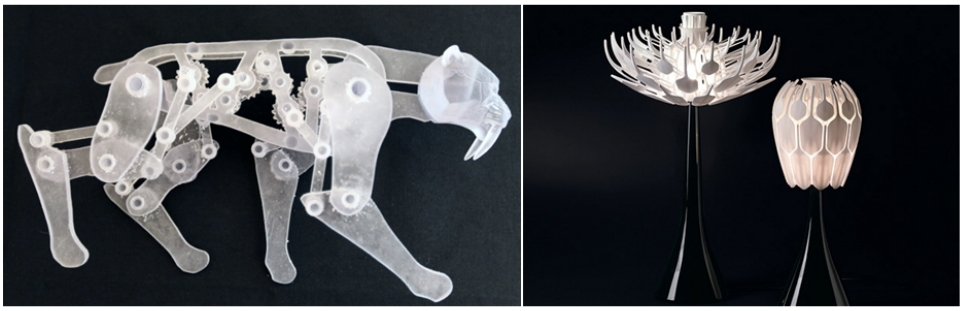
Thesis Title



Jonas Stolle

Bachelor Thesis
May 2021

Supervisors:
Moritz Geilinger
Prof. Dr. Stelian Coros

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CRL

Abstract

This thesis addresses the development of a novel sample thesis. We analyze the requirements of a general template, as it can be used with the LaTeX text processing system. (And so on...) The abstract should not exceed half a page in size!

Contents

List of Figures

List of Tables

1

Introduction

also state here that we are dealing with mechanical systems in particular

2

Related Work

Sample references are [**?**] and [**?**].

2.1. Appearance Modeling

2.1.1. Taxonomy

at the end of the one paper talk about how the goal is to extend on this by applying it to different and more complex systems.

3

Your Central Work

3.1. Fundamentals and Problem Formulation

In this section describes the relvant physical definitions from which the robustness measure is derived and which will be referred to in the later code implementation. all of which are important for understanding and some of which are directly used for the implementation.

Examples will be given n terms of laikag quadruped robot as most of the testing of the framework was done with that.

3.1.1. Dynamical Systems

Dynamical systems are distinguished by an evolution of their state $\mathbf{x}(t)$ through time. This evolution can be fully described by a set of ordinary differential equations of the form $\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t), t)$, where $\mathbf{F}$ is some nonlinear function. This simplifies to $\dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t))$ under the assumption that the dynamical system is autonomous, i.e. is not explicitly dependent on time. When solving for the explicit solution $\mathbf{x}(t)$, an initial condition $\mathbf{x_0} = \mathbf{x}(t_0)$ is required, which is a system state at an initial time. For simplicity and without loss of generality for autonomous systems, we set $t_0 = 0$. With this the Initial Value Problem (IVP) can be formulated:

$$find\ \mathbf{x}(t) \tag{3.1}$$

$$s.t.\ \dot{\mathbf{x}}(t) = \mathbf{F}(\mathbf{x}(t)) \tag{3.2}$$

$$and\ \mathbf{x}(0) = \mathbf{x}_0 \tag{3.3}$$

We denote the solution to the IVP as $\mathbf{x}(t, \mathbf{x}_0)$. It represents trajectory of the system state through time given an initial condition. The space in which this trajectory lies is spanned by all possible

states $\mathbf{x}$ and termed the $state\ space$. Note that any future state of the trajectoy $\mathbf{x}(\tau, \mathbf{x}_0)$ at time τ can be interpreted as an initial condition of the IVP itself. It turns out that the new solution coincides with the initial one, i.e. $\mathbf{x}(t, \mathbf{x}_0) = \mathbf{x}(t, \mathbf{x}(\tau, \mathbf{x}_0))$, which illustrates that any state $\mathbf{x}(t)$ of a trajectory $\mathbf{x}(t, \mathbf{x}_0)$ is suffitient to represent the trajectory as a whole.

