

Aaron Wickers

Simulation und Regelung eines Wasseroberflächen-Fahrzeugs

Bachelorarbeit

Stand: März 2019

© Lehrstuhl für Regelungstechnik
Christian-Albrechts-Universität zu Kiel

Aaron Wickers

Simulation und Regelung eines Wasseroberflächen-Fahrzeugs

Bachelorarbeit

Prüfer: Prof. Dr.-Ing. habil. Meurer
Abgabedatum: 03.03.2019

Diese Arbeit ist eine Prüfungsarbeit. Anderweitige Verwendung und die Weitergabe an Dritte, ist nur mit Genehmigung des betreuenden Lehrstuhls gestattet.

Inhaltsverzeichnis

Abstract und Kurzfassung	1
1 Einleitung	3
2 Physikalisches Konzept und Modellbildung in VREP	5
2.1 Allgemeines	5
2.2 Statischer Auftrieb	7
2.3 Dynamischer Auftrieb	8
2.3.1 Ausgleichskräfte und -momente in X-Y-Ebene	8
2.3.2 Ausgleichsmomente in Roll und Pitch	11
2.4 Strömungswiderstand	12
2.5 Antrieb	12
2.6 Ausarbeitung in VREP	13
3 Verbindung zu MATLAB	17
3.1 Kommunikation	17
3.2 Übermittelte Daten	18
4 Reglerentwurf und Simulation eines Line-Of-Sight-Reglers	21
4.1 Reglerentwurf eines Line-Of-Sight-Reglers	21
4.1.1 Grundprinzip des Line-Of-Sight-Reglers	21
4.1.2 Anwendung auf das Beispiel des Heron	23
4.2 Simulation und Auswertung des Line-Of-Sight-Reglers	25
4.2.1 Szenario A	26
4.2.2 Szenario B	27

4.2.3 Szenario C	28
5 Zusammenfassung und Ausblick	31
A Programmversionen	33
Literaturverzeichnis	35

Abstract und Kurzfassung

Abstract

The main topic of this thesis is the simulation and control of an unmanned surface vessel (USV) in calm waters. In particular the catamaran *Heron* by *Clearpath* is used as an example of a shipmodel. Since the ship can rotate and move in three degrees of freedom each, a system of six degrees of freedom is needed to create a simulation environment, which fits the dynamic behaviour of surface vessels in calm waters. Therefore, three decoupled mathematic models are presented. Together they add up to a system of six degrees of freedom. Those three individual models are needed, because only parameters of a two-dimensional model are available for the calculations of this thesis. Using this system a simulation environment for the stated shipmodel is created in the simulation program VREP. A remote-interface is presented, which is used to connect the simulation program VREP with the calculation software MATLAB . It allows MATLAB to control the behaviour of VREP and in addition enables both programs to share data with each other. Further, there is a line-of-sight-based controller implemented in MATLAB , to check if the remote-interface and the ship model work as expected. It allows the ship to move along a planned sequence of waypoints, which are handed over to the controller as a matrix. For that a simple PID-controller is used. At last the simulation results of the controlled movement are presented.

Kurzfassung

Diese Arbeit widmet sich der Simulation und Regelung eines unbemannten Wasseroberflächen-Fahrzeugs in beschränkten und ruhigen Gewässern. Im Speziellen dient der Katamaran *Heron* der Marke *Clearpath* als Vorbild. Für die Erstellung eines solchen Modells werden zunächst drei voneinander entkoppelte mathematische Modelle für das Verhalten eines Schiffes in ruhigem Gewässer herangezogen. Diese fügen sich in ein Gesamtsystem mit sechs Freiheitsgraden. Dies ist notwendig, da zur Berechnung des Systems nur die Parameter eines zweidimensionalen Modells zur Verfügung stehen. Mit Hilfe dessen wird dann eine geeignete Simulationsumgebung für ein Beispiel-Objekt - in diesem Fall der angesprochene Katamaran - im Programm VREP erschaffen. Es wird ferner eine Remote-Schnittstelle vorgestellt, mit deren Hilfe es sowohl möglich ist, die Simulationsumgebung aus der Software MATLAB zu bedienen als auch Daten zwischen beiden Programmen auszutauschen. Zum Testen des Schiffmodells und der Schnittstelle wird dann

ein Line-Of-Sight basierter Regler vorgestellt und in MATLAB implementiert, der es ermöglicht, dass das Schiff eine durch Wegpunkte vorgegebene Bahnplanung abfährt. Dazu wird ein einfacher PID-Regler verwendet. Anschließend werden die Simulationsergebnisse des geregelten Fahrverhaltens vorgestellt und im Hinblick auf Einflüsse einiger Faktoren untersucht.

Kapitel 1

Einleitung

Dass das autonome Fahren von Autos längst möglich ist und zukünftig ein wichtiger Baustein in einem modernen Verkehrssystem sein wird, daran zweifelt spätestens nach dem immensen Erfolg der Marke *Tesla* im Automotive-Sektor keiner mehr. Die Schlagzeilen im Bereich der autonomen Wasserfahrzeuge hingegen halten sich in Grenzen. Obwohl nicht so populär - sogar im Koalitionsvertrag der deutschen Bundesregierung von 2018 sind Testgebiete für autonome Wasserfahrzeuge vorgesehen. Die Vorteile für autonome Schifffahrt liegen dabei auf der Hand. Ähnlich wie im Straßenverkehr könnten so auch zu Wasser mit geringeren Personalkosten für die Betreiber und ohne den Faktor des menschlichen Versagens Güter und Personen kostengünstig und sicher transportiert werden.

Schon jetzt sind viele Assistenzsysteme auf Schiffen implementiert. So weisen Navigationsgeräte den Kurs, Systeme für die Stabilität des Schiffes sorgen auch bei unruhigem Gewässer für eine sichere und angenehmere Fahrt und ein automatisches Identifikationssystem (AIS) sorgt für die Kommunikation zwischen den einzelnen Schiffen sowie den Hafenanlagen. Auch teilautonome Fahrtsysteme wie beispielsweise ein Anlege-Assistent oder Ähnliches werden vereinzelt schon angeboten. Angesichts dieses Hintergrundes erscheint es mehr als verständlich, dass unter anderem China und einige skandinavische Länder ihre Forschungsinvestitionen im Bereich der autonomen Schifffahrt erheblich gesteigert haben. Mit diesem bislang noch ungenutzten Potential der Schifffahrt könnten so erhebliche Fortschritte erzielt werden.

Die Fachliteratur bietet für das autonome Fahren von Schiffen bereits eine Vielzahl von Konzepten und Ideen. So existieren Regelungsalgorithmen und -konzepte zur Pfadplanung, sowie zu deren Optimierung. Eine einfache Form eines gesteuerten Schiffes wird in dieser Bachelorarbeit daher konzipiert.

Dieser Arbeit liegt das unbemannte Oberflächen-Wasserfahrzeug (engl. USV = unmanned surface vehicle) *Heron* der Firma *Clearpath Robotics* zu Grunde. Dabei handelt es sich um ein transportables Wasseroberflächen-Fahrzeug, welches im Aufbau einem Katamaran gleichkommt. Ein Katamaran ist ein Wasserfahrzeug, welches aus zwei Schwimmkörpern besteht, die durch eine Fläche darüber fest verbunden sind. Da das Heron mit zwei Turbinen angetrieben wird, die an den Enden der Schwimmkörper befestigt sind, spricht man in diesem Fall auch von einem *Motorkatamaran*. Die Abbildung 1.1 zeigt ein Bild des praktischen Vorbildes für den simulierten Katamaran. Die folgende Arbeit gliedert sich wie folgt: Im zweiten Kapitel werden für eine solche Simulationsumgebung zunächst einige benötigte Grundlagen sowie mathematische Modelle präsentiert und an das vorgestellte Vorbild angepasst. Es folgt hieraus ein System mit sechs Freiheitsgraden, dass das dynamische Verhalten eines Körpers in ruhigem Wasser widerspiegelt. Das dritte Kapitel widmet sich einer programmierten Schnittstelle zwischen dem Simulations-



Abb. 1.1: Bild des *Heron*

programm VREP und dem Rechenprogramm MATLAB . Hier werden der Datenaustausch und die Kommunikation unter den beiden Programmen thematisiert. Der eigentliche Motivation dieser Arbeit, also der Regelung eines Oberflächen-Wasserfahrzeugs, wird sich im vierten Kapitel gewidmet. Hier wird zunächst das Konzept eines sogenannten Line-Of-Sight-Reglers erläutert. Anschließend wird ein solcher Regler für das erstellte Schiffsmodell erstellt und implementiert. Die Simulationsergebnisse werden dann präsentiert und ausgewertet. Zuletzt wird noch ein kleiner Ausblick auf mögliche Fortführungen dieser Arbeit und allgemein die moderne Schifffahrt gegeben.

Kapitel 2

Physikalisches Konzept und Modellbildung in VREP

In diesem Kapitel wird näher erläutert, nach welchen physikalischen Überlegungen das in dieser Arbeit verwendete Modell konstruiert und mit Hilfe welcher Vereinfachungen schließlich ein Modell in der Simulationssoftware VREP erzeugt wurde. Dazu wird zunächst unter Zuhilfenahme von Fachliteratur ein entsprechendes Konzept vorgestellt und anschließend die Entwicklung eines funktionsfähigen Modells dargestellt.

2.1 Allgemeines

Allgemein kann das in dieser Arbeit vorgestellte Modell eines Wasserfahrzeuges mit sechs Freiheitsgraden modelliert werden. Diese sechs entstehen aus der grundsätzlichen Überlegung, dass ein Körper in einem dreidimensionalen Raum sich über drei Achsen drehen sowie in drei Richtungen bewegen kann. Für diese Freiheitsgrade werden im Folgenden jeweils die englischen Begriffe der Fachliteratur verwendet. Es handelt sich also um die drei translatorische Freiheitsgrade. Dabei bezeichnet *heave* die Hebung in Richtung der Z-Achse, *sway* die Driftbewegung entlang der Y-Achse und *surge* den Schub in Richtung der X-Achse - jeweils in körperfesten Koordinaten. Hinzu kommen die drei rotatorische Freiheitsgrade. Dabei bezeichnet *roll* die Rotation um die X-Achse, *pitch* die Rotation um die Y-Achse und *yaw* die Rotation um die Z-Achse - also gleichzeitig auch die Richtung, die als Fahrtrichtung des Vehikels gilt. Eine Übersicht über diese sechs Freiheitsgrade (im Folgenden abgekürzt aus dem Englischen: DOF = degrees of freedom) ist in der Abbildung 2.1 visualisiert. Es ergibt sich also, dass die X-Y-Ebene die für eine Bahnplanung entscheidende Ebene ist, da sich das Schiff unter Vernachlässigung äußerer Störfaktoren etc. auf dieser Ebene bewegt. Um eindeutige Vektoren aufzeigen zu können, ist es notwendig, sowohl ein inertiales, absolutes Koordinatensystem festzusetzen, als auch ein dynamisches, im Folgenden als *körperfest* bezeichnetes Koordinatensystem zu definieren. Der grundsätzliche Unterschied beider Koordinatensysteme liegt darin, dass das *intertiale* System als absolute Referenz zum *körperfesten* gesehen wird. Das körperfeste Koordinatensystem bleibt dabei fest an einem Körper. Dreht man das Boot in eine beliebige Richtung, so wird trotzdem die X-Achse des körperfesten Systems durch die Spitze des Schiffes verlaufen. Um die Lage bzw. die Position von mehreren Körpern zu vergleichen, wird ein inertiales Koordinatensystem definiert, das als Vergleichs- oder auch Null-System gilt. Normalerweise spiegelt dieses Inertialsystem auch eine Umgebung wider. Mittels später vorgestellter Konzepte ist es möglich, Positionen, Geschwindigkeiten und andere Daten zwischen den Koordinatensystemen zu transformieren. Eine Positionsangabe im absoluten

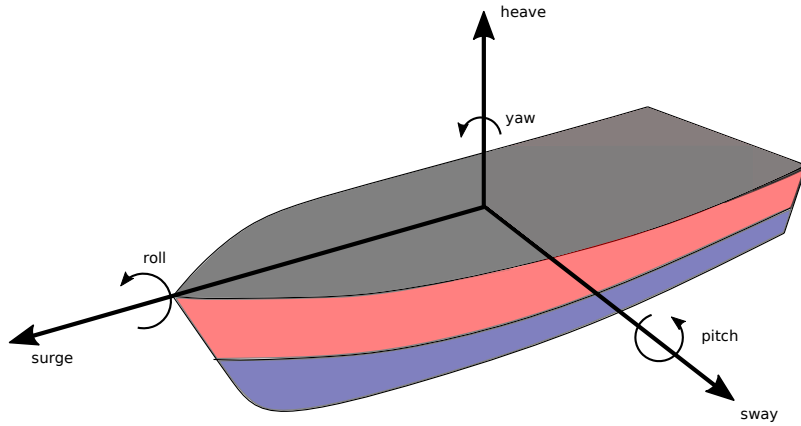


Abb. 2.1: Veranschaulichung der Freiheitsgrade eines Schiffes.

