Bechmarking control algorithms with current saturation aware references

Robot Programming and Control

Filippo Grotto VR460638

December 12, 2021

Contents

1 Introduction

The aim of Forecast project [1] is to provide a testbed to benchmark control algorithms and get in this way comparable results especially in the context of force control algorithms. The procedure uses several performances indicators to assess the controller in use such as static error, dynamic error, overshoot and bandwidth. The latter is estimated using a sinusoidal function that increments frequency over time (e.g. 1 to 10Hz in 10 seconds) until we reach -3db over the desired reference. However this estimation suffers from the presence of saturation of the motor used to drive the experiments making the result about the controller not as general as we want. In the following section we will propose a method to reduce the amplitude of the reference signal before reaching the motor saturation in order to get a much more precise bandwidth estimation using general transfer function methods in case of fully linear systems.

2 Noise Sensitivity Function

If we consider the general feedback system in Fig.1. There are lots of transfer function that we are possibile to define depending on our input-outputs pairs. In our cases we will consider $F(s) = 1$.

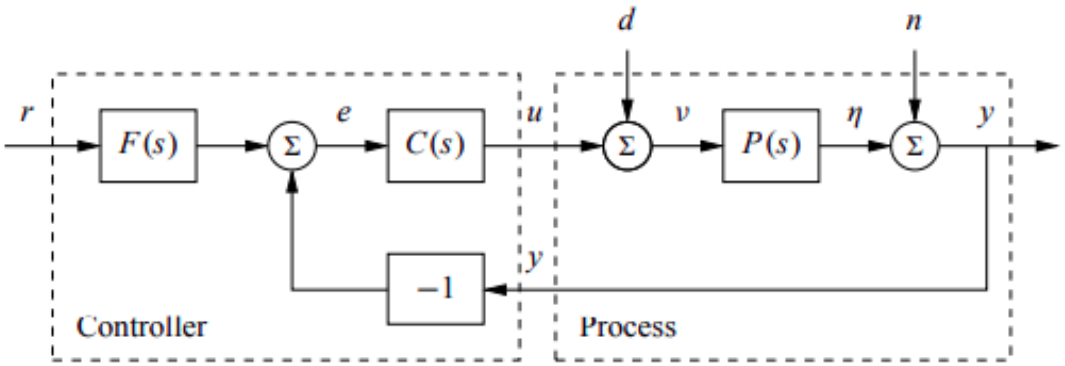

Figure 1: A general SISO feedback system taken from [2]

We are actually interested in having the relation between the input of the system r and the control input u in order to know when the current saturation is reached. We will assume our $P(s)$ plant known since we can identify all the parameters of our DC motors assuming it to be a linear system.

$$CS(s) = \frac{C(s)}{1 + C(s)P(s)} \tag{1}$$

The procedure proposed for the current saturations is derived considering the motor saturation u_{sat} usually provided in the datasheets:

$$\frac{u}{u_{sat}} < 1 \tag{2}$$

The equation (2) holds since we want u to not overcome the provided saturation. However a more strict condition can be imposed. We can then write:

$$\frac{u}{u_{sat}} = CS(s)\frac{r}{u_{sat}} \tag{3}$$

Considering the worst case scenario when $u/u_{sat} = 1$ we can derive:

$$r < \frac{u_{sat}}{CS(s)} \tag{4}$$

Note that equation (4) uses mixed time and laplace domain notations. In the following sections only the magnitude of the transfer function will be considered.

3 Control Architectures

Our aim is to replace the input of the different architectures with a correctly scaled sinusoidal function (sweep) which guarantees that motor saturation is not exceeded during the simulated experiment. To do so the signal is firstly processed by some matlab code which scaled it using the presented equations secondly the simulink model is considered using the generated signal and current/tau is considered.

