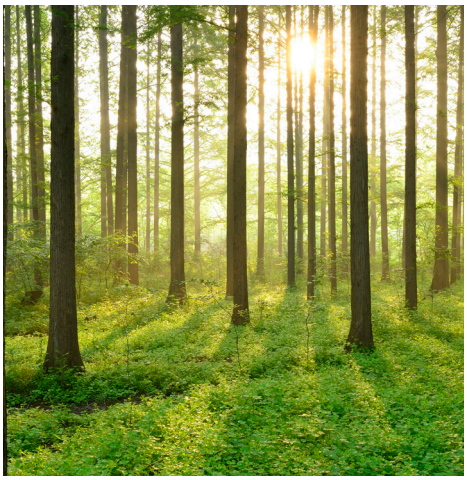
MASKININLÄRNINGSBASERAD REALTIDSSTYRNING AV FÖRNYELSEBARA OCH SÄKRA ELKRAFTSYSTEM

RAPPORT 2022:894



ELNÄTENS DIGITALISERING OCH IT-SÄKERHET

RISK- OCH TILLFÖRLITLIGHETSANALYS

Energiforsk

Maskininlärningsbaserad realtidsstyrning av förnyelsebara och säkra elkraftsystem

HANNES HAGMAR, LE ANH TUAN

ISBN 978-91-7673-894-8| © Energiforsk november 2022

Energiforsk AB | Telefon: 08-677 25 30 | E-post: kontakt@energiforsk.se | www.energiforsk.se

Förord

Denna slutrapport är ett resultat från ett samarbete mellan projekt *Maskininlärnings-baserad realtidsstyrning av förnyelsebara och säkra elkraftsystem* som ingår i programmet *Elnätens digitalisering och IT-säkerhet* och projekt *Utvärdering och osäkerhetsbedömning av maskininlärningsbaserade metoder för realtidsstyrning av förnyelsebara och säkra elkraftsystem* som ingår i programmet Risk- och tillförlitlighetsanalys. Projektet syftar till att ta fram nya och innovativa metoder för realtidsstyrning av elkraftsystem baserade på så kallad förstärkningsinlärning eller ”reinforcement learning”.

Framtidens energisystem, med en större andel varierbar elproduktion, kräver förbättrad övervakning och styrning av elnätet. Metoder baserade på förstärknings-inlärning, eller ”reinforcement learning" (RL), kan användas för att med realtidsmät-data och en mer noggrann modellering av elnätet, optimera styråtgärder, t.ex. genom förbrukningsflexibilitet, för att i realtid ta fram en mer effektiv och säkrare drift av elnätet. Denna databaserade styrning är dock känslig för mät- och modelleringsfel, och variationer inom lastdynamik och flöden kan påverka metodernas noggrannhet. Projekt har utvärderat nya utvecklade RL-metoder, för att undersöka den påverkan som mät- och modellfel har på metodernas möjlighet att styra och avvärja stabilitetsproblem vid större störningar.

Hannes Hagmar har varit projektledare för projektet. Hannes har handletts av Tuan Le, docent vid Elkraftteknik. Båda är från Chalmers Tekniska Högskola.

Energiforsk tackar referensgruppen för deras stöd och uppföljning av projektet. Referensgruppens ledamöter var:

- Massimo Bongiorno, CTH
- Ola Carlson, CTH
- Robert Eriksson, Svenska kraftnät
- Ferruccio Vuinovich, Göteborg Energi
- Per Norberg, Vattenfall Eldistribution
- Mattias Persson, RISE
- Marina Papatriantafilou, CTH

Stort tack också till programstyrelsen för programmet som består av ledamöterna:

- Kristina Nilsson, Ellevio (ordförande)
- Arne Berlin, Vattenfall Eldistribution
- Hampus Bergquist, Svenska kraftnät
- Ferruccio Vuinovich, Göteborg Energi
- Teddy Hjelm/ Per-Olov Lundqvist, Gävle Energi (Elinorr)
- Torbjörn Solver, Mälarenergi (vice ordförande)
- Magnus Sjunnesson, Öresundskraft
- Adam Nilsson/Thorsten Handler/Olle Bergström, Jämtkraft
- Magnus Brodin, Skellefteå Kraft
- Johan Örnberg, Umeå Energi Elnät
- Peter Ols, Tekniska Verken i Linköping
- Jesper Bjärvall, Karlskoga Energi
- Peter Addicksson, HEM
- Malin Wallenberg, VB Energi
- Claes Wedén, Hitachi Energy Sweden
- Katarina Porath, ABB
- Björn Ållebrand, Trafikverket
- Patrik Björnström, Sveriges Ingenjörer (Miljöfonden)
- Matz Tapper, Energiföretagen Sverige (adjungerad)

- Jenny Paulinder, Göteborg Energi
- Emil Welin, Vattenfall
- Magnus Brodin, Skellefteå Kraft
- Mattias Jonsson, Umeå Energi
- Johan Mikkelsen Öresundskraft
- Göran Hamlund/Fredrik Andersson, Elinorr
- Hampus Halvarsson, Jämtkraft Elnät
- Henric Johansson, Jönköping Energi
- Fredrik Byström Sjödin, Installatörsföretagen
- Petter Strandberg/Linus Hansson, Ellevio.se
- Carl Johan Wallnerström, Energimarknadsinspektionen
- Mari Jakobsson/Geoffrey Jordaan, Svenska kraftnät

Energiforsk framför ett stort tack för värdefulla insatser till samtliga intressenter:

Ellevio	HitachiEnergy Sweden	Blåsjön Nät
Vattenfall Eldistribution	ABB	Härjeåns Nät
Svenska kraftnät	Trafikverket	Sandviken Energi Nät
Göteborg Energi	Sveriges ingenjörer (MF)	Sundsvall Elnät
Mälarenergi Elnät	Forumet Swedish Smartgrid	Dala Energi Elnät
Öresundskraft Elnät	Teknikföretagen	Elektra Nät
Tekniska Verken i Linköping	Exeri	Gävle Energi
Skellefteå Kraft Elnät	Evado	Hamra Besparingsskog
Umeå Energi Elnät	Huawei Sverige	Hofors Elverk
Jämtkraft Elnät	Borlänge Energi	Härnösand Elnät
Elinorr ekonomisk förening	Nacka Energi	Ljusdal Elnät
Eskilstuna Strängnäs E&M	Västerbergslagens Elnät	Malungs Elnät
Karlstads El- och Stadsnät	PiteEnergi	Söderhamn Elnät
Borås Elnät	Södra Hallands Kraftförening	Åsele Elnät
Halmstad E&M Nät	Karlskoga Elnät	Årsunda Kraft& Belysn.fören
Luleå Energi Elnät	Bergs Tingslags Elektriska	Övik Energi Nät

Stockholm i oktober 2022
Susanne Stjernfeldt

Sammanfattning

Rapporten utgör slutrapportering av forskningsprojektet *Maskininlärningsbaserad realtidsstyrning av förnyelsebara och säkra elkraftsystem*. Projektet har syftat till att ta fram ett optimeringsbaserat verktyg för realtidsstyrning och kontroll av elsystem. Verktyget är tänkt att användas för att möta de nya utmaningar inom styrning och kontroll som elnätsoperatörer kan komma att möta i ett framtida energisystem som till större del är baserad på förnyelsebar och varierbar elproduktion. Metoderna bygger på djup förstärkningsinlärning, eller deep reinforcement learning (DRL), och har potentialen att i realtid föreslå optimerade styråtgärder till elnätsoperatörer för att försäkra en säker och stabil drift av elsystemet.

I rapporten utvecklas två styralgoritmer; en styralgoritm anpassad för förebyggande styrning, samt en anpassad för mer akuta styrproblem. Metoden för förebyggande styrning är anpassad till att kunna vidta styråtgärder i preventivt syfte för att kunna säkerställa att systemet bibehåller tillräckliga säkerhetsmarginaler så att en säker drift av elnätet kan bibehållas även om större systemstörningar skulle inträffa. Den utvecklade metoden testades på en rad olika drift- och störningsscenarior och visade god prestanda där den utvecklade styralgoritmen lyckades säkerställa tillräckliga säkerhetsmarginaler såväl snabbt som till en låg systemkostnad.

Metoden för akut styrning är i stället anpassad för styrning då ett större systemfel redan har inträffat och systemet behöver styras tillbaka till ett stabilt driftläge. För att stabilisera elnätet kunde styralgoritmen aktivera framtida tänkta stödtjänster från distributionsnät där exempelvis lastefterfrågeflexibilitet och styrning av energilagringssystem utnyttjades. Metoden visade god prestanda på alla de utvecklade testseten och resulterade i en signifikant bättre prestanda än då en mer konventionell styrning baserad på regelstyrd lastfrånkoppling användes. Studierna har även analyserat robustheten i de utvecklade styrsystemen där exempelvis osäkerheter i form av mätfel eller osedda driftpunkter och hur de påverkar de utvecklade styrsystemet har utvärderats.

Slutligen har även projektet undersökt möjligheterna för DRL-metoder att anpassas för att även fungera för olika styrområden på en distributionsnätsnivå. Aspekter som tillgång och kvalitet av mätdata har diskuterats och två fallstudier har analyserats mer djupgående.

Nyckelord

Förstärkningsinlärning, maskininlärning, optimerad styrning, systemtjänster, spänningsstabilitet.

Summary

This report is the final reporting of the research project called *Maskininlärningsbaserad realtidsstyrning av förnyelsebara och säkra elkraftsystem.* The project has aimed to develop an optimization-based tool for real-time control of an electric power system. The tool is intended to be used to face the new challenges that system operators may face in a future energy system that in a higher degree is based on renewable and intermittent energy generation. The methods are based on deep reinforcement learning (DRL) and have the potential to, in real-time, suggest optimized control actions to system operators to ensure a stable and secure operation of the power system.

Two different control algorithms are developed in the project; one aimed for preventive control, the other aimed for emergency control. The method for preventive control is adapted to take actions to ensure a secure operation of the system even if a larger disturbance would occur. The developed method was tested on a range of different load and disturbance scenarios and showed good performance on all the developed test sets and the developed control achieved to make the system secure to a low system cost in a single time step for almost all scenarios.

The method for emergency control is adapted for cases when a larger disturbance has already occurred and when emergency control actions are required to control the system back into a stable operation again. The controllable actions included control of future system services such as demand response and/or energy storage systems. The method showed good performance on all the developed test sets and resulted in significantly better performance than when a more conventional control that was based on a rule-based load shedding scheme was used. The studies have also analyzed the robustness in the developed control methods where, for instance, the impact of measurement errors or unseen operating conditions and their impact on the performance were evaluated.

Finally, an evaluation of how DRL-methods can be adapted to also function on other control areas in a distribution level have been performed. Aspects such as the availability and the quality of measurement data have been discussed and two different case studies and how they can be adapted for RL control have been analyzed in depth.

Innehåll

1 Introduktion 10

1.1 Bakgrund 10

1.2 Syfte och mål 11

1.3 Rapportöversikt 11

2 Teori och metoder inom djup förstärkningsinlärning 13

2.1 Markovianska beslutsprocesser 14

2.2 Grundläggande förstärkningsinlärning 15

2.3 Djup förstärkningsinlärning (DRL) 16

2.4 Policy-gradient och Actor-Critic metoder 16

2.5 Proximal Policy Optimization-algoritmen 18

3 Översikt kring DRL-baserade metoder för styrning av elsystem 20

3.1 Förebyggande styrning 20

3.2 Akut styrning 21

3.3 Test- och träningssystem 22

4 Förebyggande styrning baserad på DRL 24

4.1 Anpassning för hybridstyrning av kontinuerliga och diskreta handlingar 24

4.2 Metod och datagenerering 25

4.2.1 Beräkning av säkerhetsmarginaler 25

4.2.2 Tillstånd, handlingar och belöningsdefinitioner 26

4.2.3 Generering av träningsdata 29

4.2.4 Arkitektur för aktör- och kritikernätverk 30

4.2.5 Träning av aktör- och kritikernätverk 32

4.3 Testsets och träningsresultat 33

4.3.1 Framtagning av testset 35

4.3.2 Testresultat 35

4.4 Jämförelse mot styrsystem baserade endast på diskret handlingsrymd 36

4.4.1 Struktur för den regelbaserade styrningen 36

4.4.2 Struktur för den diskreta DRL-styrningen 37

4.4.3 Resultat 38

5 Akut styrning baserad på DRL 40

5.1 Datagenerering och metod 40

5.1.1 Tillstånd 42

5.1.2 Handlingar 43

5.2 Aktörs- och kritikernätverk 44

5.3 Träning av aktörs- och kritikernätverken 46

5.4 Simuleringar och resultat 46

5.4.1 Framtagning av testsets 46

5.4.2 Testresultat 47

5.4.3 Prestanda då aktiveringsgränser är implementerade 50

6 Förutsättningar för användning av DRL på distributionsnätsnivå 52
6.1 Mätinfrastruktur och framtida funktionskrav 52
6.2 Tillgängliga styråtgärder 53
6.3 Fallstudie 1: Styrning av resurser inom efterfrågeflexibilitet 53
6.3.1 Flexibilitetsbehov och styrmål 53
6.3.2 Definition av Markoviansk beslutsprocess för efterfrågeflexibilitet 54
6.3.3 Typ av DRL-algoritm och datagenerering 55
6.3.4 Översikt kring tidigare studier 56
6.4 Fallstudie 2: Spänningsreglering (Volt-Var styrning) 56
6.4.1 Definition av Markoviansk beslutsprocess för spänningsreglering 57
6.4.2 Typ av DRL-algoritm och datagenerering 58
6.4.3 Översikt kring tidigare studier 58
6.5 Utmaningar och praktiska aspekter 59
7 Sammanfattning 61
8 Referenser 62

1 Introduktion

Forskningsprojektet *Maskininlärningsbaserad realtidsstyrning av förnyelsebara och säkra elsystem* har utförts av Chalmers Tekniska Högskola. Projektet har finansierats med medel från dels Energiforsks forskningsprogram Elnätens Digitalisering & IT-säkerhet, dels forskningsprogrammet Risk- och Tillförlitlighetsanalys. Projektet har syftat till att utveckla samt utvärdera nya metoder för avancerad styrning baserade på så kallad ”djup förstärkningsinlärning”, vilket är en klass av maskininlärning anpassad för styrning av komplexa system. Projektet fokuserar på styrning anpassad för att undvika stabilitetsproblem på stamnätsnivå, men styrbara komponenter inkluderar även bland annat framtida systemtjänster från distributionsnät. Därutöver utförs en analys av möjligheterna att anpassa de utvecklade styrmetoderna för användning för styrning inom distributionsnät. Projektet är utfört som en fortsättning på ett tidigare doktorandprojekt som utförts vid Chalmers Tekniska Högskola[1].

1.1 BAKGRUND

Den pågående omställningen till ett förnyelsebart energisystem, med en större andel varierbar elproduktion, kommer att resultera i stora framtida utmaningar för elnätsoperatörer. Bland annat kommer upprätthållandet av kraftbalansen – alltså att den genererade effekten är *exakt* lika stor som belastningsbehovet vid varje ögonblick - att bli alltmer utmanande i en framtid med en större andel förnyelsebar energiproduktion. I det nordiska elsystemet så används reglerkraft (främst bestående av vattenkraft) samt en gemensam elmarknad med närliggande länder som främsta medel för att försäkra att kraftbalansen kan bibehållas. För att fullt ut kunna utnyttja tillgänglig reglerkraft, som ofta är lokaliserad långt från lastcentrumen, kommer det i framtiden att krävas ökad överföringskapacitet för att kunna överföra elektrisk energi från där den finns tillgänglig till där den kommer att användas.

I ett elsystem begränsas dock överföringskapaciteten av en rad stabilitetsrelaterade fenomen, som exempelvis spänning- och frekvensstabilitet. Elsystemet måste klara av att hantera större störningar, som exempelvis bortfall av produktionsanläggningar eller större nätledningar, utan att systemets stabilitet påverkas i för stor grad. Nätoperatörers möjlighet att effektivt styra elsystem för att både försäkra att det är säkert och för att kunna avvärja stabilitetsrelaterade problem är därmed av mycket hög vikt. Därutöver kommer även framtida laster (exempelvis en ökad mängd laddning av elbilar) samt elproduktion (exempelvis ökad andel vind- och solkraft) att i högre grad styras genom kraftelektronik som har en betydligt snabbare dynamisk respons vid störningar jämfört med konventionella laster och generatorer. Detta kan resultera i att dagens metoder för styrning och övervakning av elnäten kan vara för långsamma för att hantera de snabbare dynamiska förloppen som kan förväntas inträffa i framtiden. De

[1] Doktorandprojektet ”*Avancerad visualisering av spänningsstabilitetsgränser och systemskydd baserat på realtidsmätningar*”.

pågående förändringarna i elsystemet kommer även att innebära nya utmaningar inom driften av distributionsnät. Exempelvis kommer distributionsnätsägare att i en större grad behöva hantera problem som bland annat spänningsvariationer orsakade av en ökad mängd lokal förnyelsebar elproduktion eller överbelastningar orsakade av nya typer av laster (exempelvis laddning av elbilar).

Detta projekt syftar till att utveckla nya maskininlärningsmetoder baserade på så kallad djup förstärkningsinlärning, eller "*deep reinforcement learning*" (DRL), med målet att kunna hjälpa elnätsoperatörer att mer effektivt styra och avvärja stabilitetsproblem vid större störningar och därmed kunna bibehålla en hög kapacitet i nätet. DRL tillåter styrmetoderna att baseras på realtidsmätdata och en mer noggrann modellering av elnätet, samtidigt som optimerade handlingar, exempelvis nedstyrning av laster, kan aktiveras i realtid för en mer effektiv och säkrare drift av elnätet.

1.2 SYFTE OCH MÅL

Projektet syftar till att utveckla och utvärdera nya metoder för styrning av elsystem som är baserade på DRL. Projektet fokuserar på hantering av stabilitets- och säkerhetsproblem på en stamnätsnivå, men styrbara komponenter inkluderar även bland annat framtida systemtjänster från distributionsnät. En analys av möjligheterna att anpassa utvecklade metoder för användning även på distributionsnätsnivå utförs även i rapporten.

Projektets mål är följande:

- Utveckla och utvärdera en metod för *förebyggande* styrning som syftar till att vidta handlingar i preventivt syfte för att säkerställa en säker och kontinuerlig drift av elnätet även om större systemstörningar skulle inträffa. Styrbara komponenter inkluderar kapacitiva shuntar samt utnyttjande av framtida tänkta stödtjänster från distributionsnät.
- Utveckla och utvärdera en metod för *akut* styrning som syftar till att optimalt styra systemet tillbaka till ett stabilt driftläge om en större systemstörning redan har inträffat. Styrbara komponenter inkluderar styrning av efterfrågeflexibilitet samt energilagringssystem.
- Utvärdera möjligheterna för DRL-baserade styrmetoder att även användas på distributionsnätsnivå. Specifik kommer möjligheterna att reglera (1) spänning- och reaktiva effektflöden samt (2) styra flexibla resurser som laster och elproduktion att utvärderas.
- Utveckling av metodik för utvärdering av påverkan från diverse osäkerheter, som exempelvis mätfel, på de utvecklade metoderna.

1.3 RAPPORTÖVERSIKT

Rapporten är strukturerad enligt följande. Kapitel 2 ger en övergripande introduktion kring förstärkningsinlärning och de metoder som kommer att användas i rapporten. I kapitel 3 ges en översikt kring stabilitets- och säkerhetsutvärdering på en stamnätsnivå samt ger en presentation kring hur de utvecklade metoderna är tänkta att användas i framtiden. I kapitel 4 presenteras den utvecklade metoden för *förebyggande* styrning samt resultat från de teststudier

som har utförts. I kapitel 5 presenteras den utvecklade metoden för *akut* styrning samt resultat från de teststudier som utförts. I kapitel 6 presenteras en utvärdering kring förutsättningar för förstärkningsinlärningsbaserade metoder att användas på distributionsnätsnivå. Detta analyseras djupare genom två fallstudier. Slutligen presenteras slutsatser från rapporten i kapitel 7.

2 Teori och metoder inom djup förstärkningsinlärning

I följande del ges en översikt av grundläggande teori inom förstärkningsinlärning och de metoder som använts i projektets fallstudier. Teorin presenteras endast övergripande där fokus är att läsaren ska få en tillräcklig bakgrund för att kunna förstå metodernas funktion och uppbyggnad. För ytterligare detaljer kring teori hänvisas läsaren till de referenser som anges senare i kapitlet. Maskininlärning (eng. *machine learning*, ML) kan definieras som en uppsättning metoder och statistiska modeller som används för att utföra uppgifter utan att använda uttryckliga instruktioner och där algoritmen i stället lär sig genom att identifiera mönster och dra slutsatser om data. Förstärkningsinlärning är en klass av maskininlärning som används specifikt för styrning där styralgoritmen tränas genom att iterativt interagera med ett styrobjekt och därmed lär sig att agera optimalt.

Det kanske mest populära valet i litteraturen har varit att kombinera förstärkningsinlärning med djupinlärning baserat på neurala nätverk, vilket då betecknas som ”djup” förstärkningsinlärning (eng. *deep reinforcement learning*, DRL). DRL kombinerar fördelarna med djupinlärning och förstärkningsinlärning och har visat sig effektivt för att lösa många avancerade styrproblem. I rapporten presenteras ingen ingående teori kring djupinlärning och neurala nätverk, utan funktionen diskuteras endast övergripande. En relativt enkel sammanfattning av neurala nätverk och djupinlärning kan fås i [1].

DRL är ett förhållandevis nytt forskningsområde och svenska översättningar på metoder och begrepp är till stor del ännu ej definierade och allmänt vedertagna. För att undvika sammanblandning mellan algoritmer och för att hålla terminologin enkel kommer delvis de engelska förkortningarna att användas i rapporten. Nedan presenteras en kort lista på de begrepp och de förkortningar som kommer användas löpande i rapporten:

Tabell 1. Sammanställning av förkortningar och begrepp som används i rapporten

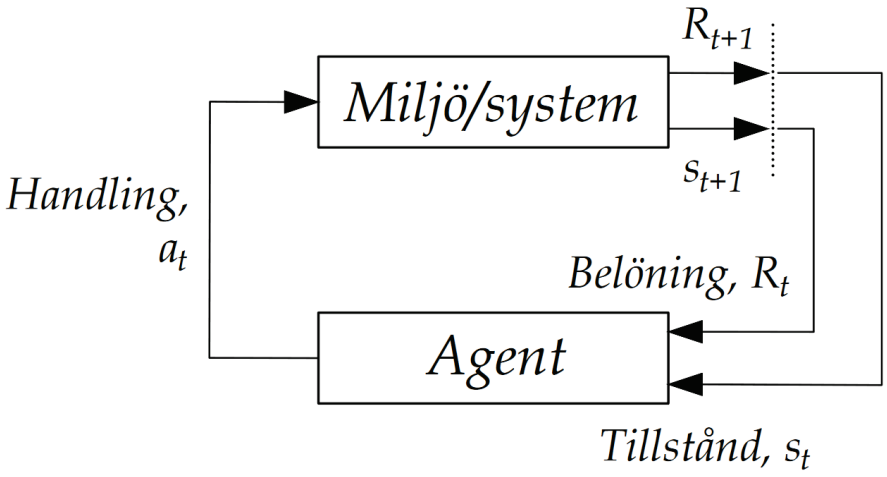
Förkortning	Engelskt begrepp	Svenskt begrepp
MDP	Markov Decision Process	Markoviansk beslutsprocess
AC	Actor-Critic	Aktör-Kritiker
DRL	Deep Reinforcement Learning	Djup förstärkningsinlärning
RL	Reinforcement Learning	Förstärkningsinlärning
PPO	Proximal Policy Optimization	-
NN	Neural network	Neuralt nätverk

2.1 MARKOVIANSKA BESLUTSPROCESSER

Förstärkningsinlärning bygger på så kallade Markovianska beslutsprocesser (MDP:er), vilket är en matematisk beräkningsprocess som används för modellering av sekventiella styrproblem [2]. Dessa används vanligtvis för tidsdiskreta sekventiella problem där utfallet av åtgärderna är delvis stokastiska och delvis påverkas av beslutsfattarens handlingar. Matematiskt definieras en MPD av [3]:

- $s \in \mathcal{S}$: en uppsättning av tillståndsvariabler, s (eng. *states*) som definierar tillståndsrymden, $\mathcal{S}$ (eng. *state space*).
- $a \in \mathcal{A}$: en uppsättning av handlingar, a (eng. *actions*) som definierar handlingsrymden, $\mathcal{A}$ (eng. *action space*).
- $\mathcal{T}$: distribution av dynamik för tillståndsövergångar.
- $\mathcal{R}$: en belöningsfunktion $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$ som definierar den belöning som förväntas vid olika handlingar och tillstånd.
- γ: en diskonteringsfaktor $\gamma \in [0,1]$.

