Control Hierarchies and Tropical Algebras

Jörg Raisch\textsuperscript{1,2}, Xavier David-Henriët\textsuperscript{1,2,3}, Tom Brunsch\textsuperscript{1,3}, Laurent Hardouin\textsuperscript{3}

\textsuperscript{1}Fachgebiet Regelungssysteme
Fakultät Elektrotechnik und Informatik, TU Berlin

\textsuperscript{2}Fachgruppe System- und Regelungstheorie
Max-Planck-Institut für Dynamik komplexer technischer Systeme

\textsuperscript{3}Laboratoire d’Ingénierie des Systèmes Automatisés
Université d’Angers
Outline

1 Motivation
2 A Behavioural View on Control Hierarchies
3 A Specific Scenario
4 A Few Essentials of Dioid (Tropical) Algebras
5 Specific Scenario Revisited
Outline

1. Motivation
2. A Behavioural View on Control Hierarchies
3. A Specific Scenario
4. A Few Essentials of Dioid (Tropical) Algebras
5. Specific Scenario Revisited
Motivation and Aims

Motivation

- Want to address large-scale/complex control problems
- Too many degrees of freedom for monolithic controller design
- Need to impose structure to reduce degrees of freedom
- Hierarchical control architecture particularly intuitive
- Heuristically designed hierarchical control ubiquitous in industry

Aims

- Want a formal framework that guarantees “proper interaction” of control layers to minimize trial and error during design
- Hierarchical structures need not be “rigid”; may be embedded into consensus-type distributed systems, with top-level functionality temporarily assigned to a node
Motivation

- Want to address large-scale/complex control problems
- Too many degrees of freedom for monolithic controller design
- Need to impose structure to reduce degrees of freedom
- Hierarchical control architecture particularly intuitive
- Heuristically designed hierarchical control ubiquitous in industry

Aims

- Want a formal framework that guarantees “proper interaction” of control layers to minimize trial and error during design
- Hierarchical structures need not be “rigid”; may be embedded into consensus-type distributed systems, with top-level functionality temporarily assigned to a node
Outline

1. Motivation

2. A Behavioural View on Control Hierarchies

3. A Specific Scenario

4. A Few Essentials of Dioid (Tropical) Algebras

5. Specific Scenario Revisited
Abstraction and Refinement

- Have been investigated in different scenarios
- Behavioural point of view allows conceptionally (and notationally) simple explanation of main ingredients

Dynamical system with input/output structure:

\[ \Sigma = (T, U \times Y, \mathcal{B} \subseteq (U \times Y)^T) \]

Abstractions and refinements:

- \( \Sigma_a = (T, U \times Y, \mathcal{B}_a) \) is an abstraction of \( \Sigma \) if \( \mathcal{B} \subseteq \mathcal{B}_a \)
- \( \Sigma_r = (T, U \times Y, \mathcal{B}_r) \) is a refinement of \( \Sigma \) if \( \mathcal{B}_r \subseteq \mathcal{B} \)

Interpretation: abstraction (refinement) corresponds to adding (removing) uncertainty
Abstraction and Refinement

- Have been investigated in different scenarios
- Behavioural point of view allows conceptionally (and notationally) simple explanation of main ingredients

Dynamical system with input/output structure:

\[ u(t) \in U \quad t \in T \quad y(t) \in Y \]

\[ \Sigma = \left( T, U \times Y, B \subseteq (U \times Y)^T \right) \]

Abstractions and refinements:

- \[ \Sigma_a = (T, U \times Y, B_a) \] is an abstraction of \( \Sigma \) if \( B \subseteq B_a \)
- \[ \Sigma_r = (T, U \times Y, B_r) \] is a refinement of \( \Sigma \) if \( B_r \subseteq B \)

Interpretation: abstraction (refinement) corresponds to adding (removing) uncertainty
Abstraction and Refinement

- Have been investigated in different scenarios
- Behavioural point of view allows conceptionally (and notationally) simple explanation of main ingredients

**Dynamical system with input/output structure:**

\[
\begin{align*}
\Sigma = \left( T, U \times Y, \mathcal{B} \subseteq (U \times Y)^T \right)
\end{align*}
\]

**Abstractions and refinements:**

- \( \Sigma_a = (T, U \times Y, \mathcal{B}_a) \) is an **abstraction** of \( \Sigma \) if \( \mathcal{B} \subseteq \mathcal{B}_a \)
- \( \Sigma_r = (T, U \times Y, \mathcal{B}_r) \) is a **refinement** of \( \Sigma \) if \( \mathcal{B}_r \subseteq \mathcal{B} \)

