ETH zürich

Zürich Autonomous Race Car

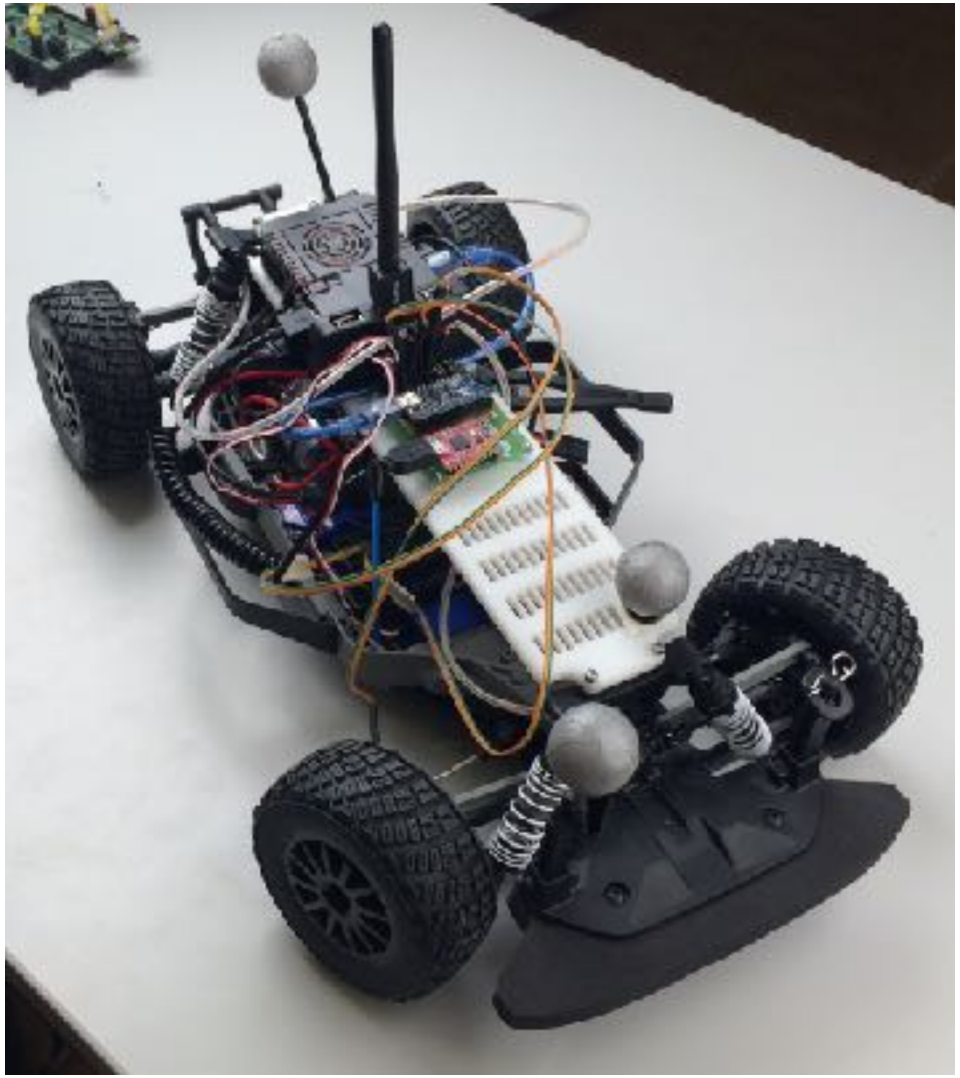
Brett Stephens

Professor Emilio Frazzoli

Supervisor: Edo Jelavic

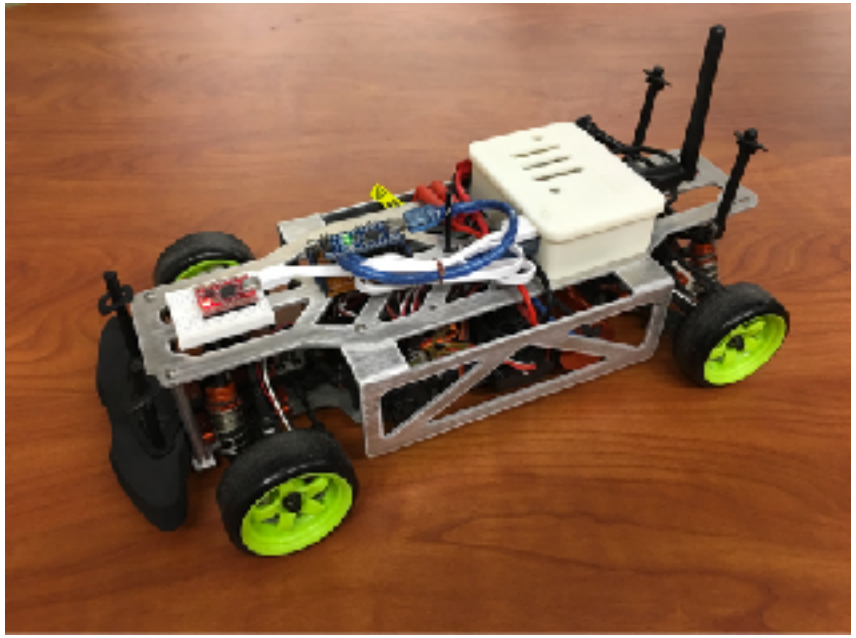
Presentation Overview

- Project Background
- Mechanical Design
- System Components
    - Odroid, Arduino, sensors
- System Architecture
- Vehicle Dynamics
    - System Identification
- Future Work
- Demonstration



