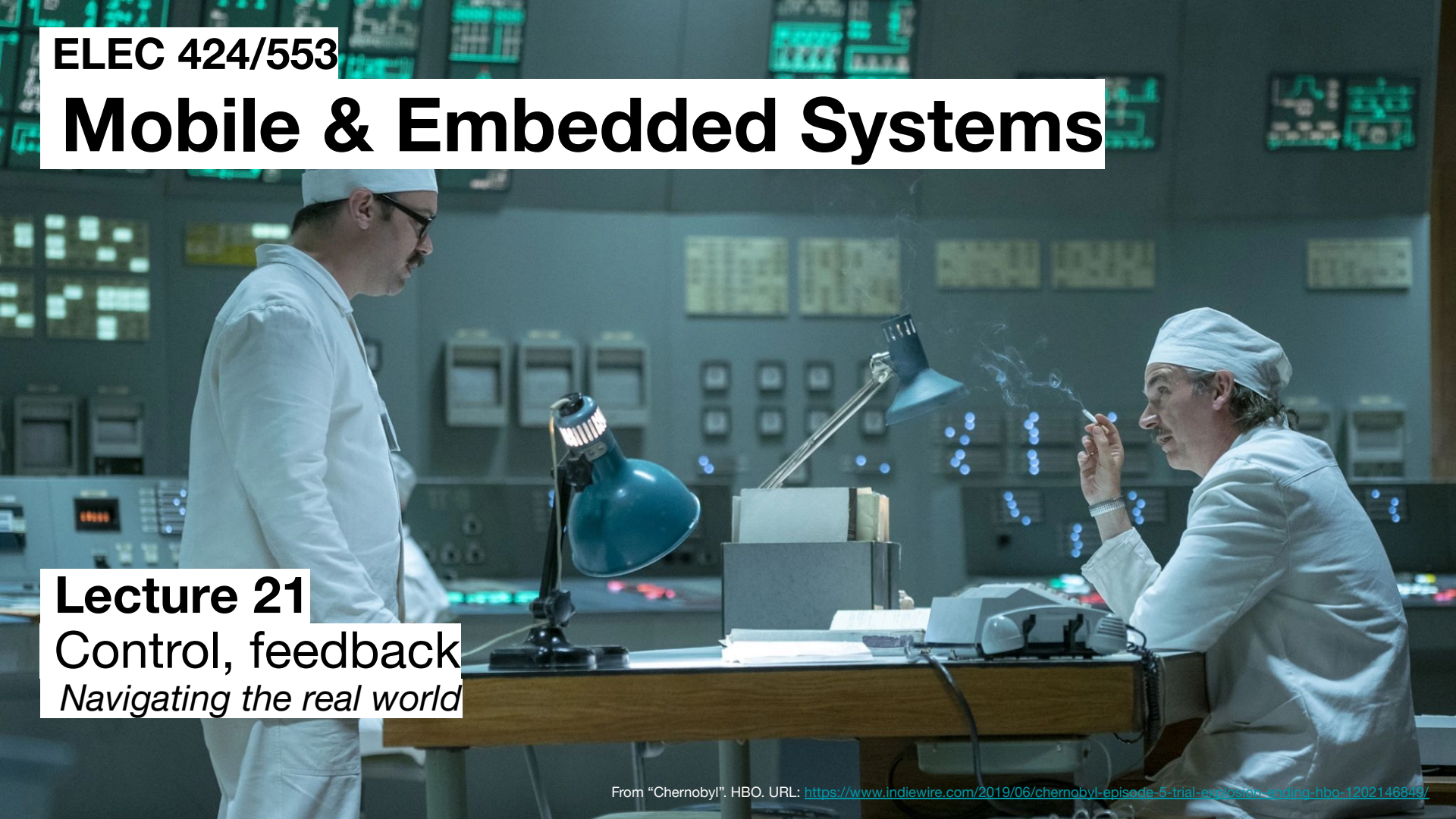
ELEC 424/553

# Mobile & Embedded Systems



## Lecture 21

Control, feedback

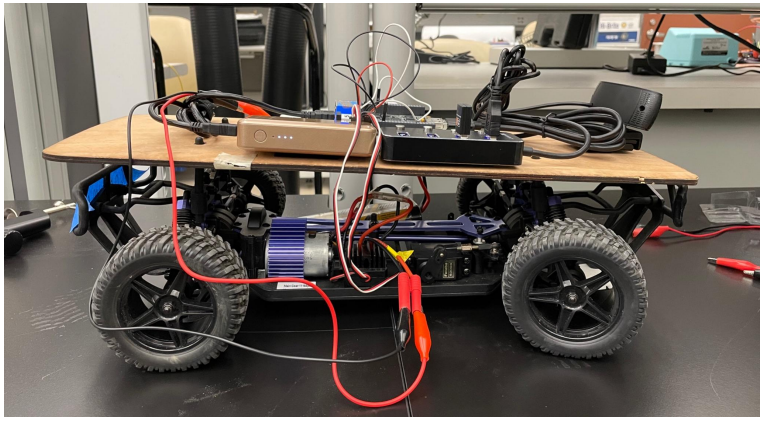
*Navigating the real world*

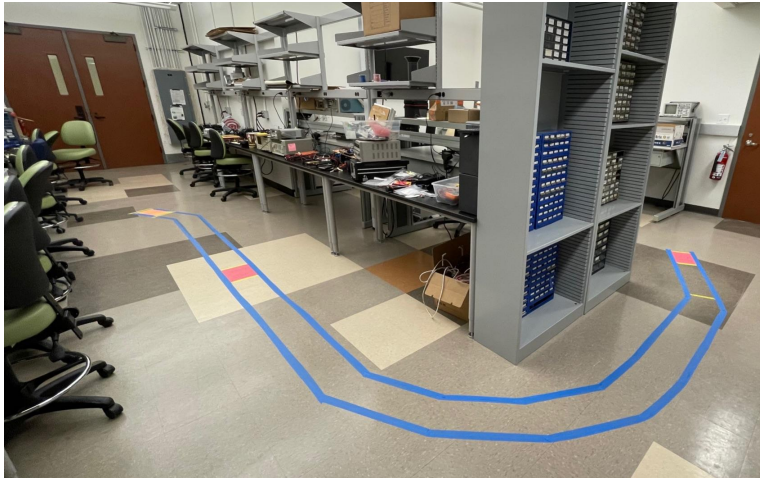
From “Chernobyl”. HBO. URL: https://www.indiewire.com/2019/06/chernobyl-episode-5-trial-explosion-ending-hbo-1202146849/

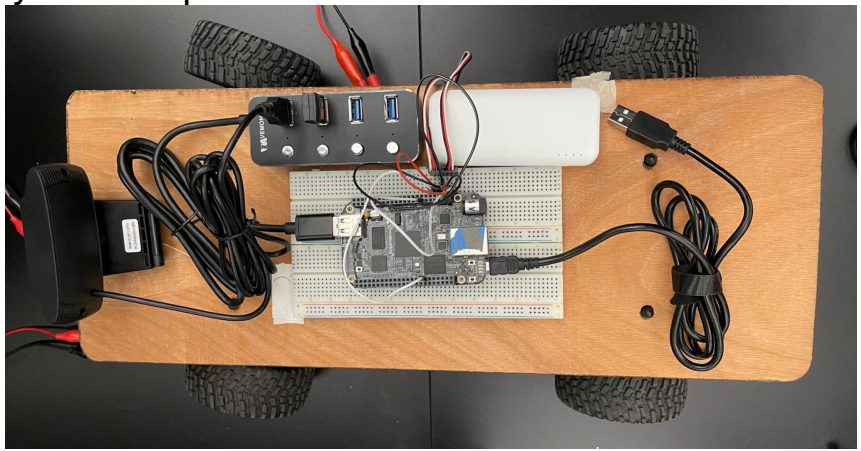
# What Is The Final Project?

Build a lane-keeping car using:

- RC car
- (undergraduate teams) RPi 4
- (graduate teams) Beaglebone AI-64
- Optical speed encoder
- Batteries (portable phone charger; 7.2 V battery)
- WiFi adapter
- Webcam
- Python & openCV



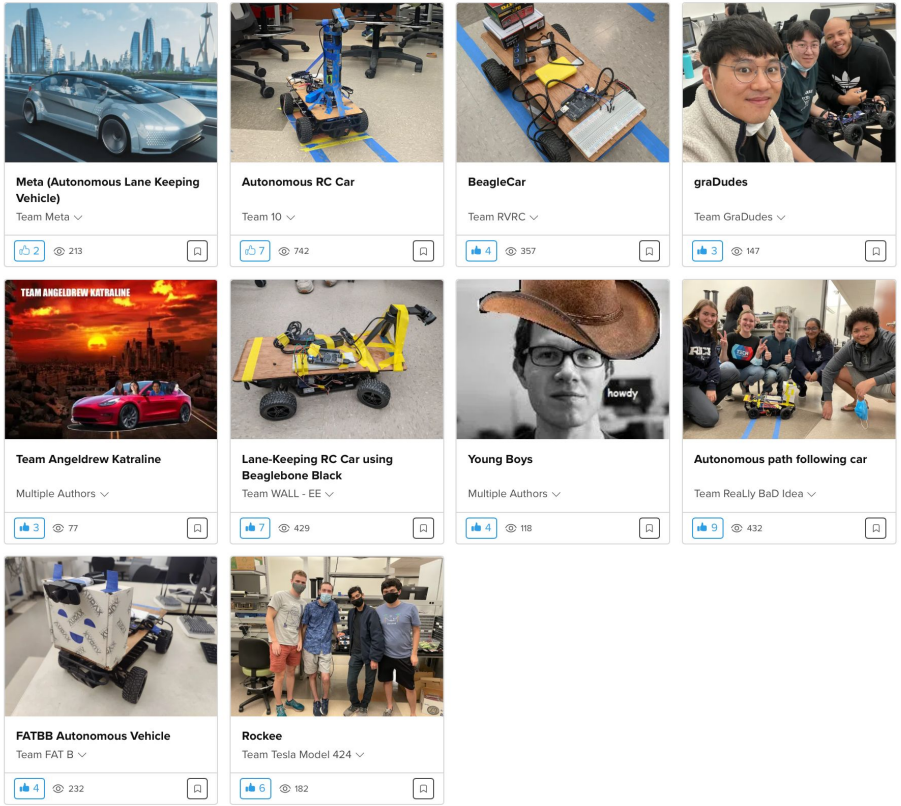




# Code From Last Year Available

## Code on Hackster

You can use that code liberally, but be sure to cite it in your code files and in your written submission



