

**TEAM MEMBERS:**

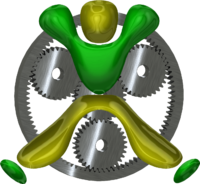
**Zhaoliang Zheng (**[**zhz503@eng.ucsd.edu**](mailto:zhz503@eng.ucsd.edu)**)**

**Dominique Meyer (demeyer@ucsd.edu)**

**Neal Bansal (nkbansal@ucsd.edu)**

**Pengcheng Cao (p5cao@eng.ucsd.edu)**

**Junchao Lin (jul025@eng.ucsd.edu)**



BeagleBone Project

EduRover V2

Catalog

[0 Introduction 3](#_Toc5195830)

[1 Structure 3](#_Toc5195831)

[1.1 Configuration in different modes 3](#_Toc5195832)

[1.2 Sensors and Actuators 4](#_Toc5195833)

[2 Mathematical Modeling 6](#_Toc5195834)

[2.1 Notation 6](#_Toc5195835)

[2.2 Two-wheeled Inverted Pendulum Model and Differential-drive Model 6](#_Toc5195836)

[2.2.1 Motion equations of two-wheeled inverted pendulum 8](#_Toc5195837)

[2.2.2 Lagrangian equations 9](#_Toc5195838)

[2.2.3 State equation linearization 10](#_Toc5195839)

[3 Control system design 12](#_Toc5195840)

[3.1 Control System Theory 12](#_Toc5195841)

[3.1.1 Input & Output 12](#_Toc5195842)

[3.1.2 Stability 12](#_Toc5195843)

[3.2 Controller design 12](#_Toc5195844)

[3.2.1 Equivalent system for balance mode 12](#_Toc5195845)

[3.2.2 Simulation 13](#_Toc5195846)

[4 Higher level application design 15](#_Toc5195847)

[4.1 Way points following based on Motion Capture system 16](#_Toc5195848)

[4.1.1 Closed loop feedback control 16](#_Toc5195849)

[4.1.2 Motion Capture System 16](#_Toc5195850)

[4.1.3 PID Control Algorithm 16](#_Toc5195851)

[4.2 Path planning Algorithm on board 16](#_Toc5195852)

[4.2.1 Workflow 16](#_Toc5195853)

[Appendix A Table 1-3 Servo Motor Properties 17](#_Toc5195854)

[Appendix B The derivation of Lagrangian equations 17](#_Toc5195855)

[Reference 18](#_Toc5195856)

# 

# 0 Introduction

This is an application of the embedded system: Beaglebone. EduRover is originally designed by James Strawson (jstrawso@eng.ucsd.edu) in the Done Lab@UCSD and Coordinated Robotics Lab@UCSD. In version 2, we made some modification on it, redesigned some part of it. In our case, high torque motors could be perfectly embedded into our design and with the higher level control algorithm, we can manually drive EduRover in different modes. Meanwhile, with the high torque motors, we can balance EduRover by using only two wheels when it faces the wall.

# 1 Structure

This chapter describes the structure and some hardware of EduRover as well as the properties of these components. All the Solidworks files can be found in the link: <https://github.com/zhz503/Beaglebone-project-EduRover-V2/tree/master/Edu%20Rover%20Solidworks%20Version2>

1.1 Configuration in different modes  
The figure 1-1 shows the structure of EduRover and in the balance condition.

The figure 1-2 shows the different four-wheeled EduRover driving mode.

