Synthesizing Robust Systems

Paulo Tabuada

Ayca Balkan, Sina Caliskan, Yasser Shoukry, Rupak Majumdar (MPI)

Cyber-Physical Systems Laboratory
Department of Electrical Engineering
University of California at Los Angeles
Software systems are designed based on assumptions about their environment.
Software systems are designed based on assumptions about their environment. But the real environment is either unknown at design time or changing over time.
Robustness
The need for robustness

- Software systems are designed based on assumptions about their environment.
- But the real environment is either unknown at design time or changing over time.
- Hence, the assumptions will be violated and current design methodologies offer no assurance on how software behaves when such violations occur.
Robustness
The need for robustness

- Software systems are designed based on assumptions about their environment.
- But the **real** environment is either unknown at design time or changing over time.
- Hence, the assumptions will be violated and current design methodologies offer no assurance on how software behaves when such violations occur.

   Ideally, we would like a modest deviation from the assumptions to lead to a modest deviation from the nominal correctness guarantees.
Software systems are designed based on assumptions about their environment.

But the real environment is either unknown at design time or changing over time.

Hence, the assumptions will be violated and current design methodologies offer no assurance on how software behaves when such violations occur.

Ideally, we would like a modest deviation from the assumptions to lead to a modest deviation from the nominal correctness guarantees.

Robustness!
Robustness
Motivation from control theory

Our starting point:

- Robustness is a very familiar concept in control theory;
Robustness
Motivation from control theory

Our starting point:

- Robustness is a very familiar concept in control theory;
- It is well understood that the models (assumptions) used for controller design are precious but always wrong:
  - Weight of a car (1 passenger vs 5 passengers);
  - Aerodynamic characteristics of a car (surfboard on the top of the car or bicycle mounted on a rack in the back);
  - etc.
Robustness
Motivation from control theory

Our starting point:

- Robustness is a very familiar concept in control theory;
- It is well understood that the models (assumptions) used for controller design are precious but always wrong:
  - Weight of a car (1 passenger vs 5 passengers);
  - Aerodynamic characteristics of a car (surfboard on the top of the car or bicycle mounted on a rack in the back);
  - etc.
- The most basic controller designs do not explicitly address robustness, but they are robust against unmodeled disturbances.
Robustness
Motivation from control theory

Our starting point:

- Robustness is a very familiar concept in control theory;
- It is well understood that the models (assumptions) used for controller design are precious but always wrong:
  - Weight of a car (1 passenger vs 5 passengers);
  - Aerodynamic characteristics of a car (surfboard on the top of the car or bicycle mounted on a rack in the back);
  - etc.
- The most basic controller designs do not explicitly address robustness, but they are robust against unmodeled disturbances.
- Can the same be done for software?
Robustness

What is known about software robustness?

In Computer Science:

- Recent work by Bloem, Chatterjee, Chaudhuri, Gulwani, Henzinger, Jobstman, Majumdar, ...

- Older work by Dijkstra (self-stabilizing algorithms).
Robustness
What is known about software robustness?

In Computer Science:
- Recent work by Bloem, Chatterjee, Chaudhuri, Gulwani, Henzinger, Jobstman, Majumdar, ...
- Older work by Dijkstra (self-stabilizing algorithms).

In Control Theory:
- There is a subfield of control theory called robust control;
- The following classification will be useful:
  - State based methods (modern view) (first part of the talk);
  - Input-output based methods (older view originated from the analysis of amplifiers and other electrical circuits) (second part of the talk).
State based robustness
Towards a definition

We start with a plain automaton.

Definition

A finite-state automaton is a triple $A = (Q, \Sigma, \delta)$ consisting of:

- A finite set of states $Q$;
- A finite set of (control) inputs $\Sigma$;
- A transition function $\delta : Q \times \Sigma \rightarrow Q$. 
We start with a plain automaton.

**Definition**

A finite-state automaton is a triple $A = (Q, \Sigma, \delta)$ consisting of:

- A finite set of states $Q$;
- A finite set of (control) inputs $\Sigma$;
- A transition function $\delta : Q \times \Sigma \rightarrow Q$.

How to reason about *modest* deviations from the nominal behavior?
State based robustness
Towards a definition

We introduce metric automata.

**Definition**

A finite-state metric automaton is a sextuple $A_\beta = (Q, d, \Sigma, X, \beta, \delta)$ consisting of:

- A finite set of states $Q$;
- A metric $d : Q \times Q \rightarrow \mathbb{R}^+_0$;
- A finite set of (control) inputs $\Sigma$;
- A finite set of (disturbance) inputs $X$ including a special symbol $\epsilon$ denoting nominal (no disturbance) behavior;
- A parameter $\beta \in \mathbb{R}^+_0$ defining the “power” of the disturbance;
- A transition function $\delta : Q \times \Sigma \times X \rightarrow Q$. 

It seems that we are explicitly modeling the disturbances through the transition function $\delta$. 

Paulo Tabuada (CyPhyLab - UCLA)
State based robustness
Towards a definition

We introduce metric automata.

**Definition**

A finite-state metric automaton is a sextuple $A_\beta = (Q, d, \Sigma, X, \beta, \delta)$ consisting of:

- A finite set of states $Q$;
- A metric $d : Q \times Q \to \mathbb{R}_0^+$;
- A finite set of (control) inputs $\Sigma$;
- A finite set of (disturbance) inputs $X$ including a special symbol $\epsilon$ denoting nominal (no disturbance) behavior;
- A parameter $\beta \in \mathbb{R}_0^+$ defining the “power” of the disturbance;
- A transition function $\delta : Q \times \Sigma \times X \to Q$.

It seems that we are explicitly modeling the disturbances through the transition function $\delta$. 
Nominal transition:

\[ q \xrightarrow{(\sigma, \epsilon)} \delta(q, \sigma, \epsilon) \]
State based robustness

Disturbance model

All the disturbed transitions:

\[ d(\delta(q, \sigma, \epsilon), \delta(q, \sigma, x)) \leq \beta \quad \forall q \in Q, \sigma \in \Sigma, x \in X. \]
State based robustness
Disturbance model

All the disturbed transitions:

\[ d(\delta(q, \sigma, \epsilon), \delta(q, \sigma, x)) \leq \beta \quad \forall q \in Q, \sigma \in \Sigma, x \in X. \]

The parameter \( \beta \) does not need to be known: results will be parameterized by \( \beta \).
State based robustness
Towards a definition

We consider first reachability objectives encoded by a set $F \subseteq Q$.

- A trace $s$ of $A_\beta$ is winning for a reachability objective $F$ if it enters $F$ in finite time.
State based robustness
Towards a definition

We consider first reachability objectives encoded by a set $F \subseteq Q$.

- A trace $s$ of $A_\beta$ is winning for a reachability objective $F$ if it enters $F$ in finite time.

