Provably Safe Obstacle Avoidance of Autonomous Robotic Ground Vehicles

Stefan Mitsch
joint work with Khalil Ghorbal and André Platzer

Computer Science Department,
Carnegie Mellon University

November 21, 2013

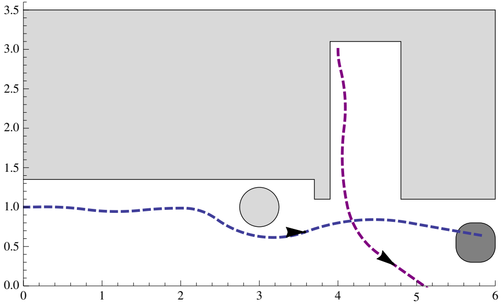
Case Study: Delivery Robot

Scenario

How can we build a robot that is safe?

High-level Requirements

Safety	"Do not collide with obstacles"
Liveness	"Arrive at a destination"



Model Variations and Verification

Obstacle Avoidance

Dynamic Window specifies robot kinematics, decouples safety from optimization $\rightsquigarrow$ well suited for hybrid safety verification

Handle complexity

Dimension 1D $\rightsquigarrow$ 2D $\rightsquigarrow$ Add floor levels

Steering Manhattan $\rightsquigarrow$ Differential $\rightsquigarrow$ Omnidirectional drive

Safety Static $\rightsquigarrow$ Passive $\rightsquigarrow$ Passive friendly $\rightsquigarrow$ Active

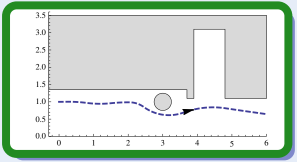
Uncertainty Sensor uncertainty $\rightsquigarrow$ Sensor failure $\rightsquigarrow$ Actuator disturbance $\rightsquigarrow$ Differential inequality models of disturbance

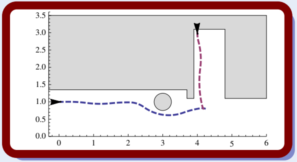
Liveness Cross goal line $\rightsquigarrow$ Before deadline $\rightsquigarrow$ Cross intersection with obstacles $\rightsquigarrow$ Before deadline $\rightsquigarrow$ Reach goal $\rightsquigarrow$ Before deadline $\rightsquigarrow$ In tricky environments $\rightsquigarrow$ Escape

Interface & Tools

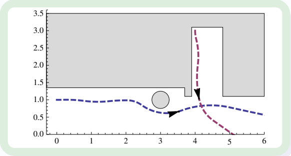
What is safe?

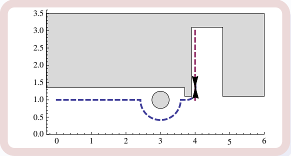
Static safety



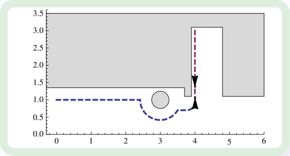


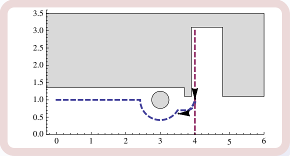
Passive safety





Passive friendly safety

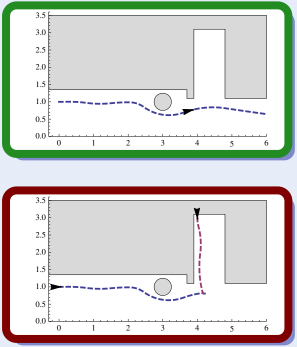




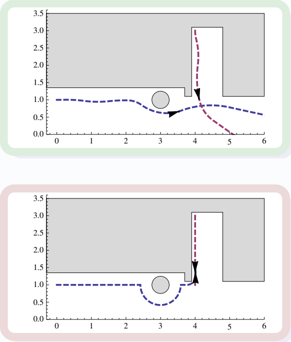
✓ Verified with KeYmaera

What is safe?

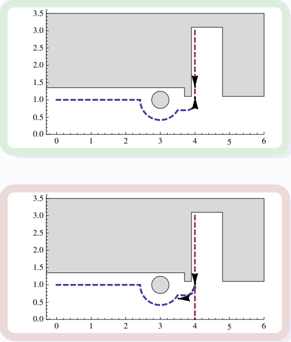
Static safety



Passive safety



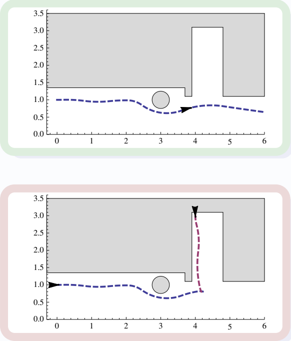
Passive friendly safety



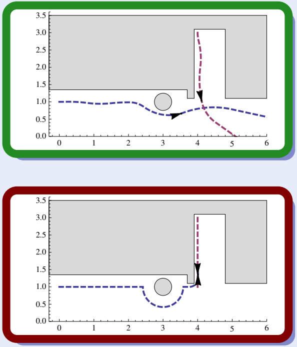
✓ Verified with KeYmaera

What is safe?

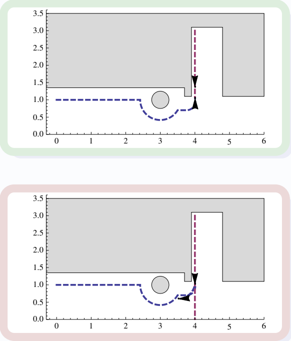
Static safety



Passive safety



Passive friendly safety



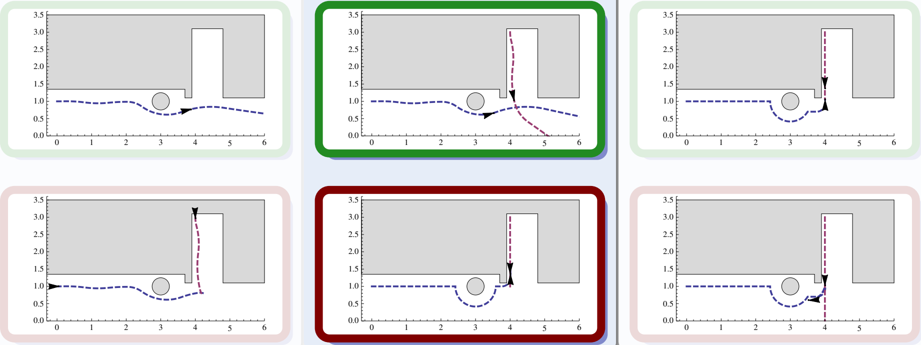
✓ Verified with KeYmaera

What is safe?

Static safety

Passive safety

Passive friendly safety



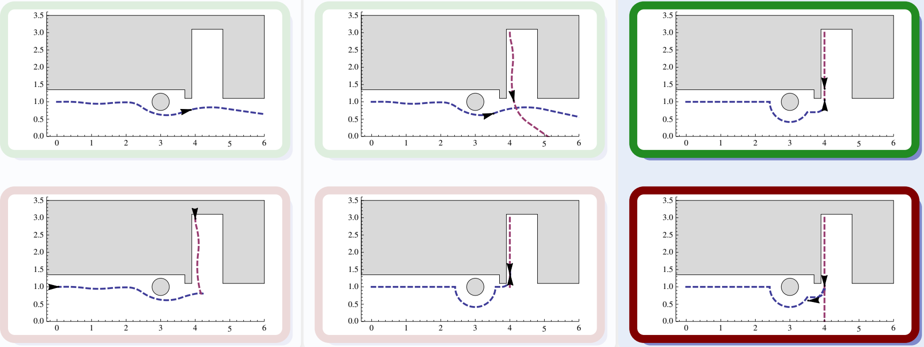
✓ Verified with KeYmaera

What is safe?

Static safety

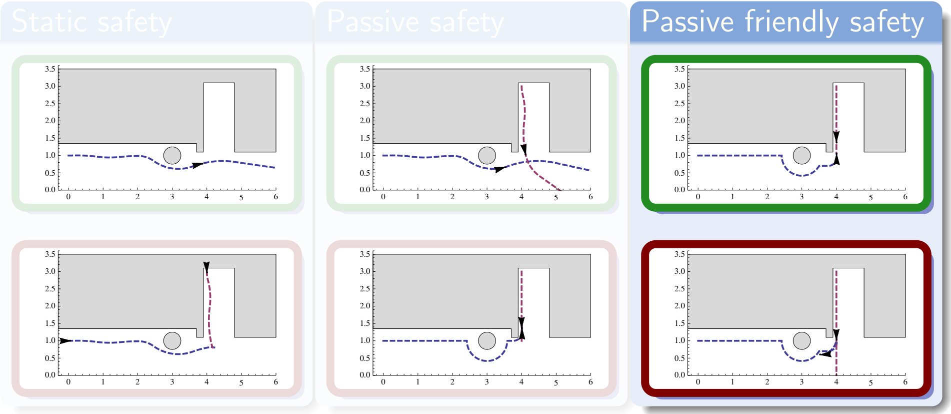
Passive safety

Passive friendly safety



✓ Verified with KeYmaera

What is safe?