# Reliable Source: TikTok

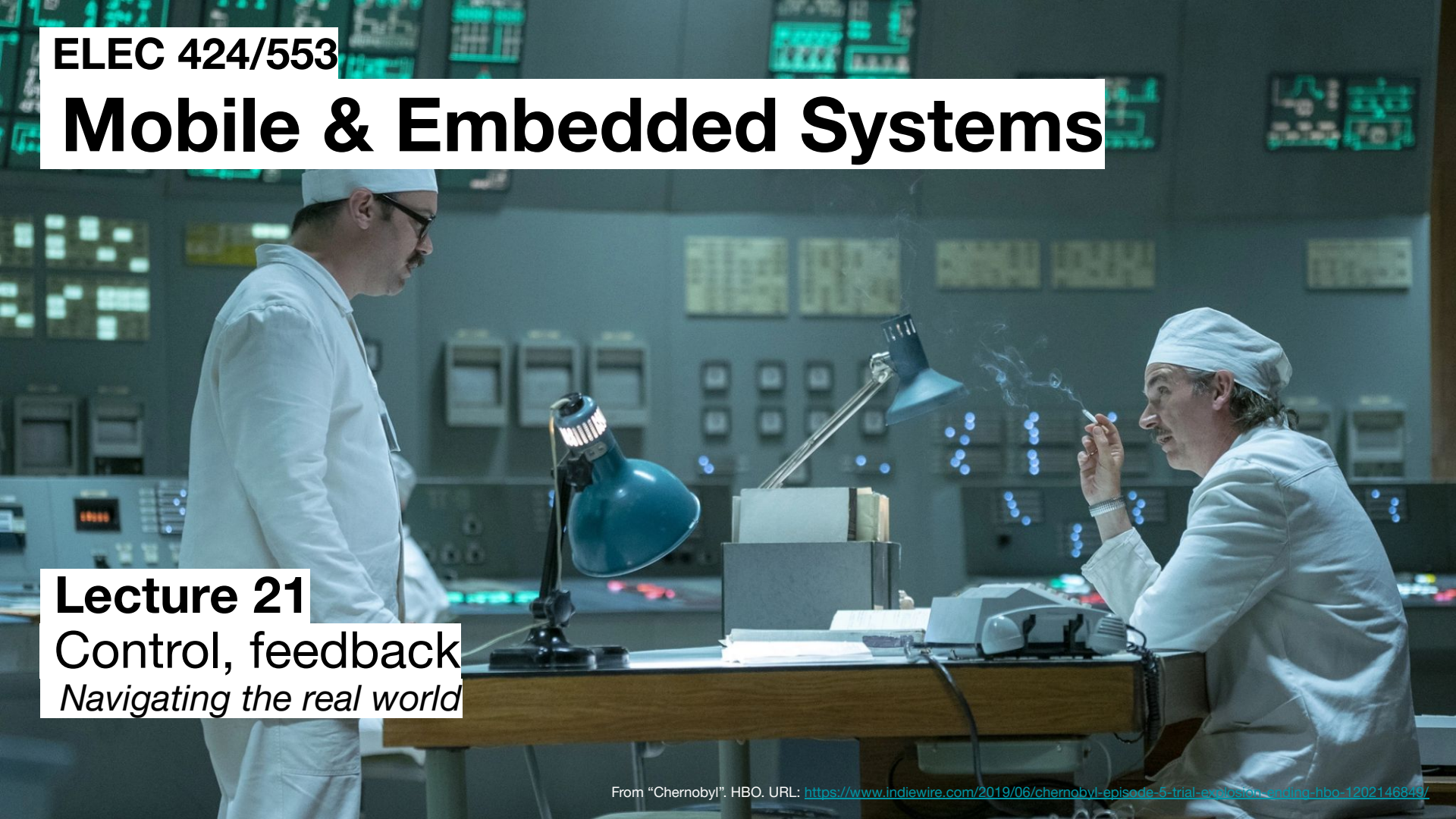
https://www.tiktok.com/@mamari_18/video/7041642065960226095?is_copy_url=0&is_from_webapp=v1&item_id=7041642065960226095

# Zoom Lecture On Monday 11/20

https://9gag.com/gag/aO7LWq2

ELEC 424/553

# Mobile & Embedded Systems



## Lecture 21

Control, feedback

*Navigating the real world*

From “Chernobyl”. HBO. URL: https://www.indiewire.com/2019/06/chernobyl-episode-5-trial-explosion-ending-hbo-1202146849/

# Control & Feedback

“Chernobyl scene #1: Valery Legasov explains, how an RBMK reactor works”. Channel: Verlin. Originally from HBO’s “Chernobyl”.
URL: https://www.youtube.com/watch?v=TmlEl4ky-Zc

# Technology Controversy In

## Tesla vehicle in 'Full Self-Driving' beta mode 'severely damaged' after crash in California

*It would appear to be the first crash involving Tesla's controversial new driver assist feature*

By Andrew J. Hawkins | @andyjayhawk | Nov 12, 2021, 10:09am EST

SHARE


https://www.theverge.com/2021/11/12/22778135/tesla-full-self-driving-beta-crash-fsd-california

ars TECH

"I KNOW LESS THAN

You sha
judge di

Judge: iPad pinch-to-zoom could "insert more items" into video of shootings.

JON BRODKIN - 11/11/2021, 3:02 PM

Getty Images | Pool

Enlarge / Judge Bruce Schroeder reprimands Assistant District Attorney Thomas Binger during cross-examination of Kyle Rittenhouse at the Kenosha County Courthouse in Wisconsin on November 10, 2021.

https://arstechnica.com/tech-policy/2021/11/rittenhouse-trial-judge-disallows-ipad-pinch-to-zoom-read-the-bizarre-transcript/

"I don't know what the state's going to do next, but I suspect that it's something along the lines of... they're going to use the iPad, and Mr. Binger was talking about pinching the screen. iPads, which are made by Apple, have artificial intelligence in them that allow things to be viewed through three dimensions and logarithms."

"[The iPad] uses artificial intelligence or their logarithms to create what they believe is happening."

Judge: "I know less than anyone in the room about all of this stuff"

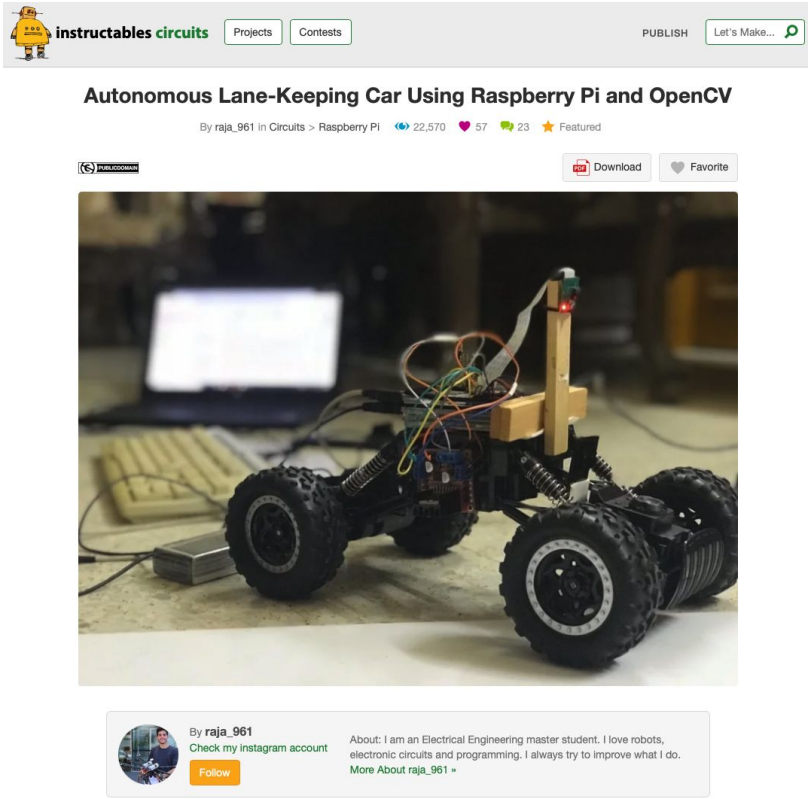
**spear** / Seniorius Lurkius JUMP TO POST

The funny thing is that what they ended up doing -- using a PC to display the video on a 4k TV -- very likely involved scaling of the picture by the video player software and/or the TV.

