Mechatronics Systems Design Laboratory

ECE 491

Igor Paprotny

Upcoming Checkout

- This week:
  - DC – DC Converter – Lab 6
- Next week:
  - Line Camera – Lab 7

PCB Board 1

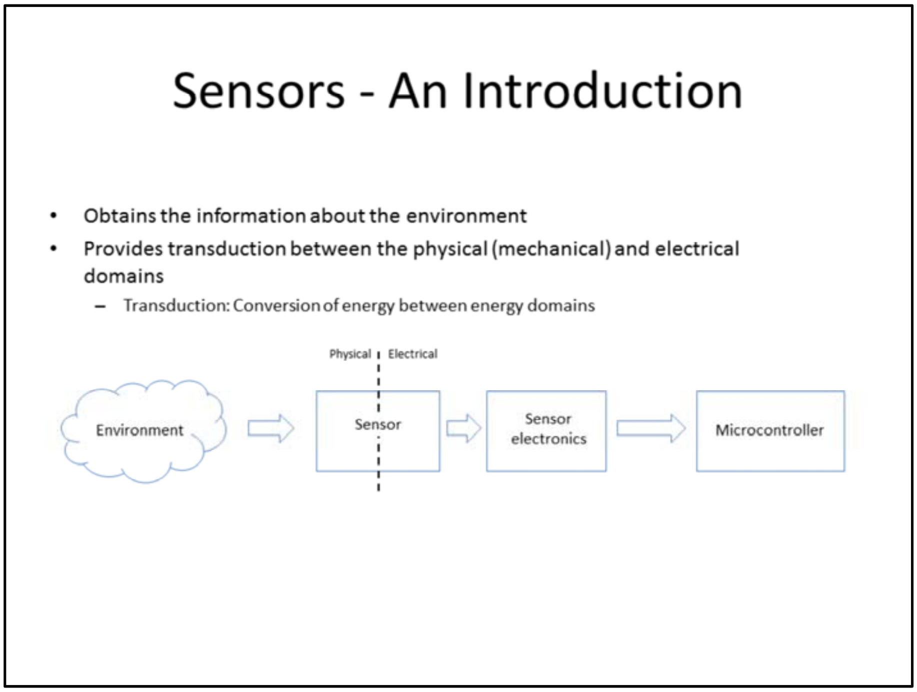
- Designs to John by Friday 3/10/2017
  - Noon
- Use template provided for you
- Shall contain:
  - power supply (DC-DC)
    - Remember disconnect switch
  - Motor controllers

Midterm (3/14/2017)

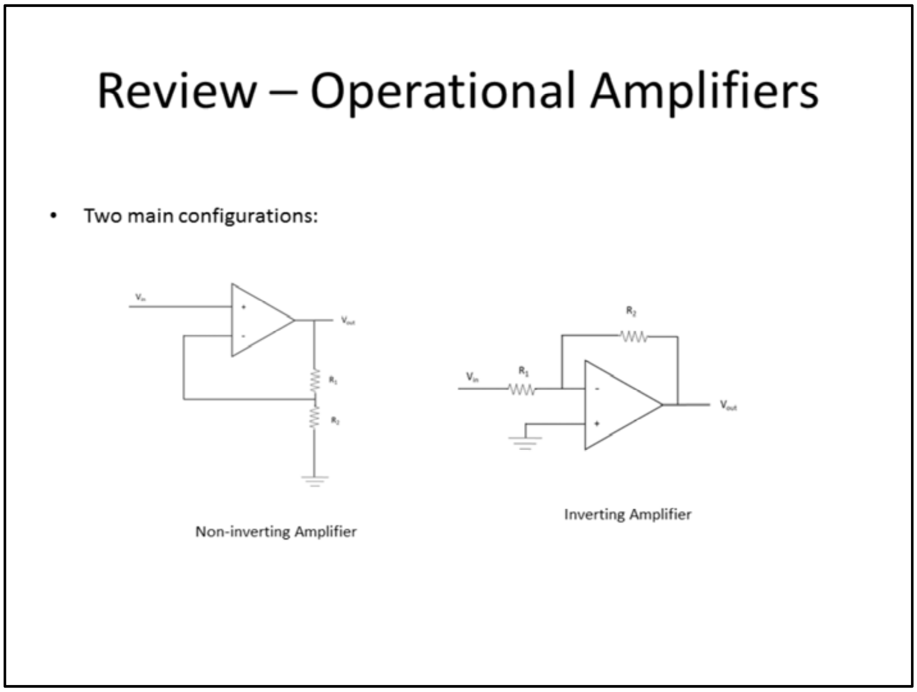
- Open Book/Open Notes
- Topics:
  - Motors
  - Motor controllers
  - FET review
  - DC-DC converter
  - OP-Amps review
  - Encoders

Other Upcoming Stuff

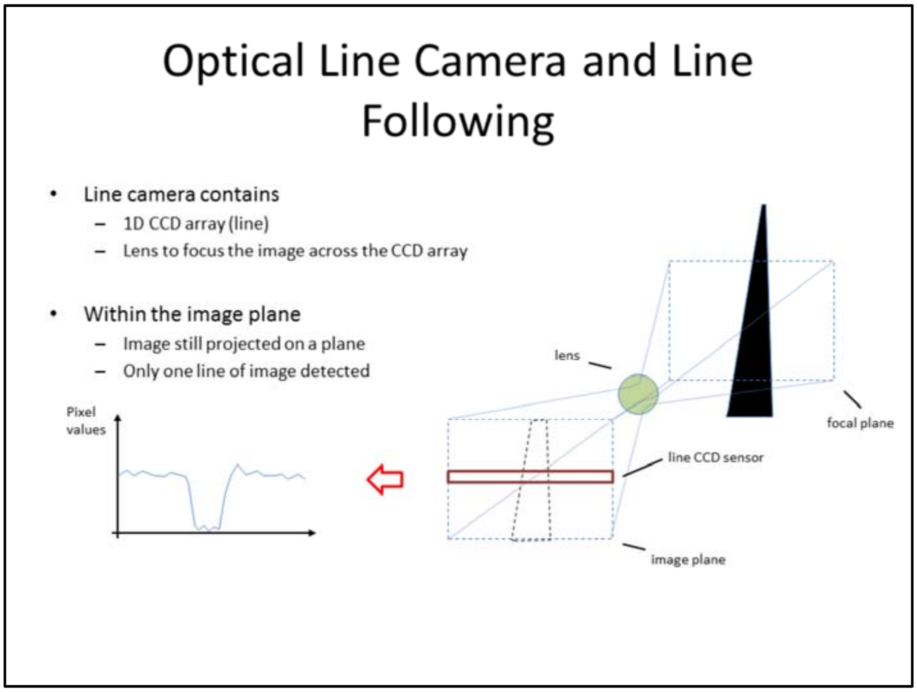
- Project Progress Report
  - Due Friday 3/17
- Round 1
  - Friday April 7th TBD



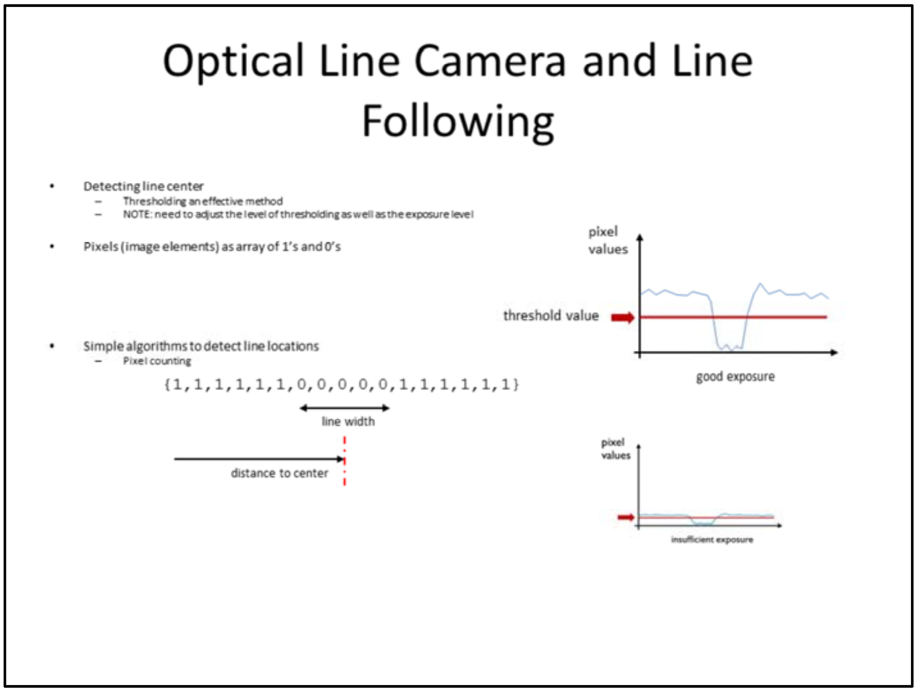
Sensors are tasked with obtaining the information about the environment (the physical system surrounding the autonomous car) and the input of a microcontroller. Sensors can be viewed as to provide transduction between the physical and (most often) electrical domains. Transduction implies conversion of energy between energy domains, and a sensors most often is a transducer that converts energy from the mechanical (physical) domain to the electrical domain. A typical sensing systems consists with the environment influencing the sensor is some way (e.g. change in the temperature). The sensor converts that influence into some for of electrical signal (e.g. change in resistance). That change is now in the electrical domain, but most often the signal is too weak to be interpreted by the microcontroller directly. Sensor electronics take care of conditioning the signal out of the sensor to interface with the microcontroller. This sensor electronics is often using operational amplifiers to amplify the signal to levels required by the microcontroller input. The microcontroller reads the conditioned input signal through its analog or digital input ports, and uses that information in its control loop.



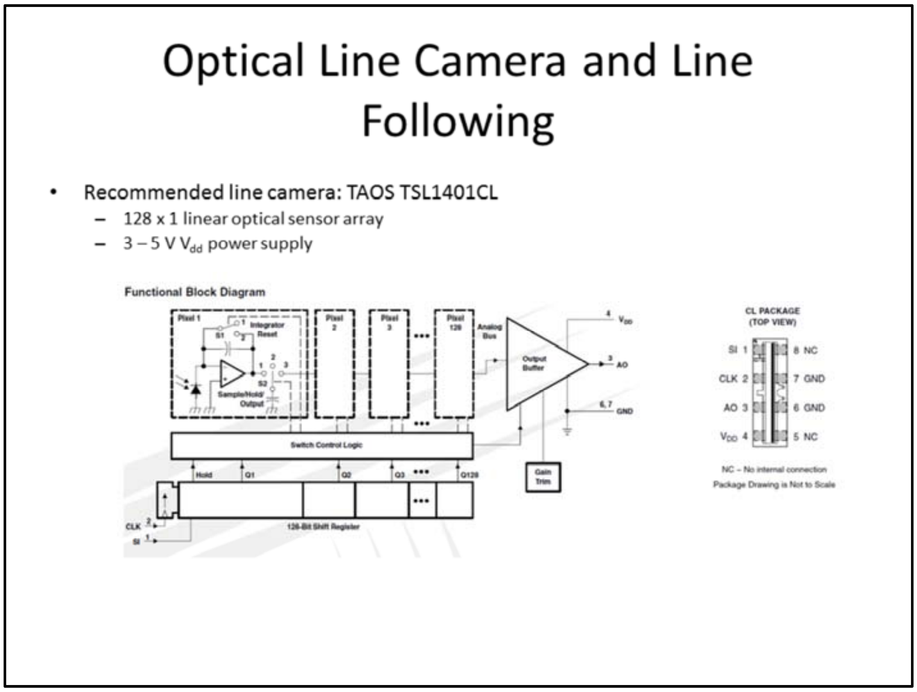
The OpAmps come in two flavors, non-inverting or inverting. We will analyze both in the next slides. In general, although the gain of the OpAmp is assumed infinite, infinite gain is not very usefull. So, in a practical amplifier circuit, the gain is set to a fixed value using a feedback circuit.



The line camera will still project the image through a lens on the image plane, just like a regular camera, however the CCD sensor will only detect (capture) the signal from one line of the image. The line will be represented by analog values, with the high values representing light pixels, with dark values representing dark pixels. Just like a regular camera, the image has to be made sharp by adjusting the focal plane of the line camera. However, the plane only has to be in focus at the image that is project over the line sensor. An in-focus line means a less blurry line. How well the line is detected depends on the focus, and the orientation of the camera. Both should be adjusted for optimal results

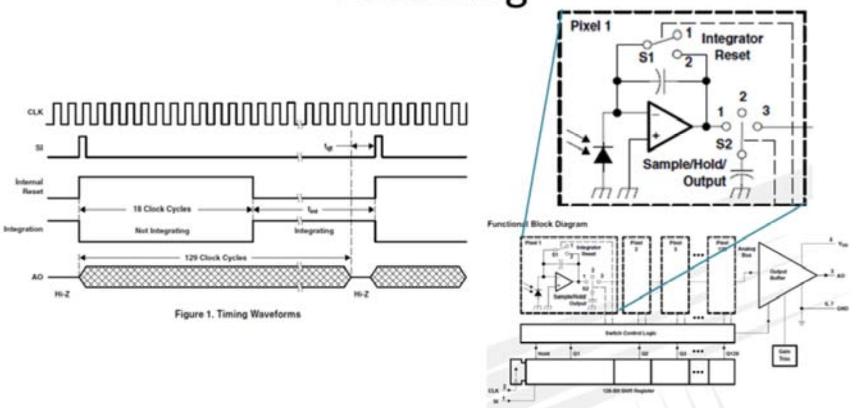


An easy way to detect the line is to simply threshold the pixels values, assigning a digital array of 1's and 0's to values above and below the threshold value, respectively. Note that the input from the line camera depends on the exposure, and that the thresholding should be adjusted based on the exposure. Signal noise will likely be present, and the lightning conditions may vary. It is strongly recommended that periodic scaling of the exposure time is done such that the signal always remains within a reasonable range of the ADC scale. The location of the center line can then be detected by counting the pixels, detecting the line, (1 to 0 transition), measuring the width of the line (counting until next 0 to 1 transition), and diving the number of line pixels in half. Several measurements should be used to **average** the input signal.



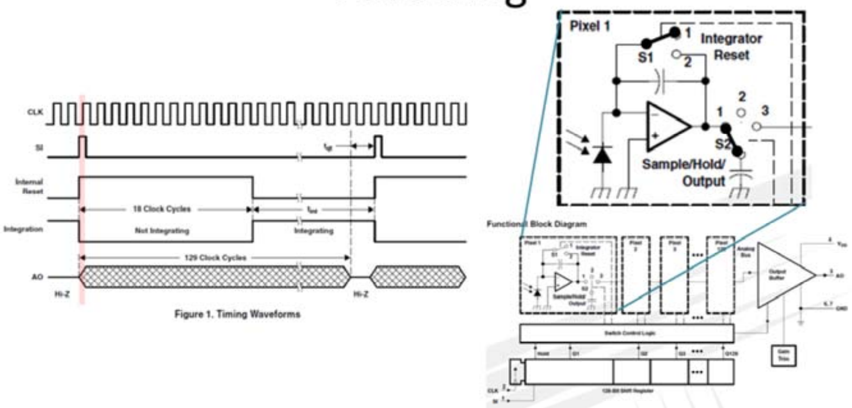
A recommended sensor that is supported in this module is based on the TAOS TSL1401CL sensor by texas instruments. It’s a 128 pixel line camera. One camera is included in the autonomous car kit, and it includes the focusing optics on the plane on the line camera sensor. The sensor is controlled by two digital inputs (CLK and SI) and the pixels are read serially through the AO output pin. The AO signal is rail-to-rail based on the supply voltage Vdd, and hence can be compatible with the ADC input of the microcontroller.

Optical Line Camera and Line Following



The clocking of the sensor can be further visualized by the following signal sequence.

Optical Line Camera and Line Following



The reset cycle is initiated by the SI pulse. First the charge is transferred to the storage capacitor via S2.

