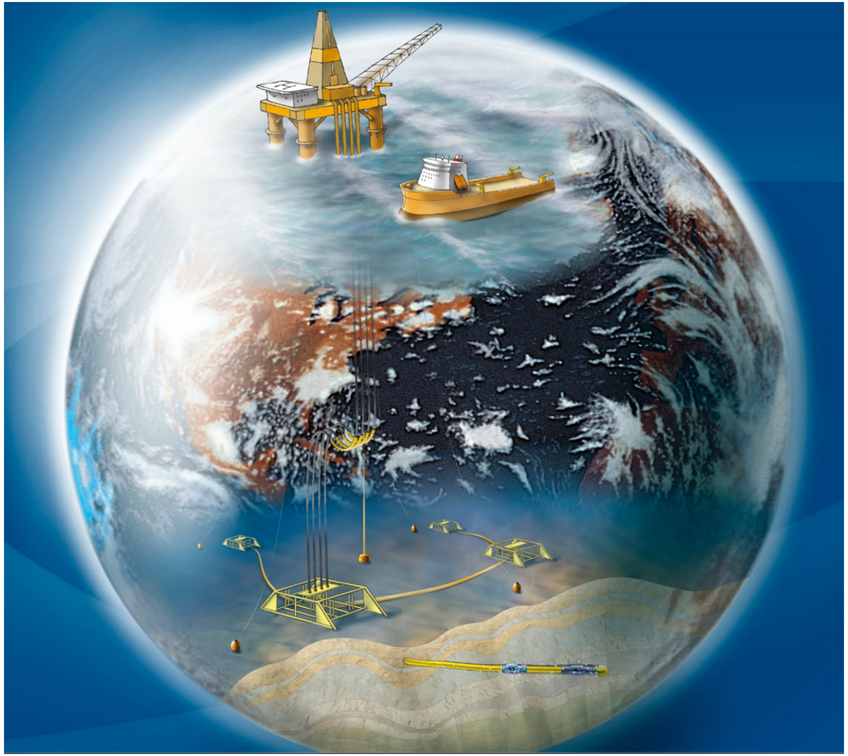
Chapter 11 – Introduction to Part II

11.1 Guidance, Navigation and Control Systems
11.2 Control Allocation

When implementing motion control systems, it is important to have a good software architecture to simplify software updates and maintenance. Another important aspect is cybersecurity, which puts constraints on the system architecture.

Because of this, the motion control system is usually constructed as three independent blocks denoted as the guidance, navigation and control (GNC) systems.



Chapter Goals

GNC Architecture:

- Understand the meaning of the G, N and C blocks in a GNC architecture for autonomous vehicles and marine craft.
- Cybersecurity: Understand that GNC architectures and signal transmissions between the G, N and C blocks are vulnerable for cyber threats. Solution: encrypt the sensor and control signals.

Control Allocation:

- Understand how control forces computed by feedback control systems are distributed as actuator commands using constrained and unconstrained control allocation methods.
- Be able to express forces from azimuth thrusters, tunnel thrusters, propellers, control surfaces, etc. in terms of the (extended) thrust configuration matrix **T** and force coefficient matrix **K**.
- Be able to design an unconstrained control allocation algorithm using the generalized inverse.
- Understand the principles for constrained control allocation and how dynamic programming can be used to solve these problems.

11.1 Guidance, Navigation and Control

Guidance *is the action or the system that continuously computes the reference (desired) position, velocity and attitude of a marine craft to be used by the motion control system. These data are usually provided to the human operator and the navigation system.*

Navigation *is the science of directing a craft by determining its position/attitude, course and distance traveled. In some cases velocity and acceleration are determined as well.*

Control, *or more specifically motion control and control allocation, is the action of determining the necessary control forces and moments to be provided by the craft in order to satisfy a certain control objective.*

11.1 Guidance, Navigation and Control

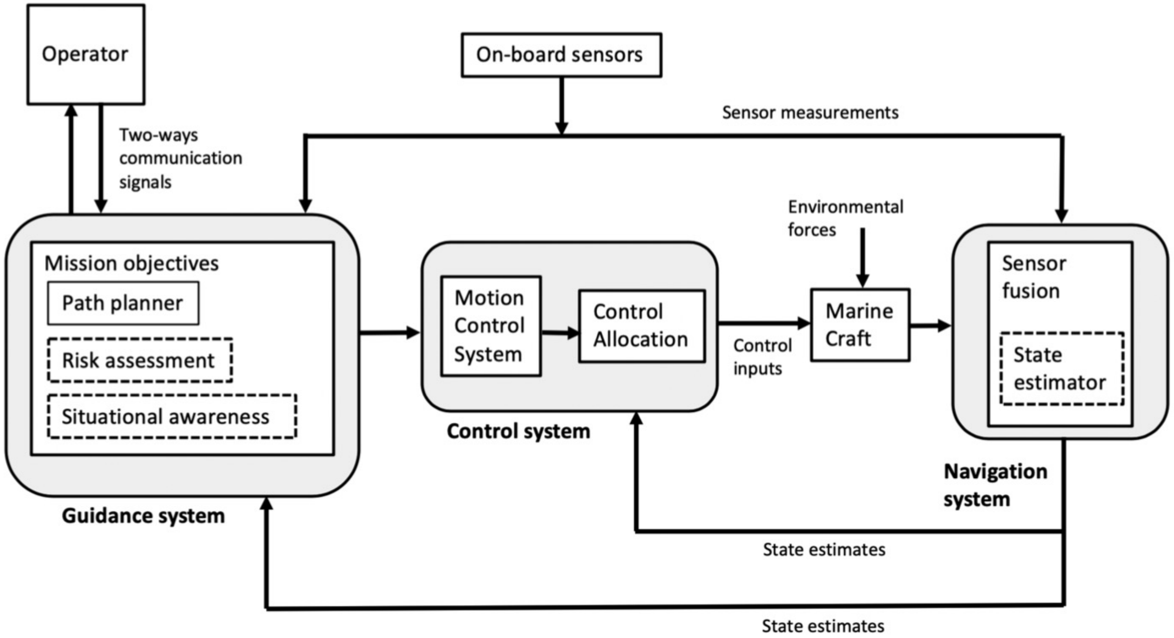
From Wikipedia, the free encyclopedia

Guidance, navigation and control (GNC) is a branch of engineering dealing with the design of systems to control the movement of vehicles, especially, automobiles, ships, and spacecraft, aircraft

In many cases these functions can be performed by trained humans. However, because of the speed of, for example, a rocket's dynamics, human reaction time is too slow to control this movement. Therefore, systems—now almost exclusively digital electronic—are used for such control. Even in cases where humans can perform these functions, it is often the case that GNC systems provide benefits such as alleviating operator workload, smoothing turbulence, fuel savings, etc. In addition, sophisticated applications of GNC enable automatic or remote control.

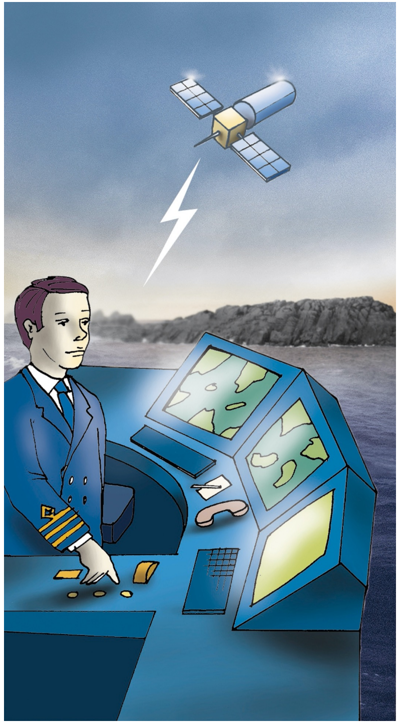
- **Guidance** refers to the determination of the desired path of travel (the "*trajectory*") from the vehicle's current location to a designated target, as well as desired changes in velocity, rotation and acceleration for following that path (main tool: dynamic optimization)
- **Navigation** refers to the determination, at a given time, of the vehicle's location and velocity (the "*state vector*") as well as its attitude (main tool: Kalman filter)
- **Control** refers to the manipulation of the forces, by way of steering controls, thrusters, etc., needed to execute guidance commands whilst maintaining vehicle stability (main tool: feedback control theory)

11.1 Motion Control System Architecture

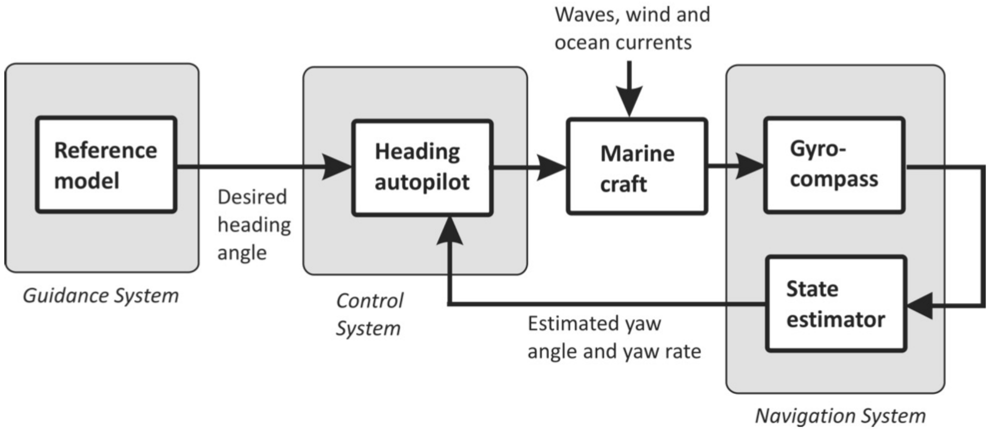


GNC blocks where the guidance system makes use of the estimated alternatively measured positions and velocities (closed-loop guidance system).