The following code was used to generate the reference signal:

```
function [sweep, suggested_sweep, t] = reference_signal(start_T,
                                                        end_T,
                                                        start_freq,
                                                        end_freq,
                                                        duration,
                                                        amplitude,
                                                        u_sat,
                                                        CS,
                                                        dt)

  t = linspace(start_T, end_T, duration/dt);
  sweep = amplitude * sin(2*pi*t.*(start_freq + ((end_freq-start_freq)/(duration))*t ));
  freq = (start_freq + ((end_freq-start_freq)/(duration))*t);

  for i = 1:length(freq)
      [mag,~] = bode(u_sat/CS, 2*pi*freq(i));
      limit(i) = mag;
  end

  sweep_expected = limit .* sin(2*pi*t.*freq);

  epsilon = 0.2;
  saturation_reached_at = -1;
  for i = 1:length(sweep)
     if (sweep(i) > 0 && sweep(i) > sweep_expected(i) + epsilon)
         saturation_reached_at = i;
         break;
     end
     if (sweep(i) < 0 && sweep(i) < sweep_expected(i) - epsilon)
         saturation_reached_at = i;
         break;
     end
  end

  suggested_sweep = sweep;
  if (saturation_reached_at > 0)
      suggested_sweep(saturation_reached_at:end) = sweep_expected(saturation_reached_at:end);
  end
end
```

3.1 Parameters Used

The parameters used for the motor comes from direct identification of a maxon DCX22L, in particular $J_m = 0.0064$, $d_m = 0.0068$, $K_t = 1.46$ and $u_{sat} = 2.26$. The controller used for all the architectures is a PID/PD controller with different parameters. To give and example $P = 15$, $D = 0.1$ and $I = 0.1$. In the force control case the stiffness of the environment was considered $h = 100$.

3.2 Position Control

Let's consider a basic position control architecture for a linear motor model considering only the mechanical subsystem as depicted in Fig 2

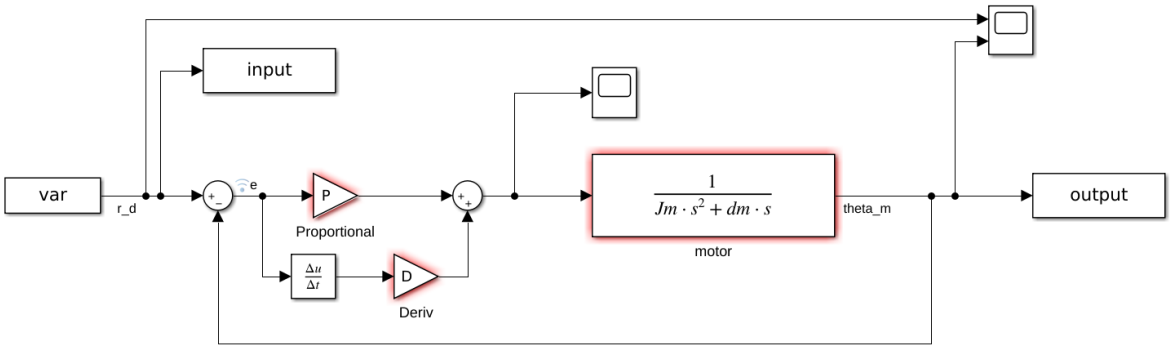

Figure 2: Simulink model of position control architecture

The related equations of the plant P_l and the PD controller C are:

$$P_l = \frac{1}{J_m s^2 + d_m s} \qquad C = P + Ds$$

The two transfer functions $T(S)$ (r to y) and $CS(s)$ (r to u) are presented in Fig 3 and the related original and suggested reference signal are reported in Fig 4. As it is visible the suggested approach is pretty aggressive especially due to the *epsilon* considered in the $reference\_signal$ function presented above, moreover we are considering a worst case scenario.

