

How to write a protocol for the MSc laboratory course at the IST^{*}

Kimon Beyer, Yves Gaßmann, Justin Buhl

Abstract: In diesem ersten Protokoll handelt es sich um eine Zusammenfassung des Vorgehens der Gruppe AWe1 des Praktikums "Konzepte der Regelungstechnik" am Institut für Systemtheorie und Regelungstechnik der Universität Stuttgart. In diesem ersten Protokoll werden die Ergebnisse von L1 und H1 dargestellt, wobei sich der erste Teil um L1 handelt und der zweite Teil um H1. Während sich L1 mit der Betriebsnahme des Versuchsstandes beschäftigt, wird im anschließenden H1 Teil die Modellierung des 3DOF Helikopters durchgeführt, mit abschließender Respektive auf die in L2 zu erledigende Modellidentifizierung.

1. EINLEITUNG

In diesem Protokoll werden die Ergebnisse der ersten Laboraufgabe (L1) und der ersten Hausaufgabe (H1) des Praktikums "Konzepte der Regelungstechnik" am Institut für Systemtheorie und Regelungstechnik der Universität Stuttgart zusammengefasst. Ziel von L1 ist es, den Versuchsstand in Betrieb zu nehmen und die grundlegenden Funktionen zu verstehen. H1 konzentriert sich auf die Modellierung des 3DOF Helikopters, um eine Grundlage für die anschließende Modellidentifizierung in L2 zu schaffen. Beim zu regelnden Versuchsstand handelt es sich um einen 3DOF Helikopter, der entlang seiner Hauptachsen/Gelenkachsen (Travel angle, elevation angle, pitch angle) bewegt werden kann. Die Bewegung um die Achsen wird dabei durch zwei Elektromotoren ermöglicht. Ziel des Praktikums ist es eine vorgegebene Trajektorie innerhalb einer gewissen Zeit abzufliegen und dabei die jeweiligen Beschränkungen einzuhalten. Dabei wird im folgenden die Inbetriebnahme des Versuchsstandes beschrieben und die Modellierung des Helikopters durchgeführt. Dieses Modell wird im laufenden Praktikum genutzt um ein Regelentwurf durchzuführen und am Teststand zu implementieren.



Fig. 1. Versuchsstand: 3DOF-Helikopter. Besteht aus Haupt- und Nebenarm, sowie Gegengewicht und am Ende des Arms befindlichen Helikopter.

2. ERSTER LABORTAG (L1)

Um den Versuchsstand in Betrieb zu nehmen wird die "Quanser Quarc" Schnittstelle, eine Erweiterung für Matlab/Simulink, verwendet. Diese Schnittstelle ermöglicht es, die Hardware mit Matlab Simulink zu verbinden und die Motoren/Signale über die jeweilige Blöcke anzusteuern und auszulesen. Über einen "Write Analog" Block werden die zwei 12V Motoren die zur Verfügung stehen, sowie der Magnet angesteuert. Der Magnet wird benötigt um das Verfrachten einer Metallkugel als Teil der Praktikumsaufgabe zu ermöglichen. Eine Spannungsdifferenz an den Motoren ermöglicht nun die Steuerung des Pitch Winkels und somit auch des travel Angels. An diesem Block kann auch der "Game Controller" angeschlossen werden, mittels dem der Versuchsstand manuell angesteuert werden kann. Dafür muss jedoch eine möglichst intuitive Bedienung mit passender Skalierung der Eingabewerte implementiert werden. Dafür könnte an dieser Stelle noch ein PID-Controller implementiert werden. Der "Read Encoder" Block ermöglicht das Auslesen der Winkelinkremente der drei Achsen der verbauten Encoder, die relativ zur Ausgangsposition gemessen werden. Diese Winkelinkremente müssen noch umgerechnet und mit richtigem Vorzeichen versehen werden, um die nötigen Konventionen von Rechtshandsystemen und der Modellbildung zu gewährleisten. Diese Blöcke werden letztendlich in 3 Simulink Subsystems unterteilt, "Input Generation", "Data acquisition" und "Plant". Die Plant soll dabei durch ein Modell des Helikopters ersetzt werden können. Im Subsystem "Input generation" der Write Digital/Analog Block sowie der Game Controller zusammen gefasst. Im Subsystem "Data acquisition" werden die Encoderwerte ausgelesen und aufbereitet.

3. ERSTE HAUSAUFGABE (H1)

Für die Modellierung des 3DOF Helikopters wurde versucht eine möglichst einfache Modellannahme zu treffen um die Komplexität des Modells gering zu halten und mit einem möglichst überschaubaren Zeitaufwand eine möglichst gute Abbildung des Systemverhaltens zu extrahieren. Zunächst war nämlich geplant, den Versuchsstand in SolidWorks zu modellieren und mit anschließenden Matlab tools die Systemgleichungen zu extrahieren.

