

# ELEC 3224 — Guidance Systems

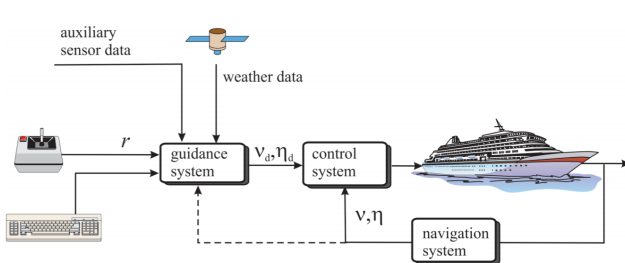
Professor Eric Rogers

School of Electronics and Computer Science  
University of Southampton  
etar@ecs.soton.ac.uk  
Office: Building 1, Room 2037

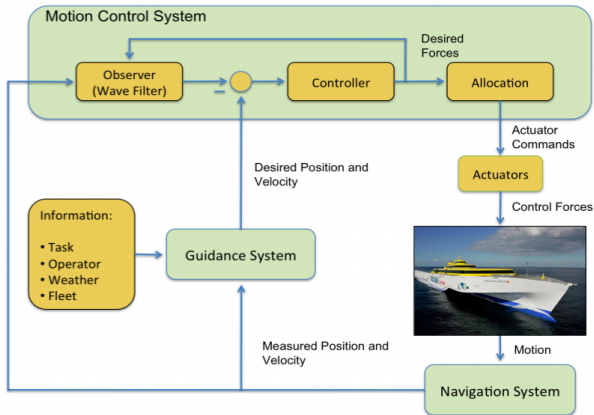
# Introduction

- ▶ Guidance and control systems usually composed of:
- ▶ **An attitude control system.**
- ▶ **A path-following control system.**
- ▶ **Basic attitude control system (marine example but extends to other cases (with appropriate changes) —**  
heading autopilot while roll and pitch are regulated to zero or left uncontrolled.
- ▶ The path-following controller is tasked with keeping the craft on the prescribed path with some pre-defined dynamics, e.g., a speed control system.

# Introduction



# Introduction



# Introduction

- ▶ Guidance is the action or the system that continuously computes the reference (desired) position, velocity and attitude (PVA) of the vehicle to be used by the control system.
- ▶ Charles Stark Draper (widely recognised as the father of inertial navigation) stated: 'Guidance depends upon fundamental principles and involves devices that are similar for vehicles moving on land, on water, under water, in air, beyond the atmosphere, within the gravitational field of the earth and in space outside this field.' (<http://www.draper.com/>).

# Motion Control Problems

- ▶ **Setpoint regulation (point stabilisation):** a special case where the desired position and attitude are chosen to be constant.
- ▶ **Trajectory Tracking:** — force the system output  $y(t) \in \mathbb{R}^m$  to track a desired output  $y_d(t) \in \mathbb{R}^m$ . The desired trajectory can be computed using: i) using reference models generated by low-pass filters, ii) optimisation methods or iii) simulating the motion of the vehicle using an ‘adequate model’ of the dynamics.
- ▶ **Path Following:** — follow a pre-defined path independent of time.

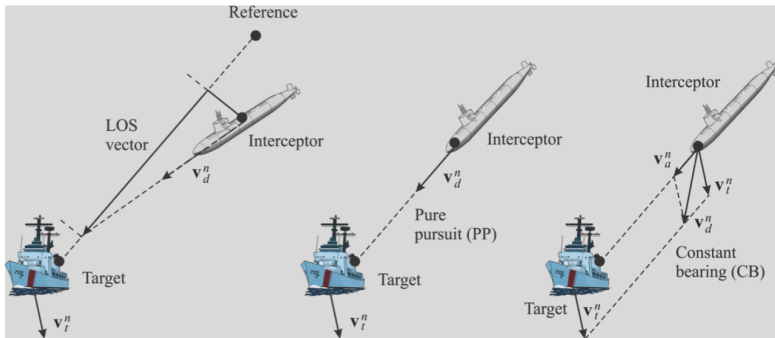
# Motion Control Problems

- ▶ In the last case there are **no temporal constraints but spatial constraints can be added to represent obstacles and other positional constraints if they are known in advance.**
- ▶ Guided missiles (2nd World War) — research in this area has been in existence as long as there has been research on control theory.
- ▶ A guide missile is a 'space-traversing unmanned vehicle that carries within itself the means of controlling its flight path.'
- ▶ It is possible to distinguish between maneuvering (accelerating) and non-maneuvering (non-accelerating).
- ▶ This area is known as **Target Tracking (Velocity Control).**