I en MPD kallas styralgoritmen för en ”*agent*”. Systemet som styrs och som agenten interagerar med kallas för ”miljö” (eng. *environment*). Den typiska interaktionen mellan agenten och miljön som används i en MPD och i RL presenteras i Figur 1. Agenten och miljön interagerar i sekvenser av diskreta ($t = 0, 1, 2, 3 \ldots$) tidssteg. Vid varje tidpunkt (t) mottar agenten en representation av miljöns tillstånd (s_t). Beroende på tillståndet och agentens styrpolicy, så väljs olika handlingar (a_t). De tagna handlingarna påverkar i sin tur miljön, vilket skapar ett nytt uppdaterat tillstånd ($s_t \to s_{t+1}$). Handlingarna och dess påverkan på miljön skapar även belöningar (R_{t+1}), vilka agenten generellt försöker maximera genom dess val av nya handlingar [1]. Den kontinuerliga interaktionen mellan agenten och miljön ger upphov till en sekvens av tillstånds-, handlings- och belöningspar. Om det går att definiera ett sista tidssteg i MDP:n brukar den definieras som *episodisk*, medan om MDP:n inte naturligt kan brytas in i identifierbara episoder så kallas den för *icke-episodisk*.



Figur 1. Interaktion mellan en agent och en miljö i en MPD.

Dynamiken i en MDP modelleras genom en distribution av dynamik för tillståndsövergångar: $\mathcal{T}$: $p(s_{t+1} = s'|s_t = s, a_t = a)$, där sannolikheten för en övergång från tillstånd s till nästkommande tillstånd s', då handlingen a har tagits vid tidpunkten t, modelleras. Den *förväntade* belöningen för alla tillstånd- och handlingspar kan modelleras av en belöningsfunktion. Belöningen ger ett kvantitativt mått som används för att förbättra agentens styrpolicy. Handlingar och en styrpolicy sägs vara optimala om de leder till en maximal kumulativ belöning över tid. En viktig egenskap i en MDP är även den så kallade Markov-egenskapen: $p(s_{t+1}|s_1, a_1, \dots, s_t, a_t) = p(s_{t+1}|s_t, a_t)$ som i huvudsak innebär att den betingade sannolikhetsfördelningen för ett framtida tillstånd, givet det *nuvarande* tillståndet och alla historiska tillstånd, är *oberoende* av de historiska tillstånden.

2.2 GRUNDLÄGGANDE FÖRSTÄRKNINGSINLÄRNING

Samtliga RL-algoritmer baseras på MDP:er, men olika klasser av RL-algoritmer gör olika matematiska antaganden. Huvuduppgiften för samtliga algoritmer är dock att lära sig en optimal *styrpolicy* (eng. *policy*), vilket generellt betecknas med det grekiska tecknet π. Styrpolicyn är en funktion som beskriver sambandet mellan ett visst tillstånd och vilken handling som ska tas i det givna tillståndet. En styrpolicy kan antingen vara stokastisk eller deterministisk beroende på utformningen av RL-algoritmen. En stokastisk styrpolicy $\pi(a|s)$ ger en *betingad* sannolikhet att ta en handling i ett givet tillstånd, vilket matematiskt kan formuleras med:

$$\pi(a|s): \quad \mathcal{S} \rightarrow \mathcal{P}(\mathcal{A}) \tag{2-1}$$

där $\mathcal{P}(\mathcal{A})$ representerar sannolikheterna att ta respektive handling i den definierade handlingsrymden $\mathcal{A}$, givet tillståndet s. För en deterministisk styrpolicy så är i stället alla handlingar som tas av styrpolicyn bestämda för varje givet tillstånd:

$$\pi(s): \quad \mathcal{S} \rightarrow \mathcal{A} \tag{2-2}$$

RL-metoder kan delas in i metoder som använder ett modellbaserat och ett modellfritt tillvägagångssätt. Modellbaserad RL modellerar uttryckligen både belöningsfunktionen och distributionen av dynamik för tillståndsövergångar. Modellfria RL-algoritmer förlitar sig i stället uteslutande på träningsdata som genereras från miljön. I rapporten kommer uteslutande modellfria RL-algoritmer att behandlas och modellbaserad RL kommer därmed ej att detaljeras vidare.

En RL-agents mål är att maximera en förväntad kumulativ avkastning, där avkastningen (eng. *return*) betecknas som G_t^λ. I allmänhet diskonteras framtida belöningar och avkastningen formuleras då som:

$$G_t^\lambda = \sum_{k=t}^{T} \lambda^{k-t} R(s_k, a_k) \tag{2-3}$$

Där λ kan variera mellan 0 och 1 beroende på hur mycket framtida belöningar önskas diskonteras. De flesta RL-algoritmer försöker lära sig en approximering av den så kallade värdefunktionen, vilket representerar den *förväntade* avkastningen om agenten startar i ett visst tillstånd och sedan följer den gällande styrpolicyn därefter:

$$V^{\pi}(s) = \mathbb{E}\left[G_t^{\lambda} | s_t = s;\ \pi\right] \tag{2-4}$$

En annan ofta använd funktion inom RL är handlings-värdefunktionen, vilket representerar den förväntade avkastningen när agenten befinner sig i tillstånd s och tar handlingen a, och *därefter* följer den gällande styrpolicyn:

$$Q^{\pi}(s,a) = \mathbb{E}\left[G_t^{\lambda} | s_t = s, a_t = a;\ \pi\right] \tag{2-5}$$

Det finns ett flertal olika RL-algoritmer som är utformade för att lära sig den optimala styrpolicyn på olika sätt. Klassiska exempel inkluderar Q-learning [4], SARSA[5] eller policy-baserade metoder som exempelvis REINFORCE [6] och ”Natural gradients”-metoder [7]. Vilken algoritm som är bäst lämpad är ofta beroende på den definierade miljön och vilken typ av styrning som önskas för det aktuella problemet.

2.3 DJUP FÖRSTÄRKNINGSINLÄRNING (DRL)

Den grundläggande formuleringen av RL är främst tillämplig på lågdimensionella miljöer eller då en kontinuerlig tillståndsrymd kan delas upp i ett begränsat antal diskreta steg. För sådana miljöer är tabellbaserade metoder, där värdefunktionen och handlingsvärdesfunktionen kan läras direkt för varje möjligt tillstånd, effektiva. För de flesta faktiska problem är dock tillståndsrymden för stor (i en kontinuerlig tillståndsrymd är antalet tillstånd oändligt) och tabellbaserade metoder är olämpliga för användning i sådana miljöer. För att hantera dessa problem är det vanligt att använda någon form av parametriserad funktion som kan definiera exempelvis sambandet mellan ett visst tillstånd och vilken handling som ska utföras, eller utföra en approximation av värdefunktionen för ett visst tillstånd. Som tidigare nämnts så har det mest populära valet i litteraturen och forskningen varit att kombinera RL med djupinlärning: vilket då betecknas som ”djup” förstärkningsinlärning (DRL). DRL kombinerar fördelarna med djupinlärning och neurala nätverk med RL och har visat sig effektivt för att lösa många avancerade styrproblem. En enklare sammanfattning kring djupinlärning och neurala nätverk kan fås i [1].

2.4 POLICY-GRADIENT OCH ACTOR-CRITIC METODER

En policy-gradientmetod är en modellfri RL-algoritm som lär sig en parametriserad styrpolicy som kan välja handlingar utan att behöva tillfråga en värdefunktion [1]. Dessa metoder försöker maximera en definierad målfunktion $J(\theta)$, parametriserad av θ, där uppdateringar av parametrarna sker genom gradvisa förbättringar av målfunktionen:

$$\theta_{t+1} = \theta_t + \alpha\widehat{\nabla J(\theta_t)} \tag{2-6}$$

där $\widehat{\nabla J(\theta_t)}$ är ett stokastiskt estimat på gradienten för målfunktionen med avseende på θ_t, och där α är en hyperparameter[2] som styr inlärningshastigheten (eng. *learning rate*). En av de vanligaste utformningarna av gradientestimatet har formen:

$$\widehat{\nabla J(\theta_t)} = \hat{\mathbb{E}}_t\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\, \hat{A}_t\right] \tag{2-7}$$

där $\hat{\mathbb{E}}_t$ representerar ett empiriskt medelvärde av ett antal scenarion som har samlats in från den definierade MDP:n; $\nabla_\theta \log \pi_\theta(a_t|s_t)$ representerar gradienten av den parametriserade styrpolicyn; och $\hat{A}_t$ är ett estimat av den så kallade fördelsfunktionen (eng. *advantage function*) vid tidpunkten t:

$$\hat{A}_t = \hat{Q}_\varphi^\pi(s_t, a_t) - \hat{V}_\varphi^\pi(s_t) \tag{2-8}$$

där $\hat{V}_\varphi^\pi$ och $\hat{Q}_\varphi^\pi$ är estimat av värdefunktionen respektive handlings-värdefunktionen. Handlings-värdefunktionen kan estimeras på flera olika sätt. Ett relativt rättframt sätt är att använda sig av den beräknade avkastningen G_t^γ från ekvation (2-3), vilket ger ett estimat på $\hat{Q}_\varphi^\pi$ utan att något systematiskt fel (eng. *bias*) introduceras. En nackdel med att använda sig av avkastningen för att estimera handlings-värdefunktionen är dock att estimatet tenderar att ha en hög varians, vilket kan leda till att metoden konvergerar långsamt. Ett alternativ för att minska estimatets varians, på bekostnad att ett visst systematiskt fel skapas, är att använda tidsskillnadsmetoder (eng. *temporal difference*). Ett alternativt sätt att formulera fördelsfunktionen baserat på just tidsskillnadsmetoder ges av:

$$\hat{A}_t = \hat{Q}_\varphi^\pi(s_t, a_t) - \hat{V}_\varphi^\pi(s_t) = \tag{2-9}$$

$$= \underbrace{R_t + \gamma \hat{V}_\varphi^\pi(s_{t+1}) - \hat{V}_\varphi^\pi(s_t)}_{\boldsymbol{\delta-error}}$$

där handlings-värdefunktionen ersätts med summan av belöningen R_t som fås efter att handlingen a_t tas i tillstånd s_t samt det estimerade (samt diskonterade) värdet av *nästkommande* tillstånd $\gamma \hat{V}_\varphi^\pi(s_{t+1})$. Eftersom $\hat{V}_\varphi^\pi(s_{t+1})$ är en approximation av den faktiska värdefunktionen så kommer det att bidra till att ett visst systematiskt fel introduceras, dock med fördelen att variansen för estimatet minskar. Används detta uttryck för att beskriva fördelsfunktionen så kallas den generellt för δ-felet eller TD-felet (från engelskans Temporal-Difference error).

Värdefunktionen är generellt okänd och måste läras samtidigt som styrpolicyn. Om värdefunktionen lärs utöver styrpolicyn och δ-felet används för att approximera fördelsfunktionen, så sägs det vanligtvis att algoritmen har en så kallad *Aktör-Kritiker*-struktur (eng. *Actor-Critic*). Aktören är i en sådan struktur ansvarig för att estimera styrpolicyn π_θ, medan värdefunktionen $\hat{V}_\varphi^\pi$ estimeras av Kritikern. I de fall då Aktör-Kritiker-strukturen baseras på djupinlärning och neurala nätverk, så motsvarar parametrarna som används för att definiera styrpolicyn (θ) och värdefunktionen (φ) de vikter som sammankopplar de noder som finns i de neurala nätverk som används för att approximera de två funktionerna.

[2] En hyperparameter är en parameter vars värde används för att styra *hur* träningen av en algoritm utförs. Detta kan jämföras med en vanlig parameter vilket (exempelvis de nodvikter som finns i neurala nätverk) vilka lärs genom att träna på data.

2.5 PROXIMAL POLICY OPTIMIZATION-ALGORITMEN

Grundformuleringen av policy-gradientmetoder begränsas av dels 1) dataeffektivitet (eng. *sample efficiency*) och 2) robusthetsproblem då algoritmen tränas. Policy-gradientmetoder är så kallade "on-policy"-metoder, vilket betyder att den träningsdata som genererats måste baseras på den *gällande* styrpolicyn. Så fort styrpolicyn uppdaterats från insamlade träningsdata måste de ersättas med nya insamlade data. Detta resulterar i att "on-policy"-metoder är generellt mindre dataeffektiva i jämförelse mot "off-policy" metoder (som exempelvis Q-learning), där handlings-värdefunktionen fortfarande kan tränas på gamla insamlade träningsdata. För att förbättra dataeffektiviteten för "on-policy"-metoder kan det därför framstå som lockade att utföra flera steg (eng. *epochs*) av optimeringsalgoritmen på samma mängd träningsdata, eller att använda en större inlärningsfaktor (α). Studier har dock visat att det ofta leder till *för* stora steg i styrpolicyn, vilket i sin tur kan leda till en plötsligt kraftigt försämrade styrpolicy.

Proximal Policy Optimization (PPO) algoritmen som först presenterades i [8], har en relativt enkel lösning på dessa två problem. I de algoritmer som används i den här rapporten kommer den så kallade "klippta" varianten av PPO-algoritmen att användas, där målfunktionen definieras som:

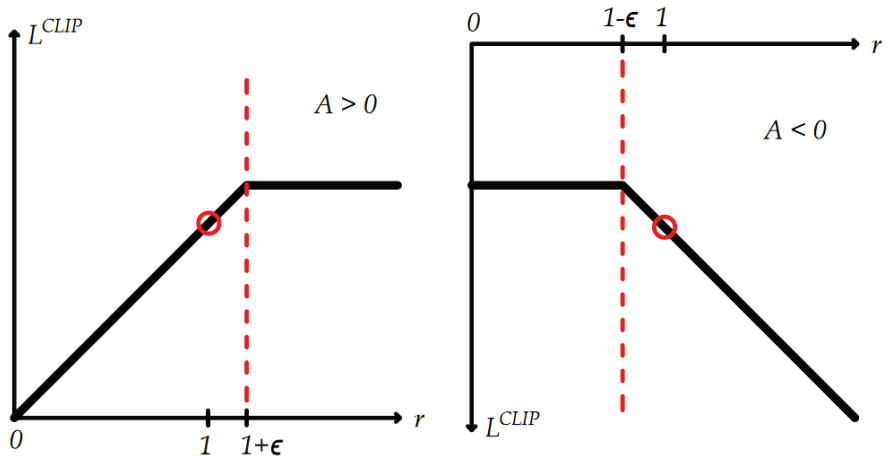
$$J^{clip}(\theta) = \hat{\mathbb{E}}_t\left[min\left(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right]\right. \tag{2-10}$$

där r_t är en sannolikhetskvot som definieras:

$$r_t = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{2-11}$$

och där θ_{old} refererar till den styrpolicy som använts för att generera den träningsdata som används vid uppdateringen av policyn. Funktionen "$clip$" används för att "klippa" värdet på sannolikhetskvoten till mellan $1-\epsilon$ och $1+\epsilon$ där ϵ är en hyperparameter (oftast mellan 0,1–0,2) som styr hur kraftigt kvoten bör begränsas. På grund av minimeringsfunktionen i målfunktionen (min) tas sedan det minsta värdet som ges av antingen 1) $r_t(\theta)\hat{A}_t$, 2) $(1-\epsilon)\,\hat{A}_t$, eller 3) $(1+\epsilon)\hat{A}_t$. Den klippta målfunktionen försäkrar att styrpolicyn inte ändras *för* mycket från den styrpolicy som använts för att generera träningsdata. Genom att införa denna begränsning kan därefter flera epochs av optimeringsalgoritmen utföras utan att styrpolicyn försämras.

Intuitionen av algoritmen kan förklaras i två delar, dels då fördelsfunktionen $\hat{A}_t$ är positiv, dels då den är negativ. Innan det första optimeringssteget är utfört är sannolikhetskvoten r_t lika med 1, då den nuvarande policyn är densamma som använts för att samla in träningsdata. Målfunktionerna för de två olika scenarierna är illustrerade i Figur 2. Den röda pricken indikerar startpunkten för optimeringen medan den streckade linjen indikerar gränsen för hur mycket sannolikhetskvoten r_t kan öka innan den begränsas (av antingen $1-\epsilon$, eller $1+\epsilon$).



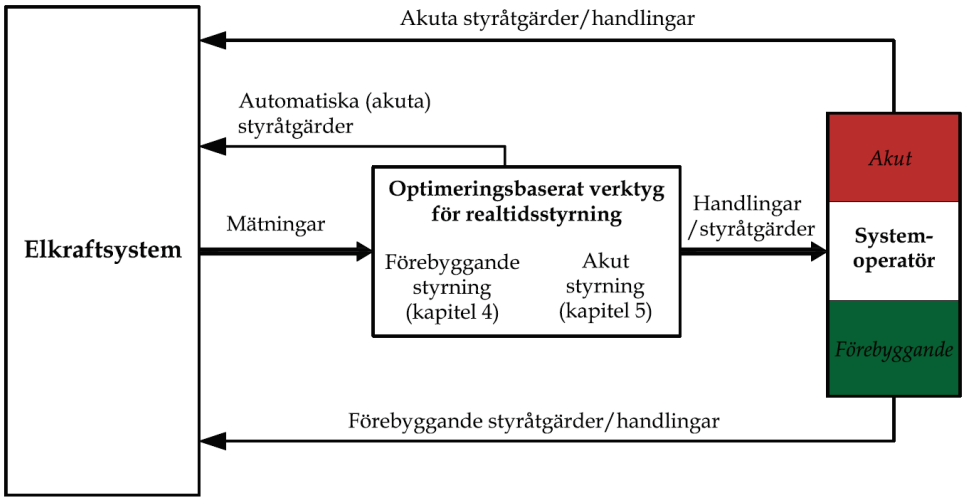
Figur 2. Den ”klippta” målfunktionen för PPO-algoritmen för en positiv samt negativ fördelsfunktion. Anpassad från [8].

- **Positiv fördelsfunktion:** En positiv fördelsfunktion för ett givet tillstånd och handling indikerar att den tagna handlingen gav en högre belöning än vad Kritikern förväntade sig. Således kommer målfunktionen att öka om den handlingen bli *mer sannolik* när agenten befinner sig i det givna tillståndet: detta motsvarar en ökning av $\pi_\theta(a_t|s_t)$, eller ekvivalent om sannolikhetskvoten r_t ökar till ett värde över 1.0. Mininmeringstermen i kostandsfunktionen sätter dock en gräns för hur mycket sannolikheten får öka, och när $\pi_\theta(a_t|s_t) > (1+\epsilon)\,\pi_{\theta_{old}}(a_t|s_t)$ så reduceras gradienten för målfunktionen till noll.
- **Negativ fördelsfunktion:** En negativ fördelsfunktion för ett givet tillstånd och handling indikerar att den tagna handlingen gav en lägre belöning än vad Kritikern förväntade sig. Därmed kommer målfunktionen att öka om den handlingen bli *mindre sannolik* när agenten befinner sig i det givna tillståndet: detta motsvarar en minskning av $\pi_\theta(a_t|s_t)$, eller ekvivalent om sannolikhetskvoten r_t minskar till ett värde under 1.0. Mininmeringstermen i kostandsfunktionen sätter dock en gräns för hur mycket sannolikheten får minska, och när $\pi_\theta(a_t|s_t) < (1-\epsilon)\,\pi_{\theta_{old}}(a_t|s_t)$ så reduceras gradienten för målfunktionen till noll.

3 Översikt kring DRL-baserade metoder för styrning av elsystem

I följande del presenteras en översikt av det optimeringsbaserade verktyget för realtidsstyrning av elkraftsystem som tagits fram inom projektet. Därutöver presenteras även en översikt kring det testsystem som använts för att utvärdera de utvecklade metoderna. Metoderna är uppdelade på två olika typer av styrning, dels 1) *förebyggande styrning*, dels 2) *akut styrning*. De två metoderna och resultat presenteras i mer detalj i kapitel 4, respektive kapitel 5.

En översikt av det optimeringsbaserade verktyget för realtidsstyrning och de två olika typerna av DRL-baserade styrsystem och dess interaktion med ett elkraftsystem presenteras i Figur 3. Verktyget samlar i realtid in mätningar från elkraftsystemet som därefter bearbetas av det upptränade DRL-baserade styrsystemet. Styrsystemet ger förslag till handlingar/styråtgärder som hanteras och aktiveras i systemet av systemoperatörerna. Akuta styråtgärder kan behöva aktiveras automatiskt för att säkerställa att systemets stabilitet hinner återställas i tid vid större störningar. Eventuella funktionskrav (som exempelvis uppdateringsfrekvens) på mätningarna diskuteras i respektive kapitel där de olika metoderna presenteras.



Figur 3. Översikt av det föreslagna optimeringsbaserade verktyget för realtidsstyrning av elkraftsystem.

3.1 FÖREBYGGANDE STYRNING

Elkraftsystem på en transmissionsnätsnivå styrs och efterföljer generellt den så kallade *N-1 regeln*, vilket innebär att de måste kunna hantera ett större dimensionerade fel (exempelvis bortfall av en större generator eller kraftledning) utan att systemets stabilitet förloras. Ett system som uppfyller det kriteriet refereras generellt till som *säkert*. För att bibehålla ett säkert system så övervakar systemoperatörer bland annat säkerhetsmarginaler och initierar förebyggande styråtgärder så fort systemet bedöms vara osäkert.

Säkerhetsmarginaler kan beräknas med olika metoder, men i den här rapporten används måttet som på engelska oftast kallas *Secure Operating Limit* (SOL) [9]. Vid beräkningen av SOL så beräknas marginalen till den teoretiskt högst lastade driftpunkten i systemet som fortfarande kan hantera ett större dimensionerande felfall utan att stabilitetskriterier överträds. Således ger SOL ett mått på hur mycket mer systemet kan lastas från den nuvarande driftpunkten och fortfarande bibehålla N-1 säkerhet. Beräkningen av SOL är beräkningsmässigt krävande och dynamiska simuleringar krävs för att kunna beakta det dynamiska förlopp som sker efter en större störning vid den teoretiskt maximalt lastade driftpunkten. Fördelen mot mer konventionella metoder som är baserade på statiska utvärderingar är att SOL generellt ger ett mer noggrant mått på den faktiska säkerhetsmarginal som systemet har. Ett alternativ är att använda mer approximativa metoder som exempelvis så kallade kvasi-statiska (eng. *quasi-steady state*) metoder, vilket dock kommer bidra med en viss förlust av modellnoggrannhet.

