

```

%%Documento con código fuente%%
%Fichero bag_model.m para extraer los datos del fichero .bag y
%generar el .m
% clear rosbag_wrapper;
% clear ros.Bag;
clear all
close all
clc

%% Load a bag and get information about it (Read all messages on
a set of topics)
% Using load() lets you auto-complete filepaths.

folder = 'rosbag/';
file = 'prueba_todo';

matfile = [folder file '.mat'];
bagfile = [folder file '.bag'];

bag = ros.Bag(bagfile);
% bag = ros.Bag.load(bagfile);

%% Get information about the bag

bag.info()

%% ff_board (throttle and brake force)

topic_ff_board = '/ff_board';
[msgs_ff_board, meta_ff_board] = bag.readAll(topic_ff_board);

if(~isempty(msgs_ff_board))

ff_board.time_ini=double(msgs_ff_board{1,1}.header.stamp.sec)+1e-9*double(msgs_ff_board{1,1}.header.stamp.nsec);
    for i=1:length(msgs_ff_board)

ff_board.time(i)=double(msgs_ff_board{1,i}.header.stamp.sec)+1e-9*double(msgs_ff_board{1,i}.header.stamp.nsec)-ff_board.time_ini;

ff_board.steer_torque_neg(i)=double(msgs_ff_board{1,i}.ADCSteerTorqueNeg);

ff_board.steer_torque_pos(i)=double(msgs_ff_board{1,i}.ADCSteerTorquePos);

ff_board.accel_force(i)=double(msgs_ff_board{1,i}.ADCAccelForce);
;

ff_board.brake_force(i)=double(msgs_ff_board{1,i}.ADCBrakeForce);
;
        end

end

```

```

%% bt_board (steering wheel torque)

topic_bt_board = '/bt_board';
[msgs_bt_board, meta_bt_board] = bag.readAll(topic_bt_board);

if(~isempty(msgs_bt_board))

bt_board.time_ini=double(msgs_bt_board{1,1}.header.stamp.sec)+1e-9*double(msgs_bt_board{1,1}.header.stamp.nsec);
    for i=1:length(msgs_bt_board)

bt_board.time(i)=double(msgs_bt_board{1,i}.header.stamp.sec)+1e-9*double(msgs_bt_board{1,i}.header.stamp.nsec)-bt_board.time_ini;

bt_board.steer_torque_neg(i)=double(msgs_bt_board{1,i}.ADCSteerTorqueNeg);

bt_board.steer_torque_pos(i)=double(msgs_bt_board{1,i}.ADCSteerTorquePos);

bt_board.accel_force(i)=double(msgs_bt_board{1,i}.ADCAccelForce);
;

bt_board.brake_force(i)=double(msgs_bt_board{1,i}.ADCBrakeForce);
;
    end

end

%% imu (orientation, angular speed and linear acceleration)

topic_imu = '/xsens/imu';
[msgs_imu, meta_imu] = bag.readAll(topic_imu);

if(~isempty(msgs_imu))

    imu.time_ini=double(msgs_imu{1,1}.header.stamp.sec)+1e-9*double(msgs_imu{1,1}.header.stamp.nsec);
    imu_time_i_prev=0;
    imu_vx_i_prev=0;
    imu_vy_i_prev=0;
    imu_vz_i_prev=0;
    g=9.81;
    for i=1:length(msgs_imu)
        imu.time(i)=double(msgs_imu{1,i}.header.stamp.sec)+1e-9*double(msgs_imu{1,i}.header.stamp.nsec)-imu.time_ini;
        imu.qw(i)=msgs_imu{1,i}.orientation(1);
        imu.qx(i)=msgs_imu{1,i}.orientation(2);
        imu.qy(i)=msgs_imu{1,i}.orientation(3);
        imu.qz(i)=msgs_imu{1,i}.orientation(4);
        quat = [imu.qw(i) imu.qx(i) imu.qy(i) imu.qz(i)];
        eul = quat2eul(quat);
        imu.ex(i) = eul(1);
    end
end

```

```

        imu.ey(i) = eul(2);
        imu.ez(i) = eul(3);
%       Rx = [1 0 0; 0 cos(eul(1)) -sin(eul(1)); 0 sin(eul(1))
cos(eul(1))];
%       Ry = [cos(eul(2)) 0 sin(eul(2)); 0 1 0; -sin(eul(2)) 0