Koordinatensystem soll im Folgenden durch den Vektor $\boldsymbol{\eta} = [x, y, z]$ ausgedrückt werden. Die Geschwindigkeitsangabe eines Körpers in allen sechs Freiheitsgrade in absoluten Koordinaten wird durch den Vektor

$$\boldsymbol{v}_0 = [\dot{x} \ \dot{y} \ \dot{z} \ \dot{\varphi}_0 \ \dot{\theta}_0 \ \dot{\psi}_0]^T \quad (2.1)$$

definiert. Ferner wird ein Vektor mit den Geschwindigkeiten in körperfesten Koordinaten als

$$\boldsymbol{v}_1 = [u \ v \ w \ \dot{\varphi}_1 \ \dot{\theta}_1 \ \dot{\psi}_1]^T \quad (2.2)$$

bezeichnet. Dabei seien u , v und w die translatorischen Geschwindigkeiten (entsprechend *surge*, *sway*, *heave*), $\dot{\varphi}_1$, $\dot{\theta}_1$ und $\dot{\psi}_1$ die drei rotatorischen Winkelgeschwindigkeiten (entsprechend *roll*, *pitch*, *yaw*). Der Index 0 bzw. 1 beschreibt dabei, in welchem Koordinatensystem eine Geschwindigkeit angegeben wird. Es gilt der Index 0 für das Inertialsystem und der Index 1 für das körperfeste System. Wie im Folgenden ausgeführt wird, besteht das in dieser Arbeit aufgestellte Simulationsmodell aus der Zusammenführung verschiedener voneinander entkoppelter Einzelmodelle, welche jeweils entweder konkrete Phänomene der Schifffahrt oder aber Ausgleichskräfte und -momente widerspiegeln. Es werden dafür in Unterkapitel 2.2 ein eindimensionales Modell für die Auftriebskraft eines Körpers, in Unterkapitel 2.3 ein zweidimensionales Modell für das dynamische Verhalten eines Schiffes in der X-Y-Ebene, sowie ein weiteres eindimensionales Modell für die Ausgleichskräfte um den *pitch*-Winkel vorgestellt. Mit Hilfe dieser lässt sich das berechnete Simulationsobjekt näherungsweise vollständig beschreiben. Der Vorteil, die sechs Freiheitsgrade aus drei verschiedenen Modellen zusammenzufügen, liegt darin, dass ein umfangreiches 6-DOF-Modell eine Parametrisierung im Bezug auf Massenverteilung etc. benötigen würde. Es existieren allerdings nach der Quelle [7] für das genannte zweidimensionale Modell in der X-Y-Ebene entsprechende Parameter, die näherungsweise für einen einzelnen Schwimmkörper genutzt werden können. Erweitert man dieses Modell um die genannten Einzelmodelle, fügt sich dies zu einem Gesamtsystem, welches für eine Simulationsumgebung als hinlänglich genau angenommen wird. Eine tatsächliche eigene Parametrisierung eines Schiffes in allen sechs Freiheitsgraden würde nicht nur ein physisches Exemplar des Schiffes sondern auch einen großen Zeitaufwand, komplizierte Tests und teures Equipment erfordern. Daher wird für dieses Modell mit den genannten Parametern gearbeitet.

2.2 Statischer Auftrieb

Der sogenannte statische Auftrieb ist der Grundstein für die Simulation eines Wasserfahrzeugs. Auch bekannt als Archimedisches Prinzip ist hiermit die Formel

$$F_a = g\rho V \quad (2.3)$$

gemeint. Dabei ist F_a die Auftriebskraft, g die Erdbeschleunigung, ρ die Dichte des Stoffes und V das vom eintauchenden Körper verdrängte Volumen. Die Formel stellt also eine Beziehung zwischen der Kraft, die ein Körper als Auftrieb in Wasser erfährt, und dem von ihm verdrängten Volumen her. Ihr entgegen wirkt grundsätzlich auf der Erde die Gewichtskraft

$$F_g = mg. \quad (2.4)$$

Dabei bezeichnet F_g die resultierende Gewichtskraft und m die Masse eines Körpers. Durch eine Gleichsetzung dieser Kräfte lässt sich demnach bestimmen, ob ein Körper überhaupt schwimmen kann oder ob er untergeht. Im Gleichgewichtsfall folgt also für diese Gleichung

$$F_a = F_g, \quad (2.5)$$

beziehungsweise

$$g\rho V = mg \quad (2.6)$$

und somit

$$\rho V = m. \quad (2.7)$$

Hiermit ist also eine Abhängigkeit von der Beschaffenheit eines verdrängten Stoffes, dessen Volumen, sowie der Masse eines Körpers gegeben. Ein Körper kann demnach nur schwimmen, wenn die auf ihn wirkende Gewichtskraft kleiner als die Auftriebskraft ist - also nur, wenn die Masse des Körpers kleiner als das verdrängte Volumen multipliziert mit der Dichte des verdrängten Stoffes ist. Eine veranschaulichende Grafik ist in Abb. 2.2 zu finden. Im ersten Fall ist die Auftriebskraft größer als die Gewichtskraft des Körpers. Der Körper schwebt im zweiten Fall, da Auftriebskraft und Gewichtskraft gleich groß sind. Im dritten Fall sinkt der Körper auf den Grund, da die Gewichtskraft größer ist als die Auftriebskraft. Für das Modell dieser Arbeit bedeutet diese Überlegung also, es muss eine Kraft in Z-Richtung wirken, für die gilt

$$F_a = \iiint_V \rho g \, dV. \quad (2.8)$$

Dabei sei das Integral über V ein von der Geometrie des eintauchenden Körpers abhängiges Volumen, welches exakt der Wasserverdrängung des Körpers entspricht. Da die Unterschiede gering sind und eine erheblich leichtere Berechnung daraus folgt, wird für das in dieser Arbeit konstruierte Modell angenommen, dass der Schwimmkörper näherungsweise einem Quader entspricht. Dadurch vereinfacht sich die Kraft zu

$$F_h = A_{xy} g \rho t, \quad (2.9)$$

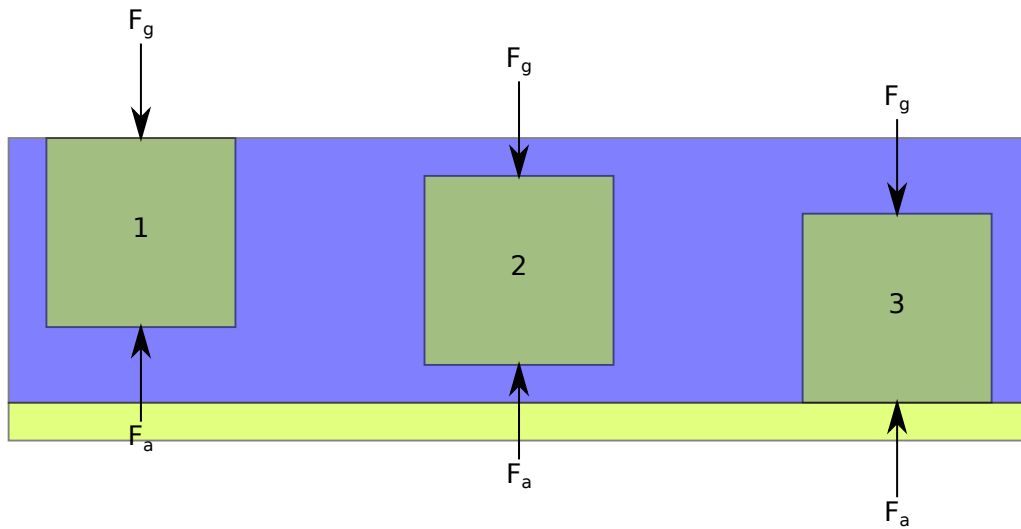


Abb. 2.2: Veranschaulichung des statischen Auftriebs.

wobei A_{xy} die Fläche des Quaders in der X-Y-Ebene darstellt. Die einzige nicht konstante Variable in dieser genäherten Gleichung ist somit die Eintauchtiefe t , welche sich ohne weiteres ermitteln lässt. Mit Hilfe der so berechneten Kraft ist es möglich, einen sich nicht bewegend schwimmenden Quader zu simulieren. In der Realität entspräche dies einem einfachen Ponton.

2.3 Dynamischer Auftrieb

Wie bereits festgestellt, verfügt das hier verwendete Modell eines Wasserfahrzeuges über sechs Freiheitsgrade (drei Translation, drei Rotation). Um in einem Raum nun nicht nur das Verhalten eines Pontons sondern das eines Wasserfahrzeugs simulieren zu können, muss überlegt werden, welche Kräfte auf einen dynamischen Körper in Wasser wirken. Diese Kräfte fasst man allgemein unter dem Begriff des dynamischen Auftriebs zusammen. Mit dem bereits in 2.2 beschriebenen statischen Auftrieb ist ein translatorischer Freiheitsgrad abgedeckt, nämlich der Auftrieb in Richtung der Z-Achse. Es verbleiben also Ausgleichskräfte in Richtung der X- und Y-Achse, sowie Ausgleichsmomente in den Rotationsrichtungen *roll*, *pitch* und *yaw*. Die Realisierung dieser Kräfte erfolgt durch zwei verschiedene Modelle.

2.3.1 Ausgleichskräfte und -momente in X-Y-Ebene

Das hier verwendete Prinzip basiert auf dem in [4] vorgestellten Modell für Wasserfahrzeuge. Angenommen wird zunächst einzig und allein die Bewegung in der X-Y-Ebene. Ein solches System bezeichnet man als planar. Es werden mit Hilfe dieses Modells die Kräfte in X- und Y-Richtung – F_u und F_v – sowie ein Drehmoment in der X-Y-Ebene – M_{ψ_1} – errechnet. Für diesen Teil werden logischerweise nur drei der sechs verfügbaren Freiheitsgrade benötigt, womit sich die Geschwindigkeitsvektoren in diesem Ansatz zu

$$\mathbf{v}_0 = [\dot{x} \ \dot{y} \ \dot{\psi}_0]^T \quad (2.10a)$$

$$\mathbf{v}_1 = [u \ v \ \dot{\psi}_1]^T \quad (2.10b)$$

vereinfachen lassen. Der Vektor \mathbf{v}_1 wird ferner als Zustandsvektor für das weiter unten vorgestellte Gleichungssystem verwendet. Aus der Abbildung 2.3 wird ersichtlich, dass für die X-Y Ebene das körperfeste Koordinatensystem in Referenz zum absoluten Koordinatensystem ein um die Z-Achse mit dem Winkel ψ gedrehtes Koordinatensystem ist. Daher lässt sich mittels der Rotationsmatrix

$$R = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \quad (2.11)$$

die Umrechnung von absoluten Koordinaten in körperfeste Koordinaten durchführen. Eine Rückführung lässt sich durch die inverse Matrix

$$R^{-1} = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix} \quad (2.12)$$

ausüben.

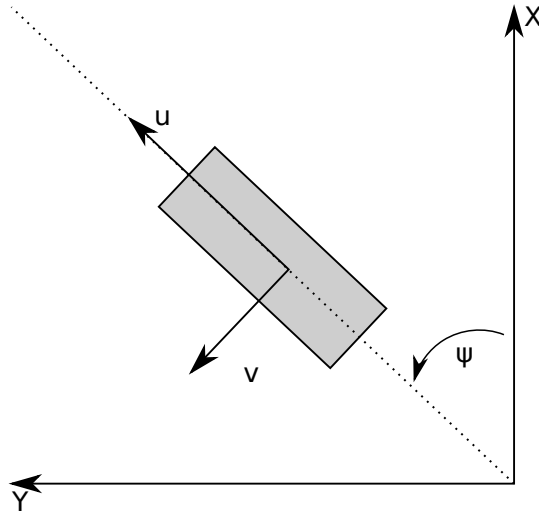


Abb. 2.3: Veranschaulichung der Koordinaten-Transformation von einem inertialen zu einem körperfesten System.

Gemäß der Fachliteratur [4] gilt für die Kräfte in einem dynamischen Modell eines 3-DOF-Körpers in der X-Y-Ebene der Zusammenhang

$$\mathbf{v}_0 = R(\psi) \mathbf{v}_1 \quad (2.13)$$

$$M\dot{\mathbf{v}}_1 = -C(\mathbf{v}_1)\mathbf{v}_1 - D(\mathbf{v}_1)\mathbf{v}_1 + \boldsymbol{\tau}. \quad (2.14)$$

Die Geschwindigkeit \mathbf{v} gilt nach Definition

$$\mathbf{v} = \frac{d\boldsymbol{\eta}}{dt} \quad (2.15)$$

Dabei beschreibt $\boldsymbol{\tau} = [\tau_u \ \tau_v \ \tau_{\psi_1}]^T$ einen Kraftvektor, welcher äußere Einflüsse widerspiegelt. Die Matrix M bezeichnet die Massenmatrix, die Matrix $C(\mathbf{v}_1)$ realisiert die entsprechenden Coriolis- und Zentrifugalkräfte, während die Matrix $D(\mathbf{v}_1)$ als Dämpfungsmatrix bezeichnet wird. Demnach beinhaltet die Formel (2.13) die bereits erwähnte Rotation bzw. Umrechnung zwischen *körperfestem* und *absolutem* Koordinatensystem, die Formel (2.14) hingegen berechnet das dynamische Verhalten eines Wasservehikels. Mit Hilfe von (2.14) lassen sich also die Kräfte F_u , F_v und das Moment M_{ψ_1} berechnen. Per Definition gilt nach [4] für die Matrizen M , $C(\mathbf{v}_1)$ und $D(\mathbf{v}_1)$:

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{\psi}_1} \\ 0 & mx_g - N_{\dot{v}} & I_{zz} - N_{\dot{\psi}_1} \end{bmatrix}, \quad (2.16a)$$

$$C(\mathbf{v}_1) = \begin{bmatrix} 0 & 0 & -M_{22}v - \frac{M_{23}+M_{32}}{2}\psi_1 \\ 0 & 0 & M_{11}u \\ M_{22}v + \frac{M_{23}+M_{32}}{2}\psi_1 - M_{11}u & 0 & 0 \end{bmatrix}, \quad (2.16b)$$

$$D(\mathbf{v}_1) = \begin{bmatrix} X_u + X_{|u|u}|u| & 0 & 0 \\ 0 & Y_v + Y_{|v|v}|v| & -Y_{\dot{\psi}_1} \\ 0 & -N_v & N_{\dot{\psi}_1} + N_{|\dot{\psi}_1|\dot{\psi}_1}|\dot{\psi}_1| \end{bmatrix}. \quad (2.16c)$$

Die darin enthaltenen Elemente sind durch verschiedene Test- und Messverfahren ermittelbar. Es handelt sich dabei um Abmessungen, Massenverteilung, Trägheitsmoment und andere spezielle Parameter. Die hier benötigten Werte werden in dieser Arbeit als bekannt vorausgesetzt und sind in der Tabelle 2.1 aufgelistet, welche aus der Quelle [7] übernommen wurden.

