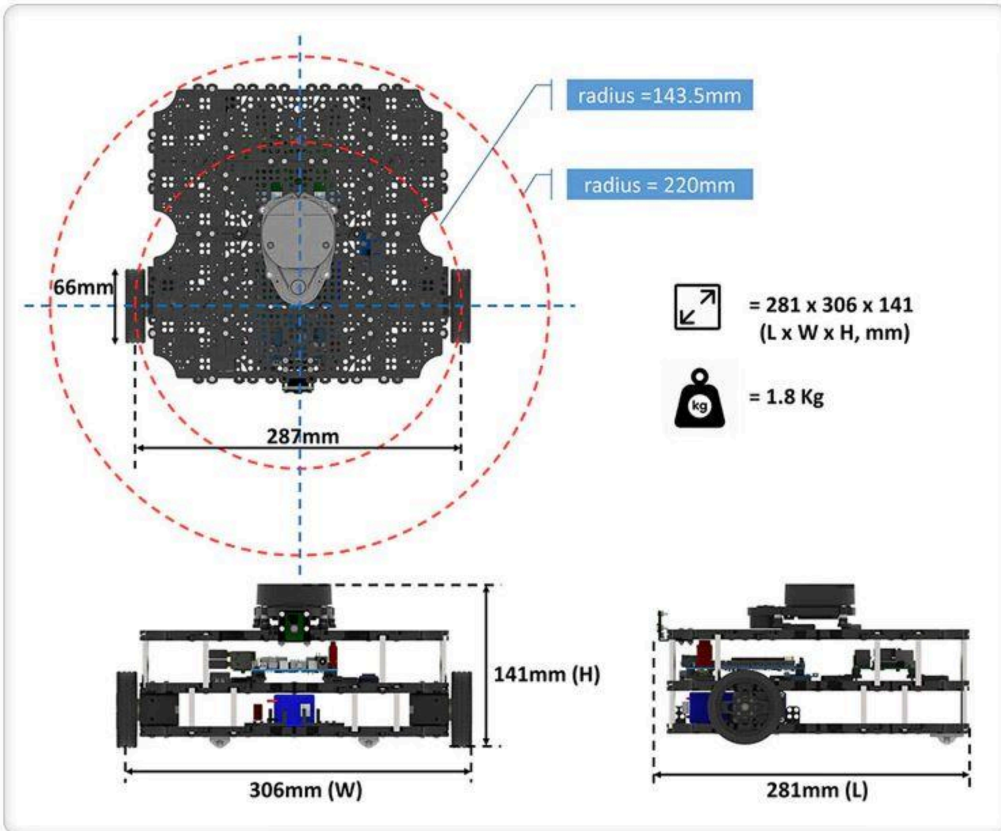
```
%----------------------------------------------------------------------------
%----------------------------------------------------------------------------
% COURSE:   RBE502 ROBOT CONTROL
%
% ASSIGNMENT:TERM PROJECT
%
% AUTHORS:  KYLE CANTRELL (KJCANTRELL@WPI.EDU)
%           CRAIG MILLER  (CDMILLER@WPI.EDU)
%           JORDAN NELSON (JVNELSON@WPI.EDU)
%
% DATE:     04/07/2019
%----------------------------------------------------------------------------
% TITLE: Dynamical_Model_TB3_OM
%----------------------------------------------------------------------------

% Dynamical Model for Turtlebot 3 Waffle Pi with OpenManipulator
% Based on paper from Dhaouadi & Hatab 2013:
% https://www.omicsonline.org/open-access/dynamic-modelling-of-differential
% drive-mobile-robots-using-lagrange-and-newtoneuler-methodologies-a-
% unified-framework-2168-9695.1000107.pdf
```



• Specification

Items	Waffle Pi
Maximum Translational Velocity	0.26m/s
Maximum Rotational Velocity	1.82rad/s (104.27 deg/s)
Maximum Payload	30kgs
Size (L x W x H)	281mm × 306mm × 141mm
Weight (+ SBC + Battery + Sensors)	1.8kgs
Operating Time	About 2hr
Charging Time	About 2hr 30min
DYNAMIXEL	XM430-W210-T
SBC	Raspberry Pi 3
Embedded Controller	OpenCR (32-bit ARM® Cortex®-M7)
Sensor	Raspberry Pi Camera 360° LiDAR 3-Axis gyroscope 3-Axis accelerometer 3-Axis magnetometer