^{*} Institute for Systems Theory and Automatic Control, University of Stuttgart, Germany. <http://www.ist.uni-stuttgart.de>

Dies hat jedoch zu Problemen geführt, wodurch eine erfolgreiche Extraktion der Modellgleichungen leider ausblieb. Dennoch wurde das Modell genutzt um Modellparameter zu extrahieren. Darauf wird jedoch noch im folgenden eingegangen. Für die physikalische Modellbildung wurde nun zunächst Grundannahme getroffen, die einzelnen Achsen des Helikopters unabhängig voneinander zu betrachten. Die Dynamik der einzelnen Achsen ergibt sich dann durch das Momentengleichgewicht und

$$M_{\text{ges}} = M_{\text{grav}} + M_{\text{motor}} \quad (1)$$

$$M = I * \ddot{\Phi} \quad (2)$$

Dabei bezeichnet M_{ges} das gesamte Moment, M_{grav} das durch die Gravitationskraft verursachte Moment und M_{motor} das durch die Motoren erzeugte Moment. I ist das Trägheitsmoment der jeweiligen Achse und $\ddot{\Phi}$ die jeweilige Winkelbeschleunigung um die betrachtete Achse. Die Dynamik der drei Rotationen um die jeweilige Achse ergibt sich letztendlich durch zweifache Integration der Winkelbeschleunigung.

Als Grundannahme betrachten wir die einzelnen rotationen Einzeln. D.h wir betrachten für jede Rotationen einen gesamten Starrkörper. Das heist für den Pitch Winkel um die Pitch Achse, bewegt sich lediglich der Helikopter als Starrkörper. Für die Bewegung um die Elevation Achse, betrachten wir den Helikopter und einen gesamten Arm (d.h Hauptarm, Magnethalterung, Nebenarm und Gegengewicht wird alles als ein starrer Körper betrachtet), beides zusammen als Starrkörper mit lediglich einem Freiheitsgrad um die Elevation Achse. Für die Bewegung um die Travel Achse, betrachten wir nun den Helikopter, einen Arm und einen zusätzlichen vertikalen Arm, alle drei Teile als ein starrer Starrkörper mit einem Freiheitsgrad um die Travel Achse.

Des Weiteren wurden folgende Annahmen getroffen:

- Reibungen sowie Luftwiderstände werden vernachlässigt.
- Der Schwerpunkt des Helikopters selbst liegt im Rotationspunkt der Drehachse. Diese Symmetrie hat zur Folge, dass für die Pitch Achse kein Gravitationsmoment berücksichtigt werden müssen, da sich die Gewichtskräfte gegenseitig aufheben.

Zur Extrahierung der Massenträgheitsmomente wurde nun das Modell (Verweis!!!) aus SolidWorks genutzt. Dabei wurden folgende Massenträgheiten extrahiert sowie mit folgenden Längen als Hebelarme des Starrkörpers festgelegt:

Table 1. Modellparameter

Variable	Wert	Beschreibung
I_{α}	–	Trägheitsmoment um die Travel-Achse
I_{β}	–	Trägheitsmoment um die Elevation-Achse
I_{γ}	–	Trägheitsmoment um die Pitch-Achse.
l_{γ}	–	Hebelarm für das Pitch-Differenzialmoment, auch von der Rotationsachse bis zum Motor
l_{Heli}	–	Abstand Rotationsachse um Elevation Achse zur Rotationsachse der Pitch-Achse
$l_{\text{Massenmittelpunkt}}$	–	Abstand Rotationsachse um Elevation Achse zum Massenmittelpunkt des Armes und des Helikopters

Bei den jeweiligen Trägheiten

$$I_{\alpha}, I_{\beta}, I_{\gamma}$$

die aus dem SolidWorks Modell extrahiert worden sind, handelt es sich um die addierten Trägheitsmomente der, für die jeweilige Achsrotation relevante, angenommenen Starrkörper.

Es fehlt noch die Beschreibung von F_g

Letztendlich ergeben sich folgende Differentialgleichungen für die drei Achsen:

$$\ddot{\gamma} = \frac{(F_{\text{vorne}} - F_{\text{hinten}}) * l_{\gamma}}{I_{\gamma}} \quad (3)$$

$$\ddot{\beta} = \frac{(F_{\text{vorne}} + F_{\text{hinten}}) * l_{\text{Heli}} * \cos(\gamma) - F_g * l_{\text{MMP}} * \cos(\beta)}{I_{\beta}} \quad (4)$$

$$\ddot{\alpha} = \frac{-(F_{\text{vorne}} + F_{\text{hinten}}) * l_{\text{Heli}} * \sin(\gamma)}{I_{\alpha}} \quad (5)$$

4. LATEX

The protocols have to be typeset with \LaTeX . A good German introduction to \LaTeX can be found in [Jürgens and Feuerstack, 2011]. An alternative in English is [Oetiker et al., 2011]. A short guide concerning the mathematical part is given by [Downes, 2002].

5. SUBMISSION

Two files have to be submitted to the Praktikumskoordinator until 11:59pm four days before the laboratory by uploading them into Ilias:

- A pdf-file of the protocol and
- a zip-file containing all the relevant Matlab-files.

Every student has to be *responsible* for the protocol and the zip-files *once* during the course.