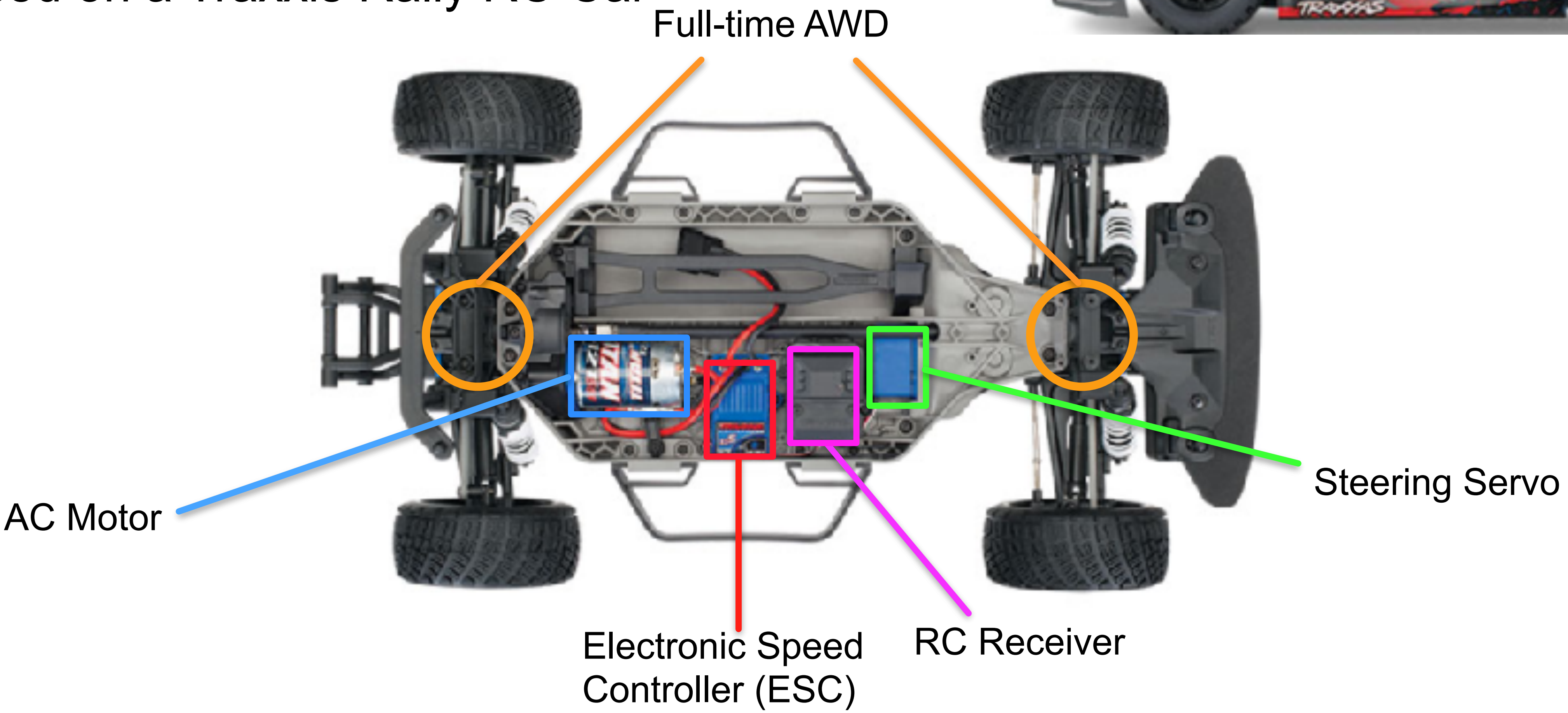
Project Background

- Based on BARC (UC Berkeley).
- Allows for the development of a variety of autonomous driving features on a compact, easy to use platform.
- Quick implementation of concepts.
- Low cost.
- Use of familiar tools - Ubuntu OS with Robot Operating System (ROS).

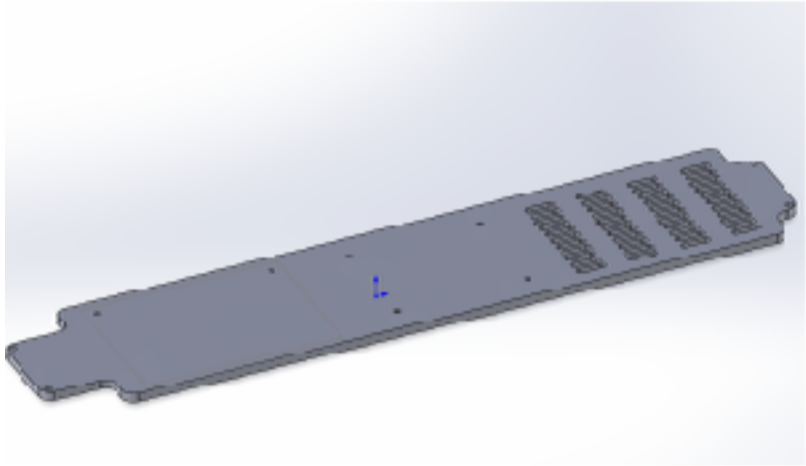


Mechanical Design

- Based on a Traxxis Rally RC Car

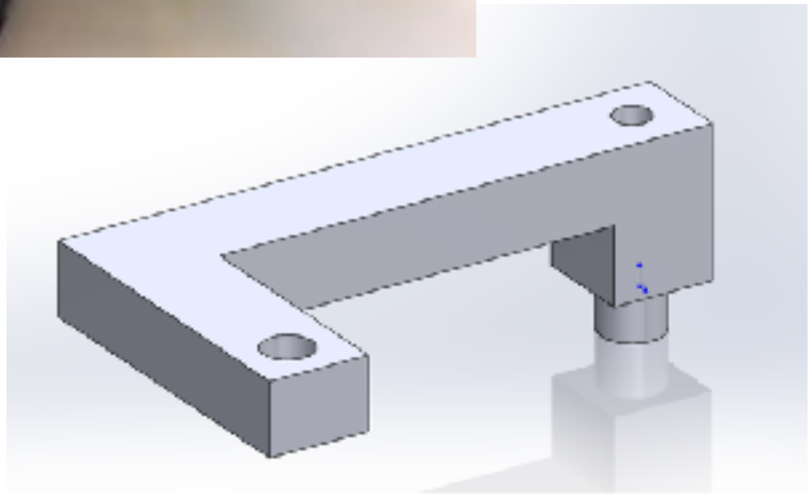


Mechanical Design



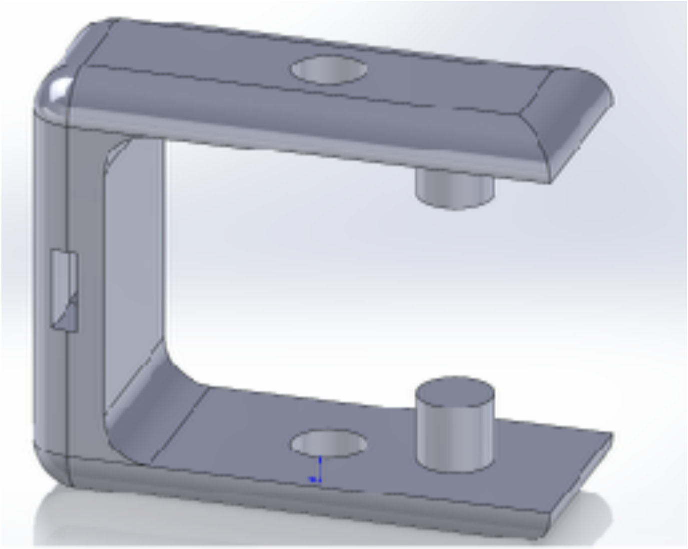
Top Platform

- Fixture for Odroid, Arduino, myAHRS+ and USB extension hub.



Hall Effect Sensor Mounts

- Mounted on each wheel hub.
- Paired with four magnets per wheel.

