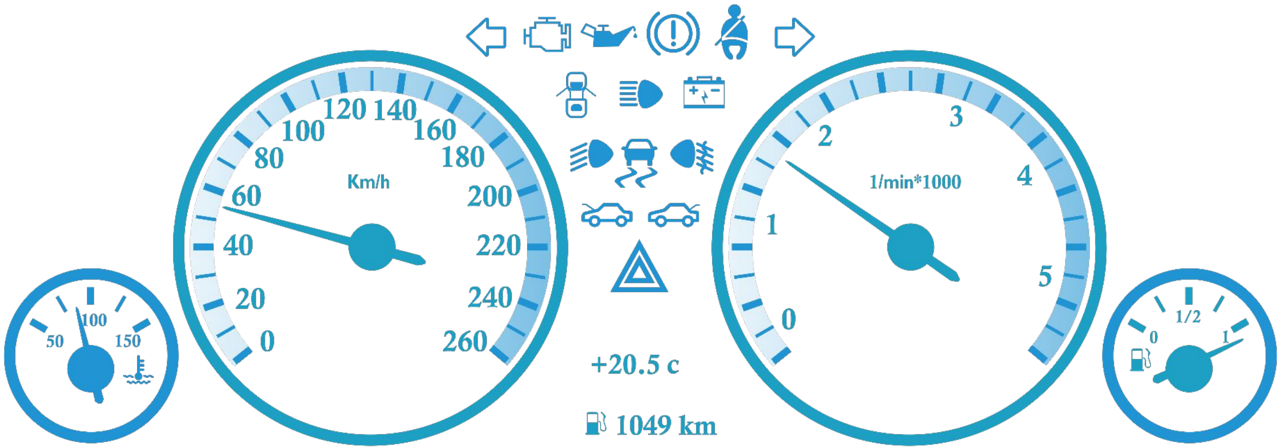
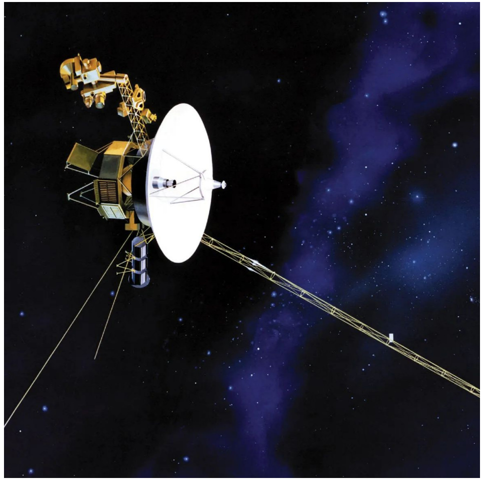


Detecting Bugs in Cars

finding needles in a haystack

Automated Model-in-the-Loop Testing of Continuous Controllers Using Search

Velocity

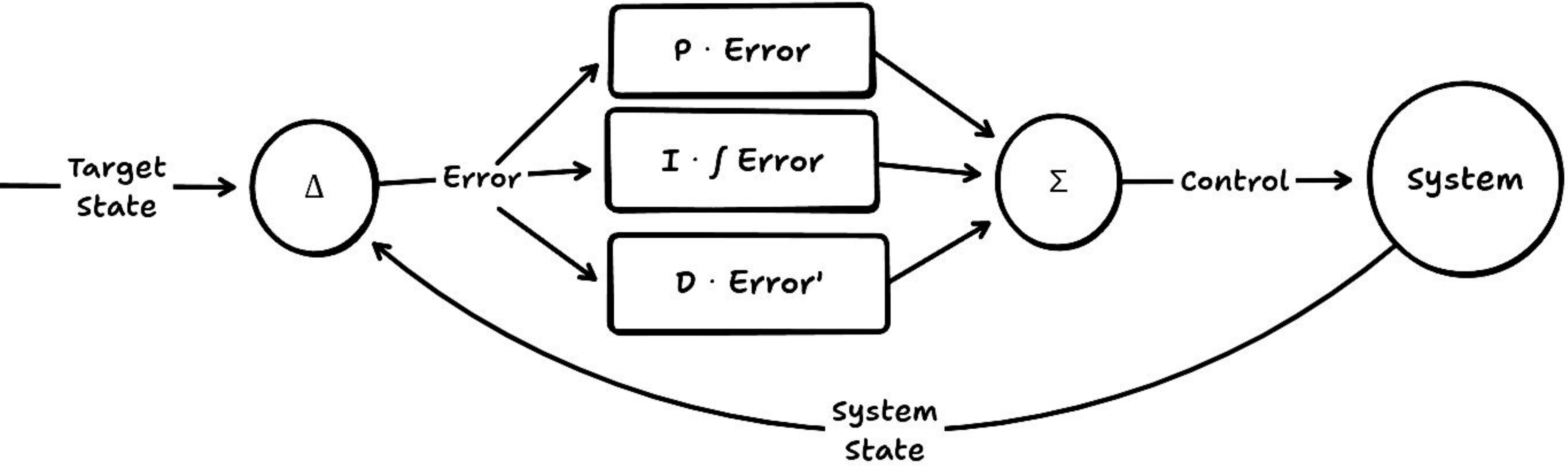


Orientation

Positioning

Error = Desired State - Actual State

PID Controllers in a Nutshell



P, I, D are the tuning parameters of the controller.

Model-in-the-Loop

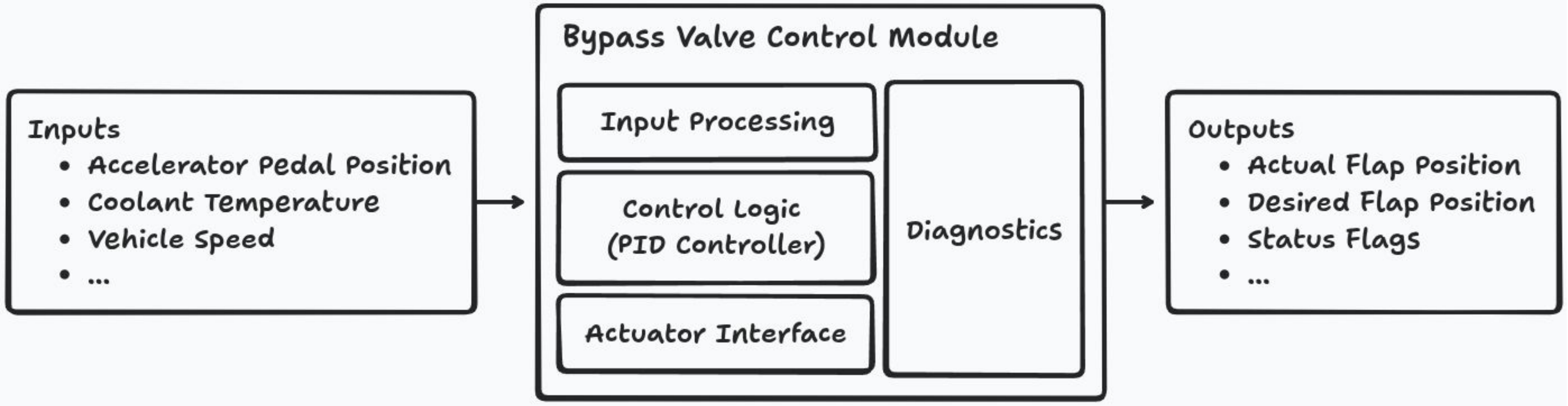
On-Road

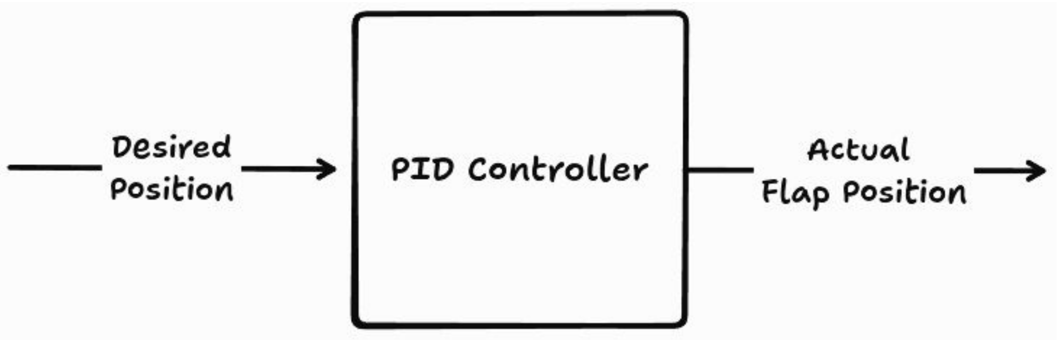
- Expensive - physical prototypes, facilities
- Time taking - setup, logistics, iterations
- Dangerous - real parts, real risks
- Non exhaustive - limited feasible tests

Model-in-the-Loop (MiL)

- Cost-effective - no physical hardware
- Fast - rapid iteration and automation
- Safe - no physical risks
- Comprehensive - virtual edge case tests

Delphi


Bypass Valve Control Module
Inputs
Accelerator Pedal Position
Coolant Temperature
Vehicle Speed
...
Input Processing
Control Logic
(PID Controller)
Actuator Interface
Diagnostics
Outputs
Actual Flap Position
Desired Flap Position
Status Flags
...



