

Holonomic Turtle Robot



Kevin J. Walchko, PhD

Code: <https://github.com/walchko/soccer2>
Email: kevin.walchko@outlook.com

December 14, 2014

Abstract

This paper describes the development of a holonomic robot using omni directional wheels. Both a development of the kinematic and dynamical equations of motion are derived and used as a foundation for gaining further insight into the capabilities of the robot. A discussion of the vision system used to detect objects and obstacles during navigation.

1 Variables

Parameter	Units	Description
R	m	robot radius
M	kg	mass of robot
I	$kg \cdot m^2$	inertia of robot
J_w	$kg \cdot m^2$	inertia of wheel
v_i	m/sec	velocity in the i^{th} direction in either frame W or B
$\dot{\theta}_i$	rad/sec	i^{th} wheel speed
r_w	m	wheel radius
r_m	m	offset distance between frames B' and B
τ_i	N·m	wheel torque
F_i	N	traction force (vector)
f_i	N	magnitude of traction force ($\ F_i\ $)
ϕ_i	rads	wheel location relative to B
ψ	rads	angle of B with respect to W

2 Introduction

Robots come in a variety of types and configurations: wheeled, tracked, legs, flying, etc. Common wheeled robots typically have two wheels (directly driven) with a caster wheel to make the robot stable. There are some without the caster wheel and employ a control system to keep them upright (inverted pendulum problem) and resemble a Segway scooter. All of these two wheeled robot are non-holonomic systems.

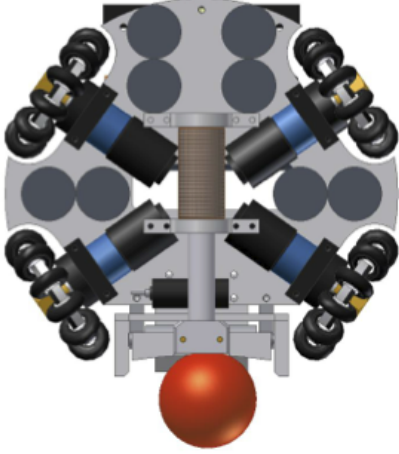


Figure 1: Holonomic soccer robot using 4 omni directional wheels and a kicking motor used to hit the red ball into a goal. [1].



Figure 2: Omni directional wheel allows movement in any direction.

Definition 1 *A non-holonomic system in physics and mathematics is a system whose state depends on the path taken to achieve it. An automobile is an example of a non-holonomic vehicle. The vehicle has three degrees of freedom: its position in two axes, and its orientation relative to a fixed heading. Yet it has only two controllable degrees of freedom: acceleration/braking and the angle of the steering wheel with which to control its position and orientation. [9]*

Due to these constraints, a holonomic robot (**Figure 1**) which could travel in any direction and immediately change its position and orientation is much more desirable. There are a variety of different wheels which make this type of robot possible such as mecanum or omni wheels (**Figure 2**).

Omni wheels operate like standard wheels in that the force is produced normal to the motor's axis of rotation and as a result of friction. However, there are a series of smaller wheels which ring the main wheel and allow the wheel to slip in the direction of the motor rotational axis. Note that no force is produced parallel to the motor axis, just slippage.

3 Holonomic Dynamics

The dynamics for a holonomic robot, such as **Figure 3**, with 4 omni directional wheels (can be derived using Euler-Largrange (\mathcal{L}) which defines a system's kinetic (T) and potential (V) energies in relation to a set of generalized coordinates (q) and generalized forces (Q):

$$\mathcal{L} = T - V \quad (1)$$

$$\frac{d}{dt} \left\{ \frac{\partial \mathcal{L}}{\partial \dot{q}} \right\} - \frac{\partial \mathcal{L}}{\partial q} = Q \quad (2)$$

$$T = \frac{1}{2} M v_w^2 + \frac{1}{2} J \dot{\psi}^2 + \frac{1}{2} J_w (\dot{\theta}_1^2 + \dot{\theta}_2^2 + \dot{\theta}_3^2 + \dot{\theta}_4^2) \quad (3)$$

$$V = 0 \quad (4)$$

However, the dynamics must be calculated from an inertial reference frame (W) and take into account the rotating body frame dynamics (B'). Now, assume the body frame is offset from the center of mass (CM) by