# We’ve Heard This Before

From YouTube channel: everymeme. URL: https://www.youtube.com/watch?v=lTonHRerMC4

# Making Our Own Mobile & Embedded System




https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/



User raja_961, “Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV”. https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/

# Pipeline of Image Processing From Instructable

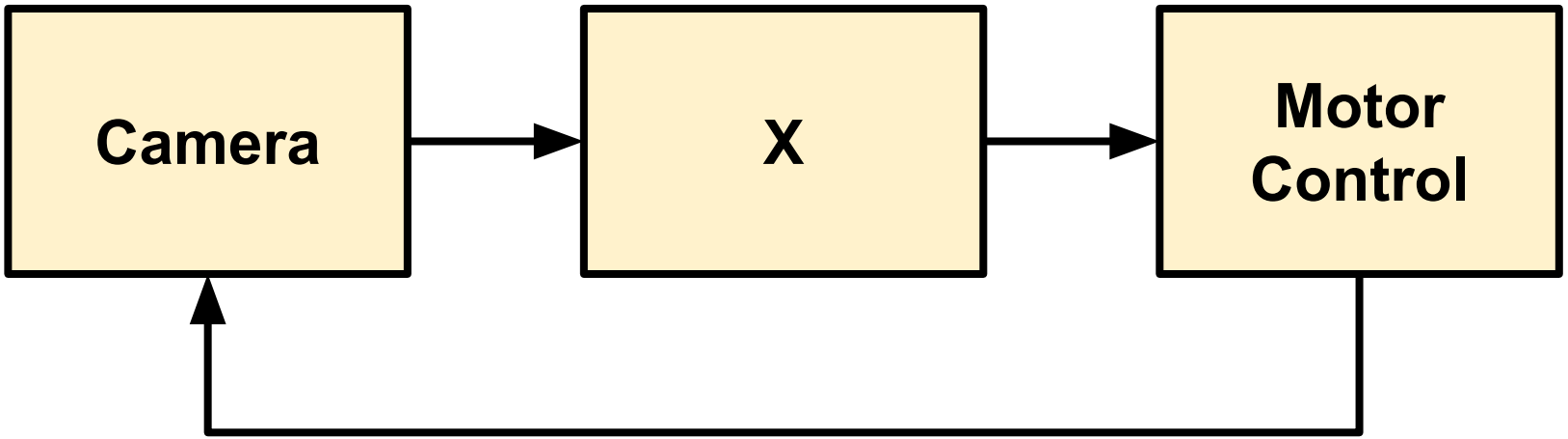



User raja_961, “Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV”. https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/


# What Do We Call This Approach

- Pipeline approach
- Tesla
- Likely ever other major player apart from Comma AI

# That's Our X



# Tesla's X & Comma AI's X

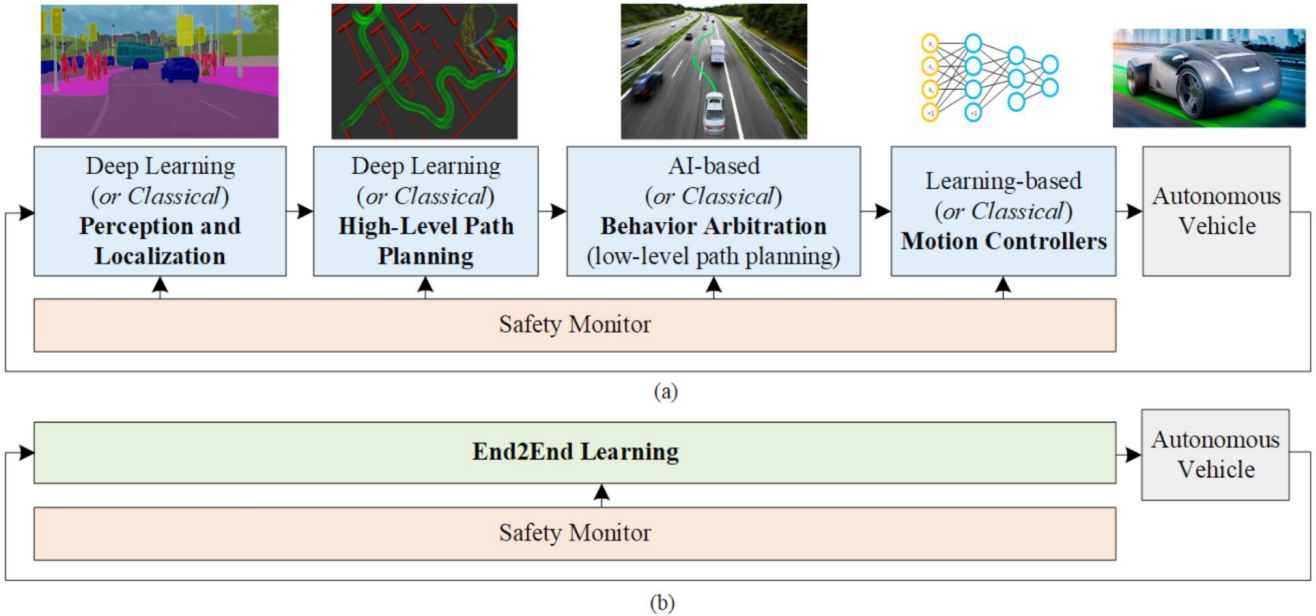
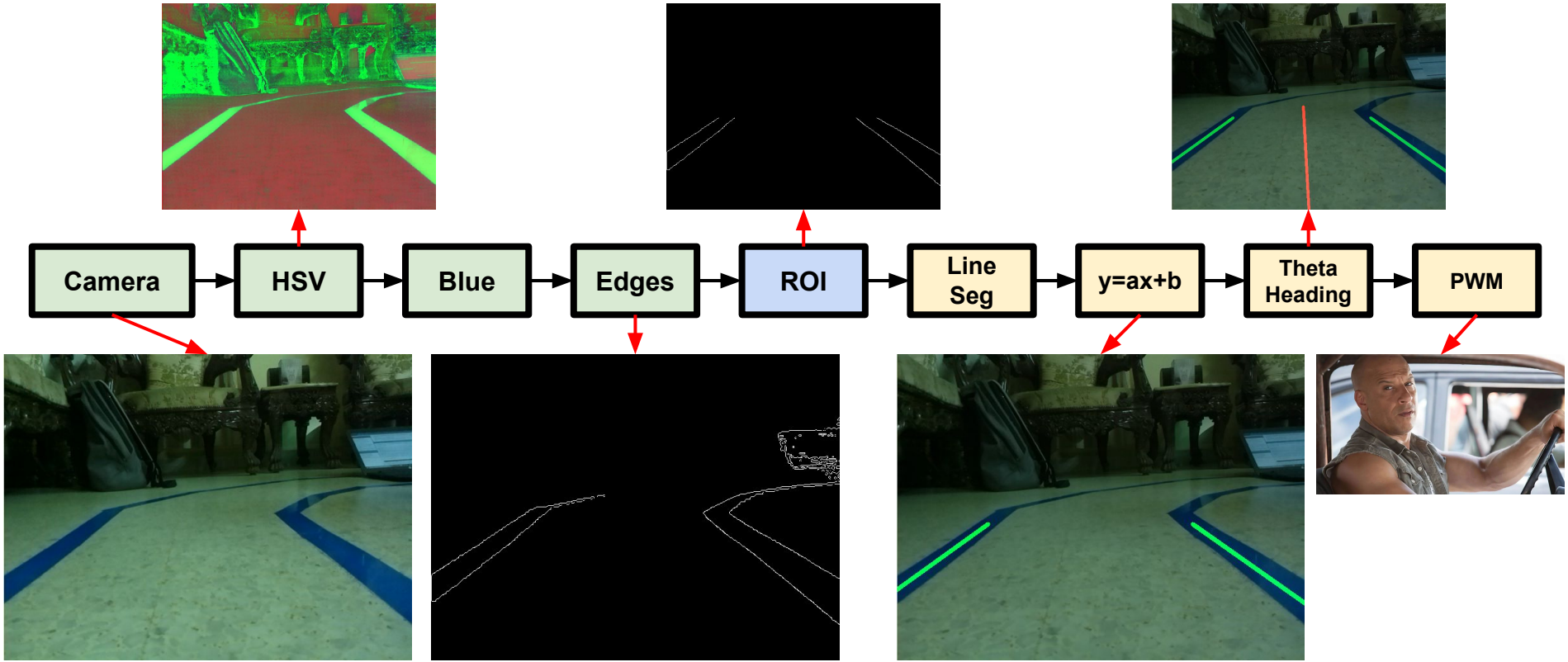



Figure 1: **Deep Learning based self-driving car**. The architecture can be implemented either as a sequential perception-planing-action pipeline (a), or as an End2End system (b). In the sequential pipeline case, the components can be designed either using AI and deep learning methodologies, or based on classical non-learning approaches. End2End learning systems are mainly based on deep learning methods. A safety monitor is usually designed to ensure the safety of each module.