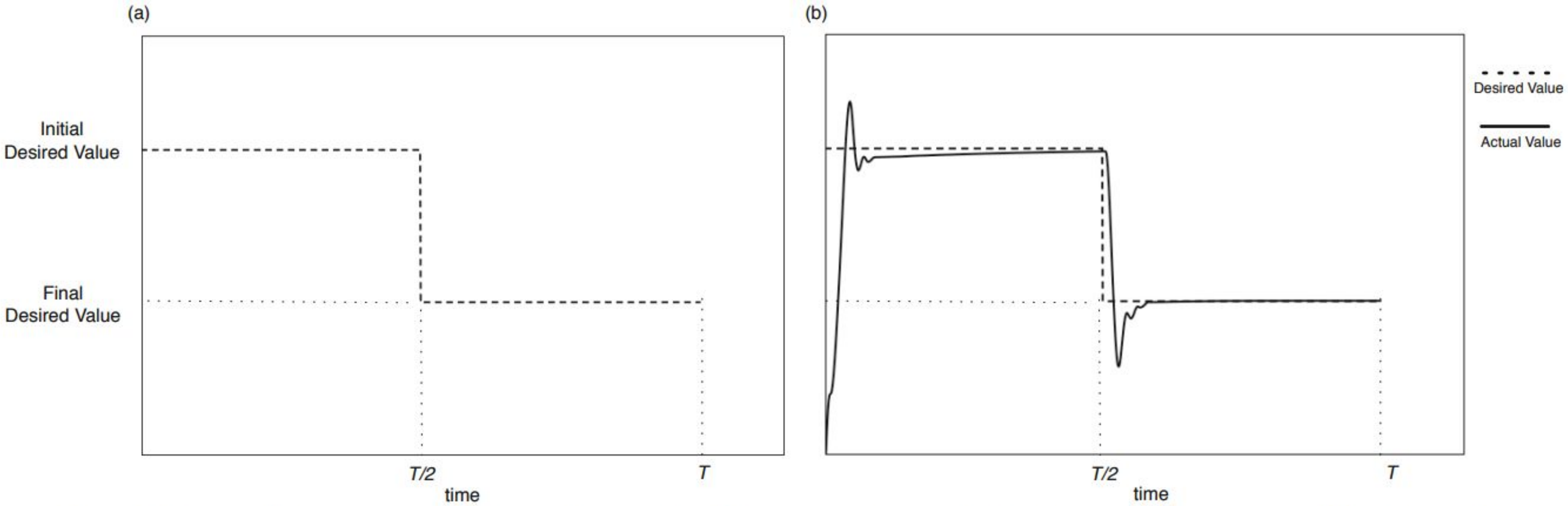
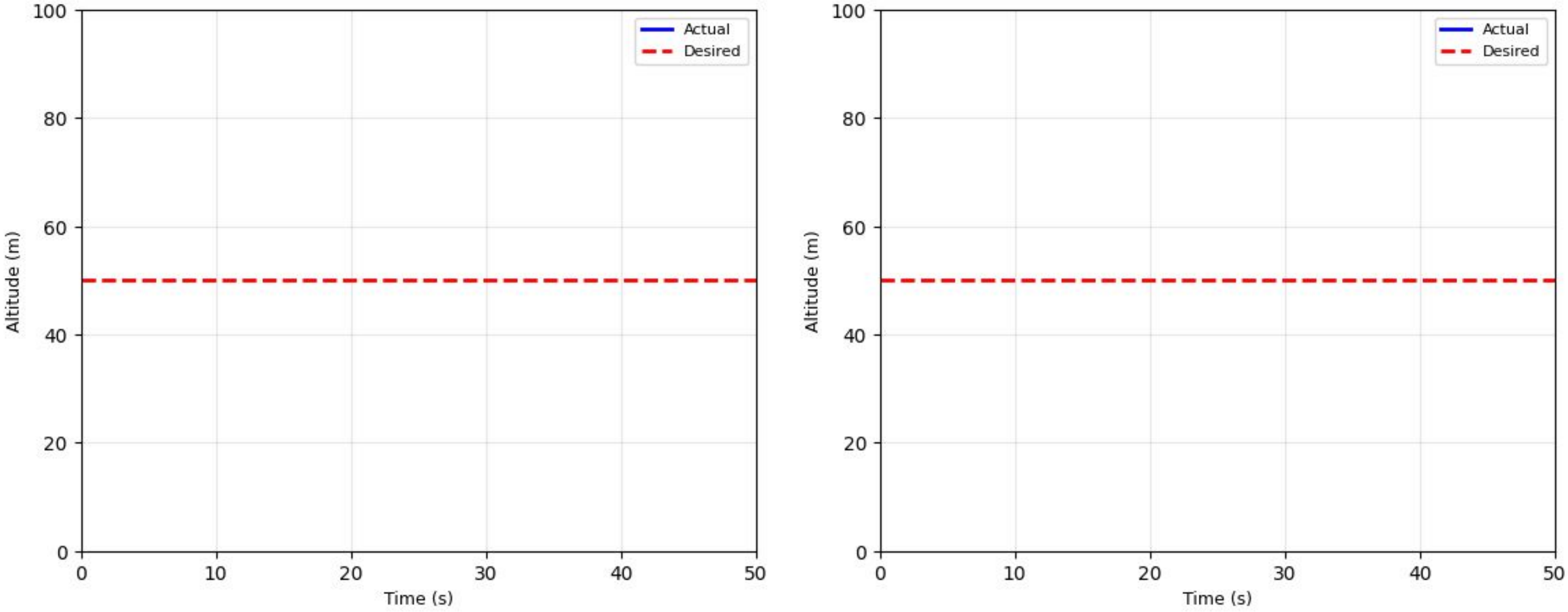
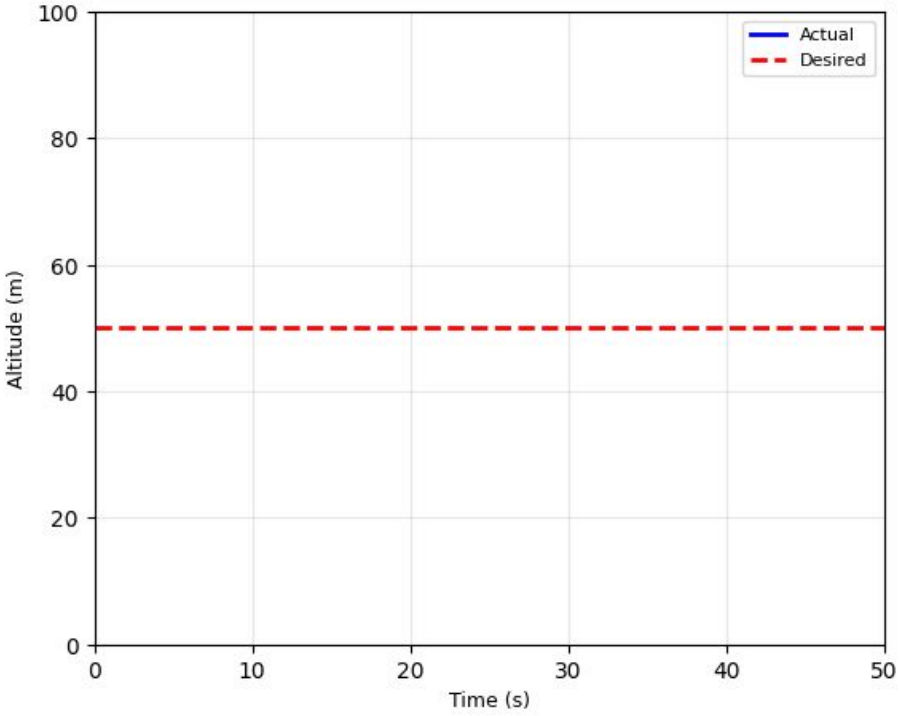


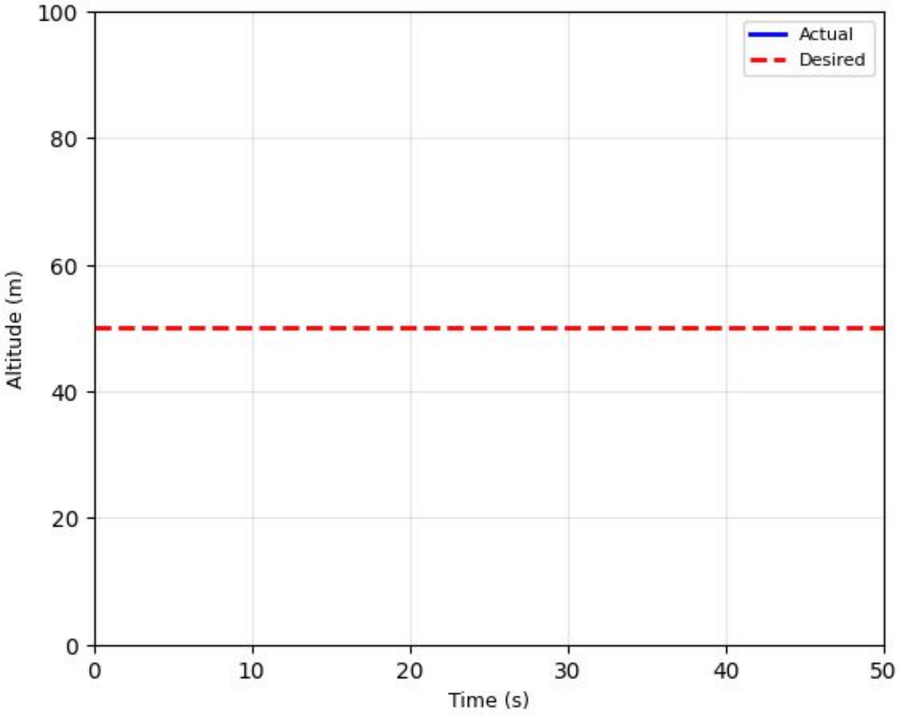
Fig. 3. Controller input step functions: (a) Step function. (b) Output of the controller (actual) given the input step function (desired).