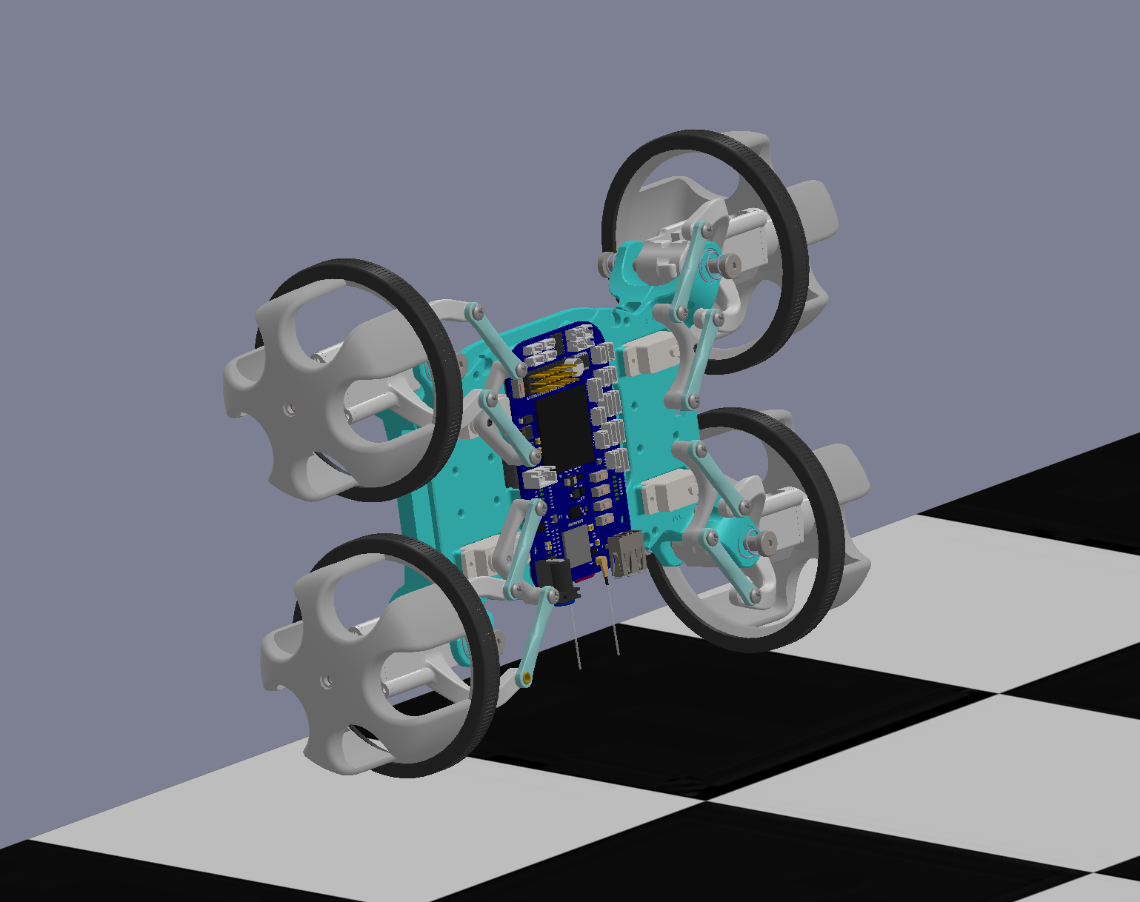


Figure 1-1 Balance Configuration Mode

(a) (b)

(c) (d)

Figure 1-2 (a) is Spinning Mode, (b) is Normal-car Mode, (c) is Quarter Turn Mode, (d) is Sidling Mode

## 1.2 Sensors and Actuators

Table 1-1 Sensors

|  |  |  |
| --- | --- | --- |
| Sensor | Output | Unit |
| Encoder | Angle | rad or deg |
| IMU: Gyroscope | Angular velocity | deg/sec |
| IMU: Accelerometer | Linear velocity | m/sec |

Table 1-2 Actuators

|  |  |  |
| --- | --- | --- |
| Actuator | Output | Unit |
| Servo Motor | Position or duty circle | % |
| DC Motor | PWM | % |

The illustration of DC motor[1] and Servo Motor can be found in appendix A and reference.

# 2 Mathematical Modeling

This chapter describes the mathematical model and motion equations of EduRover. Based on the basic knowledge, the EduRover balance mode can be simplified as a two-wheeled inverted pendulum model[2], and EduRover four-wheeled normal-car mode can be simplified as Ackermann drive (bicycle) model. Spining Mode, Quarter Turn Mode, Sidling Mode can be simplified as Omni Wheel model.

