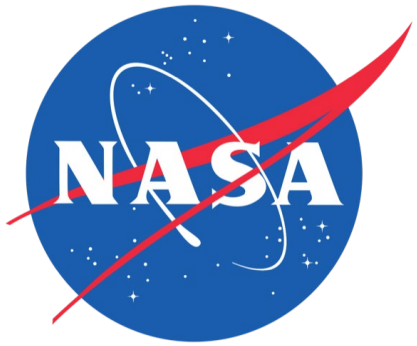
A Formal Verification Framework for Runtime Assurance

J. Tanner Slagel, Lauren M. White, Aaron Dutle,
César A. Muñoz, and Nicholas Crespo

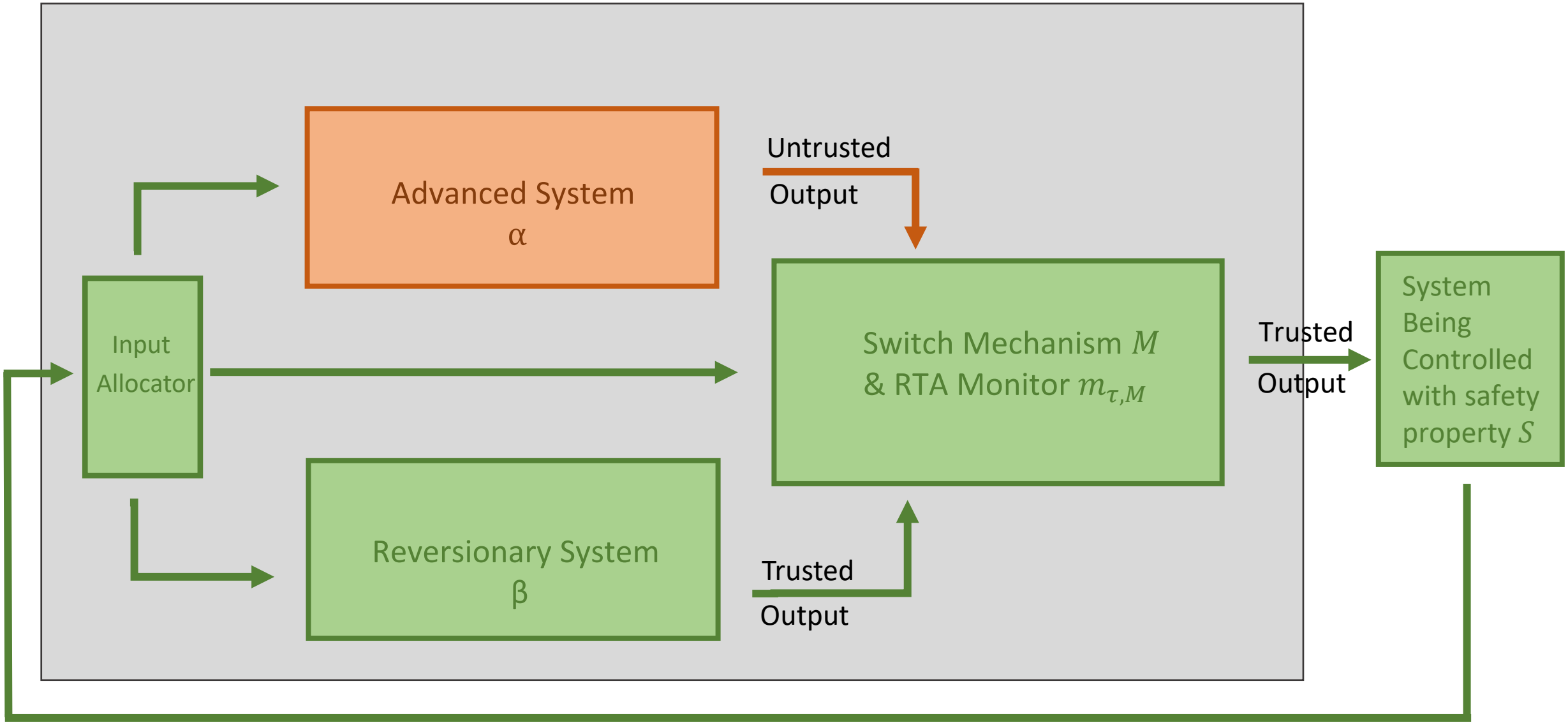
NASA Langley Research Center
Hampton, VA, USA

