Holonomic Soccer Robot

Kevin J. Walchko, PhD

Code: https://github.com/walchko/soccer
Email: walchko@verizon.net

September 1, 2012

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
I_w	kg·m^2	inertia of wheel
a_i	m/sec^2	acceleration in the i^{th} direction
v_i	m/sec	velocity in the i^{th} direction
ω_i	rad/sec	i^{th} wheel speed
r_w	m	wheel radius
τ_i	N·m	wheel torque
F_i	N	traction force (vector)
f_i	N	magnitude of traction force ($\|F_i\|$)

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.

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 freedomits position in two axes, and its orientation relative to a fixed heading. Yet it has only two controllable degrees of freedomacceleration/braking and the angle of the steering wheelwith which to control its position and orientation. [9]*

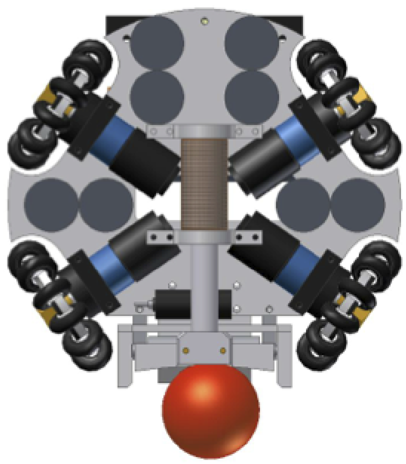


Figure 1: Holonomic soccer robot using 4 omni directional wheels and a kicking motor used to hit the red ball into a goal. [1].

Figure 2: Omni directional wheel used on the soccer robot which allows movement in any direction.

Due to these constraints, a holonomic robot (**Figure 4**) 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 (no slip assumption). 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 4**, with 4 omni directional wheels (see **Figure 2**, can be derived using Euler-Largrange ($\mathcal{L}$) which defines a system's kinectic (T) and potential (V) energies in relation to a set of generalized coordinates (q) and generalized forces (Q):

$$\mathcal{L} = T - V \tag{1}$$

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

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

$$V = 0 \tag{4}$$

$$F_x = M\ddot{x} \qquad F_y = M\ddot{y} \tag{5}$$

$$\tau = J_R\dot{\psi} \tag{6}$$

$$\tau_w = J_w\ddot{\theta}_1 \qquad \tau_w = J_w\ddot{\theta}_2 \qquad \tau_w = J_w\ddot{\theta}_3 \qquad \tau_w = J_w\ddot{\theta}_4 \tag{7}$$

$$\tag{8}$$

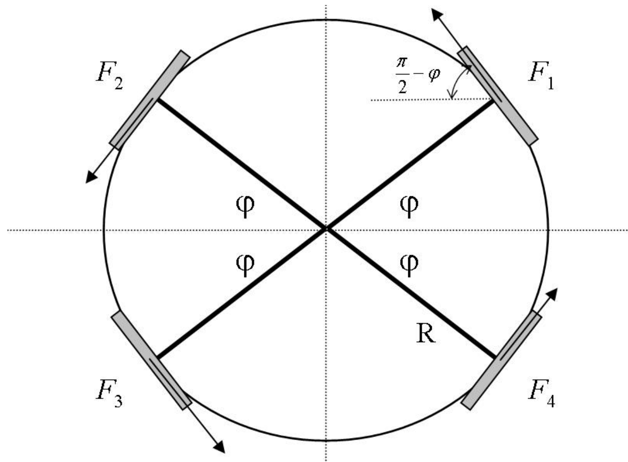


Figure 3: Coordinate system tied to the body of the robot with the origin located at the center of mass. Note that the x-axis points straight up and the y-axis points to the right. Also, the motor angle ϕ is defined as the angle measured from the y-axis. The forces (F) are the results of the motors spinning 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.

where the generalized forces (Q) are defined as:

$$\begin{aligned} Q &= F \\ &= 3 \end{aligned}$$

$$a = \sum_{i=0}^{4} \frac{F_i}{M_i} = \frac{1}{M}(F_1 + F_2 + F_3 + F_4) \tag{9}$$

$$\dot{\omega} = \sum_{i=0}^{4} \frac{\tau_i}{J_R} = \frac{R}{J_R}(f_i + f_2 + f_3 + f_4) where \tau_i = Rf_i \tag{10}$$