Interpretation: abstraction (refinement) corresponds to adding (removing) uncertainty
Abstraction and Refinement

- Have been investigated in different scenarios
- Behavioural point of view allows conceptionally (and notationally) simple explanation of main ingredients

Dynamical system with input/output structure:

\[ \Sigma = \left( T, U \times Y, \mathcal{B} \subseteq (U \times Y)^T \right) \]

Abstractions and refinements:

- \( \Sigma_a = (T, U \times Y, \mathcal{B}_a) \) is an abstraction of \( \Sigma \) if \( \mathcal{B} \subseteq \mathcal{B}_a \)
- \( \Sigma_r = (T, U \times Y, \mathcal{B}_r) \) is a refinement of \( \Sigma \) if \( \mathcal{B}_r \subseteq \mathcal{B} \)

Interpretation: abstraction (refinement) corresponds to adding (removing) uncertainty
Generic Two-Level Control Structure

\[ \mathcal{B}^L_{sup} : \text{high-level supervisor} \]

\[ u^H \]

\[ y^H \]

\[ \mathcal{B}_{im} : \text{aggregation & low-level control} \]

\[ u^L \]

\[ y^L \]

\[ \mathcal{B}^L_p : \text{low-level plant model} \]

... can be extended to arbitrary number of control layers ...

- Low-level signal space: \( W_L = U_L \times Y_L \).
- Low-level process model: \( \mathcal{B}^L_p \) ... behaviour on \( W_L \).
- Inclusion-type specification: \( \mathcal{B}^L_{spec} \) ... defined on \( W_L \).
- High-level signal space: \( W_H = U_H \times Y_H \).
- High-level supervisor: \( \mathcal{B}^H_{sup} \) ... behaviour on \( W_H \).
- Low-level control: \( \mathcal{B}_{im} \) ... behaviour on \( W_H \times W_L \).
Motivation
A Behavioural View on Control Hierarchies
A Specific Scenario
Dioid Algebras
Specific Scenario Revisited

Generic Two-Level Control Structure

... can be extended to arbitrary number of control layers...

- Low-level signal space: $W_L = U_L \times Y_L$.
- Low-level process model: $B^L_p$ ... behaviour on $W_L$.
- Inclusion-type specification: $B^L_{\text{spec}}$ ... defined on $W_L$.
- High-level signal space: $W_H = U_H \times Y_H$.
- High-level supervisor: $B^H_{\text{sup}}$ ... behaviour on $W_H$.
- Low-level control: $B_{\text{im}}$ ... behaviour on $W_H \times W_L$. 
Design Procedure

Define high-level signal space (assumed given in this talk).

**Low-level control:**
- Define (inclusion-type) specs $\mathcal{B}_{\text{spec}}^{\text{HL}}$ for lower control layer – intended relation between high-level and low-level signals.
- Design low-level control $\mathcal{B}_{\text{im}}$ enforcing specs $\mathcal{B}_{\text{spec}}^{\text{HL}}$.

**High-level control:**
- Synthesise $\mathcal{B}_{\text{sup}}^{\text{H}}$ for $\mathcal{B}_{\text{p}}^{\text{H}} = \mathcal{B}_{\text{im}}[\mathcal{B}_{\text{p}}^{\text{L}}]$. Can be done abstraction-based!
  - Use high-level proj. $P^{\text{H}}(\mathcal{B}_{\text{spec}}^{\text{HL}})$ of $\mathcal{B}_{\text{spec}}^{\text{HL}}$ as abstraction of $\mathcal{B}_{\text{p}}^{\text{H}}$.
  - Define high-level spec. $\mathcal{B}_{\text{spec}}^{\text{H}}$ such that $\mathcal{B}_{\text{spec}}^{\text{HL}}[\mathcal{B}_{\text{spec}}^{\text{H}}] \subseteq \mathcal{B}_{\text{spec}}^{\text{L}}$.
  - Find high-level control $\mathcal{B}_{\text{sup}}^{\text{H}}$ such that $P^{\text{H}}(\mathcal{B}_{\text{spec}}^{\text{HL}}) \cap \mathcal{B}_{\text{sup}}^{\text{H}} \subseteq \mathcal{B}_{\text{spec}}^{\text{H}}$.

$$\mathcal{B}_{\text{p}} \cap \mathcal{B}_{\text{im}}[\mathcal{B}_{\text{sup}}^{\text{HL}}] \subseteq \mathcal{B}_{\text{sup}}^{\text{L}}$$
Design Procedure

Define high-level signal space (assumed given in this talk).