Grigorescu et al. "A Survey of Deep Learning Techniques for Autonomous Driving". arXiv. (2020)

# Pipeline of Image Processing From Instructable

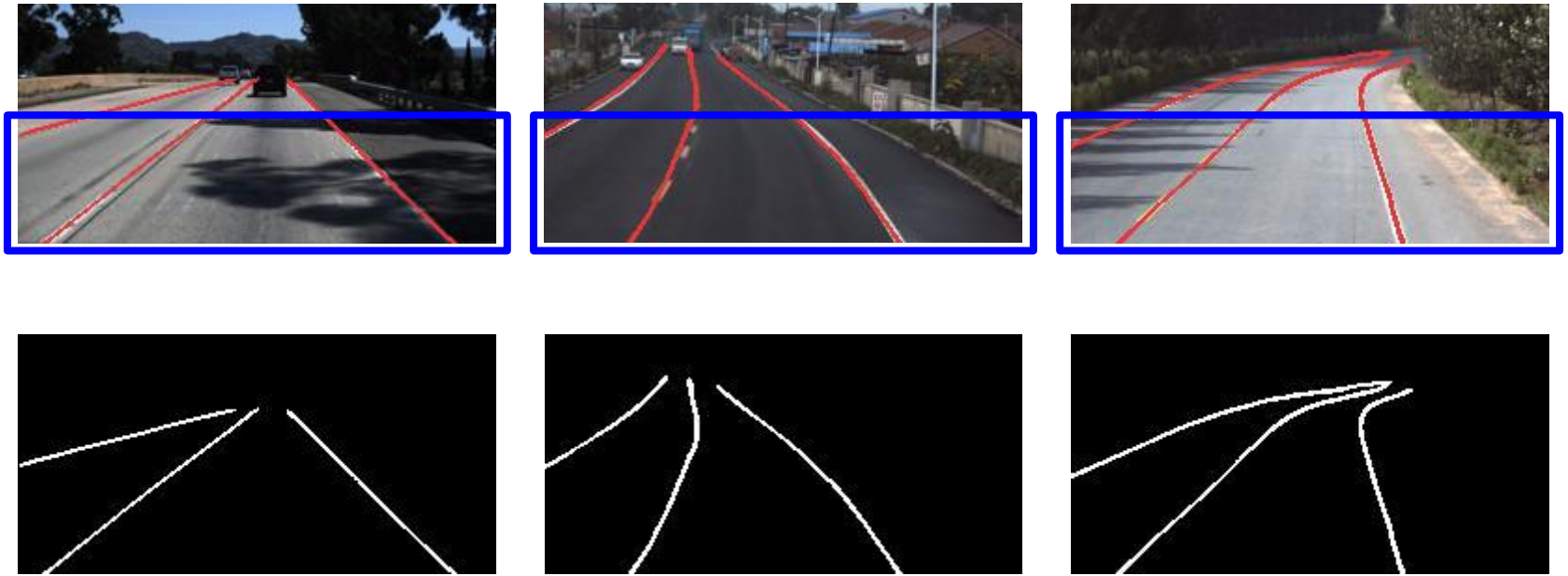





User raja_961, “Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV”. https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/
Dom picture from Fast & Furious franchise, Universal Pictures. URL: https://dashfun.net/fast-and-furious-9-vin-diesel-plans-a-spin-off-on-dominic-toretto/

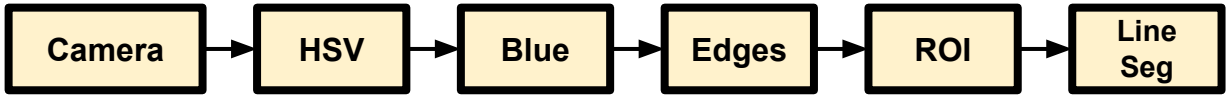

# ROI: Region of Interest

## Do We Need The Whole Scene?





Images from: “Robust Lane Detection From Continuous Driving Scenes Using Deep Neural Networks”. URL: https://github.com/qinnzou/Robust-Lane-Detection




# Line Segment Detection

- How do we infer and present a shape?
- We have edges
- But there are holes
- Python: **`cv2.HoughLinesP()`**
  - frame, rho, theta, min_threshold, minLineLength, maxLineGap
  - Returns line endpoints

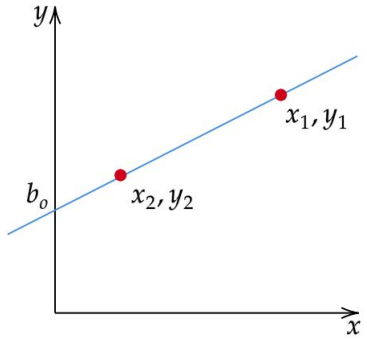
```
def detect_line_segments(cropped_edges):
    rho = 1
    theta = np.pi / 180
    min_threshold = 10
    line_segments =
        cv2.HoughLinesP(cropped_edges,
            rho,
            theta,
            min_threshold,
            np.array([]),
            minLineLength=5,
            maxLineGap=0)
    return line_segments
```



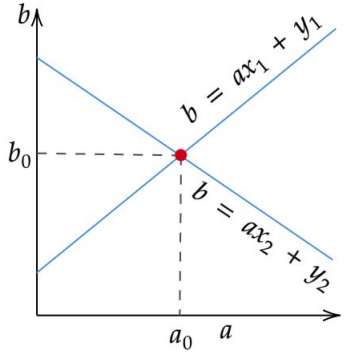
User raja_961, “Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV”. https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/

# Hough Transform: Locate Shapes

- Feature extraction
- Patented! (1962)
- Normally plot x and y for a line
  - Multiple points
  - One line
- Line: y=ax+b [y=mx+b]
- Can also plot line as “a” (slope; m) and “b”
  - Multiple lines
  - One point




**Hough (Parameter) Space**

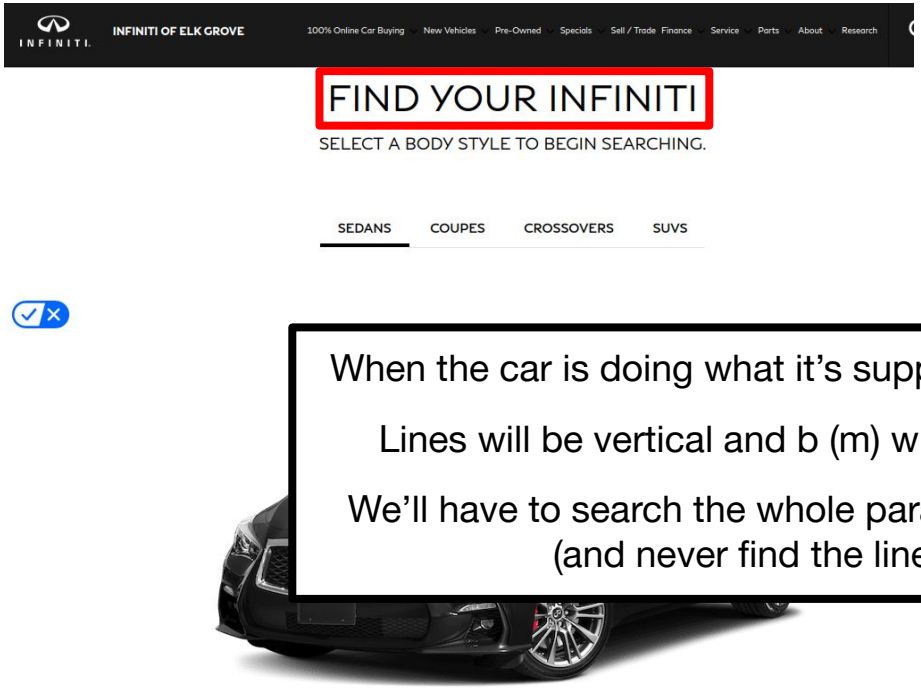



https://towardsdatascience.com/lines-detection-with-hough-transform-84020b3b1549



User raja_961, “Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV”. https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/

# Problem




When the car is doing what it's supposed to do …

Lines will be vertical and b (m) will be infinity

We'll have to search the whole parameter space (and never find the line)

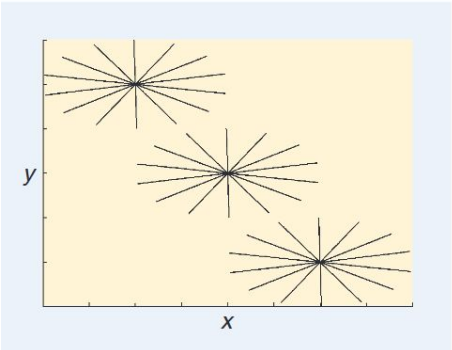


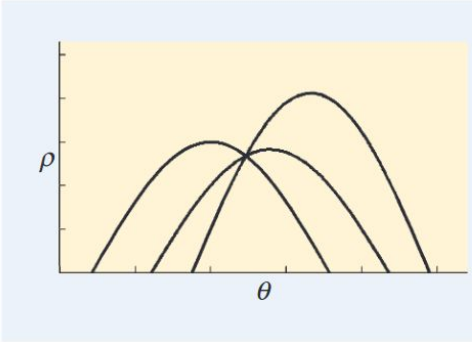
b = ?

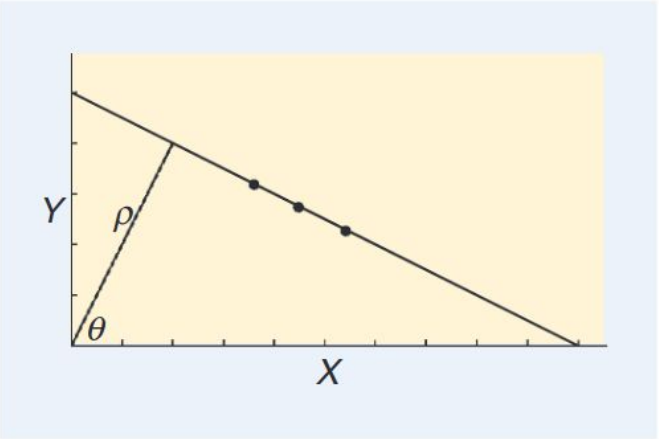
Image from: https://www.infinitiofelkgrove.com/

# Rho & Theta

- Notice **`cv2.HoughLinesP()`** takes rho & theta rather than slope and intercept
  - This function concerned with precision of rho and theta, not actual rho and theta values
- Radius and angle
- $y=-x\cos\theta/\sin\theta + \rho/\sin\theta$



