Control Synthesis for Large Collections of Systems with Mode-Counting Constraints

Petter Nilsson, Necmiye Ozay

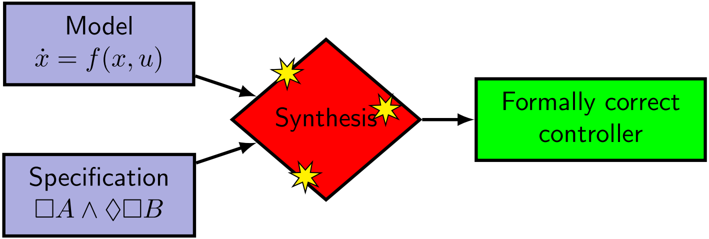
2016 Conference on Hybrid Systems: Computation and Control
Vienna, April 14, 2016

UNIVERSITY OF MICHIGAN

Research funded by NSF

What is Synthesis?

- Problem: Given a dynamic model and a specification, synthesize a controller that enforces the specification



Summary of contributions

Some state-of-the-art methods:

- Abstraction-based
- HJB equation
- Reachable set computation

Summary of contributions

Some state-of-the-art methods:

- Abstraction-based
- HJB equation
- Reachable set computation
- They all suffer from the curse of dimensionality!

Summary of contributions

Some state-of-the-art methods:

- Abstraction-based
- HJB equation
- Reachable set computation
- They all suffer from the curse of dimensionality!

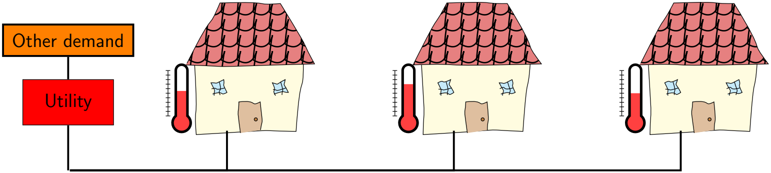
In this work:

- Scalable synthesis for mode-counting problems
- Scalability by exploiting symmetry in dynamics and spec
- Applied to a system with 20,000 state variables

Outline

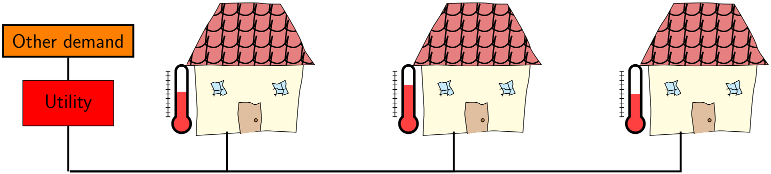
Thermostatically Controlled Load (TCL) Scheduling I

- A TCL can be `on` or `off`
- State constraint: Each TCL should maintain temperature within a desired temperature range
- Specification: Aggregate electricity consumption should be controlled over time
- The flexibility in individual specifications can be leveraged to control aggregate demand to for instance mitigate fluctuations



Thermostatically Controlled Load (TCL) Scheduling I

- A TCL can be `on` or `off`
- State constraint: Each TCL should maintain temperature within a desired temperature range
- Specification: Aggregate electricity consumption should be controlled over time
- The flexibility in individual specifications can be leveraged to control aggregate demand to for instance mitigate fluctuations



- Previous work on TCL scheduling does not provide guarantees

Thermostatically Controlled Load (TCL) Scheduling II

Assumptions

- All TCLs identical with respect to dynamics
- All TCLs have the same desired temperature range

Thermostatically Controlled Load (TCL) Scheduling II

Assumptions

- All TCLs identical with respect to dynamics
- All TCLs have the same desired temperature range

Problem features:

- Subsystem dynamics independent and identical
- Subsystems coupled through aggregate demand
- Permutation symmetry in aggregate demand

Outline

Mode-Counting Problem

- Given:
  - N identical subsystems x_i with switched dynamics

$$\dot{x}_i(t) = f_{\sigma_i(t)}(x_i(t)), \quad \sigma_i(t) \in \{1, 2, \ldots, M\}$$

  - State constraint: unsafe set $\mathcal{U}$

$$x_i(t) \notin \mathcal{U}$$

- Objective: find switching strategy that enforces state constraints and bounds on aggregate number of subsystems in each mode

$$\underline{K}_m \leq \sum_{i=1}^{N} \mathbb{1}_{\{m\}}\left(\sigma_i(t)\right) \leq \overline{K}_m \quad \forall t > 0$$

