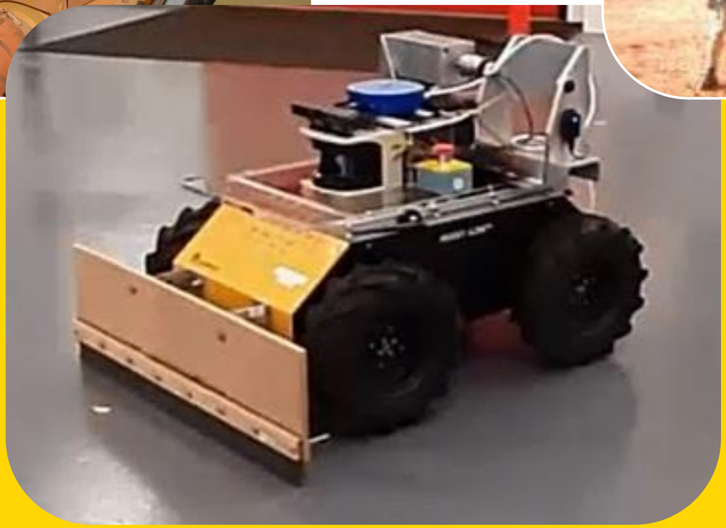
Advanced Autonomous Systems

MTRN 4010

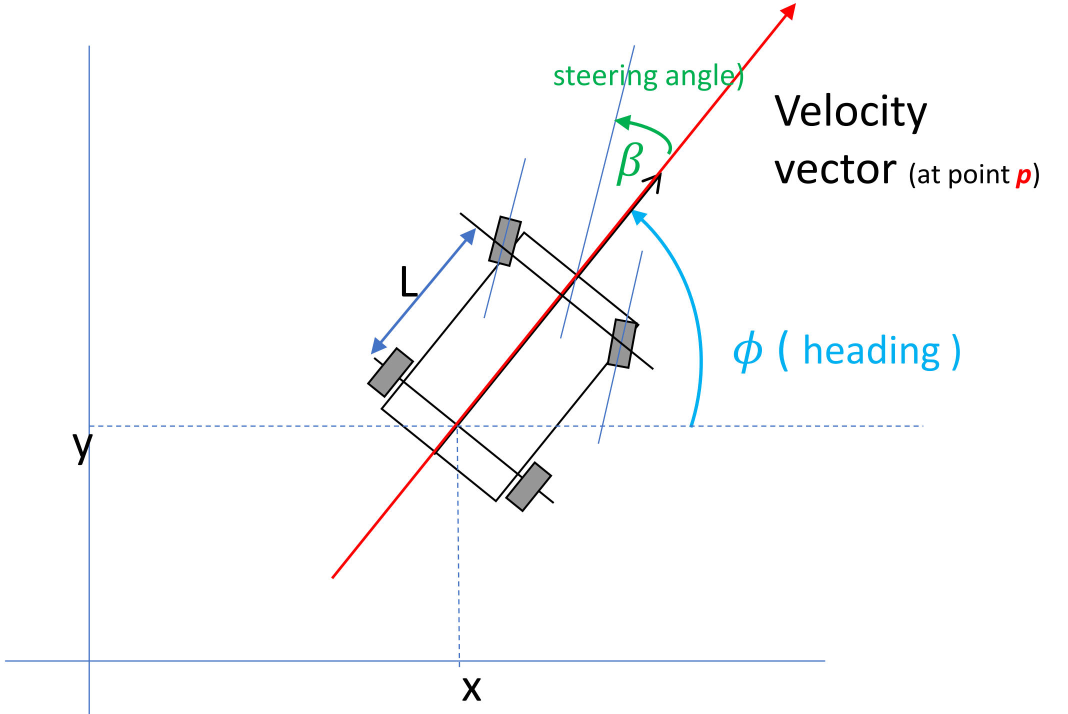
Lecturer:
Jose
Guivant



UNSW
AUSTRALIA

Never Stand Still | Faculty of Engineering | School of Mechanical and Manufacturing Engineering

KINEMATIC MODELS



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.

