EasyDyn Problem: valve–lifter mechanism

Olivier VERLINDEN, Georges KOUROUSSIS

1 Description of the system

With one century of existence, the four–stroke internal combustion engine is one of the most amazing machines we use on a daily basis. Almost all cars today use a reciprocating internal combustion engine because this engine is:

- relatively efficient (compared to an external combustion engine),
- relatively inexpensive (compared to a gas turbine),
- relatively easy to refuel (compared to an electric car).

These advantages beat any other existing technology for moving a car around.

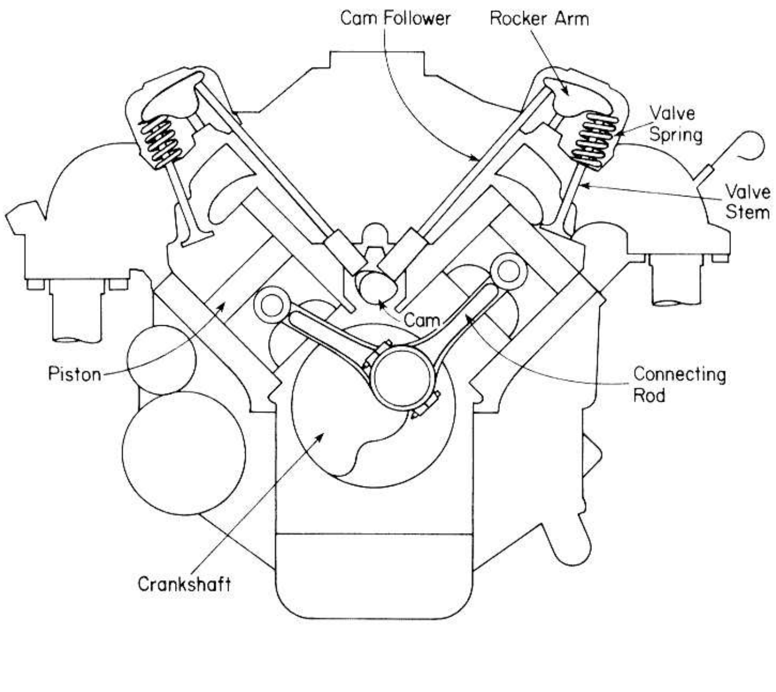

Figure 1: V–8 engine section

Let us consider the valve–lifter mechanism represented in figure 2. For this 2–D model, a outlying disk represents the cam[1]. This body goes round point O with respect to crankcase, physically outlying at its center A. The cam activates a cam follower in contact point B without friction force. This follower, in permanent translation, is linked up the rocker arm by a

[1]Simplified in this case!

revolute–translational joint in point F. The motion is therefore transmitted to the valve stem by a same joint. The valve stem can open or close. The tocker arm is atached to crankcase by a revolution joint around his center of gravity C. Points D and E are respectively Cam follower and valve stem centers of gravity.

To preserve the valve stem on closed position on compression stroke, a spring of stiffness equal to $10\,N/cm$ is used. It comes at a rest length when the rocked arm is on horizontal position. All the geometrical and inertial data are indicated on figure 2 and table 1.