Technical Assumption

- The individual dynamics are incrementally stable for all modes
  - There is a $\mathcal{KL}$-function β_m s.t. the flow ϕ_t^m of $\dot{x} = f_m(x)$ satisfies

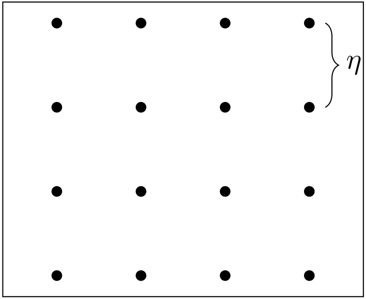
$$\|\phi_t^m(x) - \phi_t^m(y)\| \leq \beta_m \left(\|x - y\|, t\right)$$

Outline

Abstraction of a Single Subsystem

Construct abstraction in space (η) and time (τ)

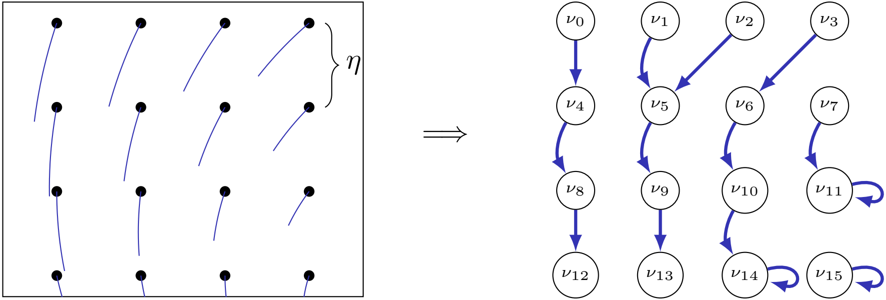
1. Grid state space of an individual system using a grid size η



Abstraction of a Single Subsystem

Construct abstraction in space (η) and time (τ)

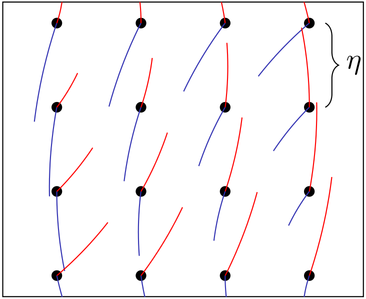
1. Grid state space of an individual system using a grid size η
2. Determine transitions by simulating trajectories on $[0, \tau]$ starting in every grid point, for each mode



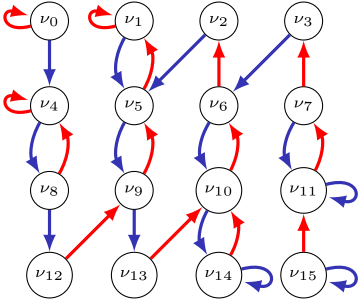
Abstraction of a Single Subsystem

Construct abstraction in space (η) and time (τ)

1. Grid state space of an individual system using a grid size η
2. Determine transitions by simulating trajectories on $[0, \tau]$ starting in every grid point, for each mode



$\Longrightarrow$



Abstraction Features

- Deterministic
- Under stability assumption and if $\beta(\epsilon, \tau) \leq \epsilon - \eta/2$, abstraction is ϵ-approximately bisimilar to original time-sampled system

Aggregate Dynamics on Abstraction

- Now consider N identical subsystems
- Let the abstraction states be $V = \{\nu_1, \dots \nu_K\}$

Aggregate Dynamics on Abstraction

- Now consider N identical subsystems
- Let the abstraction states be $V = \{\nu_1, \ldots \nu_K\}$

Introduce

- aggregate states w_k^m: the number of subsystems in mode m at the abstraction state ν_k
- aggregate controls $r_k^{m_1,m_2}$: the number of systems at ν_k that switches from mode m_1 to mode m_2

Aggregate Dynamics on Abstraction

- Now consider N identical subsystems
- Let the abstraction states be $V = \{\nu_1, \ldots \nu_K\}$

Introduce

- aggregate states w_k^m: the number of subsystems in mode m at the abstraction state ν_k
- aggregate controls $r_k^{m_1,m_2}$: the number of systems at ν_k that switches from mode m_1 to mode m_2

Aggregate dynamics as discrete-time LTI system

$$\left(w_k^{m_1}\right)^+ = \sum_{j \in \mathcal{N}_k^{m_1}} \left(w_j^{m_1} + \sum_{m_2 \,:\, m_2 \neq m_1} \left(r_j^{m_2,m_1} - r_j^{m_1,m_2} \right) \right)$$

Compact representation $\mathbf{w}^+ = A\mathbf{w} + B\mathbf{r}$

Feasibility of Mode-Counting Problem on Abstraction

The abstraction is ϵ-approximately bisimilar to the original system

Theorem

If there is a solution of the mode-counting problem with margin $+\epsilon$ for the abstraction, there is a solution for the original system.