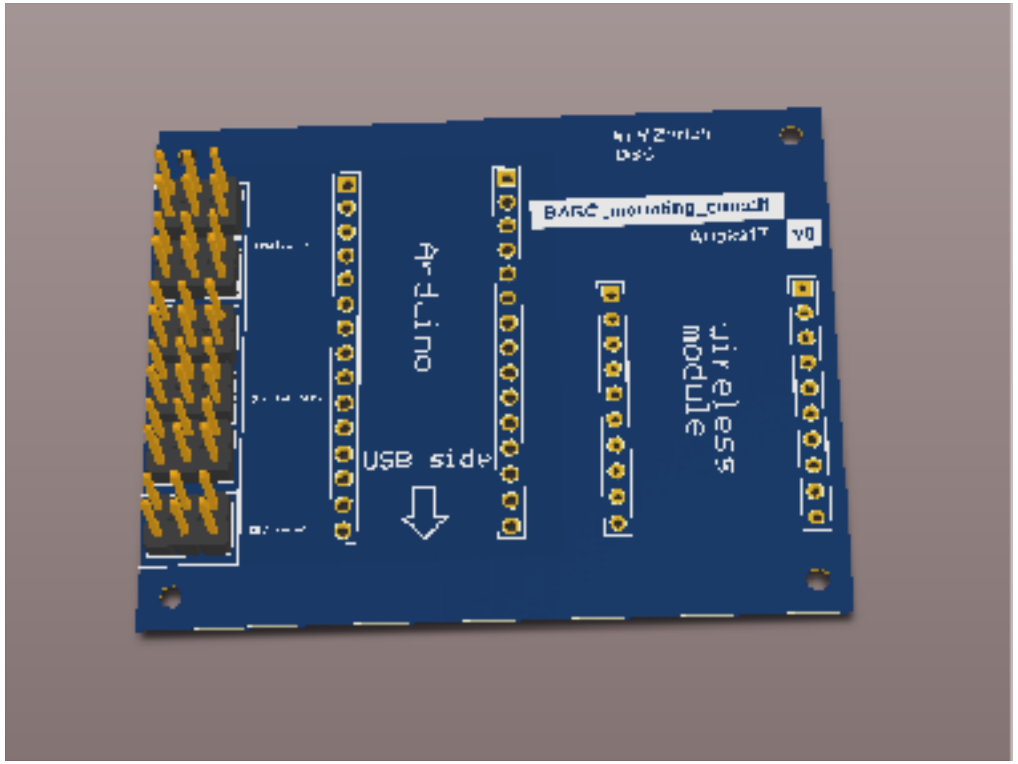


Rear

Front

Mechanical Design

- Mounting circuit for the Arduino and IMU
- Easy pin-out locations
- Allows for clean assembly of incoming sensor and actuator cables



System Components

- Odroid XU4
  - Running Ubuntu 16.04.
  - 2 GB RAM.
  - 16 GB eMMC storage.
  - 8 core ARM processor.
  - High level computing done here.

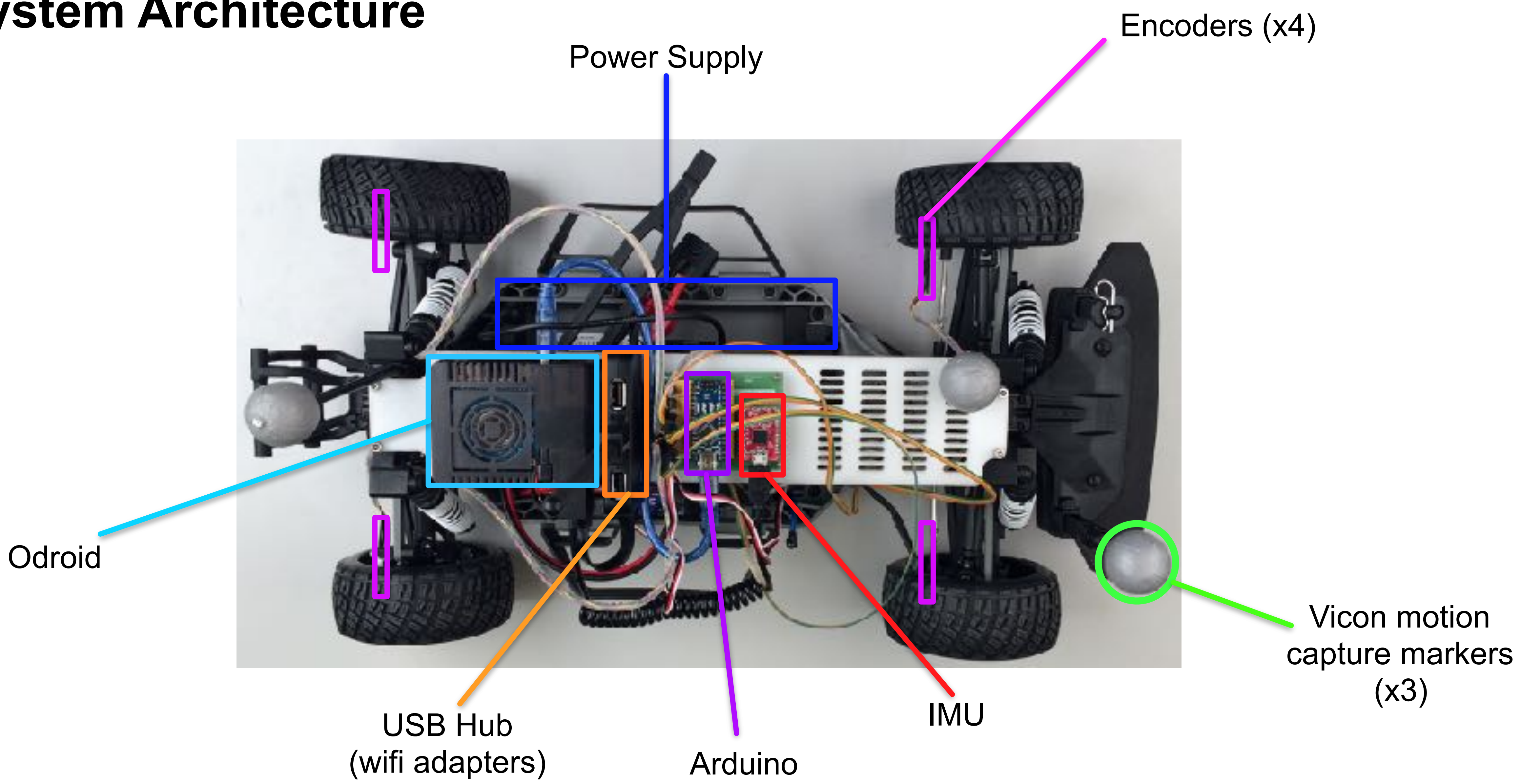
- Arduino Nano
  - Hardware interface between Odroid and sensors.
  - Communication with sensors/ actuators: steering servo, ESC, encoders.
  - Sends analog and digital sensor readings to the Odroid via USB.

