Lab Session 3: Position-Tracking Control

1 - Introduction

Trajectory-tracking control uses predefined position or force trajectories based on unimpaired individuals or information of the non-paretic side to provide assistance or resistance (see Fig. 1). The low-level controller focuses on only one aspect, to move the exoskeleton joint to a desired angle. It will change the motor's torque until the desired position is reached. Position control systems are very useful to support rigidly guided movements, or to limit the range of motion of a user.

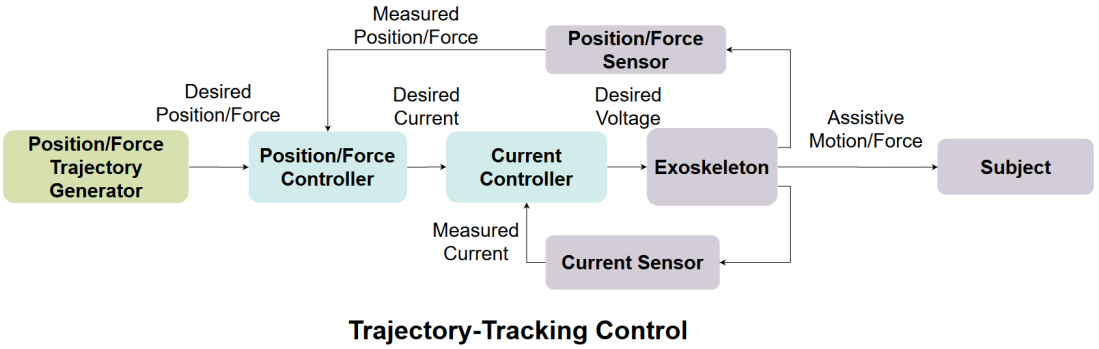

Fig. 1. Control Scheme of the Trajectory-Tracking Control.

This approach alone might be the most effective for people who do not preserve any volitional control or have an acute weakness or paralysis, like people with SCI or acute stroke. Nevertheless, excessive physically guided assistance may have a negative influence on motor learning, as the dynamics of the target task is different from the learned task. **The user learns to rely on the guidance and fails to learn the motor commands.** Trajectory-tracking control strategies that follow an assist-as-needed philosophy look for the maximum participation of the user, providing tailored assistance based on gait performance metrics. Passively guided movements may also improve the performance of the user wearing the exoskeleton, but the neurological benefits are very different. The device should challenge the users wearing the exoskeleton to initiate each action actively and control their movements. For this reason, assistive control strategies need to guarantee **patient empowerment** by inducing a progressive and tailored assistance.

2 - Reference Generation

Exercise 2.1 - Create a sine wave based on the running time of the system with an offset of 45 degrees, an amplitude of 20 degrees, and a frequency of **5 rad/s**. Fill the function *ReferenceGenerator().* Please remember that before applying the reference values, you need to perform the two calibrations needed (sensor calibration and motor calibration; see also lab 2 for examples).

Exercise 2.2 - Copy the data from the serial monitor and save it in a .txt file or in an Excel file. Plot the values of the reference position and actual position against (vs.) the running time. Is the position sensor good? The following Matlab code plots the values of the reference position and actual position against the running time:

```
%% Extract Data
[file_name, file_path] = uigetfile('*.txt');
experimental_data = load (fullfile(file_path, file_name));
time_stamp = experimental_data(:,1);
desired_pos = experimental_data(:,2);
actual_pos = experimental_data(:,3);

t_sim = cumsum(time_stamp);
```

```
%% Plot Data
ax1 = subplot(2,1,1); plot(t_sim, desired_pos, t_sim, actual_pos);
xlabel('Time [s]'); ylabel('Actuator Position [º]');
legend('Reference', 'Actual');

ax2 = subplot(2,1,2); plot(t_sim, desired_pos-actual_pos);
xlabel('Time [s]'); ylabel('Position Error[º]');


linkaxes([ax1,ax2],'x');
```

3 - System Identification in the Frequency Domain

The exoskeleton can be approximated as a **stable linear system** $G(s) = \frac{y(s)}{u(s)}$, where $y(s)$is the output position and $u(s)$ is the reference position. In case the system is excited with an input $u(t) = U \cdot sin(wt)$, it will generate an output in steady-state $y_{ss}(t) = lim_{t\to\infty} y(t) = Y \cdot sin(wt + \Phi)$. The response of the system will oscillate at the same frequency w attenuated by a factor $|G(jw)| = \frac{Y}{U}$ and shifted an angle $\Phi = t_d w = \lfloor G(jw)$, where t_d is the time delay measured in time units (see Fig. 2).

The methodology of this experiment is to excite the exoskeleton with sinusoidal position references at different frequencies and analyze the output response. From this experimental data, it can be extracted its frequency response and then a transfer function which approximates the behavior of this system. This identification will provide the bandwidth of the system, which is an important parameter of the system. **The bandwidth frequency is defined as the frequency at which the closed-loop magnitude drops 3 dB below its magnitude at DC (magnitude as the frequency approaches 0)**. In case the frequency of the reference is set to a value higher than the bandwidth frequency of the system, the output response of the system will be distorted with respect to the input.