Let $F = \{q_6\}$ be a reachability objective.
State based robustness
Towards a definition

We consider first reachability objectives encoded by a set $F \subseteq Q$.

- A trace $s$ of $A_\beta$ is winning for a reachability objective $F$ if it enters $F$ in finite time.

The shortest path strategy chooses the control input $b$ at every state.
State based robustness
Towards a definition

We consider first reachability objectives encoded by a set $F \subseteq Q$.

- A trace $s$ of $A_{\beta}$ is winning for a reachability objective $F$ if it enters $F$ in finite time.

The shortest path strategy chooses the control input $b$ at every state.

 Guarantee from $q_0$: some state in the green ellipsis will be reached in finite time.
State based robustness
Towards a definition

We consider first reachability objectives encoded by a set $F \subseteq Q$.

- A trace $s$ of $A_\beta$ is winning for a reachability objective $F$ if it enters $F$ in finite time.

Our strategy chooses the control input $a$ at every state.
State based robustness
Towards a definition

We consider first reachability objectives encoded by a set $F \subseteq Q$.

- A trace $s$ of $A_\beta$ is winning for a reachability objective $F$ if it enters $F$ in finite time.

Our strategy chooses the control input $a$ at every state.

Guarantee from $q_0$: some state in the blue ellipsis is reached in finite time.
Some standard definitions:

- A trace $s \in Q^* \cup Q^\omega$ of the automaton $A_\beta$ is a (finite or infinite) sequence of states $s = q_0 q_1 q_2 \ldots$ from $Q$ for which there exist control inputs $\sigma_0, \sigma_1, \sigma_2, \ldots$ and disturbance inputs $x_0, x_1, x_2, \ldots$ satisfying $\delta(q_i, \sigma_i, x_i) = q_{i+1}$ for $i \geq 0$;
Some standard definitions:

- A trace \( s \in Q^* \cup Q^\omega \) of the automaton \( A_\beta \) is a (finite or infinite) sequence of states \( s = q_0 q_1 q_2 \ldots \) from \( Q \) for which there exist control inputs \( \sigma_0, \sigma_1, \sigma_2, \ldots \) and disturbance inputs \( x_0, x_1, x_2, \ldots \) satisfying \( \delta(q_i, \sigma_i, x_i) = q_{i+1} \) for \( i \geq 0 \);

- A memoryless (control) strategy for an automaton \( A_\beta \) is a function \( S : Q \to \Sigma \) specifying a control input choice for each state \( q \in Q \);
State based robustness
Towards a definition

Some standard definitions:

- A **trace** \( s \in Q^* \cup Q^\omega \) of the automaton \( A_\beta \) is a (finite or infinite) sequence of states \( s = q_0 q_1 q_2 \ldots \) from \( Q \) for which there exist control inputs \( \sigma_0, \sigma_1, \sigma_2, \ldots \) and disturbance inputs \( x_0, x_1, x_2, \ldots \) satisfying \( \delta(q_i, \sigma_i, x_i) = q_{i+1} \) for \( i \geq 0 \);

- A **memoryless (control) strategy** for an automaton \( A_\beta \) is a function \( S : Q \rightarrow \Sigma \) specifying a control input choice for each state \( q \in Q \);

- A memoryless (control) strategy is winning for an automaton \( A_\beta \) if every trace of \( A_\beta \) complying with \( S : Q \rightarrow \Sigma \) satisfies the acceptance condition.
Definition

A winning strategy for the automaton $A_0$ and reachability objective $F \subseteq Q$ is $\gamma$-robust if for any $\beta \in \mathbb{R}_0^+$ it is winning for the automaton $A_\beta$ with reachability objective $B_{\gamma\beta}(F)$:

$$B_{\gamma\beta}(F) = \{ q \in Q \mid d(q, F) \leq \gamma\beta \}.$$
State based robustness

A definition

**Definition**

A winning strategy for the automaton $A_0$ and reachability objective $F \subseteq Q$ is $\gamma$-robust if for any $\beta \in \mathbb{R}_0^+$ it is winning for the automaton $A_\beta$ with reachability objective $\mathcal{B}_{\gamma\beta}(F)$:

$$\mathcal{B}_{\gamma\beta}(F) = \{q \in Q \mid d(q, F) \leq \gamma\beta\}.$$ 

- Note that if there are no disturbances, $\beta = 0$ and $\mathcal{B}_{\gamma\beta}(F) = F$. 
State based robustness

A definition

**Definition**

A winning strategy for the automaton $A_0$ and reachability objective $F \subseteq Q$ is $\gamma$-robust if for any $\beta \in \mathbb{R}_0^+$ it is winning for the automaton $A_\beta$ with reachability objective $B_{\gamma\beta}(F)$:

$$B_{\gamma\beta}(F) = \{q \in Q \mid d(q, F) \leq \gamma\beta\}.$$ 

- Note that if there are no disturbances, $\beta = 0$ and $B_{\gamma\beta}(F) = F$.
- The parameter $\gamma$ describes how much $F$ is inflated to obtain $B_{\gamma\beta}(F)$. 

State based robustness

A definition

Definition

A winning strategy for the automaton $A_0$ and reachability objective $F \subseteq Q$ is $\gamma$-robust if for any $\beta \in \mathbb{R}_0^+$ it is winning for the automaton $A_\beta$ with reachability objective $B_{\gamma\beta}(F)$:

$$B_{\gamma\beta}(F) = \{ q \in Q \mid d(q, F) \leq \gamma\beta \}. $$

- Note that if there are no disturbances, $\beta = 0$ and $B_{\gamma\beta}(F) = F$.
- The parameter $\gamma$ describes how much $F$ is inflated to obtain $B_{\gamma\beta}(F)$.
- The map transforming environment strategies to the language accepted by $A_\beta$ is uniformly continuous with modulus of continuity $\gamma$. 

State based robustness
Verification and synthesis

Given an automaton $A_0$, $\gamma \in \mathbb{R}_0^+$, and a strategy $S$ one can ask:

- **Verification**: Is $S$ $\gamma$-robust?
- **Optimal verification**: What is the smallest $\gamma \in \mathbb{R}_0^+$ for which $S$ is $\gamma$-robust?
State based robustness
Verification and synthesis

Given an automaton $A_0$, $\gamma \in \mathbb{R}_0^+$, and a strategy $S$ one can ask:

- **Verification**: Is $S$ $\gamma$-robust?
- **Optimal verification**: What is the smallest $\gamma \in \mathbb{R}_0^+$ for which $S$ is $\gamma$-robust?
- **Synthesis**: Can we synthesize a $\gamma$-robust strategy?
Given an automaton $A_0$, $\gamma \in \mathbb{R}_0^+$, and a strategy $S$ one can ask:

- **Verification**: Is $S$ $\gamma$-robust?
- **Optimal verification**: What is the smallest $\gamma \in \mathbb{R}_0^+$ for which $S$ is $\gamma$-robust?
- **Synthesis**: Can we synthesize a $\gamma$-robust strategy?
- **Optimal synthesis**: What is the smallest $\gamma \in \mathbb{R}_0^+$ for which we can synthesize a $\gamma$-robust strategy?