# Target Tracking (Velocity Control)

- ▶ Three terminal guidance strategies are:
- ▶ **Line-of-Sight (LOS) Guidance.**
- ▶ **Pure Pursuit (PP) Guidance.**
- ▶ **Constant Bearing (CB) Guidance.**
- ▶ See the next figure.

# Target Tracking (Velocity Control)



# Target Tracking (Velocity Control)

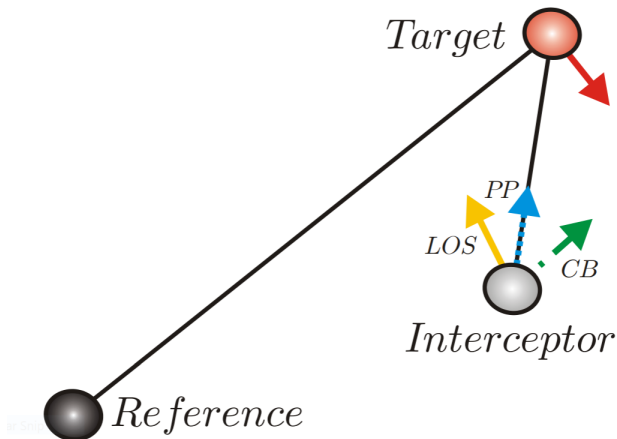
- ▶ The control objective of a target-tracking scenario can be formulated as

$$\lim_{t \rightarrow \infty} (p^n(t) - p_t^n(t)) = 0$$

- ▶ where  $p_t^n = [N_t, E_t] \in \mathbb{R}^2$  is the 2-D position of the target in North-East co-ordinates.
- ▶  $p_t^n(t)$  represents either a **stationary point** or a **vehicle moving at a non-zero and bounded NED (North East Direction) velocity**:

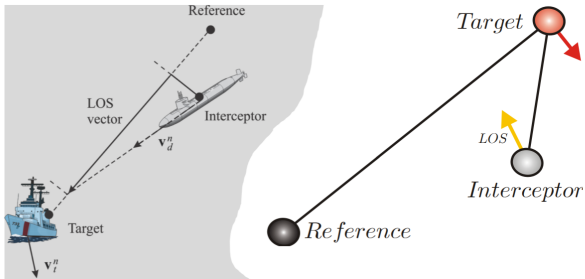
$$v_t^n(t) = \dot{p}_t^n(t) \in \mathbb{R}^2$$

# Target Tracking (Velocity Control)



**The Big Picture.**

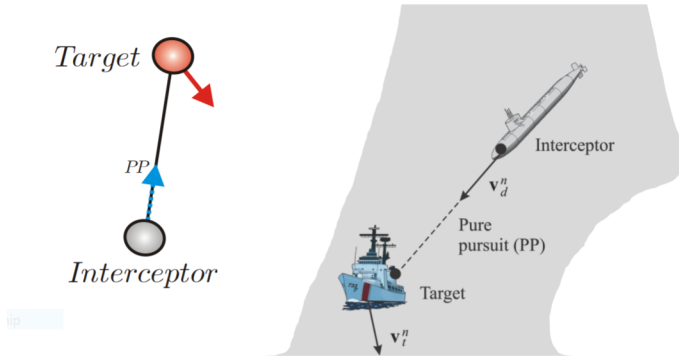
# LOS Guidance (Velocity Control)



# LOS Guidance (Velocity Control)

- ▶ 3-point guidance scheme.
- ▶ **The interceptor must constrain its motion along the reference-target line of sight.**
- ▶ Commonly used in surface-to-air missiles (bean-rider guidance).
- ▶ The interceptor (approach) velocity  $v_a^n$  is pointed to the LOS vector to obtain desired velocity to obtain the desired velocity  $v_d^n$ .

