ABSTRACT

Investigating a Roving Agent from Inception
by
Joseph Churchwell

Investigation into the design challenge in the creation of an omnidirectional autonomously roving agent and explores methods to simplify the design complexity using advanced machine learning algorithms, low level design control, and lessons learned.

TABLE OF CONTENTS

PAGE

ABSTRACT....#

LIST OF TABLES....#

LIST OF FIGURES....#

ACKNOWLEDGEMENTS....#

CHAPTER

Introduction....4

1.1 Low Level Design Philosophy....4

1.2 Control Qualifiers....5

1.3 Challenges of Control....5

1.4 Percepts....5

1.5 Summary of Subsequent Chapters....5

Past Generation of the Roving Platform....6

2.1 Principles of Rover Control....6

2.2 Rover 5 Mobile Platform....8

2.3 Sensors....9

2.3.1 Odometry....10

2.3.2 current....11

2.3.3 Inertial measurement....11

2.3.4 lidar....11

2.3.5 vision....11

New generation of the roving platform....12

3.1 OMNIDIRECTIONAL DESIGN....12

3.2 Motor Drive Modeling....15

3.3 Motor Turning Requirements....25

3.4 Selected Motor Specifications....29

LIST OF TABLES

PAGE

Table 3.2.1 – Drive Wheel Moment of Inertia..22
Table 3.2.2 – Drive Wheel Input Parameter Assumptions 23
Table 3.2.3 – Drive Wheel Output Parameters 24
Table 3.3.1 – Wheel Turning Motor Input Parameter Assumptions..27
Table 3.3.2 – Wheel Turning Motor Output Parameters 28
Table 3.4.1 – Selected Wheel Drive Motor Specifications...29
Table 3.4.2 – Selected Wheel Turn Motor Specifications 31

CHAPTER 1

INTRODUCTION

While there are a lot of algorithms available to control roving vehicles, it is still difficult to control their low-level application. A roving robotic application will need to be built from the ground up for it to be sophisticated enough for a good use. It will need a wheel (or track) and something to drive it. Each motor will need an H-Bridge driver (forward and back movement) with controller and current sensing, to implement decent control. Subsequently, the application will require PID controllers that are tuned to control the wheels on the roving vehicle which can be a daunting task for a robot builder.

1.1 Low Level Design Philosophy

Stating “fix it in software” is a complete misnomer that just doesn’t make sense and overcomplicates the programmer’s role. To make a system easy to implement and save time necessitates an adherence to remarkable low level design philosophy. Technology today makes that which was complicated a decade ago attainable. Makers are fueled by the maker movement to come up with design strategies starting from a root level.

The robot described hitherto adheres to a root level design strategy that may be used to build larger systems. Lessons learned from a previous robot, also detailed here, are incorporated and Classic Mechanics is used to define the system using as stated assumptions about the robotic platform. Motors are selected using static analysis and their performance curves are defined. Electrical systems are designed and controllers are specified that can fundamentally handle all low-level input/output.

1.2 Control Qualifiers

Sensors are evaluated and commissioned to ensure the success of the controlling paradigm. These sensors include quadrature encoders, current feedback, homing switches, accelerometers, gyroscopes, magnetometers, LiDAR, ultrasonic, and stereo vision. System performance is also of concern as the system will need the correct response time to function properly. Controls are vetted to ensure success over a multitude of environment scenarios using the associated sensors and dynamic response of the system as building blocks.

1.3 Challenges of Control

To control an omnidirectional robot is not as easy to do as a standard 4 wheeled rover or even a two-wheeled inverted pendulum. However, this structure of robot will be able to be a fully-fledged smooth and sensible 2 DOF platform. Dynamic control of four planar joints needs to be evaluated and multiple controllers will be used for all eight motors adding to system complexity. Trajectory planning is required to make the robot move in a smooth fashion. On top of this low-level control will be the autonomous agent that will command the low-level actuation. The autonomous agent will need to move unimpeded through its environment and accept high level commands. The task defined is to simplify the control methodology as much as possible.

1.4 Percepts

TBD

1.5 Summary of Subsequent Chapters

TBD

CHAPTER 2

PAST GENERATION OF THE ROVING PLATFORM

There are many forms of robotic rovers from those based on the least expensive radio controlled device (RC Car) to modern self-driving vehicles to the most expensive robotic platform ever created by mankind (Nasa's Curiosity Martian Rover). Applications of these robots range wildly while their basic components remain similar. Many of the problems associated with roving platforms arise from their control and algorithmic complexity. For example, this project started with an off-the-shelf base robotic platform that included tracks like a tank. Turning a 4-wheeled powered roving vehicle created difficulties resulting from its intrinsic turning behavior (slip turning) as the robot would experience jitter.

2.1 PRINCIPLES OF ROVER CONTROL

Equation 2.1.1 below describes the classic control theory algorithm called PID – Proportional, Integral, and Differential control loop. PID feedback control loops are ubiquitously used. Some examples of use include HVAC, cruise control, washing machines, turbo-machinery, and autopilot systems where a physical object (heating element) is controlled with an output variable (voltage applied) and control feedback utilizes a feedback sensing parameter (thermocouple).

$$u_t = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \qquad 2.1.1$$

Classic control theory has utilized PID controllers for many wheeled roving systems to control wheel position, robot orientation, internal temperature control, object avoidance, etc. The gains for the system need to be tuned using a heuristic (Ziegler–Nichols), simple guess and check (Manual Tuning), or more advanced automatic algorithms. In recently PID controller gains have been determined automatically using machine learning algorithms such as genetic algorithms and neural networks.

A myriad of PID control loops are of use to control various control outputs. Typical low level control loops include position and velocity control. Using these two PID controllers allows the roving vehicle to move in all directions with some adverse behavior. Wheels lose traction when their orientation is not changed. When this happens there is error in the motor encoder reading of actual position. More intricate control will be required to make the rover better able to perform turning maneuvers in Euclidean space.

It is important for any autonomous agent to know where it is in relation to its ambient environment. Assuming a map of the immediate environment has been pre-defined, in a non-evolving space, the rover can determine where it is without discovery. The function of localization is typically based on wheel encoder movements and camera and/or LiDAR to procure identifiable features of the space used to feedback the rover's position and orientation (POSE). An Inertial Measurement Unit (IMU) is utilized to determine POSE of the robot as well as familiar environmental features (extrinsically based on a priori knowledge). These ideas have been evaluated to one degree or another.

2.2 ROVER 5 MOBILE PLATFORM

The first robotic platform created through this project was based on the Dagu Rover 5 Tracked Platform. The base platform utilized four drive motors with encoder signals and proved to be a great starting point for automatic control algorithm discovery.