Optical Line Camera and Line Following

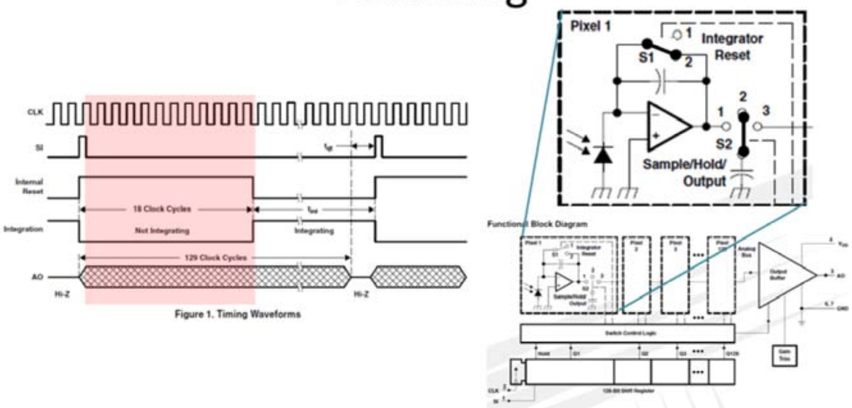


Figure 1. Timing Waveforms

After charge transfer, the storage capacitor is placed in high-impedance mode (disconnected), and the integrating capacitor is shorted to zero out the charge.

Optical Line Camera and Line Following

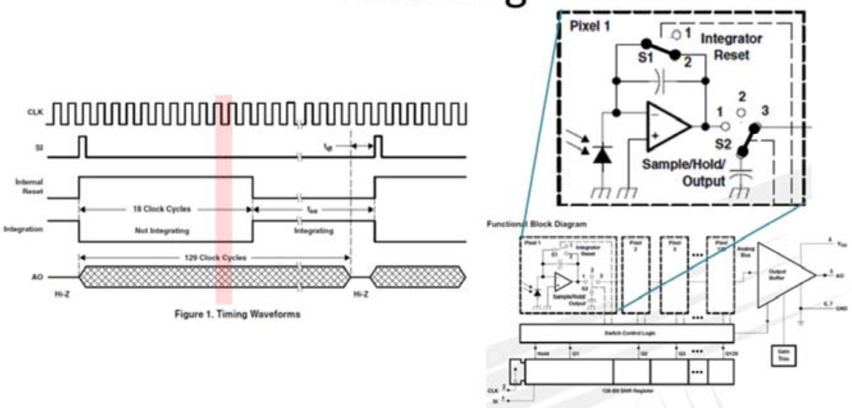
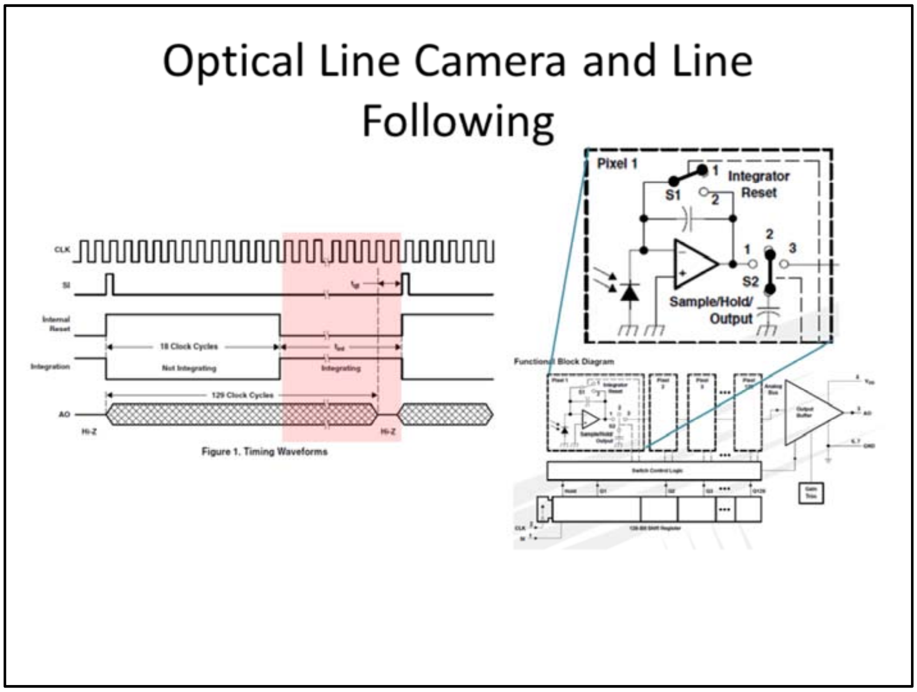
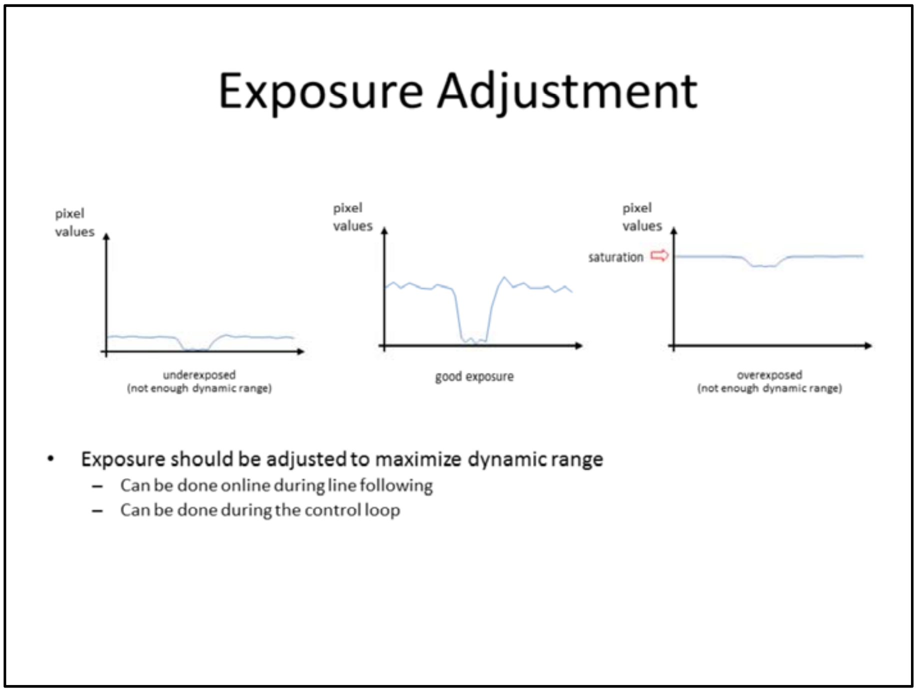


Figure 1. Timing Waveforms

As the clock clocks through the 128 clocking cycles, the storage capacitors are one-by-one connected to the output, to read out the pixels, and then back to the high impedance state. This can happen during the reset cucle or the integration cycle, but only will happen during the first 129 clock cycles. After that, all the storage capacitors are in high impedance state.



After the 18 cycles, the integrating capacitor is unshorted, and keeps integrating the input signal.



Determining the exposure time can be done

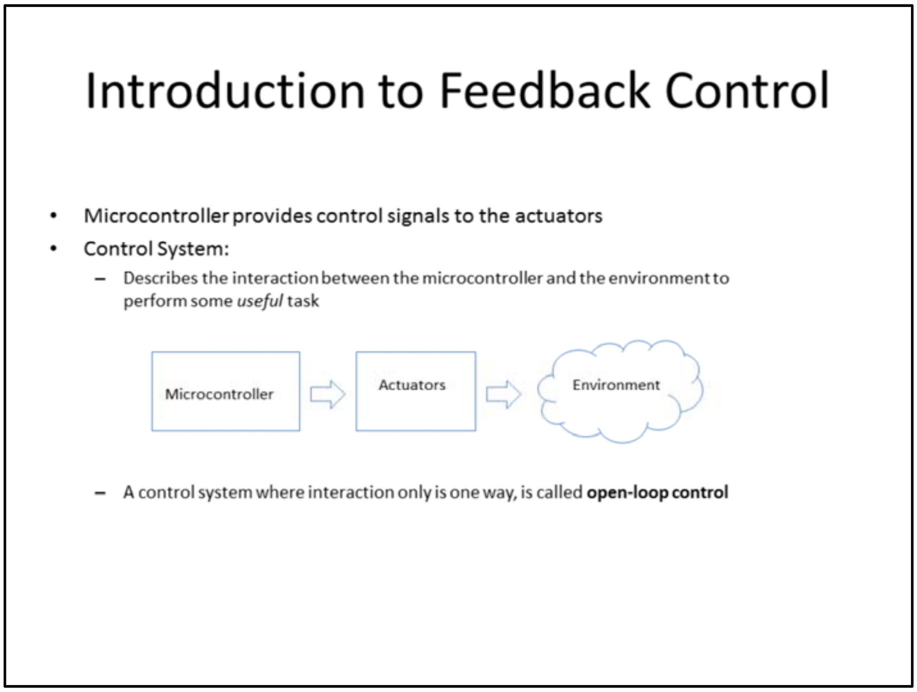
Summary: Optical Line Camera and Line Following

- A 2D light-sensitive pixel array is used in cameras for image capture
- A 1D pixel array (line) can be used for line detection – line camera
- Can be used for optical line following
  - Focus sensor on the line
  - Thresholding can be used to determine the center of the line
- Line camera provided with the kit uses TAOS TSL1401CL sensor
  - 128 pixels
  - Variable integration (exposure) time
  - Sequential (serial) output via AO, controlled through CLK and SI
- Exposure can be varied to accommodate changes in lighting conditions
  - Changing the CLK frequency
  - Can be done dynamically to account for changes in light conditions

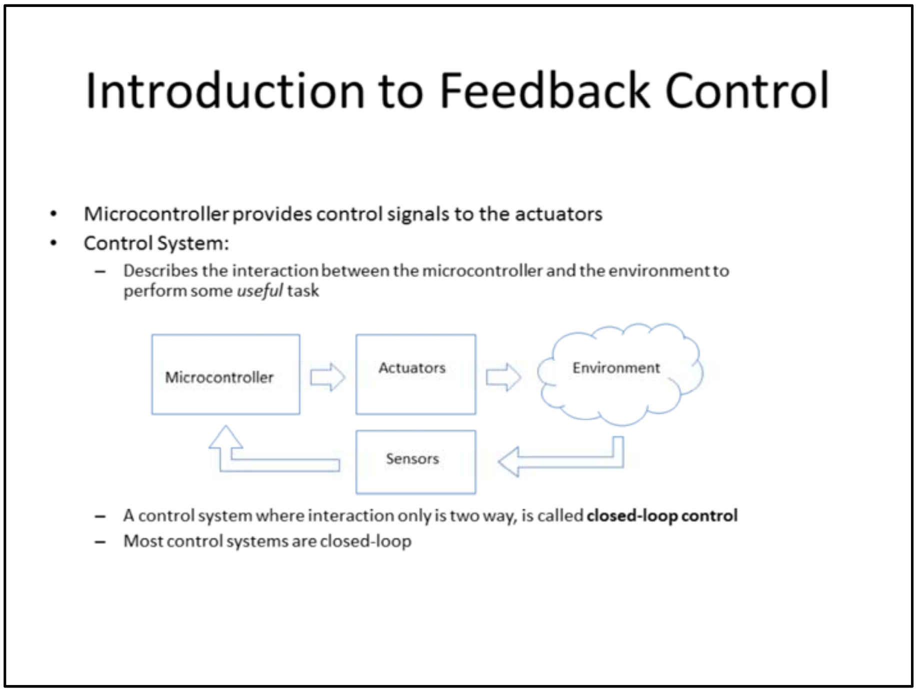
Summary

Feedback Control For Autonomous Car (1)

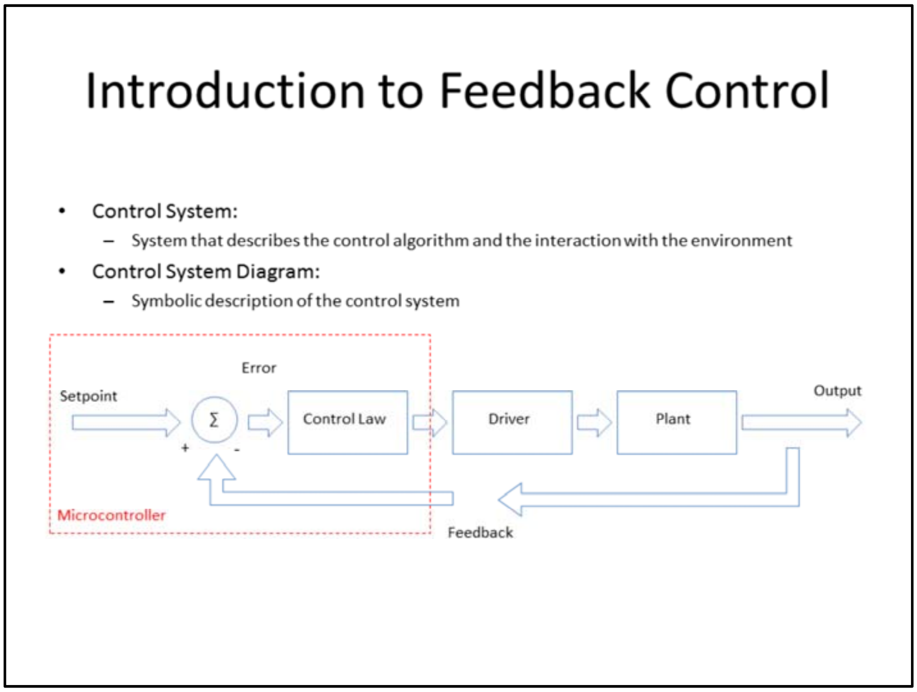
- Introduction to Feedback Control
- Nonholonomic Modeling of an Autonomous Car
- Simple Control
  - Velocity control
  - Steering
- Summary and Conclusion



In an autonomous car, the microcontroller provides input to the actuators, which propel the car (hopefully) in the desired direction. A control system describes the linkage (interaction) between the different components of the autonomous car. Hopefully all these components are helping to propel the car in the right direction. The act of moving the car affects how the environment where the car is situated.
A control system where the interaction only goes one way, i.e. the microcontroller sets an output without checking the consequences, is called open-loop control. It is in principle possible to drive an autonomous car open-loop, however the control algorithm would have to anticipate everything that happens along the way perfectly in order for this control to work.



Rather, most control systems use some for of feedback using sensors to double-check if the action had the desired effect, and correct the action based on the error. This type of control is called feedback control or closed-loop control. Sensors provide the information (feedback) as to the state of the environment, and based on the sensory input the algorithm makes a decision as to the output.