# Pure Pursuit (PP) Guidance (Velocity Control)



lip

# Pure Pursuit (PP) Guidance (Velocity Control)

- ▶ 2-point guidance.
- ▶ The interceptor must align its linear velocity along the interceptor-target line of sight.
- ▶ Equivalent to a predator chasing a prey in the animal world.
- ▶ Typically employed in air-to-surface missiles.
- ▶ Deviated pursuit guidance is a variant of PP (also termed fixed-lead guidance).

# Pure Pursuit (PP) Guidance (Velocity Control)

- ▶ The interceptor aligns its velocity  $v_a^n$  along the LOS vector between the interceptor and target by choosing the desired velocity as follows.

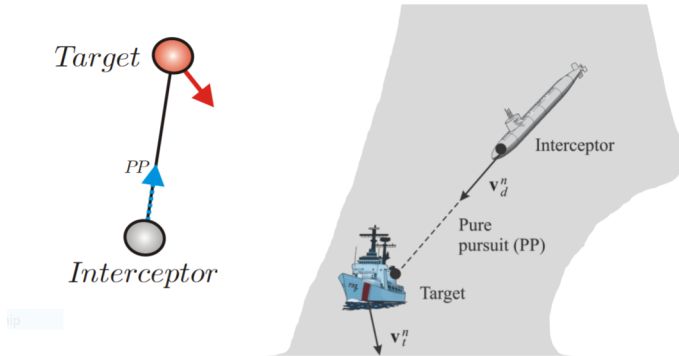


$$v_d^n = -K \frac{\tilde{p}_n}{||\tilde{p}_n||}$$



$$\tilde{p}_n = p^n - p_t^n$$

# Constant Bearing (CB) Guidance (Velocity Control)

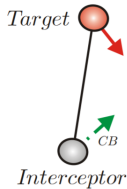


lip

# Constant Bearing (CB) Guidance (Velocity Control)

- ▶ 2-point guidance scheme — also known as **parallel navigation**.
- ▶ Major use — air-to-air missiles.
- ▶ Centuries old — used to avoid collisions at sea.
- ▶ **Proportional navigation** is the most common implementation.
- ▶ The interceptor aligns the interceptor-target velocity  $v_a^n$  along the LOS vector between the interceptor and the target.

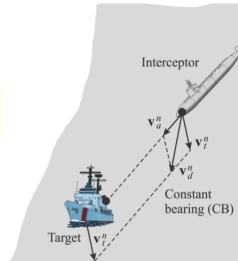
# Constant Bearing (CB) Guidance (Velocity Control)



$$\mathbf{v}_d^n = \mathbf{v}_t^n + \mathbf{v}_a^n$$

$$\mathbf{v}_a^n = -\kappa \frac{\tilde{\mathbf{p}}^n}{\|\tilde{\mathbf{p}}^n\|}$$

$$\tilde{\mathbf{p}}^n := \mathbf{p}^n - \mathbf{p}_t^n$$



# Trajectory Tracking (Position Control)

- ▶ Tracking of a time-varying reference trajectory in 3 DOF (surge, sway and yaw in a marine application) is achieved by minimising the **tracking error**:

$$e(t) = \begin{bmatrix} N(t) - N_d(t) \\ E(t) - E_d(t) \\ \psi(t) - \psi_d(t) \end{bmatrix}$$

- ▶ **Classification of trajectory-tracking control laws is by the number of available actuators.**

# Trajectory Tracking (Position Control)

- ▶ **Three or more controls:** known as fully actuated dynamic positioning — typical (marine) applications are crab-wise motions (low-speed maneuvering) and **station keeping**, where the objective is to achieve  $e(t) \rightarrow 0$ .
- ▶ **Two controls and trajectory tracking:** Trajectory-tracking in 3 DOF,  $e(t) \in \mathbb{R}^3$  with only two controls  $u(t) \in \mathbb{R}^2$ .
- ▶ This is an **under actuated control problem that cannot be solved using linear theory/design.**

# Trajectory Tracking (Position Control)

- ▶ **Two controls and path following:** a standard approach is to define a 2-D workspace — along-track and cross-track errors – and minimise using a LOS path following controller.
- ▶ This means it is possible to follow a path using only two controls.
- ▶ Since the input and output vectors are of dimension 2 the 6-DOF system model must be stable.
- ▶ **One control:** It is not possible to design station keeping and trajectory-tracking control systems in 3-DOF using only one control.