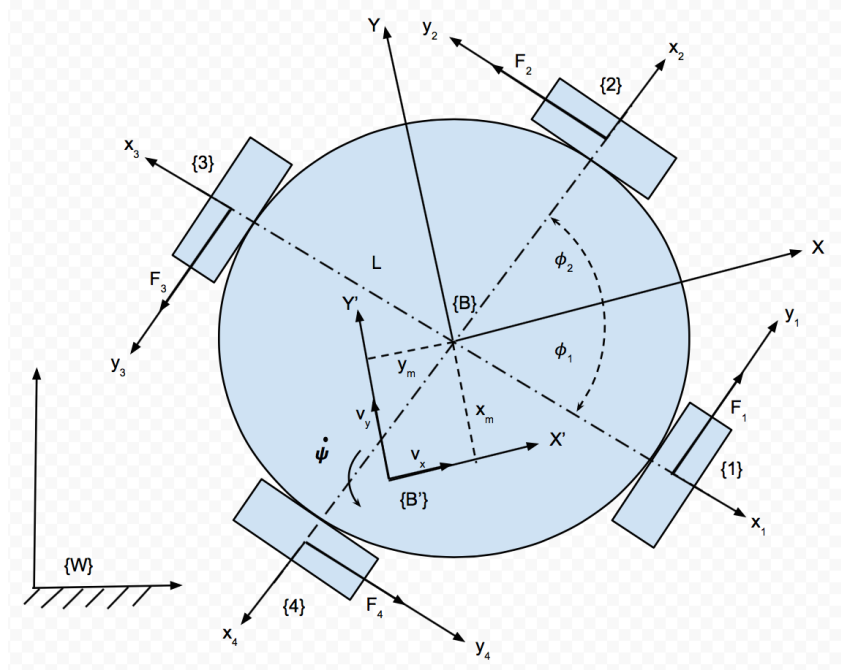


Figure 3: Coordinate system tied to the body of the robot with the origin located at the center of mass. Note that the x-axis points straight up and the y-axis points to the right. Also, the motor angle ϕ is defined as the angle measured from the y-axis. The forces (F) are the results of the motors spinning in the positive direction according to the right hand rule. Note also that no force is produced parallel to the motor's axis of rotation.

x_m and y_m which compose a vector r_m . Thus the velocity of the robot in the rotating frame would be:

$$v_w = v_{B'} + \dot{\psi} \times r_m \quad (5)$$

$$v_w = v_{B'} + [0 \ 0 \ \dot{\psi}]^T \times [x_m \ y_m \ 0]^T = [\dot{x} \ \dot{y} \ 0]^T + [-y_m \dot{\psi} \ x_m \dot{\psi} \ 0]^T \quad (6)$$

$$v_{B'} = [\dot{x} \ \dot{y} \ 0]^T \quad (7)$$

where $v_{B'}$ is the speed of the body frame. Now substituting that into the above kinetic energy equation T , we get:

$$T = \frac{1}{2}M((\dot{x} - \dot{\psi}y)^2 + (\dot{y} + \dot{\psi}x)^2) + \dots \quad (8)$$

$$T = \frac{1}{2}M(\dot{x}^2 - 2\dot{\psi}y_m\dot{x} + \dot{\psi}^2y_m^2 + \dot{y}^2 + 2\dot{\psi}x_m\dot{y} + \dot{\psi}^2x_m^2) + \frac{1}{2}J\dot{\psi}^2 + \frac{1}{2}J_w(\dot{\theta}_1^2 + \dot{\theta}_2^2 + \dot{\theta}_3^2 + \dot{\theta}_4^2) \quad (9)$$

$$\frac{d}{dt} \left\{ \frac{\partial \mathcal{L}}{\partial \dot{x}} \right\} = M(\ddot{x} - \ddot{\psi}y - \dot{\psi}\dot{y}) \quad \frac{\partial \mathcal{L}}{\partial x} = M(\dot{\psi}\dot{y} + \dot{\psi}^2x) \quad (10)$$