Error = Desired Position - Actual Position

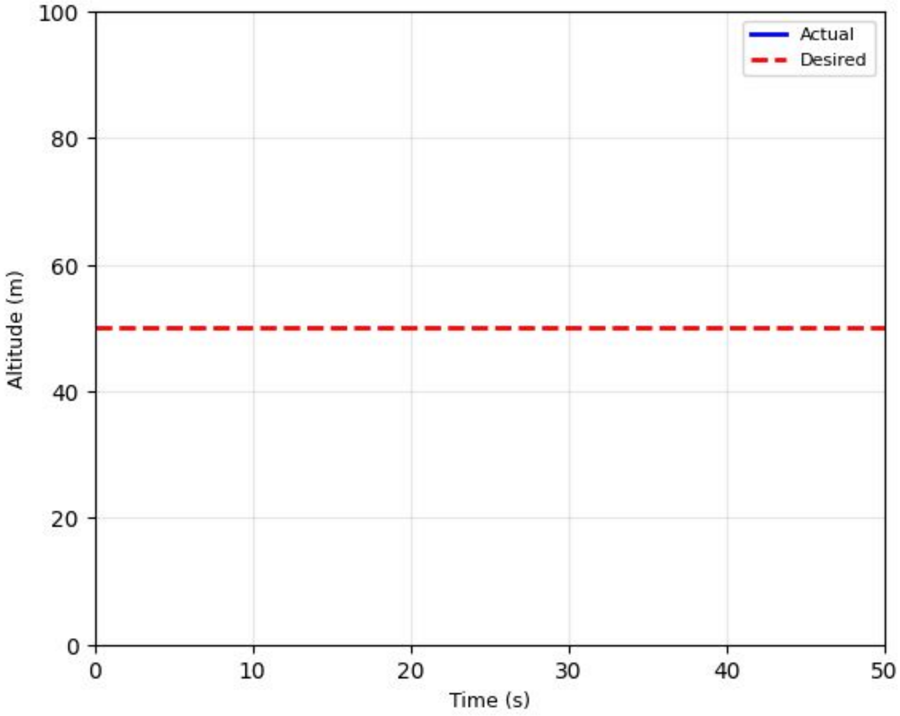


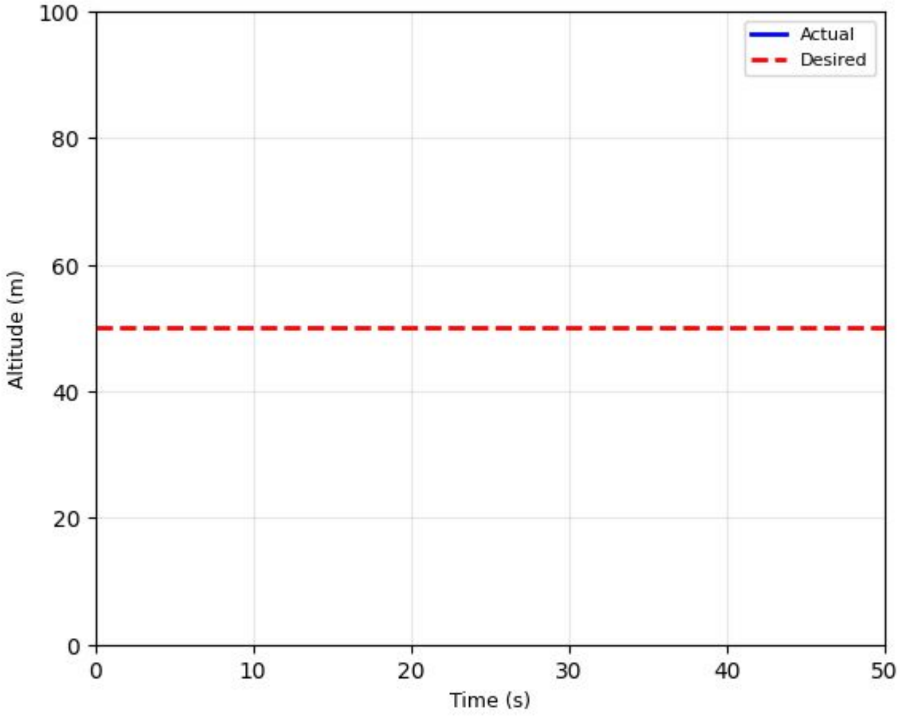
*disregard label names





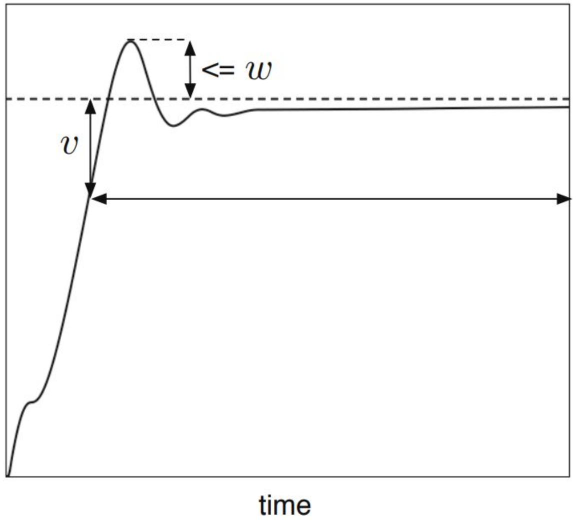
*disregard label names



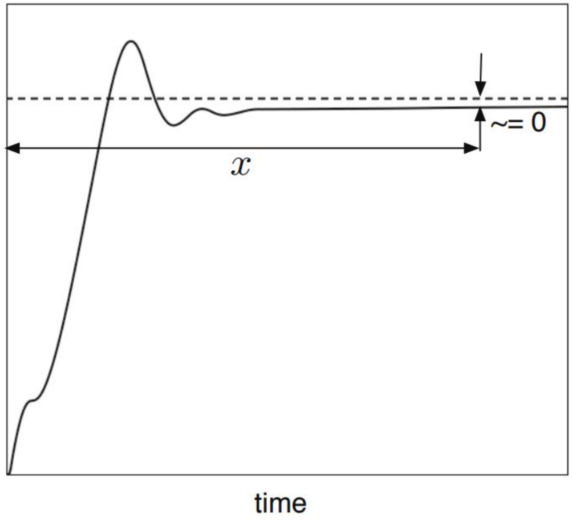


*disregard label names

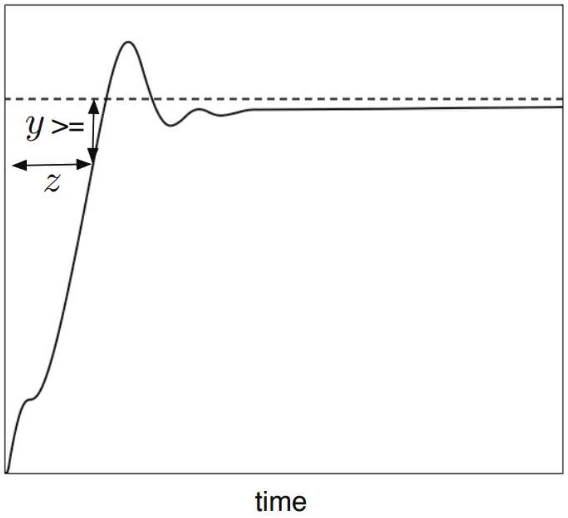
Overshoot (**Smoothness**)



Convergence (**Liveness**)

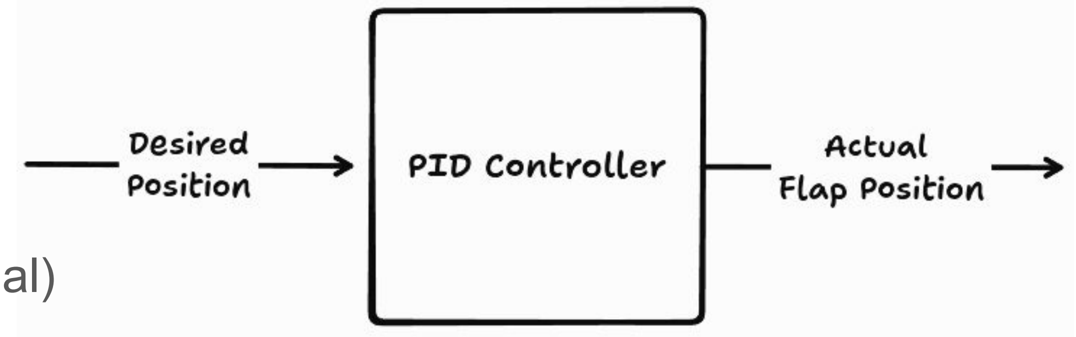


Responsiveness



Representation

(Desired, Actual)



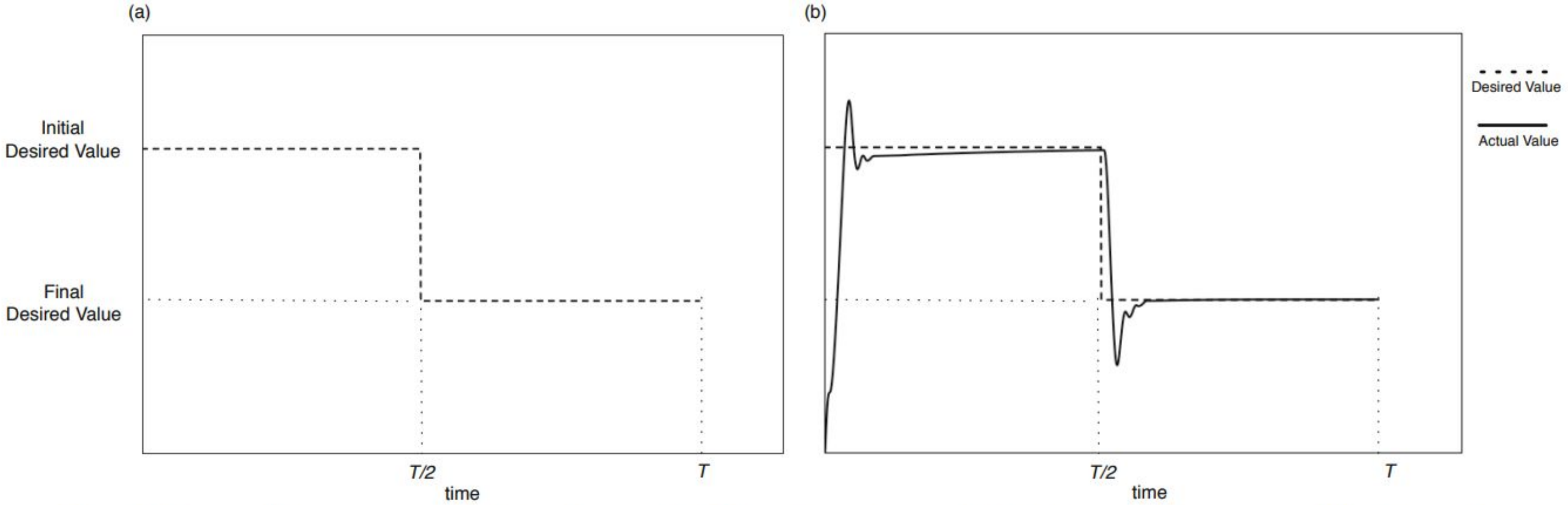


Fig. 3. Controller input step functions: (a) Step function. (b) Output of the controller (actual) given the input step function (desired).

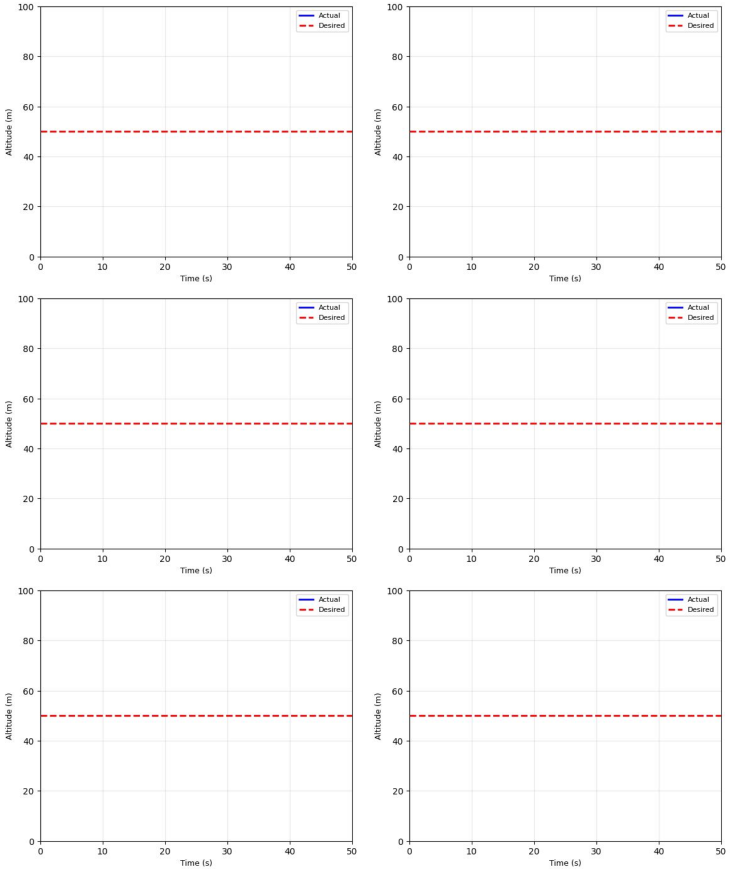
Fitness O(Desired, Actual)

Overshoot
(**Smoothness**)

Convergence
(**Liveness**)