Now looking at figure 3, and breaking equation 9 up into its x and y components

$$a_x = \frac{1}{M}(-f_1 \sin(\phi) - f_2 \sin(\phi) + f_3 \sin(\phi) + f_4 \sin(\phi)) \tag{11}$$

$$a_y = \frac{1}{M}(f_1 \cos(\phi) - f_2 \cos(\phi) - f_3 \cos(\phi) + f_4 \cos(\phi)) \tag{12}$$

$$\dot{\omega} = \frac{R}{J_R}(f_1 + f_2 + f_3 + f_4) \tag{13}$$

Now combining these into a matirx.

$$\begin{bmatrix} a_x \\ a_y \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{1}{M} & 0 & 0 \\ 0 & \frac{1}{M} & 0 \\ 0 & 0 & \frac{1}{J_R} \end{bmatrix} \begin{bmatrix} \sin(\phi) & 0 & 0 \\ 0 & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \tag{14}$$

$$\begin{bmatrix} a_x \\ a_y \\ \dot{\omega} \end{bmatrix} = \beta\,[\phi]\,\gamma\,[f_i] \tag{15}$$

where ϕ is the angle of the motors as defined in **Figure 3**, f_i is the magnitude of the force produced by the motors, and R is the radius of the robot. This can be simplified into the following:

$$P = \frac{1}{M} A(\phi) C F \tag{16}$$

where P is the vector of accelerations, $A(\phi)$ is the matrix which defines the orientation of the motors, C defines the coupling of the motor forces, and F is the vector of forces produced by the motors. The individual motor forces can further be expressed as:

$$f_i = \frac{\tau_i}{r_w} \tag{17}$$

$$F = \frac{1}{r_w} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \frac{1}{r_w} T \tag{18}$$

where τ_i is the torque produced by the i^{th} motor and r_w is the radius of the wheel. Here we are assuming that all wheels have the same radius r_w.

$$P = \frac{1}{M r_w} A(\phi) C T \tag{19}$$

For a given motor orientation (e.g., $\phi = 45°$) this equations provides the acceleration states as a function of motor velocity. However, the reverse would be more useful. Meaning, given a desired set of states, what are the motor velocities required?

$$T = M r_w pinv(C) A(\phi)^{-1} P \tag{20}$$

where

$$A(\phi)^{-1} = \begin{bmatrix} \frac{1}{\sin(\phi)} & 0 & 0 \\ 0 & \frac{1}{\cos(\phi)} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \tag{21}$$

$$pinv(C) = \frac{1}{4} \begin{bmatrix} -1 & 1 & 1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{22}$$

where $pinv()$[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.

$$\begin{bmatrix}\tau_1\\ \tau_2\\ \tau_3\\ \tau_4\end{bmatrix} = \frac{Mr_w}{4}\begin{bmatrix}-1 & 1 & 1\\ -1 & -1 & 1\\ 1 & -1 & 1\\ 1 & 1 & 1\end{bmatrix}\begin{bmatrix}\frac{1}{\sin(\phi)} & 0 & 0\\ 0 & \frac{1}{\cos(\phi)} & 0\\ 0 & 0 & \frac{1}{2}\end{bmatrix}\begin{bmatrix}a_x\\ a_y\\ R\dot{\omega}\end{bmatrix} \tag{23}$$

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.

3.1 State Space

Now rearranging equations (11), (12), and (13) into a state space representation where the state vector is $X = \begin{bmatrix}x & y & R\theta & v_x & v_y & R\omega\end{bmatrix}^T$ and its derivative is $\dot{X} = \begin{bmatrix}v_x & v_y & R\omega & a_x & a_y & R\dot{\omega}\end{bmatrix}^T$, the standard formulation is:

$$\dot{X} = AX + Bu \tag{24}$$

$$Y = CX \tag{25}$$

where A is the state transition matrix, B is the input matrix, u is the control input to the system, C is the output matrix which determines which states are visible, and Y is the output vector which is the observable states. Again, the nice thing about formulating the state vector X this way is that all of the units for position are in meters and all of the velocity units are in m/sec.