cos(eul(2))];
%       Rz = [cos(eul(3)) -sin(eul(3)) 0; sin(eul(3))
cos(eul(3)) 0; 0 0 1];
        Rx = rotation(eul(1), 'x');
        Ry = rotation(eul(2), 'y');
        Rz = rotation(eul(3), 'z');
        B = Rx*Ry*Rz*[0;0;1];
        dt=imu.time(i)-imu_time_i_prev;
        imu.wx(i)=msgs_imu{1,i}.angular_velocity(1);
        imu.wy(i)=msgs_imu{1,i}.angular_velocity(2);
        imu.wz(i)=msgs_imu{1,i}.angular_velocity(3);
        imu.ax(i)=msgs_imu{1,i}.linear_acceleration(1)+g*B(1);
        imu.ay(i)=msgs_imu{1,i}.linear_acceleration(2)+g*B(2);
        imu.az(i)=msgs_imu{1,i}.linear_acceleration(3)+g*B(3);
        imu.vx(i)=imu.ax(i)*dt+imu_vx_i_prev;
        imu.vy(i)=imu.ay(i)*dt+imu_vy_i_prev;
        imu.vz(i)=imu.az(i)*dt+imu_vz_i_prev;
        imu_time_i_prev=imu.time(i);
        imu_vx_i_prev=imu.vx(i);
        imu_vy_i_prev=imu.vy(i);
        imu_vz_i_prev=imu.vz(i);
    end

    imu_ez=imu.ez;
    loops=0;
    for j=2:length(imu_ez)
        if(imu_ez(j-1)>0.9*pi)&&(imu_ez(j)<-0.9*pi)
            loops=loops+1;
        elseif(imu_ez(j)>0.9*pi)&&(imu_ez(j-1)<-0.9*pi)
            loops=loops-1;
        end
        imu.ez(j) = imu_ez(j)+loops*2*pi;
    end

end

%% imu (orientation: roll, pitch, yaw)

topic_imu_rpy = '/xsens/rpy';
[msgs_imu_rpy, meta_imu_rpy] = bag.readAll(topic_imu_rpy);

if(~isempty(msgs_imu_rpy))

imu_rpy.time_ini=double(msgs_imu_rpy{1,1}.header.stamp.sec)+1e-
9*double(msgs_imu_rpy{1,1}.header.stamp.nsec);
    for i=1:length(msgs_imu_rpy)

imu_rpy.time(i)=double(msgs_imu_rpy{1,i}.header.stamp.sec)+1e-
9*double(msgs_imu_rpy{1,i}.header.stamp.nsec)-imu_rpy.time_ini;
        imu_rpy.r(i)=deg2rad(msgs_imu_rpy{1,i}.vector(1));
    end
end

```

```

        imu_rpy.p(i)=deg2rad(msgs_imu_rpy{1,i}.vector(2));
        imu_rpy.y(i)=deg2rad(msgs_imu_rpy{1,i}.vector(3));
    end

    imu_rpy_y=imu_rpy.y;
    loops=0;
    for j=2:length(imu_rpy_y)
        if(imu_rpy_y(j-1)>0.9*pi)&&(imu_rpy_y(j)<-0.9*pi)
            loops=loops+1;
        elseif(imu_rpy_y(j)>0.9*pi)&&(imu_rpy_y(j-1)<-0.9*pi)
            loops=loops-1;
        end
        imu_rpy.y(j) = imu_rpy_y(j)+loops*2*pi;
    end

end

%% imu (velocity)

topic_imu_vel = '/xsens/velocity';
[msgs_imu_vel, meta_imu_vel] = bag.readAll(topic_imu_vel);

if(~isempty(msgs_imu_vel))

    imu_vel.time_ini=double(msgs_imu_vel{1,1}.header.stamp.sec)+1e-
9*double(msgs_imu_vel{1,1}.header.stamp.nsec);
    for i=1:length(msgs_imu_vel)

        imu_vel.time(i)=double(msgs_imu_vel{1,i}.header.stamp.sec)+1e-
9*double(msgs_imu_vel{1,i}.header.stamp.nsec)-imu_vel.time_ini;
        imu_vel.vx(i)=msgs_imu_vel{1,i}.twist.twist.linear(1);
        imu_vel.vy(i)=msgs_imu_vel{1,i}.twist.twist.linear(2);
        imu_vel.vz(i)=msgs_imu_vel{1,i}.twist.twist.linear(3);
        imu_vel.wx(i)=msgs_imu_vel{1,i}.twist.twist.angular(1);
        imu_vel.wy(i)=msgs_imu_vel{1,i}.twist.twist.angular(2);
        imu_vel.wz(i)=msgs_imu_vel{1,i}.twist.twist.angular(3);
    end

end

%% imu (temperature)

topic_imu_tmp = '/xsens/temperature';
