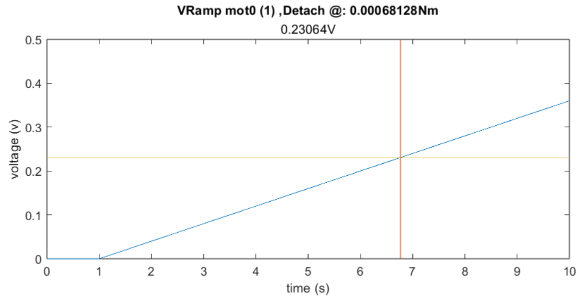
# Friction

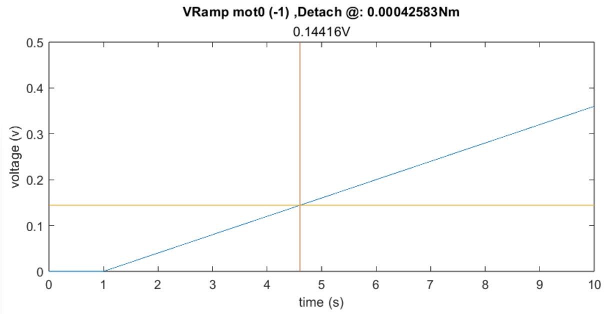
## 1) Static Friction

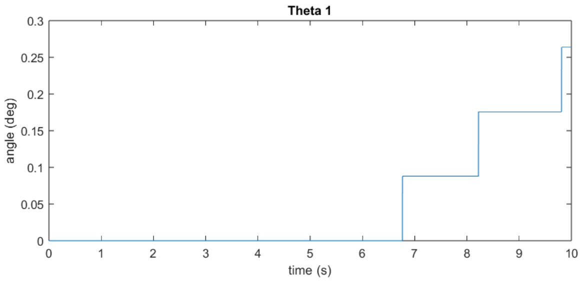
We saw that we have a small “stiction” effect at the start. We can estimate it by feeding a slow ramp of voltage and recording the value at which the “detach” occours:

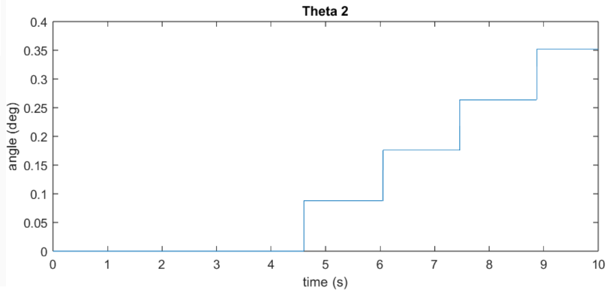
The zero velocity torque (“locked rotor”) is given by:

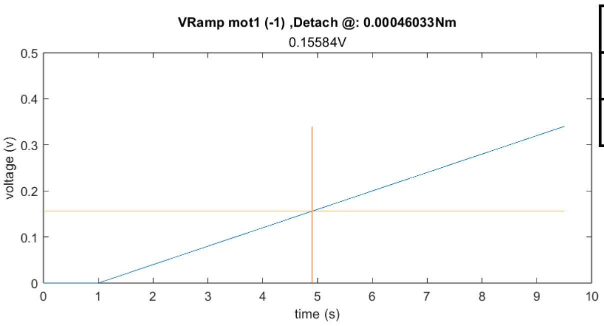
$$
\begin{aligned}
T &= k_e i \\
i &= \frac{V}{R_a} \\
T &= V \frac{k_e}{R_a}
\end{aligned}
\tag{1.}
$$