Typiska styråtgärder inom förebyggande styrning inkluderar (1) hantering av reaktiva effektresurser och spänningsstyrning genom styrning av transformatorer med lindningsomkopplare samt genom in- och urkoppling av shuntkopplade kondensatorer/reaktorer, samt (2) avlastning av belastade ledningar och stationer genom nedstyrning av laster och styrning av produktionsanläggningar. I framtiden kommer med stor sannolikhet även andra styrbara tjänster och komponenter att kunna användas. Exempel kan vara utnyttjande av lastflexibilitet och energilagringssystem, eller reaktivt effektbidrag från underliggande distributionssystem. Den utvecklade metoden för förebyggande styrning bygger på att använda en tränad DRL-agent som i realtid kan föreslå lämpliga kontrollåtgärder då säkerhetsmarginalerna (SOL) understiger fördefinierade tröskelnivåer. Systemoperatören kan välja att automatiskt initiera dessa åtgärder i systemet eller manuellt bedöma de föreslagna åtgärderna och sedan vidta åtgärderna manuellt.

3.2 AKUT STYRNING

Förebyggande styrning används således för att försäkra att ett elkraftsystem alltid kan hantera åtminstone *ett* större dimensionerade fel. Vid allvarligare fel, som exempelvis att flera samtidiga störningar inträffar simultant, måste systemoperatörerna i stället förlita sig på akuta styrmetoder för att stabilisera systemet. Akut styrning används när systemet har gått in i ett nödläge, och systemoperatören måste då snabbt vidta åtgärder för att säkerställa att systemets stabilitet återställs. Här är huvudmålet ofta att undvika en större systemkollaps eller en större bortkoppling av nätet, och mer omfattande och kostsamma styråtgärder finns därmed tillgängliga för systemoperatören.

Den utvecklade metoden för akut styrning bygger på att använda en tränad DRL-agent som i realtid kan föreslå lämpliga kontrollåtgärder närhelst systemet kan vara på väg mot ett instabilt tillstånd. Då tiden mellan detektering och åtgärdande av fel kan vara kritisk i många akuta situationer, kan de åtgärder som föreslås av DRL-agenten behöva aktiveras automatiskt i systemet för att undvika att systemet kollapsar. Vid långvarig spänningsinstabilitet, där tiden mellan då störningen

inträffar och ett eventuellt instabilt tillstånd kan utvecklas är relativt lång, kan systemoperatörerna ha tid att manuellt bedöma de föreslagna åtgärderna innan de aktiveras. Akut styrning inkluderar vanligtvis åtgärder som bortkoppling av laster, styrning av HVDC-länkar, eller kontrollerad ödrift.

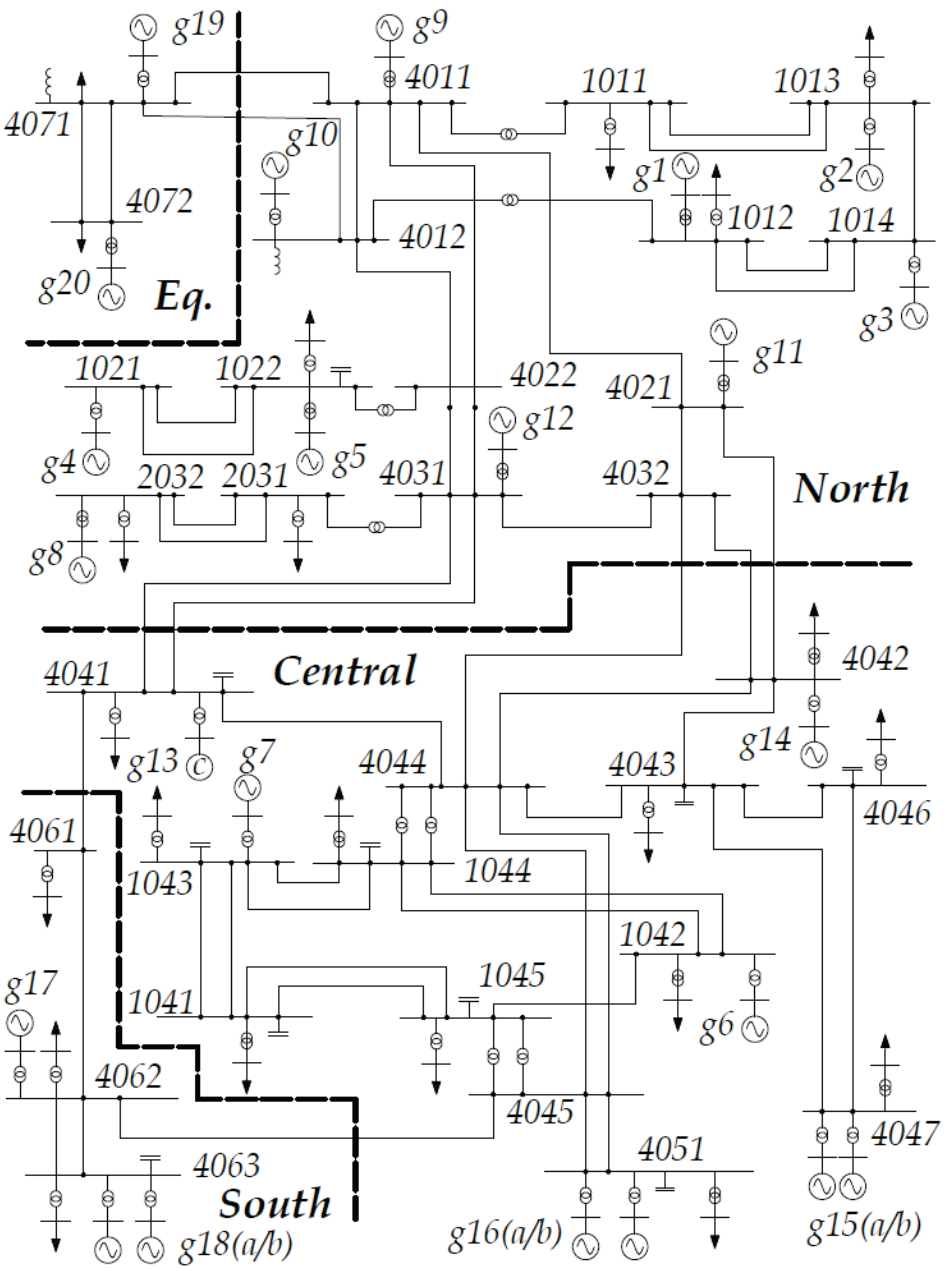
3.3 TEST- OCH TRÄNINGSSYSTEM

De utvecklade metoderna testas på en reducerad simuleringsmodell av det nordiska elsystemet. Testsystemet är uppdaterad version av testsystemet ”Nordic32”, och presenteras i sin helhet i referens [10]. Systemet är fiktivt, men har egenskaper som efterliknar det svenska och nordiska kraftsystemet. En översikt över testsystemet presenteras i Figur 4. Systemet är uppdelat i fyra olika områden med olika last- och genereringskaraktäristik:

- ”North”: huvudsakligen vattenkraft samt några mindre laster.
- ”Central”: det största lastcentret med både betydande generering av termisk kraftgenerering samt en större mängd laster.
- ”Eq”: En motsvarighet till ett externt system anslutet till ”North”-området.
- ”South”: Ett område med termisk generering som är löst kopplat till ”Central”-området.

Testsystemet karaktäriseras av långa transmissionsledningar på 400 kV och 220 kV nominell spänning. Testsystemet inkluderar även en representation av regionala system med en driftspänning på 130 kV. I Tabell 2 presenteras den aktiva elproduktionen och lasten i respektive område och för hela testsystem som en helhet. Systemet är hårt belastat med stora kraftöverföringar främst mellan områdena ”North” och ”Central”. Den överförbara effekten begränsas av generatorers reaktiva effektkapacitet i dessa båda områden. En störning i överföringskapaciteten som förbinder områdena ”North” och ”Central” är avgörande för systemets stabilitet. En minskad överföringskapacitet, orsakad av exempelvis en frånkopplad ledning mellan områdena, skulle öka belastningen på de återstående transmissionsledningarna i systemet. Den ökade strömmen i de ledningarna skulle öka reaktiva effektförluster i systemet, vilket i sin tur skulle bidra till lägre systemspänningar. Spänningsberoende laster återställs då transformatorer med lindningsomkopplare försöker återställa spänningsnivåer till nominella nivåer. Laståterhämtningen har en negativ effekt på systemets spänningsstabilitet eftersom de återhämtade lasterna kommer att orsaka en ökad belastning på de återstående ledningarna i systemet och kan få systemet att försämras ytterligare.

För det utvecklade testsystemet finns två olika driftpunkter definierade i [10]. Driftpunkt A är en osäker driftpunkt och ett avbrott antingen i en större termisk generator i ”Central”-området eller en frånkoppling av en transmissionsledning som förbinder ”North”- och ”Central”-områdena kan orsaka instabilitet. Driftpunkt B är en säker driftpunkt som ska kunna motstå en större störning (N-1 stabilt) utan att systemet blir instabilt. Systemet modelleras i kraftsimuleringsmjukvaran PSS®E med dess inbyggda dynamiska modeller [11].



Figur 4. Det uppdaterade Nordic32 testsystem som använts i rapporten. Bild anpasssad från [10].

Tabell 2. Aktiv genererad effekt samt aktiv last inom olika områden för det uppdaterade Nordic32 testsystemet.

Område	Aktiv genererad effekt [MW]	Aktiv last [MW]
”North”	4 629	1 180
”Central”	2 850	6 190
”South”	1 590	1 390
”Eq.”	2 437	2 300
Totalt	**11 506**	**11 060**

4 Förebyggande styrning baserad på DRL

I följande del presenteras en metod för förebyggande styrning baserad på DRL. Metoden bygger på att träna en DRL-agent till att i realtid kunna föreslå optimerade styråtgärder då elsystemets säkerhetsmarginaler understiger fördefinierade tröskelnivåer. DRL-agenten övervakar systemets nuvarande tillstånd genom en uppsättning mätningar. Om DRL-agenten bedömer att systemet inte är säkert eller har en för liten marginal mot säkerhetsgränsen, initieras styråtgärder för att styra systemet till ett säkrare tillstånd. Andra typer av styrning, som exempelvis att hålla systemspänningar inom acceptabla gränser, ingår inte i den utvecklade styrningen, men skulle kunna läggas till som ytterligare styrmål. Säkerhetsmarginalen som DRL-agenten styr baseras på det mer avancerade säkerhetsmåttet SOL, som är bättre anpassat till att ta hänsyn till dynamiska stabilitetsgränser, tidigare beskrivet i del 3.1. Metoden är även anpassad för att kunna styra *både* kontinuerliga (eng. *continuous*) *och* diskreta (eng. *discrete*) handlingar samtidigt[3]. Detta ger exempelvis möjlighet att kunna styra reaktiv effekt genom omkoppling av shuntinkopplade kondensatorer/reaktorer (en diskret handling), *samtidigt* som effektflöden genom nedstyrning av laster på underliggande nät kan regleras (en kontinuerlig handling). Därutöver undersöks flera andra aspekter kring metoden, som exempelvis robustheten för olika störningsscenarier samt brus och fel i mätdata.

4.1 ANPASSNING FÖR HYBRIDSTYRNING AV KONTINUERLIGA OCH DISKRETA HANDLINGAR

För att DRL-agenten ska kunna hantera en hybridstyrning där både kontinuerliga och diskreta handlingar kan utföras samtidigt så krävs vissa anpassningar i hur styrpolicyn definieras. Implementeringen som används följer till stor del den som tidigare utvecklats i [12]. En stokastisk hybridpolicy $\pi_\theta(\boldsymbol{a}|s)$ kan definieras som en tillståndsberoende fördelning som gemensamt modellerar såväl kontinuerliga som diskreta sannolikhetsfördelningar. De olika definierade sannolikhetsfördelningarna (och därmed även de olika handlingarna) antas vara oberoende av varandra, vilket gör att hybridpolicyn kan skrivas som:

$$\pi_\theta(\boldsymbol{a}|s) = \pi_\theta^{\mathcal{C}}(\boldsymbol{a}^{\mathcal{C}}|s)\pi_\theta^{\mathcal{D}}(\boldsymbol{a}^{\mathcal{D}}|s) = \prod_{a^i \in \boldsymbol{a}^{\mathcal{C}}} \pi_\theta^{\mathcal{C}}(a^i|s) \prod_{a^i \in \boldsymbol{a}^{\mathcal{D}}} \pi_\theta^{\mathcal{D}}(a^i|s) \tag{4-1}$$

där a^i representerar en enskild handling (kontinuerlig eller diskret), $\boldsymbol{a}^{\mathcal{C}}$ och $\boldsymbol{a}^{\mathcal{D}}$ är delmängder för de kontinuerliga och diskreta handlingsmängderna (där $\mathcal{C}$ respektive $\mathcal{D}$ representerar kontinuerliga respektive diskreta handlingsmängder), och $\boldsymbol{a}$ är en vektor bestående av både kontinuerliga respektive diskreta handlingar. Hur de stokastiska styrpolicyerna för de kontinuerliga ($\pi_\theta^{\mathcal{C}}$) respektive diskreta handlingarna $\pi_\theta^{\mathcal{D}}$ definieras kan varieras, men ofta används en normalfördelning för definitionen av de kontinuerliga handlingarna, och en Bernoullifördelning för

[3] Med en kontinuerlig handling avses en handling som kan ta vilket värde som helst (inom ett visst intervall). Med en diskret handling avses en handling som endast kan vissa värden, som exempelvis heltal.

de diskreta handlingarna. Används en normalfördelning kan de kontinuerliga styrpolicyerna definieras med:

$$\pi_\theta^c(a^i|s) = \mathcal{N}\left(\mu_{i,\theta}(s), \sigma_{i,\theta}^2(s)\right) \tag{4-2}$$

där $\mathcal{N}$ representerar en normalfördelning, och där medelvärdet $\mu_{i,\theta}$ och variansen $\sigma_{i,\theta}^2$ är normalfördelningens karakteristiska parametrar. Används en Bernoullifördelning kan de diskreta styrpolicyerna definieras med:

$$\pi_\theta^D(a^i|s) = Bern(P_\theta(s)) \tag{4-3}$$

där $Bern$ representerar en Bernoullifördelning parametriserad av den tillståndsberoende parametern $P_\theta(s)$. Andra typer av sannolikhetsfördelningar, som exempelvis en Betafördelning för en kontinuerlig handlingsrymd, eller en Softmaxfördelning för en representation av en flerklassklassificering, är möjliga alternativ. I den här studien antas parametrarna $\mu_{i,\theta}$, $\sigma_{i,\theta}^2$ samt P_θ vara de resulterande utgångsvärdena på ett neuralt nätverk. θ representerar parametrarna i styrpolicyn som önskas optimeras och motsvarar i det neurala nätverket de vikter som sammanbinder neuronerna/noderna i nätverket.

4.2 METOD OCH DATAGENERERING

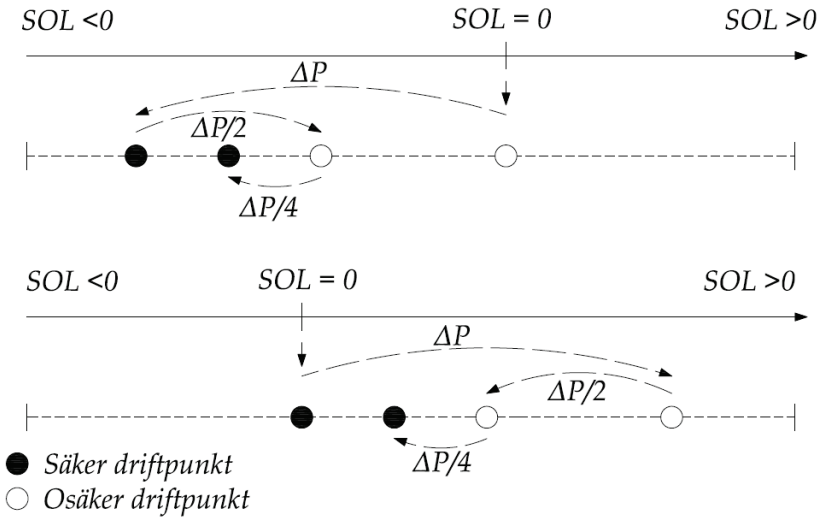
I den här delen presenteras den övergripande metoden och datagenereringen som använts för att träna DRL-agenten. DRL-agenten tränas på olika last- och störningsscenarier för att återspegla de olika driftförhållanden som kan uppstå i ett verkligt kraftsystem. Alla simuleringar har testats på det uppdaterade testsystemet Nordic32, tidigare presenterat i kapitel 3 samt i [10]. Systemet är känsligt mot långsam spänningsinstabilitet och styrmetoden kommer att begränsas till att hantera främst detta instabilitetsfenomen. Alla last- och störningsscenarier som använts har genererats med PSS®E 35.0.0 med dess inbyggda dynamiska modeller.

4.2.1 Beräkning av säkerhetsmarginaler

Säkerhetsmarginalen SOL beräknades genom en sökprocess som i mer noggrannhet diskuteras i [13]. Processen illustreras i Figur 5 och baseras på att iterativt testa systemsäkerheten inom ett krympande intervall med olika nivåer av systembelastning och för olika dimensionerande störningar. Sökprocessen startar alltid genom att en dynamisk simulering initieras. Därefter testas systemets säkerhet genom att en av de dimensionerande störningarna (beskrivna senare i del 4.2.3) appliceras i systemet för att avgöra systemets tillstånd. Den övre delen i Figur 5 illustrerar sökprocessen när det initiala tillståndet är osäkert, medan den undre delen illustrerar sökprocessen när det initiala tillståndet är säkert, där svarta prickar indikerar ett säkert tillstånd, medan vita prickar indikerar ett tillstånd som är osäkert.

Sökprocessen exemplifieras här då ett *säkert* initialt tillstånd har identifierats. Då det initiala tillståndet var säkert, så ökades belastningen i systemet för den givna initiala driftpunkten genom att öka lasten i ”Central”-regionen med totalt $\Delta P = 128$ MW, samtidigt som genereringen i ”North”-regionen justerades upp med samma mängd produktion. Effektfaktorerna för samtliga laster behölls vid deras initiala värden och fördelningen av den ökade belastningen och produktionen baserades

återigen på den initiala belastningen för respektive lastbuss och de nominella kapaciteterna för varje generator. Efter att lasterna och produktionen uppdaterats så utfördes ytterligare en lastflödesberäkning som utgångspunkt för en ny dynamisk simulering med ett hårdare lastat system. Systemets säkerhet testades därefter igen genom att återigen applicera de dimensionerande störningarna i systemet. Om det nya, mer belastade systemet, visade sig vara säkert så ökades systembelastningen återigen med ΔP. Om det inte visade sig vara säkert så reducerades systembelastningen med $\Delta P/2$. Sökprocessen fortsatte sedan med ett smalare och smalare intervall av olika systembelastningar tills en säker driftpunkt identifierats och tills ΔP hade minskat till 1 MW. De dimensionerande störningarna som applicerades för varje driftpunkt var en bortkoppling av någon av de tre största generatorerna som finns i "Central"-området: generator *g14*, generator *g15* eller generator *g16*. Den störning som resulterade i den lägsta säkerhetsmarginalen är alltid dimensionerande och användes därefter som den faktiska säkerhetsmarginalen.



Figur 5. Illustrering av sökprocessen för beräkning av SOL för en säker respektive en osäker driftpunkt.

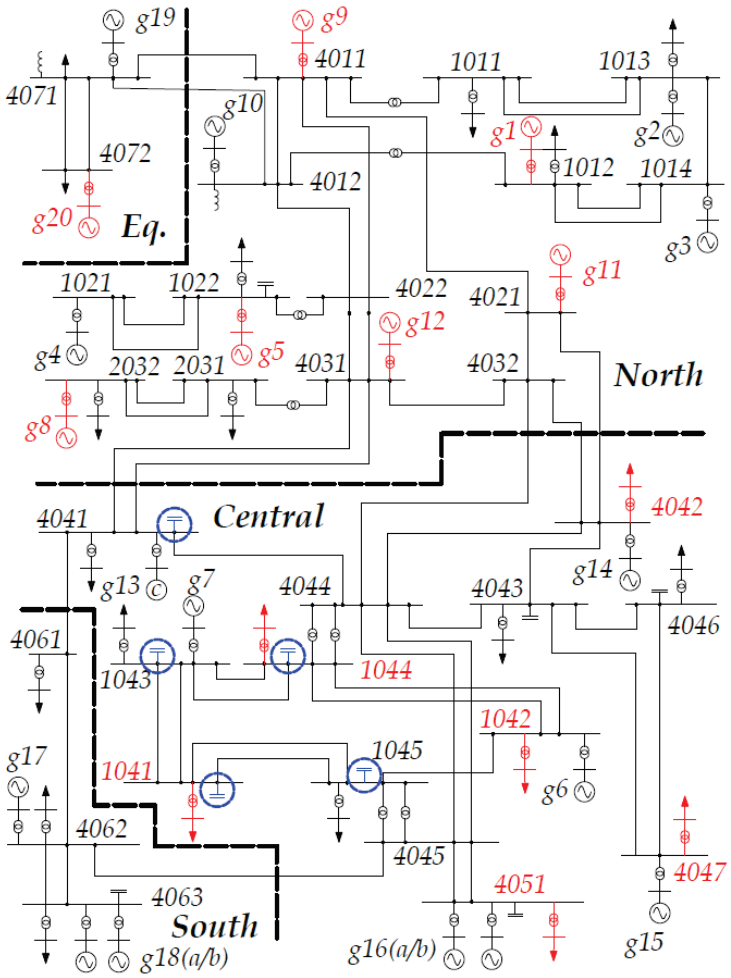
Varje dynamisk simulering kördes maximalt i 600 sekunder, men stoppades i förväg om 1) systemet kollapsade (någon spänningsmagnitud understeg 0,7 per unit) eller 2) om systemet stabiliserades i förtid. Systemet ansågs vara säkert om samtliga transmissionsbusspänningar var över 0,9 per unit vid slutet av den dynamiska simuleringen. Detta tillvägagångssätt säkerställde att systemet antingen hade stabiliserats eller blivit instabilt i slutet av varje simulering.