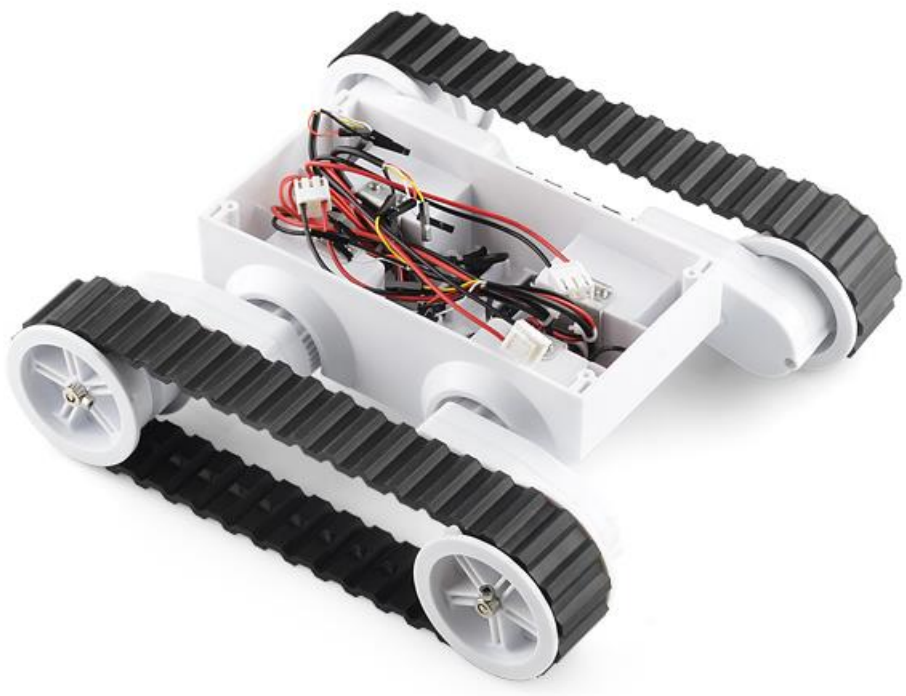

Figure 2.2.1 Rover 5 Tracked Platform

To drive the motors on the Rover 5 a motor driver board had to be selected. A matching driver board was procured through the same vendor which included 4 drivers, XOR encoder mixing, and current feedback. Since there were no controllers available to control the driver board, read the quadrature encoder signals, and monitor current a board was created using a Teensy 3.2 micro-controller.

2.3 SENSORS

Sensors are inherently flawed in that none are absolutely discretized. IMU sensors have problems with sensor drift over time which makes them difficult to use on their own to determine the robot POSE. Encoders are then used in combination. However, encoders have their own flaws resulting from the fact that no roving robot will have a wheel coefficient of friction equal to one. Adding to this, it may be impossible to calculate a coefficient of friction for all materials under all conditions. LiDAR and vision can also be used to ensure all sensing can be adjusted by incorporation of feature space.

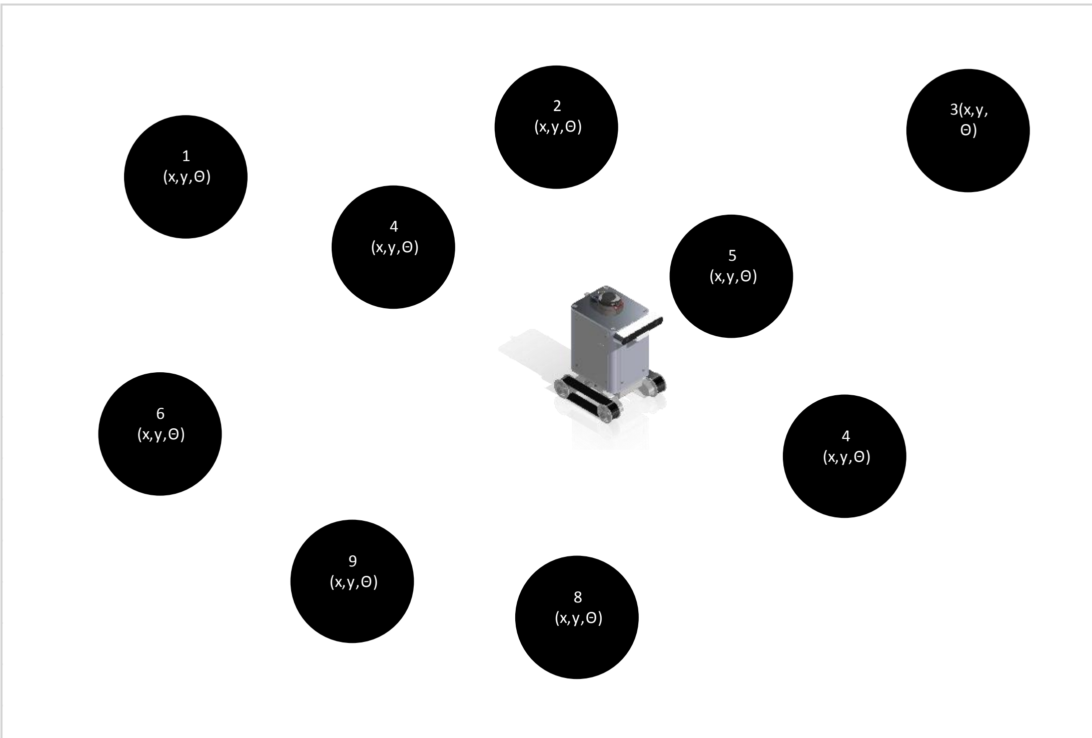

Figure 2.3.1 – Robot Feature State Space

2.3.1 ODOMETRY

Odometry is a measure of positional change over time and (in the case of a roving platform) refers to encoder signals. On the Rover 5 platform Odometry is performed using quadrature encoders on all four wheels. To measure translational distance encoder pulses are used with their relationship with wheel diameter as defined in equation 2.3.1.1 and detailed in Figure 2.3.1.

$$L = \pi D * \frac{N_C}{N_R} \qquad 2.3.1.1$$

Where:

$L = Translational\ Distance$

$D = Diamter\ of\ the\ wheel$

$N_C = Encoder\ counts$

$N_R = Encoder\ counts\ per\ revolution$

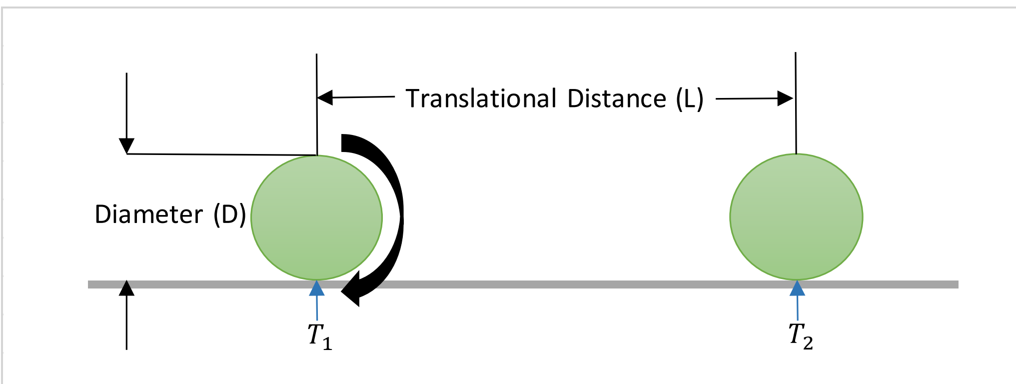

Figure 2.3.1 –Translational Distance

2.3.2 CURRENT

A current sensing device is used to determine the required torque on the motor. Current sensing is accomplished on the motor driver board. Current sensing for all 4 motors is required to ensure the motors don't stall, enables free run detection when the other motors are under loading conditions, detection of substrate material friction (in conjunction with IMU).

