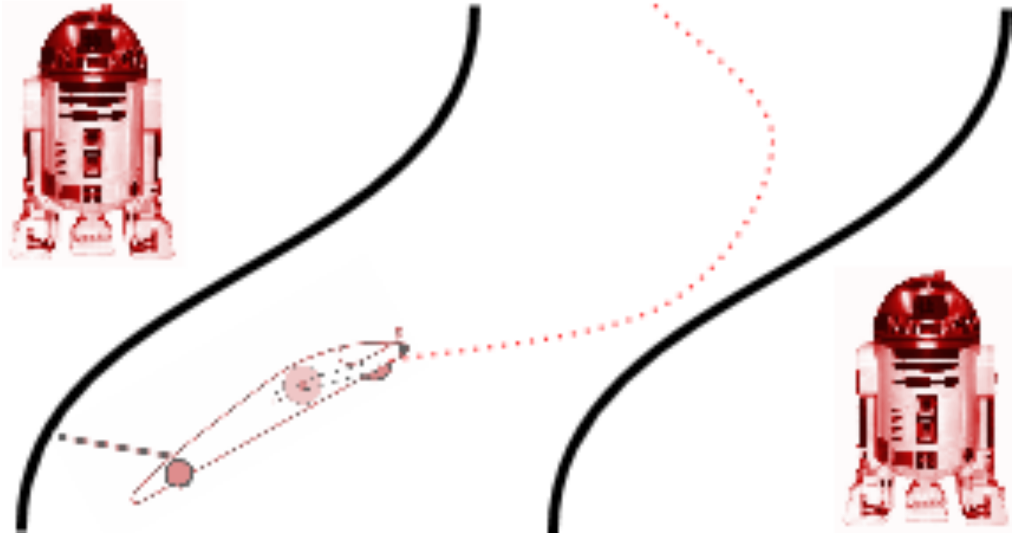
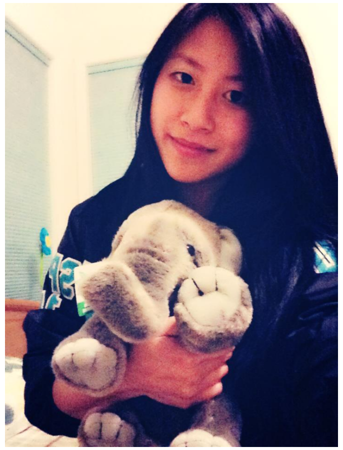
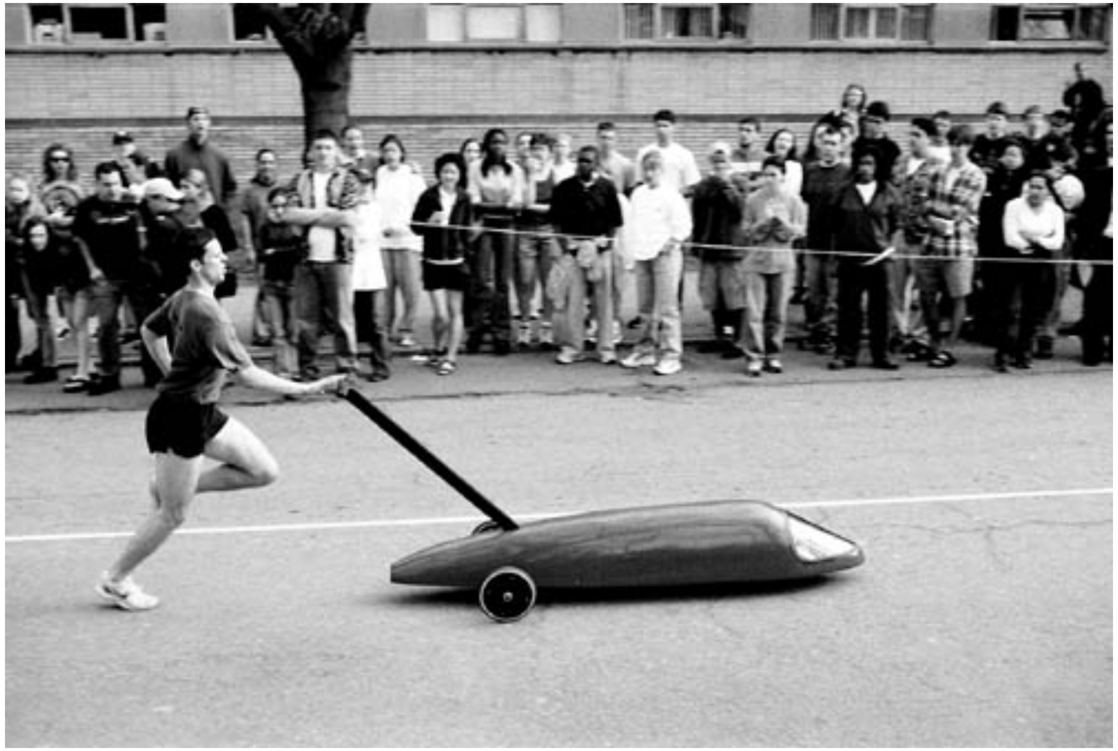


R2-D2 Goes to Buggy



Emily Yeh & Anastassia Kornilova

BUGGY



Vehicle Safety in the Real World

R2-D2 enters the races

R2D2 Goes to Buggy by Anastassia Kornilova & Emily Yeh

Basics of the Model

Track: Helix with fixed width and varying parameters

Parameters:

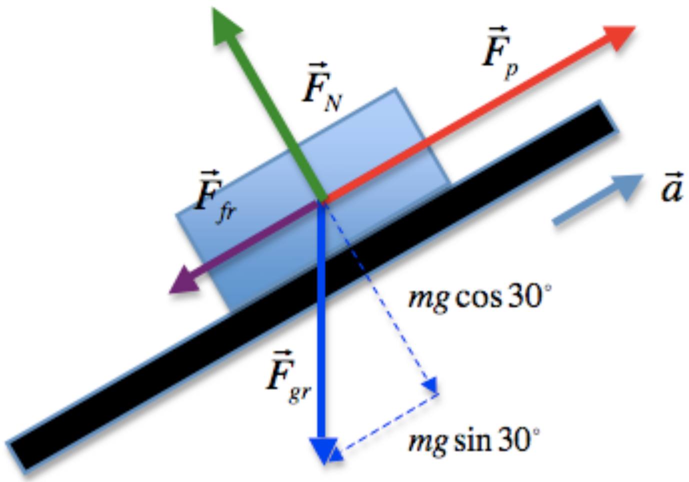
Model 1. Varying helix radii

Model 2. Varying helix slope

Buggy Control (with R2D2): acceleration

Buggy Evolution: circular motion

Physics: Circles and Inclines



$$\sum F_x = F_p - F_{fr} - mg\sin 30^\circ = ma$$

$$\sum F_y = F_N - mg\cos 30^\circ = 0$$

$$F_{fr} = \mu F_N$$

$$F_N = mg\cos 30^\circ$$

Figure from: http://physatwes.com/SecondLawHonors.aspx

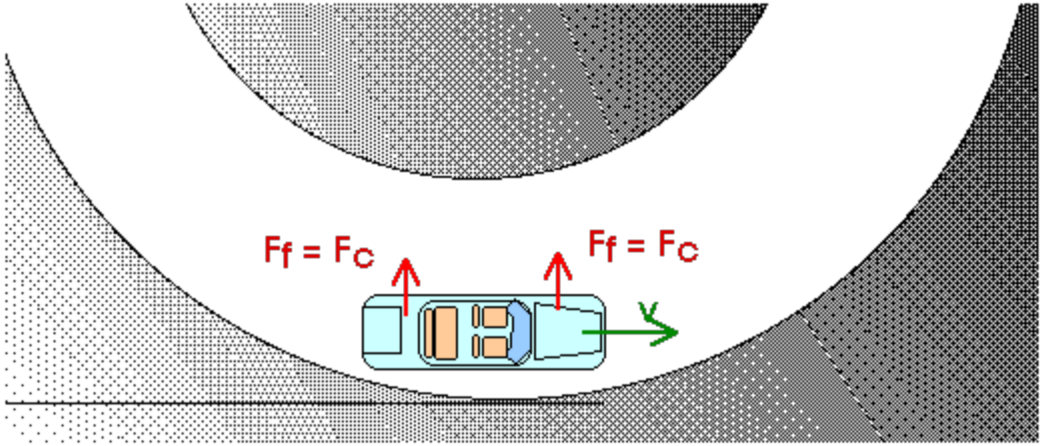
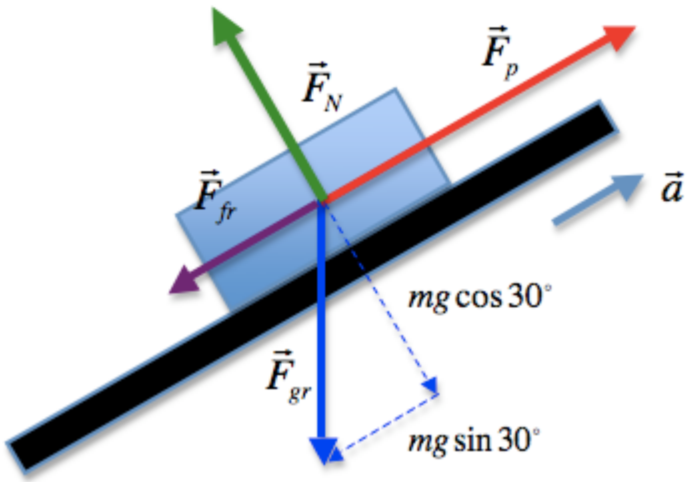


Figure form: http://www.ux1.eiu.edu/~cfadd/1150/05UCMGrav/Curve.html

Physics: Circles and Inclines



$$\sum F_x = F_p - F_{fr} - mg\sin 30^\circ = ma$$

$$\sum F_y = F_N - mg\cos 30^\circ = 0$$

$$F_{fr} = \mu F_N$$

$$F_N = mg\cos 30^\circ$$

Figure from: http://physatwes.com/SecondLawHonors.aspx

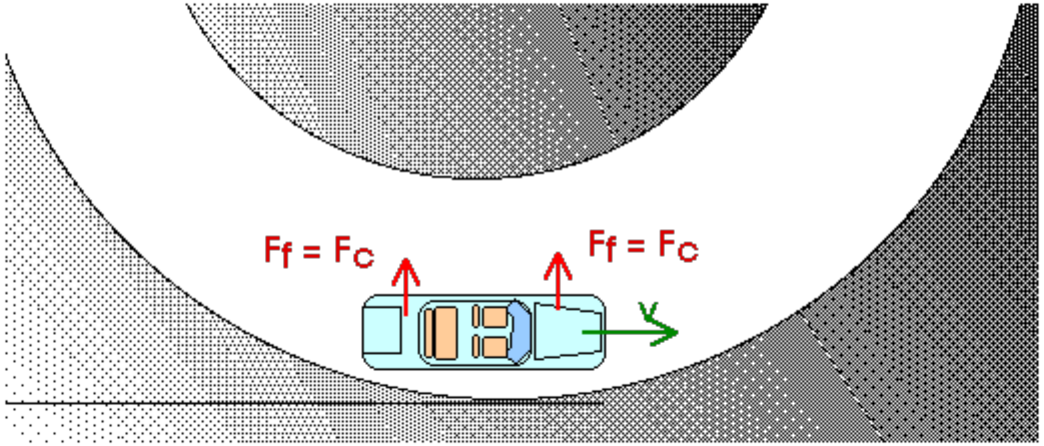


Figure form: http://www.ux1.eiu.edu/~cfadd/1150/05UCMGrav/Curve.html

$$\mu * m * g * \cos(\theta) = m * v^2/r$$

$$v^2/fr = r$$

Are we safe? Are we efficient?

- Buggy's Radius: buggyR = v^2/fr
- Stay in track: trackR <= buggyR <= trackR+width
- Maintain reasonable velocity: vMin <= v <= vMax

Approach to Problem

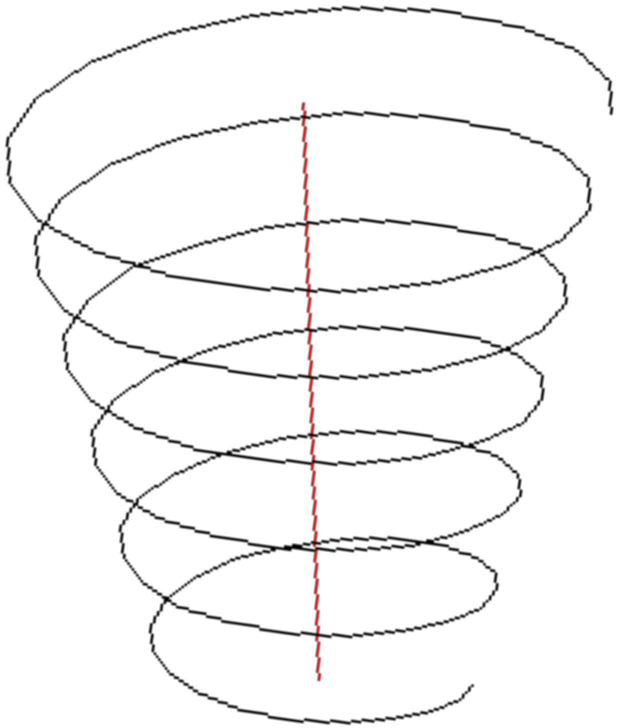
```
initial conditions ->
( /* track generation decision */
  tRate = -inRate OR tRate = -outRate OR tRate=0

  /* acceleration */
  a := A if safe OR
  a := -B if safe OR
  a := 0 if safe

/* ODEs - continuous evolution by physics */
)@loop invariant
/* ensure final safety conditions */
```

