Universität Stuttgart
Institut für Energiewirtschaft und Rationelle Energieanwendung
IER

Studienarbeit

Entwicklung eines adaptiven autoregressiven Modells zur Kurzfristlastprognose im liberalisierten Energiemarkt

Bastian Buchholz

Band 376

Entwicklung eines adaptiven autoregressiven Modells zur Kurzfristlastprognose im liberalisierten Energiemarkt

Studienarbeit im Hauptfach Energiesysteme

angefertigt von
cand. mach. Bastian Buchholz
Herweghstr. 12
70197 Stuttgart
Matr. Nr.: 1849655

Betreuer:	Prof. Dr.-Ing. A. Voß, Dipl.-Ing. D. J. Swider, IER

Studienrichtung:	Maschinenwesen
1. Hauptfach:	Mechanische Verfahrenstechnik
2. Hauptfach:	Energiesysteme

Beginn der Arbeit:	22. Juli 2002
Ende der Arbeit:	20. November 2002

Institut für Energiewirtschaft und Rationelle Energieanwendung, Stuttgart
Prof. Dr.-Ing. A. Voß
Abteilung Energieanwendung und Energiemanagement (EAM)
Dr. oec. C. Weber

Kurzfassung

Für Unternehmen in der Energiewirtschaft ist es von großer Bedeutung, die elektrische Lastnachfrage seitens der Verbraucher, vorhersagen zu können. Auf diese Weise können zum einen Stromproduzenten den Betrieb ihrer Kraftwerke, und zum Anderen die durch die Liberalisierung des Energiemarktes neu entstandenen Stromhändler ihre Strategie beim An- und Verkauf von elektrischer Energie besser planen.

Ziel dieser Arbeit ist es, auf Basis eines autoregressiven Ansatzes (AR-Ansatz), verschiedene Modelle zu erstellen, welche eine Kurzfrist-Prognose, das heisst für den Zeitraum der nächsten 24 Stunden, der elektrischen Lastnachfrage ermöglichen. Der reine AR-Ansatz wird hierfür erweitert. Zuerst durch ein adaptives Prognosevorgehen, welcher es dem Modell ermöglicht sich an veränderliche Bedingungen anzupassen, wie sie zum Beispiel durch die Fluktuationen im Kundenstamm ,verursacht durch die Liberalisierung des Energiemarktes entstehen. Anschließend durch einen X-Anteil (ARX-Ansatz) welcher den Einfluss der Temperatur berücksichtigt und Konstanten zur Abbildung spezifischer elektrischer Tageslastverläufe enthält. Die Verwirklichung eines Moving-Average-Anteils (ARMAX-Ansatz) scheitert daran, dass das sich ergebende, nicht-lineare Gleichungssystem mit dem hier verwendeten Lösungsalgorithmen nicht zufriedenstellend gelöst werden kann.

Die Ergebnisse zeigen, dass der ARX-Ansatz, kombiniert mit einem adaptiven Prognosevorgehen, die besten Ergebnisse liefert. Werktage, speziell in der warmen Jahreszeit, lassen sich mit diesem Modell gut abbilden. Die Prognose von Wochenend- und Feiertagen, sowie Tagen in der kalten Jahreszeit muss noch verbessert werden. Dabei ist bei der Prognose von Tagen der kalten Jahreszeit eine Verfeinerung des Verfahrens zur Berücksichtigung der Temperatur vorzunehmen und eine Einbeziehung weiterer Einflussgrößen wie zum Beispiel der Helligkeit vorzusehen.

Inhaltsverzeichnis

1. Einleitung **1**

2. Mathematische Grundlagen **3**
2.1. Multiple lineare Regression 3
2.2. ARIMAX Ansatz 4
2.3. Statistische Fehleranalyse 7

3. Analyse der Datengrundlage **9**
3.1. Verläufe von elektrischer Lastnachfrage 9
3.2. Analyse der Daten mit ACF und PACF 16

4. Methodisches Vorgehen **23**
4.1. AR(I)X-Ansatz 23
4.2. AR(I)MAX-Ansatz 30
4.3. Erklärungen zum Programm prognoser 31

5. Ergebnisse und Diskussion **35**

6. Zusammenfassung und Ausblick **41**

A. Ergebnisse **43**

B. Source Codes **47**
B.1. checkdata 48
B.2. prognoser_pure01 50
B.3. prognoser_pure02 52
B.4. prognoser 54

Literaturverzeichnis **77**

Abbildungsverzeichnis

3.1. Elektrischer Lastnachfrageverlauf für Werktage zu verschiedenen Jahreszeiten. 11
3.2. Elektrischer Lastnachfrageverlauf für zwei Wochen im Sommer und zugehöriger Temperaturverlauf. 12
3.3. Elektrischer Lastnachfrageverlauf für zwei Wochen im Winter und zugehöriger Temperaturverlauf. 13
3.4. Elektrischer Lastnachfrageverlauf mit Feiertag und Brückentag, und zugehöriger Temperaturverlauf. 14
3.5. Elektrische Lastnachfrage verschiedener Werktage. 15
3.6. Elektrischer Lastnachfrageverlauf für den Zeitraum um Weihnachten herum und zugehöriger Temperaturverlauf. 16
3.7. *ACF*, berechnet für alle Tage der ersten der beiden Wochen aus Abbildung 3.2 17
3.8. *ACF*, berechnet für alle Tage der zweiten der beiden Wochen aus Abbildung 3.3 18
3.9. *ACF*, berechnet für alle Tage der zweiten der beiden Wochen aus Abbildung 3.4 19
3.10. *ACF*, berechnet für alle Tage der zweiten der beiden Wochen aus Abbildung 3.6 20
3.11. *PACF*, berechnet für den Mittwoch der ersten der beiden Wochen aus Abbildung 3.2 . , und *PACF*, berechnet für den Mittwoch der zweiten der beiden Wochen aus Abbildung 3.3 21

4.1. Prinzip der Abbildungskonstanten. 27
4.2. Trainings- und Prognoseprinzip. 31
4.3. Struktur des Skriptes *prognoser*. 32
4.4. Grafische Ausgabe von *prognoser_plot*. 33

5.1. Vergleich der Prognoseergebnisse der grundlegenden Modellansätze. 36
5.2. Vergleich der Prognoseergebnisse einmal ohne und einmal mit Berücksichtigung der Temperatur. 38
5.3. Vergleich der Prognoseergebnisse für verschiedene Tagtypen, berechnet mit dem *ARX*-Ansatz. 39

5.4. Prognose der Beispielwochen aus Abbildung 3.2, berechnet mit dem ARX-Ansatz. 39
5.5. Prognose der Beispielwochen aus Abbildung 3.6, berechnet mit dem $ARXTe$-Ansatz. 40

Tabellenverzeichnis

4.1. Struktur einer Lastdatenmatrix. 25

5.1. Modellübersicht. 35

Abkürzungsverzeichnis

Formelzeichen

Symbol	Erläuterung	Dimension
$\underline{\mathbf{y}}$	Vektor der Zielgröße	
$\bar{y}$	arithmetischer Mittelwert der Zielgröße	
$\underline{\mathbf{x}}_i$	Vektor der erklärenden Eingangsgröße	
$\underline{\hat{\mathbf{y}}}$	Vektor der geschätzten Zielgröße	
$\underline{\hat{\beta}}$	Vektor der geschätzten Prognosekoeffizienten	
$\underline{\mathbf{u}}$	Vektor der Prognosefehler	
$\underline{\mathbf{e}}$	Einheitsvektor	
$\mathbf{X}$	Matrix der Eingangsgrößen	
$\mathbf{Y}$	Matrix der autoregressiven Größen	
$\mathbf{U}$	Matrix der Prognosefehler	
n	Zeitindex	
i	Index der Eingangsgrößen	
p	maximaler zeitlicher Lag des AR-Anteils	
q	maximaler zeitlicher Lag des MA-Anteils	
m	Maximalwert aus p und q	
d	Grad der Differenzierung bei I-Anteil	
Δ	Differenzenoperator	
$\hat{\gamma}$	Schätzwert für Autokovarianz	
$\hat{\varrho}$	Schätzwert für Autokorrelation	
$\hat{\mathbf{R}}$	Schätzwert für Autokorrelationsmatrix	
$\hat{\phi}$	Yule-Walker Schätzer	
h	maximaler zeitlicher Lag für Autokovarianz, Autokorrelation und partielle Autokorrelation	
l	negativer Log-Likelihood	
$\underline{\beta}^*$	Vektor mit Startwerten für $\underline{\beta}$	
u_n^*	Prognosefehler berechnet mit $\underline{\beta}^*$	
σ_u^*	Standardabweichung der Prognosefehler u_n^*	

Symbol	Erläuterung	Dimension
S_*	Quadratsumme der Prognosefehler u_n^*	
r^2	Bestimmtheitsmaß	
d_{MoDi}	Dummy für die Abbildung Montag auf Dienstag	
d_{DoFr}	Dummy für die Abbildung Donnerstag auf Freitag	
d_{FrMo_01}	erster Dummy für die Abbildung Freitag auf Montag	
d_{FrMo_02}	zweiter Dummy für die Abbildung Freitag auf Montag	
x_{meanSo}	Mittelwert der Lastnachfrage eines Sonntags	MW
ϑ_n	viertelstündlicher Mittelwert der Temperatur	$°C$
$\bar{\vartheta}$	Tagesmittelwert der Temperatur	$°C$
x_ϑ	Eingangsgröße für die Temperaturberücksichtigung	$°C$
d_ϑ	Abbildungsdummy für die Temperaturberücksichtigung	

Abkürzungen

Symbol	Erläuterung	Dimension
AR	autoregressiver Modellanteil	
MA	Moving-Average-Modellanteil	
I	integrativer Modellanteil	
X	Modellanteil mit äußeren Einflussgrößen	
ACF	Autokorrelationsfunktion	
$PACF$	Partielle Autokorrelationsfunktion	
$rmse$	Root Mean Square Error	MW
mbe	Mean Baised Error	
$mape$	Mean Absolute Percentage Error	
$maxae$	Maximum Absolute Error	MW
$maxape$	Maximum Absolute Percentage Error	

1. Einleitung

Die Liberalisierung des Strommarktes hat für Anbieter und Verbraucher elektrischer Energie weitreichende Folgen. Die Anbieter, jetzt durch ein Unbundeling, also einer Entflechtung der Geschäftsbereiche, unterteilt in Hersteller und Vertreiber elektrischer Energie, sind unter anderem durch die Möglichkeit der Verbraucher, ihren Stromanbieter frei zu wählen, vor neue Herausforderungen gestellt. Eine der neuen Herausforderungen ist die Vorhersage der Nachfrage nach elektrischer Last seitens der Verbraucher. Die Fluktuation durch das Weggehen und Hinzukommen neuer Kunden macht eine präzise Prognose schwierig. Vor allem die Vertreiber haben ein Interesse daran, mittels kurzfristiger Prognosen, also innerhalb der nächsten 24 Stunden, eine Aussage treffen zu können, wieviel sie an elektrischer Last bei den Herstellern, also den Kraftwerksbetreibern, nachfragen müssen. Dabei ist aktuell wie auch in der Vergangenheit die Jahre lange Erfahrung von Mitarbeitern auf diesem Gebiet gefragt. Diese verwenden bei ihren Prognosen sowohl die Methode des historischen Vergleiches, also die Lastverläufe von vergangenen Tagen, an denen ähnliche Verhältnisse geherrscht haben wie an dem zu prognostizierenden Tag, als auch Software Prognose Programme. Nur mit Hilfe solcher Programme, welche sich Methoden der mathematischen Statistik wie der der Neuronalen Netze (ANN: Artificial Neuronal Networks) oder der Regression bedienen, können Effekte wie die oben genannte Kundenfluktuation berücksichtigt werden. Eine Möglichkeit dieses zu tun besteht darin, adaptive Modellansätze zur Lastprognose zu verwenden. Diese Modelle bestimmen die notwendigen Parameter bei jeder Prognose eines Tages-Lastverlaufes aufs Neue. Somit können sie sich an Veränderungen in den Bedingungen anpassen.