email: lauren.m.white@nasa.gov



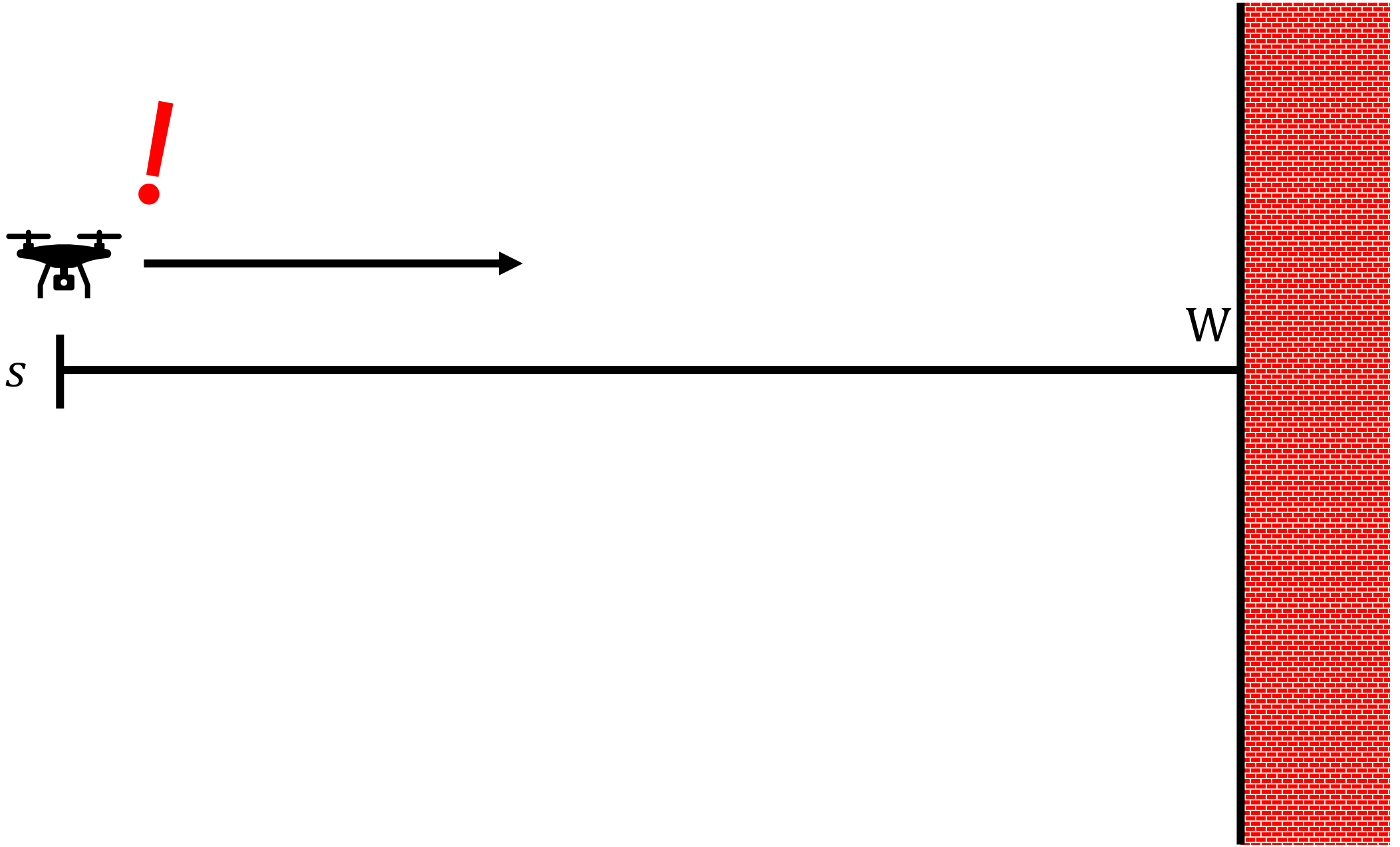
June 2024 at NASA Formal Methods (NFM) 2024

Simplex Runtime Assurance (RTA)

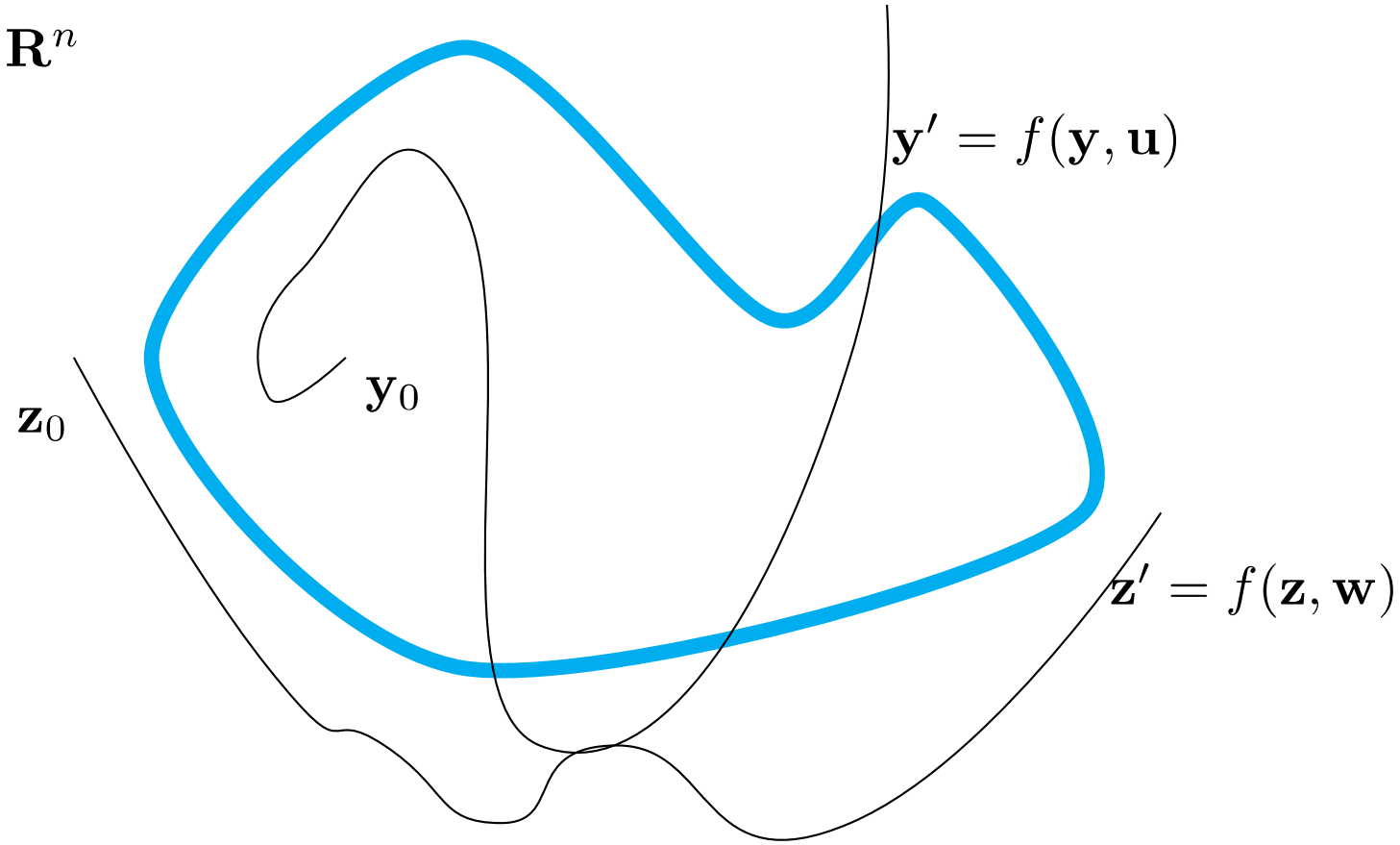


[1]: Runtime Assurance Framework Development For Highly Adaptive Flight Control Systems, SBIR Phase III Report, J.D. Schierman et. al.
https://apps.dtic.mil/dtic/tr/fulltext/u2/1010277.pdf

A UAV approaching a wall



Hybrid Programs and Differential Dynamic Logic



Hybrid Programs

Hybrid programs allow **formal specification** of hybrid systems:

- Discrete jump set:

$$(x_1 := \theta_1, \ldots, x_n := \theta_n)$$

- Differential equations:

$$\{x_1' := \theta_1, \ldots, x_n' := \theta_n \;\&\; \chi\}$$

  - $\{x_i\}_{i=1}^n$ variables
  - $\{\theta_i\}_{i=1}^n$ assignments (e.g. – functions of existing variable values)
  - χ first order formula that describes domain

Example:

$$(x := 0\,, y := c\,); \{x' = y, y' = -x \;\&\; y \geq 0\}$$

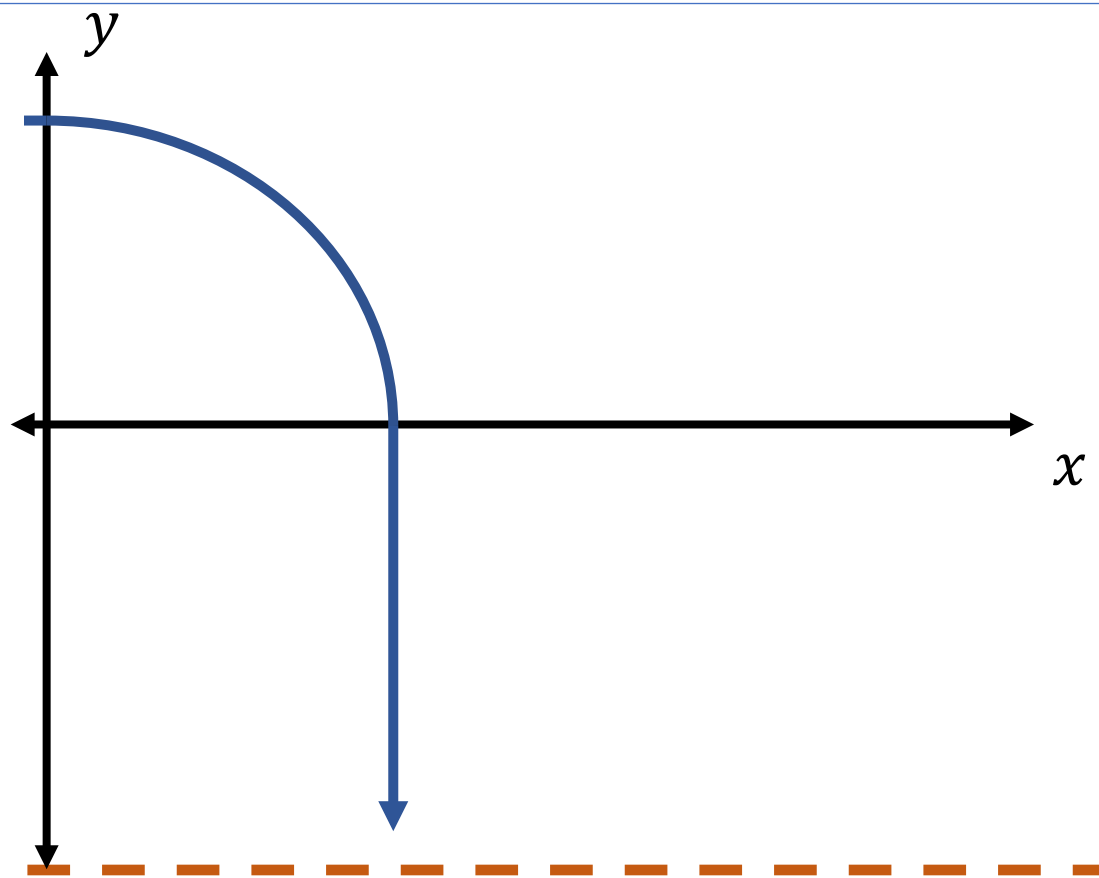
Hybrid Programs (continued)

For hybrid programs Hp_1, Hp_2, first-order formula χ:

- Choice $(Hp_1 \cup Hp_2)$
- Sequence $(Hp_1; Hp_2)$
- Repeat $(Hp_1)^*$
- Test $(?\,\chi)$

Example: $$\left\{ \begin{array}{c} \big((?\,y > 0); \{x' = y, y' = -x \;\&\; y \geq 0\}\big) \\ \cup \\ \big((?\,y \leq 0); \{y' = -c\;\}\big) \end{array} \right\}^*$$

Hybrid Programs (continued)



Example: $\left\{ \begin{array}{c} \left((?\, y > 0); \{x' = y, y' = -x \ \&\ y \geq 0\} \right) \\ \cup \\ \left((?\, y \leq 0); \{y' = -c \ \} \right) \end{array} \right\}^*$

dL: Differential Dynamic Logic

dL allows **formal reasoning** of hybrid programs:

- For hybrid program Hp and predicate P
  - All runs $[Hp]\mathrm{P}$
  - Some runs $\langle Hp\rangle\mathrm{P}$

Example: Let

$$Hp \equiv \big((?\,y > 0); \{x' = y, y' = -x \;\&\; y \geq 0\}\big)$$

$$\mathrm{P} = (x^2 + y^2 = c^2),$$

then

$$\mathrm{y = c\,, x = 0} \rightarrow [Hp]P \qquad \mathrm{y = c\,, x = 0} \rightarrow \langle Hp\rangle(\mathrm{y = 0})$$

dL: Differential Dynamic Logic – Rule Schema

Union axiom:

$$\frac{[Hp_1]P \wedge [Hp_2]P}{[Hp_1 \cup Hp_2]P}$$

Loop rule:

$$\frac{\Gamma \vdash J \qquad J \vdash [\alpha]J \qquad J \vdash P}{\Gamma \vdash [\alpha^*]P}$$

Differential invariant rule:

$$\frac{\Gamma, q(x) \vdash p(x) \qquad q(x) \vdash [x' := f(x)](p(x))'}{\Gamma \vdash [x' = f(x)\,\&\,q(x)]p(x)}$$

….and many more! [4]

[2] Differential Dynamic Logic, Logical Foundations of Cyber-Physical Systems. André Platzer. 2018 https://doi.org/10.1007/978-3-319-63588-0

[3] **dL** "Cheat Sheet," André Platzer, https://symbolaris.com/logic/dL-sheet.pdf

dL: Proof

Differential invariant rule:

$$\frac{\Gamma, q(x) \vdash p(x) \qquad q(x) \vdash [x' := f(x)](p(x))'}{\Gamma \vdash [x' = f(x)\ \&\ q(x)]p(x)}$$