4.2.2 Tillstånd, handlingar och belöningsdefinitioner

Styrproblemet definieras som en episodisk MDP. I början av varje episod så mottar DRL-agenten en representation av tillståndet s_t vilket baseras på en uppsättning mätningar i systemet. Beroende på den aktuella styrpolicyn π_θ så väljer DRL-agenten olika handlingar som därefter aktiveras i systemet. De vidtagna handlingarna och dynamiken för tillståndsövergångar gör att systemtillståndet

ändras ($s_t \rightarrow s_{t+1}$) och ger även upphov till en belöning R_t. Om DRL-agenten lyckas återställa SOL till över det definierade tröskelvärdet på 30 MW, eller om ett maximalt antal tidsssteg har uppnåtts, så avslutas episoden. Annars fortsätter DRL-agenten att observera nya tillstånd, ta nya handlingar, och motta nya belöningar. Den använda MDP:n kan beskrivas av:

- *Tillstånd:* består av en tillståndsvektor s_t bestående av mätningar av 1) spänningsmagnituder för alla bussar i systemet samt 2) aktiva och reaktiva effektflöden på alla transmissionsledningar och genom alla transformatorer. Det aktuella tidssteget t lades också till i tillståndsvektorn. Neurala nätverk är känsliga för brus i ingångsvärden så för att göra agenten mer robust mot sådana fel så multiplicerades varje tillståndsvärde med ett slumpmässigt genererat tal från en normalfördelning med ett medelvärde på 1,0 och med en standardavvikelse på 0,001. Då tillståndsvärdena kan variera relativt kraftigt så normaliserades även tillståndsvektorns data genom att subtrahera medelvärdet för varje tillståndsvärde och sedan dividera med dess standardavvikelse. Medelvärdet och standardavvikelsen för varje tillstånd beräknades från tidigare samplade tillstånd och från en dubbelslutlig lista (eng. *double-ended queue*) med data där maximalt 10 000 samplade tillstånd hade lagrats. Därmed, när fler och fler iterationer utförts så fylldes den dubbelslutliga listan på med nya insamlade tillstånd och äldre data togs bort från listan.
- *Handlingar:* DRL-agenten kan aktivera antingen kontinuerliga och/eller diskreta handlingar. De kontinuerliga handlingarna användes till att förändra kraftgenerering och utföra lastbegräsningar för att minska kraftöverföringen genom systemet och på så sätt förbättra systemets säkerhet. Detta uppnåddes genom att minska förbrukningen i ”Central”-regionen för vissa deltagande lastbussar: 4042, 4047, 4051, 1041, 1042, 1044, samtidigt som genereringen i ”North”-regionen ökar hos vissa deltagande generatorer: G1, G5, G8, G9, G11, G12, G20. Minskningen distribuerades och delades därefter jämnt ut på alla deltagande lastbussar, där distributionen baserades på den initiala lasten som respektive lastbuss hade före minskningen. Effektfaktorn för alla laster behölls åter konstant.
  Den minskade förbrukningen i ”Central”-regionen kan åstadkommas genom utnyttjande av framtida stödtjänster som exempelvis lastflexibilitet eller utnyttjande av energilagringssystem. Sådana system har möjligheten att mer flexibelt anpassa sig åt styrningen från DRL-agenten än exempelvis automatiska lastbortkopplingssystem har, som generellt kräver att större delar av laster kopplas bort simultant. De diskreta handlingarna inkluderade en möjlig inkoppling av ytterligare 100 MVAr reaktiv effekt från någon, eller flera, shuntinkopplade kondensatorer. De diskreta handlingarna styrde shuntinkopplade kondensatorer som var inkopplade vid följande bussar: $D_1 = 1041$, $D_2 = 1043$, $D_3 = 1044$, $D_4 = 1045$, $D_5 = 4041$. De deltagande bussarna för såväl den kontinuerliga handlingen samt för de diskreta handlingarna är markerade i rött, respektive blått, i enlinjediagrammet i Figur 6.
- *Dynamiken för tillståndsövergångar:* Tillståndsövergångsdynamiken är deterministisk och styrs av en uppsättning differentialekvationer och algebraiska ekvationer som används för att bygga den dynamiska modellen i PSS®E.



Figur 6. Enlinjediagram av det modifierade Nordic32-systemet. Laster och generatorer inkluderade i den kontinuerliga handlingen är markerade i rött, medan shuntinkopplade kondensatorer inkluderade i de diskreta handlingarna är markerade i blått.

- *Belöningar:* Belöningen R_t för de tagna handlingarna beräknades som en kombination av den resulterande säkerhetsmarginalen SOL och kostnaden för de kontinuerliga (C_{cont}) och diskreta handlingarna (C_{disc}). I den här studien är alla belöningar enhetslösa, men i verkliga applikationer så bör belöningarna exempelvis reflektera de uppskattade monetära kostnader som olika handlingar har i verkligheten. Varje aktivering av en diskret handling bidrog till en negativ belöning på -5, vilket representerar den kostnad för mekaniskt slitage som är involverat vid in- och urkoppling av shuntkopplade kondensatorer. Kostanden för den kontinuerliga handlingen bidrog med till en negativ belöning på -0.1 per anpassad MW i lastöverföringen i systemet. Således skulle en minskning av överföringen på -200 MW resultera i ett negativt bidrag till belöningen på totalt -20. Detta är tänkt att reflektera den kostnad som är involverad vid last- och genereringsändringar i ett elsystem. Styrmålet för DRL-agenten är att alltid återställa SOL till ett värde lika med eller högre än 30 MW, vilket säkerställer att det finns en tillräcklig säkerhetsmarginal och att N-1 kriteriet är uppfyllt med en viss marginal. Om

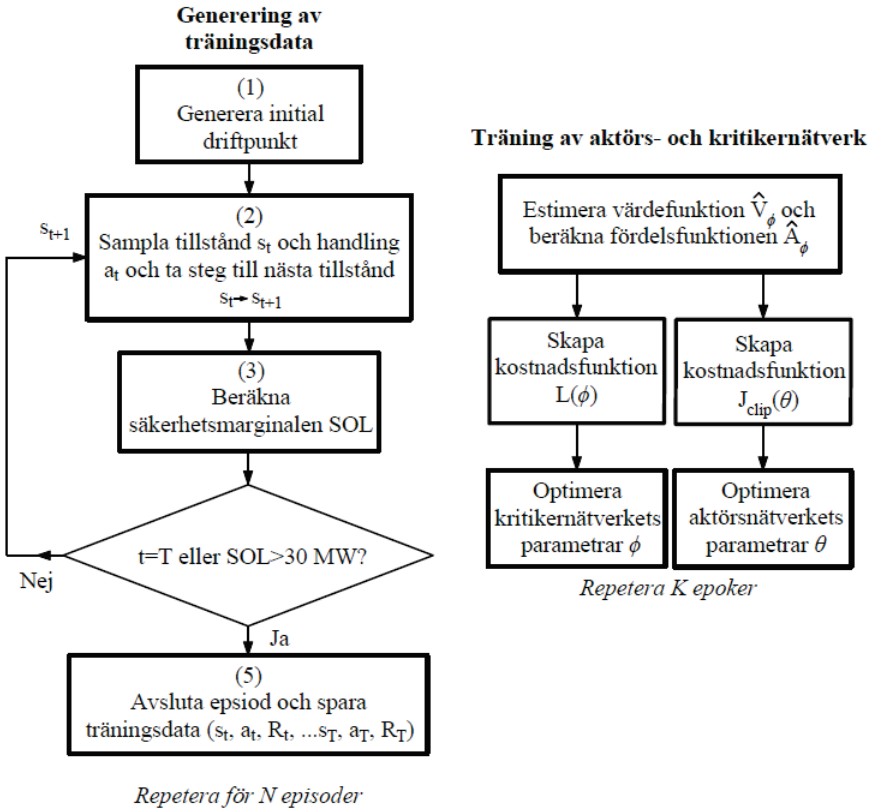
SOL är lägre än 30 MW resulterade detta i en större negativ belöning på -50 adderades till den totala belöningen för det aktuella tidssteget. Belöningen baserades även på det faktiska värdet på säkerhetsmarginalen SOL, där en lägre säkerhetsmarginal resulterade i en högre (negativ) belöning. Den slutliga belöningen för varje tidssteg kan beskrivas enligt följande:

$$R_t = \begin{cases} C_{cont} + C_{disc} + SOL - 50 & \text{om SOL } \leq 30 \text{ MW} \\ C_{cont} + C_{disc} & \text{annars} \end{cases} \quad (4\text{-}4)$$

4.2.3 Generering av träningsdata

En översikt över stegen vid generering av träningsdata och träningen av DRL-agenten illustreras i Figur 7. De olika stegen detaljeras nedan.

1. *Generering av initiala driftpunkter*: Ett stort antal olika initiala driftpunkter för testsystemet genererades för att fungera som träningsdata för DRL-agenten. Driftpunkterna genererades kring den stabila driftpunkten för det modifierade testsystemet Nordic32 som betecknads som ”driftpunkt B” enligt [9]. Alla laster i systemet varierades slumpmässigt och individuellt genom att multiplicera den aktiva lasten med en slumpmässig variabel tagen från en likformig sannolikhetsfördelning (med 80 % av den ursprungliga aktiva lasten som nedre gräns och 130 % som av den ursprungliga lasten som övre gräns). Variationen i lasterna användes för att spegla de olika belastningsnivåer som kan förekomma i ett kraftsystem över tid. I verkliga implementeringar kan träningsdata med fördel samlas in både från historiska verkliga driftpunkter samt för genererade driftscenarier. Effektfaktorn för alla laster behölls konstant och den reaktiva effekten anpassades därmed efter variationen i aktiv last. Den totala förändringen i aktiv last distribuerades därefter mellan alla generatorer i systemet där en generator med en högre nominell kapacitet fick en högre sannolikhet att kompensera för en större andel av den förändrade belastningen. När en initial driftpunkt hade skapats så utfördes därefter en lastflödesberäkning med hjälp av Newton-Raphson metoden för att beräkna fram systemets statiska parametrar. Om lastflödesberäkningen ej lyckades att konvergera så initierades en ny slumpmässig driftpunkt.
2. *Sampla tillstånd s_t och handling a_t från styrpolicyn och gå till nästa tillstånd*: När en initial driftpunkt genererats, så samplades tillståndet s_t från systemet och skickades till aktörsnätverket. Aktörsnätverket används för att ta fram parametrar som formar den aktuella styrpolicyn $\pi_\theta(\boldsymbol{a}|s)$, från vilken handlingar därefter samplas från. Aktörsnätverket och hur den används för att definiera den använda styrpolicyn är detaljerat i sektion 4.2.4. När väl handlingarna samplats och aktiverats i systemet så utfördes ytterligare en lastflödesberäkning vilket därmed resulterade i en tillståndsövergång från $s_t \rightarrow s_{t+1}$.



Figur 7. Flödesschema som visar genereringen av träningsdata och träningen av aktörs- och kritikernätverket.

1. *Beräkna säkerhetsmarginalen SOL:* När lastflödesberäkningen utförts så beräknades säkerhetsmarginalen SOL enligt de steg som finns beskrivna i sektion 4.2.1.
2. *Avsluta episod och spara träningsdata:* Episoden avslutades om antingen säkerhetsmarginalen SOL beräknades till över 30 MW (det antagna gränsvärde för då systemet kunde antas vara säkert) eller om det aktuella tidssteget var lika med T=8. Vid slutet på varje episod så samlades och sparades all träningsdata från episoden ($s_t, a_t R_t \dots s_T, a_T, R_T$) och användes senare under träningen av aktörs- och kritikernätverken. Genereringen av träningsdata utfördes totalt $N = 64$ episoder innan nätverken tränades på den data som genererats.

4.2.4 Arkitektur för aktör- och kritikernätverk

Aktörsnätverket illustreras i Figur 8 och detaljeras ytterligare i Tabell 3. Det har ett gemensamt dolt neuronlager, och sedan används ett mindre separat dolt neuronlager för respektive aktiveringsfunktion. Aktörsnätverket används för att ta fram parametrar som definierar en Normalfördelning och fem olika Bernoullifördelning, varifrån en kontinuerlig handling respektive fem olika diskreta handlingarna kan samplas från. Normalfördelningen parametriseras dels av ett medelvärde μ_{cont} samt en standardavvikelse σ_{cont}, medan Bernoullifördelningen parametriseras av en sannolikhetsparameter $P(D_x|s)$ som varierar mellan 0–1. Medelvärdet μ_{cont} fås genom att använda en linjär

aktiveringsfunktion i ett av de slutliga aktiveringslagren. Standardavvikelsen σ_{cont} fås genom att använda en aktiveringsfunktion som bygger på en ”Softplus”-funktion som säkerställer att standardavvikelsen aldrig kan bli negativ. Slutligen fås sannolikhetsparametern som används för Bernoullifördelningen genom att använda en sigmoidfunktion i ett av de slutliga aktiveringslagren, som säkerställer att värdet är bundet mellan 0 och 1.

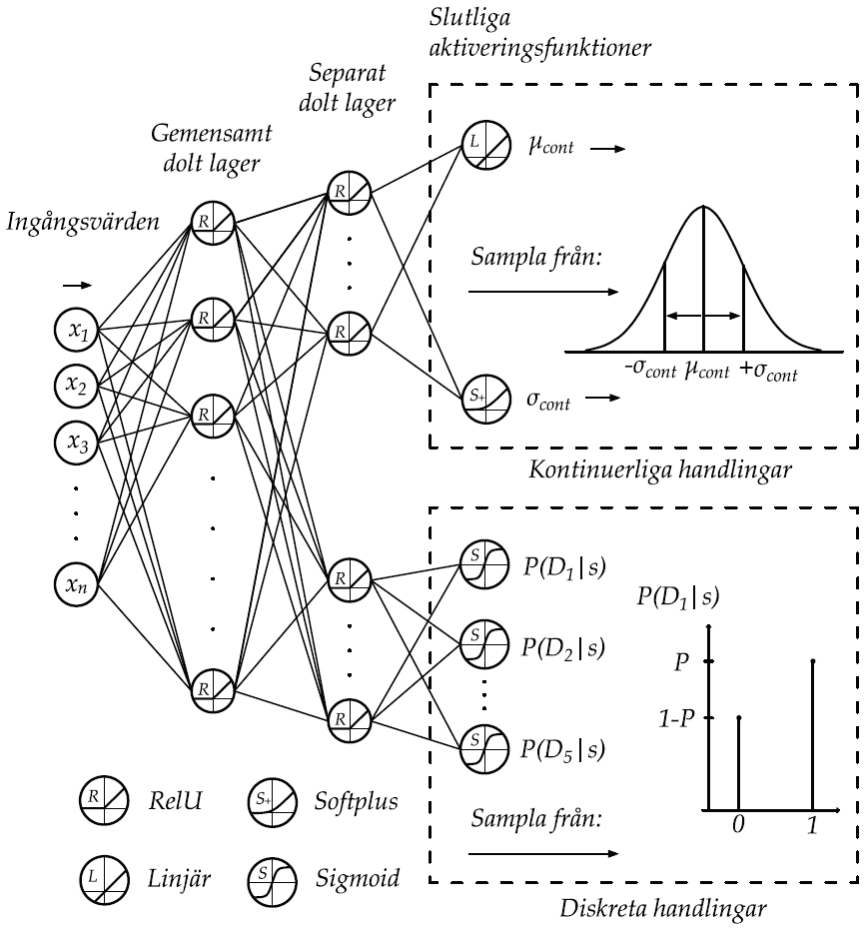
Kritikernätverket är separat definierat från aktörsnätverket och består av ett neuralt nätverk med två dolda lager och en slutlig linjär aktiveringsfunktion. De parametrar som använts för att definiera kritikernätverkets presenteras i Tabell 3.

Tabell 3. Design och hyperparametrar använda under träning och generering av träningsdata.

		Parametrar	Värden
Struktur för neurala nätverk	*Kritikernätverk*	Antal ingångsvärden/tillstånd	499
		Neuroner i gemensamma lagret	128
		Neuroner i respektive separat dolt lager	64
		Slutgiltig aktiveringsfunktion	Linjär
		Aktiveringsfunktion för dolda neuronlager	RelU
	Aktörnätverk	Antal ingångsvärden/tillstånd	499
		Neuroner i gemensamma lagret	128
		Neuroner i respektive separat dolt lager	64
		Slutgiltig aktiveringsfunktion för μ_{cont}	Linjär
		Slutgiltig aktiveringsfunktion för σ_{cont}	Softplus
		Slutgiltig aktiveringsfunktion för $P(D_x\|s)$	Sigmoid
		Aktiveringsfunktion för dolda neuronlager	RelU
	Träning	Maximal episodlängd (T)	8
		Antal epochs per träningsiteration (K)	10
		PPO klippningsparameter (ϵ)	0,2
		Diskonteringsfaktor (γ)	0,99
		Optimeringsalgoritm	RMSprop [14]

Tabell 4. Parametrar för inlärningshastigheter som använts för olika träningsiterationer.

Träningsiteration	$\alpha_{aktör}$	$\alpha_{kritiker}$
≤ 250	$1 \cdot 10^{-3}$	$5 \cdot 10^{-3}$
> 250	$1 \cdot 10^{-4}$	$5 \cdot 10^{-4}$



Figur 8. Struktur och uppbyggnad av aktörnätverket.

4.2.5 Träning av aktör- och kritikernätverk

När väl en "batch" av totalt = 64 episoder motsvarande träningsdata samlats in så tränades aktörs- och kritikernätverken. Träningen utfördes med hjälp av mjukvaran Tensorflow i programmeringsspråket Python som automatiskt beräknar gradienterna för de definierade målfunktionerna. Kritikernätverket användes först för att beräkna värdefunktionen $\hat{V}_{\varphi}^{\pi}$ för varje samplat tillstånd. Avkastningen G_t^{γ} estimerades genom att använda ekvation (2-3) och fördelsfunktionen för varje tillstånd beräknades därefter genom ekvation (2-8) och det estimerade värdet för $\hat{V}_{\varphi}^{\pi}$ vilket gavs från kritikernätverket. Ett värde på $\gamma = 0{,}99$ användes för beräkningarna. Målfunktionen $J^{clip}(\theta)$ som används för att träna aktörsnätverket beräknas med det beräknade medelvärdet av ekvation (2-10) för alla samplade värden och för samtliga N episoder. Målfunktionen som används för att träna kritikernätverket $L(\phi)$ beräknades genom att ta medelkvadratfelet på samtliga värden beräknade från ekvation (2-9) för alla N episoder, och därefter beräkna medelvärdet av dessa värden. När målfunktionerna för aktörs- och kritikernätverken beräknats, optimerades dessa med hjälp av RMSprop-algoritmen vilket är en optimeringsalgoritm anpassad för gradientbaserad optimering av stokastiska målfunktioner [14]. Aktörsnätverket maximerades med avseende på θ, medan kritikernätverket optimerades genom att minimera målfunktionen med

avseende på φ. Träningen utfördes med $K = 10$ epochs för samtliga N episoder samtidigt. För att öka hastigheten för träningen av nätverken och för att i senare delen av träningen stabilisera konvergensen så anpassades inlärningshastigheten enligt specificering i Tabell 4.

Tabell 4. Parametrar för inlärningshastigheter som använts för olika träningsiterationer.

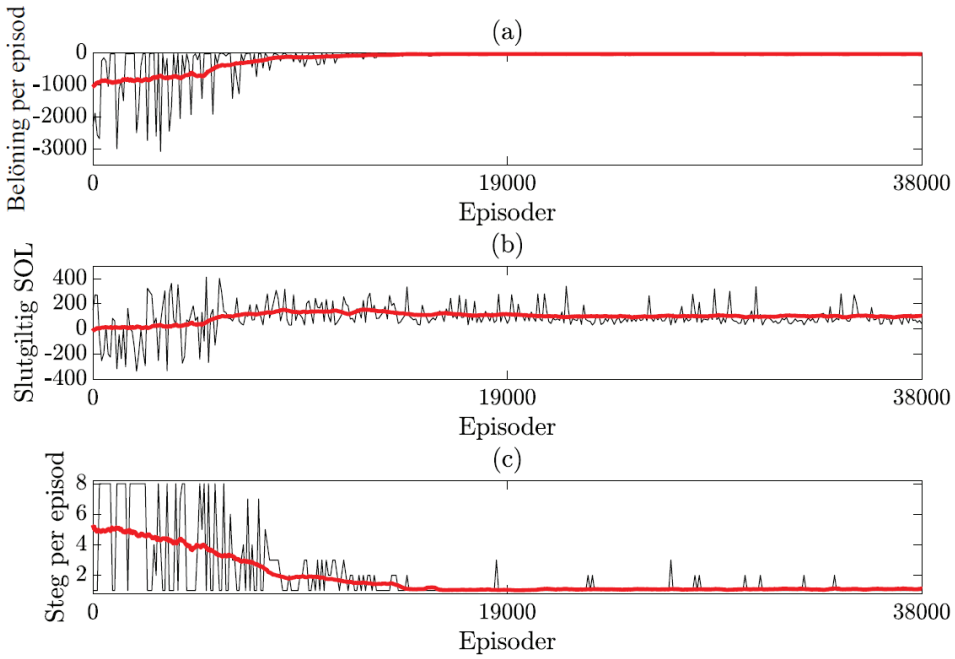
Träningsiteration	$\alpha_{\text{aktör}}$	α_{kritiker}
≤ 250	$1 \cdot 10^{-3}$	$5 \cdot 10^{-3}$
> 250	$1 \cdot 10^{-4}$	$5 \cdot 10^{-4}$

4.3 TESTSETS OCH TRÄNINGSRESULTAT

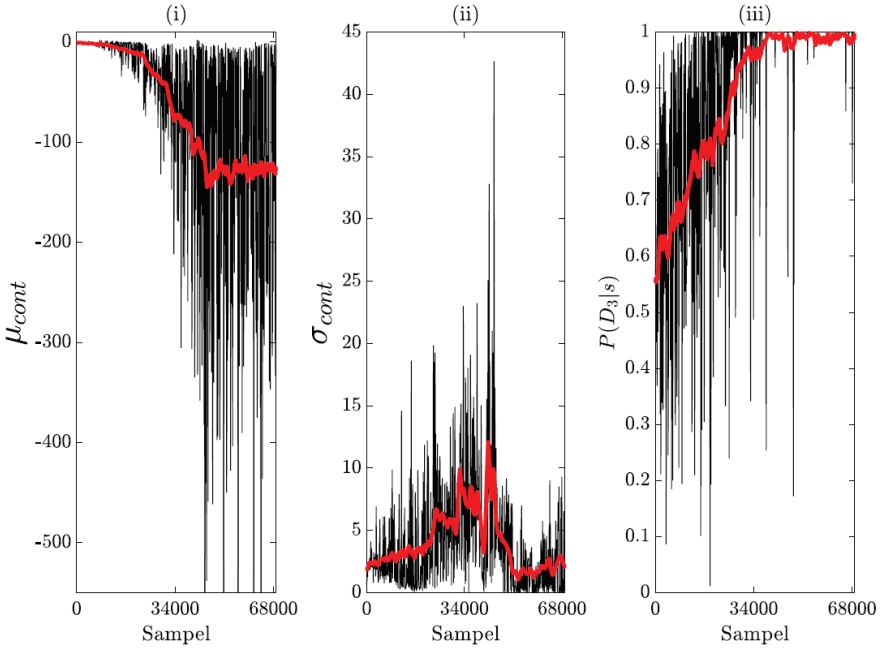
DRL-agenten tränades totalt 600 iterationer, vilket motsvarar 38 400 episoder av olika last- och störningsscenarier (och runt 68 900 individuella sampels). Utvecklingen för prestandan över tid presenteras i Figur 9. Den totala belöningen för respektive episod presenteras i delfigur (i) och den slutgiltiga resulterande SOL och antalet tidssteg för respektive episod presenteras i delfigur (ii) samt (iii). Den röda linjen visar ett centrerat glidande medelvärde beräknat som medelvärdet av resultatet från 500 datapunkter. För att bättre visualisera resultaten så illustreras endast var hundrade värde i figuren.

Resultaten visar att prestandan förbättrades snabbt efter att DRL-agenten tränats på runt 19 000 episoder, varefter styrpolicyn lyckades uppnå en SOL som låg över tröskelvärdet på 30 MW med bara ett enda tidssteg för en majoritet av alla episoder. I delfigur (ii) kan det observeras att prestandan fortsatte att förbättras även efter 19 000 episoder, men i en betydligt långsammare takt. Den huvudsakliga ökningen av prestandan efter 19 000 episoder är främst ett resultat av styrpolicyns minskade utforskningsgrad samt att optimera aktiveringsnivån för de olika handlingarna för respektive scenario.

I Figur 10 presenteras utvecklingen av de olika parametrarna som används för att definiera styrpolicyn. Parametern som styr standardavvikelsen σ_{cont} ökade kraftigt i början av träningen, samtidigt som medelvärdet μ_{cont} reducerades. Efter att modellen tränats på cirka 40 000 datapunkter så stabiliserades medelvärdet μ_{cont}, samtidigt som standardavvikelsen slutade att öka. Därefter förbättrades modellen främst genom att utforskningsgraden (som styrs av standardavvikelsen σ_{cont} samt av slumpmässigheten i de diskreta parametrarna $P(D_i)$ reducerades. I delfigur (iii) i Figur 10 illustreras sannolikheten för den vidtagna diskreta åtgärden D_3. Resultaten visar att styrpolicyn snabbt anpassade sig och kunde med en hög sannolikhet avgöra huruvida den diskreta handlingen skulle aktiveras eller ej. Liknande utveckling för de andra diskreta handlingarna observerades även i resultatet.