**Low-level control:**
- Define (inclusion-type) specs $\mathcal{B}_{\text{spec}}^{\text{HL}}$ for lower control layer – *intended* relation between high-level and low-level signals.
- Design low-level control $\mathcal{B}_{\text{im}}$ enforcing specs $\mathcal{B}_{\text{spec}}^{\text{HL}}$.

**High-level control:**
Synthesise $\mathcal{B}_{\text{sup}}^H$ for $\mathcal{B}_{\text{p}}^H = \mathcal{B}_{\text{im}}[\mathcal{B}_{\text{p}}^L]$. Can be done abstraction-based!
- Use high-level proj. $P^H(\mathcal{B}_{\text{spec}}^{\text{HL}})$ of $\mathcal{B}_{\text{spec}}^{\text{HL}}$ as abstraction of $\mathcal{B}_{\text{p}}^H$.
- Define high-level spec. $\mathcal{B}_{\text{spec}}^H$ such that $\mathcal{B}_{\text{spec}}^{\text{HL}}[\mathcal{B}_{\text{spec}}^H] \subseteq \mathcal{B}_{\text{spec}}^L$.
- Find high-level control $\mathcal{B}_{\text{sup}}^H$ such that $P^H(\mathcal{B}_{\text{spec}}^{\text{HL}}) \cap \mathcal{B}_{\text{sup}}^H \subseteq \mathcal{B}_{\text{spec}}^H$.

$$\mathcal{B}_{\text{p}}^L \cap \mathcal{B}_{\text{im}}[\mathcal{B}_{\text{sup}}^H] \subseteq \mathcal{B}_{\text{spec}}^L$$
Design Procedure

Define high-level signal space (assumed given in this talk).

**Low-level control:**
- Define (inclusion-type) specs $\mathcal{B}^{\text{HL spec}}$ for lower control layer – *intended* relation between high-level and low-level signals.
- Design low-level control $\mathcal{B}_{\text{lm}}$ enforcing specs $\mathcal{B}^{\text{HL spec}}$.

**High-level control:**
Synthesise $\mathcal{B}^{\text{sup}}$ for $\mathcal{B}^{\text{H}} = \mathcal{B}_{\text{lm}}[\mathcal{B}^{\text{L}}]$. Can be done abstraction-based!
- Use high-level proj. $\mathcal{P}^{\text{H}}(\mathcal{B}^{\text{HL spec}})$ of $\mathcal{B}^{\text{HL spec}}$ as abstraction of $\mathcal{B}^{\text{H}}$.
- Define high-level spec. $\mathcal{B}^{\text{H spec}}$ such that $\mathcal{B}^{\text{HL spec}}[\mathcal{B}^{\text{H spec}}] \subseteq \mathcal{B}^{\text{L spec}}$.
- Find high-level control $\mathcal{B}^{\text{sup}}$ such that $\mathcal{P}^{\text{H}}(\mathcal{B}^{\text{HL spec}}) \cap \mathcal{B}^{\text{H sup}} \subseteq \mathcal{B}^{\text{H spec}}$.

\[ \mathcal{B}_p \cap \mathcal{B}_{\text{lm}}[\mathcal{B}^{\text{sup}}] \subseteq \mathcal{B}^{\text{spec}} \]
Where Can Things Go Wrong?

Low-level specification $\mathcal{B}_{\text{spec}}^{\text{HL}}$ too demanding:

- I.e., we cannot find appropriate low-level control.
- Need to relax low-level specifications and replace $\mathcal{B}_{\text{spec}}^{\text{HL}}$ by an abstraction $\mathcal{B}_{\text{spec},a}^{\text{HL}}$ such that $\mathcal{B}_{\text{spec}}^{\text{HL}} \subseteq \mathcal{B}_{\text{spec},a}^{\text{HL}}$.

Illustration: robot moving in a restricted area:

\[
\begin{align*}
\dot{x}_1(t) &= v(t) \cos \theta(t) \\
\dot{x}_2(t) &= v(t) \sin \theta(t) \\
\dot{\theta}(t) &= u_1(t) \\
\dot{v}(t) &= u_2(t)
\end{align*}
\]

$u^l = (u_1, u_2)$ low-level inputs

$y^l = (x_1, x_2)$ low-level outputs

$u^{\text{HI}} \in \{\text{go up, } \ldots \}$ high-level input

$y^{\text{HI}} = \text{quant}(x_1, x_2)$ high-level output.
Where Can Things Go Wrong?

Low-level specification $\mathcal{B}_{\text{spec}}^{\text{HL}}$ too demanding:

- I.e., we cannot find appropriate low-level control.
- Need to relax low-level specifications and replace $\mathcal{B}_{\text{spec}}^{\text{HL}}$ by an abstraction $\mathcal{B}_{\text{spec},a}^{\text{HL}}$ such that $\mathcal{B}_{\text{spec}}^{\text{HL}} \subseteq \mathcal{B}_{\text{spec},a}^{\text{HL}}$.