✓ Verified with KeYmaera

Robot Invariants and Constraints

Safety	Invariant + Safe Control (RSS'13)
static	$\|p_r - p_o\|_\infty > \frac{v_r^2}{2b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon v_r\right)$
passive	$v_r = 0 \lor \|p_r - p_o\|_\infty > \frac{v_r^2}{2b} + V\frac{v_r}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon(v_r + V)\right)$
+ sensor	$\|\hat{p}_r - p_o\|_\infty > \frac{v_r^2}{2b} + V\frac{v_r}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon(v_r + V)\right) + U_p$
+ disturb	$\|p_r - p_o\|_\infty > \frac{v_r^2}{2bU_m} + V\frac{v_r}{bU_m} + \left(\frac{A}{bU_m}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon(v_r + V)\right)$
+ failure	$\|\hat{p}_r - p_o\|_\infty > \frac{v_r^2}{2b} + V\frac{v_r}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon(v + V)\right) + U_p + g\Delta$
friendly	$\|p_r - p_o\|_\infty > \frac{v_r^2}{2b} + \frac{V^2}{2b_o} + V\left(\frac{v_r}{b} + \tau\right) + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon(v_r + V)\right)$

Robot Invariants and Constraints

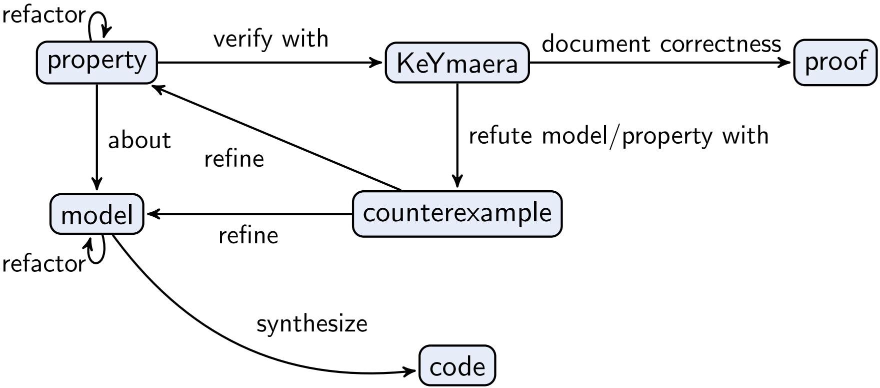
Safety	Invariant + Safe Control (RSS'13)
static	$\|p_r - p_o\|_\infty > \frac{v_r^2}{2b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon v_r\right)$
passive	$v_r = 0 \lor \|p_r - p_o\|_\infty > \frac{v_r^2}{2b} + V\frac{v_r}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon(v_r + V)\right)$
+ sensor	$+ \varepsilon(v_r + V)\big) + U_p$
+ disturb	$\|p_r - p_o\|_\infty > \frac{v_r^2}{2bU_m} + V\frac{v_r}{bU_m} + \left(\frac{A}{bU_m}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon(v_r + V)\right)$
+ failure	$\|\hat{p}_r - p_o\|_\infty > \frac{v_r^2}{2b} + V\frac{v_r}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon(v + V)\right) + U_p + g\Delta$
friendly	$\|p_r - p_o\|_\infty > \frac{v_r^2}{2b} + \frac{V^2}{2b_o} + V\left(\frac{v_r}{b} + \tau\right) + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon(v_r + V)\right)$

Question

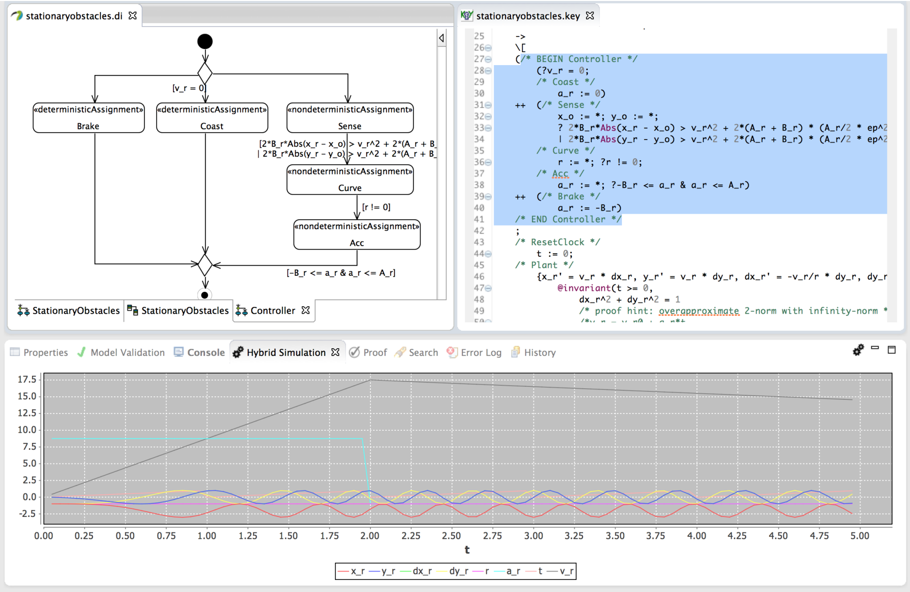
How to find and justify constraints?

Process: Modeling, Verification, Synthesis

1. Construct model along with its proof
   - Model
   - Verify $\rightsquigarrow$ Counterexample
   - Reiterate until proved
2. Synthesize code from model



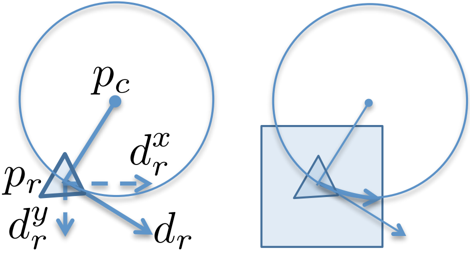
Sphinx: Graphical and Textual Modeling



Robot: State

State

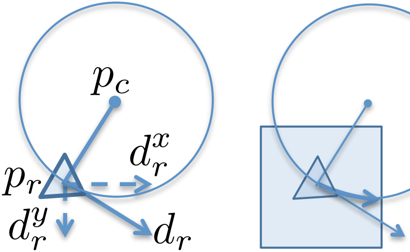
Position	$p_r = (p_r^x, p_r^y)$
Orientation	$d_r = (d_r^x = cos\theta, d_r^y = sin\theta)$
Translational velocity, acceleration	v_r, a_r
Rotational velocity	ω_r



Robot: Motion Dynamics

Dynamics

translational ODE	$p_r' = v_r d_r$	$v_r' = a_r$
rotational DAE	$\omega_r' \|p_r - p_c\| = a_r$	
	$d_r^{x\prime} = -\omega_r d_r^y$	$d_r^{y\prime} = \omega_r d_r^x$

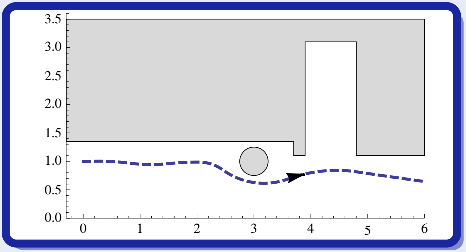


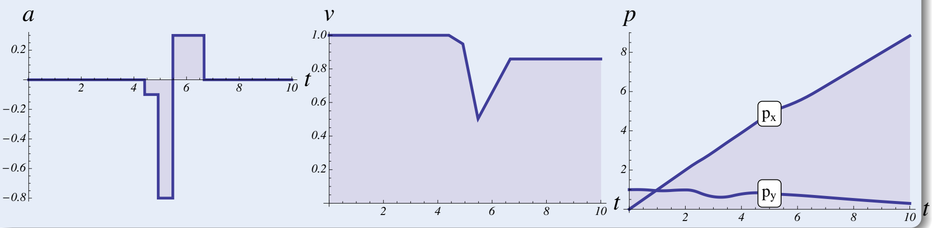
Example (Differential invariants)

1. Move on circle: $p_r - p_c = \omega d_r^\perp$
2. Stay in the box: $\|p_r - p_0\|_\infty \le v_r t + \frac{a_r}{2} t^2$

Challenge (Hybrid Systems)

$$a_r := -b$$
$$\cup\ (a_r := *; ?-b \le a_r \le A;$$
$$\omega_r := *; ?-\Omega \le \omega_r \le \Omega;$$
$$?\text{SafeCtrl})$$
$$\cup\ (?v_r = 0; a_r := 0; \omega_r := 0)$$





Robot: Control (2D)

Challenge (Hybrid Systems)

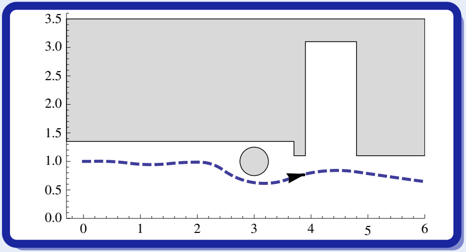
$a_r := -b$

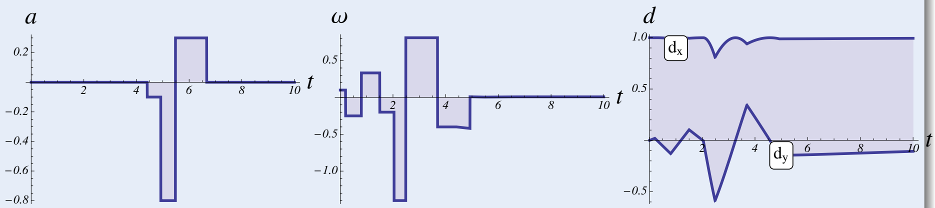
$\cup \ (a_r := *; ? - b \leq a_r \leq A;$