Responsiveness

Abnormal Normal



Operators

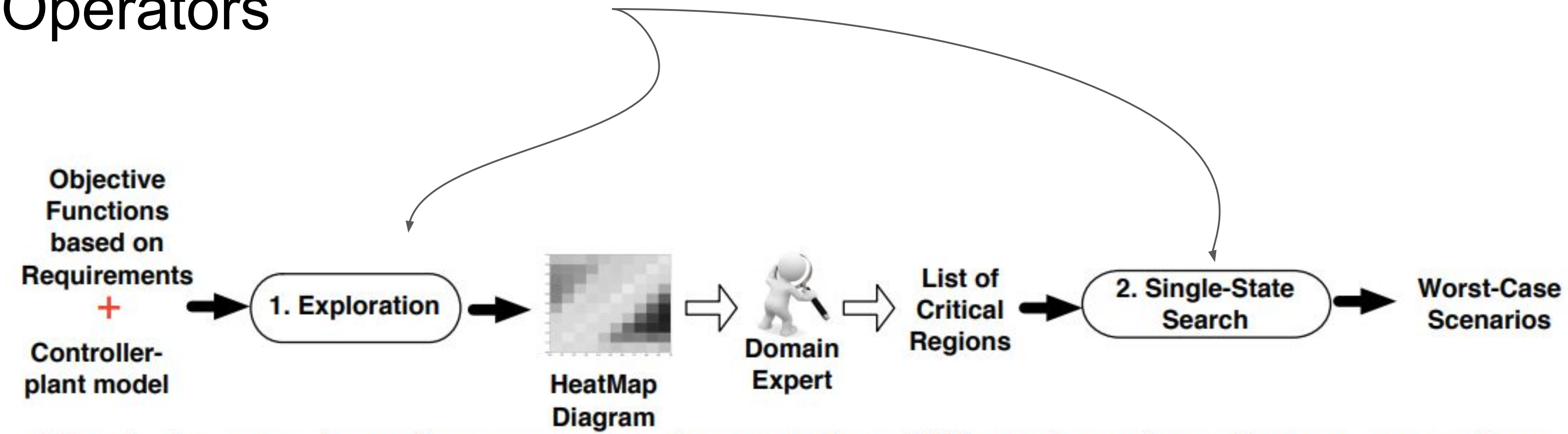
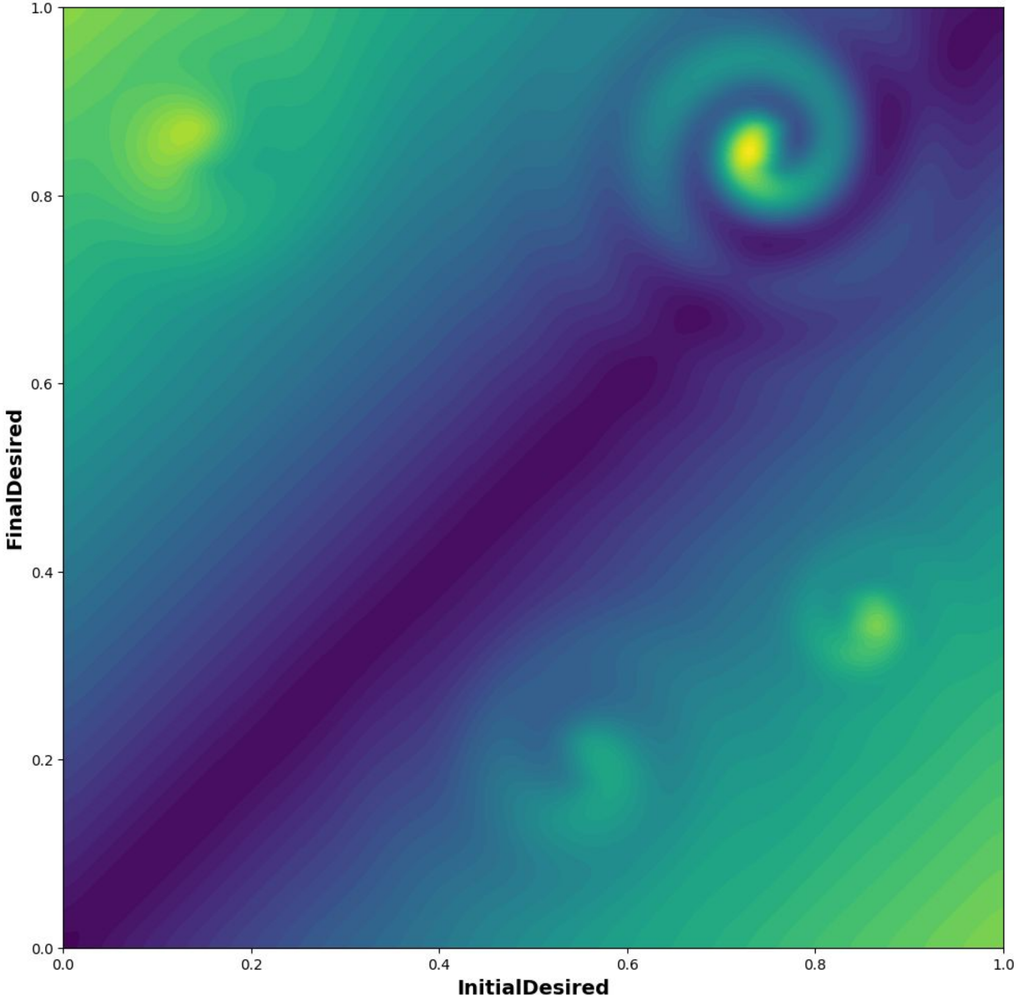


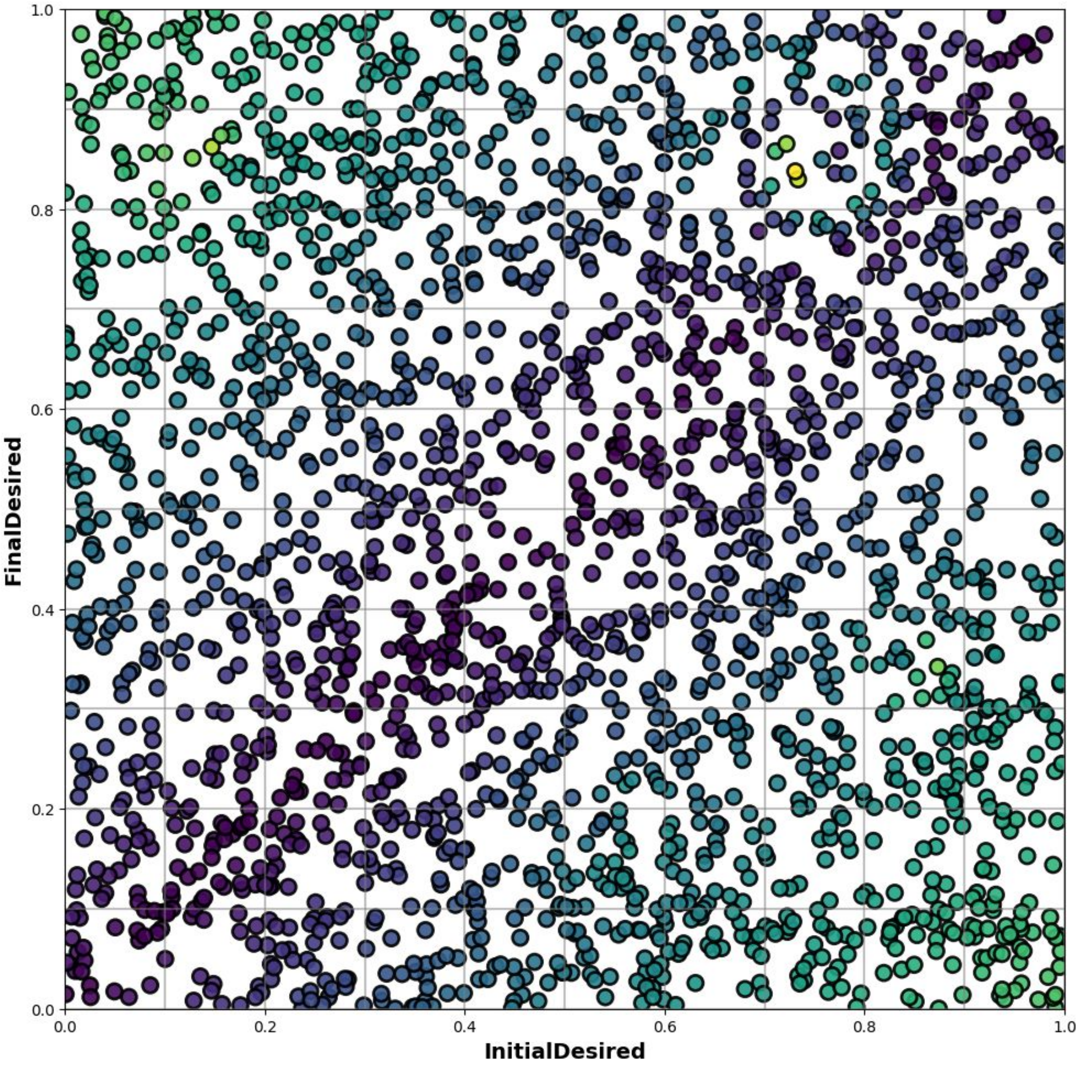
Fig. 4. An overview of our automated approach to MiL testing of continuous controllers.



FinalDesired
InitialDesired
0.0
0.2
0.4
0.6
0.8
1.0

Algorithm. RANDOMEXPLORATION
Input: A controller-plant model M with input search space S.
An objective function O. An observation time T.
Output: An overview diagram (HeatMap).

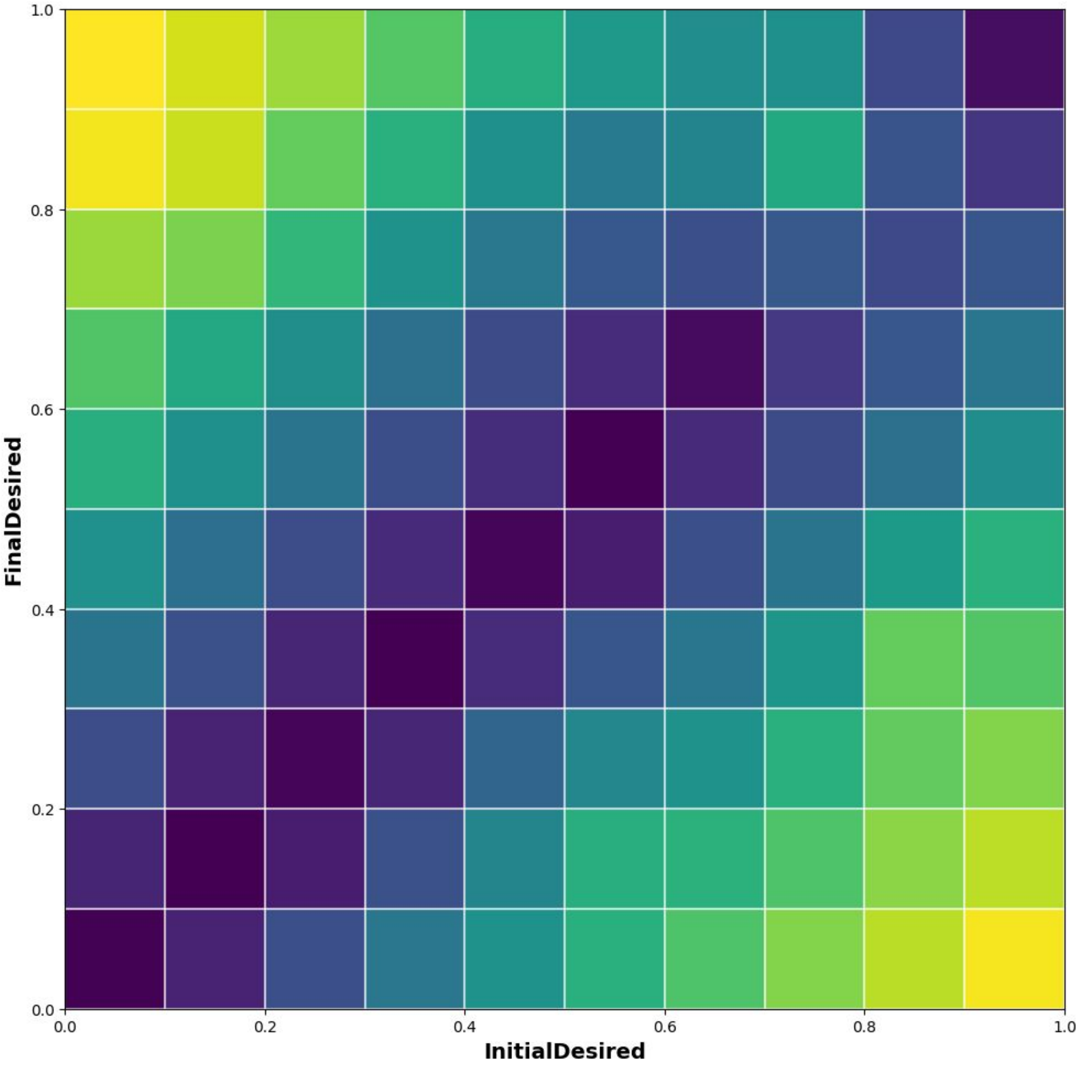
1. Partition S into equal sub-regions
2. Let $P = \{\}$
3. **repeat**
4. Let $p = (\texttt{Initial Desired}, \texttt{Final Desired})$ be a random point in S
5. Let `Desired` be a step function generated based on p and T
6. Run M with the `Desired` input to obtain the `Actual` output
7. $o = O(\texttt{Desired}, \texttt{Actual})$
8. $P = \{(p, o)\} \cup P$
9. **until** there are at least N points in each region of S **do**
10. Create a HeatMap diagram based on P



(1) Confidence about the behaviors over
the dark shaded regions.