[msgs_imu_tmp, meta_imu_tmp] = bag.readAll(topic_imu_tmp);

if(~isempty(msgs_imu_tmp))

    imu_tmp.time_ini=double(msgs_imu_tmp{1,1}.header.stamp.sec)+1e-
9*double(msgs_imu_tmp{1,1}.header.stamp.nsec);
    for i=1:length(msgs_imu_tmp)

        imu_tmp.time(i)=double(msgs_imu_tmp{1,i}.header.stamp.sec)+1e-
9*double(msgs_imu_tmp{1,i}.header.stamp.nsec)-imu_tmp.time_ini;
        imu_tmp.t(i)=msgs_imu_tmp{1,i}.temperature;
    end

end

```

```

        end

end

%% imu (pressure)

topic_imu_pre = '/xsens/pressure';
[msgs_imu_pre, meta_imu_pre] = bag.readAll(topic_imu_pre);

if(~isempty(msgs_imu_pre))

imu_pre.time_ini=double(msgs_imu_pre{1,1}.header.stamp.sec)+1e-
9*double(msgs_imu_pre{1,1}.header.stamp.nsec);
    for i=1:length(msgs_imu_pre)

imu_pre.time(i)=double(msgs_imu_pre{1,i}.header.stamp.sec)+1e-
9*double(msgs_imu_pre{1,i}.header.stamp.nsec)-imu_pre.time_ini;
        imu_pre.p(i)=msgs_imu_pre{1,i}.fluid_pressure;
    end

end

%% gps (position)

topic_gps = '/xsens/gps_data';
[msgs_gps, meta_gps] = bag.readAll(topic_gps);

if(~isempty(msgs_gps))

    gps.time_ini=double(msgs_gps{1,1}.header.stamp.sec)+1e-
9*double(msgs_gps{1,1}.header.stamp.nsec);
    gps_latitude=msgs_gps{1,1}.latitude;
    gps_longitude=msgs_gps{1,1}.longitude;
    gps_altitude=msgs_gps{1,1}.altitude;

[gps_x_i_prev,gps_y_i_prev,f]=ll2utm(gps_latitude,gps_longitude)
;
    gps_z_i_prev=gps_altitude;
    gps_time_i_prev=0;
    for i=1:length(msgs_gps)
        gps.time(i)=double(msgs_gps{1,i}.header.stamp.sec)+1e-
9*double(msgs_gps{1,i}.header.stamp.nsec)-gps.time_ini;
        gps.latitude(i)=msgs_gps{1,i}.latitude;
        gps.longitude(i)=msgs_gps{1,i}.longitude;
        gps.altitude(i)=msgs_gps{1,i}.altitude;

[gps.x(i),gps.y(i),f]=ll2utm(gps.latitude(i),gps.longitude(i));
        gps.z(i)=gps.altitude(i);
        if((gps.x(i)-gps_x_i_prev)==0)&&(i~=1))
            gps.vx(i)=gps.vx(i-1);
            gps.vy(i)=gps.vy(i-1);
            gps.vz(i)=gps.vz(i-1);
        else
            dt=gps.time(i)-gps_time_i_prev;
            gps.vx(i)=(gps.x(i)-gps_x_i_prev)*1000/dt;

```

```

        gps.vy(i)=(gps.y(i)-gps_y_i_prev)*1000/dt;
        gps.vz(i)=(gps.z(i)-gps_z_i_prev)*1000/dt;
        dt=gps.time(i)-gps_time_i_prev;
        gps_x_i_prev=gps.x(i);
        gps_y_i_prev=gps.y(i);
        gps_z_i_prev=gps.z(i);
    end
end

end

%% gps (velocity)

topic_gps_vel = '/xsens/gps_vel';
[msgs_gps_vel, meta_gps_vel] = bag.readAll(topic_gps_vel);

if(~isempty(msgs_gps_vel))

gps_vel.time_ini=double(msgs_gps_vel{1,1}.header.stamp.sec)+1e-
9*double(msgs_gps_vel{1,1}.header.stamp.nsec);
    for i=1:length(msgs_gps_vel)

gps_vel.time(i)=double(msgs_gps_vel{1,i}.header.stamp.sec)+1e-
9*double(msgs_gps_vel{1,i}.header.stamp.nsec)-gps_vel.time_ini;
        gps_vel.vx(i)=msgs_gps_vel{1,i}.twist.twist.linear(1);
        gps_vel.vy(i)=msgs_gps_vel{1,i}.twist.twist.linear(2);
        gps_vel.vz(i)=msgs_gps_vel{1,i}.twist.twist.linear(3);
    end

end

end

%% Plot body/imu over time

topic_mtw_imu{1} = '/mtw_driver/imu/E';
