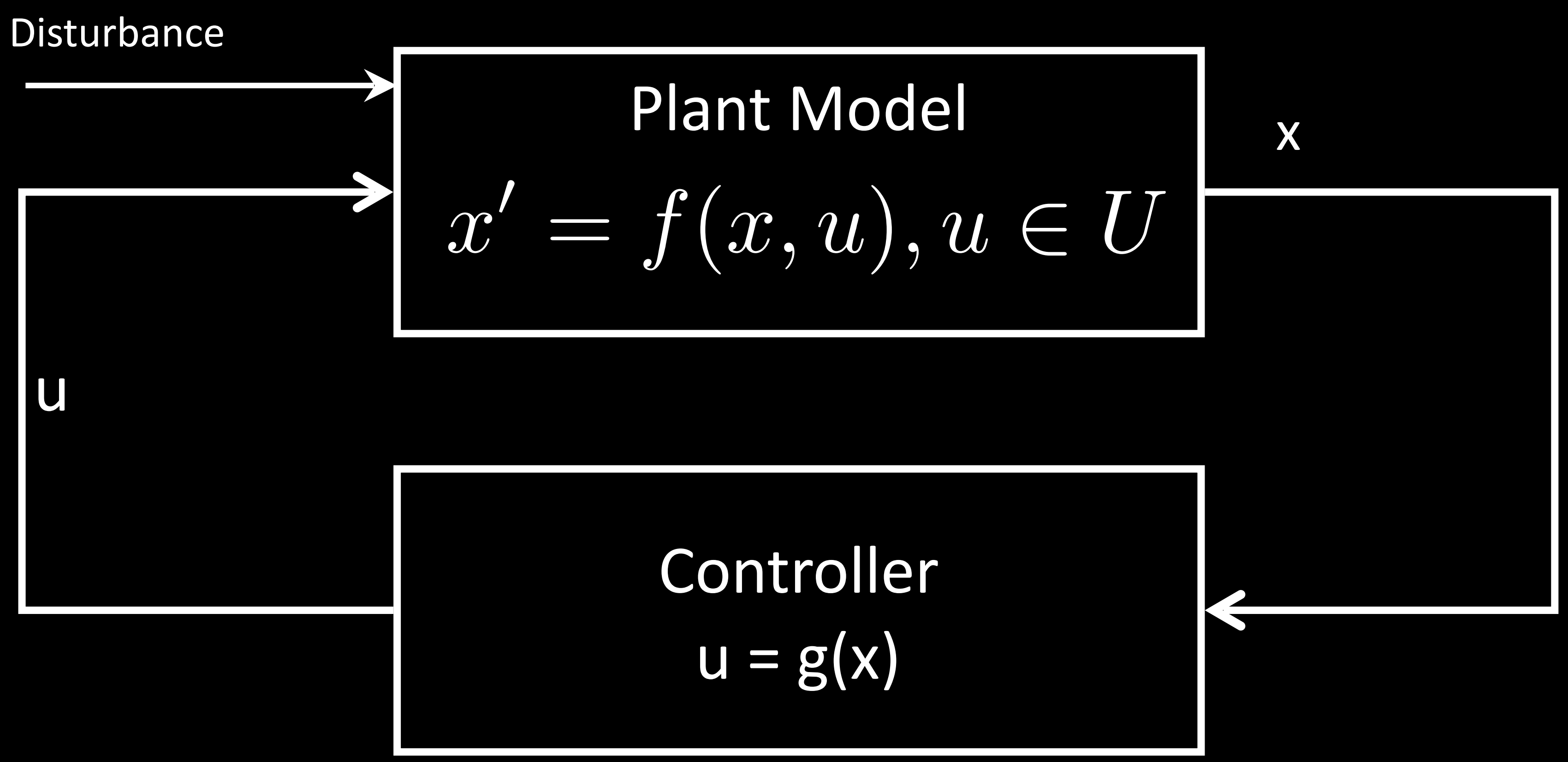
A class of control certificates to ensure Reach-While-Stay for Switched Systems

Hadi Ravanbakhsh and Sriram Sankaranarayanan

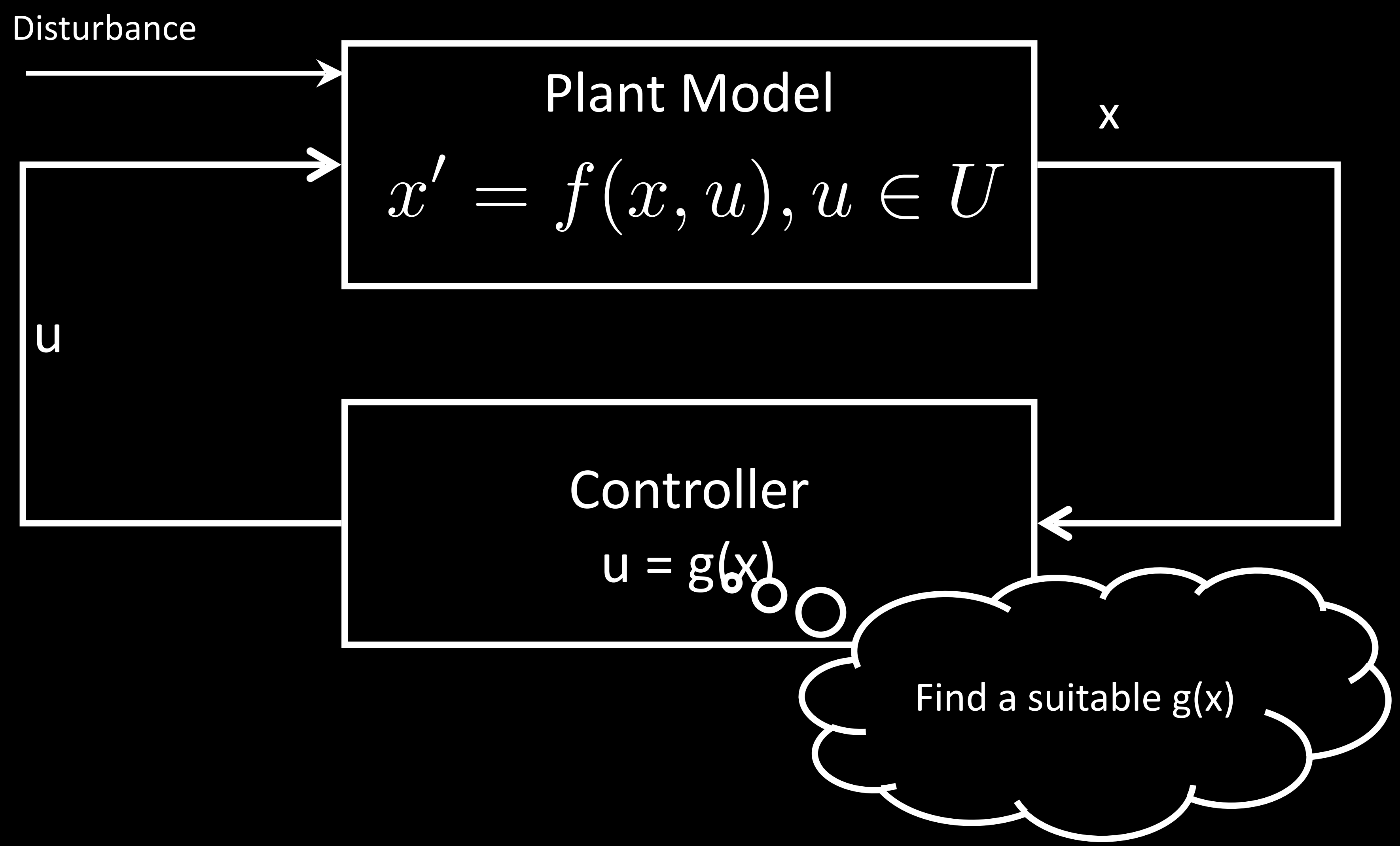
Presented by Sergio Mover

University of Colorado Boulder

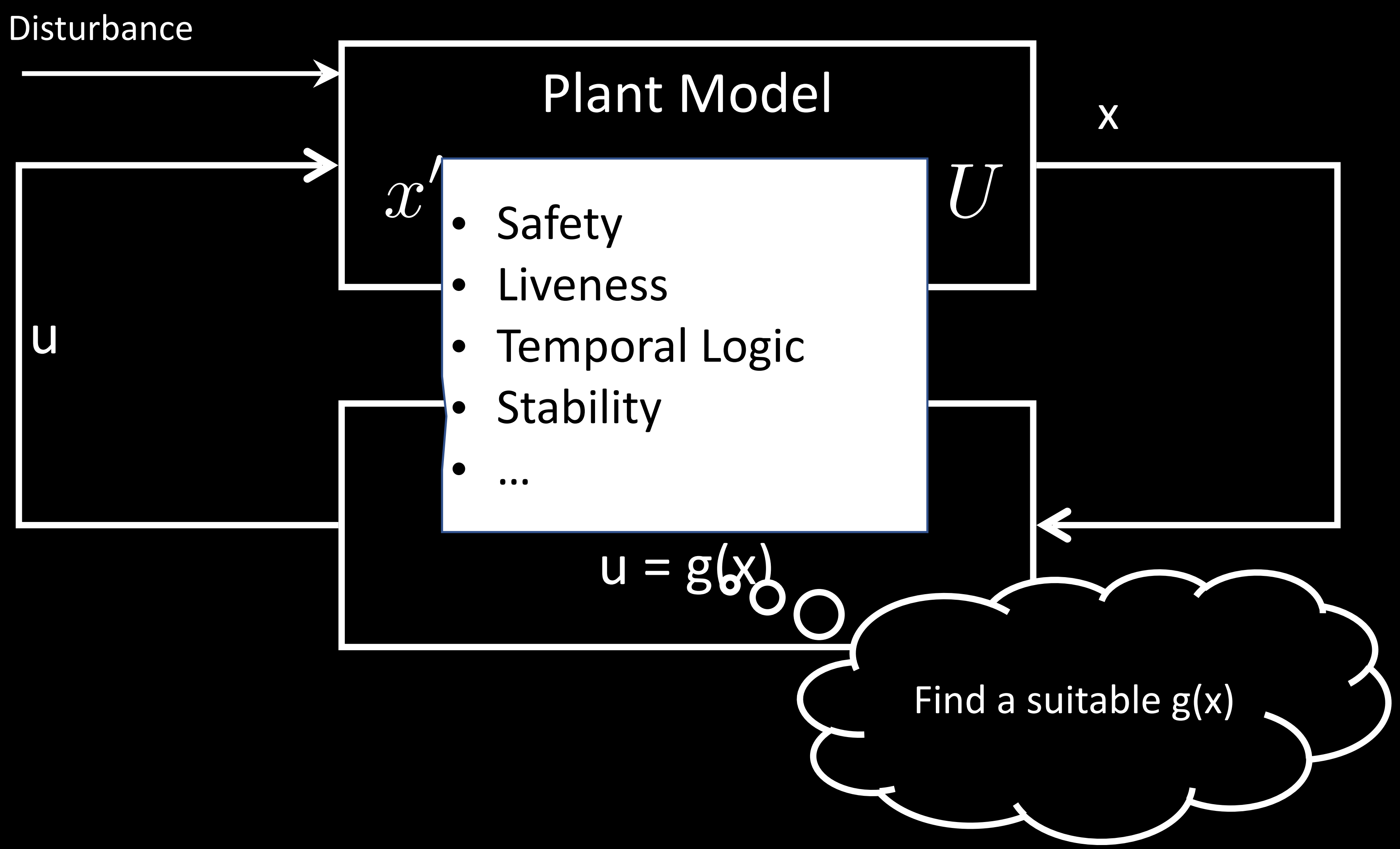
Synthesis



Synthesis


Disturbance
Plant Model
$x' = f(x, u), u \in U$
x
u
Controller
u = g(x)
Find a suitable g(x)

Synthesis



Disturbance
Plant Model
x
x'
U
u
• Safety
• Liveness
• Temporal Logic
• Stability
• …
u = g(x)
Find a suitable g(x)