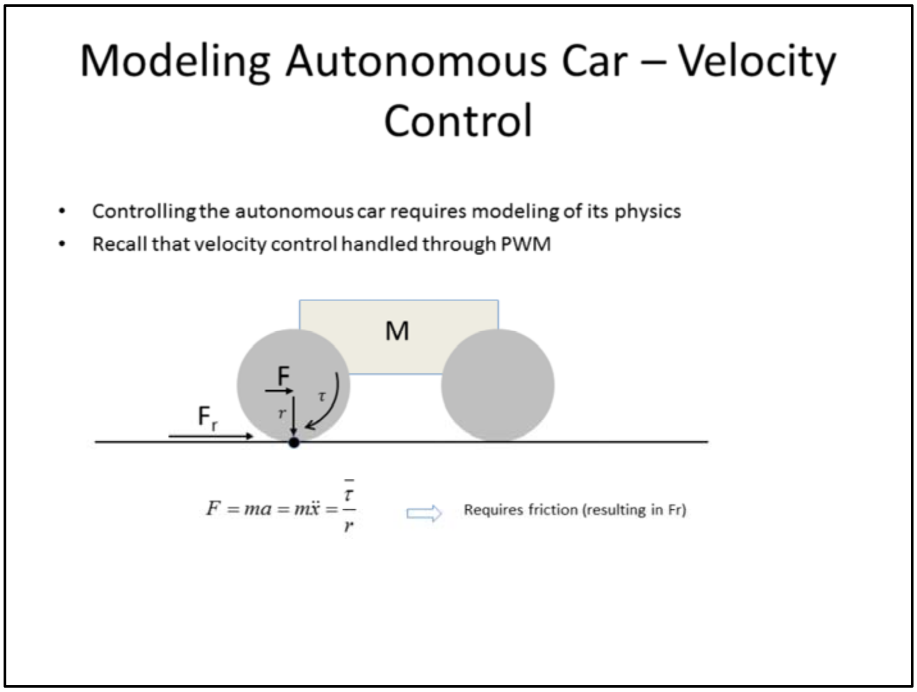


Traditionally, a control system is described using a control system diagram, and part of the algorithm used for control is described as a control system law. The algorithm actually compares the setpoint input (e.g. the desired velocity) with the actual value (obtained by the sensors) and based on the difference between the desired and the actual output, action is taken. The output of the microcontroller ins amplified by the driver, which represent the amplifiers and actuators, e.g. motors, before the environment is affected. The effect on the environment results in an output, i.e. some desired (or not desired!) changes, labeled as output. The environment is labeled as “plant”, because traditionally the control systems were used to control industrial processes in production plants. Feedback looks at the output and registers what happened. The summation between the actual result and the setpoint (denoted by sigma) provides an error to the algorithm based on which the action is taken. This block diagram is stil a simplicication, but in general represents how a control system works. It is also well suited to a proportional control of the output, but more on this later.

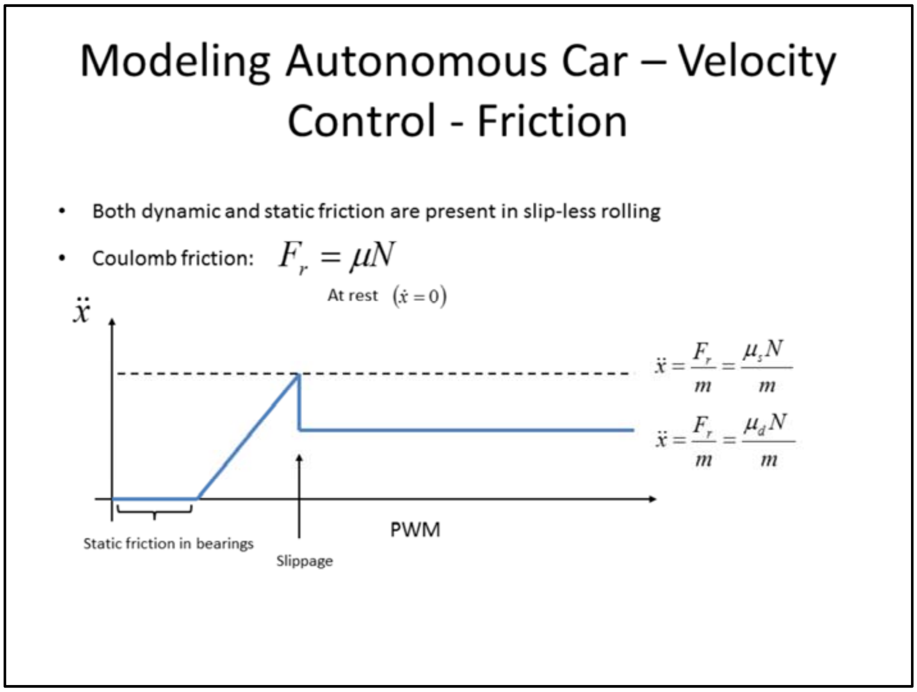
Modeling Autonomous Car

- Controlling the autonomous car requires modeling of its physics
- Simplified control problem:
  - **Velocity Control:** Consider the dynamics of motion, drag, and torque affecting the forward motion of the car, but disregard effects due to steering
  - **Steering Control:** Consider line following, subject to non-holonomic constraints associated with steering, but no dynamics (quasi-static system only)
- Nonholonomic Constrains (for Autonomous Car):
  - Constrains that prevent motion in all directions. The car cannot move sideways.
- Configuration Space (C-space):
  - Parameters used to completely define the coordinates of the autonomous car.

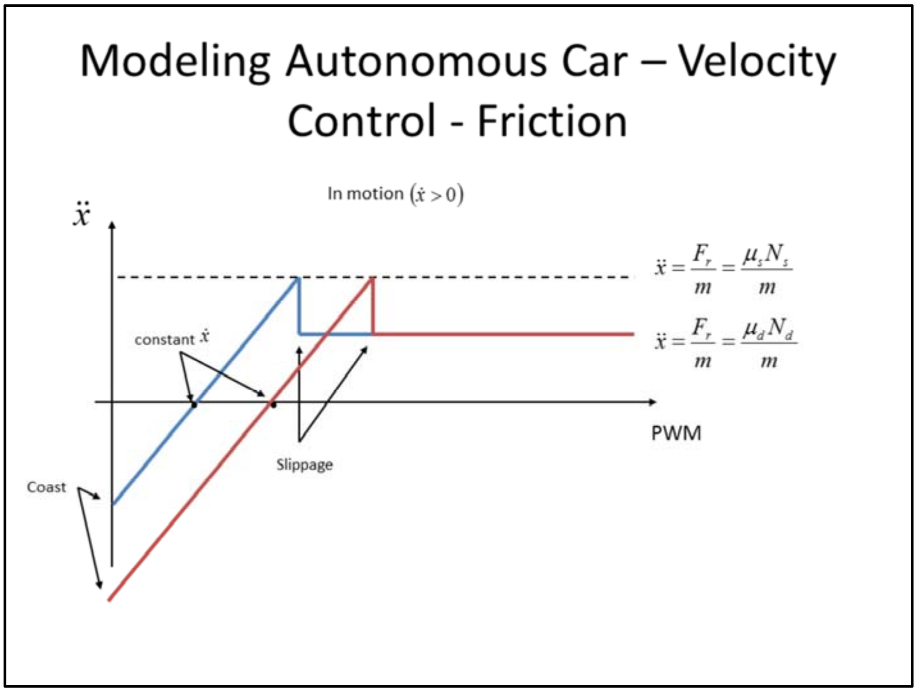
In order to be able to control the autonomous car, its physics has to be modeled, i.e. we need to understand the mapping between our degrees of freedom for control to the degrees of freedom of motion. The problem of controlling an autonomous car can be boiled down to two distinct components, that are only somewhat loosely connected. First, velocity control, i.e. designing a control system that can accelerate and maintain a set velocity along the track. Second, steering control, where the car is steered along the track, i.e. it has to follow curves, turn. In principle the two are unrelated, however we may want to slow down at the curves, i.e. adjusting the set point of the velocity control. The steering is necessary, because the autonomous car is subject to non-holonomic constraints. Non-holonomic constraints imply that the car cannot move in all directions in its configurations space (c-space for short). C-space contains all the parameters necessary to describe the quasi-static state of the autonomous car, in this case, the configuration space for a planar rigid body (such as a car) in 2D (planar space) consists of x and y coordinates, as well as rotation theta_c. The car is non-holonomic because it cannot move sideways.



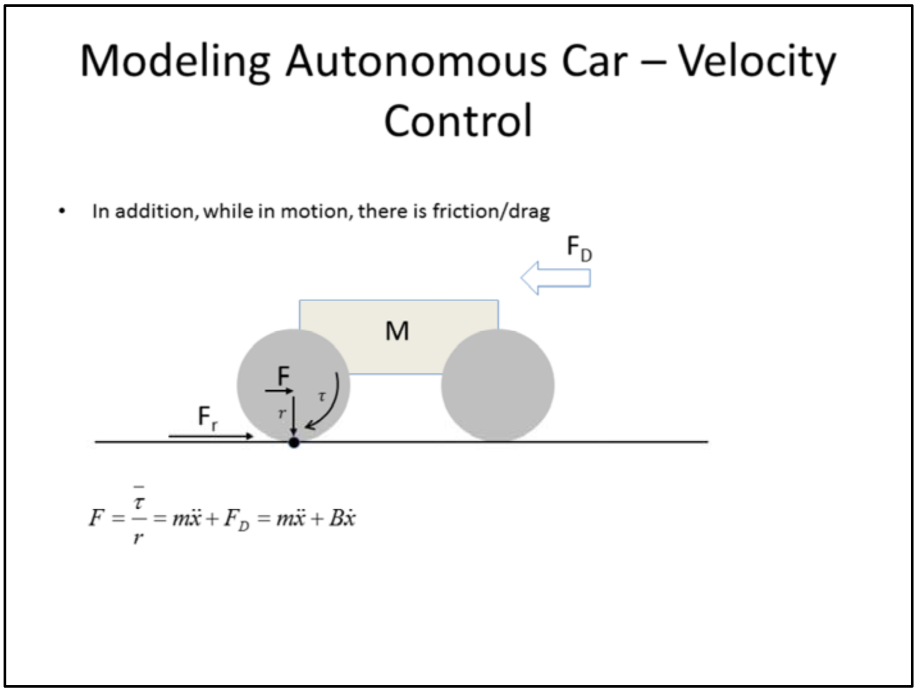
Looking at velocity control, we can consider a 1D (straight line) motion of the car. The car has some mass, and is subject to Newton's second law, F=ma. F is how the mass of the car is accelerated. F is of course acting through the car chassis, and is dependent on the interaction of the wheels with the underlying substrate. The force at the interface of the wheel with the substrate is a function of the torque on the wheel tau, divided by the wheel radius r. For rotation, one can assume instatanious rotation along the point of contact (black dot), and the force F acting on the wheel center, assuming no slip condition, i.e. F = F_r, where F_r is the reaction force at the surface interface. Recall that the way we will control the torque is using PWM. Hence, our degree of freedom for controlling velocity is PWM.



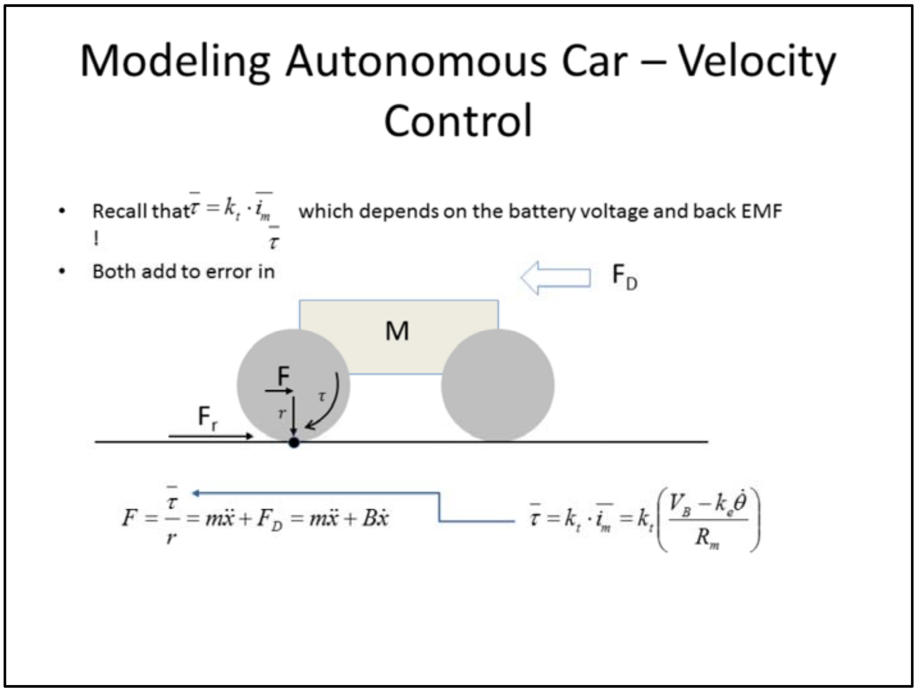
Before we proceed it is important to analyze the friction, and its interaction with our car model. The most common model for friction is Coulomb friction, od damping. The common model for this is that the force resulting from friction is a coefficient of friction, mu times the normal force. Friction always opposes the motion, or actuating force. There are two types of friction, static friction (mu_s) and dynamic friction (mu_d). In general, static friction is larger than dynamic friction (one notable exception is Teflon). Let us consider the case where the initial velocity of the car is zero (car is at rest). As we start increasing the PWM, initially nothing happens, this is where the friction in bearings is resisting the still weak torque exerted by the motor shaft. As we increase the PWM, very soon this static friction is overcome, and the car starts accelerating. The wheels start rolling, because the static friction at the wheel contact point creates the necessary reaction force F_r. As we increase the PWM, the torque increases linearly (recall tau is proportional to the duty cycle of PWM). The reaction force grows, because we are accelerating faster, so ma increases. Eventually, we reach the limit of static friction, and F_r is no longer able to counteract desired ma. At this point wheel slippage occurs. The wheels are now spinning, and we are in the dynamic friction regime. Note that the F_r is independent of the PWM, because it only depends on mu_d and N. Hence, any increase in PWM does not result in increase in acceleration. Note, that the above analysis is done from the car being at rest, i.e. before the car starts moving to fast.



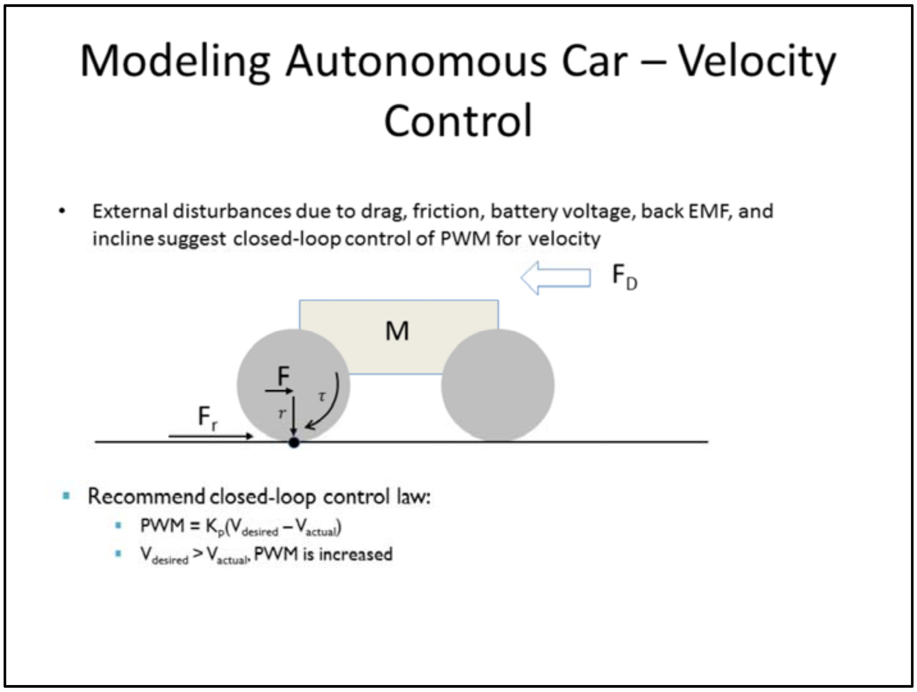
During the motion, as the car has gained some velocity (x dot). Now, some PWM has to be used to maintain constant velocity (x dot constant). However, drag, friction, and back EMF will require us to maintain some PWM. Because drag (and back EMF) in general is a function of the velocity (B x dot), a larger PWM is required to maintain constant larger velocity. As such, the blue curve represents car at some velocity, and red curve the car at a higher velocity. If the PWM is set to zero (disconnect battery) drag, friction, and back EMF will cause the car to slowly lose it velocity, and eventually coast to a stop.



Consequently, we can add drag to the electromechanical modeling of the autonomous car.



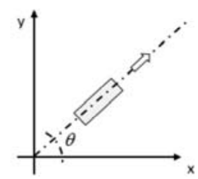
Furthermore, we should expand the model to include the influence of battery voltage (V_b, which will change as the battery discharges), and back EMF to the model.