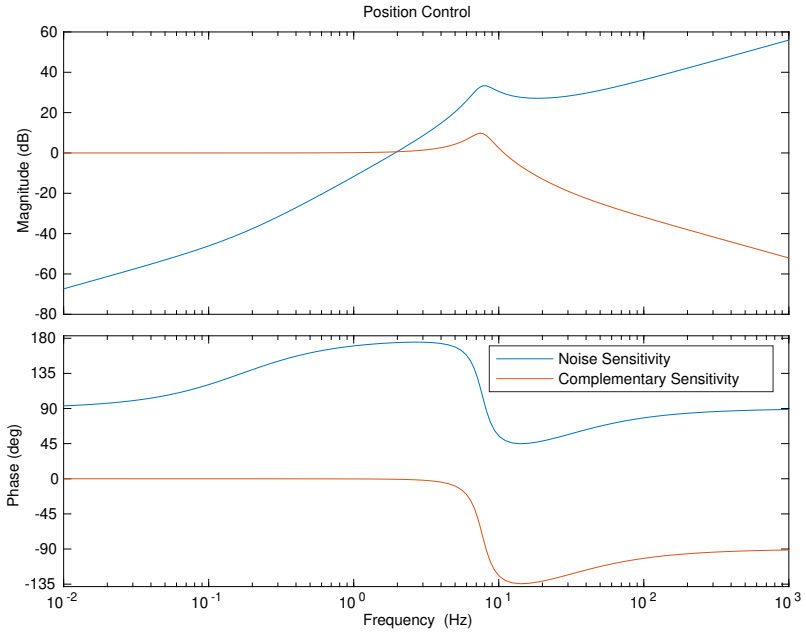

Figure 3: Position control bode plot of the complementary sensitivity function and noise sensitivity function

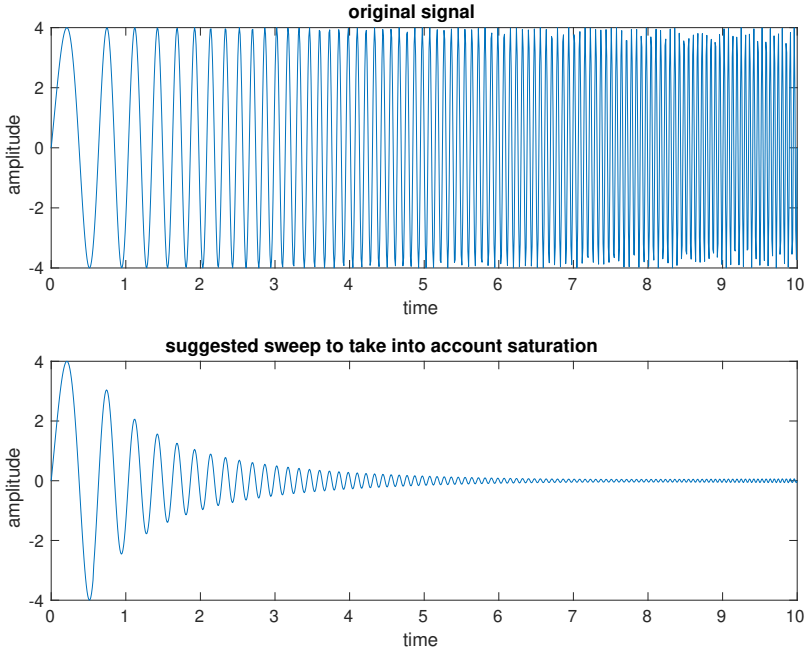

Figure 4: original and scaled sweep with amplitude 4 rad

The obtained *tau* provided to the motor in the position control simulink model is the following:

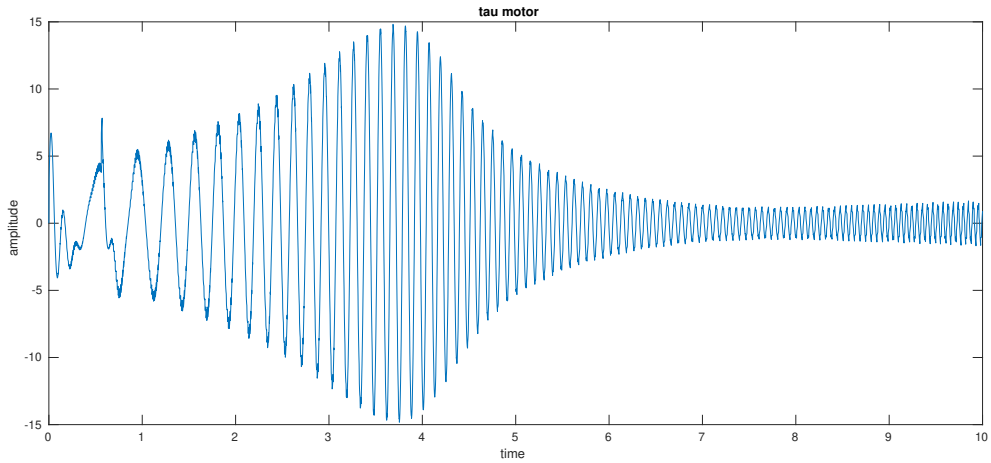

Figure 5: tau signal provided to the motor in the position control

3.3 Force Control

Let's consider a basic force control architecture for a linear motor model considering only the mechanical subsystem. The related equations of the plant P_l and the PD controller C are:

$$G = \frac{1}{\frac{J_m}{h}s^2 + \frac{d_m}{h}s + 1} \qquad C = P + Ds + \frac{I}{s};$$

The two transfer functions $T(S)$ (r to y) and $CS(s)$ (r to u) are presented in Fig 6 and the related original and suggested reference signal are reported in Fig 7. As it is visible the suggested approach is pretty aggressive especially due to the *epsilon* considered in the $reference\_signal$ function presented above, moreover we are considering a worst case scenario.

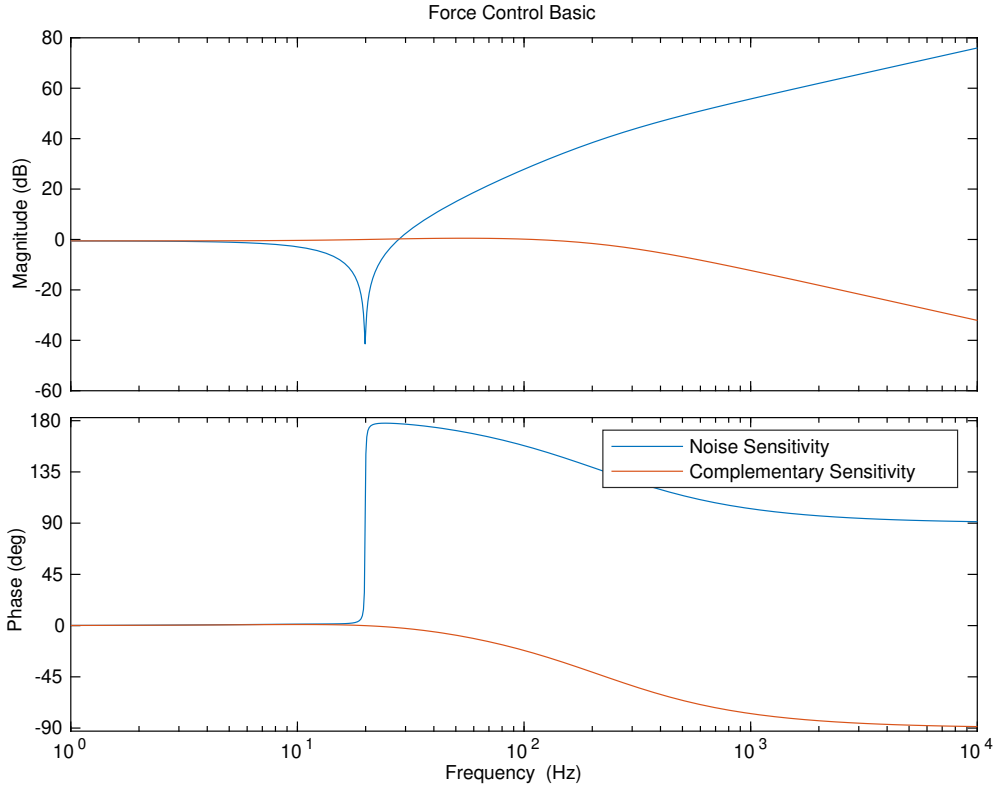

Figure 6: Force control bode plot of the complementary sensitivity function and noise sensitivity function

