Matlab® Script for Hexapod Analysis

André Boné[1,*]

[1] CENTRA-Sim, Faculdade de Ciências da Universidade de Lisboa
[*] andre.bone@sim.ul.pt

Compiled February 14, 2021

This is the documentation for the Matlab® Script used for the analysis of the Hexapod. The analytical models are shown as well as a description of the code and how to use and adapt it.

1. MOTIVATION

The hexapod analysis code was designed to find the geometries that would minimize the load on the hexapod arms under an different loads.

The code tests a mesh of geometries for the hexapod and applies loads on it. These loads can be, for example, earthquake static accelerations. The load on each arm is calculated, and the maximum load value is saved. The code chooses afterwards the best geometry and calculates on it several other properties, such as the effects of CM position change, required precision for the arms, and length and angle variation for several positions.

The code is commented and can be easily changed for other purposes. More detailed instructions are on section 4.

2. MODEL OF THE HEXAPOD

A mathematical description of the hexapod is necessary to introduce it on a numerical script.

The hexapod consists of a basis and a platform, and six variable length arms between them, as shown in Figure 1.

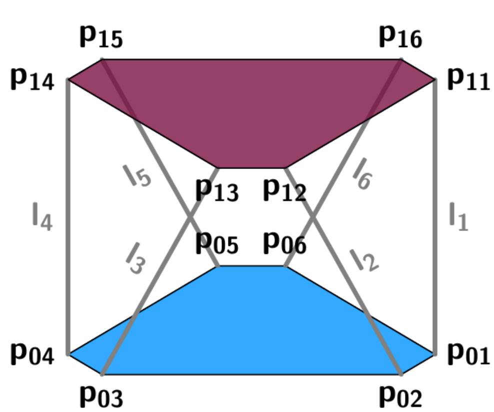

Fig. 1. Schematics of an Hexapod.

The attachment point of the basis $\boldsymbol{p}_{0i}$ connects to the attachment point at the platform $\boldsymbol{p}_{1i}$, through the arm l_i with $i = 1, ..., 6$. The arms must be able to change length freely. When this happens the top platform will move into a new position and alignment, and the arms will have a different angle with the platform, which must be accommodated by the joints.

We further define the values r_0 and r_1, the radii of the base and platform respectively. The angles α_0 and α_1 represent the spacing between consecutive attachment points in the base and platform, and h the height of the platform relative to the base. These are better shown in Figure 2 and Table 1.

Table 1. Definition of the hexapod fixation points $\boldsymbol{p}_{ji} = [r_j \sin(\beta_{ji} \pm \alpha_j), r_j \cos(\beta_{ji} \pm \alpha_j), c_j]^T$.

j	i	c_j	β_{ji}	sign(α_j)
0	1	$-h$	30°	-
	2			+
	3		150°	-
	4			+
	5		270°	-
	6			+
1	1	0	330°	+
	2		90°	-
	3			+
	4		210°	-
	5			+
	6		330°	-

The reference frame has its origin at the center of the base, with x and y being the horizontal coordinates, and z the vertical one.

3. NUMERICAL MODELS

In order to describe the Inverse Kinematics problem for the Hexapod we must define a translation vector $\mathbf{t}$ and a rotation matrix $\mathbf{R} = \mathbf{R_z}(\psi) \cdot \mathbf{R_y}(\phi) \cdot \mathbf{R_x}(\theta)$, where θ, ϕ, ψ are the $Z_\psi \, Y_\phi \, X_\theta$ Tait-Bryan angles. We then have:

$$\boldsymbol{p}'_{1i} = \boldsymbol{t}(x, y, z) + \mathbf{R}(\theta, \phi, \psi) \cdot \boldsymbol{p}_{1i}, \tag{1}$$

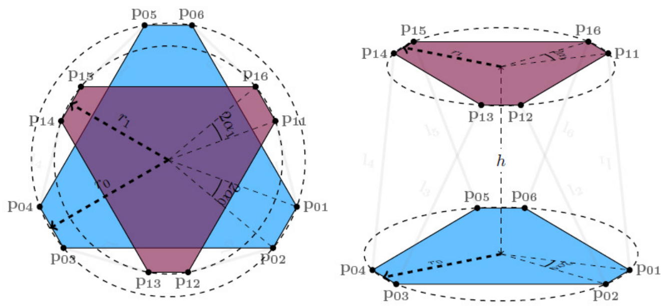

Fig. 2. Scheme showing the hexapod values r_0, r_1, α_0, α_1 and h and what they represent.

where $\mathbf{p_{1i}}$ are the fixation points in the platform coordinate system.