Theorem

If there is no solution of the mode-counting problem with margin $-\epsilon$ for the abstraction, there is no solution for the original system.

- Margin $+\epsilon$: unsafe sets are enlarged by ϵ
- Margin $-\epsilon$: unsafe sets are shrunk by ϵ

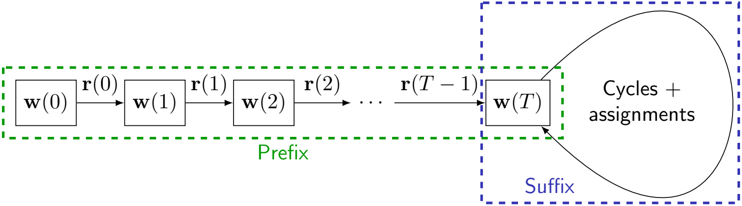
Outline

Solution as Prefix and Suffix

- Must guarantee mode-counting bounds and respect state constraints over an infinite horizon

Solution as Prefix and Suffix

- Must guarantee mode-counting bounds and respect state constraints over an infinite horizon
- Idea: make synthesis problem finite-dimensional by steering to graph cycles



Cycle Concepts

- A cycle assignment for a cycle $C = \{\nu_1, \ldots, \nu_{|C|}\}$ of length $|C|$ is a mapping $\alpha : \{1, \ldots, |C|\} \to \mathbb{R}^+$

[1] $\Xi_C(\nu_i)$ is the outgoing mode at ν_i in C, i.e., the mode of edge (ν_i, ν_{i+1})

Cycle Concepts

- A cycle assignment for a cycle $C = \{\nu_1, \ldots, \nu_{|C|}\}$ of length $|C|$ is a mapping $\alpha : \{1, \ldots, |C|\} \to \mathbb{R}^+$
- Maximal mode-m-count: maximal number of subsystems simultaneously in mode m of cyclical α-permutations in C

$$\overline{\Psi}^m(C, \alpha) = \max_k \sum_{i:\, \Xi_C(\nu_i) = m} \alpha\left((k+i) \mod |C|\right)$$ [1]

[1] $\Xi_C(\nu_i)$ is the outgoing mode at ν_i in C, i.e., the mode of edge (ν_i, ν_{i+1})

Cycle Concepts

- A cycle assignment for a cycle $C = \{\nu_1, \ldots, \nu_{|C|}\}$ of length $|C|$ is a mapping $\alpha : \{1, \ldots, |C|\} \to \mathbb{R}^+$
- Maximal mode-m-count: maximal number of subsystems simultaneously in mode m of cyclical α-permutations in C

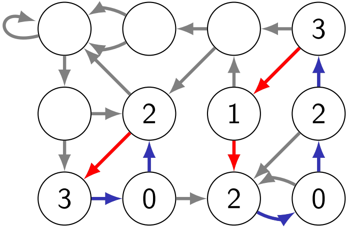
$$\overline{\Psi}^m(C, \alpha) = \max_k \sum_{i:\, \Xi_C(\nu_i) = m} \alpha\left((k+i) \mod |C|\right)^{1}$$

- Minimal mode-m-count: minimal number of subsystems simultaneously in mode m of cyclical α-permutations in C

$$\underline{\Psi}^m(C, \alpha) = \min_k \sum_{i:\, \Xi_C(\nu_i) = m} \alpha\left((k+i) \mod |C|\right)$$

[1] $\Xi_C(\nu_i)$ is the outgoing mode at ν_i in C, i.e., the mode of edge (ν_i, ν_{i+1})