11.1 Motion Control System Architecture

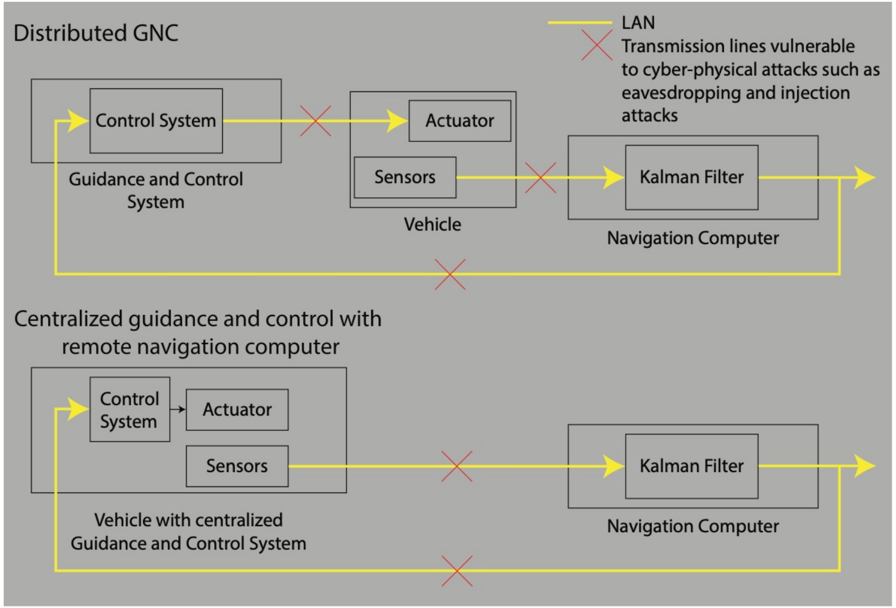


Copyright © Bjarne Stenberg



GNC blocks with an open-loop guidance system represented by a reference model.

GNC Architectures for Enhanced Cybersecurity



: A generic schematic of the signal flow in vehicles with a distributed GNC architecture and a centralized guidance and control computer with a remote navigation computer. Surfaces that are vulnerable to attacks are marked, and the goal is to make the system resistant against attacks on these surfaces.

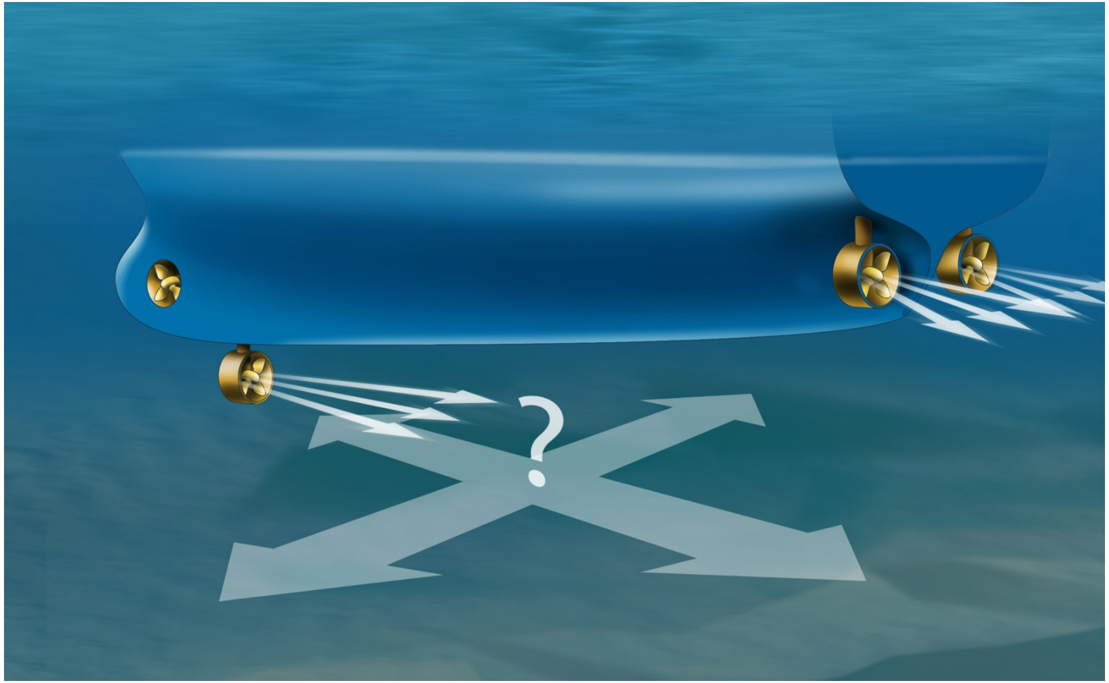
By transmitting signals between the GNC modules, motion control systems become vulnerable to cyber-physical attacks. For example, passive eavesdropping attacks may result in a leak of confidential system and control parameters. Active deception attacks may manipulate the state estimators, controllers, and actuators through the injection of spoofed data. To prevent such attacks, the transmitted signals must remain confidential across the transmission channels, and spoofed data cannot be allowed to enter the GNC loop.

The **Crypto Toolbox** contains tools for implementing high-performance algorithms in real time; even large data streams from cameras, radars, and lidars can be handled. The toolbox includes algorithms such as the Advanced Encryption Standard (AES), the AEGIS stream cipher, the Keyed-Hash Message Authentication Code (HMAC), and the stream ciphers from the eSTREAM portfolio.

GitHub repository:
https://github.com/pettsol/CryptoToolbox

Petter Solnør (2020). A Cryptographic Toolbox for Feedback Control Systems. *Modeling Identification and Control,* MIC- 41(4):2020:313-332.
https://www.mic-journal.no/PDF/2020/MIC-2020-4-3.pdf

11.2 Control Allocation



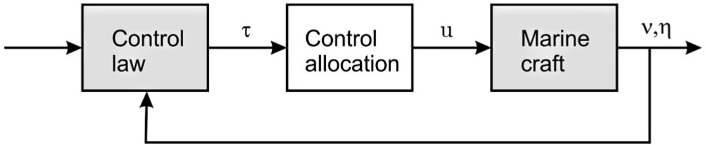
Control allocation is the problem of distributing the generalized control forces $\boldsymbol{\tau} \in \mathrm{R}^n$ to effectors/actuators with physical control inputs $\mathbf{u} \in \mathrm{R}^r$

11.2 Control Allocation

For marine craft moving in n DOFs it is necessary to distribute the generalized control forces $\boldsymbol{\tau} \in \mathbb{R}^n$ given by

$$\boldsymbol{\tau} = \boldsymbol{B}\boldsymbol{u}$$

to the actuators in terms of control inputs $\mathbf{u} \in \mathbb{R}^r$



$r \geq n$ fully actuated control

$r < n$ underactuated control

The input matrix $\boldsymbol{B}$ is square for $r = n$, that is the number of actuators is equal to n DOFs.

Computation of $\boldsymbol{u}$ from $\boldsymbol{\tau}$ is a model-based optimization problem which in its simplest form is unconstrained. The unconstrained solution is the pseudoinverse

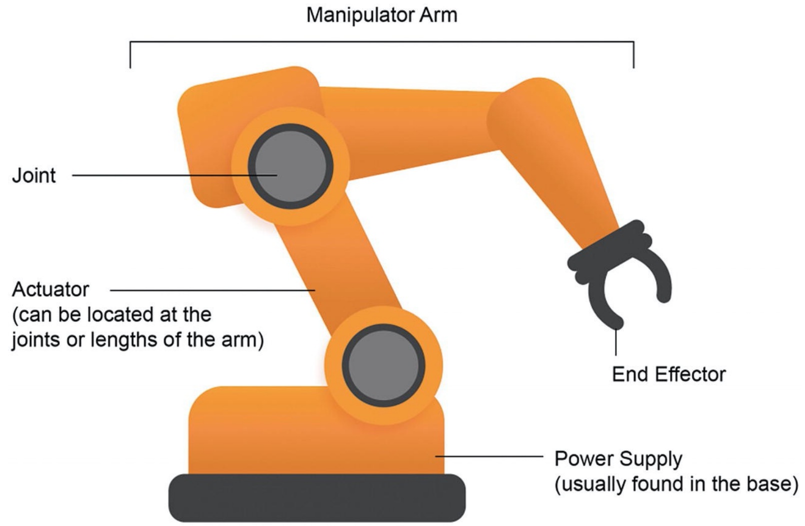
$$\boldsymbol{u} = \boldsymbol{B}^{\dagger}\boldsymbol{\tau} \quad \text{where} \quad \boldsymbol{B}^{\dagger} = \boldsymbol{B}^{\top}(\boldsymbol{B}\boldsymbol{B}^{\top})^{-1}$$

Physical limitations like input amplitude and rate saturations imply that a constrained optimization problem must be solved.

11.2 Effectors vs Actuators

- **Effectors:** mechanical devices such as control surfaces, rudders, fins, propellers, jets, engines and thrusters that can generate time-varying forces and moments to control a marine craft

- **Actuators:** electromechanical devices that are used to control the magnitude and/or direction of forces generated by the individual effectors



There may be more effectors than needed to meet the motion control objectives of a given application. Hence, for overactuated systems, the controllability of the chosen states and outputs would also be achieved with less control inputs.

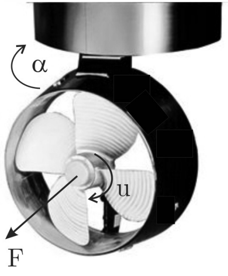
Overactuated systems are favorable due to effector redundancy allowing for fault tolerance and control reconfiguration. However, this usually involves solving an optimization problem to find the optimal actuator setpoints.

11.2.1 Propulsion and Actuator Models

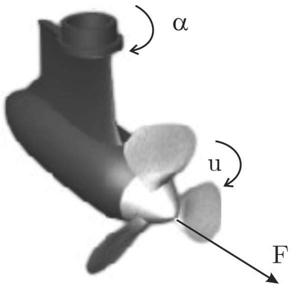
Some actuators that can be rotated at the same time as they produce control forces. This is also called vector thrust.

Rotatable Thrusters and Propellers
Examples are azimuth thrusters on an offshore supply vessel, podded propellers that can be rotated, and contra-rotating propellers.

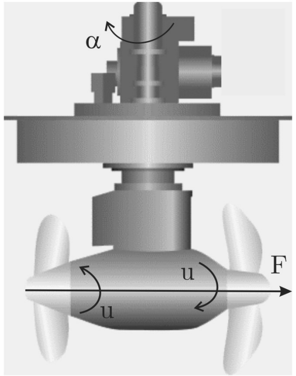
This increases the number of available controls from *r* to *r+p*, where *p* denotes the number of rotatable actuators for which additional nonlinearities are introduced.