It now becomes clear that a set PWM can result in vastly different velocity x dot, depending on the battery voltage, drag (which may vary), back EMF, and incline (if the robot is moving on a non-planar surface). All these aspects suggest a closed-loop control for the PWM such that the system can quickly accelerate to set x dot. We will go over different types of controls later in this module. However, for now lets assume a very simple control low, which is based on the following control law: lets make the PWM proportional to the error between the desired and actual velocity. In this way, if this error is large, we result in a large PWM. This type of control is called *proportional control*, with K_p being the constant of proportionality. Lets leave this for now, and revisit this later when we talk about and analyze the PID controller.

Modeling Autonomous Car – Non-Holonomic System

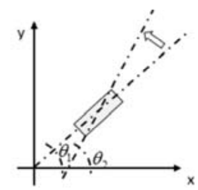
- An autonomous car is non-holonomic, i.e. its motion is subject to non-holonomic constrains of its motion. This means that it cannot move in all directions.
  - Only move forward and turn
- Simplified Systems: Unicycle
  - Configuration $\vec{q} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$
  - Equation of motion: $\dot{q} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u_1 + \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} u_2$



The autonomous car can be described by equations of motion, which governs how it moves. We started looking at this when we analyzed the velocity control, and the dynamics of the autonomus car. However, when looking at steering, we are more concerned with the path of the car as opposed to the dynamics. An autonomus car cannot move in every direction, hence it is non-holonomic. The car can only move forward or turn along a certain radius. However initially we can start explaining a non-holomic system in plane by using the analogy of a unicycle. A unicycle is a non-holomonic system consisting of a single wheel that can either move forward or turn (rotate) in place. Its configuration in plane can be described as position (x,y) and rotation. Those are the three states of the system that are crucial to describing its motion. Again, the unicycle is non-holonomic because it cannot move sideways. Configuration space, is the space of all its configuratsions. q dot is the velocity vector, and assuming a quasi-static view (disregarding the dynamics) can be viewed as having two controls (i.e. these are the two control inputs to moving the unicycle along its trajectory) namely turn un place (u1) and move forward (u2) along the angle in which the unicycle is facing.

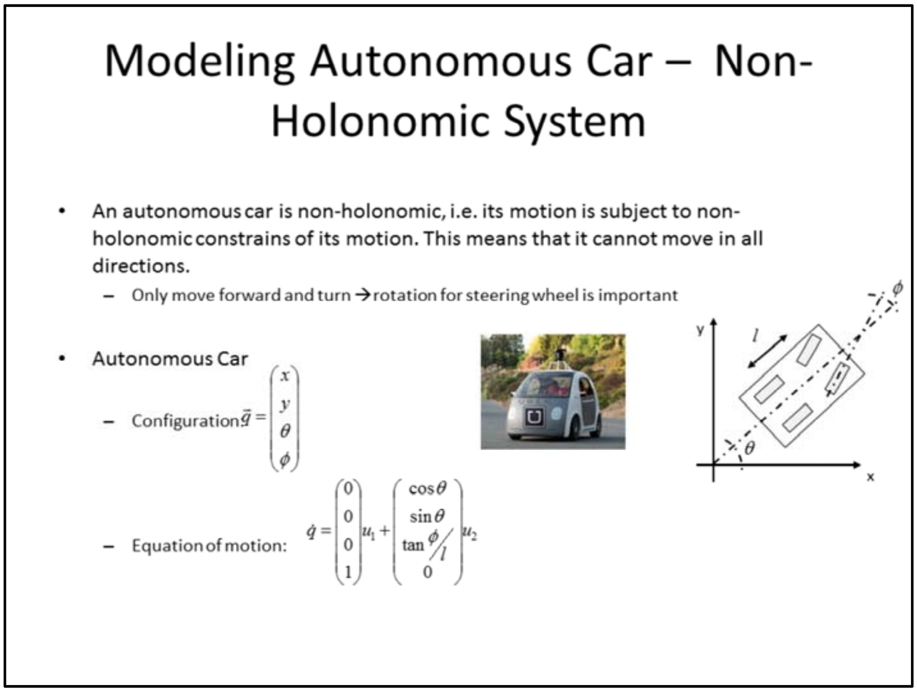
Modeling Autonomous Car – Non-Holonomic System

- An autonomous car is non-holonomic, i.e. its motion is subject to non-holonomic constrains of its motion. This means that it cannot move in all directions.
  - Only move forward and turn

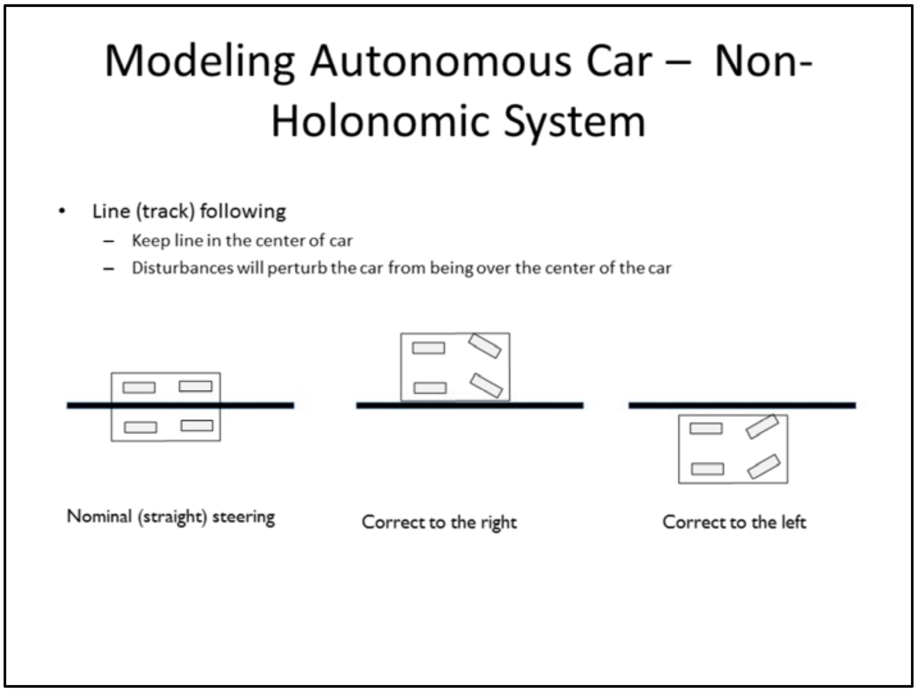


- Simplified Systems: Unicycle
  - Configuration $\vec{q} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$
  - Equation of motion: $\dot{q} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u_1 + \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} u_2$

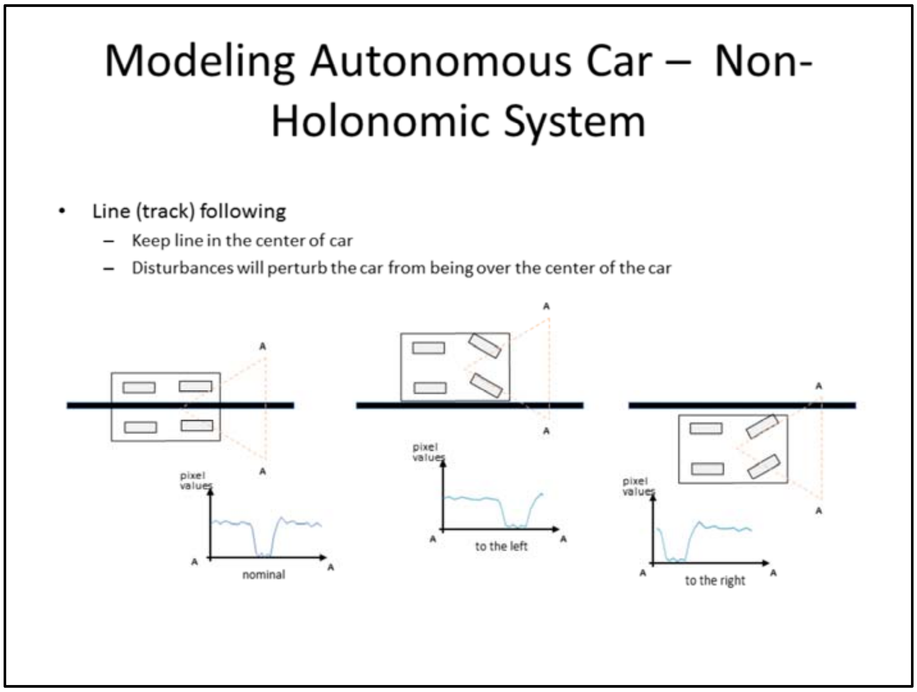
In this case, the wheel of the unicycle rotates from angle theta 1 to theta 2



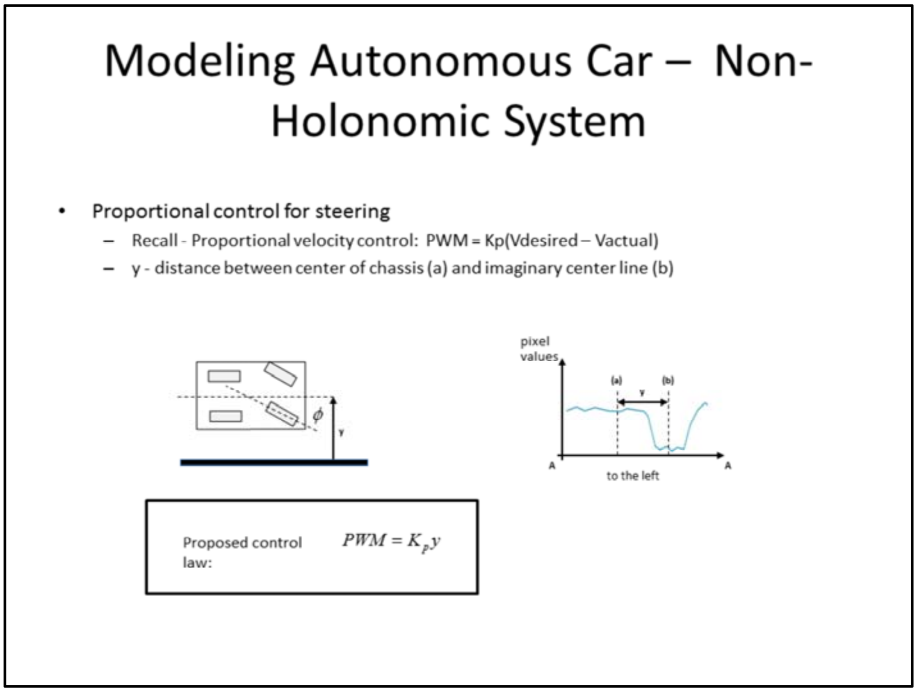
When it comes to describing the equations of motion for a car, things get more complicated, because there are more wheels involved ! The turning is now decided by the state of which direction is the steering wheel (i.e. the wheels) are pointing. That is decided by the angle phi. The control u1 is essentially the output from the servo, which determines which way the wheels are turning. The control u2 is essentially the velocity that is set by the car through PWM and the drive motors, as described earlier.



In order to keep the autonomous car to follow the line, the control system needs to ensure that the line remains underneath the center of the car. The steering wheels are used to correct for the disturbance, that would move the car to either side of the line. If the car moves to the left of the line, then the wheels should correct to the right, and if the car moves to the right of the line, then the steering should overcorrect to the left.



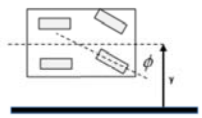
The position of the car, with respect to the line is ofcourse recorded by the line camera, in the form of pixel value. Consequently, as the signal coming from the line sensor starts moving away from the imaginary center line, the wheels should correct to point to the center. Note that a) it is important to make sure that the focus is such that the line is sharply visible, and b) once the line is outside the field of vision, the control system can no longer determine the distance to the imaginary line, and the system breaks down.

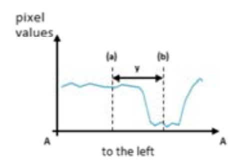


Recall that for the velocity control, proportional control was PWM = Kp times the difference between an actual and a desired velocity. For the steering, we can estimate the distance to the center line using the line camera. The distance y can then be used to steer the servo such that the wheels point towards the line. The farther away from the line (the larger y) the more do the wheels turn (feedback).

Modeling Autonomous Car – Non-Holonomic System

- Proportional control for steering
  - Recall - Proportional velocity control: PWM = Kp(Vdesired – Vactual)
  - y - distance between center of chassis (a) and imaginary center line (b)



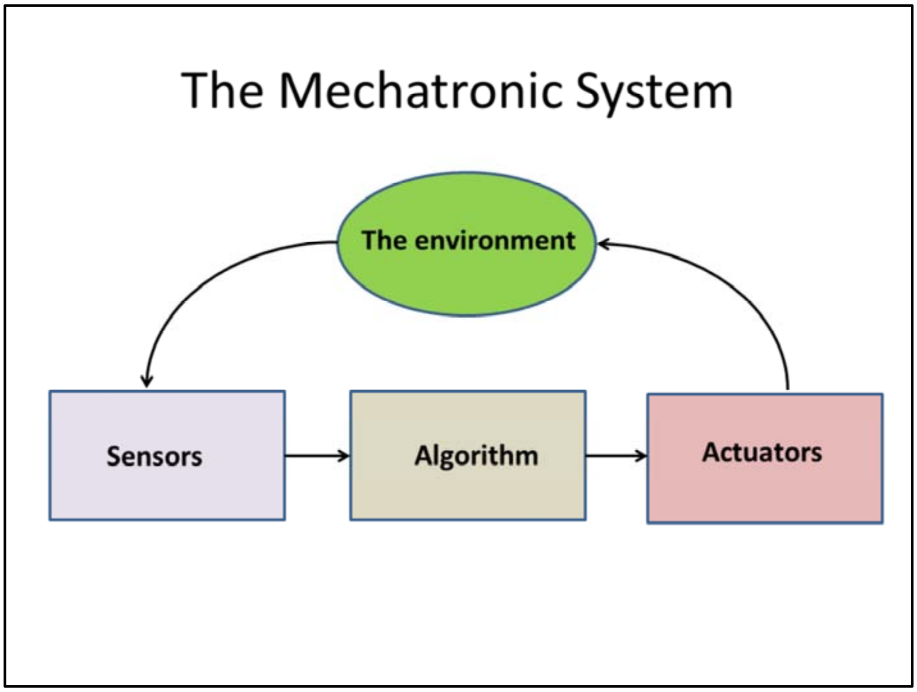


Proposed control law: $PWM = K_p y$

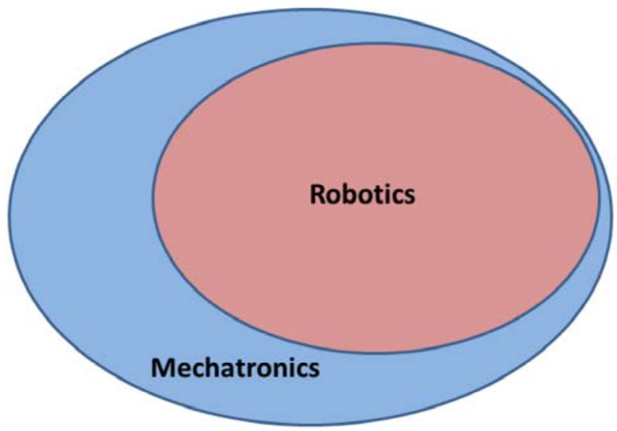
BAD

Recall that for the velocity control, proportional control was PWM = Kp times the difference between an actual and a desired velocity. For the steering, we can estimate the distance to the center line using the line camera. The distance y can then be used to steer the servo such that the wheels point towards the line. The farther away from the line (the larger y) the more do the wheels turn (feedback).