## 2.1 Notation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Items | Parameter | Value | Unit | Meaning |
| EduRover Parameter |  | 9.81 |  | Gravity acceleration |
|  | 0.03852 |  | Wheel weight |
|  | 0.0425 |  | Wheel radius |
|  |  |  | Wheel inertia moment |
|  | 0.6 |  | Body weight |
|  | 0.150 |  | Body width |
|  | 0.048 |  | Body depth |
|  | 0.18282 |  | Body height |
|  | 0.08869 |  | Distance of the center of mass from the wheel axle |
|  |  |  | Body pitch inertia moment |
|  |  |  | Body yaw inertia moment |
|  | 0.0022 |  | Friction coefficient between body & DC motor |
|  | Depends[4]:0.1 |  | Rolling Friction between wheel & floor |
| DC motor |  | 0.1765 |  | Motor torque |
|  | 1.6 |  | Motor working current |
|  |  |  | DC motor inertia moment |
|  | 2.7 |  | DC motor resistance |
|  |  |  | DC motor back EMF constant |
|  |  |  | DC motor torque constant |
|  | 99 |  | Gear ratio |

Note:

DC motor parameters can be found in [1].

are hard to verify, so they are a approximation.

[4] Rubber & Concrete(Dry)

## 2.2 Two-wheeled Inverted Pendulum Model and Differential-drive Model

The EduRover can be simplified as a two-wheeled inverted pendulum model, as described in Figure 2.2-1

