Chapter 2

Modeling

This sections is devoted to:

1. Different state-space representations of a tracked target.
2. Different motion models for a tracked target.
3. Different sensors and their models

2.1 State-space

Assuming the target's motion on a *two-dimensional tracking plane*, then two different state-space descriptions can be used depending on the expression of the target's *velocity vector*:

1. Cartesian velocity model

$$\mathbf{x} = [x, y, \dot{x}, \dot{y}, \omega]^T$$

2. Polar velocity model

$$\mathbf{x} = [x, y, \upsilon, \psi, \omega]^T$$

where

$$\cancel{\upsilon} = \sqrt{\dot{x}^2 + \dot{y}^2} \quad (2.1a)$$

$$\psi = \arctan\left(\frac{\dot{y}}{\dot{x}}\right) \quad (2.1b)$$

$$\omega = \dot{\psi} \quad (2.1c)$$

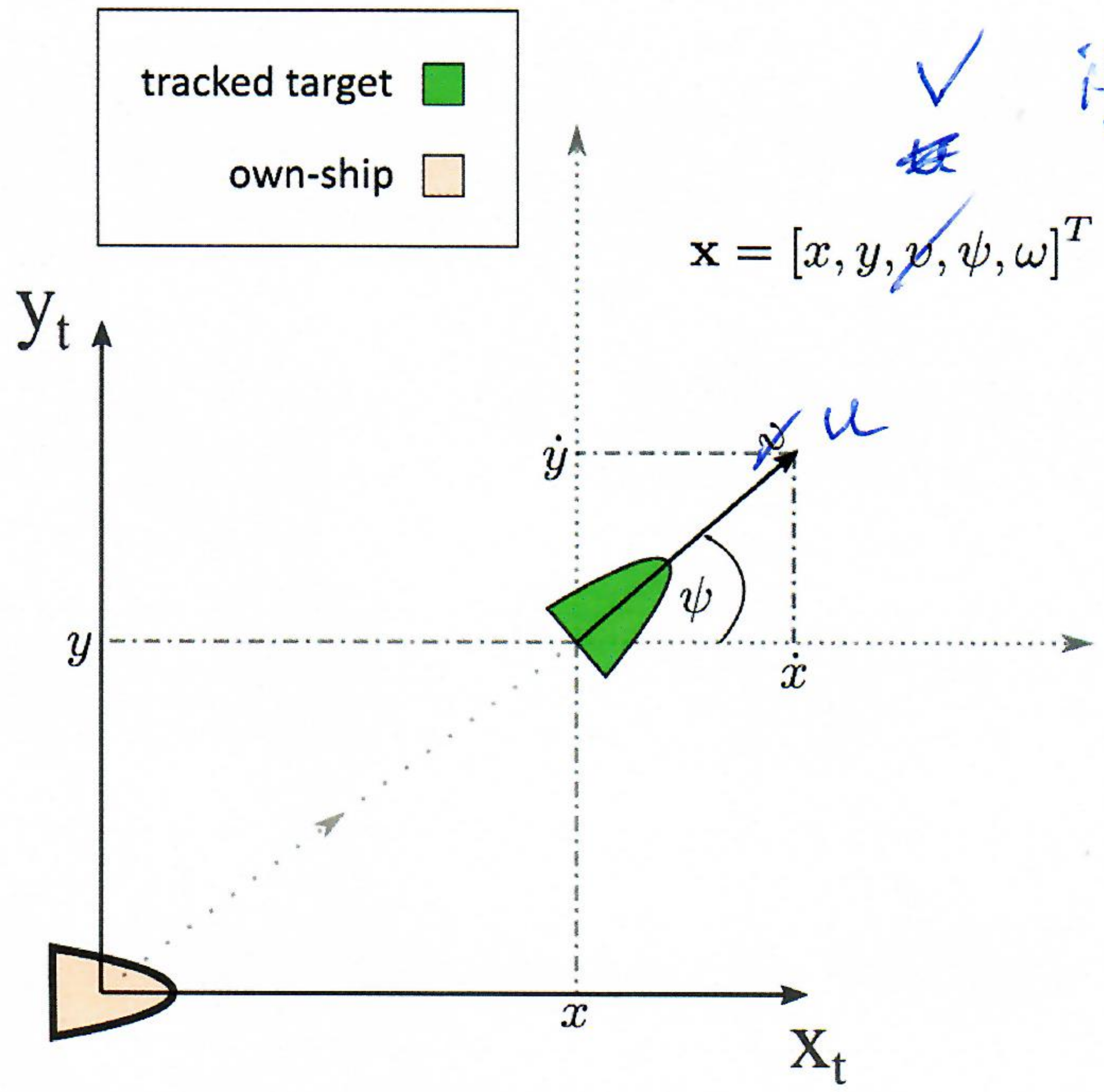
(x, y) : the displacement relative to the local tracking coordinate system