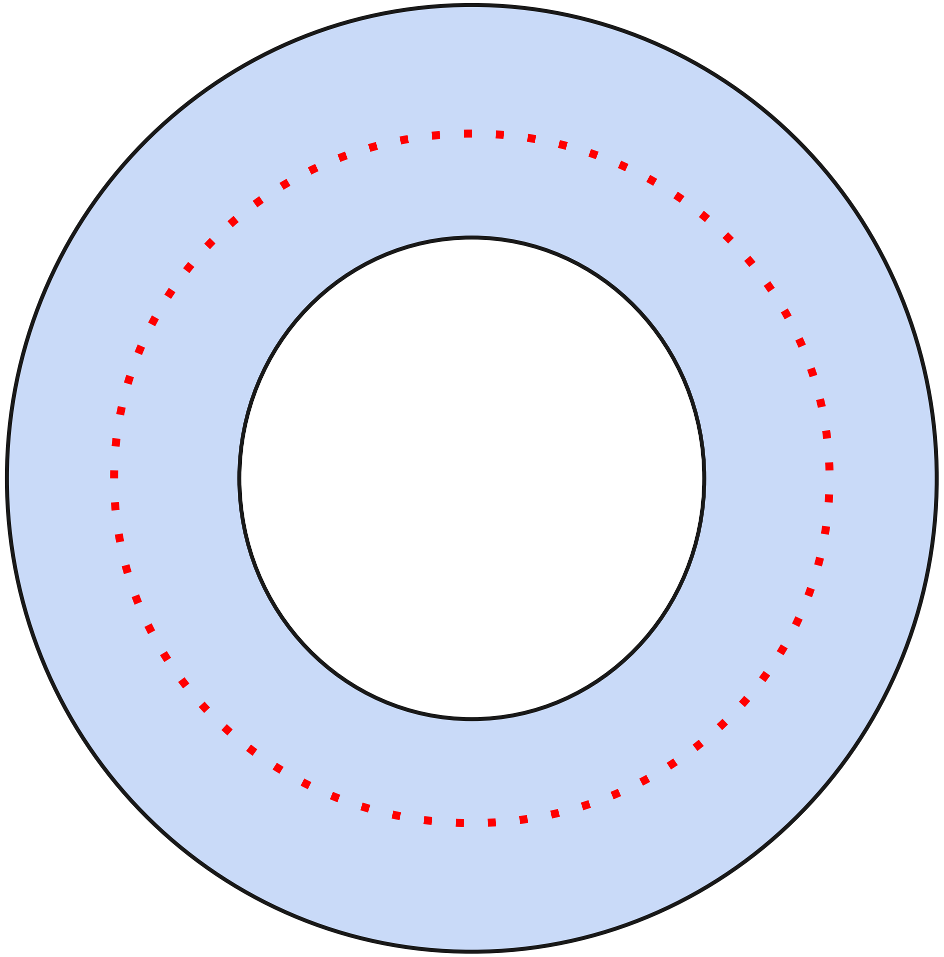
MODEL 1. THE HELIX

- Constant slope
- Radius can expand, shrink or remain the same
- Two challenges:
  - How to test for safety?
  - How to ensure safe condition exists?

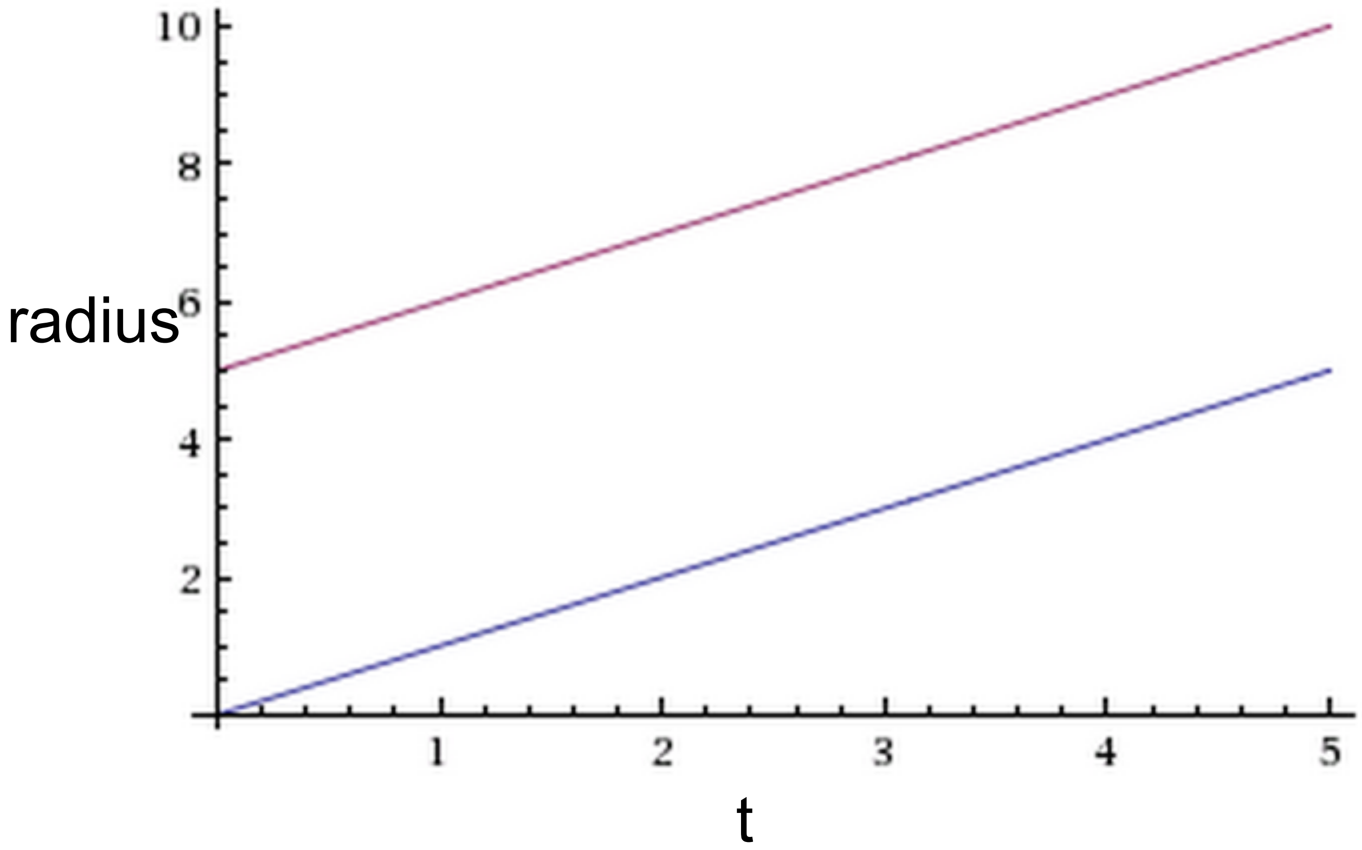


Simple Case: The Circular Track

- Track Radius does not change
- Can coast safely (a=0)

