

Exercise 2: Planar two-link flexible manipulator

Consider the two-link flexible manipulator shown in Figure 1. This manipulator has been introduced as a benchmark by Schiehlen and Leister [1] and has been quoted in several papers. Some properties are given in Table 1. Gravitation is included as illustrated by the vector \vec{g} in Figure 1(a). Although in the original benchmark a prescribed joint motion is defined, we will consider a controlled trajectory motion in this exercise. The joints are driven with torques generated by a PD-like controller, analogous to the controller used in section 3.11 of the SPACAR manual.

Table 1: Properties of the two-link manipulator (adapted from [1, 2]).

Property		Node 1	Node 2
Joint mass	m	1.0 kg	3.0 kg
Property		Link 1	Link 2
Length	l	0.545 m	0.675 m
Density	ρ	2700 kg/m ³	
Young's modulus	E	$7.3 \cdot 10^{10}$ N/m ²	
Height (in xy-plane)	h	0.015 m	0.010 m
Width (in z-direction)	b	0.060 m	0.040 m
Cross-sectional area	$A = bh$	$9.0 \cdot 10^{-4}$ m ²	$4.0 \cdot 10^{-4}$ m ²
Cross-sectional area moment of inertia (2D)	$I = \frac{1}{12}bh^3$	$1.69 \cdot 10^{-8}$ m ⁴	$3.33 \cdot 10^{-9}$ m ⁴

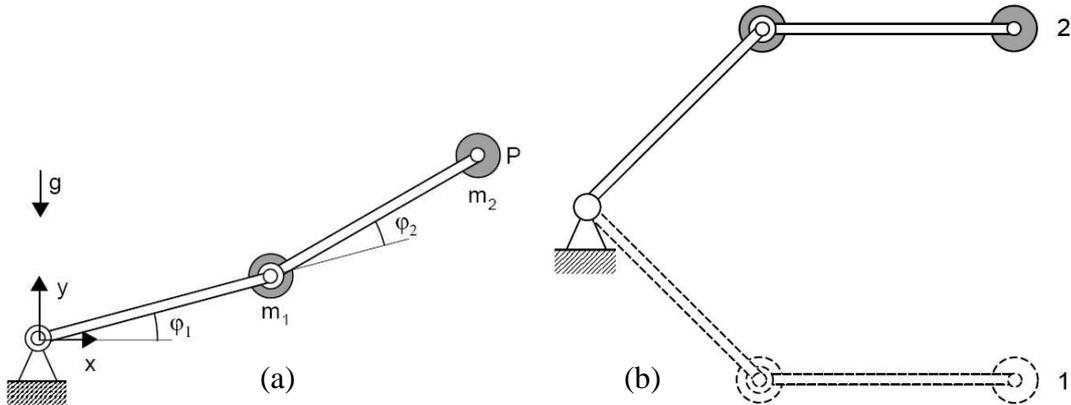


Figure 1: Planar two-link manipulator: (a) Definitions of coordinate system and joint angles. (b) Initial configuration (1) with $\phi_1 = -\phi_2 = -\pi/4$ and final configuration (2) with $\phi_1 = -\phi_2 = \pi/4$.

The links should move from the initial configuration to the final configuration, both shown in Figure 1(b). During the motion the second link should remain horizontal and its vertical position should move in 0.5 s. It should accelerate with constant acceleration in 0.1 s, move at a constant (vertical) speed and finally decelerate with constant deceleration in 0.1 s. In the initial condition for the simulation at $t = 0$ the mechanism is at rest and the deformations of the links are determined by the gravity forces. For simulations planar beam elements can be used to define each link. The longitudinal deformations of the elements are ignored, so of each element only the two bending deformation parameters are taken into account.

On page 3 some hints on the use of SPACAR and MATLAB/SIMULINK are provided. Please read these hints a priori.