$\omega_r := *; ? - \Omega \leq \omega_r \leq \Omega;$

$?\text{SafeCtrl})$

$\cup \ (?v_r = 0; a_r := 0; \omega_r := 0)$





Challenge (Hybrid Systems)

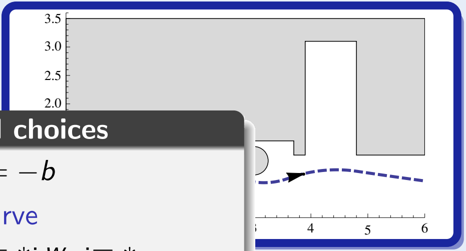
$a_r := -b$

$\cup\ (a_r := *; ?-b \le a_r \le A;$

$\omega_r := *; ?-\Omega \le \omega_r \le \Omega;$

$?$SafeCtrl$)$

$\cup\ (?v_r = 0; a_r := \ldots$



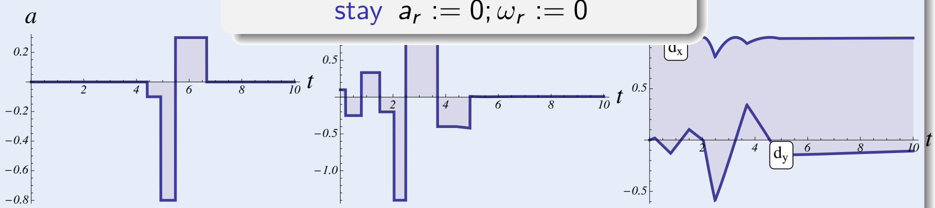
Control choices

brake $a_r := -b$

accelerate, new curve

$a_r := *; w_r := *$

stay $a_r := 0; \omega_r := 0$

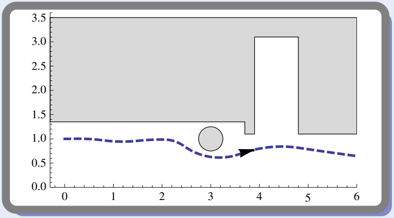


Robot: Drive Variants

Differential drive

Controller picks only curve radius

$\leadsto$ smooth orientation changes



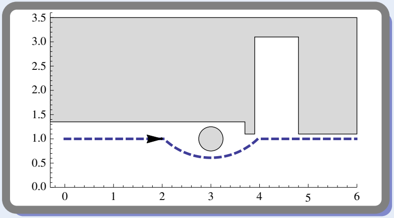
$$r := *;\ ?r \neq 0;$$

$$?r\omega_r = v_r$$

Omnidirectional drive

Controller picks orientation too

$\leadsto$ sudden orientation changes



$$p_c := *;\ ?\|p_r - p_c\| > 0;$$

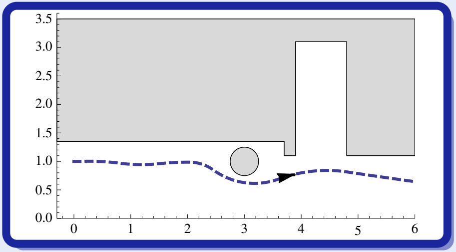
$$?\|(p_r - p_c)\omega_r\| = v_r;$$

$$d_r := \frac{(p_r - p_c)^\perp}{r}$$

Static Safety

Challenge (Hybrid Systems)

$$a_r := -b$$
$$\cup\ (a_r := *; ?-b \le a_r \le A;$$
$$\omega_r := *; ?-\Omega \le \omega_r \le \Omega;$$
$$?\text{SafeCtrl})$$
$$\cup\ (?v_r = 0; a_r := 0; \omega_r := 0)$$



Safety Condition: Non-Zero Distance to Obstacle

$$\|p_r - p_o\| > 0$$

Static Safety (d$\mathcal{L}$ Model)

Verified Property

$$\varphi_{\text{static}} \rightarrow [\text{dw}_{\text{static}}]\,(\|p_r - p_o\| > 0)$$

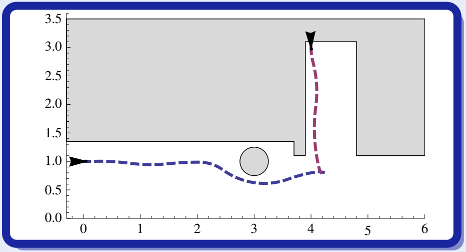
$$\begin{aligned}
\text{dw}_{\text{static}} &\equiv (\text{ctrl}_r;\ \text{dyn})^* \\
\text{ctrl}_r &\equiv \quad (a_r := -b) \\
&\quad \cup (?v_r = 0;\ a_r := 0) \\
&\quad \cup \big(a_r := *;\ ? - b \leq a_r \leq A;\ \omega_r := *;\ ? - \Omega \leq \omega_r \leq \Omega; \\
&\qquad r := *;\ ?r \neq 0 \wedge r\omega_r = v_r;\ p_o := *; ?\text{SafeCtrl}\big) \\
\text{SafeCtrl} &\equiv \|p_r - p_o\|_\infty > \frac{v_r^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon v_r\right) \\
\text{dyn} &\equiv \{p_r' = v_r d_r,\ v_r' = a_r,\ d_r' = \omega_r d_r^{\perp},\ \omega_r' = \frac{a_r}{r},\ t' = 1 \\
&\qquad \&\ v_r \geq 0 \wedge t \leq \varepsilon\}
\end{aligned}$$

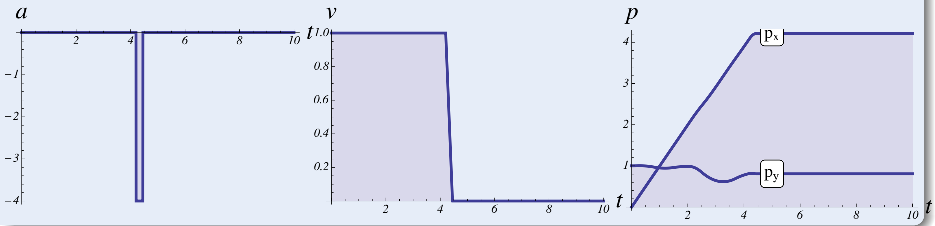
Passive Safety

Challenge (Hybrid Systems)

Moving obstacles: distance on current curve not enough

- Dynamic obstacles (other agents)
- Avoid collisions (define safety)



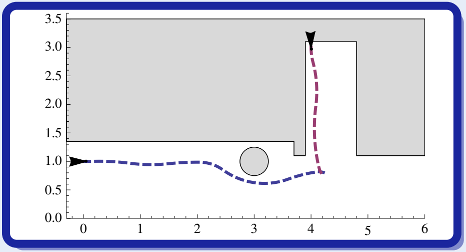


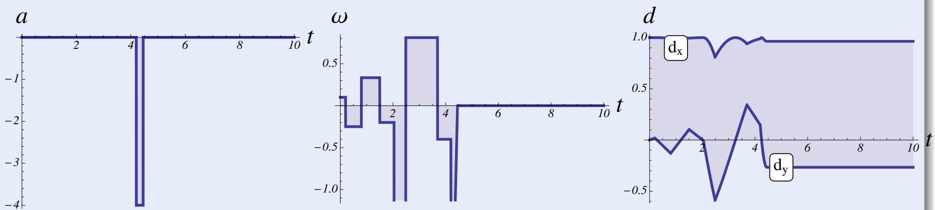
Passive Safety

Challenge (Hybrid Systems)

Moving obstacles: distance on current curve not enough

- Dynamic obstacles (other agents)
- Avoid collisions (define safety)

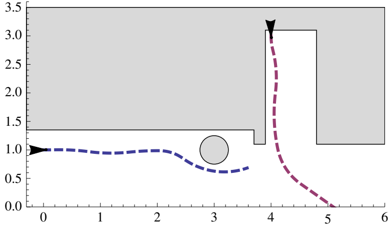




Passive Safety

Challenge (Hybrid Systems)

$$
\begin{aligned}
& a_r := -b \\
\cup\ & (a_r := *; ?-b \le a_r \le A; \\
& \omega_r := *; ?-\Omega \le \omega_r \le \Omega; \\
& ?\text{SafeCtrl}) \\
\cup\ & (?v_r = 0; a_r := 0; \omega_r := 0)
\end{aligned}
$$



Safety Condition: Already Stopped or Sufficient Space to Stop

$$v_r = 0 \lor \|p_r - p_o\| > \frac{v_r^2}{2b} + V\frac{v_r}{b}$$

Passive Safety (d$\mathcal{L}$ Model)

Verified Property

$$\varphi_{\text{ps}} \rightarrow [\text{dw}_{\text{ps}}] \left(v_r = 0 \vee \|p_r - p_o\| > \frac{v_r^2}{2b} + V \frac{v_r}{b} \right)$$

$$\text{dw}_{\text{ps}} \equiv (\text{ctrl}_o;\ \text{ctrl}_r;\ \text{dyn})^*$$

$$\text{ctrl}_o \equiv v_o := *;\ ?\|v_o\| \leq V;$$

$$\text{ctrl}_r \equiv \text{see static safety}$$

$$\text{SafeCtrl} \equiv \|p_r - p_o\|_\infty > \frac{v_r^2}{2b} + V \frac{v_r}{b} + \left(\frac{A}{b} + 1\right) \left(\frac{A}{2}\varepsilon^2 + \varepsilon(v_r + V)\right)$$