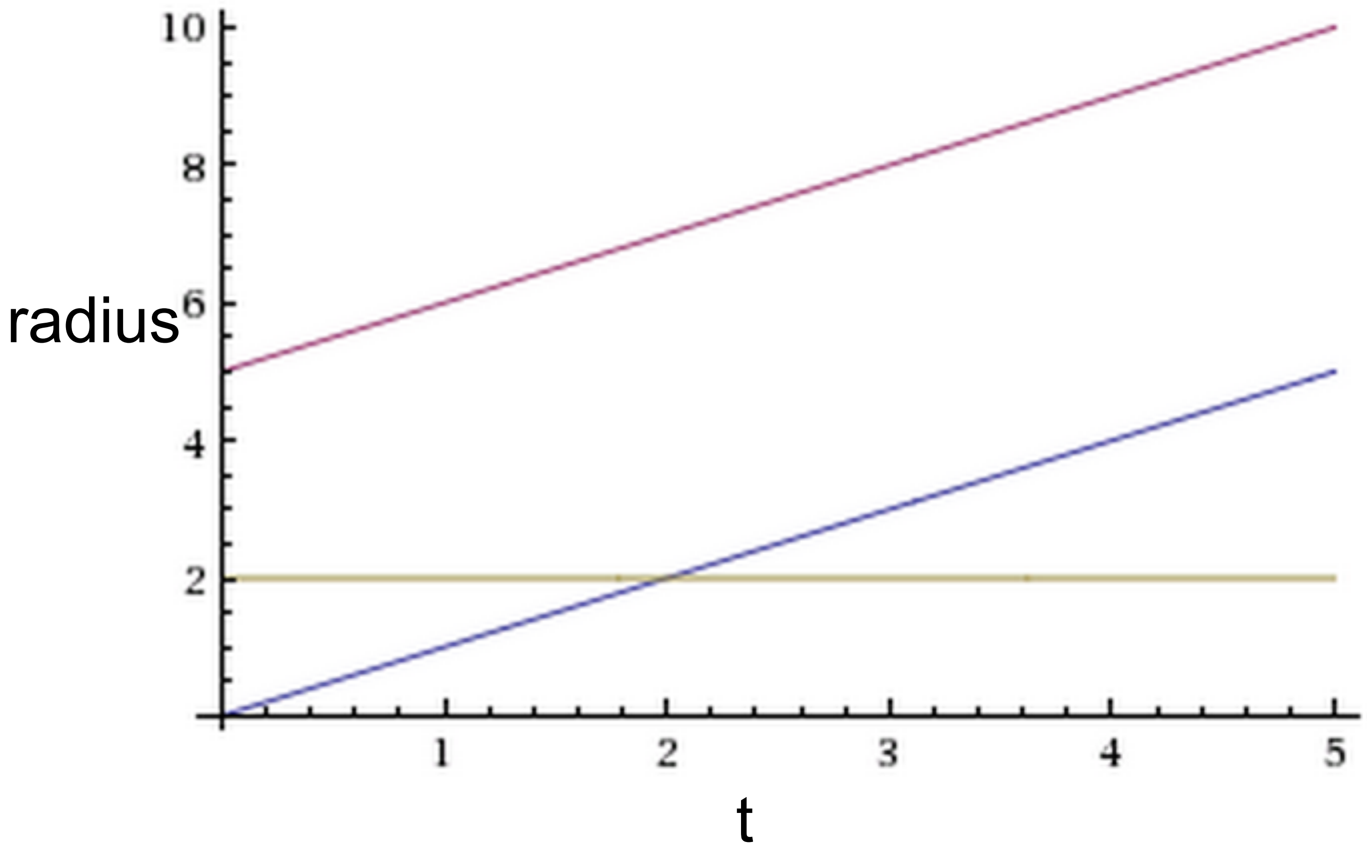


Finding a Safe Decision



inner = trackR_1+outRate*t
outer = trackR_1+width+outRate*t

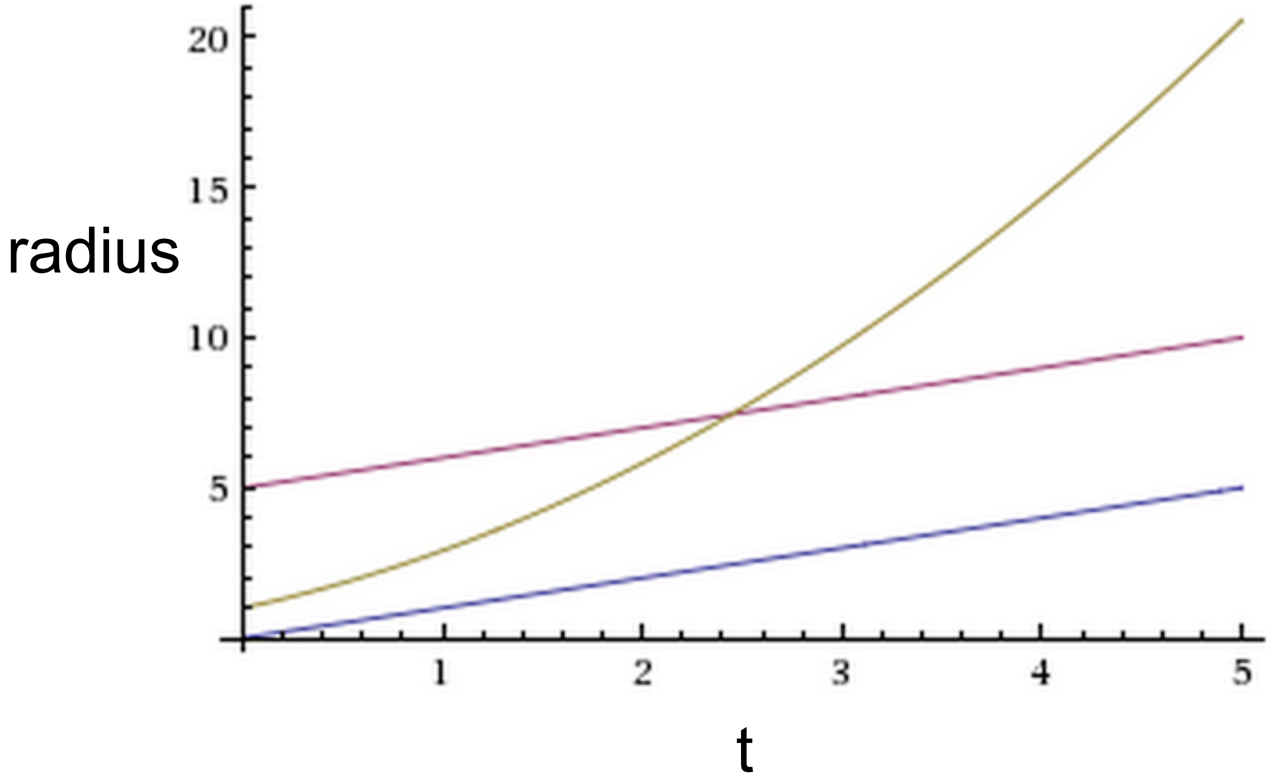
Finding a Safe Decision: Coasting?



inner = trackR_1+outRate*t
outer = trackR_1+width+outRate*t

Finding a safe decision: accelerating?

buggyR = (v+A*t)^2/fr



Finding a Safe Decision: Defining Constraints

- When the track is expanding:
  - Coast safely in the outer half
  - Accelerate safely in the inner one
- When the track is shrinking:
  - Coast safely in the inner half
  - Brake in the outer half
- When track is not changing:
  - Coast safely everywhere
- Define formulas to ensure these