All the above problems can be reduced to dynamic programming and are thus polynomially solvable.
Given an automaton $A_0$, $\gamma \in \mathbb{R}^+_0$, and a strategy $S$ one can ask:

- **Verification**: Is $S$ $\gamma$-robust?
- **Optimal verification**: What is the smallest $\gamma \in \mathbb{R}^+_0$ for which $S$ is $\gamma$-robust?
- **Synthesis**: Can we synthesize a $\gamma$-robust strategy?
- **Optimal synthesis**: What is the smallest $\gamma \in \mathbb{R}^+_0$ for which we can synthesize a $\gamma$-robust strategy?

All the above problems can be reduced to dynamic programming and are thus polynomially solvable.

All these results extend to Büchi and parity objectives\(^1\).

---

\(^1\) A theory of $\omega$-regular robust software synthesis
Rupak Majumdar, Elaine Render, and Paulo Tabuada
State based robustness

Critical assessment

- Results for reachability objectives were obtained by a simple analogy with existing results in control theory.
State based robustness

Critical assessment

- Results for reachability objectives were obtained by a simple analogy with existing results in control theory.
- The fact that the results naturally extended to Büchi and parity objectives was rewarding.
State based robustness
Critical assessment

- Results for reachability objectives were obtained by a simple analogy with existing results in control theory.
- The fact that the results naturally extended to Büchi and parity objectives was rewarding.
- Along the way we had to extend known ideas towards robustness: equivalence between the existence of winning strategies and rank functions or progress measures.
State based robustness

Critical assessment

- Results for reachability objectives were obtained by a simple analogy with existing results in control theory.
- The fact that the results naturally extended to Büchi and parity objectives was rewarding.
- Along the way we had to extend known ideas towards robustness: equivalence between the existence of winning robust strategies and rank functions or progress measures control Lyapunov functions.
Results for reachability objectives were obtained by a simple analogy with existing results in control theory.

The fact that the results naturally extended to Büchi and parity objectives was rewarding.

Along the way we had to extend known ideas towards robustness: equivalence between the existence of winning robust strategies and rank functions or progress measures control Lyapunov functions.

State based robustness requires a metric.

What if I have two different automata defining the same language?

How to reason about robustness before having an implementation with states?

How to handle refinement and abstraction?
Rather than automata we now consider transducers $f : \Sigma^* \rightarrow \Lambda^*$;
- Rather than automata we now consider transducers $f : \Sigma^* \rightarrow \Lambda^*$;
- Rather than a metric we now use cost functions $I : \Sigma^* \rightarrow \mathbb{N}_0$ and $O : \Lambda^* \rightarrow \mathbb{N}_0$ to place costs on input and output strings, respectively;
Rather than automata we now consider transducers $f : \Sigma^* \rightarrow \Lambda^*$;

Rather than a metric we now use cost functions $I : \Sigma^* \rightarrow \mathbb{N}_0$ and $O : \Lambda^* \rightarrow \mathbb{N}_0$ to place costs on input and output strings, respectively;

A notion of robustness should have the following two properties:

- Bounded disturbances should lead to bounded consequences;
Rather than automata we now consider transducers $f : \Sigma^* \rightarrow \Lambda^*$;

Rather than a metric we now use cost functions $I : \Sigma^* \rightarrow \mathbb{N}_0$ and $O : \Lambda^* \rightarrow \mathbb{N}_0$ to place costs on input and output strings, respectively;

A notion of robustness should have the following two properties:

- Bounded disturbances should lead to bounded consequences;
- The effect of a sporadic disturbance should disappear in finitely many steps;
Input/output based robustness
Towards a definition

- Rather than automata we now consider transducers $f : \Sigma^* \to \Lambda^*$;
- Rather than a metric we now use cost functions $I : \Sigma^* \to \mathbb{N}_0$ and $O : \Lambda^* \to \mathbb{N}_0$ to place costs on input and output strings, respectively;
- A notion of robustness should have the following two properties:
  - Bounded disturbances should lead to bounded consequences;
  - The effect of a sporadic disturbance should disappear in finitely many steps;
- Well known requirements in control theory that recently appeared as two separate notions of robustness: $^2$ and $^3$.

---

$^2$ Synthesizing Robust Systems
R. P. Bloem, K. Greimel, T. Henzinger, B. Jobstmann

$^3$ Robustness of Sequential Circuits
L. Doyen, T.A. Henzinger, A. Legay, and D. Nickovic
Input/output based robustness
A definition

Some notation: \(|\sigma|\) denotes the length of the string \(\sigma \in \Sigma^*\) and \(\preceq\) denotes the prefix partial order.

Based on the control theoretic notion of Input-to-State Dynamic Stability we propose:

Definition
Given parameters \(\gamma, \eta \in \mathbb{N}\), we say the transducer \(f: \Sigma^* \rightarrow \Lambda^*\) is \((\gamma, \eta)\)-Input-Output Stable (IOS) if for each \(\sigma \in \Sigma^*\) we have:

\[ O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \gamma \| \sigma' \| - \eta \| \sigma \| \).

The parameter \(\gamma\) is called the robustness gain. It measures how much the disturbance is amplified. The parameter \(\eta\) is called the rate of decay. It measures how quickly the effects of a disturbance disappear.
Input/output based robustness

A definition

Some notation: $|\sigma|$ denotes the length of the string $\sigma \in \Sigma^*$ and $\preceq$ denotes the prefix partial order.

Based on the control theoretic notion of Input-to-State Dynamic Stability we propose:

**Definition**

Given parameters $\gamma, \eta \in \mathbb{N}$, we say the transducer $f : \Sigma^* \rightarrow \Lambda^*$ is $(\gamma, \eta)$-Input-Output Stable (IOS) if for each $\sigma \in \Sigma^*$ we have:

$$O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \left\{ \gamma I(\sigma') - \eta (|\sigma| - |\sigma'|) \right\}.$$
Input/output based robustness

A definition

Some notation: \(|\sigma|\) denotes the length of the string \(\sigma \in \Sigma^*\) and \(\preceq\) denotes the prefix partial order.

