In this document, we briefly discuss about kinematic models for wheeled platforms. We discuss about these models because we use them, very frequently, for working in our projects.

State space representation for Dynamical Systems

Equations

A dynamic linear system could be represented in the state space form,

$$\frac{dX}{dt} = A \cdot X + B \cdot u$$
$$X \in \mathbb{R}^n \quad , u \in \mathbb{R}^m, \quad A \in \mathbb{R}^{n \times n} \quad , \quad B \in \mathbb{R}^{n \times m} \tag{E1}$$

where the vector ***X*** represents the state variables and the vector ***u*** the *control inputs*.

The general case is represented as a non-linear system. The models for those systems have the general form

$$\frac{dX}{dt} = F\left(X, u\right)$$
$$X \in \mathbb{R}^n \quad , u \in \mathbb{R}^m, \tag{E2}$$

The dynamic model for representing the position and attitude of a wheeled vehicle is usually a non-linear model. The state vector involves variables such as position of the robot, attitude and velocities.

If the robot operates on a (almost) flat two-dimensional world then we can represent its states through two position variables and its orientation.

Vehicle / Robot Kinematics

We define a mathematical model for describing the evolution of the pose (position and orientation) of the platform. This model is defined by a set of first order differential equations.
For example, we present the model of a tricycle (car-like platform), in 2D.

$$\frac{dx}{dt} = v(t) \cdot \cos\left(\theta(t)\right)$$
$$\frac{dy}{dt} = v(t) \cdot \sin\left(\theta(t)\right)$$
$$\frac{d\theta}{dt} = \frac{v(t)}{L} \cdot \tan\left(\alpha(t)\right) \tag{E3}$$

$v(t)$: speed at the center-back of the vehicle, at time t

$\alpha(t)$: steering angle at time t

$x(t), y(t), \theta(t)$: vehicle's pose.

(pose: position and heading respect to a fixed coordinate frame)

It shows that the variation of X and Y are function of the speed (longitudinal velocity) and the current heading of the platform. According to this model, the platform can only move in a direction which is parallel to the current heading. This constrained variation of X and Y is called NON-HOLONOMIC kinematics. It means that the instantaneous velocity vector is constrained by the current state of the system, i.e. the heading of the platform for this case.
In this model, the inputs of the system are the measured speed (at the center-back of the platform) and the steering angle (front wheels). Suppose you have sensors which measure these variables, then you would be able to estimate the evolution of the vehicle's pose, by using this approximate model.

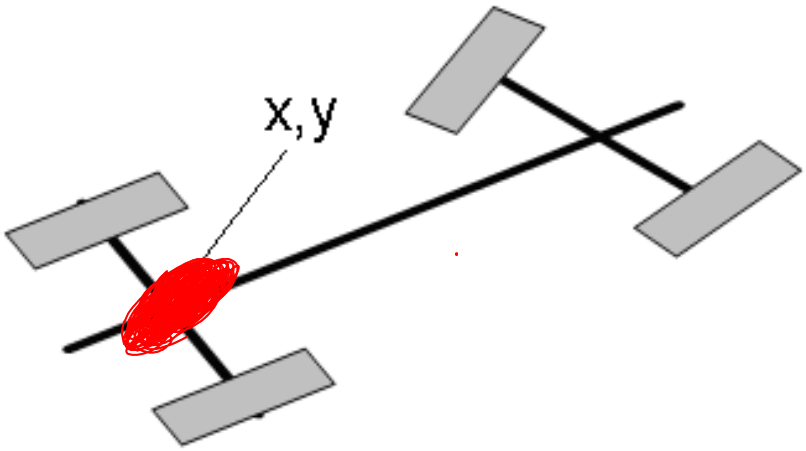

Figure 1 In this platform the driving wheels (the ones that produce the traction) are the back wheels (no motor is shown in the picture). The front wheels do not perform traction but steering. This system can be approximated by the model of a tricycle.

