Håkon Eidsvik

Dynamic Simulation of Power Systems Based on a Second Order Predictor-Corrector Scheme

Master's thesis in Energy and environment
Supervisor: Olav Bjarte Fosso
June 2019

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electric Power Engineering

NTNU
Norwegian University of
Science and Technology

Håkon Eidsvik

Dynamic Simulation of Power Systems Based on a Second Order Predictor-Corrector Scheme

Master's thesis in Energy and environment
Supervisor: Olav Bjarte Fosso
June 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electric Power Engineering

Kunnskap for en bedre verden

Abstract

In this thesis, the second order predictor-corrector method Gear's method has been implemented in the Python programming language. This is an extension of the implementation in the specialization project. Gear's method is a numerical integration method used to solve systems of differential-algebraic equations (DAE systems). The method utilizes self adaptive step lengths, where the step length depends on estimated tracking error in each step. This enables the method to simulate state-space power system models with a wide range of time constants, implying that the system model describes both rapid and slow phenomena. Every step and every calculation involved in Gear's method is extensively described in the appendix.

A literature study of numerical integration methods, their stability, applications and motivation has been carried out. This resulted in a theory section that serves as a basis for understanding the application of numerical integration methods in power systems, the need for stable methods and the motivation for utilizing variable step length.

A strategy for adapting the step length is applied to two system models. The first system consists of a synchronous generator connected to an infinite bus via two parallel transmission lines, in which one of them is subject to a short-circuit. The second system consists of a voltage regulator, amplifier, generator and a measuring device. The step length strategy is found to improve performance of the algorithm, and it is able to identify steady state quickly and increase the step length accordingly. This harmonizes well with findings made in literature and the specialization project.

However, problems arise when the two systems are subject to disturbances. At the time the disturbance is introduced, the step length drops. It takes some time to increase again, and several step length adaptions are required. Because this is computationally expensive, a strategy for re-initializing the state variables and their derivatives after a disturbance is applied. Three performance metrics related to step length adaption are identified for the purpose of having a basis of comparison between applying the re-initialization strategy and letting the algorithm handle it without added functionality.

The regulator/amplifier system described above is subjected to saturation effects, and the generator/infinite bus system is subjected to the aforementioned short circuit fault. For the short circuit case, performance is improved by approximately 5 % when re-initializing the system with the strategy w.r.t. the performance metrics. For the saturation case, no performance increase is recorded; the result is identical to letting the algorithm handle the disturbance on its own.

All simulation results are presented graphically, and tables summarizing the most interesting aspects are included. Some inconsistencies in the short circuit simulation related to the step length are observed, discussed and further assessed in the appendix.

Lastly, further work is recommended. This includes testing the step length adaption strategy and the re-initialization strategies on systems of higher complexity, and further investigating the inconsistent observations made in the short circuit case.

Sammendrag

I denne masteroppgaven har den andreordens prediksjons-korreksjonsmetoden Gears metode blitt implementert i programmeringsspråket Python. Dette er en utvidelse av implementasjonen i spesialiseringsprosjektet. Gears metode er en numerisk integrasjonsmetode som brukes for å løse systemer av differensial-algebraiske ligninger (DAL-systemer). Metoden benytter selvregulerende steglengder, hvor steglengden bestemmes basert på estimert feil i hvert steg. Dette gjør metoden i stand til å simulere tilstandsrommodeller av kraftsystemer som består av et bredt spekter av tidskonstanter, som impliserer at systememodellen beskriver både raske og trege fenomener. Hvert steg og hver kalkulasjon i Gears metode er utfyllende beskreved i appendikset.

Et litteraturstudie av numeriske integrasjonsmetoder, deres stabilitet, bruk og motivasjon har blitt utført. Dette resulterte i en teoriseksjon som fungerer som en base for å forstå numeriske integrasjonsmetoders applikasjon i kraftsystemer, behovet for stabile metoder og motivasjonen for å bruke variabel steglengde.

En strategi for justering av steglengden er brukt på to systemmodeller. Det første systemer består av en synkrongenerator koblet til et stivt nett via to parallelle transmisjonslinjer, hvor den ene utsettes for en kortslutning. Det andre systemet består av en spenningsregulator, forsterker, generator og en måleanordning. Steglengdestrategien viser seg å forbedre algoritmens ytelse, og den er i stand til å detektere stabil tilstand raskt og øke steglengden i henhold til det. Dette harmoniserer med funn gjort i litteratur og spesialiseringsprosjektet.

Problemer oppstår imidlertid når de to systemet utsettes for forstyrrelser. I tidspunktet forstyrrelsen introduseres synker steglengden. Den bruker litt tid på å øke igjen, og flere justeringer av steglengden er nødvendig. Fordi dette er beregningsmessig dyrt brukes en strategi for å reinitialisere tilstandsvariablene og deres deriverte etter en forstyrrelse. Tre ytelsesparametre relatert til justering av steglengde identifiseres for å ha et grunnlag for sammenligning mellom å bruke strategien for reinitialisering og å la algoritmen håndtere det uten ekstra funksjonalitet.

Regulator/forsterkersystemet beskrevet over utsettes for metningseffekter, og generator/stivt nett-systemet utsettes for den nevnte kortslutningsfeilen. I kortslutningstilfellet forbedres ytelsen med omtrent 5 % når systemet reinitialiseres med strategien mhp. ytelsesparameterne. I metningstilfellet registreres det ingen forbedring; resultatet er identisk med å la algoritmen håndtere forstyrrelsen på egenhånd.

Alle simuleringsresultatene presenteres grafisk, og tabeller som oppsummerer de mest interessante aspektene er inkludert. Noen uoverenstemmelser knyttet til steglengden i kortslutningssimuleringen observeres, diskuteres og vurderes videre i appendikset.

Til slutt presenteres anbefalt videre arbeid. Dette inkluderer å teste strategien for justering av steglengde og reinitialiseringsstrategiene på mer komplekse systemer, og å undersøke de inkonsekvente observasjonene fra kortslutningssimuleringen videre.

Preface

This master's thesis is the final result of the work done in the course "TET4900 - Elektrisk energiteknikk og smarte nett, masteroppgave" at the Department of Electric Power Engineering. A specialization project was carried out as preliminary work to lay the groundwork for this thesis during the fall 2018. This thesis represents the final assignment of the five year master's program Energy and environment at the Norwegian Univeristy of Science and Technology, NTNU, and corresponds to a workload of 30 ECTS.

Numerical analysis, methods/algorithms and dynamic simulation were all concepts I had only scratched the surface of a year ago. Complex concepts that are somewhat on the side of one's background experience and knowledge require time to mature. I believe that any author should strive to enable any peer to grasp the motivation, method, observations and conclusions from any publication. For this reason, I've put effort into making derivations, concept explanations and descriptions of the algorithm extensive and tangible enough so that anyone who are at the level I was a year ago should be able to grasp the subject of the thesis.

I would like to thank my supervisor, professor Olav B. Fosso. He laid the groundwork for this interesting thesis back in 1992/1993, and has provided me with insights and some state-space models for testing of the algorithm. I would also like to extend my gratitude to postdoctoral researcher Jalal Khodaparast Ghadikolaei for answering my questions with patience and an educational approach and providing tips about source material. Lastly, thanks to my fellow students at Energy and Environmental Engineering who have made these five years in Trondheim an amazing end to my 18 years in the Norwegian educational system.

Trondheim, June 10, 2019
Håkon Eidsvik

Contents

List of Figures

List of Tables

1 Introduction

This section aims to outline the background and motivation for Gear's method, and the framework on which this master's thesis is based.

1.1 Background and motivation

As penetration of renewable energy sources (RES) and energy storage in the power system increases, so does the number of power electronic devices that control them. An increasing amount of generation units with power electronic grid interfaces are becoming part of the world's power supply. Distributed photovoltaic (PV) systems are experiencing rapid growth worldwide, in many countries incentivized by energy policies [13]. This is turning consumers into prosumers, i.e. they are both consumers *and* producers of power – to quote my supervisor: "it was easier before, when load was simply load."

The increasing number of power electronic devices necessitate more detailed power system models, which increases the complexity of the systems of differential-algebraic equations (DAEs) that are used for state-space models. The DAE systems include more states and a wider range of time constants. Such DAE systems are simulated with numerical integration methods which estimate the solution at given points in time. Traditionally, a highly detailed simulation of such a model has been restricted to keeping the step length to a fraction of the smallest time constant to accurately simulate all phenomena in the system [7]. Due to the small step length, computation time quickly becomes a concern. This limits the simulation period that is practical to simulate. Performing simulations on medium-long time scales thus involves using a more aggregated power system model, neglecting the most detailed dynamics.

This is the motivation behind Gear's method, which is an integration method that is able to adapt its time steps to the transients occurring in the system. During rapid transients the step length is kept low, and during slow transients the step length is increased. This allows for dynamic simulation of highly complex, large power system models with the same method in the same simulation for arbitrarily large simulation periods.

1.2 Objective

The objective of this thesis can be split into two parts, where part two depends on the successful completion of part one.

The first objective is expanding the implementation of Gear's method from the specialization project in the Python programming language and assessing its performance in dynamic simulation of a simple power system. This builds on the work done in the specialization project, where a strategy for constraining the step length adjustment was developed and implemented based on the work done in literature reference [7]. The effect of this step length strategy is to be analyzed in dynamic simulation of power systems.

Once the first objective is completed, the performance of Gear's method in the case of disturbances is to be assessed. A strategy for re-initialization of state variables and their derivatives following a disturbance will be developed, implemented and compared with not using a strategy. The second objective is then to compare the simulation results of these approaches, and assess their performance in relation to each other based on several performance indicators to identify viable strategies. The findings will then be subject to further work in a master's thesis next year.

1.3 Method

A literature study on numerical integration methods, their stability and their use in different software applications for power system dynamic simulation will be carried out. The dynamic simulation functionality of three popular and widely used software applications will be studied, namely DIgSILENT PowerFactory, PSS/E and MATLAB.

Based on the work done in the specialization project and literature reference [7], Gear's method and its implementation will be described in depth. The implementation of the method will be extended from the specialization project, which will enable it to perform dynamic simulation of power systems based on the set of DAEs describing the system.

Further, strategies for re-initialization of state variables and their derivatives in the event of a disturbance will be implemented and simulated. Performance metrics of these strategies will be identified, and the strategies will be compared and assessed w.r.t. these metrics.

1.4 Scope and limitations

The implementation is limited to simulating two system models: one model of a synchronous machine connected by a parallel transmission line to a stiff grid, and one model of an automatic voltage regulator with amplifier controlling a generator with feedback through a measuring device.

The scope is limited to assessing the performance of Gear's method in the two systems in the event of a disturbance. The first system will be subject to a disturbance in the form of a short circuit of part of the transmission system, and the second system will be subject to a disturbance in the form of saturation of the amplifier output.

1.5 Relation between specialization project and master's thesis

This master's thesis is intended to be a continuation of the work performed in the specialization project, where Gear's method was implemented and tested on several different types of differential equations and a system of differential equations. While none of these simulations were related to the field of power system analysis, they were able to bring out several interesting characteristics of the step length adaption of Gear's method related to both damped and undamped sine waves, in addition to discontinuities in state variables. This has had great relevance for the work performed in this master's thesis.

The experience gained from the initial implementation and testing led to the development of a strategy with several measures to improve the step length adaption, and by extension the run time of the algorithm. More experience with the use of the step length strategy will be gathered in a power system context in the process of fulfilling the first objective of this thesis.

1.6 Structure

This section briefly describes the content of each section to give an overall idea of how the thesis is structured.

Section 2 aims to provide the reader with a good understanding of numerical integration of differential equations and its application in power system simulation. This is done through a thorough description of what numerical integration is, an example of a numerical integration method and a description of the equations that arise from power systems.

Section 3 describes the problem of how the step length behaves in the event of a disturbance in more detail than what is done here in the brief introduction. Further, the re-initializing strategy is introduced along with the performance metrics which will be used to assess its performance.

Section 4 contains the results from the implementation of Gear's method on the two power systems introduced here in section 1.4. The strategy for constraining the step length adjustment is implemented and tested on the two systems as a natural continuation of the specialization project. All simulations are presented with brief discussion of the results.

Section 5 takes the two systems from section 4 and subjects them to disturbances. The re-initialization strategies are implemented and tested, and all simulations are presented with brief discussion of the results.

Section 6 discusses the results obtained in sections 4 and 5 further. The causes for the observations are discussed, comparison of the different cases is carried out and some inconsistent results are pointed out and discussed.

Section 7 contains the conclusion. It summarizes the properties of the step length adaption strategy, the observations made in the implementation and their causes.

Section 8 recommends further work, which is intended to be a natural continuation of the work done here for the master's thesis to be conducted on the same topic next year.

Appendices A and B contains the full description of the second order Gear's method and some further testing related to inconsistent observations made in section 4.1.

2 Theory and preliminaries

This section aims to provide the reader with sufficient insight in numerical integration methods and their stability to understand Gear's method and the motivation behind its self-adapting step lengths.

2.1 Numerical integration of differential equations

In practically all fields of engineering, one encounters cases in which it is impossible to find the solution of a differential equation or set of differential equations analytically. These cases can be resolved by applying methods in which the solution to the differential equation(s) is approximated by splitting the function into sufficiently narrow intervals and then approximating the numerical value of the solution at the end of each of these intervals. The width of each interval is known as the step length, namely the amount of time we allow to pass before computing the next approximation of the numerical value of the solution.

The family of such methods is known as numerical integration methods, and the type of methods looked at in this thesis is called linear multistep methods [20]. One aspect that separates numerical and analytical integration is that the solution we obtain by using a numerical integration method is a set of approximate state values $(y_0, y_1, \ldots, y_{\text{final}})$ with corresponding time values $(t_0, t_1, \ldots, t_{\text{final}})$. This is opposed to analytical integration, where the solution obtained is a function expression from which we can calculate derivatives, areas and integrals.

Another aspect of numerical integration is that no exact answer is calculated, the solution is only approximated in each step. There will always be some truncation error present, and thus estimating and monitoring this error in each step is very important. To minimize this always present error, it is important that the step length of the integration is chosen sufficiently small.

A whole host of numerical integration methods exist which solve differential equations, as well as systems of differential equations, using a constant step length throughout the whole integration period. However, using a constant step length is impractical in numerical integration of differential equations [2]. Using a small step may lead to good accuracy, but it will increase calculation time and number of operations performed. Using a large step length may lead to low calculation time, but the calculation error may exceed acceptable limits and the calculation might diverge.

To illustrate the problem of having a large step length, the forward Euler's method will be used. It is a well known method for numerical integration of differential equations. In Euler's method, the differential equation to solve is given on the form

$$\frac{dy}{dt} = \dot{y} = f(y, t) \tag{1}$$

Additionally, the numerical value of the solution is known at a given time instant, i.e.

$$y(t_0) = y_0 \qquad \text{given} \tag{2}$$

Given the value in t_0, the aim is to calculate an approximation for the value in the next time, $y(t_1) = y_1$. t_1 is given as $t_1 = t_0 + h$ where h is the *step length*, i.e. the amount of time we allow to pass before performing another approximation of the solution. In Euler's method, y_1 is calculated as

$$y_1 = y_0 + hf(y_0, t_0) \tag{3}$$

More general, the numerical value of y in step ν is calculated as

$$y_{\nu+1} = y_\nu + hf(y_\nu, t_\nu) \tag{4}$$

Let $f(y,t) = \cos(t)$ for this illustration. Using a large step length leads to large error, and using a small step length leads to smaller error, as illustrated in fig. 1 below.

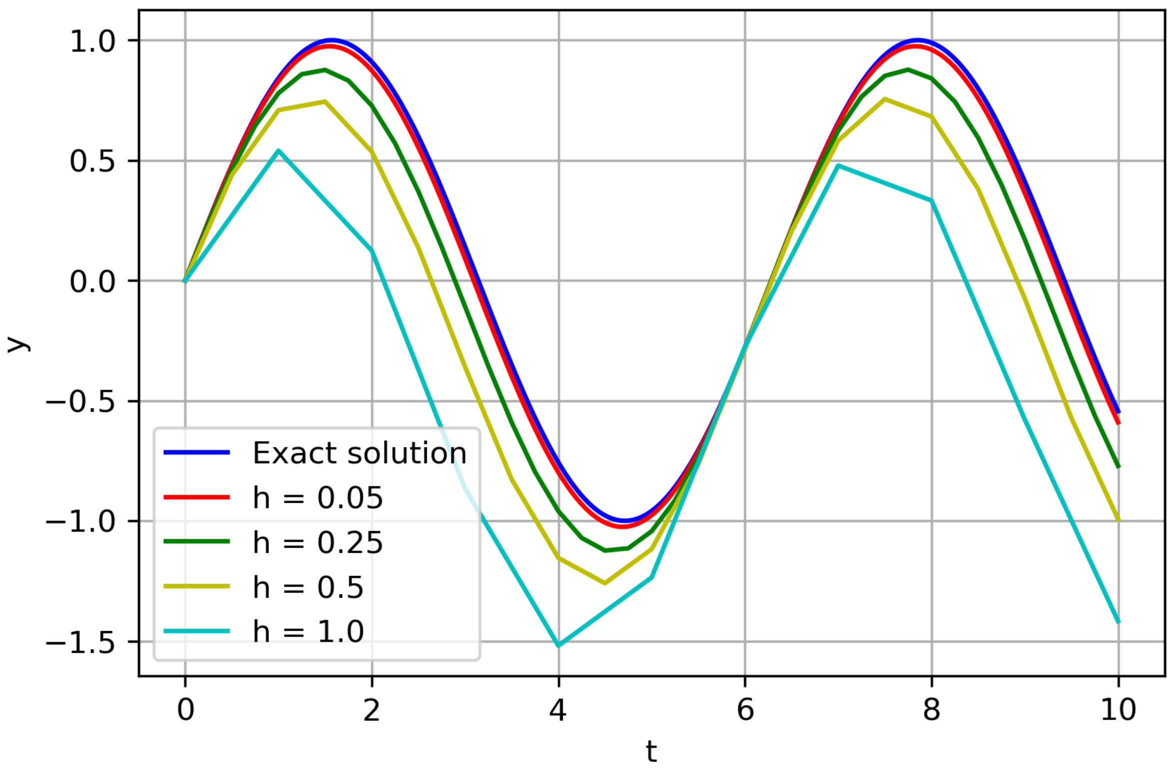


Figure 1: Euler's method applied to sine function

As can be seen from the figure, error is present in the approximation for all step lengths. This error can be largely mitigated by reducing the step length even further, but doing so increases computation time. The same considerations apply to systems of differential equations, as was discussed in the specialization project [6].