2.3.3 INERTIAL MEASUREMENT

An Inertial Measurement Unit (IMU) is a device that reports back linear and rotation movement of a body. IMUs are, typically, mems accelerometers, gyroscopes, and magnetometers that are combined to provide a body's yaw, pitch and roll (or quaternions for singularity free requirements) when sensor fusion is incorporated. The Rover 5 platform has three IMUs to include the BNO055, BNO080 and UM7-LT so that the rover can determine POSE in unison with encoders, LIDAR, and vision.

2.3.4 LIDAR

LiDAR is an acronym for Light Detection and Ranging which can determine distance to the nearest object using time of flight. The sensor used on the Rover 5 platform is the RP LiDAR A2 which can rotate 360° using a brushless motor to create a 2-dimensional map of the surrounding environment.

2.3.5 VISION

A 3D vision system is incorporated to sense the environment around the Rover 5 platform. The Zed camera is a stereoscopic vision sensors that gives the robot information regarding the distance of objects surrounding the robot.

CHAPTER 3

NEW GENERATION OF THE ROVING PLATFORM

The first generation of robotic rover had many problems derived from instability while turning, slow speeds, plastic gear stripping and wear, and small overall size so a ground up re-design was orchestrated. A new platform offers many benefits but requires more analytical thinking to make it capable. Many options exist to turn including changing orientation of the wheels to 45°, differential, Ackermann, Mecanum wheels, and directly changing the orientation of the wheel (casters). Directly changing the orientation of the wheel is said to be the best and most efficient design for a 4-wheeled robot platform.

3.1 OMNIDIRECTIONAL DESIGN

Vibration and power inefficiency on the original robot, stemming from the stationary assignment of the wheels, resulted in poor performance. Turning had to be completed using differential steering between each side which required the robot to slip along the wheel base. A purely omnidirectional wheel design was implemented using two degrees of freedom to ensure the second iteration of the robot did not face the same challenge. Mecanum wheels were not used since encoders will no longer work with them and they are not as efficient as a fully powered 2DOF wheel system. The conceptual 2DOF design is described in Figure 0.1, Figure 0.2, and Figure 0.3 below. The 2DOF design gives rise to movement in 2 dimensions so as to be Omnidirectional (although not holonomic).

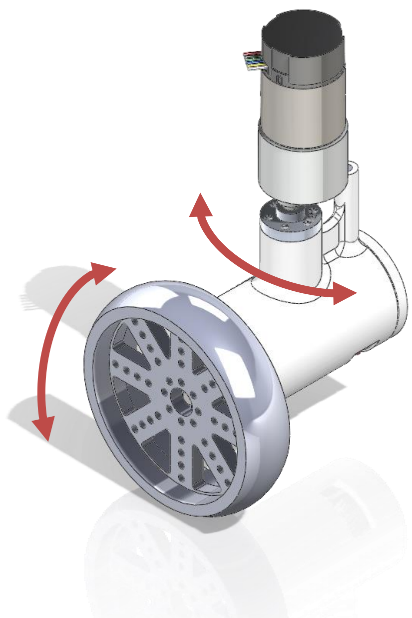

Figure 0.1 – 2DOF Wheel Isometric View

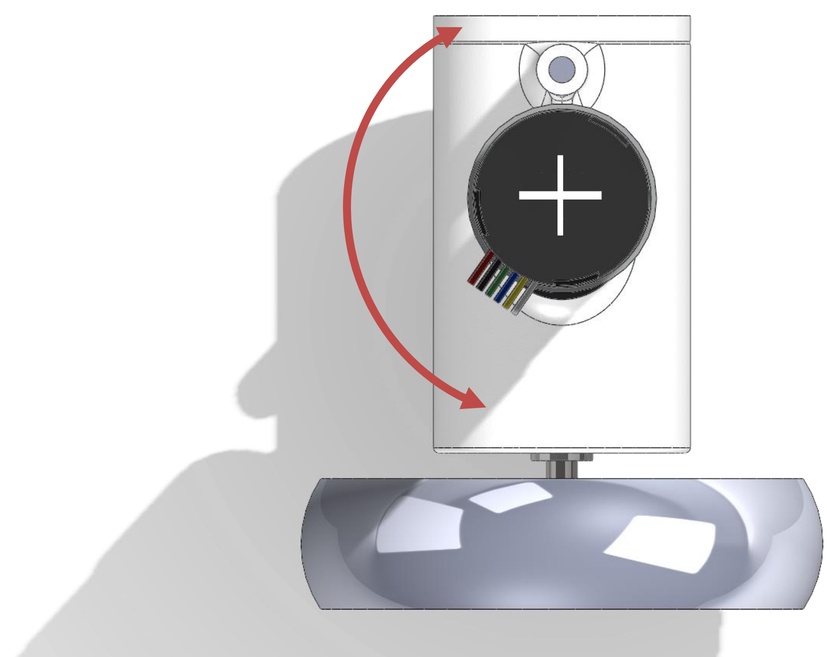

Figure 0.2 – 2DOF Wheel Top View

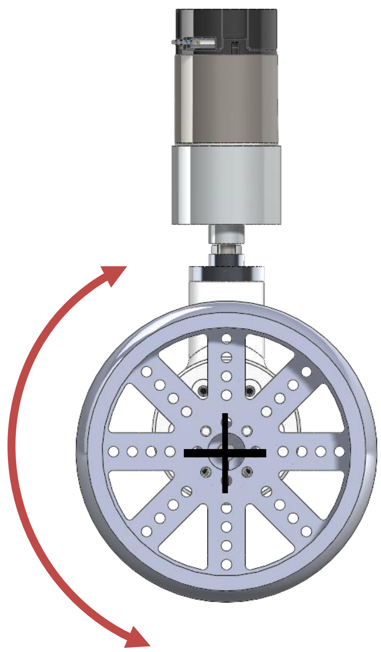

Figure 0.3 – 2DOF Wheel Front View

3.2 Motor Drive Modeling

The design of a new robotic roving platform should start with the motors that drive the wheels. One must know the component forces the rover is under when the worst design circumstance is observed. Free body diagrams (FBD) are a useful method to visualize component forces which can be seen in Figure 3.2.1 below. Classic Mechanics is used assuming static equilibrium of rigid bodies.