Ziel dieser Arbeit ist es, einen adaptiven-autoregressiven Modellansatz zur kurzfristigen Lastprognose zu entwickeln. Die Autoregression ist eine spezielle Form der Regression. Sie beruht darauf, den zukünftigen Verlauf einer Zeitreihe aus vergangenen Werten dieser Zeitreihe zu bestimmen. Die mathematischen Grundlagen zur Regression, zur Autoregression und deren Erweiterungen (ARMAX [1]-Ansätze) sowie zu Lösungsmethoden und zu Methoden der Zeitreihenanalyse werden in Kapitel 2 geliefert. Letztere sind notwendig, um die vorhandene Datengrundlage zu analysieren und daraus dann Strukturen der Modellansätze bestimmen zu können. Diese Datenanalyse und eine grundlegende Diskussion von

[1] Die Erläuterung der hier verwendeten Abkürzungen erfolgt in Kapitel 2.

Lastverläufen erfolgen in Kapitel 3. Das methodische Vorgehen bei der Ableitung der Modellansätze ist in Kapitel 4 beschrieben. Hier ist ebenfalls die Integration der Modellansätze in Prognoseskripte dargelegt. Als Programmiersprache dient MATLAB^{TM}. In Kapitel 5 werden die Prognoseergebnisse von AR- und ARX-Ansätzen miteinander verglichen und diskutiert. Kapitel 6 gibt abschließend eine Zusammenfassung und einen Ausblick darauf, welche Aspekte noch zu behandeln sind. Anhang A enthält eine vollständige Auflistung aller mit den Prognoseskripten ermittelten Ergebnisse, während sich in Anhang B die MATLAB^{TM} Sourcecodes dieser Skripte finden.

2. Mathematische Grundlagen

Die in dieser Arbeit diskutierten Prognosemodelle beruhen auf der multiplen linearen Regression und ihren Erweiterungen, zusammengefasst unter dem Begriff der *ARIMAX*-Ansätze. In diesem Kapitel sind kurz die Grundlagen zur Methodik der Regression und der weiterführenden Modelle sowie der benötigten Verfahren der Zeitreihenanalyse beschrieben. Darüber hinaus werden alle in dieser Arbeit verwendeten statistischen Maßzahlen erläutert.

2.1. Multiple lineare Regression

Wird angenommen, dass zwischen einer abhängigen Zielgröße y (zu erklärende Größe) und mehreren unabhängigen Größen $x_1, x_2, \ldots, x_i$ (erklärende Größen) ein linearer Zusammenhang besteht, so lässt sich dieser Zusammenhang in Form einer linearen Regressionsgleichung schreiben,

$$y = \hat{\beta}_0 + \hat{\beta}_1 \cdot x_1 + \hat{\beta}_2 \cdot x_2 + \cdots + \hat{\beta}_i \cdot x_i + u = \underline{\mathbf{x}} \cdot \underline{\hat{\beta}} + u \tag{2.1}$$

mit $\underline{\hat{\beta}} = (\hat{\beta}_0, \ldots, \hat{\beta}_i)^T$ als Koeffizienten der erklärenden Größen $\underline{\mathbf{x}} = (1, x_1, \ldots, x_i)$ und u als Fehler (Residuum). Als erklärende Größen kommen sowohl Variablen als auch Konstanten in Frage. In dieser Arbeit ist die Temperatur ein Beispiel einer variablen Größe, während Werte, welche den Einfluss unterschiedlicher Tagtypen beschreiben, als Konstanten eingehen. Zur Bestimmung der unbekannten Koeffizienten $\underline{\hat{\beta}}$ eignet sich vor allem das Verfahren zur Minimierung der Quadrate der Fehler u^2 (kleinste Fehlerquadrate/Least Squares). Hierbei handelt es sich um ein statistisches Schätzverfahren (im Folgenden werden alle geschätzten Größen mit einem Dachsymbol $\hat{\square}$ gekennzeichnet). Dieses wird umso robuster je mehr Beobachtungen von $\underline{\mathbf{y}} = (y_1, \ldots, y_n)^T$ und den jeweiligen Größen $\underline{\mathbf{x_i}} = (x_{1,i}, \ldots, x_{n,i})^T$ vorliegen. Gleichung 2.1 muss also in Vektorform umgeschrieben werden

$$\underline{\mathbf{y}} = \mathbf{X} \cdot \underline{\hat{\beta}} + \underline{\mathbf{u}} \tag{2.2}$$

mit

$$\mathbf{X} = \begin{bmatrix} \underline{\mathbf{e}} & \underline{\mathbf{x}}_1 & \underline{\mathbf{x}}_2 & \ldots & \underline{\mathbf{x}}_i \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \ldots & x_{1,i} \\ 1 & x_{2,1} & x_{2,2} & \ldots & x_{2,i} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_{n,1} & x_{n,2} & \ldots & x_{n,i} \end{bmatrix} \tag{2.3}$$

und

$$\underline{\mathbf{u}} = (u_1, u_2, \ldots, u_n) \ . \tag{2.4}$$

Das Verfahren zur Minimierung der Fehlerquadrate, angewendet in Gleichung 2.1, ergibt als eindeutige Lösung für $\underline{\hat{\beta}}$ (siehe auch [2])

$$\underline{\hat{\beta}} = (\mathbf{X}^T \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^T \cdot \underline{\mathbf{y}} \ . \tag{2.5}$$

2.2. ARIMAX Ansatz

Im Begriff *ARIMAX* sind verschiedene Abkürzungen von Modellansätzen enthalten, aus denen sich das gesamte Modell zusammensetzt. *AR* steht für *A*uto *R*egressive, *I* für *I*ntegrative, *MA* für *M*oving *A*verage (gleitender Durchschnitt) und *X* bezeichnet weitere Eingangsgrößen neben den *AR*- und *MA*- Anteilen.

Eine Autoregression liegt vor, wenn für die Regression einer Zeitreihe $\underline{\mathbf{y}}$ als erklärende Größen Werte von $\underline{\mathbf{y}}$ zu beliebigen früheren Zeitpunkten berücksichtigt werden, die abhängige Größe $\underline{\mathbf{y}}$ wird also auf sich selbst bezogen. Der reine *AR*-Ansatz lässt sich wie folgt schreiben

$$y_n = \hat{\beta}_0 + \hat{\beta}_1 \cdot y_{n-1} + \hat{\beta}_2 \cdot y_{n-2} + \cdots + \hat{\beta}_p \cdot y_{n-p} + u \ . \tag{2.6}$$

Zur Berechnung der Koeffizienten kann wieder das Least Squares Verfahren aus Gleichung 2.5 dienen. Die Bestimmung des maximalen zeitlichen Lags p erfolgt über Verfahren der Zeitreihenanalyse. Ziel ist ein Maß für die Autokorrelation zwischen verschiedenen Werten der Zeitreihe $\underline{\mathbf{y}}$. Die Autokorrelationsfunktion (*A*uto*C*orrelation *F*unction/*ACF*) und die partielle Autokorrelationsfunktion (*P*artitial *A*uto*C*orrelation *F*unction/ *PACF*) beschreiben einen solchen Zusammenhang. Für eine Zeitreihe $\underline{\mathbf{y}}$ mit n Beobachtungen lässt sich nach [2] als Ausdruck für die Schätzung der Autokovarianz als Funktion des zeitlichen Lags h angeben:

$$\hat{\gamma}(h) = n^{-1} \sum_{k=1}^{n-h} (y_k - \bar{y})(y_{k+h} - \bar{y}) \qquad h = 0, 1, 2, \ldots, (n-1) \ . \tag{2.7}$$

Hier ist

$$\bar{y} = \frac{1}{n} \cdot \underline{\mathbf{y}} \tag{2.8}$$

der arithmetische Mittelwert der Zeitreihe $\underline{\mathbf{y}}$. Aus Gleichung 2.7 erhält man nach Division durch die Autokovarianz des Lags Null $\hat{\gamma}(0)$ die Schätzung der Autokorrelationsfunktion

$$\hat{\varrho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)} \ . \tag{2.9}$$

Der Wert der *ACF* für einen zeitlichen Lag h entspricht dem Mittelwert der Korrelation aller Wertepaare die sich für diesen Lag in einer Zeitreihe ergeben.