2.2 Implicit vs. explicit methods

Euler's method as described above is an explicit numerical integration method. This means that the vector containing all state variables is calculated from the values of the state variables from the previous steps, i.e. $y_{\nu+1}$ is explicitly calculated based on known values y_ν from previous calculations. This is opposed to an implicit method, where $y_{\nu+1}$ is a function of itself. An example of an implicit method is the trapezoidal rule, in which a step is performed as follows:

$$y_{\nu+1} = y_\nu + \frac{h}{2}(f(y_\nu, t_\nu) + f(y_{\nu+1}, t_{\nu+1})) \tag{5}$$

Note that $y_{\nu+1}$ appears on both the left and the right hand side. It is impossible to express $y_{\nu+1}$ without it being a function of itself. Thus, eq. (5) is implicit in $y_{\nu+1}$, and the trapezoidal rule is an implicit integration method. An iterative procedure needs to be applied in each step to iteratively find $y_{\nu+1}$. This means that implicit methods may be more complicated to implement and they require more arithmetic operations per step, meaning that they are more computationally expensive than explicit methods.

Implicit methods are used for solving arbitrarily stiff differential equation systems [10], i.e. systems having a large stiffness ratio. The stiffness ratio SR is defined as

$$SR = \frac{\Re(\lambda_{\max})}{\Re(\lambda_{\min})} \quad [12] \tag{6}$$

A high stiffness ratio often, but not in all cases, indicates that the system is comprised of differential equations describing both fast and slow phenomena, i.e. the system has a wide range of time constants associated with it.

2.2.1 Stability of explicit and implicit methods

While explicit methods are less computationally demanding and may be easier to implement, they suffer from the fact that they are not *numerically A-stable* [9, 11, 20]. This means that the step length is restricted by the rapid transients of the system, and must be kept to a fraction of the smallest time constant to maintain accurate tracking in the whole simulation period. Too large step length will cause the algorithm to diverge and the solution to "blow up."

In literature, the concept of numerical A-stability is useful to classify integration methods. To determine whether a method is A-stable or not, the integration method is applied to the linear test function

$$\dot{y} = \lambda y \tag{7}$$

where λ is a complex number $\alpha + j\beta$ with $\alpha \leq 0$ [5, 9]. The state variable is estimated as

$$y_{\nu+1} = R(h\lambda) y_\nu \tag{8}$$

where $R(h\lambda)$ is referred to as the *stability function* of the system. The function R is unique for every method. The test system is stable when λ has negative real part. For the numerical integration to be stable, we require:

$$|y_{\nu+1}| \leq |y_\nu| \tag{9}$$

for large ν. The condition in eq. (9) is fulfilled if and only if

$$|R(hq)| \leq 1 \tag{10}$$

which sets constraints for the step length h for each method in order for the numerical integration to be stable [5]. If an integration method is stable for all systems, not only the linear test system in eq. (7), the method is said to be A-stable [5, 9]. That is, when the equation system is on the form $\dot{y} = \mathbf{A}y$, we have

$$|R(\lambda_i h)| \leq 1, \quad \Re(\lambda_i) \leq 1 \quad \forall i \tag{11}$$

where $\Re(\lambda_i)$ denotes the real part of the eigenvalue λ_i of the system matrix $\mathbf{A}$.

If the method used to solve $\dot{y} = \mathbf{A}y$ is not A-stable, it implies that steady state may never be reached in the simulated system, leading to the conclusion that numerical A-stability is a necessity for dynamic simulation of power systems.

It can be shown that no explicit linear integration method is A-stable, and that the highest possible order of an A-stable implicit linear integration method is 2 [9]. The order of Gear's method in the implementation in this thesis is 2.

The trapezoidal rule discussed above (eq. (5)) is numerically A-stable, and the stiffness of the system to be solved will affect simulation accuracy but not numerical stability [11]. Using a small step length h will yield high accuracy, but may be unnecessary if no rapid transients are occurring. Using a large step length h will perform the simulation faster, but may yield inaccurate results in time periods where rapid phenomena dictate the system behaviour.

2.3 Application of numerical integration in power systems

In power system analysis, numerical integration methods must be applied in the simulation of transient behavior in the event of a disturbance or altered conditions for steady state. Such a change in conditions might occur due to a change in load demand, a change in production dispatch or any sort of disturbance. This type of simulation is known as dynamic simulation.

The transient behaviour of a power system is governed by a system of DAEs which model the components constituting the power system [16]. The differential equations arise from the dynamics of the rotors of generators and various controllers (voltage controllers, turbine governors, power electronic devices etc.) in the system, and the algebraic equations arise from the stators of generators, coordinate transformation to and from the d-q axis and the transmission network equations [14].

To simulate transient behaviour in instantaneous real time terms, one must apply methods which are able to solve the aforementioned DAEs with sufficient accuracy. For simulation of larger systems, computing time also becomes a concern. Therefore, the notion of variable step length is interesting to investigate for dynamic simulation.

2.4 The need for variable step length in numerical integration

As stated in section 1.1, the number of power electronic devices in the power system is increasing as penetration of RES is increasing. Each part of the block diagram of the control system of each of these devices contributes its own differential equation, the transient response of which is governed by a unique time constant belonging to that part of the control system. The time constants may vary in magnitude by a factor of several hundreds. These contributions of differential equations to the set of DAEs that govern the power system makes dynamic simulation an increasingly complex task to perform.

Having the DAE system consist of equations whose transient response is governed by time constants differing in size means that the DAE system contains equations representing both rapid and slow varying phenomena. As briefly discussed in section 2.2, DAE systems having this wide range of time constants are known as stiff systems. During rapid transients, a small step length is needed to simulate the trajectories of the state variables with sufficient accuracy. When the rapid transients die out and the slow ones take over as the system approaches steady state, the step length may be gradually increased while keeping the level of accuracy high. Once steady state has been reached and no transients are present, the step length may be further increased to its user defined maximum value. This maximum value must be sufficiently small in order for the integration method to be able to detect the presence of any disturbance and quickly reduce the step length accordingly.

This thesis, and the specialization project upon which it is based, is concerned with the numerical integration method known as Gear's method which incorporates all the topics discussed in the last pages.

2.5 Dynamic simulation in established software applications

In this thesis, three popular software applications for power system simulation (including dynamic simulation) have been looked at.

The first is DIgSILENT PowerFactory. On their website, it is stated that PowerFactory is a leading power system analysis software application that combines reliable and flexible system modelling capabilities with state-of-the-art algorithms [3]. In their brochure of advanced features, the following is stated about their algorithms for dynamic simulation: "*A-stable numerical integration algorithms supporting long-term stability simulations with integration step sizes ranging from milliseconds to minutes, individually selectable for each model*" [4].

The second is PSS/E, a popular power system analysis software application provided by Siemens. As of april 2018, PSS/E does *not* incorporate adaptive step size algorithms in dynamic simulation [15]. DIGsILENT PowerFactory outperforms PSS/E in computation time and accuracy when simulating a very large power system including 21 500 buses and 1 150 synchronous generators [15]. It is evident that PSS/E performs worse than DIGsILENT PowerFactory, and that it may suffer from not utilizing adaptive step size algorithms.

Lastly, the integration methods (solvers) of the MATLAB programming have been investigated. MATLAB offers four integration methods for solving stiff systems. They are `ode15s`, `ode23s`, `ode23t` and `ode23tb`, with `ode15s` being the recommended solver for DAE systems [17]. `ode15s` can vary its order from 1 to 5, and it has an option to use backward differentiation formulas (BDFs, discussed in [6]) [18]. The documentation states that "[...] it can use the backward differentiation formulas (BDFs, also known as Gear's method) [...]" [18]. This is a very interesting find, and a deep-dive into the documentation and source code of `ode15s` should be performed to compare it with the implementation of Gear's method this thesis is concerned with.

3 Problem description and method

As described in sections 1 and 2, an integration method for dynamic simulation of detailed power system models will need to be able to adapt its behaviour to both fast and slow transients in the same simulation. It is interesting to see how the components of the system react to disturbances, i.e. discrete events that change the operating conditions in a short amount of time.

In this thesis, two types of events are investigated to see how Gear's method as described in appendix A behaves in the case of a disturbance. The two events are short circuit of a transmission line and saturation of a voltage regulator.

In both events, two aspects are investigated:

1. **A strategy for better step length adaptation.** In [7], a strategy for safe guarding the step length adaptation was introduced. This has been further developed here, and its effects are quantified and compared with the no safe guarding case for both events. This is investigated in section 4.
2. **A strategy for limiting step length drop.** In a disturbance, the step length must decrease to accurately track the rapid transients taking place in the time during and following the event. A strategy for re-initialization of the state variables and the derivatives after a disturbance is implemented and assessed for both events. This is investigated in section 5.

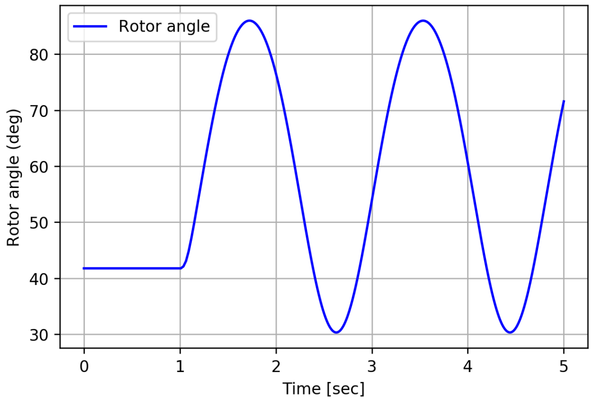
3.1 Short circuit of a transmission line

The active power delivered by a generator with internal field voltage E_f and rotor angle δ over a transmission network with equivalent reactance X_T (including synchronous reactance, neglecting all resistance) to a bus with voltage E_B is given as:

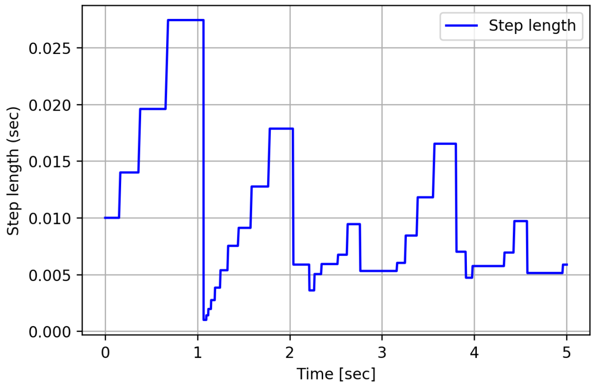
$$P_e = \frac{E_f E_B}{X_T} \sin \delta \tag{12}$$

If part of the transmission network experiences a fault, e.g. a short circuit in one of two parallel transmission lines, the reactance X_T will change from one instant to the next. Assuming that the fault is cleared within the critical clearing time, the rotor speed ω and angle δ need to adjust to the new operating scenario.

Gear's method can be used to simulate how ω and δ behave in this scenario. In steady-state conditions before the fault, the step length is expected to steadily increase before rapidly decreasing to a minimum when the short circuit occurs. The expectation harmonizes well with the simulation results presented in fig. 2 below.



(a) Tracking of δ



(b) Step length

Figure 2: Short circuit simulation. $t \in [0, 5]$

Notice the step length drop around $t = 1$ second, where the short circuit fault is detected by the algorithm. Many step length adjustments are performed to increase the step length after the fault. While a fine time resolution is necessary to track the state trajectories accurately during and immediately after a fault, the observed drop in step length may be excessively large, making it necessary to perform many step length adjustments to increase the step length. This is further investigated in section 5.1.

3.2 Saturation of a state variable

The regulators involved in exciters, governors and power electronic devices are subject to saturation effects when the operating condition of the system demands that the regulator output exceeds the maximum output value of the regulator for a period of time. The algorithm has problems handling simulation during the period of saturation.

When a regulator is saturated, it produces a constant output. Thus, the following constraints are put in place when a state variable y_i is saturated:

$$y_i = y_{i,\max} \tag{13}$$

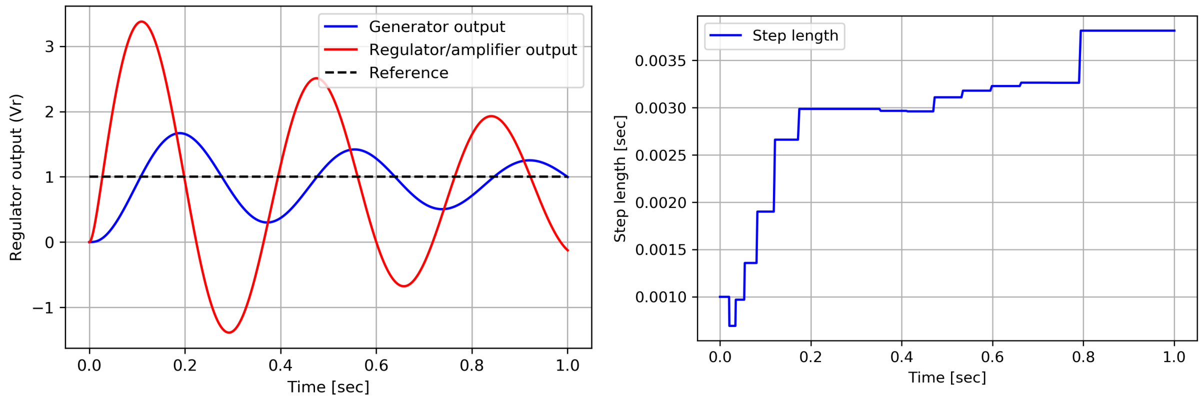
$$\dot{y}_i = \ddot{y}_i = 0 \tag{14}$$

The derivatives are forced to 0, and all corrector calculations related to the saturated variable are overridden during a state of saturation. When the derivatives are forced to 0, they change discontinuously from one instant to the next. As will be shown later, the step length may drop more than what is ideal when the derivatives experience such discontinuities.

This thesis aims to address the problem of excessive step length reduction in the case of a saturated state variable. The following subsections aim to display differences in the observed behaviour of the step length adaptation in "clean" problems without disturbances and problems where disturbances in the form of saturation effects are present. Once the difference in behaviour between the two types of problems has been displayed, different strategies for improving poor step length adaption will be presented along with the method of how these strategies will be compared.

3.2.1 "Clean" problems

The term "clean" problem refers to a problem in which there are no disturbances in the form of saturation effects present. The figure below presents a simulation result that is representative for the behaviour of the step length in a "clean" problem.



(a) Tracking of $y(t)$ (b) Step length

Figure 3: Simple voltage regulator, clean problem, $t \in [0, 1]$

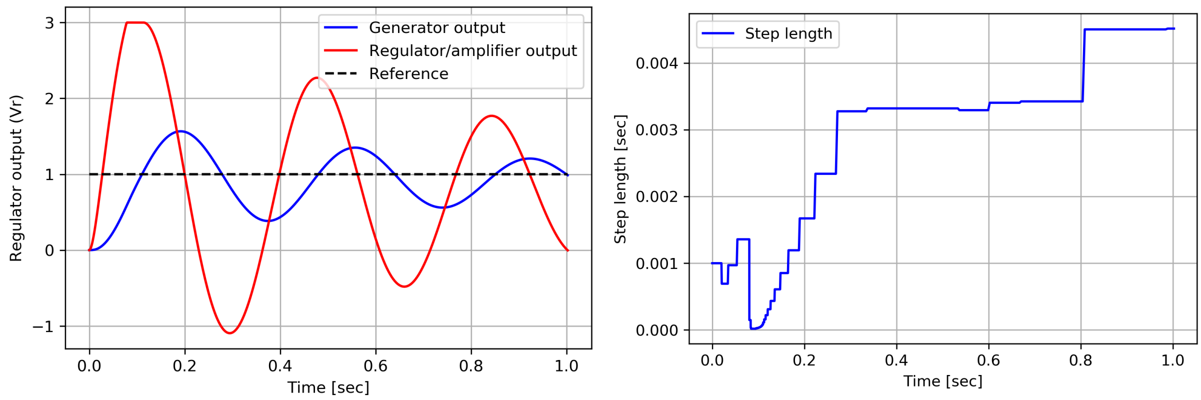
As is evident from fig. 3, the step length increases as the state variables oscillate with less amplitude and approach steady state. The step length increases rapidly due to a low initial value before steadily increasing as the magnitude of the derivatives of the state variables decrease.