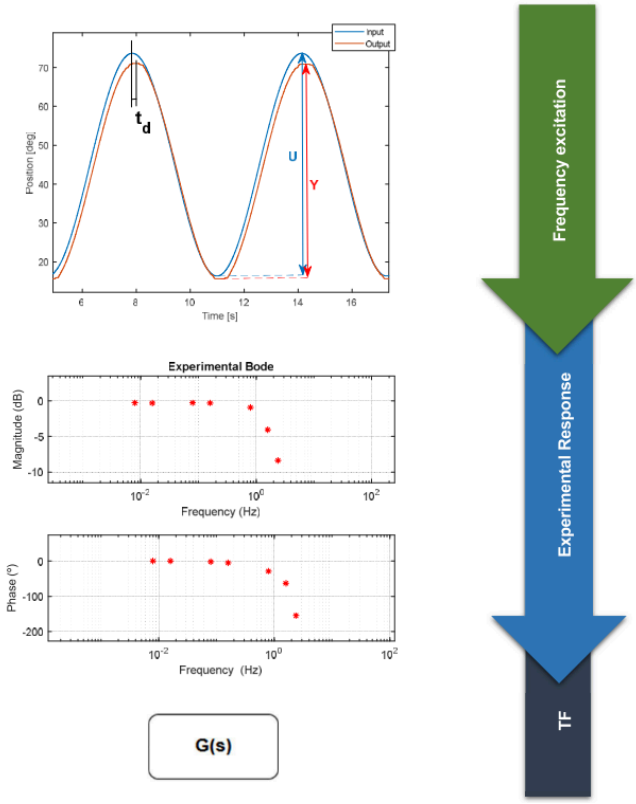

Fig. 2. Description of the methodology to identify the dynamics in the frequency domain. Firstly, the system is excited at different frequencies to measure the difference with the output in amplitude and frequency. Secondly, the experimental frequency response of the system is obtained. Finally, from this information, it is possible to extract a transfer function (TF) which collects the dynamics of the system.

Exercise 3.1 - Excite the systems at the following range of frequencies [0.5, 9] rad/s and amplitudes [2, 10] degrees and record the time stamp, actual position and reference position. Extract the difference in amplitude (*gain_exp*) and time between the input and the output signal (*time_delay*) per each frequency. The experimental Bode can be plotted using the following Matlab code:

```
% 1 radian per second approximately 0.159155 Hz
rads2hz = 0.159155; % in Hz.

% Experimental data
w_exp      = []; % [rad/s]
gain_exp = [];
time_delay   = []; %[s]
fase_exp = (-180/pi)*time_delay.*w_exp; % [º]

figure;
ax1 = subplot(2,1,1);
semilogx(rads2hz*w_exp,20*log10(gain_exp),'r*');
xlabel('Frequency (Hz)'); ylabel('Magnitude (dB)');
ax2 = subplot(2,1,2);
semilogx(rads2hz*w_exp,fase_exp, 'r*');
xlabel('Frequency  (Hz)');
ylabel('Phase (º)');
linkaxes([ax1,ax2],'x');
```

Exercise 3.2 - Consider a first order system and find the values of gain and constant time that fits the values of the experimental response. Plot the experimental bode and the estimated bode on the same plot. This Matlab code to plot the experimental bode together with the estimated bode:

```
% 1 radian per second approximately 0.159155 Hz
rads2hz = 0.159155;

% Theoretical transfer function
K = 1;
tau = 0.1;
num = K; den = [tau, 1];

% Range of frequencies to plot the theoretical bode [rad/s]
w_min= 1; w_max= 100;

% Vector of frequencies between w_min y w_max with 100 point logarithmic equispaced
w_theoretical=logspace(log10(w_min), log10(w_max), 100);

% Theoretical bode
[gain_theo, phase_theo]=bode(num,den,w_theoretical);

% Experimental data
w_exp      = []; % [rad/s]
gain_exp = [];
time_delay   = []; %[s]
fase_exp = (-180/pi)*time_delay.*w_exp; % [º]

figure;
ax1 = subplot(2,1,1);
semilogx(rads2hz*w_exp,20*log10(gain_exp),'r*', rads2hz*w_theoretical, 20*log10(gain_theo));
xlabel('Frequency (Hz)');ylabel('Magnitude (dB)');
ax2 = subplot(2,1,2);
semilogx(rads2hz*w_exp,fase_exp, 'r*', rads2hz*w_theoretical, phase_theo);
xlabel('Frequency  (Hz)');
ylabel('Phase (º)');
linkaxes([ax1,ax2],'x');
```

Exercise 3.3 - Which is the bandwidth of your estimated system in Hz? Is it a good bandwidth for an upper-limb exoskeleton?