Figur 9. Prestanda och utveckling över träningsepisoder. Delfigurer som visar (i) total belöning per episod, (ii) slutgiltigt SOL, samt (iii) antal steg per episod. Den röda linjen visar ett glidande medelvärde beräknat över medelvärdet av 500 datapunkter. För bättre visualisering illustreras endast vart 100:e värde.



Figur 10. Utveckling av parametrar för styrpolicyn. Delfigurer visar (i) medelvärdet μ_{cont}, (ii) standardavvikelsen σ_{cont}, samt (iii) den diskreta sannolikheten för $P(D_3|s)$.

4.3.1 Framtagning av testset

Den tränade DRL-agenten testades därefter på tre olika testsets för att bland annat utvärdera dess kapacitet att hantera olika typer av scenarier som inte ingått i den träningsdata som genererats. Under träningen så använde aktörsnätverket en stokastisk styrpolicy som gjorde det möjligt för den att automatiskt utforska det tillgängliga handlingsutrymmet. När styrpolicyn implementeras är det generellt mer fördelaktigt att omvandla styrpolicyn till en deterministisk där alltid de handlingar som med högst sannolikhet är de optimala som alltid tas. Vid testresultaten för DRL agenten så styrdes därmed den kontinuerliga handlingen direkt av medelvärdet μ_{cont} och var och en av de diskreta handlingarna D_i aktiverades när någon av de definierade diskreta sannolikheterna uppfyllde $P(D_i|s) \geq 0{,}5$. De tre testseten beskrivs nedan:

- **Testset 1**: Data genererad på samma sätt som för använd träningsdata, men då en deterministisk styrpolicy används i stället för att utvärdera resultatet.
- **Testset 2**: Nya osedda driftpunkter introduceras genom att förändra variationen av generering- och belastningspunkter. I stället för att slumpmässigt variera varje belastning mellan 80 % - 120 % som för träningsdata, så varierades belastningarna mellan 70 % - 130 %. Återigen så användes en deterministisk styrpolicy för att utvärdera resultatet.
- **Testset 3**: Slumpmässiga mätfel introduceras i tillstånden. De simulerade mätfelen åstadkoms genom att varje tillståndsvärde multipliceras med ett slumpmässigt tal som genererats genom att sampla från en normalfördelning med ett medelvärde på 1 och en standardavvikelse på 0,01. Återigen så användes en deterministisk styrpolicy för att utvärdera resultatet.

4.3.2 Testresultat

Prestandan för DRL-agenten när den testas på de olika testseten presenteras i Tabell 5. Varje testset består av totalt 200 episoder. För testset 1 var det genomsnittliga antalet steg per episod 1,09, vilket indikerar att DRL-agenten lyckades säkerställa en tillräcklig säkerhetsmarginal (SOL) genom ett enda tidssteg för en majoritet av alla scenarier. Även för testset 2, där nya driftpunkter som ej ingått i den träningsdata som DRL-agenten tränats på inkluderades, visade DRL-agenten på god prestanda. Det genomsnittliga antalet steg per episod ökade något till 1,12, vilket indikerar att några av de osedda driftpunkterna resulterade i att DRL-agenten krävde ytterligare steg innan säkerhetsmarginalerna återställts. Den genomsnittliga belöningen för dessa testset var $-28{,}6$ respektive $-26{,}3$. För testset 3, där inverkan av slumpmässiga mätfel på tillståndsvärdena utvärderades, sjönk prestandan för DRL-agenten något. Den genomsnittliga belöningen minskade till $-34{,}7$ medan genomsnittliga antalet steg per episod ökade till 1,19 steg. Således krävde flera av scenarierna i testsetet att DRL-agenten utförde flera handlingssteg innan SOL återställdes över tröskelvärdet. Resultaten visar på vikten att inkludera slumpmässiga fel som finns i verkliga kraftsystem, men som i allmänhet inte finns med när metoden tränas på rent simulerade data. Det är således viktigt att inkludera sådana slumpmässiga fel även under träningsfasen, vilket skulle tvinga DRL-agenten till att bli mer robust med avseende på mindre slumpmässiga störningar i tillståndssignalen.

Tabell 5. Beräknad prestanda (medelvärde) för DRL-agenten på de tre olika testseten.

	Belöning	Steg per episod [steg]
Testset 1	-28,6	1,09
Testset 2	-26,3	1,12
Testset 3	-34,7	1,19

4.4 JÄMFÖRELSE MOT STYRSYSTEM BASERADE ENDAST PÅ DISKRET HANDLINGSRYMD

Den största fördelen med den föreslagna hybridbaserade DRL-arkitekturen är kapaciteten att kunna kontrollera både diskreta och kontinuerliga handlingsvariabler simultant. För att utvärdera påverkan av den funktionaliteten så jämförs den utvecklade styrningen mot två andra styrsystem som endast har funktionalitet att styra en diskret handlingsrymd: dels ett baserat på ett regelbaserat styrsystem, dels ett baserat på styrning som bygger på DRL, men som endast är anpassat för diskreta handlingsrymder. Nedan beskrivs de diskreta styrsystemen samt de anpassningar som utförts för att kunna jämföra prestandan mot den hybridbaserade DRL-arkitekturen.

4.4.1 Struktur för den regelbaserade styrningen

På grund av komplexiteten för systemoperatörer att utvärdera ett systems tillstånd direkt genom att studera rå mätdata så analyseras generellt systemet först genom att beräkna fram systemets säkerhetsmarginaler och, i de fall då säkerhetsmarginalerna understiger definierade tröskelvärden, så aktiveras handlingar för att säkerställa att säkerhetsmarginalerna kan återställas. Vilka handlingar som därefter aktiveras beror på tidigare fastställda regler, där lämpliga handlingar har definierats beroende på olika nivåer av de beräknade säkerhetsmarginalerna. Således kräver ett sådant regelbaserat styrsystem att man (1) först utvärderar säkerhetsmarginalen för en given driftpunkt, (2) eventuellt aktiverar handlingar för att återställa säkerhetsmarginalerna, samt slutligen (3) åter beräknar säkerhetsmarginalen för att bekräfta att de aktiverade handlingarna har varit effektiva och tillräckliga.

Eftersom en regelbaserad styrning kräver att säkerhetsmarginalen även beräknas *innan* några handlingar kan aktiveras, så behövs en negativ belöning läggas till för att kunna jämföra prestandan mot den hybridbaserade DRL-styrningen. I praktiken motsvarar förbehandlingen av mätdata och den initiala beräkningen av säkerhetsmarginalen SOL ett ytterligare beräkningssteg för den hybridbaserade DRL-styrningen. Således läggs en negativ belöning på -50 och ett ytterligare tidssteg för respektive episod till vid beräkningen av prestandan för den regelbaserade styrningen[4]. För testscenarion där den initierade säkerhetsmarginalen SOL beräknades till ett värde över 30 MW (vilket var det

[4] Den negativa belöning som lades till för varje ytterligare tidssteg DRL-agenten krävde för att uppnå ett säkert tillstånd enligt ekvation (4-4).

antagna tröskelvärdet) så behövdes inga handlingar aktiveras och således lades inte någon extra negativ belöning till för dessa scenarion.

Den regelbaserade styrningen kan för varje scenario välja olika handlingar beroende på den initialt estimerade säkerhetsmarginalen SOL. Sex olika handlingar (H1-H6) kunde aktiveras, där varje handling är specificerad i Tabell 6. Handlingarna styr lastbegränsning samt inkoppling av kapacitiva shuntar vilka motsvarar de kontinuerliga respektive diskreta handlingar som finns specificerade i del 4.2.2. Alla sex definierade handlingar är utformade så att de alltid återställer säkerhetsmarginalen över tröskelvärdet på 30 MW i ett enda steg.

Tabell 6. Diskreta handlingar för den regelbaserade styrningen samt den diskreta DRL-styrningen.

		Handlingar aktiverade	
Handlings-nummer	Initialt estimerad SOL [MW]	Lastbortkoppling [MW]	Inkoppling av kapacitiv shunt
H1	SOL < 30	0	Ingen
H2	0 < SOL ≤ 30	-50	D1
H3	-100 < SOL ≤ 0	-100	D1+D2
H4	-250 < SOL ≤ -100	-200	D1+D2+D3
H5	-350 < SOL ≤ -250	-350	D1+D2+D3
H6	SOL ≤ -350	-500	Samtliga

4.4.2 Struktur för den diskreta DRL-styrningen

Den diskreta DRL-styrningen bygger på samma struktur som den hybridbaserade DRL-styrningen, med den enda skillnaden att de tillgängliga handlingarna är begränsade till att endast vara diskreta. Den diskreta DRL-styrningen kan välja från samma handlingar (H1-H6) som finns definierade i Tabell 6, dock med skillnaden att den diskreta DRL-styrningen ej kräver en initial estimering av säkerhetsmarginalen SOL. Aktörsnätverket för den diskreta styrningen har en identisk arkitektur som den som används för hybridstyrningen, med skillnaden att endast en så kallad ”Softmax”-aktiveringsfunktion har använts i det slutgiltiga neuronlagret. Den använda Softmax-aktiveringsfunktionen normaliserar utgångsvärdena till en kategorisk sannolikhetsfördelning bestående av sex olika tal (motsvarande de diskreta handlingarna H1-H6) där varje tal motsvarar sannolikheten att aktivera någon av de diskreta handlingarna. Från den definierade sannolikhetsfördelningen kunde sedan olika diskreta handlingar samplas på samma sätt som olika kontinuerliga och diskreta handlingar kunde samplas från den styrpolicy som definierats i den hybridbaserade styrningen. Exakt samma hyperparametrar och träningsupplägg användes med enda skillnaden att parametrarna för inlärningshastigheterna minskats något för att undvika att den diskreta DRL-styrningen inte konvergerade för snabbt till en icke-optimal styrpolicy. Inlärningsparametrarna för aktörs- respektive kritikernätverket var $\alpha_{aktör} = 1 \cdot 10^{-5}$ samt $\alpha_{kritiker} = 5 \cdot 10^{-5}$.

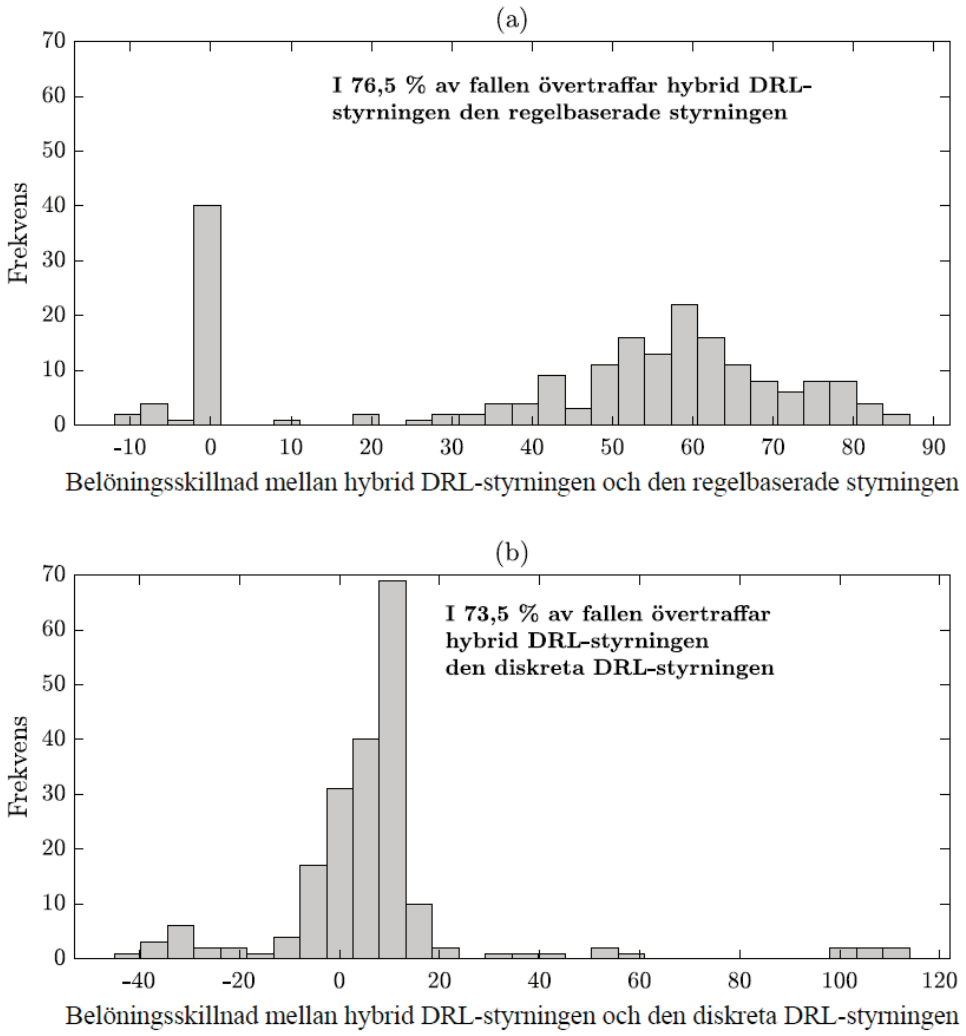
4.4.3 Resultat

I Tabell 7 så presenteras medelvärdet för den totala belöningen per episod samt antalet steg per episod för den regelbaserade styrningen och den diskreta DRL-styrning då dessa testats på de tre definierade testseten. Den procentuella skillnad i prestanda jämfört med när den hybridbaserade styrningen använts presenteras i parentes efter varje värde för respektive testset. Resultaten visar att den föreslagna hybridbaserade DRL-styrningen presterar signifikant bättre på samtliga definierade testset. Exempelvis ökar den negativa medelbelöningen från 109,0 % för testset 3, upp till 153,7 % för testset 1 när den regelbaserade styrningen användes. Då den diskreta DRL-styrningen användes så ökade den negativa medelbelöningen från 14,7 % för testset 3, upp till 25,2 % för testset 1.

Tabell 7. Beräknad prestanda (medelvärde) när en regelbaserad styrning samt en diskret DRL-styrning använts. Värden inom parentes motsvarar den procentuella skillnad i prestanda jämfört med när den hybridbaserade styrningen använts.

	Regelbaserad styrning		*Diskret DRL-styrning*	
	Belöning per episod	Steg per episod [antal]	Belöning per episod	Steg per episod [antal]
Testset 1	-72,6 (153,7 %)	1,77 (61,9 %)	-35,9 (25,2 %)	1,10 (1,4 %)
Testset 2	-62,3 (136,4 %)	1,65 (47,3 %)	-33,3 (26,4 %)	1,11 (-0,9 %)
Testset 3	-72,6 (109,0 %)	1,77 (47,7 %)	-39,9 (14,7 %)	1,16 (-3,3 %)

Antalet steg per episod var signifikant mycket högre när den regelbaserade styrningen användes, vilket beror på det ytterligare steg som krävs för att initialt beräkna säkerhetsmarginalen innan några handlingar kan aktiveras. Antalet steg per episod för den diskreta DRL-styrningen var i samma storleksordning som för den hybridbaserade för de olika testseten. Trots detta hade den hybridbaserade DRL-styrningen en signifikant lägre negativ medelbelöning per episod, vilket indikerar att kapaciteten att simultant kunna styra såväl diskreta som kontinuerliga handlingsvariabler oberoende resulterade i en mer effektiv styrpolicy. I Figur 11 så visas ett histogram där den skillnaden i belöning per episod mellan (a) den hybridbaserade DRL-styrningen och den regelbaserade styrningen samt (b) den hybridbaserade DRL-styrningen och diskreta DRL-styrningen presenteras för de olika scenarion som var inkluderade i testset 1. Resultaten visar att i 76,5 % respektive 73,5 % av alla scenarion så resulterar den hybridbaserade DRL-styrningen i en mer effektiv styrpolicy än de andra typerna av styrning.



Figur 11. Histogram som visar på skillnaden i total belöning per episod mellan (a) den hybridbaserade DRL-styrningen och den regelbaserade styrningen samt (b) den hybridbaserade DRL-styrningen och diskreta DRL-styrningen för de olika scenarion inkluderade i testset 1.

5 Akut styrning baserad på DRL

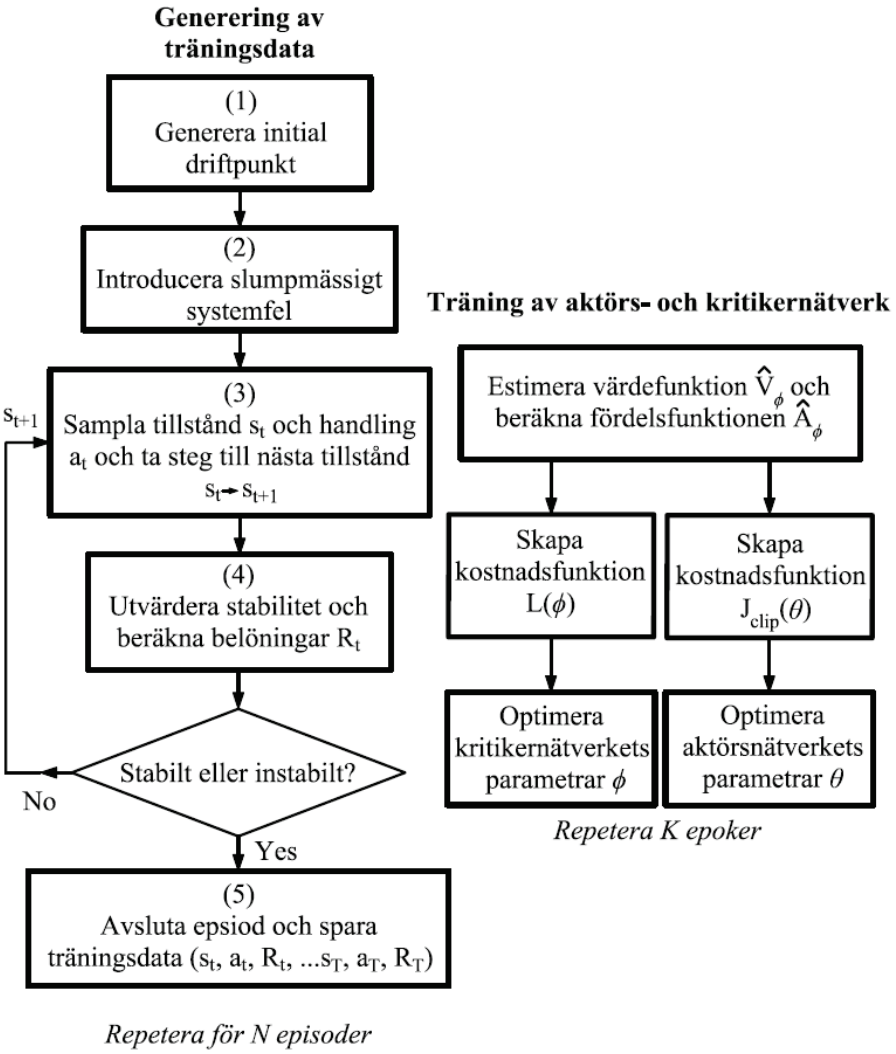
I följande del presenteras en metod för akut styrning baserad på DRL där styrproblemet är anpassat för att undvika långsam spänningsinstabilitet. Styrproblemet är ett högst icke-linjärt och icke-konvext optimeringsproblem där DRL agenten i varje tidssteg behöver kunna utvärdera systemets tillstånd och därefter välja den handling som mest effektivt kan stabilisera systemet till en lägsta totalkostnad för systemet. Komplexiteten ligger dels i att tolka systemets tillstånd, dels i att bestämma vilka handlingar som är mest optimala att vidta i det aktuella tillståndet. Metoden bygger på att DRL-agenten tränas på en stor mängd störnings- och belastningsscenarier. Genom att träna en DRL-agent på dessa scenarier så utvecklas en styrpolicy π_θ som kan bilda en funktion från tillståndet (som består av realtidsmätningar från elsystemet) till en handling. När DRL-agenten har tränats kan den i realtid föreslå optimerade handlingar till systemoperatören för att stabilisera och åtgärda långsam spänningsinstabilitet.

DRL-agenten är tränad för att utnyttja systemtjänster från efterfrågeflexibilitet och energilagringssystem som ett mer ekonomiskt och flexibelt alternativ för att stabilisera systemet än exempelvis tvingad lastbortkoppling (eng. *load shedding*). Studien undersöker specifikt förmågan hos DRL-agenten att hantera den osäkerhet som är involverad när sådana systemtjänster används (där exempelvis priset och tillgängligheten för tjänsten kan variera), som ett alternativ till konventionell lastbortkoppling. Slutligen utförs även en utvärdering av metodens robusthet och förmåga att hantera last- och störningsscenarier som inte har ingått i träningen av DRL-agenten för att kunna analysera hur väl dess styrpolicy generaliseras till osedda data. Eftersom antalet tillstånd och möjliga kombinationer med olika störningsscenarier är närmast oändliga i ett verkligt elsystem är en DRL-agents kapacitet att generalisera styrpolicyn en mycket viktig aspekt för att kunna implementera metoden rent praktiskt.

5.1 DATAGENERERING OCH METOD

En översikt av stegen involverade i genereringen av träningsdata och i träningen av DRL-agenten visas i Figur 12. De olika stegen i generering av träningsdata beskrivs i respektive del nedan.

(1) *Generera initialt drifttillstånd*: För Nordic32 testsystemet genererades de initiala drifttillstånden slumpmässigt runt den osäkra driftpunkten betecknad som ”driftpunkt A” i [10]. Alla laster i systemet varierades slumpmässigt och individuellt genom att multiplicera den aktiva lasten med en slumpmässig variabel genererad från en likformig sannolikhetsfördelning (med 95 % av den ursprungliga lasten som en nedre gräns, och 105 % av den ursprungliga lasten som en övre gräns). Effektfaktorn för alla laster hölls konstant. En lastflödesberäkning baserad på Newton-Raphson-metoden utfördes därefter där eventuella förändringar i den totala belastningen i systemet (som exempelvis förändringar i ledningsförluster) kompenserades för genom systemets slackbuss (eng. *slack bus*) generator g20, se Figur 4.



Figur 12. Flödesschema som visar genereringen av träningsdata och träningen av aktörs- och kritikernätverket.

(2) *Introducera slumpmässig störning*: Efter att den initiala driftpunkten skapats, så initierades en dynamisk simulering där en större störning introducerades. DRL-agenten tränades för att kunna hantera olika typer av störningar och med samma sannolikhet för varje scenario så kopplades antingen en ledning bort mellan systembussarna (i) 4032-4044, (ii) 4032-4042, (iii) 4031-4041, (iv) 4021-4042, eller så kopplades en av följande generatorer (v) g5, (vi) g7, bort från systemet. Störningarna valdes då de orsakar betydande påverkan på systemet i ”Central”-regionen och som utan lämpliga styråtgärder i de flesta belastningsscenarier leder till spänningsinstabilitet. I verkliga tillämpningar så bör alla störningar som har en potential att orsaka instabilitet utvärderas och inkluderas i träningen av DRL-agenten.