$$A = \begin{bmatrix}0_{3x3} & I_{3x3}\\ 0_{3x3} & 0_{3x3}\end{bmatrix} \tag{26}$$

$$B = \frac{1}{Mr_w}\begin{bmatrix} & 0_{3x4} & & \\ -\sin(\phi) & -\sin(\phi) & \sin(\phi) & \sin(\phi)\\ \cos(\phi) & -\cos(\phi) & -\cos(\phi) & \cos(\phi)\\ 2 & 2 & 2 & 2\end{bmatrix} \tag{27}$$

$$u = \begin{bmatrix}\tau_1 & \tau_2 & \tau_3 & \tau_4\end{bmatrix}^T \tag{28}$$

$$C = \begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix} \tag{29}$$

The only directly measurable states come from the compass (θ) and the gyro (ω) on the IMU. Position and velocity are not directly measured. Acceleration is also measurable from the IMU and can be used to estimate the velocity and position.

$$\dot{M}X = AX + Bu \tag{30}$$

$$\begin{bmatrix}v_x\\ v_y\\ R\omega\\ a_x\\ a_y\\ R\dot{\omega}\end{bmatrix} = \begin{bmatrix}0_{3x3} & I_{3x3}\\ 0_{3x3} & 0_{3x3}\end{bmatrix}\begin{bmatrix}x\\ y\\ R\theta\\ v_x\\ v_y\\ R\omega\end{bmatrix} + \frac{1}{Mr_w}\begin{bmatrix} & 0_{3x4} & & \\ -\sin(\phi) & -\sin(\phi) & \sin(\phi) & \sin(\phi)\\ \cos(\phi) & -\cos(\phi) & -\cos(\phi) & \cos(\phi)\\ 2 & 2 & 2 & 2\end{bmatrix}\begin{bmatrix}\tau_1\\ \tau_2\\ \tau_3\\ \tau_4\end{bmatrix} \tag{31}$$

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

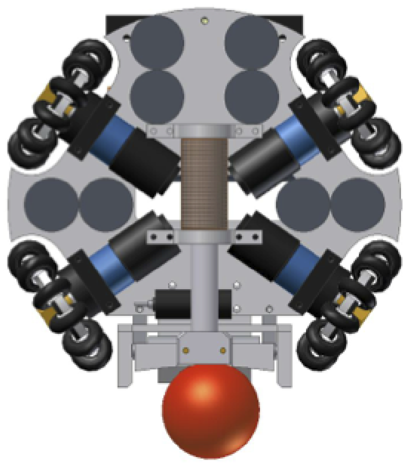


Figure 4: Configuration of three groups of motors where ϕ is 30, 45, and 60 degrees.

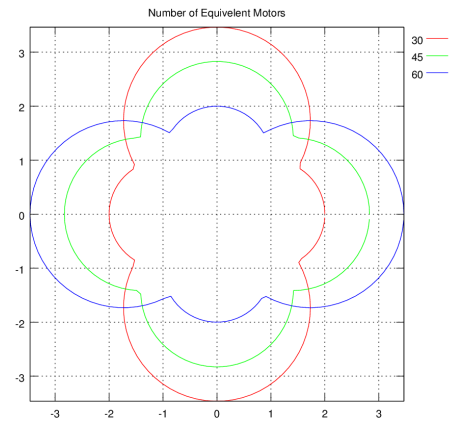


Figure 5: Number of equivalent motors for any direction under linear movement only, no rotational movement allowed.

Now using this we can show that this system is controllable but not fully observable.

$$rank\left(\begin{bmatrix} B & AB & A^2B & ... & A^5B \end{bmatrix}\right) = 6 \tag{32}$$

$$rank\left(\begin{bmatrix} C \\ CA \\ \vdots \\ CA^5 \end{bmatrix}\right) = 2 \tag{33}$$

An estimator must be developed in order to properly control the system and keep it stable.