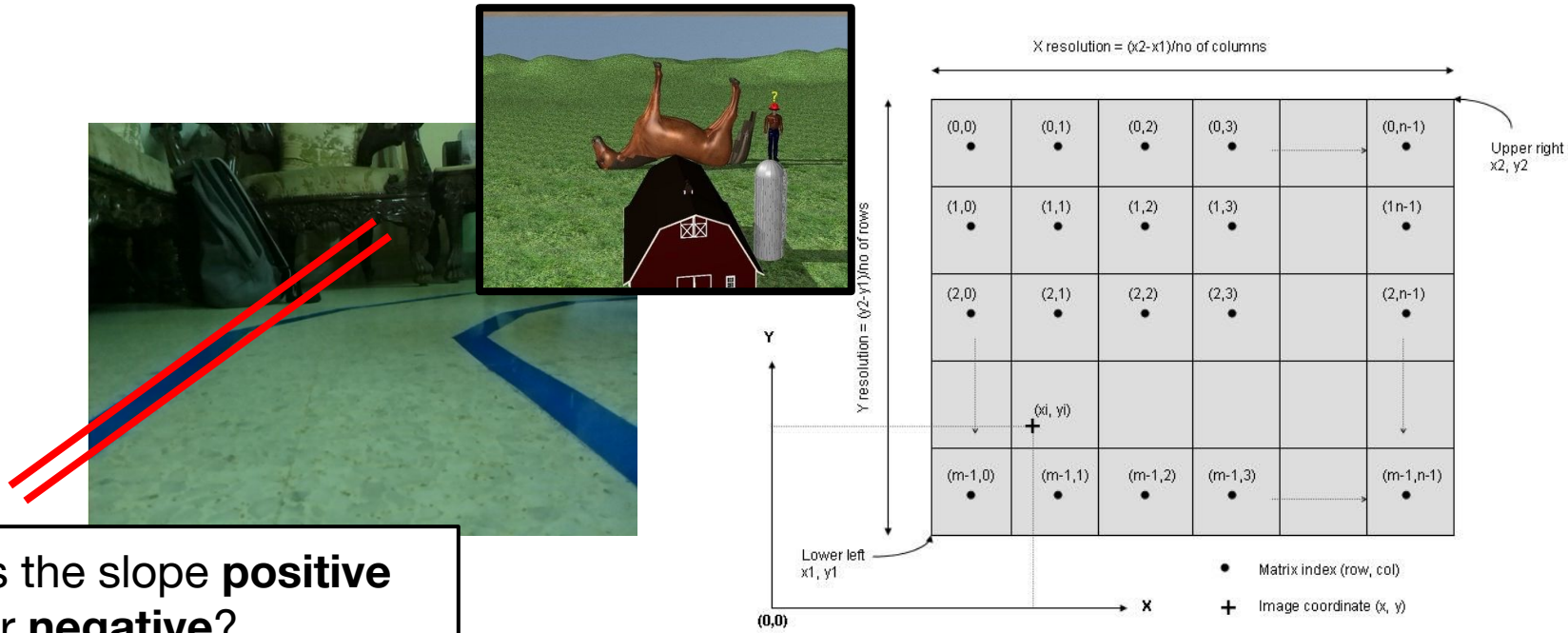




[FIG3] Using the normal parameterization of a straight line resolved the problem of "throwing a line at random" and also suggested a superior transform for computer vision purposes.



Images & info from: Peter E. Hart, "How the Hough Transform Was Invented". *IEEE Signal Processing Magazine*, Vol. 26 (2009) .URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5230799

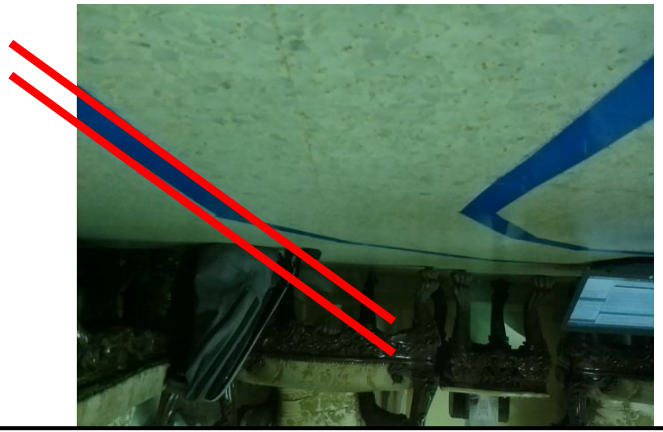
# Getting Our Lane Lines: Image Coordinate System



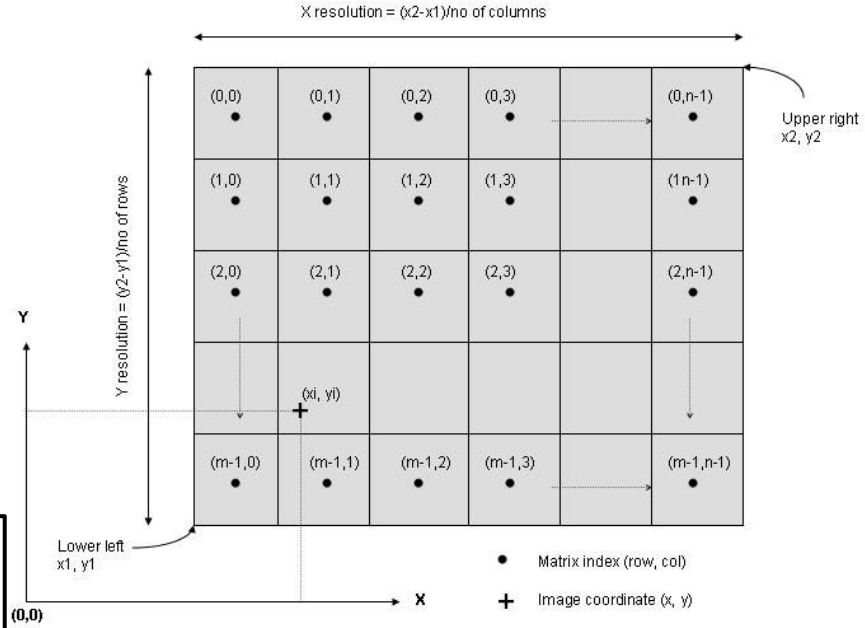

Is the slope **positive** or **negative**?

Real lane image from: User raja_961, "Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV". https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/
Image indexing figure from: "Understanding Coordinate System". URL: http://noobeed.com/nb_coord_system.htm
Farm picture from: https://www.wordseye.com/view-picture/48818

# Getting Our Lane Lines: Image Coordinate System

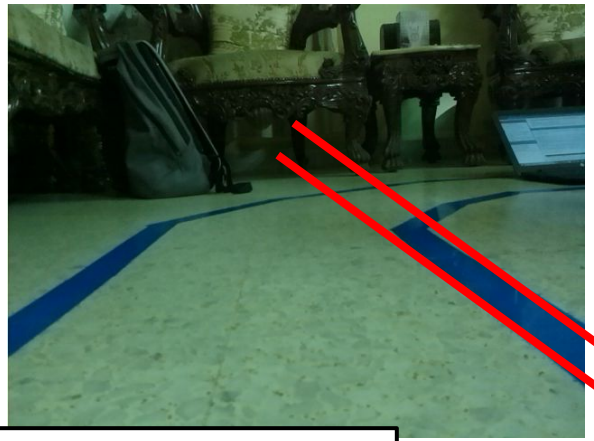


Slope is **negative**
**Left lane** marker has **negative slope**

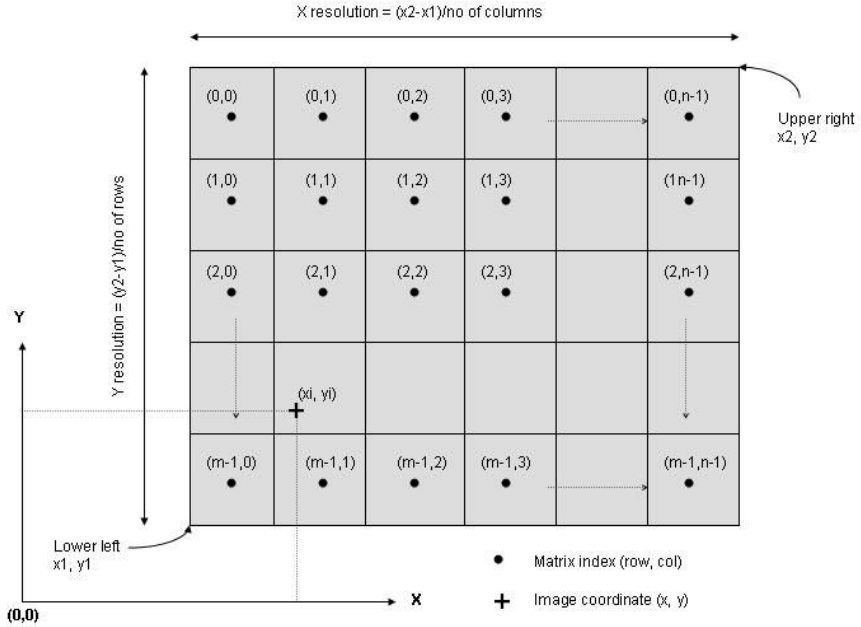



Real lane image from: User raja_961, "Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV". https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/
Image indexing figure from: "Understanding Coordinate System". URL: http://noobeed.com/nb_coord_system.htm

# Getting Our Lane Lines: Image Coordinate System



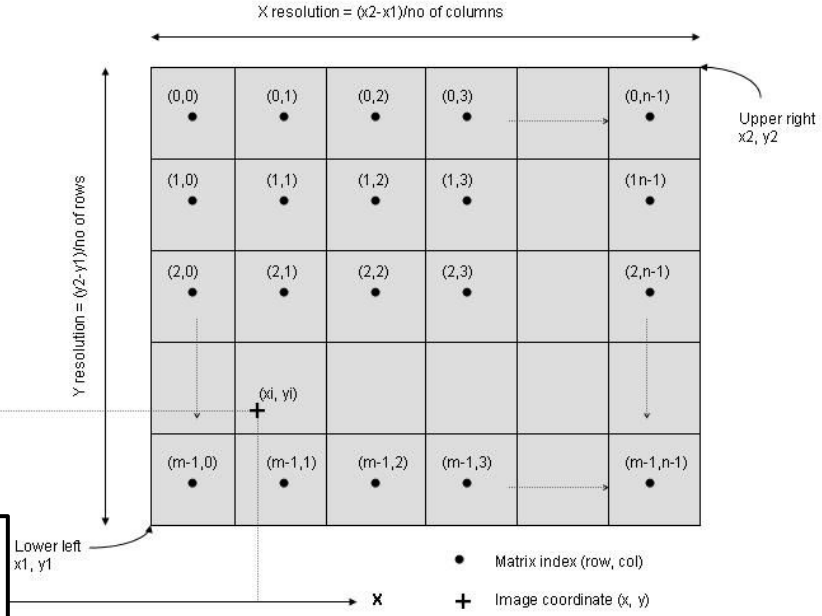
Is the slope **positive** or **negative**?