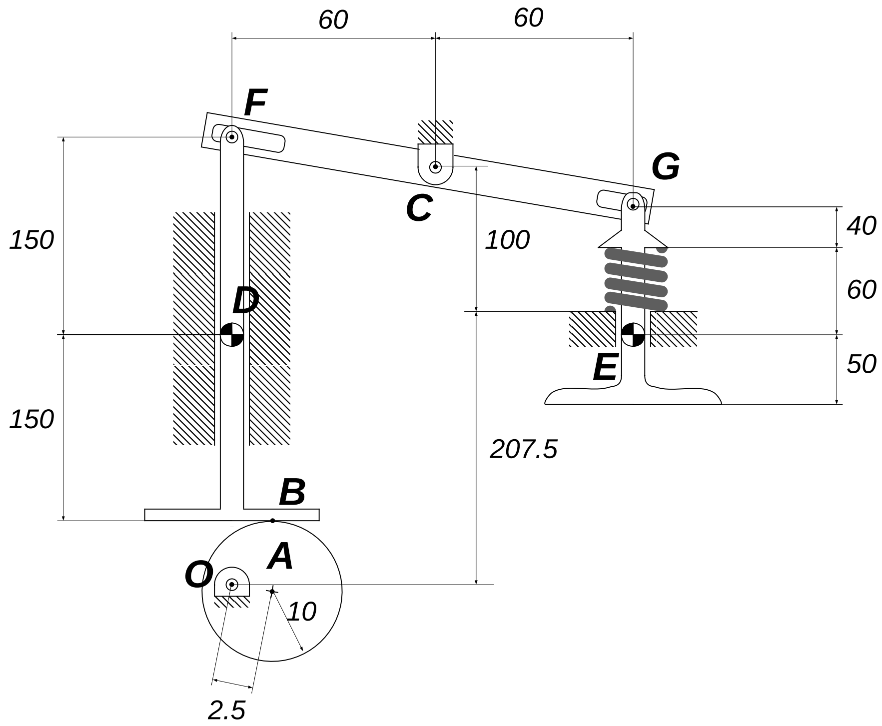

Figure 2: Valve–lifter mechanism scheme

Table 1: Inertial properties of the mechanism

Body	cam	cam follower	rocker arm	valve stem
Mass (g)	30	120	80	60
Inertial momentum ($g.cm^2$)	15	2250	1800	800

2 Requested results

It is asked to determine the equilibrium position of the system. Simulate the mechanism from 0 à $0.06\,s$ when the cam motion turn uniformelly at $3000\,rpm$ with equilibrium position as initial condition.

Note : On `CAGeM`, you can set SIMPLIFY to 0 in case to avoid long CPU–time.

3 Typical results

Figure 3 to figure 5 show the expected behaviour.