Massenmatrix		Dämpfungsmatrix			
M_{11}	25.8 kg	X_u	12.0 kg/s	$X_{ u u}$	2.5 kg/m
M_{22}	33.8 kg	Y_v	17.0 kg/s	$Y_{ v v}$	4.5 kg/m
M_{23}	6.2 kg m	Y_r	0.2 kg m/s		
M_{32}	6.2 kg m	N_v	0.5 kg m/s		
M_{33}	2.76 kg m ²	N_r	0.5 kg m ² /s	$N_{ \dot{\psi}_1 \dot{\psi}_1}$	0.1 kg m ²

Tabelle 2.1: Parameter des Modell-Schiffs.

Der Vektor $\boldsymbol{\tau}$ wird in diesem Modell ebenfalls nur bedingt genutzt und entfällt zumindest für diesen Abschnitt, da äußere Kräfte/Momente nur durch die Antriebsturbinen auf den Schwimmkörper wirken. Da die Simulationssoftware die absolute Position und Geschwindigkeit ausgeben kann, wird (2.13) umgestellt zu

$$\mathbf{v}_1 = R^{-1} R \mathbf{v}_1 = R^{-1} \mathbf{v}_0. \quad (2.17)$$

Mit Hilfe der so ermittelten körperfesten Geschwindigkeiten können im Modell dann gemäß der Gleichungen (2.17) und (2.14) die Kräfte F_u , F_v und das Moment $M_{\dot{\psi}_1}$ errechnet werden.

Da die Kräfte in der Simulationssoftware VREP nur in absoluten Koordinaten auf den Körper gegeben werden können, wird mit Hilfe der Rotationsmatrix R eine Rückführung auf dieses Koordinatensystem durchgeführt, wodurch sich dann die Kräfte F_x , F_y und das Moment $M_{\dot{\psi}_0}$ er-

geben. Es folgt also aus dem beschriebenen Vorgehen für dieses Modell zunächst die Bestimmung der körperfesten Geschwindigkeiten zu

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{\psi}_1 \end{bmatrix} = \begin{bmatrix} \cos(\psi_0) & \sin(\psi_0) & 0 \\ -\sin(\psi_0) & \cos(\psi_0) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi}_0 \end{bmatrix}. \quad (2.18)$$

Mit diesen Geschwindigkeiten ergeben sich die Kräfte in körperfesten Koordinaten zu

$$\begin{bmatrix} F_u \\ F_v \\ M_{\dot{\psi}_1} \end{bmatrix} = -[C(\mathbf{v}_1) + D(\mathbf{v}_1)] \begin{bmatrix} u \\ v \\ \dot{\psi}_1 \end{bmatrix}. \quad (2.19)$$

Die Rückführung in absolute Koordinaten liefert schließlich die Kräfte

$$\begin{bmatrix} F_x \\ F_y \\ M_{\dot{\psi}_0} \end{bmatrix} = \begin{bmatrix} \cos(\psi_0) & -\sin(\psi_0) & 0 \\ \sin(\psi_0) & \cos(\psi_0) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_u \\ F_v \\ M_{\dot{\psi}_1} \end{bmatrix}. \quad (2.20)$$

Addiert man diese Größen nun auf den Schwerpunkt des Schwimmkörpers, simuliert man damit die Kräfte, die der Körper in der X-Y-Ebene durch seine Bewegung als Driftkräfte bzw. als Dämpfung erfährt.

2.3.2 Ausgleichsmomente in Roll und Pitch

Nachdem nun alle 3 translatorischen Ausgleichskräfte und ein rotatorisches Ausgleichsmoment berücksichtigt wurden, werden nun noch die Rotationsmomente hinzugefügt, die ein Kippen des Schiffes über die X-Achse (Pitch) und die Y-Achse (Roll) berücksichtigen. Dazu wird nach [4] eine Vereinfachung für quaderförmige Vehikel mit kleinen Winkelauslenkungen angenommen. Dies ist sinnvoll, da die realen Schwimmkörpervolumina nahezu äquivalent sind zu einem entsprechenden Quader und eine große Auslenkung in ruhigem Gewässer nicht möglich ist. Diese Überlegung führt zu den Momenten $M_{\dot{\varphi}_1}$ und $M_{\dot{\theta}_1}$ in der Form

$$M_{\varphi_1} = \rho g V G M_l \varphi_1 \quad (2.21a)$$

$$M_{\theta_1} = \rho g V G M_t \theta_1. \quad (2.21b)$$

Dabei sei wie schon beim statischen Auftrieb g die Erdbeschleunigung, V das vom eintauchenden Körper verdrängte Volumen und ρ die Dichte des verdrängten Stoffes. $G M_l$ und $G M_t$ seien die sogenannten *metazentrischen Höhen* in *longitudinaler* und *transversaler* Richtung. Das Metazentrum ist ein virtueller Punkt in dem sich die vertikale Achse des Schwerpunkts eines Körpers mit der vertikalen Achse des um einen Winkel φ_1 bzw. θ_1 gedrehten Schwerpunkts trifft. Der Abstand zwischen diesem Metazentrum und dem realen Schwerpunkt eines Körpers ist als *metazentrische Höhe* definiert. Im Modell dieser Arbeit wird weiterhin die Vereinfachung angenommen, dass ein Moment M_{θ_1} zu vernachlässigen ist. Der Aufbau eines Katamarans bedingt, dass auf Grund des langen Hebelarms ein unwahrscheinlich großes rotatorisches Moment M_{θ_1} wirken müsste, damit eine tatsächliche Bewegung des Schiffes entgegen der Gewichtskraft

resultieren würde. Diese wird in stillen Gewässern nicht vorkommen, daher kann M_{θ_1} als null angenommen werden. Durch diese Ergänzung sind alle sechs Freiheitsgrade abgedeckt.

2.4 Strömungswiderstand

Es sollen in diesem Modell ebenfalls solche Kräfte und Momente berücksichtigt werden, die die Strömungswiderstände eines Körpers in Wasser oder Luft simulieren. Dadurch wird einerseits eine bedeutend realistischere Umgebung geschaffen, andererseits fungieren diese auch als Dämpfung. Für die Widerstandskraft F_r eines dynamischen Körpers in einer Umgebung mit konstanter Dichte ρ , gilt im Allgemeinen die Formel

$$F_r = c_r A \frac{1}{2} \rho v^2. \quad (2.22)$$

Dabei bezeichnet c_r den sogenannten Strömungswiderstandskoeffizienten, A die Bezugsfläche der Bewegungsrichtung und v die Geschwindigkeit der Bewegungsrichtung. Diese Kräfte werden in allen sechs Freiheitsgraden in Abhängigkeit der körperfesten Geschwindigkeiten

$$\mathbf{v}_1 = [u \ v \ w \ \dot{\varphi}_1 \ \dot{\theta}_1 \ \dot{\psi}_1]^T \quad (2.23)$$

berücksichtigt, jedoch müssen diese nicht in allen der bereits vorgestellten mathematischen Modelle auch hinzugefügt werden. In dem zweidimensionalen Modell aus Abschnitt 2.3.1 sind diese bereits integriert durch die Dämpfungsmatrix. In Abschnitt 2.3.2 wird bereits erklärt, dass das Ausgleichsmoment M_{φ_1} in Näherung nicht berücksichtigt wird. Daher werden nur die Kraft F_w und das Moment M_{θ_1} für die Strömungswiderstände aufgestellt. Sie werden dem Modell nur noch in diesen Freiheitsgraden entgegen gesetzt aufaddiert. Es folgt dafür also der zusätzliche Vektor

$$\mathbf{F}_r = \begin{bmatrix} 0 & 0 & -F_{rw} & 0 & -M_{r\theta_1} & 0 \end{bmatrix}^T. \quad (2.24)$$

In diesem Teil des Modells spiegelt sich auch die Überlegung wider, dass ein Körper eine größere Kraft erfährt, je schneller er bewegt wird. Durch das Zusammenfügen der in 2.2, 2.3 und 2.4 aufgestellten Kräfte ist das Modell unter Berücksichtigung der genannten Vereinfachungen vollständig beschrieben.

2.5 Antrieb

Der Antrieb des in dieser Arbeit modellierten Schiffes besteht in der Realität aus zwei Turbinen, die am hinteren Ende der beiden Schwimmkörper befestigt sind. Für die Funktion einer Turbine bietet die Fachliteratur der Physik – im Speziellen die Hydrodynamik – eine Vielzahl von Gleichungen zur Bestimmung von wirkenden Kräften. Da in diesem Modell allerdings eine Spannungsansteuerung oder Ähnliches zumindest jetzt noch nicht vorgesehen ist, erfüllt eine Turbine in der Simulation nur den Zweck, an einer bestimmten Stelle des Schiffes eine Kraft auszuüben. Da das Modell ferner äußere Faktoren wie Wasserverwirbelungen, Wind sowie Wellengang nicht berücksichtigt, sondern im Gegenteil von einer gänzlich ruhigen Wasseroberfläche

ausgeht, kann eine Turbine also vereinfacht dargestellt werden als ein Punkt, der an der Stelle, an der die Turbine in der Realität lokalisiert ist, eine Kraft entlang der körperfesten X-Achse ausübt. Diese Kraft bezeichnet man auch als Schub. Im Falle des Heron-Modells ist der Punkt für die anliegende Kraft in der Abb. 2.4 durch rote Pfeile visualisiert. Die Größe der Pfeile trifft keine Aussage über mögliche Kräfte.

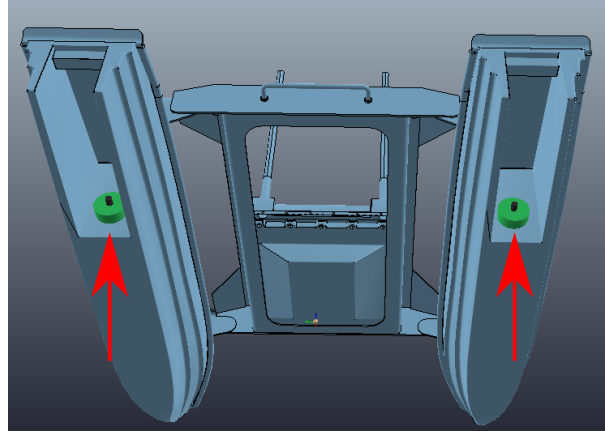


Abb. 2.4: Veranschaulichung des Schiffantriebes.

2.6 Ausarbeitung in VREP

Bevor es zur tatsächlichen Erstellung eines Modells in VREP kommen kann, müssen noch einige mathematische Grundlagen der Winkeldefinition erklärt und implementiert werden. Das Programm VREP verwendet nämlich – wie viele Simulationsprogramme – eine Winkeldefinition, die als Kardan-Winkel bezeichnet werden. Kardan-Winkel sind eine spezielle Form der Euler-Winkel-Definition und bestehen aus einem Satz von drei Winkeln, mit deren Hilfe die Rotation eines Körpers beschrieben werden kann. Dabei ist die Reihenfolge der Achsen, um die ein Körper gedreht wird, unbedingt anzugeben, da für die Rotation mittels dieser Winkel keine Kommutativität gegeben ist. Für Euler-Winkel gilt im Allgemeinen, dass die erste und die dritte Rotationsachse dieselbe sind. Im Spezialfall der Kardan-Winkel gilt, dass um drei verschiedene Achsen gedreht wird. Es folgt also beispielsweise für die Euler-Winkel-Definition die Rotationsmatrizen-Multiplikation

$$R = R_x(\gamma) R_y(\beta) R_x(\alpha). \quad (2.25)$$

Dabei sind R_x , R_y und R_z jene 3x3-Matrizen, die eine Rotation um die jeweilige Achse ausüben. Es folgt demnach für diese Matrizen

$$R_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\varphi & -s_\varphi \\ 0 & s_\varphi & c_\varphi \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}, \quad R_z(\psi) = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.26)$$