topic_mtw_imu{2} = '/mtw_driver/imu/C';
topic_mtw_imu{3} = '/mtw_driver/imu/ED';
topic_mtw_imu{4} = '/mtw_driver/imu/BD';
topic_mtw_imu{5} = '/mtw_driver/imu/AD';
topic_mtw_imu{6} = '/mtw_driver/imu/MD';
topic_mtw_imu{7} = '/mtw_driver/imu/EI';
topic_mtw_imu{8} = '/mtw_driver/imu/BI';
topic_mtw_imu{9} = '/mtw_driver/imu/AI';
topic_mtw_imu{10} = '/mtw_driver/imu/MI';
[msgs_mtw_imu{1}, meta_mtw_imu{1}] =
bag.readAll(topic_mtw_imu{1});
[msgs_mtw_imu{2}, meta_mtw_imu{2}] =
bag.readAll(topic_mtw_imu{2});
[msgs_mtw_imu{3}, meta_mtw_imu{3}] =
bag.readAll(topic_mtw_imu{3});
[msgs_mtw_imu{4}, meta_mtw_imu{4}] =
bag.readAll(topic_mtw_imu{4});
[msgs_mtw_imu{5}, meta_mtw_imu{5}] =
bag.readAll(topic_mtw_imu{5});
[msgs_mtw_imu{6}, meta_mtw_imu{6}] =
bag.readAll(topic_mtw_imu{6});

```

```

[msgs_mtw_imu{7}, meta_mtw_imu{7}] =
bag.readAll(topic_mtw_imu{7});
[msgs_mtw_imu{8}, meta_mtw_imu{8}] =
bag.readAll(topic_mtw_imu{8});
[msgs_mtw_imu{9}, meta_mtw_imu{9}] =
bag.readAll(topic_mtw_imu{9});
[msgs_mtw_imu{10}, meta_mtw_imu{10}] =
bag.readAll(topic_mtw_imu{10});

for j=1:length(msgs_mtw_imu)
    if(~isempty(msgs_mtw_imu{j}))
        mtw_imu{j}.topic = topic_mtw_imu{j};

mtw_imu{j}.time_ini=double(msgs_mtw_imu{j}{1,1}.header.stamp.sec
)+1e-9*double(msgs_mtw_imu{j}{1,1}.header.stamp.nsec);

    [t_time_ini,i_time_ini]=min(abs(mtw_imu{j}.time_ini-
mtw_imu{j}.time_ini));
%     quat_imu_ini =
[msgs_mtw_imu{j}.qw(i_time_ini),mtw_imu{j}.qx(i_time_ini),mtw_imu{j}.
qy(i_time_ini),mtw_imu{j}.qz(i_time_ini)];
%     quat_imu_ini_inv = quatinv(quat_imu_ini);

    quat_imu_ini = [msgs_mtw_imu{j}{1,1}.orientation(1)
msgs_mtw_imu{j}{1,1}.orientation(2)
msgs_mtw_imu{j}{1,1}.orientation(3)
msgs_mtw_imu{j}{1,1}.orientation(4)];
    quat_imu_ini_inv = quatinv(quat_imu_ini);

    for i=1:length(msgs_mtw_imu{j})

mtw_imu{j}.time(i)=double(msgs_mtw_imu{j}{1,i}.header.stamp.sec)
+1e-9*double(msgs_mtw_imu{j}{1,i}.header.stamp.nsec)-
mtw_imu{j}.time_ini;

mtw_imu{j}.qw(i)=msgs_mtw_imu{j}{1,i}.orientation(1);
mtw_imu{j}.qx(i)=msgs_mtw_imu{j}{1,i}.orientation(2);
mtw_imu{j}.qy(i)=msgs_mtw_imu{j}{1,i}.orientation(3);
mtw_imu{j}.qz(i)=msgs_mtw_imu{j}{1,i}.orientation(4);
    quat = [mtw_imu{j}.qw(i) mtw_imu{j}.qx(i)
mtw_imu{j}.qy(i) mtw_imu{j}.qz(i)];
    eul = quat2eul(quat);
    mtw_imu{j}.ex(i) = eul(1);
    mtw_imu{j}.ey(i) = eul(2);
    mtw_imu{j}.ez(i) = eul(3);

    [t_time,i_time]=min(abs(mtw_imu{j}.time(i)-
mtw_imu{j}.time));
    quat_imu =
[msgs_mtw_imu{j}.qw(i_time),mtw_imu{j}.qx(i_time),mtw_imu{j}.qy(i_tim
e),mtw_imu{j}.qz(i_time)];

```

```

        eul =
quat2eul(quatmultiply(quatinv(quatmultiply(quat_imu_ini_inv,quat
_imu)),quatmultiply(quat_imu_ini_inv,quat)));
        mtw_imu{j}.R(i) = eul(1);
        mtw_imu{j}.P(i) = eul(2);
        mtw_imu{j}.Y(i) = eul(3);

mtw_imu{j}.wx(i)=msgs_mtw_imu{j}{1,i}.angular_velocity(1);
mtw_imu{j}.wy(i)=msgs_mtw_imu{j}{1,i}.angular_velocity(2);
mtw_imu{j}.wz(i)=msgs_mtw_imu{j}{1,i}.angular_velocity(3);
mtw_imu{j}.ax(i)=msgs_mtw_imu{j}{1,i}.linear_acceleration(1);