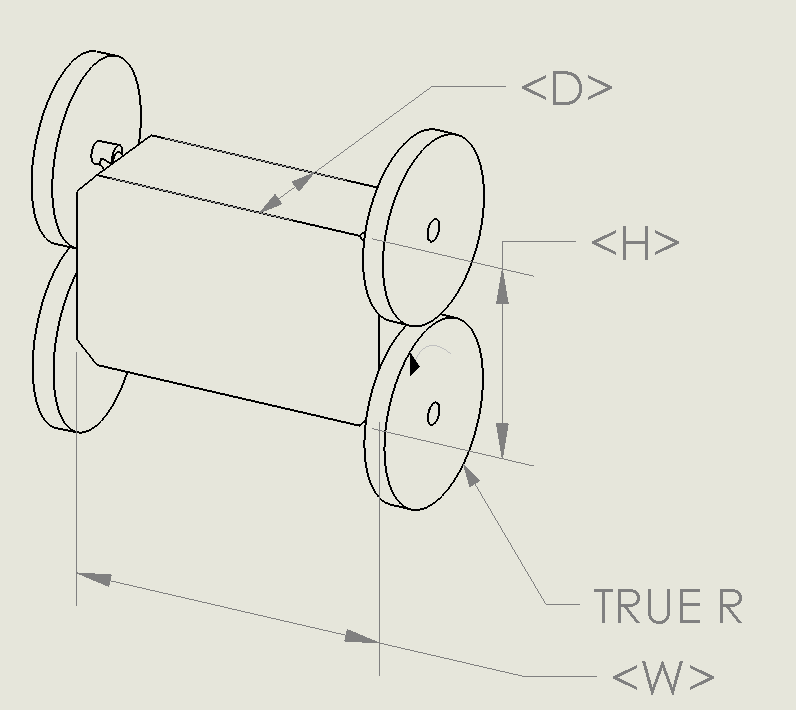


Figure 2.2-1 Two-wheeled Inverted Pendulum Model

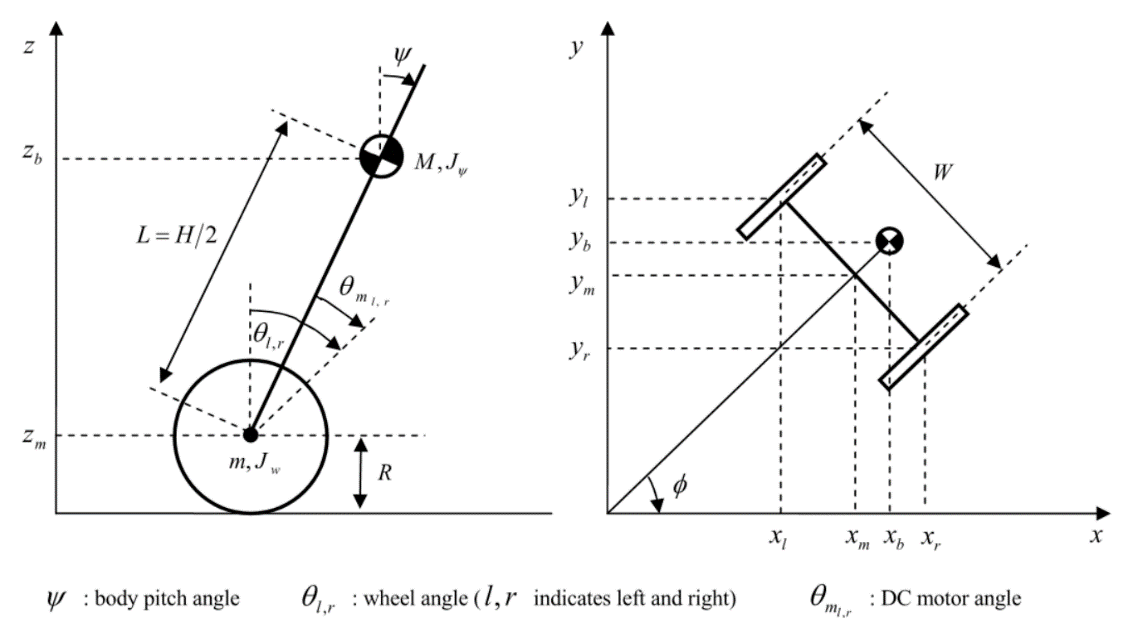


Figure 2.2-2 Side view and plane view of the model

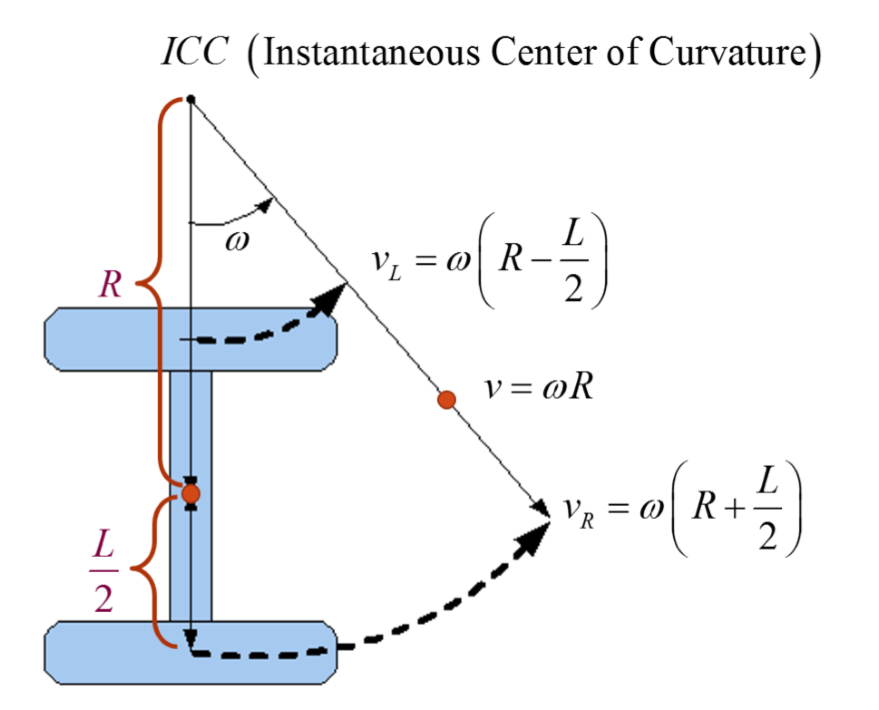


Figure 2.2-3 Differential-drive Model

### 2.2.1 Motion equations of two-wheeled inverted pendulum

We can derive motion equations for inverted pendulum based on figure 2.2-2 and the differential model in figure 2.2-3 differential-drive model [3].