Illustration: robot moving in a restricted area:

\[
\begin{align*}
\dot{x}_1(t) &= v(t) \cos \theta(t) \\
\dot{x}_2(t) &= v(t) \sin \theta(t) \\
\dot{\theta}(t) &= u_1(t) \\
\dot{v}(t) &= u_2(t)
\end{align*}
\]

$u^L = (u_1, u_2)$ low-level inputs
$y^L = (x_1, x_2)$ low-level outputs
$u^H \in \{\text{go up, \ldots}\}$ high-level input
$y^H = \text{quant}(x_1, x_2)$ high-level output.
Where Can Things Go Wrong?

**Low-level specification $\mathcal{B}_{\text{spec}}^{\text{HL}}$ too demanding:**

- I.e., we cannot find appropriate low-level control.
- Need to relax low-level specifications and replace $\mathcal{B}_{\text{spec}}^{\text{HL}}$ by an abstraction $\mathcal{B}_{\text{spec},a}^{\text{HL}}$ such that $\mathcal{B}_{\text{spec}}^{\text{HL}} \subseteq \mathcal{B}_{\text{spec},a}^{\text{HL}}$.

Illustration: robot moving in a restricted area:

\[
\begin{align*}
\dot{x}_1(t) &= v(t) \cos \theta(t) \\
\dot{x}_2(t) &= v(t) \sin \theta(t) \\
\dot{\theta}(t) &= u_1(t) \\
\dot{v}(t) &= u_2(t)
\end{align*}
\]

- $u^L = (u_1, u_2)$ low-level inputs
- $y^L = (x_1, x_2)$ low-level outputs
- $u^H \in \{\text{go up, ...}\}$ high-level input
- $y^H = \text{quant}(x_1, x_2)$ high-level output.

Ex.: “go right” $\Rightarrow$ is too demanding.
What Else Can Go Wrong?

Low-level specification $\mathcal{B}_{\text{spec}}^{\text{HL}}$ too coarse:

- $P^H(\mathcal{B}_{\text{spec}}^{\text{HL}})$ serves as abstraction of plant under low-level control.
- We cannot find appropriate high-level control.
- Need to refine low-level specifications by $\mathcal{B}_{\text{spec},r}^{\text{HL}} \subseteq \mathcal{B}_{\text{spec}}^{\text{HL}}$.

Example:

Recap:

- choice of low-level specs $\mathcal{B}_{\text{spec}}^{\text{HL}}$ depends on engineering intuition
- often involves trade-off between control layers
- key advantage: solution of low- & high-level control problems will provide a solution for the overall problem (guaranteed!)
What Else Can Go Wrong?

Low-level specification $\mathcal{B}_{\text{spec}}^{\text{HL}}$ too coarse:

- $P^H(\mathcal{B}_{\text{spec}}^{\text{HL}})$ serves as abstraction of plant under low-level control.
- We cannot find appropriate high-level control.
- Need to refine low-level specifications by $\mathcal{B}_{\text{spec},r}^{\text{HL}} \subseteq \mathcal{B}_{\text{spec}}^{\text{HL}}$.

Example: “go right” $\sim$

Recap:

- choice of low-level specs $\mathcal{B}_{\text{spec}}^{\text{HL}}$ depends on engineering intuition
- often involves trade-off between control layers
- key advantage: solution of low- & high-level control problems will provide a solution for the overall problem (guaranteed!)
Motivation
A Behavioural View on Control Hierarchies
A Specific Scenario
Dioid Algebras
Specific Scenario Revisited

What Else Can Go Wrong?

Low-level specification $\mathcal{B}_{\text{spec}}^{\text{HL}}$ too coarse:

- $\mathcal{P}^{\text{H}}(\mathcal{B}_{\text{spec}}^{\text{HL}})$ serves as abstraction of plant under low-level control.
- We cannot find appropriate high-level control.
- Need to refine low-level specifications by $\mathcal{B}_{\text{spec}, r}^{\text{HL}} \subseteq \mathcal{B}_{\text{spec}}^{\text{HL}}$.