System Components

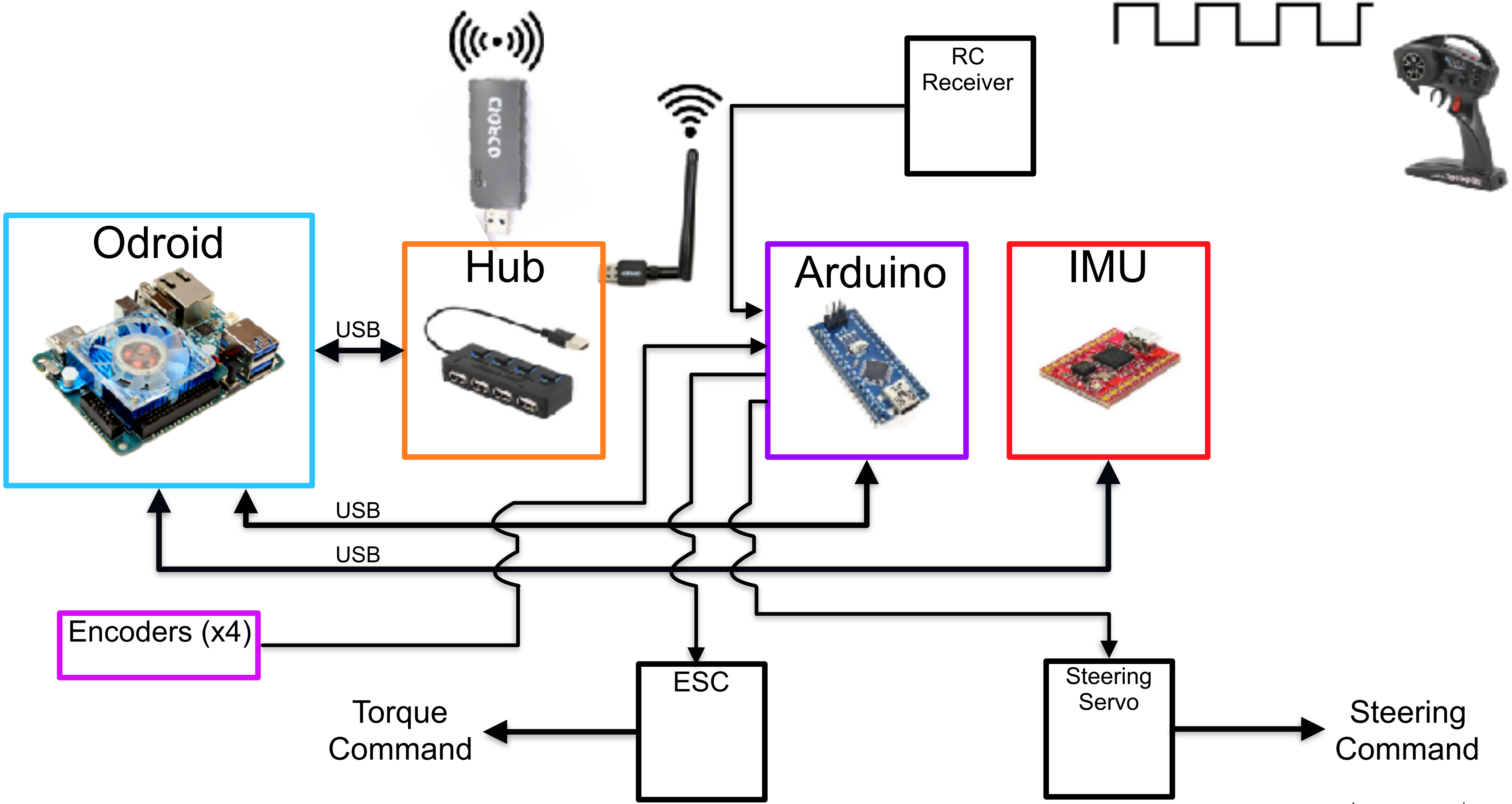
- myAHRS+ Inertial Measurement Unit (IMU)
  - Includes a magnetometer, accelerometer and gyroscope.
  - Interfaces with Odroid via USB.

- Encoders
  - Composed of hall effect sensor and magnets fixed on the inside of each wheel.
  - Four counts per wheel rotation.
  - Interface with arduino via digital pins.

System Architecture



System Architecture



Communication Architecture

ROS

Odroid
- Publishes:
  - ECU PWM
  - …
- Subscribes:
  - RC inputs
  - encoder count
  - IMU signals
  - …

Arduino
- Interface between Odroid and sensors, actuators
- Publishes:
  - RC inputs
  - encoder count
  - …
- Subscribes:
  - ECU PWM
  - …

IMU
- Publishes:
  - orientation
  - linear acc.
  - angular velocity
  - magnetic field
  - temperature

Odroid

Arduino

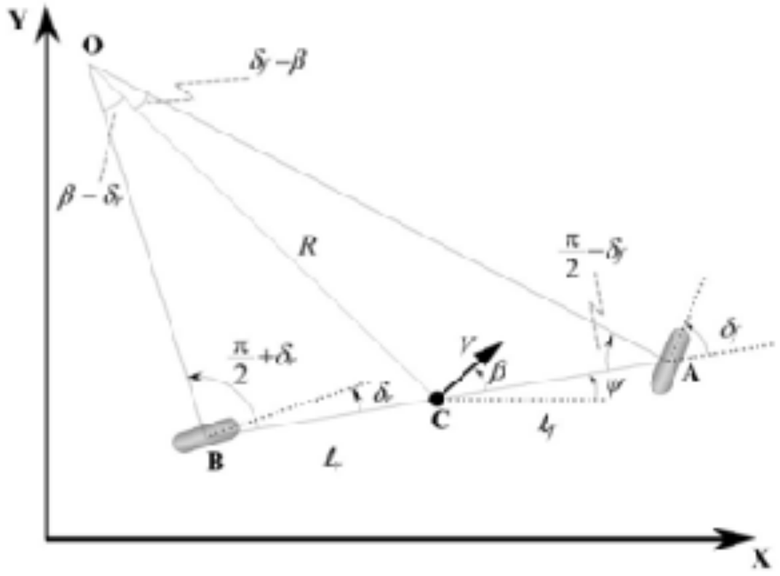
IMU

USB

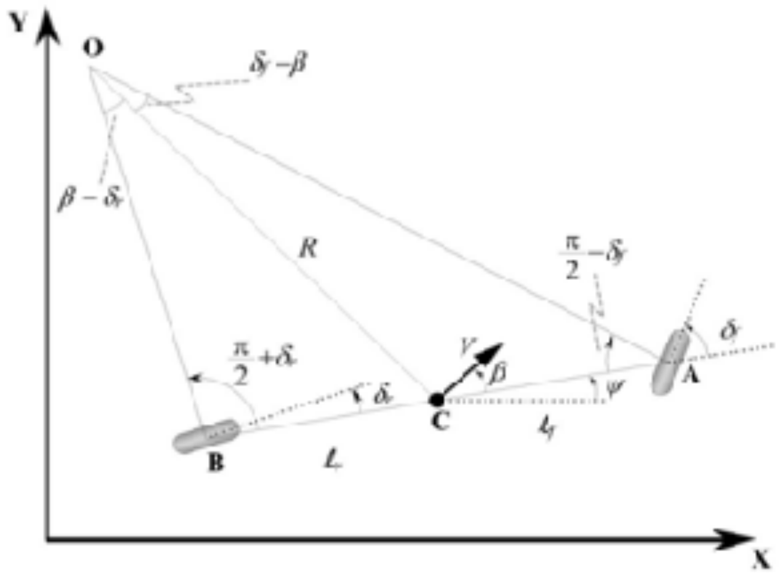
USB

Steering Mapping

- In order to reliably control the vehicle, the conversion from a pulse width modulated (PWM) servo input to a steering angle must be understood
- Use of the kinematic bicycle model - assuming low, uniform vehicle speed