Where,

: average rotational angle of left and right wheel

: Body pitch angle

: Body yaw angle

Based on the kinetic energy theorem and the potential energy theorem, the translational kinetic energy and the rotational kinetic energy , the potential energy are:

Note: the last two terms in equation are the rotational kinetic energy of armature in the left and right DC motor.

### 2.2.2 Lagrangian equations

Based on the Lagrangian method, The Lagrangian has the following expression:

We use the variables as the generalized coordinate. The Lagrangian equation can be derived as:

Expand the equations ()-(), we can derive the following equations, the whole derivation is in the Appendix:

Considering the rolling friction between motor and wheel as well as the friction between wheel and floor, the generalized force are:

Where is DC motor current.

Since the input of our system is voltage of DC motor, in the above equations, we cannot used current directly. Therefore, we have to derive the relationship between current and voltage . From DC motor model, The applied voltage equals the voltage drop across the coil resistance, R, and the inductor, L, plus the back-EMF[5], which is as follows:[4]:

Where, DC motor resistance, DC motor inductance.

For the steady-state torque-speed relationship, has no effect, so term is negligible and is approximated as zero. Therefore, rewrite equation() as:

When substitute the equation() to equations()-(), the generalized force can be expressed as the function of voltage:

Where,

### 2.2.3 State equation linearization

So far, we’ve had all the equations we need. At the balance point, when the body angle changes at about equation() can be linearized. In this case, approximates to be 0, meaning that ,and . The linearized Lagrangian equations() can be rewritten as follows:

Now, we can combine the equ()-() and the equations()-(), and rewrite these equations in matrix form:

Where,

,,

Now, we can consider as the state, as input, we can rewrite equations() as state space equations form:

The state space realization:

,

,

Where, in the MATLAB:

# 3 Control system design

## 3.1 Control System Theory

### 3.1.1 Input & Output

A. Balance Mode

Input :

Output:

B. Four-wheeled modes

Input:

Output:

### 3.1.2 Stability

(not yet finish)

## 3.2 Controller design

### 3.2.1 Equivalent system for balance mode

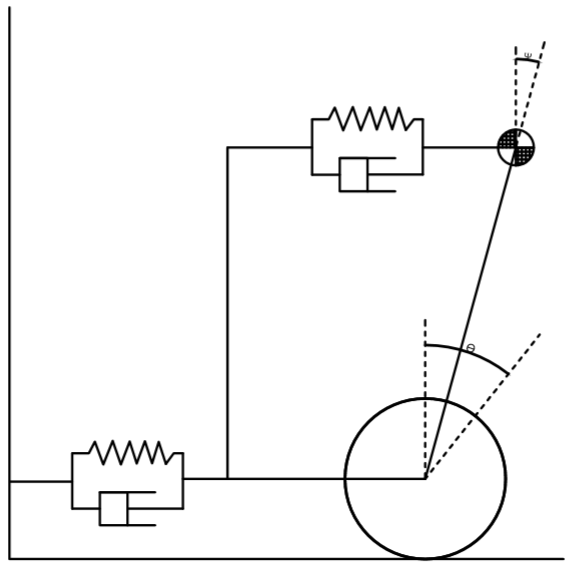


Figure 3.2-1

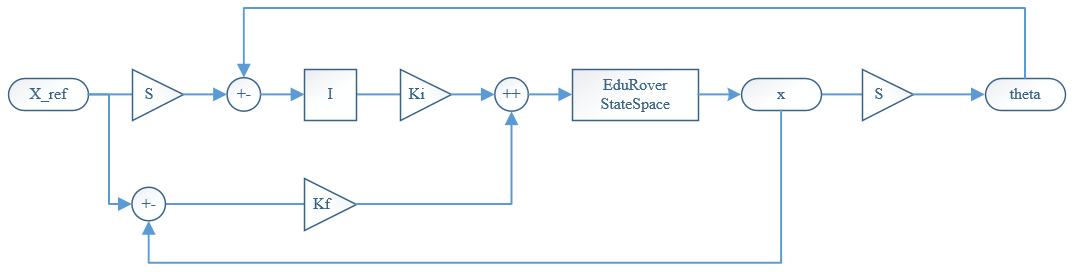


Figure 3.2-2

### 3.2.2 Simulation

***A. Balance Mode***

The EduRover balance mode can only be activated when it faces the wall. In the simulation, we only consider the small angle approximation, which is around . We can set the initial condition: . The simulation runs in Simulink. The result is as follows.