Exam Review


The Mechatronic System
The environment
Sensors
Algorithm
Actuators

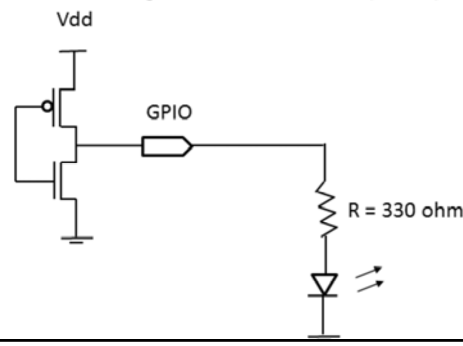
Mechatronics vs. Robotics



Vague distinction: Inputs provided vs. self-obtained

General Purpose I/O's (GPIO)

- General Purpose Input Outputs (GPIO)
  - reconfigurable Input/Output
  - digital (1 (high) or 0 (low))
  - need to be configured in software (lab 2)

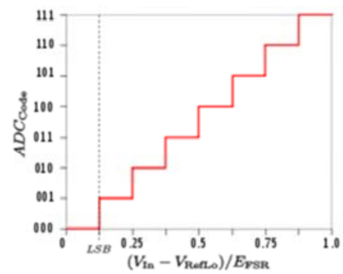


Analog to Digital Converters (ADCs)

- Analog to Digital Converter
  - Can read in the analog value
  - Input pins need to be configured

- FDM board uses:
  - 16-bit Successive Approximation (SARC) ADC
  - 12-bit DAC

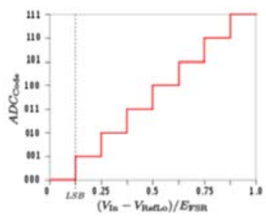
Use processor Expert

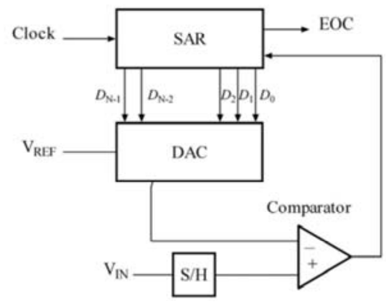


from Wikipedia

ADCs

- Successive Approximation ADC
  - Allow the Microcontroller to sample an analog waveform
  - Common type of ADCs
  - S/N: Sample and hold register
  - SAR: Successive Approximation Register
  - DAC: Digital to Analog Converter
  - EOC: end of conversion





from Wikipedia

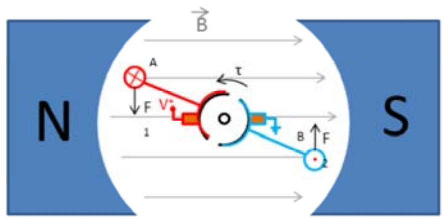
Timers and Interrupts

- Interrupts:
  - Allow code execution on an event.
  - External interrupts or software interrupts (timers)
- Timers:
  - Allows code to be triggered after some elapsed time
  - Can be used to trigger PWM
  - Sleep timer (low power mode)

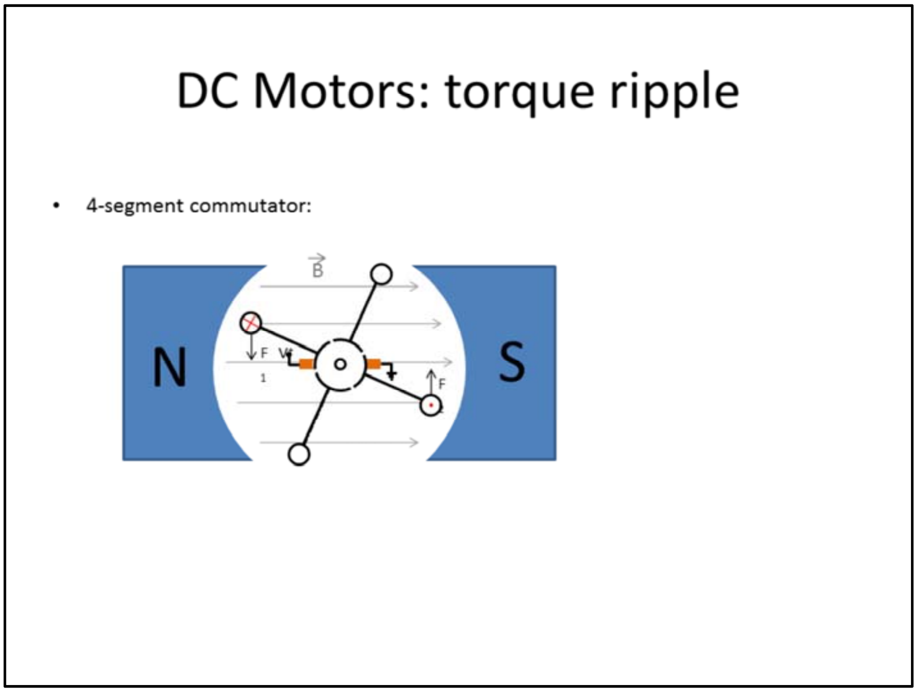
Set in processor Expert

from Wikipedia

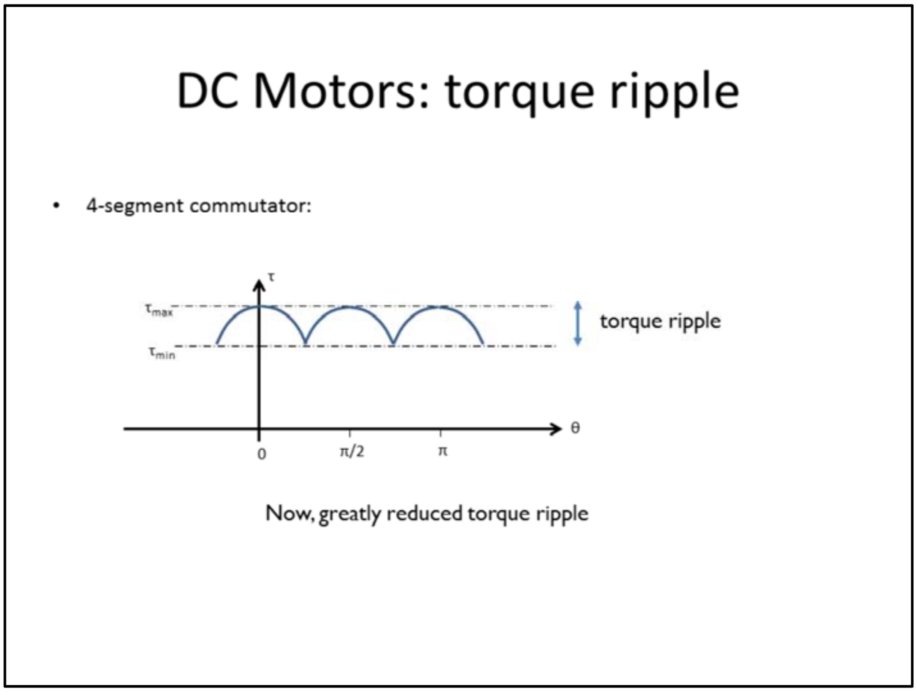
DC Motors: commutator



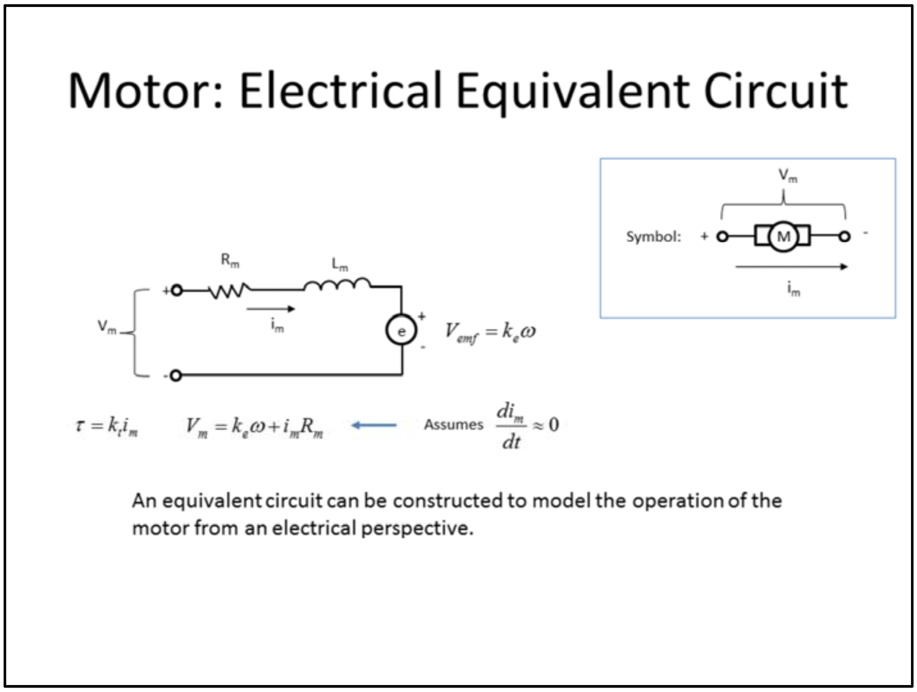
[Need a bit of touchup] As the current is flowing into the wire cross section A, and out of cross section B, the tourque Tau is counter clockwise.



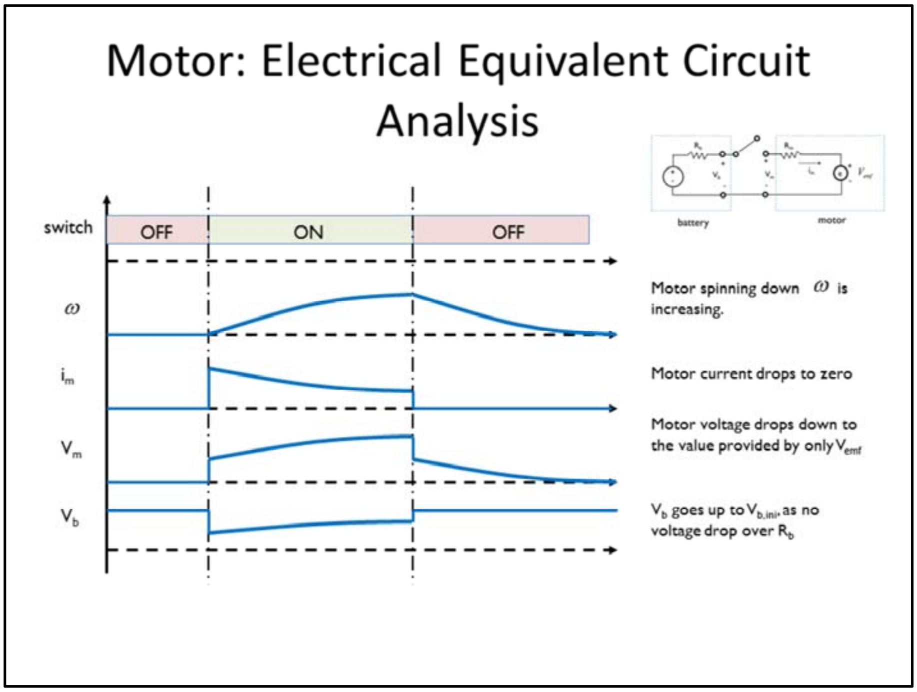
4-segment commutator dramatically reduces the torque ripple



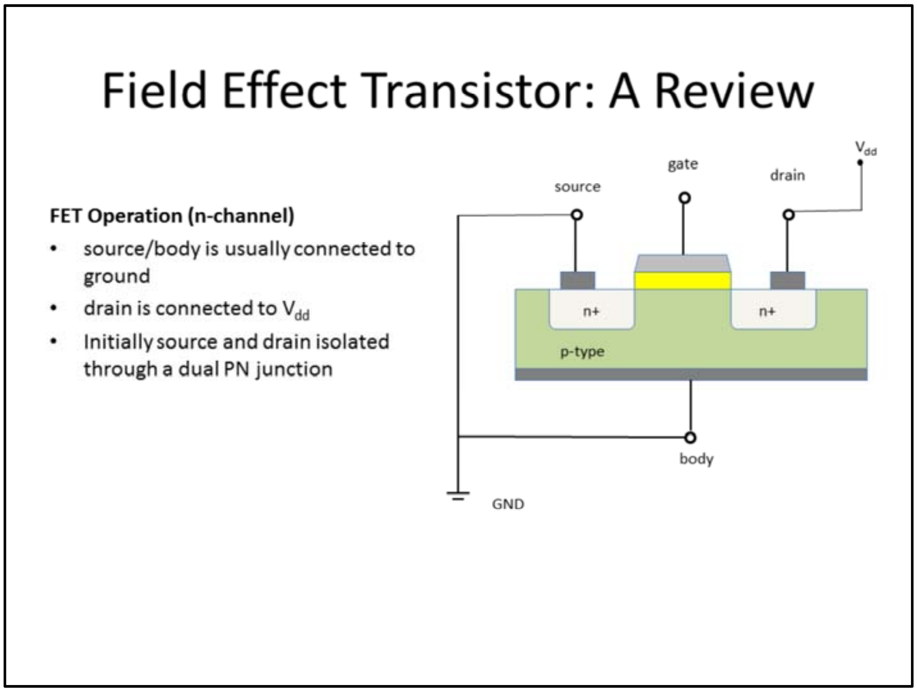
The 4-segment commutator helps to dramatically reduce the torque ripple.



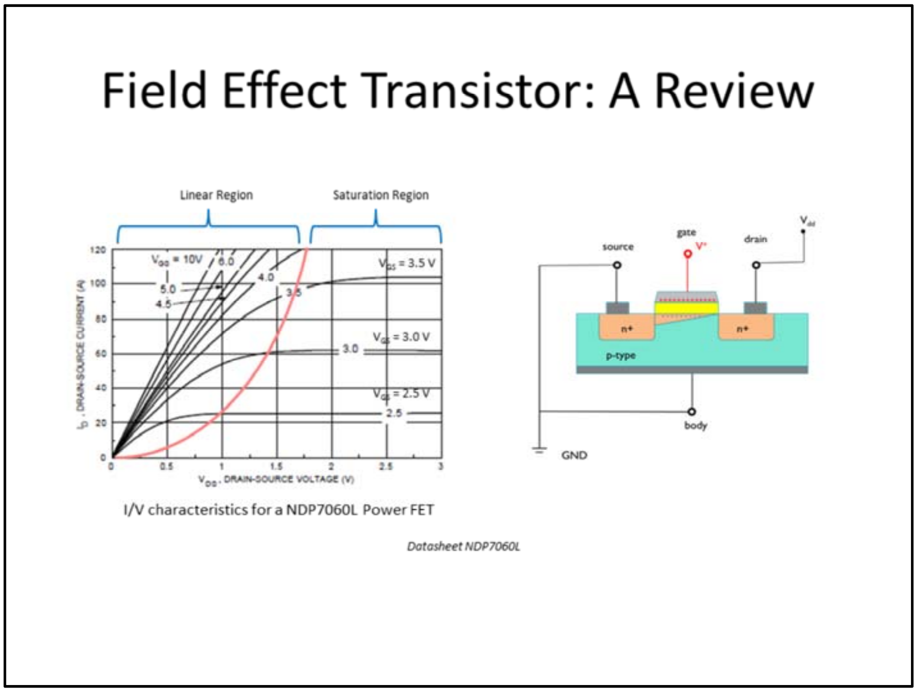
An equivalent circuit can be constructed that models the operation of the motor from an electrical perspective, i.e. the relationship between the motor current and the motor voltage. The insides of a motor consist of a winded wire (coil), which has some internal resistance, called **winding resistance** of the motor, R_m. The wire is also an inductor, with some **motor inductance** L_m, however the motor inductance only comes into play when the current changes abruptly, we come back to this when it comes to snubbing. However, a DC motor is also a generator, because as the shaft turns, the winding moves through a magnetic field (B), generating a reverse potential across the windings, which counteracts some of the applied voltage Vm, and reduces the motor current i_m. This voltage is called **back EMF** (electromotive force), V_emf. V_emf is equal to some constant k_e times the angular velocity of the shaft omega (which corresponds to how fast the coil wires move through the magnetic field). At steady state, or when the current does not change too fast, the Vm is equal to the sum of the voltage drop across the winding resistance *and* V_emf. Note that V_emf limits how fast the shaft will spin, i.e. if the shaft increases in its angular velocity, eventually all applied voltage V_m will be negated by V_emf. In other words, a DC motor is self-limiting in angular velocity.