Azimuth thruster



Podded propeller

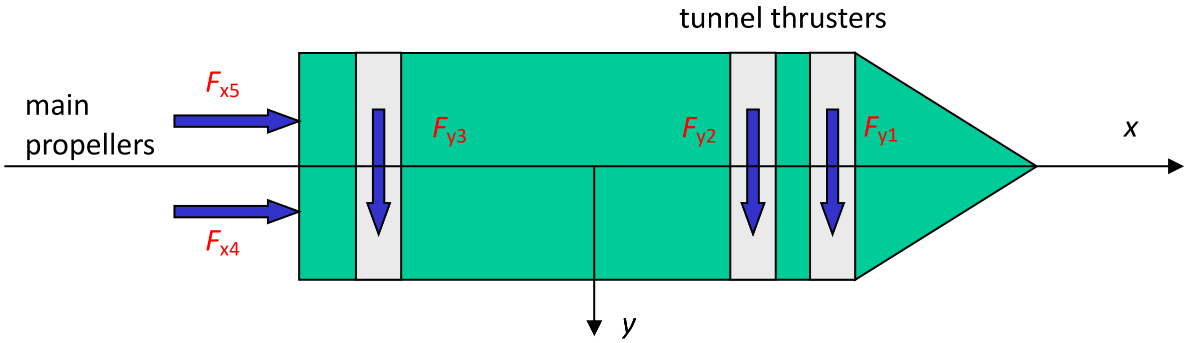


Contra-rotating propeller

11.2.1 Propulsion and Actuator Models

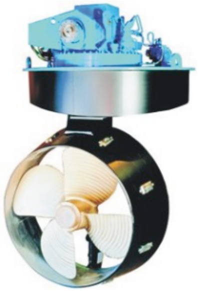
Main propellers: the main propellers of the vessel are mounted aft of the hull usually in conjunction with rudders. They produce the necessary force F_{xi} in the *x*-direction needed for transit.

Tunnel thrusters: transverse thrusters going through the hull of the vessel. The propeller unit is mounted inside a transverse tube and it produces a force F_{yi} in the *y*-direction. Tunnel thrusters are only effective at low speed which limits their use to low-speed maneuvering and dynamic positioning.

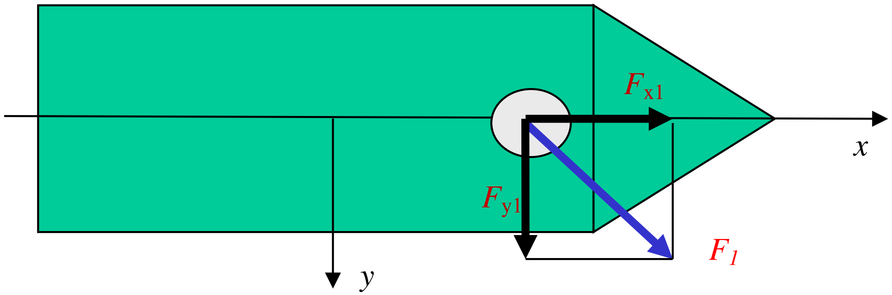


11.2.1 Propulsion and Actuator Models

Azimuth thrusters: thruster units that can be rotated an angle about the *z*-axis and produce two force components (F_{x1}, F_{y1}) in the horizontal plane are usually referred to as azimuth thrusters.



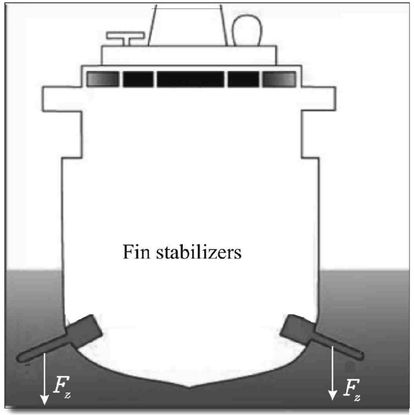
Azimuth thrusters are usually mounted under the hull of the vessel and the most sophisticated units are retractable. Azimuth thrusters are frequently used in DP since they can produce forces in different directions leading to an overactuated control problem that can be optimized with respect to power and possible failure situations.



By Siemens AG - Siemens AG (Siemens agreed the publication at de:Bild:Siemens Schottel Propulsor.jpg), CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=397859

11.2.1 Propulsion and Actuator Models

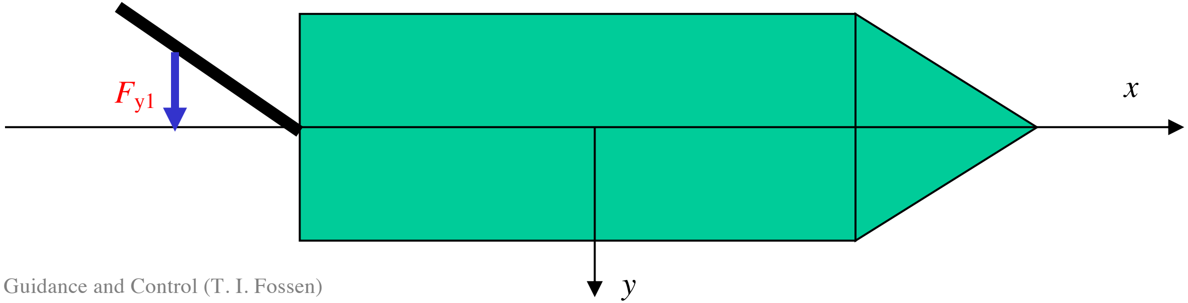
Stabilizing fins are used for damping of vertical vibrations and roll motions. They produce a force F_{zi} in the z-direction which is a function of the fin deflection. For small angles, this relationship is linear. Fin stabilizers can be retractable allowing for selective use in bad weather. The lift forces are small at low speed, so the most effective operating condition is in transit.



$$F_z \approx -\frac{1}{2}\rho U_r^2 A_F C_{L_\alpha} \alpha_F \qquad \tau_4 = l_y F_z$$

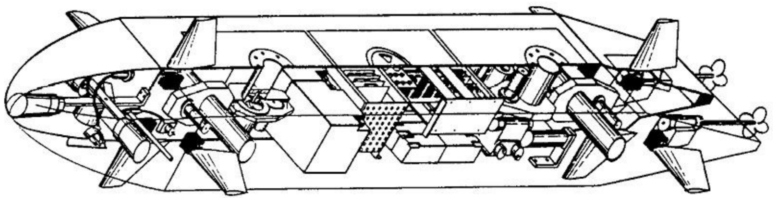
Aft rudders: rudders are the primary steering device for conventional vessels. They are located aft of the vessel and the rudder force F_{y1} will be a function of the rudder angle.

$$\boldsymbol{\tau}_R = \begin{bmatrix} -\frac{1}{2}(1-t_R)\rho U_R^2 A_r C_N \sin^2(\delta) \\ -\frac{1}{4}(1+a_H)\rho U_R^2 A_r C_N \sin(2\delta) \\ -\frac{1}{4}(x_R + a_H x_H)\rho U_R^2 A_r C_N \sin(2\delta) \end{bmatrix} \approx \begin{bmatrix} -X_{\delta\delta}\,\delta^2 \\ -Y_\delta\,\delta \\ -N_\delta\,\delta \end{bmatrix}$$



11.2.1 Propulsion and Actuator Models

Control surfaces: control surfaces can be mounted at different locations to produce lift and drag forces. For underwater vehicles these could be fins for diving, rolling, and pitching, rudders for steering, etc.



Healey, A. J. and D. Lienard (1993). Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles. IEEE Journal of Ocean Engineering, 18(3), 327–339.

https://nps.edu/web/cavr/auv

Stern and bow planes

$$X_S = -\frac{1}{2}\rho U_r^2 A_S C_{L_\delta} \delta_S^2, \qquad X_B = -\frac{1}{2}\rho U_r^2 A_B C_{L_\delta} \delta_B^2$$

$$Z_S = -\frac{1}{2}\rho U_r^2 A_S C_{L_\delta} \delta_S \qquad Z_B = -\frac{1}{2}\rho U_r^2 A_B C_{L_\delta} \delta_B$$

Rudder

$$X_R = -\frac{1}{2}\rho U_r^2 A_r C_{L_\delta}\, \delta_R^2$$

$$Y_R = -\frac{1}{2}\rho U_r^2 A_r C_{L_\delta}\, \delta_R$$

11.2.1 Propulsion and Actuator Models

The control force due to a propeller, a rudder, or a fin is (assuming linearity):

$$F_i = K_i u_i \quad \Longrightarrow \quad u_i = \frac{1}{K_i} F_i$$

K_i is the force coefficient
u_i is the control input depending on the actuator

Forces and moments in 6 DOFs corresponding to the force vector

$$\boldsymbol{F}_i^b = [F_{x_i}, F_{y_i}, F_{z_i}]^\top$$

and moment arms $\boldsymbol{r}_{bp_i}^b = [l_{x_i}, l_{y_i}, l_{z_i}]^\top$ can be written

$$\boldsymbol{\tau} = \sum_{i=1}^{r} \begin{bmatrix} \boldsymbol{F}_i^b \\ \boldsymbol{r}_{bp_i}^b \times \boldsymbol{F}_i^b \end{bmatrix} = \sum_{i=1}^{r} \begin{bmatrix} F_{x_i} \\ F_{y_i} \\ F_{z_i} \\ F_{z_i} l_{y_i} - F_{y_i} l_{z_i} \\ F_{x_i} l_{z_i} - F_{z_i} l_{x_i} \\ F_{y_i} l_{x_i} - F_{x_i} l_{y_i} \end{bmatrix}$$

11.2.1 Propulsion and Actuator Models

Example (Linear model describing nonlinear monotonic control forces):

Consider the linear model:

$$F_i = K_i u_i$$

This model can also be used to describe nonlinear monotonic control forces.

For instance, if the rudder force F_i is quadratic in rudder angle δ_i, that is

$$F_i = K_i |\delta_i| \delta_i := K_i u_i$$

Choosing

$$u_i = |\delta_i| \delta_i \quad \Rightarrow \quad \delta_i = \mathrm{sgn}(u_i)\sqrt{|u_i|}$$

and the resulting model is linear in u_i.