The step length is increased 10 times and decreased 4 times during the simulation period. The minimum step length is 0.693 ms and the average step length is 2.663 ms.

The simulated system consists of a regulator with an amplifier, a generator and a measuring unit in the feedback loop. The largest and smallest time constants differ by a factor of 20. More on the details of the system modelling in section 4.2.

3.2.2 Saturation and discontinuities

When introducing saturation effects to the same system as the one in fig. 3, the step length behaviour changes. A maximum value of 3.0 is implemented to the state variable representing the output of the amplifier of the regulator. The following graphs are then produced:



(a) Tracking of $y(t)$ (b) Step length

Figure 4: Simple voltage regulator, saturated problem, $t \in [0, 1]$

It is observed in fig. 4 that the step length is dramatically reduced during the state of saturation. The discontinuity in the state variable and its derivatives seem to cause an excessively large drop in step length.

For comparison with the "clean" problem: The step length is increased 40 times and decreased 6 times during the simulation period. The minimum step length is 0.019 ms and the average step length is 0.676 ms. The difference between the simulation of the "clean" problem and the problem with saturation is presented in table 1.

Table 1: "Clean" vs. saturated problem, $t \in [0, 1]$

Problem	Steps taken	Step incr.	Step red.	Total adj.	h_{min} [ms]	h_{avg} [ms]
Clean	376	10	4	14	0.693	2.663
Saturated	1479	40	6	46	0.019	0.676

3.3 Re-initialization strategies post disturbance

In this thesis, two approaches to step length adaptation following a disturbance are assessed and compared. They are:

1. **Letting the corrector iterations handle it.** In this case, the disturbance is introduced without any measures regarding the drop in step length. The step length is allowed to drop as much as the corrector iterations warrant. This strategy will subsequently be referred to as strategy 1.

2. **Applying re-initialization measures.** In this case, the corrector iteration taking place in the step following the disturbance is overridden by a re-initialization calculation. The derivation of the re-initialization formulas for the first and second order derivatives is done in [7], and re-iterated in the following paragraphs. This strategy will subsequently be referred to as strategy 2.

Re-initializing the first order derivatives simply consists of performing an explicit calculation:

$$\dot{y}_d = f(y_d, y_a, t) \tag{15}$$

where y_d denotes the differential state variables and y_a denotes the algebraic state variables of the system. Together, the vectors y_d and y_a constitute the state variable vector y:

$$y = \begin{bmatrix} y_d \\ y_a \end{bmatrix} \tag{16}$$

Using the chain rule of derivation, the second order derivatives can be found as follows:

$$\ddot{y}_d = \frac{\partial f}{\partial y_d}\frac{\partial y_d}{\partial t} + \frac{\partial f}{\partial y_a}\frac{\partial y_a}{\partial t} \tag{17}$$

In literature reference [7], the algebraic equations are all represented by a function of bus voltages referred to the global D-Q reference frame. All algebraic variables are then of the form $y_a = V_{DQ}$, and eq. (17) can then be rewritten as

$$\ddot{y}_d = \frac{\partial f}{\partial y_d}\frac{\partial y_d}{\partial t} + \frac{\partial f}{\partial V_{DQ}}\frac{\partial V_{DQ}}{\partial t} \tag{18}$$

To find an expression for the time derivative of the voltages, the network equation is used. Using the well known admittance matrix formulation:

$$Y_{DQ}V_{DQ} = I(y_d, V_{DQ}) \tag{19}$$

Derivating eq. (19) to express the time derivative of the voltages:

$$Y_{DQ}\frac{\partial V_{DQ}}{\partial t} = \frac{\partial I}{\partial y_d}\frac{\partial y_d}{\partial t} + \frac{\partial I}{\partial V_{DQ}}\frac{\partial V_{DQ}}{\partial t} \tag{20}$$

When rearranging eq. (20), one obtains

$$\frac{\partial V_{DQ}}{\partial t} = \left(Y_{DQ} - \frac{\partial I}{\partial V_{DQ}}\right)^{-1}\frac{\partial I}{\partial y_d}\dot{y}_d \tag{21}$$

As discussed in [7], solving eq. (21) is time consuming. The significance of the contribution from the voltages, i.e. the solution of eq. (21), to the second order derivatives has to be evaluated for larger systems. For now however, it is assumed that the contribution from eq. (21) can be neglected. This greatly simplifies the re-initialization expression of the second order derivatives in eq. (17) to:

$$\ddot{y}_d = \frac{\partial f}{\partial y_d}\dot{y}_d \tag{22}$$

The expression obtained in eq. (22) will be used later in section 5 for re-initialization of the derivatives of the state variables after a disturbance.

3.4 Performance metrics

To quantitatively compare the two strategies for re-initialization, some performance metrics that they have in common are required. A performance metric refers to a metric that reflects performance of the algorithm in key areas related to step length adaptation. In order to have a basis of comparison for the different re-initialization strategies, performance metrics related to measurements of efficiency in regards to computing time are needed. The strategies will be ranked based on how they score on these metrics.

The following metrics will be used to rank the performance of the strategies:

1. **Total number of step length adjustments.** There are two aspects as to why one wants to keep the number of step length adjustments to a minimum.

   (a) Each adjustment of the step length necessitates a recalculation of the correction matrix (see appendix A.1.2). This is computationally expensive, especially for larger systems, and should be avoided when possible.

   (b) Each step in the simulation involves at least one corrector iteration. As long as the step length remains constant, the corrector iteration is performed exactly once each step. Changing the step length means that another iteration must be performed in the same step. Because the corrector iteration involves applying Newton-Raphson to solve an equation system of the same dimension as the original system, it is the most computationally expensive part of a given step.

   Thus, any strategy for re-initialization should aim to minimize the number of step length adjustments.

2. **Average step length during the simulation.** Having a high average step length indicates that the algorithm is able to perform the simulation with desired accuracy by undertaking fewer steps than an algorithm with lower average step length. Fewer steps taken, accompanied by fewer step length adjustments, indicates less computational resources used.

3. **Minimum step length during the simulation.** As previously discussed, excessive drop in step length following a discontinuity is a problem. The step length takes longer to increase the more it drops in the period of saturation. Therefore, the metric of minimum step length is included as a performance metric.

4 Power system implementation of Gear's method

This section describes the results from implementation of Gear's method in a power system context. As stated in section 1.5, the specialization project included an implementation of Gear's method which performed simulations of quadratic functions, sine waves, damped sine waves and a system of differential equations. This work produced valuable experience, but none of the simulations were performed on a model of a power system.

In this section, an implementation is done on the two systems described the introduction. This is done to ensure that the original implementation from the specialization project, including the strategy for adapting step length (see appendix A.2), is applicable and functional in a power system context.

The circuit and operating conditions for the short circuit case are taken from example 13.1 in Kundur [11]. The model of the voltage regulator/amplifier system is based on lecture slides provided by professor and supervisor of this thesis, Olav B. Fosso.

4.1 Implementation for simple power system

Figure 5 below is a circuit equivalent drawing of the power system from example 13.1 in Kundur. It consists of a synchronous generator modelled by the classical model (described by differential equations governing $\Delta\omega$ and δ), a transformer and two parallel transmission lines. Load demand is represented by an infinite bus at constant voltage and frequency. All per unit values are referred to a 2220 MVA, 24 kV base.

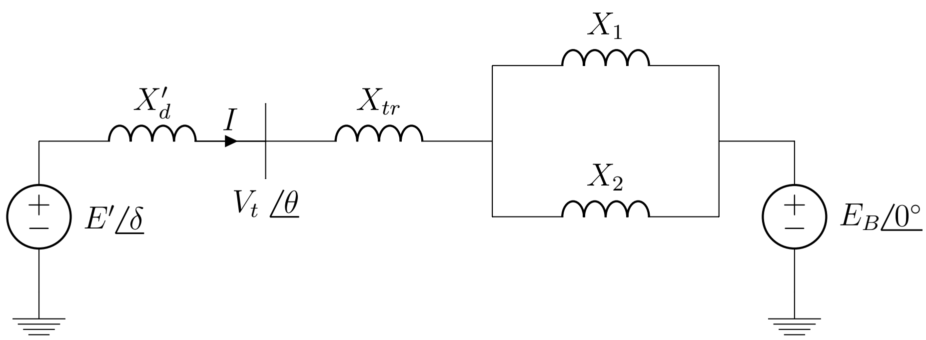


Figure 5: System equivalent circuit from ex. 13.1, Kundur

In this example, a three phase bolted short circuit fault is simulated in transmission line 2 by connecting X_2 directly to ground. The fault is cleared by removing the line completely, i.e. removing the reactance X_2 from the system. The internal voltage angle δ is referred to the infinite bus voltage E_B. For the generator parameter values, we have:

$$X'_d = 0.3 \text{ pu} \qquad H = 3.5 \text{ s} \qquad K_D = 0$$

The per unit reactances of the transmission system are:

$$X_{tr} = 0.15 \text{ pu} \qquad X_1 = 0.5 \text{ pu} \qquad X_2 = 0.93 \text{ pu}$$

The initial per unit operating conditions are:

$$P = 0.9 \text{ pu} \qquad Q = 0.436 \text{ pu} \qquad |V_t| = 1.0 \text{ pu}$$

which enables us to calculate the following initial values for our system:

$$V_t = 1.0\underline{/28.34^\circ} \text{ pu} \qquad E_B = 0.90081\underline{/0^\circ} \text{ pu} \qquad E' = 1.1626\underline{/41.77^\circ} \text{ pu}$$

The classical model of the synchronous machine in this example is represented by the following differential equation system:

$$f_1 = \Delta\dot{\omega}_r = \frac{1}{2H}(P_m - P_{\max}\sin\delta - K_D\Delta\omega_r) \tag{23}$$

$$f_2 = \dot{\delta} = \omega_0\Delta\omega_r \tag{24}$$

where $P_m = 0.9$ pu and $\omega_0 = 2\pi f_{\text{el}}$. The system frequency f_{el} is 60 Hz. The expression for $P_{\max}$ is known from electrical machinery fundamentals, and is given by

$$P_{\max} = \frac{|E'||E_B|}{X_{\text{eq}}} \tag{25}$$

The equivalent transmission system reactance X_{eq} changes as the operating scenario changes from steady state pre-fault to short circuited during fault to post-fault. This results in the following values for $P_{\max}$ in per unit:

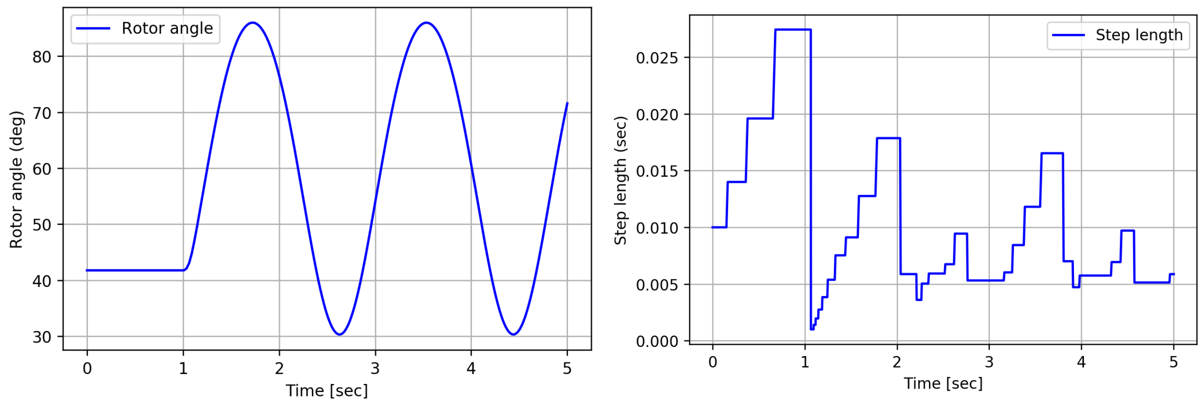
$$\text{Pre-fault: } P_{\max} = 1.351 \text{ pu} \tag{26}$$

$$\text{During fault: } P_{\max} = 0.0 \text{ pu} \tag{27}$$

$$\text{Post-fault: } P_{\max} = 1.1024 \text{ pu} \tag{28}$$

4.1.1 Simulation results, no damping

The three phase bolted fault inception is at $t = 1$ s. The fault is cleared after $t = 0.086$ s, which is close to the critical clearing time of the fault [11]. The following graphs are then produced for the solution of differential eq. (24) and the step length:



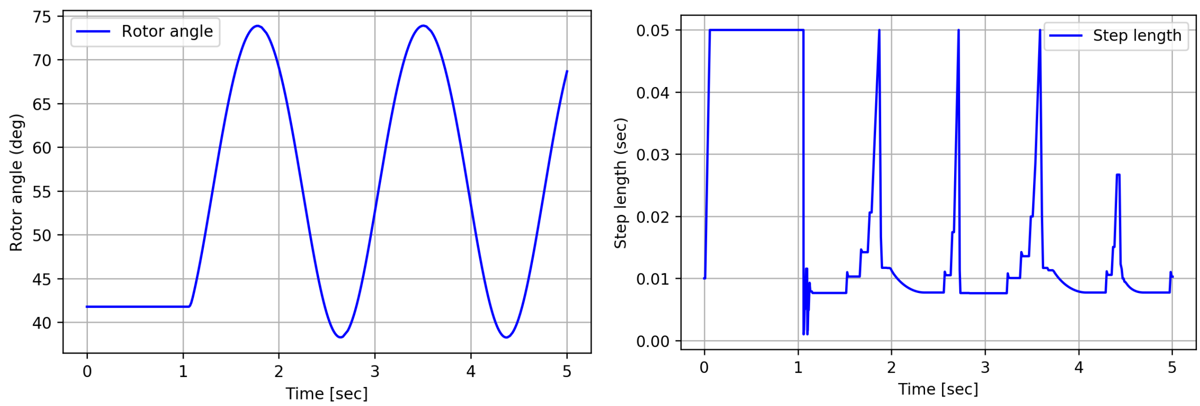
(a) Tracking of δ (b) Step length

Figure 6: Short circuit simulation. Strategy applied. $t \in [0, 5]$

The simulation results shown in fig. 6 demonstrates that Gear's method is able to simulate a simple power system governed by differential equations while adapting step length to maintain satisfactory accuracy. The step length is steadily increasing before $t = 1$ s, before rapidly decreasing to its minimum to accurately simulate the transients taking place in the system. As the sine-shaped oscillations of δ approach maximum and minimum points, the first order derivatives become smaller and smaller before changing sign. It is around these changes of first order derivative signs that the step length is at its highest. Around the changes of second order derivative signs, the step length is reduced. This harmonizes with the observations made in [6] for general sine and damped sine waves.

In the simulation above, the step length is allowed to change every 15 steps and constraints of $h_{\min} = 0.1$ ms, $h_{\max} = 50$ ms, $h_0 = 100h_{\min}$ and $h_{\nu+1} \leq 2h_\nu$ are in place in order to safe guard the step length adaption.

If the step length is allowed to change every step and increase to more than twice the value of the previous step length, the following graphs are produced:



(a) Tracking of δ (b) Step length

Figure 7: Short circuit simulation. No strategy. $t \in [0, 5]$

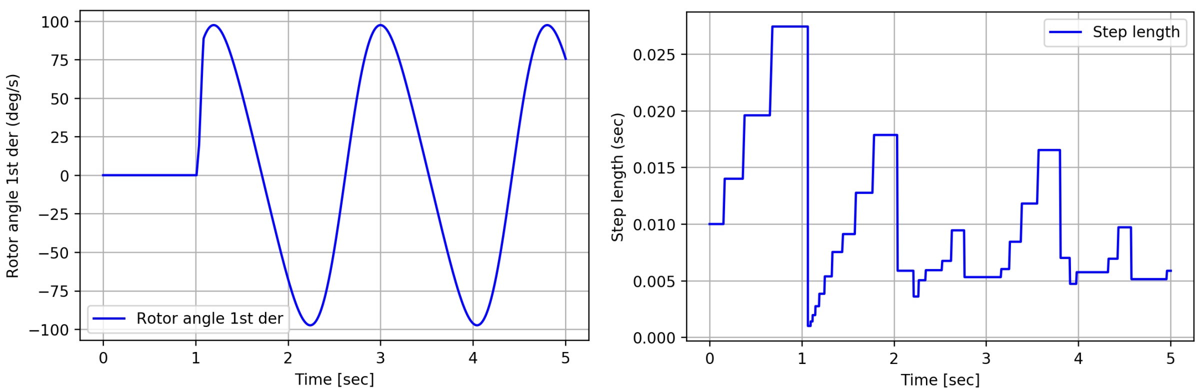
The floor/ceiling values of $h_{\min}$ and $h_{\max}$ are kept. The difference between the step length adaption in the two cases is summarized in table 2 below.