4. INSTRUCTIONS AND SETTINGS

Most of the settings are defined at the beginning of the code. The user can change them however he wants, with single values, lists of values, or any other way of creating arrays. In this case, the code will loop over all options corresponding to the tensor product of the arrays. Units are in degrees for angles, mm for lengths and normalized to g in case of forces and loads.

A. Inputs

You can change the radii of the base and platform, r_0 and r_1, through the variables *r0_val* and *r1_val*.

The angular distance of the attachment points α_0 and α_1 is given by the variables *sp0_val* and *sp1_val* respectively.

h_val is the variable controlling the height h.

Next comes the Center of Mass (CM) Position. *rcm*, the angle *spcm* and *zcm_val* are the values of the position of the CM in Cyllindrical coordinates. The origin of the coordinate system is at the center of the top platform, as the CM related to the instrument that is attached to it. The coordinate system is shown in Figure 3. However, unlike before, *zcm_val* is the only one that can be an array. The idea behind this is that you can study the relative position of the top platform attachment points to the CM.

The variables *ezcm* and *ercm* are the uncertainties for the vertical position and radius of the CM.

x_val, *y_val*, *z_val*, *th_val*, *ph_val* and *ps_val* are the deviations for the position (x, y, z) and angles (θ, ϕ, ψ) for the platform.

dx, *dy*, *dz*, *dth*, *dph* and *dps* are the precisions that you need to achieve in $x, y, z, \theta, \phi, \psi$.

F_val is a list of 3-vectors representing the force that is applied to the CM (normalized to g). These are the force vectors that are applied to the load in the study.

B. Outputs

The outputs of this script are:

1. A figure with a scale model of the hexapod with the best geometry;
2. The values for the best geometry, including necessary resolution, and expected maximum load on an arm;

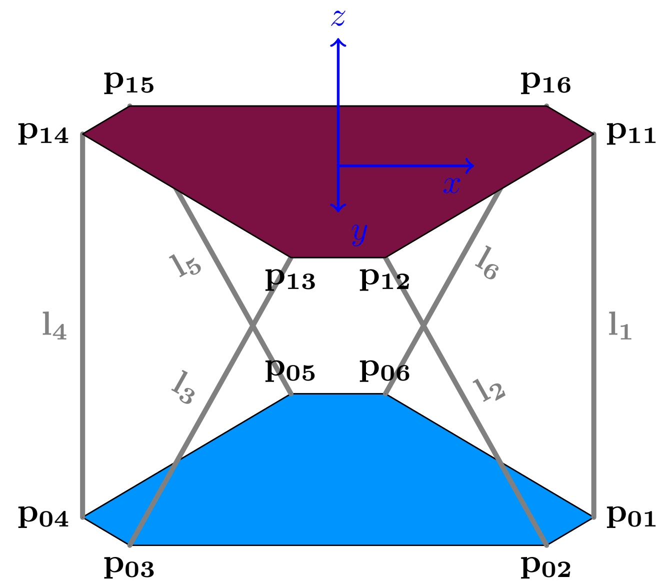

Fig. 3. Example of coordinate system.

3. A figure showing the effect on the load on an arm in function of the CM position for earthquake conditions;
4. The minimum and maximum loads on an arm under achievable under earthquake conditions with the CM uncertainties;
5. A figure showing the effect on the load on an arm in function of the CM position for normal gravity conditions;
6. The minimum and maximum loads on an arm under achievable under normal gravity with the CM uncertainties;
7. The minimum and maximum arm length, and the difference between them.

5. CYCLE

The cycle begins by defining the coordinates of the fixation points. Then it applies the necessary transformations. If you want to see the current geometry, you can uncomment *figure(1)*, right before Force Analysis.

The Force Analysis computes the load on each arm and saves the maximum values. Then repeats the process for all values of the center of mass defined by the variable *zcm_val*. It will then compare it and choose the position that minimizes the force.

The precision analysis is made AT THE CENTRAL POSITION. It changes slightly the position of the platform according to your necessary precisions, in order to compute the arm length variations that you need. It filters out very small values ($< 10^{-10}$ mm) and chooses smaller from the remaining. This indicates how much precision your arms require.

When writing to the variable *config*, the last element is a cost function. The script will choose the configurations that minimize the cost function. As of now it's simply set to the arm tension, but can be written as best suits the user.

6. RECALC

This section chooses the best configuration, and from there analyses the arm load distribution for several positions of the center of mass, within the ranges given by the uncertainties for the radius and vertical position. This will give all possible values of the load under normal and earthquake conditions.

The final part is still under construction, where the objective is to input the limit positions for the platform, and compute the necessary length variation.