Durch Auflösen der Gleichungen kann eingesehen werden, dass die Kommutativität für diese Art von Winkeldefinitionen nicht gegeben ist. Wie schon im zweidimensionalen Modell in 2.3.1

gezeigt wurde, entsteht durch die Rotation um eine Achse ein neues Koordinatensystem. Für die Aneinanderreihung von drei Rotationen folgt demnach, dass es nicht nur ein Zielkoordinatensystem sondern auch zwei Koordinatensysteme als Zwischenschritte geben muss. Innerhalb dieser sind die Kardan-Winkel definiert. Anders ausgedrückt kann gesagt werden, dass die Rotation eines Körpers in Kardan-Winkeln immer um die drei körperfesten Rotationsachsen ausgeübt wird. Für die in VREP angegebenen Winkel gilt für die Rotation eines Körpers vom inertialen Koordinatensystem in ein rotiertes Koordinatensystem die Transformationsvorschrift nach [1]

$$R = R_z(\psi) R_y(\theta) R_x(\varphi), \quad (2.27)$$

was bedeutet, dass zuerst um die körperfeste x-Achse, dann um die körperfeste y-Achse und zuletzt um die körperfeste z-Achse gedreht wird. Um nun aus der so berechneten Rotationsmatrix R die gesuchten Winkel zu berechnen, muss diese zu den drei unbekannten Winkeln hin aufgelöst werden. Nach geschicktem Auflösen kann – wie im Skript [5] gezeigt – für die körperfesten Winkel φ_1 , θ_1 und ψ_1 der Vektor

$$\begin{bmatrix} \varphi_0 \\ \theta_0 \\ \psi_0 \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{R_{2,1}}{R_{1,1}}\right) \\ \arctan\left(-\frac{R_{3,1}}{\sqrt{(R_{3,2})^2 + (R_{3,3})^2}}\right) \\ \arctan\left(\frac{R_{3,2}}{R_{3,3}}\right) \end{bmatrix} \quad (2.28)$$

aufgestellt werden. Dabei bezeichnen die Variablen $R_{a,b}$ die jeweiligen Elemente der berechneten Rotationsmatrix R . Die Winkelgeschwindigkeiten eines rotierenden Körpers werden in VREP im inertialen System $\boldsymbol{\eta} = [x, y, z]$ angegeben. Da für die im Vorherigen vorgestellten Modelle jedoch Winkel und Winkelgeschwindigkeiten im körperfesten System des Wasservehikels benötigt werden, muss eine Berechnung dieser gewünschten Werte erfolgen. Nach Definition aus [5] und [6] gilt für die Umrechnung zwischen zwei Koordinaten-Systemen für die gesuchten relativen Winkelgeschwindigkeit der Zusammenhang

$$S = \dot{R}R^T = \begin{bmatrix} 0 & -\dot{\psi}_1 & \dot{\theta}_1 \\ \dot{\psi}_1 & 0 & -\dot{\varphi}_1 \\ -\dot{\theta}_1 & \dot{\varphi}_1 & 0 \end{bmatrix}. \quad (2.29)$$

Dabei ist \dot{R} als die zeitliche Ableitung der Rotationsmatrix R und R^T die transponierte Matrix der Rotationsmatrix R . Die Ergebnismatrix S ist eine schiefsymmetrische Matrix, die die Werte der Winkelgeschwindigkeiten im körperfesten Koordinatensystem, also $\dot{\varphi}, \dot{\theta}, \dot{\psi}$ enthält. Die Berechnung hierfür wird analytisch mit Hilfe des Programms wx-Maxima ausgeführt. Nach Auslesen und Zusammenfassen der Terme ergibt sich für die gesuchten Winkelgeschwindigkeiten ein Vektor $\boldsymbol{\omega} = [\dot{\varphi}_1, \dot{\theta}_1, \dot{\psi}_1]$ zu

$$\boldsymbol{\omega} = \begin{bmatrix} \dot{\varphi}_0 \sin \theta_0 + \dot{\psi}_0 \\ \dot{\theta}_0 \cos \psi_0 - \dot{\varphi}_0 \sin \psi_0 \cos \theta_0 \\ \dot{\theta}_0 \sin \psi_0 + \dot{\varphi}_0 \cos \psi_0 \cos \theta_0 \end{bmatrix}. \quad (2.30)$$

Für die tatsächliche Erstellung einer Simulation des *Heron* in VREP wird nun zunächst die Oberfläche des Heron als Objekt in die Simulation eingefügt. Die beiden Schwimmkörper des Katamaran-Modells werden dabei getrennt betrachtet. Jeder der beiden Körper erhält eigene Antriebs-, Auftriebs- und Ausgleichkräfte, welche auf ihn wirken. Der Gesamtaufbau des Katamarans wird durch eine Fixierung der relativen Positionen der beiden Körper zueinander

erreicht. Der Vorteil dieser Vorgehensweise ist offensichtlich der, dass das Simulationsprogramm deutlich weniger und simplere dynamische Oberflächen und Körper berechnen muss. Außerdem können so leichter das Verhalten der einzelnen Bestandteile des Katamarans studiert werden. Alle oben aufgeführten Berechnungen werden direkt in VREP in sogenannten *Child-Scripts* in VREP implementiert. Diese Skripte stellen eine Programmier-Oberfläche in der Programmiersprache *Lua* dar, mit deren Hilfe Berechnungen ausgeführt werden können, aber auch Einfluss auf das Verhalten einzelner Simulationsobjekte genommen werden kann. Mit dieser Implementierung können also alle für die besprochenen mathematischen Modelle benötigten Werte und Berechnungen in VREP ausgelesen oder berechnet werden.

Die Vorgehensweise zur Erzeugung des Modells in VREP läuft dann wie folgt ab. Um mit den vorangegangenen Ausführungen ein entsprechendes Modell zu entwickeln, wird zunächst die Oberfläche des Heron-Schiffes in VREP als Oberfläche importiert, um die genaue Form übernehmen zu können. Die Gewichtskräfte des finalen Modells werden dadurch in VREP mit der tatsächlichen Geometrie des *Heron* berechnet, wobei allerdings noch immer die Vereinfachungen im Bezug auf die Auftriebskräfte (Quader-Näherung) gilt. Die beiden Schwimmkörper des Katamarans werden in VREP extrahiert bzw. freigelegt. Da über die originalen Gewichtsverteilungen des Heron keine Aussagen getroffen werden können, wird in diesem Modell angenommen, jeder Schwimmkörper würde sich wie ein Schiff mit den Daten aus Abschnitt 2.3 verhalten. Um nun mit den Objektumrandungen ein Wasserfahrzeug zu simulieren, werden mit Hilfe von Scripten in VREP selbst die in den vorigen Kapiteln besprochenen Kräfte eingefügt. Die beiden unabhängigen Schwimmkörper werden mit zwei gekoppelten Dummys verbunden, die dafür sorgen, dass die beiden Körper zu jeder Zeit ihren Abstand und ihre Orientierung zueinander beibehalten, egal welche Kräfte auf sie ausgeübt werden.

Damit das Modell anschließend auch grafisch sinnvoll aussieht, werden die beiden Schwimmkörper nur zur Berechnung in einer nicht angezeigten Ebene des Programms hinterlegt. Oberflächlich wird die vollständige Schiffsoberfläche über die beiden Schwimmkörper gelegt. Dies hat keinen Einfluss auf die Bewegung des Wasserfahrzeugs oder die Berechnungen des Programmes, es dient lediglich einer optischen Aufwertung. Des Weiteren werden auf Höhe des zuvor definierten Wasserspiegels bei $z=0$ einige Texturen eingefügt, die wie eine Wasseroberfläche aussehen. Die Abbildung 2.5 zeigt das dreidimensionale Modell des *Heron* innerhalb der Simulationsumgebung in VREP. Mit Hilfe von zwei unterschiedlichen Stellgrößen kann nun entlang der Schub-Achse des jeweiligen Schwimmkörpers eine Kraft auf das Modell ausgeübt werden. Dies entspricht dem beabsichtigten Aufbau eines Katamarans mit 2 Antriebsturbinen.

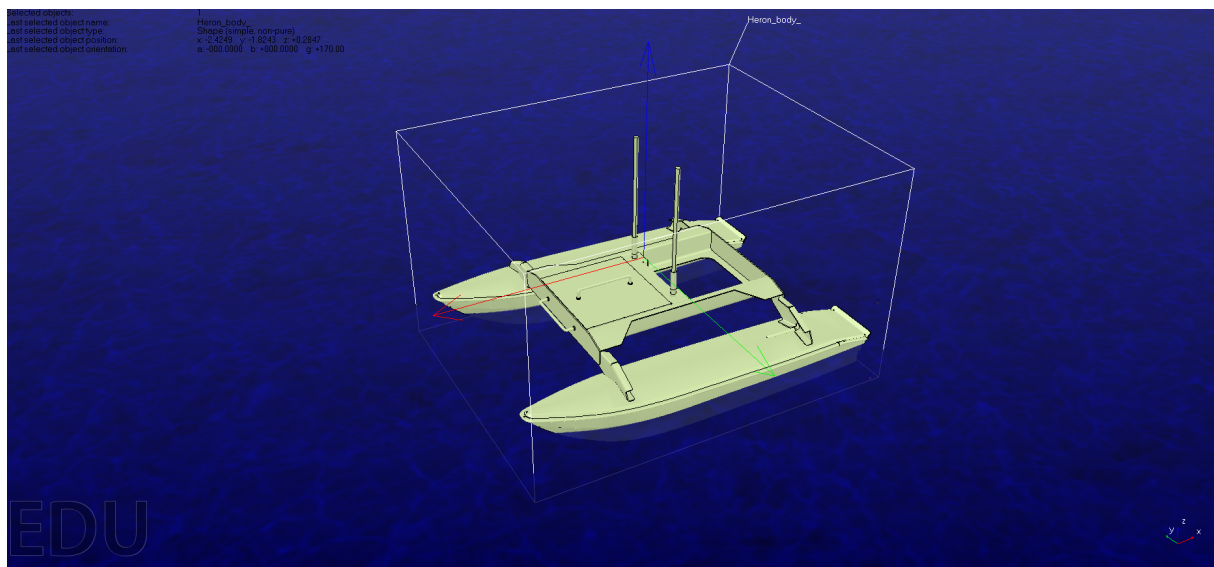


Abb. 2.5: Simulationsmodell des *Heron* in VREP

Kapitel 3

Verbindung zu MATLAB

In diesem Kapitel wird der Austausch von Daten und Simulationsparametern zwischen den Programmen MATLAB und VREP beschrieben. Eine Schnittstelle wie diese ist notwendig, da das Programm VREP in dieser Arbeit nur für das Erschaffen eines Simulationsraumes genutzt werden soll. Dies beinhaltet die in Kapitel 1 vorgestellten Modelle zur Schiffsdynamik. Der Antrieb sowie Steuer- und Regelungsalgorithmen sollen mit dem Programm MATLAB realisiert werden. Die Informationen zu dem Verhalten des Programms wurden dabei aus [1] übernommen.

3.1 Kommunikation

Die Kommunikation zwischen den Programmen wird durch einen virtuellen Server ermöglicht. Dieser wird einerseits durch mehrere Befehle in MATLAB gestartet. Andererseits muss in einem beliebigen Objekt der Simulation in der Initialisierungs-Funktion eines Child-Scripts ein Start-Befehl für VREP hinterlegt werden, damit einmalig der Verbindungs-Befehl aufgerufen wird. Der Initialisierungs-Befehl ermöglicht diverse Einstellungen. So können beispielsweise time-out-Zeiten oder einige verfügbare Verbindungs-Möglichkeiten eingestellt werden. Über diesen Server können die Programme nun miteinander kommunizieren. Dabei sind zwei generelle Modi möglich, eine asynchrone und eine synchrone Simulation. Grundsätzlich besteht der Unterschied zwischen diesen zwei Modi darin, dass im synchronen Modus innerhalb eines Simulationsschrittes die Programme aufeinander warten. Dies heißt im einzelnen, dass das Programm VREP einen Simulationsschritt startet. Dies sendet ein Trigger-Signal über den virtuellen Server an MATLAB. Dort werden wiederum alle Befehle und Berechnungen ausgeführt, die vor dem Trigger-Befehl im Programmcode von MATLAB stehen. Sobald jedoch dieses Trigger-Signal an V-REP zurück gesendet wurde, verarbeitet das Programm alle Daten innerhalb des Simulationsschrittes und startet, sobald alle Schritte abgeschlossen wurden, einen neuen Simulationsschritt. Beim asynchronen Modus hingegen laufen die beiden Programme unabhängig voneinander. Dies hat zur Folge, dass vor allem bei umfassenderen Rechnungen und schnellen Bewegungen in MATLAB eventuell falsche bzw. veraltete Werte abgerufen und berechnet werden. In jedem Fall ist also der synchrone Modus zwar langsamer, aber auch präziser. Da aber für Regler-Tests etc. Berechnungszeiten von mehreren Minuten nicht sinnvoll erscheinen und die Berechnungen in MATLAB nicht aufwändig sind, wird in dieser Arbeit der asynchrone Modus verwendet. Ein genauer Ablauf der Kommunikation im *streaming*-Modus in VREP ist in Abb. 3.1 gezeigt.