$\vdash 0 = 0$ ✓

Arithmetic! ✓ $\vdash 0^2 + c^2 = c^2$ $\qquad \vdash 2\,x\,y + 2\,y\,(-x) = 0$

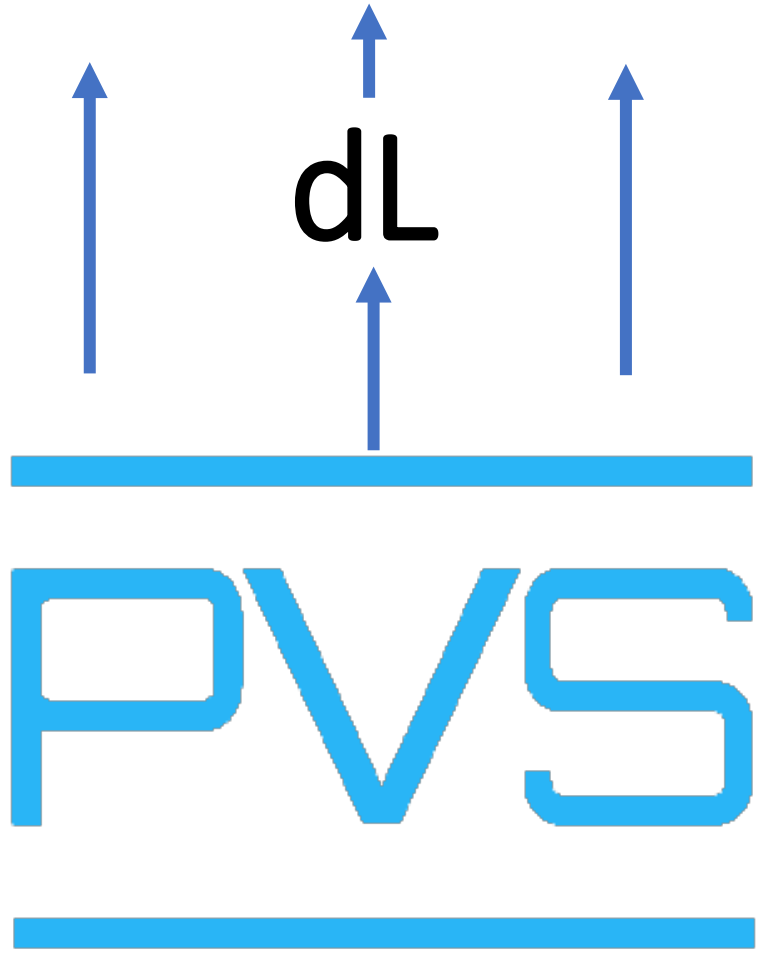
Apply substitution $y = c, x = 0 \vdash x^2 + y^2 = c^2$ $\qquad \vdash [x' := y, y' := -x](2\,x\,x' + 2\,y\,y' = 0)$

Apply Di rule $y = c\,, x = 0 \vdash [\{x' = y, y' = -x\}](x^2 + y^2 = c^2)$

[4] Differential Dynamic Logic, Logical Foundations of Cyber-Physical Systems. André Platzer. 2018 https://doi.org/10.1007/978-3-319-63588-0

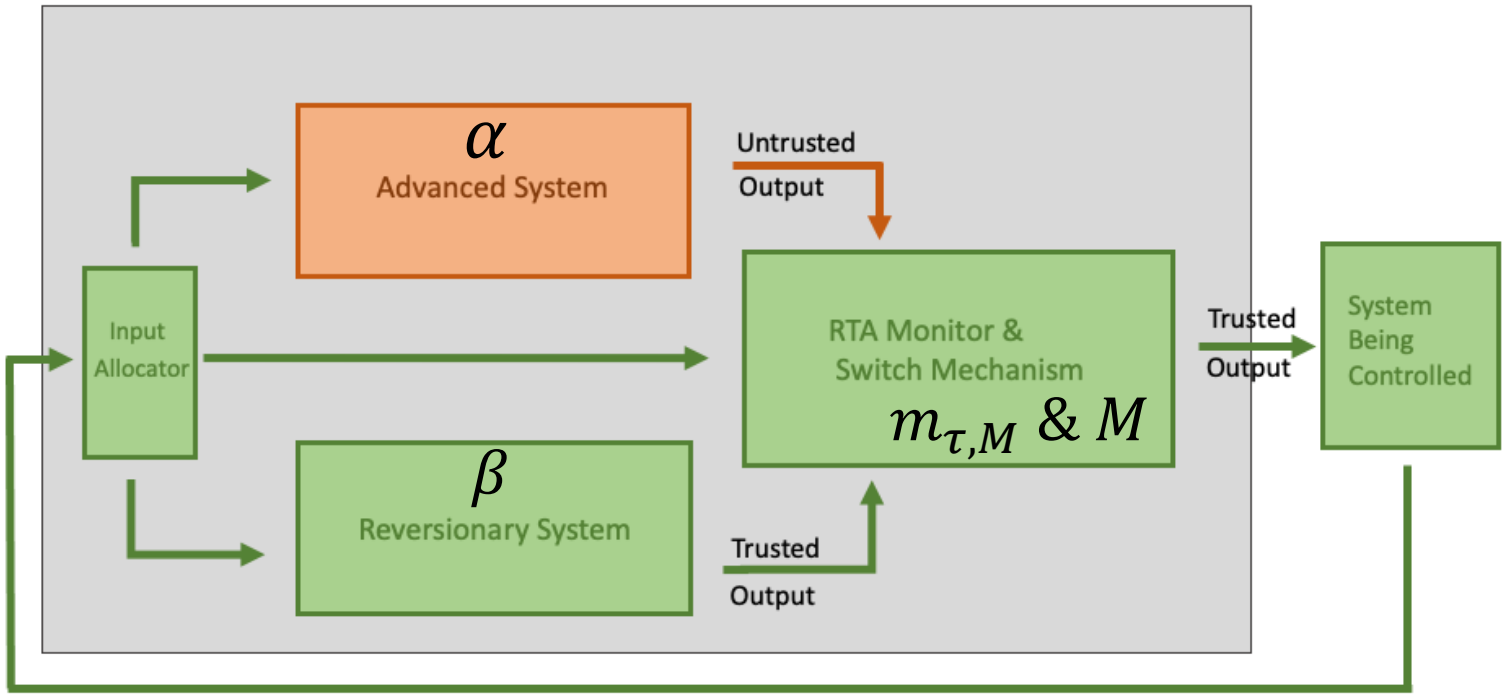
[5] **dL** "Cheat Sheet," André Platzer, https://symbolaris.com/logic/dL-sheet.pdf