Es ist ersichtlich, dass dieser Mittelwert mit Zunahme von h an Robustheit verliert, da die Anzahl der Werte zu dessen Ermittlung abnimmt. Dies muss bei der späteren Interpretation der Ergebnisse berücksichtigt werden. Oft wird als maximaler Lag, für den die Robustheit der Werte noch gegeben ist, $\frac{n}{2}$ verwendet. Die gleichen Aussagen gelten für die *PACF*, da diese aus der *ACF* berechnet wird. Ein mögliches Verfahren hierzu ist das Aufstellen der Yule-Walker-Gleichungen. Dafür ist die Definition der Autokorrelationsmatrix nach

$$\hat{\mathbf{R}} = \begin{bmatrix} \hat{\varrho}(0) & \hat{\varrho}(1) & \ldots\ldots\ldots & \hat{\varrho}(n-1) \\ \hat{\varrho}(1) & \hat{\varrho}(0) & \hat{\varrho}(1) \ \ldots & \hat{\varrho}(n-2) \\ \ldots\ldots & \ldots\ldots & \ldots\ldots & \ldots\ldots \\ \hat{\varrho}(n-1) & \ldots\ldots & \ldots\ldots & \hat{\varrho}(0) \end{bmatrix} \tag{2.10}$$

nötig. Die Yule-Walker-Gleichungen

$$\underline{\hat{\varrho}} = \hat{\mathbf{R}} \cdot \underline{\hat{\phi}} \tag{2.11}$$

mit einem modifizierten *ACF*-Vektor

$$\underline{\hat{\varrho}} = \begin{bmatrix} \hat{\varrho}(1) \\ \hat{\varrho}(2) \\ \ldots \\ \hat{\varrho}(h) \end{bmatrix} \tag{2.12}$$

und dem gesuchten *PACF*- Vektor

$$\underline{\hat{\phi}} = \begin{bmatrix} \hat{\phi}(1) \\ \hat{\phi}(2) \\ \ldots \\ \hat{\phi}(h) \end{bmatrix} \tag{2.13}$$

lassen sich nach eben diesem auflösen. Man erhält [1]

$$\underline{\hat{\phi}} = (\hat{\mathbf{R}}^T \cdot \hat{\mathbf{R}})^{-1} \cdot \hat{\mathbf{R}}^T \cdot \underline{\hat{\varrho}} \ . \tag{2.14}$$

Für den Wert $\hat{\phi}(0)$ ergibt sich nach Definition

$$\hat{\phi}(0) = 1 \ . \tag{2.15}$$

Im Gegensatz zur *ACF* beschreibt die *PACF* nur die Korrelation zwischen festen Werten, nämlich dem Anfangswert der Zeitreihe und dem sich für den jeweiligen Lag h ergebenden Wert. Dies erkärt auch die Definition aus Gleichung 2.15. Die Ergebnisse der Analyse mit *ACF* und *PACF* sind in Abschnitt 3.2 beschrieben.

[1] Die Gleichung 2.11 ist vollständig in Schätzwertform ohne Fehlerterm angegeben. Wird sie hingegen wie Gleichung 2.2 formuliert, so kann Gleichung 2.14 auch als Ergebnis eines Least-Squares Verfahren verstanden werden.

Weist ein Modell einen I-Anteil auf, so wurde die Zeitreihe $\underline{\mathbf{y}}$ vor der Regression differenziert. Auf diese Weise kann ein Trend, das heißt der Anstieg beziehungsweise Abfall des Mittelwertes von $\underline{\mathbf{y}}$, eliminiert werden. Bei unstetigen Zeitreihen wie sie hier im Fall der Lastprognose vorliegen, führt das Ganze zur Bildung von Differenzen:

$$\Delta\, y_n = y_{n+1} - y_n \ . \tag{2.16}$$

Zu beachten ist, dass sich die Anzahl der Elemente im Vektor $\Delta\ \underline{\mathbf{y}}$ bei jeder Differenzierung um den Wert 1 verringert. Der Grad der Differenzierung wird allgemein mit d bezeichnet, Gleichung 2.16 steht also für einen I-Anteil mit $d = 1$.

Für den MA-Anteil werden als Eingänge Zufallsgrößen verwendet, wie zum Beispiel der Prognosefehler

$$u_n = y_n - \hat{y}_n \tag{2.17}$$

zwischen Schätzwert $\hat{y}_n$ und Originalwert y_n zu einem beliebigen früheren Zeitpunkt. Die Zusammenfassung von AR- und MA-Anteil ergibt den $ARMA$-Ansatz:

$$y_n = \hat{\beta}_0 + \hat{\beta}_1 \cdot y_{n-1} + \cdots + \hat{\beta}_p \cdot y_{n-p} + \hat{\beta}_{p+1} \cdot u_{n-1} + \cdots + \hat{\beta}_{p+q} \cdot u_{n-q} + u_n \ . \tag{2.18}$$

Der maximale zeitliche Lag des MA-Anteiles wird mit q bezeichnet. Verfahren zur Bestimmung von q, wie sie zum Beispiel in [1] beschrieben sind, können im Rahmen dieser Arbeit nicht berücksichtigt werden. Vielmehr soll q identisch zu p gewählt werden.

Schließlich können noch weitere Eingangsgrößen x_i, wie in Abschnitt 2.1 beschrieben, in den Ansatz mit einbezogen werden. Dieser Anteil wird als X-Anteil bezeichnet. Als Gesamtgleichung für einen $ARMAX$-Ansatz lässt sich schreiben:

$$\begin{aligned} y_n = \hat{\beta}_0 + \hat{\beta}_1 \cdot y_{n-1} + \cdots + \hat{\beta}_p \cdot y_{n-p} + \hat{\beta}_{p+1} \cdot u_{n-1} + \cdots + \hat{\beta}_{p+q} \cdot u_{n-q} \\ + \hat{\beta}_{p+q+1} \cdot x_1 + \cdots + \hat{\beta}_{p+q+i} \cdot x_i + u_n \ . \end{aligned} \tag{2.19}$$

Oder wie in Gleichung 2.5 für eine Zeitreihe $\underline{\mathbf{y}}$ in Vektorform:

$$\underline{\mathbf{y}} = \underline{\mathbf{e}} \cdot \hat{\beta}_0 + \mathbf{Y} \cdot \underline{\hat{\beta}}_{AR} + \mathbf{U} \cdot \underline{\hat{\beta}}_{MA} + \mathbf{X} \cdot \underline{\hat{\beta}}_{X} + \underline{\mathbf{u}} \ . \tag{2.20}$$

Dabei gelten folgende Definitionen

$$\underline{\mathbf{y}} = \begin{bmatrix} y_m \\ y_{m+1} \\ \cdots \\ y_n \end{bmatrix} \qquad \mathbf{Y} = \begin{bmatrix} y_m & y_{m-1} & \cdots\cdots & y_1 \\ y_{m+1} & y_m & y_{m-1} & y_2 \\ \cdots & \cdots & \cdots & \cdots \\ y_{n-1} & y_{n-2} & \cdots\cdots & y_{n-m} \end{bmatrix} \tag{2.21}$$

sowie

$$\mathbf{U} = \begin{bmatrix} u_m & u_{m-1} & \cdots\cdots & u_1 \\ u_{m+1} & u_m & u_{m-1} & u_2 \\ \cdots & \cdots & \cdots & \cdots \\ u_{n-1} & u_{n-2} & \cdots\cdots & u_{n-m} \end{bmatrix} \qquad \underline{\mathbf{u}} = \begin{bmatrix} u_m \\ u_{m+1} \\ \cdots \\ u_n \end{bmatrix} \ . \tag{2.22}$$

Für den Index m gilt:

$$m = \max(p, q) \ . \tag{2.23}$$

Er berücksichtigt den durch den maximalen zeitlichen Lag verursachten Offsetbereich in $\underline{\mathbf{y}}$.

Durch das Einbeziehen des *MA*-Anteils in den Gleichungen 2.18 , 2.19 und 2.20 werden diese nichtlinear, da der Prognosefehler $\underline{\mathbf{u}}$ vor der Regression noch nicht bekannt ist. Zur Bestimmung der unbekannten Koeffizienten ist daher ein iteratives Vorgehen erforderlich. Ein gut geeignetes Verfahren hierfür ist die Minimierung des negativen Log-Likelihood. Nach [1] wird dieser definiert durch

$$l(\underline{\beta}^*, \sigma_u^*) = -n \ln(\sigma_u^*) - \frac{S_*(\underline{\beta}^*)}{2{\sigma_u^*}^2} \ . \tag{2.24}$$

Der Koeffizientenvektor $\underline{\beta}^*$ und die Standardabweichung des Prognosefehlers σ_u^* sind Startwerte für die Iteration. Der Ausdruck

$$S_*(\underline{\beta}^*) = \sum_{k=1}^{n} {u_n^*}^2 \tag{2.25}$$

steht für die Quadratsumme der Prognosefehler u_n^*, welche sich für $\underline{\beta}^*$ ergeben. Mit einem geeigneten Suchverfahren lassen sich nun Werte für $\underline{\hat{\beta}}$ so bestimmen, dass der Ausdruck in 2.24 minimal wird.

2.3. Statistische Fehleranalyse

Der bei einer Regression entstandene Fehler

$$\underline{\mathbf{u}} = \underline{\mathbf{y}} - \underline{\hat{\mathbf{y}}} \tag{2.26}$$

wird zur Berechnung statistischer Maßzahlen verwendet, welche die Güte eines Verfahrens erkennen lassen. Im Folgenden sind alle für diese Arbeit relevanten Maßzahlen aufgeführt.

- Die Wurzel des mittleren quadratischen Fehlers (Root Mean Square Error / rmse), welcher in diesem Fall der Standardabweichung der Fehler u_n entspricht.

$$\text{rmse} = \sqrt{\frac{1}{n} \cdot \sum_{k=1}^{n} u_k^2} \tag{2.27}$$

- Der Mean Baised Error (mbe), welcher angibt, ob tendenziell über- oder unterschätzt wurde.

$$\text{mbe} = \frac{\sum_{k=1}^{n} u_k}{\sum_{h=1}^{n} y_h} \tag{2.28}$$

- Der mittlere absolute prozentuale Fehler (Mean Absolute Percentage Error / mape).

$$\text{mape} = \frac{1}{n} \cdot \sum_{k=1}^{n} \frac{|u_k|}{y_k} \tag{2.29}$$

- Der maximale absolute Fehler (Maximum Absolute Erroe / maxae).

$$\text{maxae} = \max|_n \left(|u_n|\right) \tag{2.30}$$

- Der maximale absolute pronzentuale Fehler (Maximum Absolute Percentage Error / maxape).

$$\text{maxape} = \max|_n \left(\frac{|u_n|}{y_n}\right) \tag{2.31}$$

- Das Bestimmtheitsmaß, um abzuschätzen, ob der Fehler $\underline{\mathbf{u}}$ zufällig oder aufgrund falscher Modellannahmen auftritt.

$$r^2 = \frac{1}{n} \cdot \frac{\sum_{k=1}^{n} y_k^2 - \bar{y}^2}{\sum_{k=1}^{n} \hat{y}_k^2 - \bar{\hat{y}}^2} \tag{2.32}$$

3. Analyse der Datengrundlage

Nachdem die mathematischen Grundlagen der zu erstellenden Modelle erläutert sind, muss deren Struktur festgelegt werden. Maßgebend hierfür ist eine Analyse der zu prognostizierenden Datenverläufe. Datengrundlage dieser Arbeit sind viertelstündliche Mittelwerte der elektrischen Lastnachfrage eines deutschen Versorgungsunternehmens im Zeitraum vom 1. Mai 1998 bis zum 31. April 1999. Zusätzlich liegen noch Temperaturdaten, ebenfalls als viertelstündliche Mittelwerte, für den Erhebungsbereich der vorhandenen Lastnachfragedaten vor. Im ersten Teil dieses Kapitels werden anhand von verschiedenen Lastverläufen prägende Einflüsse auf die Lastnachfrage diskutiert. Der zweite Teil legt besonderes Augenmerk auf autoregressive Einflüsse. Hierzu dienen die in Kapitel 2 vorgestellten Analysefunktionen ACF und $PACF$.