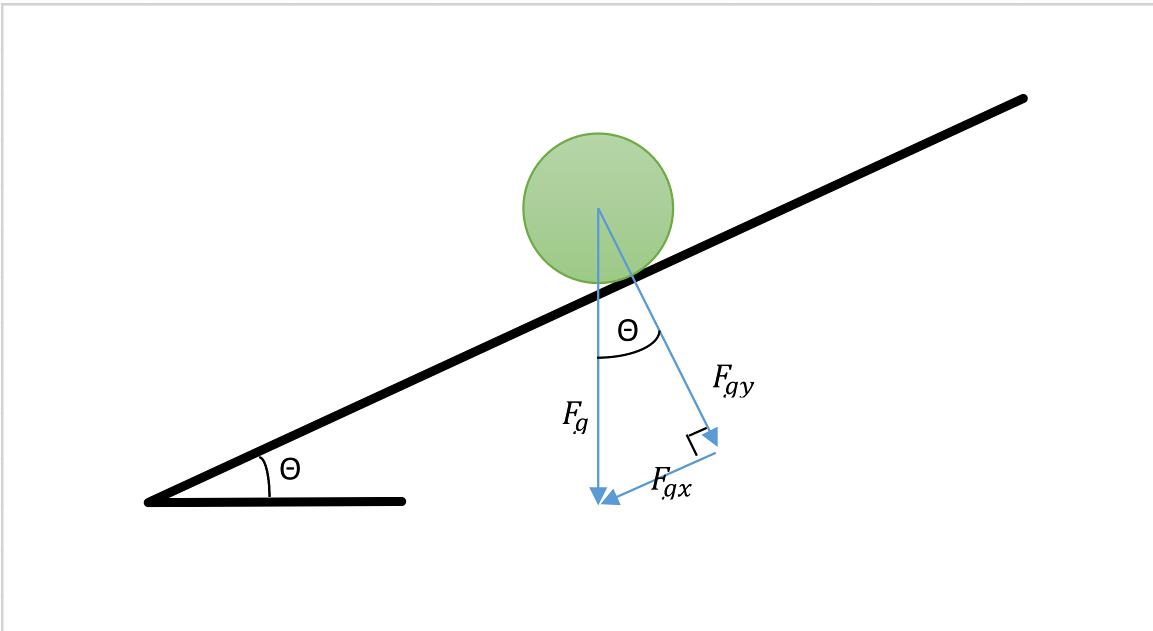

Figure 3.2.1 – Wheel Component Forces FBD

Component force equations 3.2.1 and 3.2.2 were determined from the FBD in Figure 3.2.1 above.

$$F_x = F_g * \sin\theta \qquad 3.2.1$$

$$F_y = F_g * \cos\theta \qquad 3.2.2$$

Where:

$F_g = Mass\ (m) * Acceleration\ (g)$

$F_x = Force\ Applied\ in\ the\ X - Direction$

$F_y = Force\ Applied\ in\ the\ Y - Direction$

Wheel torque also needs to be evaluated for the motor to be properly sized for the application. FBD torque inclusion is described in Figure 3.2.2 below.

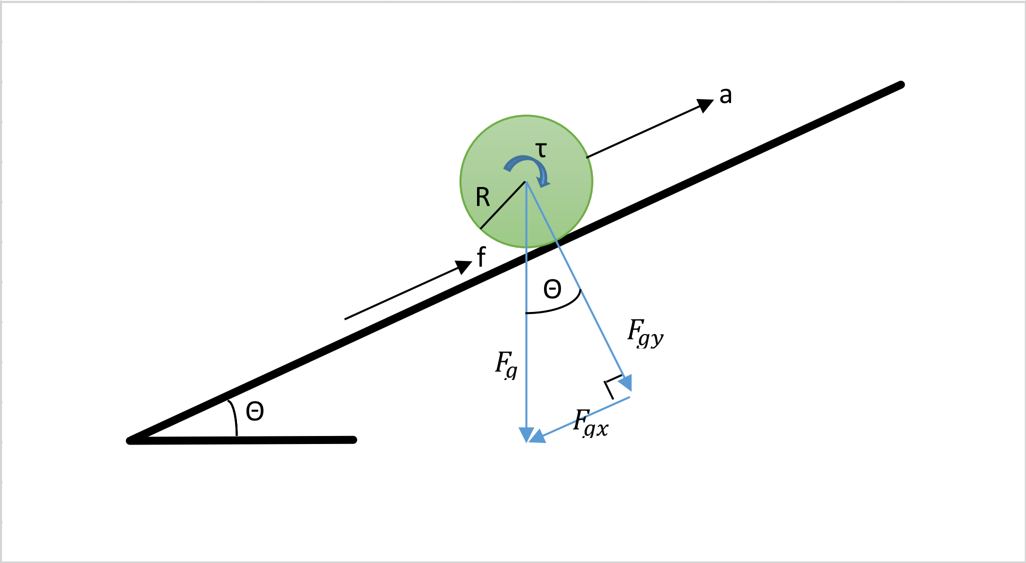

Figure 3.2.2 – Wheel Torque FBD

The torque is related to the force of friction between the wheel and surface and is noted in the FBD as ‘f’. This relation is described in equation 3.2.3.

$$\tau = f * R \qquad 3.2.3$$

Where:

$R = Radius\ of\ the\ wheel$

$f = Frictional\ force$

$\tau = Torque$

To keep the roving vehicle stationary on an incline there must be an acceleration and corresponding force up the incline to compensate for the downward pull of gravity. Using this insight yields the force balance equation 3.2.4.

$$\sum F_x = m * a_x = f - F_x \qquad 3.2.4$$

Where:

$F_x = Force\ Applied\ in\ the\ X - Direction$

$m = The\ mass\ handeled\ by\ one\ wheel$

$a_x = Acceleration\ Applied\ in\ the\ X - Direction$

$f = Frictional\ force$

We can then rearrange equation 3.2.3, solving for f, and substitute f in equation 3.2.4 with $\frac{\tau}{R}$ to determine equation 3.2.5.

$$m * a_x = \frac{\tau}{R} - F_x \qquad 3.2.5$$

Where:

$m = The\ mass\ of\ the\ robot$

$a_x = Acceleration\ Applied\ in\ the\ X - Direction$

$\tau = Torque$

$R = Radius\ of\ the\ wheel$

$F_x = Force\ Applied\ in\ the\ X - Direction$

Solving for torque in equation 3.2.5 yields equation 3.2.6

$$\tau = R * (m * a_x + F_x) \qquad 3.2.6$$

The torque is assumed to be equal among evenly proportioned motors meaning equation 3.2.6 should be divided by the number of wheels utilized in the robot design. This insight produces equation 3.2.7.