Mechanical systems (on which we will focus on form here on out) tend to be described in terms of so called generalized coordinates:

$$\mathbf{q}(t) = \begin{pmatrix} q_1(t) & q_2(t) & \dots & q_n(t) \end{pmatrix}^\mathsf{T}. \tag{3.4}$$

They are the minimal set of coordinates needed to fully describe the position and orientation of all of the systems elements. Their dimension n cooincides with the number of degrees of freedom of the system. The corresponding differential equations are of second order, depending on $\ddot{\mathbf{q}}(t)$ in addition to $\dot{\mathbf{q}}(t)$ and $\mathbf{q}(t)$. Simply put, this is due to their derivation by Newton's second method, where forces acting on the system are related to the second time derivative via $F = ma$. Through an order reduction (Appendix) these differential equations can be cast into the reviously mentioned general form 3.1, where

$$\mathbf{x}(t) = \begin{pmatrix} \mathbf{q}(t) & \dot{\mathbf{q}}(t) \end{pmatrix}^\mathsf{T}.$$

This implies that in order to solve the IVP, initial conditions $\mathbf{q}_0$ and $\dot{\mathbf{q}}_0$ are required. We can also state that for a system with n degrees of freedom, $\mathbf{x}(t) \in \mathbb{R}^{2n}$. The particular state space spanned by generalized coordinates and velocities is termed the $phase\ space$. Within it, any states are single points and state trajectories are smooth curves. The phase space is used for generalizable qualitative analysis of the behaviour of nonlinear dynamical systems and plays a pivotal role in the fromulation of the robustness measure.

When discussing high level concepts we will refer to the system state $\mathbf{x}(t)$ for simplicity, while in the code implementation the generalized coordinates $\mathbf{q}(t)$ and its derivatives will be more relevant. Keep in mind that both are equivalent.

3.1.2. Attractors and Convergence

In nonlinear dynamical systems, there may exist sets of states in the phase space which show an attracting behaviour. By "attracting" we mean that once a trajectory reaches an element of such a set, all of its future states will also be part of that set. Define an attractor as a set of states:

$$\mathbf{A} \subset phase\ space, \tag{3.5}$$

$$s.t.\ if\ \mathbf{x}(t_0) = \mathbf{x}_0\ \in \mathbf{A}, \tag{3.6}$$

$$\mathbf{x}(t, \mathbf{x}_0) \in \mathbf{A} \quad \forall\, t > t_0. \tag{3.7}$$

Attractors can be divided into two fundamental variants. If the attracting set consists of only one state, it is called a fixed point. Fixed points are associated with the condition

$$\dot{\mathbf{x}}(t) = \mathbf{0}\ \ \forall\, t \in \mathbb{R}, \tag{3.8}$$

meaning that the state of the fixed point is unchanging and the related trajectory is reduced to a point in the phase space. Wheather a state $\mathbf{x}(t)$ is a fixed point can be easily determined by

checking $\mathbf{F}(\mathbf{x}(t)) = \mathbf{0}$. A classical example of a fixed point is the stable bottom position of a pendulum, where if given zero initial velocity, it won't leave the stable position. This is not the case for any other position as gravity will act on the mass (except for the inverted position, however this is practically not realizable). In the general case with A containing of more than one state, (reference above eq) is not true. Rather the trajectory is moving through the sets of A, visiting every element at some point in time and returning to it at future times in a periodic fashion. These types of attractors are called limit cycles. Finding them is generally a hard problem, but for simpler cases one can check:

$$If\ for\ \mathbf{x}(t, \mathbf{x}_0),\ t > 0 \quad \exists \quad \mathbf{x}(\tau) = \mathbf{x}(\tau + h),\ \tau > 0,\ h > 0, \tag{3.9}$$

$$\{\ \mathbf{x}(t) \mid t \in [\tau, \tau + h)\ \},\ is\ a\ \mathbf{limit\ cycle}. \tag{3.10}$$

An example for this case are the compliant linkages described in (ref strandbeest compliant version), where the end effector follows a cyclic trajectory, i.e. set of states if undisturbed. In the code implementation section, we provide methods for dealing with both types of attractors and the problem of applying the continuous analysis in a discrete setting.