3.1. Verläufe von elektrischer Lastnachfrage

Der Verlauf der elektrischen Lastnachfrage über einen Tag ist von vielen Faktoren abhängig. Eine umfangreiche und detaillierte Aufzählung solcher Faktoren ist in [4] gegeben. In der Diskussion der Lastverläufe sollen nur die im Folgenden aufgezählten Punkte berücksichtigt werden.

- **Tagtyp**: Die verschiedenen Tage werden in sogenannte Tagtypen unterteilt. Man unterscheidet zwischen Werktagen von Montag bis Freitag, Wochenendtagen (wobei hier weiter zwischen Samstagen und Sonntagen differenziert wird), Feiertagen und Brückentagen. Unter Brückentagen versteht man Tage, welche zwischen einem Feiertag und einem Wochenende eingeschlossen sind.

- **Temperatur**: Ihr Einfluss spielt vor allem in den kalten Jahreszeiten eine Rolle. Sinken die Temperaturen, werden verstärkt elektrische Verbraucher zu Heizzwecken eingeschaltet, was zu einer Erhöhung der Grundlast führt. Am deutlichsten ist der Einfluss in den Nachtstunden. Elektrische Nachtspeicherheizungen nutzen günstige Nachtstromtarife, um Wärme zu erzeugen, welche dann am Tag genutzt werden kann. Dies führt zu einer deutlichen Erhöhung der sonst niedrigen Lastnachfrage in der Nacht. Tem-

peraturschwankungen im Sommer haben keine Auswirkungen auf die Lastnachfrage.

- **Licht**: Schlechte Lichtverhältnisse haben die Nutzung künstlichen Lichtes zur Folge, was die elektrische Lastnachfrage ansteigen läßt. Die Lichtverhältnisse sind durch den Stand der Sonne und den Bedeckungsgrad beschreibbar. Im Herbst und Winter ist der Stand der Sonne niedrig. Ein bedeckter Himmel kann sogar tagsüber die Nutzung künstlicher Beleuchtung erforderlich machen. Im Sommer sind die Lichtverhältnisse in der Regel auch bei bedecktem Himmel ausreichend. Im Sommer ist der Einfluss des Lichtes nur in Form eines Maximums durch den Sonnenuntergang am Abend beobachtbar. Daten zu Sonnenauf- und -untergangszeitpunkten sowie dem Bedeckungsgrad lagen für diese Arbeit entweder nicht vor oder konnten aus Zeitgründen nicht direkt berücksichtigt werden. Die Aussagen in der folgenden Diskussion zum Einfluss des Lichtes sind folglich empirischer Natur. Zudem geht das Licht nicht als äußere Einflussgröße in die Modellbildung ein. Vielmehr soll eine geeignete Wahl der adaptiven und autoregressiven Modellstruktur es ermöglichen, wiederkehrende, saisonal abhängige Auswirkungen des Lichtes, wie das Maximum am Abend, indirekt zu berücksichtigen.

- **Verbraucherverhalten**: Wie auch in [4] erwähnt, können eigenständiges Verbraucherverhalten und die Reaktion auf äußere Einflüsse wie Licht und Temperatur nur schwer voneinander getrennt werden. Der Aspekt des reinen eigenständigen Handelns lässt sch nicht in Form von Daten in einem Prognosemodell integrieren. Zyklisch auftretende Effekte, wie der tägliche Gang zur Arbeit oder das Zubereiten von Mahlzeiten zur Mittagszeit sollen, wie im Falle des Lichtes über die autoregressive Struktur in dem Modell Niederschlag finden. Ein wichtiger Aspekt in Bezug auf das Verbraucherverhalten ist die freie Wahl des Stromanbieters im liberalisierten Strommarkt. Durch das Weggehen oder Hinzukommen großer Abnehmer, wie Betriebe es sind, können in der Lastnachfrage Sprünge auftreten. Ein Anliegen dieser Arbeit ist, wie schon erwähnt, die Berücksichtigung solcher Effekte durch einen adaptiven Modellansatz. Im verwendeten Datensatz ließen sich aber keine derartigen Sprünge identifizieren. Auf Grund dessen kann in dieser Arbeit nicht geklärt werden, ob ein adaptives Vorgehen dieses leisten kann.

Alle oben aufgeführten Punkte lassen sich anhand der im Folgenden dargestellten Lastverläufe belegen. Abbildung 3.1 zeigt typische Lastverläufe von Werktagen (jeweils Mittwoch) zu den verschiedenen Jahreszeiten. Angaben zu Datum und mittlerer Tagestemperatur sind der Abbildung zu entnehmen. Die Diskussion dieser Lastverläufe soll nicht nur den jahreszeitlich variierenden Einfluss von Licht und Temperatur deutlich machen, sondern auch einen groben Überblick über den Tageslastverlauf eines Werktages geben. Gemeinsam ist allen vier Lastverläufen,

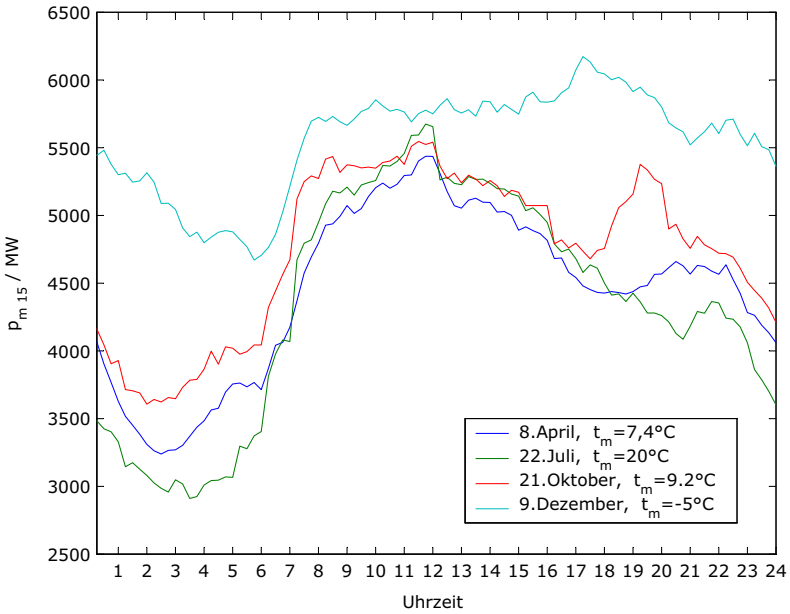

Abbildung 3.1.: Elektrischer Lastnachfrageverlauf für Werktage (Mittwoch) zu verschiedenen Jahreszeiten.

dass die Lastnachfrage nachts geringer ist als tagsüber. Der Wintertag weist die höchste Grundlast auf, und am deutlichsten erkennbar die höchste Lastnachfrage in der Nacht. Grund hierfür ist die schon erwähnte Nutzung von Beleuchtung, Heizung und Nachtspeicherheizungen. Die Verläufe für Sommer, Herbst und Frühling liegen jeweils darunter, wobei der Verlauf des Sommers am niegrigsten ist. Für diese drei Verläufe lassen sich einige Gemeinsamkeiten feststellen, wobei nicht auf Details eingegangen werden soll.

- Der morgentliche Anstieg zwischen 4 Uhr und 5 Uhr kennzeichnet den Beginn des Arbeitstages. Ab circa 8 Uhr hat die Arbeit in den meisten Betriebe begonnen, der Verlauf wird infolgedessen flacher.

- Das Maximum zwischen 11.30 und 11.45 Uhr. Hier überlagert sich die intensive Nutzung elektrischer Verbraucher zur Mittagszeit in privaten Haushalten mit dem Lasteinbruch, verursacht durch die Mittagspause, in den Betrieben.

- Der Abfall der Lastnachfrage über den Nachmittag hin zum Abend. Er kennzeichnet das allmähliche Ende des Arbeitstages.

- Das Maximum am Abend. Es kann, wie schon erwähnt, durch den Zeitpunkt des Sonnenunterganges und die daraus resultierende verstärkte Nutzung von Beleuchtungen erklärt werden.

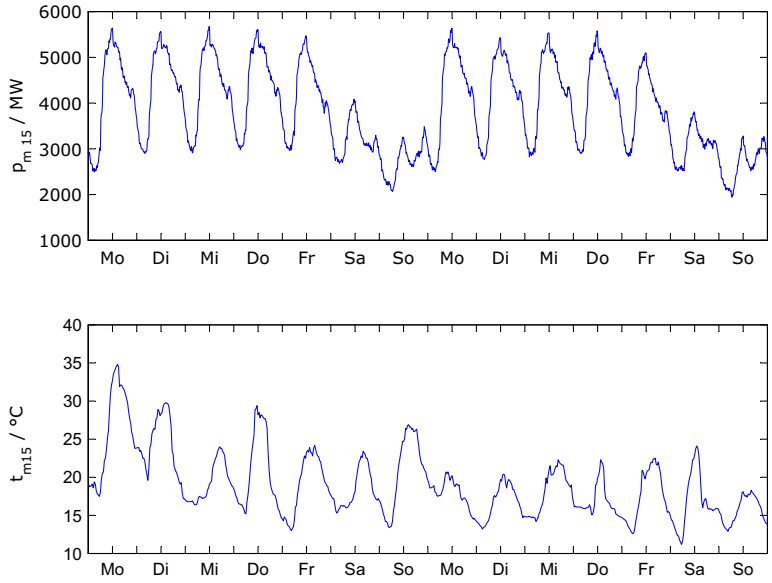

Abbildung 3.2.: Elektrischer Lastnachfrageverlauf für zwei Wochen im Sommer (20. Juli bis 2. August) und zugehöriger Temperaturverlauf.