Example: "go right" $\rightsquigarrow$

Recap:

- choice of low-level specs $\mathcal{B}_{\text{spec}}^{\text{HL}}$ depends on engineering intuition
- often involves trade-off between control layers
- key advantage: solution of low- & high-level control problems will provide a solution for the overall problem (guaranteed!)
Outline

1. Motivation
2. A Behavioural View on Control Hierarchies
3. A Specific Scenario
4. A Few Essentials of Dioid (Tropical) Algebras
5. Specific Scenario Revisited
Specific Scenario

- top layer decides on timing (not ordering!) of discrete events
- synthesis based on TEG abstraction of plant + low-level control
- TEG (Timed Event Graph) . . . specific timed Petri net

Example:

\[ x_7(k) = \max\{x_4(k) + 1, x_2(k) + 6, x_2(k + 1), x_8(k - 1)\} \]

Time relations become linear in certain dioid (tropical) algebras . . .
Specific Scenario

- top layer decides on timing (not ordering!) of discrete events
- synthesis based on TEG abstraction of plant + low-level control
- TEG (Timed Event Graph) ... specific timed Petri net

Example:

\[ x_7(k) = \max \{ x_4(k) + 1, x_2(k) + 6, x_2(k + 1), x_8(k - 1) \} \]

Time relations become linear in certain dioid (tropical) algebras ...
Specific Scenario

- top layer decides on timing (not ordering!) of discrete events
- synthesis based on TEG abstraction of plant + low-level control
- TEG (Timed Event Graph) ... specific timed Petri net

Example:

- want to compute earliest times of $k$-th occurrences of events
- doable, but time relations (non-benevolently) non-linear

$x_7(k) = \max\{x_4(k) + 1, x_2(k) + 6, x_2(k + 1), x_8(k - 1)\}$

time relations become linear in certain dioid (tropical) algebras ...
Specific Scenario

- top layer decides on timing (not ordering!) of discrete events
- synthesis based on TEG abstraction of plant + low-level control
- TEG (Timed Event Graph) ... specific timed Petri net

Example:

\[ x_7(k) = \max\{ x_4(k) + 1, x_2(k) + 6, x_2(k + 1), x_8(k - 1) \} \]

time relations become linear in certain dioid (tropical) algebras ...
Outline

1. Motivation

2. A Behavioural View on Control Hierarchies

3. A Specific Scenario

4. A Few Essentials of Dioid (Tropical) Algebras

5. Specific Scenario Revisited
A dioid is an algebraic structure with two binary operations ⊕ ("addition") and ⊗ ("multiplication") defined on a set \( D \), such that

- ⊕ is associative, commutative & idempotent \( (a ⊕ a = a \ \forall a ∈ D) \)
- ⊗ is associative and is distributive w.r.t. ⊕
- zero element \( ε \), unit element \( e \)
- \( ε \) is absorbing for ⊗, i.e., \( ε ⊗ a = a ⊗ e = ε \ \forall a ∈ D \)

**Remarks**

- a dioid is complete if it is closed for infinite sums and ⊗ distributes over infinite sums
- dioids are equipped with a natural order: \( a ⊕ b = a ⇔ a ≥ b \)
- addition and multiplication can be easily extended to matrices
A dioid is an algebraic structure with two binary operations $\oplus$ ("addition") and $\otimes$ ("multiplication") defined on a set $\mathcal{D}$, such that

- $\oplus$ is associative, commutative & idempotent ($a \oplus a = a \ \forall a \in \mathcal{D}$)
- $\otimes$ is associative and is distributive w.r.t. $\oplus$
- zero element $\varepsilon$, unit element $e$
- $\varepsilon$ is absorbing for $\otimes$, i.e., $\varepsilon \otimes a = a \otimes \varepsilon = \varepsilon \ \forall a \in \mathcal{D}$

**Remarks**

- a dioid is complete if it is closed for infinite sums and $\otimes$ distributes over infinite sums
- dioids are equipped with a natural order: $a \oplus b = a \iff a \succeq b$
- addition and multiplication can be easily extended to matrices
Example: The Max-Plus Algebra

Defined on $\bar{\mathbb{Z}} = \mathbb{Z} \cup \{-\infty\} \cup \{+\infty\}$ resp. $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$:

- addition: $a \oplus b := \max(a, b)$, zero element: $\varepsilon := -\infty$
- multiplication: $a \otimes b := a + b$, unit element: $e := 0$

Time relations for TEGs described by linear implicit difference eqns.

For our example

\[ x_7(k) = 1 \otimes x_4(k) \oplus 6 \otimes x_2(k) \oplus x_2(k + 1) \oplus x_8(k - 1) \]
Example: The Max-Plus Algebra

Defined on \( \mathbb{Z} = \mathbb{Z} \cup \{ -\infty \} \cup \{ +\infty \} \text{ resp. } \mathbb{R} = \mathbb{R} \cup \{ -\infty \} \cup \{ +\infty \} : \)

- addition: \( a \oplus b := \max(a, b) \), zero element: \( \varepsilon := -\infty \)
- multiplication: \( a \otimes b := a + b \), unit element: \( e := 0 \)

Time relations for TEGs described by linear implicit difference eqns.