Steering Mapping



$$\frac{sin(\delta_f - \beta)}{l_f} = \frac{sin(\pi/2 - \delta_f)}{R} \tag{1.1}$$

$$\frac{sin(\beta)}{l_r} = \frac{sin(\pi/2)}{R} \tag{1.2}$$

$$\frac{sin(\delta_f)cos(\beta) - sin(\beta)cos(\delta_f)}{l_f} = \frac{cos(\delta_f)}{R} \tag{1.3}$$

$$\frac{sin(\beta)}{l_r} = \frac{1}{R} \tag{1.4}$$

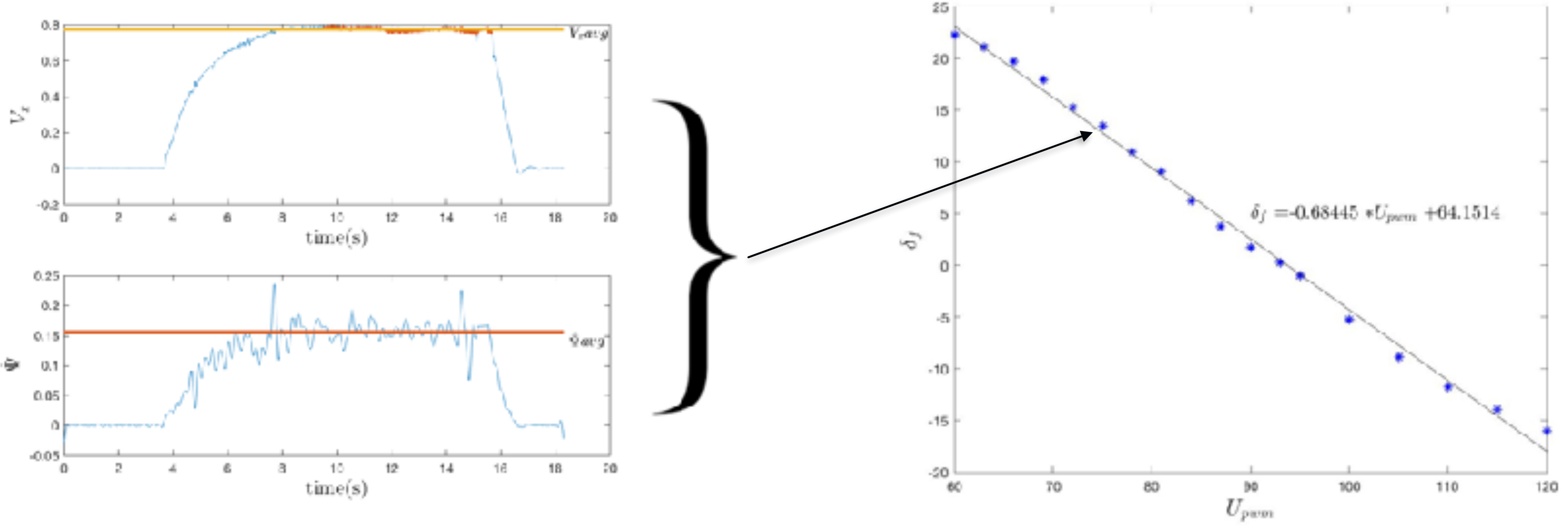
$$tan(\delta_f)cos(\beta) = \frac{l_r + l_f}{R} \tag{1.5}$$

$$\dot{\Psi} = \omega = \frac{V}{R} \tag{1.6}$$

$$\boxed{\delta_f = \arctan\left(\frac{\dot{\Psi}(l_r + l_f)}{V}\right) \qquad (1.7)}$$

Steering Mapping

- Keeping a low constant velocity, the parameters of (1.7) were measured with the Vicon motion capture system for a variety of constant steering angles δ_f
- For each δ_f, the gathered parameter data was averaged

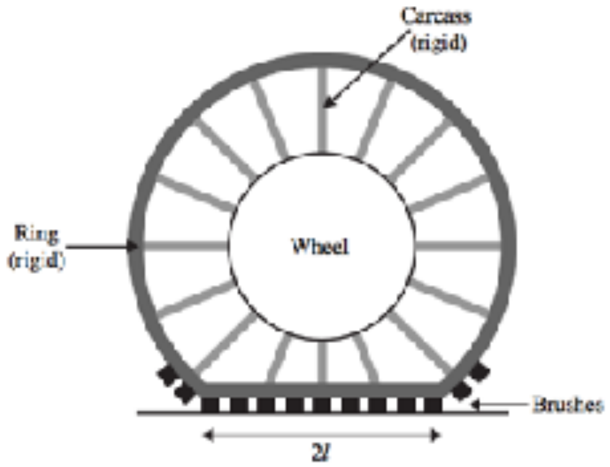


Tire Parameters

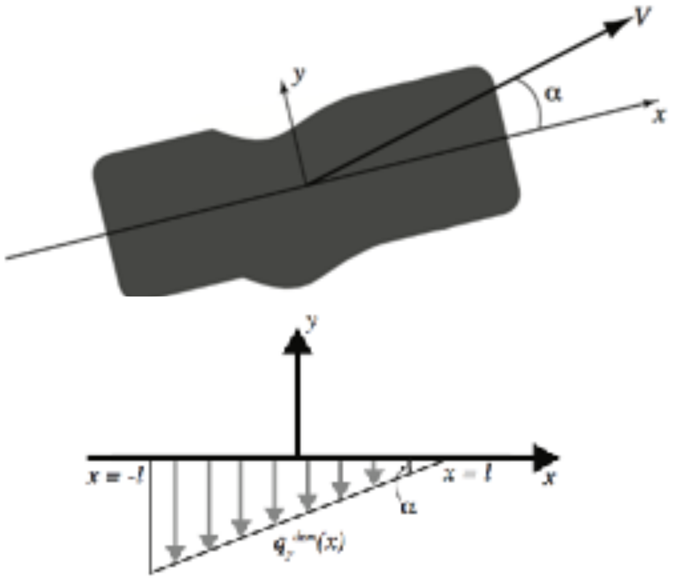
- Two approaches
  - Analytical tire model (Brush tire model).
  - Empirical tire model (Pacejka "magic formula" tire model).
- Both require data to fit model to: "ramp steer" test
  - Vehicle velocity is maintained at a constant rate while steering angle is linearly (and slowly) increased.
  - Use model of choice to fit the data. Model is characterized by tire parameters