$$\tau_w = \frac{R * (m * a_x + F_x)}{N_w} \qquad 3.2.7$$

Where:

$\tau_w = Torque$ Required per wheel

$R = Radius\ of\ the\ wheel$

$m = The\ mass\ handeled\ by\ one\ wheel$

$a_x = Acceleration\ Applied\ in\ the\ X - Direction$

$F_x = Force\ Applied\ in\ the\ X - Direction$

$N_w = Number\ of\ Wheels$

Motor torque is typically described as stall torque which is the point at which the motor will consume it's so called stall current. This is not an ideal scenario under normal operating conditions as permanent damage may result if a torque close to stall is continuous. Another quantity may be added to equation 3.2.7 to ensure our design does not approximate close to the stall current of the motor. Institution of this ideal is done by adding an efficiency multiplicative term to the equation to give rise to equation 3.2.8.

$$\tau_{w-eff} = \frac{100}{e} * \frac{R * (m * a_x + F_x)}{N_w} \qquad 3.2.8$$

Where:

$\tau_{w-eff} = Torque\ Required\ per\ wheel\ based\ on\ efficiency$

$e = Efficientcy\ desired\ as\ a\ percentage$

$R = Radius\ of\ the\ wheel$

$m = The\ mass\ of\ the\ robot$

$a_x = Acceleration\ Applied\ in\ the\ X - Direction$

$F_x = Force\ Applied\ in\ the\ X - Direction$

$N_w = Number\ of\ Wheels$

Additionally, power requirements can be determined for wheel driving motors where the power required per motor is defined in equation 3.2.9 and the associated current required per motor is defined in equation 3.2.10.

$$P_m = \tau * \omega = VI \qquad 3.2.9$$

$$I_m = \frac{\tau\omega}{V} \qquad 3.2.10$$

Where:

$P_m = Power\ required\ per\ motor$

$\tau = Torque\ of\ the\ motor$

$\omega = Angular\ Velocity$

$V = Operating\ Voltage$

$I = Motor\ Current$

Angular (Inertial) Torque (τ_a) should also be evaluated for designs that have a large inertial load. Inertia is defined as the resistance of a mass to move. The angular torque was not evaluated for the drive wheel in this application as the acceleration and moment of inertia are extremely low in relation to the drive torque requirement (Table 3.2.1). Conversely, Inertial Torque for the turning wheel is far greater as the load is not evenly distributed about the axes of rotation.

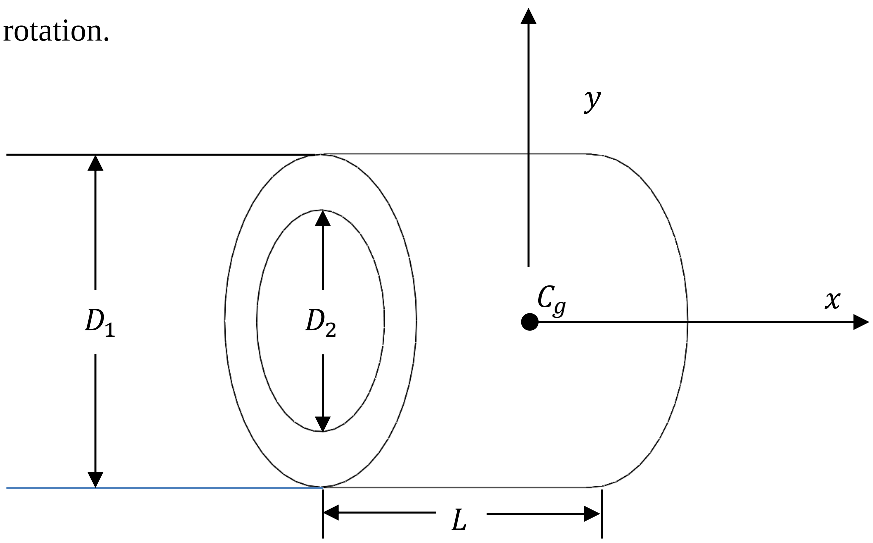

Figure 0.4 – MoI Reference Hallow Cylinder

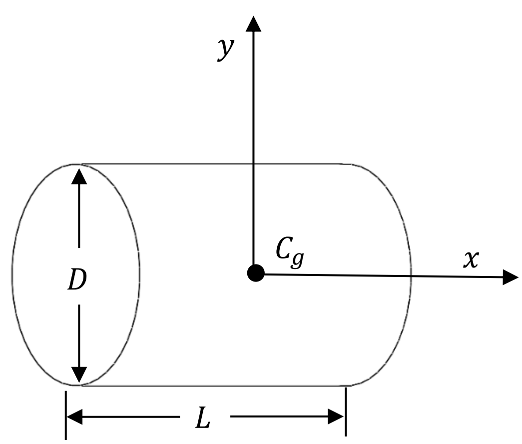

Figure 0.5 – MoI Reference Cylinder

The Moment of Inertial (MoI) for a hallow cylinder about the axis going through the hole (x-axis) is defined in equation (**3.2.1)** while the MoI for the y-axis is given in equation (**3.2.2**. Refer to Figure 0.4 for more detail. These equations (with equation (**3.2.5)**) are used to determine the rotational torque introduced by the wheel as perceived by the two motors.

$$J_x = \frac{1}{8} m({D_1}^2 + {D_2}^2) \tag{3.2.1}$$

$$J_y = \frac{1}{12} m(3{D_1}^2 + 3{D_2}^2 + L^2) \tag{3.2.2}$$

Where:

$J_x = Moment\ of\ Intertia\ about\ the\ x - axis\ of\ a\ hallow\ cylinder$

$J_y = Moment\ of\ Intertia\ about\ the\ y - axis\ of\ a\ hallow\ cylinder$

$m = Mass\ of\ the\ hallow\ cylinder$

$D_1 = Outer\ Diameter$

$D_2 = Inner\ Diameter$

$L = Length\ of\ the\ cylinder$

To model the nacelle (casing for the drive motor) the MoI equation for a hallow cylinder is used while the MoI formula for a Cylinder is used for the motor since it is full of components (i.e. gears, encoder, motor windings, etc.). These equations is given in (**3.2.3)** and (**3.2.4)**.

$$J_x = \frac{1}{2} m r^2 \tag{3.2.3}$$

$$J_y = \frac{1}{12} m(3r^2 + L^2) \tag{3.2.4}$$

Where:

$J_x = Moment\ of\ Intertia\ about\ the\ x - axis\ of\ a\ cylinder$

$J_y = Moment\ of\ Intertia\ about\ the\ y - axis\ of\ a\ cylinder$

$m = Mass\ of\ the\ cylinder$

$r = Radius\ defined\ as\ D/2$

$L = Length\ of\ the\ cylinder$

Angular Torque, defined in equation (**3.2.5)**, is used to find the torque induced by the rotation of an object about an axis (two dimensional equation).

$$\tau_a = J_x * \alpha \tag{3.2.5}$$

Where:

$\tau_a = Inertial\ Torque$

$J_x = Moment\ of\ Intertia\ through\ the\ centerof\ the\ hallow\ cylinder$

$\alpha = Angular\ Acceleration$

Table 3.2.1 – Drive Wheel Moment of Inertia

Parameter	Value	Units
m	104	g
D_1	101.6	mm
D_2	83	mm
$J_{x-calculated}$	223,750	$g\ mm^2$
$J_{x-solidworks}$	184,002	$g\ mm^2$
$\tau_{a-calculated}$	0.0007	Nm

Drive motor selection can be made using the force and torque equations garnered above using input assumptions.

Table 3.2.2 – Drive Wheel Input Parameter Assumptions

Input Parameter	Value	Units
Robot's Mass	10	Kg
# of Motors	4	$--$
Wheel Radius	0.0508	m
Robot velocity	1	m/s
Maximum Incline	45	$Degrees$
Supply Voltage	12	$volts$
Desired Acceleration	0.2	m/s^2
Desired Efficiency	75	%

Table 3.2.3 – Drive Wheel Output Parameters

Output Parameter	Value	Units	Equation
F_g	98.0665	N	$m * 9.80665$
ω	19.6850	rad/sec	$\frac{v_{robot} * 60\frac{s}{m}}{2\pi R}$
F_x	69.3435	N	**Error! Reference source not found.**
F_y	69.3435	N	**Error! Reference source not found.**
$\tau_{required}$	3.6242	$N \cdot m$	**Error! Reference source not found.**
τ_{wheel}	0.9061	$N \cdot m$	**Error! Reference source not found.**
τ_{w-eff}	1.2081	$N \cdot m$	**Error! Reference source not found.**
P_{total}	23.7812	W	**Error! Reference source not found.**
I_{motor}	1.9818	A	**Error! Reference source not found.**

3.3 Motor Turning Requirements

The motor used to turn the wheel and the wheel driving motor can be approximated using the same equations for torque for worst case analysis. The radius will be different as it is the radius between the motor output shaft and where the wheel contacts the ground. This can be seen in the following FBD. This is a top down view of the two motors where the incline angle is not seen. However, this incline is presumed to be 45° which is the same angle used in analyzing the drive motor.

Wheel turning in this configuration will be helped by using the drive wheel to rotate without slipping while a turn is being made. The torque required will be reduce to that of inertia using this methodology by eliminating the “scrubbing torque” [2000_09_Yu,_Dub_Skw.pdf]. The inertial torque is much greater than the previous case which will result in requiring further analysis.

The Parallel Axis Theorem (equation (**3.3.1)**) has to be incorporated into the analysis since the wheel is offset from the turning motor by R_t. Motor and nacelle centers of gravity nearly align with the turning motor’s output shaft axis so as not to necessitate the use of the Parallel Axis Theorem.

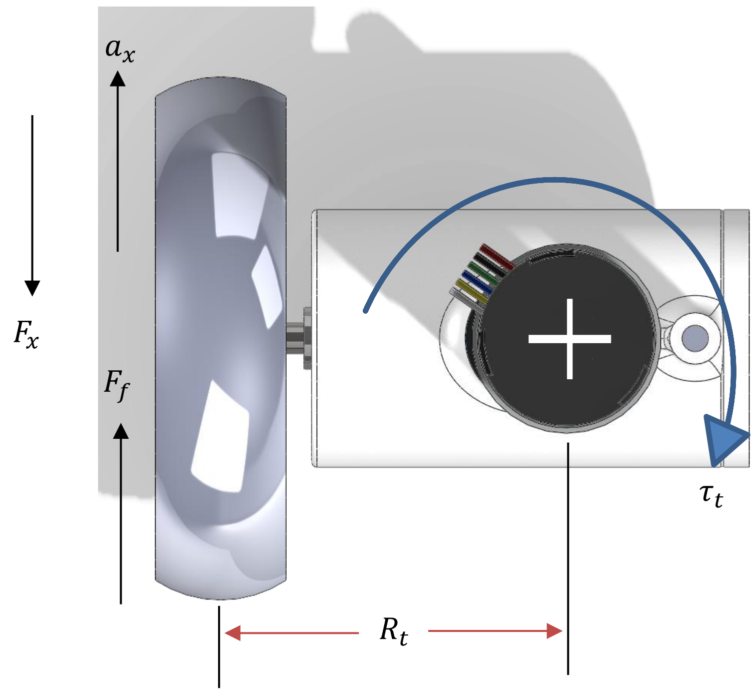

Figure 0.6 – FBD of Turning Motor

$$J_{w-t} = J' + mD^2 \tag{3.3.1}$$

$$J_{turn\_motor\_total} = [J_{y-Wheel} + mR_t^{\ 2}] + J_{y-Nacelle} + J_{y-Motor} \tag{3.3.2}$$

Where:

$J_{w-t} = Inertia\ from\ the\ remote\ parallel\ axis\ to\ the\ turning\ motor$

$J_{turn\_motor\_total} = Inertia\ felt\ by\ the\ turning\ motor$

$J_{y-Wheel} = Inertia\ of\ the\ wheel\ about\ the\ y\ axis\ and\ wheel's\ center\ of\ mass$

$J_{y-Nacelle} = Nacelle\ inertia\ about\ the\ nacelle\ y-axis$

$J_{y-Motor} = Motor\ inertia\ about\ the\ motor\ y-axis$

$J' = Moment\ of\ Intertia\ about\ the\ center\ of\ mass$

$m = Mass\ of\ the\ object$

$D = Distance\ from\ the\ center\ of\ gravity\ to\ the\ center\ of\ remote\ axis$

Table 3.3.1 – Wheel Turning Motor Input Parameter Assumptions

Input Parameter	Value	Units
Robot's Mass	10	Kg
# of Motors	4	$--$
Distance from Wheel	0.063	m
Robot Wheel Velocity	1	m/s
Maximum Incline	45	$Degrees$
Supply Voltage	12	$volts$
Desired Acceleration (a)	0.2	m/s^2
Desired Efficiency	75	%
Mass of Wheel	104	g
Mass of Nacelle	72.3	g
Mass of Motor	225	g
Motor Length	54.7	mm
Nacelle Length	80	mm
Motor Diameter	36.8	mm
Nacelle Outer Diameter	50	mm
Nacelle Inner Diameter	40	mm