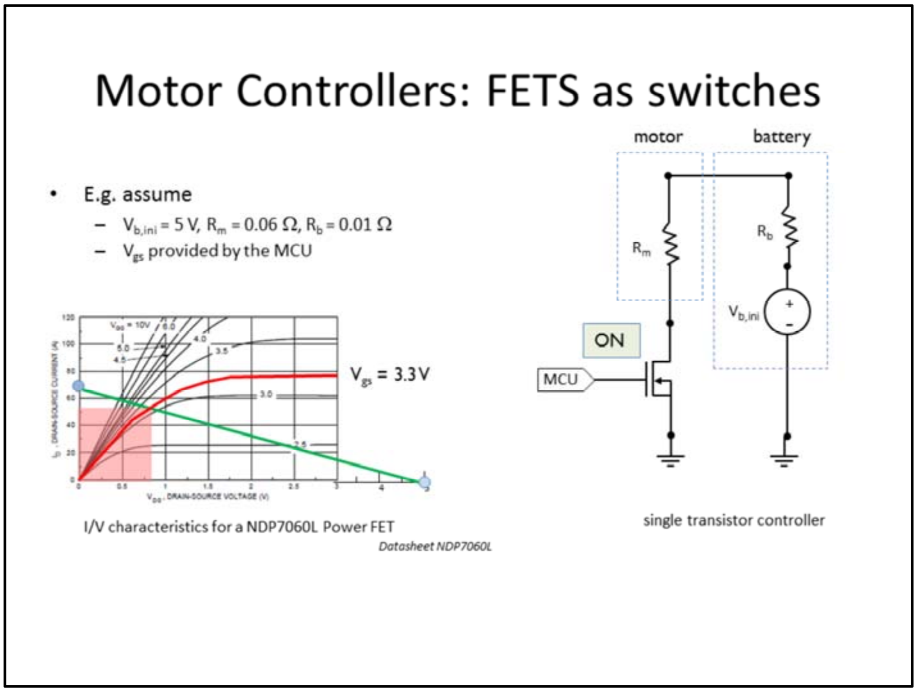
As the switch is open, the motor current is interrupted. The motor now starts to spin down, depending on losses and friction in the system. Motor voltage is reduced, however the motor is now working as a generator, with the only contribution being the back EMF, i.e. V_emf. V_b is increased to V_b,ini, since there is no more drop across the internal resistance R_b. Note that this diagram does not include the voltage spikes due to changes in i_m and interaction with L_m, which for this purpose can be ignored, they are operating at a much faster timescale.



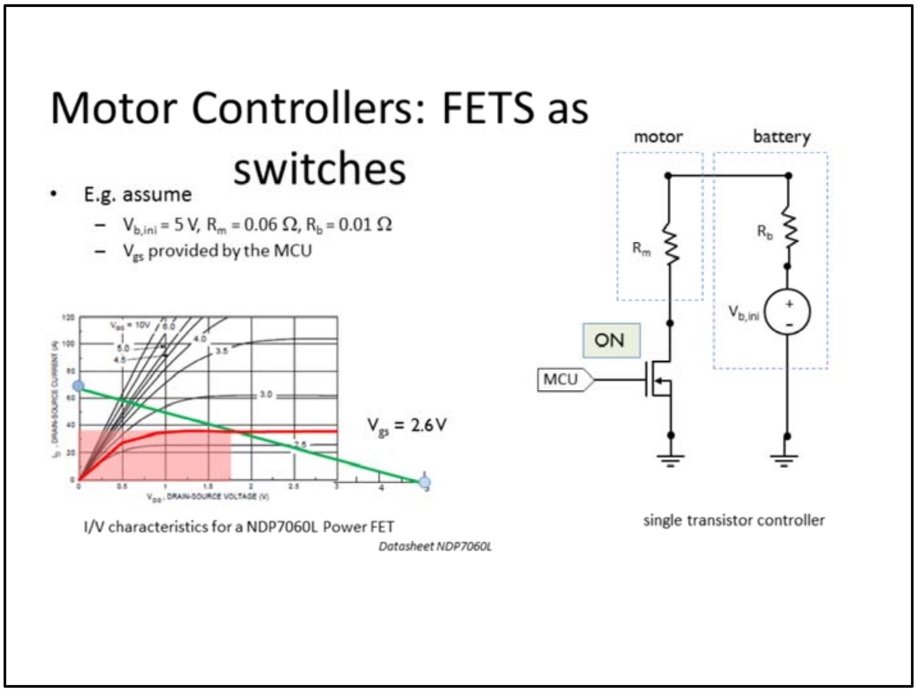
The operation of the FET and the formation of the channel is explained in the next few slides



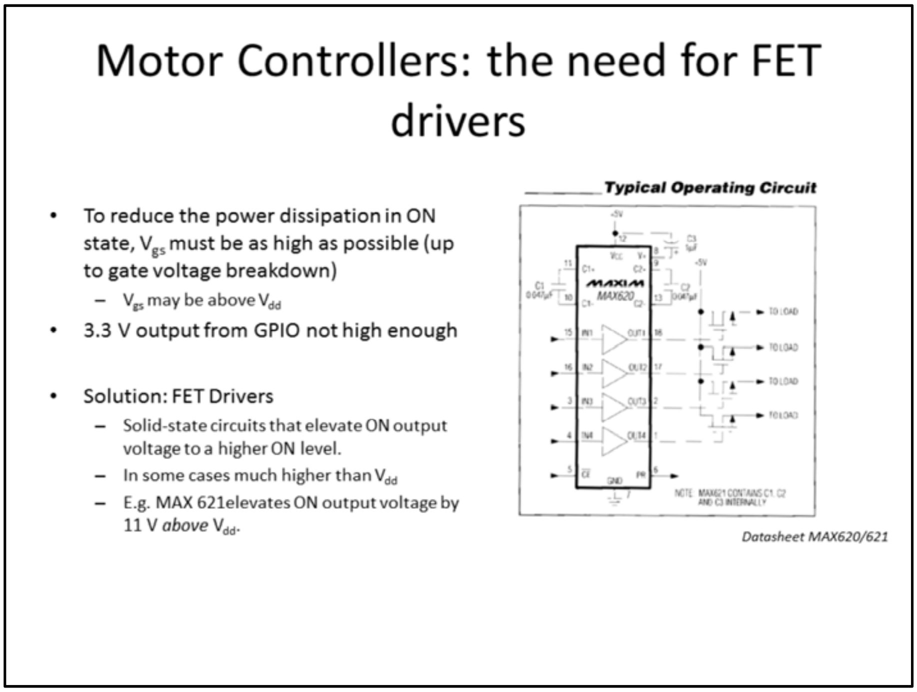
The operational characteristics for a a field effect transistor can be described using a V/I curve. The drain to source current (i_ds) can be seen as a function of drain to source voltage (V_ds) for a set gate voltage V_gs. In the linear region i_ds increases linearly with V_ds, and the FET operates in the **linear region**. Once V_gs gets large, the channel is pinched at the drain, and does not allow more lectrones to flow through it. At that point, i_ds does not increase anymore with V_gs, and the FET operates in the **saturation region**.



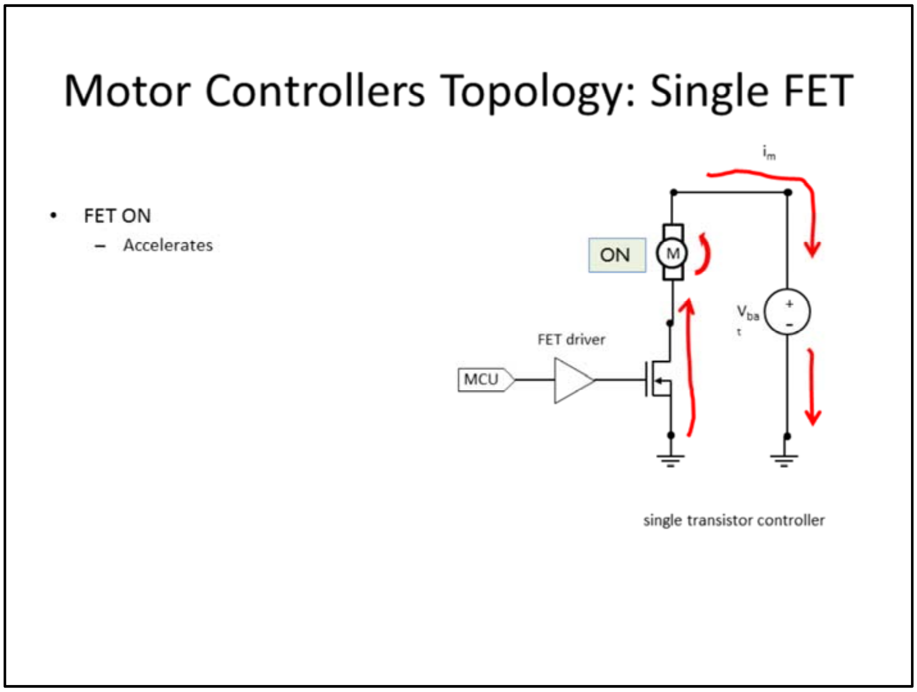
The load line illustrates a problem with connecting the GPIO to the gate of the FET directly. Assume the GPIO can deliver 3.3 V output, the resulting point on the load line corresponds to approximately 0.77 V Vds and 52 A ids. The rectangle under the load line, covered by the 0.77 Vds and 52 A ids is 40 W. This is quite high power.



Further more, it turns out that drain may not be at ground, for example in H-bridge topologies like we will observe later in this module. If the drain of the FET is elevated by 0.7 V above ground, then the Vgs would now be at 2.6 V, and ids would be approximately 35 A, and the power dissipation in FET at 61 W !



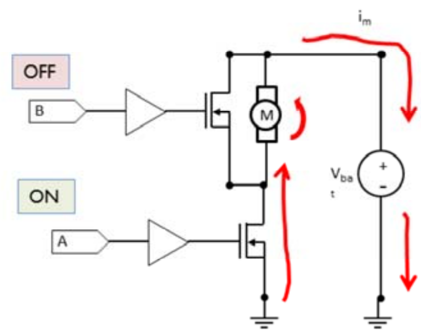
However, the higher the Vgs, the higher on the loadline we get in the ON state, and thus less power dissipation in the FET. Remember, we ideally want the FET to be completely ON. In fact, the Vgs can go above Vdd to provide the required gate voltage and alleviate the concern when the source is not at ground. In a motor controller, this is often handled by a circuit called a FET driver. For example the MAX621 circuit outputs the high signal from its input at 11 V above Vdd !



A single FET controller allows for the motor to accelerate (torque = i_m)

Motor Controllers Topology: Half-Bridge

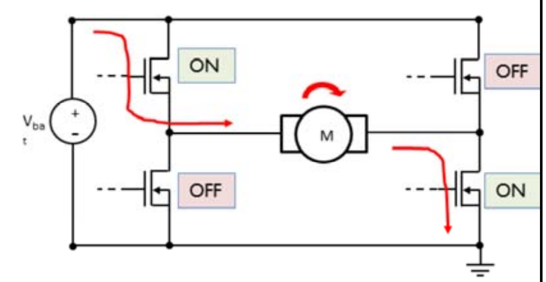
- Accelerate:
  - A ON, B OFF



Half-bridge

Motor Controllers Topology – H-Bridge

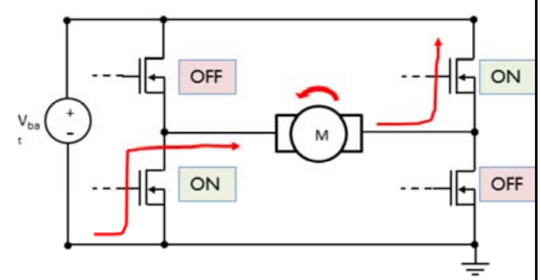
- The most versatile motor controller topology
- Requires four (4) FETs per motor
- Supports:
  - Motion forward



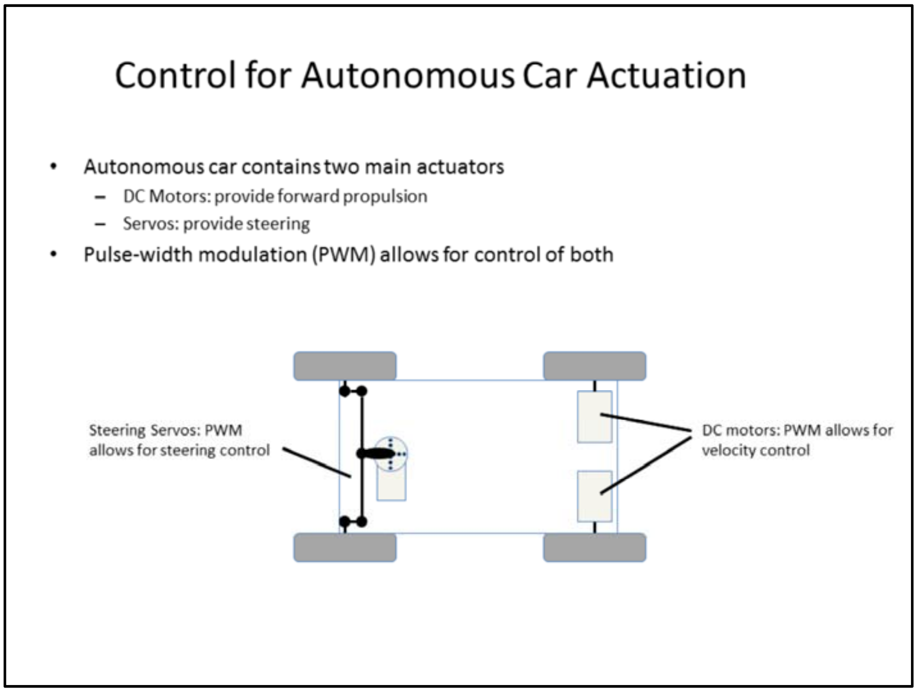
The motor can now spin forward

Motor Controllers Topology – H-Bridge

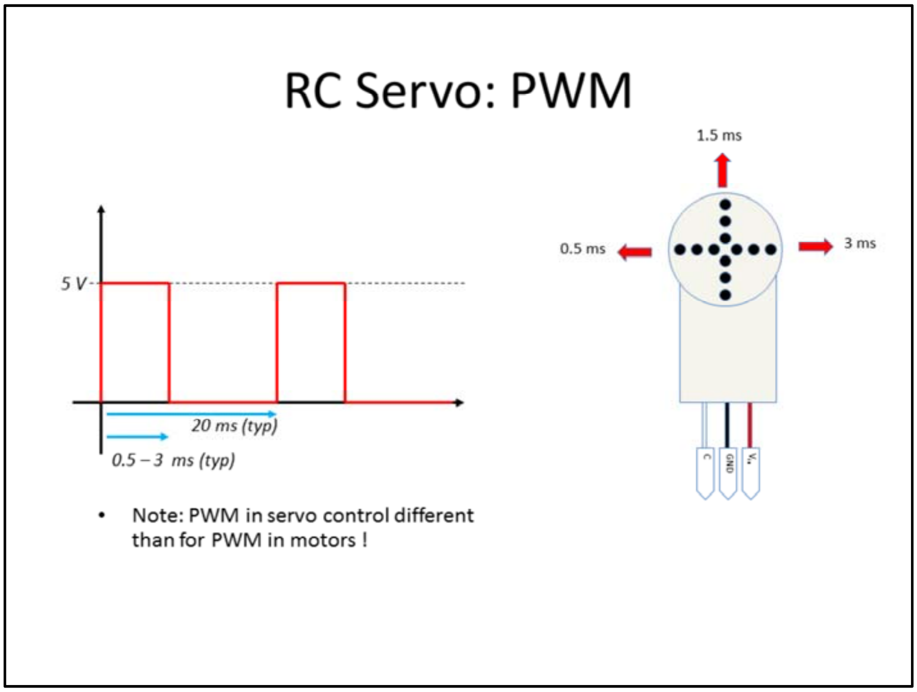
- The most versatile motor controller topology
- Requires four (4) FETs per motor
- Supports:
  - Motion forward
  - Motion backward



By changing which transistors are on, the polarity of the motor can be reversed, i.e. its direction of motion.