11.2.1 Propulsion and Actuator Models

The linear control force for the different actuators can be represented in terms of a force vector according to the following table:

Actuator	u_i (control input)	α_i (control input)	$\boldsymbol{F}_i^b$ (force vector)
Main propeller (longitudinal)	pitch and rpm	–	$[F_i,\ 0,\ 0]^\top$
Tunnel thruster (transverse)	pitch and rpm	–	$[0,\ F_i,\ 0]^\top$
Azimuth (rotatable) thruster	pitch and rpm	angle	$[F_i \cos(\alpha_i),\ F_i \sin(\alpha_i),\ 0]^\top$
Aft rudder	angle	–	$[0,\ F_i,\ 0]^\top$
Stabilizing fin	angle	–	$[0,\ 0,\ F_i]^\top$

$$
\boldsymbol{\tau} = \sum_{i=1}^{r} \begin{bmatrix} \boldsymbol{F}_i^b \\ \boldsymbol{r}_{bp_i}^b \times \boldsymbol{F}_i^b \end{bmatrix} = \sum_{i=1}^{r} \begin{bmatrix} F_{x_i} \\ F_{y_i} \\ F_{z_i} \\ F_{z_i} l_{y_i} - F_{y_i} l_{z_i} \\ F_{x_i} l_{z_i} - F_{z_i} l_{x_i} \\ F_{y_i} l_{x_i} - F_{x_i} l_{y_i} \end{bmatrix}
$$

11.2.1 Propulsion and Actuator Models

The control forces and moments $\mathbf{f} = [f_1,...,f_n]^\top$ are conveniently expressed as

$$\boldsymbol{f} = \boldsymbol{K}\boldsymbol{u}$$

Force coefficient matrix
The force coefficient matrix $\boldsymbol{K}$ is diagonal such that:

$$\boldsymbol{K} = \mathrm{diag}\{K_1, \ldots, K_r\} \quad \Longrightarrow \quad \boldsymbol{K}^{-1} = \mathrm{diag}\left\{\frac{1}{K_1}, \ldots, \frac{1}{K_r}\right\}$$

The actuator forces and moments relate to the control forces and moments by:

$$\begin{aligned} \boldsymbol{\tau} &= \boldsymbol{T}\boldsymbol{f} \\ &= \boldsymbol{T}\boldsymbol{K}\boldsymbol{u} \quad \Longrightarrow \quad \boldsymbol{B} = \boldsymbol{T}\boldsymbol{K} \end{aligned}$$

$\boldsymbol{T}$ is the thrust configuration matrix

where $\mathbf{u}$ is the control input.

11.2.1 Propulsion and Actuator Models

Thrust configuration matrix (*surge*, *sway*, *roll*, and *yaw*)

The thrust configuration matrix is defined in terms of a set of column vectors

$$\boldsymbol{T} = [\boldsymbol{t}_1, \ldots, \boldsymbol{t}_r]$$

$$\boldsymbol{t}_i = \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ -l_{y_i} \end{bmatrix}}_{\text{main propeller}}, \quad \boldsymbol{t}_i = \underbrace{\begin{bmatrix} 0 \\ 1 \\ -l_{z_i} \\ l_{x_i} \end{bmatrix}}_{\substack{\text{tunnel thruster} \\ \text{and aft rudder}}}, \quad \boldsymbol{t}_i = \underbrace{\begin{bmatrix} 0 \\ 0 \\ l_{y_i} \\ 0 \end{bmatrix}}_{\text{stabilizing fin}}$$

Azimuth thrusters

$$\boldsymbol{T}(\boldsymbol{\alpha}) = [\boldsymbol{t}_1, \ldots, \boldsymbol{t}_r]$$

$$\boldsymbol{t}_i = \underbrace{\begin{bmatrix} \cos(\alpha_i) \\ \sin(\alpha_i) \\ -l_{z_i} \sin(\alpha_i) \\ l_{x_i} \sin(\alpha_i) - l_{y_i} \cos(\alpha_i) \end{bmatrix}}_{\text{azimuth thruster}}$$

For azimuth (vector) thrust, the *T* matrix depends on the control inputs.

We can use the extended thrust configuration matrix presented on the next page to avoid this nonlinearity.

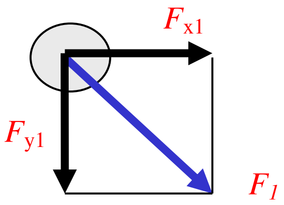
11.2.1 Propulsion and Actuator Models

Extended thrust configuration matrix for azimuth (vector) thrust

When solving the control allocation optimization problem an alternative representation is the extended thrust configuration matrix. Vector thrust can be treated as two control forces

$$\begin{aligned} F_{x_i} &= F_i \cos(\alpha_i) \\ &= K_i u_i \cos(\alpha_i) \end{aligned}$$

$$\begin{aligned} F_{y_i} &= F_i \sin(\alpha_i) \\ &= K_i u_i \sin(\alpha_i) \end{aligned}$$



Next, the extended force vector is defined according to

$$\boldsymbol{f}_e := \boldsymbol{K}_e \boldsymbol{u}_e$$

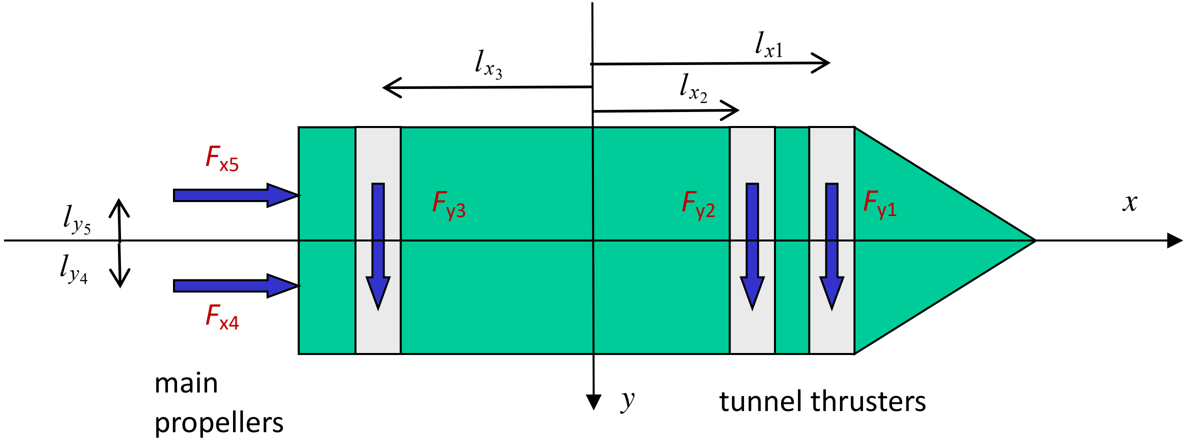
such that

$$\boldsymbol{\tau} = \boldsymbol{T}_e \boldsymbol{K}_e \boldsymbol{u}_e$$

11.2.1 Propulsion and Actuator Models

Example: 3-DOF motions (surge, sway and yaw)

$$\mathbf{T} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ l_{x1} & l_{x_2} & l_{x_3} & -l_{y_4} & -l_{y_5} \end{bmatrix} \qquad \mathbf{K} = \text{diag}\{K_1, K_2, K_3, K_4, K_5\}$$



11.2.1 Propulsion and Actuator Models

Example: Thrust configuration matrix for a marine craft

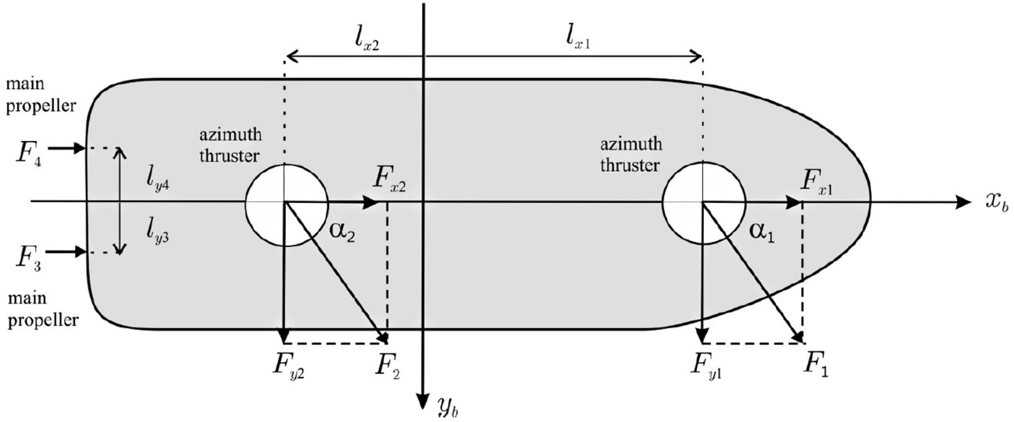
The forces and moment in surge, sway and yaw, satisfy

$$\boldsymbol{\tau} = \boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{K}\boldsymbol{u}$$

$$\Updownarrow$$

$$\begin{bmatrix} X \\ Y \\ N \end{bmatrix} = \begin{bmatrix} \cos(\alpha_1) & \cos(\alpha_2) & 1 & 1 \\ \sin(\alpha_1) & \sin(\alpha_2) & 0 & 0 \\ l_{x_1}\sin(\alpha_1) & l_{x_2}\sin(\alpha_2) & -l_{y_3} & -l_{y_4} \end{bmatrix} \begin{bmatrix} K_1 & 0 & 0 & 0 \\ 0 & K_2 & 0 & 0 \\ 0 & 0 & K_3 & 0 \\ 0 & 0 & 0 & K_4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

NB! This expression is nonlinear in the azimuth angles



11.2.1 Propulsion and Actuator Models

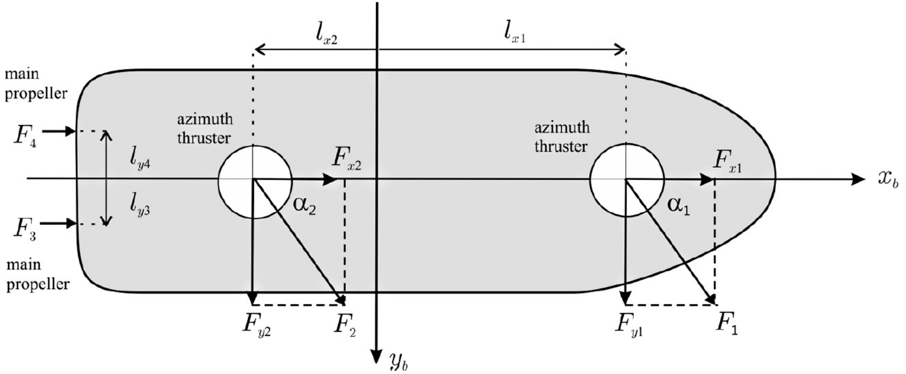
Example: Extended thrust configuration matrix for a marine craft

$$\boldsymbol{\tau} = \boldsymbol{T}_e \boldsymbol{K}_e \boldsymbol{u}_e$$

$$\Updownarrow$$

$$\begin{bmatrix} X \\ Y \\ N \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & l_{x_1} & 0 & l_{x_2} & -l_{y_3} & -l_{y_4} \end{bmatrix} \begin{bmatrix} K_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & K_4 \end{bmatrix} \begin{bmatrix} u_{1_x} \\ u_{1_y} \\ u_{2_x} \\ u_{2_y} \\ u_3 \\ u_4 \end{bmatrix}$$