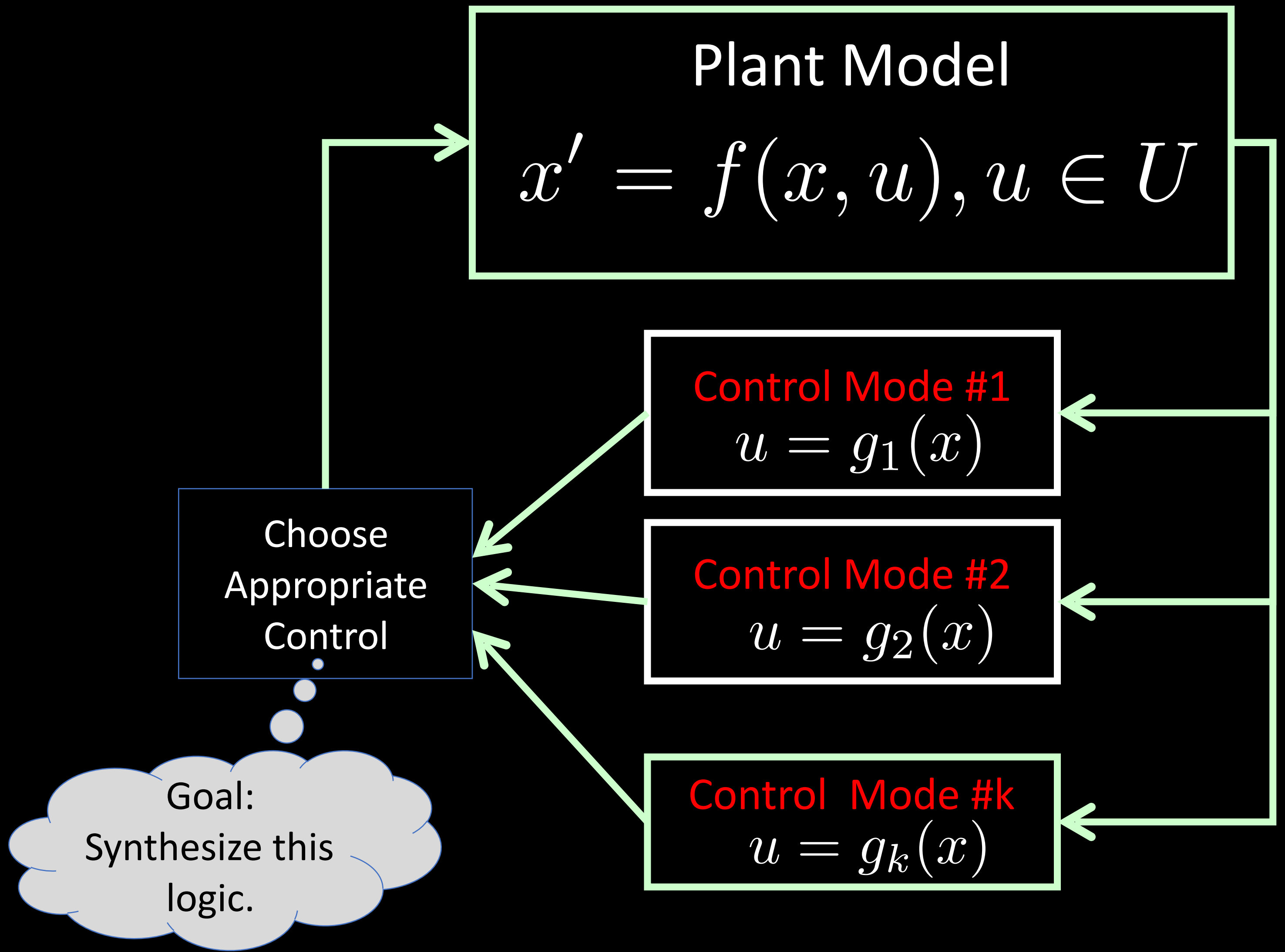
Contribution

- Synthesis of switching controller.
- Plant Model: Switched System.
- Specification: Reach-While-Stay.
- Methodology: Lyapunov-like Control Certificates.

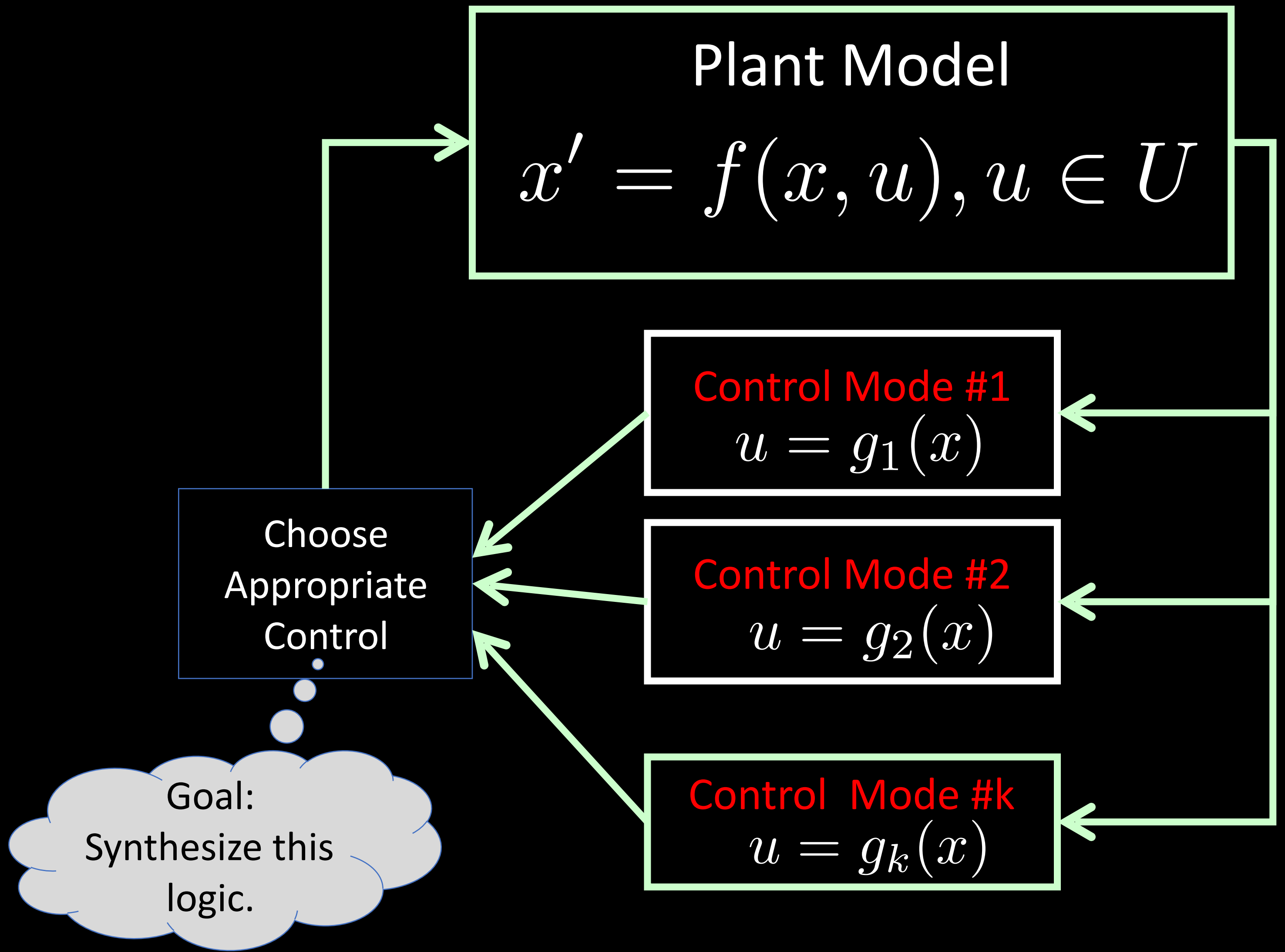
Extend to Temporal Properties.

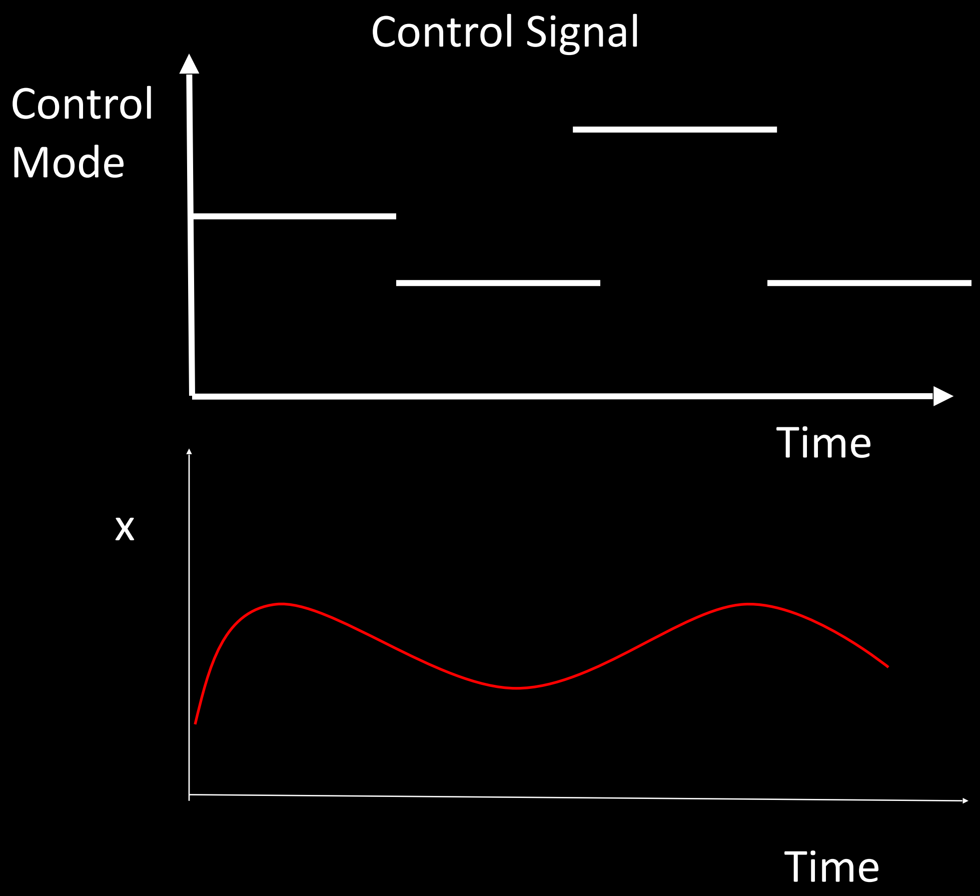
Use CEGIS procedure to synthesize.