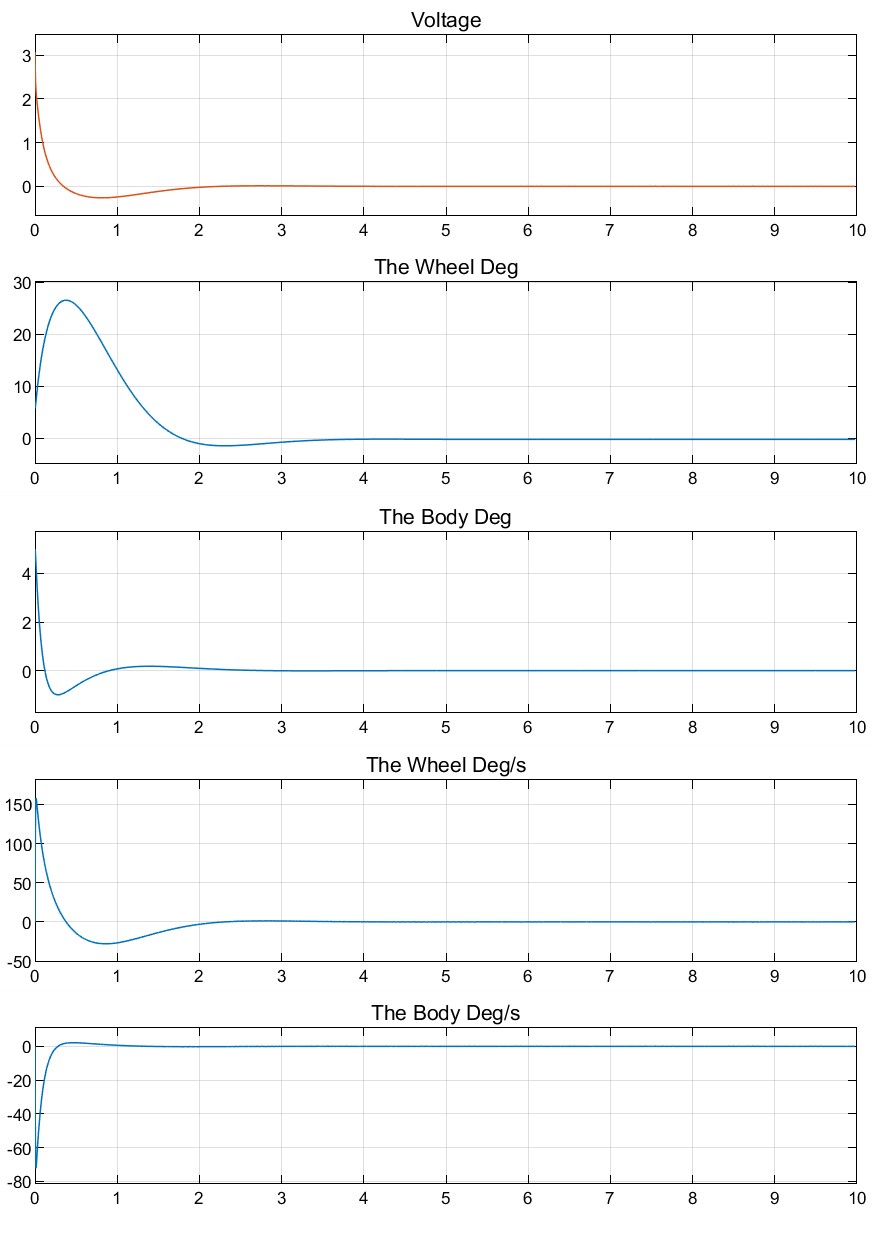


Figure 3.2-1

***B. Four-wheeled driving mode***

In the four-wheeled driving mode, four situation has to be consider: (a) Spinning mode, (b) Normal-car mode, (c) Quarter-turn mode, (d) Sidling mode. To switch from different modes, we have set up a servo logic for these four different modes.