Table 2: No constraints vs. constraints for power system, $t \in [0, 5]$

Constrained h	Steps taken	Step increases	Step reductions	Total adjustments
Yes	773	31	8	39
No	473	51	111	162

It is evident that imposing the constraints on h mentioned above leads to a higher number of steps taken to complete the simulation, i.e. a lower average step length. This requires the performance of more arithmetic operations for similar tracking accuracy. In a simulation with no constraints other than max/min values for h, the step length is adjusted more than four times as often as in a simulation with constraints, meaning that the correction matrix is recalculated more than five times as often in the unconstrained case. This is computationally expensive, and may result in longer computation time. The span between the largest and the smallest step length is also observed to be higher for the unconstrained case. The constraint strategy will need to be optimized and implemented for a larger, more complex power system in order for this result to be verified in general power system system application.

What is evident from the results presented in both fig. 6 and fig. 7 is that the step length drops to its minimum value as the operating conditions suddenly change. Observe the tracking of the derivative $\dot{\delta}$ and the step length in the following graphs:



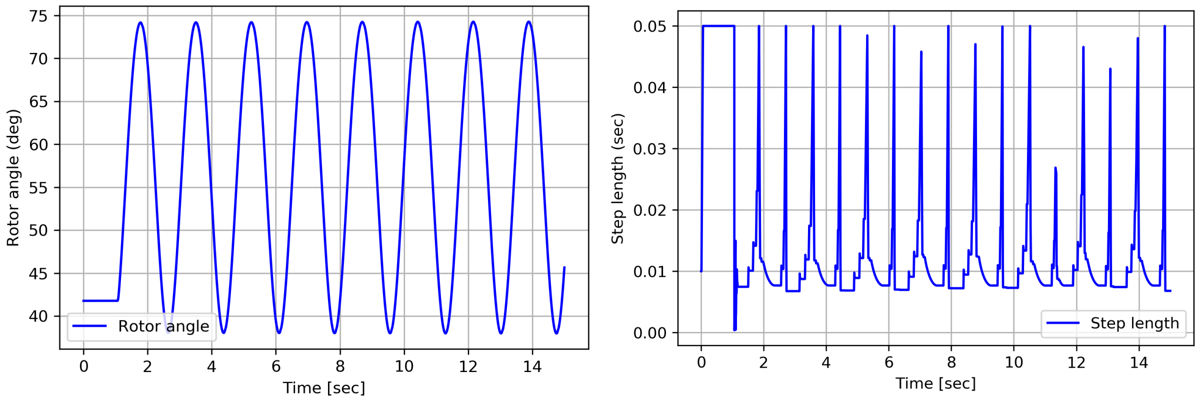
(a) Tracking of $\dot{\delta}$ (b) Step length

Figure 8: Short circuit simulation, first order derivative. $t \in [0, 5]$

The graph in fig. 8 is taken from the results of the same simulation as the graphs in fig. 6, hence the step length graphs in fig. 6b and fig. 8b are identical.

Upon fault inception, the conditions for steady state change from one step to the next. So do the derivatives, hence the discontinuity problem involving drastic step length reduction.

When extending the simulation period to 15 seconds, the same conclusions with regards to the step length strategy may be drawn. With no constraints on the step length, the following graphs are produced:



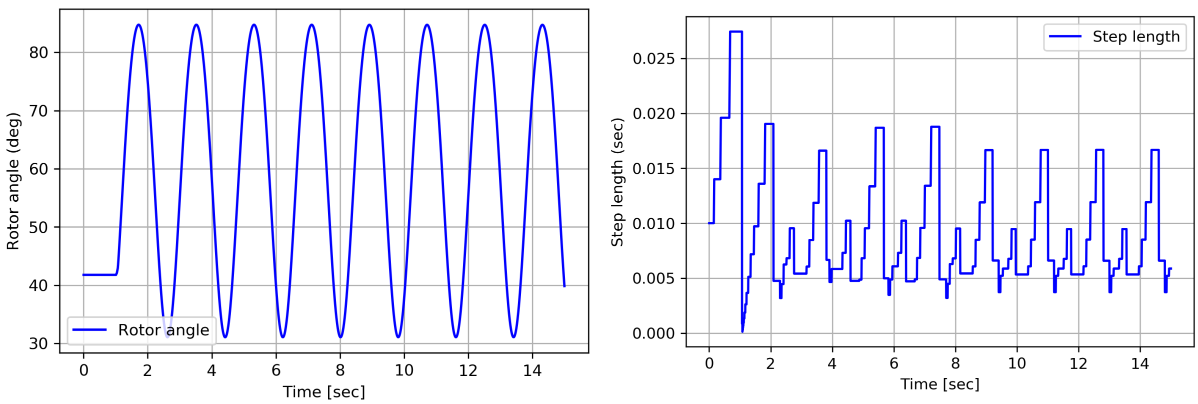
(a) Tracking of δ

(b) Step length

Figure 9: Short circuit simulation. No strategy. $t \in [0, 15]$

The step length pattern of increasing to $h_{\max}$ and then rapidly decreasing repeats throughout the simulation period. This leads to the result from the corrector iteration often being rejected, warranting a step length reduction.

When the same constraints as before are imposed on h, the following graphs are produced:



(a) Tracking of δ

(b) Step length

Figure 10: Short circuit simulation. Strategy applied. $t \in [0, 15]$

The step length fluctuates as in fig. 6, and far fewer step length adjustments are performed in this simulation than the one presented in fig. 9. The difference between the step length behaviour in the two cases is summarized in table 3 below. The results correspond well with the results for the shorter time period in table 2.

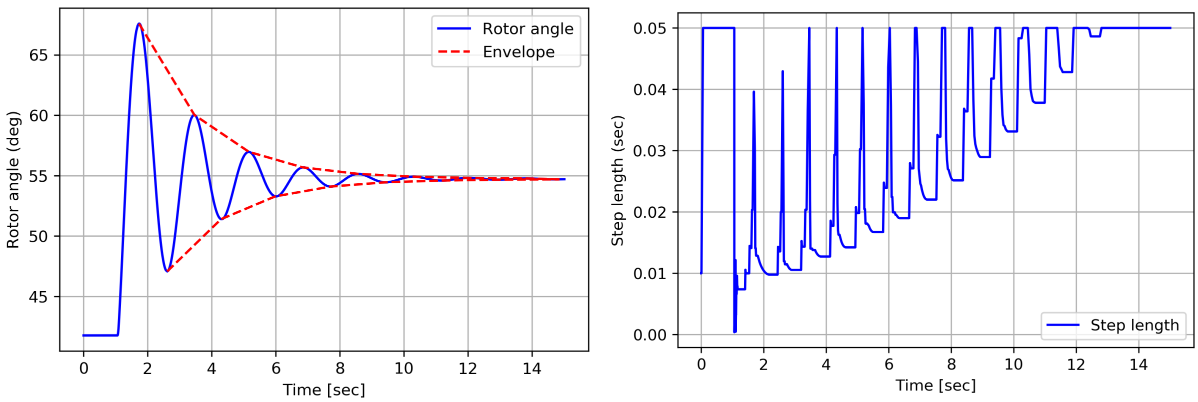
Table 3: No constraints vs. constraints for power system, $t \in [0, 15]$

Constrained h	Steps taken	Step increases	Step reductions	Total adjustments
Yes	2171	77	25	102
No	1526	97	446	543

4.1.2 Simulation results, damping

The differential eqs. (23) and (24) model the system in this case as well, but the damping coefficient is now set to $K_D = 0.30$ to induce damping effects and stabilize the voltage angle δ to its new post-fault value. Calculating δ from eqs. (12) and (28), we obtain the post-fault value $\delta = 54.7°$. The angle should stabilize to this value with a damping coefficient $K_D > 0$.

Considering the conclusions made in the specialization project, h is expected to steadily increase as the state variables approach their new steady-state values and their derivatives approach 0 [6]. With no constraints other than max/min imposed on h, the following graphs are produced when simulating for 15 seconds:



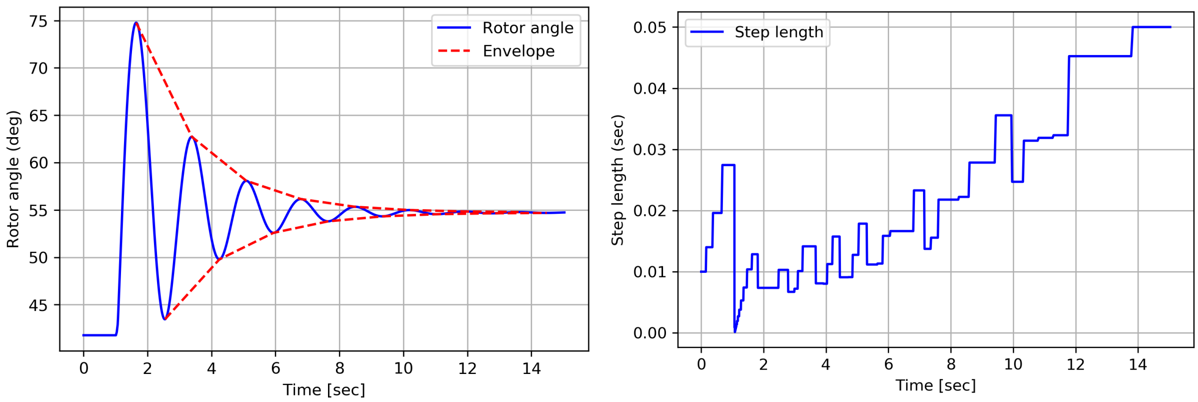
(a) Tracking of δ

(b) Step length

Figure 11: Short circuit simulation, damped. No strategy. $t \in [0, 15]$

As fig. 11 shows, the step length enters an increasing trend as δ approaches its steady state value of 54.7°. h attempts to increase beyond its maximum of 0.05 seconds and is quickly reduced after every sharp increase.

When applying the constraints from appendix A.2 to the step length and performing the same simulation, the following graphs are produced:



(a) Tracking of δ (b) Step length

Figure 12: Short circuit simulation, damped. Strategy applied. $t \in [0, 15]$

The step length behaviour enters an increasing trend in fig. 12, but no sharp increases are present. It reaches its maximum value just before $t = 14$ seconds. The difference in step length behaviour between the two cases is summarized in table 4 below.

Table 4: No constraints vs. constraints for damped power system, $t \in [0, 15]$

Constrained h	Steps taken	Step increases	Step reductions	Total adjustments
Yes	1030	40	10	50
No	645	117	184	301

As is the case for all the simulations to this point, more steps are taken with a constrained step length. When the length of the simulation period is the same, more steps means lower average step length which may imply better tracking overall. In addition, far fewer step length adjustments are needed to perform the simulation, saving computational resources related to rebuilding the correction matrix.

4.2 Implementation for voltage regulator

A large part of the motivation behind Gear's method is its ability to efficiently integrate and solve systems comprised of differential equations whose transient responses are governed by time constants of different magnitudes. Therefore, a model of a generator with a regulator, amplifier and a measuring unit is implemented in this thesis. Figure 13 below shows the block diagram for this model.

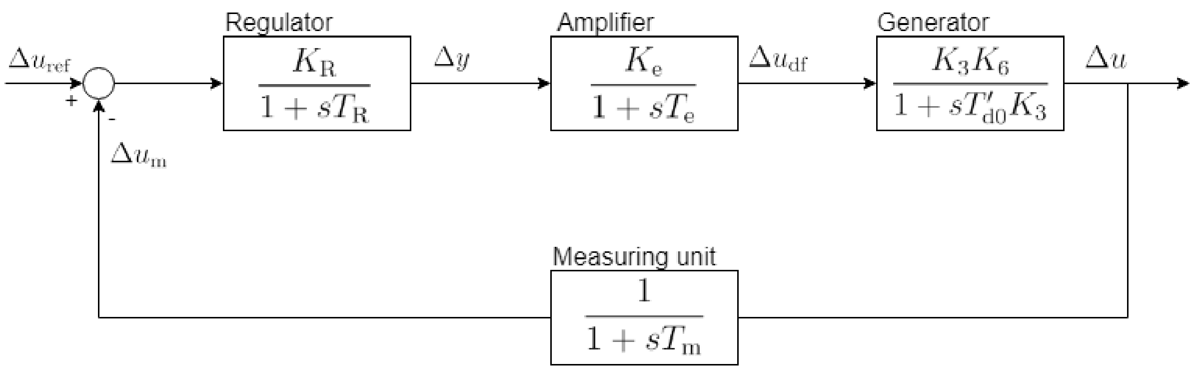


Figure 13: Voltage regulator and generator block diagram [8]

This system includes four time constants of different magnitude. An implementation of Gear's method is used in this section to simulate the system in fig. 13 to make observations related to step length adaptation with and without constraints as was done in section section 4.1.

The differential equations governing the system are:

$$f_1 = \Delta \dot{y} = \frac{1}{T_{\rm R}} \left(K_{\rm R}(\Delta u_{\rm ref} - \Delta u_{\rm m}) - \Delta y\right) \tag{29}$$

$$f_2 = \Delta \dot{u}_{\rm df} = \frac{1}{T_{\rm e}} \left(K_{\rm e}\Delta y - \Delta u_{\rm df}\right) \tag{30}$$

$$f_3 = \Delta \dot{u} = \frac{1}{T'_{\rm d0}K_3} \left(K_3K_6\Delta u_{\rm df} - \Delta u\right) \tag{31}$$

$$f_4 = \Delta \dot{u}_{\rm m} = \frac{1}{T_{\rm m}} \left(\Delta u - \Delta u_{\rm m}\right) \tag{32}$$

where $\Delta u_{\rm ref}$ is set equal to 1.0.

In the simulations, the ratio between the largest and smallest time constant is 30. The time constants are:

$$T_R = 0.01 \text{ s} \qquad T_e = 0.15 \text{ s} \qquad T'_{\rm d0} = 0.3 \text{ s} \qquad T_{\rm m} = 0.02 \text{ s}$$

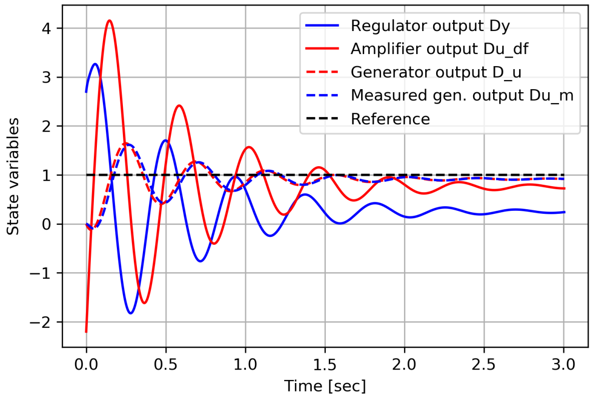
The different gains K are given as:

$$K_{\rm R} = 3.0 \qquad K_{\rm e} = 3.0 \qquad K_3 = K_6 = 1.1$$

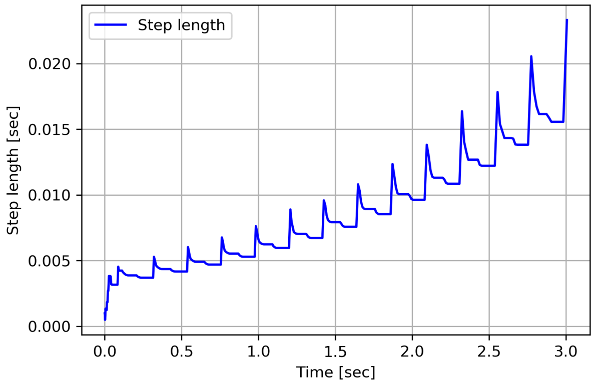
4.2.1 Simulation results

The system is modelled without any saturation effects. This is done in order to demonstrate the ability of Gear's method to simulate a system with time constants of different magnitudes, and to observe how the step length adapts as such a system approaches steady state.