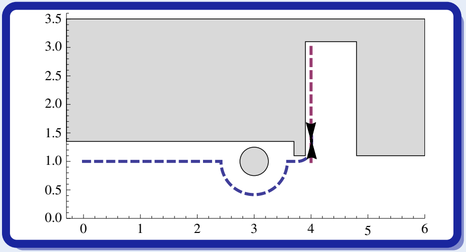
$$\text{dyn} \equiv \{p_r' = v_r d_r,\ v_r' = a_r,\ d_r' = \omega_r d_r^\perp,\ \omega_r' = \frac{a_r}{r},$$
$$p_o' = v_o,\ t' = 1\ \&\ v_r \geq 0 \wedge t \leq \varepsilon\}$$

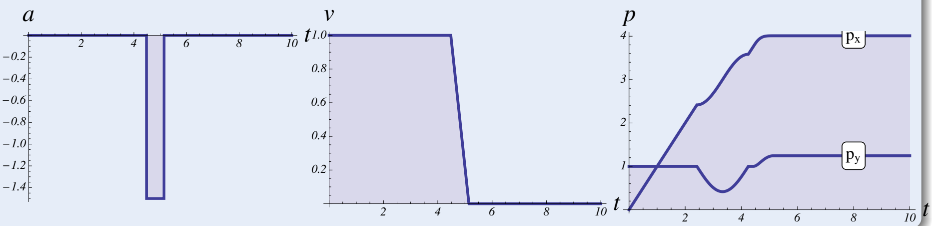
Passive Friendly Safety

Challenge (Hybrid Systems)

Passive friendly safety: don't cause unavoidable collision

- Dynamic obstacles (other agents)
- Avoid collisions (friendly safety)



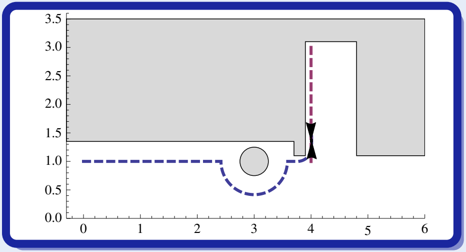


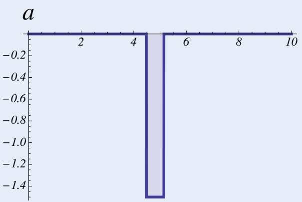
Passive Friendly Safety

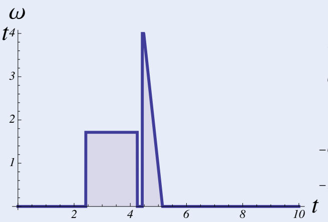
Challenge (Hybrid Systems)

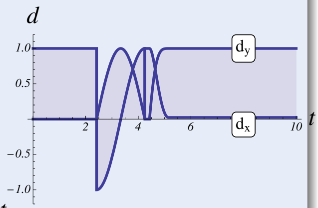
Passive friendly safety: don't cause unavoidable collision

- Dynamic obstacles (other agents)
- Avoid collisions (friendly safety)





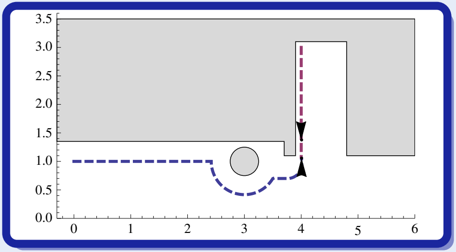




Passive Friendly Safety

Challenge (Hybrid Systems)

$$a_r := -b$$
$$\cup\ (a_r := *; ?-b \leq a_r \leq A;$$
$$\omega_r := *; ?-\Omega \leq \omega_r \leq \Omega;$$
$$?\text{SafeCtrl})$$
$$\cup\ (?v_r = 0; a_r := 0; \omega_r := 0)$$



Safety Condition: Robot and Obstacle have Sufficient Space to Stop

$$\ldots \wedge \left(v_r = 0 \wedge \|p_r - p_o\| > \frac{V^2}{2b_o} + \tau V \wedge 0 \leq v_o \leq V \right)$$
$$\rightarrow \langle obstacle \rangle \left(\|p_r - p_o\| > 0 \wedge v_o = 0 \right)$$

Passive Friendly Safety (d$\mathcal{L}$ Model)

Verified Property

$$\varphi_{\text{pfs}} \rightarrow [\text{dw}_{\text{pfs}}] \ldots \wedge \left(v_r = 0 \wedge \|p_r - p_o\| > \frac{V^2}{2b_o} + \tau V \wedge 0 \leq v_o \leq V\right) \rightarrow \langle obstacle \rangle \left(\|p_r - p_o\| > 0 \wedge v_o = 0\right)$$

$$\text{dw}_{\text{pfs}} \equiv \text{see passive safety}$$

$$\text{SafeCtrl} \equiv \|p_r - p_o\|_\infty > \frac{v_r^2}{2b} + \frac{V^2}{2b_o} + V\left(\frac{v_r}{b} + \tau\right) + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon(v_r + V)\right)$$

$$\text{obstacle} \equiv (\text{ctrl}_o;\ \text{dyn}_o)^*$$

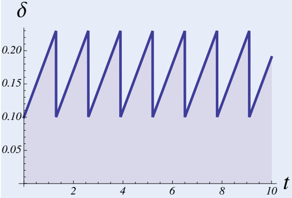
$$\text{ctrl}_o \equiv d_o := *;\ ?\|d_o\| = 1;\ a_o := *;\ ?v_o + a_o\varepsilon_o \leq V$$

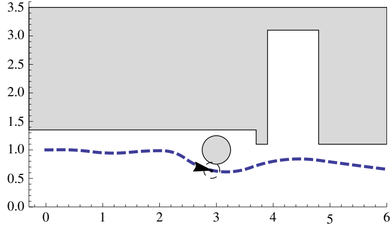
$$dyn_o \equiv (t := 0;\ p_o' = v_o d_o,\ v_o' = a_o,\ t' = 1 \ \&\ t \leq \varepsilon_o \wedge v_o \geq 0)$$

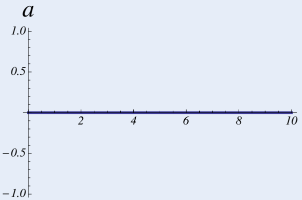
Sensor Failure and Fallback

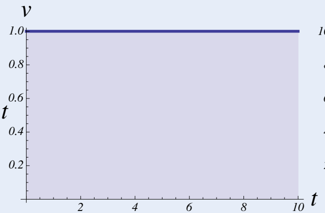
Challenge (Hybrid Systems)

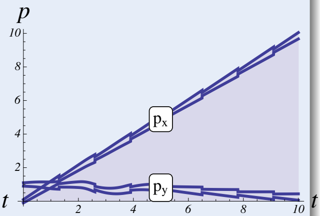
Sensor failure: Uncertainty, fallback to dead reckoning





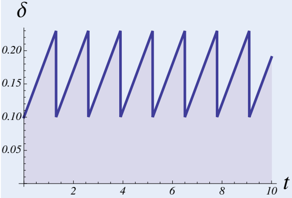


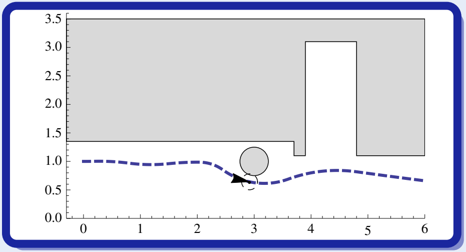


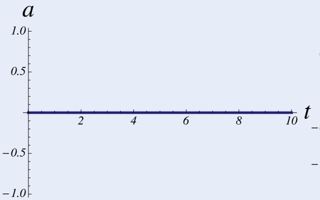


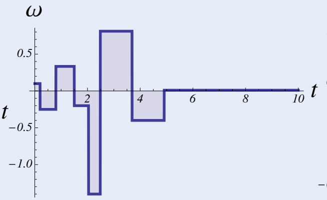
Challenge (Hybrid Systems)

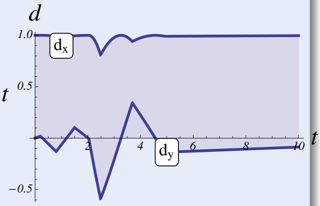
Sensor failure: Uncertainty, fallback to dead reckoning









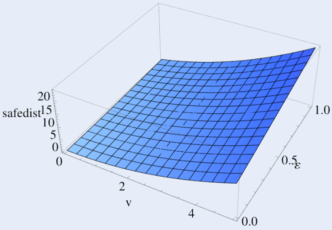


Static Obstacles: Minimum Safety Distance Estimate

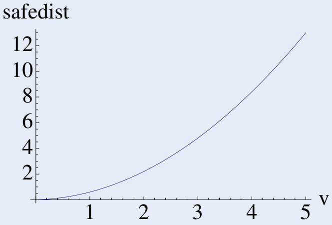
$$\|p_r - p_o\|_\infty > \frac{v_r^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon v_r\right)$$

v	A	b	ε	$\|\mathbf{p_r} - \mathbf{p_o}\|$
1	1	1	0.05	**0.61**
0.5	0.5	0.5	0.025	**0.28**
2	2	2	0.1	**1.42**
1	1	2	0.05	**0.33**
1	2	1	0.05	**0.66**