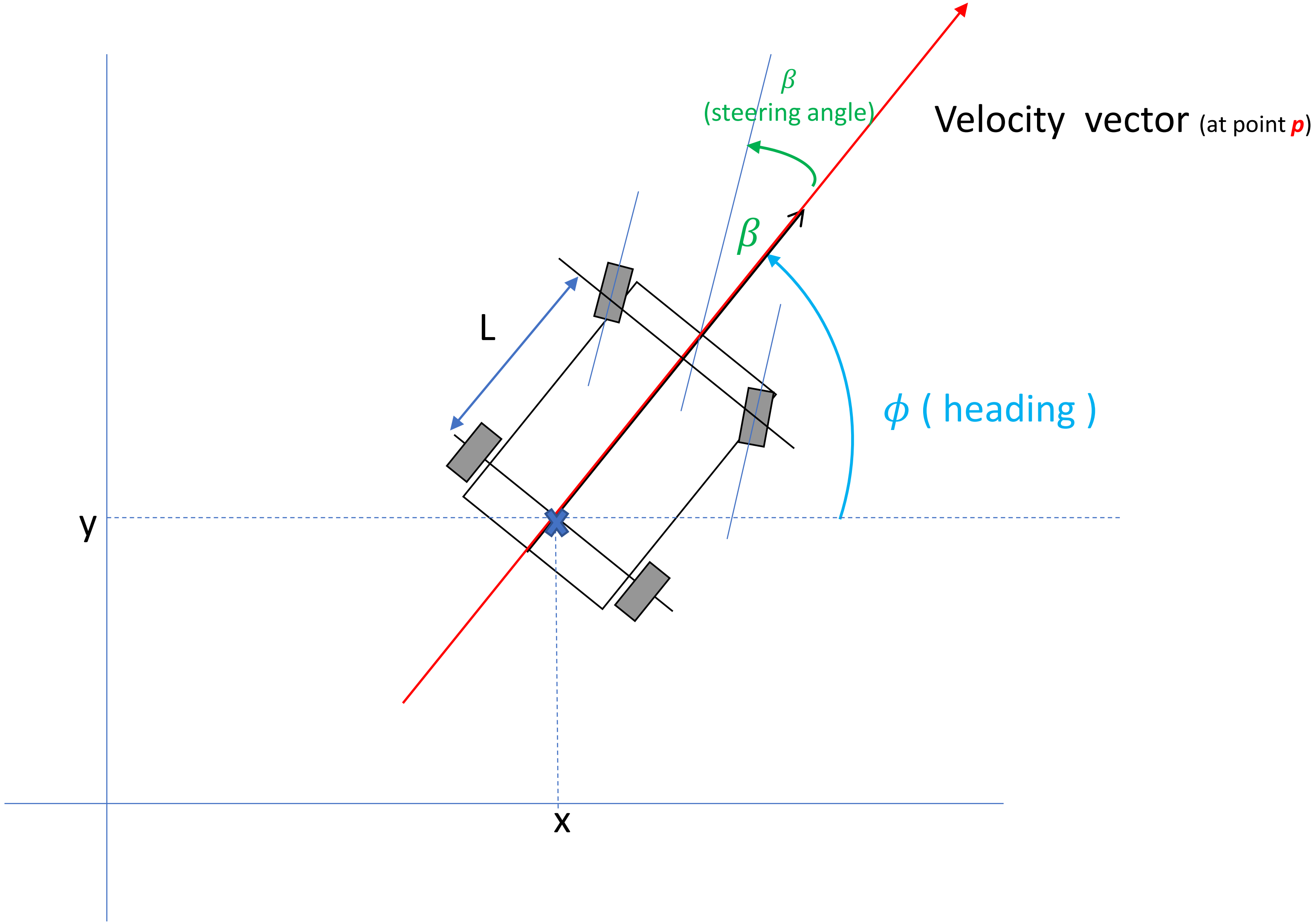
Assumptions: the velocity vector is parallel to the heading (no skidding)

