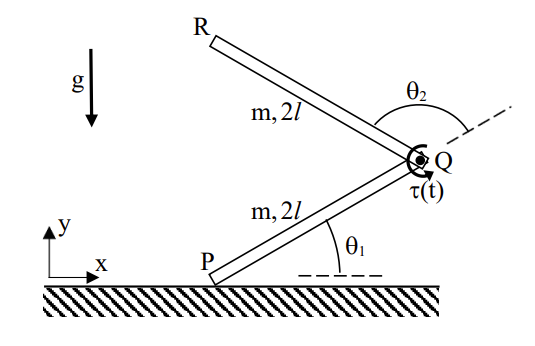
**Final Project –hybrid dynamics of a robotic hopper**

We declare that the project was solely done on our owns. And the main source of resource is based on the lecture notes.

|  |  |
| --- | --- |
| Name | ID |
| Jonathan Oh | 941170383 |
| Xingkun Huang | 941164840 |

**Configuration**

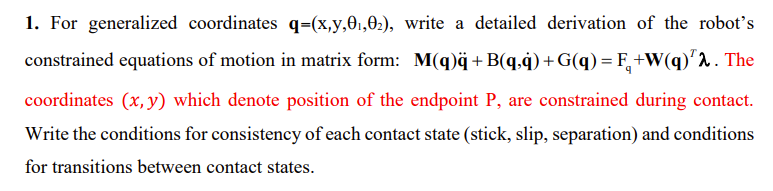
Our robot has following configuration. It’s a simplified model of a robotic hopper composed of two identical links connected by a rotational joint which is actuated by a torque.



We choose the generalized coordinates to be where is the position of the endpoint and are the relative angle between the links. During the entire motion, the actuation torque satisfies the bound . The physical values for the simulations are:

Our objective of the project is to formulate the hybrid dynamics of the robot in all possible contact states, the conditions for transitions between contact states, and the impact law. Also, we will write MATLAB code functions for dynamic simulations of “jumping forward from rest” under a prescribed actuation torque. Finally, we will choose time-profile of actuation torque in attempt to maximize the jump distance .

**Task**



First, we lay out the kinematics of the system. We find the center of gravity of each link. We denote the lower link as link number 1 and the upper one as 2. CG is the center of gravity of the robot.

Their respective velocities are:

Then we can obtain the kinetic energy of the system.

The expression is long, and we won’t display it. The exact expression can be calculated via MATLAB. As for the potential energy,

Then we can write the EOM in a matrix form via Lagrange method.

As for the constraint matrix **,** we would require the velocity at point P.

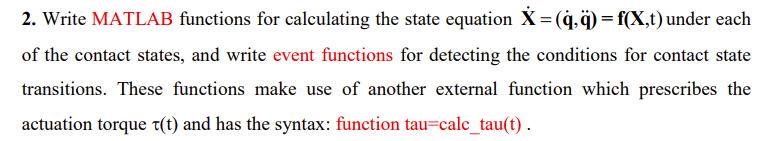
There are three phases for this configuration. To use ODE 45, we need equation of motion for each phase.

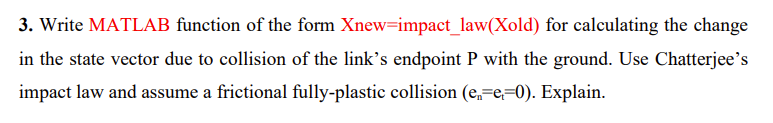
* Stick:
* Slip:
* Contact separation:

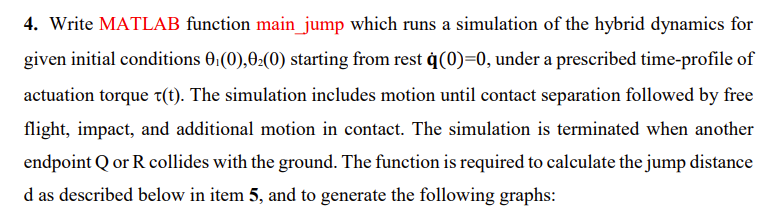
Condition for transition:

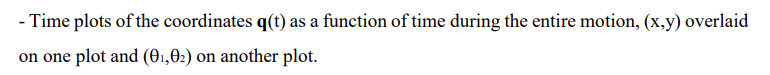
|  |  |
| --- | --- |
| No-slip |  |
| No-slip to slippage |  |
| Slipping |  |
| Slipping to separation |  |
| Separation |  |
| Separation to re-impact |  |

For MATLAB code, we need event function to detect when the transition occurs, and we can terminate the phase calculation and move to a new state.

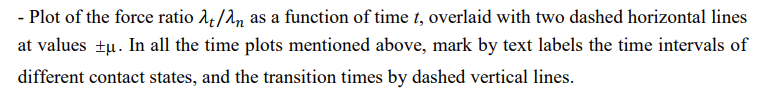


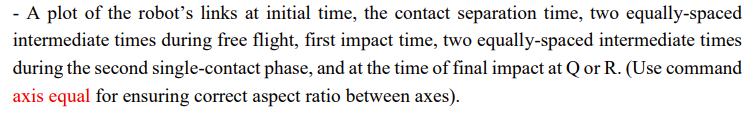


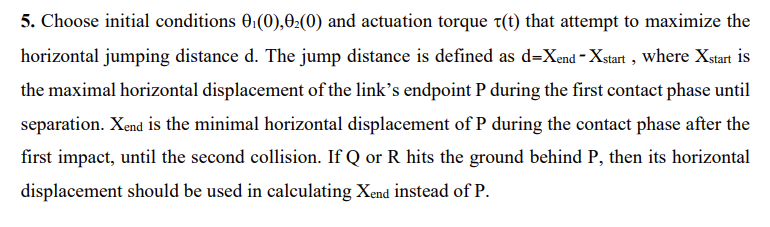












For the jumping challenge, we take the broad jump as consideration. After analyzing the motion of the athlete