

# Verified Train Controllers for the Federal Railroad Administration Train Kinematics Model: Balancing Competing Brake and Track Forces

**Aditi Kabra**

Stefan Mitsch

André Platzer

1

Computer Science Department,  
Carnegie Mellon University

INTERNATIONAL CONFERENCE ON EMBEDDED SOFTWARE 2022





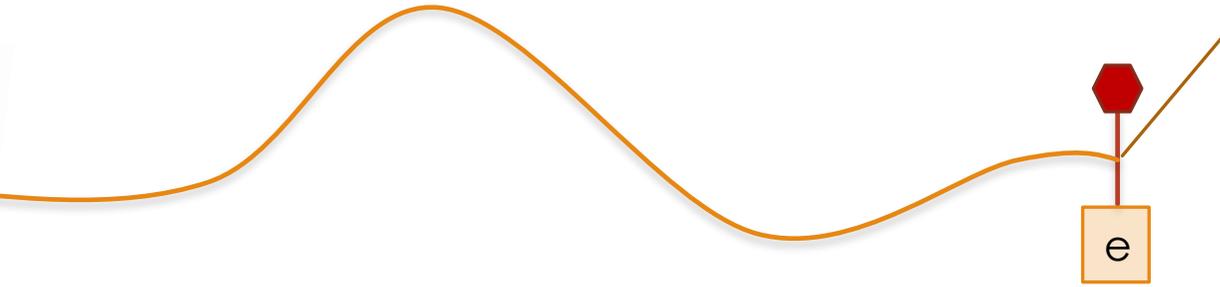
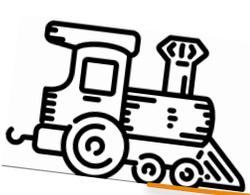


# Train Control: Complicated



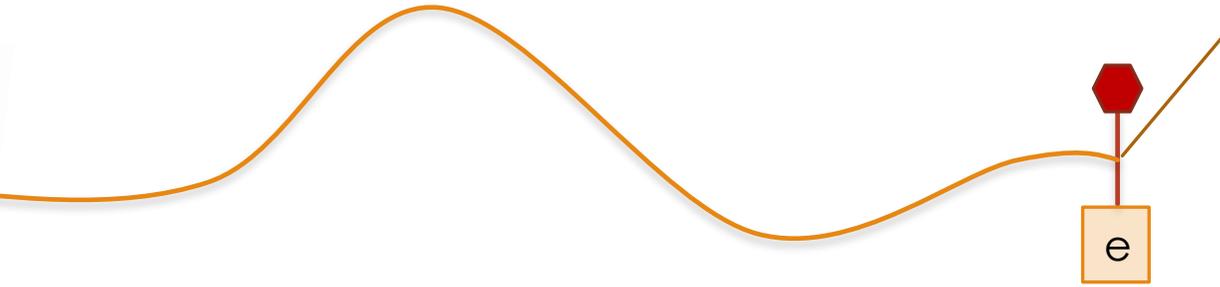
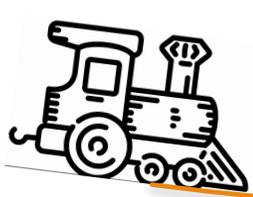
End of *movement authority*: the train must stop by this point

# Train Control: Complicated



End of *movement authority*: the train must stop by this point

# Train Control: Complicated

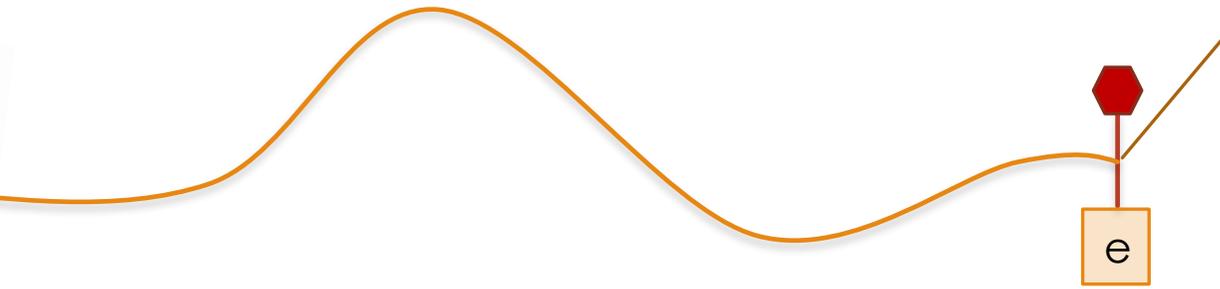


End of *movement authority*: the train must stop by this point

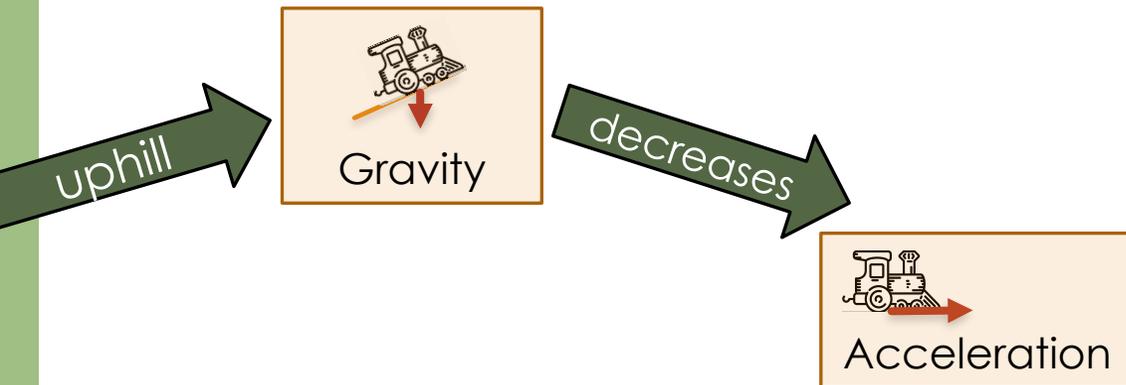


uphill

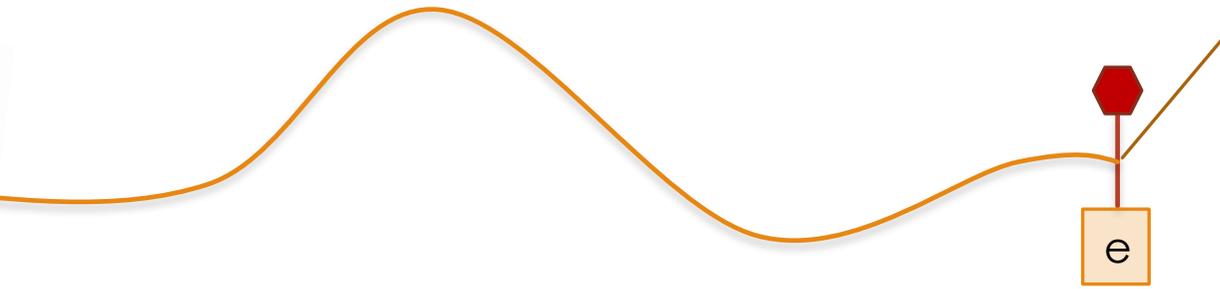
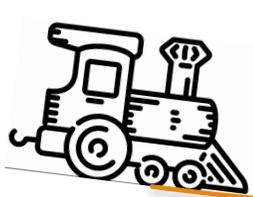
# Train Control: Complicated



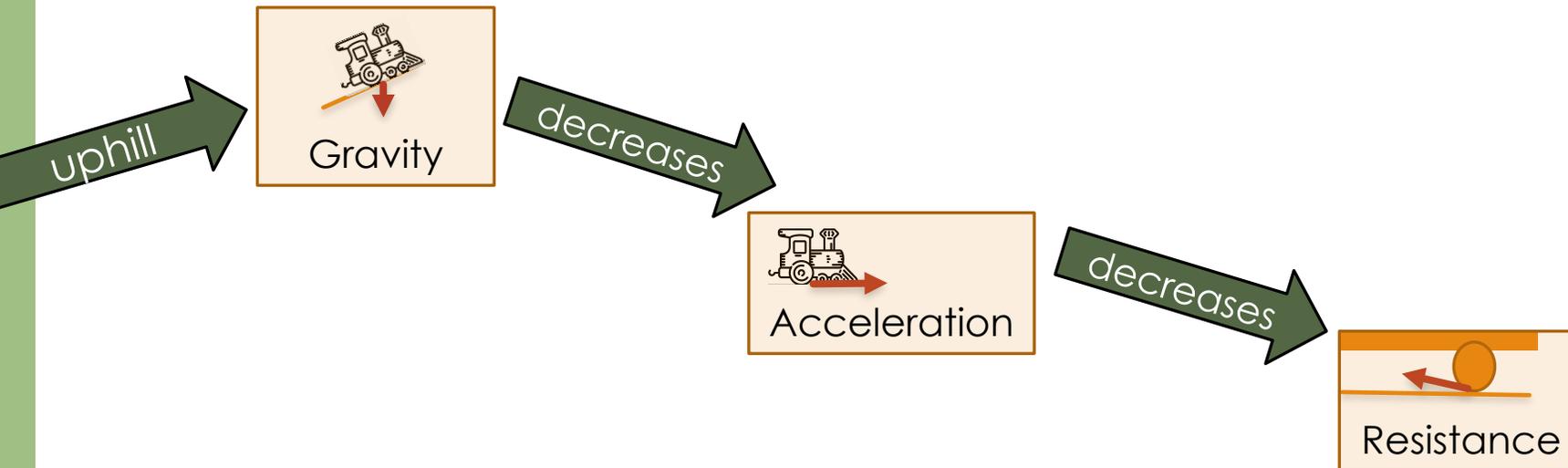
End of *movement authority*: the train must stop by this point