1. Analyse this two-link manipulator with a 2D SPACAR model while it moves along the prescribed *nominal* trajectory. Compute e.g. the nominal driving torques in the joints and the natural frequencies of the mechanical system. Check whether these frequencies change when more or less beam elements are used to model each link. Analyse the manipulator in (at least) three different configurations along the trajectory.
2. Determine the initial (steady) state of the *flexible* manipulator from a closed-loop simulation with a 2D SPACAR model. Consider a closed-loop bandwidth of approximately 10 rad/s.
Check that this initial state can also be obtained considering the system matrices from a *linearised* model of the manipulator.
3. Simulate the desired motion of the closed-loop manipulator with a non-linear simulation. Consider again a closed-loop bandwidth of approximately 10 rad/s. Is the performance improved if the bandwidth is increased?
Hints: Use only a small number of beams in this simulation as otherwise the simulation is very slow.
4. Redo the simulation with a linearised model and check the results by comparing them with the non-linear simulations.

References

- [1] Schiehlen, W. and Leister, G., 'Benchmark-Beispiele des DFG-Schwerpunktprogrammes Dynamik von Mehrkörpersystemen', Universität Stuttgart, Institut B für Mechanik, Zwischenbericht ZB-64, 1991.
- [2] Botz, M., 'Zur Dynamik von Mehrkörpersystemen mit elastischen Balken', Technische Hochschule Darmstadt, Dissertation, 1992.

MATLAB/SIMULINK help for this exercise

For the simulations the system has to be analysed with SPACAR. The example outlined in Sect. 3.11 of the SPACAR manual is a good starting point although the hints below will use a forward dynamic analysis instead of an *inverse dynamic* analysis. Furthermore, the example in the manual is about a *rigid* manipulator, while the exercise is about a *flexible* manipulator. The title of Sect. 3.12 refers to a flexible manipulator as well, but as this section is not available yet, suggestions on how to proceed are explained here:

- Again you can choose between the SPACAR GUI to create models or editing the `dat` files with your favourite text editor. The SPACAR manual uses the latter approach, but the GUI is well suited for this purpose.

Note that in general there are three steps:

1. A first forward or inverse dynamic SPACAR simulation (`mode=1` or `mode=2`) with a *rigidified* model that moves along the prescribed *nominal* trajectory.
2. Next a SPACAR `mode=3` analysis with the *rigidified* model computes e.g. the necessary driving torques and linearised system matrices during the motion along a trajectory.
3. These data are needed in a subsequent SIMULINK simulation with a closed-loop model in which the *flexible* manipulator is included. A non-linear manipulator model is defined using SPASIM which mimics a forward dynamic simulation. Alternatively, a *linear time-varying* (LTV) model can be used.

In the first and second SPACAR simulations the manipulator is considered to be *rigidified* which means that elastic deformations (like bending of beams) remain zero even though they are allowed to deform in the final analysis. Furthermore, in a `mode=3` analysis the trajectory is not defined, but it is taken from an earlier SPACAR simulation. In Sect. 3.11 a so-called *inverse dynamic* run (`mode=2`) is used for this purpose. In this exercise, the trajectory is relatively easy and a normal forward dynamic (`mode=1`) analysis can be used as will be outlined next.

- In a `mode=1` analysis the manipulator will track a path that is defined for its Input DOF's. As a rigidified manipulator is considered, only the 2-DOF joint motion needs to be defined. The standard quadratic function of time as defined in SPACAR is not suited, so a user defined `USERINP` needs to be used. A sample m-script `tlinp.m` is available for download and can be modified for a particular application. In addition to the two DOF's for the joint motion, additional DOF's *may* be defined for the elastic deformations. For the analysis of the rigidified manipulator these deformations should not play a role, so either they are ignored (i.e. set to `Zero`) or defined as an Input DOF for which the input is prescribed zero.

You have to make your own input file for this SPACAR analysis. In the rest of this help this file will be called `t1.dat` which may be edited manually or is generated by the GUI from a model `t1.spa`.

