Unicycle and Bicycle Model for Car Collision Avoidance

Harry Sood
University of Michigan
July 31st, 2019 – Updated: February 17th, 2020

Collaborator: Aakash Abhishek
Advisor: Jean-Baptiste Jeannin

Contents

Brief Review of the Unicycle Model

Ackermann Steering Geometry (ASG)

- In ASG, the car’s four wheels are all at a right angle relative to the center of turn.
- In the Unicycle Model, this is achieved by setting a radius which gives the location of the center of turn.
- ASG is present in the Bicycle Model as well. The Bicycle Model will be discussed later.

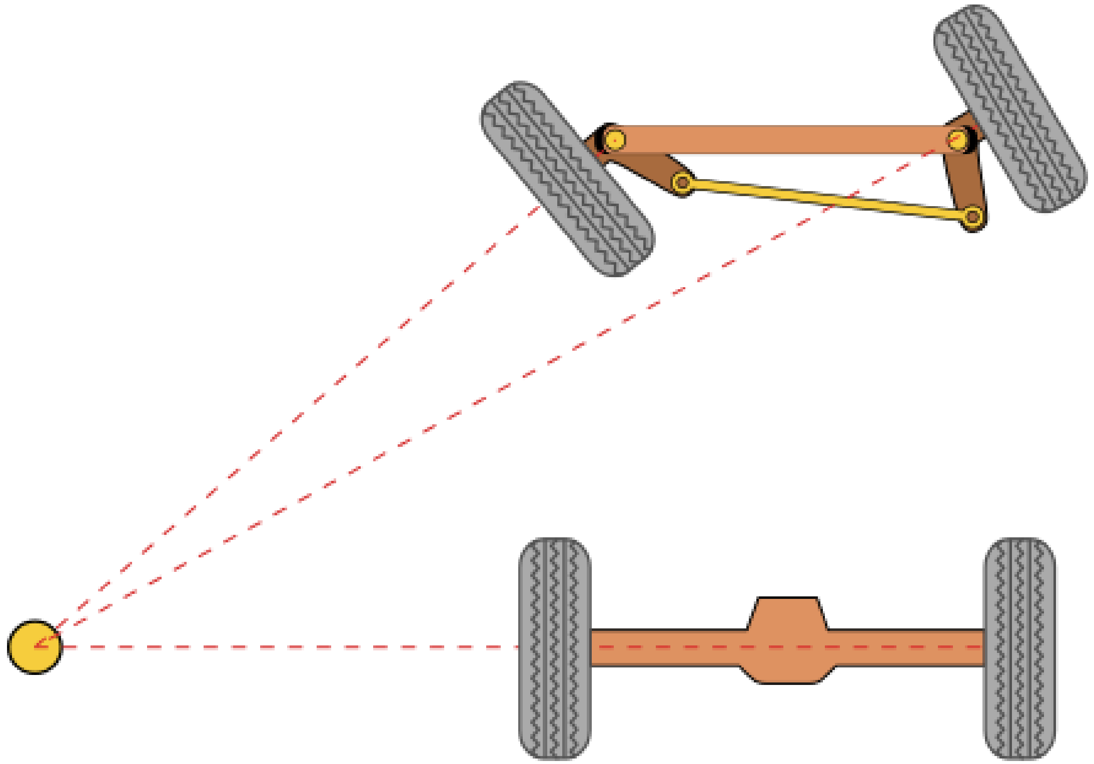
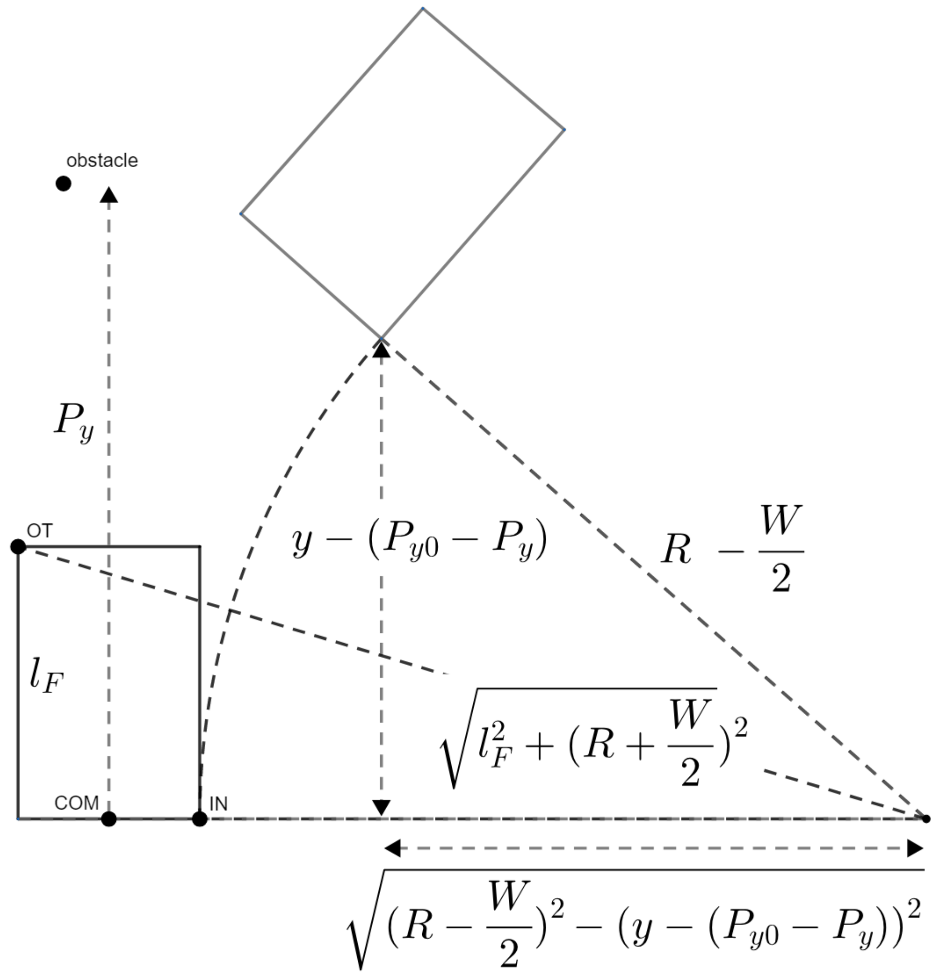