Safety distance (v_r, ε)



Safety distance (v_r)

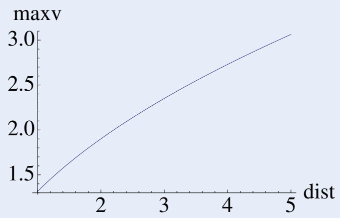


Static Obstacles: Maximum Velocity Estimate

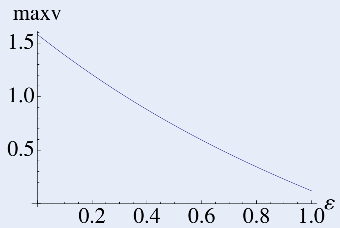
$$\|p_r - p_o\|_\infty > \frac{v_r^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon v_r\right)$$

A	b	ε	**v**
Corridor $\|p_r - p_o\| = 1.25$			
1	1	0.05	**1.48**
0.5	0.5	0.025	**1.09**
2	2	0.1	**1.85**
1	2	0.05	**2.08**
2	1	0.05	**1.43**
Door $\|p_r - p_o\| = 0.25$			
1	1	0.05	**0.61**
0.5	0.5	0.025	**0.47**
2	2	0.1	**0.63**
1	2	0.05	**0.85**
2	1	0.05	**0.56**

Maximum velocity ($\|p_r - p_o\|_\infty$)



Maximum velocity (ε)

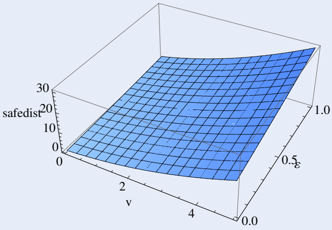


Moving Obstacles: Minimum Safety Distance Estimate

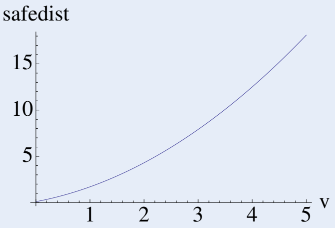
$$\|p_r - p_o\|_\infty > \frac{v_r^2}{2b} + V\frac{v_r}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon(v_r + V)\right)$$

v	A	b	V	ε	$\|\mathbf{p_r} - \mathbf{p_o}\|$
1	1	1	1	0.05	**0.61**
0.5	0.5	0.5	0.5	0.025	**0.28**
2	2	2	2	0.1	**1.42**
1	1	2	1	0.05	**0.33**
1	2	1	2	0.05	**0.66**

Safety distance (v_r, ε)



Safety distance (v_r)

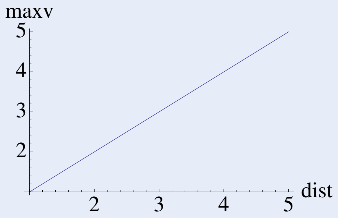


Moving Obstacles: Maximum Velocity Estimate

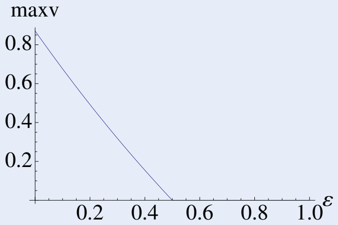
$$\|p_r - p_o\|_\infty > \frac{v_r^2}{2b} + V\frac{v_r}{b} + \left(\frac{A}{b}+1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon(v_r + V)\right)$$

A	b	V	ε	**v**
Corridor $\|p_r - p_o\| = 1.25$				
1	1	1	0.05	**0.77**
0.5	0.5	0.5	0.025	**0.69**
2	2	2	0.1	**0.61**
1	2	1	0.05	**0.4**
2	1	2	0.05	**1.3**
Door $\|p_r - p_o\| = 0.25$				
1	1	1	0.05	**0.12**
0.5	0.5	0.5	0.025	**0.18**
2	2	2	0.1	**0**
1	2	1	0.05	**0.26**
2	1	2	0.05	**1**

Maximum velocity ($\|p_r - p_o\|_\infty$)



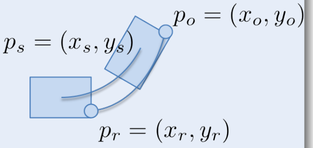
Maximum velocity (ε)



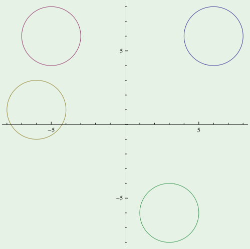
Robot Shape: Transform Obstacles

Transformation

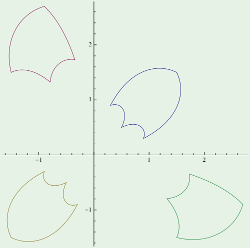
- blow up obstacles $\rightsquigarrow$ robot shape not needed
- Trajectory is **safe**, if it does **not intersect** any of the transformed regions
- Robot shape expands every point on an obstacle's shape



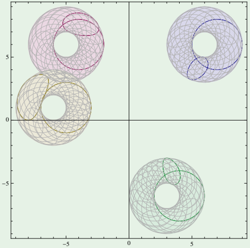
Example (Circle robot)



Example (Rectangle robot)

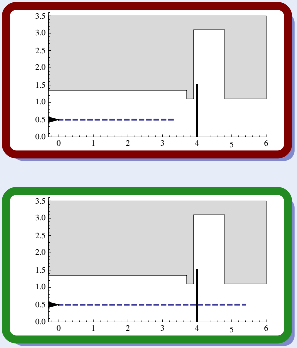


Example (Circle robot & obstacles)

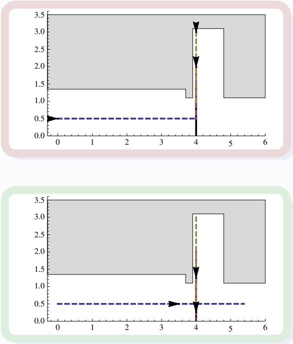


What is the goal of the robot?

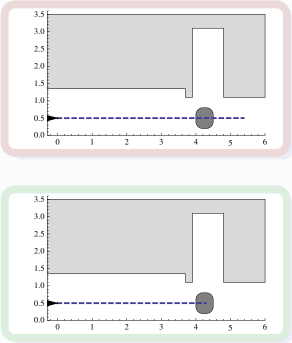
Cross finish line



Cross intersection



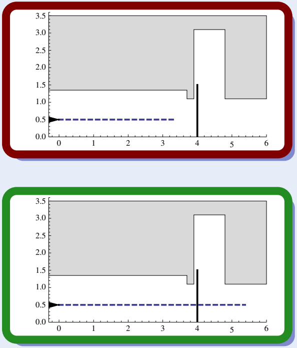
Stop at goal



✓ Verified with KeYmaera

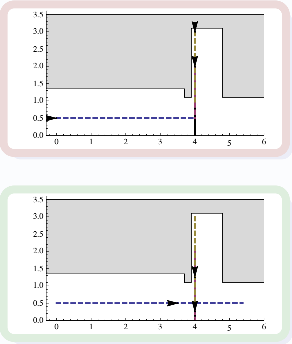
What is the goal of the robot?

Cross finish line

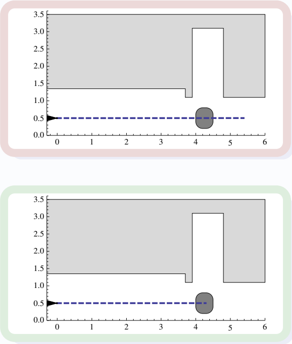


✓ Verified with KeYmaera

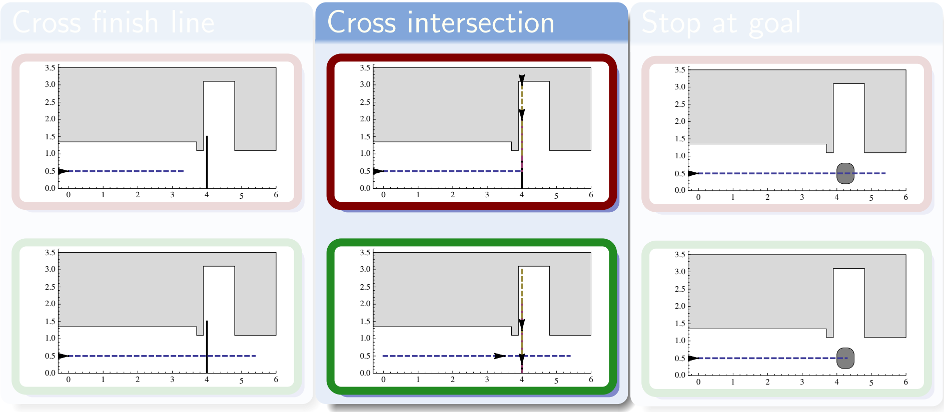
Cross intersection



Stop at goal



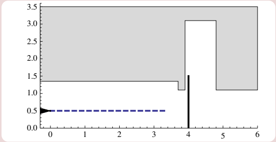
What is the goal of the robot?