For our example

\[
x_7(k) = 1 \otimes x_4(k) \oplus 6 \otimes x_2(k) \oplus x_2(k + 1) \oplus x_8(k - 1)
\]
Example: The Max-Plus Algebra

Defined on $\mathbb{Z} = \mathbb{Z} \cup \{-\infty\} \cup \{+\infty\}$ resp. $\mathbb{R} = \mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$:

- addition: $a \oplus b := \max(a, b)$, zero element: $\varepsilon := -\infty$
- multiplication: $a \otimes b := a + b$, unit element: $e := 0$

Time relations for TEGs described by linear implicit difference eqns.

For our example

$$x_7(k) = 1 \otimes x_4(k) \oplus 6 \otimes x_2(k) \oplus x_2(k + 1) \oplus x_8(k - 1)$$
The Dioid $\mathcal{M}_{in}^{ax}[\gamma, \delta]$

- $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ ... a quotient dioid in the set of 2-dim. formal power series (in $\gamma, \delta$), with Boolean coefficients and integer exponents

- interpretation of monomial $\gamma^k \delta^t$:
  - $k$th occurrence of event is at time $t$ at the earliest
  - equivalently: at time $t$, event has occurred at most $k$ times

$\leadsto$ have to consider “south-east cones” (instead of points) in $\mathbb{Z}^2$

Example: $s = \gamma^1 \delta^1 \oplus \gamma^3 \delta^2 \oplus \gamma^4 \delta^5$
The Dioid $\mathcal{M}^\text{ax}_{\text{in}} [\gamma, \delta]$ 

- $\mathcal{M}^\text{ax}_{\text{in}} [\gamma, \delta]$ ... a quotient dioid in the set of 2-dim. formal power series (in $\gamma, \delta$), with Boolean coefficients and integer exponents

- interpretation of monomial $\gamma^k\delta^t$:
  - $k$th occurrence of event is at time $t$ at the earliest
  - equivalently: at time $t$, event has occurred at most $k$ times

$\leadsto$ have to consider “south-east cones” (instead of points) in $\mathbb{Z}^2$

Example: $s = \gamma^1\delta^1 \oplus \gamma^3\delta^2 \oplus \gamma^4\delta^5$

Properties:

- $\gamma^k\delta^t \oplus \gamma^l\delta^t = \gamma^\min(k,l)\delta^t$
- $\gamma^k\delta^t \oplus \gamma^k\delta^\tau = \gamma^k\delta^\max(t,\tau)$
- $\gamma^k\delta^t \otimes \gamma^l\delta^\tau = \gamma^{(k+l)}\delta^{(t+\tau)}$
- Zero element: $\varepsilon = \gamma^{+\infty}\delta^{-\infty}$
- Unit element: $\varepsilon = \gamma^0\delta^0$
- interpretation of partial order: inclusion in $\mathbb{Z}^2$
The Dioid $\mathbb{M}^\text{ax}_{\text{in}}[\gamma, \delta]$

- $\mathbb{M}^\text{ax}_{\text{in}}[\gamma, \delta]$ ... a quotient dioid in the set of 2-dim. formal power series (in $\gamma$, $\delta$), with Boolean coefficients and integer exponents

- interpretation of monomial $\gamma^k \delta^t$:
  - $k$th occurrence of event is at time $t$ at the earliest
  - equivalently: at time $t$, event has occurred at most $k$ times

\[\rightsquigarrow\text{have to consider “south-east cones” (instead of points) in } \mathbb{Z}^2\]

Example: $s = \gamma^1 \delta^1 \oplus \gamma^3 \delta^2 \oplus \gamma^4 \delta^5$

Properties:

- $\gamma^k \delta^t \oplus \gamma^l \delta^t = \gamma^\text{min}(k,l) \delta^t$
- $\gamma^k \delta^t \oplus \gamma^k \delta^\tau = \gamma^k \delta^\text{max}(t,\tau)$
- $\gamma^k \delta^t \otimes \gamma^l \delta^\tau = \gamma^{(k+l)} \delta^{(t+\tau)}$
- Zero element: $\varepsilon = \gamma^{+\infty} \delta^{-\infty}$
- Unit element: $e = \gamma^0 \delta^0$
- interpretation of partial order: inclusion in $\mathbb{Z}^2$
The Dioid $\mathcal{M}_{in}^{ax} [\gamma, \delta]$