υ : is the target's linear velocity magnitude.

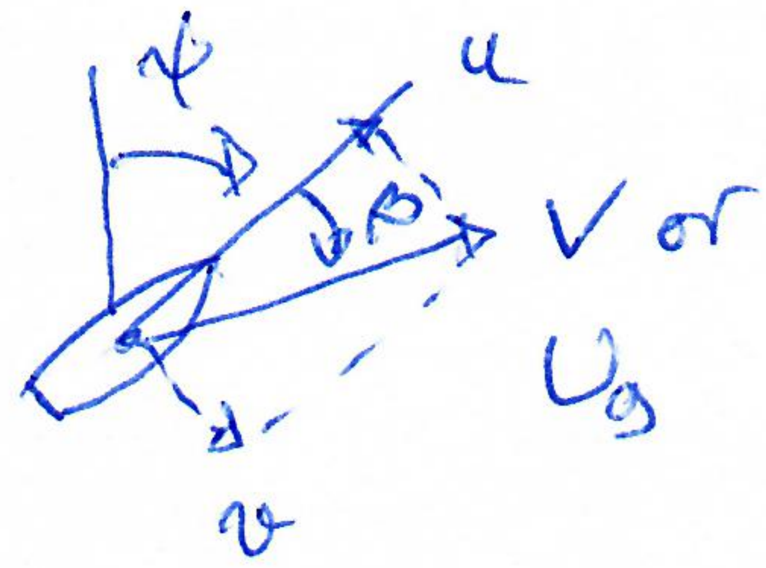
ψ : is the target's velocity vector heading angle w.r.t the tracking system's x-axis.

ω : the target's ~~turn~~ rate. in the horizontal plane

turn rate is r in the body fixed system



V if length of velocity vector
$V = \sqrt{\dot{x}^2 + \dot{y}^2}$



β is drift angle

Figure 2.1: Tracked target state vector top view, in the own-ship's tracking coordinate system

$V = \sqrt{u^2 + v^2}$

2.2.2 Discretisation of the CTRV motion model

Since the system is simulated in discrete time, one can integrate the motion model differential equations in eq. (2.3) within a sample time T and obtain the state transition function vector

V ψ+β no 1

$$\mathbf{x}(t+T) = \mathbf{x}(t) + \int_t^{t+T} (f(\mathbf{x}(\tau) + w(\tau))d\tau \quad \text{and} \quad f(\mathbf{x}(\tau)) = \begin{bmatrix} \cancel{v}cos(\psi) \\ \cancel{v}sin(\psi) \\ 0 \\ \omega \\ 0 \end{bmatrix} \tag{2.4}$$

ψ+β

where

$$\mathbf{x}(\tau) = [x, y, v, \psi, \omega]^T \tag{2.5}$$

is the state-space vector, and

$$w(\tau) \tag{2.6}$$

is the process noise vector

and hence , after ignoring noise, the discrete state transition vector g is

$$\mathbf{x}(t+T) = g\left(\mathbf{x}(t)\right) \tag{2.7}$$

where $g(\mathbf{x}(t))$ by direct integration, for $\mathbf{x}(t) = [x, y, v, \psi, \omega]^T$ is

$$g(\mathbf{x}(t)) = \mathbf{x}(t) + \begin{bmatrix} \frac{2v}{\omega} sin(\omega T) cos(\psi + \frac{\omega T}{2}) \\ -\frac{2v}{\omega} sin(\omega T) cos(\psi + \frac{\omega T}{2}) \\ 0 \\ \omega T \\ 0 \end{bmatrix}, \omega \neq 0 \tag{2.8}$$

or

~~or~~ you could define $\theta = \psi + \beta$
or put $\psi \rightarrow \psi + \beta$

$$g(\mathbf{x}(t)) = \mathbf{x}(t) + \begin{bmatrix} vTcos(\psi) \\ vTsin(\psi) \\ 0 \\ 0 \\ 0 \end{bmatrix}, \omega = 0 \tag{2.9}$$

p with pro-
se input
as well!