Based on the control theoretic notion of Input-to-State Dynamic Stability we propose:

**Definition**

Given parameters \(\gamma, \eta \in \mathbb{N}\), we say the transducer \(f : \Sigma^* \rightarrow \Lambda^*\) is \((\gamma, \eta)\)-Input-Output Stable (IOS) if for each \(\sigma \in \Sigma^*\) we have:

\[
O (f (\sigma)) \leq \max_{\sigma' \preceq \sigma} \{ \gamma l (\sigma') - \eta (|\sigma| - |\sigma'|) \}.
\]

- The parameter \(\gamma\) is called the *robustness gain*. It measures how much the disturbance is amplified.
Input/output based robustness

A definition

Some notation: $|\sigma|$ denotes the length of the string $\sigma \in \Sigma^*$ and $\preceq$ denotes the prefix partial order.

Based on the control theoretic notion of Input-to-State Dynamic Stability we propose:

**Definition**

Given parameters $\gamma, \eta \in \mathbb{N}$, we say the transducer $f : \Sigma^* \rightarrow \Lambda^*$ is $(\gamma, \eta)$-Input-Output Stable (IOS) if for each $\sigma \in \Sigma^*$ we have:

$$O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \{ \gamma l(\sigma') - \eta (|\sigma| - |\sigma'|) \}.$$  

- The parameter $\gamma$ is called the *robustness gain*. It measures how much the disturbance is amplified.
- The parameter $\eta$ is called the *rate of decay*. It measures how quickly the effects of a disturbance disappear.
Input/output based robustness
Some test cases

Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \leq \sigma} \left\{ \gamma l(\sigma') - \eta (|\sigma| - |\sigma'|) \right\} \]
Input/output based robustness
Some test cases

Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \leq \sigma} \{ \gamma I(\sigma') - \eta (|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs:

<table>
<thead>
<tr>
<th></th>
<th>(\sigma_1)</th>
<th>(\sigma_1\sigma_2)</th>
<th>(\sigma_1\sigma_2\sigma_3)</th>
<th>(\sigma_1\sigma_2\sigma_3\sigma_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(O \circ f)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Input/output based robustness

Some test cases

Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta (|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs:

<table>
<thead>
<tr>
<th></th>
<th>\sigma_1</th>
<th>\sigma_1\sigma_2</th>
<th>\sigma_1\sigma_2\sigma_3</th>
<th>\sigma_1\sigma_2\sigma_3\sigma_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(O \circ f)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

We have 3 prefixes of \(\sigma = \sigma_1\sigma_2\):
Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \geq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs:

<table>
<thead>
<tr>
<th></th>
<th>( \sigma_1 )</th>
<th>( \sigma_1\sigma_2 )</th>
<th>( \sigma_1\sigma_2\sigma_3 )</th>
<th>( \sigma_1\sigma_2\sigma_3\sigma_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( O \circ f )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

We have 3 prefixes of \( \sigma = \sigma_1\sigma_2 \):

for \( \sigma' = \sigma_1\sigma_2 \) we have \( \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 0 - \eta(2 - 2) = 0 \).

Paulo Tabuada (CyPhyLab - UCLA)
Input/output based robustness

Some test cases

Some intuition for this inequality.

\[
O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \{ \gamma l(\sigma') - \eta (|\sigma| - |\sigma'|) \}
\]

Consider the following sequence of input and output costs:

\[
\begin{array}{c|c|c|c|c}
   & \sigma_1 & \sigma_1\sigma_2 & \sigma_1\sigma_2\sigma_3 & \sigma_1\sigma_2\sigma_3\sigma_4 \\
\hline
   I & 0 & 0 & 0 & 0 \\
   O \circ f & 0 & 1 & 0 & 0 \\
\end{array}
\]

We have 3 prefixes of \( \sigma = \sigma_1\sigma_2 \):

for \( \sigma' = \sigma_1 \) we have \( \gamma l(\sigma') - \eta (|\sigma| - |\sigma'|) = \gamma 0 - \eta (2 - 1) = -\eta \).
Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs:

<table>
<thead>
<tr>
<th></th>
<th>(\sigma_1)</th>
<th>(\sigma_1\sigma_2)</th>
<th>(\sigma_1\sigma_2\sigma_3)</th>
<th>(\sigma_1\sigma_2\sigma_3\sigma_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(O \circ f)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

We have 3 prefixes of \(\sigma = \sigma_1\sigma_2\):

for \(\sigma' = \varepsilon\) we have \(\gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 0 - \eta(2 - 0) = -2\eta\).
Consider the following sequence of input and output costs:

\[
\begin{array}{c|ccccc}
& \sigma_1 & \sigma_1\sigma_2 & \sigma_1\sigma_2\sigma_3 & \sigma_1\sigma_2\sigma_3\sigma_4 \\
I & 0 & 0 & 0 & 0 \\
O \circ f & 0 & 1 & 0 & 0 \\
\end{array}
\]

We have 3 prefixes of \(\sigma = \sigma_1\sigma_2\):

for \(\sigma' = \varepsilon\) we have \(\gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 0 - \eta(2 - 0) = -2\eta\).

Hence, \(\max_{\sigma' \preceq \sigma} \{\gamma I(\sigma') - \eta(|\sigma| - |\sigma'|)\} = \max\{0, -\eta, -2\eta\} = 0\).
Input/output based robustness

Some test cases

Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs:

<table>
<thead>
<tr>
<th></th>
<th>$\sigma_1$</th>
<th>$\sigma_1\sigma_2$</th>
<th>$\sigma_1\sigma_2\sigma_3$</th>
<th>$\sigma_1\sigma_2\sigma_3\sigma_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$O \circ f$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

We have 3 prefixes of $\sigma = \sigma_1\sigma_2$:

for $\sigma' = \varepsilon$ we have $\gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 0 - \eta(2 - 0) = -2\eta$.

Hence, $\max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} = \max\{0, -\eta, -2\eta\} = 0$.

IOS requires $O(f(\sigma_1\sigma_2)) = 1 \leq 0$ which does not hold!
Input/output based robustness

Some test cases

Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \geq \sigma} \left\{ \gamma I(\sigma') - \eta (|\sigma| - |\sigma'|) \right\} \]

Consider the following sequence of input and output costs (persistent disturbance):

<table>
<thead>
<tr>
<th></th>
<th>(\sigma_1)</th>
<th>(\sigma_1\sigma_2)</th>
<th>(\sigma_1\sigma_2\sigma_3)</th>
<th>(\sigma_1\sigma_2\sigma_3\sigma_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I)</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(O \circ f)</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

We have 3 prefixes of \(\sigma = \sigma_1\sigma_2\):
Some intuition for this inequality.

\[
O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \left\{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \right\}
\]

Consider the following sequence of input and output costs (persistent disturbance):

<table>
<thead>
<tr>
<th></th>
<th>(\sigma_1)</th>
<th>(\sigma_1\sigma_2)</th>
<th>(\sigma_1\sigma_2\sigma_3)</th>
<th>(\sigma_1\sigma_2\sigma_3\sigma_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I)</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(O \circ f)</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