- $\mathcal{M}_{in}^{ax} [\gamma, \delta]$ ... a quotient dioid in the set of 2-dim. formal power series (in $\gamma$, $\delta$), with Boolean coefficients and integer exponents
- interpretation of monomial $\gamma^k \delta^t$:
  - $k$th occurrence of event is at time $t$ at the earliest
  - equivalently: at time $t$, event has occurred at most $k$ times

∴ have to consider “south-east cones” (instead of points) in $\mathbb{Z}^2$

Example: $s = \gamma^1 \delta^1 \oplus \gamma^3 \delta^2 \oplus \gamma^4 \delta^5$

Properties:

- $\gamma^k \delta^t \oplus \gamma^l \delta^t = \gamma^{\min(k,l)} \delta^t$
- $\gamma^k \delta^t \oplus \gamma^k \delta^\tau = \gamma^k \delta^{\max(t,\tau)}$
- $\gamma^k \delta^t \otimes \gamma^l \delta^\tau = \gamma^{(k+l)} \delta^{(t+\tau)}$
- Zero element: $\varepsilon = \gamma^{+\infty} \delta^{-\infty}$
- Unit element: $e = \gamma^0 \delta^0$
- interpretation of partial order: inclusion in $\mathbb{Z}^2$
The Dioid $\mathcal{M}_{in}^{ax} [\gamma, \delta]$ ctd.

Time relations for TEGs become linear algebraic eqns. in $\mathcal{M}_{in}^{ax} [\gamma, \delta]$

For our example

$$x_7 = \delta^1 \gamma^0 x_4 \oplus (\delta^6 \gamma^0 \oplus \delta^0 \gamma^{-1}) x_2 \oplus \delta^0 \gamma^1 x_8$$

In general, with input & output trans. (triggered resp. seen externally):

$$x = Ax \oplus Bu$$

$$y = Cx$$
The Dioid $\mathcal{M}_{in}^{ax} [\gamma, \delta]$ ctd.

Time relations for TEGs become linear algebraic eqns. in $\mathcal{M}_{in}^{ax} [\gamma, \delta]$

For our example

$$x_7 = \delta^1 \gamma^0 x_4 \oplus (\delta^6 \gamma^0 \oplus \delta^0 \gamma^{-1}) x_2 \oplus \delta^0 \gamma^1 x_8$$

In general, with input & output trans. (triggered resp. seen externally):

$$x = Ax \oplus Bu$$

$$y = Cx$$
Control in the Dioid $\mathcal{M}_{\text{in}}^{\text{ax}} [\gamma, \delta]$

**Plant:**
- state model $x = Ax \oplus Bu$, $y = Cx$
- i/o rel. $y = CA^* Bu$, with $A^* := \bigoplus_{i \in \mathbb{N}_0} A^i$ ... Kleene star operator

**Output feedback:**
- $u = Ky \oplus v$
- $y \triangleleft y = CA^* BKy \oplus CA^* Bv$
- $y = (CA^* BK)^* CA^* B v$

**Aim: just-in-time policy**
- find greatest $K$ s.t. $H_{\text{ref}} \succeq H_{\text{cl}}$, with
- $H_{\text{ref}}$ a given reference model
- “greatest” and “$\succeq$” in the sense of natural order in $\mathcal{M}_{\text{in}}^{\text{ax}} [\gamma, \delta]$

**Solution:**
- desired feedback $K$ can be obtained using “residuation theory”:
  - $K_{\text{opt}} = (CA^* B) \downarrow H_{\text{ref}} \uparrow (CA^* B)$
Control in the Dioid $\mathcal{M}_{in}^{ax}[\gamma, \delta]$ 

**Plant:**

- state model $x = Ax \oplus Bu, \ y = Cx$
- i/o rel. $y = CA^* Bu$, with $A^* := \bigoplus_{i \in \mathbb{N}_0} A^i$ ... Kleene star operator

**Output feedback:**

$u = Ky \oplus v$

$\Rightarrow y = CA^* BK y \oplus CA^* B v$

$y = (CA^* BK)^* CA^* B v$

**Aim: just-in-time policy**

find greatest $K$ s.t. $H_{ref} \succeq H_{cl}$, with

- $H_{ref}$ a given reference model
- “greatest” and “$\succeq$” in the sense of natural order in $\mathcal{M}_{in}^{ax}[\gamma, \delta]$

**Solution:**

desired feedback $K$ can be obtained using “residuation theory”:

$$K_{opt} = (CA^* B) \Diamond H_{ref} \not\diamond (CA^* B)$$
Control in the Dioid $\mathcal{M}_{\text{in}}^{\text{ax}}[\gamma, \delta]$