(2) The diagrams facilitate discovery of anomalies

(3) A sharp contrast may indicate abnormal behavior



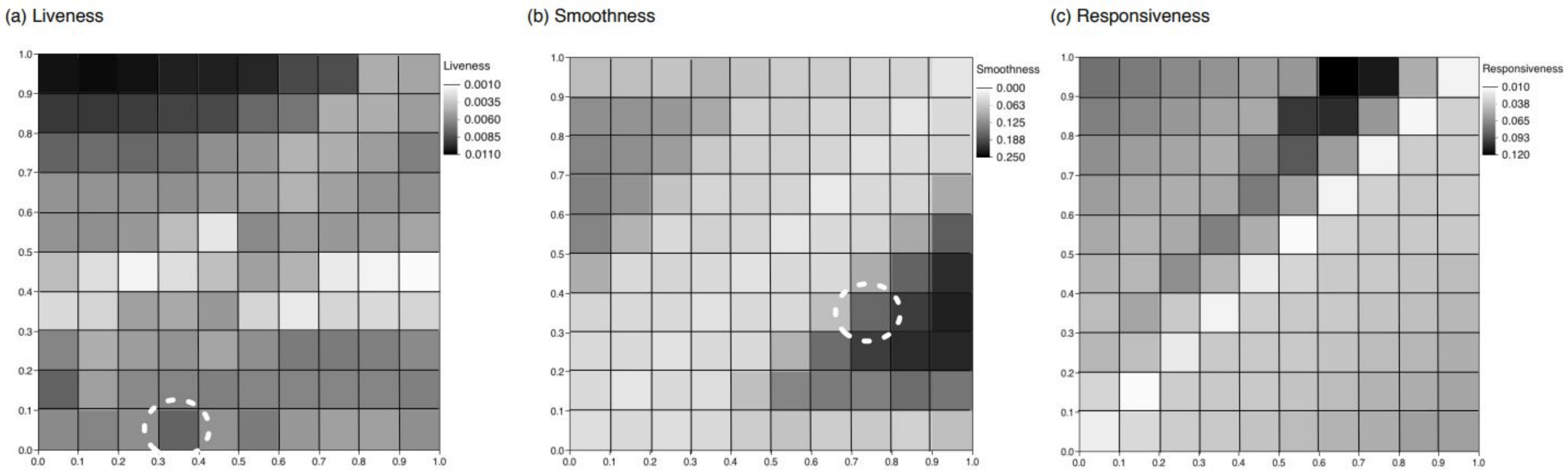


Fig. 7. HeatMap diagrams generated for our case study for the Liveness (a), Smoothness (b) and Responsiveness (c) requirements.

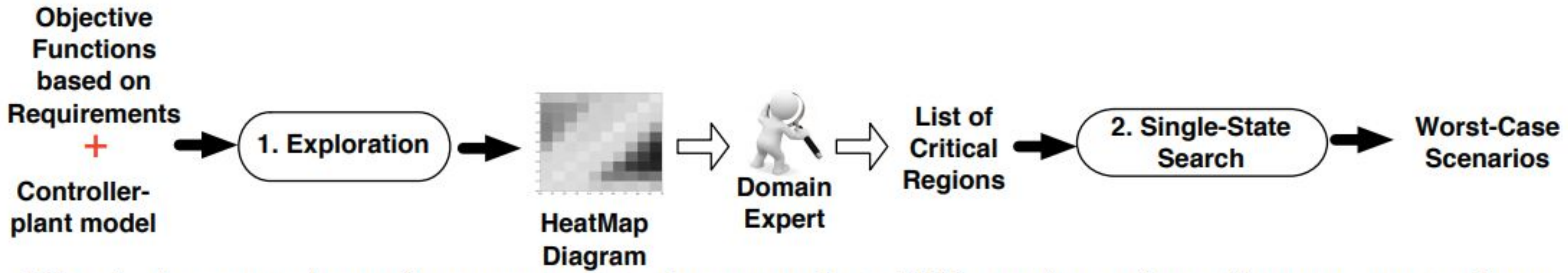
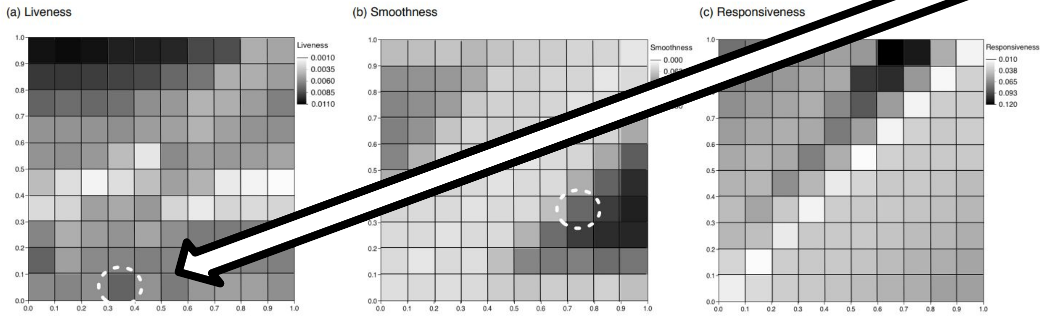


Fig. 4. An overview of our automated approach to MiL testing of continuous controllers.



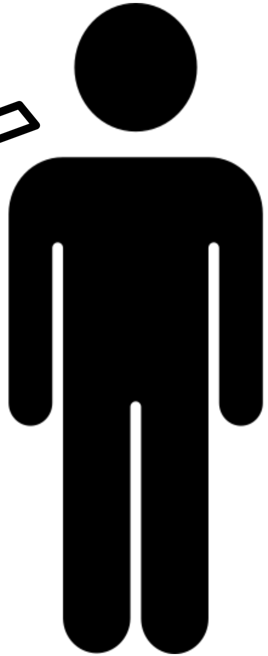


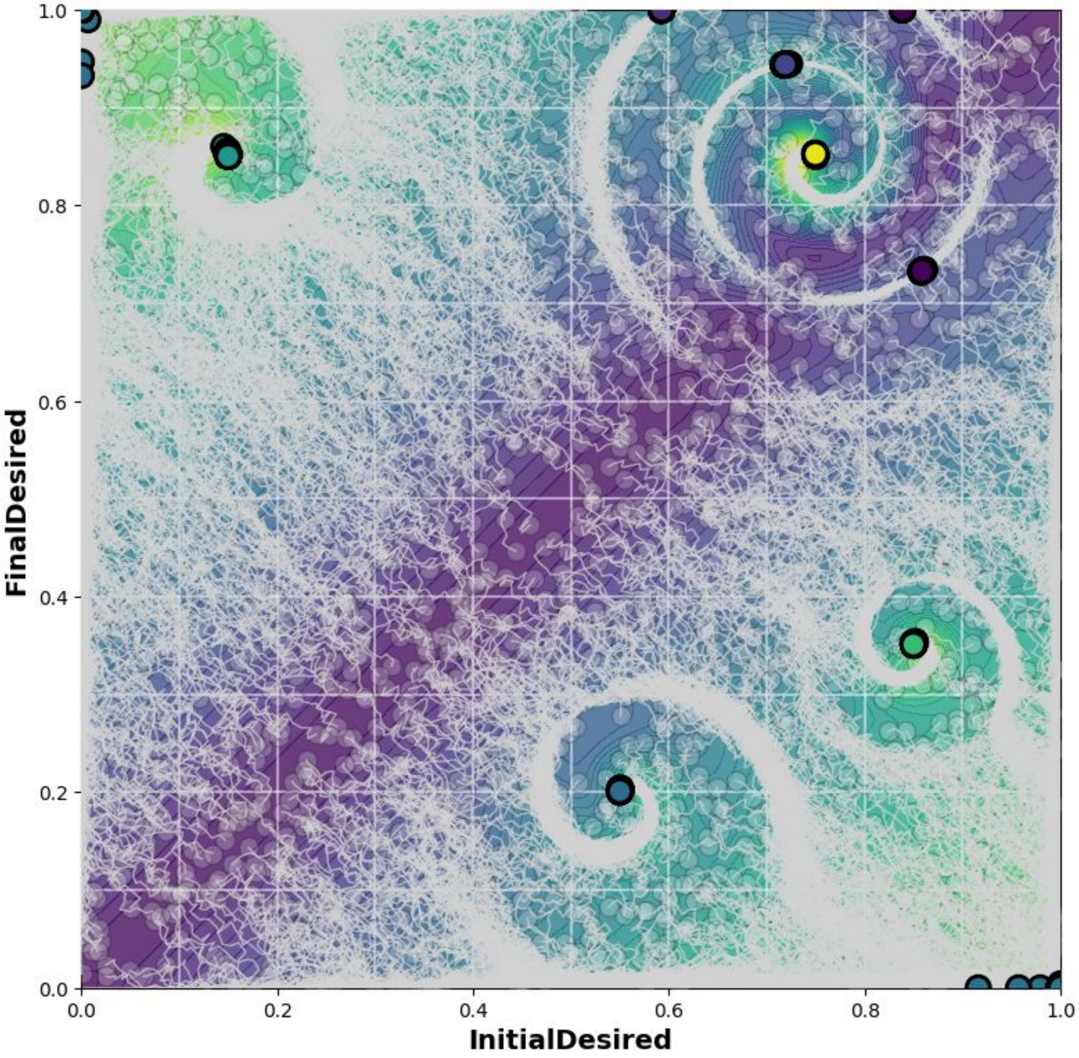
Fig. 7. HeatMap diagrams generated for our case study for the Liveness (a), Smoothness (b) and Responsiveness (c) requirements.

Algorithm. SINGLESTATESEARCH

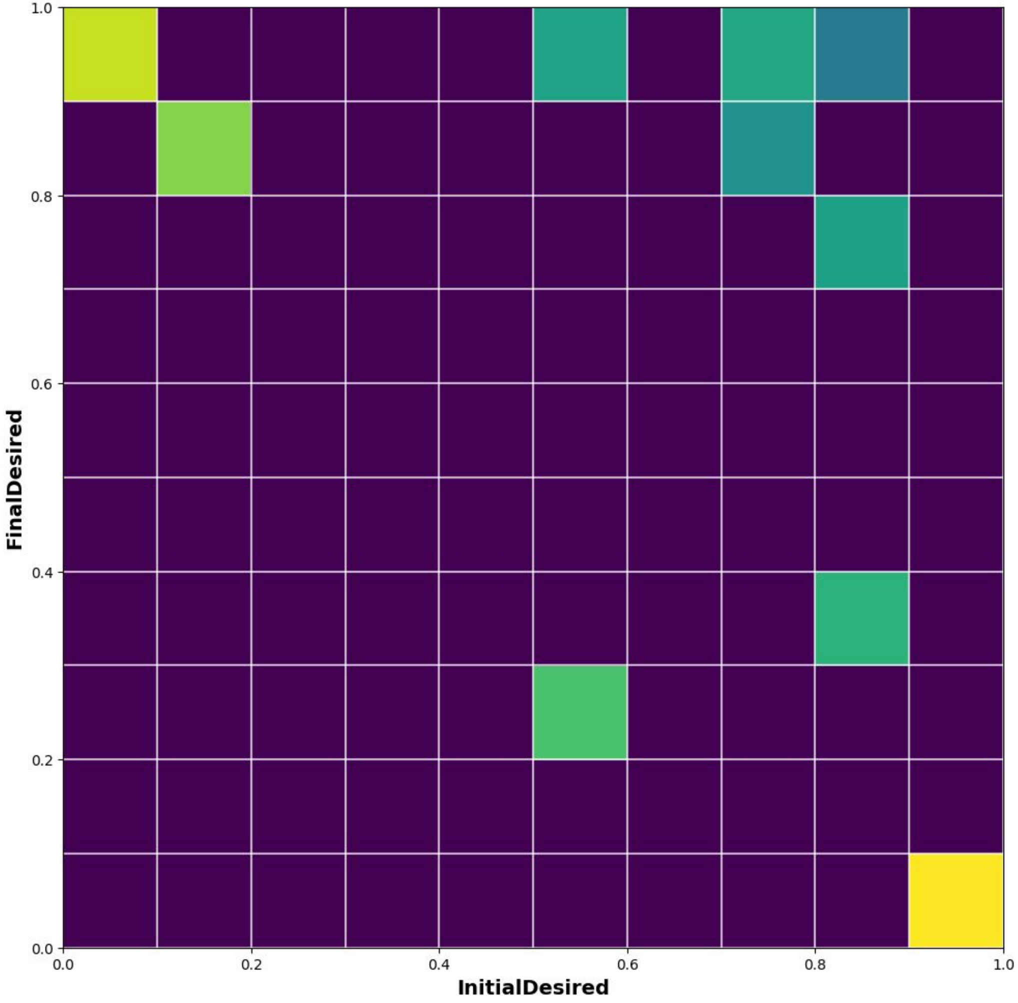
Input: A controller-plant model M. A region r.
The set P computed by the algorithm in Figure 5(a).
An objective function O.