NB! This is a linear problem where the extended thrust $\mathbf{u}_e$ can be found by matrix inversion



$$\boxed{\boldsymbol{u}_e = \boldsymbol{K}_e^{-1} \boldsymbol{T}_w^{\dagger} \boldsymbol{\tau}}$$

$$u_1 = \sqrt{u_{1_x}^2 + u_{1_y}^2}$$

$$u_2 = \sqrt{u_{2_x}^2 + u_{2_y}^2}$$

$$\alpha_1 = \mathrm{atan2}(u_{1_y}, u_{1_x})$$

$$\alpha_2 = \mathrm{atan2}(u_{2_y}, u_{2_x})$$

11.2.2 Unconstrained Control Allocation

Unconstrained least-squares (LS) optimization problem (Fossen 1991)

$$J = \min_{\boldsymbol{f}_e} \{\boldsymbol{f}_e^\top \boldsymbol{W} \boldsymbol{f}_e\}$$

$$\text{subject to:} \quad \boldsymbol{\tau} - \boldsymbol{T}_e \boldsymbol{f}_e = \boldsymbol{0}$$

W is a positive definite matrix, usually diagonal, weighting the control forces $\mathbf{f}_\mathbf{e}$.

Explicit solution to the LS optimization problem using Lagrange multipliers

$$L(\boldsymbol{f}_e, \boldsymbol{\lambda}) = \boldsymbol{f}_e^\top \boldsymbol{W} \boldsymbol{f}_e + \boldsymbol{\lambda}^\top (\boldsymbol{\tau} - \boldsymbol{T}_e \boldsymbol{f}_e)$$

$\boldsymbol{\lambda} \in \mathbb{R}^r$ is a vector of Lagrange multipliers.

$$\frac{\partial L}{\partial \boldsymbol{f}_e} = 2\boldsymbol{W}\boldsymbol{f}_e - \boldsymbol{T}_e^\top \boldsymbol{\lambda} = 0$$

$$\Downarrow$$

$$\boldsymbol{f}_e = \frac{1}{2}\boldsymbol{W}^{-1}\boldsymbol{T}_e^\top \boldsymbol{\lambda}$$

$$\Longrightarrow$$

$$\boldsymbol{\tau} = \boldsymbol{T}_e \boldsymbol{f}_e = \frac{1}{2}\boldsymbol{T}_e \boldsymbol{W}^{-1}\boldsymbol{T}_e^\top \boldsymbol{\lambda}$$

$$\Downarrow$$

$$\boldsymbol{\lambda} = 2(\boldsymbol{T}_e \boldsymbol{W}^{-1}\boldsymbol{T}_e^\top)^{-1}\boldsymbol{\tau}$$

Lagrange multipliers

Assuming that $\boldsymbol{T}_e \boldsymbol{W}^{-1}\boldsymbol{T}_e^\top$ is nonsingular

11.2.2 Unconstrained Control Allocation

Lagrange multipliers:

$$\boldsymbol{\lambda} = 2(\boldsymbol{T}_e \boldsymbol{W}^{-1} \boldsymbol{T}_e^\top)^{-1} \boldsymbol{\tau}$$

Optimal force vector:

$$\boldsymbol{f}_e = \underbrace{\boldsymbol{W}^{-1} \boldsymbol{T}_e^\top (\boldsymbol{T}_e \boldsymbol{W}^{-1} \boldsymbol{T}_e^\top)^{-1}}_{\boldsymbol{T}_w^\dagger} \boldsymbol{\tau}$$

Control allocation

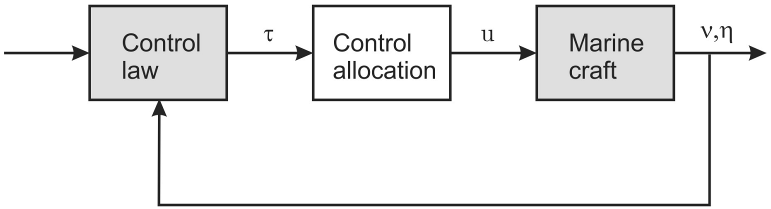
$$\boldsymbol{\tau} = \boldsymbol{T}_e \boldsymbol{K}_e \boldsymbol{u}_e$$

$$\Downarrow$$

$$\boldsymbol{u}_e = \boldsymbol{K}_e^{-1} \boldsymbol{T}_w^\dagger \boldsymbol{\tau}$$

```
function u = ucalloc(K,T,W,tau)
% u = ucalloc(K,T,W,tau) unconstrained control allocation. The generalized
%     force vector tau = T*K*u (dim n) is distributed to the input vector u
%     (dim r) where r>=n by minimizing the force f=K*u.
%
%     An unconstrained solution u = inv(K)*inv(W)*T'*inv(T*inv(W)*T')
%     exists if T*T' is non-singular.
%
%     - K is a diagonal rxr matrix of force coefficients
%     - T is a nxr constant configuration matrix.
%     - W is a rxr positive diagonal matrix weighting (prizing) the
%       different control forces f = K*u.
```

MSS Toolbox



Generalized inverse:

$$\boldsymbol{T}_w^\dagger = \boldsymbol{W}^{-1} \boldsymbol{T}_e^\top (\boldsymbol{T}_e \boldsymbol{W}^{-1} \boldsymbol{T}_e^\top)^{-1}$$

For the case **W = I**, that is equally weighted control forces reduces to the

Moore-Penrose pseudoinverse:

$$\boldsymbol{T}^\dagger = \boldsymbol{T}_e^\top (\boldsymbol{T}_e \boldsymbol{T}_e^\top)^{-1}$$

11.2.3 Constrained Control Allocation

Explicit solution for $\boldsymbol{\tau} = \boldsymbol{T}_e \boldsymbol{K}_e \boldsymbol{u}_e$ using piecewise linear functions

Optimization criterion

$$J = \min_{\boldsymbol{f}_e, \boldsymbol{s}, \bar{f}} \left\{ \boldsymbol{f}_e^\top \boldsymbol{W} \boldsymbol{f}_e + \boldsymbol{s}^\top \boldsymbol{Q} \boldsymbol{s} + \beta \bar{f} \right\}$$

subject to:

$$\begin{aligned} \boldsymbol{T}_e \boldsymbol{f}_e &= \boldsymbol{\tau} + \boldsymbol{s} \\ \boldsymbol{f}^{\min} \leq \boldsymbol{f}_e &\leq \boldsymbol{f}^{\max} \\ -\bar{f} \leq f_{e_1}, f_{e_2}, \ldots, f_{e_r} &\leq \bar{f} \end{aligned}$$

s is a vector of slack variables that should be close to zero, that is:

$\mathbf{Q} \gg \mathbf{W} > 0$

- The 1st term of the criterion corresponds to the LS criterion (pseudo-inverse)
- The 3rd term is introduced to minimize the largest force $\bar{f} = \max_i |f_i|$ among the actuators.
- The constant $\beta \geq 0$ controls the relative weighting of the two criteria.

This formulation ensures that the constraints $\boldsymbol{f}^{\min} \leq \boldsymbol{f}_e \leq \boldsymbol{f}^{\max}$ are satisfied, if necessary, by allowing the resulting generalized force $\boldsymbol{T}_\mathbf{e} \boldsymbol{f}_\mathbf{e}$ to deviate from its specification.

11.2.3 Constrained Control Allocation

Global Solution:

The global solution can be computed off-line using multi-parametric QP algorithms;
see Tøndel et al. (2001, 2003).

Local Solutions:

Explicit Solutions based on Minimum Norm and Null-Space Methods: In flight and aerospace control systems, the problems of control allocation and saturating control have been addressed by Durham (1993, 1994a, 1994b).

They also propose an explicit solution to avoid saturation referred to as the "direct method". By noticing that there are infinite combinations of admissible controls that generates control forces on the boundary of the closed subset of attainable controls, the "direct method" calculates admissible controls in the interior of the attainable forces as scaled down versions of the unique solutions for force demands.

Case Study: CyberShip I

Explicit optimal constrained least-squares solution (nonrotatable thrusters)

Supply vessel scale 1:70 equipped with
4 azimuth thruster. The controls are:

- RPM controlled
- Azimuth angles

Azimuth angles are fixed in the experiment

Problem: compute the optimal controls
$\mathbf{u} = [u_1, u_2, u_3, u_4]^T$ when u_3 saturates at 0.1 N.

NB! Cannot be solved using the Generalized Inverse

Reference:
T. A. Johansen, T. I. Fossen, P. Tøndel, Efficient Optimal Constrained Control Allocation via Multi-Parametric Programming. *AIAA Journal of Guidance, Dynamics and Control*, 2005.