2.3 Observation models

An *observation model* or *sensor model* $z_k = h(\mathbf{x_k}) + w_k$ expresses a sensor's observations z_k at time instances t_k given the state vector $\mathbf{x}$. w_k is assumed to be white gaussian noise with covariance matrix that depends on the specific sensor. The selection of the state vector $\mathbf{x}$ can lead to linear or nonlinear observation models.

The author is using the term *observations* in the tracking context to refer to the post processed information derived from raw-sensor *measurements*. In most cases the observations are a byproduct of multiple raw sensor measurements. For instance line of sight sensors such as a laser distance sensor, a radar, create dense point representations around the sensor's origin. These points can be grouped into geometric entities by various feature extraction algorithms, in order to create target *observations*. Likewise, a camera in the tracking context, instead of an imaging device, can be modeled as a relative bearing sensor, by pre-processing the raw image data through an ANN that extracts detected targets pixel locations within the frames. This abstraction level greatly simplifies the measurement models, since one can directly focus on the data-fusion and bypass the observation acquisition process, which at a certain level constitutes a separate and independent task. It is worth mentioning that the author in this thesis chose to focus on a *medium level fusion approach*[6], which means that features(target's position) are fused to obtain features of better quality that can be employed by different tasks.

guate dif-
ensor fusion
rovide a
ction?