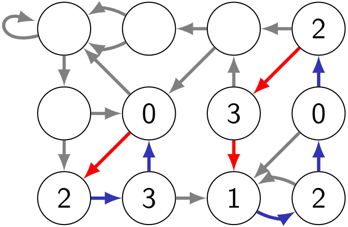
Illustration: Cycles



- Small cycle C_1, assignment $\alpha_1 = [3, 0, 2]$, gives red mode counts

- Big cycle C_2, assignment $\alpha_2 = [2, 0, 2, 3, 1]$, gives red mode counts

Illustration: Cycles



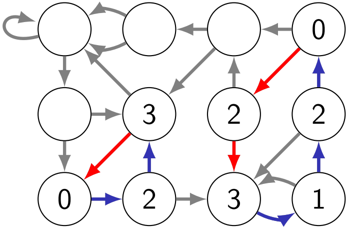
- Small cycle C_1, assignment $\alpha_1 = [3, 0, 2]$, gives red mode counts

$$\underline{\Psi}(C_1, \alpha_1) = 0$$

- Big cycle C_2, assignment $\alpha_2 = [2, 0, 2, 3, 1]$, gives red mode counts

$$\overline{\Psi}(C_2, \alpha_2) = 5$$

Illustration: Cycles



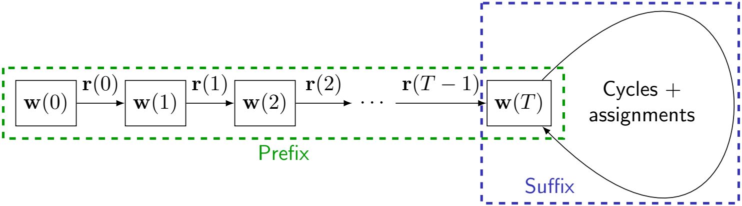
- Small cycle C_1, assignment $\alpha_1 = [3, 0, 2]$, gives red mode counts

$$\underline{\Psi}(C_1, \alpha_1) = 0 \quad \overline{\Psi}(C_1, \alpha_1) = 3$$

- Big cycle C_2, assignment $\alpha_2 = [2, 0, 2, 3, 1]$, gives red mode counts

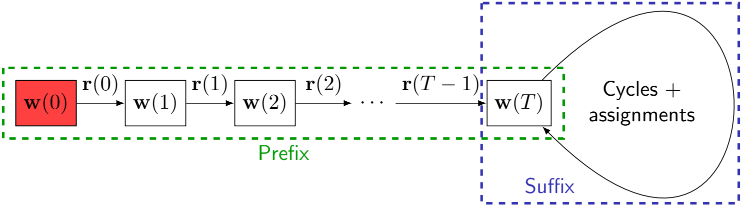
$$\underline{\Psi}(C_2, \alpha_2) = 2 \quad \overline{\Psi}(C_2, \alpha_2) = 5$$

Solution as Prefix and Suffix Parts



- Given: Given: initial positions $\lambda_0 \in \mathbb{N}^K$, mode-counting bounds $\underline{K}_m, \overline{K}_m$, set of cycles $\{C_j\}_{j \in J}$, horizon T
- Find: cycle assignments $\alpha_1, \ldots, \alpha_J$, aggregate states $\mathbf{w}(0), \ldots, \mathbf{w}(T)$, aggregate controls $\mathbf{r}(0), \ldots, \mathbf{r}(T-1)$
- Such that

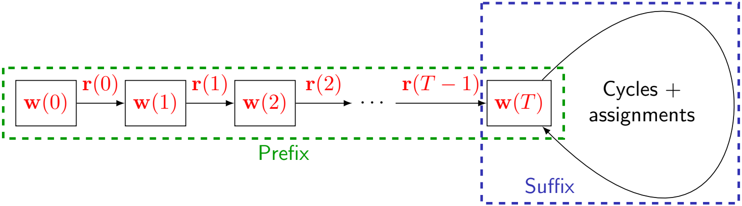
Solution as Prefix and Suffix Parts



- Given: Given: initial positions $\lambda_0 \in \mathbb{N}^K$, mode-counting bounds $\underline{K}_m, \overline{K}_m$, set of cycles $\{C_j\}_{j \in J}$, horizon T
- Find: cycle assignments $\alpha_1, \ldots, \alpha_J$, aggregate states $\mathbf{w}(0), \ldots, \mathbf{w}(T)$, aggregate controls $\mathbf{r}(0), \ldots, \mathbf{r}(T-1)$
- Such that
  - Initial aggregate state feasible