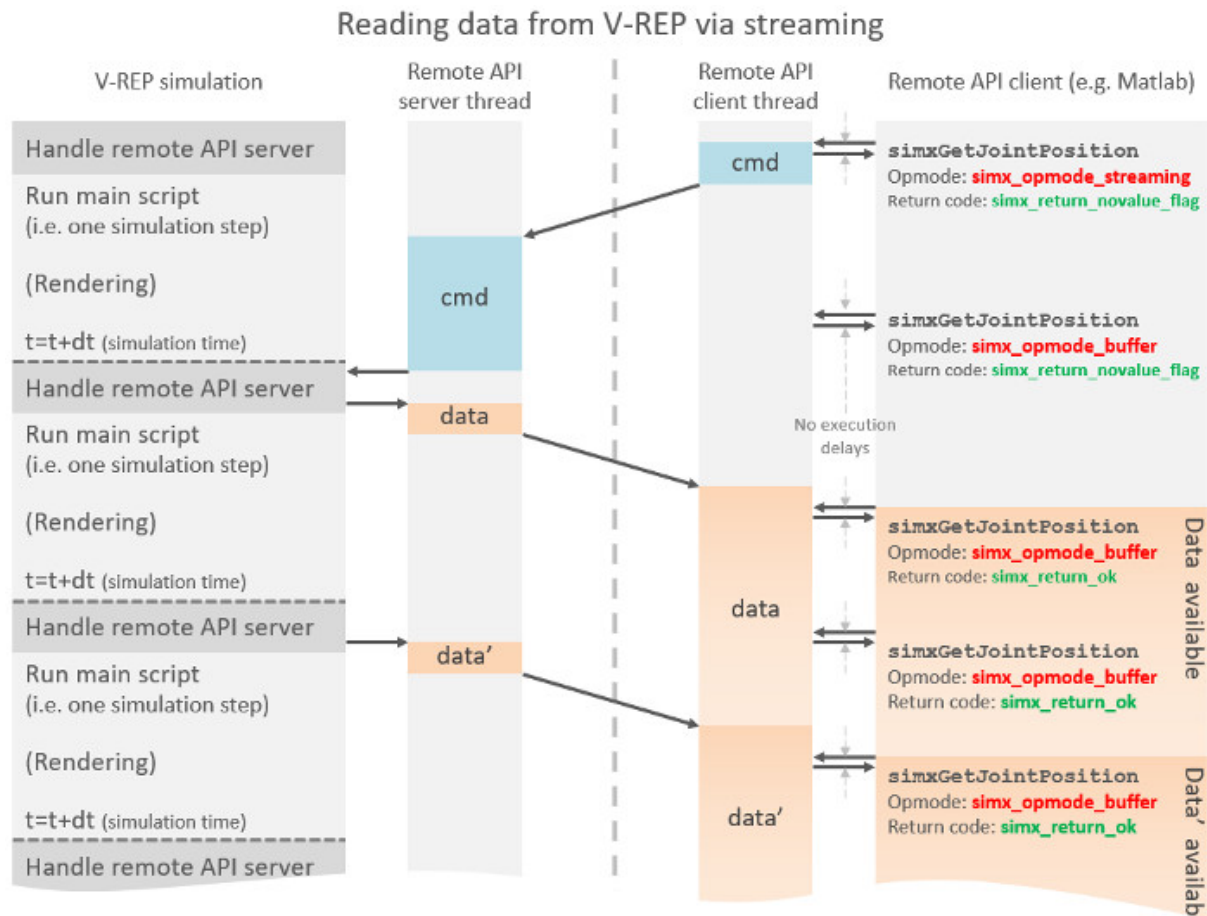


Abb. 3.1: Übersicht des Kommunikationsablaufs zwischen VREP und MATLAB nach [1]

3.2 Übermittelte Daten

Um die in Sektion 3.1 beschriebene Verbindung sinnvoll zu nutzen, wird mittels eines Remote-Startbefehls die Simulation in VREP durch das Programm MATLAB gestartet. Es ist nun innerhalb der Verbindung zwischen MATLAB und VREP möglich, mit bestimmten Remote-Befehlen Simulationsdaten als Signal zu übermitteln. Allgemein ist die Übermittlung in den Formaten Integer, Float und String möglich. Außerdem können einige Parameter der einzelnen Objekte ebenfalls über ihre Handle abgerufen werden. Da für das hier vorgestellte Modell die aktuelle Position sowie ein Ausrichtungswinkel übermittelt werden sollen, ist es am naheliegendsten, die Position des Schiffes, welche ein Objektparameter ist, über den Handle des Objektes abzurufen. Der aktuelle Ausrichtungswinkel ψ kann hingegen als einfacher Float-Wert übermittelt werden. Er ist kein Objektparameter, da es sich hierbei um einen aus der Rotationsmatrix des Objektes errechneten Wert handelt. Näheres dazu ist in der Sektion 2.6 zu finden. Um Daten zu übermitteln stehen wieder verschiedene Übertragungs-Optionen zur Verfügung. Es kann generell in blockende und nicht blockende Übertragungen unterschieden werden. Wie die Namen schon vermuten lassen, wird bei blockenden Übertragungen so lange gewartet, bis ein entsprechender Wert übermittelt wurde. Bei nicht blockenden Verfahren wird hingegen einfach mit dem Programmcode in MATLAB fortgefahren, egal wie lange es dauert, bis ein Wert vorliegt. Da sich das Modell fortlaufend bewegt, scheint in dem gewählten asynchronen Modus die *streaming*-

Option am naheliegendsten. Hierbei wird der Wert von VREP an MATLAB bzw. von MATLAB an VREP jederzeit – und damit auch in jedem Simulationsschritt aktuell – bereitgestellt und kann nach Notwendigkeit der Programme verwendet werden. Für das Modell und den in Kapitel 4 vorgestellten Regler werden die beschriebenen Daten übermittelt. Einerseits werden von MATLAB die VREP-Daten der Position und des Ausrichtungswinkels abgerufen. Dabei ist auf eine einmalige Initialisierung des Daten-Streams gefolgt von einem Streaming-Befehl zu achten. Die Position wird dann also über die Verzweigung des Handles in die Positions-Variablen in MATLAB gestreamt, der Ausrichtungswinkel ist als Float-Stream vorliegend. Wie ausführlicher in Kapitel 4 erläutert wird, sendet MATLAB im Anschluss an eine Reihe von Berechnungen zwei Float-Werte an VREP, die die Kraft widerspiegeln, die die Turbinen des Schiffes ausüben sollen. Dazu wird seitens MATLAB wiederum ein Streaming-Signal initialisiert und in VREP in einem an das entsprechende Objekt - der linke oder der rechte Schwimmkörper - geknüpfte Child-Script durch einen Empfangs-Befehl abgerufen. Es folgen also für die übermittelten Daten die Vektoren

$$\boldsymbol{\lambda}_{0,1} = [x \ y \ \psi]^T, \quad (3.1)$$

$$\boldsymbol{\lambda}_{1,0} = [F_R \ F_L]^T. \quad (3.2)$$

Dabei steht der Index 0 für die VREP-Seite und der Index 1 für die MATLAB -Seite der Übertragung. Somit beschreibt der Vektor $\boldsymbol{\lambda}_{0,1}$ also die Übertragung von VREP an MATLAB und der Vektor $\boldsymbol{\lambda}_{1,0}$ die Übertragung von MATLAB an VREP. Die von VREP an MATLAB übermittelten Werte bilden die grundlegenden Informationen aus dem Modell in 2.3.1. Ein Verweis zum vollständigen Programm-Code in MATLAB und VREP ist im Anhang zu finden.

Kapitel 4

Reglerentwurf und Simulation eines Line-Of-Sight-Reglers

In diesem Kapitel wird ein Ansatz zur Regelung eines Wasservehikels vorgestellt, auf das in den Kapiteln 1 und 2 aufgezeigte Beispiel-Modell angewandt und mittels der in Kapitel 3 beschriebenen Schnittstelle aus dem Programm MATLAB an die Simulationsumgebung in VREP übertragen. Motivation für dieses Vorgehen ist es, die Funktionstüchtigkeit des entworfenen Modells mitsamt der Schnittstelle und dem im folgenden Kapitel vorgestellten Reglers zu testen.

4.1 Reglerentwurf eines Line-Of-Sight-Reglers

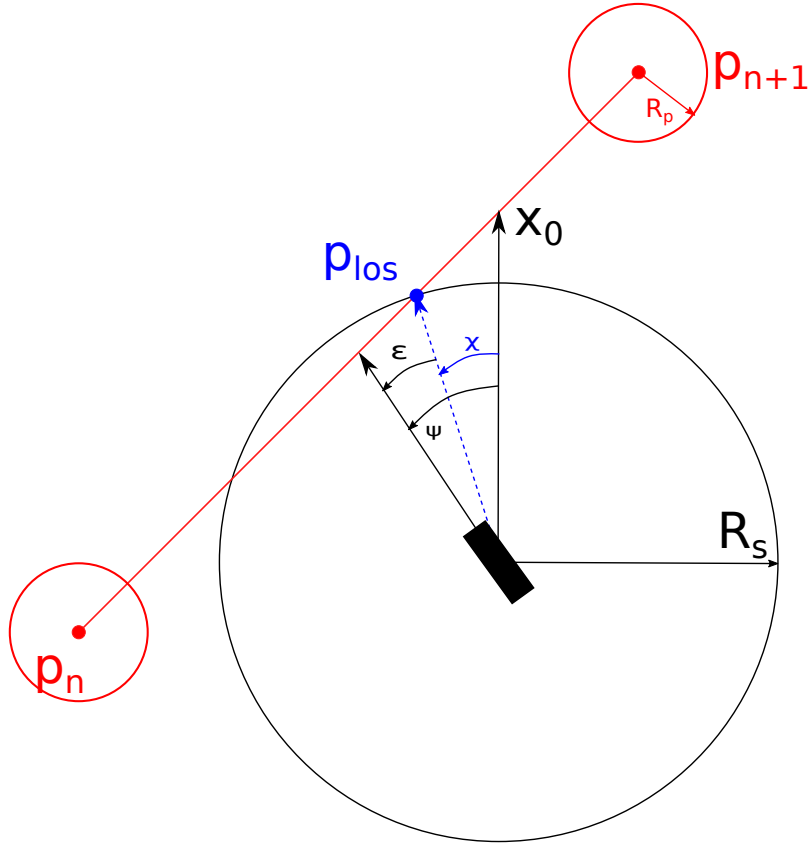
Wie auch bei den vorherigen Kapiteln werden zunächst einige Grundlagen vorgestellt und erklärt, um die dann folgende Anwendung des Line-Of-Sight-Reglers in einer Simulation übersichtlich zu halten. Es wird im folgenden für Line-Of-Sight das Akronym *LOS* verwendet.

4.1.1 Grundprinzip des Line-Of-Sight-Reglers

Die grundsätzliche Idee einer LOS-Regelung ist in 4.1 abgebildet und entstammt in ihren Grundzügen [2], [3] und [4]. Es wird zunächst angenommen, dass es eine geplante Strecke gibt, die von einem Vehikel abgefahren werden soll. Diese Strecke wird in diesem Beispiel durch n Wegpunkte mit $p_n = (x_n|y_n)$ dargestellt, die allesamt in der X-Y-Ebene liegen. Es entsteht also eine Abfolge von Punkten, die sinnvollerweise in der Form

$$P = \begin{bmatrix} x_0, x_1, \dots, x_n \\ y_0, y_1, \dots, y_n \end{bmatrix} = [p_0, p_1, \dots, p_n] \quad (4.1)$$

dargestellt werden kann. Ziel des LOS-Reglers ist es, das Vehikel entlang eines aufgestellten Polygonzugs zu manövrieren. Ein Polygonzug ist per Definition die Zusammensetzung von endlich vielen Geradenstücken, die an Wegpunkten verbunden sind. Wird eine imaginäre Gerade durch den Ausgangspunkt p_0 und den ersten Zielpunkt p_1 gezogen, so bildet sich mit dem Geradenstück zwischen den beiden Punkten die erste Gerade des Polygonzugs. Wiederholt man dies nun für alle in P eingetragenen Punkte, so erhält man den vollständigen gewünschten Polygonzug. Ausgehend davon, dass das Fahrzeug einen Abstand R_s zu einem Punkt auf dieser imaginären Linie aufweist, kann nun ein Winkel bestimmt werden, der auf den Schnittpunkt zwischen dem

Abb. 4.1: Veranschaulichung des *LOS*-Konzepts.

Kreis um das Fahrzeug mit Radius R_s und der imaginären Gerade deutet. Jener Kreis kann nun mit der Gerade null, ein oder zwei Schnittpunkte haben. Bei null Schnittpunkten ist der Radius des Schiffes zu klein gewählt, weshalb die angestrebte Gerade nicht im Detektionsradius liegt. Bei genau einem Schnittpunkt ist keine weitere Fallunterscheidung nötig. Bei zwei Schnittpunkten ist allerdings darauf zu achten, dass immer der Schnittpunkt ausgewählt wird, welcher eine Bewegung in Richtung des Zielpunkts und nicht in Richtung des Ausgangspunkts zur Folge hat. Der entsprechende Schnittpunkt wird als *LOS-Punkt* $p_{\text{los}} = (x_{\text{los}}|y_{\text{los}})$ bezeichnet. Er ist abhängig von der momentanen Position des Fahrzeugs, sodass er sich kontinuierlich ändert. Für die Berechnung des *LOS-Punktes* werden die Gleichungen

$$[x_{\text{los}} - x(t)]^2 + [y_{\text{los}} - y(t)]^2 = R_s^2 \quad (4.2a)$$

$$\frac{y_{\text{los}} - y_n}{x_{\text{los}} - x_n} = \frac{y_{n+1} - y_n}{x_{n+1} - x_n} \quad (4.2b)$$

gemäß [4] aufgestellt und nach x_{los} und y_{los} aufgelöst. Die Variable $x(t)$ bezeichnet dabei die momentane Position des Schiffes, die Punkte x_n und x_{n+1} bezeichnen den Ausgangspunkt und den momentanen Zielpunkt. Die Gleichung (4.2a) stellt dabei eine einfache Abstandsbestimmung dar, Gleichung (4.2b) stellt sicher, dass die Steigung in besagtem Punkt auch der Steigung der Geraden entspricht. Beim Auslösen müssen verschiedene Fallunterscheidungen vorgenommen werden, um nicht definierte Fälle – beispielsweise $x_{\text{los}} = x_n$ – ebenfalls berechnen zu können. Eine Ausführung darüber ist im Einzelnen in [4] nachzulesen. Auch ist hier die Fallunterscheidung für die oben genannten Schnittpunkte zu finden. Nach Bestimmen des *LOS-Punktes* ist eine