Der Verlauf des Wintertages weicht in einigen der genannten Punkte ab. Der morgentliche Anstieg wird durch die Nutzung von Nachtspeicherheizungen überlagert und beginnt somit zu einem späteren Zeitpunkt. Für das Verschwinden des Maximums am Mittag ist keine Erklärung gefunden worden. Möglich ist, dass dies mit der Zusammensetzung des Kundenstammes im hier betrachteten Erhebungsgebiet zusammenhängt. Der langsame Anstieg während des Nachmittags kann durch zurückgehende Temperaturen und eine Verschlechterung der Lichtverhältnisse begründet werden. Nur das Maximum am Abend lässt sich wieder wie für alle Tage deuten.

Die bis hierher erläuterten Punkte verdeutlichen nicht nur den Einfluss von Temperatur und Licht, sondern zeigen auch, inwiefern ein adaptives, autoregressives Prognosemodell die geringe Anzahl äußerer Einflussgrößen zur Beschreibung der Lastverläufe in dieser Arbeit kompensieren kann. Da durch den autoregressiven Ansatz die Daten auf sich selbst bezogen werden, können regelmäßig auftretende Merkmale wie die Maxima und Lastanstiege beziehungsweise -abfälle auf diese Weise abgebildet werden. Ein adaptives Vorgehen berücksichtigt die saisonale Abhängigkeit dieser Effekte.

Die Abbildungen 3.2 und 3.3 dienen sowohl als weiterer Beleg für den Einfluss der Temperatur als auch der Verdeutlichung der Tagtypensystematik. Zu sehen ist jeweils der Verlauf der Lastnachfrage, einmal für zwei Wochen im Sommer (Abbildung 3.2) und einmal für zwei Wochen im Winter (Abbildung 3.3),

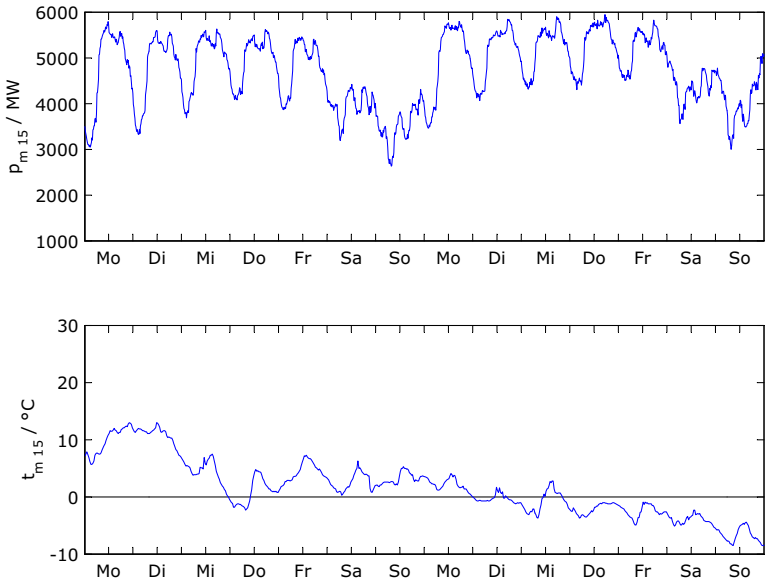

Abbildung 3.3.: Elektrischer Lastnachfrageverlauf für zwei Wochen im Winter (9. bis 22. November) und zugehöriger Temperaturverlauf.

sowie die dazu gehörenden Temperaturverläufe. Deutlich erkennbar ist, dass für sinkende Temperaturen in den Winterwochen die Last kontinuierlich ansteigt. In den beiden Sommerwochen sinken die Temperaturen zwar auch, befinden sich aber auf einem höheren Niveau. Ein Einfluss auf den Lastverlauf kann hier nicht identifiziert werden.

Dass die zu Beginn dieses Abschnittes definierten Tagtypen unterschiedliche charakteristische Lastverläufe aufweisen, ist in beiden Abbildungen zu erkennen. Die Verläufe von Samstag und Sonntag grenzen sich deutlich von denen der Werktage ab. Sie weisen eine deutlich geringere Grundlast auf, wobei die des Sonntags noch unter der des Samstags liegt. Grund hierfür ist die geringe Lastnachfrage in den Betrieben, welche zum Großteil am Wochenende geschlossen sind. Eine genaue Diskussion des Verlaufes von Wochenendtagen soll aus Gründen des Umfangs hier nicht erfolgen. Abbildung 3.4 zeigt beispielhaft den Verlauf der beiden anderen Tagtypen, der Feier- und Brückentage. Der Feiertag, hier der Donnerstag der zweiten Woche, schließt den Freitag zwischen sich und dem Wochenende ein, was diesen per Definition zum Brückentag macht. Der Feiertag ist in seinem Verlauf nahezu identisch mit dem Sonntag; diese Erkenntnis liegt durchaus nahe, da anzunehmen ist, dass an Feiertagen in ebenso wenigen Betrieben gearbeitet wird wie an Sonntagen. Ein Brückentag wird von vielen Arbeitnehmern als verlängertes Wochenende genutzt. Sein Verlauf ist ähnlich zu dem eines Werktages, nur geringer. Ihn mit einem Samstag zu vergleichen führt zu keinem Erfolg.

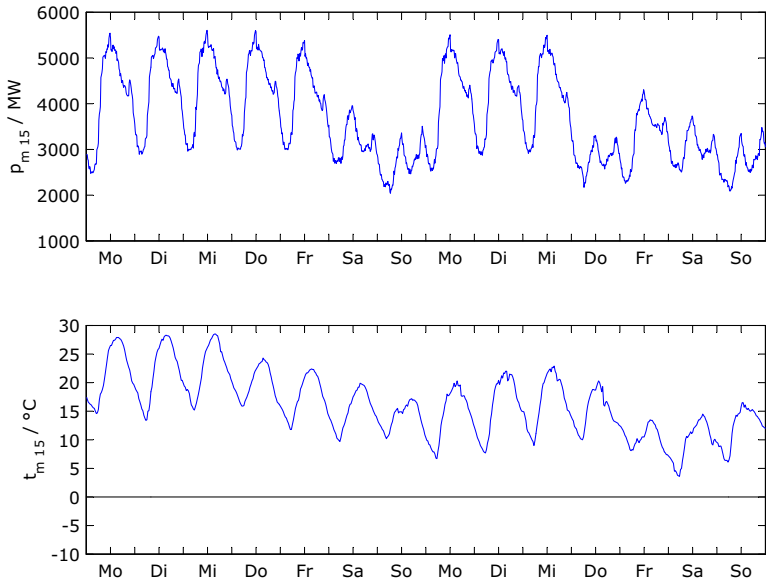

Abbildung 3.4.: Elektrischer Lastnachfrageverlauf mit Feiertag und Brückentag (Zeitraum 11. bis 24.Mai) und zugehöriger Temperaturverlauf.

Da die Verläufe verschiedener Tagtypen sich stark voneinander unterverschieden, wird für jeden Tagtyp ein separates Prognosemodell mit eigener Datenbasis vorgesehen. Dies ist notwendig, um robuste Prognoseparameter $\underline{\hat{\beta}}$ zu erhalten. Zwei Kriterien sollte die Datenbasis erfüllen:

- Die Datenbasis muss so umfangreich wie möglich sein (vergleiche Kapitel 2).
- Die enthaltenen Daten sollten regelmäßig sein, das heißt einen möglichst gleichmäßigen Verlauf aufweisen (vergleiche hierzu auch [4]).

Während für Werktage die tagtypenspezifische Unterteilung von Modellen und zugehöriger Datenbasis die bestmögliche Robustheit der Prognoseparameter gewährleistet, führt das Vorgehen bei den anderen Tagtypen zu Problemen. Durch die tagtypenspezifische Unterteilung der Daten wird die Datenbasis für Samstage, Sonntage und Feiertage, welche ja wie Sonntage behandelt werden sollen, sehr gering, wodurch der erste Punkt der oben genannten Kriterien verletzt wird. Dies hat, wie später in Kapitel 5 zu sehen sein wird, starken Einfluss auf die Ergebnisse. Brückentage müssen bei einem solchen Vorgehen sogar ganz aus der Prognose ausgeschlossen werden. Da sie weder durch einen anderen Tagtypen erklärt werden können, noch im betrachteten Zeitraum in ausreichender Anzahl vorliegen, ist die spezifische Datenbasis zur Berechnung der Prognoseparameter

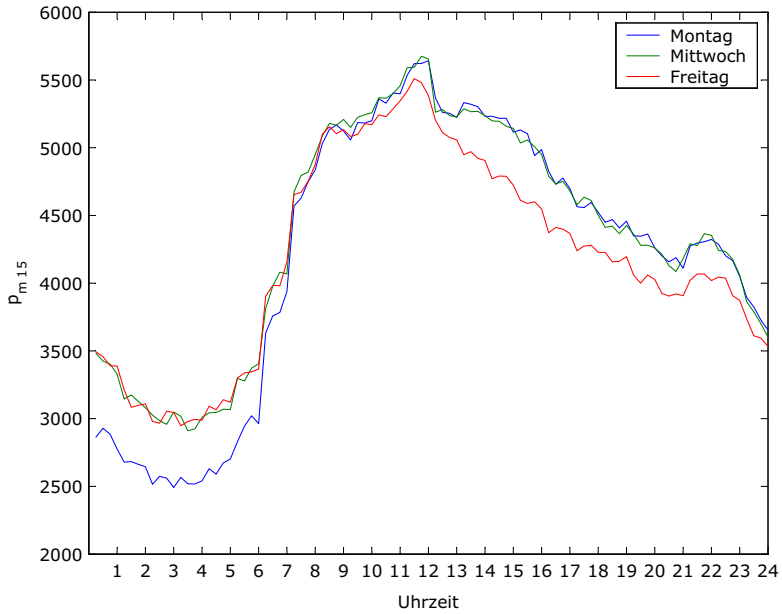

Abbildung 3.5.: Elektrische Lastnachfrage verschiedener Werktage.

zu gering. Die Umsetzung der tagtypspezifischen Unterteilung von Modellen und Datenbasis ist in Kapitel 4 beschrieben.