# Reference Models

- ▶ Simplest form is a low-pass filter

$$\frac{x_d}{r}(s) = h_{lp}(s) = \frac{b}{s + a}$$

where  $r$  is the command and  $x_d$  is the desired state.

- ▶ **How to chose the filter?**
- ▶ The filter should be chosen to reflect the dynamics of vehicle feasible trajectory: speed and acceleration limitations.
- ▶ Also the bandwidth of the reference model should be less than that of the vehicle control system.

# Reference Models

- ▶ In some cases, e.g., marine vehicles, reference models motivated by the dynamics of spring-mass-damper systems are used

$$h_{lp}(s) = \frac{\omega_{n_i}^2}{s^2 + 2\zeta_i\omega_{n_i}s + \omega_{n_i}^2}$$

- ▶  $\zeta_i$  — relative damping ratio  $i = 1, 2, \dots, n$
- ▶  $\omega_{n_i}$  — natural frequency  $i = 1, 2, \dots, n$
- ▶ MIMO model

$$M_d\ddot{\eta}_d + D_d\dot{\eta}_d + G_d\eta_d = G_dr$$

- ▶  $M_d, D_d$  and  $G_d$  – positive design matrices.

# Reference Models

- ▶ State-space model

$$\dot{X}_d = A_d X + B_d r$$

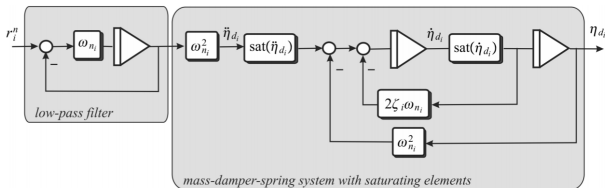


$$A_d = \begin{bmatrix} 0 & I \\ -M_d^{-1}G_d & -M_d^{-1}D_d \end{bmatrix}, B_d = \begin{bmatrix} 0 \\ M_d^{-1}G_d \end{bmatrix}, C_d = \begin{bmatrix} I & 0 \end{bmatrix}$$

# Reference Models

- ▶ A drawback with linear reference models is that the time constants in the model often give satisfactory response for one operating point of the system but the response for the others may result in a completely different behavior.
- ▶ This is due to the exponential convergence of the signals in a linear system.
- ▶ The performance of a linear reference model can be improved by including saturation elements for velocity and acceleration.
- ▶ See the next figure.

# Reference Models



►  $v_i \leq v_i^{max}, \dot{v}_i \leq \dot{v}_i^{max}$

$$\text{sat}(x) = \begin{cases} \text{sgn}(x)x_{max} & \text{if } |x| \geq x_{max} \\ x & \text{elsewhere} \end{cases}$$

# Optimal Trajectory Generation

- ▶ Optimization methods can be used for trajectory and path generation.
- ▶ Provides a systematic method for the inclusion of **static and dynamic** constraints.
- ▶ This, however, come at the cost that an optimisation problem **must be solved online** to generate a feasible time-varying trajectory.
- ▶ Solution can be by linear programming (LP), quadratic programming (QP) and nonlinear methods — all require a solver that can be implemented in the program.

# Optimal Trajectory Generation

$$J = \min_{\eta_d, v_d} \{\text{power, time}\}$$

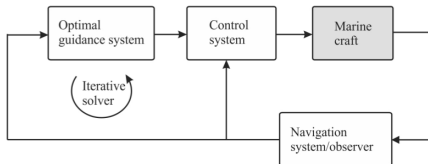
$|U| \leq U_{\max}$  (max speed)

$|r| \leq r_{\max}$  (max turning rate)

$|u_i| \leq u_{i,\max}$  (saturating limit of control  $u_i$ )

$|\dot{u}_i| \leq \dot{u}_{i,\max}$  (saturating limit of rate  $\dot{u}_i$ )

etc.



# Path Following for Straight-Line Paths

- ▶ A trajectory describes the motion of a moving object through space as a function of time – object can be a craft, projectile or a satellite etc.
- ▶ A trajectory can be described **mathematically either by the geometry of the path or as the position of the object over time.**
- ▶ Path following is the task of following a predefined path independent of time, i.e., no temporal constraints
- ▶ Spatial constraints can, however, be added for, eg., obstacle avoidance.