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t)\right)$$

$$\Downarrow$$

$$\mathbf{x}(t) = \begin{bmatrix} x(t) \\ y(t) \\ \phi(t) \end{bmatrix}, \quad \mathbf{u}(t) = \begin{bmatrix} v(t) \\ \beta(t) \end{bmatrix}$$

$$\frac{d\mathbf{x}(t)}{dt} = \frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \\ \phi(t) \end{bmatrix} = \begin{bmatrix} v(t)\cdot\cos(\phi(t)) \\ v(t)\cdot\sin(\phi(t)) \\ \frac{v(t)}{L}\cdot\tan(\beta(t)) \end{bmatrix}$$

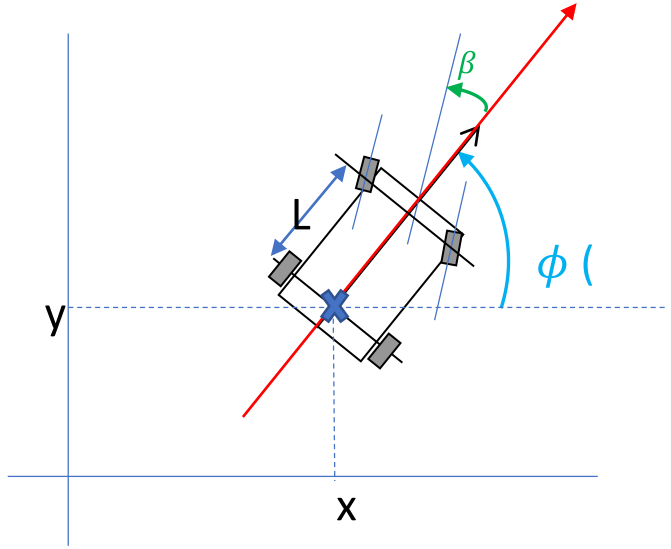


$$\frac{d\mathbf{x}(t)}{dt} = f\left(\mathbf{x}(t), \mathbf{u}(t)\right)$$

$$\Downarrow$$

$$\mathbf{x}(t) = \begin{bmatrix} x(t) \\ y(t) \\ \phi(t) \end{bmatrix}, \quad \mathbf{u}(t) = \begin{bmatrix} v(t) \\ \beta(t) \end{bmatrix}$$

$$\frac{d\mathbf{x}(t)}{dt} = \frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \\ \phi(t) \end{bmatrix} = \begin{bmatrix} v(t)\cdot\cos(\phi(t)) \\ v(t)\cdot\sin(\phi(t)) \\ \frac{v(t)}{L}\cdot\tan(\beta(t)) \end{bmatrix}$$



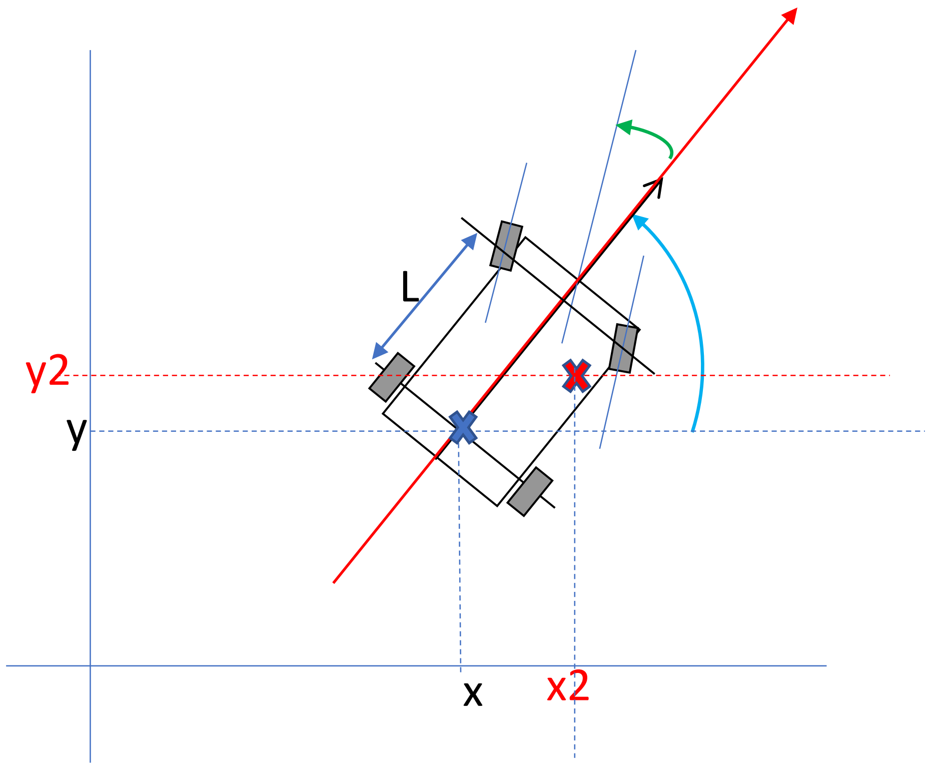
$v(t)$: speed at the center-back of the vehicle, at time t