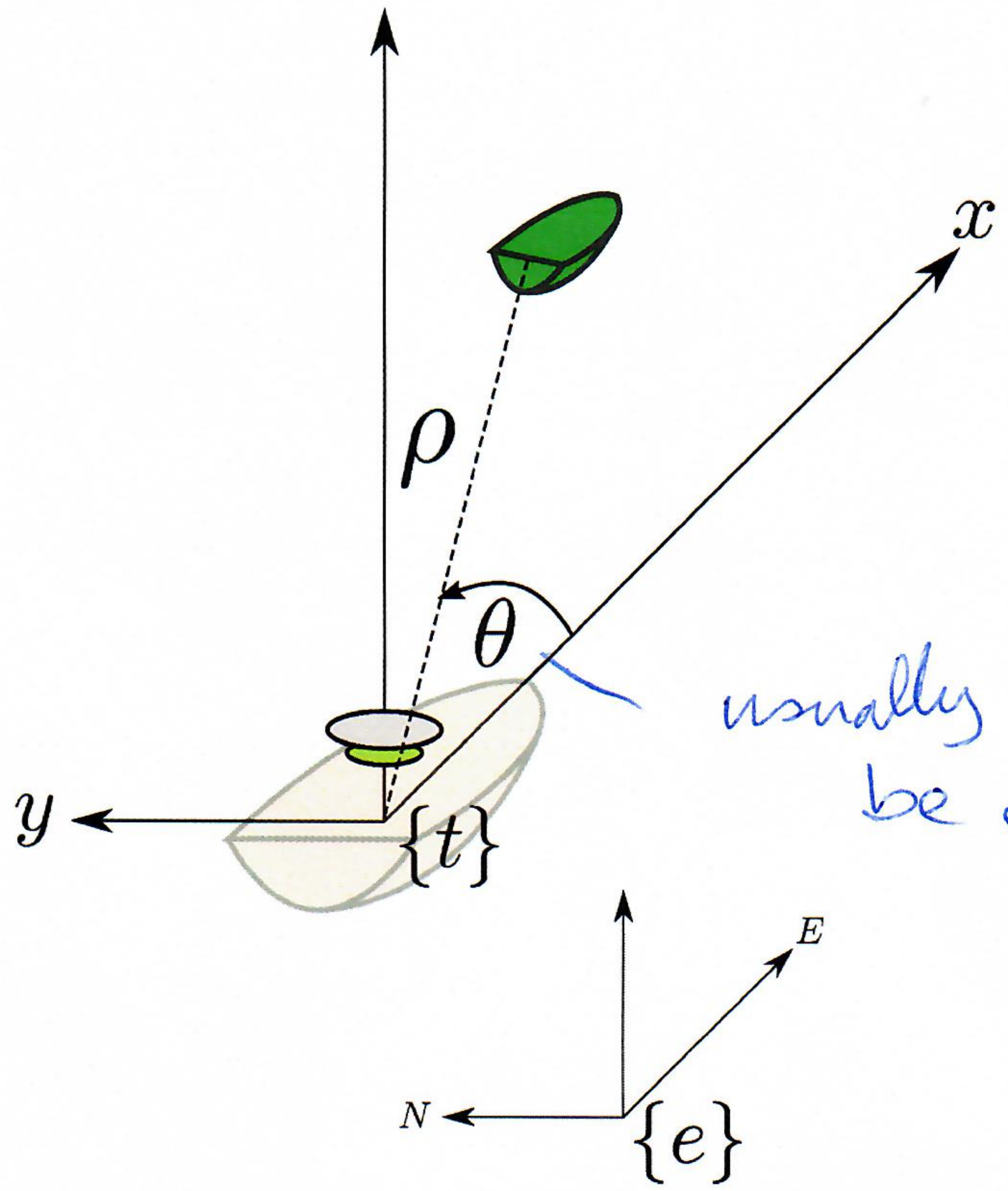
2.3.1 Radar model

dar pre and
cessing clus-
ction, as
ANN detec-
ges

The output of the radar observation model is assumed to be the relative range and bearing measurements of the centroids of clustered raw radar measurements. The model is thus derived by calculating the relative position of a target to the sensor in polar form $z_{radar} = [\rho, \theta]^T$, given the target's position $[x_t, y_t]^T$ w.r.t the sensor's coordinate system.

bearing, usually β

based on
arget and



usually β but just be consistent

Figure 2.2: Radar observation model and different coordinate systems

If the radar's coordinate system, is for simplicity chosen to coincide with a *tracking coordinate system*, attached to the own-ships sea-keeping coordinate system. Then the information about the own-ship is not necessary in the calculations and as such

$$z_{radar}(\mathbf{x}) = \begin{bmatrix} \rho \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{x^2 + y^2} \\ arctan(\frac{y}{x}) \end{bmatrix} + w_{radar} \tag{2.10}$$

where the observation noise w is commonly assumed to be white, zero-mean Gaussian with covariance $R_{radar} = \text{diag}\left[\sigma_\rho^2, \sigma_\theta^2\right]$

A reasonable sizing of the noise variances can be

write part or values select

$\sigma_\rho^2 = 10\,\text{m}$

$\sigma_\theta^2 = 3\frac{\pi}{180}\,\text{rad}$

2.3.2 Converted measurements GPS model

The relative GPS observation model calculates the target's relative position $[x, y]^T$ w.r.t the own-ship's tracking coordinate system, which is directly a part of the target's state vector$\mathbf{x}$.

needs rewriti

1. The own-ship's geodetic coordinates $[\phi_0, \lambda_0]^T$
2. The own-ships sea-keeping heading angle from true north ψ_O
3. The target's measured geodetic coordinates $[\phi, \lambda]^T$ which is usually available at sea through the Automatic Identification System.

If $\mathbf{R}_z(\psi)$ is the Euler's angle transformation for a rotation ψ around the z-axis, then the transformation from *sea-keeping/tracking to ENU*

$$\mathbf{R}_{\textit{sea-keeping}}^{\textit{ENU}}$$

given the own-ship's *heading from true-north* ψ_O is ,

$$R_{\textit{sea-keeping}}^{\textit{ENU}} = R_z(\frac{\pi}{2} - \psi_O) \tag{2.16}$$

$$\begin{bmatrix} dE \\ dN \end{bmatrix} = R_{\textit{sea-keeping}}^{\textit{ENU}} \begin{bmatrix} x \\ y \end{bmatrix}_{\{t\}} \tag{2.17}$$