**Plant:**
- state model $x = Ax \oplus Bu$, $y = Cx$
- i/o rel. $y = CA^* Bu$, with $A^* := \bigoplus_{i \in \mathbb{N}_0} A^i$ ... Kleene star operator

**Output feedback:**
- $u = Ky \oplus v$
- $\Rightarrow y = CA^* BKy \oplus CA^* Bv$
- $y = (CA^* BK)^* CA^* Bv$

**Aim: just-in-time policy**
- Find greatest $K$ s.t. $H_{\text{ref}} \succeq H_{\text{cl}}$, with
  - $H_{\text{ref}}$ a given reference model
  - “greatest” and “$\succeq$” in the sense of natural order in $\mathcal{M}_{\text{in}}^{\text{ax}}[\gamma, \delta]$

**Solution:**
- Desired feedback $K$ can be obtained using “residuation theory”:
  $$K_{\text{opt}} = (CA^* B) \downarrow H_{\text{ref}} \uparrow (CA^* B)$$
Control in the Dioid $\mathcal{M}_{in}^{ax} \left[ \gamma, \delta \right]$
Outline

1. Motivation
2. A Behavioural View on Control Hierarchies
3. A Specific Scenario
4. A Few Essentials of Dioid (Tropical) Algebras
5. Specific Scenario Revisited
Tradeoff Between Control Layers

- $K_{opt}$ … greatest feedback $K$ s.t.

$$ (H_{spec}K)^*H_{spec} \preceq G_{spec} $$

for a given overall spec. $G_{spec}$

- $H_{spec}$ … low-level spec., i.e., abstraction for plant under low-level control

Result:

- Given overall specification $G_{spec}$
- Given low-level specifications $H_{spec_1}, H_{spec_2}$, with $H_{spec_1} \preceq H_{spec_2}$ (and some “natural” restrictions in place)
- Compute corresponding optimal feedback control $K_{opt_1}, K_{opt_2}$
- Can show that $K_{opt_1} \succeq K_{opt_2}$ (“stricter low-level specs allow for more relaxed high-level control”)
**Motivation**

A Behavioural View on Control Hierarchies

A Specific Scenario

Dioid Algebras

Specific Scenario Revisited

**Tradeoff Between Control Layers**

- \( K_{\text{opt}} \) \( \ldots \) greatest feedback \( K \) s.t.
  
  \[
  (H_{\text{spec}}K)^* H_{\text{spec}} \preceq G_{\text{spec}}
  \]

  for a given overall spec. \( G_{\text{spec}} \)

- \( H_{\text{spec}} \) \( \ldots \) low-level spec., i.e., abstraction for plant under low-level control

---

**Result:**

- Given overall specification \( G_{\text{spec}} \)
- Given low-level specifications \( H_{\text{spec}_1}, H_{\text{spec}_2} \), with \( H_{\text{spec}_1} \preceq H_{\text{spec}_2} \) (and some “natural” restrictions in place)
- Compute corresponding optimal feedback control \( K_{\text{opt}_1}, K_{\text{opt}_2} \)
- Can show that \( K_{\text{opt}_1} \succeq K_{\text{opt}_2} \) (“stricter low-level specs allow for more relaxed high-level control”)
**Tradeoff Between Control Layers**

- $K_{\text{opt}}$ ... greatest feedback $K$ s.t.
  
  $$(H_{\text{spec}}K)^*H_{\text{spec}} \preceq G_{\text{spec}}$$

  for a given overall spec. $G_{\text{spec}}$

- $H_{\text{spec}}$ ... low-level spec., i.e., abstraction for plant under low-level control

**Result:**

- Given overall specification $G_{\text{spec}}$
- Given low-level specifications $H_{\text{spec}_1}, H_{\text{spec}_2}$, with $H_{\text{spec}_1} \preceq H_{\text{spec}_2}$ (and some “natural” restrictions in place)
- Compute corresponding optimal feedback control $K_{\text{opt}_1}, K_{\text{opt}_2}$
- Can show that $K_{\text{opt}_1} \succeq K_{\text{opt}_2}$ (“stricter low-level specs allow for more relaxed high-level control”)

\[\text{Diagram:} K_{\text{opt}} \rightarrow H_{\text{spec}} \rightarrow \text{Input/Output/Observation layers}\]
Interpreted trade-off between layers in a hierarchical control system from a behavioural point of view

Formally investigated this trade-off for a specific scenario where top layer is responsible for timing of discrete events

Resulting setup conveniently described in the dioid $\mathcal{M}^{ax}_{in} [\gamma, \delta]$

Verified that stricter low-level specs indeed allow for more relaxed high-level control
More Details