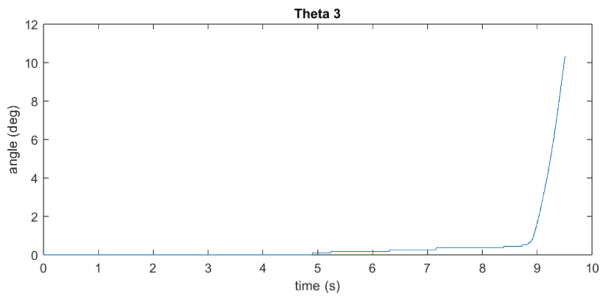



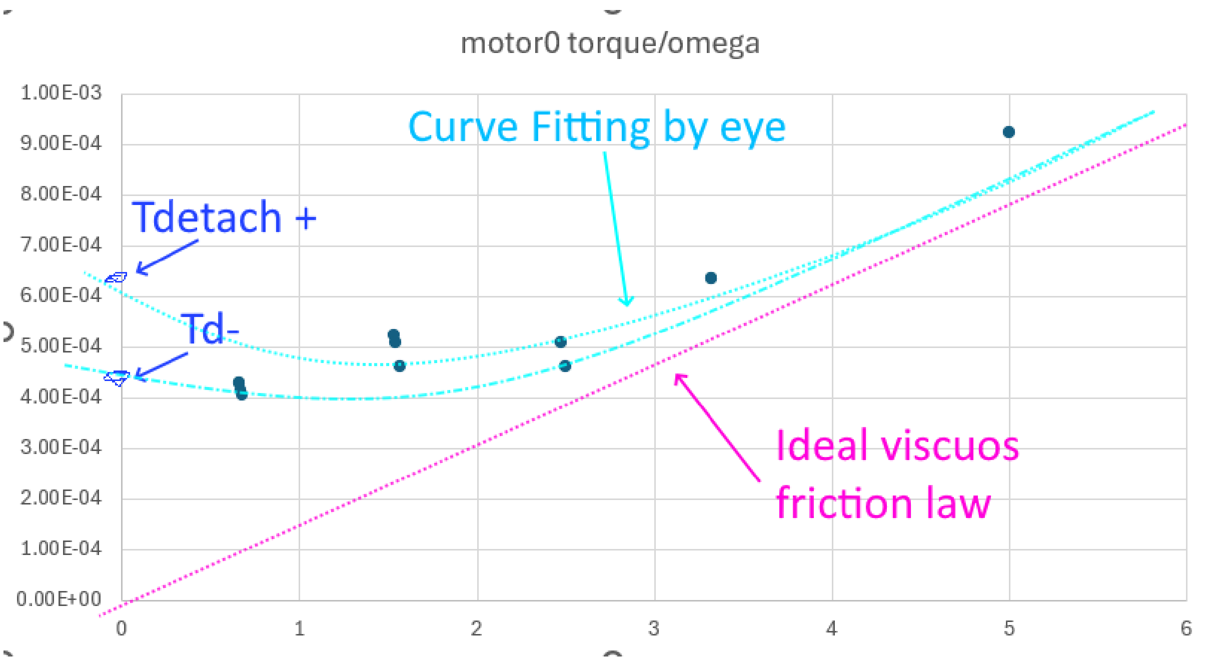



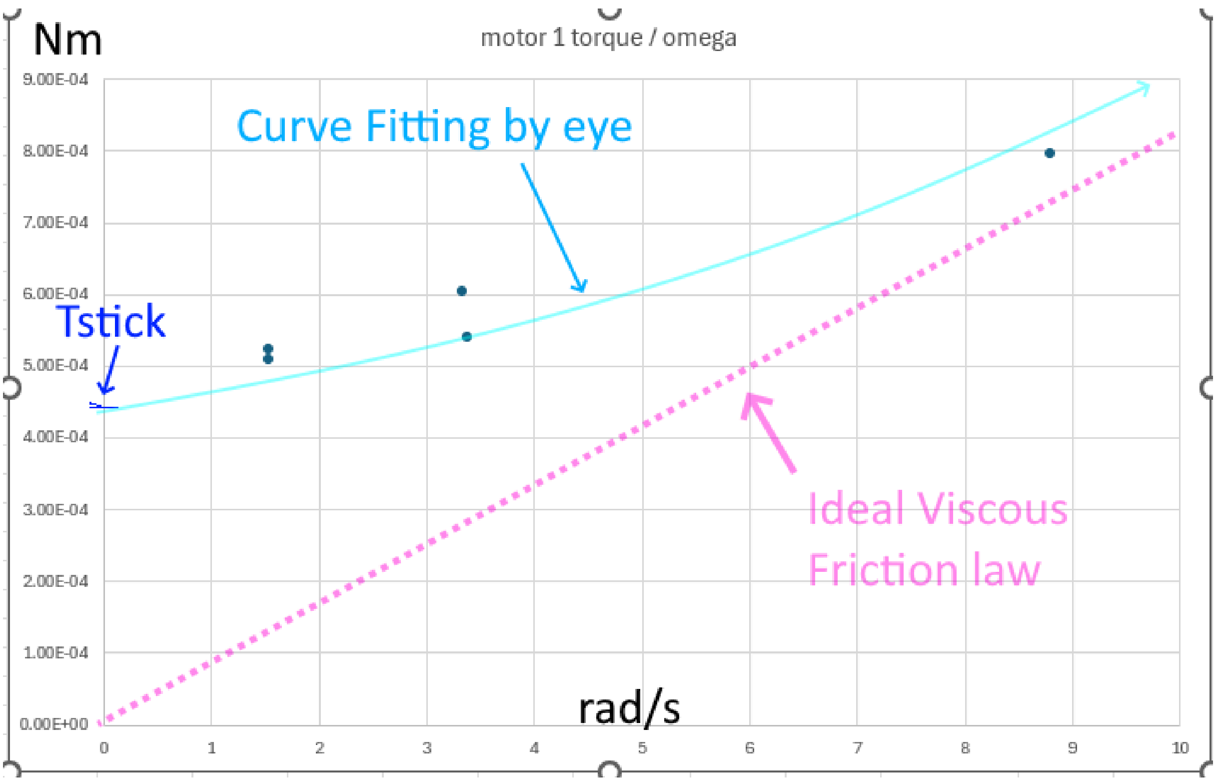




| | Mot 0 | Mot 1 |
|---|---|---|
| V+ | 6.8E-0.4 | N/A |
| V- | 4.2E-0.4 | 4.6E-0.4 |



You can see how the static friction torque is not "stable" even in the same motor.




## 2) Non Linear Friction

Taken from: *Control of Machines with Friction (B Armstrong-Hélouvry)*

Quick overview of what real friction looks like: the "viscous law" $T_f = \beta\omega$ is an approximation that only works at high velocities (when the light-blue line asymptotically approaches the purple line)