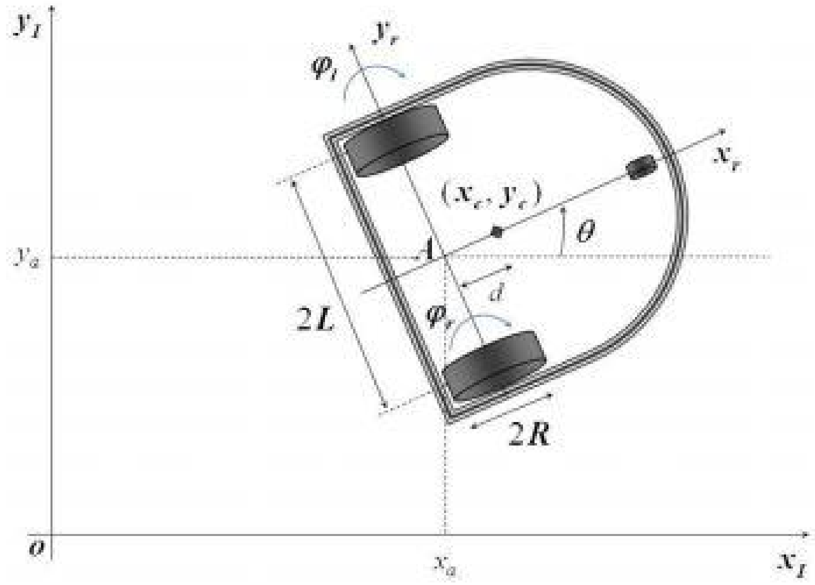


Figure 1: Differential Drive Mobile Robot (DDMR).

```
clear all; clc;

syms theta_dot phi_r phi_l phi_r_dot phi_l_dot I_c
% Define Turtlebot 3 Waffle Pi & OpenManipulator Parameters
R=0.033;     %meters, wheel radius
L=0.1435;    %meters, CG to wheel centerline.  Assumed to be constant. (estimate)
m_tb3=1.800; %kg, mass of Turtlebot 3 Waffle Pi
m_om=0.70;   %kg, mass of OpenManipulator
m=m_tb3+m_om; %kg, mass of TB3 and OM
m_w=0.030;   %kg, mass of each wheel
m_c=m-2*m_w; %kg, mass of turtlebot less wheels
d=-0.033;     %meters, wheel axis to CG (estimate)
I_c=0.0360;  %kg*m2, moment of inertia of the DDMR about the vertical
                   % axis through the center of mass (estimate)
I_w=0.000013;%kg*m2, moment of inertia of each
                   % driving wheel with a motor about the wheel axis (estimate)
I_m=0.000024;%kg*m2, moment of inertia of each driving wheel with
                   % a motor about the wheel diameter (estimate)
I=I_c+m_c*d^2+2*m_w*L^2+2*I_m; %kg*mm2, Total inertia of robot (estimate)

% Define Inertia Matrix (M_bar)
M11=I_w+(R^2/(4*L^2))*(m*L^2+I);
M12=(R^2/(4*L^2))*(m*L^2-I);
M21=(R^2/(4*L^2))*(m*L^2-I);
M22=I_w+(R^2/(4*L^2))*(m*L^2+I);
M_bar=[M11 M12;
       M21 M22];
% Define centripetal & coriolis matrix (V_bar)
V11=0;
V12=(R^2/2*L)*m_c*d*theta_dot;
V21=-(R^2/2*L)*m_c*d*theta_dot;
V22=0;
```

```
V_bar=[V11 V12;
       V21 V22];
% Define input matrix (B_bar)
B_bar=eye(2);
% Define wheel position & velocities
nu=[phi_r;
    phi_l];
nu_dot=[phi_r_dot;
        phi_l_dot];
% Show dynamic equations in general form
tau=B_bar\(M_bar*nu_dot+V_bar*nu); % [tau_right; tau_left]
tau=simplify(expand(tau))
```

tau =

$$\begin{pmatrix} \frac{5628815048954133\,\varphi_{l,\mathrm{dot}}}{36893488147419103232} + \frac{1408501907426143\,\varphi_{r,\mathrm{dot}}}{1152921504606846976} - \frac{7427681072376443\,\varphi_l\,\theta_{\mathrm{dot}}}{1180591620717411303424} \\ \frac{1408501907426143\,\varphi_{l,\mathrm{dot}}}{1152921504606846976} + \frac{5628815048954133\,\varphi_{r,\mathrm{dot}}}{36893488147419103232} + \frac{7427681072376443\,\varphi_r\,\theta_{\mathrm{dot}}}{1180591620717411303424} \end{pmatrix}$$