We have 3 prefixes of \(\sigma = \sigma_1\sigma_2\):

for \(\sigma' = \sigma_1\sigma_2\) we have \(\gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 2 - \eta(2 - 2) = 2\gamma\).
Input/output based robustness

Some test cases

Some intuition for this inequality.

\[
O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \}
\]

Consider the following sequence of input and output costs (persistent disturbance):

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>2</th>
<th>2</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I)</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(O \circ f)</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

We have 3 prefixes of \(\sigma = \sigma_1\sigma_2\):

for \(\sigma' = \sigma_1\) we have \(\gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 0 - \eta(2 - 1) = -\eta\).
Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs (persistent disturbance):

<table>
<thead>
<tr>
<th></th>
<th>\sigma_1</th>
<th>\sigma_1\sigma_2</th>
<th>\sigma_1\sigma_2\sigma_3</th>
<th>\sigma_1\sigma_2\sigma_3\sigma_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I)</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(O \circ f)</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

We have 3 prefixes of \(\sigma = \sigma_1\sigma_2\):

for \(\sigma' = \varepsilon\) we have \(\gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 0 - \eta(2 - 0) = -2\eta\).
Input/output based robustness
Some test cases

Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs (persistent disturbance):

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>( I )</th>
<th>( O \circ f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sigma_1 )</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \sigma_1\sigma_2 )</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>( \sigma_1\sigma_2\sigma_3 )</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>( \sigma_1\sigma_2\sigma_3\sigma_4 )</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

We have 3 prefixes of \( \sigma = \sigma_1\sigma_2 \):

for \( \sigma' = \varepsilon \) we have \( \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 0 - \eta(2 - 0) = -2\eta \).

Hence, \( \max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} = \max\{2\gamma, -\eta, -2\eta\} = 2\gamma \).
Input/output based robustness
Some test cases

Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs (persistent disturbance):

<table>
<thead>
<tr>
<th></th>
<th>(\sigma_1)</th>
<th>(\sigma_1\sigma_2)</th>
<th>(\sigma_1\sigma_2\sigma_3)</th>
<th>(\sigma_1\sigma_2\sigma_3\sigma_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I)</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(O \circ f)</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

We have 3 prefixes of \(\sigma = \sigma_1\sigma_2\):

for \(\sigma' = \varepsilon\) we have \(\gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 0 - \eta(2 - 0) = -2\eta\).

Hence, \(\max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} = \max\{2\gamma, -\eta, -2\eta\} = 2\gamma\).

IOS requires \(O(f(\sigma_1\sigma_2)) \leq 0 \leq 2\gamma\).
Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \leq \sigma} \left\{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \right\} \]

Consider the following sequence of input and output costs (persistent disturbance):

<table>
<thead>
<tr>
<th></th>
<th>(\sigma_1)</th>
<th>(\sigma_1\sigma_2)</th>
<th>(\sigma_1\sigma_2\sigma_3)</th>
<th>(\sigma_1\sigma_2\sigma_3\sigma_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I)</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(O \circ f)</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

We have 3 prefixes of \(\sigma = \sigma_1\sigma_2\):

for \(\sigma' = \varepsilon\) we have \(\gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 0 - \eta(2 - 0) = -2\eta\).

Hence, \(\max_{\sigma' \leq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} = \max\{2\gamma, -\eta, -2\eta\} = 2\gamma\).

IOS requires \(O(f(\sigma_1\sigma_2)) = 0 \leq 2\gamma\). At this point we can take \(\gamma = 0\).
Input/output based robustness

Some test cases

Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \leq \sigma} \{ \gamma I(\sigma') - \eta (|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs (persistent disturbance):

<table>
<thead>
<tr>
<th></th>
<th>( \sigma_1 )</th>
<th>( \sigma_1\sigma_2 )</th>
<th>( \sigma_1\sigma_2\sigma_3 )</th>
<th>( \sigma_1\sigma_2\sigma_3\sigma_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I )</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>( O \circ f )</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

We have 4 prefixes of \( \sigma = \sigma_1\sigma_2\sigma_3 \):
Input/output based robustness

Some test cases

Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \leq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs (persistent disturbance):

<table>
<thead>
<tr>
<th></th>
<th>$\sigma_1$</th>
<th>$\sigma_1\sigma_2$</th>
<th>$\sigma_1\sigma_2\sigma_3$</th>
<th>$\sigma_1\sigma_2\sigma_3\sigma_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I$</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$O \circ f$</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

We have 4 prefixes of $\sigma = \sigma_1\sigma_2\sigma_3$:

for $\sigma' = \sigma_1\sigma_2\sigma_3$ we have $\gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 2 - \eta(3 - 3) = 2\gamma$. 
Input/output based robustness

Some test cases

Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs (persistent disturbance):

<table>
<thead>
<tr>
<th></th>
<th>( \sigma_1 )</th>
<th>( \sigma_1\sigma_2 )</th>
<th>( \sigma_1\sigma_2\sigma_3 )</th>
<th>( \sigma_1\sigma_2\sigma_3\sigma_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I )</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>( O \circ f )</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

We have 4 prefixes of \( \sigma = \sigma_1\sigma_2\sigma_3 \):

for \( \sigma' = \sigma_1\sigma_2 \) we have \( \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 2 - \eta(3 - 2) = 2\gamma - \eta \).
Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \subseteq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs (persistent disturbance):

<table>
<thead>
<tr>
<th>\sigma</th>
<th>\sigma_1</th>
<th>\sigma_1\sigma_2</th>
<th>\sigma_1\sigma_2\sigma_3</th>
<th>\sigma_1\sigma_2\sigma_3\sigma_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>\sigma_1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>\sigma_2</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

We have 4 prefixes of \( \sigma = \sigma_1\sigma_2\sigma_3 \):

for \( \sigma' = \sigma_1 \) we have \( \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 0 - \eta(3 - 1) = -2\eta \).
Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs (persistent disturbance):

\[
\begin{array}{c|c|c|c|c}
 & \sigma_1 & \sigma_1\sigma_2 & \sigma_1\sigma_2\sigma_3 & \sigma_1\sigma_2\sigma_3\sigma_4 \\
\hline
I & 0 & 2 & 2 & 2 \\
O \circ f & 0 & 0 & 4 & 4 \\
\end{array}
\]

We have 4 prefixes of \( \sigma = \sigma_1\sigma_2\sigma_3 \):

for \( \sigma' = \varepsilon \) we have

\[ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 0 - \eta(3 - 0) = -3\eta. \]
Input/output based robustness

Some test cases

Some intuition for this inequality.

\[
O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \left\{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \right\}
\]

Consider the following sequence of input and output costs (persistent disturbance):

\[
\begin{array}{c|c|c|c|c}
\sigma_1  & \sigma_1\sigma_2  & \sigma_1\sigma_2\sigma_3  & \sigma_1\sigma_2\sigma_3\sigma_4 \\
\hline
I         & 0         & 2         & 2         & 2         \\
O \circ f & 0         & 0         & 4         & 4         \\
\end{array}
\]

We have 4 prefixes of \( \sigma = \sigma_1\sigma_2\sigma_3 \):

for \( \sigma' = \epsilon \) we have \( \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 0 - \eta(3 - 0) = -3\eta \).