$$\Lambda(\mathbf{w}(0)) = \lambda_0$$

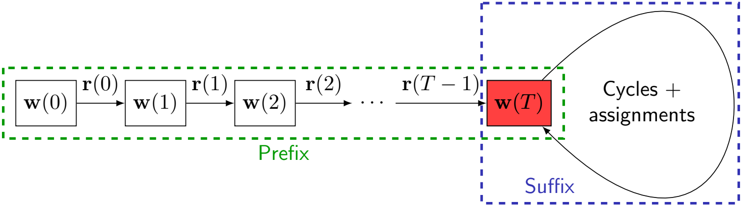
Solution as Prefix and Suffix Parts



- Given: Given: initial positions $\lambda_0 \in \mathbb{N}^K$, mode-counting bounds $\underline{K}_m, \overline{K}_m$, set of cycles $\{C_j\}_{j \in J}$, horizon T
- Find: cycle assignments $\alpha_1, \ldots, \alpha_J$, aggregate states $\mathbf{w}(0), \ldots, \mathbf{w}(T)$, aggregate controls $\mathbf{r}(0), \ldots, \mathbf{r}(T-1)$
- Such that
  - Dynamics obeyed

$$\mathbf{w}(s+1) = A\mathbf{w}(s) + B\mathbf{r}(s), \quad s = 0, \ldots, T-1$$

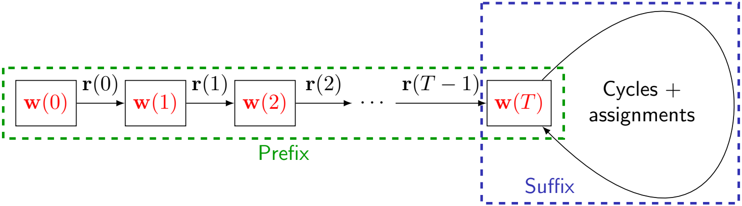
Solution as Prefix and Suffix Parts



- Given: Given: initial positions $\lambda_0 \in \mathbb{N}^K$, mode-counting bounds $\underline{K}_m, \overline{K}_m$, set of cycles $\{C_j\}_{j \in J}$, horizon T
- Find: cycle assignments $\alpha_1, \ldots, \alpha_J$, aggregate states $\mathbf{w}(0), \ldots, \mathbf{w}(T)$, aggregate controls $\mathbf{r}(0), \ldots, \mathbf{r}(T-1)$
- Such that
  - Prefix and suffix parts connected

$$\Lambda(\mathbf{w}(T)) = \sum_j \Phi_{C_j}(\alpha_j)$$

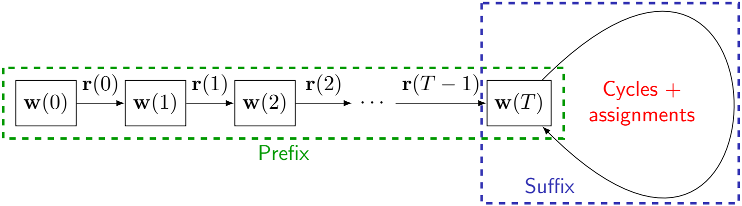
Solution as Prefix and Suffix Parts



- Given: Given: initial positions $\lambda_0 \in \mathbb{N}^K$, mode-counting bounds $\underline{K}_m, \overline{K}_m$, set of cycles $\{C_j\}_{j \in J}$, horizon T
- Find: cycle assignments $\alpha_1, \ldots, \alpha_J$, aggregate states $\mathbf{w}(0), \ldots, \mathbf{w}(T)$, aggregate controls $\mathbf{r}(0), \ldots, \mathbf{r}(T-1)$
- Such that
  - Prefix mode-counting constraints

$$\underline{K}_m \leq \sum_{k \in [K]} w_k^m(s) \leq \overline{K}_m, \quad s = 0, \ldots, T$$

Solution as Prefix and Suffix Parts



- Given: Given: initial positions $\lambda_0 \in \mathbb{N}^K$, mode-counting bounds $\underline{K}_m, \overline{K}_m$, set of cycles $\{C_j\}_{j \in J}$, horizon T
- Find: cycle assignments $\alpha_1, \ldots, \alpha_J$, aggregate states $\mathbf{w}(0), \ldots, \mathbf{w}(T)$, aggregate controls $\mathbf{r}(0), \ldots, \mathbf{r}(T-1)$
- Such that
  - Suffix mode-counting constraints

$$\underline{K}_m \leq \sum_j \underline{\Psi}^m(C_j, \alpha_j), \quad \sum_j \overline{\Psi}^m(C_j, \alpha_j) \leq \overline{K}^m$$