(3) *Sampla tillstånd s_t och handling a_t från styrpolicyn och ta steg till nästa tillstånd*: Tillståndet samlades därefter in från systemet och skickades till

aktörsnätverket. Tillståndet passerar genom aktörsnätverket vars utgångsparametrar används för att definiera den aktuella styrpolicyn $\pi_\theta(a|s)$ från vilken en handling sedan samplas från. Den samplade handlingen aktiveras sedan i systemet och den dynamiska simuleringen fortsatte att köras till nästa tidssteg, vilket bildar tillståndsövergången från $s_t \rightarrow s_{t+1}$. Tiden mellan varje steg i simuleringen var 5 sekunder. Tillstånden och handlingarna diskuteras vidare i respektive avsnitt nedan.

(4) *Utvärdera stabilitet och beräkna belöningar* R_t: Efter att den dynamiska simuleringen nått nästa tidssteg så utvärderades systemets stabilitet. Om någon transmissionsbusspänning V_{TS} understeg 0,7 per unit antogs systemet vara instabilt och episoden avslutades. Om någon V_{TS} vid slutet av den dynamiska simuleringen var under 0,9 per unit så antogs systemet också vara instabilt. Detta upplägg gör att systemspänningarna kan understiga 0,9 per unit under kortare tidsperioder, men att sådana låga systemspänningar inte är acceptabla i ett längre tidsperspektiv. Vid varje tidssteg beräknades också en belöning. Belöningsnivåerna utformades för att alltid motivera DRL-agenten att aktivera diverse handlingar snarare än att låta systemet bli instabilt, samtidigt som kostnaderna för handlingarna minimerades. Belöningen R_t vid varje tidssteg var en kombination av kostnaden för den vidtagna handlingen C_a, en mindre kostnad på -1 om någon V_{TS} var under 0,90, eller en större kostnad på -500 om systemet hade blivit instabilt. Kostnaderna för handlingarna C_a diskuteras vidare i senare avsnitt. Belöningen för varje tidssteg beräknades sedan som:

$$R_t = \begin{cases} C_a - 500 \cdot 0{,}99 & \text{om systemet är instabilt} \\ C_a & \text{annars om alla } V_{TS} \geq 0{,}90 \text{ per unit} \\ C_a - 1 \cdot 0{,}99 & \text{annars om någon } V_{TS} < 0{,}90 \text{ per unit} \end{cases} \quad (5\text{-}1)$$

Såväl den mindre kostnaden på -1 samt den större på -500 multiplicerades med en diskonteringsfaktor på 0,99, vilket resulterade i lägre negativa belöningar om instabilitet och låga systemspänningar inträffade senare snarare än tidigt i en episod. I denna studie antas samtliga belöningar vara utan någon enhet, men de bör i verkliga tillämpningar återspegla de faktiska monetära kostnaderna för olika åtgärder och belöningar när styrmålet antingen uppnås eller misslyckas.

(5) *Avsluta episod och spara träningsdata*: Samtliga episoder kördes maximalt i $T = 1000$ sekunder i den dynamiska simuleringen om inte systemet blev instabilt och avslutades i förväg. I slutet av varje episod så lagrades all data från simuleringen (tillstånd, handlingar, belöningar) och användes senare under träningen av aktörs- och kritikernätverken.

5.1.1 Tillstånd

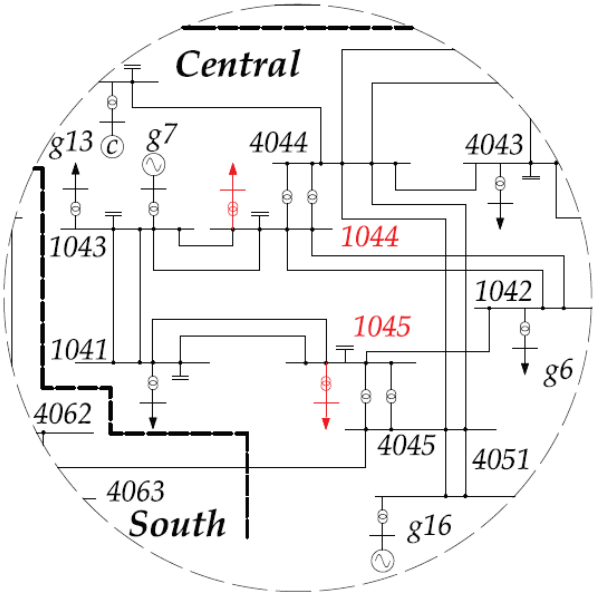
Tillstånden samlades in från mätningar tagna från den dynamiska simuleringen. Tillstånden bestod av en vektor av i) spänningsmagnituder från alla systembussar, ii) aktiva effektflöden mellan alla systembussar, och iii) reaktiva effektflöden mellan alla systembussar. För att även fånga systemets dynamik så användes även mätningar från det föregående tidssteget $t - 1$ som en del i tillståndsvektorn, vilket

därmed fördubblade tillståndsvektorns längd. För att stabilisera träningen så normaliserades tillståndsvektorernas värden genom att subtrahera medelvärdet för varje tillståndsvärde och sedan dividera de värdena med deras standardavvikelse. Medelvärdet och standardavvikelsen för varje tillståndsvärde beräknades från tidigare sanplade tillstånd och en lista med maximalt 10 000 samplade tillstånd lagrades. När totalt 10 000 samplade tillstånd lades till i listan så fastställdes medelvärdet och standardavvikelsen som användes för att normalisera tillstånden.

5.1.2 Handlingar

För att stabilisera systemet kan DRL-agenten ta olika handlingar för att minska belastningen i systemet. Handlingarna styr och utnyttjar systemtjänster av efterfrågeflexibilitet och energilagringssystem som kan tillhandahållas från aktörer på exempelvis en distributionsnätsnivå. Såväl efterfrågeflexibilitet samt styrning av energilagringssystem kan ses som en tillgänglig lastbegräsning som upphandlats genom en marknad för systemtjänster [15]. Genom att använda sådana tjänster är det möjligt för systemoperatörer att styra undan instabilitet i ett elsystem på ett liknande sätt som om mer klassisk förbrukningsbortkoppling hade använts. Den främsta skillnaden består av att dels en högre grad av flexibilitet i styrningen är tillgänglig där aktiveringsnivån av belastningsbegräsningen kan styras i betydligt mindre steg än vid förbrukningsbortkoppling, dels att påverkan på slutanvändare är betydligt mindre då medverkan i dessa systemtjänster är frivillig och bygger på monetära ersättningar för medverkan.

Styrningen av efterfrågeflexibiliteten och energilagringssystemet modelleras implicit genom att låta DRL-agenten justera lastnivåer vid två deltagande lastbussar inom ”Central”-regionen. De två lastbussarna som deltar i styrningen är placerade vid buss 1044 och buss 1045, se Figur 13. Tillgängligheten och priset på de marknadsbaserade systemtjänsterna fluktuerar vilket är en osäkerhet som behöver modelleras vid träningen av DRL-agenten. För att modellera detta varieras nivån av lastbegränsning som är tillgänglig på respektive av de två deltagande lastbussarna i varje störningsscenario. Kapaciteten för lastbegräsning bestäms genom att sampla från en slumpmässiga likformig sannolikhetsfördelning med en lägre nivå på 300 MW och en övre nivå på 500 MW. Dessutom varieras även priset för aktivering av lastbegräsningen mellan de två deltagande bussarna vilket uppnås genom att priset för lastbegräsningen för varje deltagande buss slumpmässigt varierades i början av varje störningsscenario. Priset för att aktivera lastbegräsning vid varje buss bestäms genom att sampla från en ny likformig sannolikhetsfördelning med en lägre kostnad på $-0{,}1$/MW och en övre nivå på $-0{,}2$/MW.



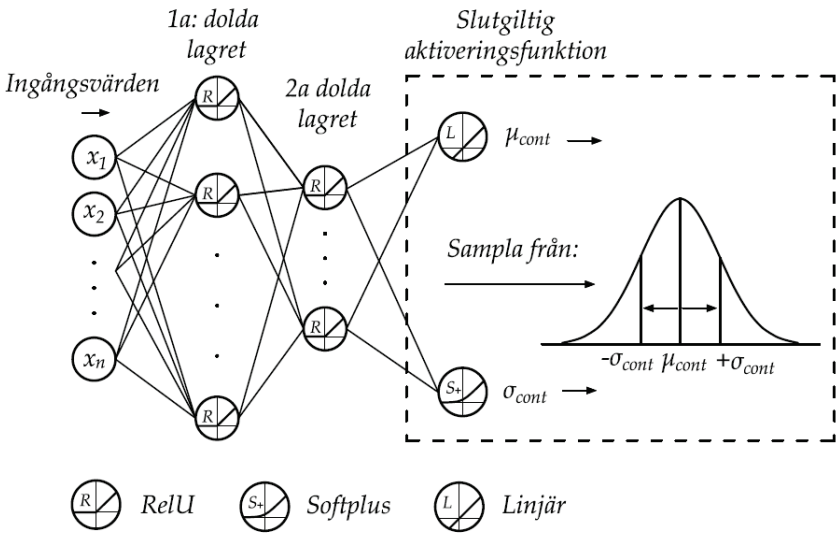
Figur 13. Inzoomat enlinjediagram för det modifierade Nordic32 systemet. Lastbussar 1044 och 1045 medverkar som styrbara laster i systemet.

DRL-agenten kontrollerar därefter *totalnivån* för den belastningsminskning som tas vid varje tidssteg. Den lastbuss som har lägst pris aktiveras först, men om den inte har tillräcklig kapacitet för att justera sin last så aktiveras även den andra lastbussen (med det högre priset för att justera ned lasten). Variationen i pris är tänkt att modellera ett marknadsbaserat system, där det satta priset för systemtjänsterna kommer att variera beroende på tillgänglighet. Det valda tillvägagångssättet säkerställer således att det finns en osäkerhet i *var* handlingarna kommer att aktiveras, till *vilken nivå* handlingarna är tillgängliga och även till *vilken kostnad* för systemet. Beroende på vilken buss som belastningsbegränsningen aktiveras först så kan det även påverka effektiviteten för handlingens kapacitet att motverka instabilitet, vilket är en ytterligare osäkerhet som DRL-agenten måste ta hänsyn till.

5.2 AKTÖRS- OCH KRITIKERNÄTVERK

Aktörsnätverket, illustrerat i Figur 14 och ytterligare detaljerat i Tabell 8, skapar funktionen som tillåter DRL-agenten att gå från det uppmätta tillståndet till styrpolicyn $\pi_\theta(a|s)$ från vilken handlingarna som ska aktiveras kan samplas från. Nätverket har två dolda lager följt av ett slutgiltigt aktiveringslager med två olika aktiveringsfunktioner som används för att estimera parametrarna som definierar styrpolicyn. Nätverket används för att estimera parametrar som i sin tur används för att definiera en normalfördelning $\mathcal{N}$. Normalfördelningen definierar därefter den stokastiska styrpolicyn som används för att sampla olika handlingar:

$$\pi_\theta(a|s) = \mathcal{N}\left(\mu_\theta(s), \sigma_\theta^2(s)\right) \tag{5-2}$$



Figur 14. Struktur för aktörsnätverket.

Normalfördelningen parametriseras av dels ett medelvärde μ_θ och en standardavvikelse σ_θ, där medelvärdet beräknas med hjälp av en linjär aktiveringsfunktion i det sista lagret, medan standardavvikelsen beräknas med hjälp av en softplus-aktiveringsfunktion som säkerställer att värdet aldrig blir negativt. Kritikernätverket är skilt från aktörsnätverket och består av ett enkelt neuralt nätverk med ett enda dolt lager och en linjär slutgiltig aktiveringsfunktion, mer detaljerad i Tabell 8.

Tabell 8. Design och hyperparametrar använda under träning och generering av träningsdata.

		Parametrar	Värden
Struktur för neurala nätverk	*Kritiker-nätverk*	Antal ingångsvärden/tillstånd	976
		Neuroner i gemensamma lagret	128
		Slutgiltig aktiveringsfunktion	Linjär
		Aktiveringsfunktion för dolda neuronlager	RelU
	Aktörnätverk	Antal ingångsvärden/tillstånd	976
		Neuroner i gemensamma lagret	64
		Neuroner i respektive separat dolt lager	32
		Slutgiltig aktiveringsfunktion för μ_{cont}	Linjär
		Slutgiltig aktiveringsfunktion för σ_{cont}	Softplus
		Aktiveringsfunktion för dolda neuronlager	RelU
	Träning	Antal epochs per träningsiteration (K)	5
		PPO klippningsparameter (ϵ)	0,2
		Batchstorlek (N)	64
		Optimeringsalgoritm	Adam [16]

5.3 TRÄNING AV AKTÖRS- OCH KRITIKERNÄTVERKEN

När totalt $N = 64$ episoder utförts och data insamlats så tränades aktörs- och kritikernätverken. Kritikernätverket används först för att beräkna värdet av $\widehat{V_{\phi}^{\pi}}$ för varje tillstånd och varje tidssteg, och därefter beräknades den uppskattade fördelsfunktionen för varje tillstånd med hjälp av ekvation (2-9). Målfunktionen som används för att träna aktörsnätverket beräknas med det beräknade medelvärdet av ekvation (2-10) för alla samplade värden och för samtliga N episoder. Målfunktionen som används för att träna kritikernätverket L_{ϕ} beräknades genom att ta medelkvadratfelet på alla δ-fel från ekvation (2-9) för alla samplade värden och för samtliga N episoder, och därefter beräkna medelvärdet av dessa värden. Träningen av båda nätverken utfördes med hjälp av programvaran Tensorflow i Python som automatiskt beräknar gradienterna på de definierade målfunktionenerna. Totalt utfördes 5 epochs av träningsalgoritmen på hela batchen av träningsdata. Värdena för inlärningsparametrarna och andra hyperparametrar som använts i träningen är specificerade i Tabell 8. När nätverken väl tränats på de insamlade data så genererades nya träningsdata och tidigare insamlade träningsdata från den tidigare styrpolicyn kasserades.

5.4 SIMULERINGAR OCH RESULTAT

DRL-agenten tränades totalt under 200 träningsiterationer, vilket motsvarar 12 800 individuella episoder, innan styrpolicyn konvergerade. Träningsprestandan presenteras i Figur 15 där den totala belöningen per episod och huruvida episoden resulterade i ett stabilt eller instabilt tillstånd, visas. Den röda linjen visar ett centrerat glidande medelvärde beräknat över 250 episoder. Resultaten i delfigur (i) visar att prestandan förbättrades snabbt fram till cirka 4 000 episoder, varefter styrpolicyn lyckades med målet att undvika instabilitet helt för samtliga episoder. Efter 4 000 episoder fortsatte prestandan att förbättras genom att i huvudsak optimera nivån för hur mycket handlingarna borde aktiveras i respektive last- och störningsscenario.

5.4.1 Framtagning av testsets

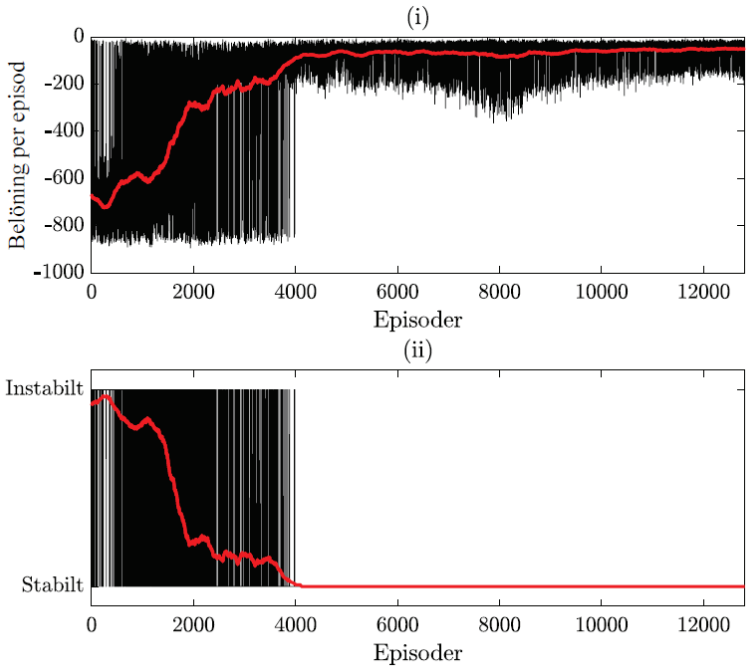
Under träningen använde DRL-agenten en stokastisk policy för att automatiskt kunna utforska det tillgängliga handlingsutrymmet. När DRL-agenten är färdigtränad är det dock generellt bättre att omvandla styrpolicyn till en deterministisk och alltid välja de handlingar som med högst sannolikhet är mest effektiva. Vid test av DRL-agenten så styrdes handlingarna i stället direkt av medelvärdet μ_{cont} som är en av parametrarna som aktörsnätverket beräknar. Den tränade DRL-agenten testades sedan på tre olika testset. Totalt 100 olika testscenarier beräknades för varje testset. De använda testseten består av:

- **Testset 1**: Data genererad på samma sätt som för använd träningsdata, men då en deterministisk styrpolicy används i stället för att utvärdera resultatet.
- **Testset 2**: Nya osedda driftpunkter introduceras genom att förändra variationen av generering- och belastningsnoder. I stället för att slumpmässigt variera varje belastning mellan 95 %–105 % som för träningsdata, så varierades

belastningarna mellan 90 % - 110 %. Återigen så användes en deterministisk styrpolicy för att utvärdera resultatet.

- **Testset 3**: Introduktion av nya osedda driftpunkter genom att introducera ett fel som inte tidigare använts vid träning av DRL-agenten. Den nya störningen är en bortkoppling av ledningen mellan bussarna 4011–4021. Återigen så användes en deterministisk styrpolicy för att utvärdera resultatet.

Prestandan för den utvecklade DRL-agenten jämfördes även med ett regelstyrt systemskydd baserad på lastbortkoppling. Lastbortkopplingen aktiveras direkt när *någon* transmissionsspänning i systemet understiger 0,9 per unit. I det fallet så tas totalt 100 MW last bort från systemet, lika fördelat mellan lasterna som finns på buss 1044 och 1045. För att möjliggöra en rättvis jämförelse mellan de två olika sätten at styra systemet så valdes kostnaden för aktivering av lastbortkopplingen (C_a) till -0,15/MW, vilket är medelvärdet av det varierande priset för aktivering av efterfrågeflexibilitet och styrning energilagringssystem som DRL-agenten styr.



Figur 15. Prestanda och utveckling över tid. Delfigurer visar (i) totala belöningar per episod, samt (ii) huruvida episoden slutade i ett stabilt eller instabilt tillstånd. Den röda linjen visar ett glidande medelvärde beräknat över 250 datapunkter.

5.4.2 Testresultat

Den genomsnittliga belöningen för respektive episod för de olika testseten presenteras i Tabell 9. I den sista kolumnen presenteras den relativa skillnad i belöningen mellan då DRL-styrningen används och då lastbortkopplingsstyrningen användes. Resultaten visar att DRL-agenten lyckades att få en signifikant lägre negativ genomsnittlig belöning jämfört med då lastbortkoppling användes för alla de olika definierade testseten. Till exempel, för testset 1 så resulterade den lastbortkopplingen i en 128,6 % högre negativ belöning jämfört med när DRL-styrningen användes. Även för testset 2 och 3, som DRL-agenten inte specifikt tränats på, lyckades DRL-agenten att generalisera den

inlärda styrpolicyn och uppnå en betydligt effektivare styrning än för den klassiska lastbortkopplingen. Förbättringen var dock väsentligen mindre märkbar på testset 3 (33,5 %) där en ny störning som inte hade ingått i den träningsdata som använts för att träna DRL-agenten utvärderades. Det bör även noteras att i samtliga testscenarior för alla de definierade testseten så lyckads både DRL-styrningen och den klassiska lastbortkopplingen att stabilisera alla fall av spänningsinstabilitet.

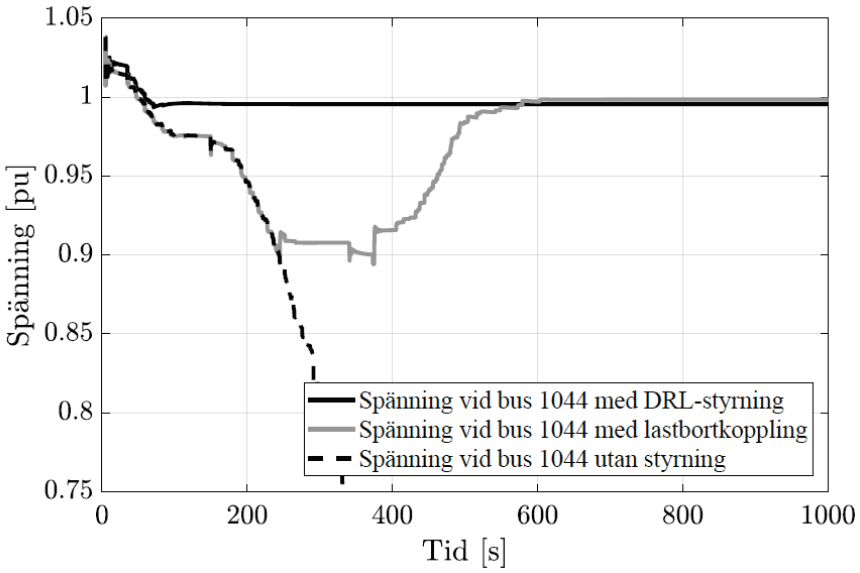
Tabell 9. Genomsnittlig prestanda för olika testset och styrmetoder.

	Genomsnittlig belöning per episod		**Differens**
	DRL-styrning	*Lastbortkoppling*	*[%]*
Testset 1	-37,9	-86,7	128,6 %
Testset 2	-36,5	-114,2	213,9 %
Testset 3	-16,7	-22,3	33,5 %

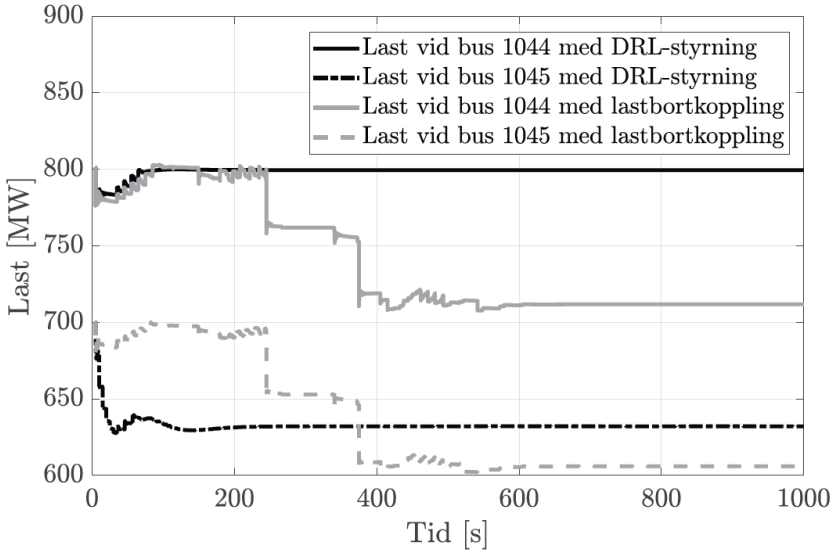
I Tabell 10 presenteras den genomsnittliga mängden av bortkopplad aktiv last som de två olika styrmetoderna krävde för respektive testset. Måttet representerar alltså hur mycket belastning varje styrmetod behövde koppla bort innan systemet stabiliserades. Återigen så presenteras den relativa skillnaden mellan de två styrsystemeter i den sista kolumnen i tabellen. Resultaten visar att DRL-agenten krävde *betydligt* mindre belastningsminskning i systemet för att stabilisera det i jämförelse med den regelbaserade lastbortkopplingen. Exempelvis så krävde den regelbaserade lastbortkopplingen 259,4 % mer bortkopplad aktiv last i genomsnitt mot vad DRL-styrningen krävde på samma testset. Skillnaderna mellan DRL-styrningen och den regelbaserade lastbortkopplingen exemplifieras i Figur 16 och i Figur 17. I Figur 16 visas spänningsmagnituden för buss 1041 för ett testscenario från testset 1. Spänningsmagnituden över tid presenteras för fallen då i) DRL-styrningen används, ii) när den regelbaserade lastbortkopplingen används, och iii) då ingen styrning används. I Figur 17 visas även lasten vid de styrda lastbussarna 1044 och 1045, vilket visar skillnaden i hur mycket bortkopplad aktiv last som krävds vid de olika styrmetoderna.