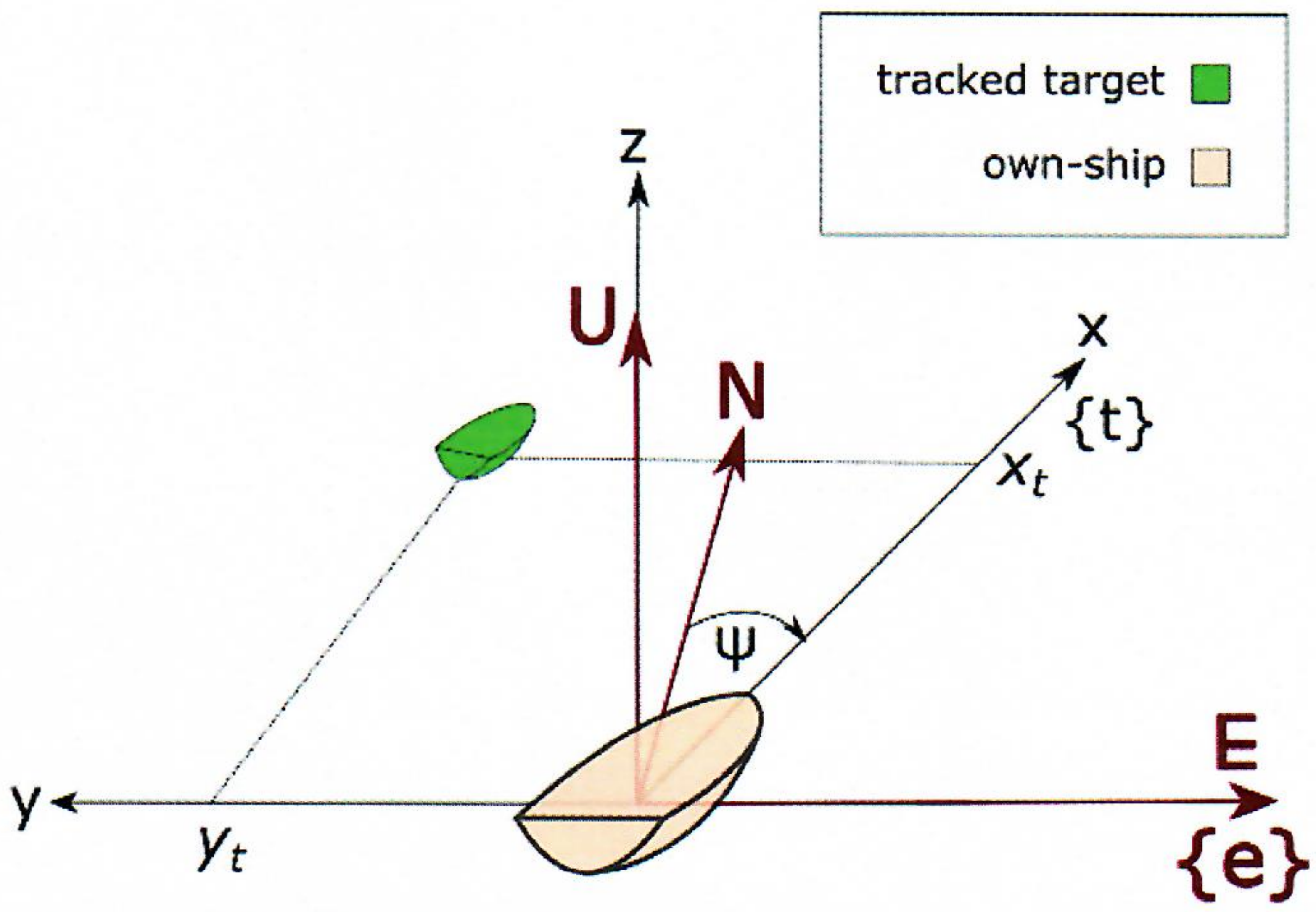


Figure 2.3: Tracking frame and own-ship sensor frame

and thus, by combining the above equations, one can solve for the observed $[\phi, \lambda]^T$,

$$\begin{aligned} \begin{bmatrix} \phi \\ \lambda \end{bmatrix} = \begin{bmatrix} \phi_0 \\ \lambda_0 \end{bmatrix} + \mathbf{T}(\phi_0)^{-1} \begin{bmatrix} dE \\ dN \end{bmatrix} + w_{\textit{AIS}} = \\ \begin{bmatrix} \phi_0 \\ \lambda_0 \end{bmatrix}_{\textit{own-ship}} + \mathbf{T}(\phi_0)^{-1} \mathbf{R}_{\textit{sea-keeping}}^{\textit{ENU}}(\psi_O) \begin{bmatrix} \overset{x}{\cancel{X}}_{\textit{target}} \\ \underset{y}{\cancel{Y}}_{\textit{target}} \end{bmatrix}_{wrt_{\textit{sea-keeping}}} + w_{\textit{AIS}} \end{aligned} \tag{2.18}$$