Hence, \( \max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} = \max\{2\gamma, 2\gamma - \eta, -\eta, -2\eta\} = 2\gamma \).
Input/output based robustness

Some test cases

Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs (persistent disturbance):

<table>
<thead>
<tr>
<th></th>
<th>(\sigma_1)</th>
<th>(\sigma_1\sigma_2)</th>
<th>(\sigma_1\sigma_2\sigma_3)</th>
<th>(\sigma_1\sigma_2\sigma_3\sigma_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I)</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(O \circ f)</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

We have 4 prefixes of \(\sigma = \sigma_1\sigma_2\sigma_3\):

for \(\sigma' = \varepsilon\) we have \(\gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) = \gamma 0 - \eta(3 - 0) = -3\eta\).

Hence, \(\max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta(|\sigma| - |\sigma'|) \} = \max\{2\gamma, 2\gamma - \eta, -\eta, -2\eta\} = 2\gamma\).

IOS requires \(O(f(\sigma_1\sigma_2\sigma_3)) = 4 \leq 2\gamma\).
Input/output based robustness

Some test cases

Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta (|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs (persistent disturbance):

<table>
<thead>
<tr>
<th></th>
<th>(\sigma_1)</th>
<th>(\sigma_1\sigma_2)</th>
<th>(\sigma_1\sigma_2\sigma_3)</th>
<th>(\sigma_1\sigma_2\sigma_3\sigma_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I)</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(O \circ f)</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

We have 4 prefixes of \(\sigma = \sigma_1\sigma_2\sigma_3\):

for \(\sigma' = \varepsilon\) we have \(\gamma I(\sigma') - \eta (|\sigma| - |\sigma'|) = \gamma \cdot 0 - \eta (3 - 0) = -3\eta\).

Hence, \(\max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta (|\sigma| - |\sigma'|) \} = \max\{2\gamma, 2\gamma - \eta, -\eta, -2\eta\} = 2\gamma\).

IOS requires \(O(f(\sigma_1\sigma_2\sigma_3)) = 4 \leq 2\gamma\). A similar analysis for the remaining strings leads to IOS with \(\gamma = 2\).
Input/output based robustness

Some test cases

Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \{ \gamma I(\sigma') - \eta (|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs (sporadic disturbance):

<table>
<thead>
<tr>
<th></th>
<th>(\sigma_1)</th>
<th>(\sigma_1\sigma_2)</th>
<th>(\sigma_1\sigma_2\sigma_3)</th>
<th>(\sigma_1\sigma_2\sigma_3\sigma_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I)</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(O \circ f)</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>
Input/output based robustness

Some test cases

Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \left\{ \gamma I(\sigma') - \eta (|\sigma| - |\sigma'|) \right\} \]

Consider the following sequence of input and output costs (sporadic disturbance):

<table>
<thead>
<tr>
<th></th>
<th>(\sigma_1)</th>
<th>(\sigma_1\sigma_2)</th>
<th>(\sigma_1\sigma_2\sigma_3)</th>
<th>(\sigma_1\sigma_2\sigma_3\sigma_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I)</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(O \circ f)</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

A similar analysis leads to the following constraints:

\[
\begin{align*}
O(f(\sigma_1)) &= 0 \leq 2\gamma \\
O(f(\sigma_1\sigma_2)) &= 4 \leq 2\gamma - \eta \\
O(f(\sigma_1\sigma_2\sigma_3)) &= 3 \leq 2\gamma - 2\eta \\
O(f(\sigma_1\sigma_2\sigma_3\sigma_4)) &= 2 \leq 2\gamma - 3\eta 
\end{align*}
\]
Input/output based robustness

Some test cases

Some intuition for this inequality.

\[ O(f(\sigma)) \leq \max_{\sigma' \leq \sigma} \{ \gamma I(\sigma') - \eta (|\sigma| - |\sigma'|) \} \]

Consider the following sequence of input and output costs (sporadic disturbance):

<table>
<thead>
<tr>
<th></th>
<th>(\sigma_1)</th>
<th>(\sigma_1\sigma_2)</th>
<th>(\sigma_1\sigma_2\sigma_3)</th>
<th>(\sigma_1\sigma_2\sigma_3\sigma_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I)</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(O \circ f)</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

A similar analysis leads to the following constraints:

\[
\begin{align*}
O(f(\sigma_1)) &= 0 \leq 2\gamma = 6 \\
O(f(\sigma_1\sigma_2)) &= 4 \leq 2\gamma - \eta = 6 - 1 = 5 \\
O(f(\sigma_1\sigma_2\sigma_3)) &= 3 \leq 2\gamma - 2\eta = 6 - 2 = 4 \\
O(f(\sigma_1\sigma_2\sigma_3\sigma_4)) &= 2 \leq 2\gamma - 3\eta = 6 - 3 = 3
\end{align*}
\]

IOS holds for \(\gamma = 3\) and \(\eta = 1\).
Based on the control theoretic notion of Input-to-State Dynamic Stability we propose:

**Definition**

Given parameters $\gamma, \eta \in \mathbb{N}$, we say the transducer $f : \Sigma^* \rightarrow \Lambda^*$ is $(\gamma, \eta)$-input-output stable if for each $\sigma \in \Sigma^*$ we have

$$O(f(\sigma)) \leq \max_{\sigma' \preceq \sigma} \left\{ \gamma I(\sigma') - \eta (|\sigma| - |\sigma'|) \right\}.$$ 

- The parameter $\gamma$ is called the *robustness gain*. It measures how much the disturbance is amplified.
- The parameter $\eta$ is called the *rate of decay*. It measures how quickly the effects of a disturbance disappear.
- The notion of $(\gamma, \eta)$-input-output stability captures the two desired properties:
  - Bounded disturbances should lead to bounded consequences;
  - The effect of a sporadic disturbance should disappear in finitely many steps;
When is a transducer IOS?

**Problem (\((\gamma, \eta)\)-IOS Verification)**

*Given a transducer* \( f : \Sigma^* \rightarrow \Lambda^* \), *input and output cost functions* \( I : \Sigma^* \rightarrow \mathbb{N}_0 \) *and* \( O : \Lambda^* \rightarrow \mathbb{N}_0 \), *respectively, and parameters* \( \gamma, \eta \in \mathbb{N} \), *is the transducer* \( f (\gamma, \eta) \)-IOS *with respect to* \((I, O)\)?
When is a transducer IOS?