For any initial condition not part of the attractor, the related trajectory may land and stay on the attractor after some time $t > t_0$. We define this occurance:

$$Given\ an\ attractor\ \mathbf{A},\ for\ any\ \mathbf{x}_0 \notin \mathbf{A} \tag{3.11}$$

$$if\ \lim_{t\to\infty} \mathbf{x}(t, \mathbf{x}_0) \in \mathbf{A}\ \Rightarrow\ \mathbf{x}(t, \mathbf{x}_0)\ \mathbf{converges}\ to\ \mathbf{A} \tag{3.12}$$

Should the definition above not hold, we denote the trajectoy as diverging (as in cell mapping methods).

Note that there may exist any number of attractors in the phase space of a system (reference paper with multitude of attractors). Convergence is always defined with respect to a particular attractor $\mathbf{A}$, which needs to be specified. Therefore if the trajectory converges to a different attractor, it is still defined as diverging from the attractor of interest.

The set of all states for which the related trajectories converge to the attractor of interest is called the $Basin\ of\ Attraction$ (BoA) and is defined as:

$$\{\ \mathbf{x}_0 \in \mathbb{R}^{2n} \mid \lim_{t\to\infty} \mathbf{x}(t, \mathbf{x}_0) \in \mathbf{A}\ \} \tag{3.13}$$

where $\mathbf{A}$ is an attractor as defined in 3.8.

3.1.3. Robustness Measure

INTRO

The robustness measure proposed in REF and implemented in the following is derived from basic concepts of nonlinear dynamic outlined in the previous section. In the actual implementation and testing compromises had to be made (rephrase) in order to improve feasibility, because of which we propose some slight reframing of some definitions to accomodate for those variations.

REF METHOD

Given a target behaviour of a system in form of an attractor, convergence of a state trajectory can be interpreted as a successfull recovery from a disturbance. Looking at a set of disturbances, a larger portion of recoveries means a larger robustness within that set. This can be taken as a scalar measure of robustness within that set. If one could define a set of all possible disturbances, one could theoretically compute general robustness. At this point already one might notice that the set of disturbances may lie in a continuous space, making

Choosing instantaneous external forces as disturbances, their effect on the system can be captured by a change on the systems state. Forces are directly related to the generalized accelerations $\ddot{\mathbf{q}}$ following Newton's second law, which in turn changes $\mathbf{q}$ and $\dot{\mathbf{q}}$ via the underlying differential equations. This implies that the set of disturbances in this case coincides with the elements of the phase space. This is precisely the approch chosen for sampling of disturbances and measurement of robustness in (reference paper). A nice implications of this is that the set of disturbances from which the system recovers from coincides with the basin of attraction (defined previously). There exists literature analyzing . Also the scaling issue between q and qdot, maybe just reference it and go into detail later.

However it also assumes that the system experiences a disturbance at exacltly one point and not after. This is of course quite limiting. Effects like resonance for example cannot be captured HOOWEVER

ADAPTATIONS / REDEFINITION

IMPLICATIONS OF ADAPTATIONS

The good thing of the conservative measure of the minimal Radius is has next to no requirements onto the shape of the attracting set of disturbances and can easily be implemented in different spaces.

TRANSITION TO NEXT CHAPT

we ultimately want to optimize over robustness measure, so we need to look into that.

... We will have a single or a set of states of our system that we have as a goal (= attractor). In the context of a quadruped this might be an upright standing position (fixed point) or a periodic walking gait. If the system is perturbed, it's state will inevitably be changed such that it is most likely not part of the attractor anymore. Finding the trajectory shows how the system will further evolve. Now convergence means that under the disturbance applied, in the long term, we will still return to and stay on the attroctor, or the system recovered, it is successful in compensating the disturbance. Divergence means something else happened, which we will just interpret as failure. So if the trajectory related to an initial condition caused by a particular disturbance converges to the attractor, the system is in a sense robust to that disturbance.

If one wants to maximize robustness, really one wants to change the system or rather its parameters in such a manner that the set of disturbances which result in convergence of the state trajectory is maximized. This is the basic formulation.