For the step length, the following is implemented: $h_{\rm min} = 0.1$ ms, $h_{\rm max} = 50$ ms and $h_0 = 10h_{\rm min}$. When no strategy to impose constraints other than min and max on the step length is implemented, the following graphs are produced for all state variables and the step length:



(a) Tracking of y

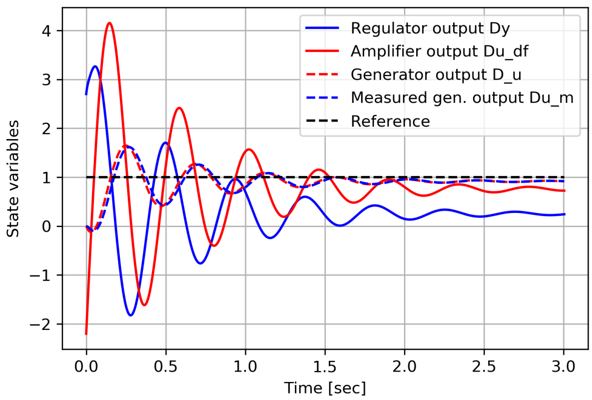


(b) Step length

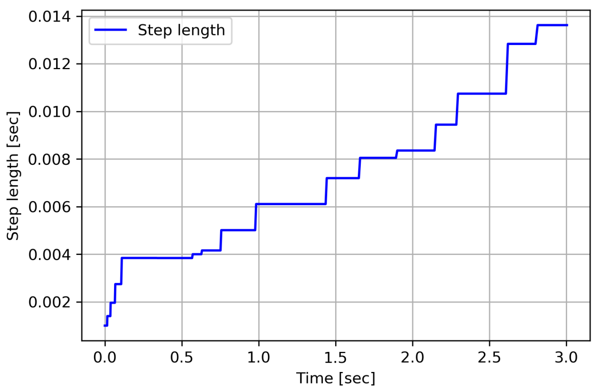
Figure 14: Voltage regulator system simulation. No strategy. $t \in [0, 3]$

The simulation results shown in fig. 14 demonstrate that Gear's method is able to simulate the system described in section 4.2. The same observations that were made in simulation of the power system in section 4.1 can be made here. As the state variables approach steady state, the step length enters an increasing trend. However, because no constraints are imposed on the step length, it is able to increase in an uncontrolled manner. It is observed that every increase in h is immediately followed by a series of decreases. This is again unfortunate. By allowing this uncontrolled increase of h, the solution in the next step is more likely to be rejected in the correction phase due to excessive truncation error.

It is clear that a strategy for constraining the step length adaption is needed. The following strategy is applied to the same system: the step length is kept constant for 15 steps before being allowed to change and $h_{\nu+1} \leq 2h_\nu$. The following graphs are then produced for the same system with the same simulation period:



(a) Tracking of y



(b) Step length

Figure 15: Voltage regulator system simulation. Strategy applied. $t \in [0, 3]$

From the results shown in fig. 15, it is clear that the tracking of the state variables is good. No differences are observed between fig. 14a and fig. 15a. The key difference is the step length behaviour. With constraints imposed on h, one can observe a steadily increasing trend in the step length as in the case without constraints. Only a single decrease in h occurs around $t \approx 0.34$ s, where the step length decreases by $2.16 \cdot 10^{-6}$ seconds which can be considered negligible.

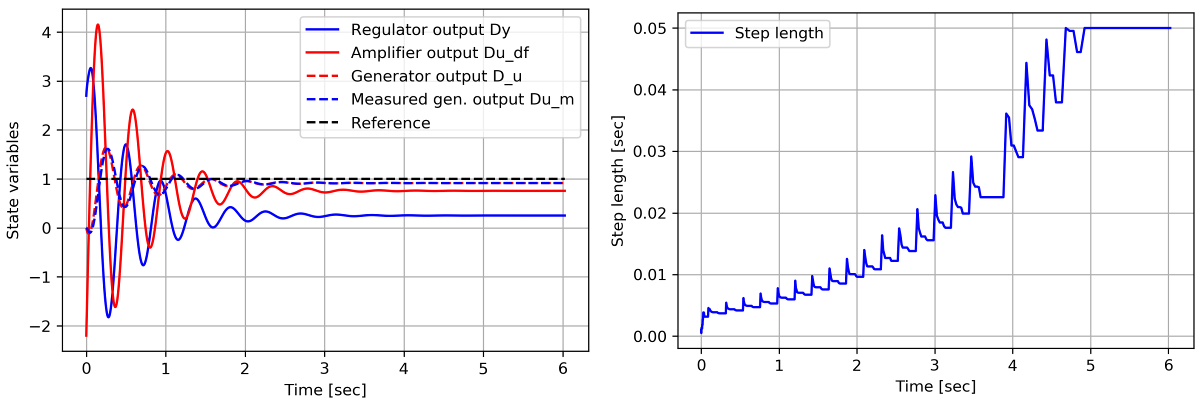
The simulation results regarding step length behaviour of the unconstrained and constrained case are summarized in table 5 below.

Table 5: No constraints vs. constraints for AVR system, $t \in [0, 3]$

Constrained h	Steps taken	Step increases	Step reductions	Total adjustments
Yes	510	16	1	17
No	441	19	153	172

The same general observations are made here as in table 2. A higher number of steps are taken in the constrained case, indicating lower average step length which may imply better accuracy. The key difference is in the number of step length adjustments. For the constrained case, the step length is adjusted only one tenth as often as in the unconstrained case. This is means that when all differential equations are linearized, the correction matrix only has to be recalculated one tenth as often with constraints imposed on the step length.

When doubling the simulation period to 6 seconds, the following graphs are produced when no constraining strategy is imposed on h:

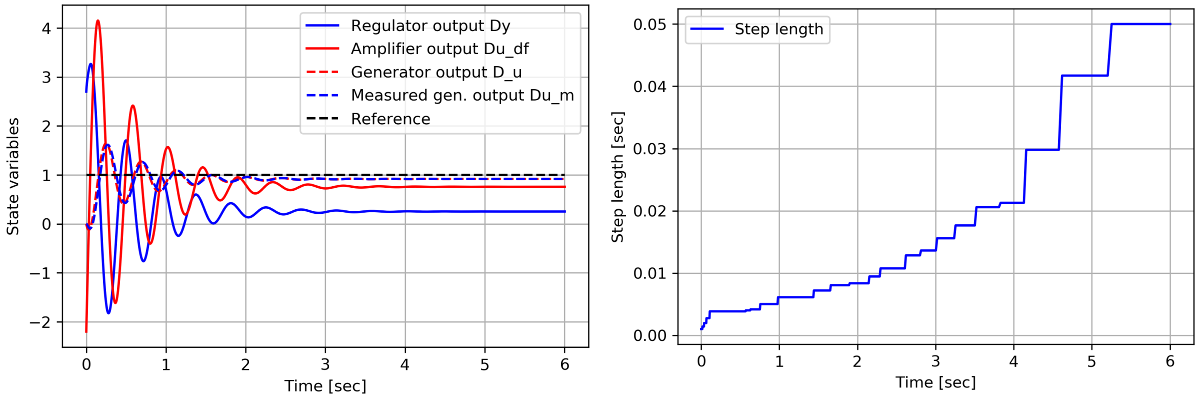


(a) Tracking of y (b) Step length

Figure 16: Voltage regulator system simulation. No strategy. $t \in [0, 6]$

The pattern of sharp step length increases followed immediately by a series of decreases observed in fig. 14 is present in fig. 16 as well. When extending the simulation period, the pattern continues and the rate of step length increase keeps rising until h reaches its maximum permitted value of $h_{\text{max}} = 50$ ms.

When applying the step length constraining strategy to the simulation with the same simulation period, the following graphs are produced:



(a) Tracking of y (b) Step length

Figure 17: Voltage regulator system simulation. Strategy applied. $t \in [0, 6]$

In fig. 17, the same increasing trend in h as in fig. 16 is observed. The step length increases more and more rapidly as the state variables approach steady state. h reaches its maximum of $h_{\text{max}} = 50$ ms only slightly later than in the case without step length constraints. The step length results from the two cases are summarized in table 6 below.

Table 6: No constraints vs. constraints for AVR system, $t \in [0, 3]$

Constrained h	Steps taken	Step increases	Step reductions	Total adjustments
Yes	616	23	1	24
No	532	45	179	224

When simulating a more complex system, the observations related to steps taken and total number of step length adjustments from section 4.1 make thsemselves even clearer in favor of the strategy for adjusting the step length. 16 % more steps are taken when h is constrained, and 933 % more step length adjustments are performed when h is not constrained. This is opposed to 42 % more steps taken with constrained h and 532 % more step length adjustments performed for non-constrained h in the short circuit case. The tests on a more complex system thus point in favor of the strategy for adjusting the step length.

From this point, all simulations will be performed with fixed step length for a number of steps and $h_{\nu+1} \leq 2h_\nu$ unless otherwise is stated.

5 Disturbances and discontinuities

In this section, the two systems from section 4 will be used for simulation with the goal of implementing strategies that can prevent the step length from experiencing an excessive decrease in the event of a disturbance. The system from section 4.1 will be subject to a disturbance in the form of a short circuit (as was the case in section 4.1), and the system from section 4.2 will be subject to a disturbance in the form of a saturation of the amplifier output.

The problem of excessive step length decrease in the event of a disturbance was described in section 3, and the decrease phenomenon itself has been observed in figs. 4b, 6b, 7b and 8b.

5.1 Short circuit in simple power system

The simulation results from fig. 6b will be used as a basis for comparison, as the simulation has implemented the same strategy for constraining the step length as the simulation in this section does.

Functionality that is able to detect the disturbance is added to the program. The functionality includes the following calculations for re-initialization of the derivatives:

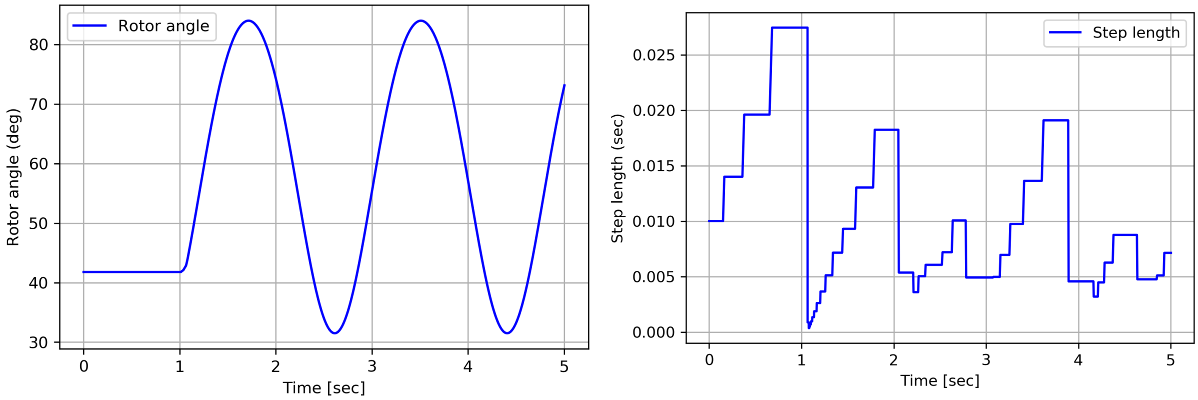
$$\dot{y}_{\nu+1} = f(y_\nu) \tag{33}$$

$$\ddot{y}_{\nu+1} = \frac{\partial f}{\partial y}\dot{y}_{\nu+1} \tag{34}$$

where eq. (34) uses the estimate from eq. (33) to estimate the second order derivative. These are the equations discussed in section 3.3. The idea is that eqs. (33) and (34) are able to provide better approximations for the derivatives than what the corrector iterations do on their own following a disturbance.

5.1.1 Simulation results

Figure 6 shows the simulation results when strategy 1 is in place, i.e. letting the corrector iterations handle the step length adjustment on their own. In fig. 6, one can observe how the step length drops when the short circuit fault occurs. When applying strategy 2, i.e. the re-initialization strategy discussed above, the following graphs are produced:



(a) Tracking of δ (b) Step length

Figure 18: Short circuit simulation. Strategy 2 applied. $t \in [0, 5]$

While it may be hard to observe any noticeable differences between figs. 6 and 18, the detailed comparison of the two simulations reveal interesting observations. The step length doesn't drop as far. The total number of step length adjustments are reduced by 2, and the number of steps taken goes down.

The differences between the simulations with strategy 1 and strategy 2 are summarized in the table below.

Table 7: Re-initialized vs. not re-initialized for power system, $t \in [0, 5]$

Strategy	Steps taken	Step incr.	Step red.	Total adj.	h_{min} [ms]	h_{avg} [ms]
Strategy 1	773	31	8	39	0.127	6.474
Strategy 2	731	29	8	37	0.346	6.848

The results presented in table 7 reveal some interesting effects of re-initialization. Approximately 5.5 % fewer steps are taken, and the total number of step length adjustments is approximately 5% lower.

5.2 Saturation of a state variable

The implementation described in section 4.2 will be used for simulations in this section. Functionality to include saturation effects is added, and the block diagram for the model is as shown in fig. 19 below. The max/min values for the output signal of the amplifier Δu_{df} are set to $\pm$ 3.5.

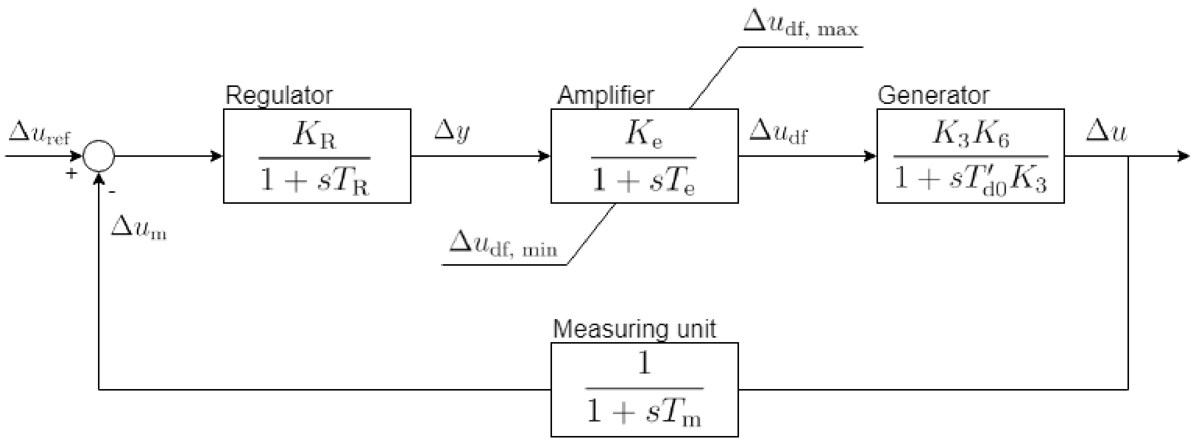
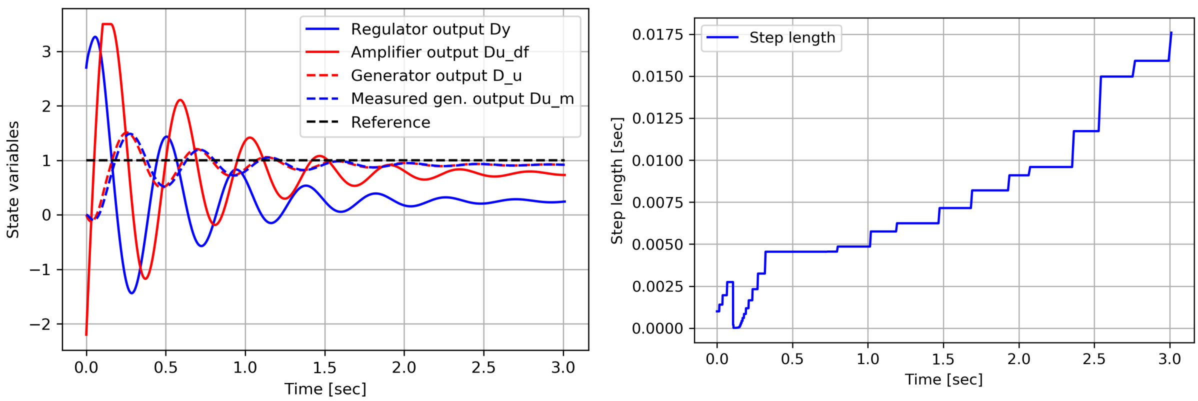


Figure 19: Voltage regulator w/ saturation and generator block diagram

All simulations in this section will be performed with a simulation period of 3.0 seconds.

5.2.1 Simulation results

Applying strategy 1, i.e. letting the corrector iterations handle the disturbance, the following graphs are produced:



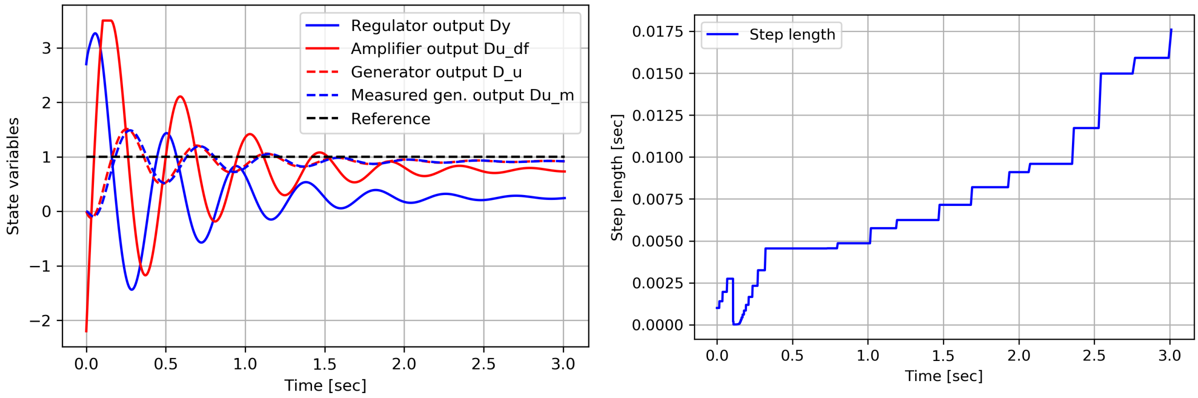
(a) Tracking of y

(b) Step length

Figure 20: Voltage regulator system simulation. Strategy 1 applied. $t \in [0, 3]$

The step length immediately drops to $h = 15$ μs and stays there throughout the period of time that the amplifier is saturated. Once the saturation period is over, the step length rises again and enters a rising trend similar to the behaviour observed in fig. 15. The simulation results presented in fig. 15 are comparable because they have the same strategy for constraining the step length adaption implemented.