Real lane image from: User raja_961, "Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV". https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/

Image indexing figure from: "Understanding Coordinate System". URL: http://noobeed.com/nb_coord_system.htm

# Getting Our Lane Lines: Image Coordinate System




Slope is **positive**
**Right lane** marker has **positive slope**

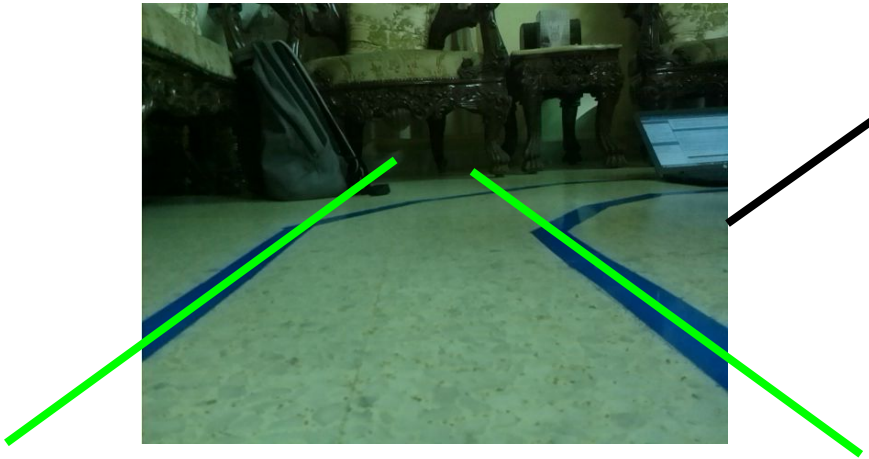
Real lane image from: User raja_961, "Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV". https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and-
Image indexing figure from: "Understanding Coordinate System". URL: http://noobeed.com/nb_coord_system.htm

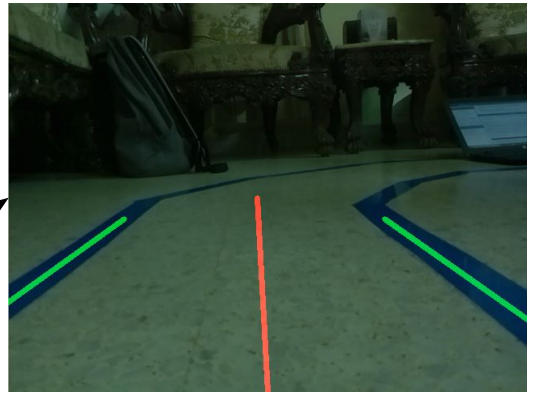

# Getting Our Lane Lines: Just Take Some Averages

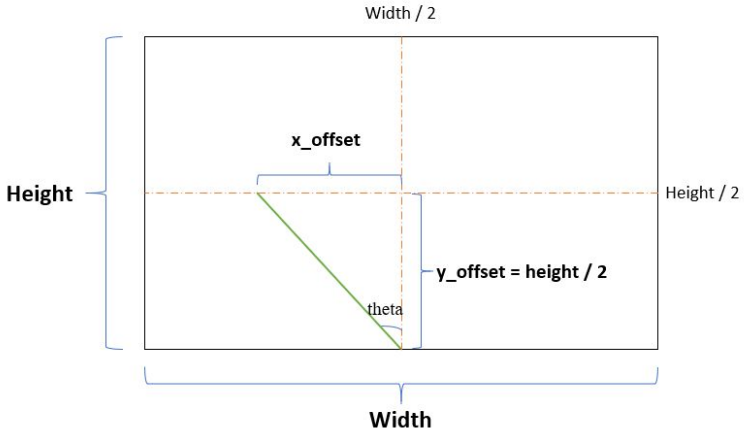
Real lane image from: User raja_961, “Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV”. https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/

Camera → HSV → Blue → Edges → ROI → Line Seg → y=ax+b → Theta Heading

# Getting Heading Line





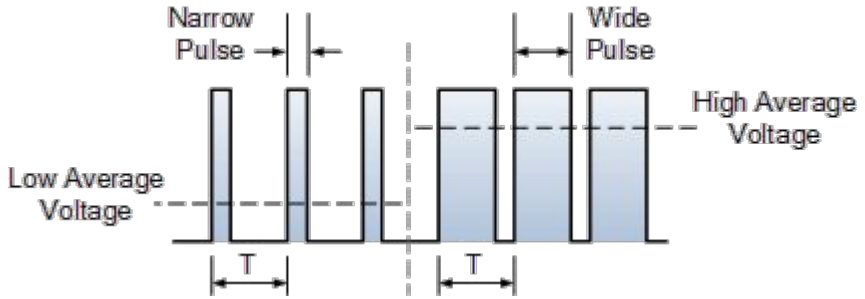



What should we use for our **error**?

Images from: User raja_961, “Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV”. https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/

# PWM: Pulse-Width Modulation

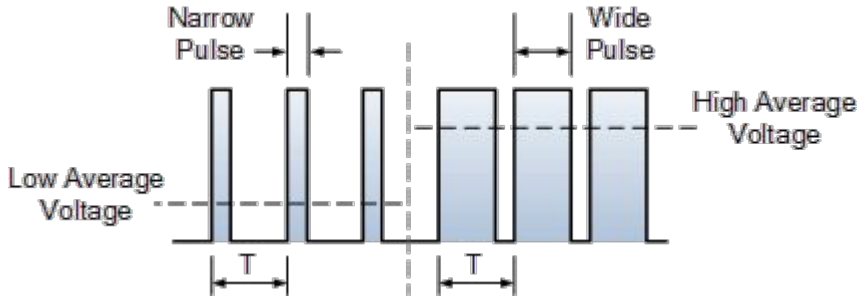
- Analog to digital encoding
- We tend to control motors via PWM
- PWM involves changing the duty cycle of a square wave
- DC of 0% means 0V DC waveform
- DC of 100% means 3.3V DC waveform
- DC of 50% means typical square wave going from 0V to 3.3V back and forth




"Pulse Width Modulation". Electronics Tutorials. URL: https://www.electronics-tutorials.ws/blog/pulse-width-modulation.html

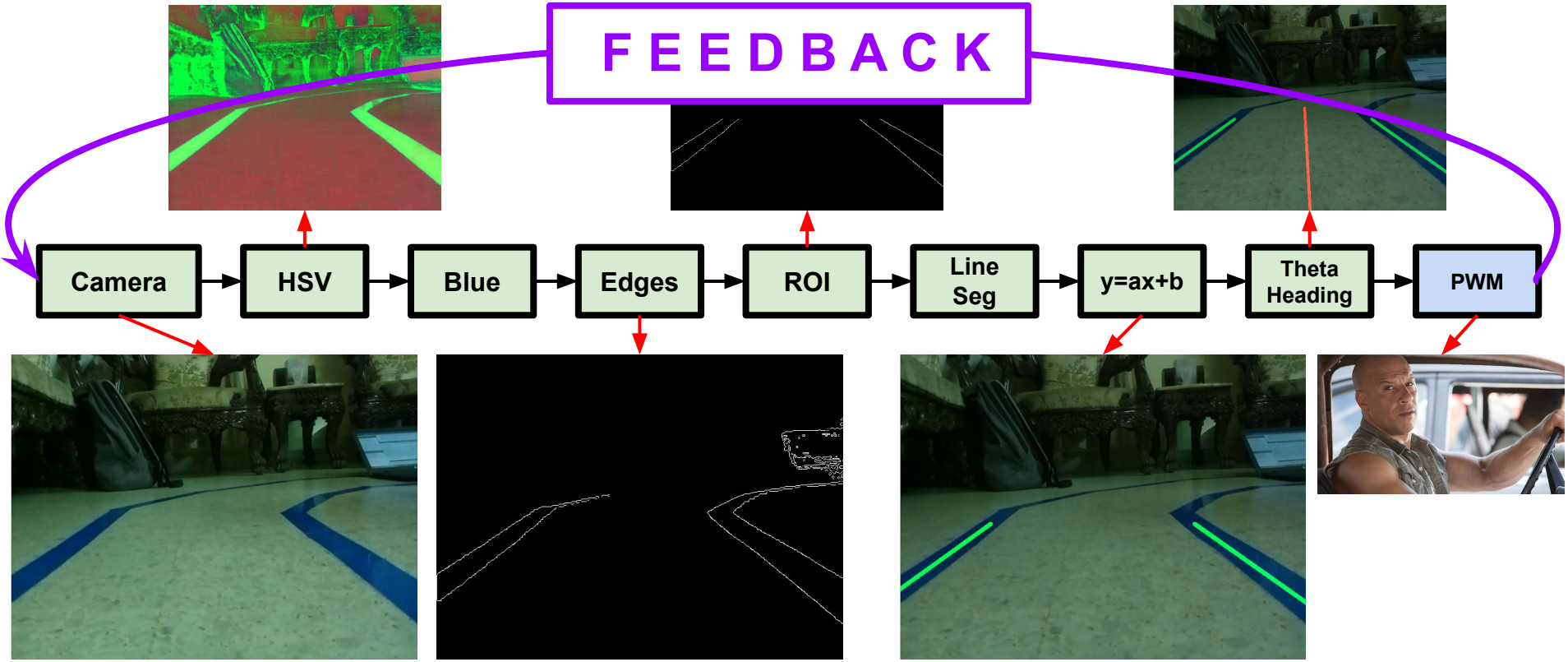
# Translating PWM To Steering & Speeds For Our RC Car