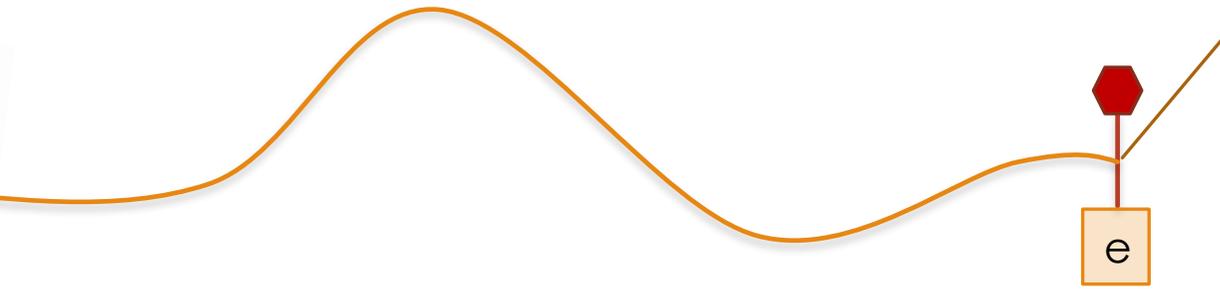
# Train Control: Complicated



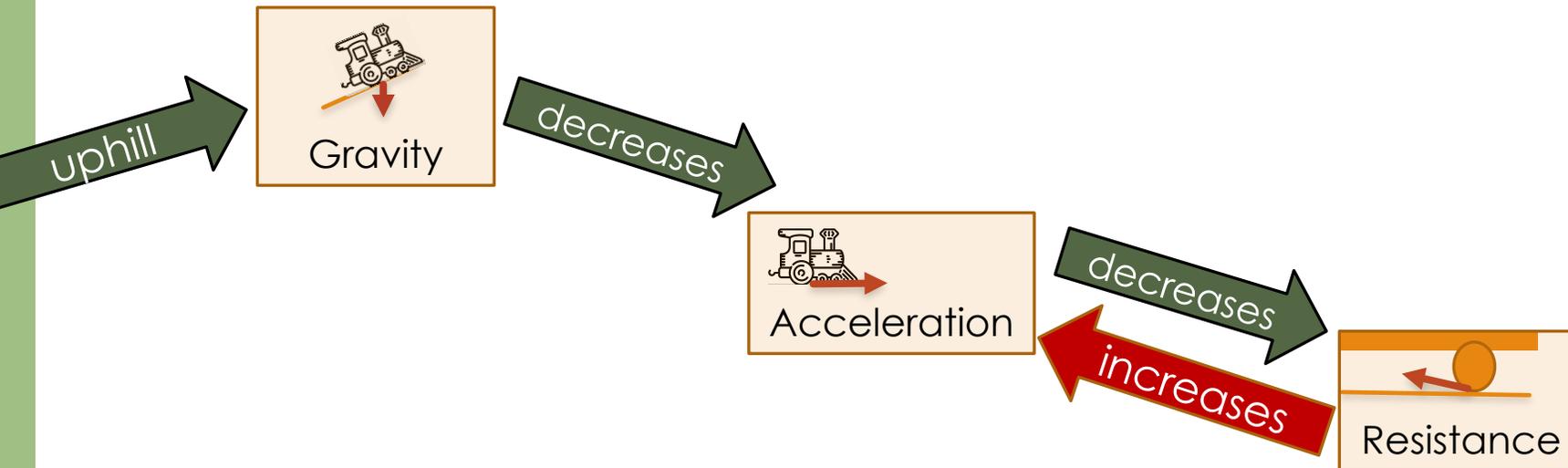
End of *movement authority*: the train must stop by this point



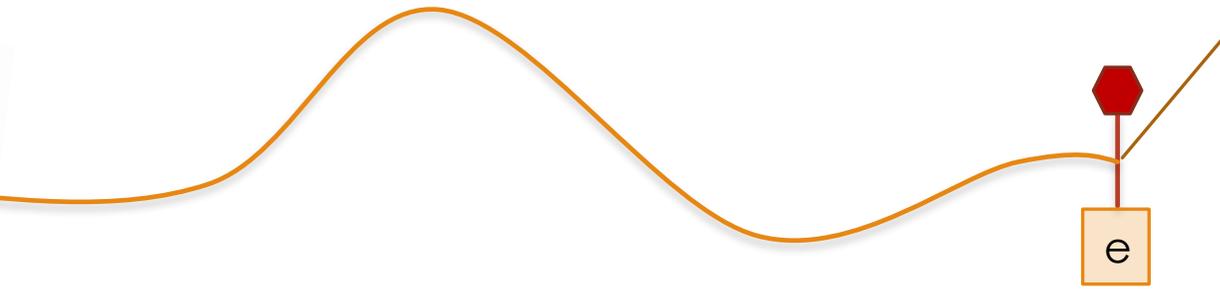
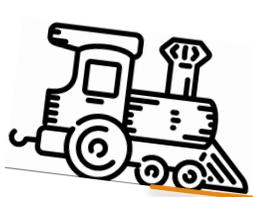
# Train Control: Complicated



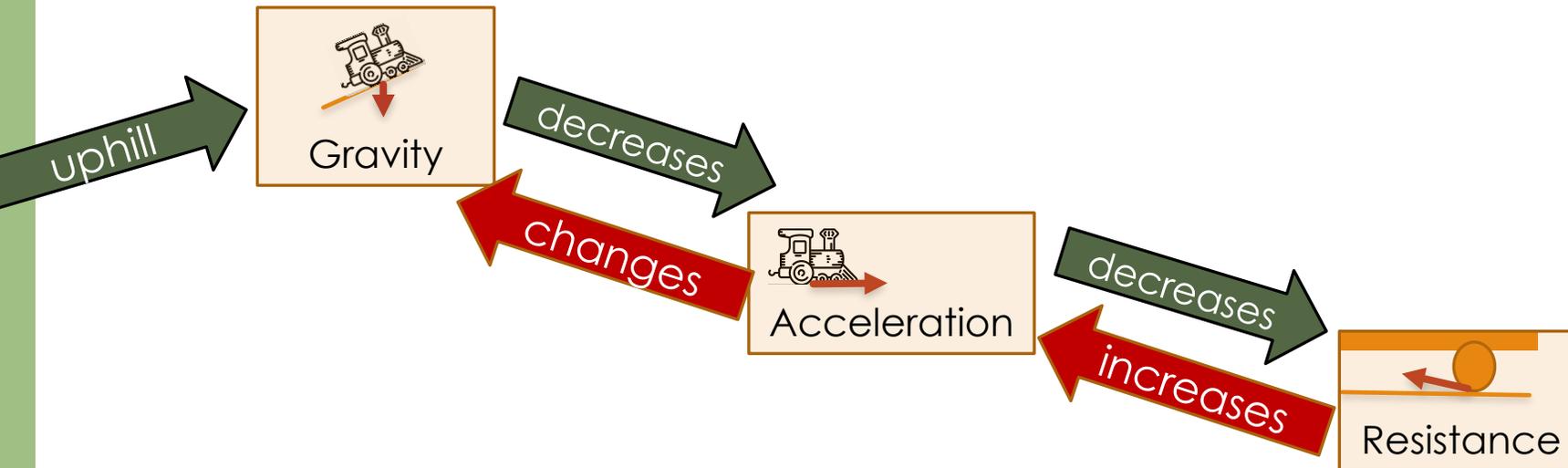
End of *movement authority*: the train must stop by this point