Berechnung des entsprechenden Winkels möglich, welcher mit Hilfe der Gleichung

$$\chi = \text{atan2}(y_{\text{los}} - y(t), x_{\text{los}} - x(t)) \quad (4.3)$$

berechnet wird. Der Winkel χ sei also jener, welcher vom Fahrzeug auf den *LOS-Punkt* zeigt. Die Differenz zwischen dem momentanen Ausrichtungswinkel des Fahrzeugs ψ und dem *LOS-Winkel* χ wird auch als *Fehlerwinkel* bezeichnet und ist als

$$\varepsilon = \psi - \chi \quad (4.4)$$

definiert. Um die Funktion der Steuerung in allen *LOS-Winkeln* zu gewährleisten, ist auf eine Besonderheit bei der Berechnung des Fehlerwinkels ε zu achten. Diese tritt auf, sobald das Schiff sich in negativer x-Richtung bewegt. Wie in der Gleichung (4.3) gezeigt wird, ist die Berechnung des Fehlerwinkels mittels der *atan2*-Funktion realisiert. Diese weist die Besonderheit auf, dass im Gegensatz zur herkömmlichen Umkehrfunktion des Tangens alle vier Quadranten, also 360° abgedeckt werden. Dabei entsteht ein Übergang zwischen $-\pi$ und π , der für eine Problematik bei der Berechnung des Winkels χ sorgt. In dem Bereich dieses Übergangs kann es dazu kommen, dass χ und ψ verschiedene Vorzeichen erhalten und sich so negieren beziehungsweise aufaddieren. Ist beispielsweise im Extremfall der Winkel χ nahezu gleich π und $\psi = \text{nahezu } -\pi$, so würden sich diese zu -2π aufaddieren. Dieser Wert entspricht aber im System des Schiffes einer Ausrichtung zu 0° anstatt der gewünschten Ausrichtung zu $\pm 180^\circ$, was zu einer Drehung um 180° führt, anstatt nur die geringe Differenz der beiden Winkel auszugleichen. Daher muss hier mittels einer Fallunterscheidung geklärt werden, aus welcher Richtung sich beide Winkel dem Übergang nähern oder entfernen, um sicherzustellen, dass sowohl das Vorzeichen als auch der Betrag des Winkels dem gewünschten Ergebnis entsprechen. Im Rahmen dieser Arbeit wurde dies durch mehrere Bedingungen in MATLAB realisiert. Zur Manövrierung zum bestimmten *LOS-Punkt* wird ein einfacher PID-Regler verwendet. Dieser berechnet in Abhängigkeit des Fehlerwinkels ε und dessen Winkelgeschwindigkeit ein Drehmoment, um das Fahrzeug in die Richtung des *LOS-Punkts* zu drehen. Der PID-Regler entspricht der Form

$$\Gamma_r = -K_p \varepsilon - K_d \dot{\varepsilon} - K_i \int_0^t \varepsilon d\tau. \quad (4.5)$$

Genauereres dazu wird in Abschnitt 4.1.2 ausgeführt. Sobald sich das Fahrzeug mit Hilfe der Regler-Ansteuerung innerhalb eines Radius R_p um den vorläufigen Zielpunkt befindet, gilt dieser Punkt als erreicht. Es wird dann der gerade erreichte Punkt zum neuen Ausgangspunkt und es wird - falls verfügbar - der nächste Wegpunkt verarbeitet. Generell ist es auch möglich, mit einer Positionsabfrage des Fahrzeugs zu starten, den nahegelegensten Punkt einer Strecke zu finden und dementsprechend dann den Radius vorerst anzupassen, damit das Fahrzeug sich auch außerhalb des vorgegebenen Umgebungsradius auf die vorgegebene Strecke begibt. Da in diesem Beispiel jedoch eine vorgegebene Strecke abgefahren werden soll, ist diese zusätzliche Funktion überflüssig.

4.1.2 Anwendung auf das Beispiel des Heron

Das Konzept des *LOS-Punktes* in Verbindung mit einem PID-Regler ermöglicht es dem in dieser Arbeit vorgestellten Schiffsmodell eine Folge von Wegpunkten abzufahren. Gemäß den Ausfüh-

rungen in Kapitel 2 verfügt das Schiff über zwei Antriebsturbinen, welche getrennt voneinander angesteuert werden können. Da diese von vornherein im Modell mit einem Kraft-Input arbeiten, muss lediglich eine Leistungs-Begrenzung für diese Turbinen getroffen werden. Nach Datenblatt des Herstellers liegt diese für eine einzelne Turbine des *Heron* bei $L_{\max} = 20$ N. Außerdem ist auf Grund der Bauweise nur eine positive Kraft möglich. Es ergibt sich also ein Output-Vektor der Form

$$F_{out} = \begin{bmatrix} F_r \\ F_l \end{bmatrix} \quad (4.6)$$

mit

$$0 \leq F_r, F_l \leq 20, \quad (4.7)$$

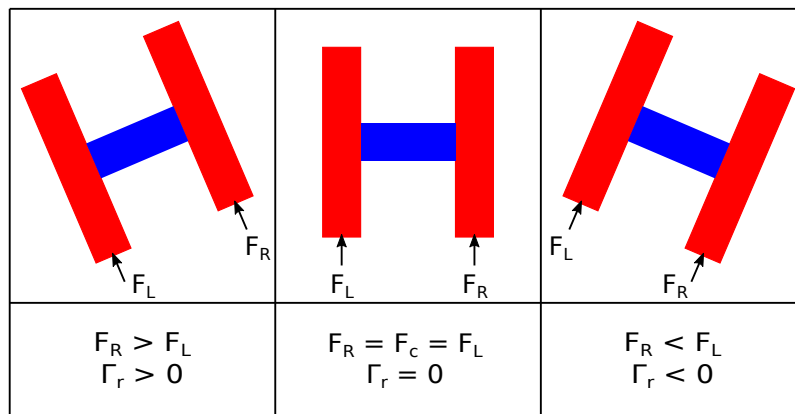
wobei die Kräfte F_r und F_l die Kraft der rechten bzw. linken Turbine angeben. Für die Zwecke eines geregelten Fahrbetriebs bietet es sich allerdings an, nicht das Maximum an Leistung für beliebige Polygonzüge zu verwenden, weshalb das Maximum der einzelnen Turbine zunächst auf die Hälfte begrenzt wird. Die Kurvenfahrt durch die Turbinen ist durch Differenzen in der Schubkraft der Turbinen möglich. Um die Fahrtrichtung nach rechts zu ändern, muss $F_r < F_l$ sein, für eine Fahrt nach links entsprechend $F_r > F_l$. Veranschaulicht ist dieser Zusammenhang in der Abbildung 4.2. Die Pfeile deuten dabei nicht den Betrag der Kräfte an, sondern wie schon zuvor im VREP-Modell nur auf den Punkt, an dem die Schubkraft ausgeübt wird. Genannter Zusammenhang führt zu dem Regler-Entwurf

$$\begin{bmatrix} F_r \\ F_l \end{bmatrix} = \begin{bmatrix} F_c + \Gamma_r \cdot h \\ F_c - \Gamma_r \cdot h \end{bmatrix}, \quad (4.8)$$

wobei das durch den PID-Regler bereitgestellte Γ_r in diesem Beispiel ohne Integratoranteil berechnet wird. Es gilt also $K_i = 0$. Der Regler hat somit die Form

$$\Gamma_r = -K_p \varepsilon - K_d \dot{\varepsilon}. \quad (4.9)$$

Der Faktor h entspricht dem Hebelarm, also dem Abstand des Schwimmkörpermittelpunktes zum Mittelpunkt des gesamten Modells. Multipliziert mit dem berechneten Drehmoment ergibt dies eine Kraft. F_c sei definiert als eine konstante Schubkraft. Demnach ist bei einer geraden Fahrt der Fehlerwinkel $\varepsilon = 0$ und nur die konstante Schubkraft F_c wird von den Turbinen ausgeübt. Sobald allerdings die Richtung geändert werden soll, wird durch den PD-Regler eine weitere Kraft addiert bzw. subtrahiert, die dafür sorgt, dass eine Fahrt in Richtung des gewünschten Ziels möglich wird. Die Vorzeichen der Fahrtrichtungen decken sich in diesem Fall mit dem zuvor gewählten Koordinatensystem für das Schiffsmodell, wobei der Ausrichtungswinkel positiv drehend entgegen dem Uhrzeigersinn definiert wurde. Die einzelnen Teile des vorgestellten Regler-Konzepts können nun also - wie im Diagramm 4.3 dargestellt - zusammengefügt werden. Wie in den obigen Gleichungen gezeigt, wird aus den vorgegebenen Wegpunkten p_n und p_{n+1} zusammen mit der aktuellen Position des Schiffes ein *LOS-Punkt* berechnet. Aus diesem lässt sich dann weiter ein Winkel χ ermitteln, der als Führungsgröße in den Regler eingegeben wird. Aus der Regelgröße - also der zeitabhängigen Ausrichtung des Schiffes ψ - wird zusammen mit der Führungsgröße ein Fehlerwinkel ε berechnet und in den PD-Regler gegeben. Dieser berechnet nun das definierte Γ_r , welches dann zu einer Kraft umgerechnet wird. Gemäß Gleichung (4.8) werden die beiden Kräfte für die linke und rechte Turbine bestimmt und anschließend durch die

Abb. 4.2: Veranschaulichung des Kräfteverhältnisses zwischen F_r und F_l .

Turbinen des Schiffes ausgeübt. Anschließend wird die Regelgröße ψ erneut abgefragt und der Prozess wiederholt sich.

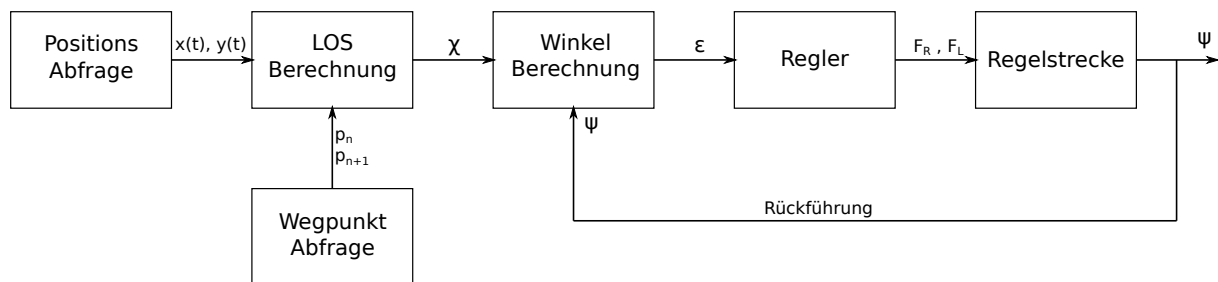


Abb. 4.3: Blockdiagramm der beschriebenen Prozesse zur LOS-Regelung.

4.2 Simulation und Auswertung des Line-Of-Sight-Reglers

Zur Simulation des vorgestellten Reglers werden drei verschiedene Szenarien gewählt, mit Hilfe von MATLAB aufgezeichnet und anschließend ausgewertet. Diese zeigen den Ausgleich eines Offsets zu Beginn der Simulation, das Abfahren eines Polygonzugs sowie zuletzt noch den Einfluss der Radien R_p und R_s . In allen drei Szenarien werden die selben Regler-Parameter, also K_p und K_d verwendet. Diese wurden empirisch festgestellt und müssen je nach Anwendung neu bestimmt werden. Beispielsweise kann bei kleineren Radien ein reaktionsschneller Regler von Vorteil sein, um ausladende Kurven gering zu halten, was allerdings bei größeren Radien zu starkem Überschwing-Verhalten führen kann.

Die verwendeten Diagramme haben alle den selben Aufbau und werden daher in diesem Zusammenhang erklärt. Das jeweilige Diagramm oben links ist eine zweidimensionale Veranschaulichung der X-Y-Ebene. Dabei ist mit *rot* der Verlauf der Position des Schiffes markiert. Der Polygonzug ist in *blau* gezeichnet. In *schwarz* ist hier der Detektionsradius der Wegpunkte eingezeichnet.

Das Diagramm oben rechts zeigt den Verlauf der vom Regler berechneten und durch die Turbinen ausgegebenen Kräfte über die Simulationszeit. In *rot* ist dabei F_r markiert, F_l ist durch die

Farbe *blau* gekennzeichnet.

Das untere Diagramm zeigt schließlich den Verlauf des Ausrichtungswinkels in *blau*, des *LOS*-Winkels in *grün* und des Fehlerwinkels in *rot* über die Simulationszeit. Dabei sind die Winkel in rad angegeben.