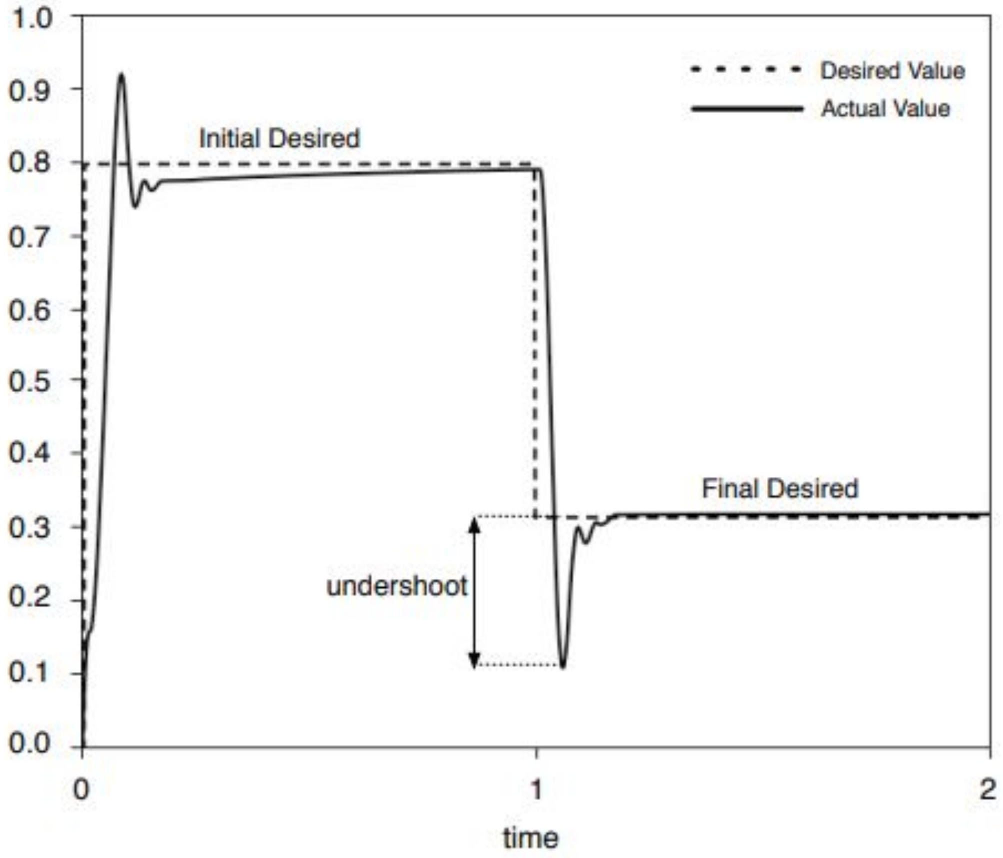
Output: The worst case scenario $testCase$.

$P' = \{(p, o) \in P \mid p \in r\}$
Let $(p, o) \in P'$ s.t. for all $(p', o') \in P'$, we have $o \geq o'$
$worstFound = o$
for K iterations **:**
  Let `Desired` be a step function generated by p
  Run M with the `Desired` input to obtain
  the `Actual` output
  $v = O(\texttt{Desired}, \texttt{Actual})$
  if $v > worstFound$ **:**
    $worstFound = v$
    $testCase = p$
  $p = Tweak(p)$
return $testCase$

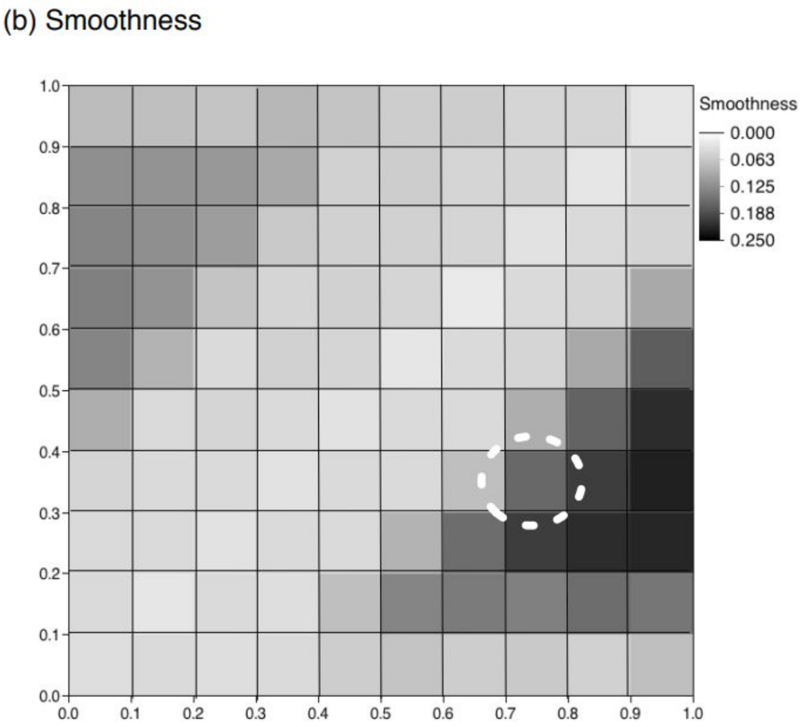


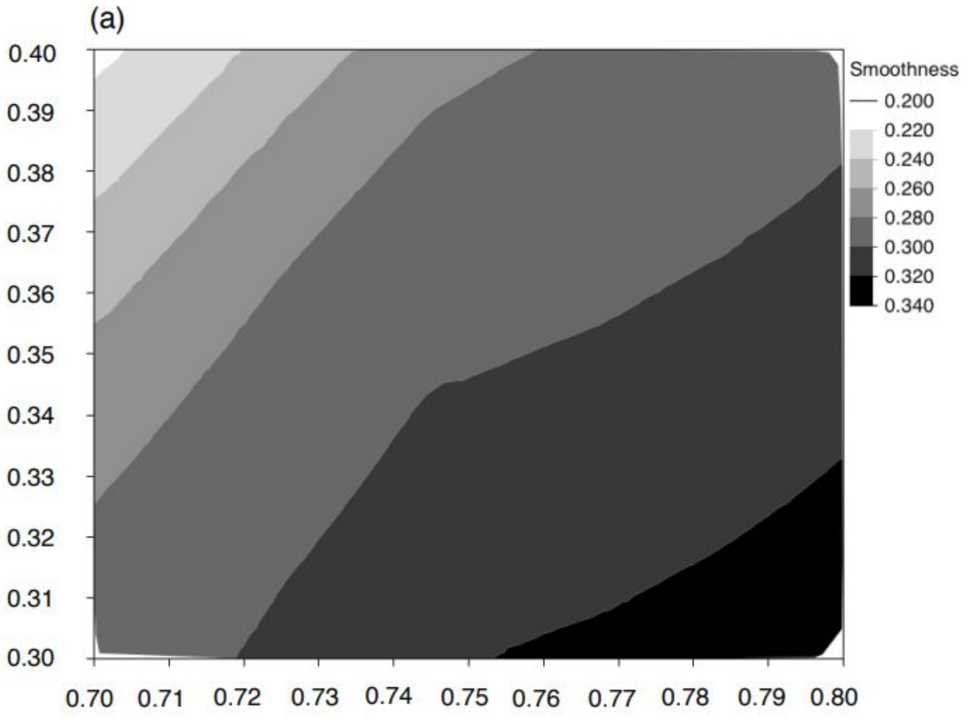
- 11 worst case scenarios
- Requirements violations
- Controller model error
- Plant model error

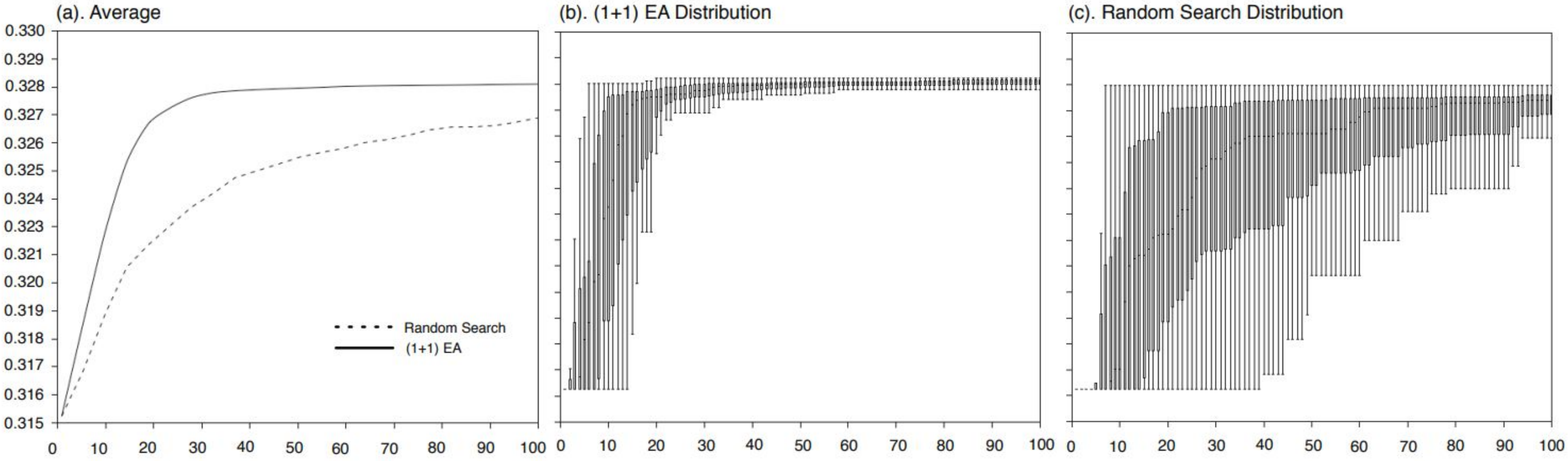




1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
time
Desired Value
Actual Value
Initial Desired
Final Desired
undershoot