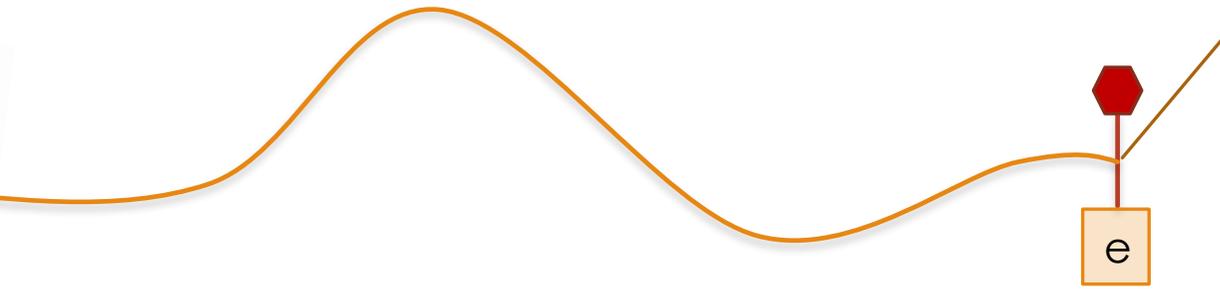
# Train Control: Complicated



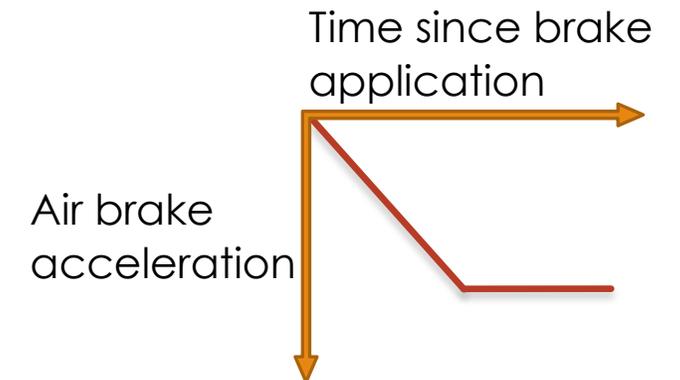
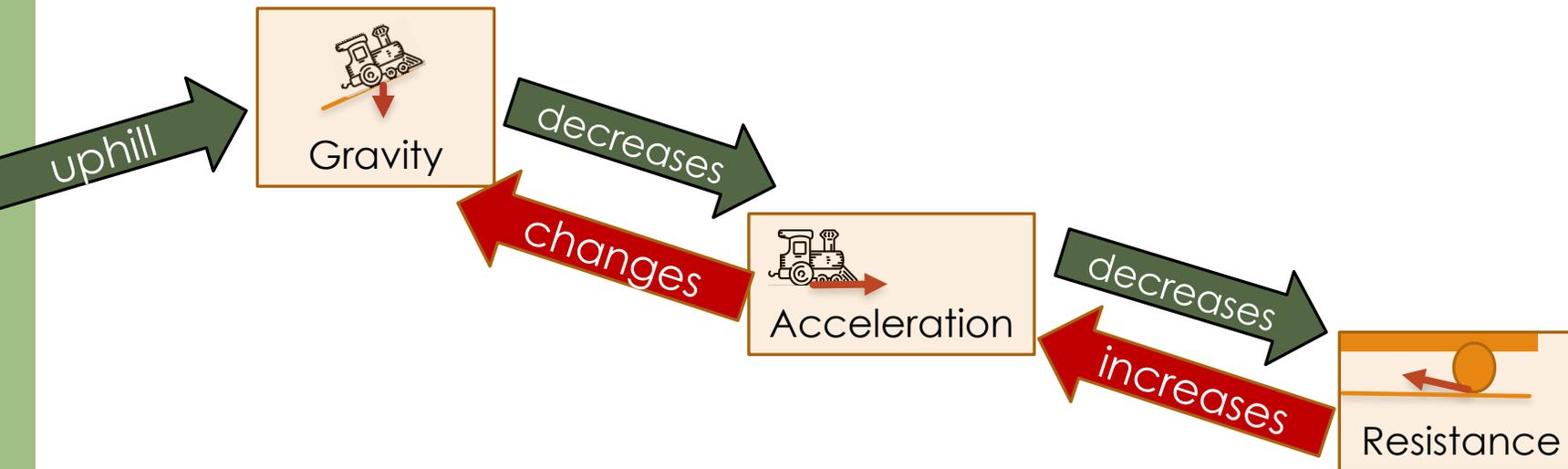
End of *movement authority*: the train must stop by this point



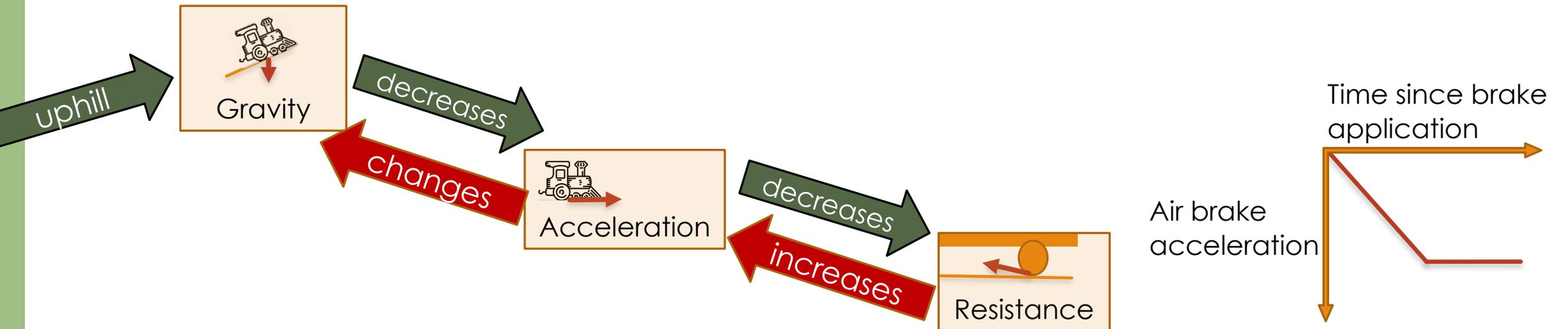
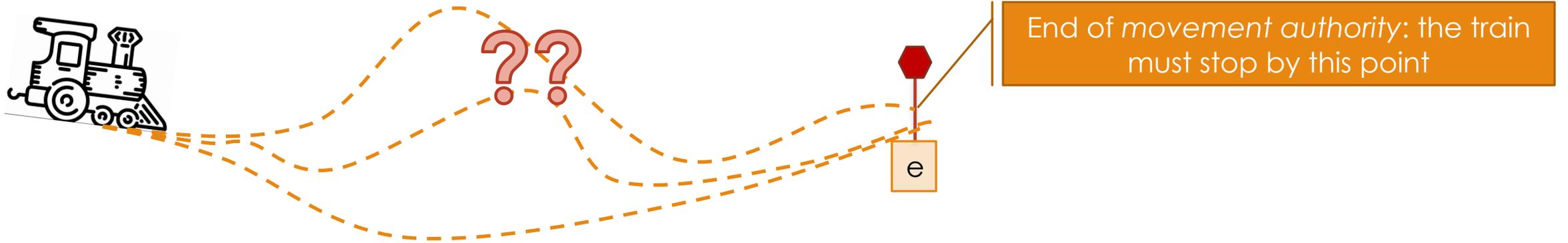
# Train Control: Complicated



End of *movement authority*: the train must stop by this point



# Train Control: Complicated



# Formal Verification



Complete  
FRA Model<sup>[1]</sup>

[1] J. Brosseau and B. M. Ede, "Development of an adaptive predictive braking enforcement algorithm", Federal Railroad Administration, 2009.

# Formal Verification



Complete  
FRA Model [1]

Formal Model

```

1 Theorem - fev_taylor_combine
2
3 Definitions
4 /* Acceleration coefficients. */
5 Real a0; /* Strict upper bound on maximal constant acceleration. */
6 Real a1; /* Accelerations that are linear in velocity. */
7 Real a2; /* Accelerations that are quadratic in velocity. */
8 Real b0; /* Maximal constant braking force (positive). */
9 Real crvDer; /* Coefficient in derivative of horizontal curve. */
10
11 /* Situational track setup */
12 Real maxSlope; /* Greatest allowed acceleration due to slope gradient. a_s in the paper. */
13 Real T; /* Time control loop period / system reaction time. */
14 Real slopeAcc(Real trainPos); /* Slope acceleration map (where trainPos is measured along the sloped track rather than along flat land). a_c in the paper. */
15 Real end; /* End of movement authority. e in the paper. */
16 Real curvature(Real trainPos); /* Acceleration due to curve resistance map (where trainPos is measured along the sloped track rather than along flat land). a_c in the paper. */
17 Real maxVertCur; /* Maximal rate of change of slope (vertical curvature). h_max in the paper. */
18 Real Rmin; /* Bound on friction due to horizontal curve (resistance at min radius). R_c in the paper. */
19 Real Apb; /* maximum penalty brake acceleration */
20 Real pressureChangeRate; /* (Linear) rate of increase in acceleration due to air brakes when train is applying them. */
21
22 /* Upper bound on velocity for current velocity vel at acceleration a0 for one time period independently from particular curve or slope. */
23 Real baseUpperV(Real a0, Real vel) = vel + (a0*maxSlope)*T;
24
25 /* Maximum acceleration due to slope. */
26 Real maxSlopeAcc(Real slopeAcc, Real vel) = min(slopeAcc+(maxVertCur*T)*vel, maxSlope);
27
28 /* Maximum acceleration due to curve (negative since curves decelerate). */
29 Real maxCurveAcc(Real curvature, Real vel) = min((curvature+crvDer*vel*T), 0);
30
31 /* The train will stop in at most this much distance if braking from speed vel. */
32 Real brakingDistance(Real vel, Real buildUp) =
33   (vel - (b0
34     + veleb
35
36 /* The train
37 Real stopping
38
39 /* The train
40 Real stopping
41
42 /* Velocity
43 Real resistar
44
45 /* Total acc
46 Real acc(Real
47
48 /* Upper bound
49 Real upperVel
50   + trainAcc
51   + maxCur

```

Proving in KeYmaera X Theorem Prover

```

[
  t := 0;
  ? end - trainPos > stoppingDistance(vel); trainAcc := *; ? - b0 ≤ trainAcc ∧ trainAcc < a0; brakeSlope := 0; airBrake := 0;
  U
  trainAcc := - b0; brakeSlope := pressureChangeRate;
  { trainPos' = vel, vel' = acc(trainAcc + max(airBrake, Apb), vel, trainPos), airBrake' = brakeSlope, t' = 1 & t ≤ T ∧ vel ≥ 0 }
  *
]
end - trainPos > 0

```

Loop Invariant Loop

loop	$\Gamma \vdash J, \Delta$	$J \vdash P$	$J \vdash [a] J$
		$\Gamma \vdash [a] P, \Delta$	

[\*] iterateb  $\{[a]^* P \leftrightarrow P \wedge [a] \{[a]^* P\}$

Gödel Vacuous gv

2545 lines of proof tactic

[1] J. Brosseau and B. M. Ede, "Development of an adaptive predictive braking enforcement algorithm", Federal Railroad Administration, 2009.