Tire Parameters

Brush tire model:



Tire slip angle:

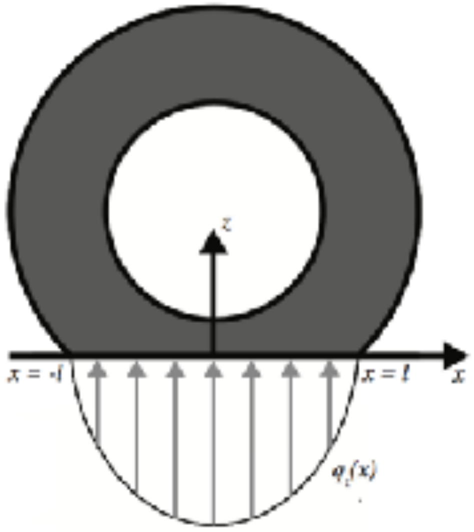


$$q_y^{dem}(x) = -c_{py}(a - x)\tanh\alpha \tag{2.11}$$

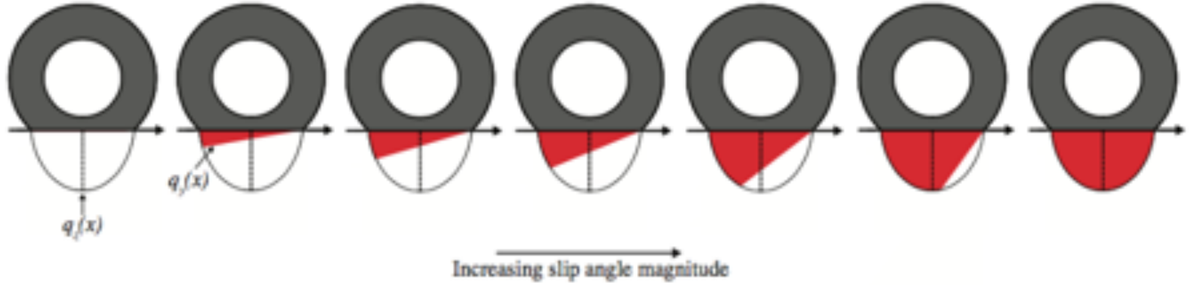
* model and figures as proposed by Hindiyeh [1]

Tire Parameters

Force available within contact patch area?



$$q_z(x) = \frac{3F_z}{4l}\left(\frac{l^2 - x^2}{l^2}\right), \tag{2.12}$$



$$|q_y(x)| \leq \mu q_z(x) \tag{2.13}$$

Behind the intersection of q_z and q_y, the contact patch is friction limited in that $q_y = \mu q_z$. This portion of the contact patch is now saturated.

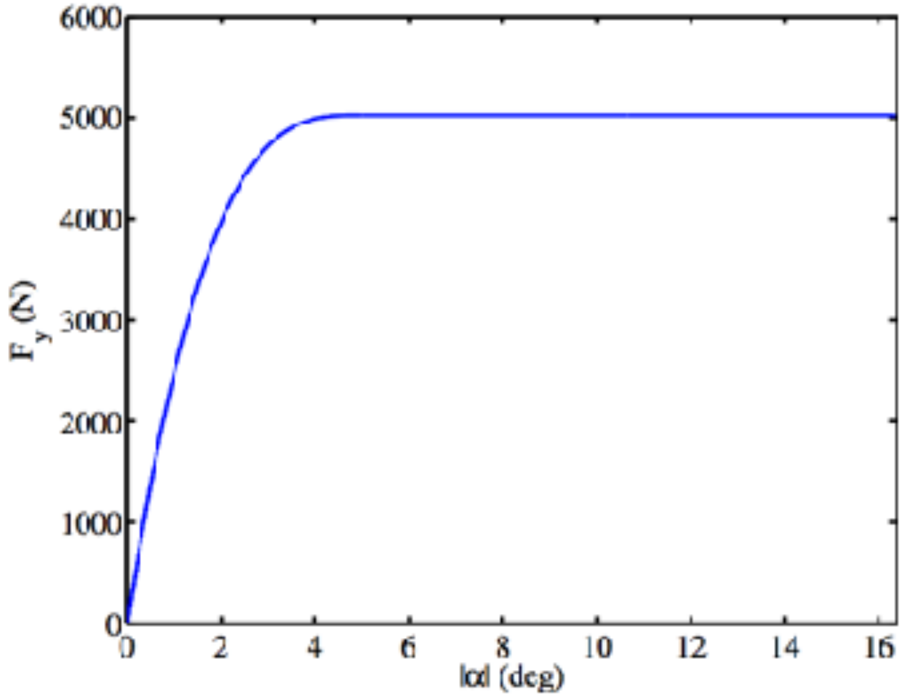
* model and figures as proposed by Hindiyeh [1]

Tire Parameters

- To arrive at an analytical model describing the lateral tire force as a function of slip angle, $q_y(x)$ can be integrated across the contact patch (characterized by α_{sl} and the “cornering stiffness” C_α).

$$F_y = \begin{cases} -C_\alpha \tan\alpha + \frac{C_\alpha^2}{3\mu F_z}|\tan\alpha|\tan\alpha - \frac{C_\alpha^3}{27\mu^2 F_z^2}\tan^3\alpha, & \text{if } |\alpha| \le \alpha_{sl}, \\ -\mu F_z sgn\alpha, & \text{otherwise} \end{cases} \quad (2.14)$$

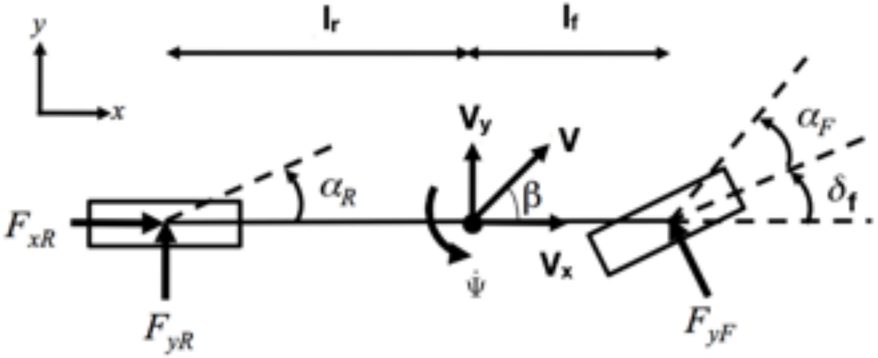
$$\alpha_{sl} = \arctan\frac{3\mu F_z}{C_\alpha} \quad (2.15)$$



* model and figures as proposed by Hindiyeh [1]

Tire Parameters

- To characterize the brush tire model, a dynamic bicycle model can be utilized to calculate the lateral force Fy and slip angle α_{sl}.