Linear Feasibility Problem

Given: initial positions $\lambda_0 \in \mathbb{N}^K$, mode-counting bounds $\underline{K}_m, \overline{K}_m$, set of cycles $\{C_j\}_{j \in J}$, horizon T,

$$
\begin{aligned}
\text{find} \quad & \alpha_1, \ldots, \alpha_J \text{ cycle assignments}, \\
& \mathbf{r}(0), \ldots, \mathbf{r}(T-1), \quad \mathbf{w}(0), \ldots, \mathbf{w}(T), \\
\text{s.t.} \quad & \underline{K}_m \leq \sum_{k \in [K]} w_k^m(s) \leq \overline{K}_m, \;\; s = 0, \ldots, T, \\
& \underline{K}_m \leq \sum_j \underline{\Psi}^m(C_j, \alpha_j), \;\; \sum_j \overline{\Psi}^m(C_j, \alpha_j) \leq \overline{K}^m, \\
& \Lambda(\mathbf{w}(T)) = \sum_j \Phi_{C_j}(\alpha_j), \\
& \mathbf{w}(s+1) = A\mathbf{w}(s) + B\mathbf{r}(s), \;\; s = 0, \ldots, T-1, \\
& \Lambda(\mathbf{w}(0)) = \lambda_0, \\
& \sum_{m_2} r_j^{m_1, m_2} = w_j^{m_1} \text{ for all } j \in \bigcup_{i \in U_{m_1}} \mathcal{N}_i^{m_1}, \\
& r_j^{m_2, m_1} = 0 \text{ for all } m_2 \in [M], j \in U_{m_1}, \\
& \text{state positivity constraints.}
\end{aligned}
$$

Feasibility Problem Analysis

The feasibility problem can be solved as a Linear Program (LP) or as an Integer Linear Program (ILP)

Proposition

If the ILP is feasible, the discrete mode-counting problem has a solution

Feasibility Problem Analysis

The feasibility problem can be solved as a Linear Program (LP) or as an Integer Linear Program (ILP)

Proposition

If the ILP is feasible, the discrete mode-counting problem has a solution

Theorem (Converse statement 1)

If there exists a solution to the discrete mode-counting problem, then there exists a finite horizon T and an integer L such that the ILP is feasible when solved for the cycle set consisting of all cycles of length at most L

Feasibility Problem Analysis

The feasibility problem can be solved as a Linear Program (LP) or as an Integer Linear Program (ILP)

Proposition

If the ILP is feasible, the discrete mode-counting problem has a solution

Theorem (Converse statement 1)

If there exists a solution to the discrete mode-counting problem, then there exists a finite horizon T and an integer L such that the ILP is feasible when solved for the cycle set consisting of all cycles of length at most L

Theorem (Converse statement 2)

If there exists a solution to the discrete mode-counting problem, then there exists a finite horizon T such that the LP is feasible when solved for the (finite) cycle set consisting of all simple cycles

Complexity of Feasibility Problem

- Number of variables

$$\mathcal{O}\left(M^2KT + \sum_{j\in J}|C_j|\right)$$

- Number of constraints

$$\mathcal{O}\left(MKT + \sum_{j\in J}|C_j|^2\right)$$

- Independent of number of subsystems $N \longrightarrow$ scalable!

Rounding a LP Solution

A non-integer assignment α for a cycle C can be rounded to an integer assignment at a cost in mode-m-counting bound at most

$$\min \left(\frac{|C|_m}{|C|} \left(|C| - \sum_i \alpha(i) \right), \left(1 - \frac{|C|_m}{|C|} \right) \sum_i \alpha(i) \right)^2$$

Procedure to avoid large ILPs

1. Solve LP feasibility problem
2. Round suffix part
3. Solve ILP with fixed suffix

[2] $|C|_m$ is the number of nodes in C with outgoing mode m

Outline

Example 1

Mode 1

$$\begin{bmatrix} \dot{x}_1^i \\ \dot{x}_2^i \end{bmatrix} = \begin{bmatrix} -(x_1^i - 1) + x_2^i \\ -(x_1^i - 1) - x_2^i - (x_2^i)^3 \end{bmatrix}$$