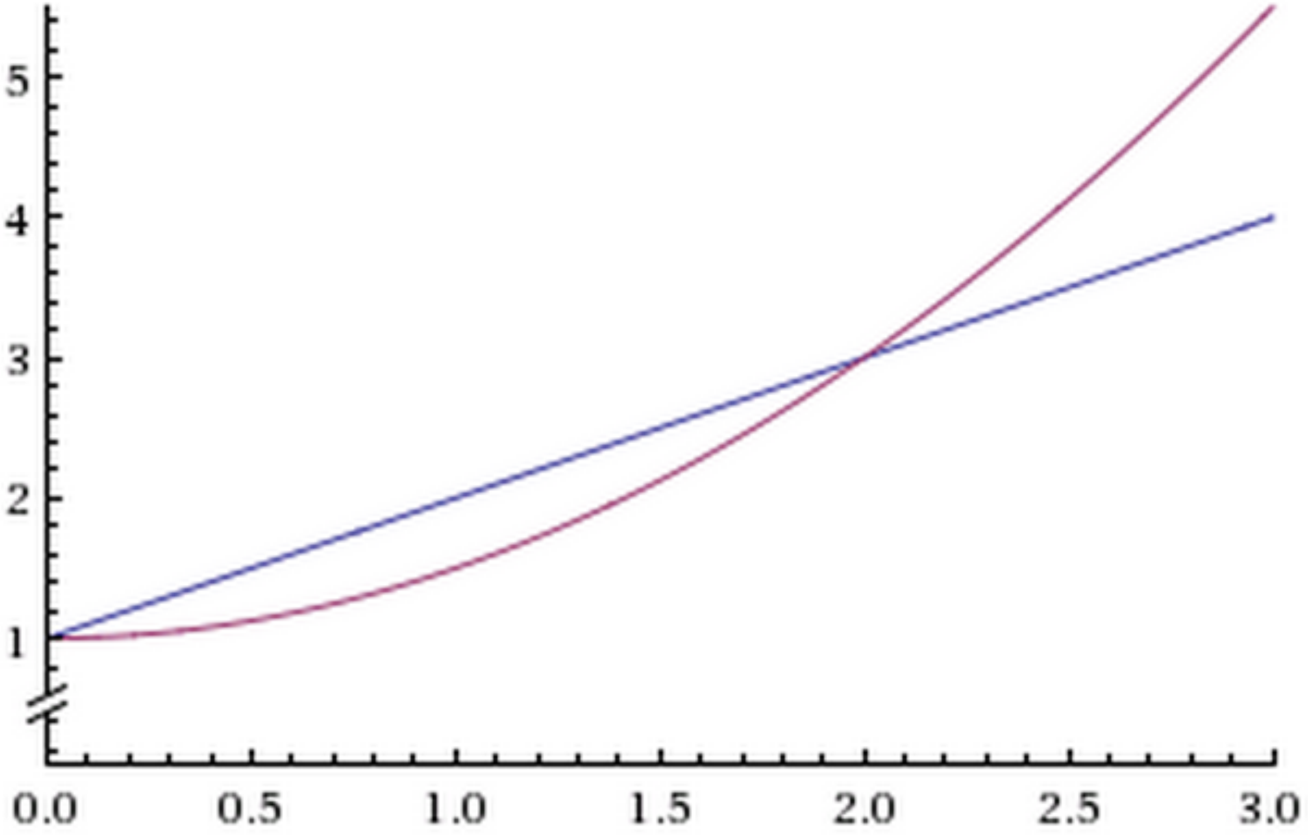
Constraint: Coasting on Expanding Track

- Outer edge is moving away - can't cause collision
- Inner Edge will approach middle the faster

$$trackR + width/2 >= trackR + inRate * t \quad \forall 0 \leq t \leq T$$

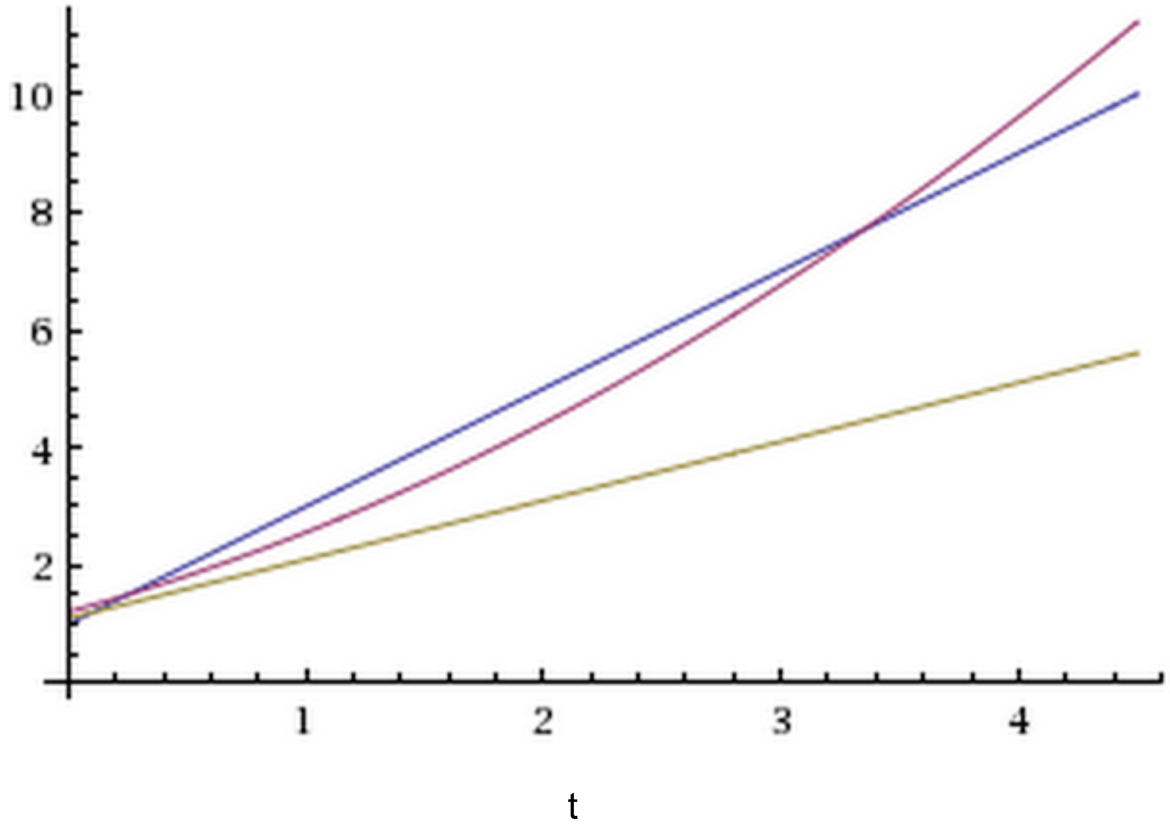
$$width/2 >= inRate * T$$

Another challenge for constraints: safe at the end, but not the middle!



R2D2 Goes to Buggy by Anastassia Kornilova & Emily Yeh

Another challenge for constraints: safe at the end, but not the middle!