# Formal Verification



Complete FRA Model [1]

**Formal Model**

```

1 Theorem "FRA_validator_combined"
2
3 Definitions
4 /* Acceleration coefficients. */
5 Real a0; /* Strict upper bound on maximal constant acceleration. */
6 Real a1; /* Accelerations that are linear in velocity. */
7 Real a2; /* Accelerations that are quadratic in velocity. */
8 Real b0; /* Maximal constant braking force (positive). */
9 Real crvDer; /* Coefficient in derivative of horizontal curve. */
10
11 /* Situational track setup */
12 Real maxSlope; /* Greatest allowed acceleration due to slope gradient. a_s in the paper. */
13 Real T; /* Time control loop period / system reaction time. */
14 Real slopeAcc(Real trainPos); /* Slope acceleration map (where trainPos is measured along the sloped track rather than along flat land). a_c in the paper. */
15 Real end; /* End of movement authority. e in the paper. */
16 Real curvature(Real trainPos); /* Acceleration due to curve resistance map (where trainPos is measured along the sloped track rather than along flat land). a_c in the paper. */
17 Real maxVertCur; /* Maximal rate of change of slope (vertical curvature). h_max in the paper. */
18 Real Rmin; /* Bound on friction due to horizontal curve (resistance at min radius). R_c in the paper. */
19 Real Apb; /* maximum penalty brake acceleration */
20 Real pressureChangeRate; /* (Linear) rate of increase in acceleration due to air brakes when train is applying them. */
21
22 /* Upper bound on velocity for current velocity vel at acceleration a0 for one time period independently from particular curve or slope. */
23 Real baseUpperV(Real a0, Real vel) = vel + (a0*maxSlope)*T;
24
25 /* Maximum acceleration due to slope. */
26 Real maxSlopeAcc(Real slopeAcc, Real vel) = min(slopeAcc+(maxVertCur*T)*vel, maxSlope);
27
28 /* Maximum acceleration due to curve (negative since curves decelerate). */
29 Real maxCurveAcc(Real curvature, Real vel) = min((curvature+crvDer*vel*T), 0);
30
31 /* The train will stop in at most this much distance if braking from speed vel. */
32 Real brakingDistance(Real vel, Real b0) =
33   (vel - b0) * T +
34   (vel - b0) * T * T * (b0 - a0) / (2 * T);
35
36 /* The train
37 Real stopping
38
39 /* The train
40 Real stopping
41
42 /* Velocity
43 Real resist
44
45 /* Total acc
46 Real acc(Real
47
48 /* Upper bound
49 Real upperVel
50   * trainAcc
51   + maxCur

```

**Formal Model**

**Proving in KeYmaera X Theorem Prover**

```

[
  t := 0;
  ? end - trainPos > stoppingDistance(vel); trainAcc := *; ? - b0 ≤ trainAcc ∧ trainAcc < a0; brakeSlope := 0; airBrake := 0;
  loop
    trainAcc := - b0; brakeSlope := pressureChangeRate;
  }
  { trainPos = vel, vel = acc(trainAcc + max(airBrake, Apb), vel, trainPos), airBrake = brakeSlope, t = 1 & t ≤ T ∧ vel ≥ 0 }
]
end - trainPos > 0

```

**Loop Invariant Loop**

loop	$\Gamma \vdash J, \Delta$	$J \vdash P$	$J \vdash [a] J$
		$\Gamma \vdash [a] P, \Delta$	

**[\*] iterateb**  $\{ [a]^* P \leftrightarrow P \wedge [a] \{ [a]^* P \}$

**Gödel Vacuous gv**

2545 lines of proof tactic

**Proof: ✓ All goals closed**

```

Provable( ==> end()-trainPos>(vel-min<< .1 < .2&
reChangeRate()*vel^(1/2)))/pressureChangeRate()*
& .0=-.1|.1>=0&.0=.1 >>(((b0()-maxSlope()-0)^2-2*
essureChangeRate(),(b0()-maxSlope()-0-abs<< .1 < 0&
0=.1|.1>=.2&.0=.2 >>((Apb()-0)/pressureChangeRa
6*pressureChangeRate()*min<< .1 < .2&.0=.1|.1>=
( )*vel^(1/2)))/pressureChangeRate())^3&(a0()>0&b0()>
ainPos (Rmin()<=curvature(trainPos)&curvature(trainPo
((slopeAcc(x) )<=maxVertCur()*x'&-(slopeAcc(x) )<=max
Pos>(vel+(a0()+min<< .1 < .2&.0=.1|.1>=.2&.0=.
Der)*(vel+(a0()+maxSlope())*T())*T(),0))*T()-min<<
2-2*pressureChangeRate()* (vel+(a0()+min<< .1 < .2&
rvature(trainPos)+crvDer)*(vel+(a0()+maxSlope())*T(
sureChangeRate(),(b0()-maxSlope()-0-abs<< .1 < 0&.
rtCur()*T())*(vel+(a0()+maxSlope())*T()),maxSlope()+
(2*

```

Infinitely many possibilities checked once and for all

[1] J. Brosseau and B. M. Ede, "Development of an adaptive predictive braking enforcement algorithm", Federal Railroad Administration, 2009.

# Formal Verification



Complete FRA Model [1]

```

1 Theorem -rev_taylor_bound
2
3 Definitions
4   /* Acceleration coefficients. */
5   Real a0; /* Strict upper bound on maximal constant acceleration. */
6   Real a1; /* Accelerations that are linear in velocity. */
7   Real a2; /* Accelerations that are quadratic in velocity. */
8   Real b0; /* Maximal constant braking force (positive). */
9   Real crvDer; /* Coefficient in derivative of horizontal curve. */
10
11 /* Situational track setup */
12 Real maxSlope; /* Greatest allowed acceleration due to slope gradient. a_s in the paper. */
13 Real T; /* Time control loop period / system reaction time. */
14 Real slopeAcc(Real trainPos); /* Slope acceleration map (where trainPos is measured along the sloped track rather than along flat land). a_c in the paper. */
15 Real end; /* End of movement authority. e in the paper. */
16 Real curvatur(Real trainPos); /* Acceleration due to curve resistance map (where trainPos is measured along the sloped track rather than along flat land). a_c in the paper. */
17 Real maxVertCur; /* Maximal rate of change of slope (vertical curvature). h_max in the paper. */
18 Real Rmin; /* Bound on friction due to horizontal curve (resistance at min radius). R_C in the paper. */
19 Real Apb; /* maximum penalty brake acceleration */
20 Real pressureChangeRate; /* (Linear) rate of increase in acceleration due to air brakes when train is applying them. */
21
22 /* Upper bound on velocity for current velocity vel at acceleration a0 for one time period independently from particular curve or slope. */
23 Real baseUpperV(Real a0, Real vel) = vel + (a0*maxSlope)*T;
24
25 /* Maximum acceleration due to slope. */
26 Real maxSlopeAcc(Real slopeAcc, Real vel) = min(slopeAcc+(maxVertCur*T)*vel, maxSlope);
27
28 /* Maximum acceleration due to curve (negative since curves decelerate). */
29 Real maxCurveAcc(Real curvature, Real vel) = min((curvature+crvDer*vel*T), 0);
30
31 /* The train will stop in at most this much distance if braking from speed vel. */
32 Real brakingDistance(Real vel, Real buildUp) =
33   (vel - (b0 +
34     + veleb
35
36 /* The train
37 Real stopping
38
39 /* The train
40 Real stopping
41
42 /* Velocity
43 Real resist
44
45 /* Total acc
46 Real acc(Real
47
48 /* Upper bound
49 Real upperVel
50   + trainAcc
51   + maxCur

```

Formal Model

Proving in KeYmaera X Theorem Prover

```

[
  t := 0;
  ? end - trainPos > stoppingDistance(vel); trainAcc := ? - b0 ≤ trainAcc ∧ trainAcc < a0; brakeSlope := 0; airBrake := 0;
  u
  trainAcc := - b0; brakeSlope := pressureChangeRate;
  { trainPos' = vel, vel' = acc(trainAcc + max(airBrake, Apb), vel) }
]
end - trainPos > 0