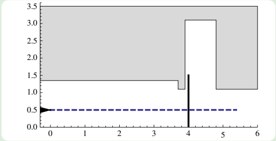


✓ Verified with KeYmaera

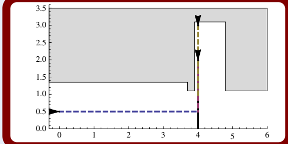
What is the goal of the robot?

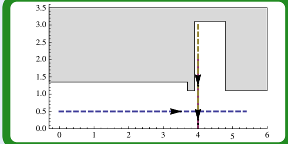
Cross finish line





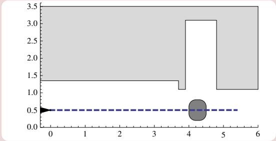
Cross intersection

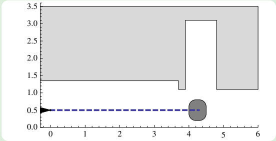




✓ Verified with KeYmaera

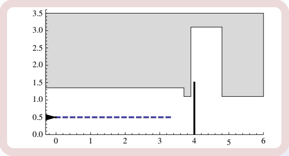
Stop at goal

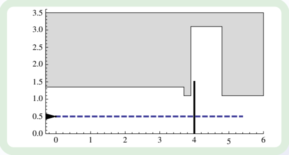




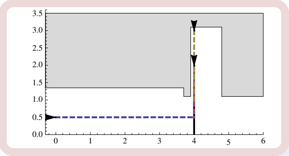
What is the goal of the robot?

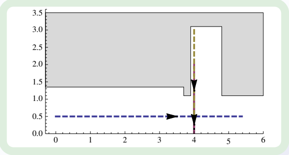
Cross finish line



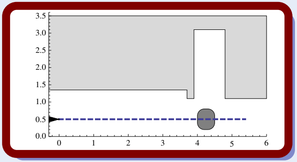


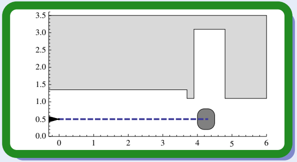
Cross intersection





Stop at goal

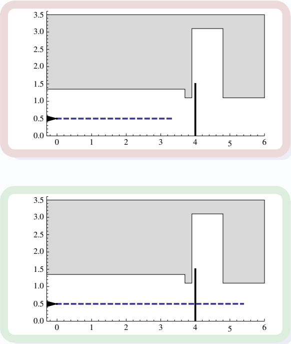




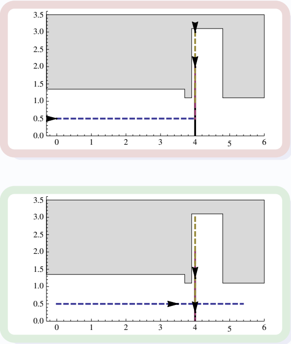
✓ Verified with KeYmaera

What is the goal of the robot?

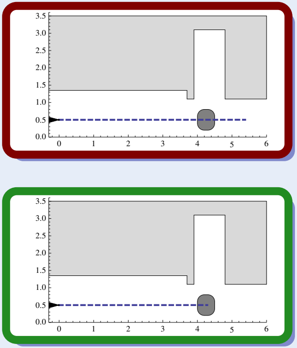
Cross finish line



Cross intersection



Stop at goal



✓ Verified with KeYmaera

Non-determinism in Verification

◀ Less deterministic   More deterministic ▶

$[\alpha]$ Safety

All runs are safe

$\langle\alpha\rangle$ Liveness

At least one run reaches the goal

$[\alpha]$ Liveness

All runs reach the goal (exclude empty set with $\langle\alpha\rangle$)

Example

$$(a_r := *) \cup (?v_r = 0; \ldots) \cup (a_r := *;\ \ldots)$$

Example

$$(a_r := *) \cup (?v_r = 0; \ldots) \cup (a_r := *;\ \ldots)$$

pick smart values

Example

if$(x_r > x_o)$ then $a_r := *;\ \ldots$

else if$(y_r < y_o)$ then $a_r := A;$

else if $\ldots$

Cross finish line ($\langle\alpha\rangle$ Liveness)

Verified Property

$$\varphi_{\text{cgl}} \rightarrow \langle cgl \rangle \,(p_g < p_r)$$

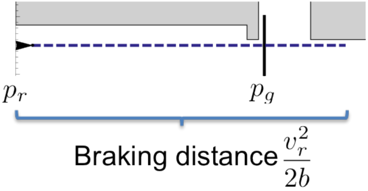
Robot is located after finish line

Verified Property

$$\varphi_{\text{cgl}} \rightarrow \langle cgl \rangle \left(v_r \geq 0 \wedge p_g < p_r + \frac{v_r^2}{2b} \right)$$

Robot reaches a point where it cannot prevent passing the finish line, not even by fully braking

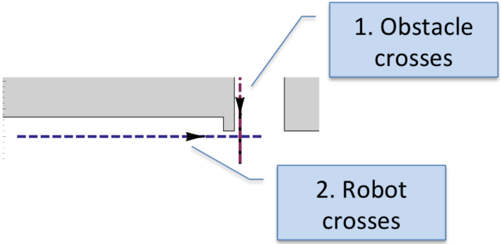
$$cgl \equiv (ctrl;\ dyn)^*$$
$$ctrl \equiv a_r := *;\ ?-b \leq a_r \leq A;$$
$$dyn \equiv (t := 0;\ p_r' = v_r,\ v_r' = a_r,\ t' = 1\ \&\ t \leq \varepsilon \wedge v_r \geq 0)$$



Cross intersection ($\langle\alpha\rangle$ Liveness for one obstacle)

Verified Property

$$\varphi_{\text{cio}} \rightarrow \langle cio\rangle \left(p_o^y > p_r^y \land \langle cio\rangle \left(p_r^x > p_o^x\right)\right)$$



Cross intersection ($\langle \alpha \rangle$ Liveness for one obstacle)

Verified Property

$$\varphi_{\text{cio}} \rightarrow \langle cio \rangle \big(p_o^y > p_r^y \wedge \langle cio \rangle \left(p_r^x > p_o^x \right) \big)$$

$$cio \equiv \big((ctrl_o \parallel ctrl_r);\ dyn \big)^*$$

$$ctrl_o \equiv a_o := *;\ ?-b \leq a_o \leq A;$$

$$ctrl_r \equiv \begin{cases} a_r := *;\ ?-b \leq a_r \leq A & \text{if AfterX} \\ a_r := *;\ ?0 \leq a_r \leq A & \text{if PassFaster} \\ a_r := 0 & \text{if PassConst} \\ (a_r := -b) \cup (?v_r = 0;\ a_r := 0) & \\ \quad \cup\ (?SafeCtrl;\ a_r := *;\ ?\dots) & \text{else} \end{cases}$$

$$\text{SafeCtrl} \equiv p_r^y < p_o^y \vee p_r^x + \frac{v_r^2}{2b} + \left(\frac{A}{b} + 1 \right) \left(\frac{A}{2}\varepsilon^2 + \varepsilon v_r \right) < p_o^x$$

$$dyn \equiv (t := 0;\ p_r^{x\prime} = v_r,\ v_r' = a_r,\ p_o^{y\prime} = v_o,\ v_o' = a_o, t' = 1$$
$$\&\ t \leq \varepsilon \wedge v_r \geq 0 \wedge v_o \geq V_{min})$$

Cross intersection ($\langle\alpha\rangle$ Liveness for one obstacle)

Verified Property

$$\varphi_{\text{cio}} \rightarrow \langle cio \rangle \left(p_o^y > p_r^y \wedge \langle cio \rangle \left(p_r^x > p_o^x \right) \right)$$

$$ctrl_r \equiv \begin{cases} a_r := *;\ ?-b \le a_r \le A & \text{if AfterX} \\ a_r := *;\ ?0 \le a_r \le A & \text{if PassFaster} \\ a_r := 0 & \text{if PassConst} \\ (a_r := -b) \cup (?v_r = 0;\ a_r := 0) & \\ \quad \cup\ (?SafeCtrl;\ a_r := *;\ ?\ldots) & \text{else} \end{cases}$$

$$\text{AfterX} \equiv p_r^x > p_o^x$$

- robot can do whatever it wants if it passed the intersection

Cross intersection ($\langle\alpha\rangle$ Liveness for one obstacle)

Verified Property

$$\varphi_{\text{cio}} \rightarrow \langle cio\rangle\big(p_o^y > p_r^y \wedge \langle cio\rangle \left(p_r^x > p_o^x\right)\big)$$

$$ctrl_r \equiv \begin{cases} a_r := *;\ ?-b \leq a_r \leq A & \text{if AfterX} \\ a_r := *;\ ?0 \leq a_r \leq A & \text{if PassFaster} \\ a_r := 0 & \text{if PassConst} \\ (a_r := -b) \cup (?v_r = 0;\ a_r := 0) & \\ \quad \cup\ (?SafeCtrl;\ a_r := *;\ ?\ldots) & \text{else} \end{cases}$$

$$\text{PassFaster} \equiv v_r > 0 \wedge \left(p_o^y + v_o \frac{p_o^x - p_r^x}{v_r} + A\left(\frac{p_o^x - p_r^x}{v_r}\right)^2 < p_r^y \right.$$
$$\left. \vee\ p_r^y < p_o^y + V_{min} \frac{p_o^x - p_r^x}{v_r + A\varepsilon}\right)$$