# Path Following for Straight-Line Paths

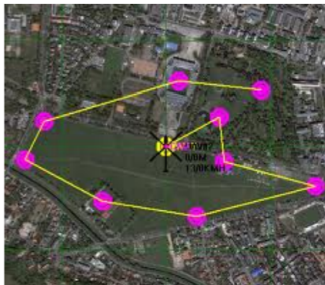
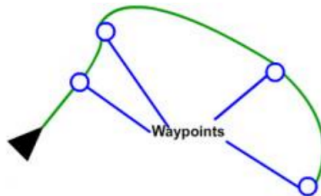
- ▶ LOS guidance can be applied for path control.
- ▶ A LOS vector from the vehicle to the next waypoint or a point on the path (straight line) between two waypoints can be used for both course and heading control.
- ▶ In some cases a heading autopilot is available.
- ▶ In this last case, the angle between the LOS vector and the prescribed path can be used as a set-point for the heading autopilot. This will force the vehicle to track the path.

# Optimal Trajectory Generation



- ▶ The waypoints are stored in the database and used to generate the trajectory or path for the moving vehicle to follow.
- ▶ Both trajectory-tracking and path-following control systems can be designed for this purpose.

# Optimal Trajectory Generation



# Waypoint based Path Generation

- ▶ **Mission:** The vehicle should move from a starting point  $(x_0, y_0, z_0)$  to the terminal point  $(x_n, y_n, z_n)$  via the waypoints  $(x_i, y_i, z_i)$ .
- ▶ **Environmental data:** e.g., avoid bad weather for energy optimal routing or safety reasons.
- ▶ **Geographical data:** e.g., for the marine case shallow water etc.
- ▶ **Obstacles:** to be avoided.
- ▶ **Collision avoidance:** avoid moving vehicles close to your own route by introducing **safety margins**.

# Waypoint based Path Generation

- ▶ **Feasibility: each waypoint must be feasible**, i.e., it must be such that it is possible to maneuver to the next waypoint without exceeding the maximum speed and other requirements, e.g. turning rate.
- ▶ The route for a vehicle is usually specified in terms of waypoints. Each waypoint is defined using Cartesian co-ordinates.
- ▶ The waypoint database is therefore composed of



$$\text{wpt.pos} = \{(x_0, y_0, z_0), (x_1, y_1, z_1), \dots, (x_n, y_n, z_n)\}$$

# Waypoint based Path Generation

- ▶ together with other properties such as

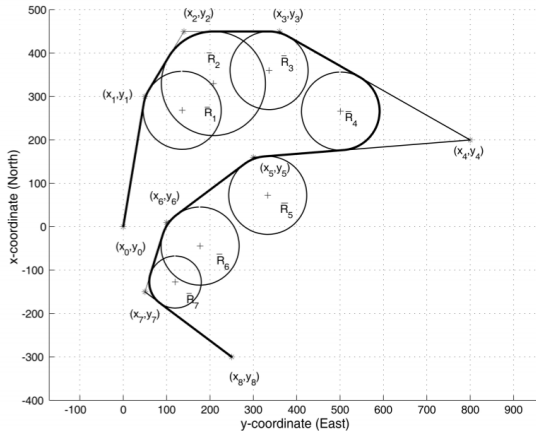
$$\text{wpt.speed} = \{U_0, U(1), \dots, U_n\}$$



$$\text{wpt.heading} = \{\psi_0, \psi_1, \dots, \psi_n\}$$

- ▶ The three states  $\{x_i, y_i, \psi_i\}$  are also termed the **pose** and they **describe the start and end configurations of the vehicle given by two waypoints.**