Switched Plant Model

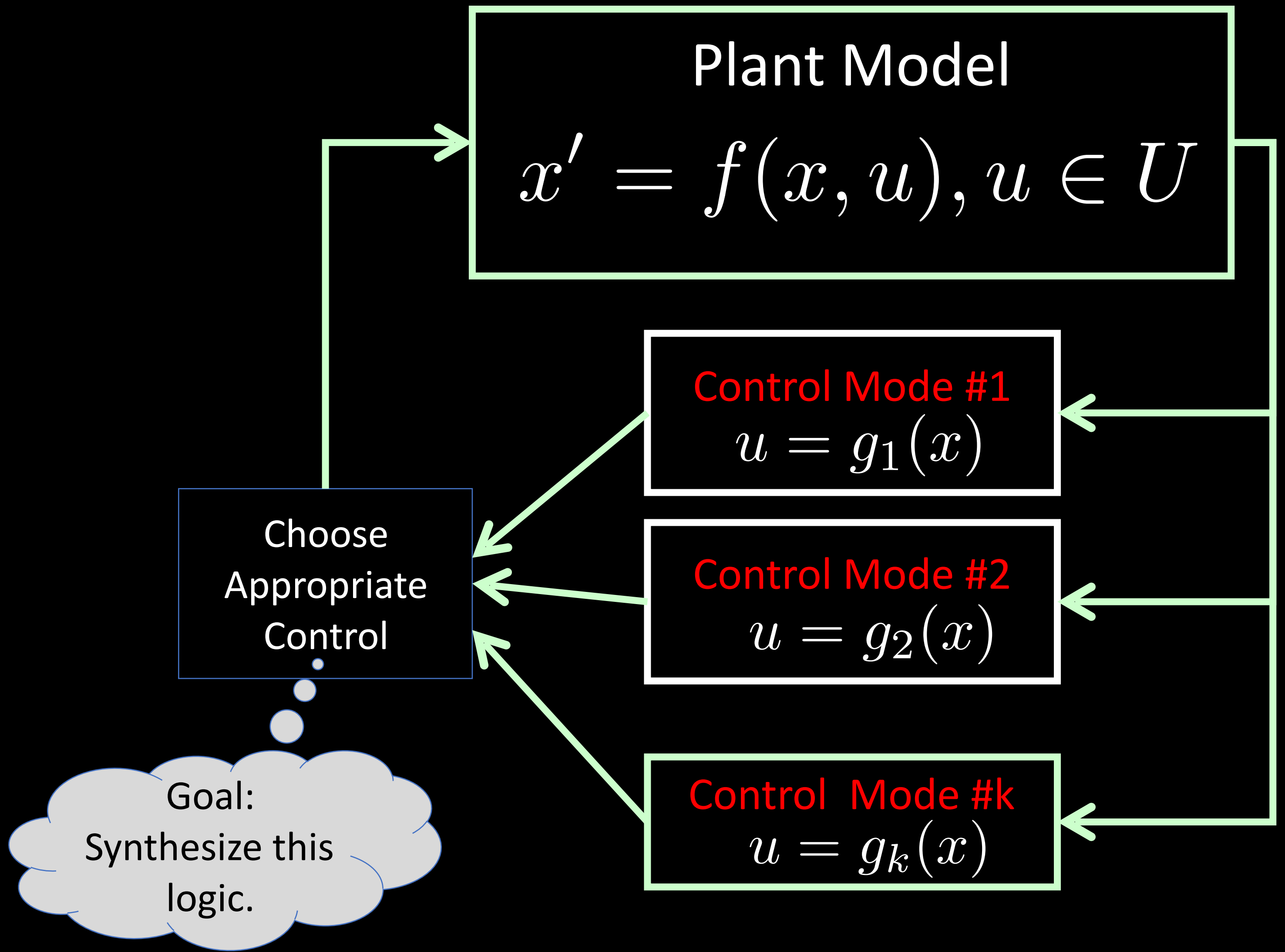


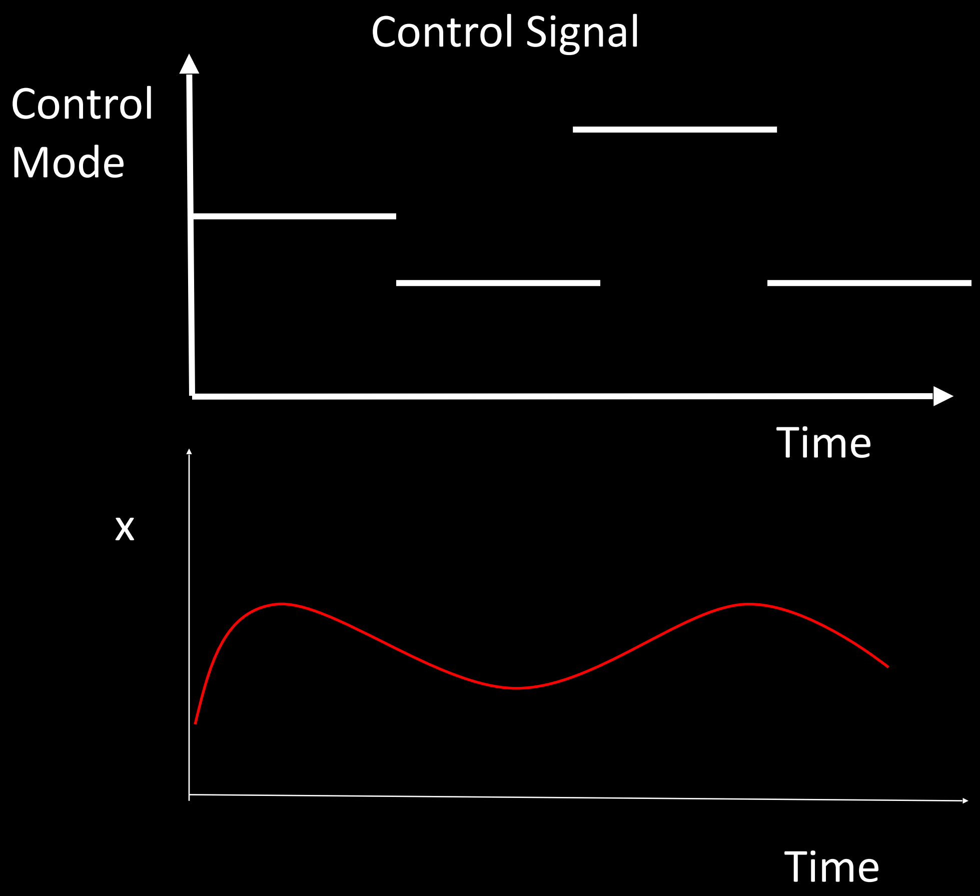
Switched Plant Model





Switched Plant Model



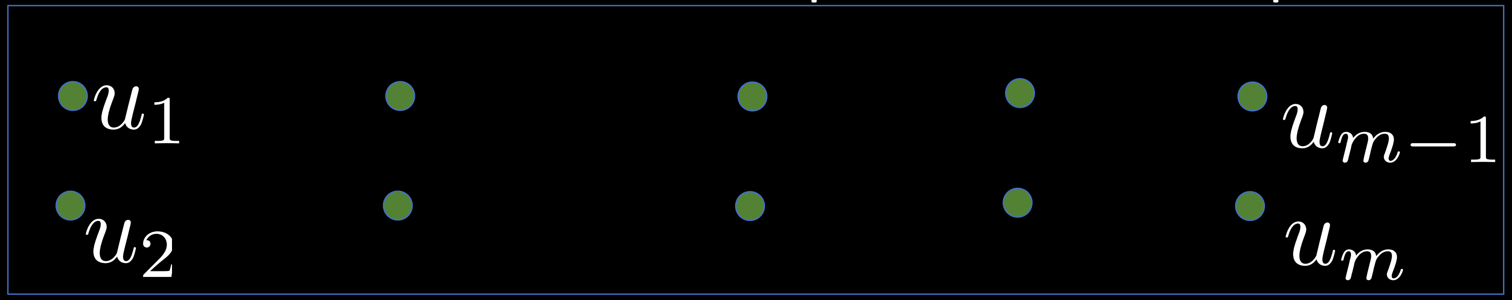


Minimum Dwell Time Constraint.

Choosing Control Modes

- Idea #1: Sample control values.

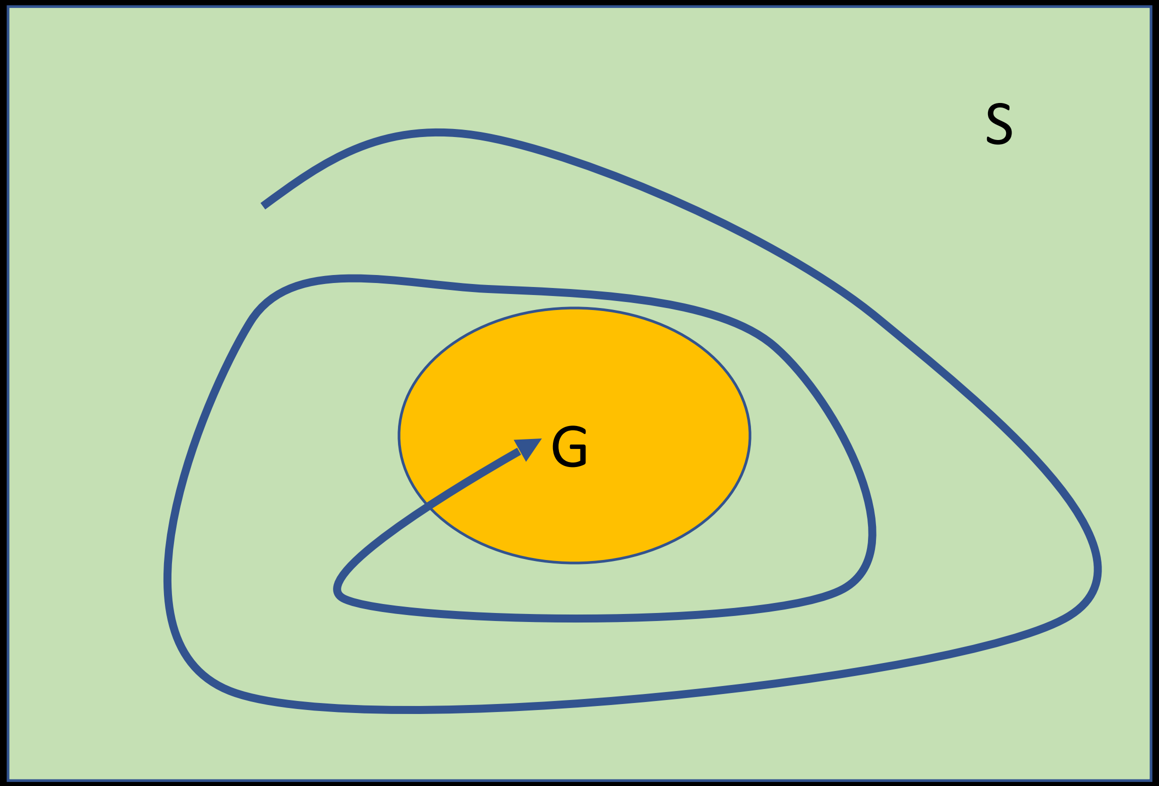
Space of Control Inputs: U



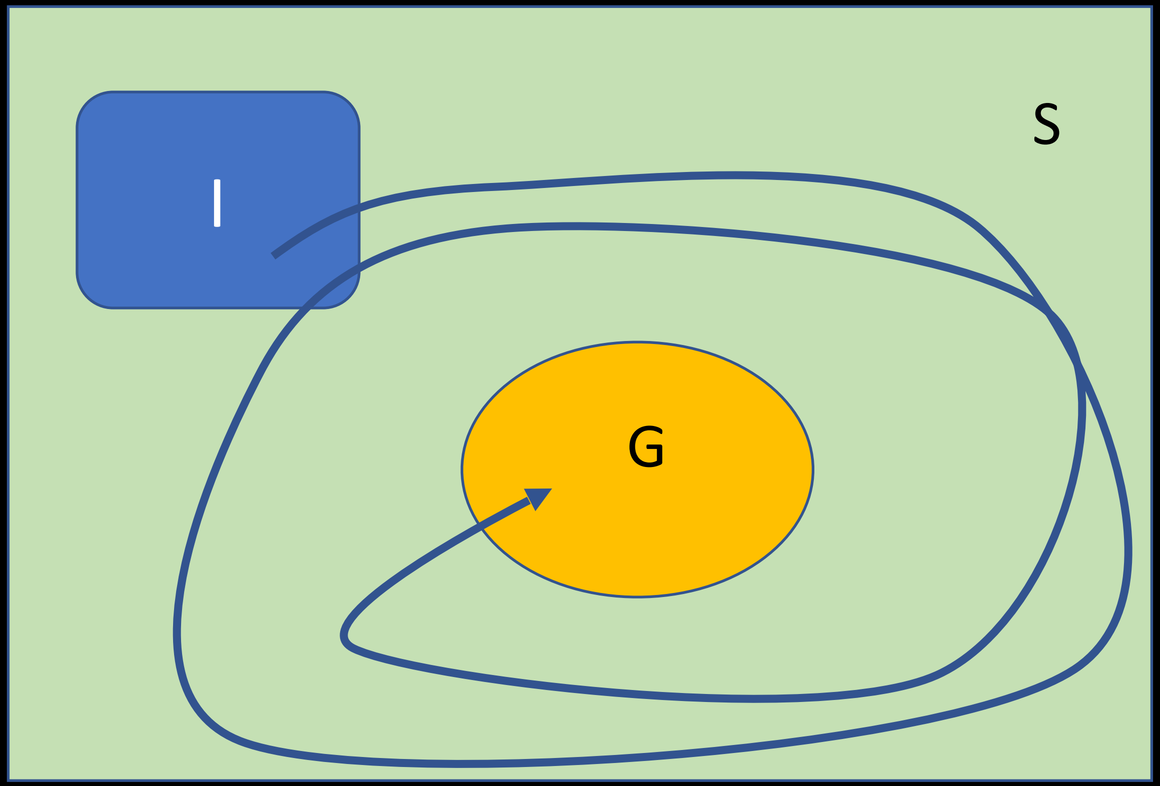
- Idea #2: Combine "useful" primitive feedback laws.
  - Eg., robotic vehicles.