8 *Control of Machines with Friction*

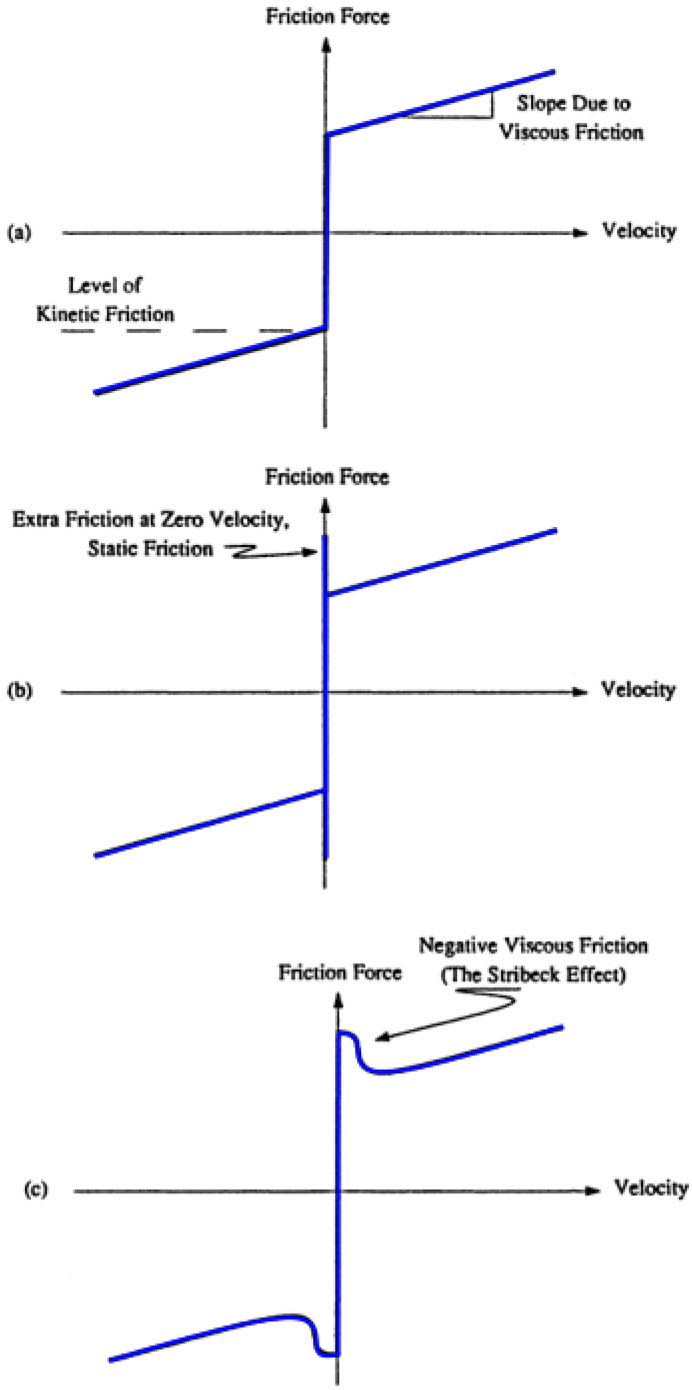
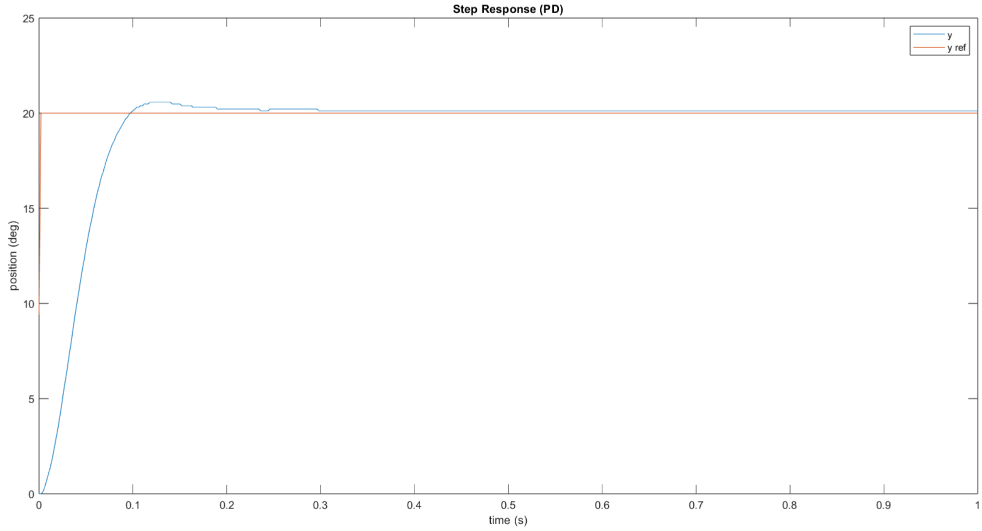
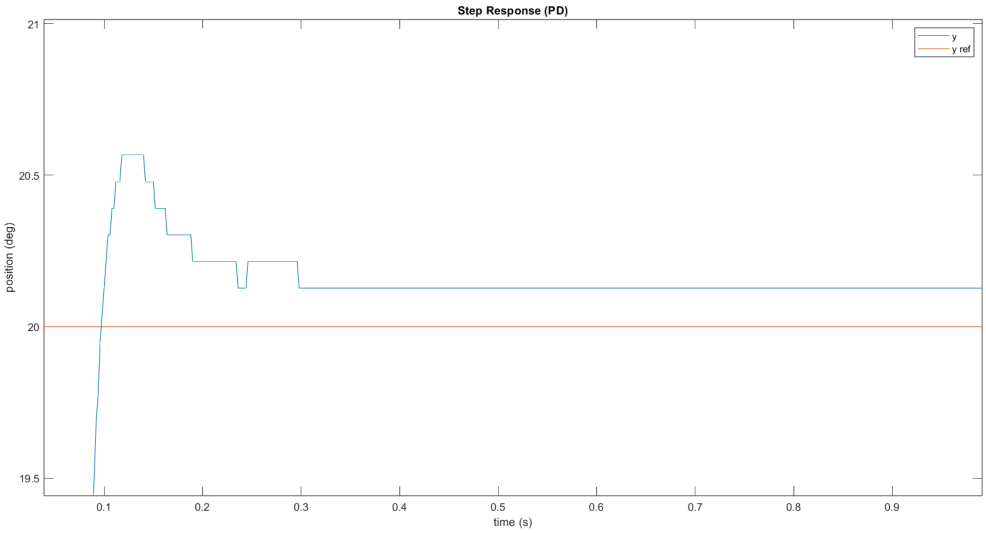



**Figure 2.1** Friction Models. a: Kinetic plus Viscous Friction Model; b: Static plus Kinetic plus Viscous Friction Model; c: Negative Viscous plus Kinetic plus Viscous Friction Model.

At zero/low velocities we have huge non-linearity that creates a loss of precision during position control:




It's possible to see how this non-linearity shows up at low voltage -> low error -> near the set point.