By applying re-initialization strategy 2, i.e. implementing eq. (22), the following graphs are produced:



(a) Tracking of y

(b) Step length

Figure 21: Voltage regulator system simulation. Strategy 2 applied. $t \in [0, 3]$

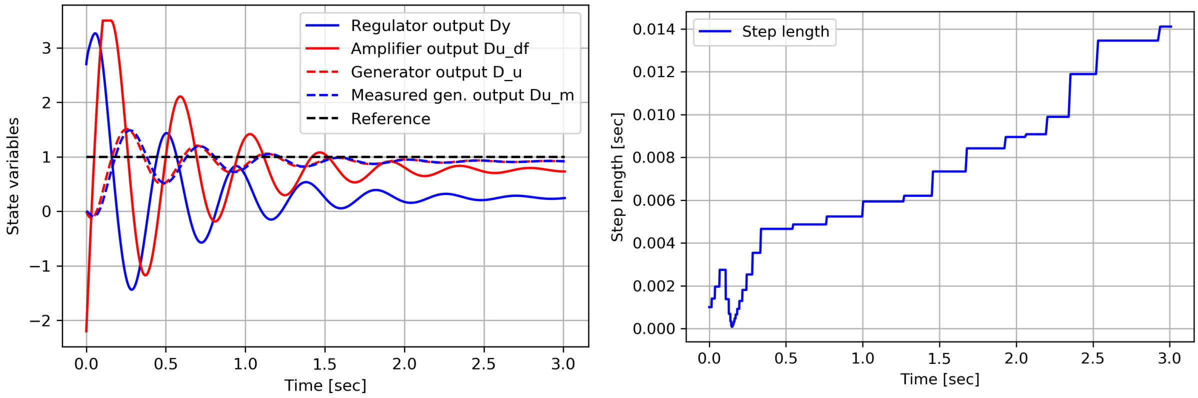
The step length in fig. 21b drops to $h = 15\ \mu$s in this case as well, and the step length behaviour appears very similar. By quantitatively comparing the step length adaption between the cases presented in figs. 20 and 21, the following table can be made:

Table 8: Re-initialized vs. not re-initialized for power system, $t \in [0, 5]$

Strategy	Steps taken	Step incr.	Step red.	Total adj.	h_{min} [ms]	h_{avg} [ms]
Strategy 1	2521	72	4	76	0.015	1.194
Strategy 2	2521	72	4	76	0.015	1.194

As table 8 shows, which of the two strategies are used seem to have no effect on the step length adaption. All performance metrics are equal, and the performance of the two strategies is thus identical.

A measure that may mitigate this is to apply point no. 4 of the step length strategy discussed in appendix A.2, i.e. never let the step length be reduced by more than half of the previous value. This may help avoid excessive step length, but should be used with caution as some events may warrant a quick large step length reduction. When $h_{\nu+1} \geq 0.5 h_\nu$ for step length reduction is applied, and using strategy 1 for re-initialization, the following graphs are produced:

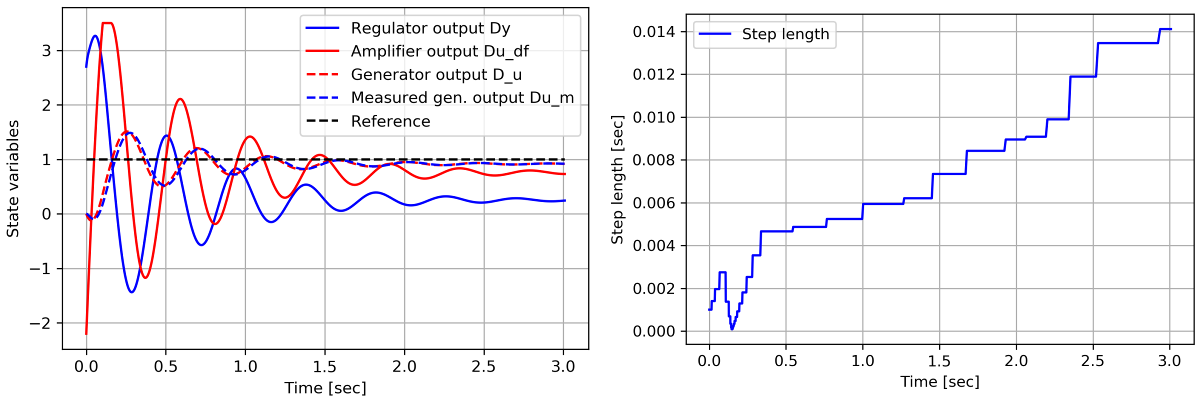


(a) Tracking of y (b) Step length

Figure 22: Voltage regulator system simulation. Strategy 1 applied. h not allowed to reduce by more than half. $t \in [0, 3]$

As fig. 22b shows, the step length is reduced to half of its pre-saturation value when the saturation is applied. It is further reduced by halving in subsequent steps. h reaches a minimum of $h = 86$ μs here as well.

When applying the same rule of $h_{\nu+1} \geq 0.5h_\nu$ in the case of a step length reduction and using strategy 2 for re-initialization, the following graphs are produced:



(a) Tracking of y (b) Step length

Figure 23: Voltage regulator system simulation. Strategy 2 applied. h not allowed to reduced by more than half. $t \in [0, 3]$

Comparing the results from the simulations shown in figs. 22 and 23, the following table can be made:

Table 9: Re-initialized vs. not re-initialized for power system. h not allowed to reduce by more than half. $t \in [0, 5]$

Strategy	Steps taken	Step incr.	Step red.	Total adj.	h_{min} [ms]	h_{avg} [ms]
Strategy 1	726	28	5	33	0.086	4.139
Strategy 2	726	28	5	33	0.086	4.139

As table 9 shows, the two strategies produce identical results. Both the pair of figs. 20 and 21 and the pair of figs. 22 and 23 are thus identical.

However, when comparing the data from table 9 with table 8, one can observe that applying the rule of $h_{\nu+1} \geq 0.5h_\nu$ in the case of a reduction produces more favourable results in terms of total number of step length adjustments, minimum step length and average step length. In other words, the $h_{\nu+1} \geq 0.5h_\nu$ rule improves performance in all three pre-defined performance metrics.

The only difference between re-initialization strategy 1 and 2 is observed when one investigates the first- and second order derivatives of the state variables. In a single step, strategy 1 and 2 produce different values for the derivatives. In strategy 2, this is overridden in the next step when the corrector iterations take over again and the results from strategy 1 and 2 are identical from there on out. More on this observation in section 6.2.

6 Discussion

This section aims to state some of the most important observations from sections 4 and 5 and discuss the simulation results further. The observations that are valid in general are discussed first, before the observations specific to each case are addressed.

The strategy for constraining the step length discussed in appendix A.2 results in a higher number of steps taken in both the short circuit- and the voltage regulator scenario. The number of steps is inversely proportional to the average step length, which means that applying the strategy leads to a lower average step length. The drawback is that more steps require more arithmetic operations to be performed. On the other hand, a shorter average step length means that the algorithm can detect any additional disturbances quicker and adapt to them in a shorter amount of time.

The total number of step length adjustments is reduced when applying vs. not applying the strategy. In the short circuit scenario, the number of step length adjustments is 4-5 times lower with the strategy applied. In the slightly more complex voltage regulator case, the number of step length adjustments is 10 times lower. This reduction in number of step length adjustments is hypothesized to be beneficial enough to justify the higher number of steps needed, especially for large systems. Further work is needed to confirm this.

When the system is modelled by n differential equations and m algebraic equations, the correction matrix is of dimension $(n + m) \times (n + m)$. Thus, the size of the correction matrix grows quadratically with the number of equations involved. The work involved in rebuilding it is therefore expected to increase quadratically as well, which further emphasizes the importance of minimizing the number of step length adjustments.

6.1 Short circuit case

Some inconsistent observations can be made by comparing fig. 6 with fig. 7, and fig. 9 with fig. 10. In the cases where the step length constraining strategy is not applied, the rotor angle δ varies between 37.98° and 74.27°. In the cases where the strategy is applied, the rotor angle δ varies between 31.05° and 84.74°. The peak-to-peak value varies by 17.40° between the two cases.

A difference in peak-to-peak value is present also between figs. 11 and 12 in the damped simulation. In the case where the step length constraining strategy is not applied, the rotor angle δ increases to 67.59° in the first local maximum before it decreases to 47.07° in the first local minimum after the short circuit event. The peak-to-peak value is 20.52° in this case. In the case where the step length strategy is applied, the rotor angle δ increases to 74.77° in the first local maximum before it decreases to 43.46°. The peak-to-peak value is 31.31° in this case, 10.79° larger than the peak-to-peak value of the unconstrained, damped case.

The only differences in the implementation between the comparable pairs of cases are as follows:

No strategy applied		Strategy applied
h can change every step	–	h can change every 15 steps
h can increase as much as it wants as long as it doesn't exceed $h_{\max}$	–	h can increase to no more than twice its previous value while staying below the same $h_{\max}$

They have identical minimum and maximum step length, and the equations that model the system are identical. It's clear that the step length has some influence on the result. Further testing of this observation is done in appendix B.

6.2 Voltage regulator case

The simulation results in the saturated case are identical for both strategy 1 and strategy 2. No difference is observed in the state variable tracking or the step length behaviour. The only measure that improves performance with regards to the performance metrics as defined in section 3.4, is implementing a constraint on how much the step length is allowed to be reduced from one step to the next. This functionality is described in appendix A.2, pt. 4. It consists of implementing the relation $h_{\nu+1} \geq 0.5h_{\nu}$ between the step length in the current step and the step length in the next step.

When implementing said functionality, the number of steps taken to perform the simulation is reduced from 2521 to 726, a reduction by a factor of 3.47, implying that the average step length is 3.47 times higher. The total number of step adjustments is more than halved, and the minimum step length is 5.73 times higher.

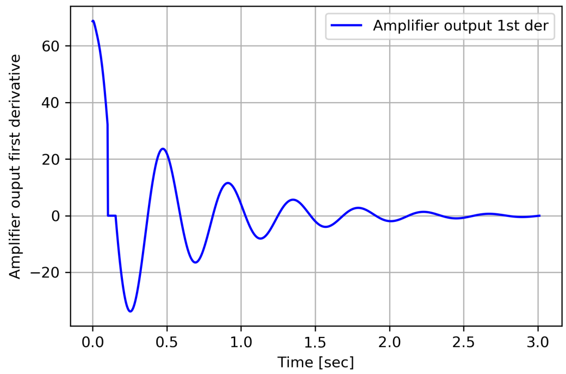
6.2.1 Comparison of derivatives

Because the tracking of the state variables is identical when simulating using strategy 1 and 2, it is interesting to investigate the estimation of the first- and second order derivatives in both cases to see if there are any differences. Because the saturation occurs close to the peak of the output of the amplifier, it is not expected that the first order derivative experiences a large jump in value, as it is expected to be close to 0 at the time of saturation. However, the second order derivative is expected to experience a large jump, as it is expected to be close to a local extreme point when the first order derivative is close to 0.

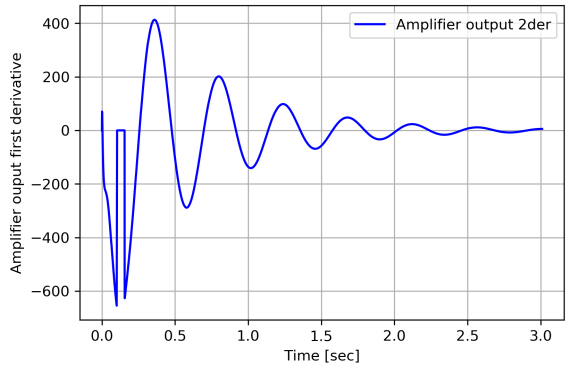
When comparing the first and second order derivatives as estimated by strategy 1 and 2, one can observe that the estimation of the first order derivatives differs slightly between the two strategies. The estimation of the second order derivatives differs greatly. The estimation of the first- and second order derivatives in the step following the disturbance is presented for both strategies in table 10 below.

Table 10: Estimation of derivatives after disturbance for strategy 1 and 2

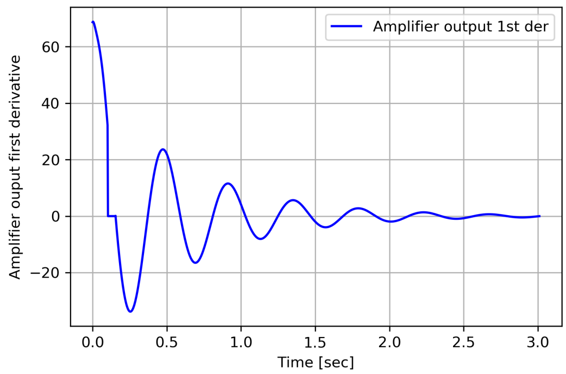
Strategy	1st der. estimate	2nd der. estimate
Strategy 1	-0.00702	-32.8747
Strategy 2	0.09240	-632.684



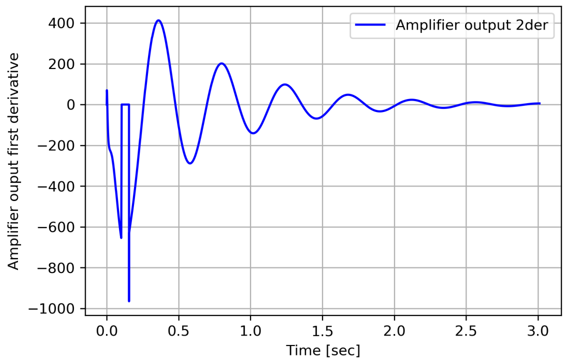
(a) First order derivative, strategy 1.



(b) Second order derivative, strategy 1.



(c) First order derivative, strategy 2



(d) Second order derivative, strategy 2

Figure 24: First and second order derivatives. Comparison of strategy 1 and 2.

As the data in table 10 show, the second order derivative estimate differs greatly between the two strategies. Further investigation of the derivative estimates show that this is only the case for the step immediately following the saturation period. The graphical representation of the derivatives in the two strategies is presented in fig. 24 above.

As can be seen from fig. 24, the second order derivative drops even further after reinitialization with strategy 2. It quickly corrects itself and, after the brief drop, strategy 1 and strategy 2 produce similar results. This mismatch in estimation of variables between the two strategies has no effect on the number of steps taken, the minimum step length or the number of step length adjustments, as shown in tables 8 and 9.

7 Conclusion

In this thesis, the second order predictor-corrector method Gear's method has been implemented in the Python programming language and extensively described in the appendix. The method has been tested on two power systems, one consisting of a synchronous generator connected to an infinite bus via two parallel transmission lines and one consisting of a voltage regulator, amplifier, generator and voltage measuring unit.

In section 4, both systems were simulated on different time scales and many observations related to the step length adjustment strategy were made and discussed. The step length adjustment strategy, as described in appendix A.2, was proven to reduce the number of step length adjustments at the cost of increasing the total number of steps required in both systems tested. This is advantageous, as the computational cost of updating the step length is higher than taking a step. Further work is needed to confirm this for larger systems.

When disturbances were imposed on the systems, the step length was shown to drop to a very low value. While this is necessary to simulate with desired accuracy, the step length spends some time re-adjusting. For this reason, two strategies were applied to reduce the number of step length adjustments and increase the average and minimum step length. They were re-initialization of the state variables and their derivatives, and letting the corrector iterations handle it on their own. The re-initialization strategy improved performance by approximately 5 % in the short circuit case. In the saturation case, the two strategies yielded identical results with the only difference being the estimation of the second order derivative after the saturation. This had no implications on the step length adaption, however.

While the re-initialization and the corrector iterations produced the same result with regards to the step length for the voltage regulator/amplifier system, imposing an additional constraint on the step length adaption does improve performance. By not allowing h to be reduced to more than half of its previous value, the number of step length adjustments was reduced by a factor of 2.30 and average step length was increased by a factor of 3.47 while maintaining desired accuracy.

In the specialization project, the algorithm had some problems determining whether or not the state variables were in steady state following a disturbance. No such observation was made in this thesis. The algorithm is able to detect steady state and increase the step length accordingly, even after a disturbance.

Some inconsistent observations were made in the short circuit case. Depending on the step length, the estimation of the state variables differed. This was further explored by simulating the system with several different step lengths and step length adjustment strategies, but no clear cause was established. This is a topic well suited for further work. In the voltage regulator/amplifier case, no inconsistensies were observed.