Reach-While-Stay (RWS) Property

$S \, \mathcal{U} \, G$

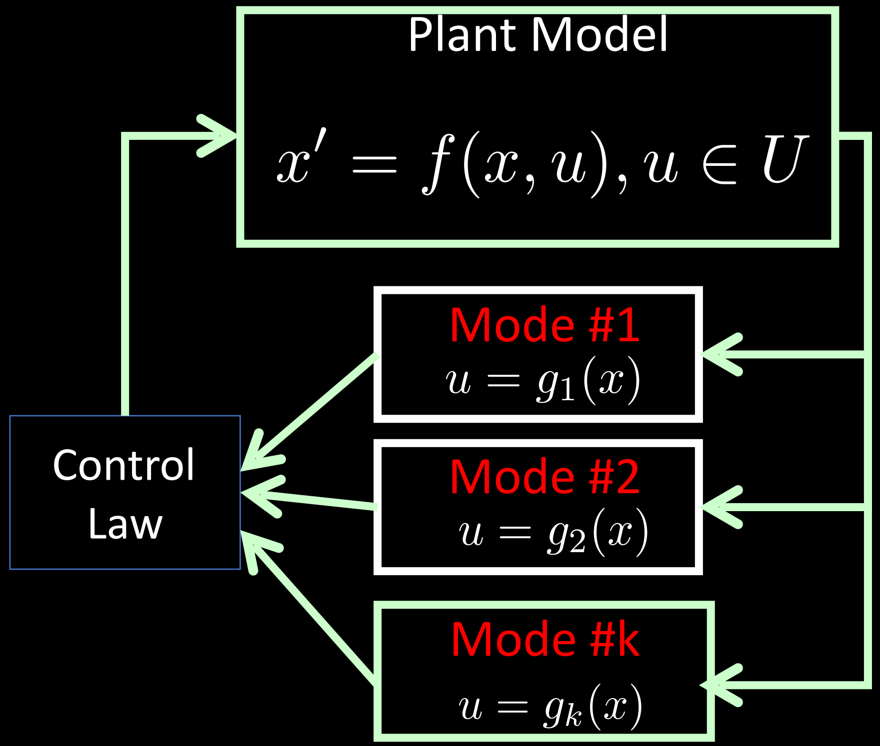


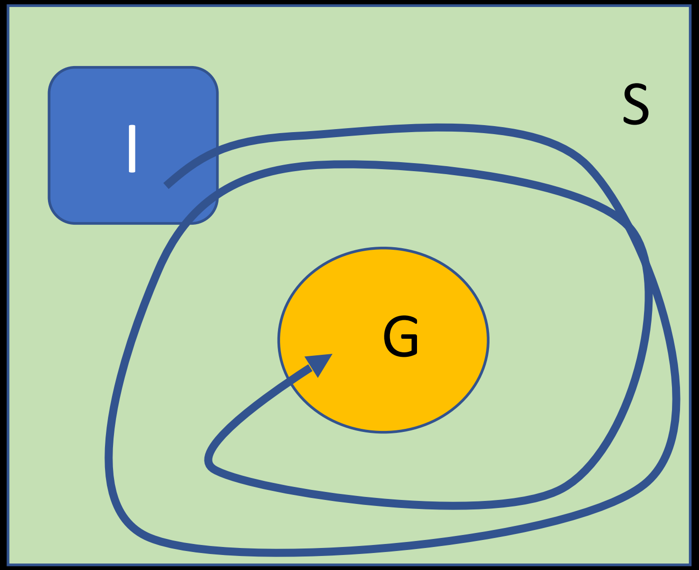
Initialized RWS $I \Rightarrow S \, \mathcal{U} \, G$



Problem Statement

- Inputs:
  - Switched Plant Model
  - (Initialized) RWS Property
- Output:
  - Control Law: $\mathbf{x} \rightarrow \{1, \ldots, m\}$
  - Minimum Dwell Time Estimate.





$I \Rightarrow S \, U \, G$

Control Certificates

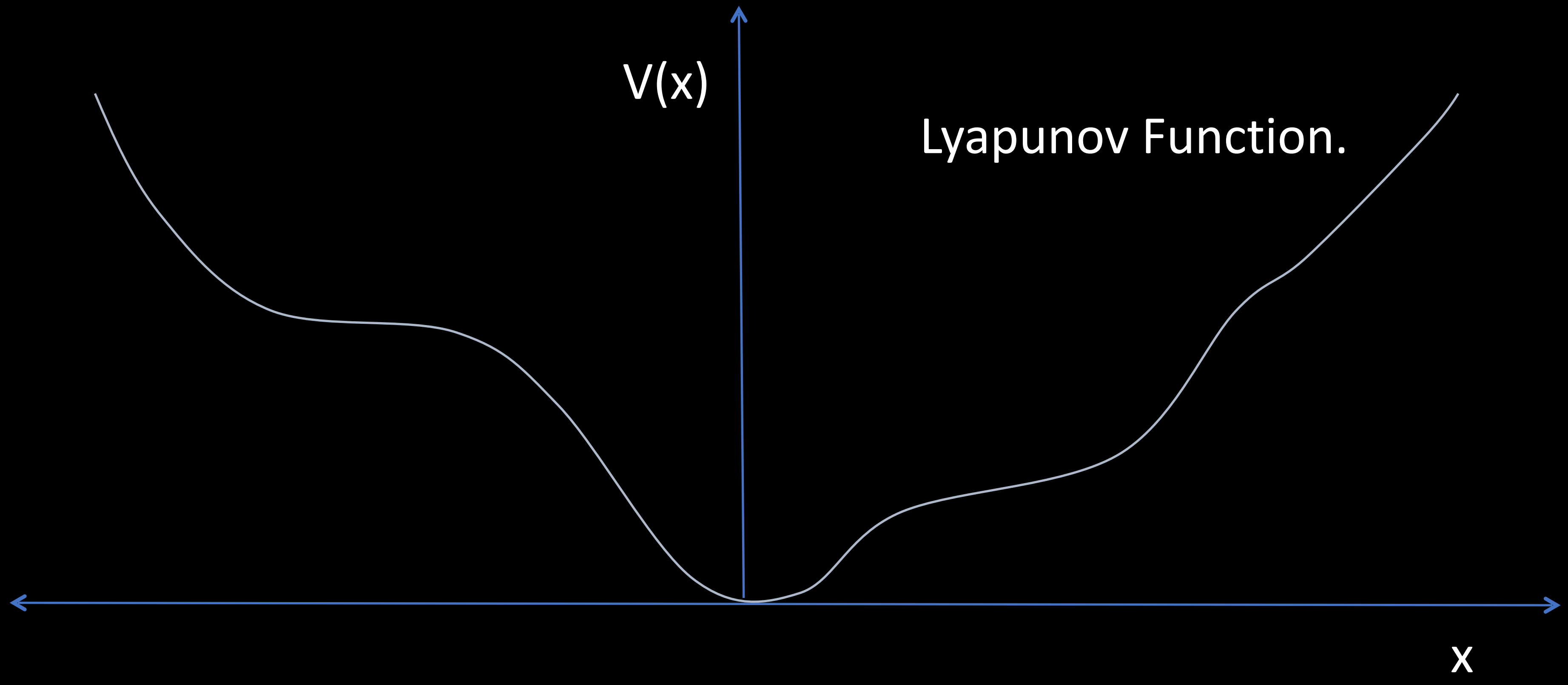
Control Lyapunov Function

[Artstein; Sontag; …]

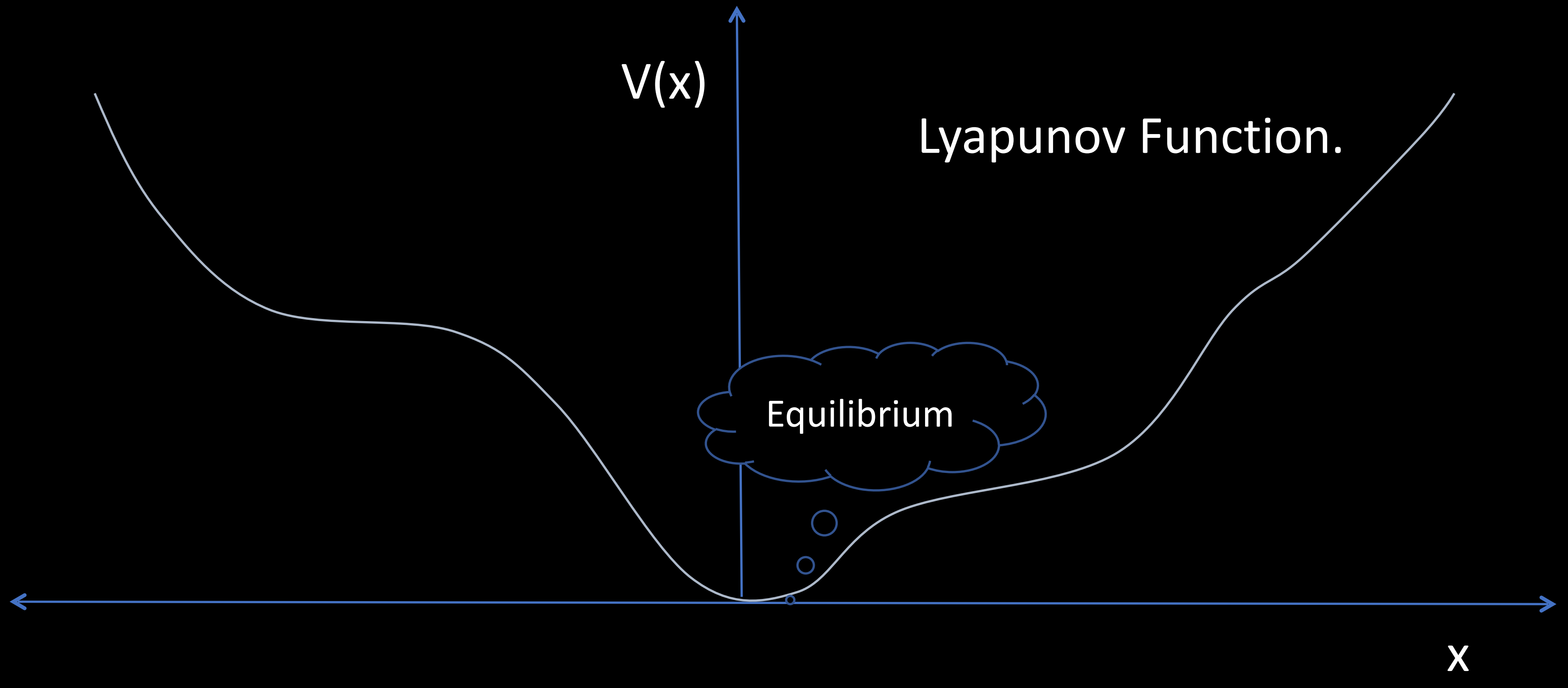
- Lyapunov function: V(x)
- V(x) is positive definite.
- A control input chosen s.t. derivative is negative definite.