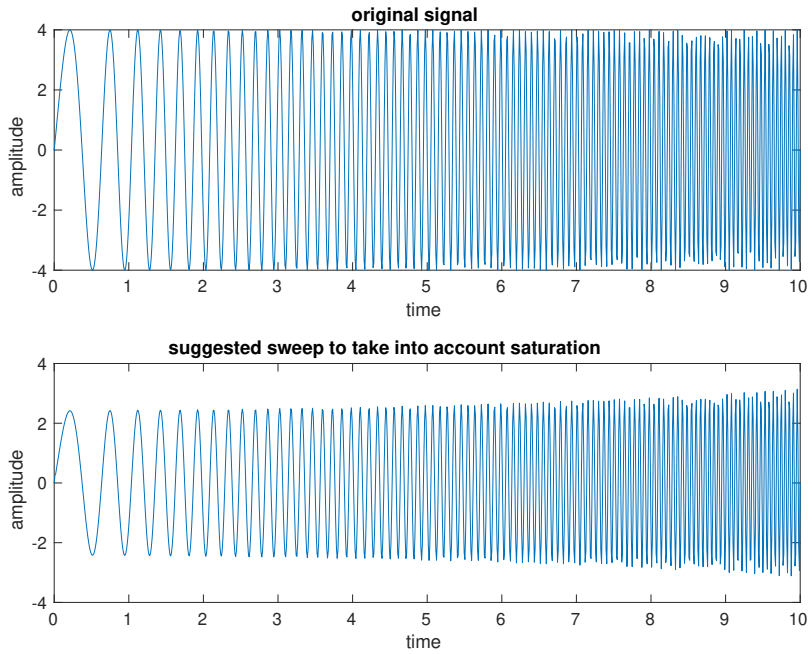

Figure 7: original and scaled sweep for the force reference. As it is visible we have a different behaviour than the position control due to a completly different $CS(s)$ transfer function

The obtained *tau* provided to the motor in the position control simulink model is the following:

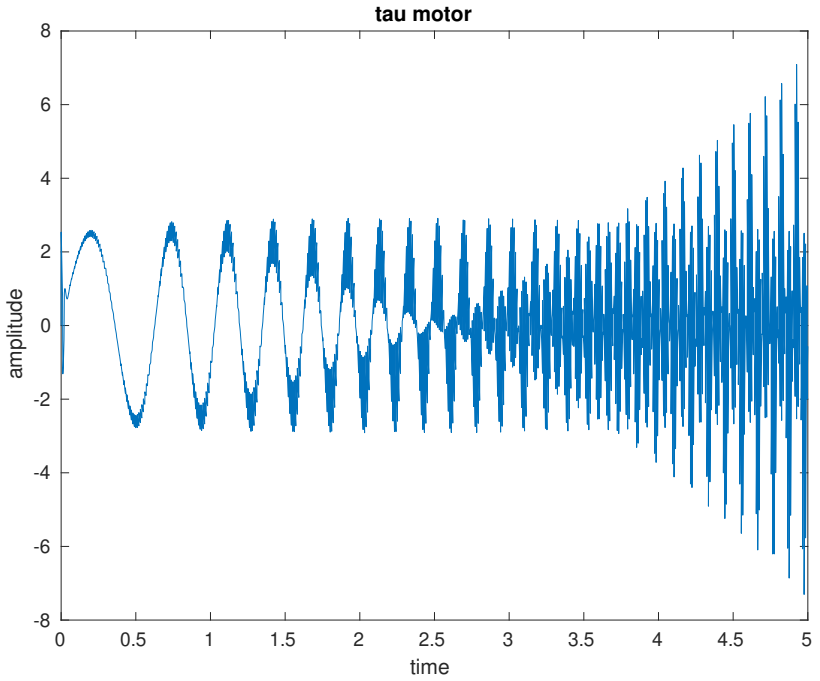

Figure 8: tau signal provided to the motor in the force control

4 Limitations

The current approach based on the use of the transfer function $CS(s)$ has lots of limitations:

- The approach assumes full knowledge of the considered system, everything has to be identified correctly and this is even a bigger limitation for force control because we don't have informations about the environment which makes the use of the transfer functions not reliable at all.
- We assume a linear system which is a big limitation since it doesn't consider non-linear behaviours of the DC motors. For this cases having at least one real experiment to validate the data is a must.
- The use of $CS(s)$ is not reliable in some scenarios due to accentuated shapes of the transfer function in Fig 9 the transfer function of a SEA with links is reported as well as its relevant equations.

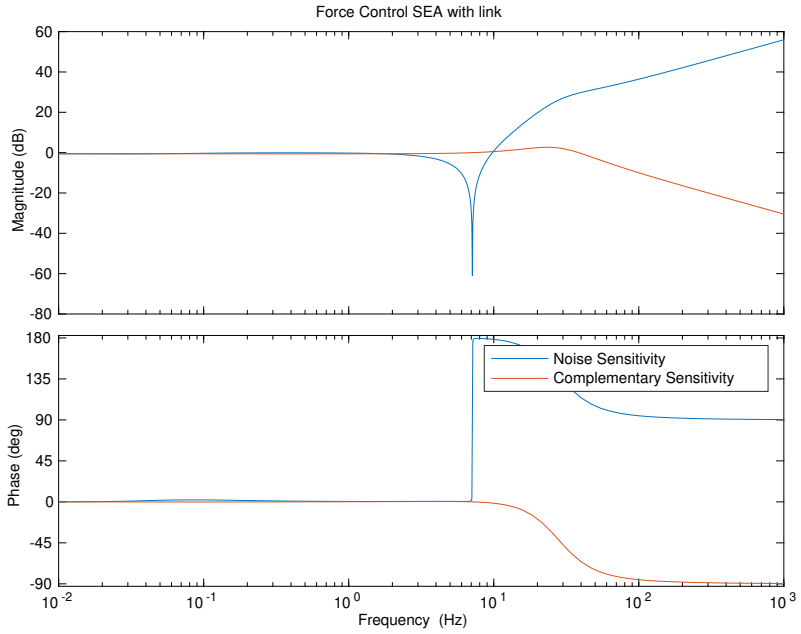

Figure 9: SEA with link force control bode plot of the complementary sensitivity function and noise sensitivity function

$$r = \frac{Ks}{Ke} \qquad E = \frac{J_e}{K_e}s^2 + \frac{B_e}{K_e}s + 1;$$

$$F = \frac{E}{(E+r)\frac{J_m}{K_s}s^2 + E} \qquad C = P + Ds;$$

5 Other approaches

Simulating the experiments in simulink is in general a good way to verify the presence of saturation but it is affected by the same problems related to prior knowledge and linear behaviours (at least without implementing a non-linear system). A good way to generate a current saturation aware reference signal is adapting the reference in real time by reading directly the current provided to the motor. If this exceeds a certain threasold just drop a certain percentange. With this approach we can't know a priori the "amplitude shape" of the signal but it will be recovered after the experiments (using the log captured by the Forecast system). The bandwith estimation can be performed using the Matlab toolbox over that data.

6 Conclusions

The report is an attempt to provide a way to generate current saturation aware signals using noise sensitivity trasfer function. The basic idea was provided as well as some simulation results. The main limitations and other possibilities were discussed and analysed.

References

[1] Forecast Project https://eurobench2020.eu/developing-the-framework/force-control-algorithms-testbench-forecast/

[2] Karl Johan Astrom, Richard M. Murray Feedback Systems An Introduction for Scientists and Engineers