Plaidypvs



[6] Embedding Differential Dynamic Logic in PVS, LSFA 2023, Slagel et. al. https://doi.org/10.48550/arXiv.2404.15214

RTA in Plaidypvs



Define

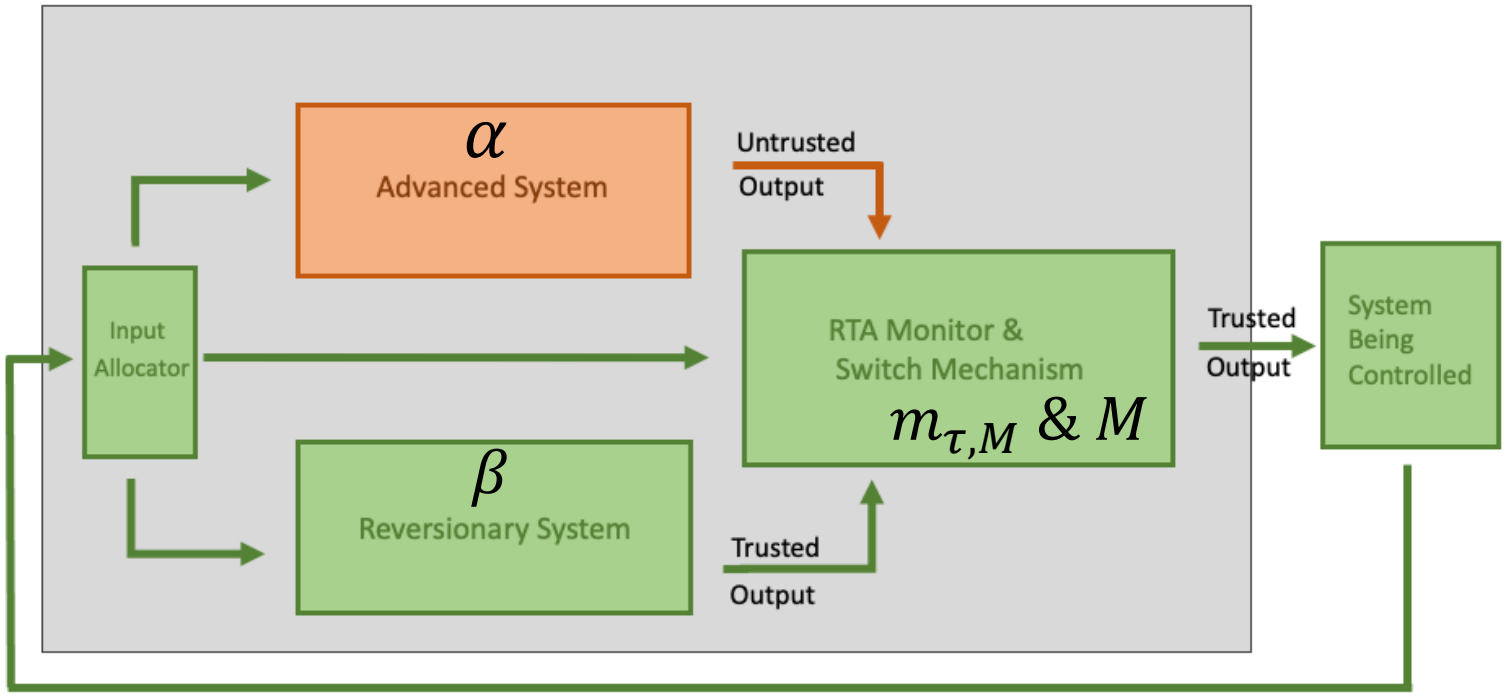
$$\left(\left(?\,M\;;m_{\tau,M}(\alpha)\right)\cup\left(?\,\neg M\;;\beta\right)\right)^{*}$$

While M is true run α with timed monitor

and

While M is not true run β with monitor

RTA in Plaidypvs



Goal:

$$[\left(\left(?\,M\,;m_{\tau,M}(\alpha)\right)\cup\left(?\,\neg M\,;\beta\right)\right]\,S$$

Want to show that for all runs of the system, some safety property S is satisfied.