$$(\forall\ x \neq x^*)\ (\exists u \in U)\ V'(x) = \nabla_x V(x) \cdot f(x, u) < 0$$

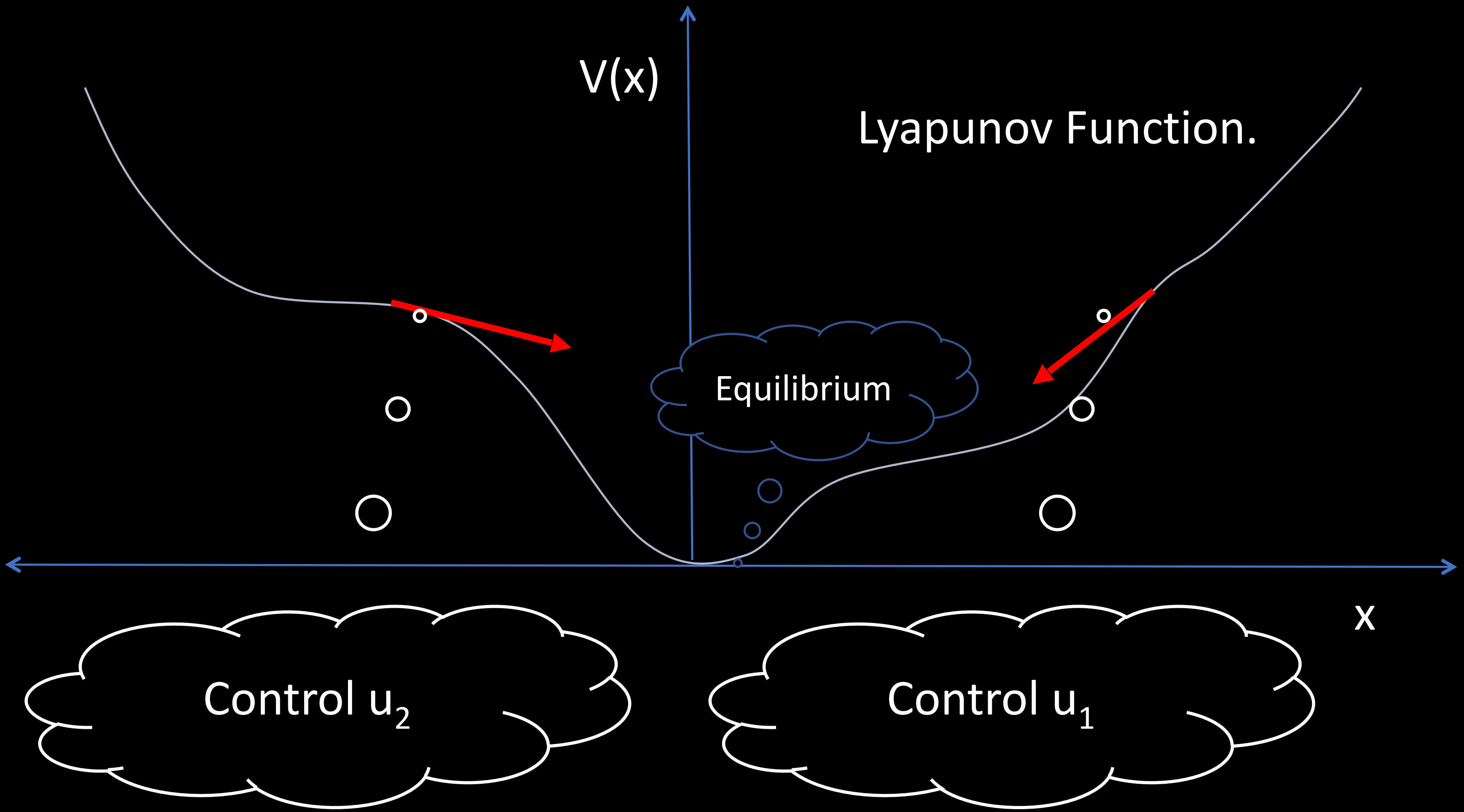
Control Lyapunov Function: Interpretation



Control Lyapunov Function: Interpretation



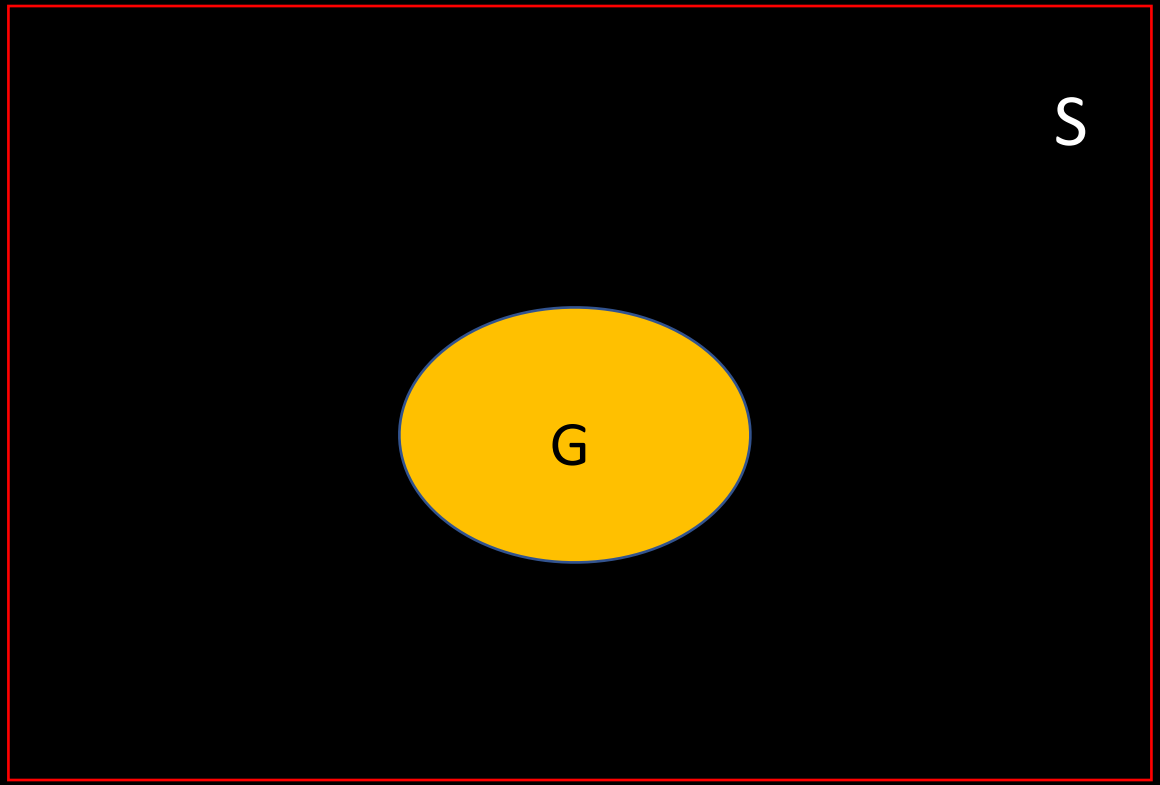
Control Lyapunov Function: Interpretation



Control Certificates for RWS

[Dimitrova+Majumdar]

Find function V

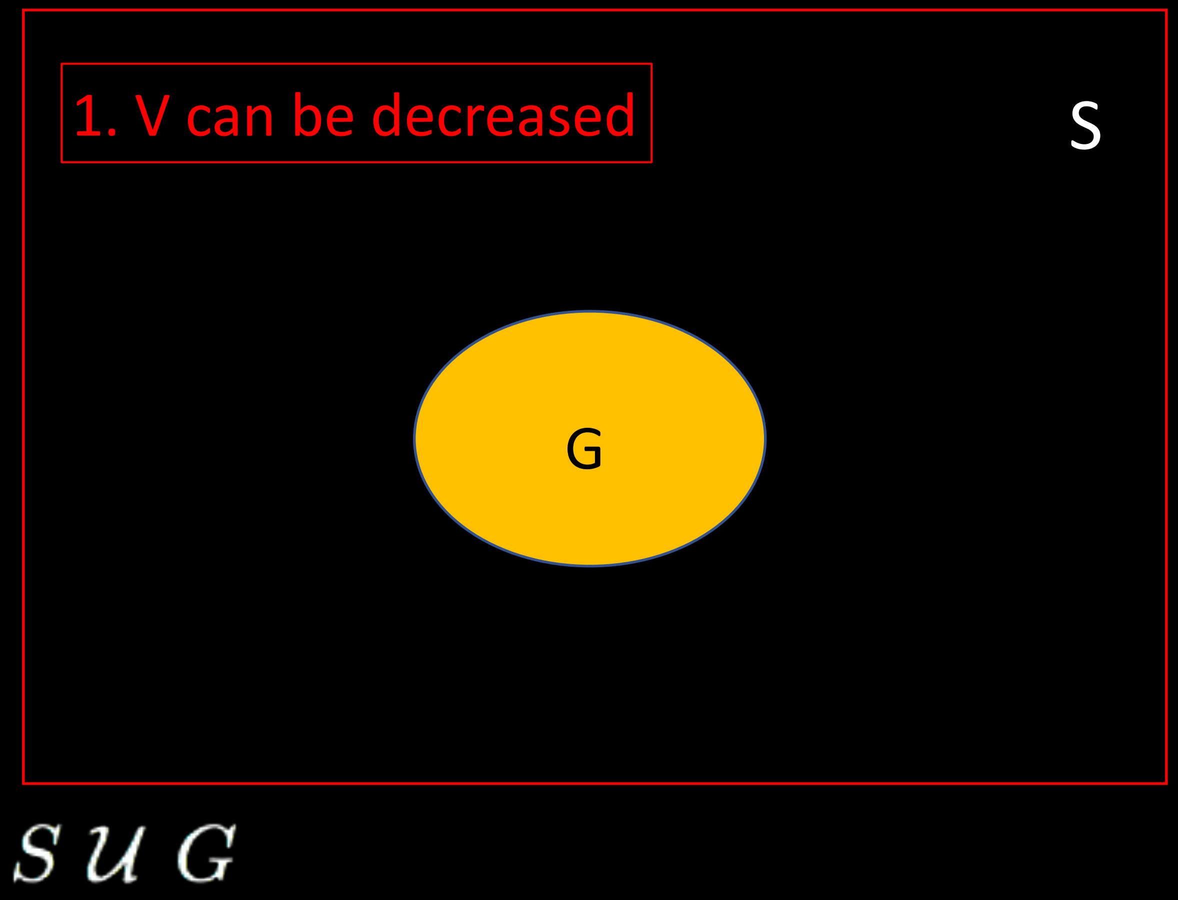


$S \mathcal{U} G$

Control Certificates for RWS

[Dimitrova+Majumdar]