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

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

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,



$$\mathbf{p}_1 = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \mathbf{p}_2 = \mathbf{p}_1 + \mathbf{R}_\varphi \cdot \begin{bmatrix} a \\ b \end{bmatrix}$$

$$+\mathbf{R}_\varphi = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix}$$

$$x_2 = x_1 + \cos(\varphi) \cdot a - \sin(\varphi) \cdot b$$

$$y_2 = y_1 + \sin(\varphi) \cdot a + \cos(\varphi) \cdot b$$

$$x_2 = x_1 + \cos(\varphi)\cdot a - \sin(\varphi)\cdot b$$

Diversity of Kinematic Models

$$y_2 = y_1 + \sin(\varphi)\cdot a + \cos(\varphi)\cdot b$$

If we evaluate the time derivatives of that point,

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

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

In which

$$\dot{\varphi}(t) = \frac{v(t)}{L}\cdot\tan\left(\beta(t)\right)$$

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

Kinematic Model Based on Angular Rate

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

$$\frac{dx}{dt} = v(t) \cdot \cos(\varphi(t))$$

$$\frac{dy}{dt} = v(t) \cdot \sin(\varphi(t))$$

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

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

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

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

Implementing discrete time versions of the models

$$\frac{dx}{dt} = v(t) \cdot \cos(\varphi(t))$$
$$\frac{dy}{dt} = v(t) \cdot \sin(\varphi(t))$$
$$\frac{d\varphi}{dt} = \frac{v(t)}{L} \cdot \tan\beta(t)$$

$$x(t+\tau) = x(t) + \tau \cdot v(t) \cdot \cos(\varphi(t))$$
$$y(t+\tau) = y(t) + \tau \cdot v(t) \cdot \sin(\varphi(t))$$
$$\varphi(t+\tau) = \varphi(t) + \tau \cdot \frac{v(t)}{L} \cdot \tan\beta(t)$$

Implementing a discrete time versions of the models

$$\frac{dx}{dt} = v(t) \cdot \cos(\varphi(t))$$

$$\frac{dy}{dt} = v(t) \cdot \sin(\varphi(t))$$

$$\frac{d\varphi}{dt} = \frac{v(t)}{L} \cdot \tan\beta(t)$$

$$x(t+\tau) = x(t) + \tau \cdot v(t) \cdot \cos(\varphi(t))$$

$$y(t+\tau) = y(t) + \tau \cdot v(t) \cdot \sin(\varphi(t))$$

$$\varphi(t+\tau) = \varphi(t) + \tau \cdot \frac{v(t)}{L} \cdot \tan\beta(t)$$

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

"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" step (should be a fraction of second, e.g. 0.02sec)
% Tricycle / Ackermann model, discrete version

```
function X = PredictVehiclePose(X0, steering, speed, dt)
        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 ;
end
```