$$\frac{d}{dt} \left\{ \frac{\partial \mathcal{L}}{\partial \dot{y}} \right\} = M(\ddot{y} + \ddot{\psi}x + \dot{\psi}\dot{x}) \quad \frac{\partial \mathcal{L}}{\partial y} = M(-\dot{\psi}\dot{x} + \dot{\psi}^2y) \quad (11)$$

$$\frac{d}{dt} \left\{ \frac{\partial \mathcal{L}}{\partial \dot{\psi}} \right\} = J\ddot{\psi} \quad \frac{\partial \mathcal{L}}{\partial \phi} = 0 \quad (12)$$

$$\frac{d}{dt} \left\{ \frac{\partial \mathcal{L}}{\partial \dot{\theta}} \right\} = J_w \sum_{i=1}^4 \ddot{\theta}_i \quad \frac{\partial \mathcal{L}}{\partial \theta} = 0 \quad (13)$$

Now we make the following assumptions: B^i is coincident with B , $x_m = 0$, $y_m = 0$, $\dot{x} = v_x$, $\dot{y} = v_y$

$$F_x = M(\ddot{x} - 2\dot{\psi}\dot{y}) \quad (14)$$

$$F_y = M(\ddot{y} + 2\dot{\psi}\dot{x}) \quad (15)$$

$$T = J\ddot{\psi} \quad (16)$$

$$\tau_w = J_w\ddot{\theta}_1 \quad \tau_w = J_w\ddot{\theta}_2 \quad \tau_w = J_w\ddot{\theta}_3 \quad \tau_w = J_w\ddot{\theta}_4 \quad (17)$$

$$\begin{bmatrix} F_x \\ F_y \\ T \end{bmatrix} = \begin{bmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & J \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & -2M\dot{\psi} & 0 \\ 2M\dot{\psi} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \mathcal{M}\ddot{X} + \mathcal{O}\dot{X} = Q \quad (18)$$

3.0.1 World Coordinates

Now the dynamics derived so far are all in the body frame and we could stop here and develop a controller which performs velocity control. However, position control is more useful and a transform needs to be performed to move the velocities and accelerations into the world frame.

$$\dot{X}^W = R_B^W \dot{X}^B \quad (19)$$

$$R_B^W = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (20)$$

$$\ddot{X}^W = \dot{R}_B^W \dot{X}^B + R_B^W \ddot{X}^B \quad (21)$$

$$\dot{R}_B^W = \begin{bmatrix} \sin \psi & -\cos \psi & 0 \\ \cos \psi & \sin \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (22)$$

Now, substituting this into the dynamics, gives dynamics in the world coordinate system of:

$$F = \mathcal{M}(\dot{R}\dot{X} + R\ddot{X}) + \mathcal{O}R\dot{X} \quad (23)$$

$$F = \mathcal{M}R\ddot{X} + (\mathcal{M}\dot{R} + \mathcal{O}R)\dot{X} \quad (24)$$

3.0.2 External Forces and Torques

Now looking at figure 4 and summing the forces into their body referenced x and y directions and the torque about the z axis, gives us:

$$\sum F_x = f_1 \sin(\phi) - f_2 \sin(\phi) - f_3 \sin(\phi) + f_4 \sin(\phi) \quad (25)$$

$$\sum F_y = f_1 \cos(\phi) + f_2 \cos(\phi) - f_3 \cos(\phi) - f_4 \cos(\phi) \quad (26)$$

$$\sum T = L(f_1 + f_2 + f_3 + f_4) \quad (27)$$

Additionally, we can simplify this by assuming all of the angles are the same (e.g., $\phi_1 = \phi_2 = \phi_3 = \phi_4$) and can now put this into a matrix form:

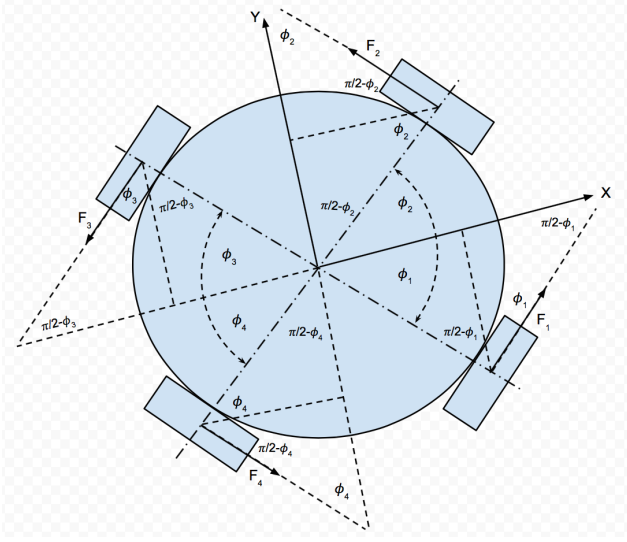


Figure 4: Configuration of three groups of motors where ϕ is 30, 45, and 60 degrees.

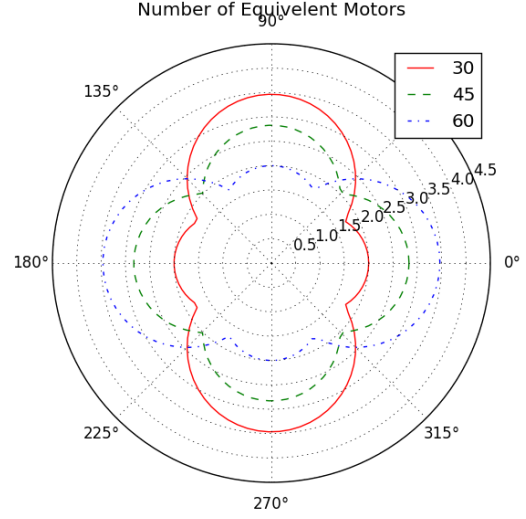


Figure 5: Number of equivalent motors for any direction under linear movement only, no rotational movement allowed.

$$\begin{bmatrix} F_x \\ F_y \\ T \end{bmatrix} = \begin{bmatrix} \sin(\phi) & 0 & 0 \\ 0 & \cos(\phi) & 0 \\ 0 & 0 & L \end{bmatrix} \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \quad (28)$$

where ϕ is again the angle of the motors as defined in **Figure 4**, f_i is the magnitude of the force produced by the motors, and L is the radius of the robot.

where $\text{pinv}()$ ¹ is defined as the pseudoinverse since $A(\phi)$ is not a square matrix. Finally, substituting these into the original equation, we can calculate the torques given the desired accelerations.

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \frac{Mr_w}{4} \begin{bmatrix} -1 & 1 & 1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sin(\phi)} & 0 & 0 \\ 0 & \frac{1}{\cos(\phi)} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ R\dot{\omega} \end{bmatrix} \quad (29)$$

Now looking at this equation, we notice that ϕ can not be equal to 0, 90, 180, 270, or 360 otherwise we get a singularity in the $A(\phi)$ matrix. This however is not an issue in the real world, since the motors would occupy the same physical space and the robot would essentially only have 2 and not 4 motors.

4 Holonomic Robot Kinematics

Now performing a similar exercise for what was done with the dynamics, looking at **Figure 3**, the velocity of motor 1 is given by $v_1 = -\sin(\phi)v_x + \cos(\phi)v_y + R\omega$. Performing this for each wheel gives:

¹Pseudoinverse: for $m > n$: $A_{left}^{-1} = (A^T A)^{-1} A^T$ or $m < n$: $A_{right}^{-1} = A^T (A A^T)^{-1}$ such that $A A^{-1} = I$ or $A^{-1} A = I$

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} -\sin(\phi) & \cos(\phi) & L \\ -\sin(\phi) & -\cos(\phi) & L \\ \sin(\phi) & -\cos(\phi) & L \\ \sin(\phi) & \cos(\phi) & L \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \sin(\phi) & 0 & 0 \\ 0 & \cos(\phi) & 0 \\ 0 & 0 & L \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \quad (30)$$

Now setting ω to zero and calculating only linear movement, we can determine the number of equivalent motors as shown in **Figure 5**. For example, setting ϕ to 30° (the red line in **Figure 5**) and traveling in the x direction only ($[v_x \ v_y \ \omega]^T = [1 \ 0 \ 0]^T$), the above equation simplifies to $4 \sin(30)$ or 2 equivalent motors. Repeating for the y direction results in $4 \cos(30)$ or 3.46 equivalent motors.