Image Credit: https://en.wikipedia.org/wiki/Ackermann_steering_geometry

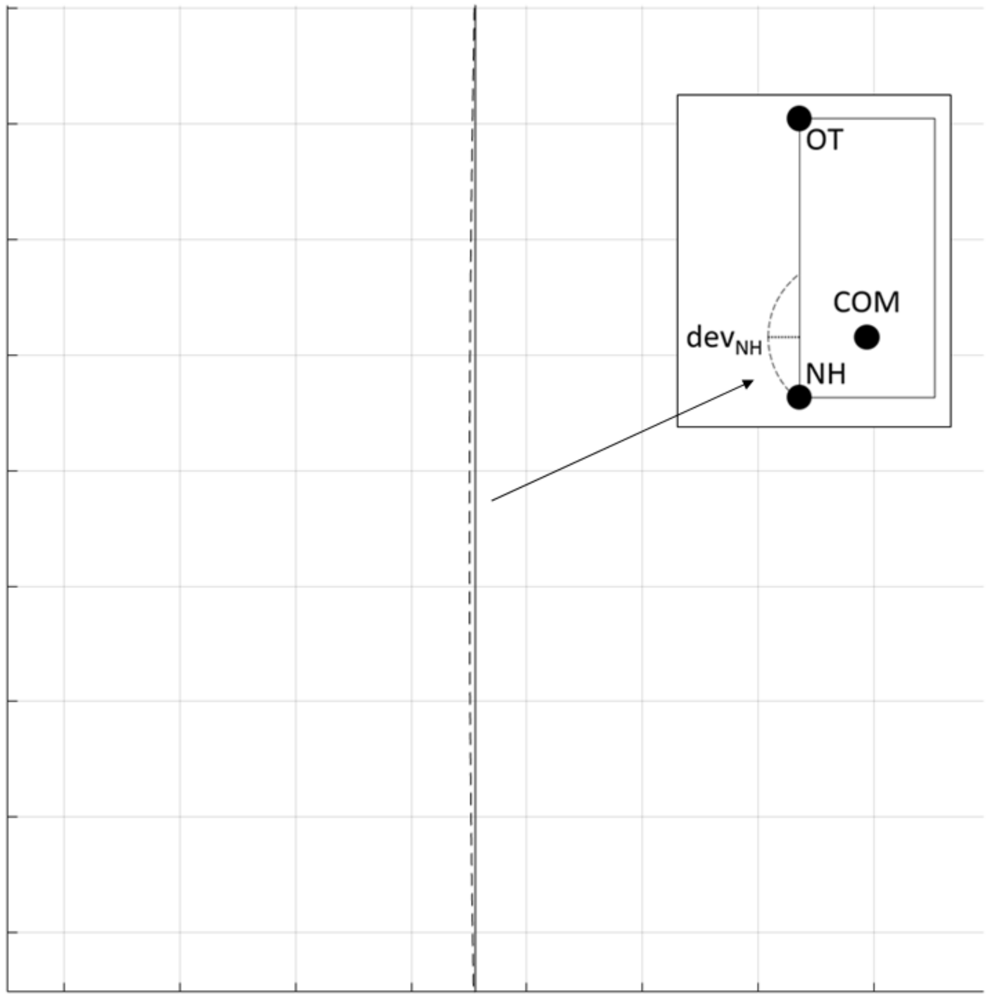
Unicycle Model

- The Unicycle Model makes the simplification that the car's motion is modeled by a single point at the center of the rear axle.
- The car still has a length and width around this point
- The inputs in this model are a radius of turn, R, and an angle of braking, ϕ (discussed later).



Notch

- Any part of the car behind the rear-axle is ignored due to the creation of a “notch”.
- The notch is an insignificant protrusion formed when the car turns. Thus, it is ignored/eliminated by not considering the rear of the car.



Circle of Forces

- The Circle of Forces is used to allocate correct amounts of force to braking and turning in order to prevent skidding. These values are decided by the angle of braking, ϕ.
- The relation follows:

  (force of static friction)^2 =

  (force of acceleration/braking)^2 +

  (force of turn)^2

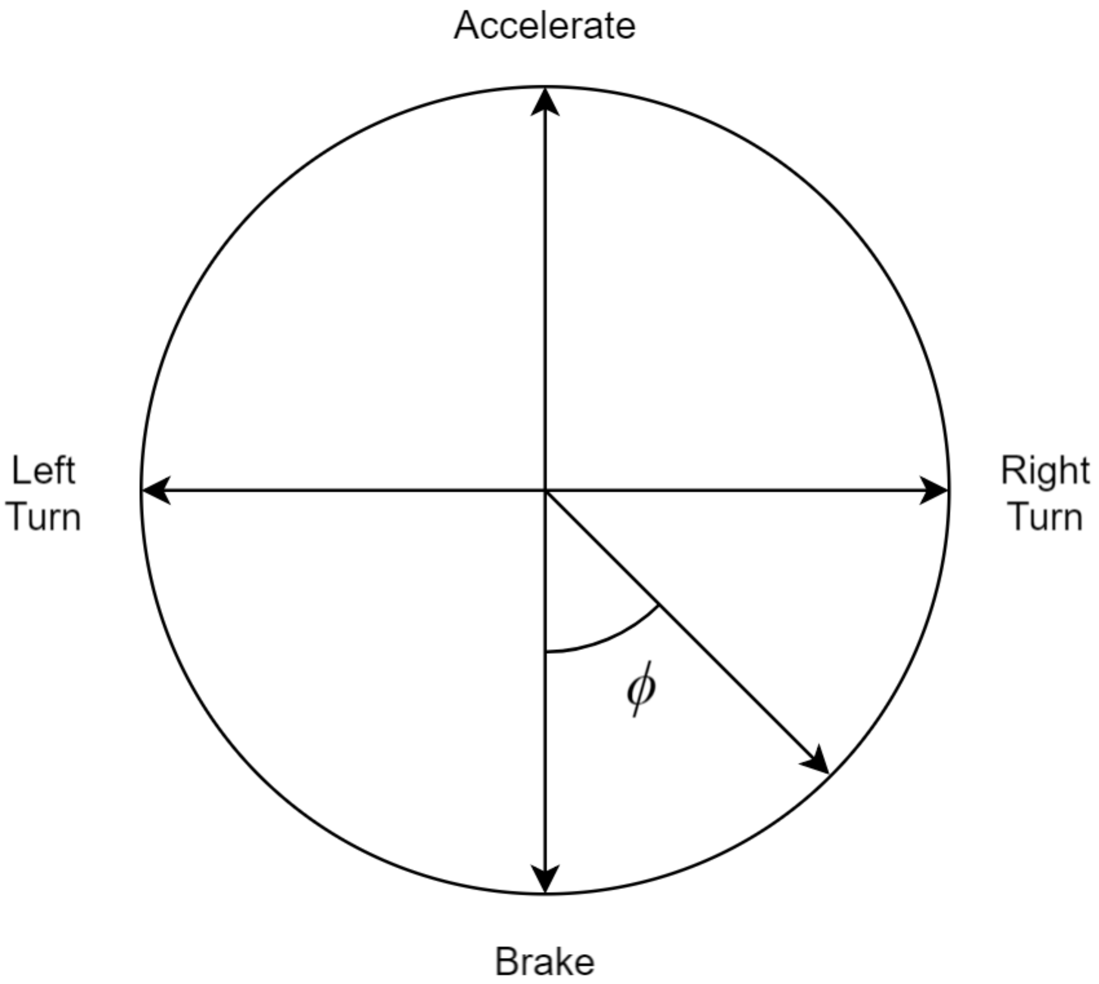
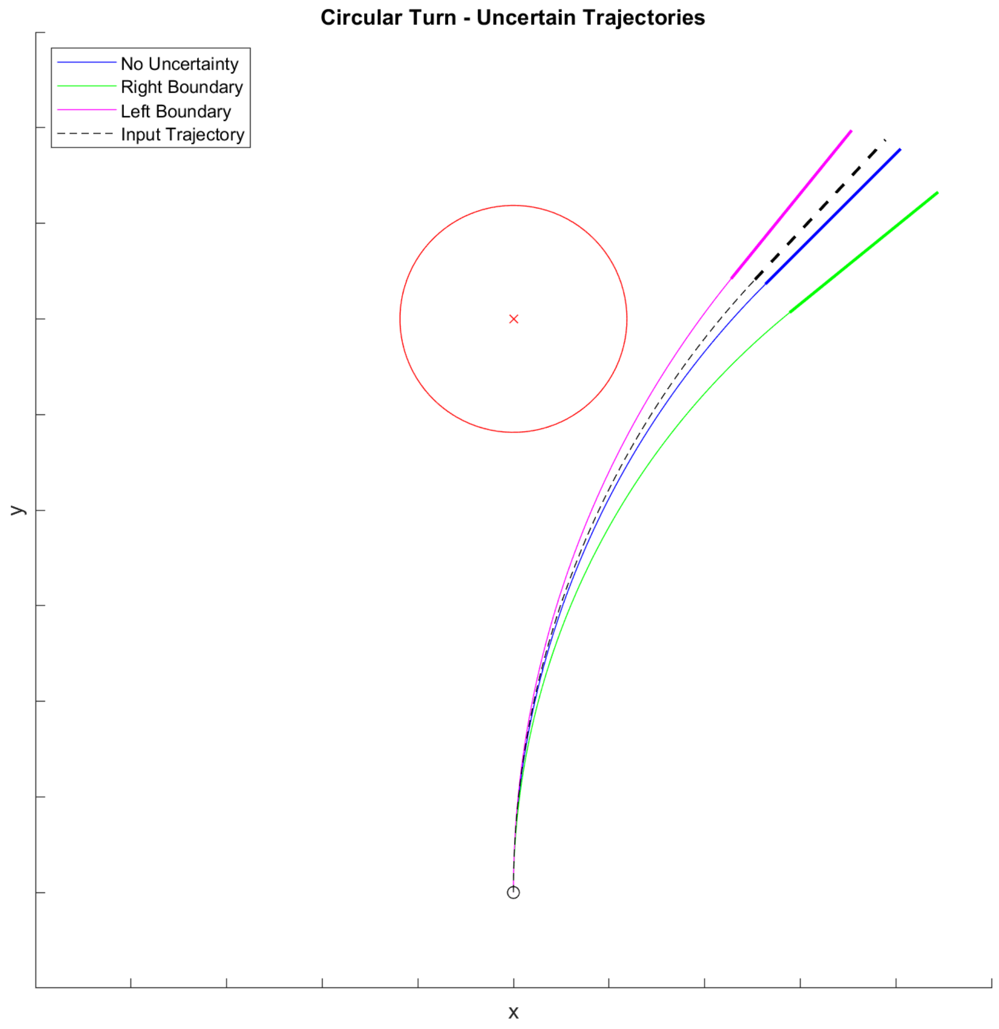