```

Loop Invariant Loop

loop  $\Gamma \vdash J, \Delta$

[\*] iterateb  $\{[a;] P \leftrightarrow P \wedge [a;] \{[a;] P\}$

Gödel Vacuous cv

2545 lines of proof tactic

Proof: ✓ All goals closed

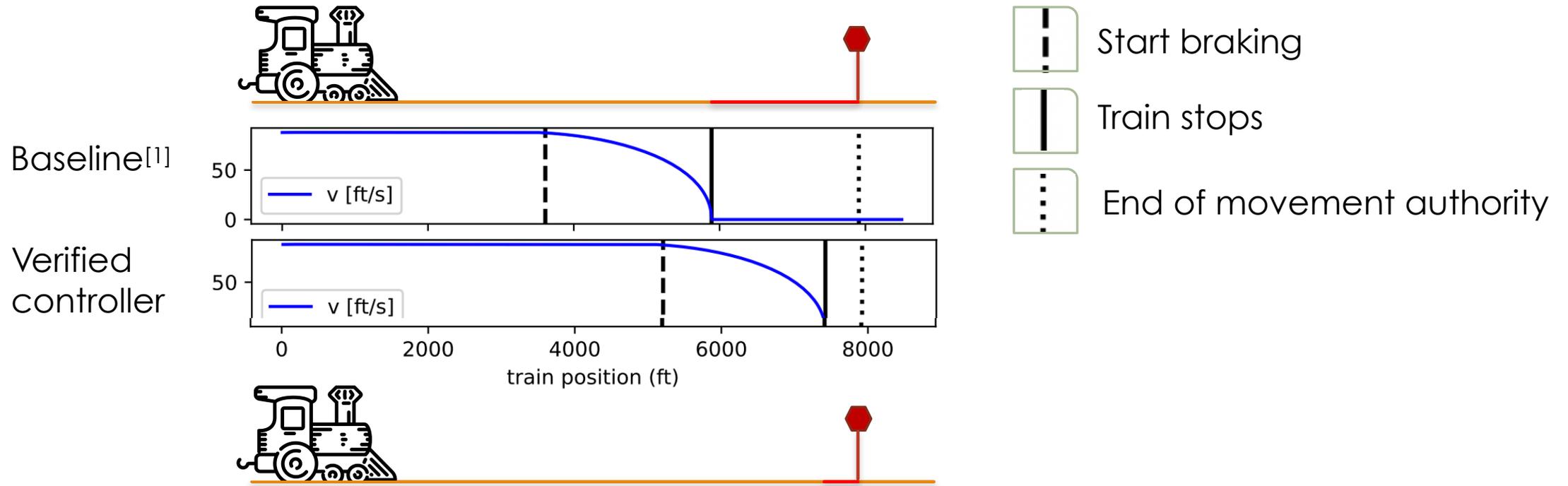
Provable(  $\implies$  end()-trainPos>(vel-min<< .1 < .2&. reChangeRate()\*vel^(1/2)))/pressureChangeRate()\* (b0 & .0=-.1|.1>=0&.0=.1 >>(((b0()-maxSlope()-0)^2-2\* essureChangeRate(),(b0()-maxSlope()-0-abs<< .1 < 0&. 0=.1|.1>=.2&.0=.2 >>((Apb()-0)/pressureChangeRa 6\*pressureChangeRate()\*min<< .1 < .2&.0=.1|.1>= (.)\*vel^(1/2)))/pressureChangeRate())^3&(a0(>0&b0(> ainPos (Rmin()<=curvature(trainPos)&curvature(trainPo ((slopeAcc(x))'<=maxVertCur()\*x'&-(slopeAcc(x))'<=max Pos>(vel+(a0()+min<< .1 < .2&.0=.1|.1>=.2&.0=. Der)\*(vel+(a0()+maxSlope()\*T))\*T(),0))\*T()-min<< . 2-2\*pressureChangeRate()\* (vel+(a0()+min<< .1 < .2&. rvature(trainPos)+crvDer)\*(vel+(a0()+maxSlope()\*T())



Generalizable

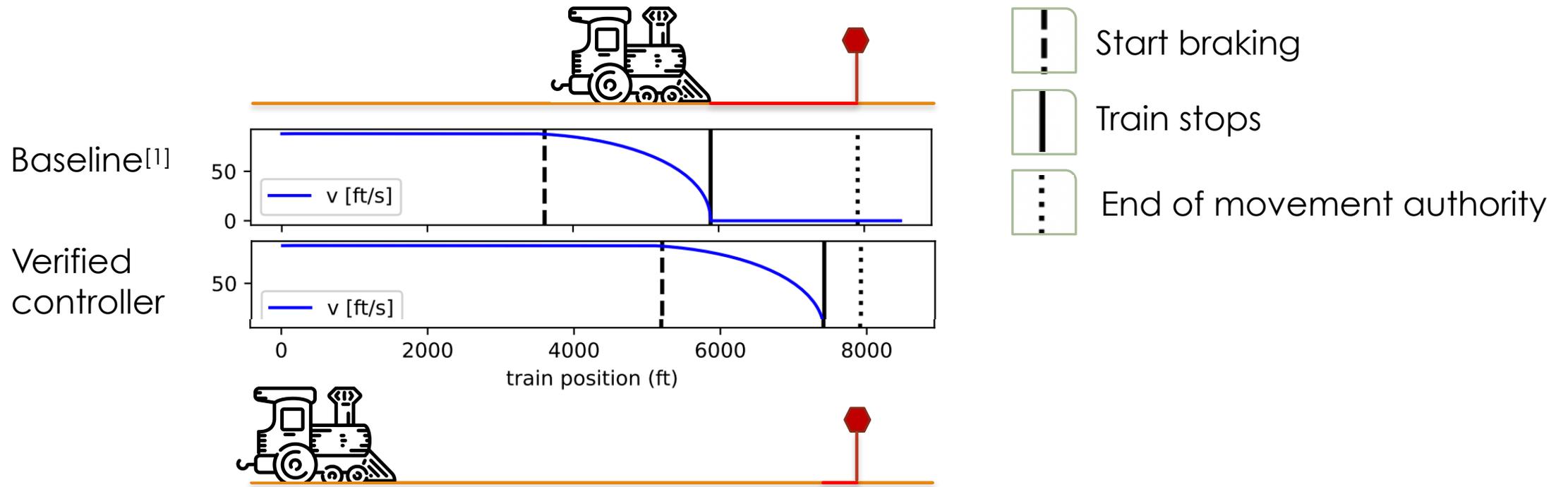
[1] J. Brosseau and B. M. Ede, "Development of an adaptive prediction Administration, 2009.

# Approach: Impact



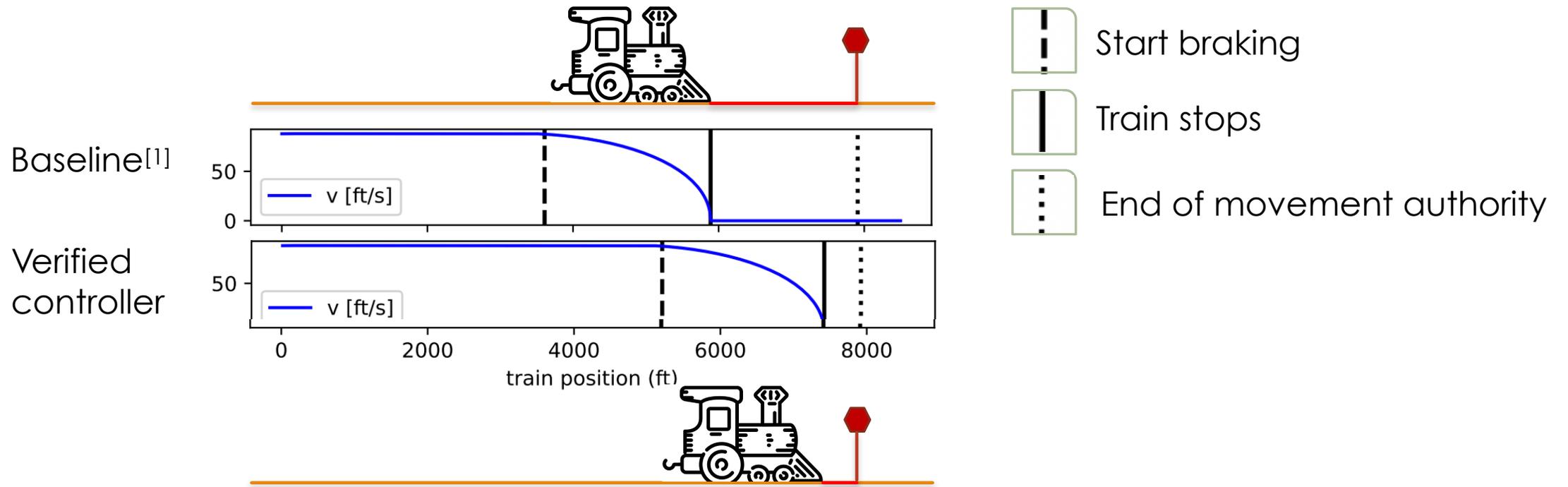
[1] J. Brosseau and B. M. Ede, "Development of an adaptive predictive braking enforcement algorithm", Federal Railroad Administration, 2009.

# Approach: Impact



[1] J. Brosseau and B. M. Ede, "Development of an adaptive predictive braking enforcement algorithm", Federal Railroad Administration, 2009.

# Approach: Impact



[1] J. Brosseau and B. M. Ede, "Development of an adaptive predictive braking enforcement algorithm", Federal Railroad Administration, 2009.