Diversity of Kinematic Models

If you read the literature, for teaching and research, you would find equivalent kinematic models for the same type of machine. They are in fact the same model but applied at different points on the vehicle's body. By doing some translations we can model the evolution at any point of the rigid body of the vehicle. For instance, suppose a point of interest at coordinates (a, b) in the vehicle body,

$$P_1 = \begin{bmatrix} x \\ y \end{bmatrix}, \quad P_2 = P_1 + R(\theta) \cdot \begin{bmatrix} a \\ b \end{bmatrix}$$

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \qquad \text{(E4)}$$

$$\begin{aligned} x_2 &= x_1 + \cos(\theta)\cdot a - \sin(\theta)\cdot b \\ y_2 &= y_1 + \sin(\theta)\cdot a + \cos(\theta)\cdot b \end{aligned} \qquad \text{(E5)}$$

If we derive P2 respect to the time

$$\dot{x}_2(t) = v(t)\cdot\cos\left(\theta(t)\right) + \left(-\sin\left(\theta(t)\right)\cdot a - \cos\left(\theta(t)\right)\cdot b\right)\cdot\dot{\theta}(t)$$

$$\dot{y}_2(t) = v(t)\cdot\sin\left(\theta(t)\right) + \left(\cos\left(\theta(t)\right)\cdot a - \sin\left(\theta(t)\right)\cdot b\right)\cdot\dot{\theta}(t)$$

Where the derivative of the heading is

$$\dot{\theta}(t) = \frac{v(t)}{L}\cdot\tan\left(\alpha(t)\right) \qquad \text{(E6)}$$

This new differential equation describes the kinematics at the point P_2.

If we measure the speed at a different place, for instance at the front wheel of a tricycle then we will have a different expression

$$\dot{\theta}(t) = v_{fw}(t)/L\cdot\sin\left(\alpha(t)\right)$$

This is because the instantaneous speed is not the same at different points on the rigid body (because this rigid body is rotating as well). Then, for instance, if we measure (or estimate) the speed at the front wheel or at some of the back wheels, that fact should be considered when the measurement is used.

Note that the steering angle has some limitations ($-\pi/2 < \alpha(t) < \pi/2$) in order to avoid the singularity that happens in equation (E6) (that limitation is present due to physical/mechanical reasons as well).

The mathematical singularity in equation (E6) is consistent with the reality. If the steering angle is 90 degrees then the machine is just rotating and then the translational speed at the point between the back wheels would be null; however, there is a variation in the heading. In the limit, equation (E6) should give the correct solution.

Kinematic Model Based on Angular Rate

If we are able to know (measure or estimate) the yaw angular rate, then we can propose another version of the kinematic model for a car-like platform in 2D

.

$$\frac{dx}{dt} = v(t)\cdot\cos\left(\theta(t)\right)$$

$$\frac{dy}{dt} = v(t)\cdot\sin\left(\theta(t)\right) \qquad \text{(E7)}$$

$$\frac{d\theta}{dt} = \omega(t)$$

$v(t)$: speed the center-back of the vehicle at time t

$\omega(t)$: yaw rate at time t

$x(t), y(t), \theta(t)$: vehicle's pose.

This model is usually used when the angular rate is known, such as in cases when we install a gyroscope (usually part of the 3D IMU devices that we use in our projects, in the course).

Note that the only difference between this model and the one that utilizes the steering angle, is in the prediction of the heading component, $\theta(t)$; the rest of the states preserve the same equations.

Simulation of the Platform Dynamics

Given a model, we can perform a simulation of the evolution of the states given a sequence of control actions (inputs to the model).
One approach is by approximating the continuous system by a discrete-time one,

$$x(t+\tau) = x(t) + \tau \cdot F\left(x(t), u(t)\right)$$

This approximation is called Euler's approximation. It just uses the first order derivatives at time t to predict the state at time $t+\tau$. If the time horizon τ is sufficiently short, then the approximation is appropriate. The term "sufficiently short" is highly dependent on the type of dynamics of the systems. Our robots are slow mechanical systems (low frequency dynamics). Sample times of milliseconds are sufficiently fast to approximate the continuous model.
The result of this simulation is the trajectory (position and heading) of the platform. If the model is a good approximation to the real system, then this estimated trajectory will be very close to the real one, at least for a short term.