4.2.1 Szenario A

In diesem Szenario enthält die Wegpunktmatrix M nur zwei Punkte, sodass sich für dieses Szenario

$$P = \begin{bmatrix} 0 & 0 \\ 0 & 5 \end{bmatrix} \quad (4.10)$$

ergibt. Das Modellschiff startet beim Punkt $p = (0 | -0.5)$, also einem Offset von 50 Zentimetern, mit einem Auslenkungswinkel von $\psi_0 = \pi$. Die Radien sind gewählt zu $R_s = 2$ und $R_p = 1$. Die Motivation des Szenarios besteht darin, dass das Schiff zunächst eine einfache Strecke problemlos erkennen, den entsprechenden Offset überwinden und im Abstand von R_p zum Stehen kommen soll. Wie die Abbildung 4.4 zeigt, ist das angenommene Resultat erreicht. Der

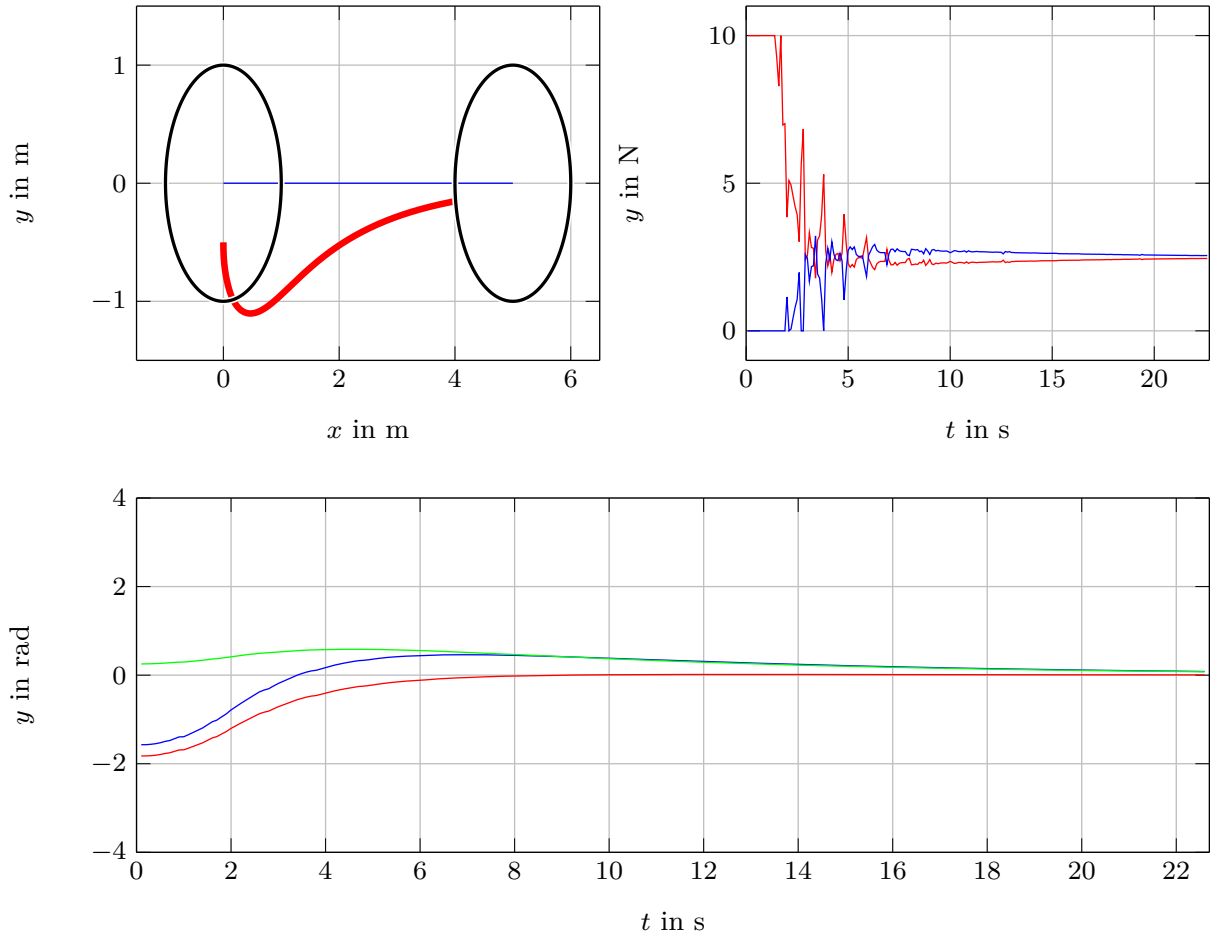


Abb. 4.4: Simulationsergebnisse Szenario A.

Ausrichtungswinkel nähert sich dem *LOS*-Winkel an. Der Fehlerwinkel nähert sich 0 an und erreicht diesen Wert zu dem Zeitpunkt, wo Ausrichtungs- und *LOS*-Winkel aufeinandertreffen. Das Kräfte-Diagramm zeigt deutlich, dass bei der starken Linkskurve zum Offset-Ausgleich zu Beginn der Simulation auch die Kraft der rechten Turbine (dargestellt in *rot*) deutlich höher ist als die der linken Turbine. Zuletzt verifiziert auch das Positionsdiagramm die Orientierung zum Polygonzug und ein Abstoppen beim Detektionsradius des Zielpunktes.

4.2.2 Szenario B

In Szenario B erhält das Schiff nun eine Abfolge von insgesamt fünf Punkten, welche in diesem Fall mit der Matrix

$$P = \begin{bmatrix} 0 & 5 & 5 & 0 & 0 \\ 0 & 0 & 5 & 5 & 0 \end{bmatrix} \quad (4.11)$$

ein Quadrat bilden sollen, sodass das Schiff wieder am Startpunkt, bzw. im Abstand R_p zum Startpunkt ankommt. Die Radien werden gewählt wie in Szenario 4.2.1. Motivation für dieses erweiterte Szenario ist das folgerichtige Einlesen der Wegpunkte. Außerdem ist zu erwarten, dass der Ausgleich des kleinen Start-Offsets auch in dieser Abbildung überwunden wird. Wie erwartet

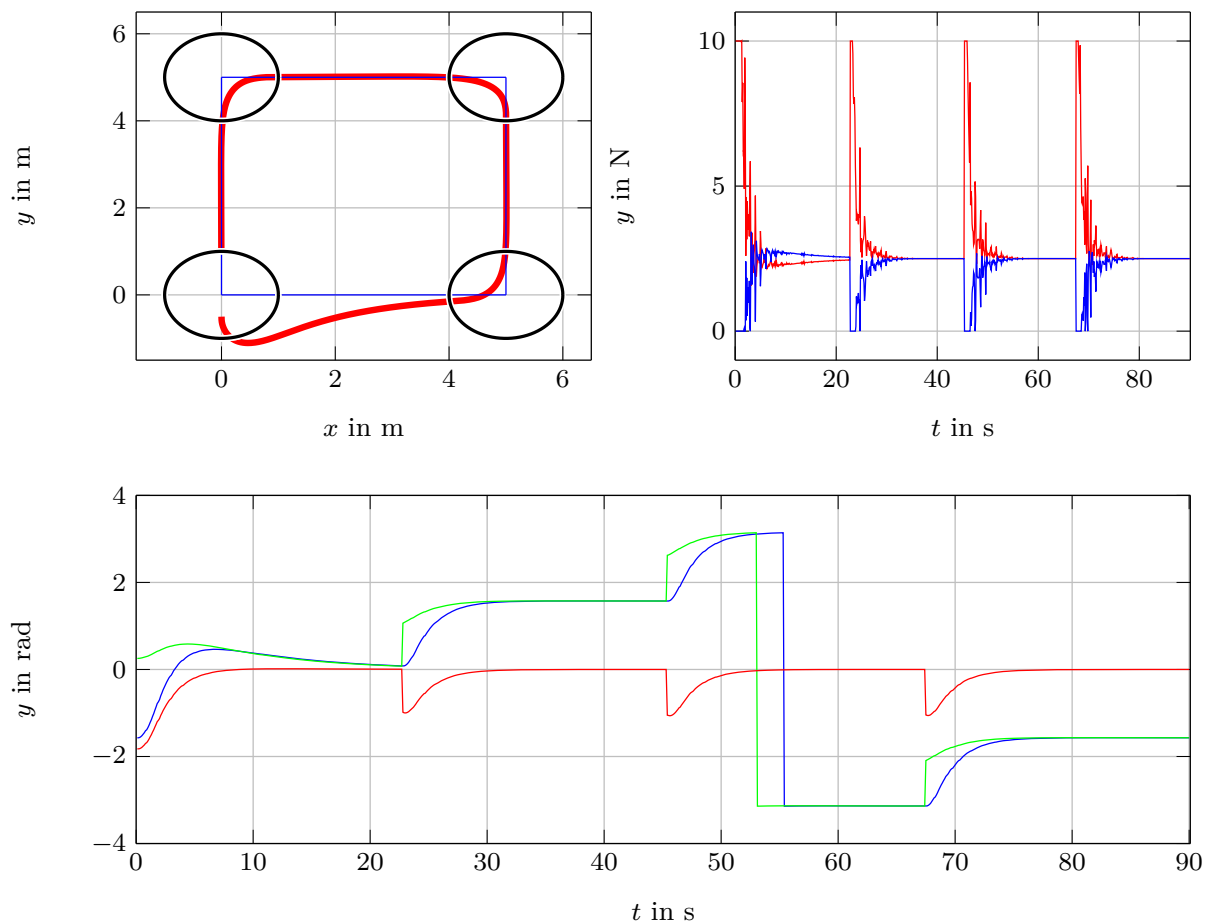


Abb. 4.5: Simulationsergebnisse Szenario B.

folgt auch in diesem Szenario (siehe Abbildung 4.5) das Schiff der geplanten Punktabfolge. Im Vergleich zu Szenario 4.2.1 ist jedoch vor allem im Winkel-Diagramm zu erkennen, wie sich die Winkel an die vier gewählten Zielwinkel anpassen, also $(0, \frac{\pi}{2}, \pi$ bzw. $-\pi$ und $-\frac{\pi}{2}$, während sich der Fehlerwinkel wie erwartet immer wieder 0 annähert. Die Sprünge kennzeichnen hier deutlich den Zeitpunkt, zu dem ein neuer Wegpunkt eingelesen wird und damit eine neue LOS-Gerade entsteht. Der Ausrichtungswinkel ist hingegen eine Messgröße, die sich nicht sprunghaft ändern kann. Die bereits angesprochene Besonderheit des Übergangs zwischen π und $-\pi$ ist im Winkel-Diagramm an der Stelle erkennbar, wo der Ausrichtungswinkel und der textitLOS-Winkel einen Sprung von π zu $-\pi$ machen. Die in MATLAB implementierte Fallunterscheidung erfüllt ihren Zweck wie vorgesehen.

4.2.3 Szenario C

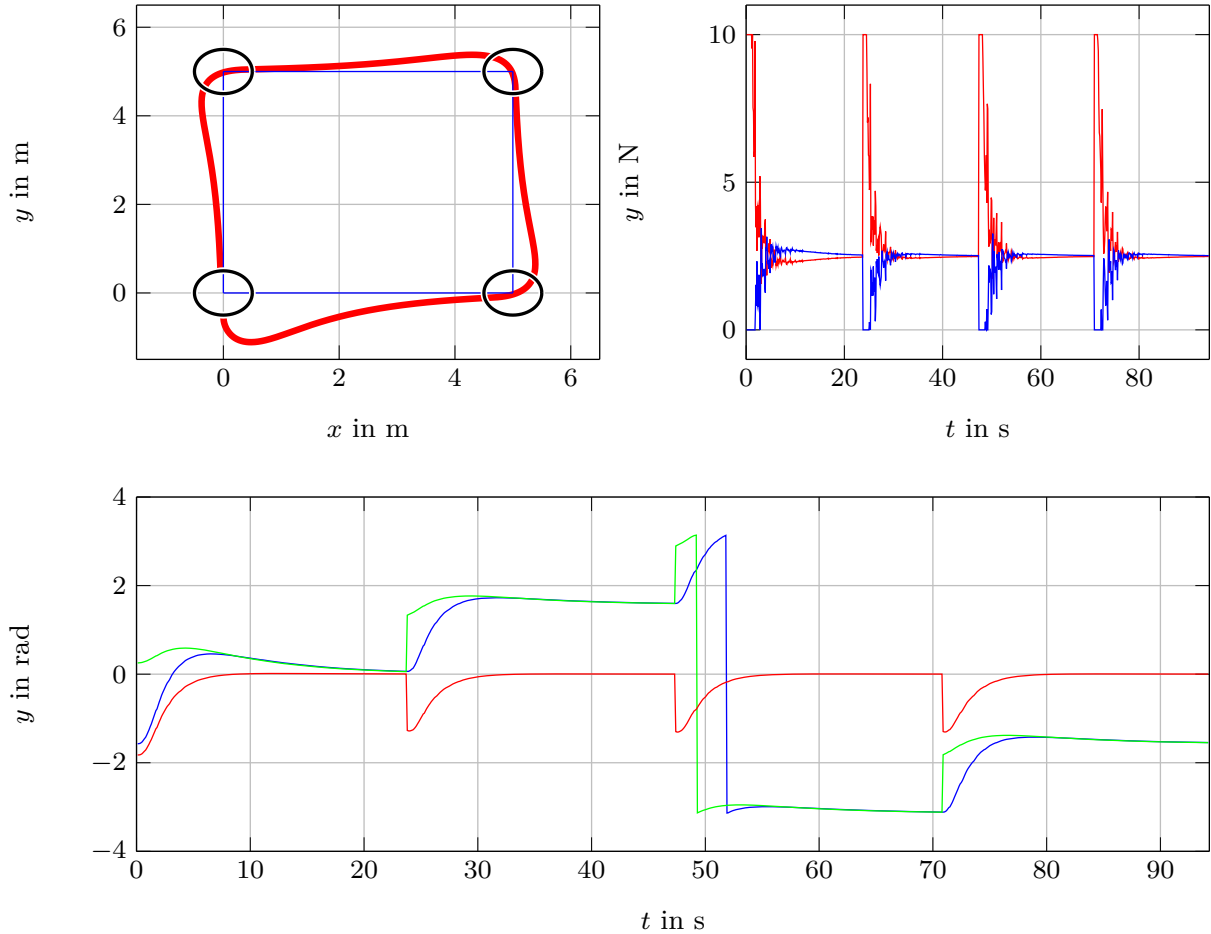


Abb. 4.6: Simulationsergebnisse Szenario C mit $R_s = 2$ und $R_p = 0.5$.