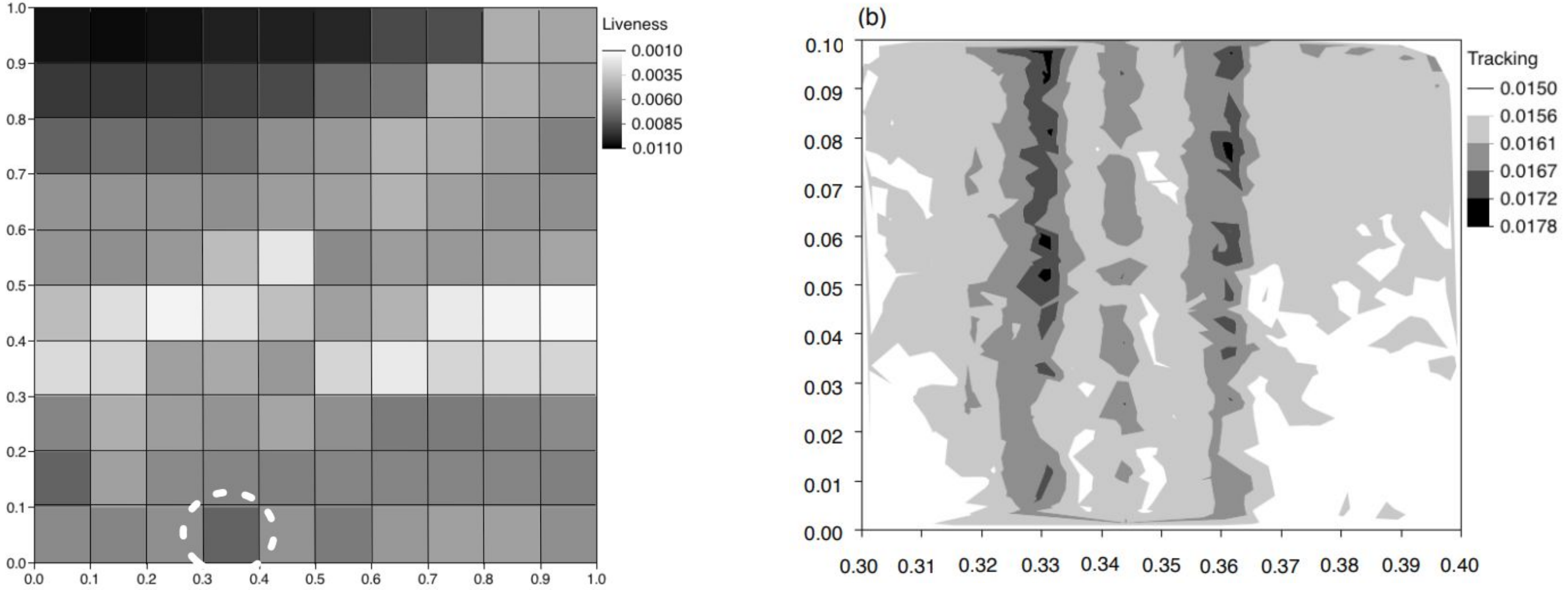

(b) Smoothness
Smoothness
0.000
0.063
0.125
0.188
0.250
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

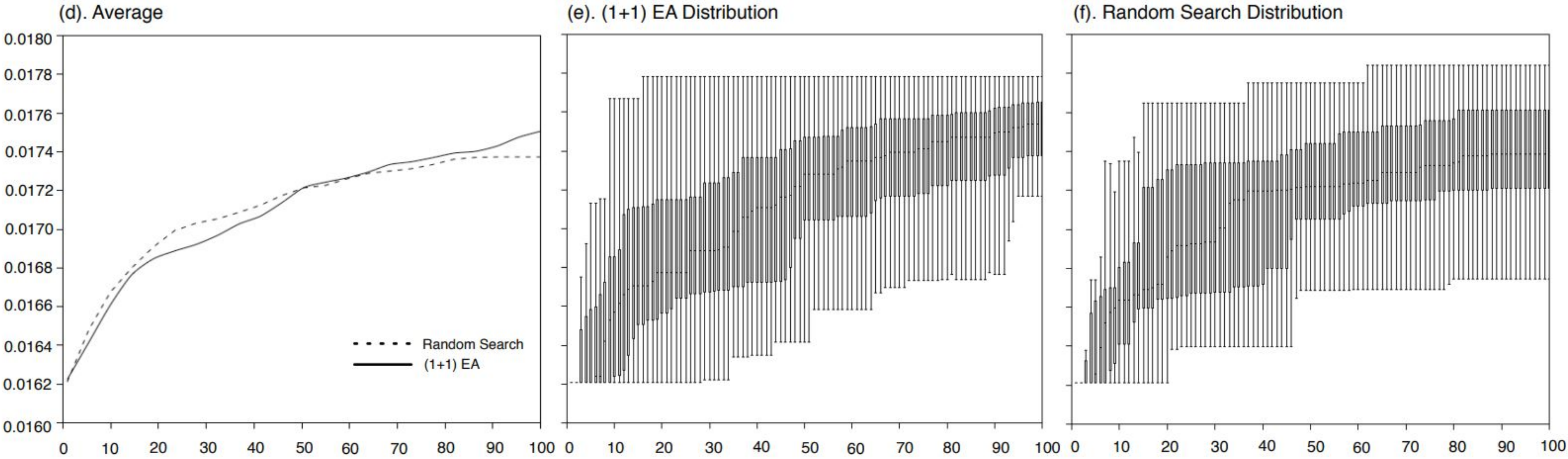

(a)
Smoothness
0.200
0.220
0.240
0.260
0.280
0.300
0.320
0.340
0.30
0.31
0.32
0.33
0.34
0.35
0.36
0.37
0.38
0.39
0.40
0.70
0.71
0.72
0.73
0.74
0.75
0.76
0.77
0.78
0.79
0.80



(a). Average
(b). (1+1) EA Distribution
(c). Random Search Distribution
Random Search
(1+1) EA
0.330
0.329
0.328
0.327
0.326
0.325
0.324
0.323
0.321
0.320
0.319
0.318
0.317
0.316
0.315
0
10
20
30
40
50
60
70
80
90
100

(a) Liveness


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Liveness
0.0010
0.0035
0.0060
0.0085
0.0110
(b)
0.10
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0.30 0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.40
Tracking
0.0150
0.0156
0.0161
0.0167
0.0172
0.0178



(d). Average
(e). (1+1) EA Distribution
(f). Random Search Distribution
0.0180
0.0178
0.0176
0.0174
0.0172
0.0170
0.0168
0.0166
0.0164
0.0162
0.0160
Random Search
(1+1) EA
0
10
20
30
40
50
60
70
80
90
100

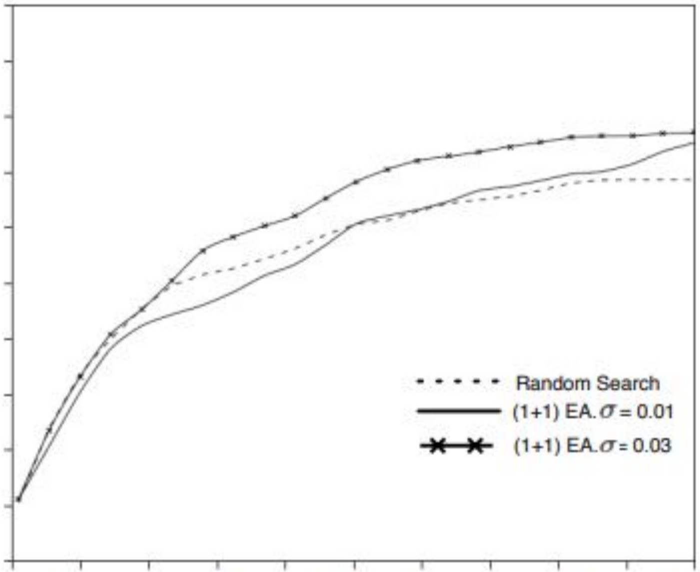
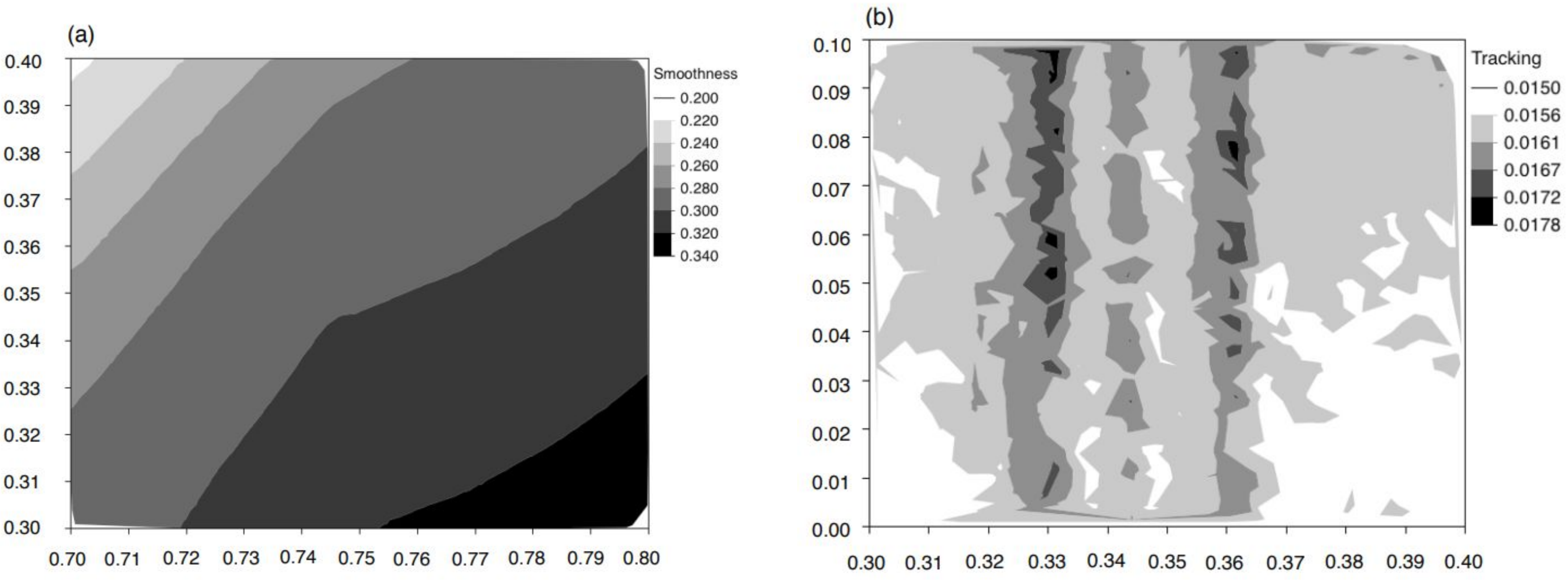


Fig. 10. Comparing average values for (1+1) EA with $\sigma = 0.01$, (1+1) EA with $\sigma = 0.03$, and random search for the region in Figure 7(a)

TAKEAWAYS


(a)
Smoothness
0.200
0.220
0.240
0.260
0.280
0.300
0.320
0.340
0.40
0.39
0.38
0.37
0.36
0.35
0.34
0.33
0.32
0.31
0.30
0.70
0.71
0.72
0.73
0.74
0.75
0.76
0.77
0.78
0.79
0.80
(b)
Tracking
0.0150
0.0156
0.0161
0.0167
0.0172
0.0178
0.10
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0.30
0.31
0.32
0.33
0.34
0.35
0.36
0.37
0.38
0.39
0.40

No one tool for the job

Random Search:

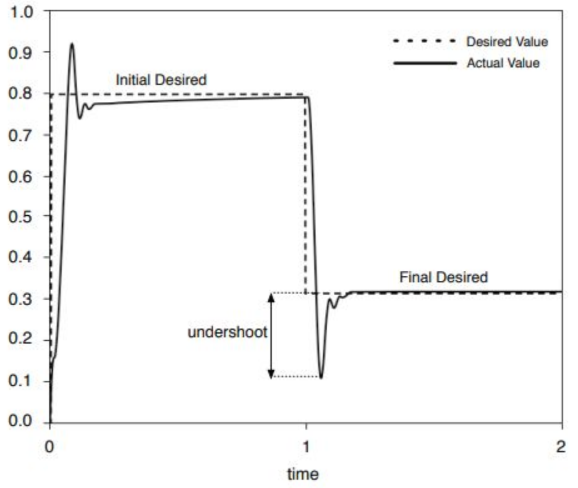
- Terrible for finding exact peaks
- Effective at finding which hills are worth climbing

Local Search:

- Terrible for global search (stuck in local peaks)
- Effective at finding the precise peak in a specific area

Turn requirements into numbers

- Translate qualitative into quantitative
- Frame as an optimization problem



Did it overshoot?