- robot can pass in front, even when obstacle accelerates for the remaining time to the intersection
- robot can pass behind obstacle, even when obstacle drives with minimum speed and robot accelerates once

Cross intersection ($\langle\alpha\rangle$ Liveness for one obstacle)

Verified Property

$$\varphi_{\text{cio}} \rightarrow \langle cio\rangle\big(p_o^y > p_r^y \land \langle cio\rangle\,(p_r^x > p_o^x)\big)$$

$$ctrl_r \equiv \begin{cases} a_r := *;\ ?-b \leq a_r \leq A & \text{if AfterX} \\ a_r := *;\ ?0 \leq a_r \leq A & \text{if PassFaster} \\ a_r := 0 & \text{if PassConst} \\ (a_r := -b) \cup (?v_r = 0;\ a_r := 0) & \\ \quad \cup\ (?SafeCtrl;\ a_r := *;\ ?\ldots) & \text{else} \end{cases}$$

$$\text{PassConst} \equiv v_r > 0 \land p_r^y < p_o^y + V_{min}\frac{p_o^x - p_r^x}{v_r}$$

- robot can pass behind obstacle, even when obstacle drives with minimum speed for the remaining time

Cross intersection ($[\alpha]$ Liveness for one obstacle)

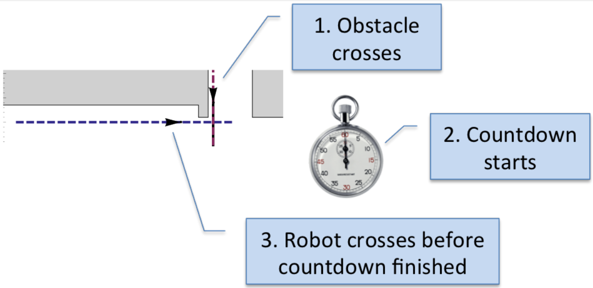
Verified Property

$$\varphi_{\text{cio}} \land T = \frac{p_r^y - p_o^y}{V_{min}} \rightarrow \exists D \Big(deadline \rightarrow [cio] \left(safe \land afterx \right) \Big)$$

$$deadline \equiv D \leq -\varepsilon \land p_r^x + \frac{A}{2} (D + \varepsilon)^2 > p_o^x$$

$$safe \equiv \|p_r - p_o\| > 0$$

$$afterx \equiv (T = D \rightarrow p_r^x > p_o^x)$$



Cross intersection ([α] Liveness for one obstacle)

Verified Property

$$\varphi_{\text{cio}} \land T = \frac{p_r^y - p_o^y}{V_{min}} \rightarrow \exists D \Big(deadline \rightarrow [cio] \, (safe \land afterx) \Big)$$

$$deadline \equiv D \leq -\varepsilon \land p_r^x + \frac{A}{2} (D + \varepsilon)^2 > p_o^x$$

$$safe \equiv \|p_r - p_o\| > 0$$

$$afterx \equiv (T = D \rightarrow p_r^x > p_o^x)$$

Characteristics

- The robot will be after the intersection exactly at the deadline
- T is a count-down to D
- Deadline D is negative and T starts positive, because we need the zero-crossing point in the proof (if $T \leq 0$, robot must accelerate to make it past intersection before deadline)
- Deadline must be at least ε, otherwise robot cannot react

Cross intersection ($[\alpha]$ Liveness for one obstacle)

Verified Property

$$\varphi_{\text{cio}} \wedge T = \frac{p_r^y - p_o^y}{V_{min}} \rightarrow \exists D \Big(deadline \rightarrow [cio] \, (safe \wedge afterx) \Big)$$

$$cio \equiv \big((ctrl_o \parallel ctrl_r);\; dyn \big)^*$$

$$ctrl_o \equiv a_o := *;\; ?-b \le a_o \le A;$$

$$ctrl_r \equiv \begin{cases} a_r := *;\; ?-b \le a_r \le A & \text{if AfterX} \\ a_r := A & \text{if ObsPassed} \\ a_r := *;\; ?0 \le a_r \le A & \text{if PassFaster} \\ a_r := 0 & \text{if PassConst} \\ (a_r := -b) \cup (?v_r = 0;\; a_r := 0) & \\ \quad \cup\, (?SafeCtrl;\; a_r := *;\; ?\ldots) & \text{else} \end{cases}$$

$$dyn \equiv (t := 0;\; p_r^{x\prime} = v_r,\; v_r' = a_r,\; p_o^{y\prime} = v_o,\; v_o' = a_o, t' = 1, T' = -1$$
$$\& \; t \le \varepsilon \wedge v_r \ge 0 \wedge v_o \ge V_{min} \wedge T \ge D)$$

Cross intersection ([α] Liveness for one obstacle)

Verified Property

$$\varphi_{\text{cio}} \land T = \frac{p_r^y - p_o^y}{V_{min}} \rightarrow \exists D \Big(deadline \rightarrow [cio] \left(safe \land afterx \right) \Big)$$

$$ctrl_r \equiv \begin{cases} a_r := *;\ ?-b \leq a_r \leq A & \text{if AfterX} \\ a_r := A & \text{if ObsPassed} \\ a_r := *;\ ?0 \leq a_r \leq A & \text{if PassFaster} \\ a_r := 0 & \text{if PassConst} \\ (a_r := -b) \cup (?v_r = 0;\ a_r := 0) & \\ \quad \cup\ (?SafeCtrl;\ a_r := *;\ ?\ldots) & \text{else} \end{cases}$$

$$\text{AfterX} \equiv p_r^x > p_o^x \qquad \text{ObsPassed} \equiv p_r^y < p_o^y$$

$$\text{PassFaster} \equiv v_r > 0 \land \left(p_o^y + v_o \frac{p_o^x - p_r^x}{v_r} + A \left(\frac{p_o^x - p_r^x}{v_r} \right)^2 < p_r^y \right.$$
$$\left. \lor\ p_r^y < p_o^y + V_{min} \frac{p_o^x - p_r^x}{v_r + A\varepsilon} \right)$$

$$\text{PassConst} \equiv v_r > 0 \land p_r^y < p_o^y + V_{min} \frac{p_o^x - p_r^x}{v_r}$$

Cross intersection ([α] Liveness for two obstacles)

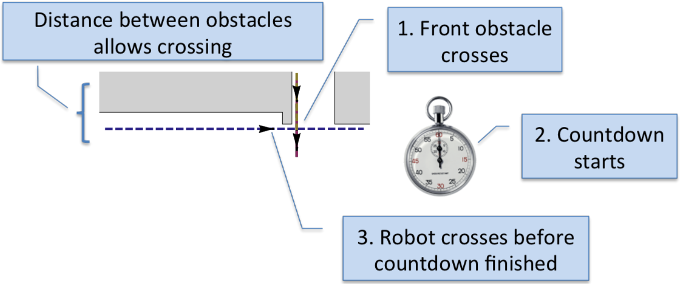
Verified Property

$$\varphi_{\text{ci2o}} \wedge T = \frac{p_{o1}^y - p_r^y}{V_{min}} \rightarrow \exists D \left(deadline \rightarrow [ci2o]\,(safe \wedge afterx) \right)$$

$$deadline \equiv \left(D \geq \varepsilon \wedge p_r^x + \frac{A}{2}(D-\varepsilon)^2 > p_o^x \wedge \frac{p_r^y - p_{o2}^y}{V_{max}} > D - T \right)$$

$$safe \equiv \|p_r - p_{o1}\| > 0 \wedge \|p_r - p_{o2}\| > 0$$

$$afterx \equiv (T \geq D \rightarrow p_r^x > p_o^x) \wedge (p_r^x \leq p_o^x \rightarrow p_{o2}^y < p_r^y)$$



Cross intersection ($[\alpha]$ Liveness for two obstacles)

Verified Property

$$\varphi_{\text{ci2o}} \wedge T = \frac{p^y_{o1} - p^y_r}{V_{min}} \rightarrow \exists D \left(deadline \rightarrow [ci2o]\,(safe \wedge afterx) \right)$$

$$deadline \equiv \left(D \geq \varepsilon \wedge p^x_r + \frac{A}{2}(D-\varepsilon)^2 > p^x_o \wedge \frac{p^y_r - p^y_{o2}}{V_{max}} > D - T\right)$$

$$safe \equiv \|p_r - p_{o1}\| > 0 \wedge \|p_r - p_{o2}\| > 0$$

$$afterx \equiv (T \geq D \rightarrow p^x_r > p^x_o) \wedge (p^x_r \leq p^x_o \rightarrow p^y_{o2} < p^y_r)$$

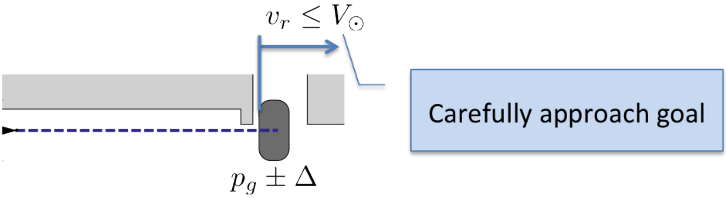
Characteristics

- The robot will be after the intersection for all times after the deadline
- T is a count-up to D
- Deadline must be large enough so that robot can make it past intersection by fully accelerating
- Second obstacle must leave sufficient space for robot to pass (maximum velocity)

