=0}^{N_S} c_i z^{N_s - i}, c_0 \neq 0$$

$$M = \begin{pmatrix} v_1^{(0)} & \cdots & v_{N_S}^{(0)} \\ v_1^{(1)} & \cdots & v_{N_S}^{(1)} \\ \vdots & \vdots & \vdots \\ v_1^{(N_S-1)} & \cdots & v_{N_S}^{(N_S-1)} \end{pmatrix},$$

$$v_j^{(k)} = \begin{cases} c_{j-1}, & k = 0 \\ 0, & k > 0 \wedge \text{if } j > N_S - k \\ v_1^{(k-1)} - v_{N_S-j}^{(k-1)} \cdot \frac{v_1^{(k-1)}}{v_{N_S-k}^{(k-1)}}, & k > 0 \wedge \text{if } j \leq N_S - k \end{cases}$$

$$S(z) \text{ is stable } \Leftrightarrow$$

$$S(1) > 0 \ \wedge \ (-1)^{N_S} S(-1) > 0 \ \wedge \ |c_0| < c_{N_S}$$

Stability Analysis via Jury's Criterion

$$S(z) = \hat{G}_n(z)\hat{C}_n(z) + \hat{G}_d(z)\hat{C}_d(z) = \sum_{i=0}^{N_S} c_i z^{N_s - i}, c_0 \neq 0$$

$$M = \begin{pmatrix} v_1^{(0)} & \cdots & v_{N_S}^{(0)} \\ v_1^{(1)} & \cdots & v_{N_S}^{(1)} \\ \vdots & \vdots & \vdots \\ v_1^{(N_S-1)} & \cdots & v_{N_S}^{(N_S-1)} \end{pmatrix},$$

$$v_j^{(k)} = \begin{cases} c_{j-1}, & k = 0 \\ 0, & k > 0 \wedge \text{if } j > N_S - k \\ v_1^{(k-1)} - v_{N_S-j}^{(k-1)} \cdot \frac{v_1^{(k-1)}}{v_{N_S-k}^{(k-1)}}, & k > 0 \wedge \text{if } j \leq N_S - k \end{cases}$$

$$S(z) \text{ is stable } \Leftrightarrow$$

$$S(1) > 0 \;\wedge\; (-1)^{N_S} S(-1) > 0 \;\wedge\; |c_0| < c_{N_S} \;\wedge\; \forall k, v_1^{(k)} > 0$$

Transfer Functions and Uncertainty

$$\hat{C}(z) = \frac{\hat{C}_n(z)}{\hat{C}_d(z)} = \frac{\sum_{i=0}^{M_C}(\beta_i + \Delta\beta_i)z^{-i}}{\sum_{i=0}^{N_C}(\alpha_i + \Delta\alpha_i)z^{-i}},$$

$$\hat{G}(z) = \frac{\hat{G}_n(z)}{\hat{G}_d(z)} = \frac{\sum_{i=0}^{M_G}(b_i + \Delta b_i)z^{-i}}{\sum_{i=0}^{N_G}(a_i + \Delta a_i)z^{-i}}.$$

- $\vec{\beta}$ and $\vec{\alpha}$ – vectors containing controller coefficients
- $\vec{b}$ and $\vec{a}$ – plant coefficients

Sources of Uncertainty

1. Model Parametric Errors (affect $\Delta G(z)$)

Sources of Uncertainty

1. Model Parametric Errors (affect $\Delta G(z)$)
2. Quantisation errors in the ADC, DAC, and controller arithmetics (modelled as external noise, do not affect transfer functions)

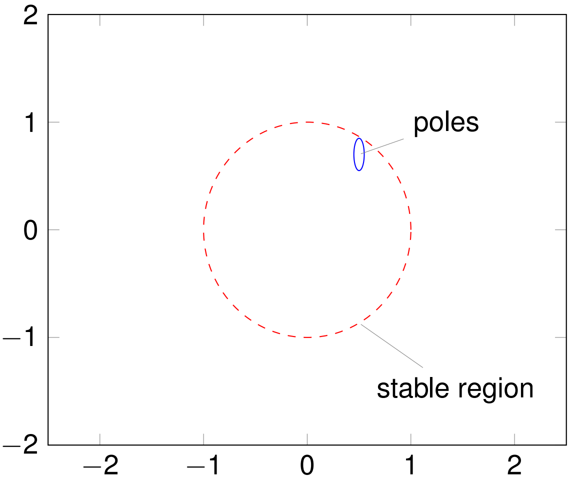
Sources of Uncertainty

1. Model Parametric Errors (affect $\Delta G(z)$)
2. Quantisation errors in the ADC, DAC, and controller arithmetics (modelled as external noise, do not affect transfer functions)
3. Roundoff errors in verification process (affect $\Delta S(z)$)