Die Zusammenfassung aller Werktage zu einem Tagtyp ist auf den ersten Blick gerechtfertigt, betrachtet man die Werktage jedoch genauer, so erkennt man geringe Unterschiede. Abbildung 3.5 zeigt den Verlauf der Tage Montag, Mittwoch und Freitag der ersten der beiden Beispielwochen in Abbildung 3.2 nocheinmal genauer im Vergleich. Auffällig ist zunächst der niegrigere Verlauf der Last in den Nacht- und Morgenstunden des Montages. Erklärt werden kann dies dadurch, dass in der Nacht zwischen Sonntag und Montag in weniger Betrieben gearbeitet wird als in den Nächten zwischen Werktagen (vergleiche hierzu auch die Abbildungen 3.2, 3.3 und 3.4). Zweites ist zu erkennen, dass der Lastverlauf am Freitag von nachmittags an bis in die Nacht flacher verläuft als an den anderen beiden Tagen. Auch hier liegt der Grund im Verhalten der Betriebe, in denen freitags in der Regel früher Feierabend gemacht wird beziehungsweise die Maschinen zur Wartung stillgelegt werden. Die Unterschiede sind jedoch nicht gravierend genug, um eine eigene Tagtypenkategorie für Freitage und Montage einzuführen, zumal dies ja die oben genannten Probleme der Datenbasisverkleinerung mit sich bringt. Vielmehr soll diesen Phänomenen durch die Einführung spezieller Abbildungskonstanten für die Prognose eines Montages beziehungsweise eines Freitages Rechnung getragen werden. Die genaue Vorgehensweise hierfür ist in Kapitel 4 beschrieben.

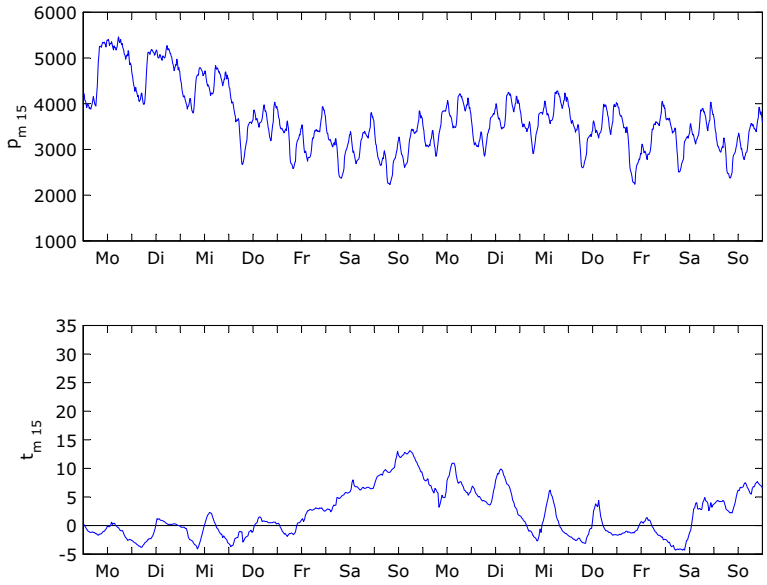

Abbildung 3.6.: Elektrischer Lastnachfrageverlauf für den Zeitraum um Weihnachten herum (21. Dezember bis 3. Januar) und zugehöriger Temperaturverlauf.

Alle bisher diskutierten Verläufe lassen sich auf Basis der zu Beginn aufgezählten Einflussgrößen erklären. Für den Bereich um Weihnachten und Neujahr gelang dies jedoch nicht. Der Verlauf aus Abbildung 3.6 lässt ab dem Mittwoch der ersten der beiden Wochen (hier ein Tag vor Heiligabend) keine tagtypenspezifische Unterscheidung mehr zu. Ein Einfluss der Temperatur wie in Abbildung 3.3 ist ebenfalls nicht zu erkennen. Wie sich in Kapitel 5 herausstellen wird, bereitet dieser Bereich bei der Prognose besondere Probleme.

3.2. Analyse der Daten mit ACF und PACF

Durch die in Abschnitt 3.1 durchgeführte Diskussion der Lastdaten ist es möglich geworden, eine Aussage darüber zu treffen, auf welche Art äußere Einflussgrößen zu berücksichtigen sind und wo die Vorteile eines adaptiven autoregressiven Vorgehens liegen. Die autoregressive Struktur des Modelles selbst, genauer der Parameter p konnte bisher jedoch nicht ermittelt werden. Hierzu dienen die in Abschnitt 2.2 vorgestellten Analysefunktionen *ACF* und *PACF*. Beide sind in einem, in der Programmiersprache MATLAB™ verfassten Skript, integriert. Der Quellcode dieses Skriptes, mit der Bezeichnung *checkdata.m* befindet sich im Anhang B . Das Skript übernimmt einen Datenvektor der Länge n, berechnet für diesen die Werte von *ACF* und *PACF* bis zum maximal möglichen zeitlichen Lag

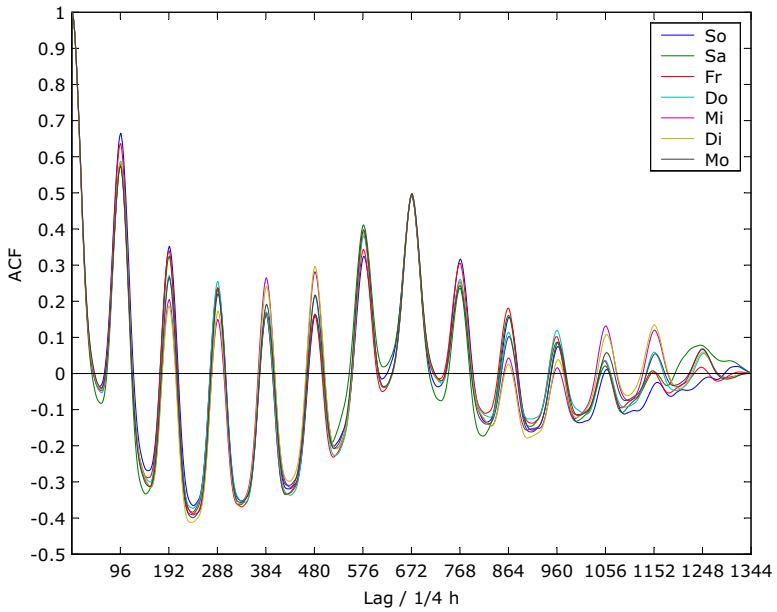

Abbildung 3.7.: *ACF*, berechnet für alle Tage der ersten der beiden Wochen aus Abbildung 3.2 .

$(n-1)$ und gibt die Ergebnisse wieder als Vektor aus.

Eine erste Analyse untersucht Lastdatenvektoren mit einer Länge von 14 Tagen. Der Zeitraum ist so gewählt, dass eine Robustheit der Werte für *ACF* und *PACF* bis zu einem Lag von sieben Tagen, also einer Woche (vergleiche Abschnitt 2.2) gegeben ist. Da die Korrelation zwischen einem Wert der Gegenwart und einem beliebigen Wert der Vergangenheit Ziel der Analyse ist, werden sowohl *ACF* als auch *PACF* rückwärtig berechnet, die Lastdatenvektoren werden also vor der Übergabe an das Skript umgedreht. Jede Analyse vergleicht jeweils sieben *ACF*-Verläufe, rückwärtig für jeden Tag einer Beispielwoche berechnet. Abbildung 3.7 zeigt das Ergebnis einer solchen Analyse, berechnet für alle Tage der ersten der beiden Wochen im Sommer aus Abbildung 3.2. Die lokalen Maxima in Schritten von 96 Viertelstunden verdeutlichen die Periodizität der Lastdaten. Bemerkenswert ist der sehr einheitliche Verlauf der verschiedenen Kurven, ein Beleg dafür, dass der Einfluss der Temperatur hier sehr gering ist, da bei variierender Temperatur (vergleiche Abbildung 3.2) die Werte für die Autokorrelation sehr ähnlich sind. Das Maximum bei einem Lag von 672 Viertelstunden (entspricht einer Woche) durchlaufen die Kurven sogar im gleichen Wert. Dies deckt sich mit den in Abschnitt 3.1 getroffenen Aussagen bezüglich der Tagtypen. Die Datenpunkte der Verläufe von Tagen des gleichen Typs, zwischen denen immer ein Zeitraum von einer Woche liegt, weisen demnach eine gleich hohe Korrelation auf.

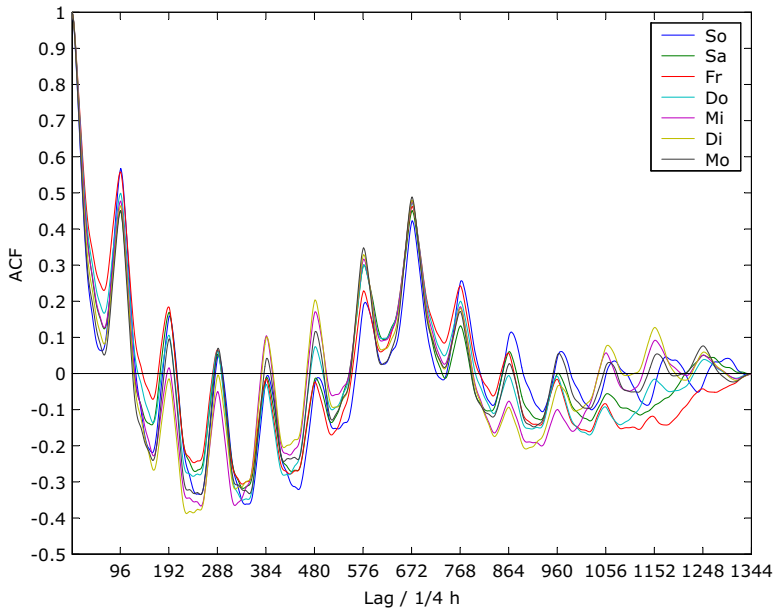

Abbildung 3.8.: *ACF*, berechnet für alle Tage der zweiten der beiden Wochen aus Abbildung 3.3 .

Das Ergebnis einer identischen *ACF* Analsyse für die zweite der beiden Wochen im Winter aus Abbildung 3.3, dargestellt in Abbildung 3.8, zeigt einen wesentlich uneinheitlicheren Verlauf der einzelnen Kurven untereinander. Zwar weisen alle Kurven weiterhin Maxima in regelmäßigem Abstand von 96 Viertelstunden auf; die Bandbreite, in denen sie diese Maxima durchlaufen, ist jedoch wesentlich höher als es in Abbildung 3.7 der Fall ist. Nur im Bereich des Maximums bei 672 Viertelstunden ist wiederum eine Annäherung der einzelnen Kurven zueinander zu erkennen. Dieser Befund bestätigt die Aussagen aus Abschnitt 3 über den steigenden Einfluss der Temperatur in kalten Jahreszeiten, was den autoregressiven Einfluss schwächt. Auch ein Feiertag im Datenzeitraum stört den einheitlichen Verlauf der *ACF* Kurven. Abbildung 3.9 zeigt die Analyse der zweiten Woche aus Abbildung 3.4. Die Kurven für Sonntag, Samstag, Freitag und Donnerstag stehen für die Datenbereiche, in denen der Feiertag und auch der Brückentag enthalten sind. Sie weichen vom Verlauf der anderen beiden Kurven ab, am deutlichsten zu erkennen bei einem Lag von 672 Viertelstunden.