# Overview

- Introduction
- Techniques
- Evaluation
- Summary

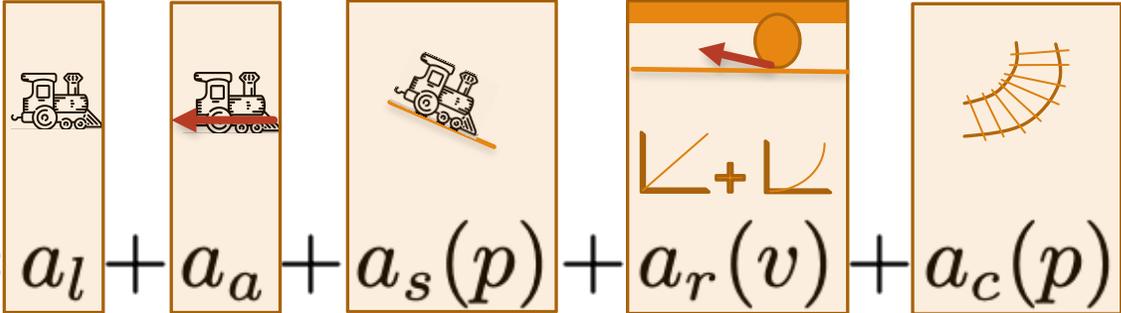


# Background: Dynamics

$$p' = v, v' = a_l + a_a + a_s(p) + a_r(v) + a_c(p), a'_b = m_b$$

with  $a_l \in [-b_{\max}, a_{\max}]$ ,  $a_a = \max(a_b, a_{b\max})$

# Background: Dynamics

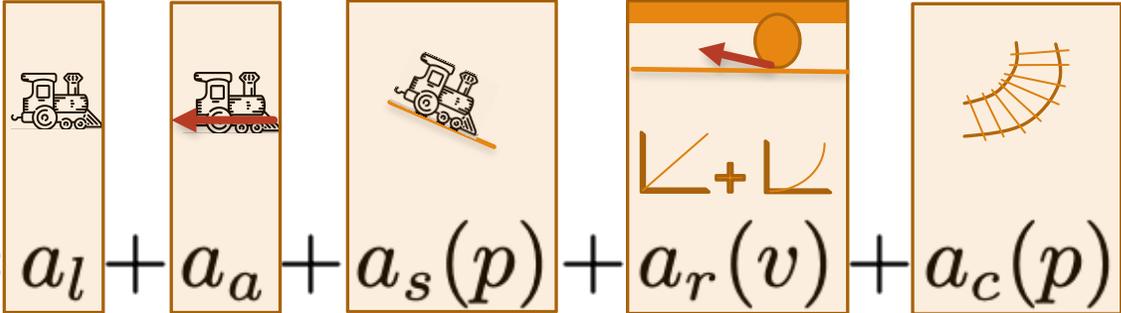
$$p' = v, v' = a_l + a_a + a_s(p) + a_r(v) + a_c(p), a'_b = m_b$$


with  $a_l \in [-b_{\max}, a_{\max}]$ ,  $a_a = \max(a_b, a_{b\max})$

Rate of change of train  
position is velocity

# Background: Dynamics

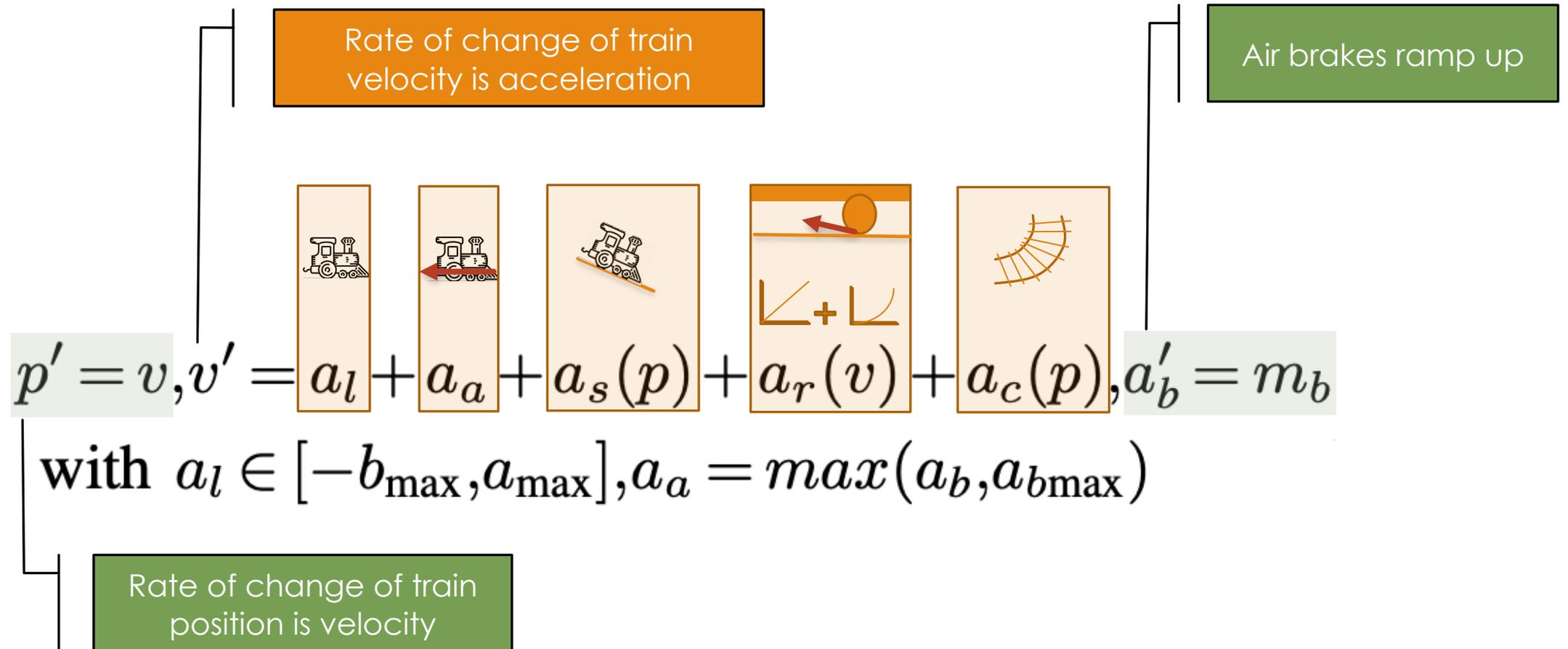
Rate of change of train velocity is acceleration

$$p' = v, v' = a_l + a_a + a_s(p) + a_r(v) + a_c(p), a'_b = m_b$$


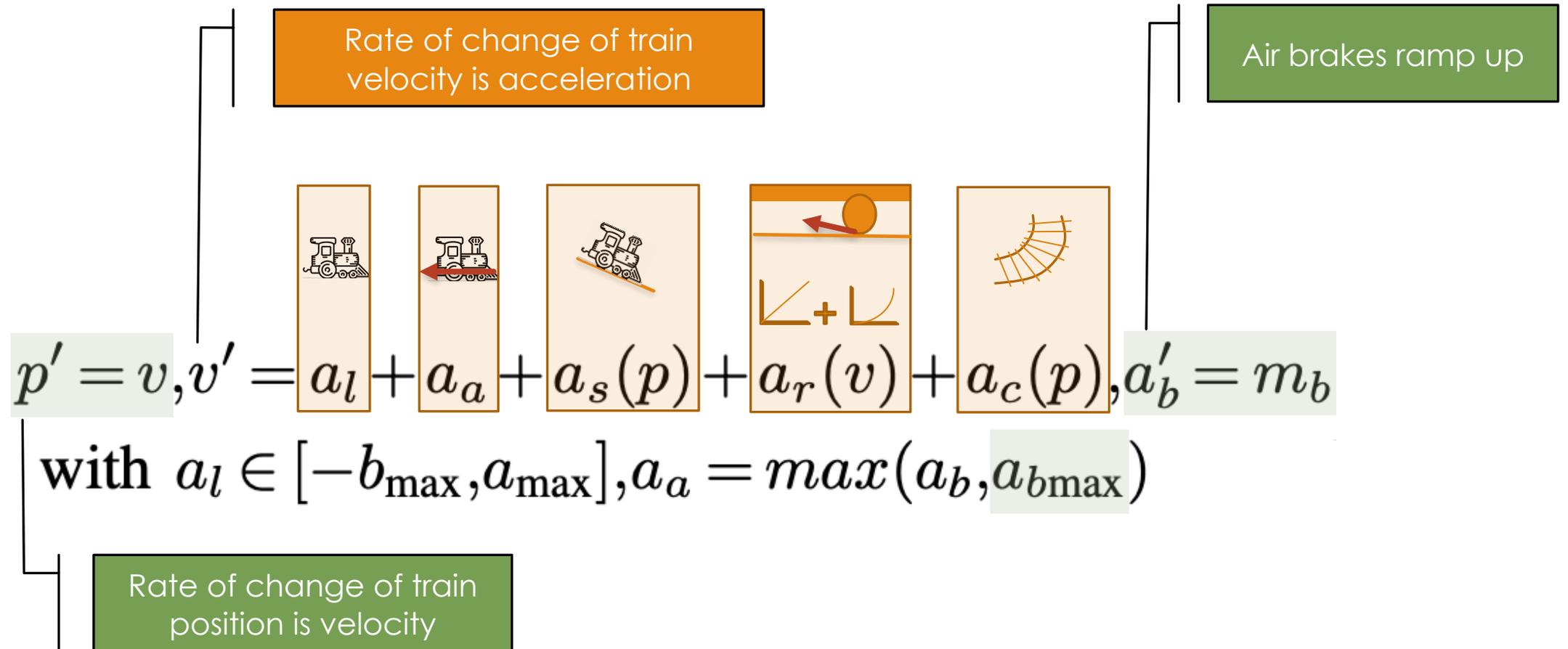
with  $a_l \in [-b_{\max}, a_{\max}]$ ,  $a_a = \max(a_b, a_{b\max})$

Rate of change of train position is velocity

# Background: Dynamics



# Background: Dynamics



# Unknown functions: slope, curve



$$p' = v, v' = a_l + a_a + a_s(p) + a_r(v) + a_c(p), a'_b = m_b$$

# Unknown functions: slope, curve



$$p' = v, v' = a_l + a_a + a_s(p) + a_r(v) + a_c(p), a'_b = m_b$$

# Unknown functions: slope, curve



$$p' = v, v' = a_l + a_a + a_s(p) + a_r(v) + a_c(p), a'_b = m_b$$

# Unknown functions: slope, curve



Use worst case value ...