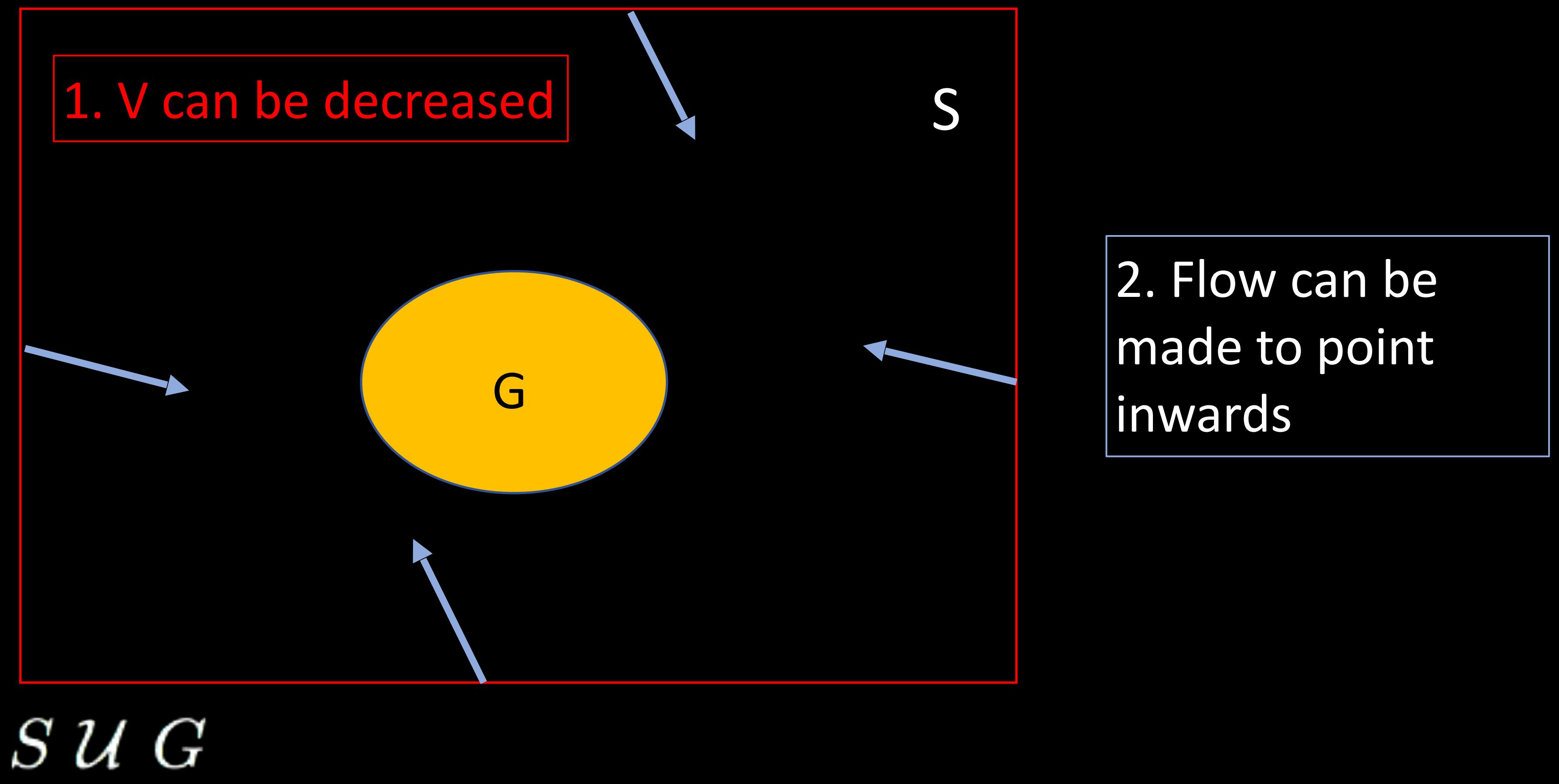
Find function V



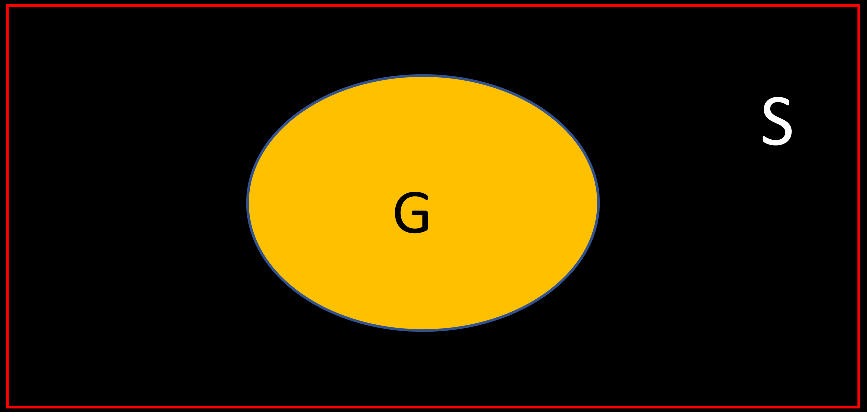
Control Certificates for RWS

[Dimitrova+Majumdar]

Find function V

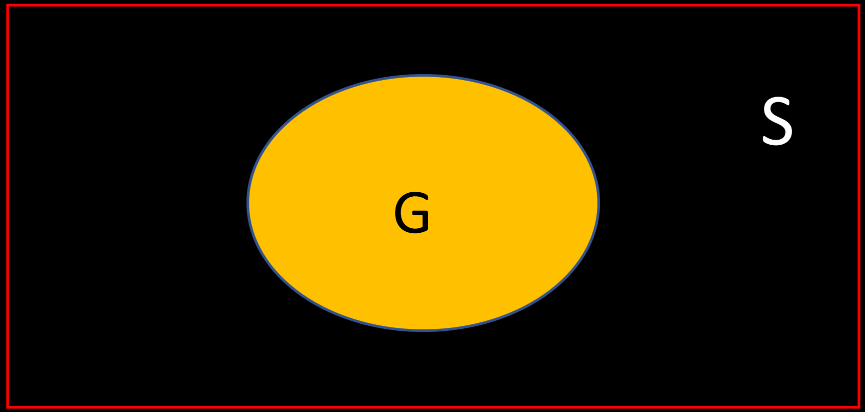


Control Certificates



$$(\forall\ \mathbf{x}) \begin{cases} \mathbf{x} \in int(S) \setminus G \implies (\exists\ q)\ \dot{V}_q(\mathbf{x}) < -\epsilon \\ \mathbf{x} \in F_1 \setminus G \implies (\exists\ q) \left(\dot{V}_q(\mathbf{x}) < -\epsilon \ \wedge \bigwedge_{p \in F_1^{=}} \dot{p}_q(\mathbf{x}) < -\epsilon \right) \\ \vdots \\ \mathbf{x} \in F_{l_k} \setminus G \implies (\exists\ q) \left(\dot{V}_q(\mathbf{x}) < -\epsilon \ \wedge \bigwedge_{p \in F_{l_k}^{=}} \dot{p}_q(\mathbf{x}) < -\epsilon \right). \end{cases}$$

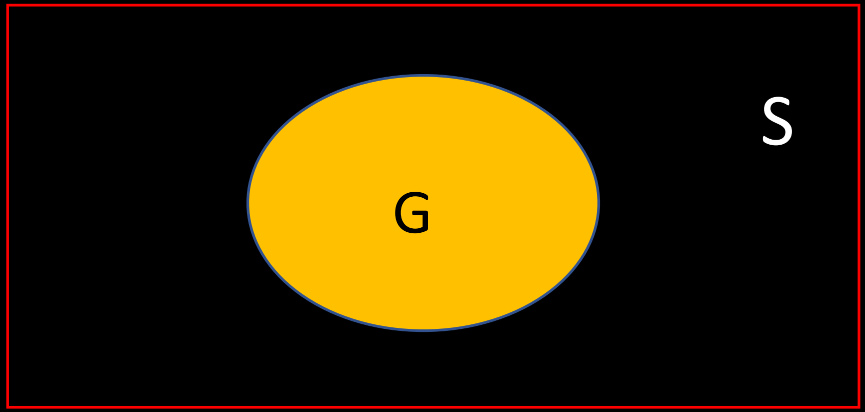
Control Certificates



"Can always choose a mode q to decrease V"

$$
(\forall\ \mathbf{x}) \begin{cases}
\mathbf{x} \in int(S) \setminus G \implies (\exists\ q)\ \dot{V}_q(\mathbf{x}) < -\epsilon \\
\mathbf{x} \in F_1 \setminus G \implies (\exists\ q) \left(\dot{V}_q(\mathbf{x}) < -\epsilon \ \wedge \bigwedge_{p \in F_1^{=}} \dot{p}_q(\mathbf{x}) < -\epsilon \right) \\
\vdots \\
\mathbf{x} \in F_{l_k} \setminus G \implies (\exists\ q) \left(\dot{V}_q(\mathbf{x}) < -\epsilon \ \wedge \bigwedge_{p \in F_{l_k}^{=}} \dot{p}_q(\mathbf{x}) < -\epsilon \right).
\end{cases}
$$

Control Certificates

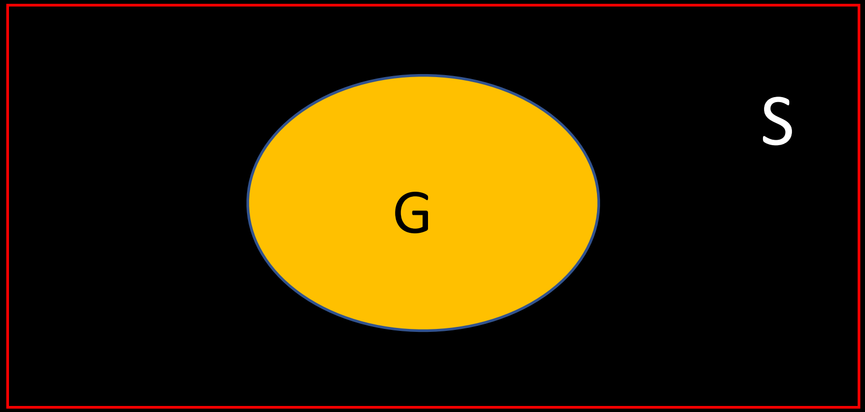


"Can always choose a mode q to decrease V"

$$(\forall\ \mathbf{x}) \begin{cases} \mathbf{x} \in int(S) \setminus G \implies (\exists\ q)\ \dot{V}_q(\mathbf{x}) < -\epsilon \\ \mathbf{x} \in F_1 \setminus G \implies (\exists\ q) \left(\dot{V}_q(\mathbf{x}) < -\epsilon \ \wedge \bigwedge_{p \in F_1^{=}} \dot{p}_q(\mathbf{x}) < -\epsilon \right) \\ \vdots \\ \mathbf{x} \in F_{l_k} \setminus G \implies (\exists\ q) \left(\dot{V}_q(\mathbf{x}) < -\epsilon \ \wedge \bigwedge_{p \in F_{l_k}^{=}} \dot{p}_q(\mathbf{x}) < -\epsilon \right). \end{cases}$$

If on the k^{th} facet of S

Control Certificates

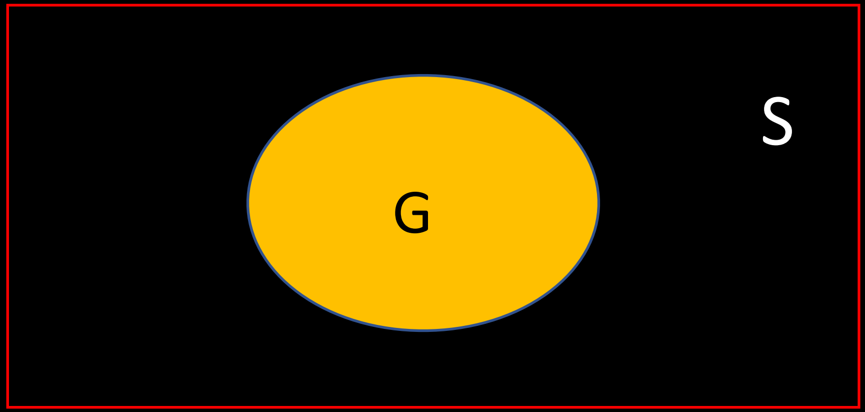


"Can always choose a mode q to decrease V"

$$(\forall\ \mathbf{x}) \begin{cases} \mathbf{x} \in int(S) \setminus G \implies (\exists\ q)\ \dot{V}_q(\mathbf{x}) < -\epsilon \\ \mathbf{x} \in F_1 \setminus G \implies (\exists\ q) \left(\dot{V}_q(\mathbf{x}) < -\epsilon \ \wedge \bigwedge_{p \in F_1^{=}} \dot{p}_q(\mathbf{x}) < -\epsilon \right) \\ \vdots \\ \mathbf{x} \in F_{l_k} \setminus G \implies (\exists\ q) \left(\dot{V}_q(\mathbf{x}) < -\epsilon \ \wedge \bigwedge_{p \in F_{l_k}^{=}} \dot{p}_q(\mathbf{x}) < -\epsilon \right). \end{cases}$$