R2D2 Goes to Buggy by Anastassia Kornilova & Emily Yeh

Constraint: Accelerating on expanding track

Outer edge: $2 * vMax * A * T / fr + A^2 * T^2 / fr <= width/2 + outRate * T$

Inner Edge: $outRate <= 2 * vMin * A / fr$

Safety of Helix Model

- Define 6 constraints for guaranteed safe decisions
- Constraints use constants and remain true
- Inequalities use extreme values - can be extended

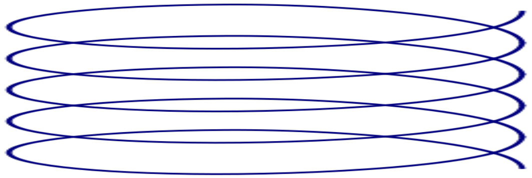
Model 2. The Helix with Hills

Track **slope** changes over time.
Track **radius** stays the same.

Safety: Don't Crash!

Track chooses new slope change,
***not** new slope.*

Case 1. Flat Slope

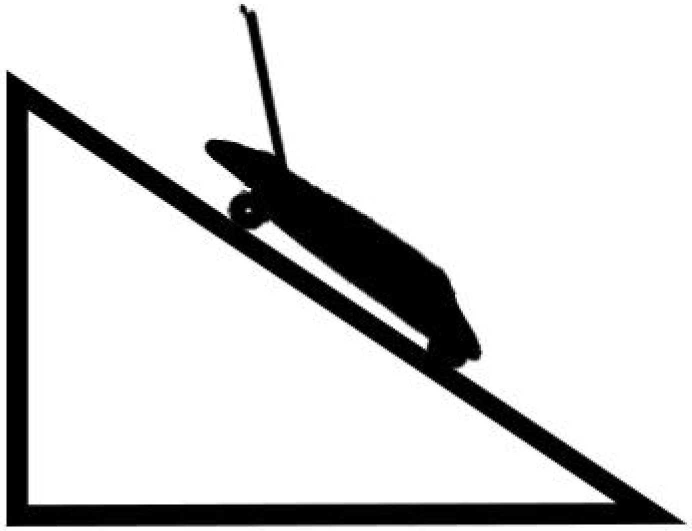


- Slope = 0
- $F_g = G*\cos(0) = G$

Risks:

- No new risks introduced by slope

Case 2. Downhill

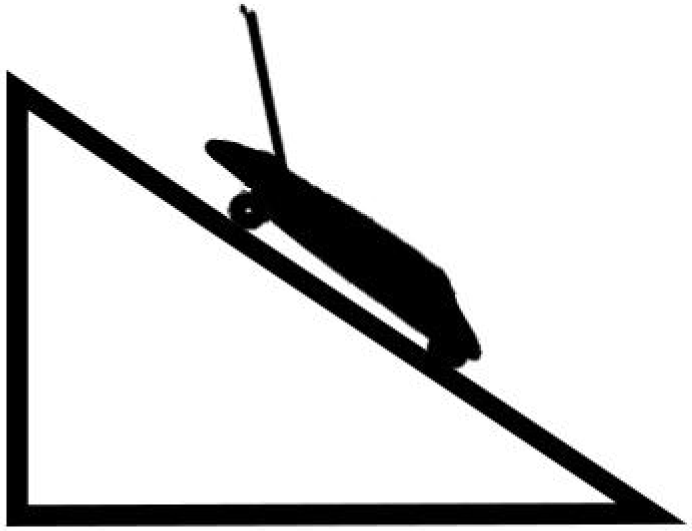


- $F_g > 0$

Risks:

- Crashing into the outer edge (high v)

Downhill Safety

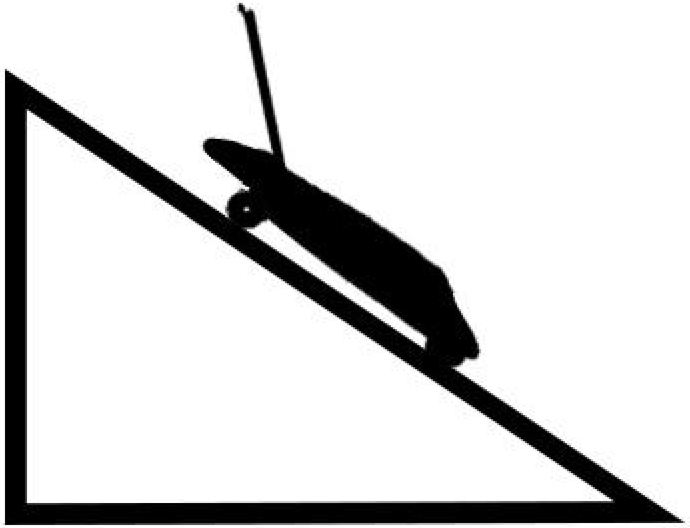


Constraints added to ensure:

Inner half of track means…
acceleration is <u>safe</u>.

Outer half of track means…
deceleration is <u>safe</u>.

Downhill Safety



Constraints added to ensure:

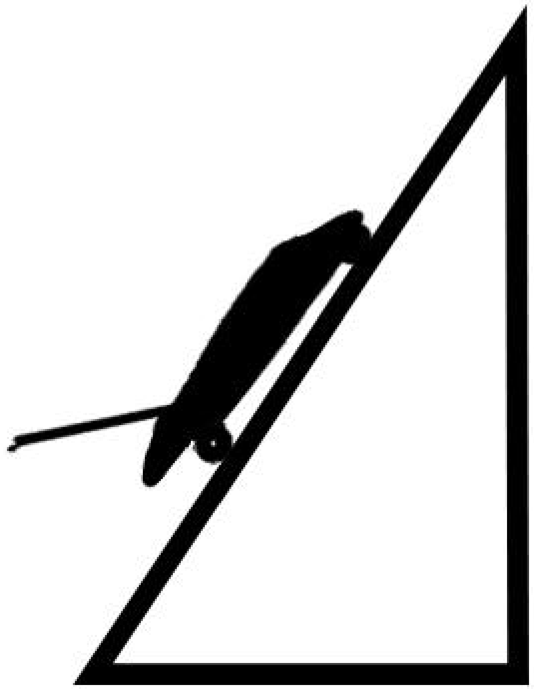
Inner half of track means…
acceleration is <u>safe</u>.

$$vMax >= (vMax + vMin)/2 + (A + G) * T + (T^2)/2$$

Outer half of track means…
deceleration is <u>safe</u>.

$$vMax >= vMax + (-B + G) * T + (T^2)/2$$

case 3. uphill

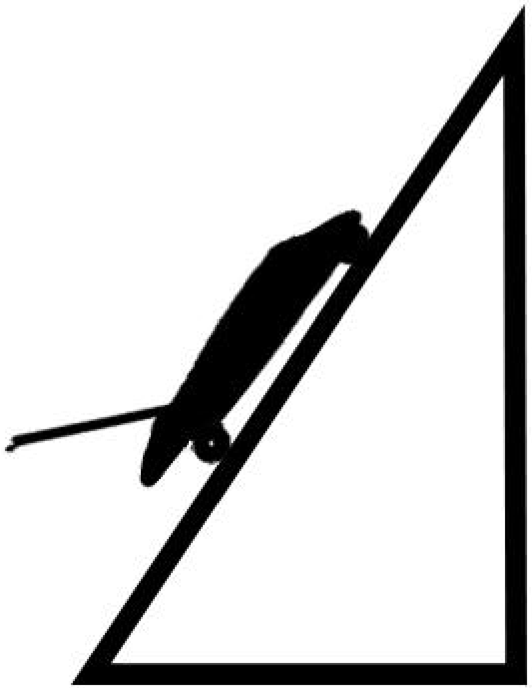


- $F_g < 0$

Risks:

- Crashing into inner edge (low v)

UPHILL SAFETY

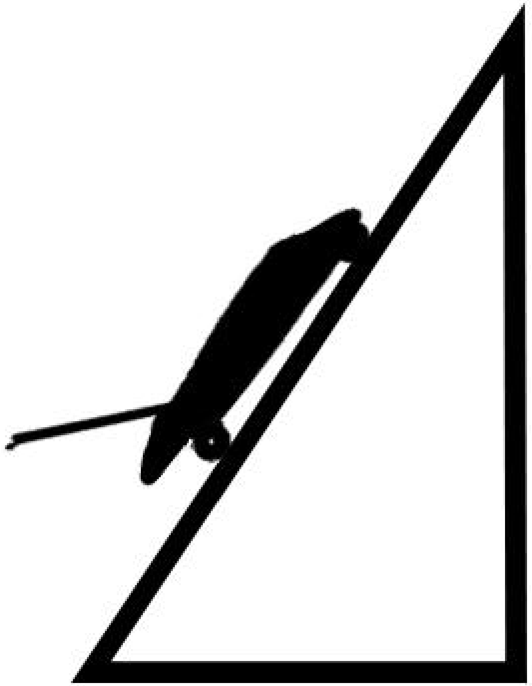


Constraints added to ensure:

Inner half of track means…
acceleration is <u>safe</u>.

Outer half of track means…
deceleration is <u>safe</u>.

UPHILL SAFETY



Constraints added to ensure:

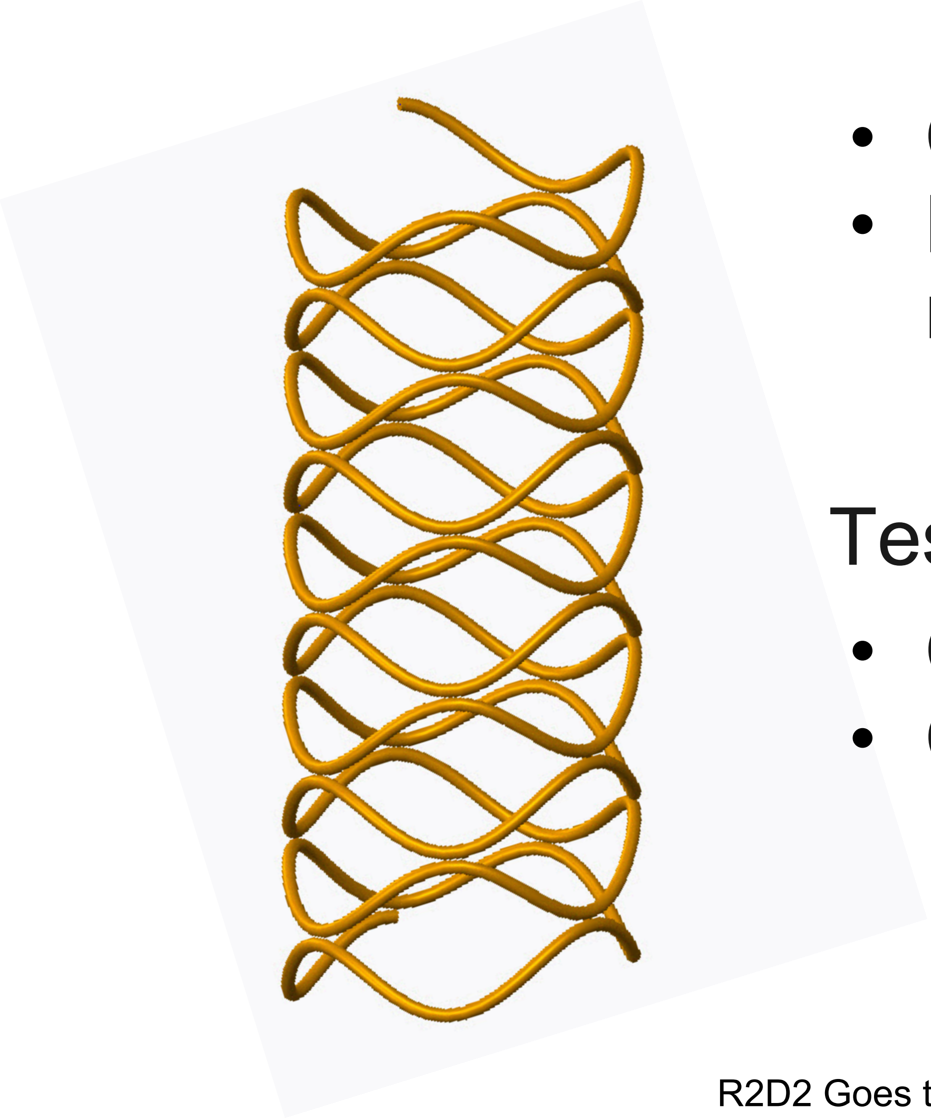
Inner half of track means…
acceleration is <u>safe</u>.

$$vMin <= vMin + (A - G) * T - (T^2)/2$$

Outer half of track means…
deceleration is <u>safe</u>.