Now it is interesting to note that when ϕ is set to 30° , the robot has more equivalent motors when going forward or backwards, while a ϕ of 60° provides more equivalent motors moving left or right. When the motors are angled at 45° , movement is clearly equally optimized for both forward/backwards and left/right ($2 \sin(45)$ is 2.83 motors) movement.

Figure 5 tells us that no matter how the 4 motors are oriented in a realistic configuration, the robot will never have the equivalent use of all 4 motors. Movement in one direction or another can be optimized, but then a sacrifice is made in another direction. This fact is intuitively obvious.

Another issue is these results are also ideal. This logic assumes that the wheels will not slip and have good traction in any orientation. Unfortunately real world results do not mimic this situation and the robot's performance will be reduced.

5 Control

Looking at the state space equations, the system is controllable but it is not observable. Using an IMU (accelerometer, gyro, and magnetometer), the heading (θ) can be determined from the magnetometer and the angular rate (ω) can be determined from the gyro. An observer must be used to estimate the position and velocity of the robot.

Typically encoders attached to the wheels (under the assumption of no slip) would be used to estimate velocity and position. However, with omni wheels, this is not possible since they rely on slippage in order to achieve holonomic motion. Wheel encoders can be useful for detecting excessive amounts of wheel slippage [2] in order to optimize movement or detect failed motors.

6 Guidance and Navigation

In order to have the robot go from one location to another, the position and velocity must be estimated. A Kalman filter using the dynamic equations above will provide this solution. The general form of the Kalman filter can be found in any text book on estimation [11] and have the form:

$$x = ax + bu \quad (31)$$

$$\tilde{x} = \hat{x} - x \quad (32)$$

where the error (\tilde{x}) is the difference between the estimated state (\hat{x}) and the true state (x).

7 Results

8 References

References

- [1] Radu.
- [2] Radu.
- [3] R. Balakrishna, Ashitava Ghosal, "Modeling of Slip for Wheeled Mobile Robots," IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 11, NO. 1, FEBRUARY 1995, pp. 126-132
- [4] J. Agullo, S. Cardona, and J. Vivancos, Kinematics of vehicles with directional sliding wheels, Mechanisms and Machine Theory, vol. 22, no. 4, pp. 295-301, 1987.
- [5] <http://www.ros.org>
- [6] http://en.wikipedia.org/wiki/HSL_and_HSV
- [7] <http://pointclouds.org>
- [8] <http://opencv.willowgarage.com>
- [9] http://en.wikipedia.org/wiki/Non-holonomic_system
- [10] http://en.wikipedia.org/wiki/Lagrangian_mechanics
- [11] kalman filter
- [12] Xiang Li, Andreas Zell, "Motion Control of an Omnidirectional Mobile Robot," ??

A Full Derivation of EOM

A.1 EOM

hi how are you

B Robot Hardware

B.0.1 MiniIMU

The mini inertial measurement unit (IMU) from <http://www.pololu.com> provides acceleration, rotation rates, and magnetic field sensing along all 3 axes. The performance specs are shown below:

table

C Robot Software

The robot software is primarily written in Python and located at <https://github.com/walchko>. The architecture is shown in [?], where multiple distributed processes communicate using a publish subscribe architecture leveraging software from <http://zeromq.org>.