**Problem \(((\gamma, \eta))-IOS\) Verification**

*Given a transducer* \(f : \Sigma^* \rightarrow \Lambda^*\), *input and output cost functions* \(I : \Sigma^* \rightarrow \mathbb{N}_0\) *and* \(O : \Lambda^* \rightarrow \mathbb{N}_0\), *respectively, and parameters* \(\gamma, \eta \in \mathbb{N}\), *is the transducer* \(f (\gamma, \eta)\)-IOS *with respect to* \((I, O)\) ?

**Problem (IOS Verification)**

*Given a transducer* \(f : \Sigma^* \rightarrow \Lambda^*\) *and input and output cost functions* \(I : \Sigma^* \rightarrow \mathbb{N}_0\) *and* \(O : \Lambda^* \rightarrow \mathbb{N}_0\), *respectively, does there exist* \(\gamma, \eta \in \mathbb{N}\) *such that* \(f\) *is* \((\gamma, \eta)\)-IOS *with respect to* \((I, O)\)? *If so, find all such* \(\gamma\) *and* \(\eta\).
Robustness
Solving the verification problem

Assume that \( f, I, \) and \( O \) are defined by finite-state (weighted) automata and compose them in the single automaton \( A \):

\[
\begin{array}{c}
\text{f} \\
\text{I} \\
\text{O}
\end{array}
\]

We now consider the lattice \( M_Q \) of functions from the set of states \( Q \) of \( A \) to \( M = \{1, 2, \ldots, \gamma_w\} \) where \( \gamma_w \) is the largest weight in the automaton defining \( I \).

On \( M_Q \) we can define the operator \( F : M_Q \rightarrow M_Q \) given by:

\[
F(W)(q) = \max_{\gamma} H_I(q), W(q), \min_{q' \in \text{Pre}(q)} W(q') - \eta_{ff}.
\]
Robustness
Solving the verification problem

Assume that $f$, $I$, and $O$ are defined by finite-state (weighted) automata and compose them in the single automaton $A$:

We now consider the lattice $M^Q$ of functions from the set of states $Q$ of $A$ to $M = \{1, 2, \ldots, \gamma w\}$ where $w$ is the largest weight in the automaton defining $I$. 
Robustness
Solving the verification problem

Assume that $f$, $I$, and $O$ are defined by finite-state (weighted) automata and compose them in the single automaton $A$:

We now consider the lattice $M^Q$ of functions from the set of states $Q$ of $A$ to $M = \{1, 2, \ldots, \gamma w\}$ where $w$ is the largest weight in the automaton defining $I$.

On $M^Q$ we can define the operator $F : M^Q \rightarrow M^Q$ given by:

$$F(W)(q) = \max \left\{ \gamma H'(q), W(q), \min_{q' \in \text{Pre}(q)} W(q') - \eta \right\}.$$
Robustness
Solving the verification problem

**Theorem ((\(\gamma, \eta\))-IOS Verification)**

Let \(f : \Sigma^* \rightarrow \Lambda^*\), \(l : \Sigma^* \rightarrow \mathbb{N}_0\), and \(O : \Lambda^* \rightarrow \mathbb{N}_0\) be defined by (weighted) finite state automata. Given \(\eta, \gamma \in \mathbb{N}\), the transducer \(f\) is \((\gamma, \eta)\)-IOS with respect to \((l, O)\) iff the infimal fixed point of \(F\), denoted by \(W^*\), satisfies the following inequality for every \(q \in Q\):

\[H^O(q) \leq W^*(q).\]

Note that \(W^*\) is computed in \(O(|Q| \cdot |\gamma w|)\) steps.
Robustness
Solving the verification problem

Theorem \(((\gamma, \eta))-IOS\) Verification

Let \(f : \Sigma^* \rightarrow \Lambda^*\), \(I : \Sigma^* \rightarrow \mathbb{N}_0\), and \(O : \Lambda^* \rightarrow \mathbb{N}_0\) be defined by (weighted) finite state automata. Given \(\eta, \gamma \in \mathbb{N}\), the transducer \(f\) is \((\gamma, \eta))-IOS\) with respect to \((I, O)\) iff the infimal fixed point of \(F\), denoted by \(W^*\), satisfies the following inequality for every \(q \in Q\):

\[ H^O(q) \leq W^*(q). \]

Note that \(W^*\) is computed in \(O(|Q| \cdot |\gamma w|)\) steps.

- For the IOS verification problem, there exists a different operator whose fixed point characterizes the existence of \((\gamma, \eta)\) for which \(f\) is \((\gamma, \eta))-IOS\).
Robustness
Solving the verification problem

**Theorem ((\(\gamma, \eta\))-IOS Verification)**

Let \(f : \Sigma^* \rightarrow \Lambda^*\), \(I : \Sigma^* \rightarrow \mathbb{N}_0\), and \(O : \Lambda^* \rightarrow \mathbb{N}_0\) be defined by (weighted) finite state automata. Given \(\eta, \gamma \in \mathbb{N}\), the transducer \(f\) is \((\gamma, \eta)\)-IOS with respect to \((I, O)\) iff the infimal fixed point of \(F\), denoted by \(W^*\), satisfies the following inequality for every \(q \in Q\):

\[
H^O(q) \leq W^*(q).
\]

Note that \(W^*\) is computed in \(O(|Q| \cdot |\gamma w|)\) steps.

- For the IOS verification problem, there exists a different operator whose fixed point characterizes the existence of \((\gamma, \eta)\) for which \(f\) is \((\gamma, \eta)\)-IOS.
- Furthermore, we can compute all the values of \(\gamma\) (but only some of the values of \(\eta\)) for which \(f\) is \((\gamma, \eta)\)-IOS.
How about synthesis?
How about synthesis?

- The set of inputs $\Sigma$ is split as $\Sigma = \Sigma_c \times \Sigma_d$ with $\Sigma_c$ being control inputs and $\Sigma_d$ being disturbance inputs.
How about synthesis?

- The set of inputs $\Sigma$ is split as $\Sigma = \Sigma_c \times \Sigma_d$ with $\Sigma_c$ being control inputs and $\Sigma_d$ being disturbance inputs.
- A controller is a map $C : \Sigma^* \times \Sigma_c \rightarrow \Sigma_c$ transforming the history of past inputs $\sigma \in \Sigma^*$ and a given control input request $\sigma_c \in \Sigma_c$ into the control input $C(\sigma, \sigma_c)$. 

Paulo Tabuada (CyPhyLab - UCLA)

Synthesizing Robust Systems

ExCAPE Seminar 12/03/12
How about synthesis?

- The set of inputs $\Sigma$ is split as $\Sigma = \Sigma_c \times \Sigma_d$ with $\Sigma_c$ being control inputs and $\Sigma_d$ being disturbance inputs.