Der letzte in Abschnitt 3 dargestellte Bereich ist der um Weihnachten und Neujahr herum. Das Ergebnis der Analyse dieses Bereiches, zu sehen in Abbildung 3.10, zeigt deutlich, dass in diesem Bereich kein einheitlich autoregressiver Zusammenhang der Daten mehr gegeben ist.

Eine ähnlich umfangreiche Analyse der Lastverläufe mit Hilfe der *PACF* würde den Rahmen dieser Arbeit sprengen. Abbildung 3.11 zeigt zwei *PACF*

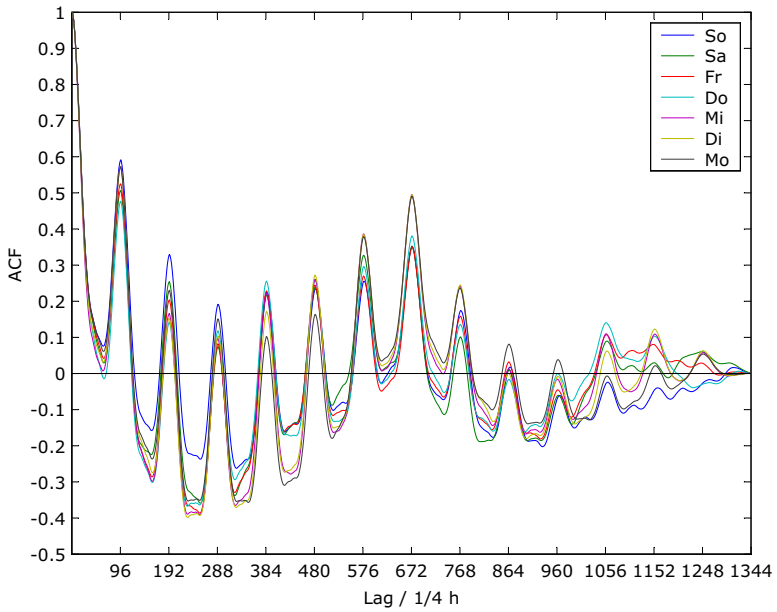

Abbildung 3.9.: *ACF*, berechnet für alle Tage der zweiten der beiden Wochen aus Abbildung 3.4 .

Verläufe. Die Daten umfassen wie bei der *ACF*-Analyse einen Zeitraum von zwei Wochen und die Berechnung der *PACF* erfolgt ebenfalls rückwärtig. Der obere Verlauf ist für den Mittwoch aus der ersten der beiden Sommerwochen aus Abbildung 3.2 ermittelt worden, der untere für den Mittwoch aus der zweiten der beiden Winterwochen aus Abbildung 3.3. In beiden Abbildungen ist deutlich der hohe Peak bei 96 Viertelstunden zu erkennen. Er kann bei jeder *PACF*-Analyse identifiziert werden. Weitere Peaks in Intervallen von 96 Viertelstunden, wie im Falle der *ACF* treten zwar auf, sind aber wesentlich geringer ausgeprägt als der erste Peak und auch nicht regelmässig. Grund hierfür ist die Eigenschaft der *PACF*, nicht Mittelwerte von Korrelationen, sondern Korrelationen zwischen einzelnen Punkten zu berechnen. Singuläre Effekte prägen somit deutlich jeden der einzelnen *PACF* Verläufe und erschweren die Ableitung allgemeiner Aussagen. In [1] ist erwähnt, dass, unter gewissen Voraussetzungen, die Werte der *PACF* den Schätzwerten der autoregressiven Koeffizienten $\underline{\hat{\beta}}$ des *AR*-Ansatzes entsprechen. Vorversuche ergaben zwar, dass dies im Fall der vorliegenden Arbeit nicht zutrifft, die Vorgehensweise selbst gibt jedoch den Anstoss dazu, die klassische autoregressive Struktur eines *AR*-Ansatzes abzuändern. Gleichung 2.6 stellt eine solche klassische Struktur dar. Hier werden alle Werte bis hin zu einem maxmimalen zeitlichen Lag p als autoregressive Größen verwendet. Bei Berücksichtigung der dargestellten *PACF*-Verläufe erscheint es jedoch sinnvoll, nicht alle Werte bis hin zu einem Lag p, sondern nur isolierte Werte zu verschiedenen Lags p_i, welche

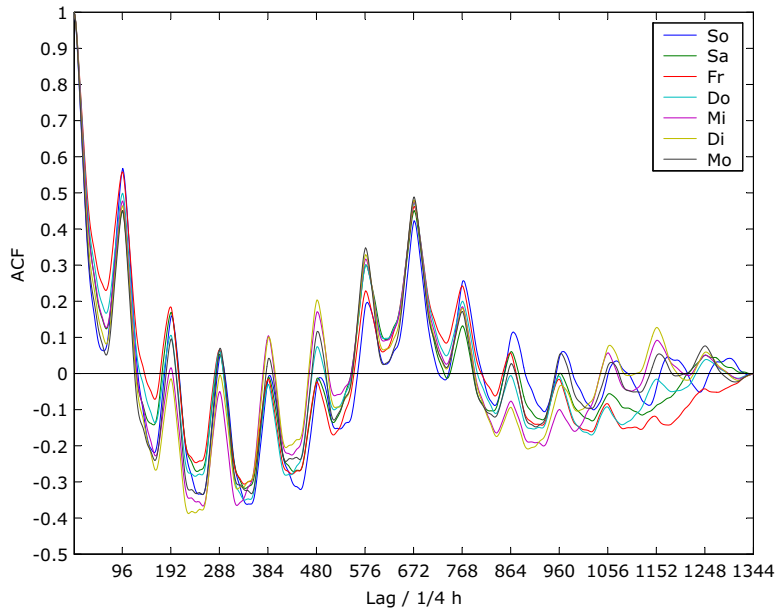

Abbildung 3.10.: *ACF*, berechnet für alle Tage der zweiten der beiden Wochen aus Abbildung 3.6 .

sich durch deutlich ausgebildete Maxima hervorheben, als autoregressive Werte zu berücksichtigen. Solch ein Wert ist hier bei einem Lag von 96 Viertelstunden gegeben. Zudem sollen Werte für einen zeitlichen Lag von einer Woche berücksichtig werden. Dies lässt sich mit den Ergebnissen der *ACF*-Ananlyse bergründen. Die Berücksichtigung von Werten der angegebenen Lags ist auch die einzigste, in Praxis anwendbare Methode. Zur Prognose des kommenden Tages sind die aktuellsten, zur Verfügnung stehenden Daten, die, genau 96 Viertelstunden, also einen Tag zuvor. Ein Vorgehen, bei dem alle Werte bis hin zu einem maximalen Lag in den Prognosegleichungen verwendet werden, ist zum Beispiel in [3] dargelegt.

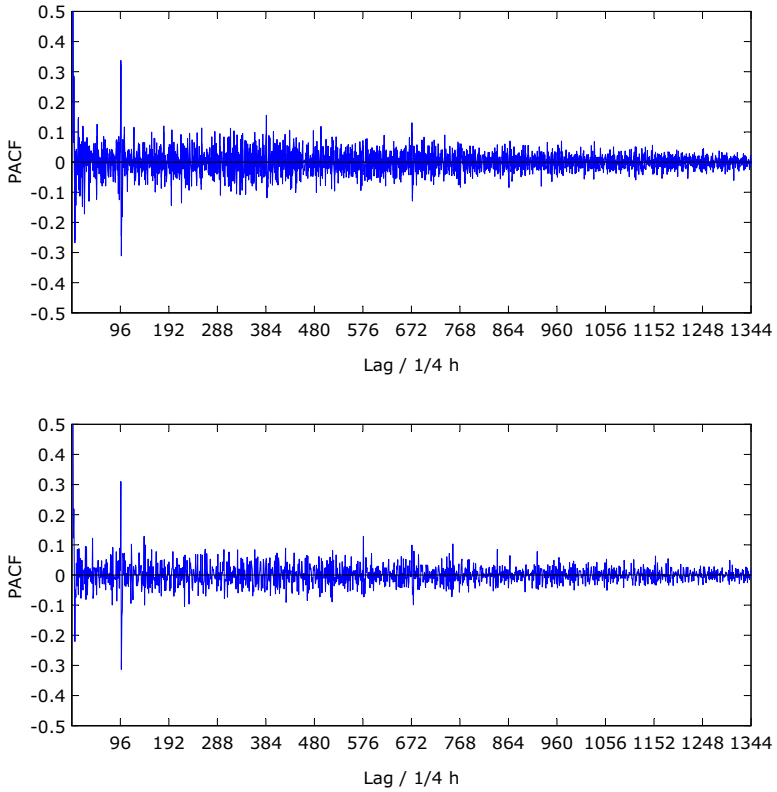

Abbildung 3.11.: Oben: *PACF*, berechnet für den Mittwoch der ersten der beiden Wochen aus Abbildung 3.2 . Unten: *PACF*, berechnet für den Mittwoch der zweiten der beiden Wochen aus Abbildung 3.3

4. Methodisches Vorgehen

Die in Kapitel 3 gewonnenen Erkenntnisse fließen in die Ableitung der notwendigen mathematischen Modellgleichungen ein. In diesem Kapitel ist die schrittweise Entwicklung der in dieser Arbeit realisierten Modelle beschrieben. Zwei erste grundlegende Modelle bauen auf einem reinen AR-Ansatz auf. Beide können nur Werktage prognostizieren, wobei eines der beiden zusätzlich adaptiv ist. Die adaptive Vorgehensweise wird für alle weiteren Modelle beibehalten. Unter Einbeziehung nichtautoregressiver Einflussgrößen erfolgt der Schritt zu einem ARX-Ansatz. Als solche Einflussgrößen gelten sowohl Abbildungskonstanten, welche korrigierend auf die Abbildung von Werktagen wirken, als auch die Temperatur, deren Einfluss in den weiteren Modellen optional berücksichtigt werden kann. Aufbauend auf dem ARX-Ansatz und unter Einbezug eines MA-Anteiles erfolgt im letzten Schritt die Ableitung eines $ARMAX$-Ansatzes. Beide, der ARX- als auch der $ARMAX$-Ansatz ermöglichen mittels separater Gleichungen die Prognose von Wochenend- und Feiertagen. Dabei kommt zusätzlich ein integratives Verfahren zum Einsatz.