As detailed in REF and REF, evaluating the exact size of the set of convergiing disturbed systems, is often hard and sometimes misleading, which is why the more conservative measure of the minimal radius found with the help of random sampling is adopted (phrasing). This is in addition to the fact, that with the numerical solver at hand (ref section below), solving long

trajectories and evaluating their convergence via a given attractor is much simpler to implement than the cell mapping methods detailed in previous work.

This basic idea was proposed and implemented in REF, with a more focused choice of disturbance space. Disturbances themselves were disregarded and only their effects in the phase space taken into consideration. So here the goal was to maximize the set of converging inital conditions independent of their cause. However as the sampled inital conditions are just effects of disturbances, in the test we sampled and applied general disturbances, which can be thought of as subsets of the full phase space. Especially as we work with high dof systems, sampling disturbances in the full state space becomes infeasible very quickly. A generalization to disturbances as opposed to initial conditions allows us to explore more easily understandable examples where physical intuition can be applied.

While we still determine convergence by checking the state trajectory wrt the attractor in the phase space, disturbances will be sampled in the DS, meaning the minimal radius also lies in the disturbance space. Nice thing is that any disturbance one can come up with can be tested, but generalizbility to other disturbances (for which one doesn't evaluate the robustness) ceases to exist (rephrase)

The nonlinear dynamics view is uesful in detecting attractors and evaluating, but sometimes when the needed information can be acquired more easily, we always chose that route.

We sample in the bounded DS, evaluate convergence of disturbed system via convergence to the specified attractor in the phase space and

In other words, any combination of disturbances which lies on the surface of the hypersphere with the minimal radius in the DS will result in a trajectory that converges to the attractor in the phase space.

Taking sampling initial conditions form the phase space as disturbances gives a nice physical interpretation and in "REF" this was denoted as general robustness (or was it?), however as a counterexample, periodic or constant disturbances are cleary not covered by this. This is why we worked with general disturbance spaces that can be chosen to work for the particular application at hand. Note that the trajectory will still lie in the phase space and convergence evaluation

Describe robustness measure from paper but for the actual definition use the general disturbance space.

3.1.4. Parameter space and Disturbance space

The choice of parameter spaces (have I mentioned them before?) and disturbance spaces I feel like this should be part of the robustness measure

separate from "dynamics", focus on

dimension disturbance space equals d

We define the $Disturbance\ Space$ (DS) as the d dimensional space in from which we sample the disturbances. As in REF, The disturbance space is defined as the set containing all combi-

nations of disturbances, against which the robustness of the system is evaluated. The choice of disturbances in "REFERENCE" are the initial conditions in the phase space (reference to the section in related work or copy that here) as they can be interpreted as the direct result of external forces. In general however, the disturbances may be chosen arbitrarily and the resulting robustness measure will quantify robustness against these. Examples of this were initial tilting of a quadruped over pitch and roll axes and oscillations with the disturbance space spanning the amplitudes and frequencies. Tuple combinations are favoured in testing because of their ease of visualization and mild computational power requirements.

Issues with the disturbance space. Scaling and boundaries will fundamentally effect the robustness measure. The choice of the phase space as the DS in REF remedied this as there is a direct relation between the coordinates and their derivatives via physical units and the test cases were ones where all the coordinates had the same units and were generally similar (multi-pendulum => only angles and angular velocities). Changing the units will already yield different results and combining various different coordinates might very quickly compicate things further.

"REFERENCE" chose the DS to coincide with the phase space, as elements of it can be directly interpreted as the result of external forces on the system. This choice is not always feasible nor necessary, as will be shown in the following sections.