mtw_imu{j}.ay(i)=msgs_mtw_imu{j}{1,i}.linear_acceleration(2);
mtw_imu{j}.az(i)=msgs_mtw_imu{j}{1,i}.linear_acceleration(3);
    end

        mtw_imu{j}.ez = angle_unwrap(mtw_imu{j}.ez);
        mtw_imu{j}.Y = angle_unwrap(mtw_imu{j}.Y);

    end
end

%% Plot body/euler over time

topic_mtw_euler{1} = '/mtw_driver/euler/E';
topic_mtw_euler{2} = '/mtw_driver/euler/C';
topic_mtw_euler{3} = '/mtw_driver/euler/ED';
topic_mtw_euler{4} = '/mtw_driver/euler/BD';
topic_mtw_euler{5} = '/mtw_driver/euler/AD';
topic_mtw_euler{6} = '/mtw_driver/euler/MD';
topic_mtw_euler{7} = '/mtw_driver/euler/EI';
topic_mtw_euler{8} = '/mtw_driver/euler/BI';
topic_mtw_euler{9} = '/mtw_driver/euler/AI';
topic_mtw_euler{10}= '/mtw_driver/euler/MI';
[msgs_mtw_euler{1}, meta_mtw_euler{1}] =
bag.readAll(topic_mtw_euler{1});
[msgs_mtw_euler{2}, meta_mtw_euler{2}] =
bag.readAll(topic_mtw_euler{2});
[msgs_mtw_euler{3}, meta_mtw_euler{3}] =
bag.readAll(topic_mtw_euler{3});
[msgs_mtw_euler{4}, meta_mtw_euler{4}] =
bag.readAll(topic_mtw_euler{4});
[msgs_mtw_euler{5}, meta_mtw_euler{5}] =
bag.readAll(topic_mtw_euler{5});
[msgs_mtw_euler{6}, meta_mtw_euler{6}] =
bag.readAll(topic_mtw_euler{6});
[msgs_mtw_euler{7}, meta_mtw_euler{7}] =
bag.readAll(topic_mtw_euler{7});
[msgs_mtw_euler{8}, meta_mtw_euler{8}] =
bag.readAll(topic_mtw_euler{8});

```



```

[msgs_mtw_euler{9}, meta_mtw_euler{9}] =
bag.readAll(topic_mtw_euler{9});
[msgs_mtw_euler{10}, meta_mtw_euler{10}] =
bag.readAll(topic_mtw_euler{10});

for j=1:length(msgs_mtw_euler)
    if(~isempty(msgs_mtw_euler{j}))
        mtw_euler{j}.topic = topic_mtw_euler{j};

mtw_euler{j}.time_ini=double(msgs_mtw_euler{j}{1,1}.header.stamp.
.sec)+1e-9*double(msgs_mtw_euler{j}{1,1}.header.stamp.nsec);

    [t_time_ini,i_time_ini]=min(abs(mtw_imu{j}.time_ini-
imu.time_ini));
%     quat_imu_ini =
[imu.qw(i_time_ini),imu.qx(i_time_ini),imu.qy(i_time_ini),imu.qz
(i_time_ini)];
%     quat_imu_ini_inv = quatinv(quat_imu_ini);

    quat_ini =
eul2quat([deg2rad(msgs_mtw_euler{j}{1,1}.vector(1))
deg2rad(msgs_mtw_euler{j}{1,1}.vector(2))
deg2rad(msgs_mtw_euler{j}{1,1}.vector(3))]);
    quat_ini_inv = quatinv(quat_ini);

%     [t_time_ini,i_time_ini]=min(abs(mtw_euler{j}.time_ini-
imu.time_ini));
%     RxI = rotation(imu.ex(i_time_ini),'x');
%     RyI = rotation(imu.ey(i_time_ini),'y');
%     RzI = rotation(imu.ez(i_time_ini),'z');
%     RI = RxI*RyI*RzI;
%     % RI = eul2rotm([imu.ez(i_time_ini) imu.ey(i_time_ini)
imu.ex(i_time_ini)]);
%
%
mtw_euler{j}.r(1)=deg2rad(msgs_mtw_euler{j}{1,1}.vector(1));
%
mtw_euler{j}.p(1)=deg2rad(msgs_mtw_euler{j}{1,1}.vector(2));
%
mtw_euler{j}.y(1)=deg2rad(msgs_mtw_euler{j}{1,1}.vector(3));
%     RxIG = rotation(mtw_euler{j}.r(1),'x');
%     RyIG = rotation(mtw_euler{j}.p(1),'y');
%     RzIG = rotation(mtw_euler{j}.y(1),'z');
%     RIG = RxIG*RyIG*RzIG;
%     % RIG = eul2rotm([mtw_euler{j}.r(3) mtw_euler{j}.r(2)
mtw_euler{j}.r(1)]);
%
%     RIL = inv(RI)*RIG;

    for i=1:length(msgs_mtw_euler{j})

mtw_euler{j}.time(i)=double(msgs_mtw_euler{j}{1,i}.header.stamp.