- A controller is a map $C : \Sigma^* \times \Sigma_c \rightarrow \Sigma_c$ transforming the history of past inputs $\sigma \in \Sigma^*$ and a given control input request $\sigma^c \in \Sigma_c$ into the control input $C(\sigma, \sigma^c)$.

Recall the automaton $A$: 

![Automaton Diagram](image-url)
How about synthesis?

- The set of inputs $\Sigma$ is split as $\Sigma = \Sigma_c \times \Sigma_d$ with $\Sigma_c$ being control inputs and $\Sigma_d$ being disturbance inputs.
- A controller is a map $C : \Sigma^* \times \Sigma_c \rightarrow \Sigma_c$ transforming the history of past inputs $\sigma \in \Sigma^*$ and a given control input request $\sigma^c \in \Sigma_c$ into the control input $C(\sigma, \sigma^c)$.

From $A$ we can construct a monitor $A_M$ for the $(\gamma, \eta)$-IOS property:

$$f \quad I \quad O \quad A \quad A_M$$

where the set of states of $A_M$ is $M = \{1, 2, \ldots, \gamma w\}$ with $w$ being the maximum weight of the automaton defining $I$. 
Robustness
Solving the synthesis problem

Theorem

Let $f : \Sigma^* \rightarrow \Lambda^*$, $l : \Sigma^* \rightarrow \mathbb{N}_0$, and $O : \Lambda^* \rightarrow \mathbb{N}_0$ be defined by (weighted) finite state automata. Given $\eta, \gamma \in \mathbb{N}$, the transducer $f$ is $(\gamma, \eta)$-IOS with respect to $(l, O)$ iff every reachable state $(q, m)$ of $A \times A_M$ satisfies $H^O(q) \leq m$. 
Robustness
Solving the synthesis problem

**Theorem**

Let \( f : \Sigma^* \rightarrow \Lambda^* \), \( I : \Sigma^* \rightarrow \mathbb{N}_0 \), and \( O : \Lambda^* \rightarrow \mathbb{N}_0 \) be defined by (weighted) finite state automata. Given \( \eta, \gamma \in \mathbb{N} \), the transducer \( f \) is \((\gamma, \eta)\)-IOS with respect to \((I, O)\) iff every reachable state \((q, m)\) of \( A \times A_M \) satisfies \( H^O(q) \leq m \).

- This result provides a different strategy for the verification problem: verify that the set \( S = \{(q, m) \in Q \times M | H^O(q) \leq m \} \) is invariant;
Robustness
Solving the synthesis problem

Theorem

Let $f : \Sigma^* \rightarrow \Lambda^*$, $I : \Sigma^* \rightarrow \mathbb{N}_0$, and $O : \Lambda^* \rightarrow \mathbb{N}_0$ be defined by (weighted) finite state automata. Given $\eta, \gamma \in \mathbb{N}$, the transducer $f$ is \((\gamma, \eta)\)-IOS with respect to \((I, O)\) iff every reachable state \((q, m)\) of $A \times A_M$ satisfies $H^O(q) \leq m$.

- This result provides a different strategy for the verification problem: verify that the set $S = \{(q, m) \in Q \times M | H^O(q) \leq m\}$ is invariant;
- It also provides a solution to the synthesis problem: synthesize a controller to render the set $S$ invariant;
Robustness
Solving the synthesis problem

Theorem

Let \( f : \Sigma^* \rightarrow \Lambda^* \), \( I : \Sigma^* \rightarrow \mathbb{N}_0 \), and \( O : \Lambda^* \rightarrow \mathbb{N}_0 \) be defined by (weighted) finite state automata. Given \( \eta, \gamma \in \mathbb{N} \), the transducer \( f \) is \((\gamma, \eta)\)-IOS with respect to \((I, O)\) iff every reachable state \((q, m)\) of \( A \times A_M \) satisfies \( H^O(q) \leq m \).

- This result provides a different strategy for the verification problem: verify that the set \( S = \{(q, m) \in Q \times M | H^O(q) \leq m \} \) is invariant;
- It also provides a solution to the synthesis problem: synthesize a controller to render the set \( S \) invariant;
- Since safety games can be solved in linear time, the complexity of synthesizing a controller enforcing \((\gamma, \eta)\)-IOS is linear in the size of \( A \times A_M \), i.e., it takes \( O(|Q| \cdot |\gamma w| \cdot |\Sigma_c|) \) time.
Several issues remain open:

- the characterization of all the $(\gamma, \eta)$ pairs for which a transducer is $(\gamma, \eta)$-IOS;
Robustness

Several issues remain open:

- the characterization of all the $(\gamma, \eta)$ pairs for which a transducer is $(\gamma, \eta)$-IOS;
- How to solve the IOS synthesis problem: existence and characterization of all the $(\gamma, \eta)$ pairs for which there exists a controller rendering a given transducer $(\gamma, \eta)$-IOS;

How to make these ideas practical so that they become more useful. In particular, how to define metrics and costs in concrete problems?

The ultimate objective is to understand robustness for cyber-physical systems.

Paulo Tabuada (CyPhyLab - UCLA)
Several issues remain open:

- the characterization of all the \((\gamma, \eta)\) pairs for which a transducer is \((\gamma, \eta)\)-IOS;
- How to solve the IOS synthesis problem: existence and characterization of all the \((\gamma, \eta)\) pairs for which there exists a controller rendering a given transducer \((\gamma, \eta)\)-IOS;
- How to make these ideas practical so that they become more useful. In particular, how to define metrics and costs in concrete problems?

The ultimate objective is to understand robustness for cyber-physical systems.
Robustness

Several issues remain open:

- the characterization of all the \((\gamma, \eta)\) pairs for which a transducer is \((\gamma, \eta)\)-IOS;
- How to solve the IOS synthesis problem: existence and characterization of all the \((\gamma, \eta)\) pairs for which there exists a controller rendering a given transducer \((\gamma, \eta)\)-IOS;
- How to make these ideas practical so that they become more useful. In particular, how to define metrics and costs in concrete problems?
- The ultimate objective is to understand robustness for cyber-physical systems.
Robustness

Relevant recent references:

- *Robust Discrete Synthesis Against Unspecified Disturbances*
  Rupak Majumdar, Elaine Render, and Paulo Tabuada
  14th International Conference on Hybrid Systems: Computation and Control 2011.

- *A theory of \(\omega\)-regular robust software synthesis*
  Rupak Majumdar, Elaine Render, and Paulo Tabuada

- *Input-Output Robustness for Discrete Systems*
  Paulo Tabuada, Ayca Balkan, Sina Caliskan, Yasser Shoukry, and Rupak Majumdar
  International Conference on Embedded Software 2012.

For preprints and other information:

tabuada@ee.ucla.edu
http://www.cyphylab.ee.ucla.edu