For manual control, we want to minimize the number of inputs as much as possible. The figure 3.2-2 shows that the servo logics in Simulink. In this servo logics, the flag constant controls the logics switching between different modes. Flag(1): Spinning Mode, flag(2): Normal Car Mode, flag(3): Quarter Turn Mode, flag(4): Sidling Mode.

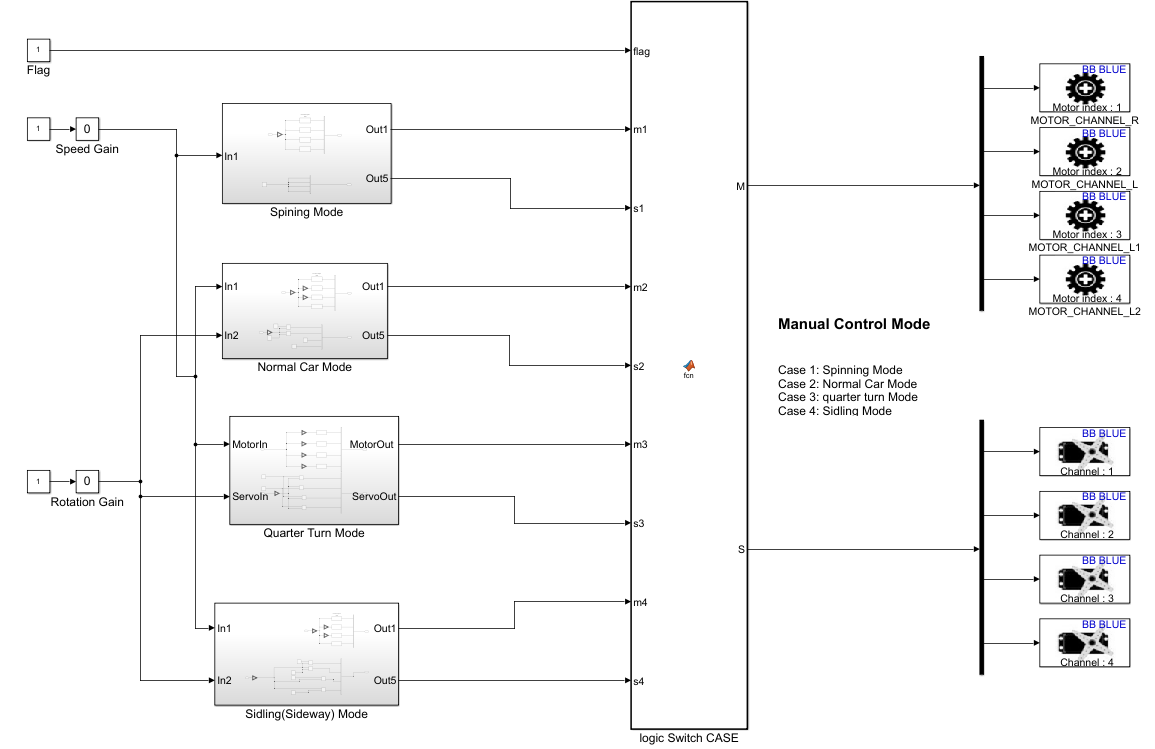


Figure 3.2-2 Servo logics

3.2.3 Real-world Experiment

***A. Balance Mode***

(not yet finish)

***B. Four-wheeled driving mode***

We can use the controller designed in Simulink to control the EduRover in real-time in real-world. The following video links can show you how it performs:

1. <https://github.com/zhz503/Beaglebone-project-EduRover-V2/blob/master/RW%20Simulink%20Control%20and%20Sensing/ManualControl.mp4>
2. <https://github.com/zhz503/Beaglebone-project-EduRover-V2/blob/master/RW%20Simulink%20Control%20and%20Sensing/logics%20switch.MP4>

# 4 Higher level application design

EduRover is a very decent platform for developer to do higher level application design. There are many applications right now.