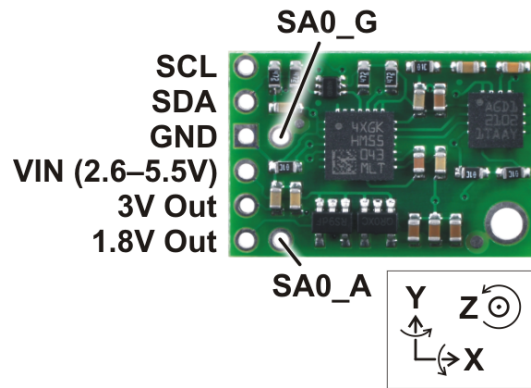


Figure 6: Pololu MiniIMU which provides 3 DoF for acceleration, rotation rate, and magnetic field.

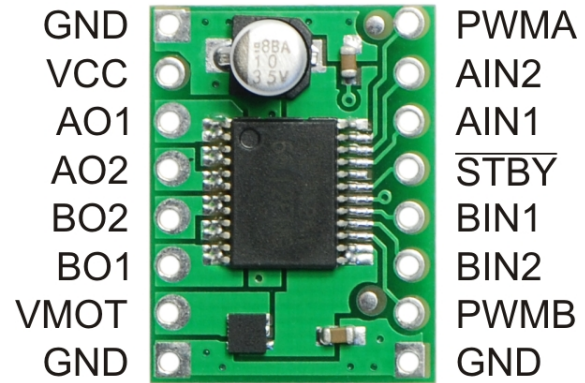


Figure 7: Two Pololu dual TNG6612 motor driver boards control four motors with omni wheels.

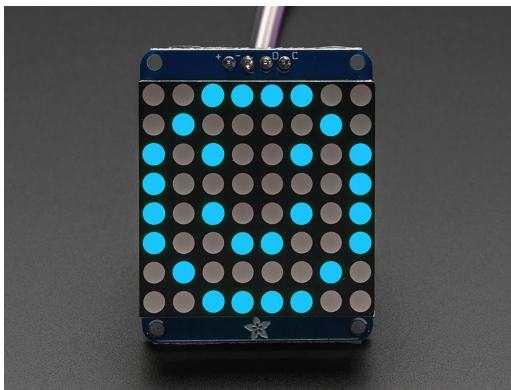


Figure 8: Adafruit LED matrix array.



Figure 9: Logitech c270 USB camera with microphone.

C.0.2 OpenCV

The vision system uses the Open Computer Vision (OpenCV [8]).

The BGR (blue, green, red) image is transformed to the HSV (hue, saturation, and value) system to help account for variable lighting conditions.

Hue The "attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors: red, yellow, green, and blue, or to a combination of two of them"

Saturation The "colorfulness of a stimulus relative to its own brightness"

Value, Lightness The "brightness relative to the brightness of a similarly illuminated white"