Case Study: CyberShip I

Least-squares thrust allocation (generalized inverse)

In the case of no saturation the control allocation problem is solved as:

$$B(\alpha)u = \tau$$

$$B_i(\alpha) = \begin{pmatrix} \cos(\alpha_i) \\ \sin(\alpha_i) \\ -\ell_{yi}\cos(\alpha_i) + \ell_{xi}\sin(\alpha_i) \end{pmatrix}$$

$$\min_u u^T W u \quad \text{subject to} \quad Bu = \tau$$

$$\boxed{u = B^+\tau}$$

Case Study: CyberShip I

Constrained least-squares thrust allocation

$$\min_{u,s,\tilde{u}} \left(s^T Q s + u^T W u + \beta \tilde{u} \right)$$

$$Bu = \tau + s$$

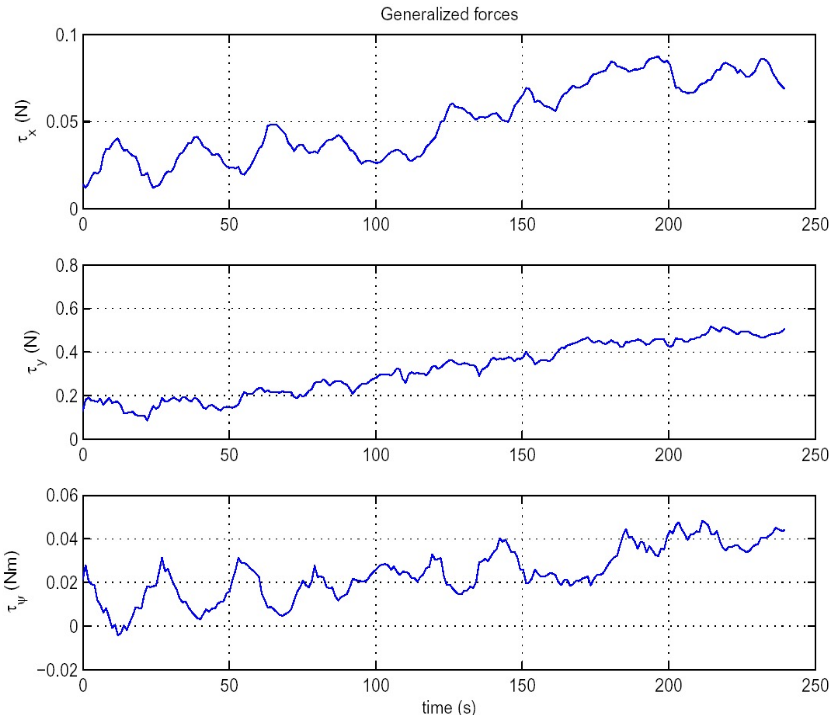
$$u_{min} \leq u \leq u_{max}$$

$$-\tilde{u} \leq u_1, u_2, \ldots, u_N \leq \tilde{u}$$

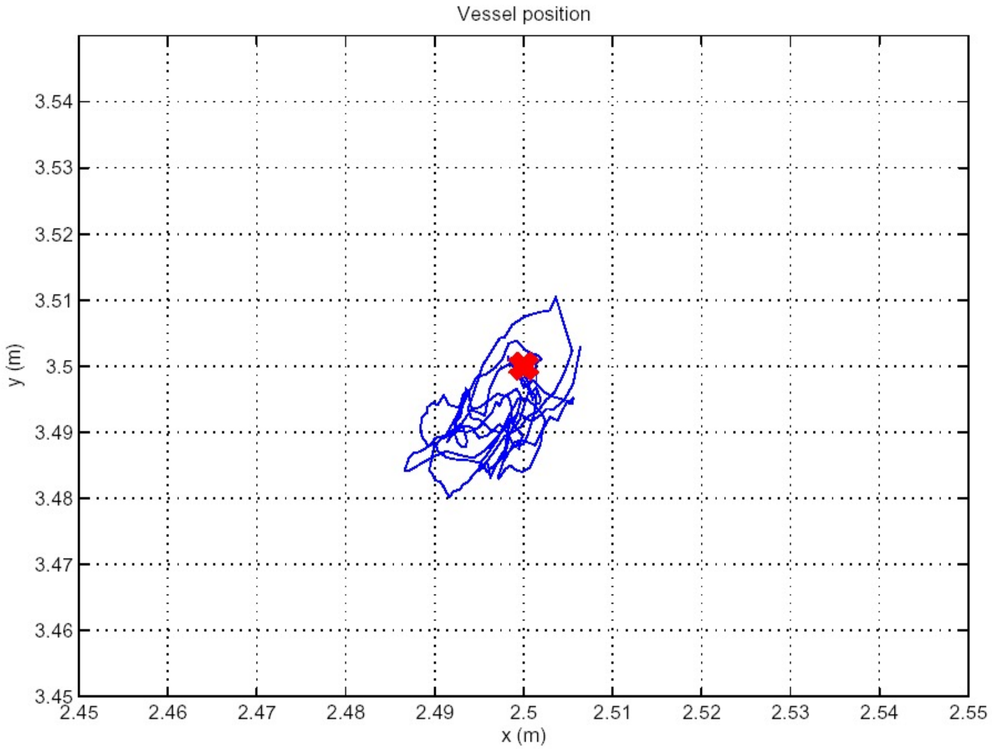
An *explicit* piecewise linear representation of the solution can be found using Multiparametric Quadratic Programming (mp-QP)

DP Experiments: Slowly Increasing Wind Loads

The increase in wind forces increases the demand for thrust and possible saturation in the control **u**

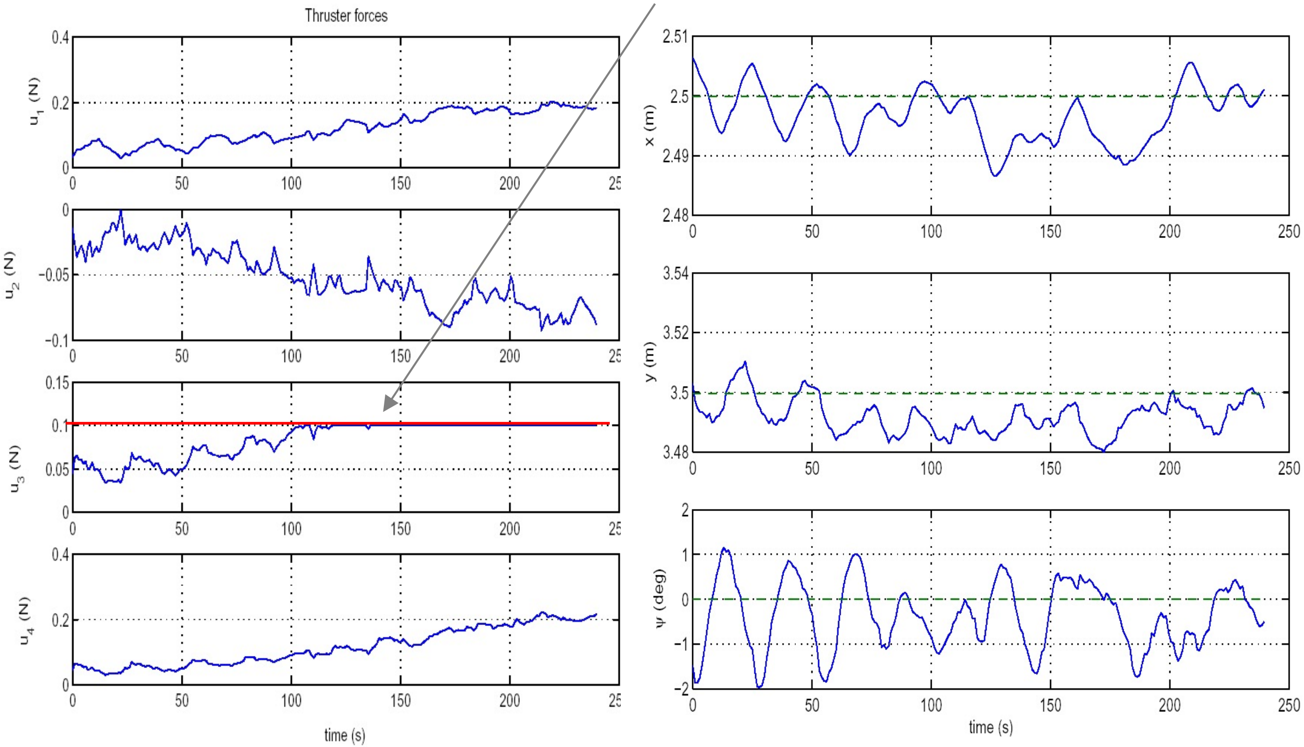


DP Experiments: Slowly Increasing Wind Loads

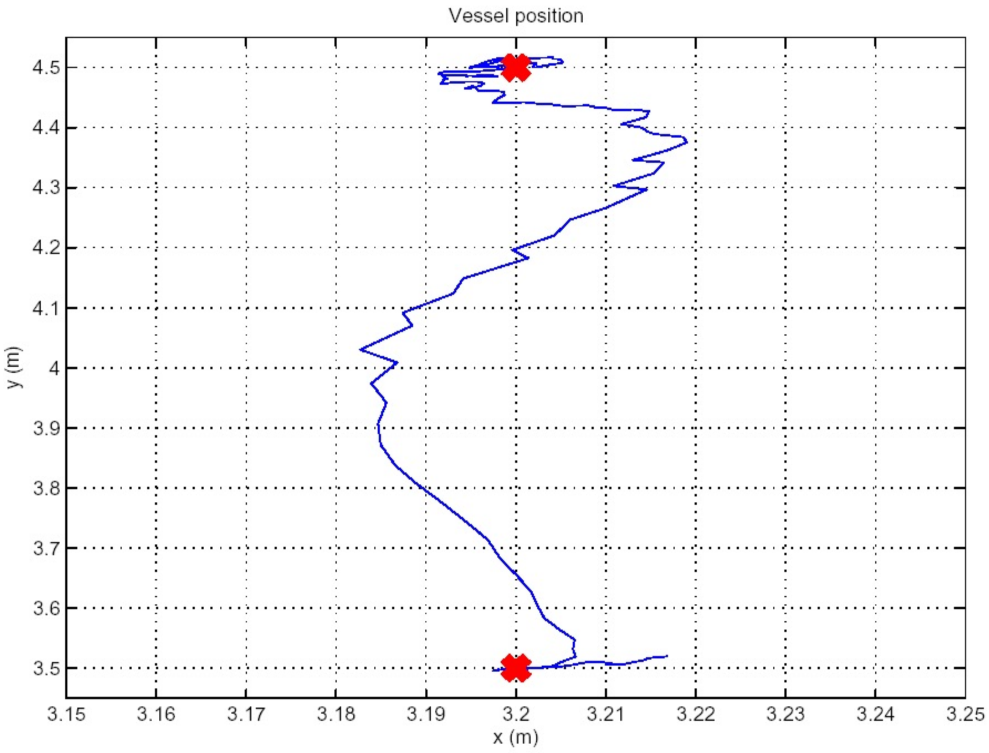


DP experiments: Position Change

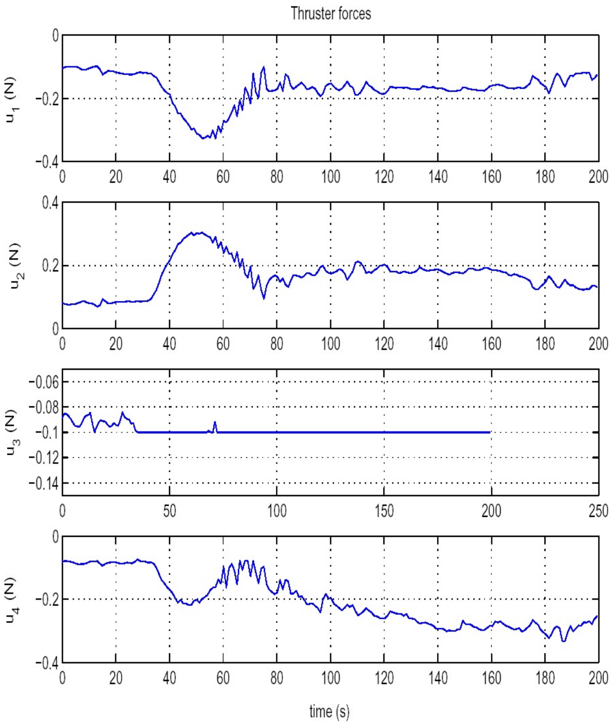
Azimuth thruster #3 hits saturation limit when wind force increases. Solution is still OPTIMAL