The autonomous car consists of dc motors that provide forward motion, and DC servo that provide steering capabilities. Both can be controlled using pulse-width modulation.



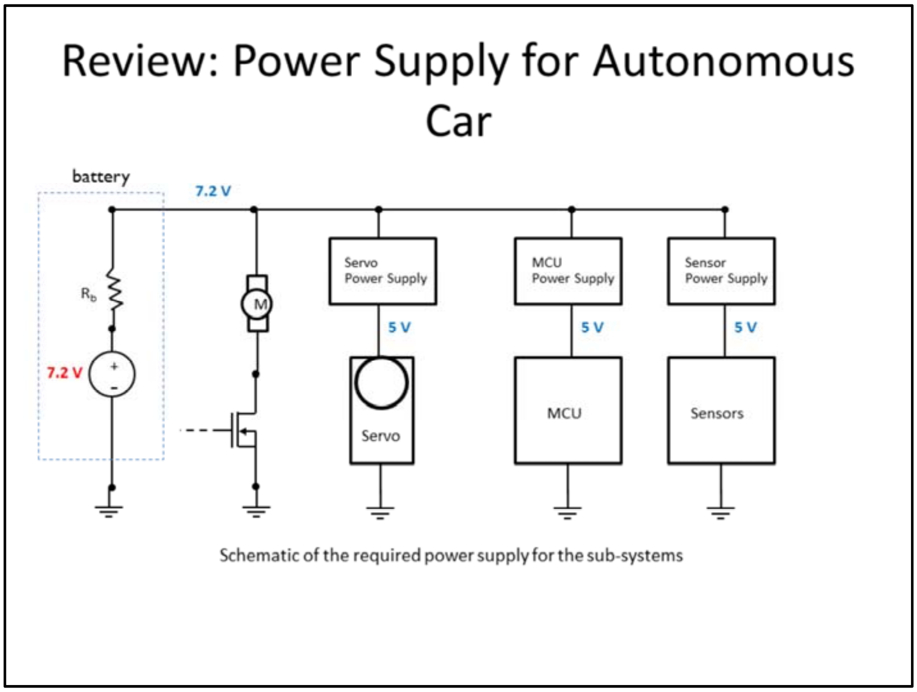
Pulse-width modulation (PWM) can be successfully used to control the servo direction, but the principle is slightly different. The servo uses the length of the pulse to determine the orientation of the output shaft of the servo, as opposed to the actual duty cycle with respect to velocity control

Review: Power Supply for Autonomous Car

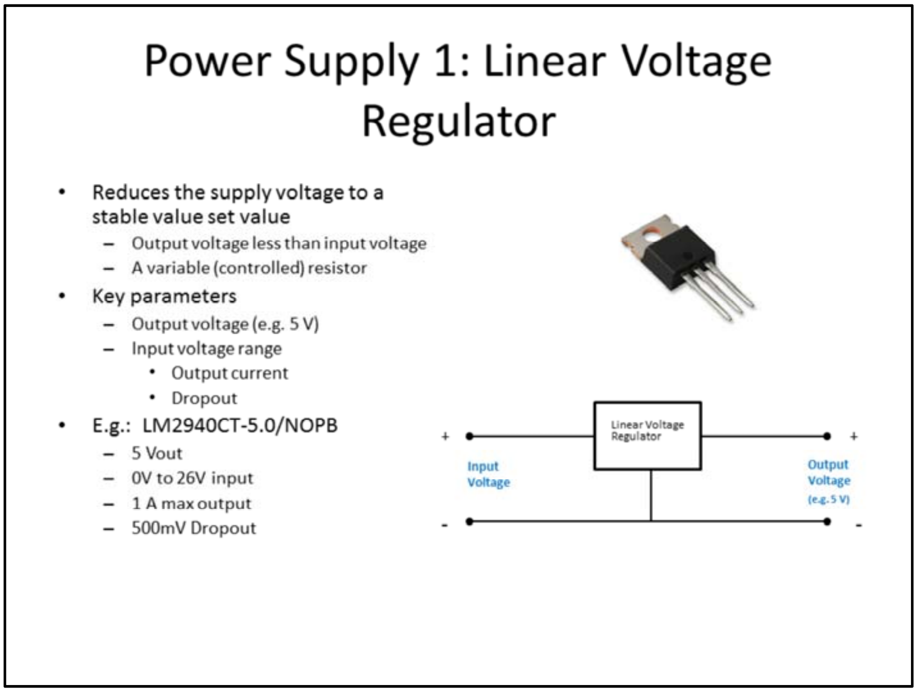
- Different power needs for various sub-systems
- On-board power supply needed to:
  - Drive the motors (large currents)
  - Drive the servo (moderate currents)
  - Power the MCU (low currents, voltage stability)
  - Power the sensors (low currents, voltage stability
- Battery is the main reservoir of on-board power
- Many battery types
  - Most common types for electric vehicles are Lithium ion batteries (high energy density, moderate cost, large discharge/charge cycle life)

http://www.wirelessmadness.com/

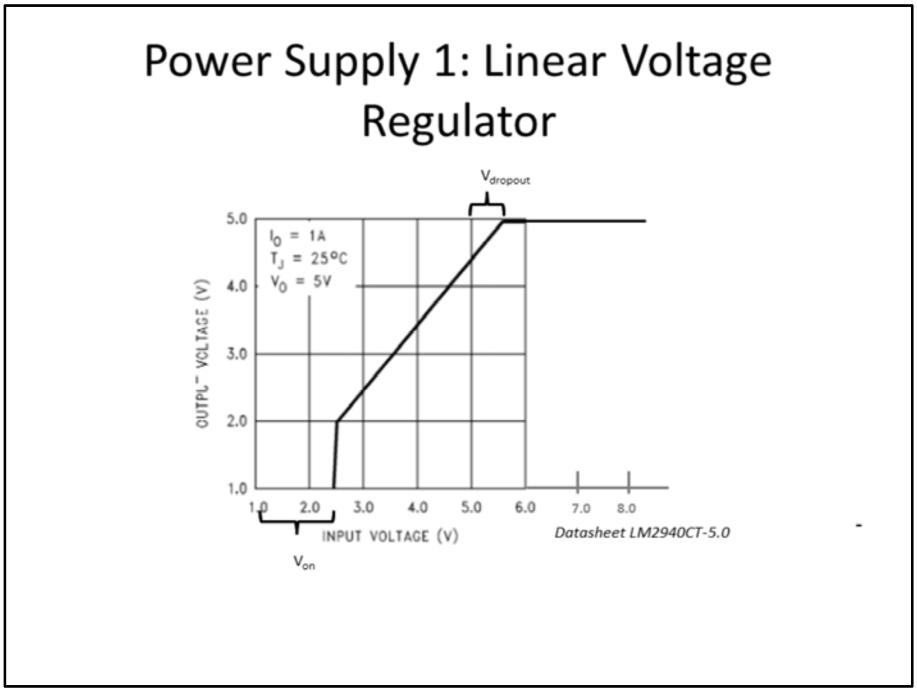
The power supply is an essential component of the autonomous car, because it ensures seamless operation of all the components while the car is driving on the track, i.e. is disconnected from any external power supply. It needs to supply the power to all the different sub-components of the system. Some of the components have different requirements. For example, a lot of power (high current) might need to be supplied to the motors to provide fast acceleration. On the other hand, the microcontroller is concerned with voltage stability rather than the ability to dissipate large amount of power. The power supply may need to be designed separately to meet these different requirements.
A battery is usually the main supplier of power for on-board power. There are many types of batteries, but the most frequently used battery for autonomous vehicle applications is Litium-ion battery, due to its relatively moderate cost, high energy density, and long lifetime.



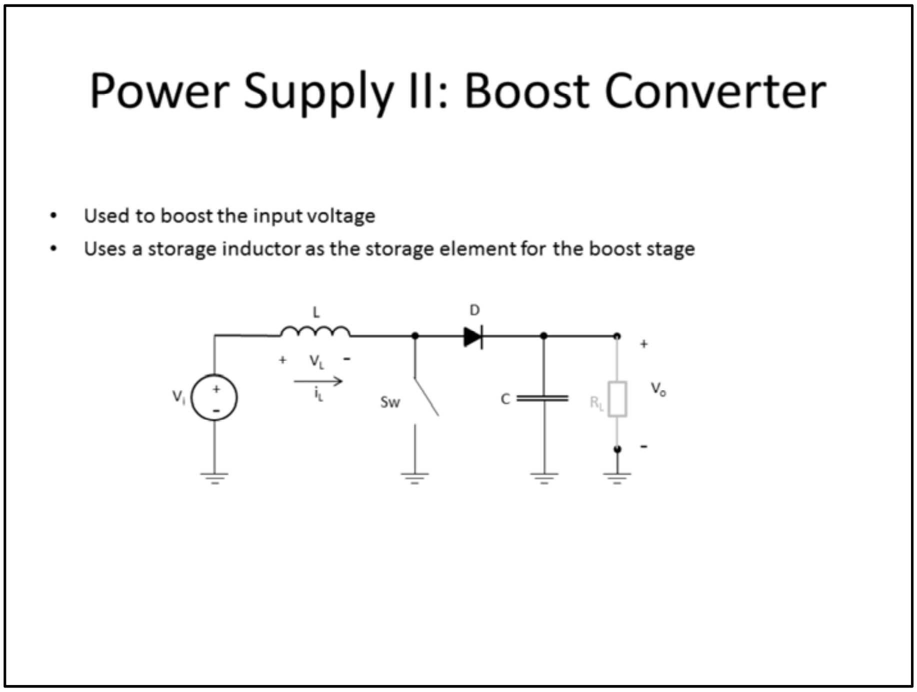
Lets consider a 7.2 V Li-Ion battery that will be the power reservoir for the autonomous car power supply. Let us assume that all the electronics require 5 V DC, so des the servo. The DC motor is not sensitive to voltage as much as it requires current to operate, so that can be considered separately. Ignoring for now the voltage drop across the resistor, the battery will output approximately 7.2 V, while the Servo, MCU and Sensor electronics require 5 V. Consequently, the principal job of the servo, MCU and Sensor power supplies is to reduce the battery voltage to a stable 5 V DC. This can be achieved in a simple way using a **linear regulator**.



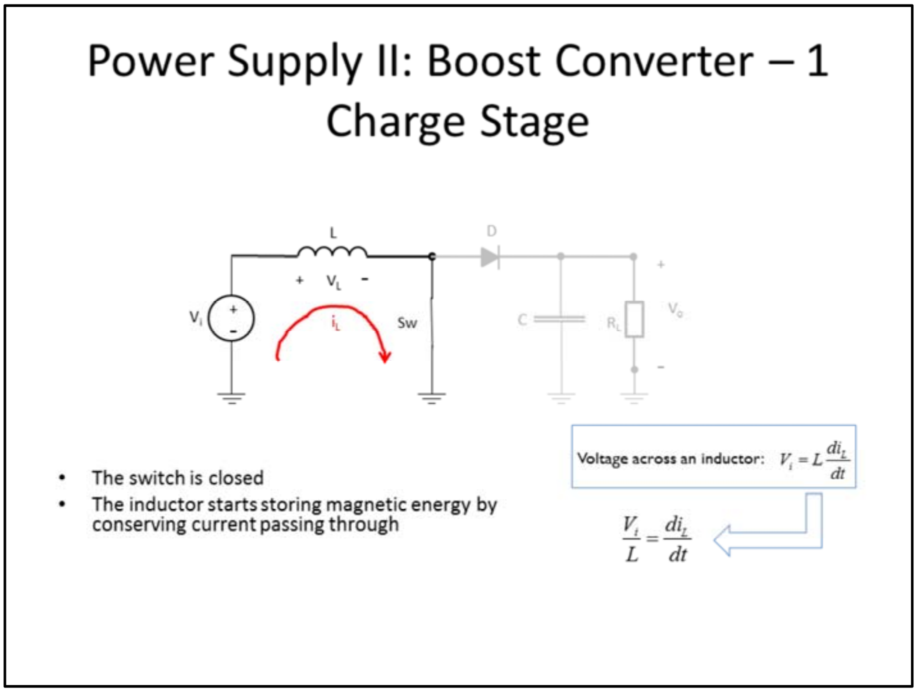
A linear regulator is essentially a variable load, which provides stable output voltage by providing enough voltage drop from the input voltage to a prescribed value. They are simple, reliable, and provide a fixed output. Parameters to watch for when selecting a linear regulator are: a) output voltage, this has to match what we want/need as output of the power supply; b) input voltage range that describes what input voltage range is tolerated by the device, usually some maximum voltage; c) maximum amperage rating, i.e. how much current the unit can supply, d) and drop-out. Drop-out is the minimum voltage drop that is taken up by the control circuit in the device before the device to start operating. We usually look for a low dropout regulator. At high current and high voltage drop across the regulator, the device will dissipate significant power (e.g. 5V output, 20 V input at 1 A, the device will dissipate 15W, so a heat sink is needed.



The output voltage of the linear regulator can be illustrated by the following relationaship between input and output voltage. The linear regulator turns on at a specific on voltage, V_on. The output voltage then increases linearly with the input voltage as long as the input voltage is below the designated output voltage of the regulator. When the input voltage exceeds the desired output voltage, the output voltage does not increase anymore. Note that the output voltage always trails the input voltage by 0.5 V (in the case of the LM2940). This is the dropout voltage Vdrouput. In all cases, the input voltage > designated output voltage + Vdropout for the regulator to function properly.

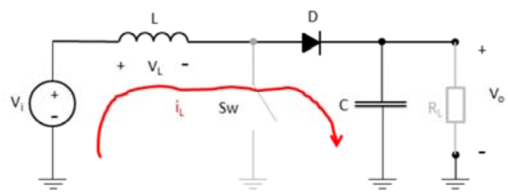


The boost converter uses an inductor (L) as the storage element. The boosted voltage is stored in an output capacitor C before it is delivered to the load R_L. A diode D ensures that the supply output capacitor only discharges through the load. The boost converter consist of two cycles, 1) the charge stage, and 2) the step-up stage.



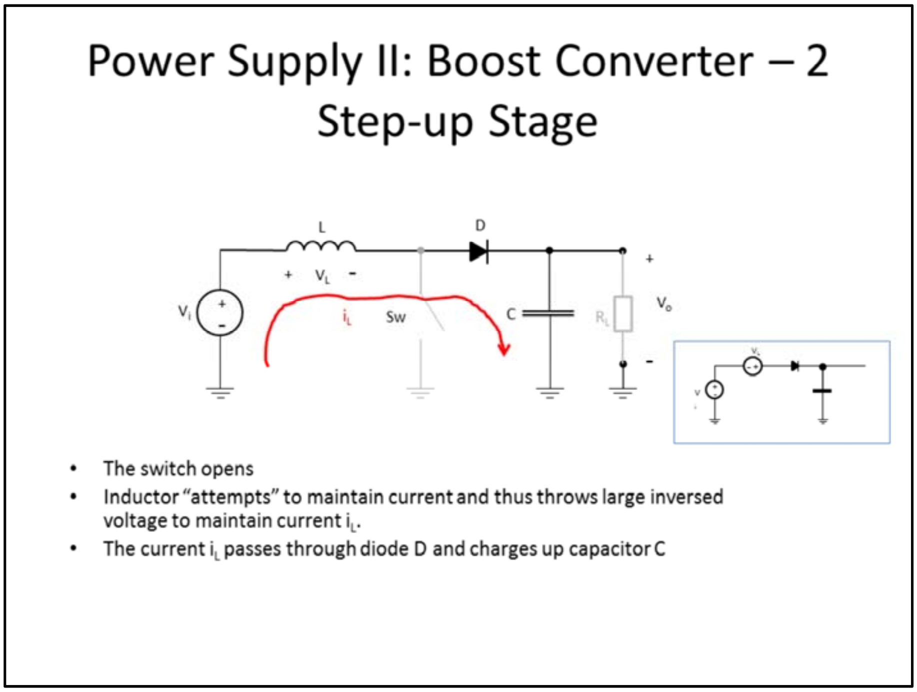
In the charge stage, the switch is closed, and the inductor is being charged with magnetic energy. The voltage across the inductor is set to input voltage Vi. The current through the inductor increases, as the magnetic energy is building up, and assuming that the time the switch is on is << than tau (LR), the current is increasing (di_L/dt) as Vi/L.