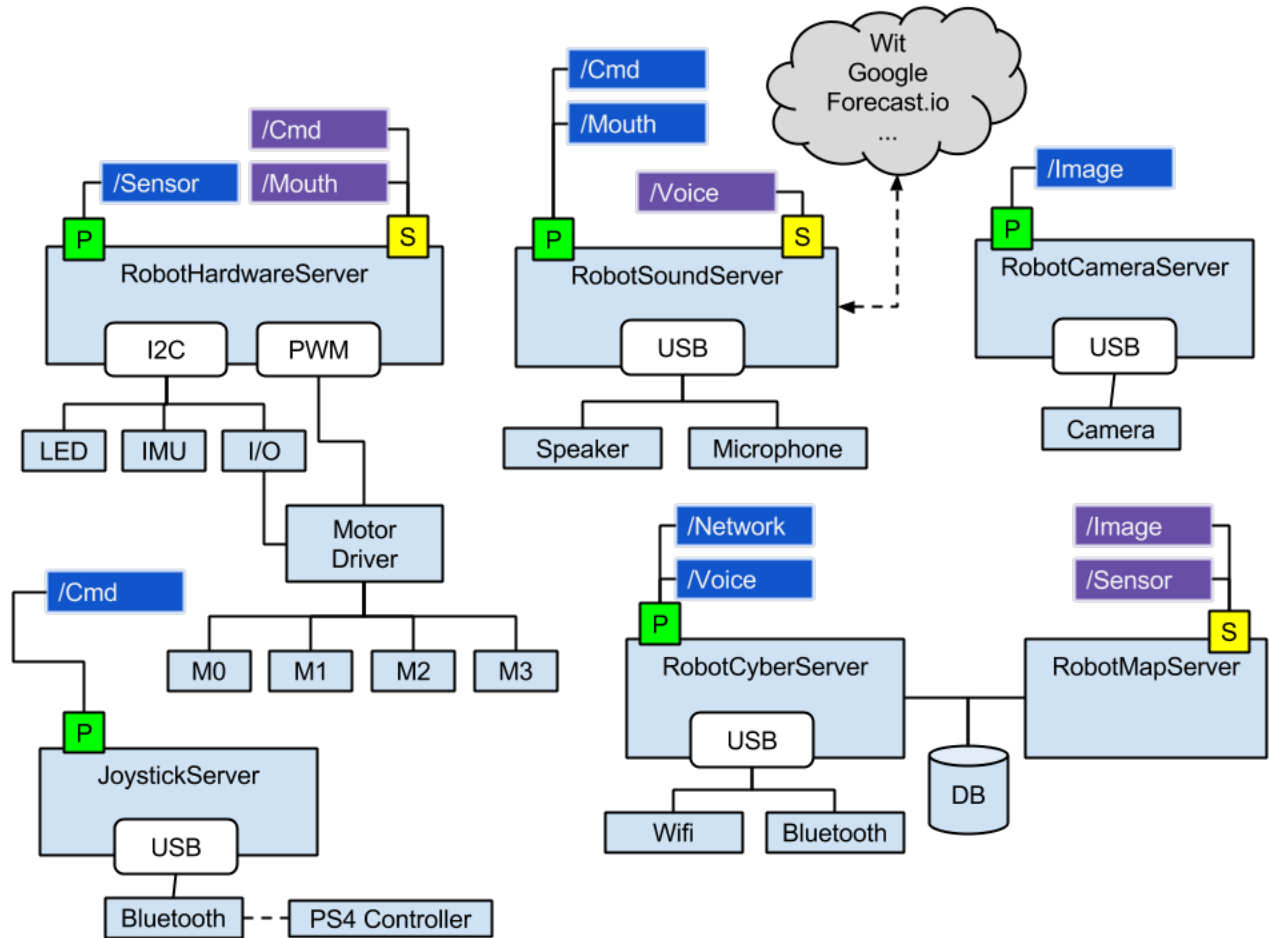


Figure 10: The software architecture composed of publish/subscribe nodes allow all of the software to run either locally on the robot or distributed across different machines on the network.

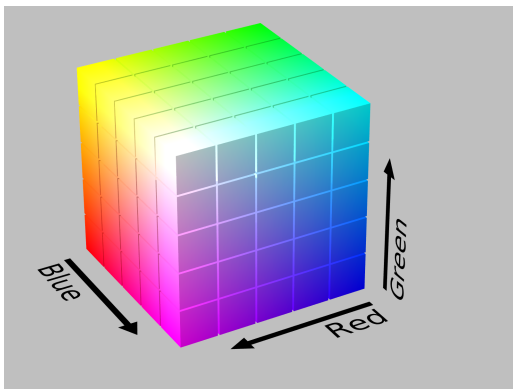


Figure 11: RGB color cube.

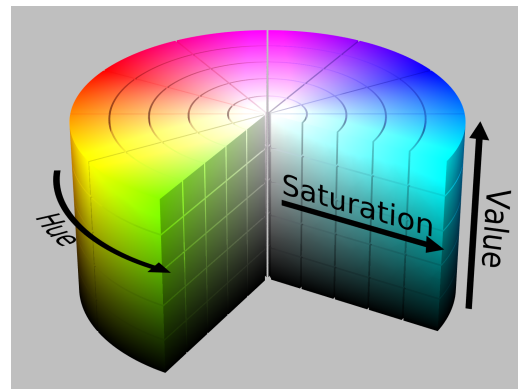


Figure 12: HSV color cylinder.