Tabell 10. Genomsnittlig bortkopplad last för olika testset och styrmetoder.

	Genomsnittlig bortkopplad aktiv last		**Differens**
	DRL-styrning [MW]	*Lastbortkoppling [MW]*	*[%]*
Testset 1	190,9	560	193,4 %
Testset 2	166,1	597	259,4 %
Testset 3	124,5	144	15,7 %



Figur 16. Spänningsmagnitud vid buss 1041 över tid för olika styrmetoder.



Figur 17. Last vid buss 1044 och buss 1045 när DRL-styrningen används och för fallet då den regelbaserade lastbortkopplingen används.

För det givna scenariot kommer systemet att kollapsa efter cirka 330 sekunder om ingen styrning initieras. Då en regelstyrd lastbortkoppling används kommer totalt 200 MW last behöva kopplas bort från systemet för att stabilisera det. Lastbortkopplingen aktiveras dels en gång vid cirka 250 sekunder, dels sker en ytterligare aktivering vid cirka 380 sekunder in i den dynamiska simuleringen. Aktiveringen av lastbortkopplingen kan ses från de relativt stora steg i lastförändring som sker i Figur 17. Efter den andra aktiveringen av lastbortkoppling börjar systemspänningarna i systemet att återställas vilket kan ses i Figur 16. När DRL-styrningen används så aktiveras lastreduceringen *direkt* efter störningen vilket gör att styrningen kan aktiveras utan att systemspänningarna

behöver försämras först. Belastningen vid buss 1045 reduceras med totalt cirka 130 MW, varefter systemet stabiliseras. DRL-styrningen lyckas även uppnå en bättre spänningsprofil efter störningen, där spänningsmagnituden behålls närmare den nominella nivå som fanns före störningen. Således lyckas DRL-styrningen med att dels aktivera en mindre mängd lastreducering, samtidigt som den uppnår en både snabbare och mer effektiv styrning för det givna scenariot.

5.4.3 Prestanda då aktiveringsgränser är implementerade

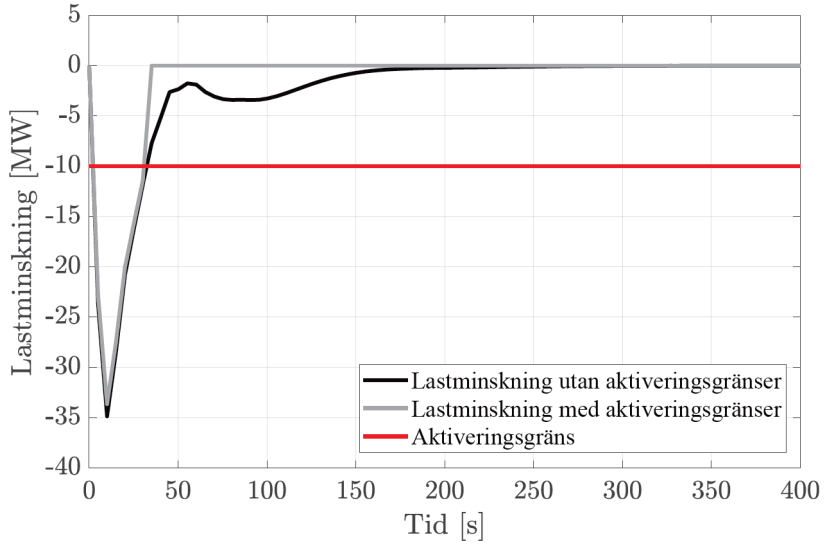
En av fördelarna med DRL-styrningen är att nivån av bortkopplad aktiv last kan styras mer flexibelt, vilket kan jämföras vid den regelstyrda lastbortkopplingen som generellt kopplar bort en större del last vid en och samma aktivering. Samtidigt som DRL-agenten är tränad till att minimera hur mycket aktiv last som kopplas bort från systemet så är det svårt att träna handlingarna (som styrs av medelvärdet μ_{cont}) att helt konvergera till noll då systemet väl har stabiliserats. Därutöver ger en utvärdering av DRL-agentens prestanda på stabila störningsscenarier (alltså för drift- och störningsscenarier som skulle resultera i ett stabilt tillstånd även om inga handlingar aktiverats i systemet), att DRL-agenten (alltså i onödan) aktiverade en låg nivå av handlingarna. Orsaken till detta beteende kan främst förklaras av det faktum att DRL-agenten tränats på en majoritet av drift- och störningsscenarier som var osäkra, vilket kan ses genom att notera andelen instabila episoder i början av träningen givet av den röda linjen i delfigur (ii) i Figur 15.

Att undvika onödig aktivering av lastbegränsning kommer att vara viktigt om DRL-styrningen ska kunna ingå verkliga styrsystem. Den primära lösningen på problemet är att träna DRL-agenten på en större andel stabila scenarier för att göra den mer robust och för att träna styrpolicyn på att bättre hantera den typen av scenarier. Detta skulle dock kunna kombineras med en tröskel för handlingsaktivering för att säkerställa att endast relativt signifikanta handlingar aktiveras i systemet. För att testa den här funktionen tillämpades en aktiveringströskel på 10 MW för DRL-agenten. Således resulterade varje tagen handling från DRL-agenten med en lägre magnitud än 10 MW i att ingen belastningsreducering aktiverades, medan varje handling som var större eller lika med 10 MW aktiverades.

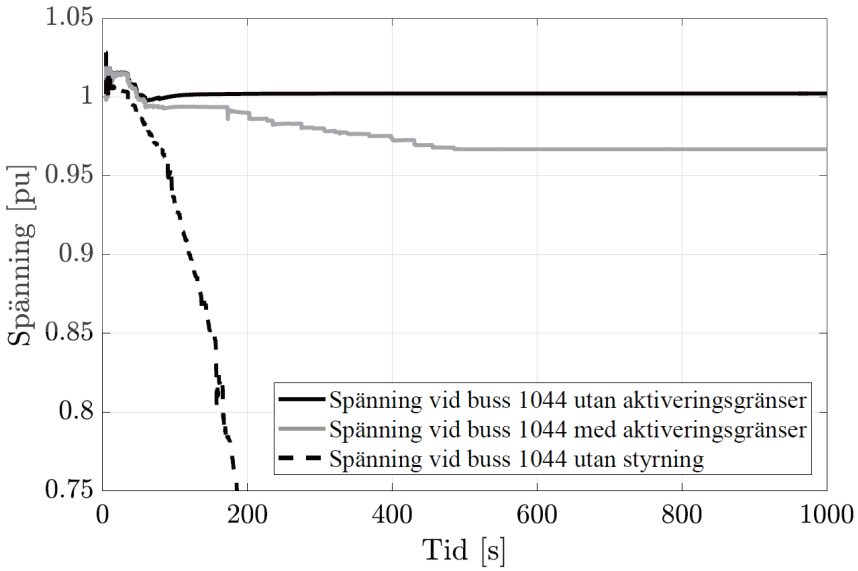
Prestandan då tröskelvärdet för handlingsaktiveringen användes testades på samma testset som utvecklades i del 5.4.1. Resultaten presenteras i Tabell 11 där den genomsnittliga belöningen per episod och den genomsnittliga belastningsbegränsningen presenteras då tröskelvärden för handlingsaktivering användes. Den procentuella skillnaden i prestanda jämfört med fallet då *inget* tröskelvärde användes (men fortfarande då DRL-styrningen används) visas inom parentes efter varje värde. Skillnaden var mest signifikant för testset 3, där den genomsnittliga mängden bortkopplad aktiv last kunde reduceras med -82,1%. Den genomsnittliga belöningen per episod försämrades dock för test 1 och test 2, där den (negativa) genomsnittliga belöningen per episod ökade med 32,7 % respektive 13,2 %. De högre negativa belöningarna orsakades av att systemspänningar förblev under 0,9 per unit under längre tid efter störning varje, vilket resulterade i en högre negativ belöning för dessa scenarier.

Tabell 11. Genomsnittlig prestanda och bortkopplad aktiv last då tröskelvärde för handlingsaktivering används.

	Genomsnittlig belöning per episod	Genomsnittlig bortkopplad aktiv last *[MW]*
Testset 1	-50,3 (32,7 %)	171,6 (-10,1 %)
Testset 2	-41,3 (13,2 %)	114,2 (-29,1 %)
Testset 3	-12,4 (-25,8 %)	22,3 (-82,1 %)



Figur 18. Last vid buss 1044 och buss 1045 när DRL-styrningen används och för fallet då den regelbaserade lastbortkopplingen används.



Figur 19. Last vid buss 1044 och buss 1045 när DRL-styrningen används och för fallet då den regelbaserade lastbortkopplingen används.

6 Förutsättningar för användning av DRL på distributionsnätsnivå

I följande del analyseras möjligheterna att anpassa utvecklade DRL-metoder för användning för styrning av olika ändamål på en distributionsnätsnivå. Skillnader i krav på mätinfrastruktur, olika typer av styråtgärder, samt en genomgång av hur datagenerering kan utföras analyseras. Därutöver kommer två fallstudier att analyseras närmare och tidigare studier inom ämnet kommer att kortfattat presenteras.

6.1 MÄTINFRASTRUKTUR OCH FRAMTIDA FUNKTIONSKRAV

DRL kräver generellt stora mängder mätdata och för att få effektiva styralgoritmer krävs det att tillgången av mätdata samt dess kvalitet är tillräcklig. Mätinfrastrukturen som traditionellt funnits tillgänglig på distributionsnätsnivå skiljer sig väsentligt från den som finns tillgänglig på stamnätsnivå. Skillnader i form av mätnoggrannhet, typ av mätstorheter som kunnat uppmätas, samt kommunikationshastighet för mätvärden, har begränsat möjligheterna att använda maskininlärningsbaserade metoder för vissa applikationer på en distributionsnätsnivå.

Tillgången på mätdata på distributionsnätsnivån har dock redan och kommer ytterligare att förbättras i framtiden. 2014 fick Energimarknadsinspektionen (Ei) i uppdrag av regeringen att ta fram nya rekommendationer för vilka funktionskrav som ska finnas på elmätare i framtiden. Uppdraget resulterade 2017 i ett författningsförslag [17] från Ei kring nya funktionskrav, och 2018 beslutade regeringen om nya funktionskrav för elmätare [18]. De nya funktionskraven inkluderar [19]:

- Elmätaren ska kunna mäta både uttag och inmatning i varje fas av ström, aktiv effekt och reaktiv effekt, mäta spänningen i varje fas samt mäta och registrera den totala aktiva energin vid uttag och inmatning av el.
- Elmätaren ska utrustas med ett kundgränssnitt som stöds av en öppen standard som möjliggör för kunden att ta del av mätuppgifterna i nära realtid.
- Elmätaren ska möjliggöra avläsning av mätdata och uppgifter om elavbrott på distans.
- Elmätaren ska kunna registrera mängden överförd energi per 15 minuter.
- Elmätaren ska kunna registrera uppgifter om tidpunkt för början och slut på elavbrott längre än tre minuter.
- Det ska vara möjligt för elnätsföretaget att uppgradera och ändra inställningar i elmätaren på distans.
- Det ska vara möjligt för elnätsföretaget att via elmätaren kunna spänningssätta och frånkoppla elanläggningar på distans.

En majoritet av funktionskraven måste vara implementerade senaste den 1 januari 2025. Mätinfrastruktur inom övriga delar på distributionsnätssidan, som exempelvis nätstationsmätningar, är inte reglerad, utan har främst styrts av nätägarens behov och nytta av ytterligare mätinfrastruktur. Tidigare studier har

visat att andelen nätstationsmätningar varierar mellan olika distributionsnätsägare, men att den totala andelen nätstationer som är uppmätta förväntas öka i framtiden [20]. Kvaliteten och omfattningen på nätstationsmätningarna skiljer sig, där både enklare elmätare samt mätningar med elkvalitetsinstrument har använts till att utföra nätstationsmätningar.

En framtida förbättrad mätinfrastruktur förenklar möjligheterna att använda DRL-baserade metoder även på en distributionsnätsnivå. Vilken typ av mätinfrastruktur som krävs är till stor del beroende på vilken typ av styralgoritm som önskas implementeras/testas, vilket analyseras mer i fallstudierna senare i kapitlet.

6.2 TILLGÄNGLIGA STYRÅTGÄRDER

Inom distributionsnät finns andra typer av komponenter som är styrbara jämfört med på en stamnätsnivå. Nedan exemplifieras ett antal komponenter och system som hade kunnat vara lämpliga för styrning av DRL-baserade metoder:

- Styrning av aktiv och reaktiv effekt från lokala produktionssystem
- Styrning av lindningsomkopplare på transformatorer
- Frånkoppling av laster
- Styrning av lastefterfrågeflexibilitet
- Styrning av reaktiva shuntar
- Styrning av batterisystem / energilagring

Vilken komponent/system som lämpar sig bäst för styrning är även det beroende av vilken funktion som önskas uppnås. I följande delar analyseras två fallstudier närmare där styråtgärder och dess funktionalitet analyseras närmare.

6.3 FALLSTUDIE 1: STYRNING AV RESURSER INOM EFTERFRÅGEFLEXIBILITET

I följande sektion undersöks möjligheten att använda DRL som en metod för styrning av resurser inom efterfrågeflexibilitet. Efterfrågeflexibilitet är en frivillig ändring av efterfrågad elektricitet från elnätet under en kortare eller längre period som sker genom att elanvändningen styrs temporärt. Genom att tillåta sin elanvändning att styras så får användaren sedan en viss monetär ersättning. Styrningen kan antingen ske *indirekt* genom att kundens användningsmönster påverkas, eller *direkt* genom att utrustning automatiskt reagerar på olika (styr)signaler. Den typ av efterfrågeflexibilitet som är relevant för användning inom DRL är direkt styrning, där DRL-agenten analyserar systemets behov och skickar styrsignaler som används för att styra laster inom systemet. Typisk utrustning som lämpar sig för direkt styrning är exempelvis värmepumpar samt laddstationer för elbilar.

6.3.1 Flexibilitetsbehov och styrmål

I kraftnätet finns olika flexibilitetsbehov, där skillnader i påverkan på nätegenskap, orsak till att behovet uppstår, samt tidsskalan då behovet finns varierar. I [21] sammanfattades kraftsystemets flexibilitetsbehov i de fyra flexibilitetsbehoven

balanstjänster, *balansreglering*, *spänningsreglering* och *nätkapacitet*. Dessa begrepp förklaras kortfattat nedan [21]:

- **Balanstjänster:** Används för att säkerställa kraftbalansen där jämnvikten mellan elektrisk produktion och elektrisk efterfrågan i varje tidsinstans måste upprätthållas. Balanstjänster kan även användas för att hantera frekvensproblem då en minskade mängd rotationsenergi i kraftsystemet minskar den naturliga trögheten mot störningar i frekvens inom elsystemet. Tidshorisonten för balanstjänster sträcker sig från delar av en sekund till upp till en timma.
- **Balansreglering:** Balansreglering refererar till liknande behov som för balanstjänster, men där tidshorisonten är betydligt längre (timmar upp till ett år). Här avses snarare planerbara behov av elenergi där variationer i säsongsberoende elektrisk produktion skapar utmaningar.
- **Spänningsreglering:** Spänningsreglering (eng. *Volt-Var control*) refererar till bibehållandet av spänningar som hålls inom reglerade nivåer vilket är viktigt för upprätthållandet av elkvaliteten och för att elektriska komponenter ej ska skadas eller få sämre prestanda. En ökad problematik kring spänningsreglering kan resultera från den kraftiga utbyggnad av decentraliserad förnybar produktion som sker på lokalnätsnivå.
- **Nätkapacitet:** Nätkapacitet refererar till nätets förutsättningar att hantera och överföra den elektricitet från den plats den produceras till den plats där den konsumeras. Ett elnät begränsas ofta av den nätkapacitet som finns tillgänglig, vilket kan resultera i problem vid ökat uttag av effekt. På distributionsnät är ofta nätkapaciteten beroende på den fysiska dimensioneringen av ledningar och transformatorer, där exempelvis termiska begränsningar avgör möjligheterna för ökad effektkapacitet.

Det bör noteras att styrning av efterfrågeflexibilitet för ett enskilt definierat flexibilitetsmål i vissa fall kan motverka andra flexibilitetsmål. Exempelvis kan en styralgoritm anpassad för att minska påverkan på ett elnäts lokala nätkapacitet innebära att andra mål, som exempelvis balansreglering, påverkas negativt. Flera simultana styrmål kan inkluderas i DRL-algoritmer, där olika behov kan viktas genom att anpassa de belöningar algoritmen mottar. Det bör noteras att ju mer komplicerade styrmål som definieras desto större mängd träningsdata kan förväntas behövas för att uppnå en effektiv styrning.

6.3.2 Definition av Markoviansk beslutsprocess för efterfrågeflexibilitet

I följande del diskuteras hur en MDP kan definieras för styrning av efterfrågeflexibilitet.

Tillstånd: Systemets tillstånd ska med fördel representera all tillgänglig information som krävs för att kunna ta beslut om vilken handling/styrsignal som är optimal vid varje tidpunkt. Vilken typ av tillstånd som krävs är beroende dels på vilken typ av laster som styrs, samt dels vilken typ av flexibilitetsbehov som efterfrågeflexibiliteten har anpassats för. Används laster som exempelvis luftvärmepumpar så kan exempelvis inomhustemperaturer vara viktiga tillstånd då komforten ej får försämras i för hög grad. Om i stället laddning av elbilar används är andra parametrar som aktuell batterinivå eller tid till nästa användning

viktig information som DRL-agenten kommer att behöva ha tillgång till. Därutöver kommer annan systeminformation och mätningar i elnätet att vara viktiga tillstånd, där det aktuella flexibilitetsbehovet kommer att styra vilka tillstånd som krävs. Är exempelvis målet att undvika lokala problem med nätkapacitet kan bland annat lokala nätstationsmätningar samt spänning och effekter vid lastpunkter att vara viktigt att inkludera i MDP:ns tillstånd. I de fall då tidsdynamik kan tänkas vara viktigt för styrningen kan det även krävas att historiska mätningar inkluderas i systemets tillstånd.

Handlingar/styrsignaler: De handlingar som DRL-agenten kan ta kan bestå av diskreta eller stegvisa handlingar som används för att exempelvis stänga av/på laster eller exempelvis välja nivåer på lindningsomkopplingar. En annan möjlighet är att definiera kontinuerliga handlingar, där exempelvis olika nivåer av styrning på batterisystem i mer detaljnivå kan användas. Generellt är binära och stegvisa handlingar enklare att använda och kräver generellt mindre träningsdata än kontinuerliga handlingar.

Belöningar: De belöningar som DRL-agenten mottar bör med fördel representera ett uppskattat värde för den monetära kostnad/inkomst som diverse handlingar uppnår. Det bör noteras att vissa belöningar är enklare än andra att uttrycka i monetära termer. Att exempelvis flytta över elförbrukning till timmar med låga elpriser, eller att undvika effektuttag då effekttaxor hade gjort förbrukningen kostsam, bör vara relativt enkelt att uppskatta det monetära värdet för. Mer problematiskt är det exempelvis att, i monetära termer, beskriva den kostnad som en slutkund upplever då dess elförbrukning flyttas från en tidpunkt till en annan.

Dynamik för tillståndsövergångar och diskonteringsfaktor: Dynamik för tillståndsövergångar kan i ett elsystem generellt antas vara deterministiska, där exempelvis en minskning i ett effektuttag alltid kommer att leda till en viss påverkan på den tillgängliga nätkapaciteten. Stokasticitet och osäkerheter ligger i stället i systemets tillstånd (exempelvis mätfel). Vilken diskonteringsfaktor som bör användas är i hög grad beroende på det definierade styrmålet, men det är rimligt att en viss diskonteringsfaktor används för att ge incitament till DRL-algoritmen att uppnå det definierade styrmålet så snabbt som möjligt.

6.3.3 Typ av DRL-algoritm och datagenerering

Vilken typ av DRL-algoritm som bör användas är beroende på hur styrningen definieras, samt vilka typer av handlingar som önskas användas. Det är generellt problematiskt att förutspå vilken typ av algoritm som är mest effektiv och det krävs ofta att olika DRL-algoritmer testas för att se vilken som klarar av det definierade styrproblemet mest effektivt. Generellt kräver ”on-policy” metoder (se kapitel 2) mer träningsdata än ”off-policy” metoder. Samtidigt är många ”off-policy”-metoder begränsade till att endast styra diskreta eller stegvisa handlingar, vilket begränsar deras applicerbarhet till mer komplicerade problem.

DRL-algoritmen kan tränas antingen genom direkt interaktion i det faktiska systemet, eller genom att träningsdata genereras i olika typer av simuleringssystem eller med modeller, där olika styrscenarion testas och där den utvecklade DRL-agenten tränas. Med stor sannolikhet krävs det att DRL-

algoritmen tränas på åtminstone någon form av simulerade träningsdata initialt, då dess styrpolicy behöver nå en åtminstone acceptabel nivå innan den kan appliceras i systemet.

Slutligen bör det noteras att det inte är säkert att det krävs djupinlärning/DRL för att dessa algoritmer ska bli effektiva då antalet tillstånd som finns definierade i MDP:n med stor sannolikhet är betydligt färre än de som krävs då dynamiska förhållanden på en stamnätsnivå analyseras. Huruvida det är möjligt att använda sig av mer konventionell förstärkningsinlärning utan djupinlärning behöver dock utredas vidare.

6.3.4 Översikt kring tidigare studier

I följande del ges en kortare översikt över tidigare studier där förstärkningsinlärning har testats för styrning av efterfrågeflexibilitet. Syftet med översikten är endast att exemplifiera vilka algoritmer och begräsningar som tidigare studier har använt sig av och är därmed inte menad att fullständigt redogöra för den litteratur som finns tillgänglig inom området. I [22] definierades styrning av energilagring i form av batteri där Q-learning algoritmen användes för att välja bland tre olika laddning/urladdningsnivåer. I [23] studerades återigen styrning av energilagring i form av batterier där en DRL-algoritm användes för att optimera laddning/urladdning samtidigt som laddningscyklernas påverkan på batteriets prestanda och livslängd inkluderades. I [24] presenterades en DRL-metod med en struktur baserad på så kallad ”dueling deep Q network” (DDQN) där laster modellerade med efterfrågeflexibilitet styrdes både med målet att hålla spänningar inom förutbestämda gränser samt för att minimera den totala driftkostnaden i systemet. Den föreslagna DDQN-strukturen jämförs mot en styrning baserade på en mer klassisk (djup) Q-learning algoritm, där DDQN-strukturen visar sig konvergera snabbare och med en betydligt lägre varians. Användning av DRL är inte alltid nödvändigt för samtliga styrproblem och i [25] modelleras termostatiska laster som kontrolleras av en mer klassisk RL-algoritm kallad ”modified Q-iteration”. Den RL-baserade styralgoritmen styrde inomhustemperaturen, samtidigt som ett mer klassiskt reglersystem verifierade att komforten ej försämrades för mycket.