In diesem Szenario wird nun der Einfluss der Radien R_s und R_p diskutiert. Diese haben ebenso wie die Regler-Parameter einen Einfluss auf das Fahrverhalten des Modells. Zunächst darf trivialerweise der Startpunkt des Schiffes immer maximal dem Abstand R_s entsprechen, da das Schiff sonst keine Bahn detektieren kann. Ferner gilt auch, dass die Radien nicht größer sein

dürfen, als die eigentliche Bahn, da sonst in jedem Punkt angenommen wird, dass das Schiff bereits am Punkt angekommen ist. Schließlich muss noch

$$R_s \geq R_p \quad (4.12)$$

gelten. Dies ist leicht einsehbar, da ein zu kleiner Radius R_s zwar die Ankunft am ersten Punkt garantiert, allerdings beim Umschalten auf den nächsten Wegpunkt nicht garantieren kann, dass die Gerade zwischen diesen Punkten auch im Detektionsradius des Schiffes liegt. Geprüft wird also lediglich, was kleinere Unterschiede der Radien zur Folge haben. Wie das Positions-Diagramm aus Abbildung 4.6 zeigt, wird durch den im Vergleich zu Sektion 4.2.2 halbierten Radius R_p eine weite Kurve gefahren. Dies liegt daran, dass das Schiff in seiner Turbinenleistung und der Reaktionszeit der Kommunikation zwischen den Programmen Matlab und VREP begrenzt ist. Es zeigt sich also, dass die in Sektion 4.2.2 gewählten Werte für die Radien gut auf die Geschwindigkeit des Schiffes angepasst sind. Das Winkel-Diagramm zeigt außerdem genau wie das Positions-Diagramm, dass sich *LOS-Winkel* und Ausrichtungswinkel mit einem Überspringen den erwarteten Werten nähern. Die Abbildung 4.7 das Verhalten des Schiffes, wenn R_s verändert wird. Die Verkleinerung von $R_s = 1.2$ bei $R_p = 1$ hat, wie das Positionsdiagramm anschaulich beweist, zur Folge, dass das Schiff schneller den Offset zur vorgesehenen Bahn überwindet. Dieses Verhalten ist in Einklang mit den geometrischen Überlegung für die Berechnung des *LOS-Punktes* aus Sektion 4.1.1, da aus einem nähergelegenen *LOS-Punkt* auch ein steilerer Winkel χ folgt, der vom Schiff anvisiert wird. Gleiches Verhalten lässt sich auch aus dem Winkel-Diagramm entnehmen. Wie deutlich ersichtlich ist, nähern sich Ausrichtungswinkel und *LOS-Winkel* deutlich schneller bzw. steiler dem gewünschten Zielwinkel an, während auch der Fehlerwinkel sehr viel weniger Zeit benötigt um 0 zu erreichen. Aus dieser Art der Konfiguration der Radien folgt also ein deutlich besseres Fahrverhalten des Schiffes.

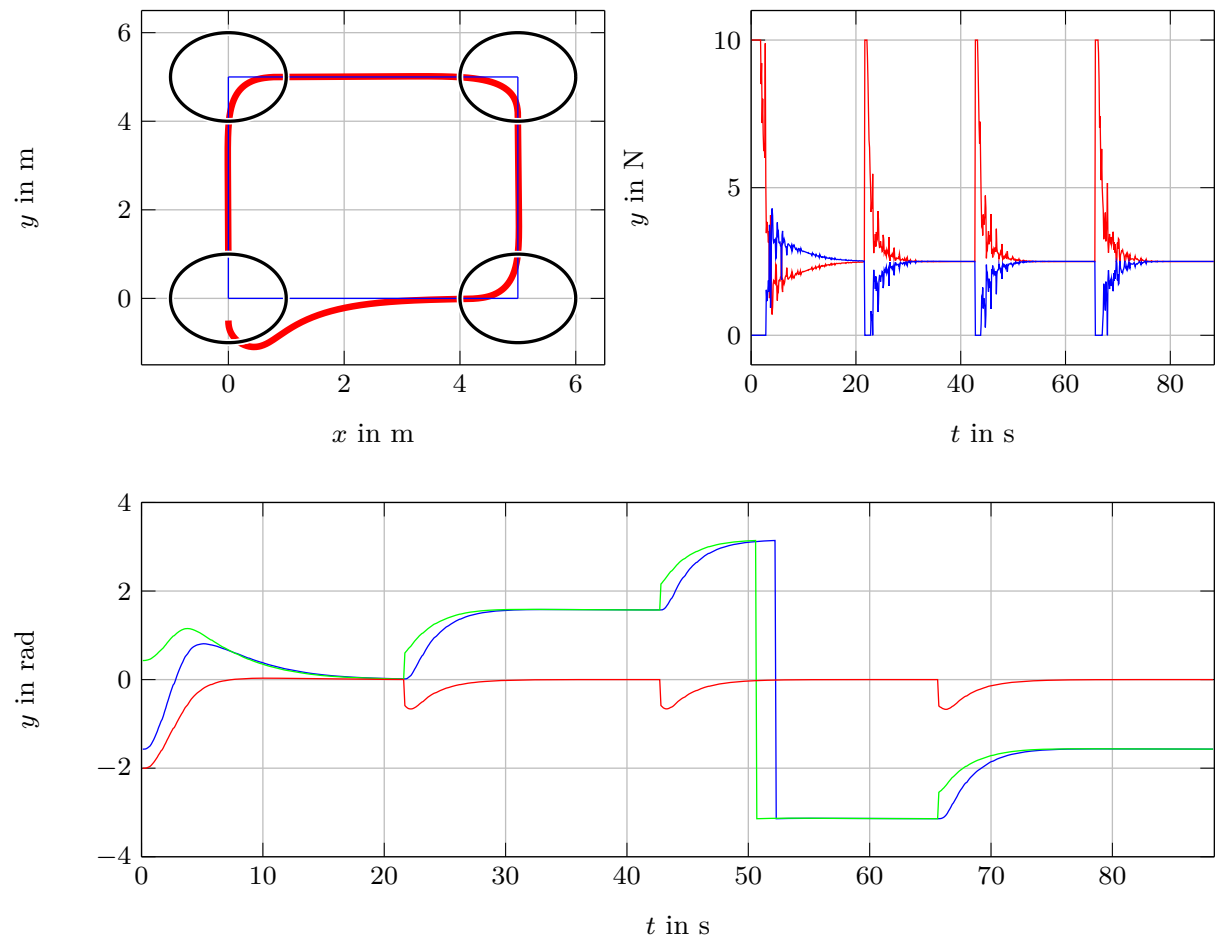


Abb. 4.7: Simulationsergebnisse Szenario C mit $R_s = 1.2$ und $R_p = 1$.

Kapitel 5

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde vorrangig ein Modell für einen Katamaran erstellt, welcher in seinen Grundzügen dem *Heron* der Firma *Clearpath* entspricht. Für die Erstellung eines entsprechenden Modells in der Simulationssoftware VREP wurden auf Grund von fehlenden Daten des praktischen Vorbildes einige Parameterdaten von vergleichbaren Objekten verwendet und dieses Modell durch voneinander entkoppelte Modelle in seinen Freiheitsgraden erweitert bzw. ergänzt. Unter Berücksichtigung von Näherungen in der Berechnung einiger Modellkräfte liegt somit eine Simulationsumgebung vor, die durch die dafür entwickelte Remote-Schnittstelle durch MATLAB gesteuert werden kann. Auch ein Austausch von weiteren Parametern und Daten zwischen den Programmen ist möglich. Die Funktionstüchtigkeit des entwickelten Modells wurde dabei durch einen dafür entworfenen *LOS*-Regler überprüft. Die vorgestellten Simulationsergebnisse haben gezeigt, dass sowohl der Regler und dessen Berechnungen als auch das Modell funktionstüchtig sind.

Verbesserungen dieses Modells sind durch eine Vielzahl von Möglichkeiten zu erzielen. Zur Erstellung eines vollständigen 6-DOF-Modells könnte so ein Parametertest durchgeführt werden. Dazu würde es jedoch ein physisches Modell erfordern, einen geeigneten Testraum (Schlepptank etc.), sowie Equipment zur Aufnahme der entsprechenden Werte. Des Weiteren würde sich die Genauigkeit des Modells ebenfalls durch eine exakte Bestimmung der Auftriebskräfte (Quader-Näherung) verbessern lassen. Außerdem könnte das Modell in seinem Verhalten weiter auch durch die bislang vernachlässigten Kräfte im Roll-Winkel ergänzt werden. Dies wäre vor allem dann von Belang, wenn ferner externe Einflüsse wie Wind, Wellengang und Wasserverwirbelung eingespeist und berücksichtigt werden sollen. Im Bereich der Schnittstelle könnten bei der Kommunikation der beiden Programme Verbesserungen erzielt werden. So könnte beispielsweise eine andere Programmiersprache und Schnittstelle dazu eine bessere Performance bieten als MATLAB. Außerdem könnte durch die Erhöhung der verfügbaren Rechenleistung eine schnellere und genauere Simulation erzielt werden. Würde dazu noch der erwähnte synchrone Simulations-Modus verwendet, könnte so eine bedeutend präzisere Simulation ermöglicht werden. Im Bereich der Regelungs-Algorithmen sind die Verbesserungs- und Erweiterungsmöglichkeiten bei weitem am größten. Es kann, sofern die Genauigkeit des Modells dem Anwender genügt, eine Vielzahl von zweidimensionalen Regelungsprinzipien und -algorithmen in MATLAB programmiert und getestet werden. Auch können Vergleiche von verschiedenen Regelungs- und Optimierungsprozessen in MATLAB programmiert und in der Simulationsumgebung VREP getestet werden. Besonders für den praktischen Einsatz des *Heron* kann mit der geschaffenen Simulationsumgebung vor einem Praxistest eine Simulation Aufschluss darüber geben, ob der geschaffene Programmcode sich wie vorgesehen verhalten wird.

Wie bereits in der Einleitung erwähnt, tendieren die moderne Schifffahrt und deren Forschungsziele zu einem vollständig autonomen Schiffsverkehr. Auch wenn das in dieser Arbeit präsentierte Modell nur eine sehr einfache Art von autonomer Schifffahrt darstellt, so zeigt sie doch auch auf, was bereits mit einfachen Mitteln möglich ist, denn eine Übertragung des programmierten Reglers auf ein Exemplar des *Heron* wäre durchaus denkbar.

Anhang A

Programmversionen

Für das in dieser Arbeit erstellte Modell wurden folgende Software-Versionen verwendet:

Programmversionen

- VREP: Version 3.5.0 (rev 4), 64bit
- MATLAB : Version R2013b, 64bit

Eine Zusammenstellung des verwendeten Programmcodes in VREP und MATLAB ist in entsprechenden Dateien geordnet unter der unten aufgeführten URL abrufbar. Auch ein Programm-Ordner für die Schnittstelle zwischen beiden Programmen sowie eine erläuternde Textdatei sind dort zu finden.

https://github.com/aaronwickers/heron_vrep

Literaturverzeichnis

1. Coppelia-Robotics (2018) VREP User Manual. Coppelia Robotics, 3rd edn
2. Fossen S Breivik (2003) Line-of-sight path following of underactuated marine craft. URL : <http://www.fossen.biz/home/papers/FossenBreivikSkjetneMCMC03.pdf>
3. Fossen T (1995) Guidance and Control of Ocean Vehicles. Wiley, New York, NY
4. Fossen T (2011) Handbook of Marine Craft Hydrodynamics and Motion Control. Wiley, DOI 10.1002/9781119994138
5. Meurer T (2018) Mathematical modeling. URL: <https://www.control.tf.uni-kiel.de/en/teaching/winter-term/mathematical-modeling-etit-619/>
6. Shabana AA (2005) Dynamics of Multibody Systems, 3rd edn. Cambridge University Press, DOI 10.1017/CBO9780511610523
7. Storr B (2018) Regelungstechnische methoden für den autonomen schiffsbetrieb. Master's thesis, Christian-Albrechts-Universität zu Kiel

Erklärung

Ich versichere, dass ich die Bachelorarbeit

Simulation und Regelung eines Wasseroberflächen-Fahrzeugs

selbständig und ohne unzulässige fremde Hilfe angefertigt habe und dass ich alle von anderen Autoren wörtlich übernommenen Stellen, wie auch die sich an die Gedankengänge anderer Autoren eng anlehnenden Ausführungen, meiner Arbeit besonders gekennzeichnet und die entsprechenden Quellen angegeben habe.

Mir ist bekannt, dass die unter Anleitung entstandene Bachelorarbeit, vorbehaltlich anders lautender Vereinbarungen, eine Gruppenleistung darstellt und in die Gesamtforschung der betreuenden Institution eingebunden ist. Daher darf keiner der Miturheber (z.B. Texturheber, gestaltender Projektmitarbeiter, mitwirkender Betreuer) ohne (schriftliches) Einverständnis aller Beteiligten, aufgrund ihrer Urheberrechte, auch Passagen der Arbeit weder kommerziell nutzen noch Dritten zugänglich machen. Insbesondere ist das Arbeitnehmererfindergesetz zu berücksichtigen, in dem eine Vorveröffentlichung patentrelevanter Inhalte verboten wird.

Kiel, 03.03.2019

Aaron Wickers