All in all, the results show that Gear's method is an elegant method well suited to simulate systems with different time constants. The method is capable of detecting both steady state and disturbances on its own, and quickly adjust the step length accordingly.

8 Further work

As previously stated, the work done in this thesis will be continued on in a master's thesis next year. The further work proposed here is therefore assumed to be a natural continuation for further exploring Gear's method and its strengths and weaknesses.

Several aspects are interesting to explore further. The recommended further work is listed in the following points:

1. The biggest challenge is the re-initialization of the state variables and their derivatives after a disturbance. This is especially interesting in the case of a saturated control system, as the re-initialization strategy tested on such a control system in this thesis was not proved effective. The re-initialization strategy should be tested on other systems, both of equivalent and higher complexity to further explore its effects.

2. Functionality should be added to the step length strategy to accommodate for planned disturbances. Until this point, all simulations have involved the algorithm detecting the disturbance on its own and reducing the step length accordingly. An aspect of a potential re-initialization strategy could therefore be to adapt the step length ahead of the disturbance.

3. In the literature review, it was discovered that there exists several software applications that utilize dynamic simulation based on implicit methods with adaptive step length. The MATLAB function `ode15s` claims to be able to use a method known as Gear's method. This function should be explored in depth and compared with the implementation of Gear's method as described here in appendix A.

4. Extend the implementation of Gear's method to be able to simulate larger, more complex power systems. A natural starting point could be a higher order model of one or more synchronous machines connected to a load via a transmission network consisting of one more more lines and buses. The goal here should be to verify the effectiveness of the step length adjusting strategy and tune it for larger systems.

5. The inconsistent observations made in section 4.1 should be investigated further and tested on arbitrary comparable short circuited systems. The goal here should be to determine the connection between step length and state estimation, and find the cause for the inconsistent observations.

With the above points, it should be possible to produce a code that functions similar to the many ODE solvers available in software in the sense that it takes in an arbitrarily large system of DAEs, solves them in the specified time interval, has some capabilities related to handling disturbances and produces plots similar to the ones presented in this thesis.

References

[1] G. D. Byrne and A. C. Hindmarsh. "A Polyalgorithm for the Numerical Solution of Ordinary Differential Equations". In: *ACM Trans. Math. Softw.* 1.1 (Mar. 1975), pp. 71–96. ISSN: 0098-3500. DOI: 10.1145/355626.355636. URL: http://doi.acm.org/10.1145/355626.355636.

[2] Gülnur Çelik Kızılkan and Kemal Aydin. "Step Size Strategies Based On Error Analysis For The Linear Systems". In: *SDU Journal of Science* 6 (Dec. 2011), pp. 149–159.

[3] DIgSILENT. URL: https://www.digsilent.de/en/powerfactory.html (visited on 05/14/2019).

[4] DIgSILENT. URL: https://www.digsilent.de/en/stability-analysis.html (visited on 05/14/2019).

[5] E. Egeland and J. T. Gravdahl. *Modeling and Simulation for Automatic Control.* Marine Cybernetics AS. Trondheim, Norway: Tapir Trykkeri, 2002. ISBN: 82-92356-01-0.

[6] H. Eidsvik. *Dynamic Simulation Based on a Second-Order Predictor-Corrector Scheme.* Specialization project report. Norwegian University of Science and Technology, NTNU. December. Unpublished. 2018.

[7] O. B. Fosso. "Implementation of an implicit method for numerical integration of differential equations with self-adaptive time-steps (GEAR's method)". In: (1992). DOI: 10.13140/RG.2.2.33826.43205.

[8] O. B. Fosso. *Lecture slides from the course Power System Stability and Control.*

[9] Germund G. Dahlquist. "A special stability property for linear multistep methods". In: *BIT* 3 (Mar. 1963), pp. 27–. DOI: 10.1007/BF01963532.

[10] B. Gear. "Backward differentiation formulas". In: *Scholarpedia* 2.8 (2007). revision #91024, pp. 31–62. DOI: 10.4249/scholarpedia.3162.

[11] P. Kundur. *Power System Stability and Control.* 1st edition. McGraw-Hill Education, 1994. ISBN: 978-0070359581.

[12] J. D. Lambert. *Numerical Methods for Ordinary Differential Systems: The Initial Value Problem.* New York, NY, USA: John Wiley & Sons, Inc., 1991. ISBN: 0-471-92990-5.

[13] N. Liu et al. "Energy-Sharing Model With Price-Based Demand Response for Microgrids of Peer-to-Peer Prosumers". In: *IEEE Transactions on Power Systems* 32.5 (Sept. 2017), pp. 3569–3583. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2017.2649558.

[14] Y. Liu and K. Sun. *Solving Power System Differential Algebraic Equations Using Differential Transformation.* Manuscript submitted for publication. Mar. 2019.

[15] DIgSILENT Pacific Pty Ltd. *Power System Model Guidelines AEMO - Comments to Draft Report.* Apr. 2018.

[16] J. Machowski, J. W. Bialek, and J. R. Bumby. *Power System Dynamics: Stability and Control.* 2nd edition. John Wiley & Sons, 2008. ISBN: 978-0-470-72558-0.

[17] Mathworks. URL: https://it.mathworks.com/help/matlab/math/choose-an-ode-solver.html (visited on 05/15/2019).

[18] Mathworks. URL: https://it.mathworks.com/help/matlab/ref/ode15s.html (visited on 05/15/2019).

[19] M. Stubbe et al. "STAG - a new unified software program for the study of the dynamic behaviour of electrical power systems". In: *IEEE Transactions on Power Systems* 4.1 (Feb. 1989), pp. 129–138. ISSN: 0885-8950. DOI: 10.1109/59.32470.

[20] E. Süli. *Numerical Solution of Ordinary Differential Equations*. Educational note. 2014.

A Gear's method

The following sections of the appendix describe the algorithm behind Gear's method and how to implement it in-depth. The thorough description is taken from [6, 7, 19], and is included here for the sake of completeness and the overall integrity of this master's thesis as a standalone work.

Gear's method is a second order predictor-corrector numerical integration method utilizing self-adaptive step lengths for DAE systems simultaneously, meaning that the differential and algebraic equations are solved simultaneously in each step as opposed to partitioning the system matrix into differential and algebraic equations and solving them separately.

Gear's method is said to be of second order because the derivatives of y up to and including the second order is calculated. Each iteration of the algorithm involves calculating the vector of state variables y, the vector of the first order derivatives of the state variables $\dot{y}$ and the vector of the second order derivatives of the state variables $\ddot{y}$.

It is said to be a predictor-corrector method because in each iteration of the algorithm, the state variables and their first and second order derivatives are first predicted in an explicit calculation based on the step length and values obtained in the previous iteration. This explicit prediction calculation is referred to as the *predictor iteration.* After the predictor iteration has been performed, the predicted values of y, $\dot{y}$ and $\ddot{y}$ are then fed into the corrector in what is referred to as the *corrector iteration.* In the corrector iteration, an approximation of the deviation Δy between the predicted and the actual values is calculated iteratively through the Newton Raphson method. Once the Newton Raphson iterations converge the state variables and their derivatives are updated, thus concluding the corrector iteration.

The corrector iteration yields approximate values for the state variables and their derivatives. This approximation is naturally subject to some error, and the truncation error is estimated after the corrector iteration. Truncation error is defined as the error between estimated and actual values when assuming that no error was present in the previous step. If the truncation error for one or more state variables is deemed to large, i.e. larger than a predetermined maximum, the result of the corrector iteration is rejected and the step size is reduced to maintain desired accuracy. The corrector iteration is then performed again with the new step length. If the estimated truncation error is well within the permitted maximum, the step length is increased to an appropriate value.

The step length adjusts itself based on the magnitude of the truncation error. Experience from implementation shows that the step length is correlated with the magnitude of the first order derivatives of the state variables, i.e. the step length is small when the system is in a state of high rate of change and vice versa. The step length will increase to its maximum value as steady state is approached. Should a disturbance occur during a part of the simulation where the step length is high, the truncation error will be high enough to warrant a reduction of the step length. The step length is then reduced, and the simulation is re-done starting from the step before the disturbance.

A flowchart of the algorithm is presented below.

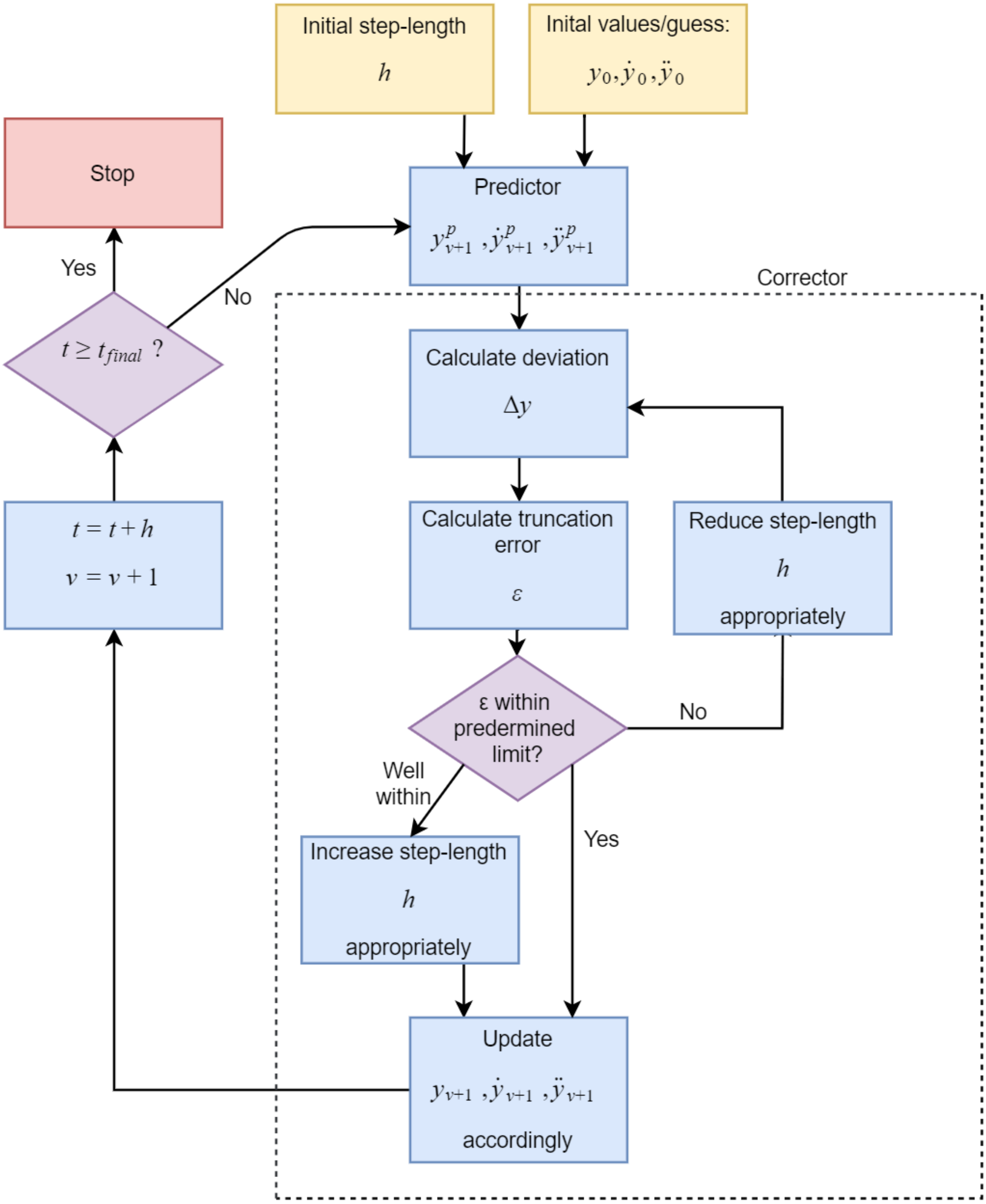


Figure 25: Flowchart of the algorithm behind Gear's method

As can be seen from fig. 25, the corrector iteration is able to iterate several times in the same step. It will not allow the algorithm to proceed to the next step before an acceptable truncation error has been achieved to ensure sufficiently accurate tracking of the state variables.

A.1 Quantitative description

The equation system to solve is given on the following form:

$$\dot{y} = f(y,t) \tag{35}$$

$$0 = g(y,t) \tag{36}$$

y	Vector of state variables (differential and algebraic)
t	Time
$f(y,t)$	Differential equations
$g(y,t)$	Algebraic equations

A.1.1 Predictor iteration

Before the first iteration, an initial step length h_0 is chosen. Initial values of the state variables and their derivatives should be based on a solved load flow case for a given operating scenario. The solution y at time $t_{\nu+1}$ is approximated using step h by Taylor series expansion of second order:

$$y^p_{\nu+1} = y_\nu + h\dot{y}_\nu + \frac{h^2}{2!}\ddot{y}_\nu \tag{37}$$

$$\dot{y}^p_{\nu+1} = \dot{y}_\nu + h\ddot{y}_\nu \tag{38}$$

$$\ddot{y}^p_{\nu+1} = \ddot{y}_\nu \tag{39}$$

Performing the prediction calculation above yields predicted values for y, $\dot{y}$ and $\ddot{y}$ at time $t_{\nu+1}$.

A.1.2 Corrector iteration

It is shown in [1] that the values of y and its derivatives are linked to the predicted values by the following relation:

$$y_{\nu+1} = y^p_{\nu+1} + (y_{\nu+1} - y^p_{\nu+1}) \tag{40}$$

$$h\dot{y}_{\nu+1} = h\dot{y}^p_{\nu+1} + l_1(y_{\nu+1} - y^p_{\nu+1}) \tag{41}$$

$$\frac{h^2}{2!}\ddot{y}_{\nu+1} = \frac{h^2}{2!}\ddot{y}^p_{\nu+1} + l_2(y_{\nu+1} - y^p_{\nu+1}) \tag{42}$$

where the constants l_1 and l_2 only depend on step length. They are defined as follows:

$$l_1 = \frac{2h_{\nu+1} + h_\nu}{h_{\nu+1} + h_\nu} \tag{43}$$

$$l_2 = \frac{h_{\nu+1}}{h_{\nu+1} + h_\nu} \tag{44}$$

The difference between $y_{\nu+1}$ and $y^p_{\nu+1}$ is denoted Δy. To calculate the desired values of the state variables $y_{\nu+1}$, an implicit function that takes the difference between our approximation of the first order derivative and the actual first order derivative from eq. (35) is defined. This function is denoted U and is defined as follows:

$$U(y_{\nu+1}) = \dot{y}_{\nu+1} - f(y_{\nu+1}, t_{\nu+1}) = 0 \tag{45}$$

where $\dot{y}_{\nu+1}$ is the approximation we want to end up with after the corrector iteration. The zeroes of U gives the desired variables for the step $\nu + 1$. By multiplying U with h and using eqs. (35) and (41), this function is rewritten as:

$$U(\Delta y) = h\dot{y}^p_{\nu+1} + l_1\Delta y - hf(\Delta y + y^p_{\nu+1}, t_{\nu+1}) = 0 \tag{46}$$

By performing this redefinition of U, the differential equations $f(y,t)$ can be explicitly calculated using calculated values of Δy and $y^p_{\nu+1}$. The zeroes of this function U gives the required correction vector Δy from step ν to step $\nu + 1$, thus enabling us to calculate the desired state variables for the time $t_{\nu+1}$. The zeroes are found through Newton Raphson iterations.

Once the iterations have converged, the state variables are updated according to eqs. (40) to (42).

A.1.2.1 Corrector implementation

In a general implementation of Gear's method, n differential equations $(f_1, f_2, \dots, f_n)$ and m algebraic equations $(g_1, g_2, \dots, g_m)$ are used to model the system. The differential variables are denoted y_d and the algebraic variables y_a. From eqs. (36) and (46) we have $U(\Delta y) = 0$ and $g(y,t) = 0$. We expand the function U to matrix form as follows:

$$\left[L_1 - h\frac{\partial f}{\partial y}\right][\Delta y] = [-\Delta U] \tag{47}$$

Then, the corrector matrix can be established:

$$\begin{bmatrix} L_1 - h\dfrac{\partial f}{\partial y_d} & -h\dfrac{\partial f}{\partial y_a} \\ \\ \dfrac{\partial g}{\partial y_d} & \dfrac{\partial g}{\partial y_a} \end{bmatrix} \begin{bmatrix} \Delta y_d \\ \\ \Delta y_a \end{bmatrix} = \begin{bmatrix} -\Delta U \\ \\ -\Delta g \end{bmatrix} \tag{48}$$

It has dimension $(n + m) \times (n + m)$. L_1 is a diagonal matrix of the form

$$L_1 = \begin{bmatrix} l_1 & 0 & \dots & 0 \\ 0 & l_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & l_1 \end{bmatrix}_{n\times n} \quad (49)$$

where l_1 is calculated on the basis of h as in eq. (43).

Iterations on eq. (48) are performed until the right hand side of eq. (48) is adequately close to 0, thus fulfilling eqs. (36) and (46). When the iterations stop, the deviations Δy_d and Δy_a are extracted and used to update the state variables.