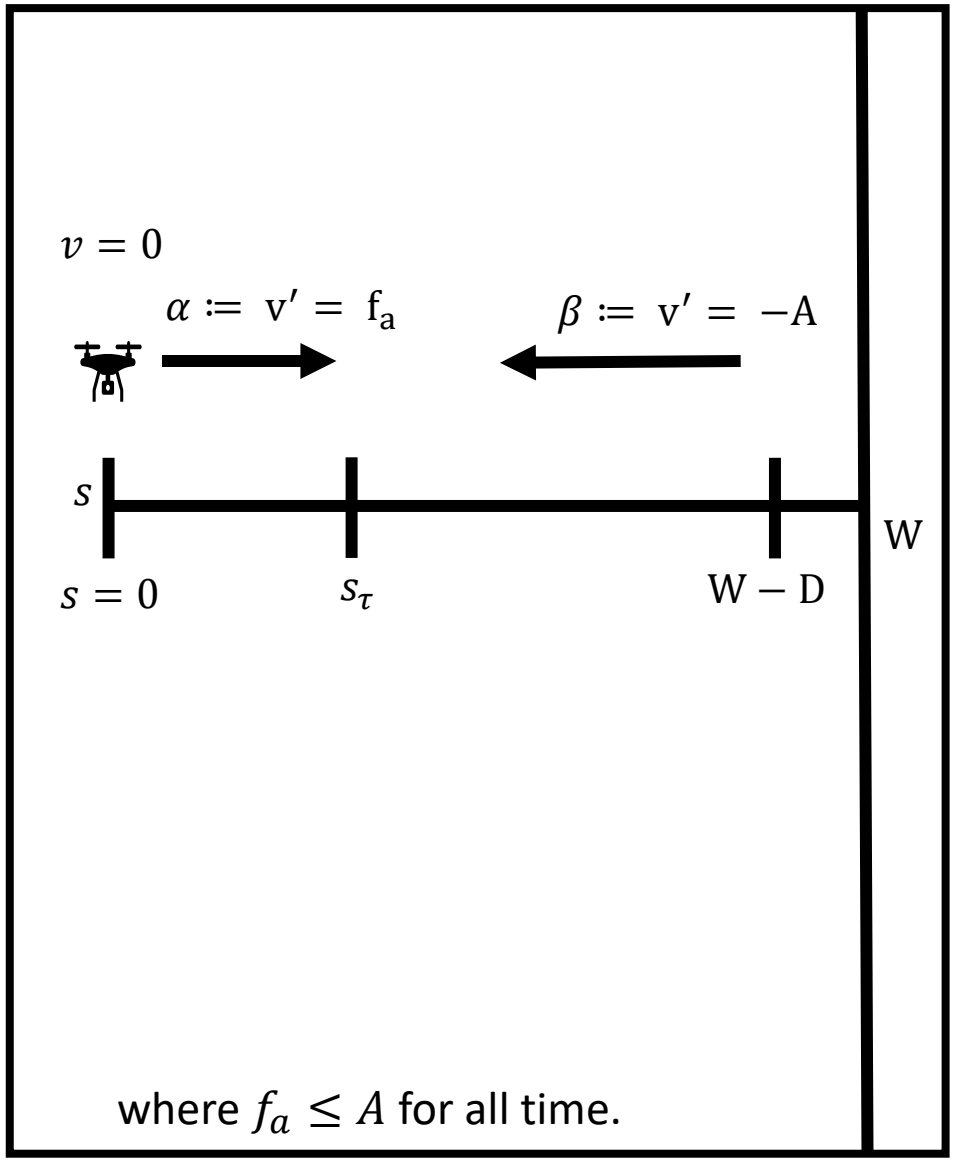
RTA rule in Plaidypvs

$$\frac{\Gamma \vdash S \wedge (M \vee G)\,, \quad S \vdash \left[m_{\tau,M}(\alpha)\right]\left(S \wedge (M \vee G)\right), \quad G \vdash [\beta^*]S}{\Gamma \vdash \left[\left(\left(?\,M\,;m_{\tau,M}(\alpha)\right) \cup (?\,\neg M\,;\beta)\right)^*\right] S}$$

Where

1. G is a user-instantiated condition that represents a property that carries over when switching from the advanced system to the reversionary system
2. M is the switch Boolean expression
3. S is the safety Boolean expression
4. $m_{\tau,M}(\alpha)$ is the monitored advanced controller
5. β is the reversionary system.

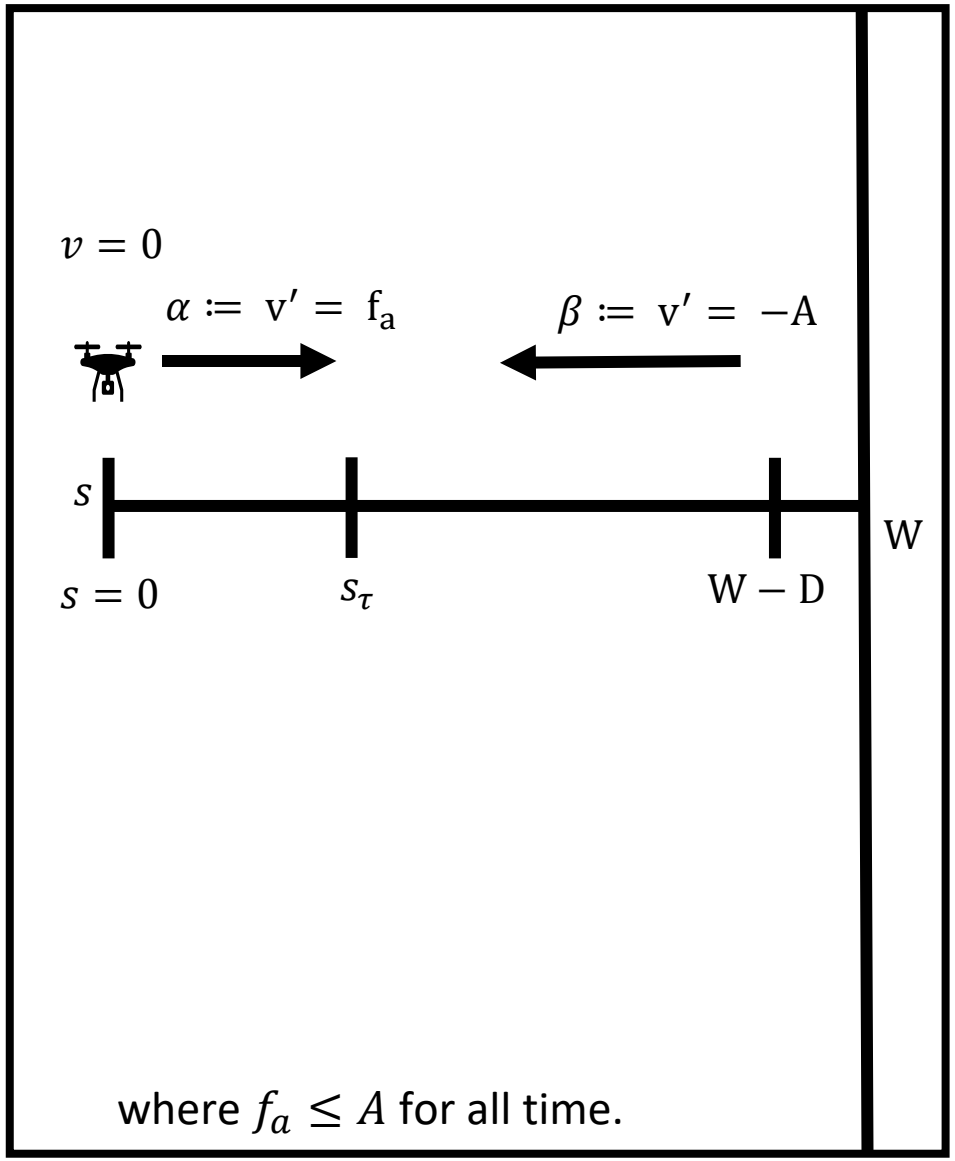
Breaking example RTA



$$[((?M\,;m_{\tau,M}(\alpha)\,)\cup(?\neg M\,;\,\beta))^*](s \leq W - D)$$

"The aircraft stays at least D distance from the wall."