In this report, I only list some of the on-going applications:

(1) Way points following based on Motion Capture system

(2) Path planning algorithm based on Motion Capture system

(3) Feature recognition based on web cam on board

(4) ORB-SLAM based on RGBD camera

## 4.1 Way points following based on Motion Capture system

### 4.1.1 Closed loop feedback control

(not yet finish)

### 4.1.2 Motion Capture System

In this project, the motion capture system we used is VICON system. According to the description [5], Vicon are the premier solution for UAV and Robotic studies because we understand what is important; A highly accurate system that provides low latency data that is easy to use. With our turnkey approach and easy to access our DataStream (via TCP, UDP, etc.) users can easily integrate accurate Vicon data into virtually any control system.

### 4.1.3 PID Control Algorithm

This PID control algorithm is designed by Dominique Meyer, which is available in the Github: <https://github.com/zhz503/Beaglebone-project-EduRover-V2/tree/master/way_points%20following>

## 4.2 Path planning Algorithm on board

### 4.2.1 Workflow

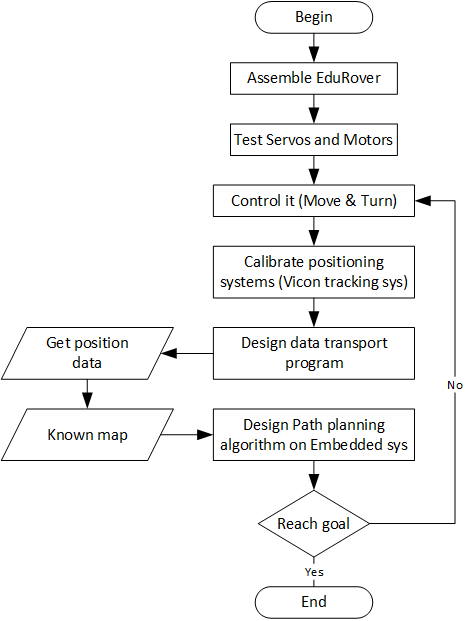


Figure 4-1

# Appendix A Table 1-3 Servo Motor Properties

Table 1-3 Servo Motor Properties

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Product | Seller | Link | No-Load Speed @ 4.8V (sec/60°) | Price | Qty | Price/Unit |
| HS-45HB | ServoCity | <https://www.servocity.com/hs-45hb-servo> | 0.14 | $15.99 | 1 | $15.99 |
| RB-Fit-03 | RobotShop | <https://www.robotshop.com/en/9g-micro-servo-motor-4-8v.html#Supplier-Product-Code> | 0.12 | $3.56 | 1 | $3.56 |
| Miuzei SG90 | Amazon | <https://www.amazon.com/Micro-Helicopter-Airplane-Remote-Control/dp/B072V529YD?ref_=fsclp_pl_dp_1&th=1> | 0.09 | $17.99 | 10 | $1.80 |
| Seamuing MG90S | Amazon | <https://www.amazon.com/dp/B07F7VJQL5/ref=psdc_2234131011_t2_B072V529YD?th=1> | 0.11 | $20.99 | 6 | $3.50 |
| SG90 Micro Servo | Amazon | <https://www.amazon.com/dp/B07F2XDZZ7/ref=psdc_2234131011_t2_B072V529YD?th=1> | 0.1 | $19.99 | 10 | $2.00 |

# Appendix B The derivation of Lagrangian equations

# Reference

1. https://www.servocity.com/270-rpm-micro-gear-motor-w-encoder
2. NXTway-GS Model-Based Design, http://www.pages.drexel.edu/~dml46/Tutorials/BalancingBot/files/NXTway-GS%20Model-Based\_Design.pdf
3. UCSD ECE motion and observation model, https://natanaso.github.io/ece276a/ref/ECE276A\_8\_MotionObservationModels.pdf
4. https://www.motioncontroltips.com/faq-whats-relationship-voltage-dc-motor-output-speed/
5. https://www.vicon.com/motion-capture/engineering