- Electronic speed controller (ESC) for speed motors
- The steering motor is a **servo motor**




"Pulse Width Modulation". Electronics Tutorials. URL: https://www.electronics-tutorials.ws/blog/pulse-width-modulation.html

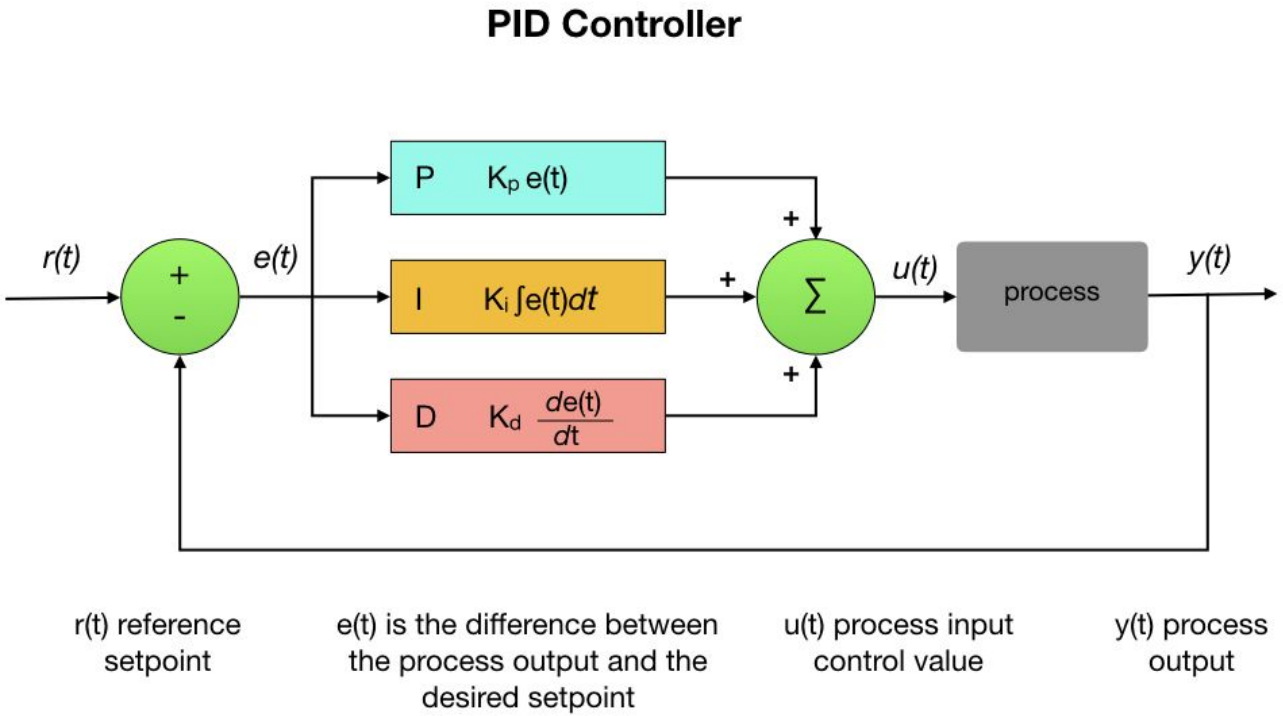
# Pipeline of Image Processing From Instructable






User raja_961, “Autonomous Lane-Keeping Car Using Raspberry Pi and OpenCV”. https://www.instructables.com/Autonomous-Lane-Keeping-Car-Using-Raspberry-Pi-and/
Dom picture from Fast & Furious franchise, Universal Pictures. URL: https://dashfun.net/fast-and-furious-9-vin-diesel-plans-a-spin-off-on-dominic-toretto/

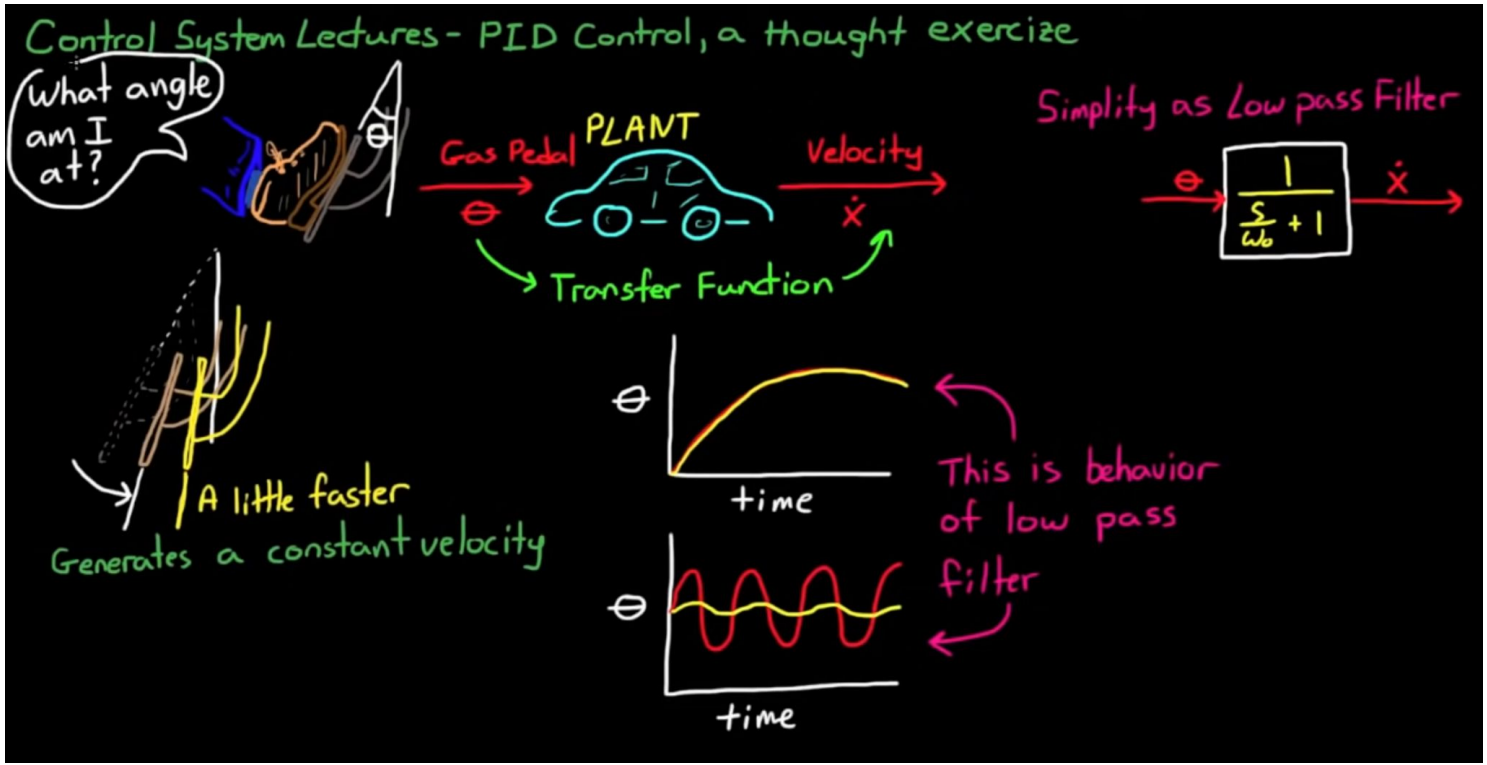
# Feedback: PID Control






"PID Control for Robotics". Programming Robots. URL: https://mjwhite8119.github.io/Robots/pid-control