When all DAEs are linearized, the partial derivatives in the correction matrix are constant and do not change no matter what step we are in. In that case, the correction matrix depends only on the step length h and the constant l_1, which is itself dependant only on h. This implies that when h is updated during or between steps, the corrector matrix must be recalculated. This is very time consuming for large systems, and therefore it is not desirable to have h change more often than necessary. A strategy for this is discussed later in appendix A.2.

A.1.3 Truncation error and step length

Local truncation error ε_t is defined as the error in step $\nu + 1$ when there is no error in step ν. An expression for this truncation error approximation is derived in [19], and it is given as follows:

$$\varepsilon_t = K_q q! l_q \|\Delta y\| \quad (50)$$

q	Order of method (here: $q = 2$)
K_q, l_q	Constants dependant only on q and h
$\|\Delta y\|$	Infinite norm of correction vector Δy

where $\|\Delta y\|$ is given by:

$$\|\Delta y\| = \max_i |y_{\nu+1,i} - y^p_{\nu+1,i}| \quad (51)$$

The error estimation performed in eq. (50) has one of the three following outcomes:

1. **The truncation error exceeds the maximum.** This signals inaccurate tracking of the state trajectory, and the result of the corrector iteration is rejected. The step length is reduced, and the corrector iteration is performed again with the new step length.

2. **The truncation error is well within the maximum.** This signals very accurate tracking. The result of the corrector iteration is accepted, and the step length for the next step is increased to reduce the number of steps and computing operations required to finish the simulation. In this thesis, the term "well within" refers to having a truncation error less than 50 % of the predetermined maximum.

3. **The truncation error is within the maximum.** Here, the truncation falls within 50-100 % of the permitted maximum signalling adequate tracking. The result of the corrector iteration is accepted, and the algorithm keeps the step length for the next step.

To adjust the step length, one can develop an expression for how the truncation error will scale with step size. Depending on the order of the algorithm, a change in step size will give approximate truncation error:

$$\varepsilon = K_q q! l_q \|\Delta y\| \left(\frac{h'}{h}\right)^q \tag{52}$$

In this thesis, we have order $q = 2$ and thus $K_q = K_2$. K_2 is defined as

$$K_2 = \frac{1}{6} \frac{(h_{\nu+1} + h_\nu)^2}{h_{\nu+1}(2h_{\nu+1} + h_\nu)} \tag{53}$$

Thus, the following relationship between maximum and required step size can be established from eqs. (50) and (52):

$$h' = h \left(\frac{\varepsilon_{\max}}{\varepsilon_t}\right)^{\frac{1}{q}} = h \sqrt{\frac{\varepsilon_{\max}}{\varepsilon_t}} \tag{54}$$

$\varepsilon_{\max}$	Predetermined maximum truncation error
ε_t	Approximated truncation error
h	Old step size
h'	New step size

A.2 Considerations on adapting time-steps

Equation (54) provides an easy-to-implement formula for selecting step length. However, in [7] it is observed that this formula needs some safe guarding measures to ensure desired algorithm behaviour. In section 4, the effects of applying vs. not applying such measures is looked at.

It is desired to keep the number of step size adjustments to a minimum. This is due to the fact that the step size is included in the equation system that Newton Raphson is applied to in the corrector iterations, meaning that this system matrix will need to be recalculated for every time the step length is adjusted as shown in eq. (48).

Additionally, the algorithm needs to be able to detect rapid transients occurring even during periods where the derivatives of the state variables are small and the system very close to or in steady state.

The following measures are therefore implemented to step length adjustment:

1. Apply a user defined scaling factor k so that equation (54) becomes

$$h' = k \cdot h \sqrt{\frac{\varepsilon}{\varepsilon_t}} \tag{55}$$

Without this factor k, the step size may be rapidly increased to the maximum permitted value. This increases the likelihood of a rejected solution in the next step due to increased probability for high truncation error. In this thesis, $k = 0.7$ is used in all cases.

2. Select the new step size as follows:

$$h' = \min(h', 2h) \tag{56}$$

This ensures that the step length is never more than doubled from the previous step. The estimate from eq. (55) is used.

3. Specify a maximum permitted step length $h_{\max}$ such that the new step length is selected as follows:

$$h' = \max(h', h_{\max}) \tag{57}$$

where the estimate from eq. (56) is used.

4. When a reduction of the step length is warranted, a maximum reduction by a factor of 2 is implemented to account for situations where a larger reduction than necessary is implied. Note that one should be very careful when applying this aspect of the strategy, as some events *do* warrant a large reduction of the step length quickly in order to stay on track and not diverge. In such a case, stable scenario may be simulated as unstable.

5. Keep the step length fixed for a user defined number of steps. This ensures that recalculation of the system matrix used in the corrector iteration is not necessary for the specified amount of steps.

The effect of these five measures is evident in the simulation results presented in section 4 and discussed in section 6.

B Further testing of inconsistent results in short circuit case

B.1 Constant step length

In the following simulations, the system described in section 4.1 is used. The step length is kept constant to observe what effect different step lengths may have on the peak-to-peak value of the oscillations of δ. The table below summarizes the observations for the undamped case, i.e. $K_D = 0$. All values are rounded to two decimal places and may be subject to some rounding error.

The values δ_{max} correspond to the first local maximum of δ, and the values in column δ_{min} correspond to the first local minimum of δ after the short circuit event. Peak-to-peak is denoted P2P and is calculated as:

$$\text{P2P} = \delta_{\text{max}} - \delta_{\text{min}} \tag{58}$$

Table 11: Max, min and P2P for constant step lengths. Undamped case. $t \in [0, 15]$

h [ms]	δ_{max} [deg]	δ_{min} [deg]	P2P [deg]
1.0	87.87	29.24	58.63
1.5	87.87	29.24	58.63
2.0	87.49	29.45	58.03
2.5	87.11	29.67	57.44
3.0	87.87	29.24	58.63
3.5	88.07	29.13	58.93
4.0	88.26	29.03	59.23
4.5	87.29	29.57	57.73
5.0	89.05	28.59	60.46
5.5	88.26	29.03	59.23
6.0	86.73	29.89	56.84
6.5	89.45	28.38	61.08
7.0	89.45	28.38	61.07
7.5	86.17	30.21	55.95
8.0	88.25	29.04	59.22
8.5	87.09	29.69	57.41
9.0	85.61	30.54	55.08
9.5	87.28	29.58	57.70
10.0	89.04	28.61	60.43

When plotting the data points for δ_{max} and δ_{min} from table 12 in a dual y-axis plot, some interesting characteristics can be observed:

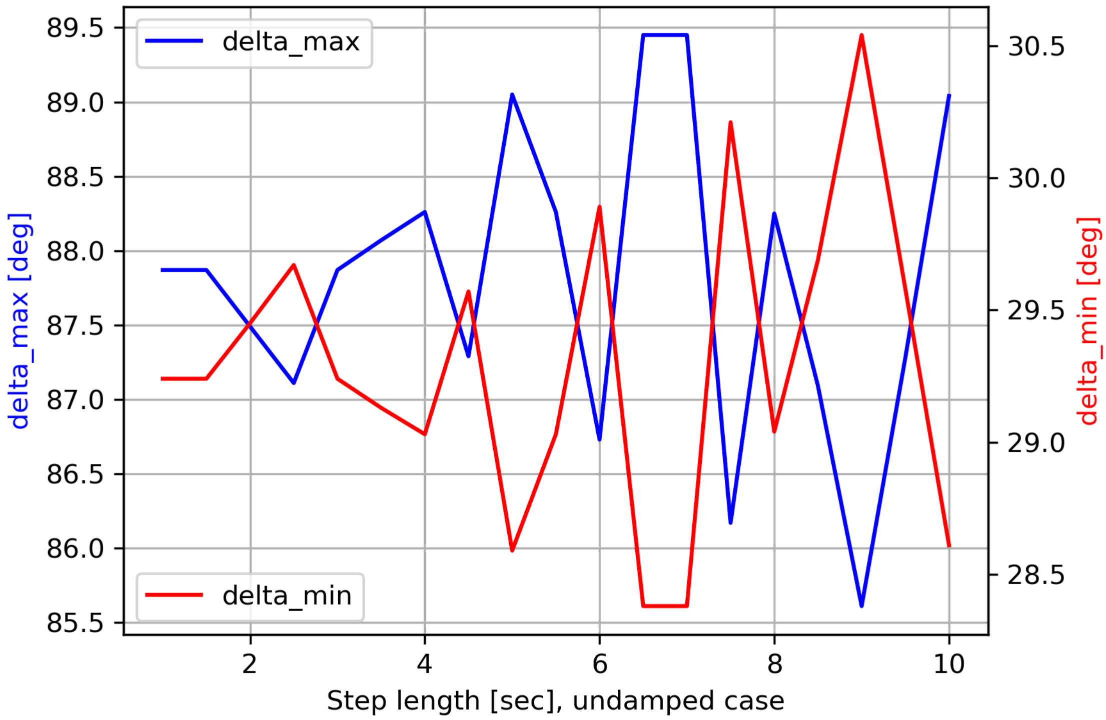


Figure 26: δ_{max} and δ_{min} for different step lengths. Undamped case. $t \in [0, 15]$

Figure 26 shows that the maximum and minimum value appear to increase and decrease symmetrically, with δ_{max} increasing roughly 2 degrees for every degree δ_{max} decreases.

In the undamped case, the same observations can be made. Table 12 below lists the simulation results with the same step lengths as in table 12.

Table 12: Max, min and P2P for constant step lengths. Damped case. $t \in [0, 15]$

h [ms]	δ_{max} [deg]	δ_{min} [deg]	P2P [deg]
1.0	76.81	42.51	34.30
1.5	76.81	42.51	34.30
2.0	76.56	42.63	33.94
2.5	76.31	42.74	33.57
3.0	76.81	42.51	34.30
3.5	76.94	42.46	34.48
4.0	77.07	42.40	34.67
4.5	76.44	42.68	33.75
5.0	77.58	42.17	35.40
5.5	77.07	42.40	34.66
6.0	76.07	42.86	33.21
6.5	77.83	42.06	35.77
7.0	77.83	42.06	35.77
7.5	75.70	43.02	32.67
8.0	77.06	42.40	34.66
8.5	76.31	42.74	33.57
9.0	75.33	43.19	32.14
9.5	76.43	42.69	33.75
10.0	77.57	42.17	35.40

Dual axis plot of the values in table 12:

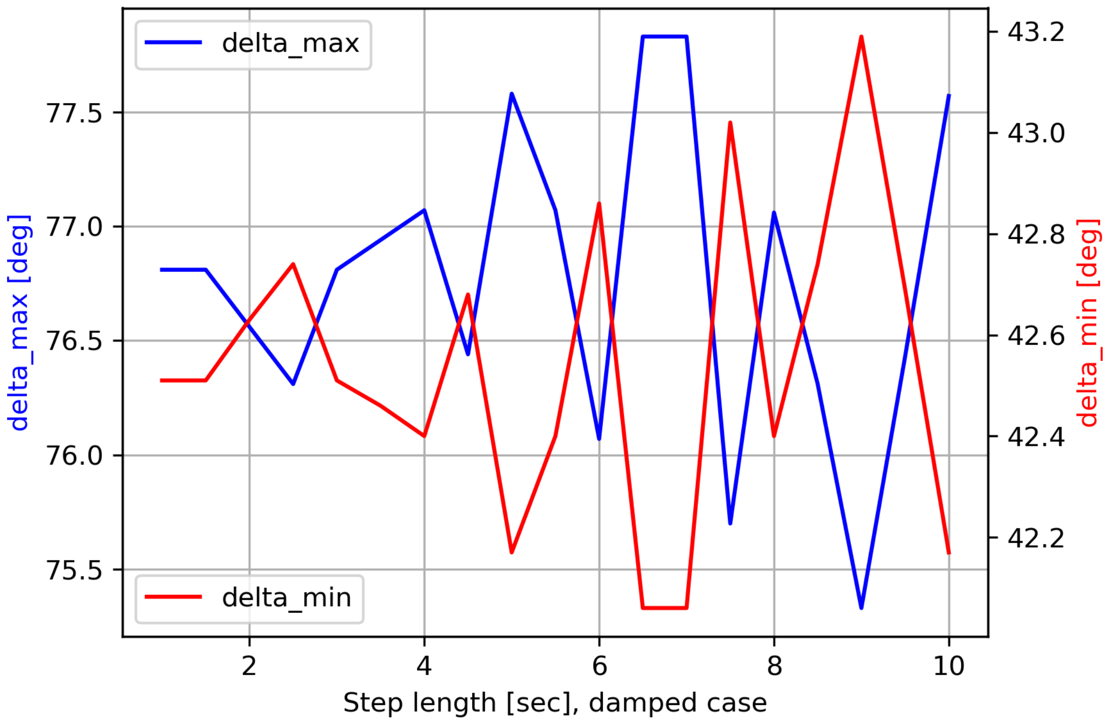


Figure 27: δ_{max} and δ_{min} for different step lengths. Damped case. $t \in [0, 15]$

The shape of $\delta_{\max}$ and $\delta_{\min}$ in fig. 27 is similar to what was observed in fig. 26, with $\delta_{\max}$ increasing a little under 2 degrees for every degree $\delta_{\min}$ decreases.

B.2 Adjusting the step length constraining strategy

In the following simulations, the same system is used. The number of steps in which the step length has to stay constant is adjusted to investigate how that affects max/min values of δ. When conducting the simulations, the following table of values can be produced:

Table 13: Max, min and P2P for different step length strategies. Undamped case. $t \in [0, 15]$

Steps	$\delta_{\max}$ [deg]	$\delta_{\min}$ [deg]	P2P [deg]
1	69.26	41.77	27.49
2	69.52	41.57	27.95
3	116.52	18.72	97.80
4	117.02	18.65	98.36
5	71.30	40.19	31.11
6	94.61	25.77	68.84
7	91.73	27.19	64.54
8	86.87	29.81	57.06
9	84.71	31.07	53.64
10	85.23	30.76	54.46
11	79.29	34.49	44.80
12	85.96	30.33	55.62
13	87.54	29.43	58.11
14	87.60	29.39	58.21
15	86.35	30.11	56.24
16	91.14	27.49	63.66
17	94.36	25.90	68.46
18	86.13	30.24	55.89
19	86.14	30.23	55.91
20	86.15	30.22	55.93

The values in the column labelled "Steps" refer to how many steps h is kept constant for before being allowed to change. Plotting the data reveals the same symmetry characteristic as in figs. 26 and 27:

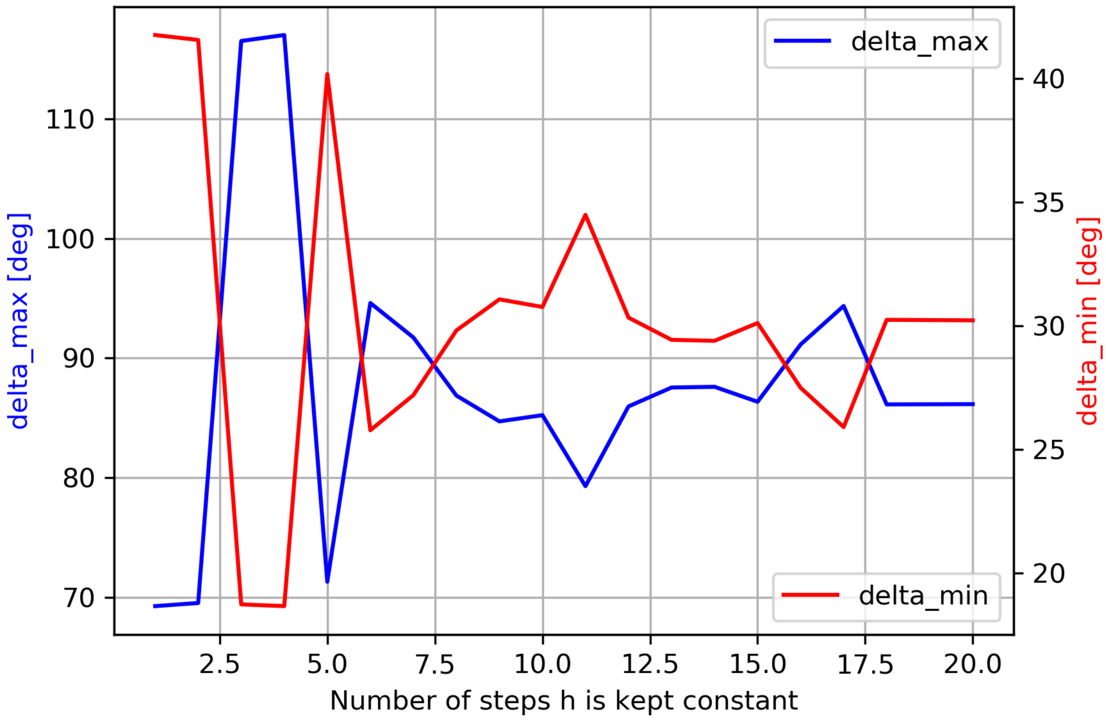


Figure 28: $\delta_{\max}$ and $\delta_{\min}$ for different step length strategies. Undamped case. $t \in [0, 15]$

NTNU
Norwegian University of
Science and Technology