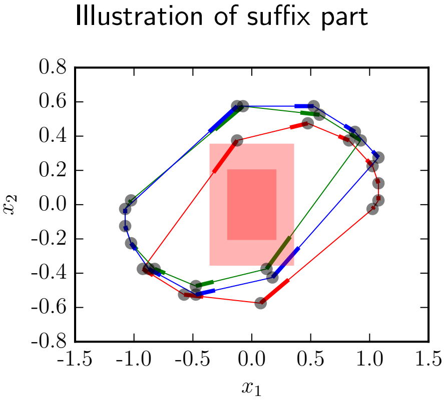
Mode 2

$$\begin{bmatrix} \dot{x}_1^i \\ \dot{x}_2^i \end{bmatrix} = \begin{bmatrix} -(x_1^i + 1) + x_2^i \\ -(x_1^i + 1) - x_2^i - (x_2^i)^3 \end{bmatrix}$$

- 10,000 subsystems
- Desired mode-1-count: 7000
- Unsafe set: $[-0.3, 0.3] \times [-0.2, 0.2] \subset \mathbb{R}^2$
- Solve using set of 100 random cycles

Example 1: Result

Repeatability Evaluated
Quality
Coverage
Instructions
Confirmed
Hybrid Systems Computation & Control



Mode-1-count guaranteed to be 7000 over time

Example 2: TCL Scheduling

TCL dynamics

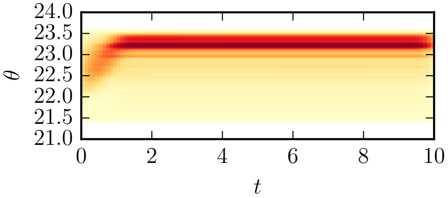
$$\dot{\theta}_i = -a(\theta_i - \theta_a) - bP_i^m$$

- Mode `on`: $P_i^m = 5.6$, Mode `off`: $P_i^m = 0$

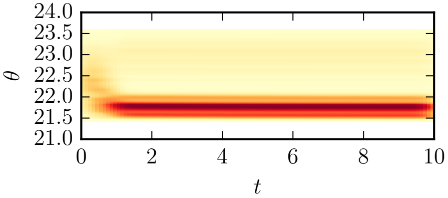
- Temperature range: $\theta_i \in [21.5, 23.5]$, outdoor temp $\theta_a = 32°$C
- 10,000 subsystems
- Solve for two desired mode-`on`-counts: 3600, 3200
- Use rounding algorithm to decrease computational complexity
- Relaxed constraints during prefix phase

Example 2: Result I

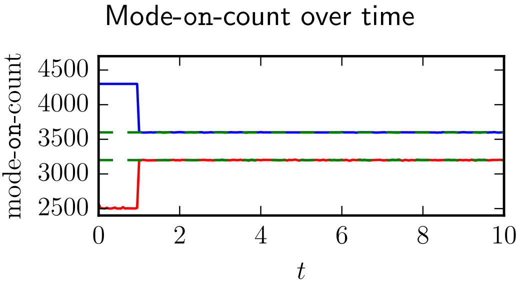
Low desired mode-on-count



High desired mode-on-count



Example 2: Result II



Guaranteed mode-counting bounds

Desired mode-count	low	high
Prefix phase bounds	$[2500, 2564]$	$[3696, 4300]$
Suffix phase bounds	$[3180, 3217]$	$[3595, 3604]$

Outline

Key takeaway

- Scalability can be achieved in formal methods by exploiting problem-specific symmetries
- Mode-counting specifications are permutation-symmetric

Generalizations and future work

- Classes of subsystems
- More general specifications over mode-counting quantities
- Generalized modes that include state-space regions
- Better rounding algorithms in order to avoid ILPs
- Robustness

Thank you for your attention

Conclusions from (In)feasibility

(I)LP	**Cycle set**	**T**	**Feasible?**	**Conclusions**
LP			Yes	Unknown
ILP			Yes	Solution
LP	Simple cycles	$\left(\!\binom{K}{N}\!\right)$	No	No solution
ILP	Length up to $\left(\!\binom{K}{N}\!\right)$	$\left(\!\binom{K}{N}\!\right)$	No	No solution