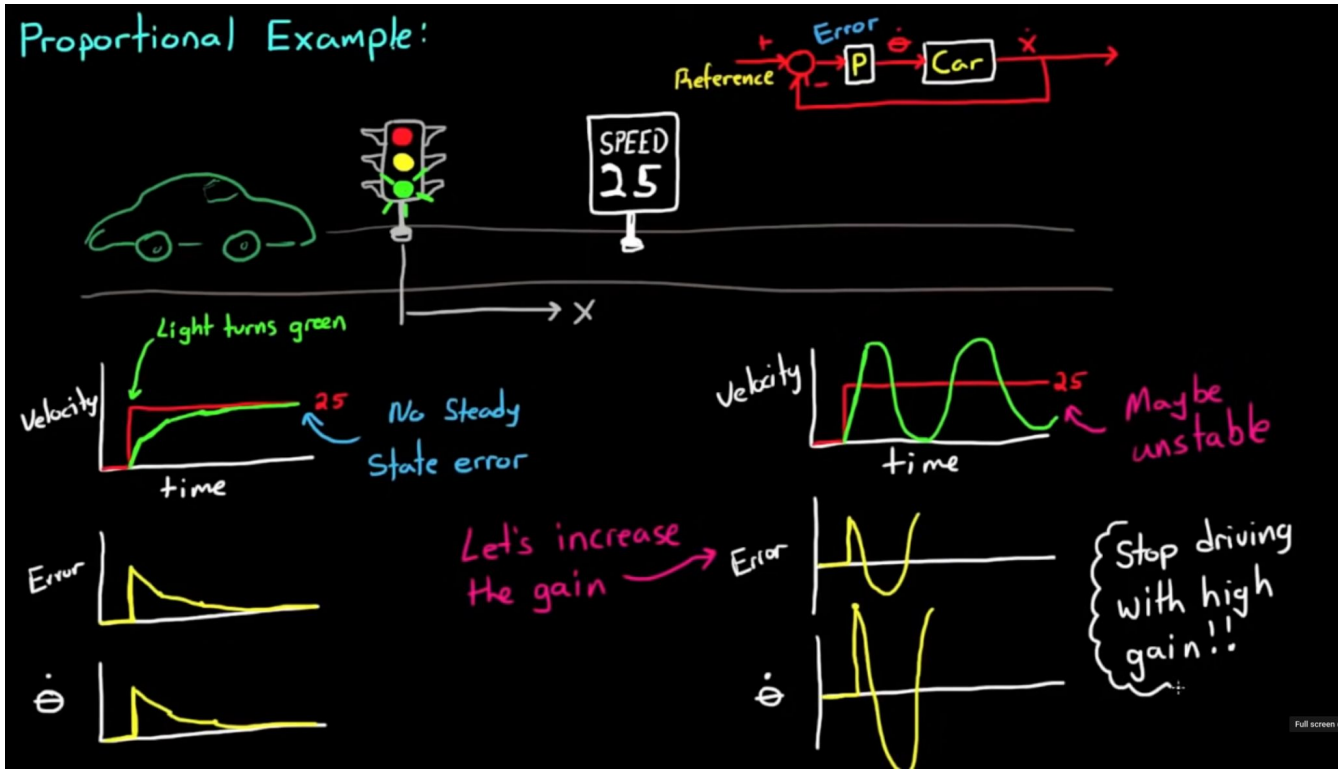
# Feedback: PID Control Responses






Screenshot from: Brian Douglas, “Simple Examples of PID Control”. URL: https://www.youtube.com/watch?v=XfAt6hNV8XM

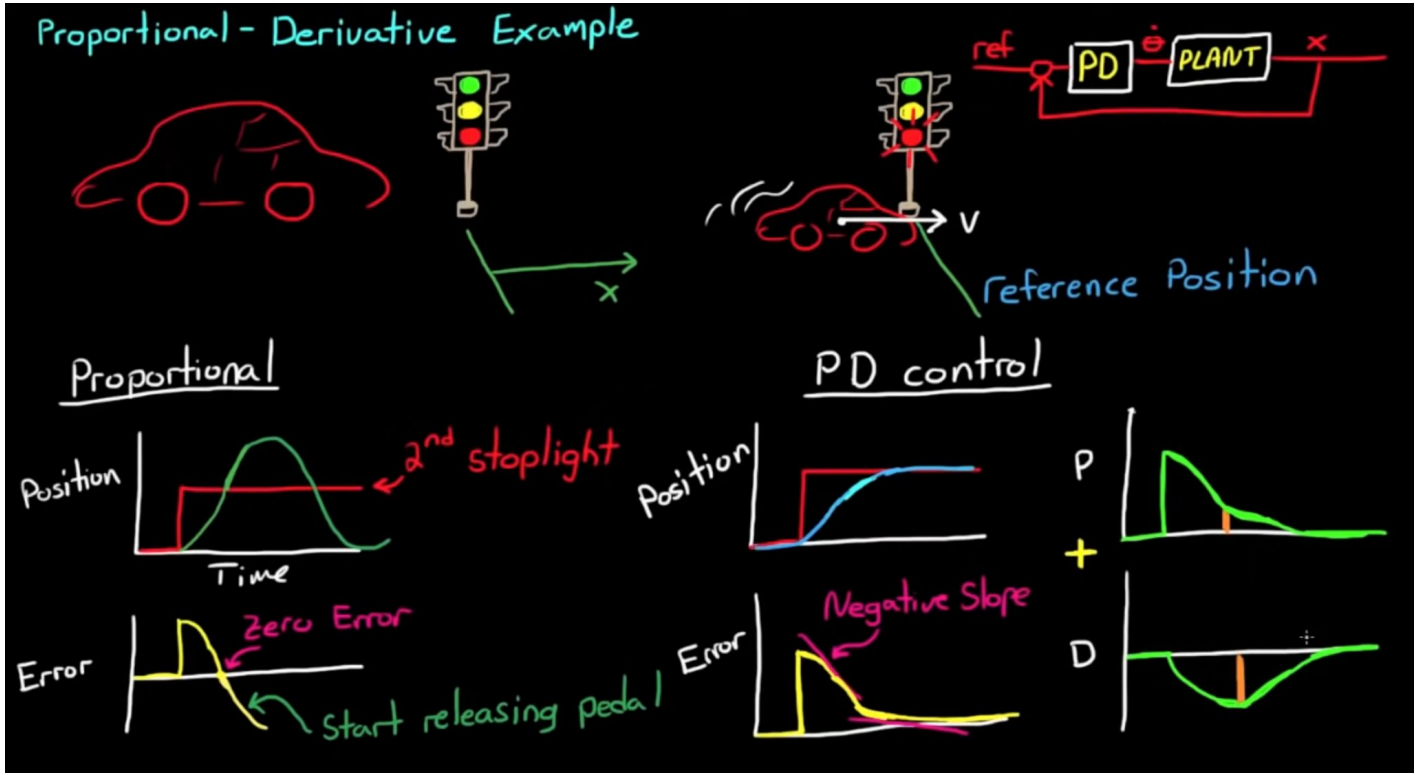
# Feedback: PID Control Responses





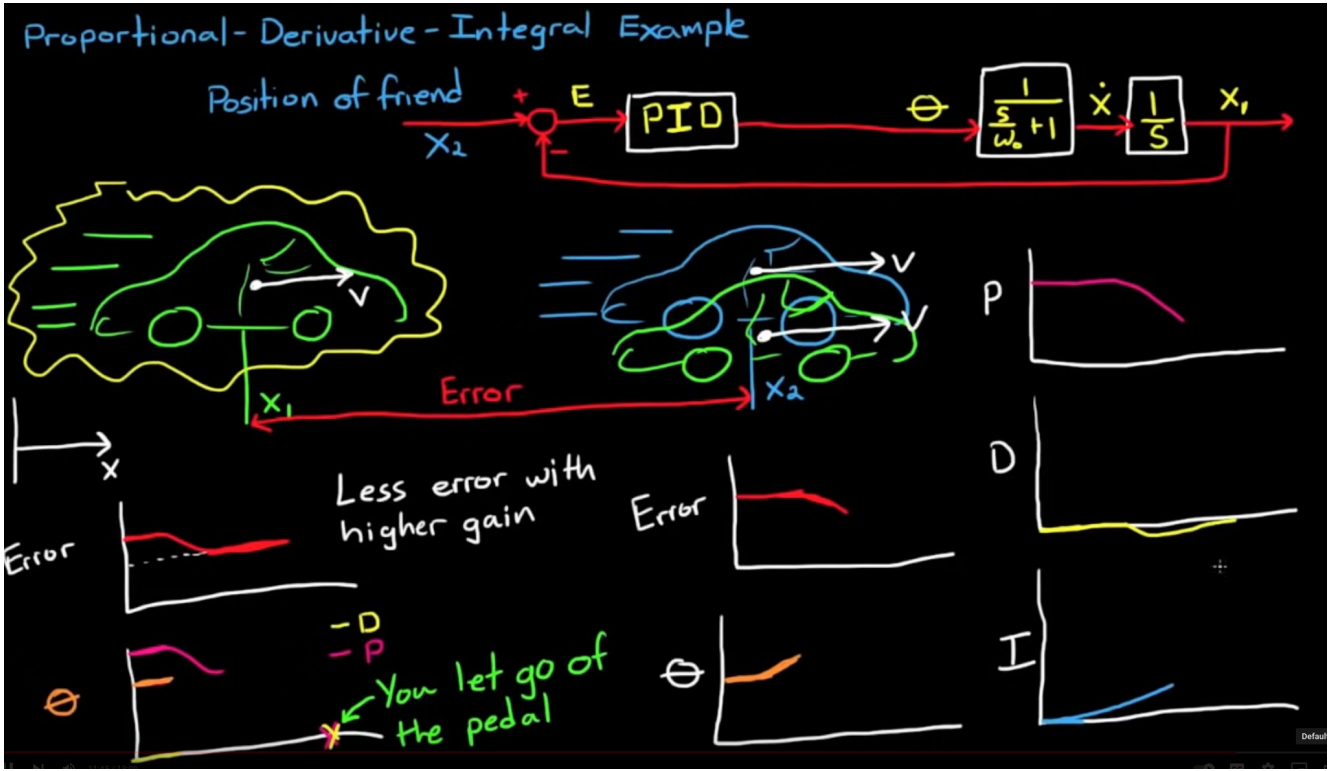

Screenshot from: Brian Douglas, “Simple Examples of PID Control”. URL: https://www.youtube.com/watch?v=XfAt6hNV8XM

# Feedback: PID Control Responses




Screenshot from: Brian Douglas, “Simple Examples of PID Control”. URL: https://www.youtube.com/watch?v=XfAt6hNV8XM

# Feedback: PID Control Responses




Screenshot from: Brian Douglas, “Simple Examples of PID Control”. URL: https://www.youtube.com/watch?v=XfAt6hNV8XM