Combat Control

Griffin Tabor

Abstract—The controls problem involved in combat robotics is quite interesting. We are working on simulating and controlling 250 lb Battlebots. The robots are designed with extremely heavy loads sticking off the sides spinning at several thousand rpm. This presents an interesting controls problem because the load is not balanced and the gyroscopic affect can cause quick acceleration to force the robot to turn over. The Lagrangian approach was used to make the dynamic model for the robot.

1 Introduction

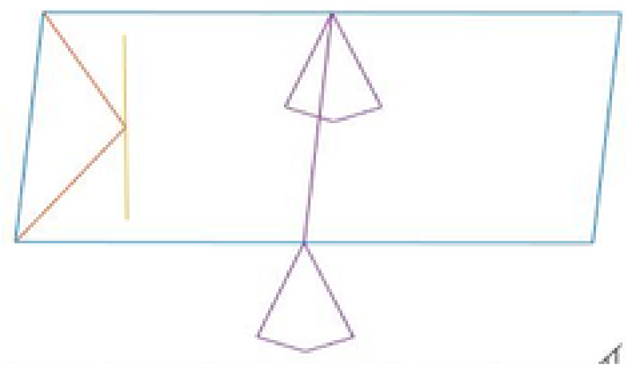
We started with a combat robot known as Minotaur as the subject of our study. This robot is interesting because the driver will regularly perform flips using the gyroscopic affects of his weapon. The driver is able to do precision turns normally while his weapon is moving and flip over with more aggressive turns. This behavior makes it both interesting and difficult to properly control.

mds

April 26, 2018

2 Robot Model

The simple model of the robot that we built defines the robot as 3 point masses.The center of the robot is treated as a single mass and the weapon is treated as 2 masses on opposite side of the center of rotation. The two masses help ensure the system is balanced without the undo burden of computing a proper cylinder.The wheels on each side of the robot are modeled as 3 points and are used for computing the ground contact. The wheel points are fixed to the chassis and do not rotate.



The dimensions of the robot were measured were applicable and interviews with the builders where measuring was insufficient.

chassis length	0.6m
chassis width	0.6m
wheel radius	0.1m
weapon offset	0.2m
weapon radius	0.0.85
weapon mass	33kg
chassis mass	80kg
weapon max speed	10k rpm

3 Dynamic Model

The dynamic model was computed using the Lagrangian method entirely symbolically. The state being tracked is x, y, z, roll, pitch, yaw and weapon location. As well as the time derivative of each. Working fully symbolically allows taking the Jacobion easily. The fully symbolic model was then converted to a matlab function with the magic matlab function matlabFunction() which resulted in a 3000x evaluation speed increase.

4 Simulation

The robot was simulated using matlab's ODE solver. The ground contact is handled using the 3 wheel points. If any wheel point is lower than the ground plane both the z acceleration and velocity of the wheel point are adjusted. Tests were conducted to have the robot spin iin circles while the weapon spun up to speed. In several experiments the robot completely flipped over while still maintaining one powered wheel on the ground.

The authors would like to thank BattleBots and Team Riobots.