(Error) e = 0.127° = 2.21E-3 rad

With Kp = −21.4 and low velocity ($\dot{e} \approx 0$)

$$u = eK_p = 0.04V \tag{2.}$$

$0.04V \ll 0.15V$ needed for detach. (see graphs at the start)

## 3) Conclusions

- Friction is bad

# 4) Countermeasures

## 4.1) Add an Integrator

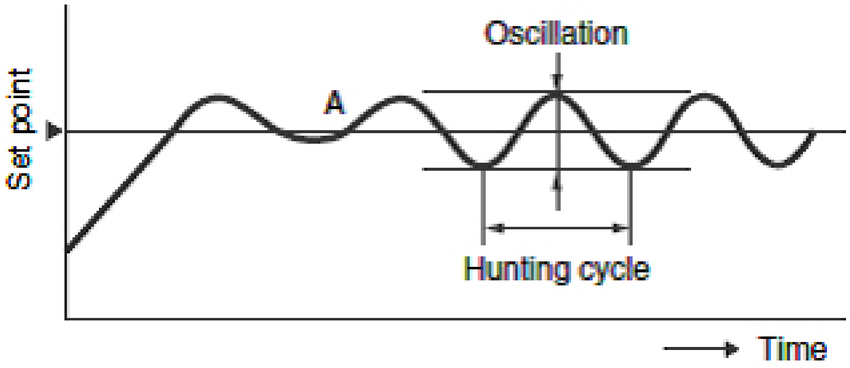
From PD to PID:

Good:

- Integrators help with "low frequency" disturbances and you can model friction as a disturbance.

Bad:

- It goes against our specifications (we have already enough integrators in our loop)
- Hunting (≈ Wind-up in reverse): the integrator accumulates error and is able to overcome the stiction, but now it has to deplete this internal energy so it overshoots. Rinse and repeat




  - Solution (?) PID reset: as soon as you reach the set point, just reset the integrator. Supported in a lot of PID and in Simulink.

**External reset — Trigger for resetting integrator and filter values**
none (default) | `rising` | `falling` | `either` | `level`

Specify the trigger condition that causes the block to reset the integrator and filter to initial conditions. (If **Use filtered derivative** is not selected, the trigger resets the integrator and differentiator to initial conditions.) Selecting any option other than `none` enables the **Reset** port on the block for the external reset signal.

**`none`**
The integrator and filter (or differentiator) outputs are set to initial conditions at the beginning of simulation, and are not reset during simulation.

**`rising`**
Reset the outputs when the reset signal has a rising edge.

**`falling`**
Reset the outputs when the reset signal has a falling edge.

**`either`**
Reset the outputs when the reset signal either rises or falls.

**`level`**
Reset the outputs when the reset signal either:

- Is nonzero at the current time step
- Changes from nonzero at the previous time step to zero at the current time step

This option holds the outputs to the initial conditions while the reset signal is nonzero.

## 4.2) Increase the Kp

From Equation 2 we can see that if Kp is high enough we can get rid of this problem:

Our desired "max error" is given by the precision of the encoder

$$\theta_{\text{err max}} = \frac{2\pi}{4096} = 1.533 \times 10^{-3} \text{ rad} \quad (3.)$$

Let's take a "deadzone voltage" of around 0.3V (highest measurment x 150% as a safety margin)

$$|K_{p,\min}| = \frac{u_{dz}}{\theta_{\text{err max}}} = 195 \quad (4.)$$

Good:

- It should just work…

Bad:

- …as long as you don't care about the controller specifications (Bandwidth, moderation, stability!)
  - It's basically asking for 10 times the previous bandwidth!

Solutions:

- Redesign the PD controller with a fixed Kp and Bandwidth by moving the Derivator's zero/poles.
- If someone is familiar with H∞ this looks like an appropriate situation for Loop Shaping.

## 4.3) Friction Compensation

**If** we have a reliable model of the friction then we can just invert it, do a table lookup with the speed and use it as a feedforward compensator.

Good:

- We place it, it works, we forget about it. All the design in frequency, state-space, whatever can be done with a linear model in mind.

Bad:

- We need a **really good model** of the friction. All errors in modelling will be amplified (chattering or instability)

## 4.4) Dithering

Feed a high frequency signal at low velocity, it helps overcoming the stiction effects.

Good:

- We place it, it works, we forget about it. All the design in frequency, state-space, whatever can be done with a linear model in mind.

Bad:

- Mechanical models don't like high frequency disturbances (wear of components, vibrations etc etc etc). We also have a lot of gears which may "damp" this signal and end up with only a partial solution of the problem.