$$vMin <= (vMin + vMax)/2 + (-B - G) * T - (T^2)/2$$

Model Controls

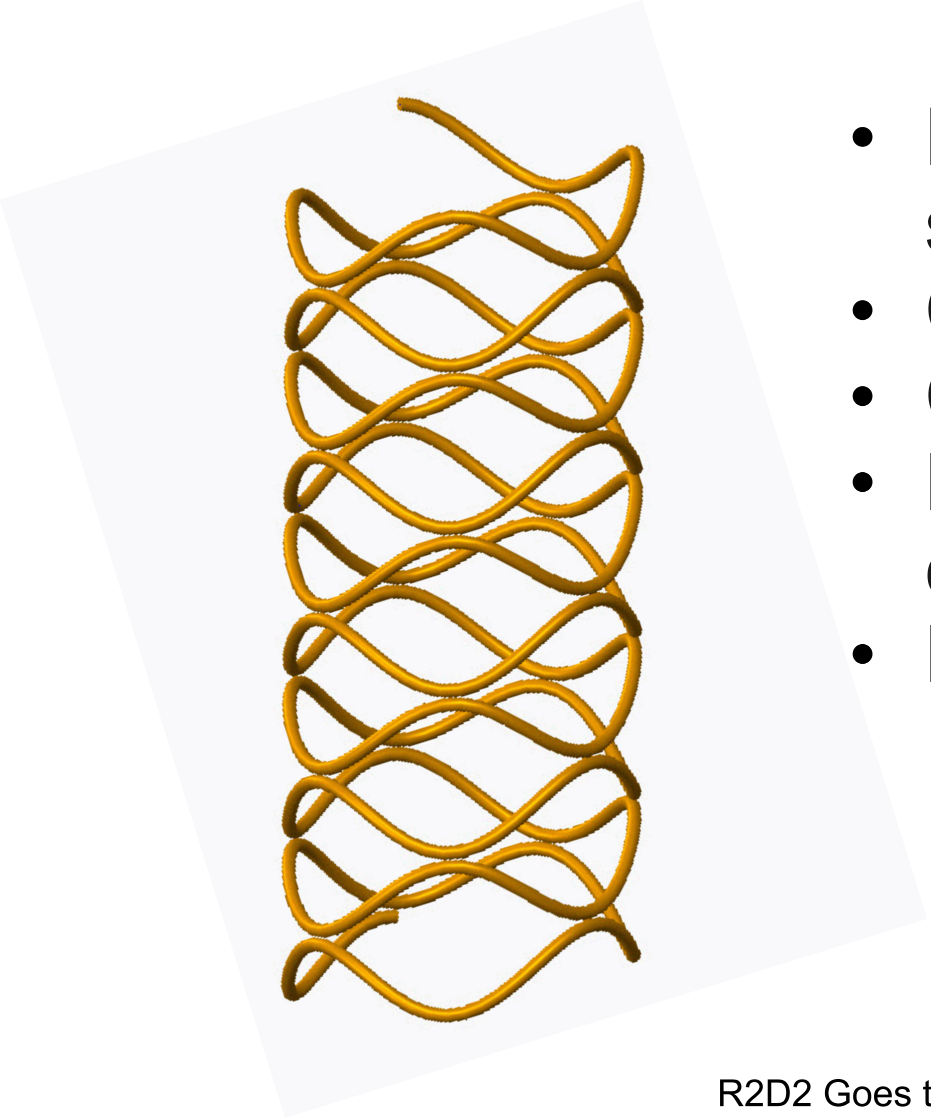


- Choices: A, -B, 0
- Makes choice based on tests that make sure we won't crash

Tests:

- Crash into inner edge?
- Crash into outer edge?

Model 2 Summary

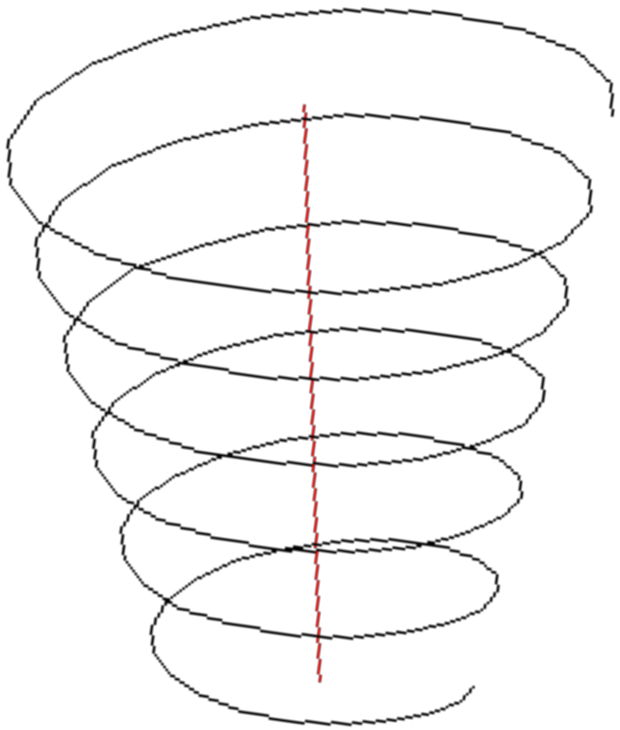


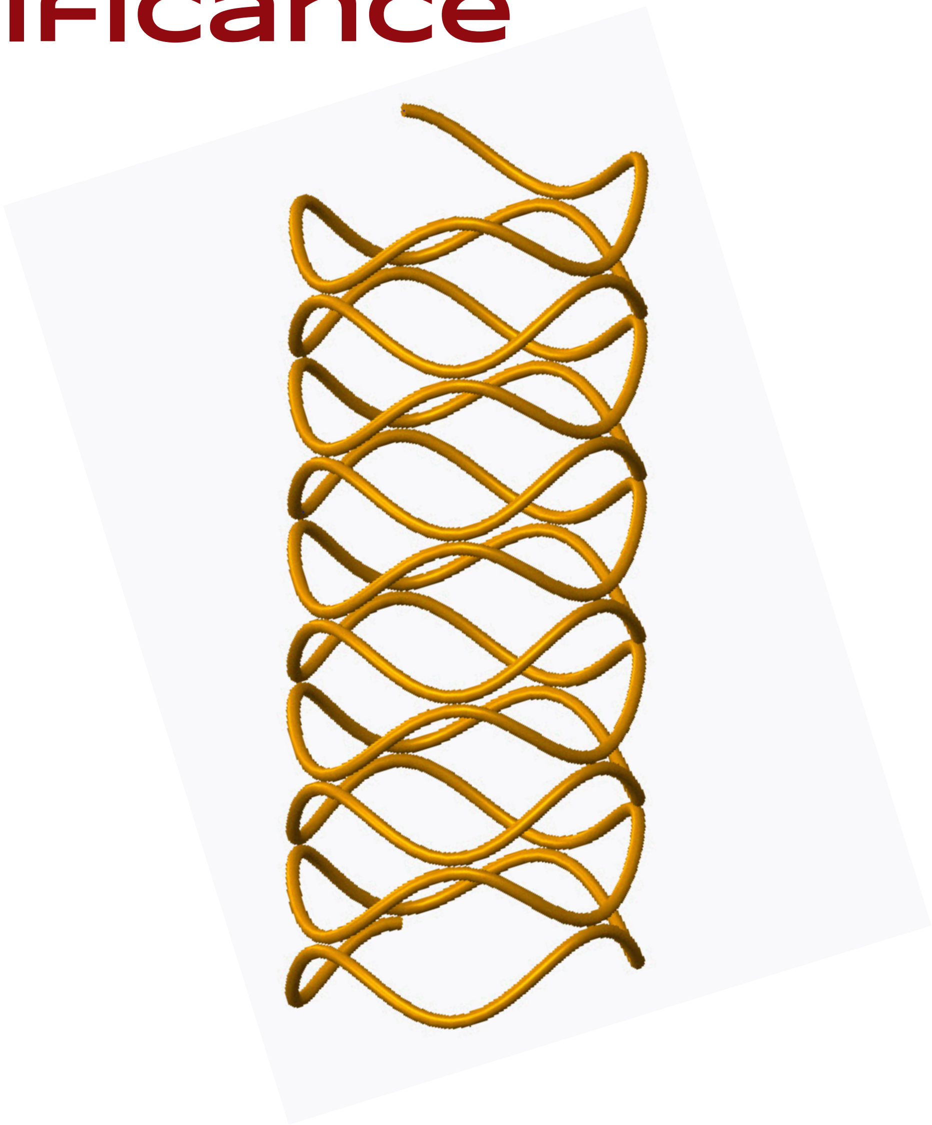
- Makes decisions based on upcoming slope changes
- Constraints use constants
- Constraints and tests ensure safety
- Model is limited by conservative constraints
- Model doesn't analyze F_{fr} change

Future Directions

- Better efficiency: find best path around track segment
- More Diverse Tracks
  - Combine changes in slope and radius
  - Allow for straight segments
- Looking ahead in tracks to find better paths
- Less synchronized controls

summary & significance





R2D2 Goes to Buggy by Anastassia Kornilova & Emily Yeh

thank you.