Image Credit: EMSOFT submission

Uncertainty in the Unicycle Model

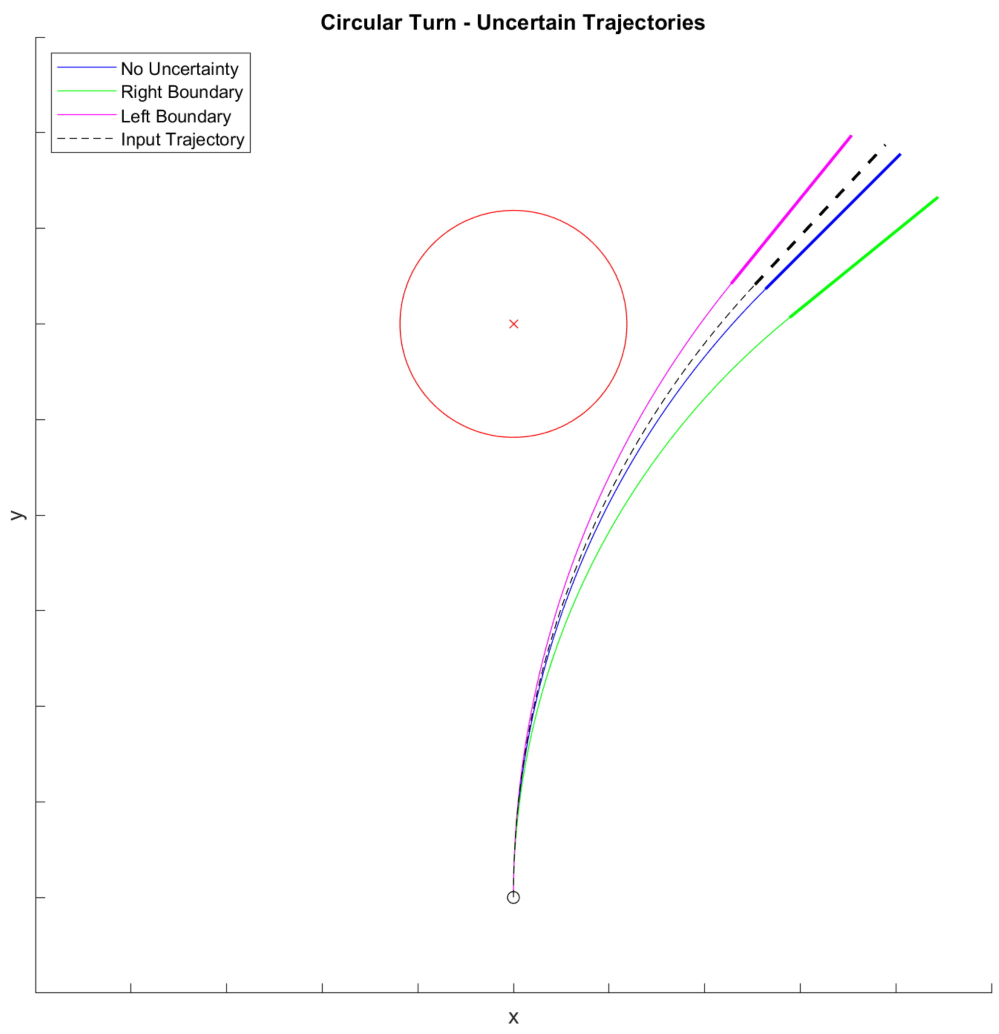
Uncertain Model

- Since R and ϕ are the input parameters, they are a source of uncertainty.
- Uncertainty can be introduced on these parameters by giving them a range of possible values.



Uncertain Model

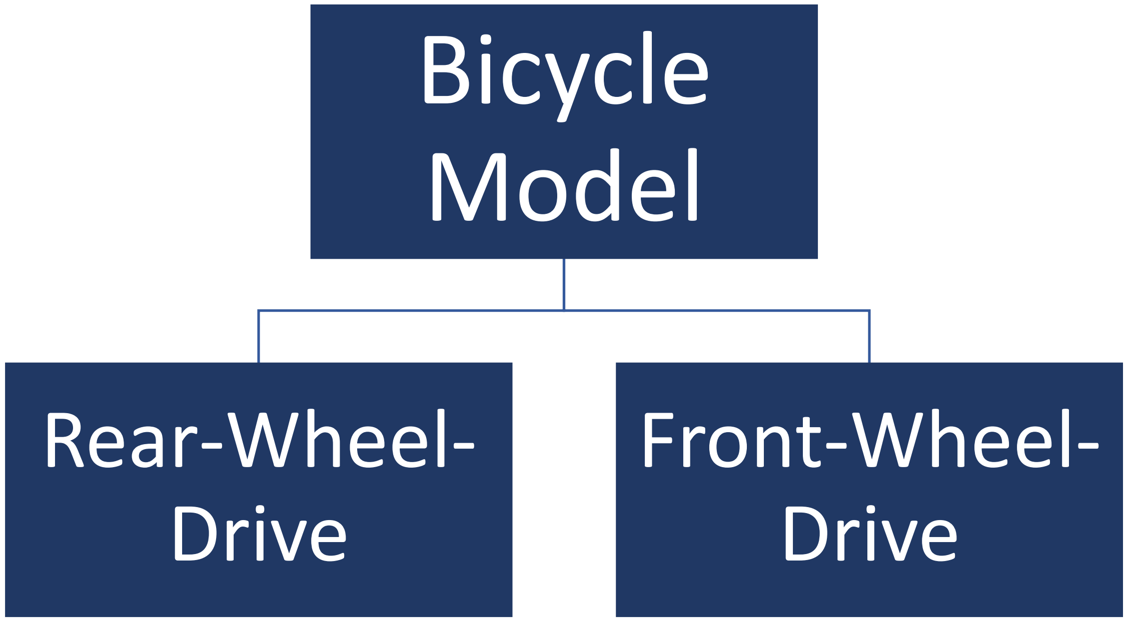
- The system shown to the right is modeled at 3 standard deviations (std) from the target trajectory.
- In a right turn, the left boundary is given by the largest R (+3 std) and the smallest ϕ (-3 std). The right boundary is given by the smallest R (-3 std) and the largest ϕ (+3 std).
- The obstacle is also given an uncertain location. It is still known to be contained within the circle.