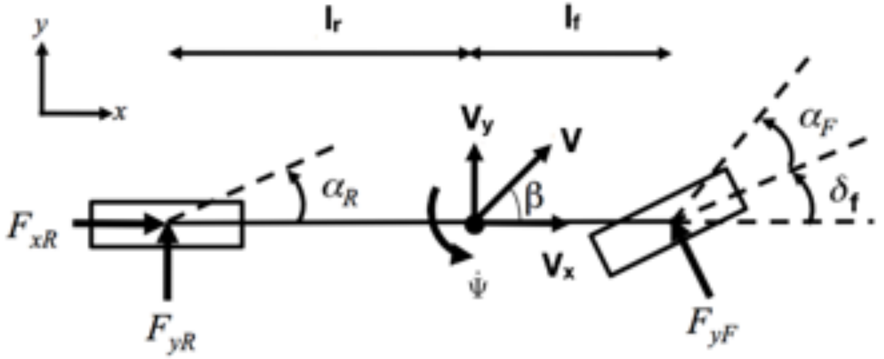
$$ma_y = F_{yF}\cos\delta_f + F_{yR} \qquad (2.1)$$

$$I_z\ddot{\Psi} = l_f F_{yF}\cos\delta_f - l_r F_{yR} \qquad (2.2)$$

$$a_y = \dot{V}_y + \dot{\Psi}V_x \qquad (2.3)$$

* model and figures as proposed by Hindiyeh [1]

Tire Parameters



$$\dot{V}_y = \frac{F_{yF} + F_{yR}}{m} - \dot{\Psi} V_x \qquad (2.4)$$

$$\ddot{\Psi} = \frac{l_f F_{yF} - l_r F_{yR}}{I_z} \qquad (2.5)$$

Small angle approximation, $\beta = \arctan \frac{V_y}{V_x} \approx \frac{V_y}{V_x}$
and $\dot{\beta} \approx \frac{\dot{V}_y}{V_x}$. (2.4) updated to:

$$\dot{\beta} = \frac{F_{yF} + F_{yR}}{mU_x} - \dot{\Psi} \qquad (2.6)$$

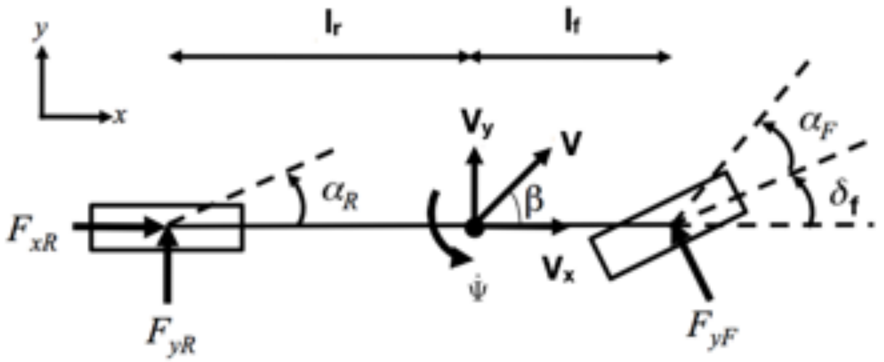
Steady state cornering: $\dot{\beta} = \ddot{\Psi} = 0$.
Combining (2.5) and (2.6) with $a_y^{SS} = \dot{\Psi} V_x$:

$$\boxed{F_{yF} = \frac{l_r}{L} m a_y^{SS} \qquad (2.7)}$$

$$\boxed{F_{yR} = \frac{l_f}{L} m a_y^{SS} \qquad (2.8)}$$

* model and figures as proposed by Hindiyeh [1]

Tire Parameters



- With (2.7-2.10), it is possible to plot a lateral force vs slip angle curve with the given vehicle parameters:

$$\alpha_F = \arctan \frac{V_y + l_f \dot{\Psi}}{V_x} - \delta_f \approx \arctan\left(\beta + \frac{l_f}{V_x}\dot{\Psi}\right) - \delta_f \qquad (2.9)$$

$$\alpha_R = \arctan \frac{V_y - l_r \dot{\Psi}}{V_x} \approx \arctan\left(\beta - \frac{l_r}{V_x}\dot{\Psi}\right) \qquad (2.10)$$

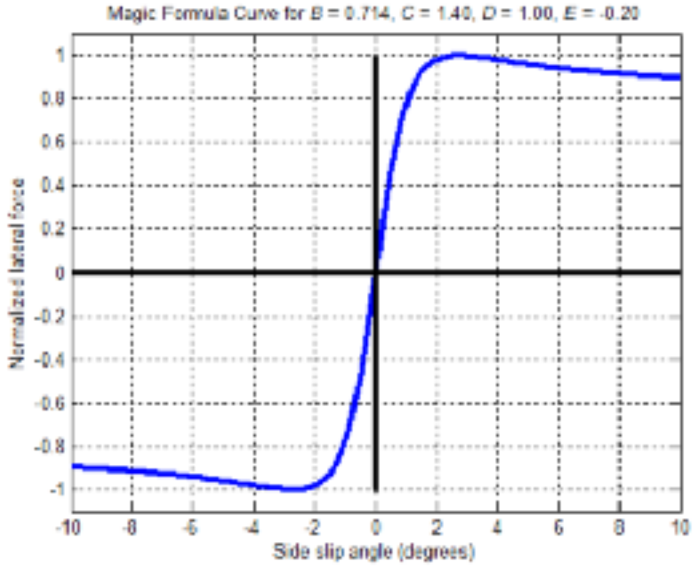
Vehicle Parameters	
Parameter	Value
F_{zF}	12.6941 N
F_{rF}	14.7454 N
l_f	0.1741 m
l_r	0.1499 m
m	2.792 kg

* model and figures as proposed by Hindiyeh [1]