sec)+1e-9*double(msgs_mtw_euler{j}{1,i}.header.stamp.nsec)-
mtw_euler{j}.time_ini;

mtw_euler{j}.r(i)=deg2rad(msgs_mtw_euler{j}{1,i}.vector(1));

```

```

mtw_euler{j}.p(i)=deg2rad(msgs_mtw_euler{j}{1,i}.vector(2));

mtw_euler{j}.y(i)=deg2rad(msgs_mtw_euler{j}{1,i}.vector(3));

%           RxG = rotation(mtw_euler{j}.r(i), 'x');
%           RyG = rotation(mtw_euler{j}.p(i), 'y');
%           RzG = rotation(mtw_euler{j}.y(i), 'z');
%           RG  = RxG*RyG*RzG;
% %         RG = eul2rotm([tf{i}.euler(j,3) tf{i}.euler(j,2)
tf{i}.euler(j,1)]);
%
%           [t_time,i_time]=min(abs(mtw_euler{j}.time(i)-
imu.time));
%           Rx = rotation(imu.ex(i_time), 'x');
%           Ry = rotation(imu.ey(i_time), 'y');
%           Rz = rotation(imu.ez(i_time), 'z');
%           R  = Rx*Ry*Rz;
% %         R = eul2rotm([imu.ez(i_time) imu.ey(i_time)
imu.ex(i_time)]);
%
%           RL = inv(RIL)*inv(R)*RG;
%           mtw_euler{j}.Y(i)=atan2(RL(2,1),RL(1,1));
%           mtw_euler{j}.P(i)=atan2(-
RL(3,1),sqrt(RL(3,2).*RL(3,2)+RL(3,3).*RL(3,3)));
%           mtw_euler{j}.R(i)=atan2(RL(3,2),RL(3,3));
%
% %         mtw_euler{j}.eulerL(i,:) = rotm2eul(RL);
% %         mtw_euler{j}.Y(i) = mtw_euler{j}.eulerL(i,1);
% %         mtw_euler{j}.P(i) = mtw_euler{j}.eulerL(i,2);
% %         mtw_euler{j}.R(i) = mtw_euler{j}.eulerL(i,3);

%           [t_time,i_time]=min(abs(mtw_euler{j}.time(i)-
imu.time));
%           quat_imu =
[imu.qw(i_time),imu.qx(i_time),imu.qy(i_time),imu.qz(i_time)];
%           quat = eul2quat([mtw_euler{j}.r(i) mtw_euler{j}.p(i)
mtw_euler{j}.y(i)]);
%           eul =
quat2eul(quatmultiply(quatinv(quatmultiply(quat_imu_ini_inv,quat
_imu)),quatmultiply(quat_ini_inv,quat)));
%           mtw_euler{j}.R(i) = eul(1);
%           mtw_euler{j}.P(i) = eul(2);
%           mtw_euler{j}.Y(i) = eul(3);

end

%           mtw_euler{j}.y = angle_unwrap(mtw_euler{j}.y);
%           mtw_euler{j}.Y = angle_unwrap(mtw_euler{j}.Y);
end
end

% tf (steering, throttle and brake transforms)

topic_tf = '/tf';
[msgs_tf, meta_tf] = bag.readAll(topic_tf);

```

```

if(~isempty(msgs_tf))

    for i=2:length(msgs_tf{1,1}.transforms)

if(double(msgs_tf{1,1}.transforms(i).header.stamp.sec)+1e-
9*double(msgs_tf{1,1}.transforms(i).header.stamp.nsec)<=double(m
sgs_tf{1,1}.transforms(i-1).header.stamp.sec)+1e-
9*double(msgs_tf{1,1}.transforms(i-1).header.stamp.nsec))

tf_time_ini=double(msgs_tf{1,1}.transforms(i).header.stamp.sec)+
1e-9*double(msgs_tf{1,1}.transforms(i).header.stamp.nsec);
    end
    end
    for i=1:length(msgs_tf{1,1}.transforms)
        tf{i}.time_ini=tf_time_ini;

tf{i}.frame_id=msgs_tf{1,1}.transforms(i).header.frame_id;

tf{i}.child_frame_id=msgs_tf{1,1}.transforms(i).child_frame_id;
        for j=1:length(msgs_tf)

tf{i}.time(j)=double(msgs_tf{1,j}.transforms(i).header.stamp.sec
)+1e-9*double(msgs_tf{1,j}.transforms(i).header.stamp.nsec)-
tf{i}.time_ini;

tf{i}.translation(j,:)=msgs_tf{1,j}.transforms(i).transform.tran
slation';

tf{i}.rotation(j,:)=msgs_tf{1,j}.transforms(i).transform.rotatio
n';

        tf{i}.euler(j,:)=quat2eul(tf{i}.rotation(j,:));

