Differential Robotic H-Joint

Brendon Espinoza
BS Mechanical Engineering

Portland State University
Maseeh College of Engineering and Computer Science

LOAD BEARING PROPERTIES

The traditional **Y-joint** used in many limb based robots has a disadvantage, it has **hanging shafts**. A **hanging shaft** allows the drive shaft to **undergo deflection** due to a force couple from the load on the shaft, and the reaction acting through the bearing. This puts a load on the gear train **wearing it down** over time.

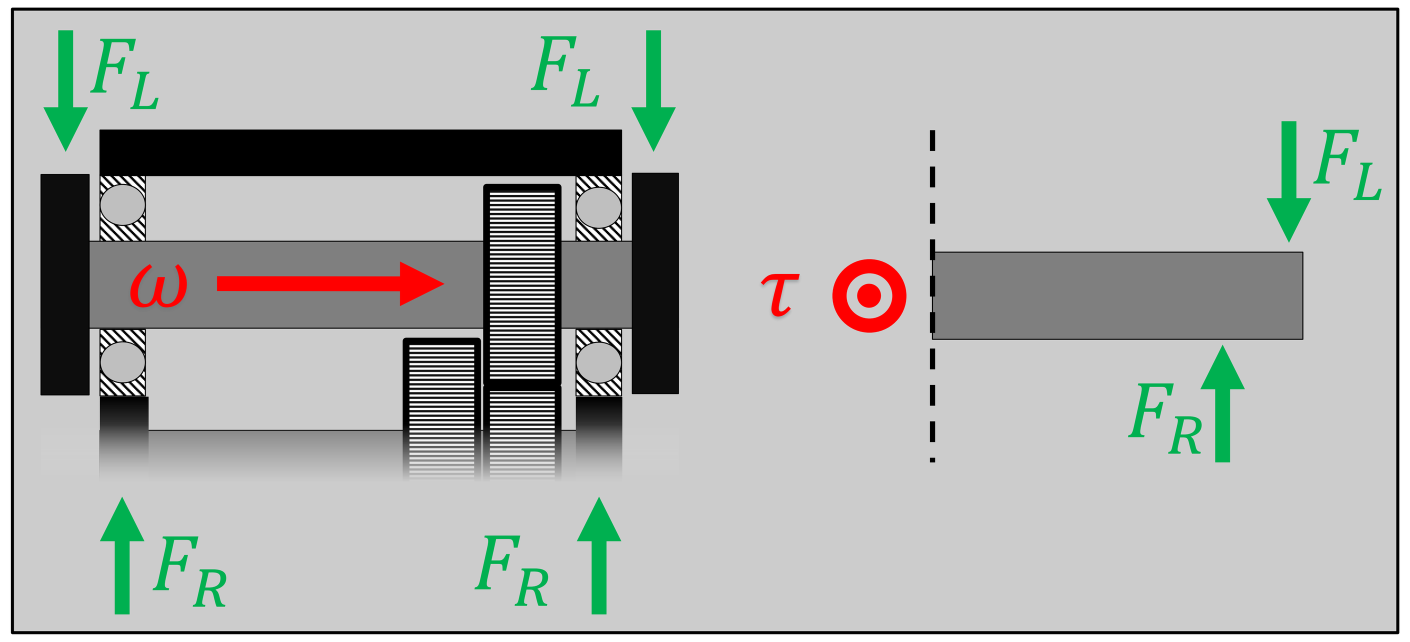
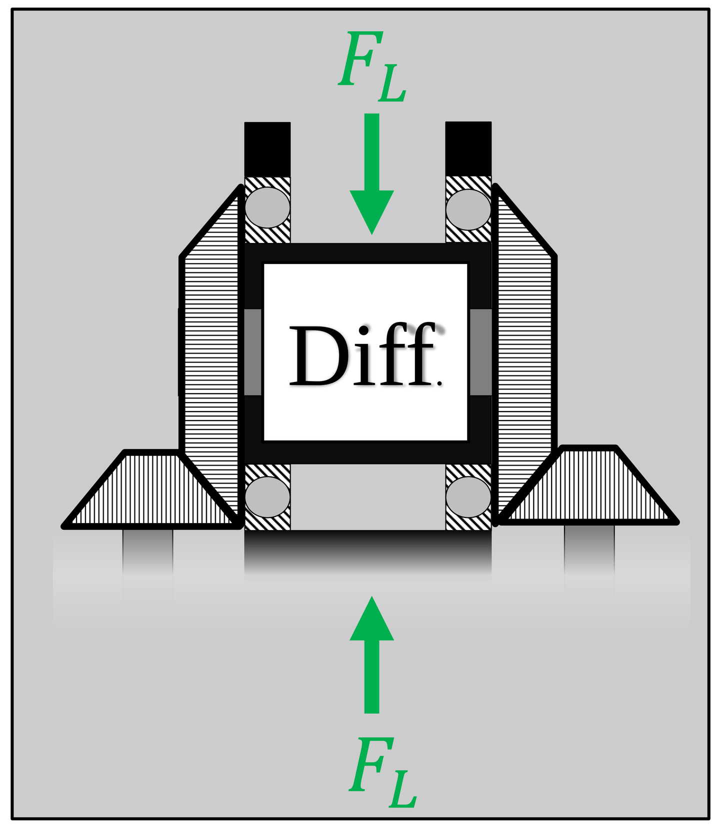