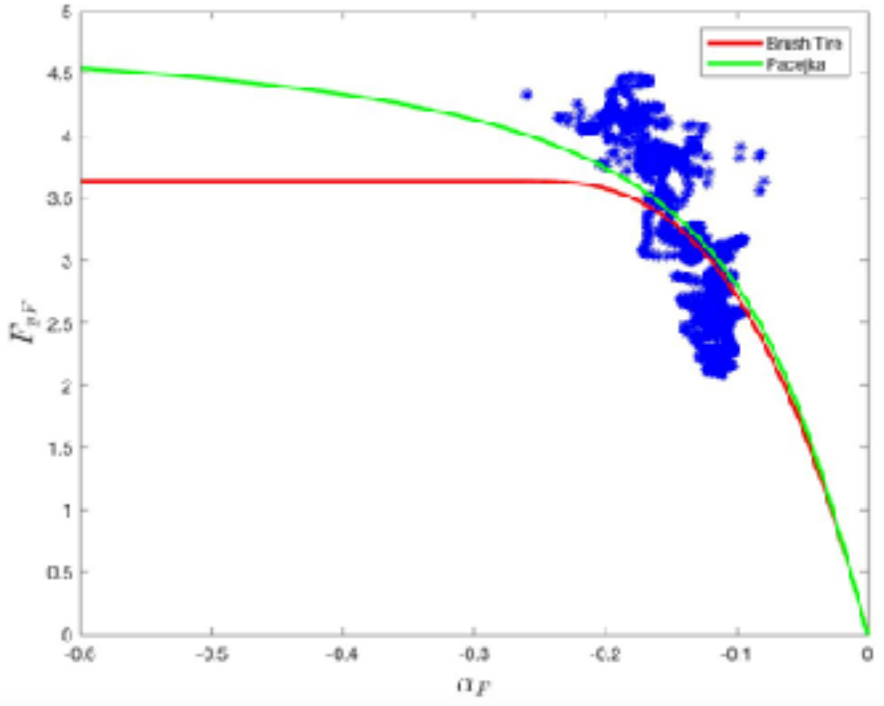
Tire Parameters

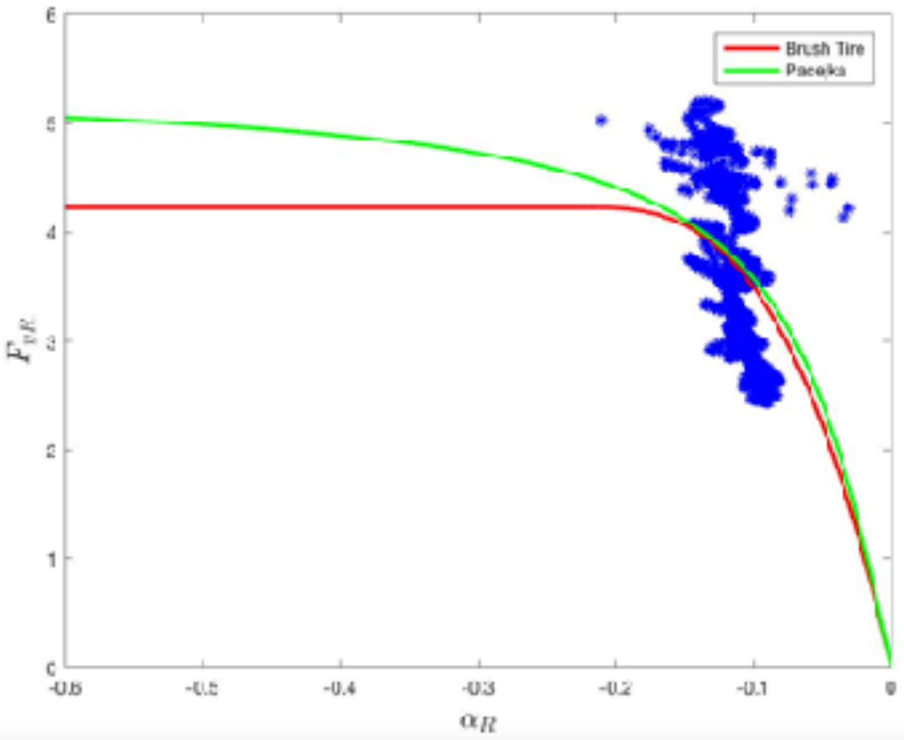
- Pacejka "magic formula"
  - Empirically derived
  - Tire is characterized by coefficients for forces it can produce at the contact patch
  - Coefficients are determined via a best fit line between the empirical formula and gathered data

$$R(k) = d \cdot \sin\{c \cdot \arctan[b\,(1-e)\,k + e \cdot \arctan(bk)]\}$$



Tire Parameters





Tire Model	
States	$\beta, \dot{\Psi}$
Input	δ_f

$$\dot{\beta} = \frac{F_{yF} + F_{yR}}{mU_x} - \dot{\Psi}$$

$$\ddot{\Psi} = \frac{l_f F_{yF} - l_r F_{yR}}{I_z}$$

Future Work

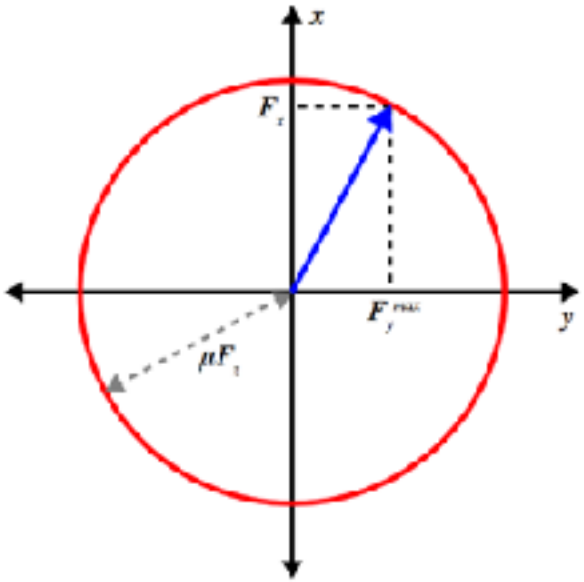
- Networking
- Encoder miscount
  - hardware fix - redesign hall sensor mounts
  - software fix (timing issue?)
- Additional sensors

Thank you.
Demo coming up!

References

- [1] Hindiyeh, Rami Yusef. “Dynamics And Control Of Drifting In Automobiles”. PhD Dissertation, Stanford University, 2013.

Appendix



$$\sqrt{F_x^2 + F_y^2} \leq \mu F_z.$$

$$F_y^{max} = \sqrt{(\mu F_z)^2 - F_x^2}.$$

$$F_y^{max} = \xi \mu F_z$$

$$\xi = \frac{\sqrt{(\mu F_z)^2 - F_x^2}}{\mu F_z}$$

$$F_y = \begin{cases} -C_\alpha \tan\alpha + \frac{C_\alpha^2}{3\xi\mu F_z} \mid \tan\alpha \mid \tan\alpha - \frac{C_\alpha^3}{27\xi^2\mu^2 F_z^2} \tan^3\alpha & |\alpha| \leq \alpha_{sl} \\ -\xi\mu F_z \text{sgn } \alpha & |\alpha| > \alpha_{sl} \end{cases}$$

$$\alpha_{sl} = \arctan\frac{3\xi\mu F_z}{C_\alpha}.$$

Appendix

$$ma_x = \frac{F_{xR} - F_{yF}\sin\delta}{m},$$

$$a_x = \dot{U}_x - rU_y \approx \dot{U}_x - rU_x\beta.$$

$$\dot{\beta} = \frac{F_{yF} + F_{yR}}{mU_x} - r$$

$$\dot{r} = \frac{aF_{yF} - bF_{yR}}{I_z}$$

$$\dot{U}_x = \frac{F_{xR} - F_{yF}\sin\delta}{m} + rU_x\beta.$$