h

Where

$\mathbf{T}(\phi) = \begin{bmatrix} N(\phi)\cos\phi & 0 \\ 0 & M(\phi) \end{bmatrix}$ is the transformation matrix from *geodetic differences* to *ENU differences.*

$N(\phi) = \frac{\alpha}{\sqrt{1-e^2\sin^2\phi}}$ is the *prime vertical radius ~~of curvature~~*

λ

$M(\phi) = \frac{\alpha(1-e^2)}{(1-e^2\sin^2\phi)^{\frac{3}{2}}}$ is the *meridional radius ~~of curvature~~*

α, β are the equitorial radius(6378.1370 km) and the Polar radius(6356.7523 km) respectively, as derived from the *WGS-84 ellipsoid model.*

[ref to book]

here a and b
previous page α and β ?

$e = \sqrt{1 - \frac{b^2}{a^2}}$ is the ellipsoid's eccentricity.

$h = 0$ since the own-ship is always floating on the geoid.

w_{AIS} is white gaussian zero-mean noise with covariance matrix $\mathbf{R}_{AIS}$ depending on the noise intensity of the AIS source.

The above equation can be rewritten to the standard observation model formulation

$$z_{AIS}(\mathbf{x}, [\phi_0, \lambda_0, \psi_0]^T) = \begin{bmatrix}\phi \\ \lambda\end{bmatrix}_{target} = \begin{bmatrix}\phi_0 \\ \lambda_0\end{bmatrix} + \mathbf{H}\mathbf{x} + w_{AIS} \tag{2.19}$$

, which is a linear observation model that is approximating the geoid with a *spherical-linearization* about the own-ship's geodetic location.

2.3.4 Camera observation model

Assuming a camera sensor mounted on a mast with the camera z-axis aligned with the *own-ship's body coordinate system*