Figure 1: Y-Joint free body diagram

In an **H-joint** deflection due to loading is eliminated. The load bearing ends of the shaft are brought to the middle of the joint, **directly supported** by bearings. Theoretically none of the other drive components bear any load.

INCLUSION OF DIFFERENTIAL



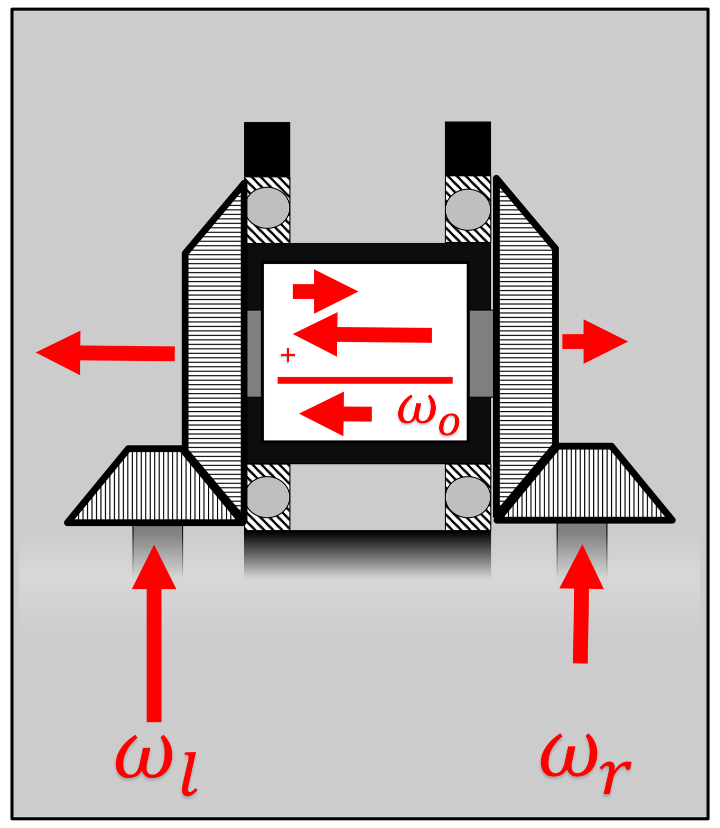


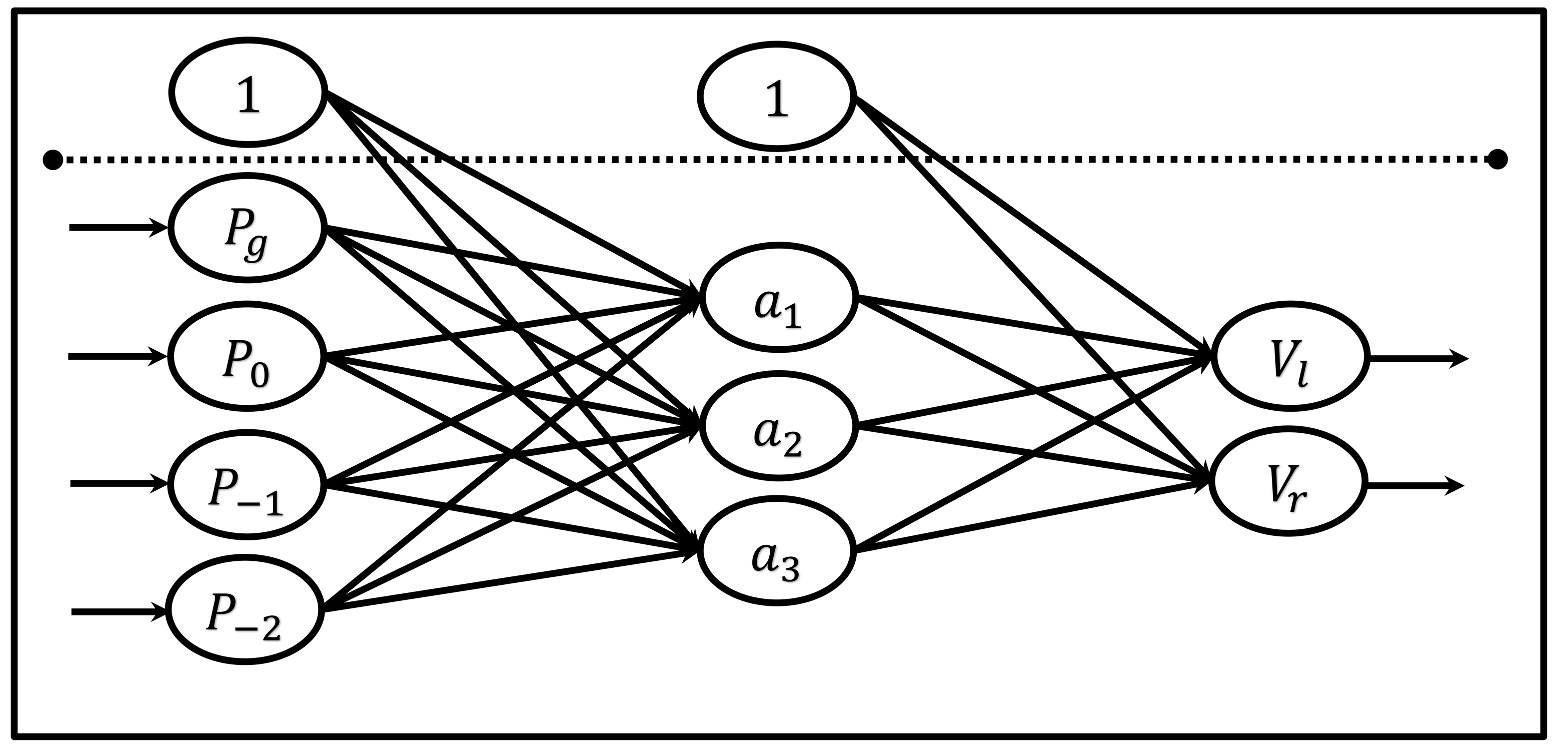
Figure 2: H-Joint free body diagram *Figure 3: Differential speed summation*

The increased volume of the load bearing member allows the inclusion of a **differential**, a speed summing mechanism. The two inputs of the differential transfer their speeds through **bevel gears**. This results in an equation for the speed of the output: $\omega_o = \omega_r - \omega_l$.

With this configuration an important advantage is introduced to the **H-joint**, the output can be driven **bidirectionally** by **unidirectional** inputs.

For **brushless DC motors** this means that there is no danger of **overload**, and the control algorithm need not include **dead-time** when performing polarity reversal. In terms of **speed reduction** the advantage is that **high backlash gear sets** can be used which reduces cost, and complexity. Additionally, the **H-joint** has ample space between the input shafts for the placement of sensors, and controllers.

NEURAL NETWORK CONTROLLER



Traditional controller design methodologies suffer from many **practical issues**, specifically the need to derive the **equations of motion** of the system being controlled in **closed form**. Whereas designing a **neural network controller** only requires that the system is simulatable.

Neural networks have the disadvantage of needing to be trained. The first choice of training method for any sequential operation is **reinforcement learning**. But for simple controllers, with few weights, a **genetic algorithm** can be used.

A **deep neural network** with 4 inputs, 3 hidden layers, and 2 outputs was implemented. The inputs were the **goal position**, **current position**, and **two previous positions** of the output. The outputs are the **speed controller input voltage** for each motor.

The idea behind the choice of inputs is that the controller needed to have information about the entire state of the system. Assuming the system can be approximated as **second order dominant** only the 0^{th}, 1^{st}, and 2^{nd} derivatives needed to be supplied, which can be built from the chosen inputs.

PHYSICAL IMPLEMENTATION

A physical prototype was made which implemented position control. Certain compromises were made in the prototype design. The bevel gears were put on the outside of the driving pinions, and the bearings were placed on the shafts instead of the cuff.