Table 3.3.2 – Wheel Turning Motor Output Parameters

Output Parameter	Value	Units	Equation
F_g	98.0665	N	$m * 9.80665$
ω	15.8730	rad/sec	$\frac{v_{robot} * 60\frac{s}{m}}{2\pi R}$
α	3174.6	rad/sec^2	a/r
F_x	69.3435	N	**Error! Reference source not found.**
F_y	69.3435	N	**Error! Reference source not found.**
$\tau_{required}$	4.4946 (636.49)	$N \cdot m$ (oz·in)	**Error! Reference source not found.**
τ_{wheel}	1.1237 (159.13)	$N \cdot m$ (oz·in)	**Error! Reference source not found.**
τ_{w-eff}	1.4982 (212.16)	$N \cdot m$ (oz·in)	**Error! Reference source not found.**
P_{total}	23.7812	W	**Error! Reference source not found.**
I_{motor}	1.9818	A	**Error! Reference source not found.**
$J_{y-Wheel}$	112,315.41	$g \cdot mm^2$	**(3.2.2)**
$J_{y-Nacelle}$	112,667.50	$g \cdot mm^2$	**(3.2.2)**
$J_{y-Motor}$	118,491.75	$g \cdot mm^2$	**(3.2.4)**
$J_{turn\_motor\_total}$	2.35 (332)	N·m (oz·in)	**(3.3.2)**

3.4 SELECTED MOTOR SPECIFICATIONS

The motor selected for the drive motor is the Pololu 50:1 Metal Gear motor with 64 CPR Encoder. It has the specifications as defined in Table 3.4.1 – Selected Wheel Drive Motor Specifications.

Table 3.4.1 – Selected Wheel Drive Motor Specifications

Input Parameter	Value	Units
Size	37D x 70L	mm
Weight	225	g
Shaft Diameter	6	mm
Gear Ratio	50:1	$--$
Free-run speed	200	rpm
Free-run current	300	mA
Stall current	5000	mA
Stall Torque	170	oz·in

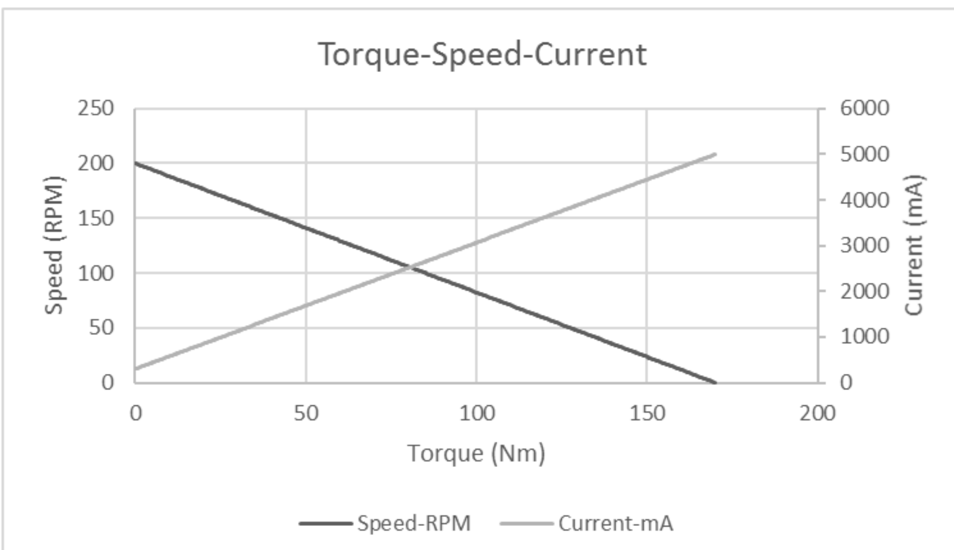

Figure 0.7 – 50:1 Metal Gear motor Torque-Speed-Current (Drive Motor)

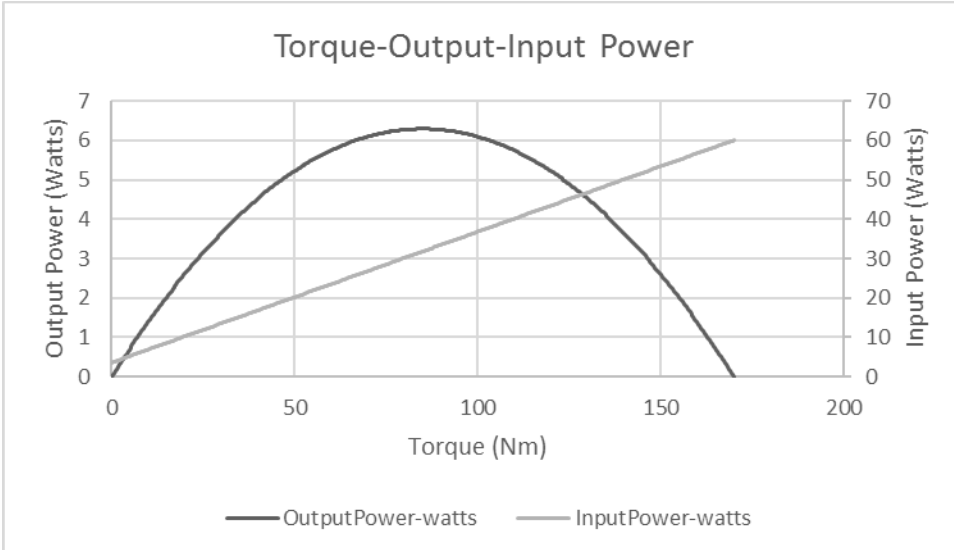

Figure 0.8 – 50:1 Metal Gear motor Torque-Output-Input Power (Drive Motor)

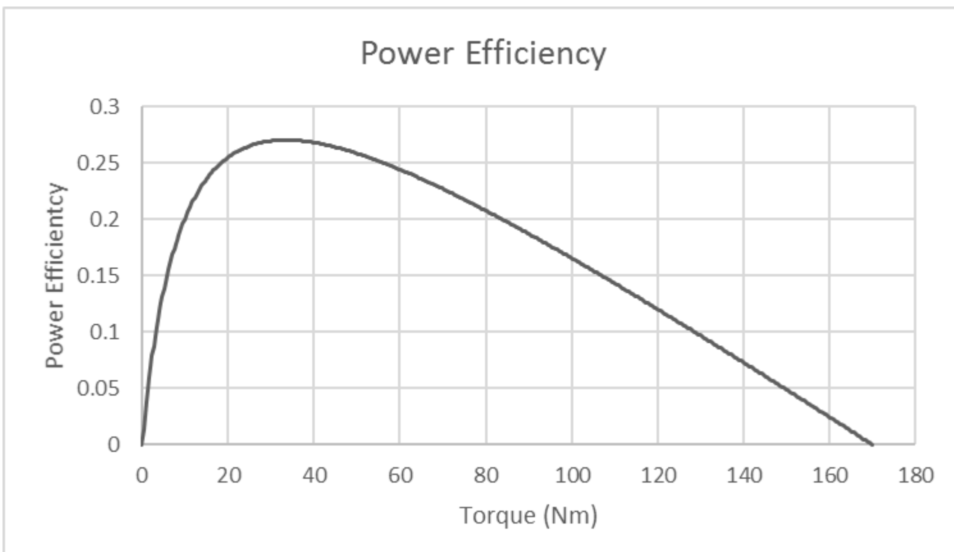

Figure 0.9 – 50:1 Metal Gear motor Power Efficiency (Drive Motor)