It may coincide with the phase space as implemented in "REFERENCE", in which case disturbances can be interpreted as the effects of external forces acting on the system. This moves the system onto a different trajectory , however it will be shown in the following sections, that this is not always feasible nor necesseary. Especially in systems with a large number of degrees of freedom n, the resulting disturbance space is 2n dimensional.

case DS = phase space: d

dimension parameter space = p The two main variable spaces of relevance move are the parameter space (PS), where each dimension represents the value of a chosen system parameter. The goal of finding a robustness measure can be though of as $RM : \mathbb{R}^p \to \mathbb{R}$, i.e. a mapping from the p-dimensional parameter space to a scalar value. (really it also depends on the disturbance space) Finding the best system parameters with respect to the resulting robustness can be formulated as an optimization problem ...

state space v phase space

3.2. Code Implementation

This section details the implementations of the robustness measure and particular challenges that were encountered and overcome. (rephrase)

3.2.1. Framework Overview

differential equations are implicitly represented as simulation objects in dde. Their parameters can be set. The state x as well. Use the solver to compute the trajectory under a disturbance for

a given amount of timesteps. check if trajectory converges or diverges. Do this repeatedly with initial conditions sample talk about how knowing the exact shape of the basin of attraction is not necessary for optimization of the robustness, with this and because of the additional work needed to implement the cell mapping algorithms, the method with full trajectories was chosen. For this we need a solver. boom. Great segue!

specify laikago as an example. Or rather briefly describe what it is and use it to illustrate issue that might arise,

here we want a nice block diagram.

Also here we want

3.2.2. Solver

Finding an explicit analytical solution to the IVP is possible for simple cases, but very hard if not impossible for more complex systems. The solution to the IVP can be found using numerical solvers. This is not an explicit solution but the trajectory must be iteratively computed by integrating the differential equations over small time steps.

Here we used dde ... write some stuff about dde.

Describe what dde puts out DDE provides q, qdot and qddot at every timestep when computing the trajectories.

We choose the method from REF over cell mapping because of it's simplicity in implementation with the given solver.

One particularity of the dde framework, x and z are the horizontal plane, and y is the vertical axis, which the reader nees to keep in mind.

Starting with a sytem of ODE's and as corresponding set of initial states, the first step towards computing robustness measure is finding the resulting trajectories. This is achieved by numerically integrating the differential equations over small timesteps until either convergence is detected or the t_max is reached. With the chosen solver it is imperative to keep this timestep constant when comparing different robustness measures. The time step has a direct influence on the solvers accuracy and by that onto the system dynamics. It is advised to find a value that is a sufficient compromise between accuracy and cumputational time and keeping this fixed from there on.

trajectories here are not continuous but sets of states for discrete timepoints $[t_1, t_2, \ldots, t_n]$.

3.2.3. Detecting Attractors

In order to evaluate wheather a trajectory converges

Automatic identification of fixed points General idea with similarity of states to attractor state

Fixed points this is rather easy fixed points are defined states where $\dot{\mathbf{x}}(t) = \mathbf{0} \quad \forall\, t \;>\; t_0$, i.e. $\mathbf{q}(t) = \mathbf{q}_{fp}, \dot{\mathbf{q}}(t) = \ddot{\mathbf{q}}(t) \forall t > t_0$

Issue that this is one specific state that does not accomodate for any deviation. With laikago standing up, we might accept translations of the robot in the xy-plane or rotations about the

Periodic Orbits not as trivial. need assumptions assume that period of attractor is the same as forcing frequency. Think controller forcing a gait given period, then the leg will also move at that period. This is not generally true and needs to be verified. Poincare => issues with high computational effort and issues with systems that exhibit oscillations with a multitude of periods.

Issue with wrong attractor

Simplification

sometimes we can loosen the requirements for convergence. For the laikago experiments, where the goal is just for the quadruped to not tip over,

3.2.4. "Evaluating" Convergence

Cleary it is impractical to let t go to infitiy, which is why some t_n, max needs to be defined at which the simulation stops.

Any way of checking and guaranteeing for divergence is actually really useful, as when continually apllied at every timestep, the simulation can be stopped if divergence is detected and computational time be saved. Example Laikgao, if we want it to stand upright and we detect it tipping over, that run has clearly failed and can be stopped early. quite similar to

3.2.5. minRad algorithm

maybe not go into too much detail on how it works (look at ref) No DO go into detail. This is the part where we actually want to explain how we find the minimal radius. Because of the random sampling of disturbances, the convergence of the minRad algorithm and therefore the robustness value is inherently stochastic and might be dominated by noise. The extent of theis noise can be reduced by larger amounts of samples. An compromize between noise and computational time must be found. Number of iterations

tuning

visualizations plot resultion as a function of iterations. NO, write down the formula $resolution = (1/2)^n$ with n being the number of iterations.

