```
function theta_list_new = NextState(theta_list_old,theta_dot_list,dt,theta_dot_max)
%NEXTSTATE  One-step kinematics simulator for the KUKA youBot.
%
%   theta_list_new =
%   NEXTSTATE(theta_list_old,theta_dot_list,dt,theta_dot_max) produces a
%   column vector (theta_list_new) describing the new configuration of the
%   KUKA youBot after 1 time step using a 1st-order Euler method given the
%   previous configuration (theta_list_old), constant joint velocities
%   (theta_dot_list), the time step duration (dt), and the maximum
%   joint/wheel velocity magnitude (theta_dot_max).
%
%   Input           Value
%   theta_list_old  A 12-element column vector containing the 3 chassis
%                   configuration variables (phi,x,y), 5 arm joint angles,
%                   and 4 wheel angles of the KUKA youBot of the current
%                   time step. Angles are in radians, x and y are in
%                   meters.
%   theta_dot_list  A 9-element column vector containing the 5 arm joint
%                   velocities and 4 wheel velocities. The velocities are
%                   in radians/second.
%   dt              The time step duration in seconds.
%   theta_dot_max   The maximum joint/wheel velocity magnitude in
%                   radians/second.
%
%   Output          Value
%   theta_list_new  A 12-element column vector containing the 3 chassis
%                   configuration variables, 5 arm joint angles,
%                   and 4 wheel angles of the next time step.
%
%   See also NEXTSTATETEST, WRAPPER.

%   Written by David Lim for the MAE 204 Final Project in WI25.
%   Last modifed on 03/08/25.

% parameters of the mecanum wheel base
l = 0.47/2;
w = 0.3/2;
r = 0.0475;

% preallocate output
theta_list_new = zeros(12,1);

% appy velocity limit
theta_dot_list = clip(theta_dot_list,-theta_dot_max,theta_dot_max);

% increment joint/wheel angles
theta_list_new(4:end) = theta_list_old(4:end) + theta_dot_list*dt;

% compute body twist of chassis
Vb = r/4*[-1/(l+w) 1/(l+w) 1/(l+w) -1/(l+w);
          1 1 1 1;
          -1 1 -1 1]*theta_dot_list(6:end);
w_bz = Vb(1);
v_bx = Vb(2);
v_by = Vb(3);
```

```
% compute chassis velocities in body frame
if w_bz == 0
    qb_dot = Vb;
else
    qb_dot = [w_bz;
              (v_bx*sin(w_bz)+v_by*(cos(w_bz)-1))/w_bz;
              (v_by*sin(w_bz)+v_bx*(1-cos(w_bz)))/w_bz];
end
phi = theta_list_old(1);

% compute chassis velocities in space frame
q_dot = [1 0 0;
         0 cos(phi) -sin(phi);
         0 sin(phi) cos(phi)]*qb_dot;

% increment chassis configuration
theta_list_new(1:3) = theta_list_old(1:3)+q_dot*dt;
```

Published with MATLAB® R2024a