If on the k^{th} facet of S

Control Certificates



"Can always choose a mode q to decrease V"

$$(\forall\ \mathbf{x}) \begin{cases} \mathbf{x} \in int(S) \setminus G \implies (\exists\ q)\ \dot{V}_q(\mathbf{x}) < -\epsilon \\ \mathbf{x} \in F_1 \setminus G \implies (\exists\ q) \left(\dot{V}_q(\mathbf{x}) < -\epsilon \ \wedge \bigwedge_{p \in F_1^{=}} \dot{p}_q(\mathbf{x}) < -\epsilon \right) \\ \vdots \\ \mathbf{x} \in F_{l_k} \setminus G \implies (\exists\ q) \left(\dot{V}_q(\mathbf{x}) < -\epsilon \ \wedge \bigwedge_{p \in F_{l_k}^{=}} \dot{p}_q(\mathbf{x}) < -\epsilon \right). \end{cases}$$

If on the k^{th} facet of S

"mode q to decrease V"

"push the flow inside the facet"

Synthesizing Control Certificates

- Fix a template (Ansatz) with unknown coefficients.

$$V(x_1, x_2) : \; c_0 + c_1 x_1 + c_2 x_2 + c_3 x_1 x_2 + c_4 x_1^2 + c_5 x_2^2$$

- Enforce CLF constraints on the unknown form.

$$(\exists\, \mathbf{c})\; (\forall\, \mathbf{x})\; (\exists\, q \in \{1, \ldots, k\})\; \cdots$$

Synthesizing Control Certificates

- Fix a template (Ansatz) with unknown coefficients.

$$V(x_1, x_2) : \; c_0 + c_1 x_1 + c_2 x_2 + c_3 x_1 x_2 + c_4 x_1^2 + c_5 x_2^2$$

- Enforce CLF constraints on the unknown form.

$$(\exists\, \mathbf{c})\ (\forall\, \mathbf{x})\ (\exists\, q \in \{1, \ldots, k\})\ \cdots$$

Quantifier Alternation

Nonlinear Constraints.

Counterexample Guided Inductive Synthesis.

Program Sketching [Solar-Lezama + Others]

```
int search(int [] a,int n, int s){
//@pre:…
     if (n <= ??){ return 0; }
     for i = 0 to ??
          if (a[i] == ??){
               return 1;
          }
     end
     return ??;
}
//@post:…
```

Find appropriate
program expressions.

CEGIS Approach

Constraints to be solved:

$$(\exists\, c)\, (\forall y)\, \psi(c, y)$$

Template Parameters

Program Behaviors

Iterative Procedure:

- Finite set Y_i : $\{y_{1,\dots,} y_k\}$
- Instantiate the $\forall$ Quantifier

Basic CEGIS Loop

$$(\exists\, c)\ (\forall y)\ \psi(c, y)$$

1. Check SATisfiability of the formula:

$$\psi(c, y_1) \ \wedge\ \psi(c, y_2) \ \wedge\ \cdots\ \wedge\ \psi(c, y_k)$$

Basic CEGIS Loop

$$(\exists\, c)\ (\forall y)\ \psi(c, y)$$

1. Check SATisfiability of the formula:

$$\psi(c, y_1) \ \wedge\ \psi(c, y_2)\ \wedge\ \cdots\ \wedge\ \psi(c, y_k)$$

UNSAT,
Failure

Basic CEGIS Loop

$$(\exists\, c)\ (\forall y)\ \psi(c, y)$$

1. Check SATisfiability of the formula:

$$\psi(c, y_1) \ \wedge\ \psi(c, y_2) \ \wedge\ \cdots \ \wedge\ \psi(c, y_k)$$

SAT, c_k

UNSAT,
Failure

Basic CEGIS Loop

$$(\exists\, c)\,(\forall y)\,\psi(c, y)$$

1. Check SATisfiability of the formula:

$$\psi(c, y_1) \;\wedge\; \psi(c, y_2) \;\wedge\; \cdots \;\wedge\; \psi(c, y_k)$$

SAT, c_k

2. Check UNSATisfiability of

$$\neg\psi(c_k, y)$$

UNSAT,
Failure

Basic CEGIS Loop

$$(\exists\, c)\, (\forall y)\, \psi(c, y)$$

1. Check SATisfiability of the formula:

$$\psi(c, y_1) \;\wedge\; \psi(c, y_2) \;\wedge\; \cdots \;\wedge\; \psi(c, y_k)$$

SAT, c_k

UNSAT, Failure

2. Check UNSATisfiability of

$$\neg\psi(c_k, y)$$

UNSAT, Success

Basic CEGIS Loop

$$(\exists\, c)\ (\forall y)\ \psi(c, y)$$

1. Check SATisfiability of the formula:

$$\psi(c, y_1)\ \wedge\ \psi(c, y_2)\ \wedge\ \cdots\ \wedge\ \psi(c, y_k)$$

SAT, c_k

SAT, y_{k+1}

UNSAT, Failure

2. Check UNSATisfiability of

$$\neg\psi(c_k, y)$$

UNSAT, Success

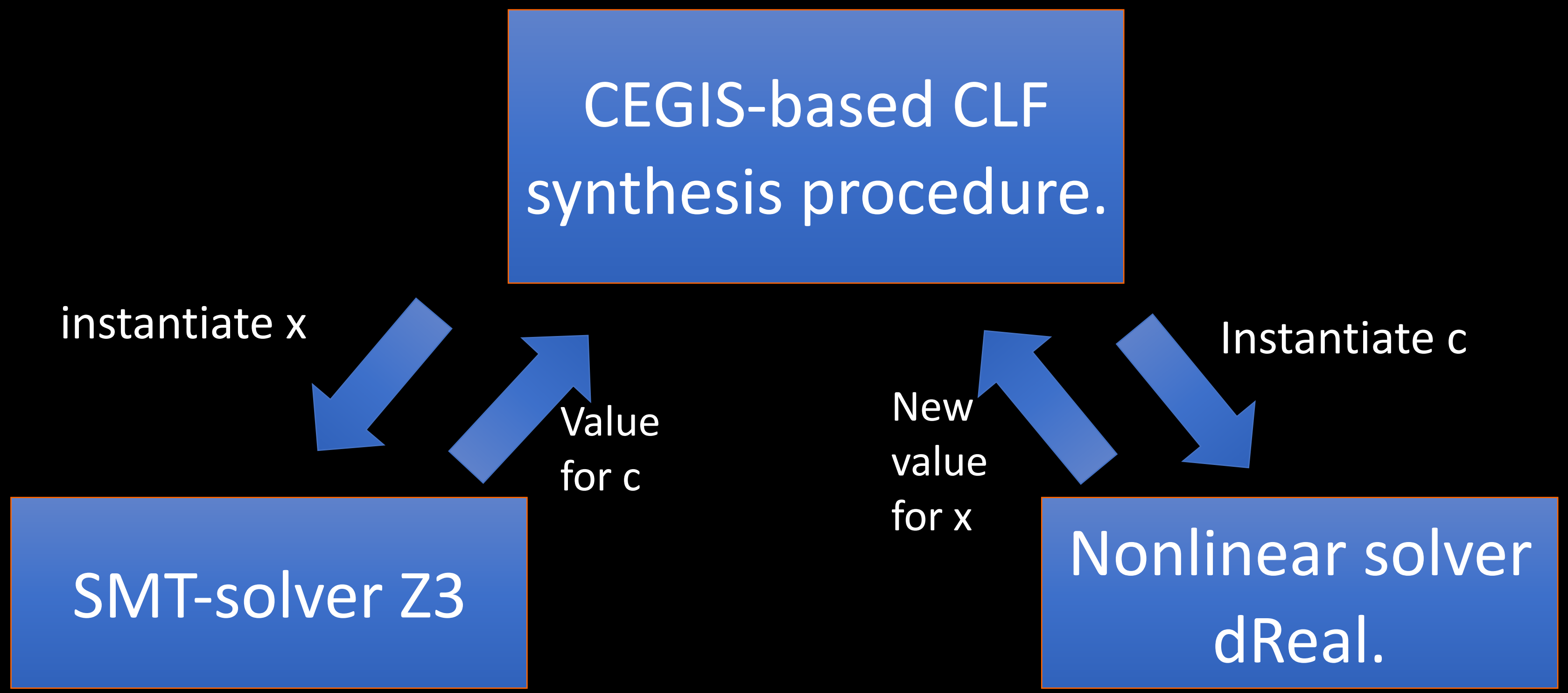
Applying CEGIS to Control Certificates

$$(\forall\ \mathbf{x}) \begin{cases} \mathbf{x} \in int(S) \setminus G \implies (\exists\ q)\ \dot{V}_q(\mathbf{x}) < -\epsilon \\ \mathbf{x} \in F_1 \setminus G \implies (\exists\ q) \left(\dot{V}_q(\mathbf{x}) < -\epsilon\ \wedge \bigwedge_{p \in F_1^{=}} \dot{p}_q(\mathbf{x}) < -\epsilon \right) \\ \vdots \\ \mathbf{x} \in F_{l_k} \setminus G \implies (\exists\ q) \left(\dot{V}_q(\mathbf{x}) < -\epsilon\ \wedge \bigwedge_{p \in F_{l_k}^{=}} \dot{p}_q(\mathbf{x}) < -\epsilon \right). \end{cases}$$

x instantiated => Linear arithmetic formula in coefficients of **V**

V instantiated => Nonlinear formula in **x**

Integrating SMT solvers



Improvement: use LMI solvers
See our CDC'15 paper.

Further Contributions

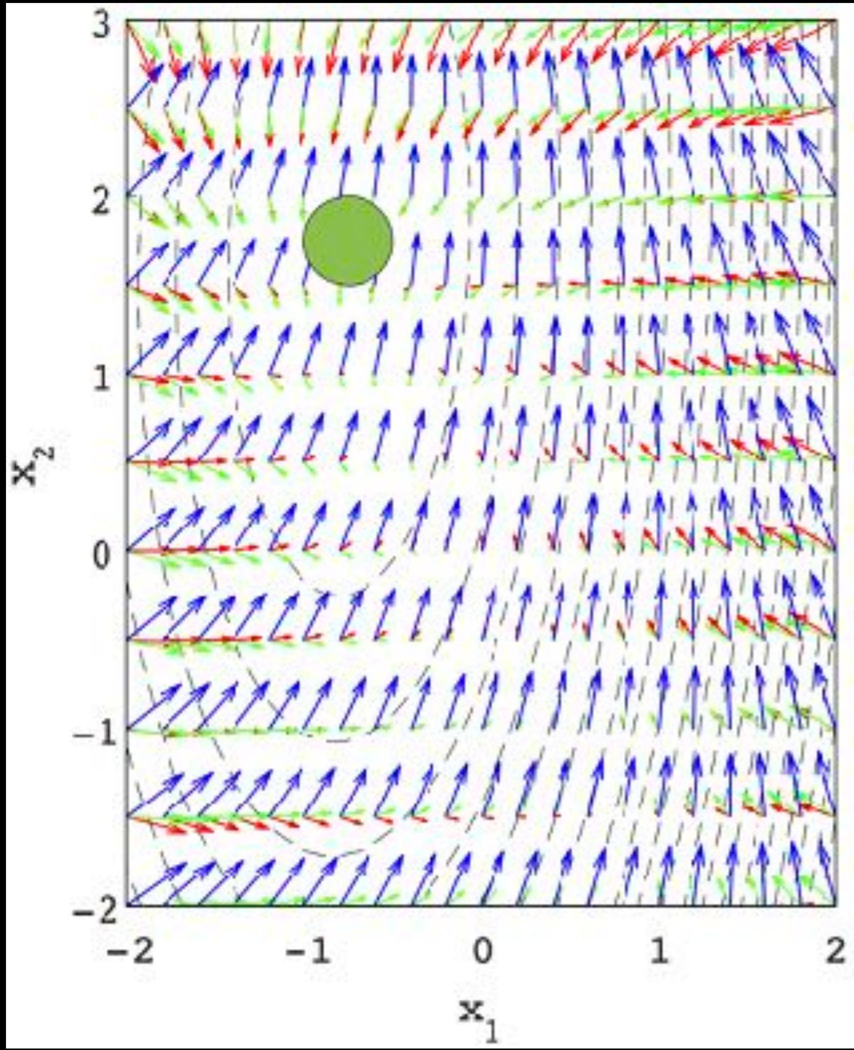
- Exponential barrier condition
- Extracting a control law: control code synthesis.
  - Guaranteeing minimum dwell time (compactness of S, G).
- Control certificates for initialized RWS problems.
- Combining RWS specifications to solve more complex problems.