3.2.6. boundarys PS DS

elephant in the room changing even the unit changes the scaling and therefore the robustness value.

explain and give examples on how boundaries were chosen.

was noted in REF that it can be an ellipse as well, but they didn't know how to choose it's parameters. We do this by analysis and finding reasonable boundaries in the DS.

order of complexity depending on PS and DS

3.2.7. Multithreading

computational time can be reduced by pseudocode

3.2.8. Application to specific systems

need to decide on PS and DS plus boundaries. Need to find or define attractor.

analysis needed to find parameter space and disturbance space boundaries and good initial guesses. analysis of high dof motion tricky (bad 3d image), rather plot every coordinate over time (image).

applyParameters

simAndEval

3.2.9. Optimization and Complexity reduction

find out how the order of computational effort when making discretization smaller vs how much one bisection algorithm iteration can do. cmaes given robustness value, any optimizer that does not depend on derivatives can be used. Of course one can also just explore the entire parameter space, howeve that is quite costly. Useful for debugging though (rough estimate if optimizer converges).

complexity reductions should probably be noted where they are implemented (i.e. not in this section)

3.3. Tests

Introduction to laikago high dof rather long computational time. how it's implemented (no walking yet) laikago, that it doesn't do anything but try to keep its limbs in the predefined state. optimization of robustness examples

Laikago Droptest Laikago Swingtest note that here we kind of broke the mathematical framework as technically the underlying diff eq were changed. But we just ignored this, NO effects of this need to be investigated further.

Maybe we could do one high precision, high resolution DS swingtest to see if we can spot resonance

This is kind of a twist on the concept as we are starting at a fixed point and continually applying disturbances to see for what disturbances the trajectory conveges, which in this case is expressed by the system staying at the initial state.

4

Conclusion and Outlook

physical explanation why there can't be that much optimization for swing and droptest

Implication that the system must be dynamical in nature (i.e. it must evolve). So a rudimentary control strategy must be implemented or outside forced must be applied. Don't quite know where to put this.

further explore effects of combining disturbences of differnent sorts and the effects of the choice of bounds.

applying to systems of high dof with full phase space. Is it even possible? How much can the code be optimized?

apply to physical dimensions of systems (intuitively more drastic effects)

using the phase space as the disturbance space but heavily restricting it (actuator limits, improbable constellations etc)

Optimizing with limit cycles as attractors.

solver (or rather finding the trajectories in general) will always be the largest bottleneck and finding methods to reduce number of trajectories to be evaluated or the computational time per trajectory would be very benefitial

Seems unlikely that this will ever be plug and play. Lots of tuning, lots of fiddling around. But there could definitely be applications, especially in novel systems where rigorous analysis is lacking.

High complexity is still an issue.

A

Information For The Few (Appendix)

dot = f(q,qdot) qddot = g(q,qdot)

order reduction:

x = [q,qdot] => xdot = [qdot, qddot] = F(x) = [f(x), g(x)]

q,qdot,qddot

A.1. Foo Bar Baz

A.2. Barontes

A.3. A Long Table with Booktabs

Table A.1.: *A sample list of words.*

ID	Word	Word Length	WD	ETL	PTL	WDplus
1	Eis	3	4	0.42	1.83	0.19
2	Mai	3	5	0.49	1.92	0.19
3	Art	3	5	0.27	1.67	0.14
4	Uhr	3	5	0.57	1.87	0.36