Power Supply II: Boost Converter – 2 Step-up Stage



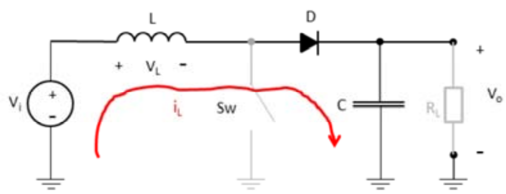
- The switch opens
- Inductor “attempts” to maintain current and thus throws large inversed voltage to maintain current i_L.
- The current i_L passes through diode D and charges up capacitor C

The switch is opened to start the step-up stage. As the switch is opened, the inductor attempts to maintain the current flowing though it by changing its voltage. A rapid negative change in current, due to the opening of the switch, will result in a large negative voltage across the inductor (negative according to the V_L definition.



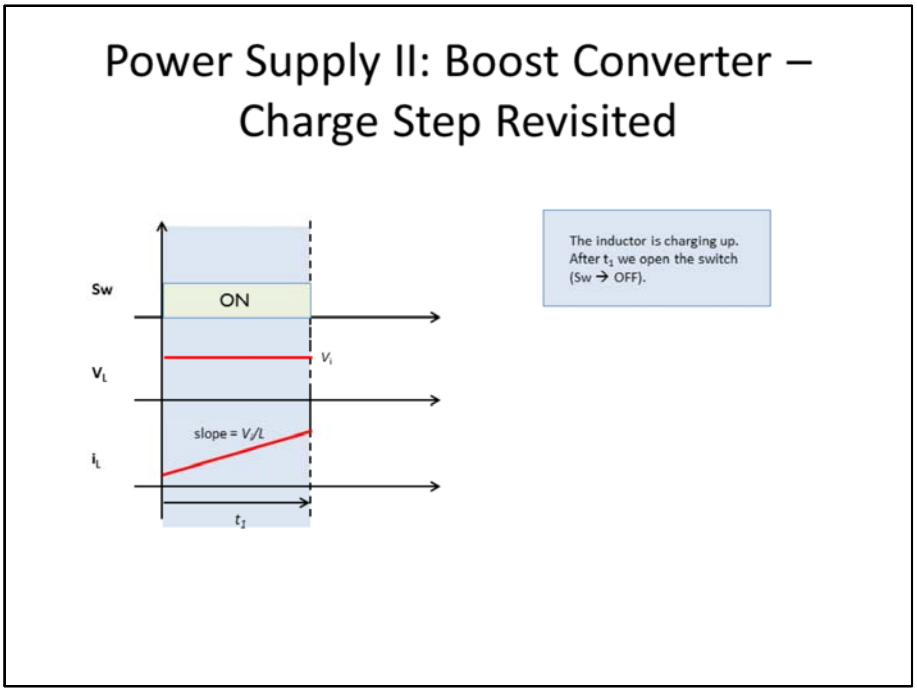
The voltage V_L adds to Vi, the added voltage is larger than Vi, even larger than V_C, because that is the only way the inductor can maintain the current across L. Consequently, charge is transferred to the capacitor C, at a higher voltage than Vi.

Power Supply II: Boost Converter – 2 Step-up Stage



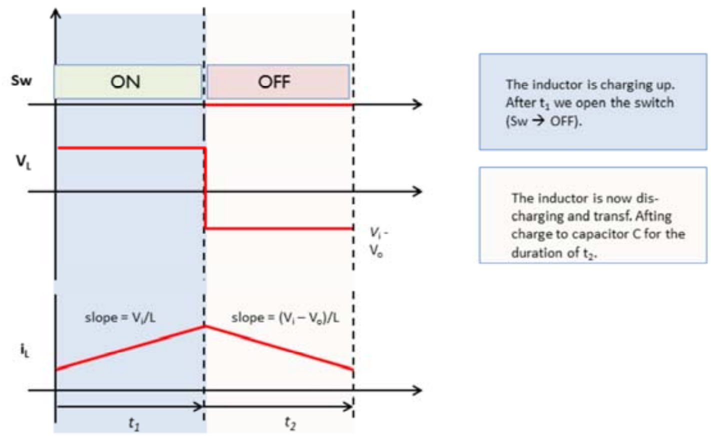
- The step-up current (slope) through the inductor is now: $\frac{V_i - V_O}{L} = \frac{di_L}{dt}$
- Inductor is discharging

The slope of the current through the inductor is now (V_i – V_o)/L, assuming that V_o does not significantly change over a cycle. This is a fair assumption due to a large storage capacitor at the output.

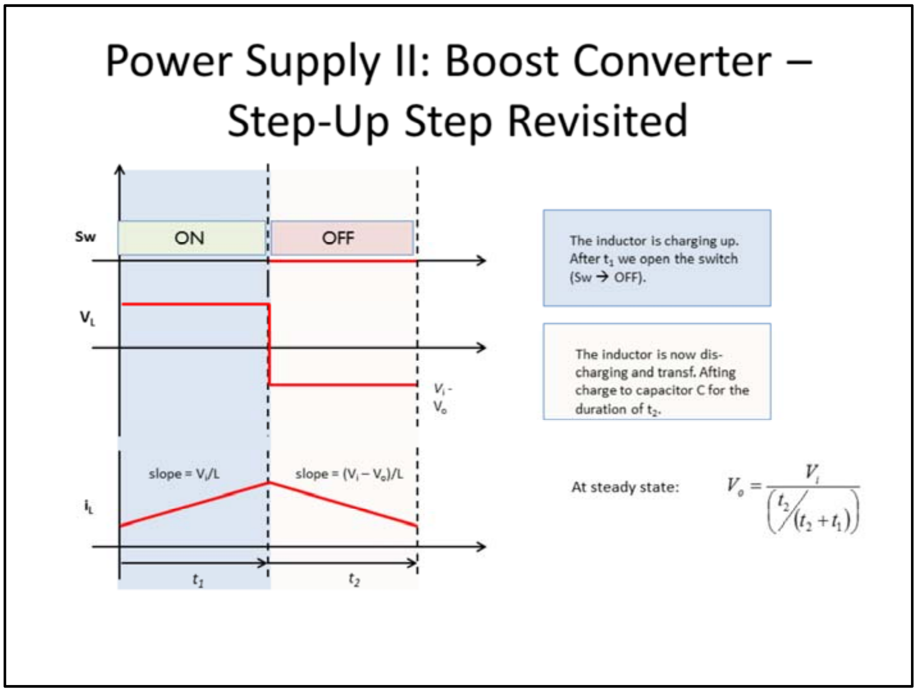


This diagram shows the voltage and current across the inductor L as a function of switch position. The switch is closed at t = 0, and remains closed for t_1.

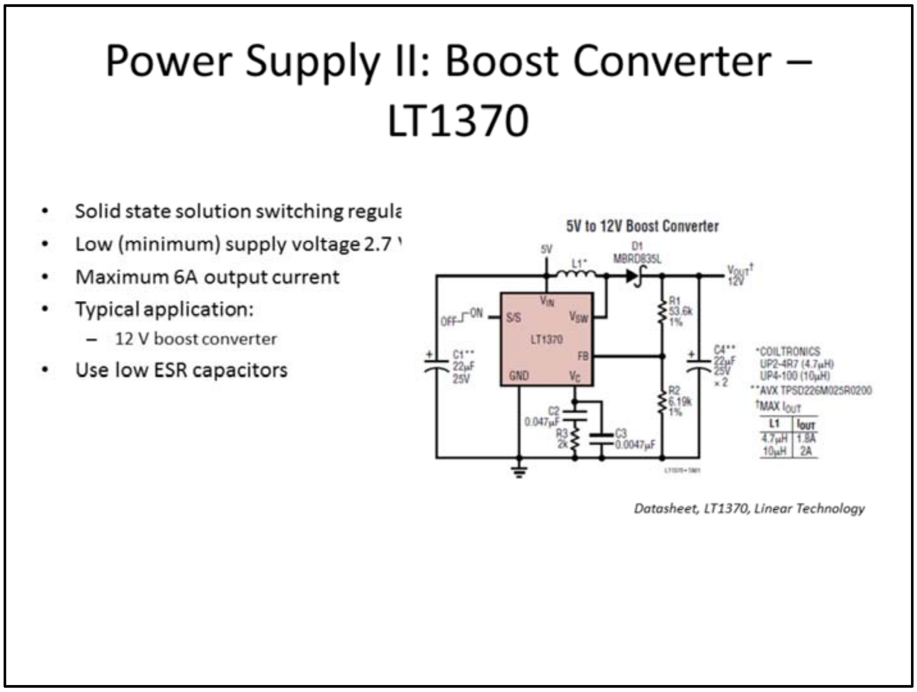
Power Supply II: Boost Converter – Step-Up Step Revisited



As the switch is closed, the current over the inductor is being reduced, and the energy is being transferred to the output storage capacitor. The voltage over the inductor is now Vi-Vo, and the current change is (V_i – V_o)/L. Note that V_o is larger than V_o, and so the change is negative (the capacitor is discharging)



By making the energy analysis, i.e. analyzing the energy transferred from the inductor at each stage, and realizing that the energy charged and discharge in each cycle must be equal, we can calculate the voltage at the output to be inversely proportional with the duty cycle of the discharge step, i.e. t_2/ (t_2+t_1).



The solid-state switching regulator LT1370 is a convenient DC-DC converter that can boost voltage starting with voltage as low as 2.7 V. The typical configuration is to boost up voltage to 12 V, which is presented on the datasheet. It is important to use low ESR (equivalent serial resistance) capacitors, however the datasheet recommends using tantalum capacitors, except for input and output resistors, which should be electrolythic. Remember to rate the maximum voltage of the electrolythic capacitors to at least 2x of the output/input voltage to avoid spikes. Consult the datasheet about the recommended minimum path between the diode D1 and output capacitors.

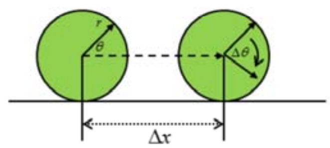
Optical Rotary Encoders and Velocity Sensing

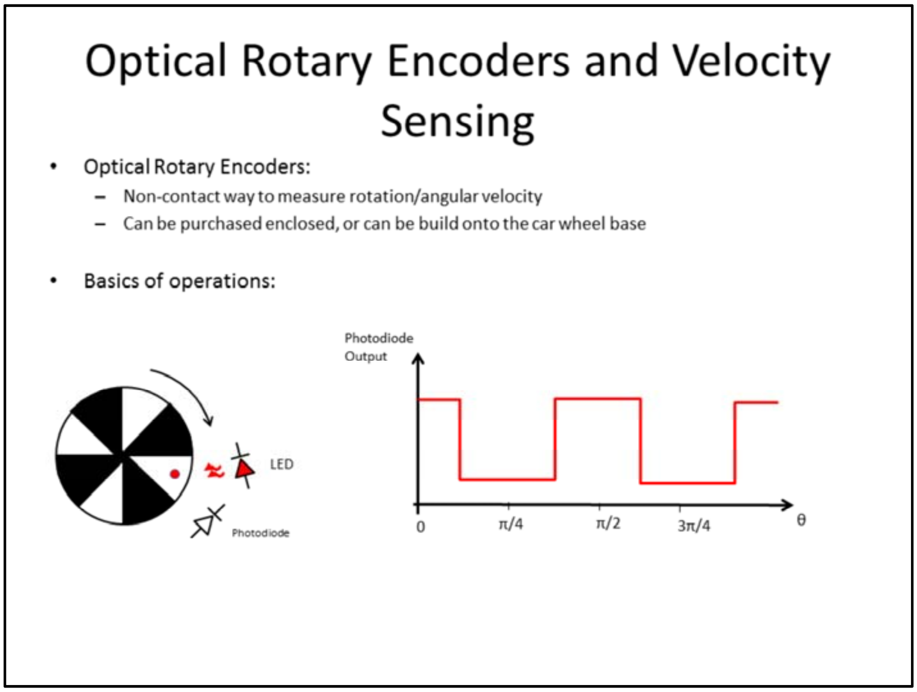
- Velocity sensing is necessary for a car to reach a set velocity
  - Recall $\tau \propto i_m$
  - To reach the desired velocity, the car has to accelerate, i.e. increase i_m
  - Once desired velocity is reach the car has to coast, reducing i_m to counteract friction and drag
  - I_m must be larger to maintain same velocity if traversing an incline
- Velocity = distance / time
- Assuming a no-slip condition:

$$\Delta x = \Delta\theta \cdot r$$

- Resulting velocity:

$$v = \frac{\Delta x}{\Delta t} = \frac{\Delta\theta}{\Delta t} \cdot r = \omega \cdot r$$





An optical rotary encoder can be used to measure the relative rotation, and velocity, of a wheel of an autonomous car. Assuming no-slip condition, the rotation of the wheel should correspond to the displacement and velocity of the car. The rotary encoder can be purchased already enclosed, with a shaft sticking out, however a lot of times it is easier to build an encoder onto the side of the wheel. What is needed is a colored wheel (as illustrated), illuminated by a focused LED, and a photodiode measuring the refection of the surface. Care has to be taken to avoid the illumination from daylight, so IR illumination and photodiode is always preferred. Sometimes, combo illumination driver and detector units, such as the HAMAMATSU S6809, can be used for this. As the wheel rotates, the detected illumination pattern is shown to the right. Each pulse (P) on the photodiode corresponds to pi/2 rotation of the wheel.

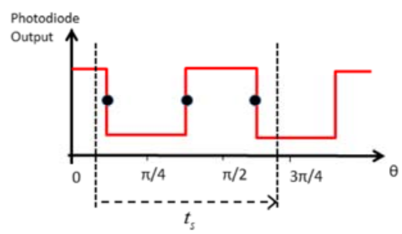
Optical Rotary Encoders and Velocity Sensing

- Count number of edges in a fixed amount of time:

$$\boxed{v = \frac{n\Delta\theta_{e-e}}{t_s} \cdot r}$$

where t_s is sampling time, n is the number of transitions, and θ_{e-e} is the angle between transitions, in this case π/4.

Error: $\pm\frac{\Delta\theta_{e-e}}{t_s} \cdot r$



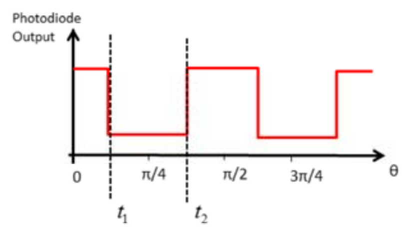
The velocity can be measured by counting number of transitions from an optical encoder (n) per a fixed unit time (t_s). This can be achieve by continuously incrementing a counter on each transition, and after a certain time looking at the counter value. Input can be continuously sampled, and transitions recorded (n is incremented if input_t-1 <> input_t), and a timer or timed interrupt is used to keep track of the time.
Maximum error can occur if the counter has been sampled or started just before a transition occurs.

Optical Rotary Encoders and Velocity Sensing

- Measure time between transitions:

$$\boxed{v = \frac{\Delta\theta_{e-e}}{t_2 - t_1} \cdot r}$$

where t_1 is the time of first transition, t_2 is the time of second transition θ_{e-e} is the angle between transitions, in this case π/4.



Error: $\pm\frac{\Delta\theta_{e-e}}{t_e} \cdot r$

where t_e is the sampling interval.

Measuring the time between the transitions allows for estimation of velocity by dividing the angular distance * r by the difference in time between the two transitions. To record the times of both transitions one can sample the input, and look for transitions from low to high. Alternatively, an interrupt can be set up on the transition. If sampling of input is used, then the resulting error is a function of the sampling interval - worst case, the input was just sampled as the transition occurred.