Qualitative	Quantitative
Yes	Os = max|error(t)| = 0.2

***Oracle problem**: automatically determining if a program's output is correct for a given input*

CRITICISMS

Novelty and Significance

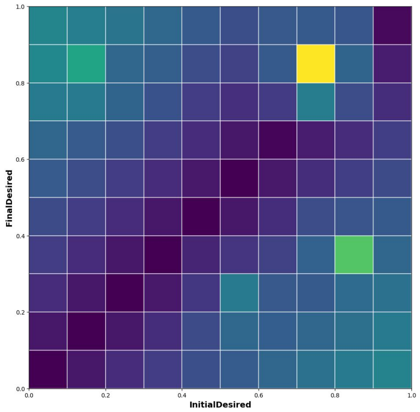
- The general concept had been already established by prior work (eg, Lindlar et al., 2009)

- Two-phase workflow
- Heatmap to convey the information
- Three generic controller metrics (reusable for this class of problem)

- Automated and more effective alternative to manual testing.
- Demonstration of requirements into objective functions.

Soundness and Verifiability

- Simplified 2D search space (fixed all other system inputs).
    - How well would the methodology scale as more parameters are added?
- Reliability of humans in the loop



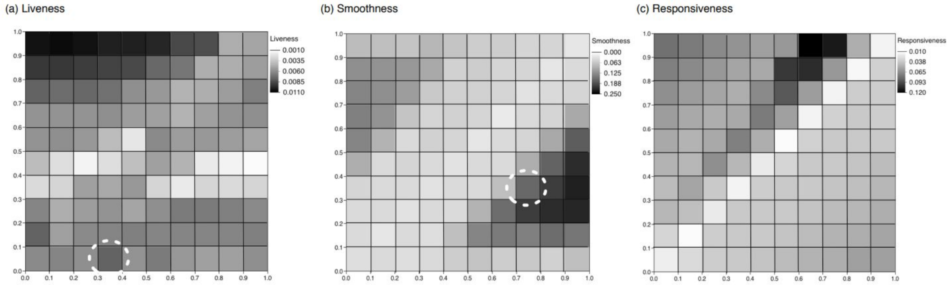
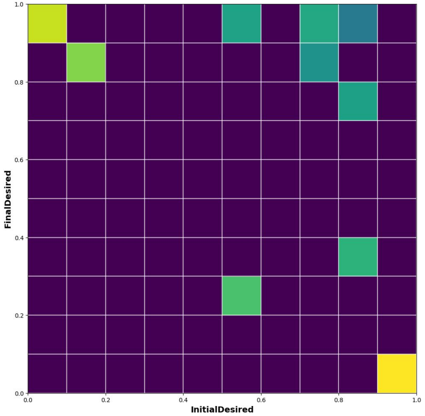
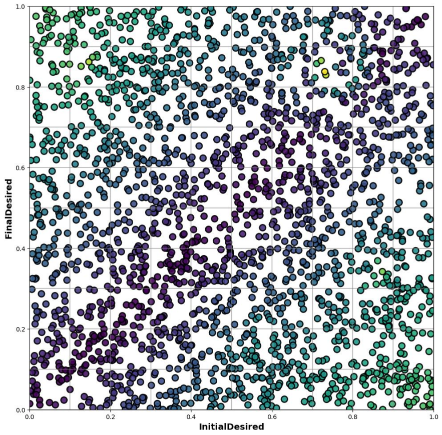
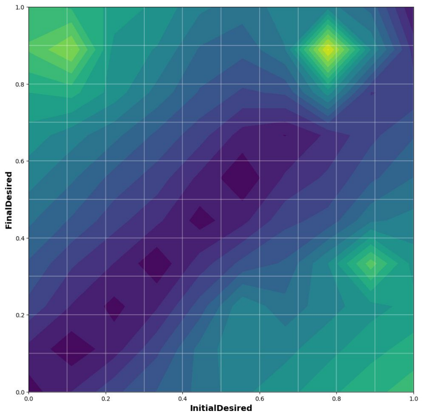


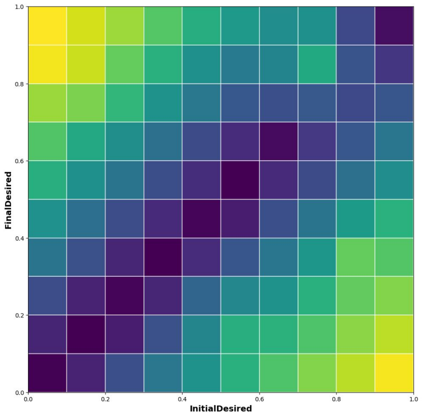
Fig. 7. HeatMap diagrams generated for our case study for the Liveness (a), Smoothness (b) and Responsiveness (c) requirements.

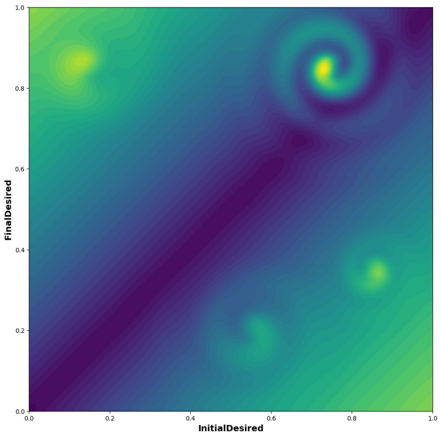


Soundness and Verifiability





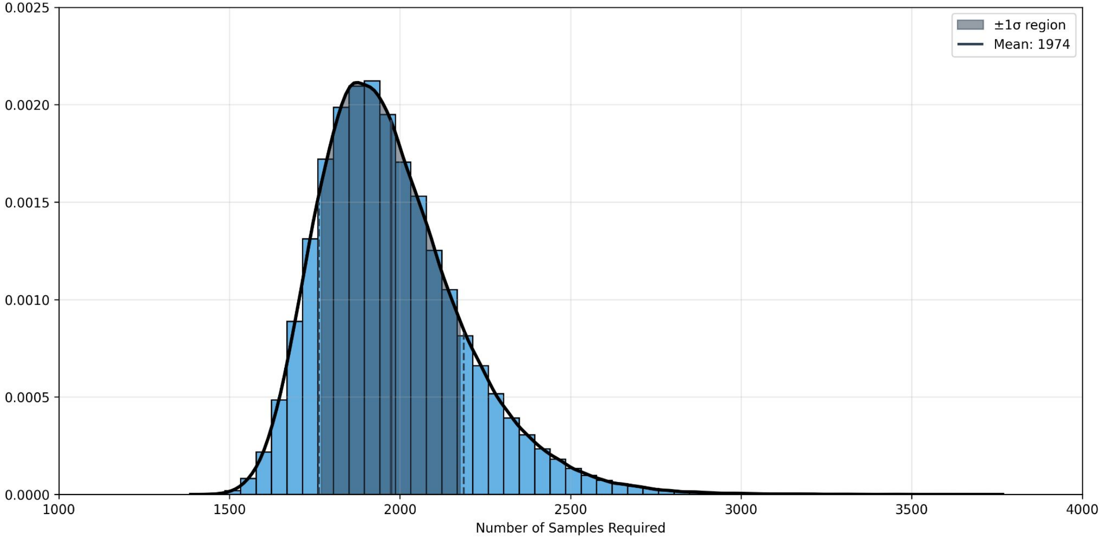




Soundness and Verifiability

Algorithm. RANDOMEXPLORATION
Input: A controller-plant model M with input search space S.
An objective function O. An observation time T.
Output: An overview diagram (HeatMap).

1. Partition S into equal sub-regions
2. Let $P = \{\}$
3. **repeat**
4. Let $p = (\texttt{Initial Desired}, \texttt{Final Desired})$ be a random point in S
5. Let `Desired` be a step function generated based on p and T
6. Run M with the `Desired` input to obtain the `Actual` output
7. $o = O(\texttt{Desired}, \texttt{Actual})$
8. $P = \{(p, o)\} \cup P$
9. **until** there are at least N points in each region of S **do**
10. Create a HeatMap diagram based on P



200k Monte Carlo Runs
E[N]: 1974
σ(N): 212

Min: 1442
Max: 3711

“We decided to generate around 1000 points in S during the exploration step.”

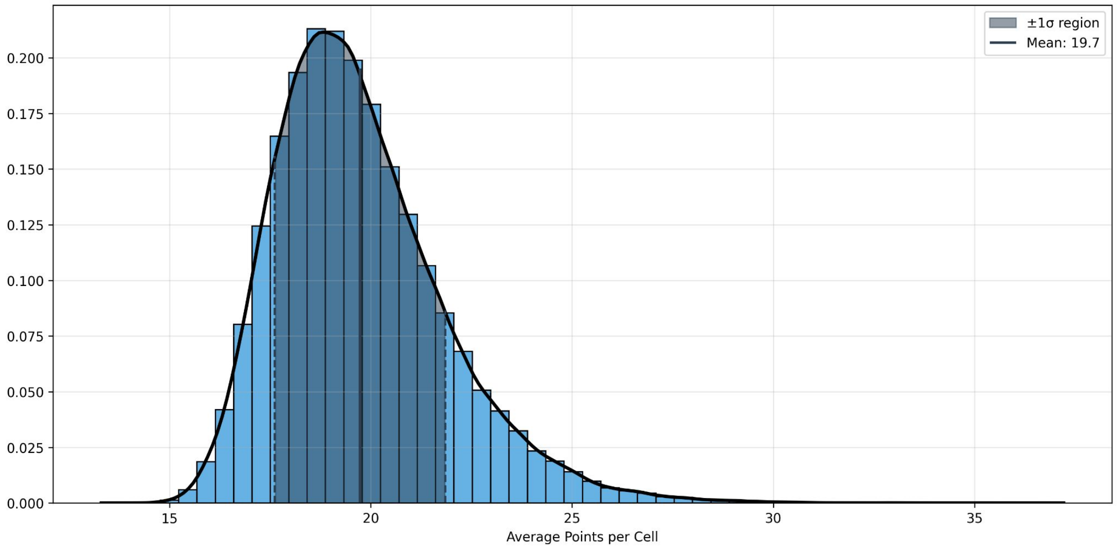
Soundness and Verifiability

Algorithm. RANDOMEXPLORATION
Input: A controller-plant model M with input search space S.
An objective function O. An observation time T.
Output: An overview diagram (HeatMap).

1. Partition S into equal sub-regions
2. Let $P = \{\}$
3. **repeat**
4. Let $p = (\texttt{Initial Desired}, \texttt{Final Desired})$ be a random point in S
5. Let `Desired` be a step function generated based on p and T
6. Run M with the `Desired` input to obtain the `Actual` output
7. $o = O(\texttt{Desired}, \texttt{Actual})$
8. $P = \{(p, o)\} \cup P$
9. **until** there are at least N points in each region of S **do**
10. Create a HeatMap diagram based on P

200k Monte Carlo Runs
E[N]: 1974
σ(N): 212

Min: 1442
Max: 3711



"We divided up S into 100 equal squares [...] at least 10 points are simulated in each region during exploration"

Final Assessment

- Well written, structured and motivated
- Good figures
- Implementation details and rationale could have been a bit more clear
- Demonstrates the search-based approach outperformed humans and random search
- Critical regions are not always trivial to identify by eye
- Rationale for sampling strategy could have been explained in more detail

THANK YOU!

+ Q&A