Reach goal ($\langle\alpha\rangle$ Liveness)

Verified Property

$$\varphi_{\mathrm{rg}} \rightarrow \langle rg \rangle \big(p_g - \Delta < p_r \wedge 0 \leq v_r \leq V_\odot \wedge \langle rg \rangle (v_r = 0)\big) \wedge [rg] (p_r < p_g + \Delta)$$



Reach goal ($\langle \alpha \rangle$ Liveness)

Verified Property

$$\varphi_{rg} \rightarrow \langle rg \rangle \left(p_g - \Delta < p_r \land 0 \leq v_r \leq V_\odot \land \langle rg \rangle (v_r = 0)\right) \land [rg] \left(p_r < p_g + \Delta\right)$$

$$\begin{aligned}
rg &\equiv (ctrl;\ dyn)^* \\
ctrl &\equiv (a_r := -b) \\
&\quad \cup \left(?p_r < p_g - \Delta \land v_r \leq V_\odot;\ a_r := *;\ ? - b \leq a_r \leq \frac{V_\odot - v_r}{\varepsilon} \leq A\right) \\
&\quad \cup (?v_r = 0;\ a_r := 0) \\
&\quad \cup \left(?p_r + \frac{v_r^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon v_r\right) < p_g + \Delta;\right. \\
&\qquad\quad \left. a_r := *;\ ? - b \leq a_r \leq A\right); \\
dyn &\equiv (t := 0;\ p_r' = v_r,\ v_r' = a_r,\ t' = 1\ \&\ t \leq \varepsilon \land v_r \geq 0)
\end{aligned}$$

Reach goal before deadline ([α] Liveness)

Verified Property

$$\varphi_{\text{rgbd}} \land T > \varepsilon + \frac{p_g - \Delta - p_r}{V_\odot} + \frac{V_\odot - v_r}{A} + \frac{V_\odot}{b}$$
$$\rightarrow [rg_{\text{bd}}] \left(p_r < p_g + \Delta \land \left(T \leq 0 \rightarrow (p_g - \Delta < p_r \land v_r = 0) \right) \right)$$

$$rg_{\text{bd}} \equiv (ctrl;\ dyn)^*$$

$$ctrl \equiv \begin{cases} (a_r := -b) \\ \quad \cup (?v_r = 0;\ a_r := 0) & \text{if } p_r > p_g - \Delta \\ a_r := A & \text{if } p_r + \frac{v_r^2 - V_\odot^2}{2b} \\ & + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2}\varepsilon^2 + \varepsilon v_r\right) \leq p_g - \Delta \\ a_r := *; \\ \quad ? - b \leq a_r \leq \frac{V_\odot - v_r}{\varepsilon} \leq A & \text{else} \end{cases}$$

$$dyn \equiv (t := 0;\ p_r' = v_r,\ v_r' = a_r,\ t' = 1, T' = -1\ \&\ t \leq \varepsilon \land v_r \geq 0)$$

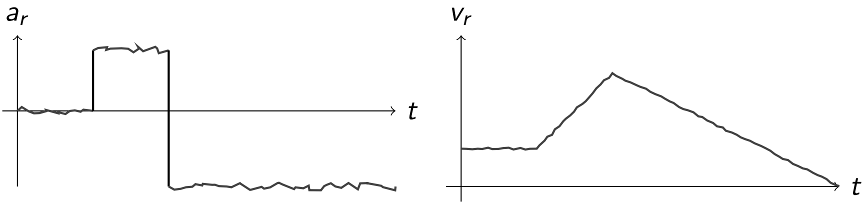
Disturbance

- Acceleration Disturbance
- Steering Disturbance

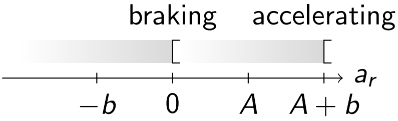
$$dyn \equiv (p_r' = v_r,\ v_r' = a_r,\ \ldots$$

$$\leadsto$$

$$dyn \equiv (p_r' = v_r,\ v_r' \leq a_r + u,\ \ldots$$



$$dyn \equiv p_r' = v_r d_r,\ v_r' \leq a_r + u,\ d' = \omega_r d_r^{\perp},\ \omega_r' = \frac{a_r}{r},\ t' = 1 \ \&\ t \leq \varepsilon \wedge v_r \geq 0$$



Verified Property

Passive Safety:

$$v_r = 0 \vee \|p_r - p_o\| > \frac{v_r^2}{2b} + V\frac{v_r}{b}$$

$$\text{init} \equiv \ldots \wedge 0 < u < b$$

$$\text{SafeCtrl} \equiv \|p_r - p_o\|_\infty > \frac{v_r^2}{2\,(b-u)} + V\frac{v_r}{b-u} + \left(\frac{A+u}{b-u} + 1\right)\left(\frac{A+u}{2}\varepsilon^2 + \varepsilon(v_r + V)\right)$$

Additive Acceleration Disturbance - Interval

Disturbance Interval

Effective acceleration/braking in interval $[-u_l, u_r]$

$$\text{SafeCtrl} \equiv \|p_r - p_o\|_\infty > \frac{v_r^2}{2\,(b - u_r)} + V\frac{v_r}{b - u_r} + \left(\frac{A + u_r}{b - u_r} + 1\right)\left(\frac{A + u_r}{2}\varepsilon^2 + \varepsilon(v_r + V)\right)$$

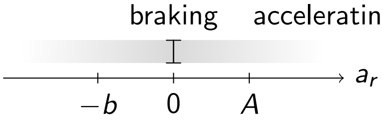
$$dyn \equiv p_r' = v_r d_r,\ a_r - u_l \leq v_r' \leq a_r + u_r,\ d' = \omega_r d_r^\perp,\ \omega_r' = \frac{a_r}{r},\ t' = 1 \ \&\ t \leq \varepsilon \wedge v_r \geq 0$$

Note

Proof for interval same as for previous model (only upper bound matters)

Multiplicative Acceleration Disturbance

$$dyn \equiv p_r' = v_r d_r,\ v_r' \le a_r u,\ d' = \omega_r d_r^{\perp},\ \omega_r' = \frac{a_r}{r},\ t' = 1$$
$$\&\ t \le \varepsilon \wedge v_r \ge 0$$



Verified Property

Passive Safety:

$$v_r = 0 \vee \|p_r - p_o\| > \frac{v_r^2}{2b} + V\frac{v_r}{b}$$

$$\text{init} \equiv \ldots \wedge 0 < u$$

$$\text{SafeCtrl} \equiv \|p_r - p_o\|_\infty > \frac{v_r^2}{2bu} + V\frac{v_r}{bu} + \left(\frac{A}{b} + 1\right)\left(\frac{Au}{2}\varepsilon^2 + \varepsilon(v_r + V)\right)$$

When to use formal verification & validation

When should I use formal V & V? E.g.

- I want to guarantee safety / correctness
- I can imagine that there are reasonable models of the relevant physics
- I want to know how the system behaves in all possible cases
- I do not yet have clear requirements and want to specify them unambiguously
- I want to know under which operating conditions my system will work

When is it challenging to use?

- The physics of the system has no reasonable models.
- Nobody understands any part of the system.

Model Variations and Verification

Obstacle Avoidance

Dynamic Window specifies robot kinematics, decouples safety from optimization $\leadsto$ well suited for hybrid safety verification

Handle complexity

- **Dimension** 1D $\leadsto$ 2D $\leadsto$ Add floor levels
- **Steering** Manhattan $\leadsto$ Differential $\leadsto$ Omnidirectional drive
- **Safety** Static $\leadsto$ Passive $\leadsto$ Passive friendly $\leadsto$ Active
- **Uncertainty** Sensor uncertainty $\leadsto$ Sensor failure $\leadsto$ Actuator disturbance $\leadsto$ Differential inequality models of disturbance
- **Liveness** Cross goal line $\leadsto$ Before deadline $\leadsto$ Cross intersection with obstacles $\leadsto$ Before deadline $\leadsto$ Reach goal $\leadsto$ Before deadline $\leadsto$ In tricky environments $\leadsto$ Escape

Interface & Tools

Model Variations and Verification

Obstacle Avoidance

Dynamic Window specifies robot kinematics, decouples safety from optimization $\rightsquigarrow$ well suited for hybrid safety verification

Handle complexity

- **Dimension** 1D $\rightsquigarrow$ 2D $\rightsquigarrow$ Add floor levels
- **Steering** Manhattan $\rightsquigarrow$ Differential $\rightsquigarrow$ Omnidirectional drive
- **Safety** Static $\rightsquigarrow$ Passive $\rightsquigarrow$ Passive friendly $\rightsquigarrow$ Active
- **Uncertainty** Sensor uncertainty $\rightsquigarrow$ Sensor failure $\rightsquigarrow$ Actuator disturbance $\rightsquigarrow$ Differential inequality models of disturbance
- **Liveness** Cross goal line $\rightsquigarrow$ Before deadline $\rightsquigarrow$ Cross intersection with obstacles $\rightsquigarrow$ Before deadline $\rightsquigarrow$ Reach goal $\rightsquigarrow$ Before deadline $\rightsquigarrow$ In tricky environments $\rightsquigarrow$ Escape

Interface & Tools