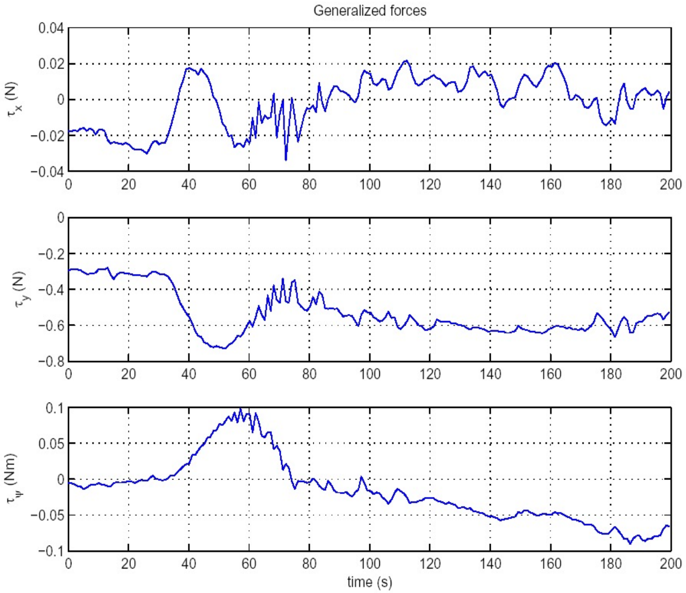


DP experiments: Position Change



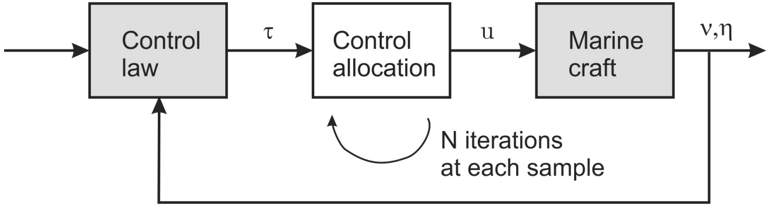
DP experiments: Position Change





11.2.3 Constrained Control Allocation

Iterative constrained control allocation for azimuth thrusters



- **Iterative Solutions:** An alternative to the explicit solution could be to use an iterative solution to solve the QP problem.
- The *m*-file function quadprog.m in the Matlab optimization toolbox can be used for computer simulations, while a stand-alone compiled QP solver must be implemented in a real-time application.
- The drawback with the iterative solution is that several iterations may have to be performed at each sample in order to find the optimal solution.
- An advantage of the iterative approach is that there is more flexibility for online reconfiguration, as for example a change in **W** may require that the explicit solutions are recalculated. Computational complexity is also greatly reduced by a "warm start", that is the numerical solver is initialized with the solution of the optimization problem computed at the previous sample.

11.2.3 Constrained Control Allocation

Iterative constrained control allocation for azimuth thrusters

The control allocation problem for vessels equipped with azimuth thrusters is in general a nonconvex optimization problem that is hard to solve.

The primary constraint is:

$$\boldsymbol{\tau} = \mathbf{T}(\boldsymbol{\alpha})\mathbf{f} \qquad \boldsymbol{\alpha} \in \mathbb{R}^p \text{ denotes the azimuth angles}$$

The azimuth angles $\boldsymbol{\alpha}$ must be computed at each sample together with the control inputs $\mathbf{u}$

Other problems:

feasible sectors $\alpha_{i,\min} \leq \alpha_i \leq \alpha_{i,\max}$

limiting turning rate $\dot{\boldsymbol{\alpha}}$

$$\boldsymbol{T}_w^{\dagger}(\boldsymbol{\alpha}) = \boldsymbol{W}^{-1}\boldsymbol{T}^{\top}(\boldsymbol{\alpha})[\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{W}^{-1}\boldsymbol{T}^{\top}(\boldsymbol{\alpha})]^{-1}$$

can be singular for certain $\boldsymbol{\alpha}$ values.

11.2.3 Constrained Control Allocation

Iterative constrained control allocation for azimuth thrusters

$$J = \min_{\mathbf{f},\boldsymbol{\alpha},\mathbf{s}} \sum_{i=1}^{r} \bar{P}_i |f_i|^{3/2} + \mathbf{s}^\top \mathbf{Q}\mathbf{s} + (\boldsymbol{\alpha} - \boldsymbol{\alpha}_0)^\top \boldsymbol{\Omega} (\boldsymbol{\alpha} - \boldsymbol{\alpha}_0) + \frac{\varrho}{\varepsilon + \det(\mathbf{T}(\boldsymbol{\alpha})\mathbf{W}^{-1}\mathbf{T}^\top(\boldsymbol{\alpha}))}$$

power consumption

ensures that the azimuth angles do not move to much within one sample

avoid singular configurations

subject to:

$$\mathbf{T}(\boldsymbol{\alpha})\mathbf{f} = \boldsymbol{\tau} + \mathbf{s}$$

$$\mathbf{f}_{\min} \leq \mathbf{f} \leq \mathbf{f}_{\max}$$

$$\boldsymbol{\alpha}_{\min} \leq \boldsymbol{\alpha} \leq \boldsymbol{\alpha}_{\max}$$

$$\Delta\boldsymbol{\alpha}_{\min} \leq \boldsymbol{\alpha} - \boldsymbol{\alpha}_0 \leq \Delta\boldsymbol{\alpha}_{\max}$$

This optimization problem is a nonconvex nonlinear program, and it requires a significant number of computations at each sample.

Two alternative implementation strategies will be mentioned.

T. A. Johansen, T. I. Fossen, and S. P. Berge. Constrained Nonlinear Control Allocation With Singularity Avoidance Using Sequential Quadratic Programming. *IEEE Transactions on Control Systems Technology*, Vol. 12 , No. 1, Jan. 2004, pp. 211-216

11.2.3 Constrained Control Allocation

Iterative solutions using quadratic programming

The nonlinear program can be locally approximated with a convex QP problem by assuming that:

1. Power consumption can be approximated by a quadratic term in $\mathbf{f}$, near the last force $\mathbf{f}_0$ such that $\mathbf{f} = \mathbf{f}_0 + \Delta\mathbf{f}$
2. The singularity avoidance penalty can be approximated by a linear term linearized about the last azimuth angle $\boldsymbol{\alpha}_0$ such that $\boldsymbol{\alpha} = \boldsymbol{\alpha}_0 + \Delta\boldsymbol{\alpha}$

The resulting QP criterion is (Johansen, Fossen and Berge 2004):

$$J = \min_{\Delta\mathbf{f},\Delta\boldsymbol{\alpha},\mathbf{s}} (\mathbf{f}_0 + \Delta\mathbf{f})^\top \mathbf{P}(\mathbf{f}_0 + \Delta\mathbf{f}) + \mathbf{s}^\top\mathbf{Q}\mathbf{s} + \Delta\boldsymbol{\alpha}^\top\boldsymbol{\Omega}\Delta\boldsymbol{\alpha} + \frac{\partial}{\partial\boldsymbol{\alpha}}\left(\frac{\varrho}{\varepsilon + \det(\mathbf{T}(\boldsymbol{\alpha})\mathbf{W}^{-1}\mathbf{T}^\top(\boldsymbol{\alpha}))}\right)\Bigg|_{\boldsymbol{\alpha}_0}\Delta\boldsymbol{\alpha}$$

subject to:

$$\mathbf{s} + \mathbf{T}(\boldsymbol{\alpha}_0)\Delta\mathbf{f} + \frac{\partial}{\partial\boldsymbol{\alpha}}(\mathbf{T}(\boldsymbol{\alpha}_0)\mathbf{f})\Big|_{\boldsymbol{\alpha}_0,\mathbf{f}_0}\Delta\boldsymbol{\alpha} = \boldsymbol{\tau} - \mathbf{T}(\boldsymbol{\alpha}_0)\mathbf{f}_0$$

$$\mathbf{f}_{\min} - \mathbf{f}_0 \leq \mathbf{f} \leq \mathbf{f}_{\max} - \mathbf{f}_0$$

$$\boldsymbol{\alpha}_{\min} - \boldsymbol{\alpha}_0 \leq \Delta\boldsymbol{\alpha} \leq \boldsymbol{\alpha}_{\max} - \boldsymbol{\alpha}_0$$

$$\Delta\boldsymbol{\alpha}_{\min} \leq \Delta\boldsymbol{\alpha} \leq \Delta\boldsymbol{\alpha}_{\max}$$

Case Study: Nonlinear Programming Approach: Power Optimality and Singularity Avoidance

$$J(\alpha, u, s) = \sum_{i=1}^{N} W_i(u_i) + s^T Q s + (\alpha - \alpha_0)^T \Omega (\alpha - \alpha_0) + \frac{\varrho}{\varepsilon + \det(B(\alpha) B^T(\alpha))}$$

Constraints:

$$\begin{aligned} s &= \tau - B(\alpha) u \\ u_{min} &\leq u \leq u_{max} \\ \alpha_{min} &\leq \alpha \leq \alpha_{max} \\ \Delta\alpha_{min} &\leq \alpha - \alpha_0 \leq \Delta\alpha_{max} \end{aligned}$$