The protocol for every laboratory day has to contain

- a result protocol of the last laboratory, if applicable,
- a protocol of the preparation and
- a working plan for the next laboratory.

A template for the working plan can be found in Table 2. Of the last laboratory day, a result protocol has to be written which is due one week after the laboratory.

To make your approach or results repeatable, comment all your Matlab-scripts and Simulink-files appropriately - preferably while working with them. Decide, which files and maybe also measurement data is necessary to repeat your results. Then, put all necessary files into one zip-file and upload it into Ilias. In the protocol you should reference the files where the information is only needed for repeatability. Furthermore, you should refer to the files while explaining your procedure. Thus, you don't have to include all information into the protocol.

You can write either in German or English. If you write in German, it is a good idea to include the packages `ngerman`, `inputenc`, and `fontenc` in the preamble of your tex-file.

Table 2. A Working Plan Template.

Time	Duration	Goal	Task	Preparation
14:00	30 min.	Test File to actuate motor	Create Quarc Simulink File	Read Quarc documentation

6. STRUCTURE

In the title include your group description and the task the protocol is about. The group description follows the syntax $IDDN-T$ with $I \in \{A, B\}$ being the cycle number, $DD \in \{Mo, Tu, We, Th\}$ the day, $N \in \{1, 2, 3, 4\}$ the group number, and $T \in \{1, 2, 3, 4, 5\}$ the task number. For example AMo1-1. Include all group members as authors. Underline the corresponding author. In the abstract, you should summarize the result within up to three sentences.

The structure on the section level has to be decided depending on the tasks. For the main sections, the structure can be similar to the problem-solving process:

- *Work task* explaining the task and the goal.
- *Solution approach* describing the chosen solution to allow the reader to repeat and assess the solution.
- *Result* containing the observations and justified conclusions.
- *Critical comments* evaluating the approach including a comparison with the working plan. Furthermore the pros and cons should be discussed.

7. FIGURES AND GRAPHICS

All figures should be created as done in the Matlab-script “plotFigureTest.m” which can be found in Ilias. By following the procedure, the axis labels and ticks are in an appropriate size, the same L^AT_EX-font is used, and the line thickness is adjusted resulting in a formally excellent figure. If you have plots in Simulink, export the data and create a Matlab figure. To label the axis, use words and not symbols. Write units in parentheses. E.g., write “Control input u (V)”. An easy and nice-looking way for the legend is to include it into the caption of the figure. See the tex-code for more details.

Graphics for the protocol can be made for example with the Open Source vector graphics editor Inkscape. Details can be found on <http://inkscape.org/>.

8. FINAL PROTOCOL

At the end of the course a protocol over the whole laboratory has to be written. This final protocol may have up to 8 pages. The final protocol may be based on your other protocols and you can reuse parts of the other protocols. But the final protocol has to be self-contained and needs to have a comprehensible line of thought.

The purpose of the final protocol is that the reader gets an idea of your whole solution. Therefore, it should contain the results of your work. To understand the results, you should also include your approaches on a more abstract level. For you, the final protocol is a chance to reflect the laboratory as a whole.

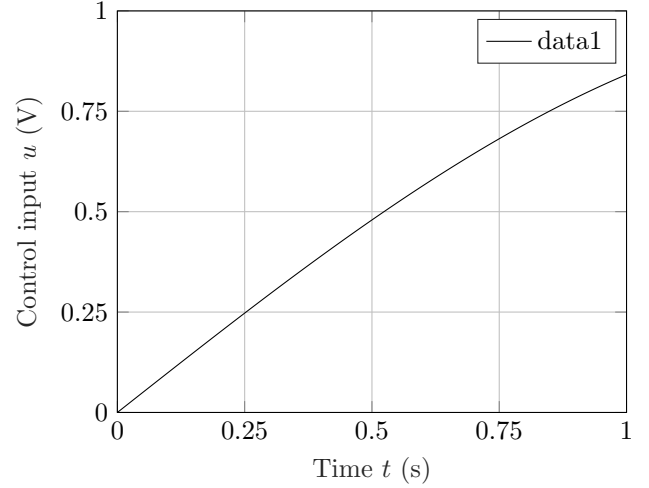


Fig. 2. Exemplary Matlab figure. The exemplary signal is plotted in (—).

REFERENCES

- Jürgens, M. and Feuerstack, T. (2011). L^AT_EX-eine Einführung und ein bißchen mehr.... Fernuniversität in Hagen, A/026/0911. Available online from http://www.fernuni-hagen.de/imperia/md/content/zmi_2010/a026_latex_einf.pdf.
- Oetiker, T., Partl, H., Hyne, I., and Schlegl, E. (2011). The not so Short Introduction to L^AT_EX 2_ε. Version 5.01, April 06, 2011. Available online from <http://tobi.oetiker.ch/lshort/lshort.pdf>.
- Downes, M. (2002). Short Math Guide for L^AT_EX. American Mathematical Society. Available online from <ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf>.
- IEEE Control Systems Magazine (2004). Writing Guidelines for IEEE Control Systems Magazine. *IEEE Control Systems Magazine*, 24(1), 89–90.