An approximated discrete-time version of the kinematic model (E3) is the following one,

$$\begin{aligned} x(t+\tau) &= x(t) + \tau \cdot v(t) \cdot \cos\left(\theta(t)\right) \\ y(t+\tau) &= y(t) + \tau \cdot v(t) \cdot \sin\left(\theta(t)\right) \\ \theta(t+\tau) &= \theta(t) + \tau \cdot \frac{v(t)}{L} \cdot \tan\alpha(t) \end{aligned} \qquad \text{(E8)}$$

This approximation is valid provided that the "sample time" τ is small enough. For our machines "small enough" means that τ in the order of milliseconds (e.g. 20ms).

Matlab example

```
% Example using the function PredictVehiclePose().
% It implements a dead-reckoning prediction (i.e. kinematic model based on current speed
% and steering).
%.....................................................................
function main()

Dt=0.05 ;           % Step, for Euler approximation: 50ms (expressed in seconds)
duration = 100 ;    % Simulation horizon

times = [0:Dt:duration];
N=length(times) ;

%.....................................
speed       = ones(N,1);
steering    = (2+ 20*cos(times*2*pi/20))*pi/180; % (behavior of certain crazy driver)
% In the arrays "speed[]" and "steering[]"
% you can define the time evolution of the speed & steering inputs (it is a simulation!)
%.....................................

% buffer for storing simulated positions
Pose=zeros(3,N);

X0 = [0;0;0] ;          %Initial Condition  (x, y, heading), in(meters, meters, radians)

X=X0;
For I=1:N,              %iterations / Loop
      X = PredictVehiclePose(X, steering(i), speed(i) ,Dt);
      Pose(:,i)=X ;
end ;

% Show the results.
figure(1) ; clf() ;
plot(Pose(1,:),Pose(2,:)) ; xlabel('X (m)'); ylabel('Y (m)'); title('Position');
%figure(2) ; clf ; plot(Pose(3,:)) ;   %heading

return;

%.....................................................................%
"X=PredictVehiclePose(x0, steering, speed,dt)"
% X0 current state ( [x; y; heading] )
% X estimated next state ( [x; y; heading] at time t+dt)
% speed : current speed (m/s)
% steering : current steering angle (at time t)(in radians)
% dt is the "integration" horizon (should be a fraction of second)
% Jose Guivant - For AAS
% Tricycle / Ackermann model, discrete version

function X = PredictVehiclePose(X0,steering,speed,dt)
      % Remember: state vector X = [x; y; heading]
      LCar=3 ;    %in this example L=3meters
      X=X0 ;
      dL = dt*speed ;
      X(3) = X0(3)+ tan(steering)*dL/LCar ;
      X(1:2) = X0(1:2)+dL*[ cos(X0(3));sin(X0(3))] ;
return ;
%.....................................................................
```

Platforms with Differential Steering

Figure 2 illustrates a robot with differential steering. Each wheel has the same radius R_w, and the wheels are separated by a distance L. The control inputs of the robot are:

Rotational velocity of left wheel: ω_l [rad/s]

Rotational velocity of right wheel: ω_r [rad/s]