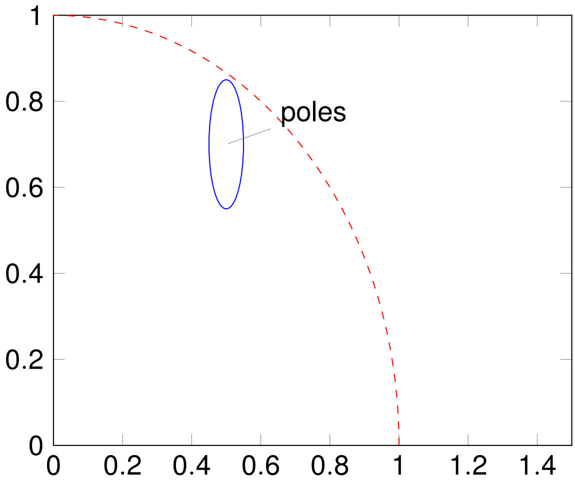
Sources of Uncertainty

1. Model Parametric Errors (affect $\Delta G(z)$)
2. Quantisation errors in the ADC, DAC, and controller arithmetics (modelled as external noise, do not affect transfer functions)
3. Roundoff errors in verification process (affect $\Delta S(z)$)
4. Discretisation errors (affect $\Delta G(z)$ and $\Delta C(z)$)

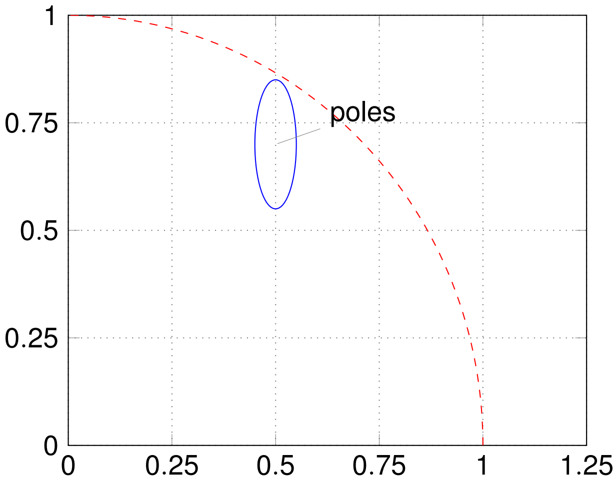
Pole Placement



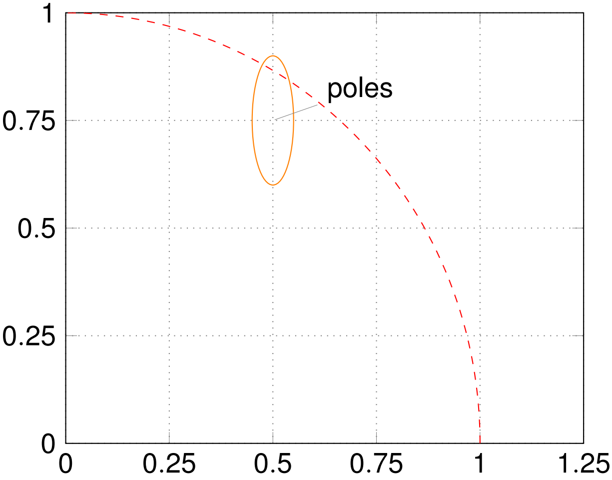
Pole Placement



Pole Placement in a Discretized Domain

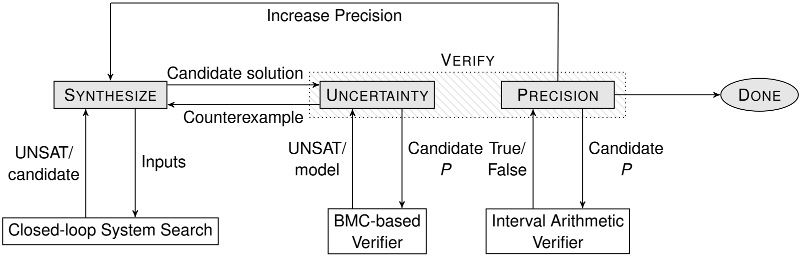


Pole Placement in a Discretized Domain



Objective

We seek to synthesise a controller where $\Delta\vec{\beta} = 0$ and $\Delta\vec{\alpha} = 0$ and which stabilizes all plants within a given range of $\Delta\vec{b}$ and $\Delta\vec{a}$.