$$x(t+\tau) = x(t) + \tau \cdot v(t) \cdot \cos(\varphi(t))$$

$$y(t+\tau) = y(t) + \tau \cdot v(t) \cdot \sin(\varphi(t))$$

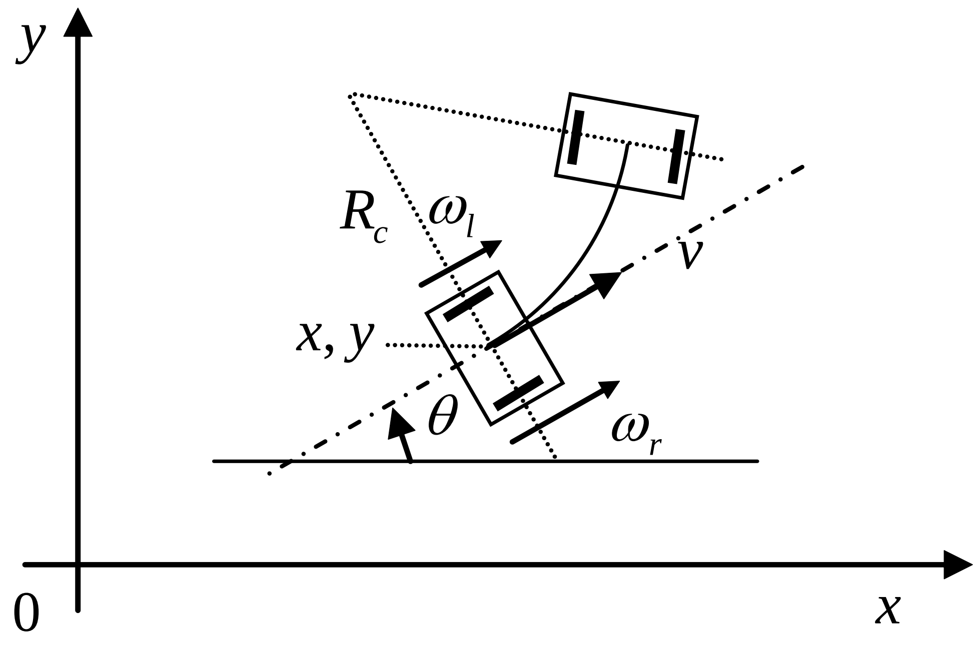
$$\varphi(t+\tau) = \varphi(t) + \tau \cdot \frac{v(t)}{L} \cdot \tan\beta(t)$$

Platforms with Differential Steering

Each wheel has the same radius , and the wheels are separated by a distance L.
The control inputs of the robot are:

Rotational velocity of left wheel: [rad/s]

Rotational velocity of right wheel: [rad/s]



The resulting longitudinal velocity, and heading rate are:

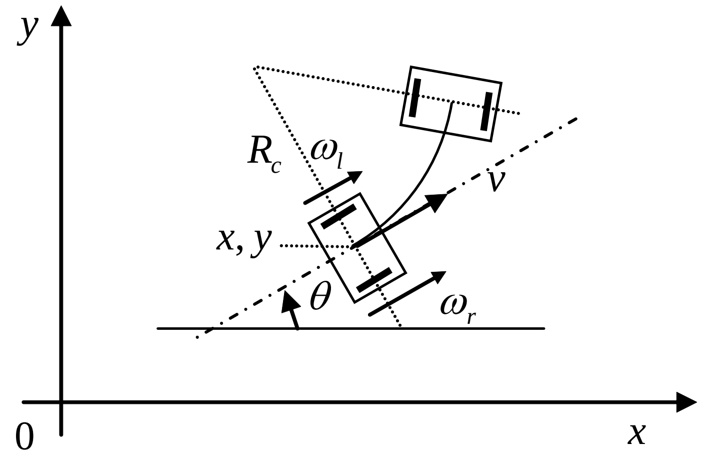
$$v = \frac{R_w}{2}\left(\omega_l + \omega_r\right), \qquad \frac{d\varphi}{dt} = \frac{R_w}{L}\left(\omega_r - \omega_l\right)$$

In which R_w is the wheel radius, and L the transversal separation between both wheels.

Each wheel has the same radius , and the wheels are separated by a distance L.
The control inputs of the robot are:

Rotational velocity of left wheel: [rad/s]

Rotational velocity of right wheel: [rad/s]



The resulting longitudinal velocity, and angular rate are:

$$v = \frac{R_w}{2}\left(\omega_l + \omega_r\right), \qquad \frac{d\varphi}{dt} = \frac{R_w}{L}\left(\omega_r - \omega_l\right)$$

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

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

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

Time to use those models for something.

In the tutorial problems (and in the project) we will need to use a Kinematic.

Let’s see those implementations, for solving the problems.

And the results.

We are already combining multiple concepts (Kinematic model, coordinate frames, IMU, LiDAR measurements)