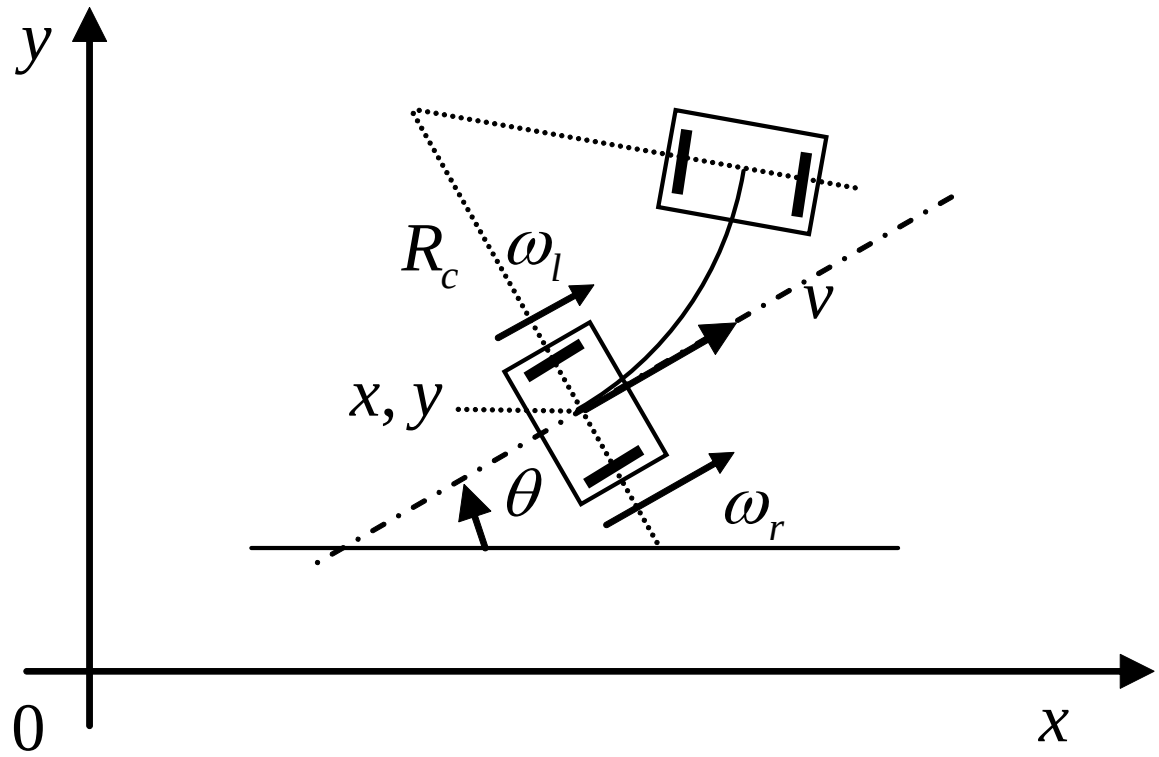

Figure 2 Differential steering platform

Instantaneous Curvature

Let us find an instantaneous curvature when $\omega_l > \omega_r$. Figure 2 shows the instantaneous curvature of such a robot. The linear and angular velocities of the center of the robot are

$$v = \frac{R_w}{2}(\omega_l + \omega_r), \qquad \omega = \frac{R_w}{L}(\omega_r - \omega_l) \tag{E9}$$

The radius of the instantaneous curvature is given by

$$R_c = \frac{v}{\omega} \tag{E10}$$

The parameters R_w and L are the wheel radius and the separation between wheels, respectively.

Orientation

The state space equation describing the orientation of the robot is given by

$$\dot{\theta} = \frac{R_w}{L}\left(\omega_r - \omega_l\right) \tag{E11}$$

When both wheels' speeds are identical, then the variation of derivative of the heading is nil. The platform would describe a straight line.

If $\omega_l = -\omega_r$, i.e. wheels rotate at the same angular speed but in opposite directions, then the platform would experiment a pure rotation (no translation, i.e. v=0).

Position

The process model for this platform is as follows,

$$\begin{cases} \dot{x} = \dfrac{R_w}{2} \cdot \left(\omega_l + \omega_r\right) \cdot \cos\theta \\ \dot{y} = \dfrac{R_w}{2} \cdot \left(\omega_l + \omega_r\right) \cdot \sin\theta \\ \dot{\theta} = \dfrac{R_w}{L} \cdot \left(\omega_r - \omega_l\right) \end{cases} \tag{E12}$$