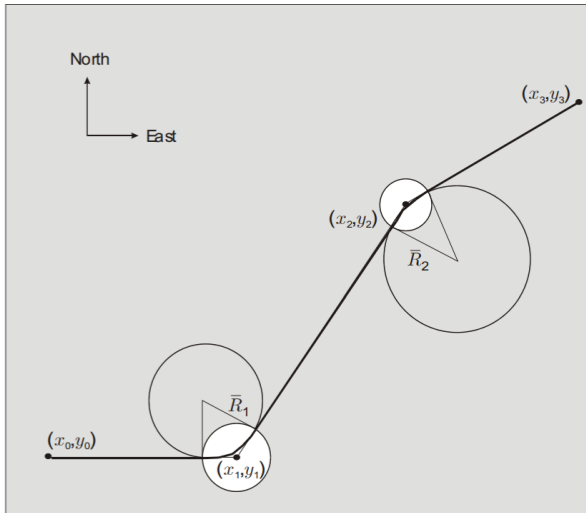
# Waypoint based Path Generation



# Waypoint based Path Generation

- ▶ In application, the desired path is often represented using **straight lines** and **circle arcs** to connect the waypoints — known as a **Dubins path**.
- ▶ **Dubins (1957)** — ‘The shortest path (minimum time) between two configurations  $(x, y, \psi)$  of a vehicle travelling at constant speed  $U$  is a path formed by straight lines and circular arc segments.’

# Waypoint based Path Generation



# Waypoint based Path Generation

- ▶ It is also possible to use other interpolations strategies, e.g., cubic.
- ▶ A drawback of the approach above against a cubic strategy is that jumps in the yaw rate  $r_d$ .
- ▶ The desired ya rate along the straight line is  $r_d = 0$  and  $r_d = \text{constant}$  on the circular arc during steady turning.
- ▶ Hence there will be a **jump in the desired yaw rate during transition from the straight line to the circular arc.**



# Waypoint based Path Generation

- ▶ The human operator typically specifies a circle of radius  $R_i$  around each waypoint

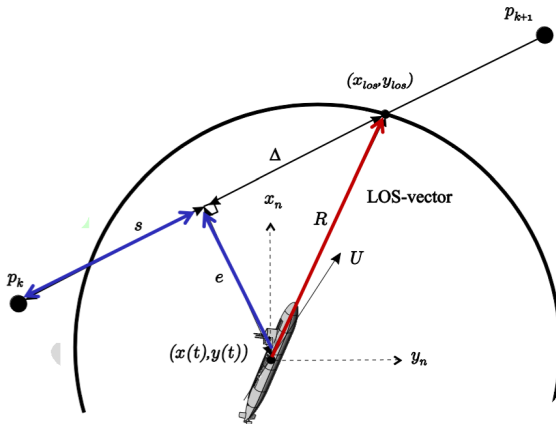
$$\text{wpt.radius} = \{R_0, R_1, \dots, R_n\}$$

- ▶ The turning point is where the circle arc intersects the turning point of the vehicle.
- ▶ The radius of the enscribed circle can be computed from  $R_i$  as

$$\overline{R}_i = R_i \tan \alpha_i$$

where  $\alpha_i$  is given in the previous figure.

# Course Control



# Course Control

- ▶ Consider a straight-line path implicitly defined by two waypoints:

$$\begin{aligned} p_k^n &= \begin{bmatrix} x_k & y_k \end{bmatrix}^T \\ p_{k+1}^n &= \begin{bmatrix} x_{k+1} & y_{k+1} \end{bmatrix}^T \end{aligned}$$

- ▶ Speed

$$U(t) := |v| = \sqrt{\dot{x}^2(t) + \dot{y}^2(t)} \geq 0$$

# Course Control

- ▶ Course angle

$$\xi(t) := \tan^{-1} 2(\dot{y}(t), \dot{x}(t)) \in S := [-\pi, \pi]$$

- ▶ Angle of the straight line w.r.t NED (North East Direction)

$$\alpha_k := \tan^{-1} 2(y_{k+1} - y_k, x_{k+1} - x_k) \in S$$

- ▶ Tracking errors

$$\epsilon(t) = R_p(\alpha_k)^T (p^n(t) - P_k^n)$$

# Course Control



$$R_p(\alpha_k) := \begin{bmatrix} \cos(\alpha_k) & -\sin(\alpha_k) \\ \sin(\alpha_k) & \cos(\alpha_k) \end{bmatrix} \in SO(2)$$

- ▶  $SO(2)$  — group of  $2 \times 2$  orthogonal (inverse is equal to the transpose) matrices.



$$\epsilon(t) = \begin{bmatrix} s(t) & e(t) \end{bmatrix}^T \in \mathbb{R}^2$$

- ▶  $s(t)$  — along-track distance (tangential to path).
- ▶  $e(t)$  = cross-track error (normal to path).