6.4 FALLSTUDIE 2: SPÄNNINGSREGLERING (VOLT-VAR STYRNING)

I följande sektion undersöks möjligheten att använda DRL som en metod för styrning av resurser inom spänningsreglering på en distributionsnätsnivå. Med en ökad mängd lokal elproduktion (från exempelvis solpaneler) ökar problematiken att hålla spänningar inom definierade nominella gränser. På senare år har en även ökad kablifiering av luftledningar på lokalnäten inneburit en ökad reaktiv effektgenerering, vilket i sin tur påverkar spänningsnivåer i näten. Spänningsvariationer kan kraftigt påverka elkvaliteten hos slutkunder och såväl överspänningar som underspänningar kan resultera i både försämrad funktion samt i värsta fall skador på elektriska komponenter.

För att bibehålla spänningar inom reglerade gränser utför distributionsnätsägare spänningsreglering, där styrbara komponenter som lindningsomkopplare och

inkopplingsbara shuntkompensatorer styrs. I en framtid med en kraftigt utbyggd lokal elproduktion, kan det även komma att bli aktuellt att aktivt styra ner lokal elproduktion för att hantera lokala spänningstoppar. Redan i dag finns krav på lokala elproduktionsanläggningar för inbyggd automatik att stänga ner produktionen då den lokala spänningen blir för hög. Men genom en mer aktiv styrning, och ett potentiellt krav på växelriktare att även kunna leverera eller absorbera reaktiv effekt, hade en mer effektiv styrning kunnat åstadkommas. Definitionen av MDP:n som styrsystemet arbetar i finns i större nivå detaljerat i delen nedan.

6.4.1 Definition av Markoviansk beslutsprocess för spänningsreglering

I följande del diskuteras hur en MPD kan definieras för styrning av efterfrågeflexibilitet.

Tillstånd: Systemets tillstånd ska representera all tillgänglig information som krävs för att kunna ta beslut om vilken handling / styrsignal som är optimal vid varje tidpunkt. Typiska mätdata som generellt alltid bör inkluderas i tillståndet är spänningsmagnituder från slutkunder och eventuella nätstationsmätningar, då dessa parametrar direkt påverkar styrmålet. Exempel på andra mätningar och parametrar kan inkludera:

- Aktuellt steg på lindningsomkopplare: ger styragenten information om hur många steg som finns tillgängliga att styra
- Aktuellt elpris: kan användas av styragenten för att estimera alternativkostnaden om exempelvis lokal elproduktion styrs ned.
- Aktiv och reaktiv effektmätning: kan exempelvis hjälpa styragenten att estimera hur mycket spänningsförändringarna vid varje handling kommer att bidra med.

Handlingar/styrsignaler: Vilka handlingar som DRL-agenten kan ta beror på hur avancerad styrning som önskas. I det enklaste fallet kan en styrning av lindningsomkopplare användas, medan mer avancerade applikationer skulle kunna vara simultan reglering av lokala produktionskällor och andra mer traditionella komponenter som lindningsomkopplare och inkoppling av shuntkompensatorer.

Belöningar: De belöningar som DRL-agenten mottar bör med fördel representera den monetära kostnad/belöning som diverse handlingar faktiskt lyckas uppnå. Återigen kommer vissa belöningar vara enklare än andra att uttrycka i monetära termer. Exempelvis är det relativt enkelt att uppskatta den faktiska kostnaden av en eventuell nedstyrning av lokala produktionskällor. Mer problematiskt är det att estimera den kostnad som det mekaniska slitage som styrning av en lindningsomkopplare innebär eller hur en kortvarig överträdelse av en spänningsgräns bör bestraffas. För att hitta lämpliga nivåer på belöningssystemet krävs det troligen att viktningar utvärderas löpande när DRL-algoritmen tränas.

Dynamik för tillståndsövergångar och diskonteringsfaktor: Dynamik för tillståndsövergångar kan i ett elsystem generellt antas vara deterministiska, där exempelvis ett steg från en lindningsomkopplare alltid kommer att leda till en viss påverkan på spänningsnivåerna i systemet (givet den dåvarande effektnivån).

Återigen ligger osäkerheter i stället i systemets tillstånd, där exempelvis mätfel bland mätningar eller saknade värden kan bidra till att noggrannheterna för metoderna försämras. Återigen är val av diskonteringsfaktor i hög grad beroende på det definierade styrmålet, men det är rimligt att anta att de flesta kostnader för diverse handlingar som aktiveras är oberoende av *när* de aktiveras. Således bör dessa generellt inte diskonteras.

6.4.2 Typ av DRL-algoritm och datagenerering

Komplexiteten på styrsystemet kommer att avgöra vilken typ av DRL-algoritm som kommer att kunna användas. Används endast diskreta handlingar som exempelvis styrning av lindningsomkopplare eller in/urkoppling av shuntkompensatorer så kan ”off-policy”-metoder som exempelvis deep Q-learning användas. Önskas kontinuerliga handlingar att användas med sådana metoder så är det även möjligt att diskretisera kontinuerliga handlingsutrymmet. Dock innebär detta en viss förlust av den flexibilitet i styrningen som ett kontinuerligt handlingsutrymme medför.

Den utvalda DRL-algoritmen bör tränas först på simulerade system för att en acceptabel nivå på styrsystemet kan uppnås innan den potentiellt appliceras i det verkliga systemet. Statisk modellering är fullt tillräckligt för simuleringen av spänningsregleringsproblemet och kraftsimuleringsprogram som kan hantera lastflödesberäkningar är således tillräckliga. Återigen bör det noteras att det inte är säkert att det krävs djupinlärning/DRL för att dessa algoritmer ska bli effektiva då antalet tillstånd som finns definierade i MDP:n med stor sannolikhet är betydligt färre än de som krävs då dynamiska förhållanden på en stamnätsnivå analyseras.

6.4.3 Översikt kring tidigare studier

I följande del ges en översikt kring tidigare studier där förstärkningsinlärning har använts vid styrning av spänningsreglering. Återigen är huvudsyftet med översikten att främst exemplifiera vilka algoritmer och begräsningar som tidigare studier har använt sig av och är därmed inte menad att fullständigt redogöra för den litteratur som finns tillgänglig inom området. I [26] användes klassisk ”Q-learning” för att styra reaktiv effekt i ett elsystem. Handlingarna inkluderade styrning av lindningsomkopplare, shuntkompensatorer, samt spänning och aktiv energi vid systembussar där generatorer av diverse slag var inkopplade. Belöningssystemen baserades endast på hur väl styrningen uppnådde den önskade spänningsprofilen och kostnaden för handlingarna var ej inkluderade i belöningsfunktionen. I [27] presenteras en algoritm baserad på ”Q-learning” och som bygger på ett så kallat ”multi-agent”-ramverk. Metoden bygger på att flera decentraliserade RL-agenter används och tränas simultant vilket gör att behovet av att kommunicera mätvärden och utföra beräkningar kan distribueras på flera olika lokala noder i nätet. Handlingarna som styrdes inkluderade generators busspänningar, shuntkompensatorer, samt steg för lindningsomkopplare på transformatorer. Återigen så baserades belöningssystemet endast på hur väl styrningen uppnådde den önskade spänningsprofilen och kostnader för handlingarna var ej inkluderade.

Att använda sig av DRL är inte enda alternativet för att hantera högdimensionella och kontinuerliga tillståndsrymder. I [28] användes en algoritm benämnd "least square policy iteration" för att styra transformatorers lindningsomkopplare, där en linjär funktionsapproximering användes för att gå från ett tillstånd (bestående av spänningsmagnituder och aktuella steg för lindningsomkopplare) till ett estimat för handlings-värdefunktionen. DRL har även använts för att utföra effektiv styrning av omriktare för att hantera spänningsregleringsproblem. I [29] användes en algoritm kallad "deep deterministic policy gradient" för att förbättra ett systems spänningsprofil och för att minimera mängden av produktion som behöver styras ned för att spänningsgränser ej ska överskridas.

6.5 UTMANINGAR OCH PRAKTISKA ASPEKTER

I följande del diskuteras generella utmaningar och praktiska aspekter då DRL används inom styrning inom distributionsnät. Utmaningarna är gemensamma för de båda genomförda fallstudierna.

1) *Tolkningsbarhet av styrpolicyer:* Neurala nätverk är mycket effektiva på att kunna approximera icke-linjära samband mellan ingångsvärden och utgångsvärden. Denna förmåga utnyttjas inom DRL för att kunna skapa optimala styrpolicyer då sambandet mellan tillstånd (ingångsvärden) och handlingar (utgångsvärden) ofta är icke-linjärt. Dock medför användningen av neurala nätverk att tolkningsbarheten av de styrpolicyer som de neurala nätverken approximerar är svår att utvärdera. Det är exempelvis mycket svårt att specifikt kunna tolka vilka kombinationer av tillstånd som resulterar i vilka typer av handlingar. Andra typer av funktionsapproximatorer (baserade på exempelvis linjära funktioner) är enklare att utvärdera och tolka, men ger generellt en sämre och mindre effektiv styrpolicy. Det finns således en avvägning mellan prestandan av en styrpolicy och dess tolkningsbarhet.

2) *Säkerhet och robusthet:* För att kunna använda DRL för styrning av elnät krävs det att den utvecklade styrpolicyn är robust och kan hantera exempelvis mätfel, oförutsedda driftpunkter, eller förändringar i systemets topologi. Hur påverkan av mätfel eller oförutsedda driftpunkter har utvärderats och exemplifierats i resultaten för de utförda studierna i kapitel 4 och kapitel 5. Det finns flera möjligheter att förbättra och försäkra en DRL-algoritms robusthet, genom exempelvis att använda verifieringsmetoder för det neurala nätverkets beteende för olika driftpunkter [30], eller genom att använda metoder som kan ta hänsyn till de fysiska lagar som beskriver elnätets dynamik [31].

3) *Dataeffektivitet:* För att en effektiv styrpolicy ska kunna utvecklas krävs generellt att en relativt stor mängd träningsdata används för att träna algoritmerna. Detta kan vara problematiskt om tillgängliga mätdata är begränsade alternativt om det är tidskrävande/problematiskt att simulera nya träningsdata. Off-policy algoritmer är generellt mer dataeffektiva då de kan återanvända tidigare insamlade data och åter träna algoritmen på dessa. Generellt bör dock DRL-algoritmer utvecklas för styrområden där

det finns tillgängligt en större mängd historiska data och där man effektivt kan modellera systemet dynamik genom simuleringssystem.

4) *Simuleringssystems och modellers noggrannhet:* I de fall då simuleringssystem eller modeller används för att skapa träningsdata till DRL-algoritmerna är det viktigt att deras noggrannhet kan verifieras. Påverkan av exempelvis felaktiga estimeringar av ledningars parametervärden, systemets topologi, modellers dynamik, kommer att leda till att de simulerade data som genereras ej kommer effektivt avspegla det faktiska systemet. Att kunna verifiera simuleringssystem och modellers noggrannhet är således av hög vikt.

5) *Validering och verifiering av styrpolicyer:* För att kunna försäkra sig att utvecklade styrpolicyer är effektiva krävs det även att de kan valideras i det faktiska systemet. För vissa styrproblem är detta trivialt och en suboptimal styrpolicy som testas i det riktiga systemet kommer i värsta fall att innebära en viss högre driftkostnad under en begränsad tid. Ett typiskt exempel här kan vara styrning av ett hushålls laster för att minimera elkostnaden; i värsta fall innebär en icke-optimal styrning att endast en aningen högre elkostnad blir resultatet. Istället är det betydligt svårare att verifiera styrpolicyer som är anpassade för akuta och sällsynta styrproblem. Ett exempel på ett sådant styrproblem är det som utvärderades i kapitel 5 där ett systems spänningsstabilitet skulle stabiliseras med hjälp av styrning av lastefterfrågeflexibilitet och energilagringssystem. Då spänningsinstabilitet inträffar extremt sällan och dessutom inte går att manuellt testa i ett system (på grund av för höga kostnader och risker för systemet) så kommer styrpolicyn inte att kunna valideras. I dessa fall måste i stället simuleringsmodellernas noggrannhet utvärderas i så hög grad som möjligt för att verifiera att de effektivt simulerar det faktiska systemets dynamik.

7 Sammanfattning

Den pågående omställningen till ett elsystem med en större andel förnyelsebar elproduktion och nya typer av laster kommer att innebära stora utmaningar för elnätsoperatörer. Att effektivt kunna styra undan säkerhets- och stabilitetsproblem till en låg systemkostnad kan både bidra till ett mer effektivt utnyttjande av det befintliga elsystemet och underlätta omställningen. I denna rapport har två styrsystem baserade på DRL utvecklats. DRL bygger på en kombination mellan djupinlärning och förstärkningsinlärning och används för att skapa optimala styrpolicyer för realtidsstyrning av komplexa system. Två olika styrproblem har utvärderats i rapporten: 1) förebyggande styrning och 2) akut styrning.

Metoden för förebyggande styrning är anpassad till att kunna vidta styråtgärder i preventivt syfte för att kunna säkerställa en säker och kontinuerlig drift av elnätet även om en större systemstörning skulle inträffa. Den utvecklade metoden testades på en rad olika drift- och störningsscenarior. Metoden visade god prestanda på de utvecklade testseten och den utvecklade DRL-agenten lyckades styra systemets handlingar så att systemets säkerhet återställdes i ett enda tidssteg och till en låg systemkostnad. Metoden utvärderades även på scenarier då mätfel introducerades i tillståndssignalen vilket resulterade i att metodens prestanda försämrades, vilket exemplifierar behovet av att inkludera mätfel även i den träningsdata som använts för att träna DRL-agenten.

Metoden för akut styrning är anpassad till att kunna vidta styråtgärder i akut syfte för att kunna styra systemet tillbaka till ett stabilt driftläge om en större systemstörning redan har inträffat. Styrbara komponenter inkluderade styrning av framtida tänkta stödtjänster från distributionsnät där exempelvis lastefterfrågeflexibilitet och styrning av energilagringssystem utnyttjades. Metoden visade god prestanda på alla de utvecklade testseten, vilket även inkluderade drift- och störningsscenarier som DRL-agenten ej hade tränats på. Därutöver så jämfördes även DRL-styrningen mot en mer konventionell styrning baserad på regelbaserad lastfrånkoppling där DRL-styrningen visade sig resultera i såväl en snabbare som mer effektiv styrning och dessutom krävde att en mindre mängd last kopplades bort från systemet.

Slutligen har även möjligheterna att använda DRL inom styrningsapplikationer för distributionsnät utvärderats. Två olika fallstudier har utvärderats i större detalj; dels styrning av resurser inom efterfrågeflexibilitet, dels styrning anpassad för spänningsreglering. Därutöver har olika praktiska aspekter som krav på tillgänglig mätinfrastruktur och olika utmaningar med att använda DRL inom styrning på distributionsnätsnivå utvärderats. DRL har möjlighet att förbättra nuvarande styrsystem, men många utmaningar kvarstår. Dels krävs stora mängder träningsdata för att uppnå en effektiv styrpolicy, dels så finns det stora utmaningar med att verifiera robustheten och stabiliteten för de utvecklade styrsystemen.

8 Referenser

[1] I. Goodfellow, Y. Bengio, och A. Courville, *Deep Learning*. MIT Press, 2016.

[2] R. S. Sutton och A. G. Barto, ”Reinforcement Learning: An Introduction (second edition)”. The MIT Press, Cambridge, Massachusetts London, England, 2018.

[3] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1:a uppl. Wiley, 1994. doi: 10.1002/9780470316887.

[4] C. J. Watkins och P. Dayan, ”Q-learning”, *Mach. Learn.*, vol. 8, nr 3–4, s. 279–292, 1992.

[5] G. A. Rummery och M. Niranjan, *On-line Q-learning using connectionist systems*, vol. 37. Citeseer, 1994.

[6] R. J. Williams, ”Simple statistical gradient-following algorithms for connectionist reinforcement learning”, *Mach. Learn.*, vol. 8, nr 3–4, s. 229–256, 1992.

[7] S. M. Kakade, ”A natural policy gradient”, *Adv. Neural Inf. Process. Syst.*, vol. 14, 2002.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, och O. Klimov, ”Proximal policy optimization algorithms”, *ArXiv Prepr. ArXiv170706347*, 2017.

[9] T. V. Cutsem, C. Moisse, och R. Mailhot, ”Determination of secure operating limits with respect to voltage collapse”, *IEEE Trans. Power Syst.*, vol. 14, nr 1, s. 327–335, feb. 1999, doi: 10.1109/59.744551.

[10] T. V. Cutsem *m.fl.*, ”Test Systems for Voltage Stability Studies”, *IEEE Trans Power Syst*, vol. 35, nr 5, s. 4078–4087, 2020.

[11] *PSS®E 35.0.0 Program Application Guide: Volume I*. Schenectady, NY: Siemens Power Technologies International, 2019.

[12] M. Neunert *m.fl.*, ”Continuous-discrete reinforcement learning for hybrid control in robotics”, i *Conference on Robot Learning*, 2020, s. 735–751.

[13] H. Hagmar, R. Eriksson, och L. Tuan, ”Fast Dynamic Voltage Security Margin Estimation: Concept and Development”, *IET Smart Grid*, vol. 3, apr. 2020, doi: 10.1049/iet-stg.2019.0278.

[14] ”Tensorflow Keras Optimizers: RMSprop”. https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/RMSprop (åtkomstdatum 23 augusti 2021).

[15] F. Rahimi och A. Ipakchi, ”Demand Response as a Market Resource Under the Smart Grid Paradigm”, *IEEE Trans. Smart Grid*, vol. 1, nr 1, s. 82–88, juni 2010, doi: 10.1109/TSG.2010.2045906.

[16] D. P. Kingma och J. Ba, ”Adam: A Method for Stochastic Optimization”, *ArXiv E-Prints*, s. arXiv:1412.6980, dec. 2014.

[17] L. Veman Tell, L. Jaakonantti, E. Grahn, T. Ny, och R. Husblad, ”Funktionskrav på elmätare”, Energimarknadsinspektionen, Eskilstuna, Ei R2017:08, 2017. [Online]. Tillgänglig vid: www.ei.se

[18] SFS 1999:716, *Förordning (1999:716) om mätning, beräkning och rapportering av överförd el*. Stockholm: Infrastrukturdepartementet.

[19] ”Funktionskrav elmätare”. Energimarknadsinspektionen. [Online]. Tillgänglig vid: https://www.ei.se/bransch/funktionskrav-elmatare

[20] H. Hagmar och A. Lindskog, ”Bedömning av nätstatus baserad på datanalys och avancerade algoritmer”, Energiforsk, Stockholm, 2017.

[21] D. Zagerholm, S. Ackeby, och C. Wiig, ”Digitalisering för efterfrågeflexibilitet”, Energiforsk, Stockholm, 2021.

[22] H. Wang och B. Zhang, ”Energy Storage Arbitrage in Real-Time Markets via Reinforcement Learning”, i *2018 IEEE Power & Energy Society General Meeting (PESGM)*, Portland, OR, aug. 2018, s. 1–5. doi: 10.1109/PESGM.2018.8586321.

[23] J. Cao, D. Harrold, Z. Fan, T. Morstyn, D. Healey, och K. Li, ”Deep Reinforcement Learning-Based Energy Storage Arbitrage With Accurate Lithium-Ion Battery Degradation Model”, *IEEE Trans. Smart Grid*, vol. 11, nr 5, s. 4513–4521, sep. 2020, doi: 10.1109/TSG.2020.2986333.

[24] B. Wang, Y. Li, W. Ming, och S. Wang, ”Deep Reinforcement Learning Method for Demand Response Management of Interruptible Load”, *IEEE Trans. Smart Grid*, vol. 11, nr 4, s. 3146–3155, juli 2020, doi: 10.1109/TSG.2020.2967430.

[25] F. Ruelens, B. J. Claessens, S. Vandael, B. De Schutter, R. Babuska, och R. Belmans, ”Residential Demand Response of Thermostatically Controlled Loads Using Batch Reinforcement Learning”, *IEEE Trans. Smart Grid*, vol. 8, nr 5, s. 2149–2159, sep. 2017, doi: 10.1109/TSG.2016.2517211.

[26] J. G. Vlachogiannis och N. D. Hatziargyriou, ”Reinforcement Learning for Reactive Power Control”, *IEEE Trans. Power Syst.*, vol. 19, nr 3, s. 1317–1325, aug. 2004, doi: 10.1109/TPWRS.2004.831259.

[27] Y. Xu, W. Zhang, W. Liu, och F. Ferrese, ”Multiagent-Based Reinforcement Learning for Optimal Reactive Power Dispatch”, *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 42, nr 6, s. 1742–1751, nov. 2012, doi: 10.1109/TSMCC.2012.2218596.

[28] H. Xu, A. D. Dominguez-Garcia, och P. W. Sauer, ”Optimal Tap Setting of Voltage Regulation Transformers Using Batch Reinforcement Learning”, *IEEE Trans. Power Syst.*, vol. 35, nr 3, s. 1990–2001, maj 2020, doi: 10.1109/TPWRS.2019.2948132.

[29] C. Li, C. Jin, och R. Sharma, ”Coordination of PV Smart Inverters Using Deep Reinforcement Learning for Grid Voltage Regulation”, i *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, Boca Raton, FL, USA, dec. 2019, s. 1930–1937. doi: 10.1109/ICMLA.2019.00310.

[30] A. Venzke och S. Chatzivasileiadis, ”Verification of Neural Network Behaviour: Formal Guarantees for Power System Applications”, *IEEE Trans. Smart Grid*, vol. 12, nr 1, s. 383–397, 2021, doi: 10.1109/TSG.2020.3009401.

[31] G. S. Misyris, A. Venzke, och S. Chatzivasileiadis, ”Physics-Informed Neural Networks for Power Systems”, i *2020 IEEE Power & Energy Society General Meeting (PESGM)*, Montreal, QC, Canada, aug. 2020, s. 1–5. doi: 10.1109/PESGM41954.2020.9282004.

MASKININLÄRNINGSBASERAD REAL-TIDSSTYRNING AV FÖRNYELSEBARA OCH SÄKRA ELKRAFTSYSTEM

I det här projektet har forskarna studerat och tagit fram två nya styrmetoder baserade på djup förstärkningsinlärning anpassade för styrning av elsystem. Djup förstärkningsinlärning bygger på en kombination mellan djupinlärning och förstärkningsinlärning och används för att kunna ta fram avancerade och optimala styrpolicyer för realtidsstyrning av komplexa system.

En av de utvecklade metoderna är anpassade för att i realtid kunna välja olika handlingar för att säkerställa att ett elsystem kan bibehålla en säker och kontinuerlig drift även om en systemstörning skulle inträffa. Den andra styrmetoden är anpassad för att snabbt och effektivt stabilisera systemet om en större, eller flera simultana, systemstörningar skulle inträffa. Metoderna möjliggör för systemoperatörer att mer effektivt kunna styra undan säkerhets- och stabilitetsproblem till en låg systemkostnad, vilket kan leda både till ett mer effektivt utnyttjande av det befintliga elsystemet och underlätta omställningen till ett förnyelsebart energisystem.

Därutöver har projektet undersökt möjligheterna för styrmetoder baserade på förstärkningsinlärning att även användas på distributionsnätsnivå. Två olika fallstudier har analyserats där 1) möjligheterna att reglera spänning genom reaktiva effektflöden och 2) möjligheterna att styra flexibla resurser som laster och elproduktion har undersökts.

Ett nytt steg i energiforskningen

Energiforsk är en forsknings- och kunskapsorganisation som samlar stora delar av svensk forskning och utveckling om energi. Målet är att öka effektivitet och nyttiggörande av resultat inför framtida utmaningar inom energiområdet. Vi verkar inom ett antal forskningsområden, och tar fram kunskap om resurseffektiv energi i ett helhetsperspektiv – från källan, via omvandling och överföring till användning av energin. www.energiforsk.se

Energiforsk