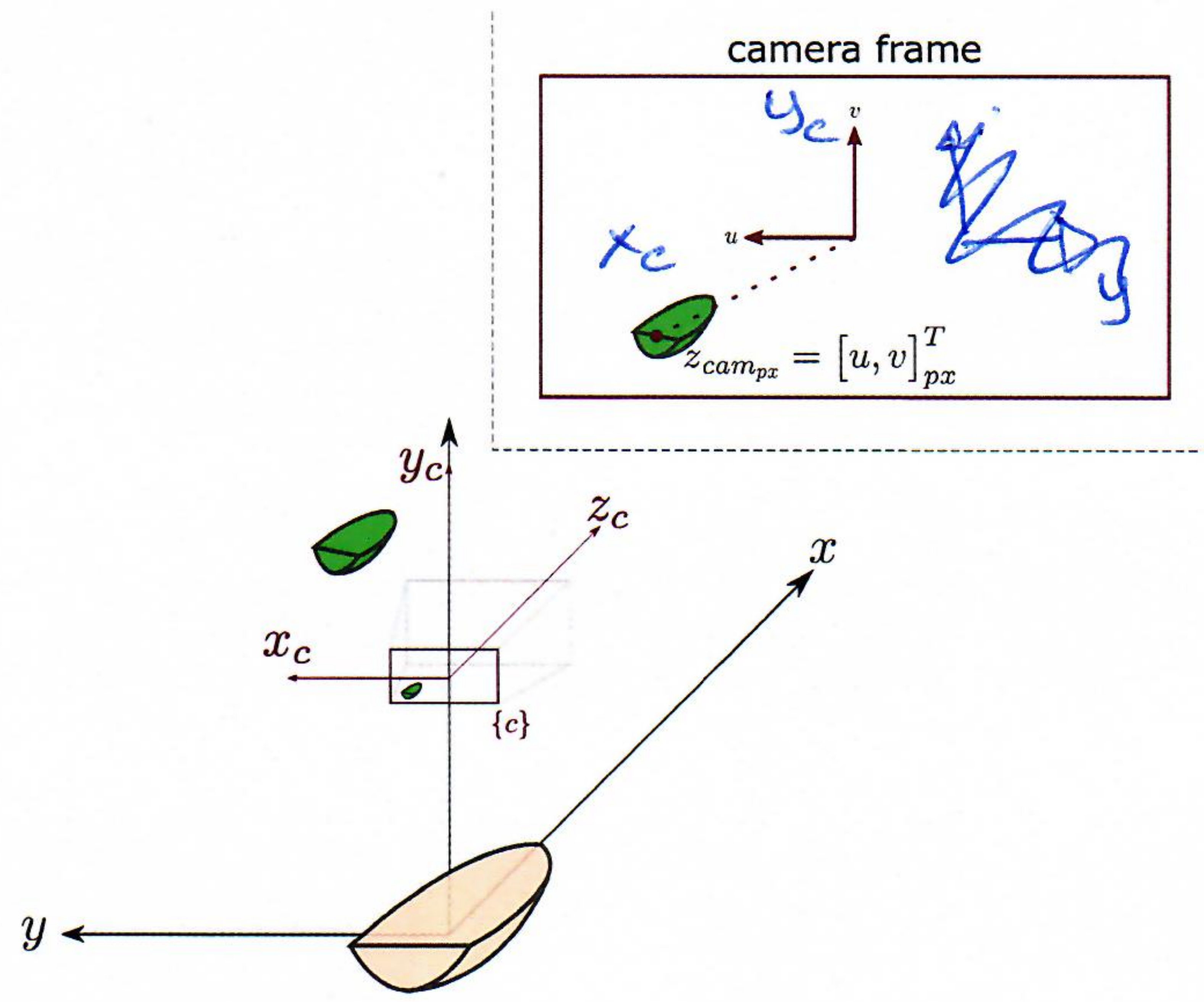


Figure 2.4: Camera alignment with the *tracking* frame

u and v are reserved for body velocities you could use x_c and y_c as in your nice figure

A top-view of the pinhole model is illustrated in fig. 2.5.

K
is the *intrinsic camera calibration* matrix

P_x were x_c
P_y y_c
on previous pages

$$\mathbf{K} = \begin{bmatrix} fm_x & 0 & p_x \\ 0 & fm_y & p_y \\ 0 & 0 & z \end{bmatrix}$$

with

- f is the intrinsic camera calibration parameter, focal length in mm - m_x, m_y are the camera's sensor pixel density in $\frac{\text{number of pixels}}{mm}$ across the two different directions of the image plane. - p_x, p_y correspond to the position of the principal point of the image in *pixel coordinates* - w_{camera} is white Gaussian zero mean noise with variance related to the classifier accuracy, or any calibration errors in the extrinsic and intrinsic camera matrices.

$\mathbf{R}^{\text{camera}}_{\text{own-ship}}$
is the *extrinsic camera rotation matrix* that describes the *orientation of the camera coordinate system* w.r.t. the *own-ship*.

C
is the *extrinsic origin of the camera coordinate system* w.r.t. the *own-ship*.

$\mathbf{R}^{\text{own-ship}}_{\text{tracking}}$
Is the transformation matrix that corresponds to the *roll-pitch* rotation of the *own-ship's body coordinate system* w.r.t the *tracking* coordinate system.. Please note that the order of the rotations matters.

$$R^{\textit{own-ship}}_{\textit{tracking}} = (R_{\textit{pitch}} R_{\textit{roll}})^T \tag{2.23}$$

Where

$$R_{\textit{pitch}}, R_{\textit{roll}}$$

are the standard rotation matrices for Euler's angles $z_{\textit{gyro}} = (\alpha, \beta)^T = (\textit{roll}, \textit{pitch})^T$.

h

$$R_{\textit{roll}} = R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \tag{2.24}$$

$\alpha \to \phi$

roll = ϕ (Warphi)
pitch = Θ

and

$$R_{\textit{pitch}} = R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \tag{2.25}$$

$\beta \Rightarrow \Theta$

A basic assumption at this point, is that the own-ship is able to measure the angles (α, β) of the *own-ship body coordinate system* w.r.t the *tracking coordinate system*. These angles are usually available on a ship by using one or multiple *gyro sensors*.

h

yro sensors