T. A. Johansen, T. I. Fossen, and S. P. Berge. Constrained Nonlinear Control Allocation With Singularity Avoidance Using Sequential Quadratic Programming. *IEEE Transactions on Control Systems Technology*, Vol. 12 , No. 1, Jan. 2004, pp. 211-216

Case Study: Nonlinear Programming Approach: Power Optimality and Singularity Avoidance

Local convex quadratic approximation at each sample
– standard QP solution

Tradeoff between maneuverability and power optimality
– singularity avoidance

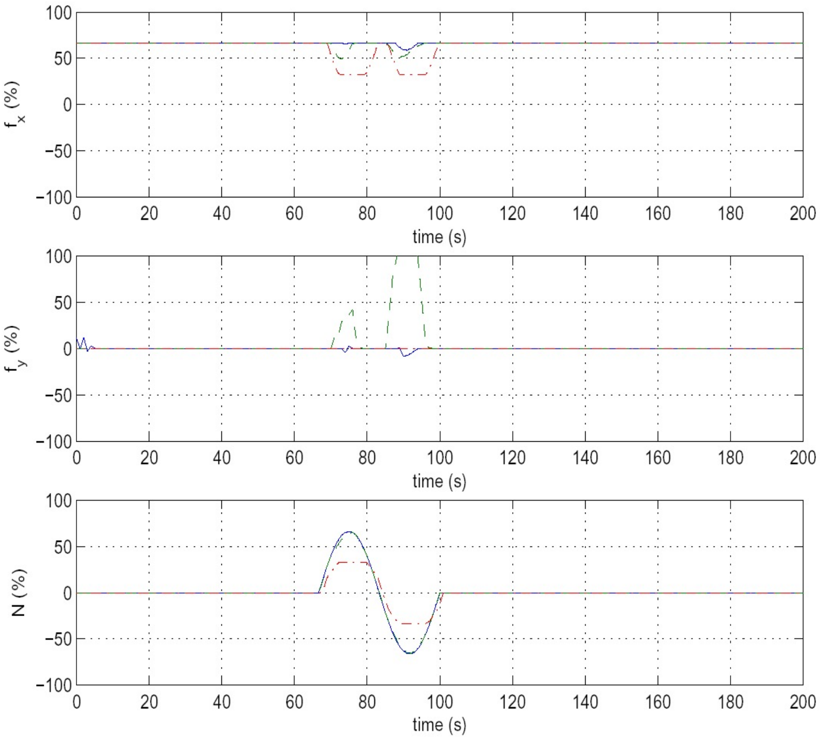
Dynamic solution that handles rate and amplitude constraints

We will consider a marine craft with four azimuth thrusters (-25° to 25° azimuth, 1 deg/s rate constraints)

Scenario: Constant forward thrust plus course-changing maneuver

1. Optimal approach with singularity avoidance
2. Optimal approach without singularity avoidance
3. Pseudo-inverse

Simulation Example

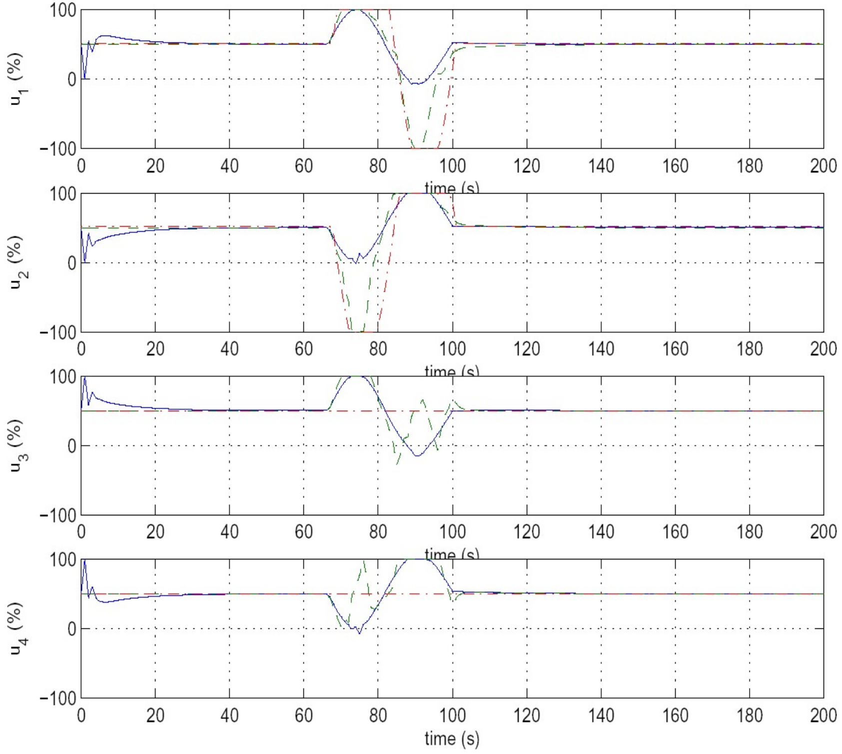


Blue – optimal

Green – optimal without singularity avoidance

Red – pseudo-inverse

Simulation Example

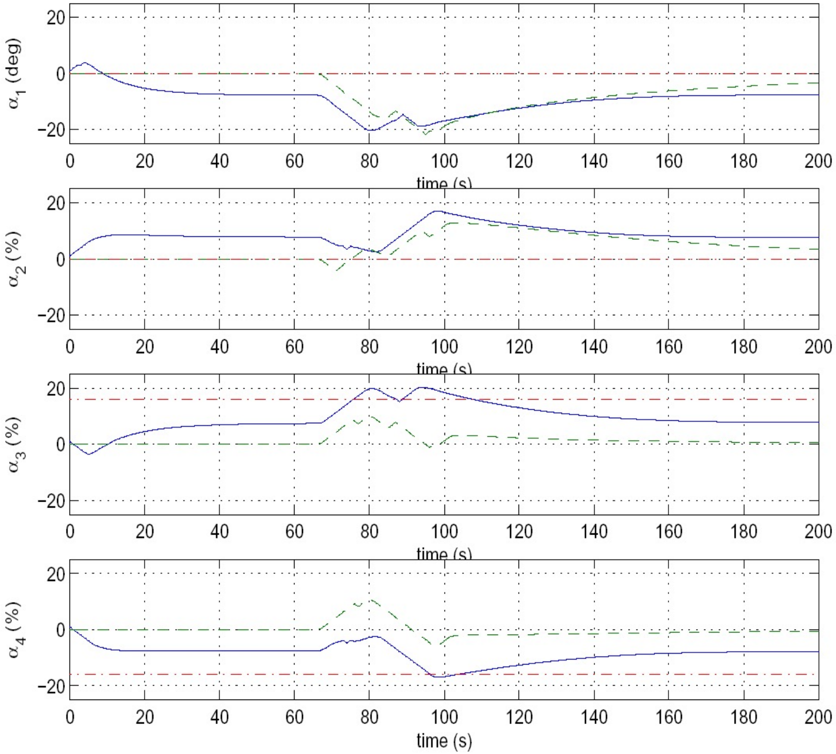


Blue – optimal

Green – optimal without singularity avoidance

Red – pseudo inverse

Simulation Example

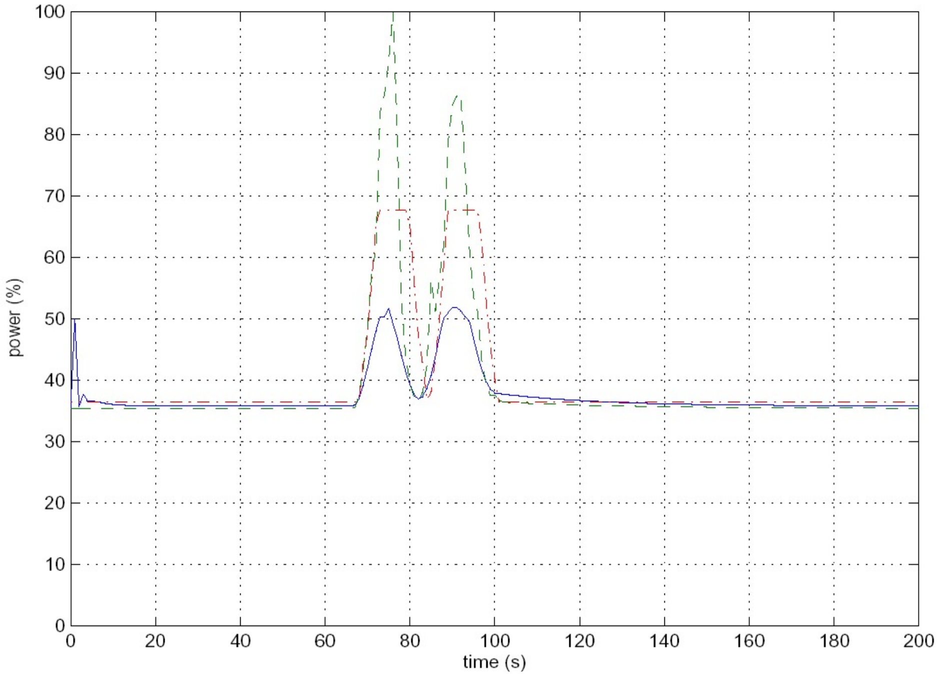


Blue – optimal

Green – optimal without singularity avoidance

Red – pseudo inverse

Simulation Example



Blue – optimal

Green – optimal without singularity avoidance

Red – pseudo inverse

Chapter Goals — Revisited

GNC Architecture:

- Understand the meaning of the G, N and C blocks in a GNC architecture for autonomous vehicles and marine craft.
- Cybersecurity: Understand that GNC architectures and signal transmissions between the G, N and C blocks are vulnerable for cyber threats. Solution: encrypt the sensor and control signals.

Control Allocation:

- Understand how control forces computed by feedback control systems are distributed as actuator commands using constrained and unconstrained control allocation methods.
- Be able to express forces from azimuth thrusters, tunnel thrusters, propellers, control surfaces, etc. in terms of the (extended) thrust configuration matrix **T** and force coefficient matrix **K**.
- Be able to design an unconstrained control allocation algorithm using the generalized inverse.
- Understand the principles for constrained control allocation and how dynamic programming can be used to solve these problems.