In the SPACAR GUI the element mass and stiffness properties can be calculated easily once the correct material properties and dimensions are specified. Do not confuse the height (in *xy*-plane) and width (in *z*-direction). After you defined a model with a single beam per link, you can easily divide a beam into `Beam Segments` by selecting this option from the menu that appears after a right-click on a beam element in the navigation pane. If you edit the `dat` file manually, you have to compute properties like mass per unit length and *EI* yourself. And you have to segment beams manually as well. In either case, do not forget to enable gravity.

To check the outcome of the simulation it is advisable to start and stop the motion while the system is at rest for about 0.1 s, see the example in the manual. In the supplied m-script `tlinp.m` this initial stand-still is present as well. Depending on the details of your system definition, you have to edit this file, so verify the motion of the manipulator e.g. from the visualisation.

- Next the analysis is continued with a `mode=3` run. This is also a forward dynamic analysis for which a second input file is needed. As mentioned above and explained in the manual, during this `mode=3` run no new trajectory is defined, but the outcome of an earlier analysis is taken. The name of the `mode=3` must be the name of the file from the previous simulation with one or more extra characters at the end, e.g. `tllin.dat`. As it is also a forward dynamic analysis, the degrees of freedom should again include the driving joint angles. Now also the elastic deformations must be defined that are taken into account. The system is still considered to be rigidified, so the elastic deformations are `Input DOF`'s which are zero at all time steps.

To prepare the SIMULINK simulation, we must also define at this stage the system's inputs and outputs. It is assumed that the system has two inputs that are the driving torques in the joints, so defined with the `NOMS` keyword. There are six outputs that are the joint angles, the joint angular velocities and the tip coordinates. These are defined with `REFE`, `REFEP` and `REFX` keywords. You may of course change this if you like, but then you have to modify e.g. the SIMULINK models (see below) accordingly.

- To answer the questions of this exercise, you need e.g. the mass and stiffness matrices. After the linearisation like `mode=3` these matrices are available in two-dimensional MATLAB arrays that include all time steps. It may be easier to access them in an alternative way. You will notice that an "LTV" file has been created. It includes also the mass matrix `m0` and a stiffness matrix `k0b` in which structural, dynamic and geometric stiffnesses are already added. Try:

```
>> m0=getfrsbf('tllin.ltv','m0',1);
>> k0b=getfrsbf('tllin.ltv','k0b',1);
```

Note that these commands create matrices (so no need to `reshape`) for the specified time step (1 in this case). Another advantage of these matrices is that they are ordered: The upper left part is the ee -part, the lower right the $\varepsilon\varepsilon$ -part, etcetera.

- For the non-linear simulation with `spasim` you need to write the input file e.g. `tlsim.dat`. It can be created with the GUI, but alternatively replacing a few keywords in the previously generated `tllin.dat` is quite straightforward. All degrees of freedom are now dynamic (`DYNE` or `DYNX`) of which initial conditions can be specified (`STARDE` or `STARTDX`). Don't forget the inputs and outputs that are now defined with keywords like `INPUTS`, `OUTE`, `OUTEP` and `OUTX`. The SIMULINK model `tlsim_cl.mdl` is available for download as an example. You may change this if you like. A few remarks:

A number of blocks read data from the "LTV" file for which the name `tllin.ltv` has been specified (without the extensions). All blocks reading from this file need to use the same file!

The block "`times M0`" is put in front of the manipulator in order to approximate it as a set of separate $1/s^2$ systems. It shows how many "modes" are used, which means that only the upper left part of the complete mass matrix with this size is used.

The bandwidth (`Omega`) and damping (`beta`) of the PD-like controller can be changed.

The provided SIMULINK model stores data in a number of variables for later analysis in MATLAB.

- For the linearised simulation the SIMULINK model `tlsim_cl_pm.mdl` can be downloaded as an example. The `spasim` block is replaced by an LTV block that uses the linearised equations of motion. Note that now only the differences compared to the nominal motion are computed. Furthermore, the $\vec{\sigma}_0^{\varepsilon m}$ are also part of the input of the LTV block.