Bicycle Model

Bicycle Model

- The Bicycle Model expands on the Unicycle model and accounts for two wheels instead of a single point.
- The rear-wheel is located at the center of the rear axle and the front wheel is located directly forward of it
- The inputs in this model are the velocity of the vehicle, v, and the turning angle of the front wheel, γ.



Rear-Wheel-Drive

- The equations governing rear-wheel-drive follow:

$$\dot{v_r}(t) = a_r$$

$$\dot{\theta}_r(t) = v_r(t)\left(\frac{\tan\gamma}{l}\right)$$

$$\dot{x_r}(t) = v_r(t)\sin\left(\theta(t)\right)$$

$$\dot{y_r}(t) = v_r(t)\cos\left(\theta(t)\right)$$

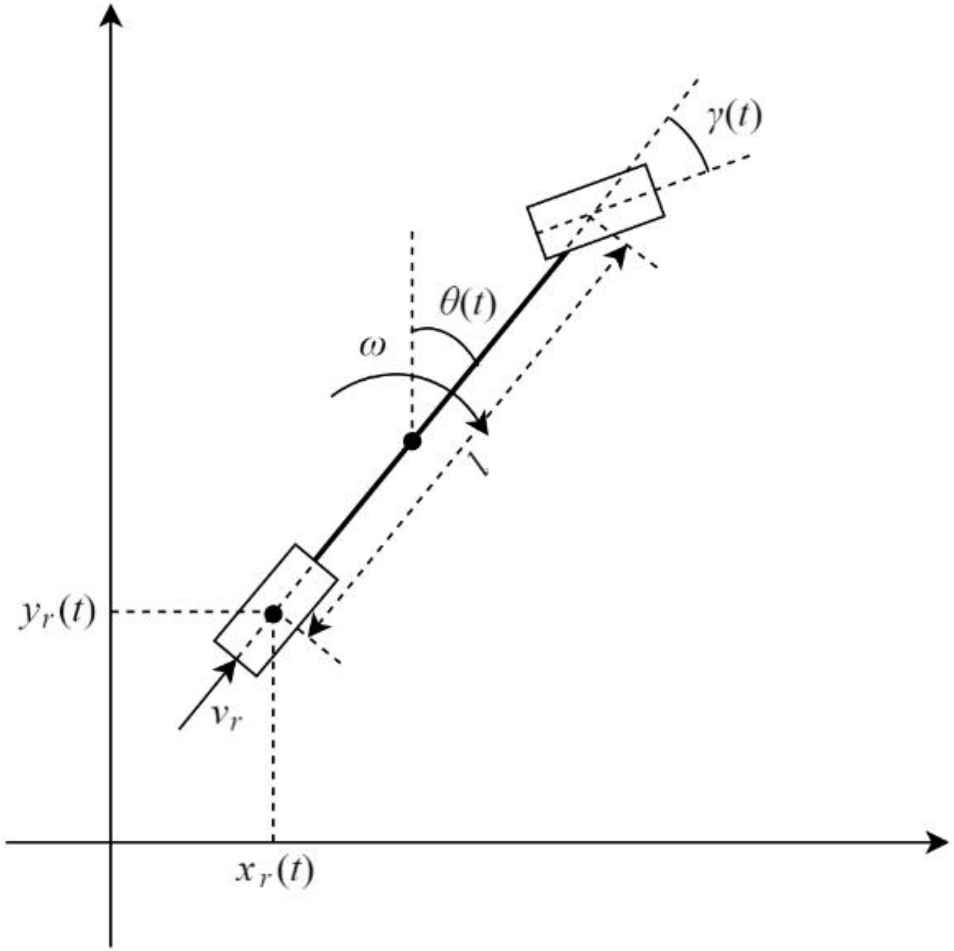


Image Credit: Aakash's Bicycle Model Report

Front-Wheel-Drive

- The equations governing front-wheel-drive follow:

$$\dot{v}_f(t) = a_f$$

$$\dot{\theta}_f(t) = v_f(t)\left(\frac{\tan\gamma}{l}\right)$$

$$\dot{x}_f(t) = v_f(t)\sin\left(\theta(t)+\gamma\right)$$

$$\dot{y}_f(t) = v_f(t)\cos\left(\theta(t)+\gamma\right)$$

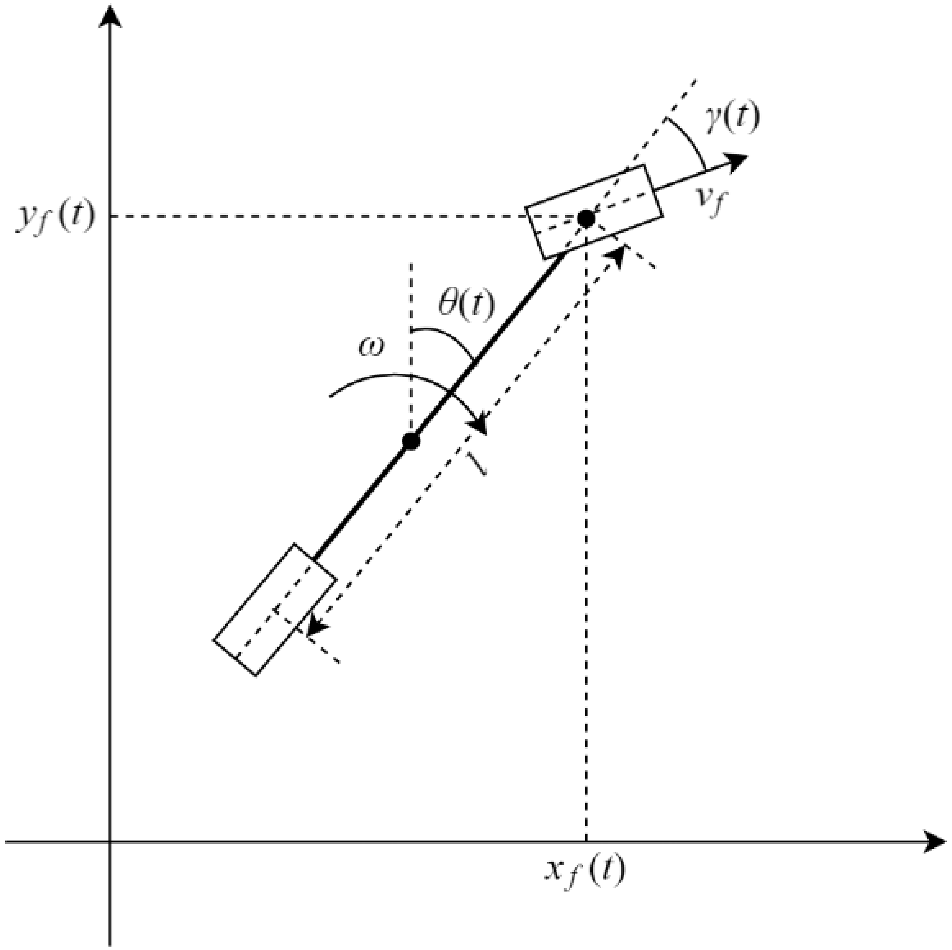


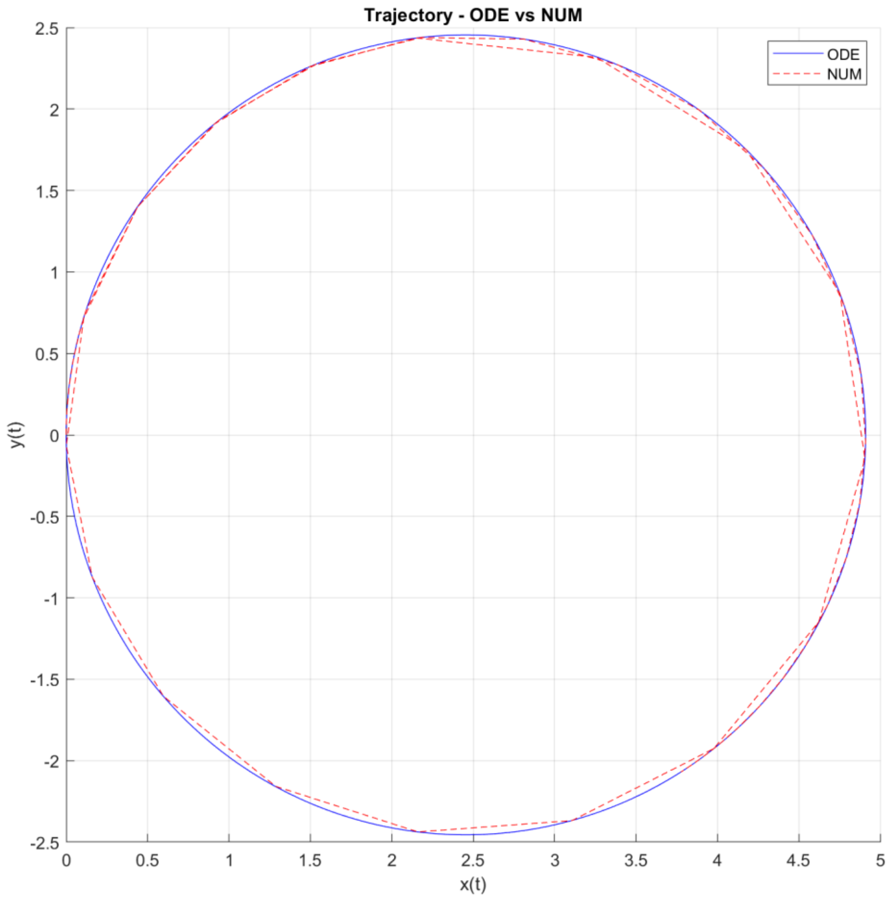
Image Credit: Aakash's Bicycle Model Report

Rear-Wheel-Drive Solutions

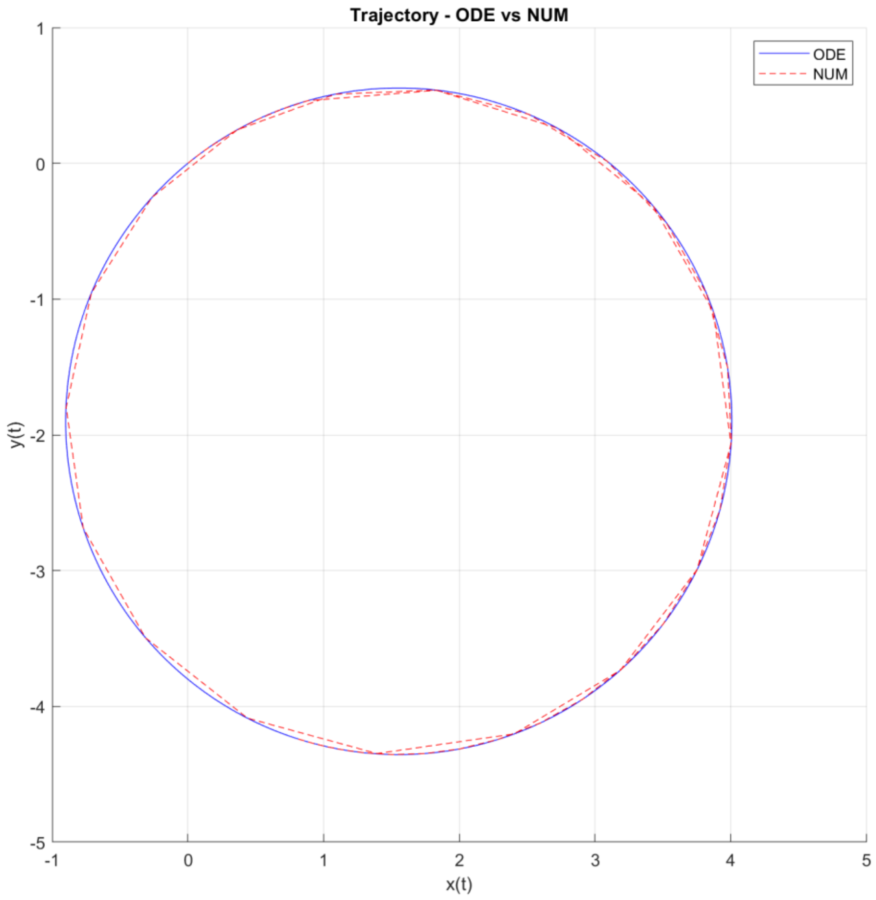
- The solutions to the Rear-Wheel-Drive equations follow:

$$
\begin{aligned}
v_r(t) &= v_{r,0} + a_r t \\
\theta_r(t) &= \left(v_{r,0}t + \frac{1}{2}a_r t^2\right)\left(\frac{\tan\gamma}{l}\right) \\
x_r(t) &= -\left(\frac{l}{\tan\gamma}\right)\left[\cos\left(\left(v_{r,0}t + \frac{1}{2}a_r t^2\right)\left(\frac{\tan\gamma}{l}\right)\right)\right] + \left(\frac{l}{\tan\gamma}\right) \\
y_r(t) &= \left(\frac{l}{\tan\gamma}\right)\left[\sin\left(\left(v_{r,0}t + \frac{1}{2}a_r t^2\right)\left(\frac{\tan\gamma}{l}\right)\right)\right]
\end{aligned}
$$

Trajectories



RWD



FWD

Proof for Rear-Wheel-Drive

Proof Steps Using KeYmaera X

1. Create an implicit model in KeYmaera X.
2. Prove the implicit model.
3. Create an equivalence between the implicit and explicit models in KeYmaera X.
4. Prove the equivalence.

Since acceleration does not change the trajectory, the proof can be approached in two ways.

- Eliminating acceleration completely from the model and showing that it does not matter by hand.
- Accounting for acceleration within the KeYmaera X model.

KeYmaera X Model – Without Acceleration

```
Problem.
    (
        (L > 0 & W > 0) &
        (vo > 0)
    ) & (
        (l >= lMin & l <= lMax) &
        (lMin > 0 & lMax > 0 & lMax >= lMin) &
        (tG >= tGMin & tG <= tGMax) &
        (tGMin > 0 & tGMax > 0 & tGMax >= tGMin) &
        (cMin > 0 & cMin <= 1)
    ) & (
        (x = -l/tG & y = 0) &
        (s = 0 & c = 1) &
        (t = 0)
    ) & (
        (mu > 0 & g > 0) &              /** condition to check compliance with circle of forces where acceletion is 0 **/
        ((mu*g)^2 >= (vo^2*tG/l)^2)     /** force of friction^2 >= force of acceleration^2 + force of turn^2 **/
    ) & (
        \forall tn \forall xn \forall yn \forall sn \forall cn
            ((( tn = 0 & cn >= cMin & sn >= 0 & sn^2 = 1 - cn^2 & xn = -l/tG*cn & yn = l/tG*sn) |
            (tn >= 0 & cn = cMin & sn >= 0 & sn^2 = 1 - cn^2 & xn = -l/tG*cn + vo*sn*tn & yn = l/tG*sn + vo*cn*tn))
            -> ((abs(cn*(xo-xn) - sn*(yo-yn)) > (W/2)) | (abs(sn*(xo-xn) + cn*(yo-yn) - (L/2)) > (L/2))))
    )
    ->
        [{
            {{s' = c*vo*tG/l, c' = -s*vo*tG/l, x' = vo*s, y' = vo*c & t = 0 & c >= cMin}
            ++ {t' = 1, x' = vo*s, y' = vo*c & t >= 0 & c = cMin}}*
        }] ((abs(c*(xo-x) - s*(yo-y)) > (W/2)) | (abs(s*(xo-x) + c*(yo-y) - (L/2)) > (L/2)))
End.
```

KeYmaera X Tactic – Without Acceleration

```
implyR(1) ; loop({`t=0&c>=cMin()&s>=0&s^2=1-c^2&x=-l()/tG()*c&y=l()/tG()*s|t>=0&c=cMin()&s>=0&s^2=1-c^2&x=-l()/tG()*c
+vo()*s*t&y=l()/tG()*s+vo()*c*t`}, 1) ; <(
  allL({`t`}, -6) ; allL({`x`}, -6) ; allL({`y`}, -6) ; allL({`s`}, -6) ; allL({`c`}, -6) ; QE,
  allL({`t`}, -7) ; allL({`x`}, -7) ; allL({`y`}, -7) ; allL({`s`}, -7) ; allL({`c`}, -7) ; QE,
  hideL(-7=={`\forall tn \forall xn \forall yn \forall sn \forall cn (tn=0&cn>=cMin()&sn>=0&sn^2=1-cn^2&xn=-l()/tG()*
cn&yn=l()/tG()*sn|tn>=0&cn=cMin()&sn>=0&sn^2=1-cn^2&xn=-l()/tG()*cn+vo()*sn*tn&yn=l()/tG()*sn+vo()*cn*tn->abs(cn*(xo
()-xn)-sn*(yo()-yn))>W()/2|abs(sn*(xo()-xn)+cn*(yo()-yn)-L()/2)>L()/2)`}) ; hideL(-3=={`L()>0`}) ; hideL(-3=={`W()>0
`}) ; hideL(-4=={`l()<=lMax()`}) ; hideL(-6=={`lMax()>0`}) ; hideL(-6=={`lMax()>=lMin()`}) ; hideL(-7=={`tG()<=tGMax
()`}) ; hideL(-13=={`tGMax()>=tGMin()`}) ; hideL(-12=={`tGMax()>0`}) ; choiceb(1) ; andR(1) ; <(
    MR({`t=0&c>=cMin()&s>=0&s^2=1-c^2&x=-l()/tG()*c&y=l()/tG()*s`}, 1) ; <(
      ODE(1),
      QE
      ),
    MR({`t>=0&c=cMin()&s>=0&s^2=1-c^2&x=-l()/tG()*c+vo()*s*t&y=l()/tG()*s+vo()*c*t`}, 1) ; <(
      ODE(1),
      QE
      )
    )
  )
```