continued on next page

Table A.1.: *(Continued)*

ID	Word	Word Length	WD	ETL	PTL	WDplus
5	Rat	3	5	0.36	1.71	0.14
6	weit	4	6	0.21	1.65	0.25
7	eins	4	6	0.38	1.79	0.26
8	Wort	4	6	0.30	1.62	0.20
9	Wolf	4	6	0.18	1.54	0.19
10	Wald	4	6	0.31	1.63	0.19
11	Amt	3	6	0.30	1.67	0.14
12	Wahl	4	7	0.36	1.77	0.42
13	Volk	4	7	0.45	1.81	0.20
14	Ziel	4	7	0.48	1.78	0.42
15	vier	4	7	0.38	1.81	0.42
16	Kreis	5	7	0.26	1.62	0.33
17	Preis	5	7	0.28	1.51	0.33
18	Re-de	4	7	0.22	1.56	0.33
19	Saal	4	7	0.75	2.10	0.43
20	voll	4	7	0.48	1.82	0.24
21	weiss	5	7	0.21	1.59	0.36
22	-ger	5	7	1.16	2.69	0.59
23	bald	4	7	0.18	1.56	0.19
24	hier	4	7	0.40	1.70	0.43
25	neun	4	7	0.17	1.52	0.26
26	sehr	4	7	0.36	1.85	0.43
27	Jahr	4	7	0.50	1.82	0.43
28	Gold	4	7	0.04	1.35	0.20
29	Ter	5	8	0.15	1.39	0.59
30	Tei-le	5	8	0.30	1.71	0.46
31	Na-tur	5	8	0.18	1.59	0.41
32	Feu-er	5	8	0.30	1.71	0.45
33	Rol-le	5	8	0.15	1.46	0.45
34	Rock	4	8	0.29	1.68	0.25
35	Spass	5	8	0.28	1.64	0.32
36	Gte	5	8	0.49	1.75	0.66
37	En-de	4	8	0.36	1.72	0.33
38	Kunst	5	8	0.26	1.59	0.35
39	Li-nie	5	8	0.45	1.88	0.63
40	Bme	5	8	0.48	1.92	0.45
41	Bh-ne	5	9	0.94	2.48	0.62
42	Bahn	4	9	0.21	1.62	0.42
43	Br-ger	6	9	0.38	1.70	0.65
44	Druck	5	9	0.60	2.03	0.31
45	zehn	4	9	0.41	1.84	0.42

continued on next page

Table A.1.: *(Continued)*

ID	Word	Word Length	WD	ETL	PTL	WDplus
46	Va-ter	5	9	0.36	1.78	0.40
47	Angst	5	9	0.29	1.56	0.35
48	lei-der	6	9	0.13	1.47	0.52
49	hfig	6	9	0.82	2.31	0.52
50	le-ben	5	9	0.38	1.85	0.40
51	aus-ser	6	9	1.20	2.26	0.57
52	be-vor	5	9	1.28	2.75	0.39
53	Kai-ser	6	9	0.92	2.37	0.53
54	Markt	5	9	0.23	1.58	0.28
55	Os-ten	5	9	0.21	1.54	0.48
56	Krieg	5	9	0.33	1.67	0.50
57	Mann	4	9	0.31	1.47	0.25
58	Hal-le	5	9	0.24	1.65	0.45
59	heu-te	5	9	0.44	1.87	0.46
60	in-nen	5	10	0.36	1.80	0.45
61	Na-men	5	10	0.28	1.72	0.41
62	jetzt	5	10	0.70	2.07	0.32
63	kei-ner	6	10	0.28	1.62	0.53
64	Schu-le	6	10	1.02	2.12	0.48
65	Ar-beit	6	10	0.34	1.70	0.52
66	An-teil	6	10	0.27	1.63	0.53
67	di-rekt	6	10	0.67	2.04	0.47
68	vor-her	6	10	0.78	2.25	0.47
69	wol-len	6	10	0.44	1.85	0.51
70	Kampf	5	10	0.70	1.96	0.27
71	dern	6	10	1.18	2.62	0.65
72	lau-fen	6	10	0.21	1.64	0.52
73	Eu-ro-pa	6	10	0.23	1.53	0.66
74	statt	5	10	1.61	2.86	0.39
75	Wes-ten	6	10	0.29	1.60	0.54