Case Study

Take a classical cruise control example from the literature.

$$G(z) = \frac{0.0264}{z - 0.9998}.$$

Using an optimization tool, the following stable high-performance controller was designed:

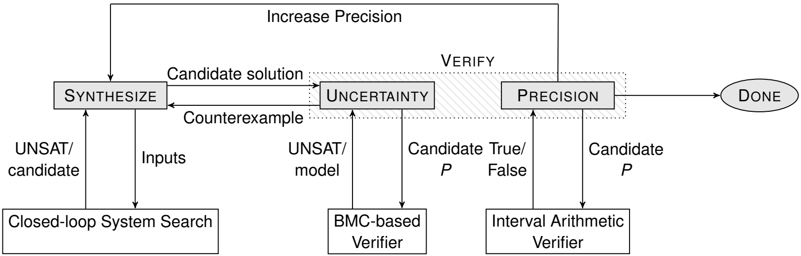
$$C(z) = \frac{2.72z^2 - 4.153z + 1.896}{z^2 - 1.844z + 0.8496}.$$

However, if we implement the controller in $\mathbb{R}\langle 4, 16\rangle$.

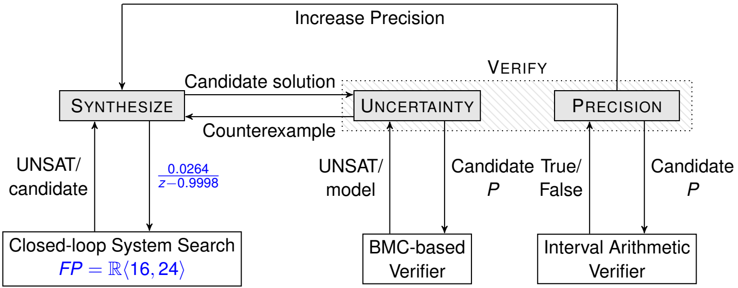
$$\tilde{C}(z) := \frac{2.7199859619140625z^2 - 4.1529998779296875z + 1.89599609375}{z^2 - 1.843994140625z + 0.8495941162109375}.$$

where $\tilde{C}(z)$ is the controller adjusted to the FWL. The resulting system is unstable.