Example: Nilsson and Ozay

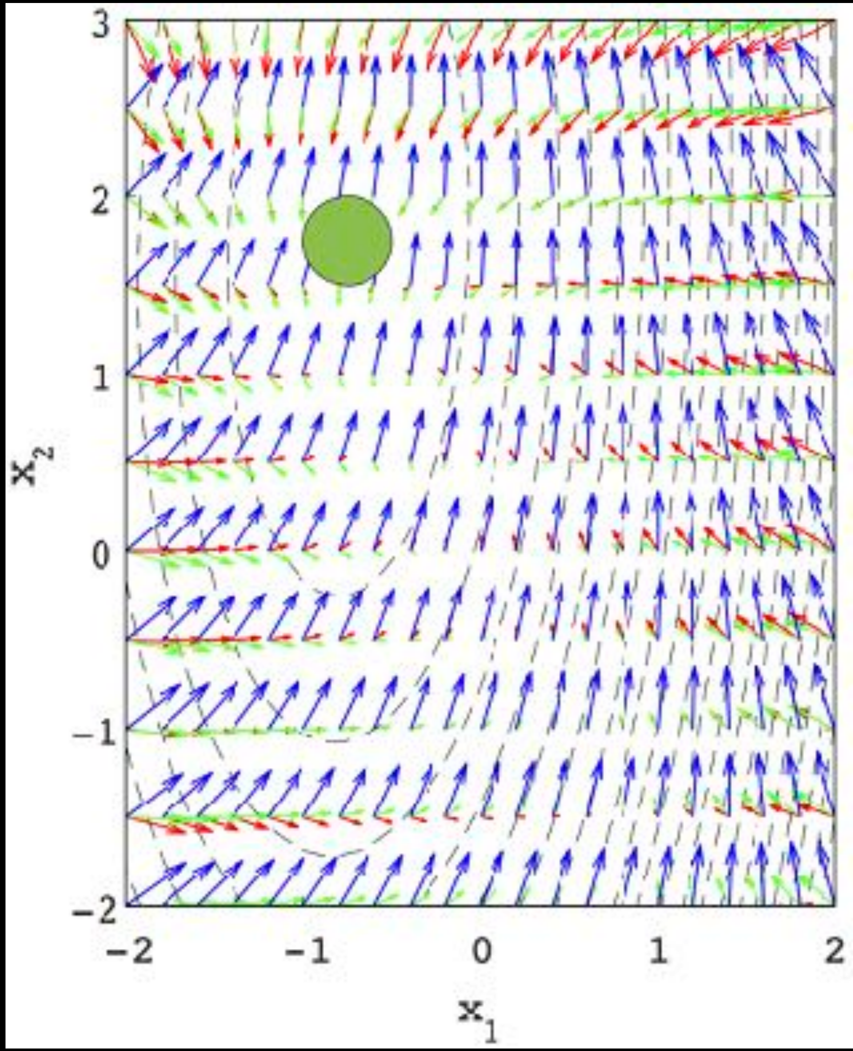
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -x_2 - 1.5x_1 - 0.5x_1^3 \\ x_1 \end{bmatrix} + B_q,$$

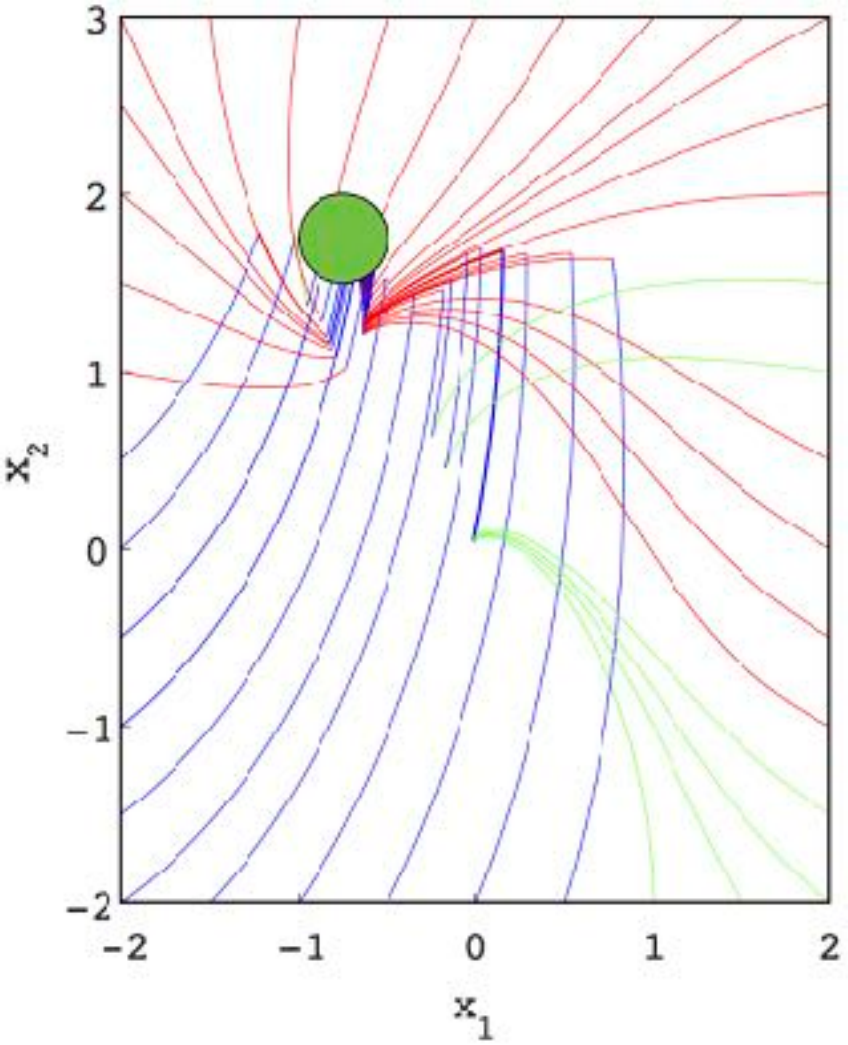
$$B_{q1} = \begin{bmatrix} 0 \\ -x_2^2 + 2 \end{bmatrix}, \; B_{q2} = \begin{bmatrix} 0 \\ -x_2 \end{bmatrix}, \; B_{q3} = \begin{bmatrix} 2 \\ 10 \end{bmatrix}.$$

$$V(x_1, x_2) : 37.782349x_1^2 - 2.009762x_1x_2 + 60.190607x_1 + 4.415093x_2^2 - 16.960145x_2 + 37.411604.$$

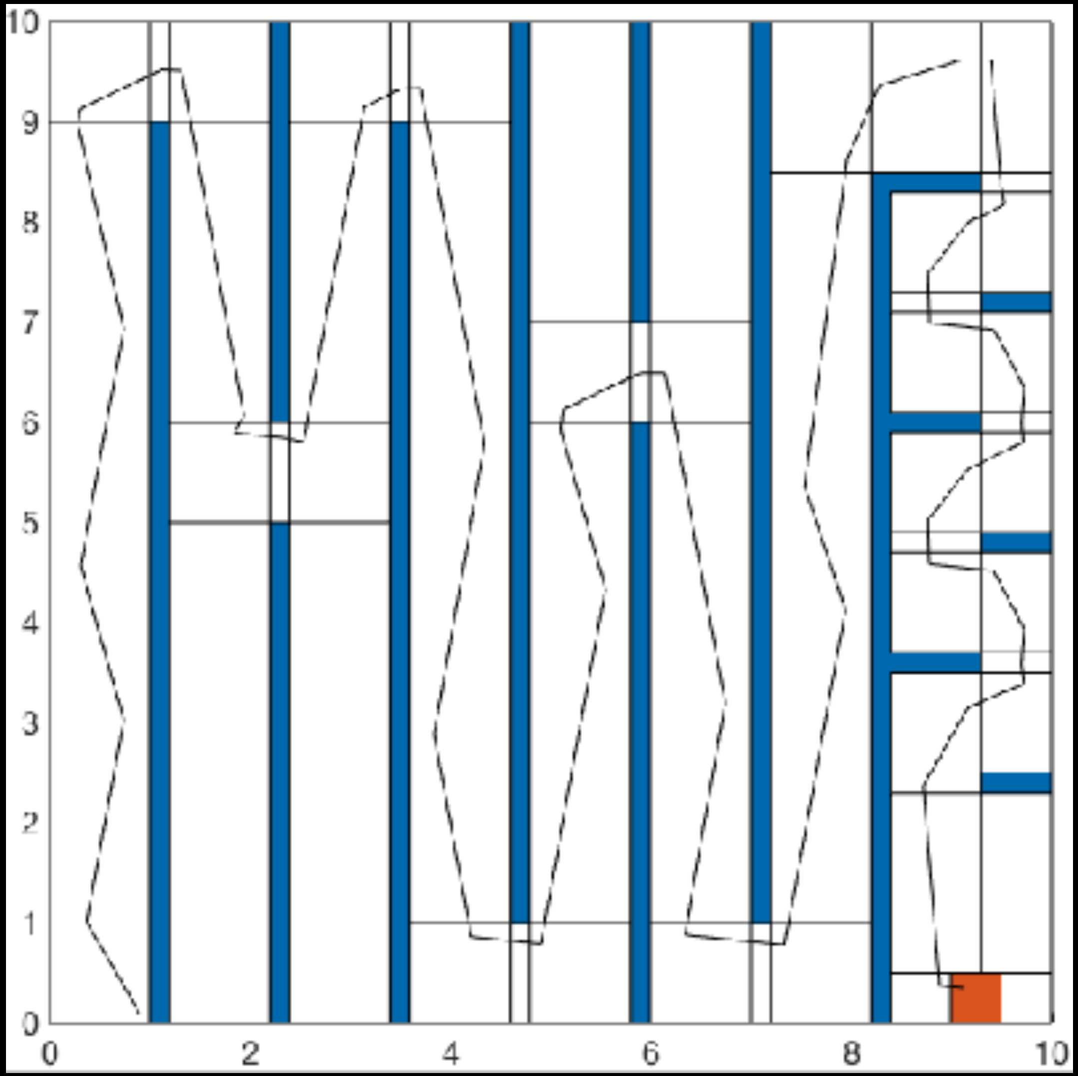


Example: Nilsson and Ozay (Continued)





Example: Unicycle Path Planning



Reach red goal set while avoiding blue obstacles.

Model: Nonlinear unicycle. Discretized steering.

Partition state-space into 53 polyhedra.

117 different RWS problems.

Comparison with SCOTS

State Variables in Plant Model

[Zamani et al.]

Problem		Parameters		SCOTS		CEGIS
ID	n	η	τ	*itr*	Time	Time
1	2	0.16^2	0.12	18	0	3
2	2	0.01^2	1.0	106	1	39
3	3	$0.2^2 \times 0.1$	0.3	404	989	1484
4	4	0.03×0.1^3	0.005	48	304	3
5	4	$0.1^2 \times 0.05^2$	0.3	TO		5296

Timeout = 10 hours

Conclusions

✓Control certificate approach can synthesize:
- Simple and easy to implement control laws.
- Minimum dwell time guarantees.

✓Restriction to polynomial certificates.
- Incomplete.
- Workaround: partition original control problem into smaller sub-problems.

✓Adds value to other existing approaches.

Thank You

This work was funded in part by NSF under award numbers SHF 1527075 and CPS 1646556.