4 Holonomic Robot Kinematics

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

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} -\sin(\phi) & \cos(\phi) & 1 \\ -\sin(\phi) & -\cos(\phi) & 1 \\ \sin(\phi) & -\cos(\phi) & 1 \\ \sin(\phi) & \cos(\phi) & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ R\omega \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \sin(\phi) & 0 & 0 \\ 0 & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ R\omega \end{bmatrix} \tag{34}$$

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 ($\begin{bmatrix} v_x & v_y & R\omega \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$), 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 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 mater 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 magnometer), the heading (θ) can be determined from the magnometer 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:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + \hat{B}_k u_k \tag{35}$$

$$\tilde{x} = \hat{x} - x \tag{36}$$

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

The navigation equations are given by Titterman and ?? [12], Chatfield [13], and Savage [14] are measured in a local navigation frame. In this case, it is the North, East, and down (NED) frame.

$$\dot{v}^e = C_b^e f^b - 2\omega_{ie}^e \times v^e + g_l^e \tag{37}$$

$$\dot{C}_b^e = C_b^e[\omega_{ib}^b \times] - [\omega_{ie}^e \times] C_b^e \tag{38}$$

$$\tag{39}$$

v^e vehicle velocity (x,y,z)

C_b^e DCM for rotating the IMU specific accelerations from the body frame to the NED navigation frame

ω_{ie}^e Earth's rotation vector

ω_{ib}^b rotation rates from the IMU gyroscopes measured in the body frame

f^b specific forces from the IMU accelerometers measured in the body frame

g_l^e local gravity vector including the centripetal force effects, this is also referred to as the "plumb bob" force

$a\times$ skew symetric matrix for vector a

Note that the term $2\omega_{ie}^e \times v^e$ is the centrifugal force exerted on the vehicle. The

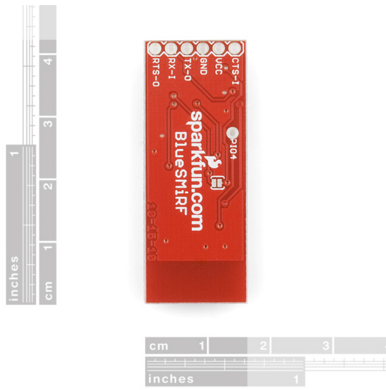
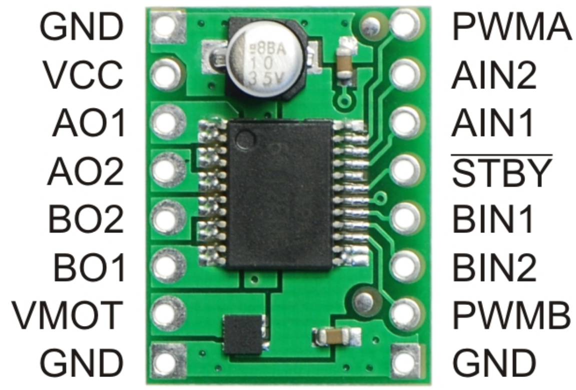


Figure 6: IMU.



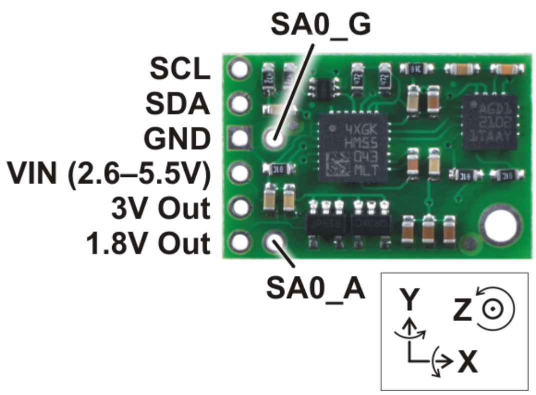
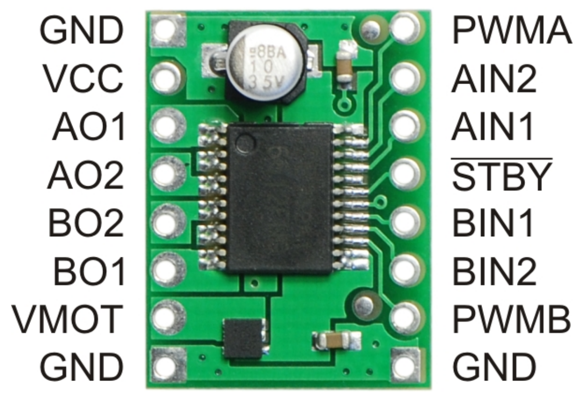


Figure 7: IMU.



