**Project Title: Hopping Robot – Model Estimation**

The prediction of model dynamics of an underactuated system using a stochastic approach is a recently introduced control strategy and is being extensively explored in the presently. Motion planning and modeling of a cart-pole has been extensively studied. A successful approach towards a simpler underactuated system provides a research bed for complex systems such as Walking Robot, Hopping Robot. A kind of well-known Hopping robot is extensively studied in both industrial and academic communities for recent decade. For a fully automated robotic system it is important that the control policy must deal effectively by predicting the trajectory motion with the fast-changing environment. Over these years trajectory optimization was been successfully used in the complex locomotion tasks. There are several methods to estimate the dynamic states but, Gaussian mixture model runs with less computation time. So, in this approach we use Gaussian Mixture model to estimate the future dynamical states of the underactuated system.

**Getting Started:**

1. **Prerequisites**

The important prerequisites of this projects are

* Python – Jupiter notebook
* MUJOCO software
* Linux OS

1. **Installing**

* Install sklearn library

**In the anaconda command prompt:** pip install sklearn

* Install numpy

**In the anaconda command prompt:** pip install numpy

* Install MUJOCO

**Link to download:** <https://www.roboti.us/index.html>

1. **Run**

* Initially design the system (robot in this case) using MUJOCO software with C++ programming. Retrieve the state variables data by using sensors in the MUJOCO software. Store the dataset which will be used in the main program.
* With the Linux OS, run the PYTHON\_SIMULATOR\_CODE.py to run the simulation of the system in MUJOCO
* The simulator code has a Machine learning algorithm to estimate the future dynamic variables and the controller uses these variables to actuate the system depending upon the feedback.

**Result:**

The following figure shows the result of the simulation which uses the machine learning algorithm to balance a inverted pendulum.

**![A picture containing object

Description automatically generated]()**

Model Estimation of a Cart pole system.

**Deployment:**

To run in a real time hardware, we need a fully working hardware with the sensors compatible with our PC. Once we get this hardware compatible with the software, we can try implementing on the real world system. E.g. Flight control systems uses model predictive controls for its control system.