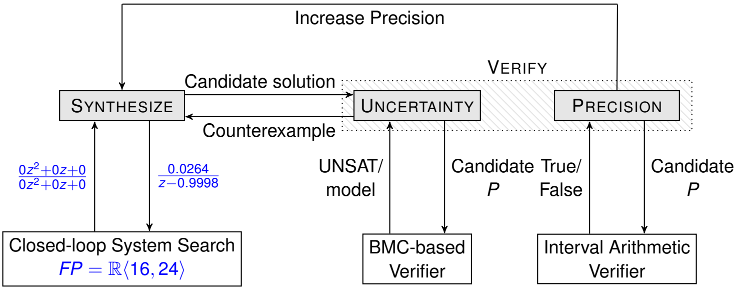
CEGIS Architecture



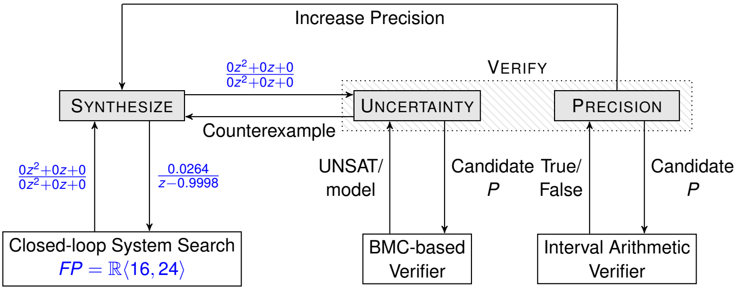
CEGIS Architecture



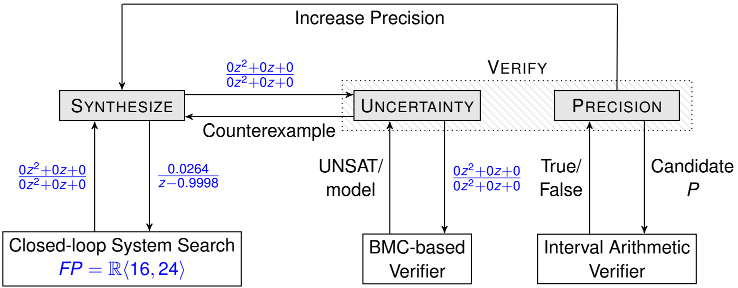
CEGIS Architecture



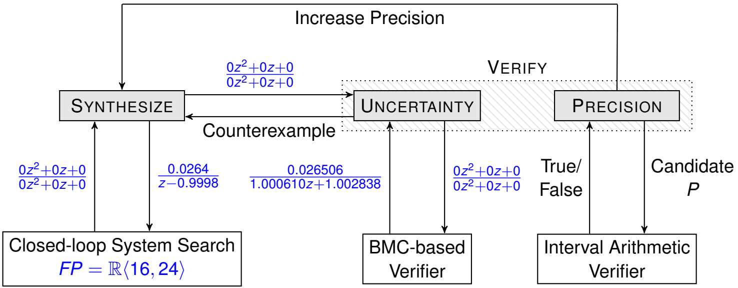
CEGIS Architecture



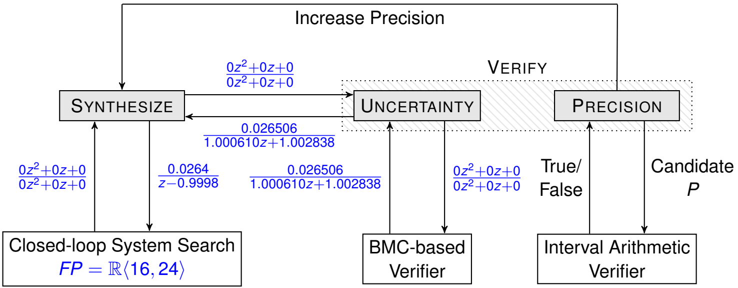
CEGIS Architecture



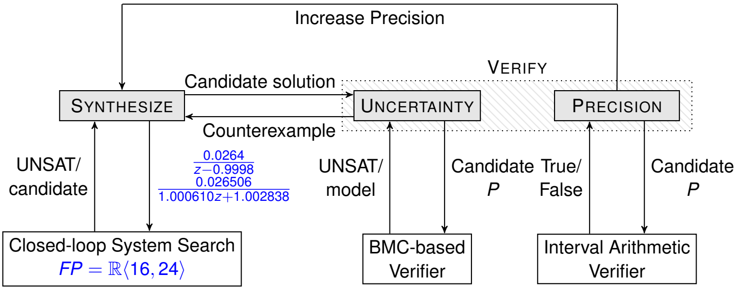
CEGIS Architecture



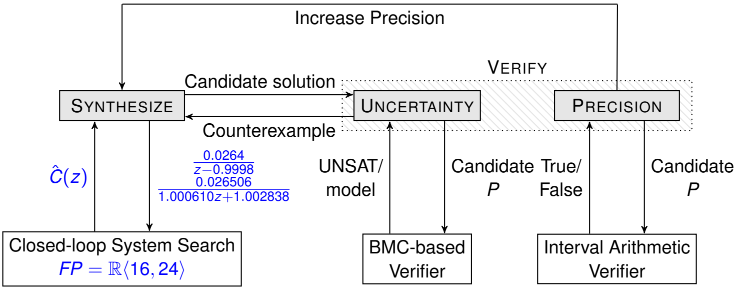
CEGIS Architecture



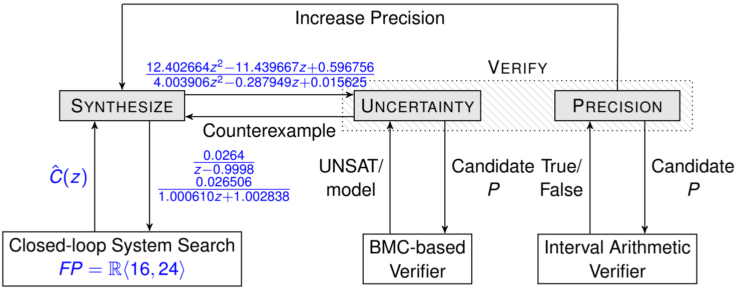
CEGIS Architecture



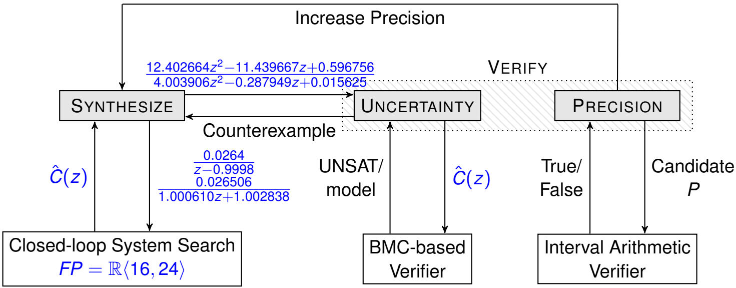
CEGIS Architecture



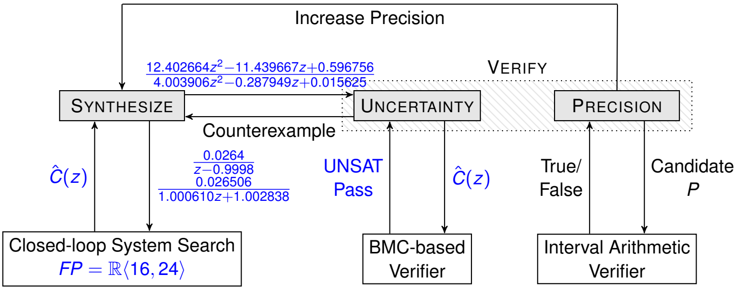
CEGIS Architecture



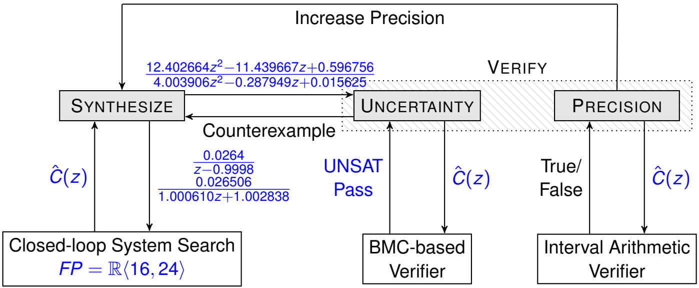
CEGIS Architecture



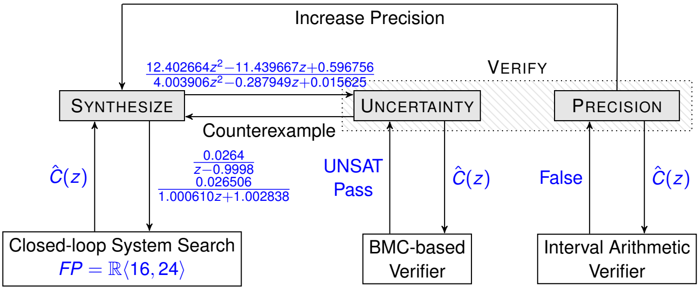
CEGIS Architecture



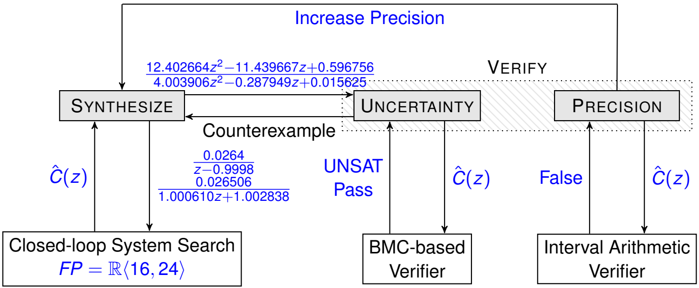