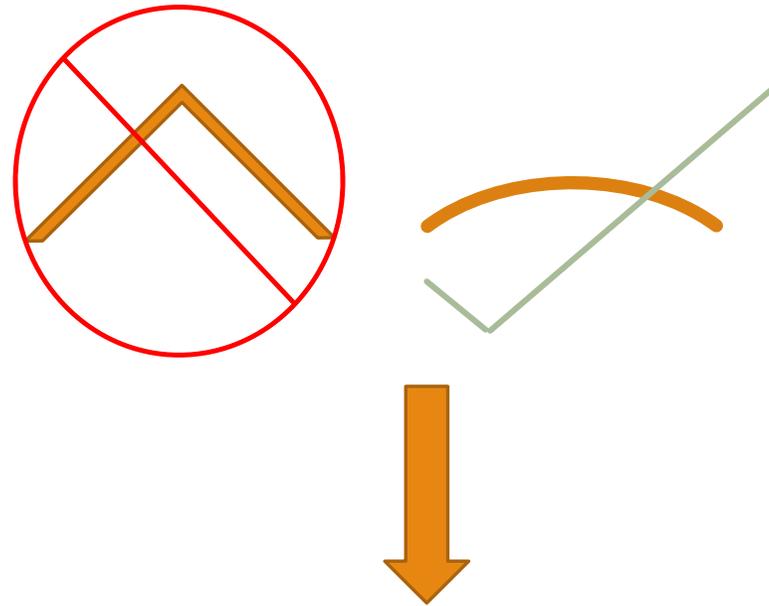
$$p' = v, v' = a_l + a_a + \overset{m_s}{a_s(p)} + a_r(v) + \overset{0}{a_z(p)}, a'_b = m_b$$

Unknown function: replace  
with worst case value  $m_s$

Unknown function: replace with  
worst case value 0

# Unknown functions: slope, curve

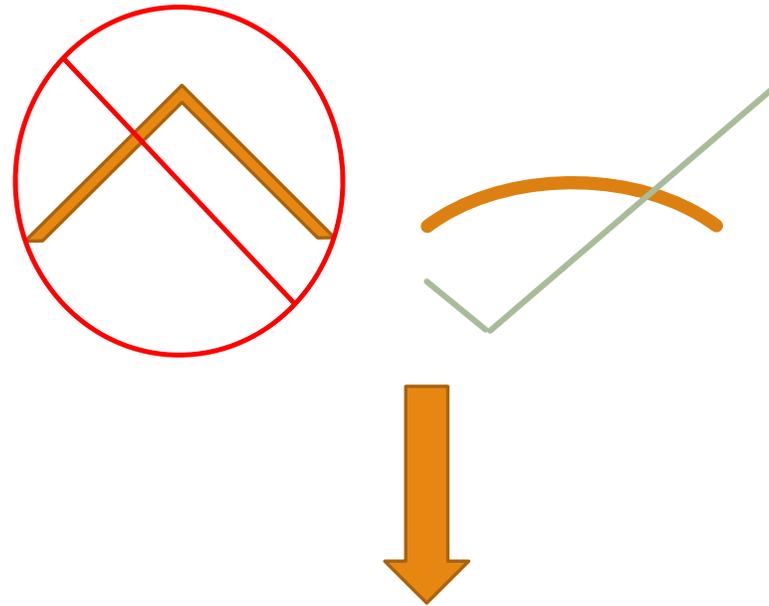
... with improving estimates.



$$a_s(p) \leq \bar{a}_s(p_0) = \min(m_s, a_s(p_0) + u \cdot h_{\max} \cdot T)$$

# Unknown functions: slope, curve

... with improving estimates.

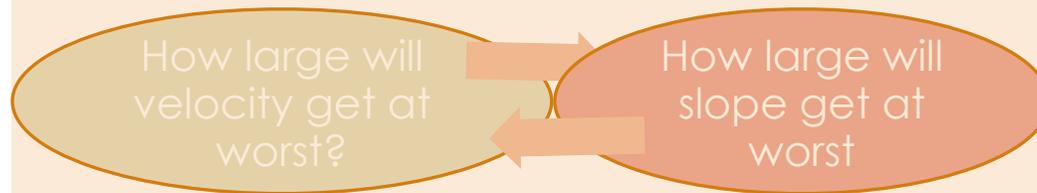


$$a_s(p) \leq \bar{a}_s(p_0) = \min(m_s, a_s(p_0) + u \cdot h_{\max} \cdot T)$$

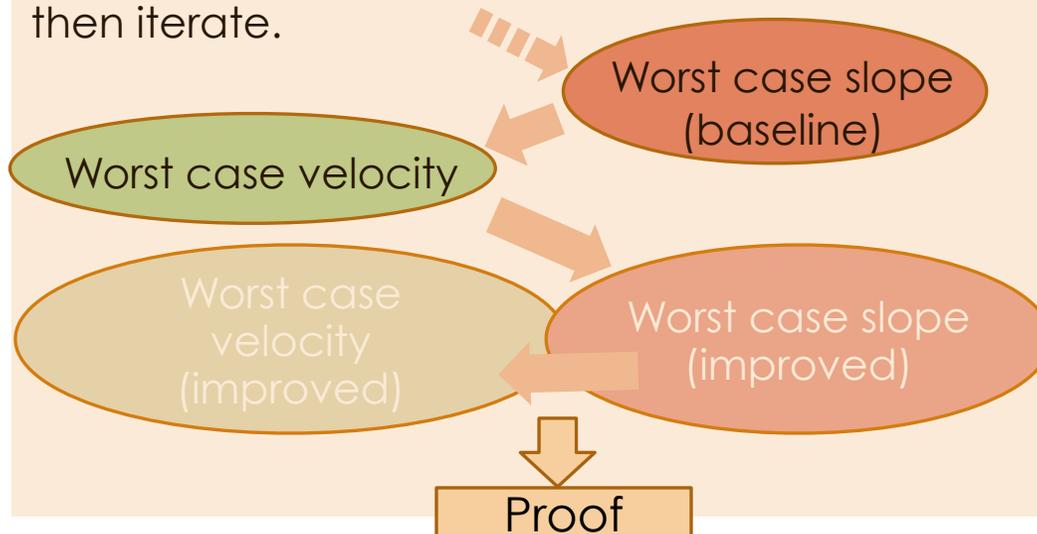
# Other Techniques

## Circular Dependencies

**Problem:** Circular dependence while estimating worst case values.



**Solution:** Bootstrap cycle with naive values, then iterate.



## Taylor Polynomial

**Problem:** Davis resistance integrates poorly.

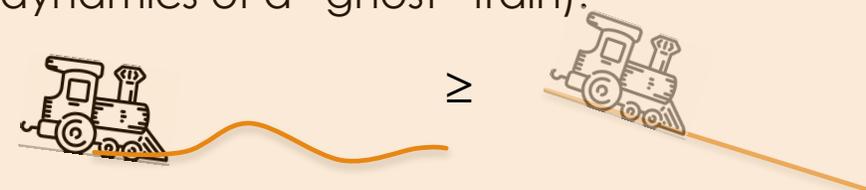
$$\frac{\left( \sqrt{4(a_l + m_s)a_2 - a_1^2} \right) \cdot \tan \left( t \frac{\sqrt{4(a_l + m_s)a_2 - a_1^2}}{2} + \tan^{-1} \left( \frac{a_1 + 2a_2 v_0}{\sqrt{4(a_l + m_s)a_2 - a_1^2}} \right) \right) - a_1}{2a_2}$$

**Solution:** Taylor polynomial approximation.

## Ghost Trains

**Problem:** Intermediate reasoning steps transcendental.

**Solution:** Reason about as ODE (here represents dynamics of a "ghost" train).

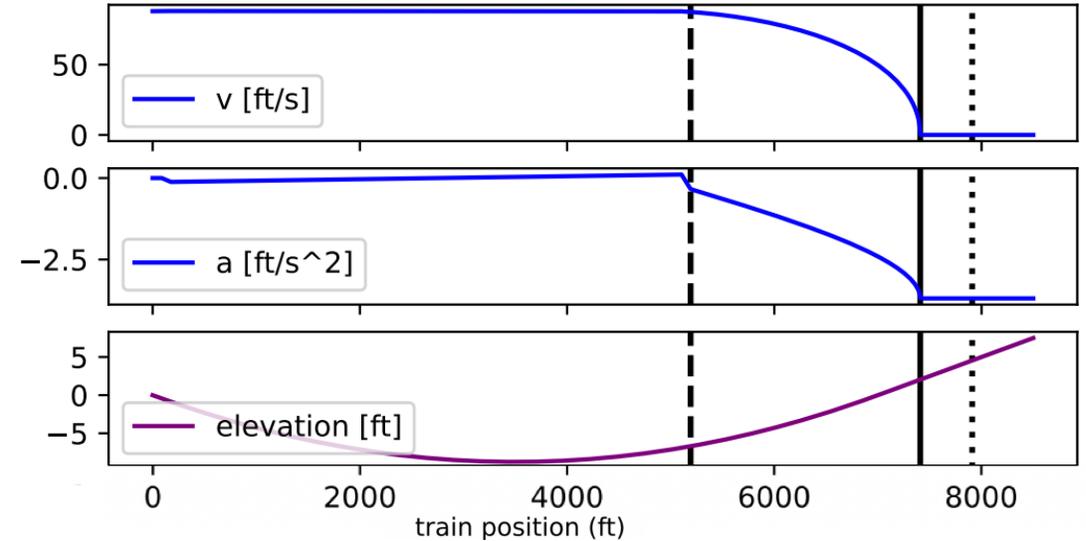
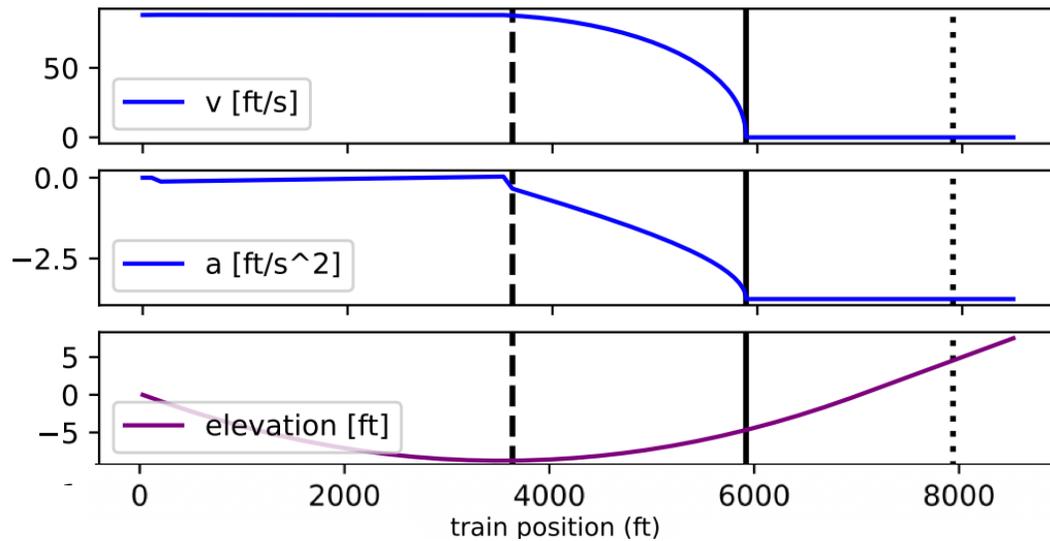