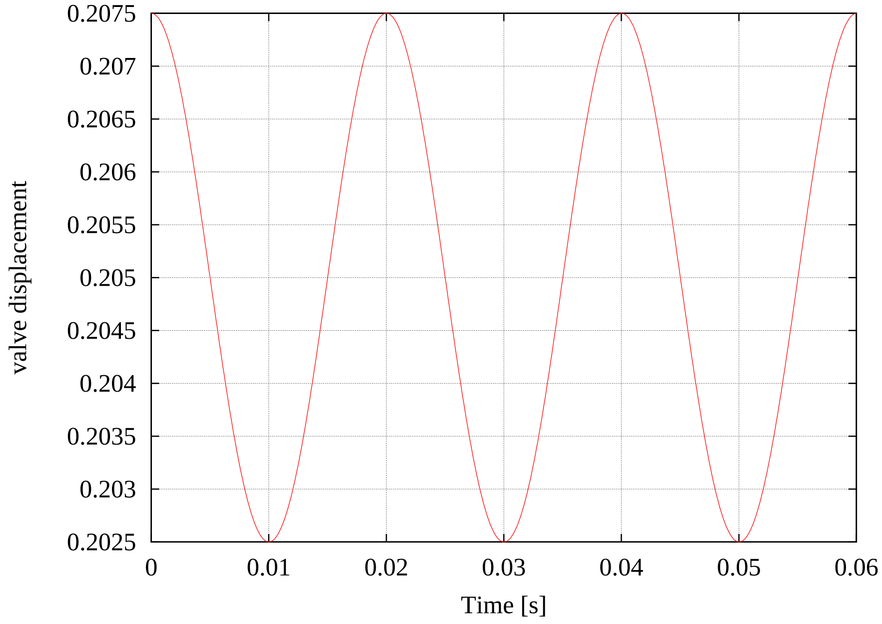

Figure 3: Evolution of configuration parameter

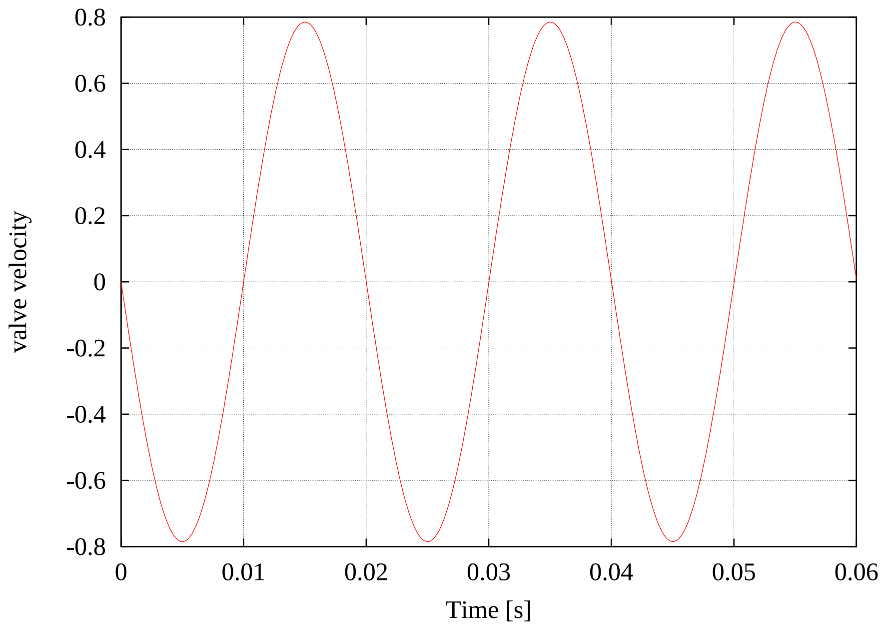

Figure 4: Evolution of first time derivatives of configuration parameter

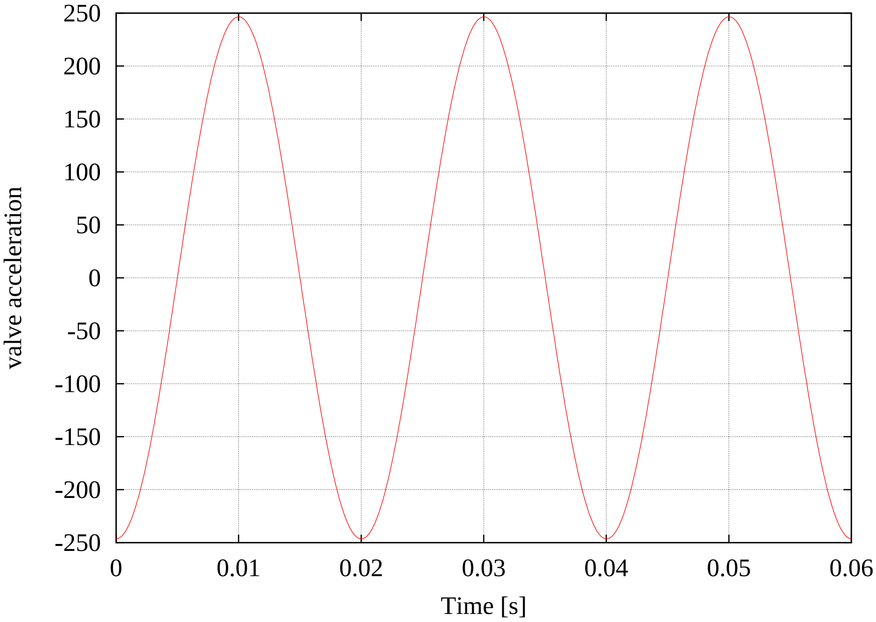

Figure 5: Evolution of second time derivatives of configuration parameter