CEGIS Architecture



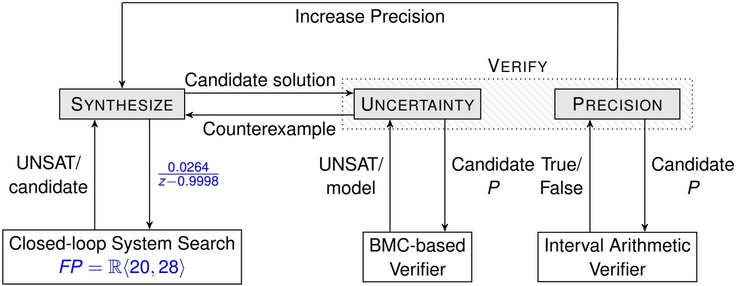
CEGIS Architecture



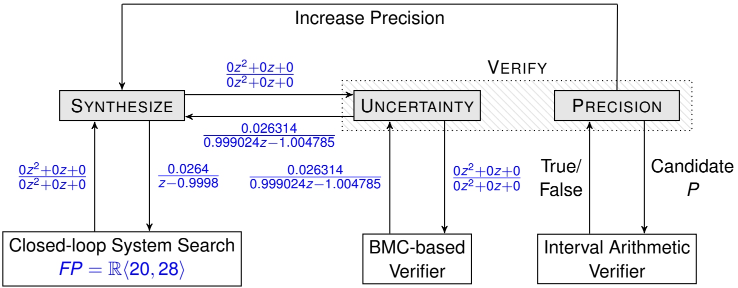
CEGIS Architecture



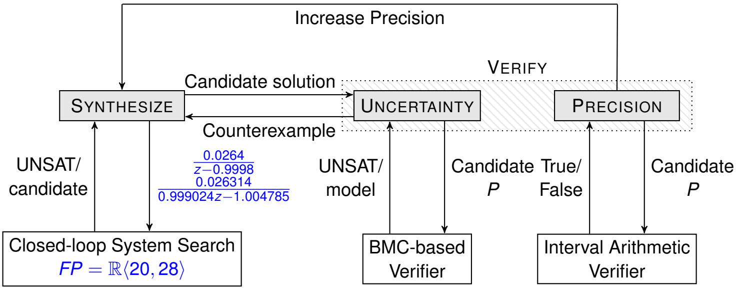
CEGIS Architecture



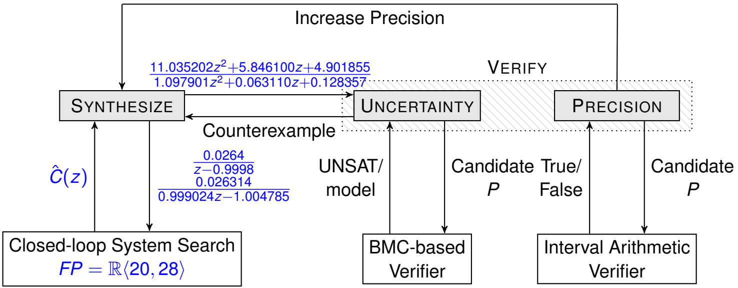
CEGIS Architecture



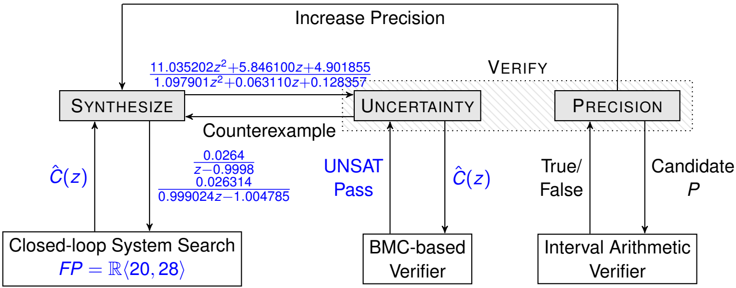
CEGIS Architecture



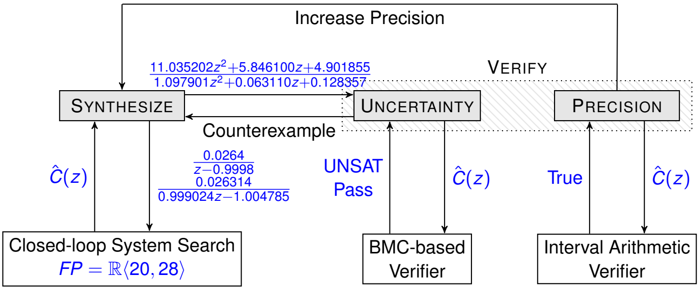
CEGIS Architecture



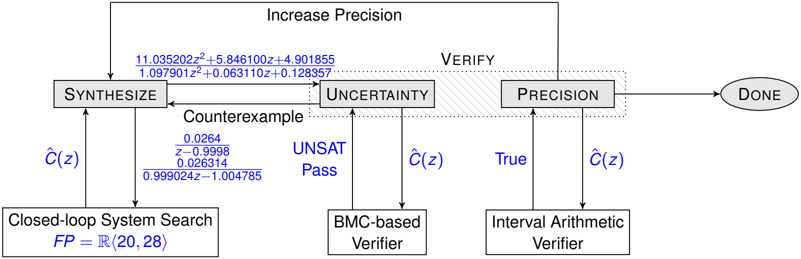
CEGIS Architecture



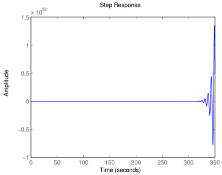
CEGIS Architecture



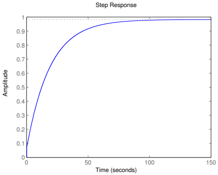
CEGIS Architecture



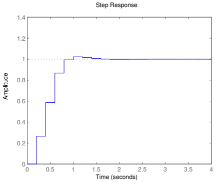
Outcomes: Step Response



(a) Original controller



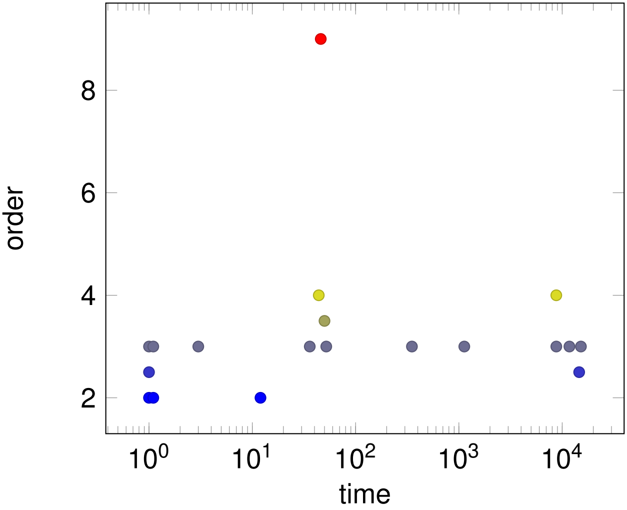
(b) Controller (first step)



(c) Controller (final step)