# Overview

- ▶ Introduction
- ▶ Techniques
- ▶ Evaluation
- ▶ Summary



# Limiting Undershoot while Maintaining Safety



Start braking

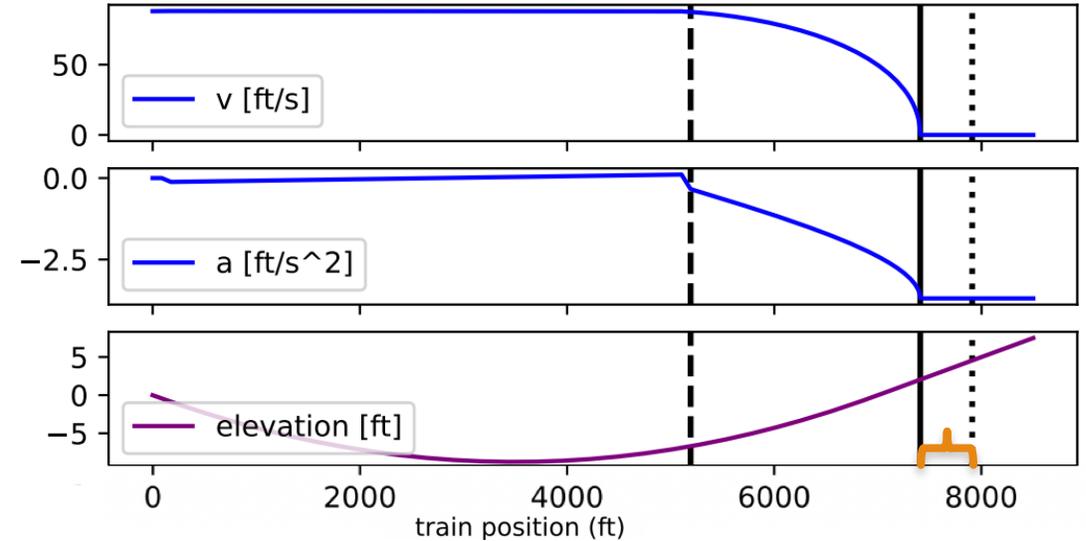
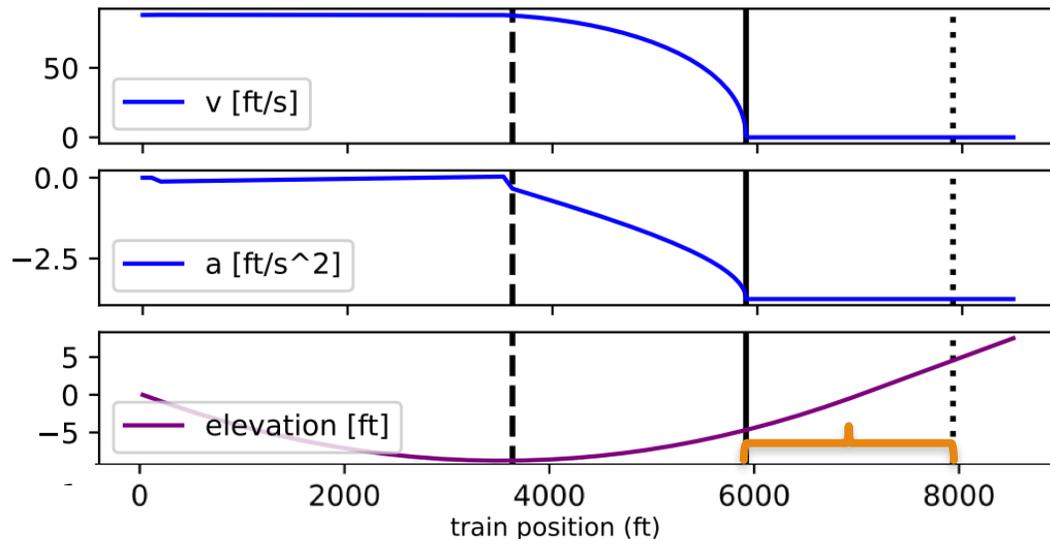


End of movement authority



Train stops

# Limiting Undershoot while Maintaining Safety



Start braking

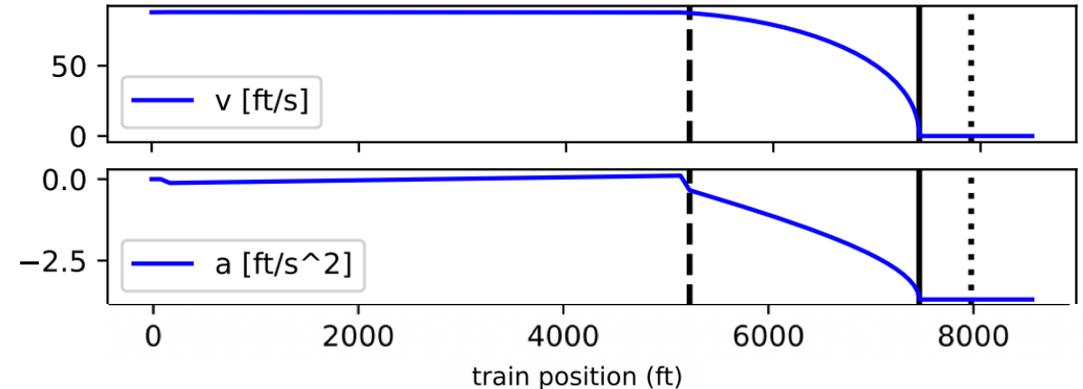
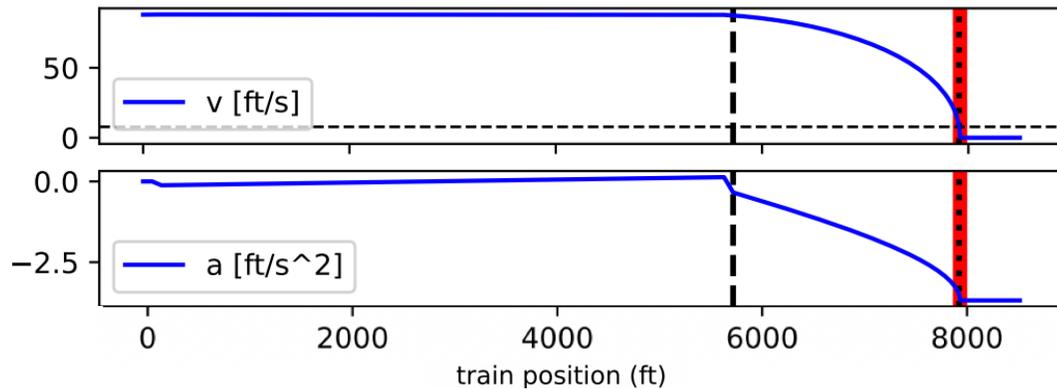


End of movement authority



Train stops

# Limiting Undershoot while Maintaining Safety



Start braking

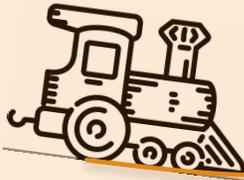


End of movement authority



Train stops

# Summary



Verified controller for full FRA model dynamics. KeYmaera X proofs available online

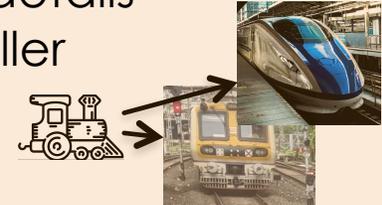
## Generalizable Techniques

- Dealing with unknown functions
- Circular dependencies
- Taylor polynomials
- Ghost dynamics



## Verified Model Generalizability

- Abstraction of physical details
- Nondeterministic controller



## Experiments

Controller limits undershoot while maintaining safety