%           RxG = rotation(tf{i}.euler(j,1),'x');
%           RyG = rotation(tf{i}.euler(j,2),'y');
%           RzG = rotation(tf{i}.euler(j,3),'z');
%           RG  = RxG*RyG*RzG;
% %           RG = eul2rotm([tf{i}.euler(j,3) tf{i}.euler(j,2)
tf{i}.euler(j,1)]);
%
%           [t_time,i_time]=min(abs(tf{i}.time(j)-imu.time));
%           RxT = rotation(imu.ex(i_time),'x');
%           RyT = rotation(imu.ey(i_time),'y');
%           RzT = rotation(imu.ez(i_time),'z');
%           RT  = RxT*RyT*RzT;
% %           RT = eul2rotm([imu.ez(i_time) imu.ey(i_time)
imu.ex(i_time)]);
%
%           [t_time_ini,i_time_ini]=min(abs(tf{i}.time_ini-
imu.time_ini));
%           RxI = rotation(imu.ex(i_time_ini),'x');
%           RyI = rotation(imu.ey(i_time_ini),'y');
%           RzI = rotation(imu.ez(i_time_ini),'z');
%           RI  = RxI*RyI*RzI;
%
%           RL = inv(inv(RI)*RT)*RG;

```

```

%           tf{i}.Y(j)=atan2(RL(2,1),RL(1,1));
%           tf{i}.P(j)=atan2(-
RL(3,1),sqrt(RL(3,2).*RL(3,2)+RL(3,3).*RL(3,3)));
%           tf{i}.R(j)=atan2(RL(3,2),RL(3,3));
%
% %           tf{i}.eulerL(j,:) = rotm2eul(RL);
% %           tf{i}.Y(j) = tf{i}.eulerL(j,1);
% %           tf{i}.P(j) = tf{i}.eulerL(j,2);
% %           tf{i}.R(j) = tf{i}.eulerL(j,3);

        [t_time,i_time]=min(abs(tf{i}.time(j)-imu.time));
        quat_imu =
[imu.qw(i_time),imu.qx(i_time),imu.qy(i_time),imu.qz(i_time)];
        quat = eul2quat([tf{i}.euler(j,1) tf{i}.euler(j,2)
tf{i}.euler(j,3)]);
        eul =
quat2eul(quatmultiply(quatinv(quatmultiply(quat_imu_ini_inv,quat
_imu)),quatmultiply(quat_ini_inv,quat)));
        tf{i}.R(j) = eul(1);
        tf{i}.P(j) = eul(2);
        tf{i}.Y(j) = eul(3);
    end

    tf{i}.euler(:,1) = angle_unwrap(tf{i}.euler(:,1));
    tf{i}.Y = angle_unwrap(tf{i}.Y);
end

end

%% odometry (position and linear speed from GPS, orientation and
angular speed from IMU)

topic_odom = '/odom';
[msgs_odom, meta_odom] = bag.readAll(topic_odom);

if(~isempty(msgs_odom))

    odom.time_ini=double(msgs_odom{1,1}.header.stamp.sec)+1e-
9*double(msgs_odom{1,1}.header.stamp.nsec);
    for i=1:length(msgs_odom)
        odom.time(i)=double(msgs_odom{1,i}.header.stamp.sec)+1e-
9*double(msgs_odom{1,i}.header.stamp.nsec)-odom.time_ini;
        % odom.time(i)=i;
        odom.x(i)=msgs_odom{1,i}.pose.pose.position(1);
        odom.y(i)=msgs_odom{1,i}.pose.pose.position(2);
        % odom.t(i)=msgs1{1,i}.pose.pose.orientation;
        odom.vx(i)=msgs_odom{1,i}.twist.twist.linear(1);
        % odom.vy(i)=msgs1{1,i}.twist.twist.linear(2);
        % odom.vz(i)=msgs1{1,i}.twist.twist.linear(3);
        % odom.wx(i)=msgs1{1,i}.twist.twist.angular(1);
        % odom.wy(i)=msgs1{1,i}.twist.twist.angular(2);
        odom.wz(i)=msgs_odom{1,i}.twist.twist.angular(3);
    end

end

end

```

```

%% joystick

topic_joy = '/joy';
[msgs_joy, meta_joy] = bag.readAll(topic_joy);

if(~isempty(msgs_joy))

    joy.time_ini=double(msgs_joy{1,1}.header.stamp.sec)+1e-
9*double(msgs_joy{1,1}.header.stamp.nsec);
    for i=1:length(msgs_joy)