- original controller was synthesised ignoring FWL effects (becomes unstable over time)

Outcomes: Controller Order



Outcomes: Benchmark

#	Plant	Benchmark	I	F	2-stage	1-stage
1	G_{1a}	CruiseControl02	4	16	**12 s**	67 s
2	G_{1b}	CruiseControl02†	4	16	14600 s	**52 s**
3	G_{2a}	SpgMsDamper	15	16	**52 s**	318 s
4	G_{2b}	SpgMsDamper†	15	16	✗	✗
5	G_{3a}	SatelliteB2	3	7	**36 s**	✗
6	G_{3b}	SatelliteB2†	3	7	✗	**4111 s**
7	G_{3c}	SatelliteC2	3	5	**3 s**	205 s
8	G_{3d}	SatelliteC2†	3	5	**50 s**	1315 s
9	G_4	Cruise	3	7	1 s	1 s
10	G_5	DCMotor	3	7	**1 s**	10 s
11	G_6	DCServomotor	4	11	**46 s**	✗
12	G_7	Doyleetal	4	11	**8769 s**	✗
13	G_8	Helicopter	3	7	**44 s**	✗
14	G_9	Pendulum	3	7	**1 s**	14826 s
15	G_{10}	Suspension	3	7	**1 s**	5 s
16	G_{11}	Tapedriver	3	7	1 s	1 s
17	G_{12a}	a_ST1_IMPL1	16	4	**11748 s**	✗
18	G_{12a}	a_ST1_IMPL2	16	8	**351 s**	✗
19	G_{12a}	a_ST1_IMPL3	16	12	**8772 s**	✗
20	G_{12b}	a_ST2_IMPL1	16	4	**1128 s**	✗
21	G_{12b}	a_ST2_IMPL2	16	8	✗	✗
22	G_{12b}	a_ST2_IMPL3	16	12	**15183 s**	✗
23	G_{12c}	a_ST3_IMPL1	16	4	✗	✗

Table: DSSynth results (✗ = time-out, † = uncertainty)

Conclusions

- We have presented a method for automatically synthesizing stable and sound controllers, implemented in a tool called DSSynth.
- DSSynth marks the first use of the CEGIS that handles plants with uncertain models and FWL effects over the digital controller.
- Our experimental results show that DSSynth is able to synthesize stable controllers for most benchmarks within a reasonable amount of time.
- Future work will extend this CEGIS-based approach to LTI systems with state space safety specifications.