7 Hardware Setup

7.1 Chasis

7.2 Electronic Power System

7.3 Electronics and Sensors

The robot is teleoperated from a laptop via a bluetooth link to an Arduino microcontroller which runs the motor drivers and sensors.

Figure 8: The software architecture composed of ROS nodes.

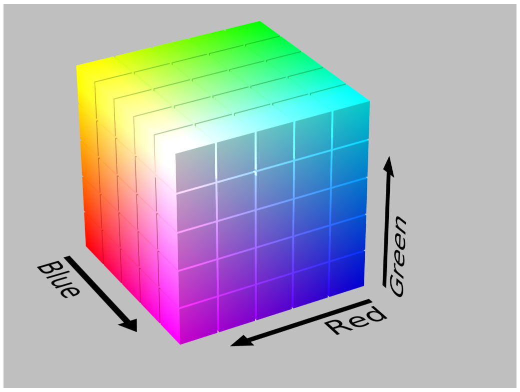


Figure 9: RGB color cube.

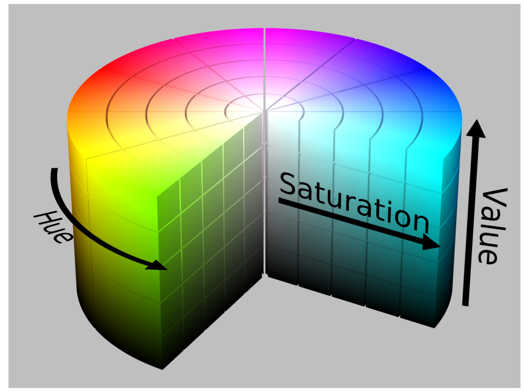


Figure 10: HSV color cylinder.

7.3.1 MiniIMU

7.3.2 Camera

7.4 Software

The robot utilizes the open source Robotic Operating System (ROS) [5]. The core software is run on a laptop and commands and telemetry is passed back and forth via serial bluetooth.

7.4.1 OpenCV

The vision system uses the Open Computer Vision (OpenCV [8]) is used to track the ball and guide the robot to the ball (visual servoing).

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"

7.4.2 Point Cloud Library

7.4.3 Kalman Filter

Covariance Matrix a covariance matrix is a matrix whose element in the i, j position is the covariance between the ith and jth elements of a random vector (that is, of a vector of random variables). Each element of the vector is a scalar random variable, either with a finite number of observed empirical values or with a finite or infinite number of potential values specified by a theoretical joint probability distribution of all the random variables.

RANSAC is an abbreviation for "RANdom SAmple Consensus". It is an iterative method to estimate parameters of a mathematical model from a set of observed data which contains outliers. It is a non-deterministic algorithm in the sense that it produces a reasonable result only with a certain probability, with this probability increasing as more iterations are allowed. An advantage of RANSAC is its ability to do robust estimation of the model parameters, i.e., it can estimate the parameters with a high degree of accuracy even when a significant number of outliers are present in the data set. A disadvantage of RANSAC is that there is no upper bound on the time it takes to compute these parameters. The algorithm was first published by Fischler and Bolles in 1981.The RANSAC algorithm is often used in computer vision, e.g., to simultaneously solve the correspondence problem and estimate the fundamental matrix related to a pair of stereo cameras.

Point Feature Histogram (PFH) representation is based on the relationships between the points in Pk2 and their normals. Simply put, it attempts to capture as best as possible the sampled surface variations by taking into account all the interactions between the directions of the estimated normals. The resultant hyperspace is thus dependent on the quality of the surface normal estimations at each point.

Fast Point Feature Histograms (FPFH), reduces the computational complexity of the algorithm to O(nk), while still retaining most of the discriminative power of the PFH.

8 Results

9 References

References

[1] Radu.

[2] Radu.

[3] R. Balakrishna, Ashitava Ghosal, "Modeling of Slip for Wheeled Mobile Robots," lEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. I I , NO. I , FEBRUARY 1995, pp. 126-132

[4] J. Agullo, S. Cardona, and J. Vivancos, Kinematics of vehicles with directional sliding wheels, Mechanisms and Muchine 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] Gelb, Applied Optimal Estimation

[12] Titterman

[13] Chatfield

[14] P. Savage. Strapdown Analytics Part 2. Strapdown Associates. Maple Plain. 2000