        joy.time(i)=double(msgs_joy{1,i}.header.stamp.sec)+1e-
9*double(msgs_joy{1,i}.header.stamp.nsec)-joy.time_ini;
        %    joy.time(i)=i;
        joy.axes1(i)=msgs_joy{1,i}.axes(1);
        joy.axes2(i)=msgs_joy{1,i}.axes(2);
        joy.axes3(i)=msgs_joy{1,i}.axes(3);
        joy.buttons1(i)=msgs_joy{1,i}.buttons(1);
        joy.buttons2(i)=msgs_joy{1,i}.buttons(2);
        joy.buttons3(i)=msgs_joy{1,i}.buttons(3);
        joy.buttons4(i)=msgs_joy{1,i}.buttons(4);
    end

end

%% biosignals 1

topic_bio1 = 'bitalino1/bio';
[msgs_bio1, meta_bio1] = bag.readAll(topic_bio1);

if(~isempty(msgs_bio1))

    bio1.time_ini=double(msgs_bio1{1,1}.header.stamp.sec)+1e-
9*double(msgs_bio1{1,1}.header.stamp.nsec);
    for i=1:length(msgs_bio1)
        bio1.time(i)=double(msgs_bio1{1,i}.header.stamp.sec)+1e-
9*double(msgs_bio1{1,i}.header.stamp.nsec)-bio1.time_ini;
        bio1.d1(i)=msgs_bio1{1,i}.d(1);
        bio1.d2(i)=msgs_bio1{1,i}.d(2);
        bio1.d3(i)=msgs_bio1{1,i}.d(3);
        bio1.d4(i)=msgs_bio1{1,i}.d(4);
        bio1.s1(i)=msgs_bio1{1,i}.s(1);
        bio1.s2(i)=msgs_bio1{1,i}.s(2);
        bio1.s3(i)=msgs_bio1{1,i}.s(3);
        bio1.s4(i)=msgs_bio1{1,i}.s(4);
        bio1.s5(i)=msgs_bio1{1,i}.s(5);
        bio1.s6(i)=msgs_bio1{1,i}.s(6);
        bio1.f1(i)=double(msgs_bio1{1,i}.f(1));
        bio1.f2(i)=double(msgs_bio1{1,i}.f(2));
        bio1.f3(i)=double(msgs_bio1{1,i}.f(3));
        bio1.f4(i)=double(msgs_bio1{1,i}.f(4));
        bio1.f5(i)=double(msgs_bio1{1,i}.f(5));
        bio1.f6(i)=double(msgs_bio1{1,i}.f(6));
    end

end

end

```

```

%% biosignals 2

topic_bio2 = 'bitalino2/bio';
[msgs_bio2, meta_bio2] = bag.readAll(topic_bio2);

if(~isempty(msgs_bio2))

    bio2.time_ini=double(msgs_bio2{1,1}.header.stamp.sec)+1e-
9*double(msgs_bio2{1,1}.header.stamp.nsec);
    for i=1:length(msgs_bio2)
        bio2.time(i)=double(msgs_bio2{1,i}.header.stamp.sec)+1e-
9*double(msgs_bio2{1,i}.header.stamp.nsec)-bio2.time_ini;
        bio2.d1(i)=msgs_bio2{1,i}.d(1);
        bio2.d2(i)=msgs_bio2{1,i}.d(2);
        bio2.d3(i)=msgs_bio2{1,i}.d(3);
        bio2.d4(i)=msgs_bio2{1,i}.d(4);
        bio2.s1(i)=msgs_bio2{1,i}.s(1);
        bio2.s2(i)=msgs_bio2{1,i}.s(2);
        bio2.s3(i)=msgs_bio2{1,i}.s(3);
        bio2.s4(i)=msgs_bio2{1,i}.s(4);
        bio2.s5(i)=msgs_bio2{1,i}.s(5);
        bio2.s6(i)=msgs_bio2{1,i}.s(6);
        bio2.f1(i)=double(msgs_bio2{1,i}.f(1));
        bio2.f2(i)=double(msgs_bio2{1,i}.f(2));
        bio2.f3(i)=double(msgs_bio2{1,i}.f(3));
        bio2.f4(i)=double(msgs_bio2{1,i}.f(4));
        bio2.f5(i)=double(msgs_bio2{1,i}.f(5));
        bio2.f6(i)=double(msgs_bio2{1,i}.f(6));
    end

end

%% PLOT DATA

plot_data();

figure
plot(bt_board.time, bt_board.accel_force*1.85)
title('Bt_ accel')
figure
plot(bt_board.time, bt_board.brake_force*1.95)
title('Bt_ brake')
figure
plot(bt_board.time, bt_board.steer_torque_pos*0.03)
title('Bt_ steer_ pos (izq)')
figure
plot(bt_board.time, bt_board.steer_torque_neg*0.03)
title('Bt_ steer_ neg (der)')

%% Save data to file

save(matfile);

```