KeYmaera X Model – With Acceleration

```
Problem.
    (
        (L > 0 & W > 0) &
        (vo > 0 & v = vo) &
        (a != 0)
    ) & (
        (l >= lMin & l <= lMax) &
        (lMin > 0 & lMax > 0 & lMax >= lMin) &
        (tG >= tGMin & tG <= tGMax) &
        (tGMin > 0 & tGMax > 0 & tGMax >= tGMin) &
        (cMin > 0 & cMin <= 1)
    ) & (
        (x = -l/tG & y = 0) &
        (s = 0 & c = 1) &
        (p = 0)
    ) & (
        (mu > 0 & g > 0) &                      /** condition to check compliance with circle of forces where acceletion is 0 **/
        ((mu*g)^2 >= (a)^2+(vo^2*tG/l)^2)       /** force of friction^2 >= force of acceleration^2 + force of turn^2 **/
    ) & (
        \forall pn \forall xn \forall yn \forall sn \forall cn
            (((pn = 0 & cn >= cMin & sn >= 0 & sn^2 = 1-cn^2 & xn = -l/tG*cn & yn = l/tG*sn) |
            (pn >= 0 & cn = cMin & sn >= 0 & sn^2 = 1-cn^2 & (yn-l/tG*sn)*sn = (xn+l/tG*cn)*cn & yn >= l/tG*sn))
            -> ((abs(cn*(xo-xn) - sn*(yo-yn)) > (W/2)) | (abs(sn*(xo-xn) + cn*(yo-yn) - (L/2)) > (L/2))))
    )
    ->
        [{
            {{p' = 0, v' = a, s' = c*v*tG/l, c' = -s*v*tG/l, x' = v*s, y' = v*c & p = 0 & c >= cMin & (mu*g)^2 >= (a)^2+(v^2*tG/l)^2 & v >= 0}
            ++ {p' = 1, x' = v*s, y' = v*c & p >=0 & c = cMin & (mu*g)^2 >= (a)^2 & v >= 0}}*
        }] ((abs(c*(xo-x) - s*(yo-y)) > (W/2)) | (abs(s*(xo-x) + c*(yo-y) - (L/2)) > (L/2)))
End.
```

KeYmaera X Tactic – With Acceleration

```
implyR(1) ; loop({`(p=0&c>=cMin()&s>=0&s^2=1-c^2&x=-l()/tG()*c&y=l()/tG()*s|p>=0&c=cMin()&s>=0&s^2=1-c^2&(y-l()/tG()*s)*s
=(x+l()/tG()*c)*c&y>=l()/tG()*s)&(p=0->y=l()/tG()*s)&(p=0->x=-l()/tG()*c)`}, 1) ; <(
  allL({`p`}, -8) ; allL({`x`}, -8) ; allL({`y`}, -8) ; allL({`s`}, -8) ; allL({`c`}, -8) ; QE,
  allL({`p`}, -8) ; allL({`x`}, -8) ; allL({`y`}, -8) ; allL({`s`}, -8) ; allL({`c`}, -8) ; QE,
  hideL(-8=={`\forall pn \forall xn \forall yn \forall sn \forall cn (pn=0&cn>=cMin()&sn>=0&sn^2=1-cn^2&xn=-l()/tG()*cn&y
n=l()/tG()*sn|pn>=0&cn=cMin()&sn>=0&sn^2=1-cn^2&(yn-l()/tG()*sn)*sn=(xn+l()/tG()*cn)*cn&yn>=l()/tG()*sn->abs(cn*(xo()-xn)
-sn*(yo()-yn))>W()/2|abs(sn*(xo()-xn)+cn*(yo()-yn)-L()/2)>L()/2)`}) ; hideL(-2=={`L()>0`}) ; hideL(-2=={`W()>0`}) ; hideL
(-3=={`vo()>0`}) ; hideL(-4=={`l()<=lMax()`}) ; hideL(-6=={`lMax()>0`}) ; hideL(-6=={`lMax()>=lMin()`}) ; hideL(-7=={`tG
()<=tGMax()`}) ; hideL(-12=={`tGMax()>0`}) ; hideL(-12=={`tGMax()>=tGMin()`}) ; choiceb(1) ; andR(1) ; <(
    MR({`p=0&c>=cMin()&s>=0&s^2=1-c^2&x=-l()/tG()*c&y=l()/tG()*s`}, 1) ; <(
      ODE(1),
      QE
      ),
    MR({`(p>=0&c=cMin()&s>=0&s^2=1-c^2&(y-l()/tG()*s)*s=(x+l()/tG()*c)*c&y>=l()/tG()*s)&(p=0->y=l()/tG()*s)&(p=0->x=-l()/
tG()*c)`}, 1) ; <(
      ODE(1),
      QE
      )
    )
  )
```

Future Work

- Complete and prove the equivalence model for the rear-wheel-drive model.
- Complete and prove the implicit model for the front-wheel-drive model.
- Complete and prove the equivalence model for the front-wheel-drive model.
- Introduction of uncertainty to the Bicycle Model.
- Sophistication of Bicycle Model to have a changing radius of turn.

Questions