Breaking example RTA



Using the RTA rule in Plaidypvs with

$$G = s \leq \frac{W - D}{2} \wedge v \leq \sqrt{A(W - D)}$$

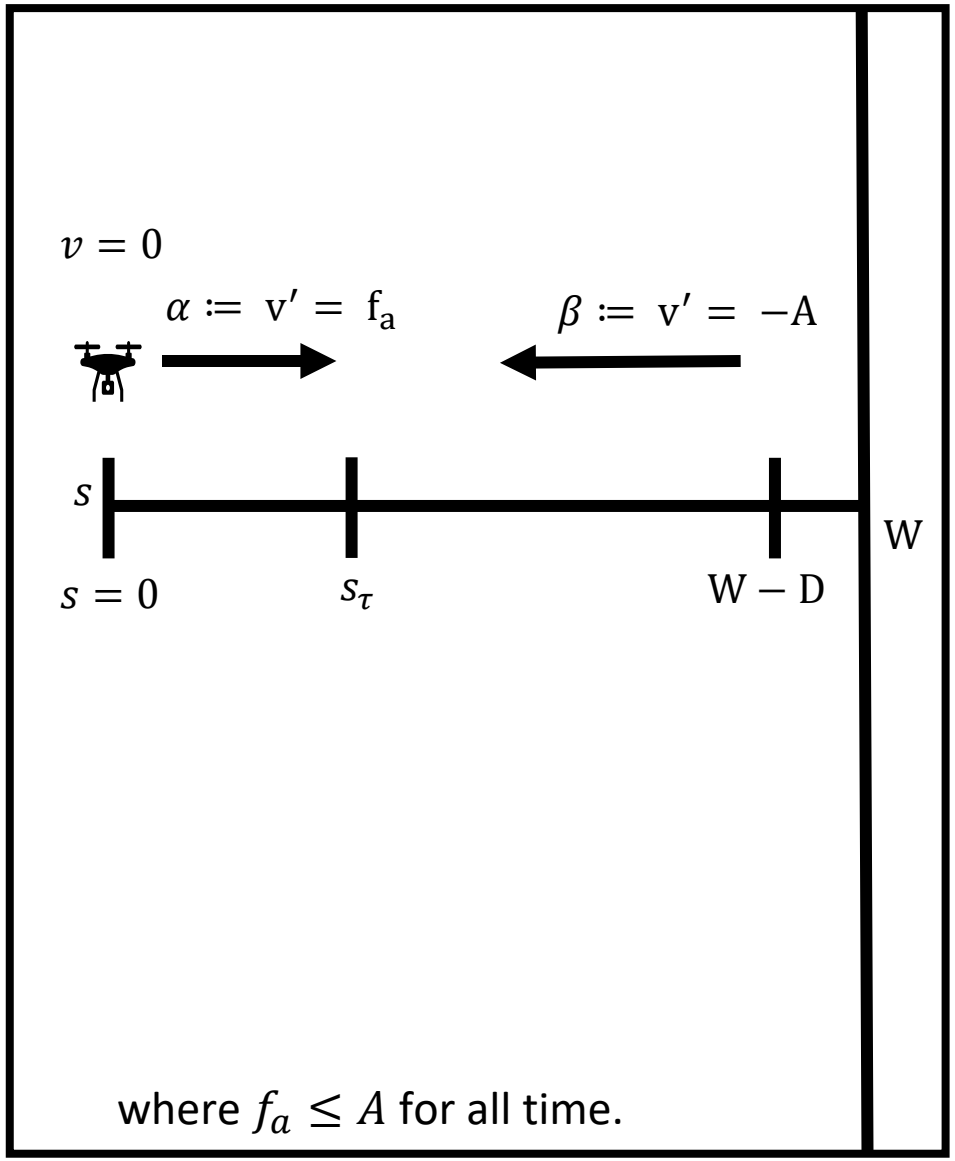
results in the three subgoals:

I. $s = 0, v = 0 \vdash (s \leq (W - D)) \wedge (M \vee G)$

II. $s \leq (W - D) \vdash [m_{\tau,M}(\alpha)]\ (s \leq (W - D)) \wedge (M \vee G)$

III. $G \vdash [\beta^*]\ (s \leq (W - D))$

Breaking example RTA



The three subgoals in the proof impose requirements for the sample rate and monitor condition:

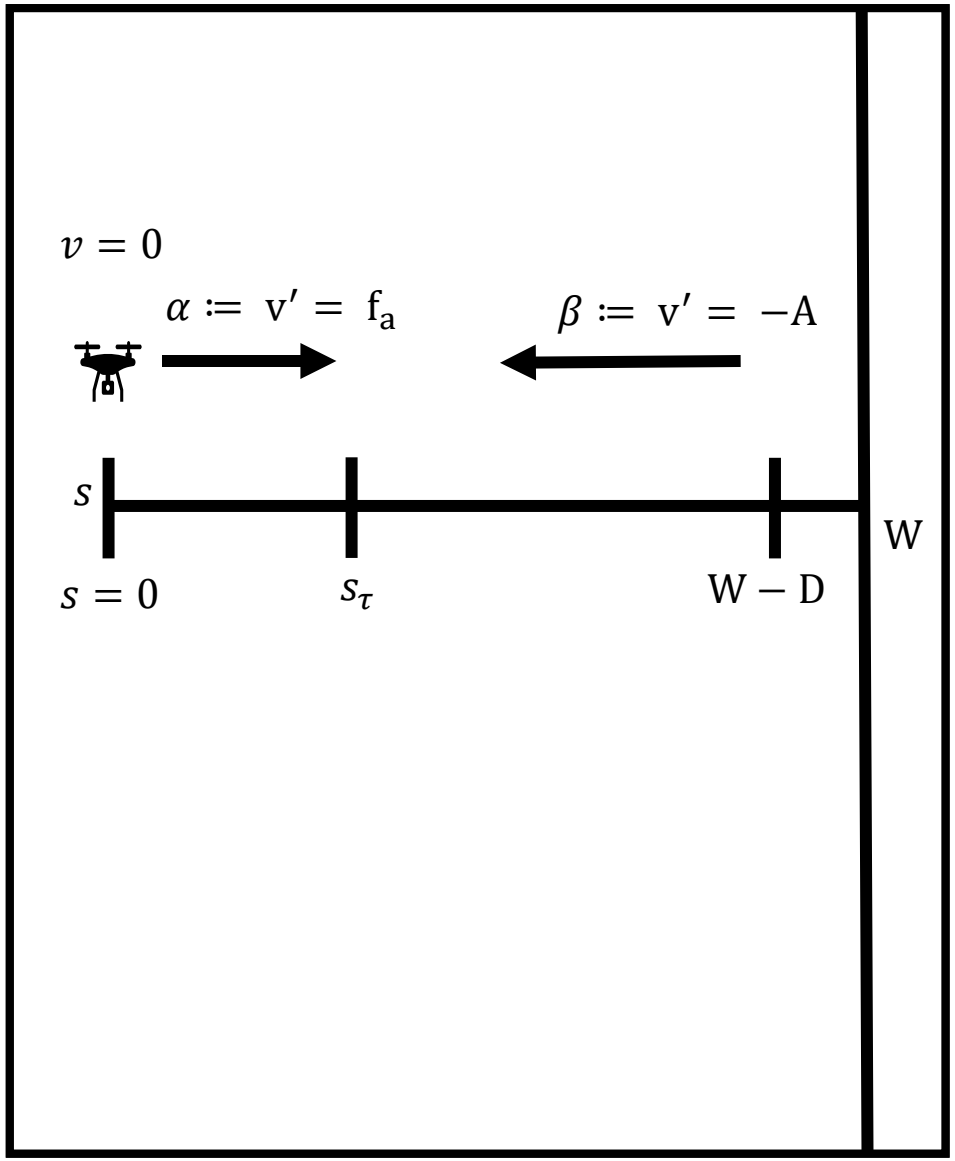
$$\tau \leq \sqrt{\frac{W - D}{A}},$$

$$M = s \leq s_\tau \wedge v \leq \sqrt{A(W - D)}$$

where

$$s_\tau = \frac{\left(\sqrt{A(W - D)} - \tau\right)^2}{2}$$

Summary

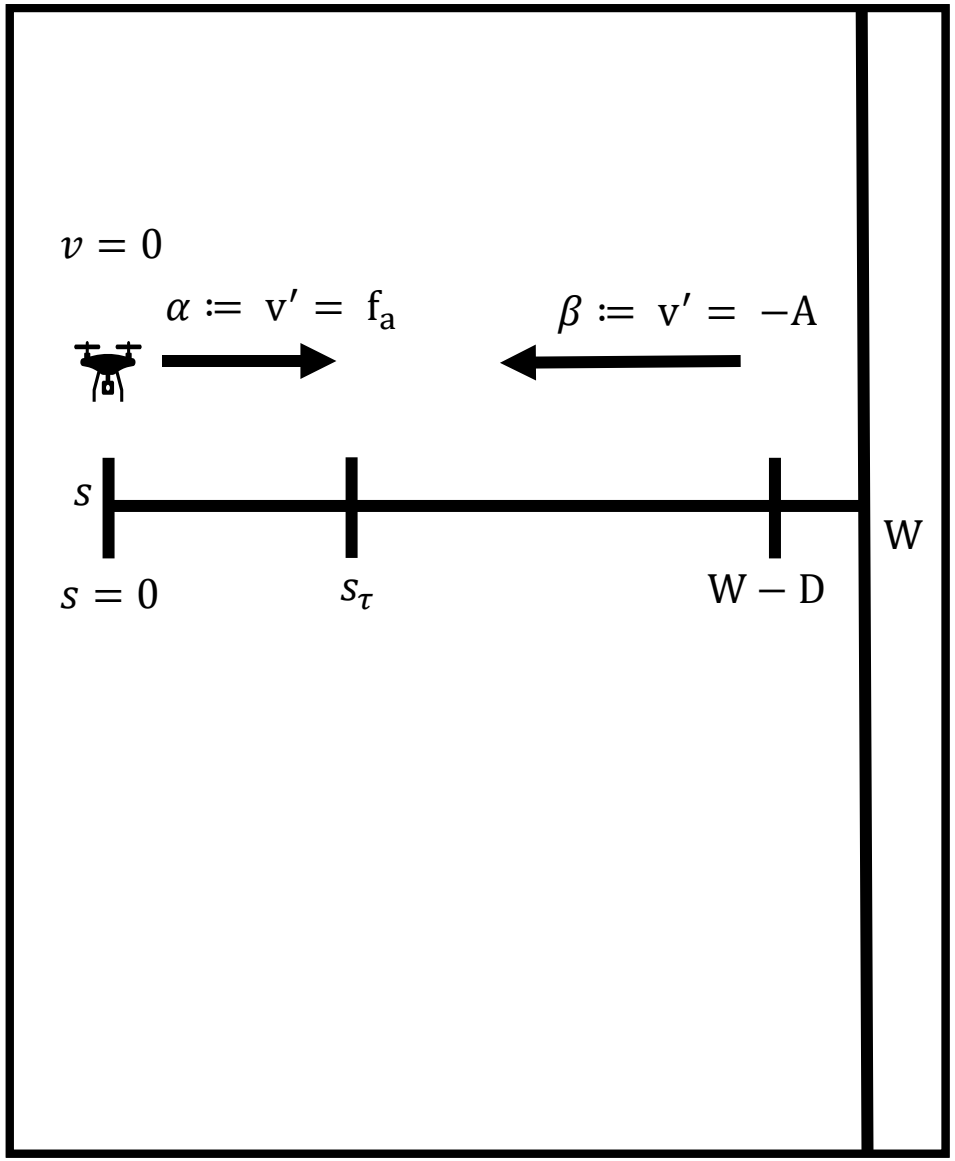


$$[((?M\,;m_{\tau,M}(\alpha)\,)\cup(?\neg M\,;\,\beta))^*](s \leq W - D)$$

Plaidypvs can

- model and formally reason about RTA Systems
- help extract and define requirements from the system

Summary



$$[((?M\,;m_{\tau,M}(\alpha)\,)\cup(?\neg M\,;\,\beta))^*](s \leq W - D)$$

Plaidypvs can

- model and formally reason about RTA Systems
- help extract and define requirements from the system

Thank you for paying attention!

Questions, comments?