The motor selected for the wheel turning motor is the Pololu 131:1 Metal Gear motor with 64 CPR Encoder. It has the specifications as defined in Table 3.4.2 – Selected Wheel Turn Motor Specifications

Table 3.4.2 – Selected Wheel Turn Motor Specifications

Input Parameter	Value	Units
Size	37D x 72.5L	mm
Weight	235	g
Shaft Diameter	6	mm
Gear Ratio	131:1	$--$
Free-run speed	80	rpm
Free-run current	300	mA
Stall current	5000	mA
Stall Torque	250	oz·in

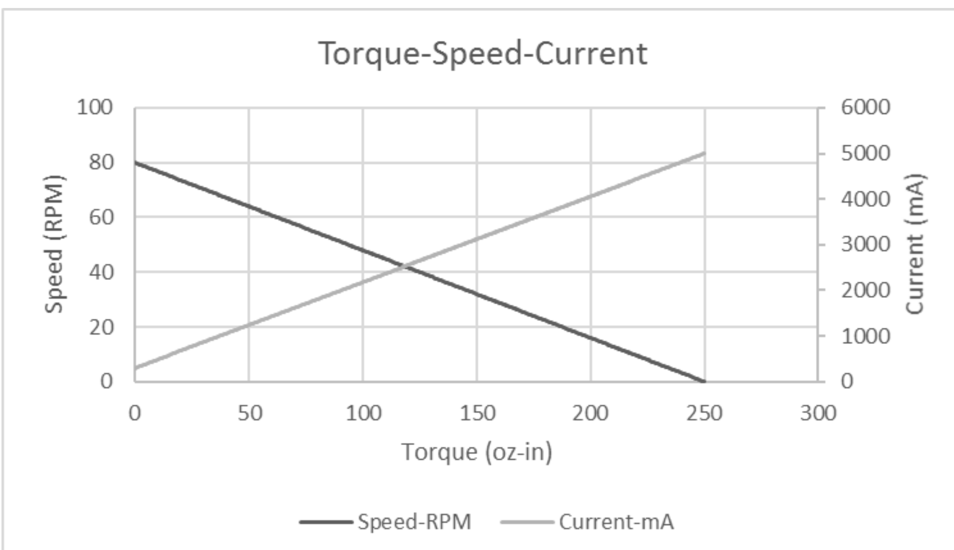

Figure 0.10 – 131:1 Metal Gear motor Torque-Speed-Current (Turn Motor)

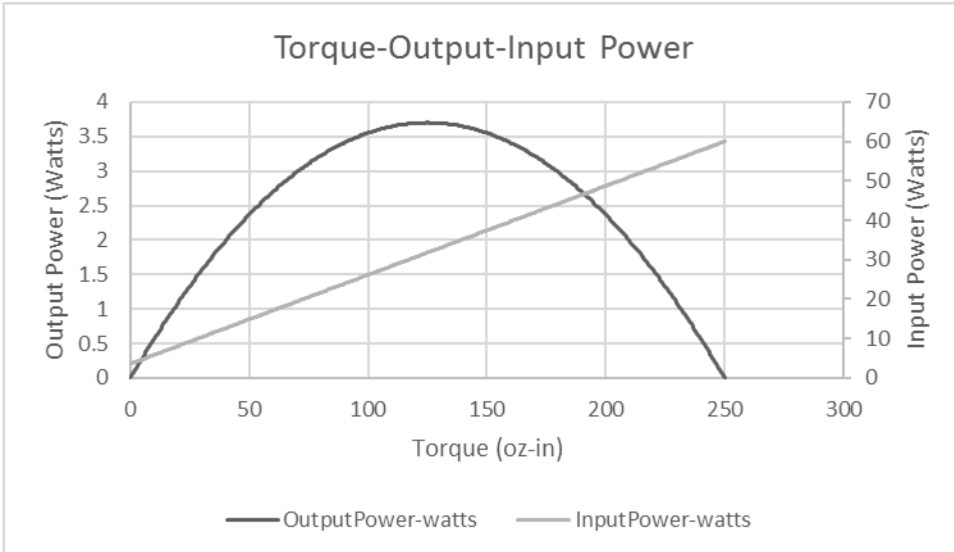

Figure 0.11 – 131:1 Metal Gear motor Torque-Output-Input Power (Turn Motor)

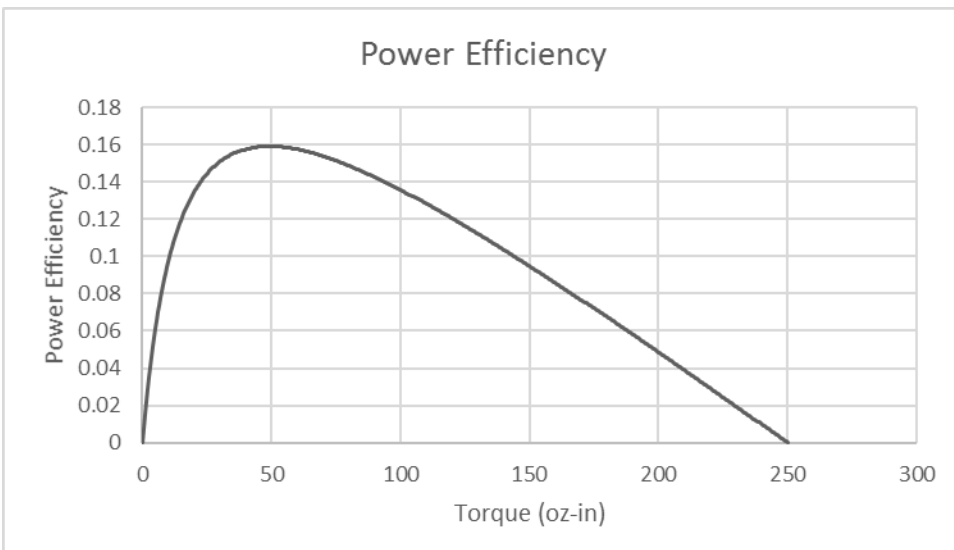

Figure 0.12 – 131:1 Metal Gear motor Power Efficiency (Turn Motor)