Der letzte Abschnitt dieses Kapitels widmet sich dem Aufbau und der Funktonsweise des in der Programmiersprache MATLAB™ verfassten Skripts *prognoser*, welches sowohl den ARX- als auch den $ARMAX$- Ansatz enthält.

4.1. AR(I)X-Ansatz

Als Grundlage für alle weiteren Ansätze dient der AR-Ansatz. Unter Berücksichtigung der in Kapitel 3 gewonnenen Erkenntnisse sind folgende Punkte für die abzuleitenden Modelle festzulegen:

- **Tagtypenspezifische Unterteilung der Modelle:** Für jeden der definierten Tagtypen ist ein separates Modell vorgesehen, wobei Feiertage wie Sonntage behandelt werden und Brückentage ganz aus der Prognose ausgeschlossen sind.

- **Die autoregressive Struktur:** Wie schon in Abschnitt 3.2 dargelegt, sollen in dieser Arbeit isolierte Werte verschiedener Lags p als autoregressive Grössen berücksichtigt werden. Jede Viertelstunde eines Werktages wird

mit der gleichen Viertelstunde des Werktages zuvor und des Werktages einer Woche zuvor erklärt. Ein Wochenendtag kann nur mit dem gleichen Tag eine Woche zuvor erklärt werden, da der Tag zuvor einem anderen Tagtyp entspricht.

- **Die Datenbasis:** Bei der Prognose eines Tageslastverlaufes mit einem AR-Ansatz sind zum Antrainieren der Prognosekoeffizienten $\underline{\hat{\beta}}$ als auch für die Prognose eines Tageslastverlaufes Basislastdaten notwendig. Diese Datenbasis ist innerhalb von MATLAB^{TM} als Matrix mit der Bezeichnung *data_prog* abgespeichert. Tabelle 4.1 zeigt ihren Aufbau. In jeder Zeile stehen alle 96 viertelstündlichen Mittelwerte der Last (p_{m15}) eines Tages. Die ersten beiden Zeilen enthalten Parameter zur Identifizierung des Tagtypen. dy beschreibt, mit Werten von 0 bis 6, welcher Wochentag vorliegt, während ft, mit den Werten 0 und 7, und bt, mit Werten von 0 und 12, angeben, ob der betreffende Tag ein Feier- beziehungsweis Brückentag ist. Die Werte sind so gewählt, dass durch einfaches Addieren von dy mit ft, beziehungsweise bt jede mögliche Kombination erfasst werden kann. Um das Auslesen der Daten bei einer Werktagsprognose zu erleichtern, sind in einer zweiten Lastdatenmatrix mit der Bezeichnung *data_wt_prog* nur Werktage gespeichert.

- **Der Trainingsbereich:** Damit wird der zeitliche Bereich bezeichnet, aus dem die Daten zum Antrainieren der autoregressiven Prognosekoeffizienten entnommen werden. Ein mögliches Vorgehen ist, den gesamten hier zur Verfügung stehenden Datenzeitraum zur Berechnung der Prognosekoeffizienten zu verwenden. Die Koeffizienten werden somit einmal fix ermittelt und anschließend zur Prognose eines jeden Tages aus diesem Datenzeitraum angesetzt. Auf diese Weise soll in dem ersten unten beschriebenen Modell vorgegangen werden. Alle weiteren Modelle sind adaptiv. Hier werden die Prognosekoeffizienten nicht fix, sondern bei jeder Prognose neu berechnet. Der Trainingszeitraum für ein solches Vorgehen soll in dieser Arbeit auf vier Wochen vor dem zu prognostizierenden Tag festgelegt werden. Diese Wahl stellt einen Kompromiss zwischen einem möglichst großen Zeitraum, welcher die Robustheit der Prognosekoeffizienten gewährleistet, und einem möglichst kleinen Zeitraum, zur besseren Abbildung der Kundenfluktuation und anderer Schwankungseffekte, dar. Ein weiterer Punkt in Bezug auf den Trainingszeitraum ist die gewünschte Gleichförmigkeit der Daten. Feiertage und Brückentage stellen Störungen dar, welche bei der Prognose eines Werktages die Robustheit der Prognosekoeffizienten beinträchtigen würden. Aus diesem Grund werden alle Feier- und Brückentage durch Prognosen der entsprechenden Werktage ersetzt und die so modifizierten Daten in einer Matrix mit den Bezeichnungen *data_wt_tr* gespeichert. Aus ihr können anschließend die Daten für das Training entnommen werden. Ersetzt man die

Feier- beziehungsweise Brückentage mit den Werktagen einen Tag oder eine Woche zuvor, so führt das zu schlechteren Ergebnissen. Dies haben Vorversuche ergeben.

Unter Berücksichtigung dieser Punkte lässt sich ein erstes Prognosemodell mit der Bezeichnung *AR1* ableiten. Dieses Modell soll nur für Werktage gelten und ist, wie schon erwähnt, nicht adaptiv. Das entsprechende MATLAB™ Skript ist mit *prognoser_pure_01* bezeichnet und im Anhang B enthalten. Als Modellansatz ergibt sich folgende Gleichung:

$$\hat{y}_n = \hat{\beta}_0 + \hat{\beta}_1 \cdot y_{(n-96)} + \hat{\beta}_2 \cdot y_{(n-480)} \ . \tag{4.1}$$

Tabelle 4.1.: Struktur einer Lastdatenmatrix.

ft/bt	dy	$p_{m15,1}$	$p_{m15,2}$	...	$p_{m15,96}$	Tag	Tagtyp
0	0	2878	2803	...	3607	Montag	Werktag
0	1	3418	3350	...	3651	Dienstag	Werktag
0	2	3499	3447	...	3492	Mittwoch	Werktag
7	3	3296	3168	...	2879	Donnerstag	Feiertag
12	4	2725	2692	...	3291	Freitag	Brückentag
0	5	3171	3027	...	2889	Samstag	Wochenendtag
0	6	2854	2811	...	3072	Sonntag	Wochenendtag
...	...	...	...	...	...	...	...

Zur Berechnung der unbekannten Prognosekoeffizienten $\underline{\hat{\beta}}$ müssen die Orginaldaten aus der Datenbasis entsprechend den Vorgaben zur autoregressiven Struktur in Form eines linearen Gleichungssystemes, ähnlich wie in Gleichung 2.6, gebracht werden. Zur Lösung dieses Gleichungsstemes dient das Least-Squares Verfahren (vergleiche Gleichung 2.5). Anschließend läßt sich eine Prognosegleichung für den zu prognostzierenden Tageslastverlauf $\underline{\hat{\mathbf{y}}} = (\hat{y}_1, \ldots, \hat{y}_{96})$ formulieren:

$$\underline{\hat{\mathbf{y}}} = \underline{\mathbf{e}} \cdot \hat{\beta}_0 + \mathbf{Y} \cdot \underline{\hat{\beta}} \tag{4.2}$$

Da die Werte für Training und Prognose aus den Matrizen *data_wt_tr* und *data_wt* ausgelesen werden, entspricht ein Lag von 480 Viertelstunden (fünf Tage) in Gleichung 4.1 den Werten der vorangehenden Woche. Aus den prognostizierten Daten $\underline{\hat{\mathbf{y}}}$ und den Originaldaten $\underline{\mathbf{y}}$ errechnen sich der Prognosefehler $\underline{\mathbf{u}}$ und somit alle statistischen Maßzahlen.

Das zweite Modell mit der Bezeichnung *AR2* berücksichtigt wiederum nur Werktage, basiert aber jetzt auf einem adaptiven Vorgehen. Die Gleichung für den Modellansatz ist identisch mit dem des ersten Modells. Da die Prognosekoeffizienten jetzt aber nicht mehr einmal fix, sondern für jede Prognose eines Tages

neu berechnet werden müssen, ist in dem entsprechenden Skript mit der Bezeichnung *prognoser_pure_2* eine Schleifenstruktur notwendig. Der adaptive Trainingszeitraum umfasst wie schon erwähnt vier Wochen, also zwanzig Werktage.

Als nächstes soll die Ableitung eines ARX-Modelles (Bezeichnug ARX) beschrieben werden. In diesem Modell werden zusätzlich berücksichtigt:

- Die Abweichungen der Verläufe von Montag und Freitag vom Verlauf der anderen Werktage.
- Die Prognose von Wochenentagen und Feiertagen.
- Der Einfluss der Temperatur.

Der gesamte ARX-Ansatz, inklusive der Prognose von Wochenendtagen, Feiertagen und einer optionalen Berücksichtigung der Temperatur ist in dem Skript *prognoser* integriert, welches sich im Anhang B befindet. Im Folgenden wird erörtert, auf welche Weise die oben genannten Punkte in den Modellansatz integriert sind.

Wie schon in Kapitel 3.1 erwähnt, unterscheiden sich auch verschiedene Werktage bezüglich des Lastnachfrageverlaufes. Montag und Freitag weichen vom Verlauf der anderen Werktage ab. Durch die Verwendung autoregressiver Daten einer Woche zuvor können die beiden Abweichungen zwar abgebildet werden, verfälschen aber trotzdem die Koeffizienten $\underline{\hat{\beta}}$ aus Gleichung 4.1 . Zur Korrektur dienen spezielle Konstanten. Diese nehmen entweder den Wert 1 oder 0 an und werden auch als Dummys bezeichnet.

$$d = \begin{cases} 0 & \text{inaktiv} \\ 1 & \text{aktiv} \end{cases} \tag{4.3}$$

Folgende Tagesabbildungen sind zu beachten:

- Montag auf Dienstag: Abbildungsdummy d_{MoDi}.
- Donnerstag auf Freitag: Abbildungsdummy d_{DoFr}.
- Freitag auf Montag: Abbildungsdummys d_{FrMo_01} und d_{FrMo_02}.

Die Abbildung von Freitag auf Montag muss von zwei verschiedenen Abbildungsdummys übernommen werden, da für die Nacht nach unten und für den Nachmittag nach oben korrigiert werden muss. In Abbildung 4.1 ist das Prinzip dargestellt. Hier ist schematisch die Abbildung jedes einzelnen Werktages dargestellt. Die Kennzeichnung mit einem X gibt an, ob ein Dummy für die jeweilige Werktagsabbildung relevant ist. Darunter sind die Dummys als Vektoren dargestellt. Zu erkennen ist, dass ein Dummy nicht bei der Abbildung jeder der 96 Viertelstunden eines Tages aktiv ist, sondern nur in dem zeitlichen Bereich, in welchem korrigiert werden muss. Die Festlegung der jeweiligen zeitlichen Bereiche erfolgt durch

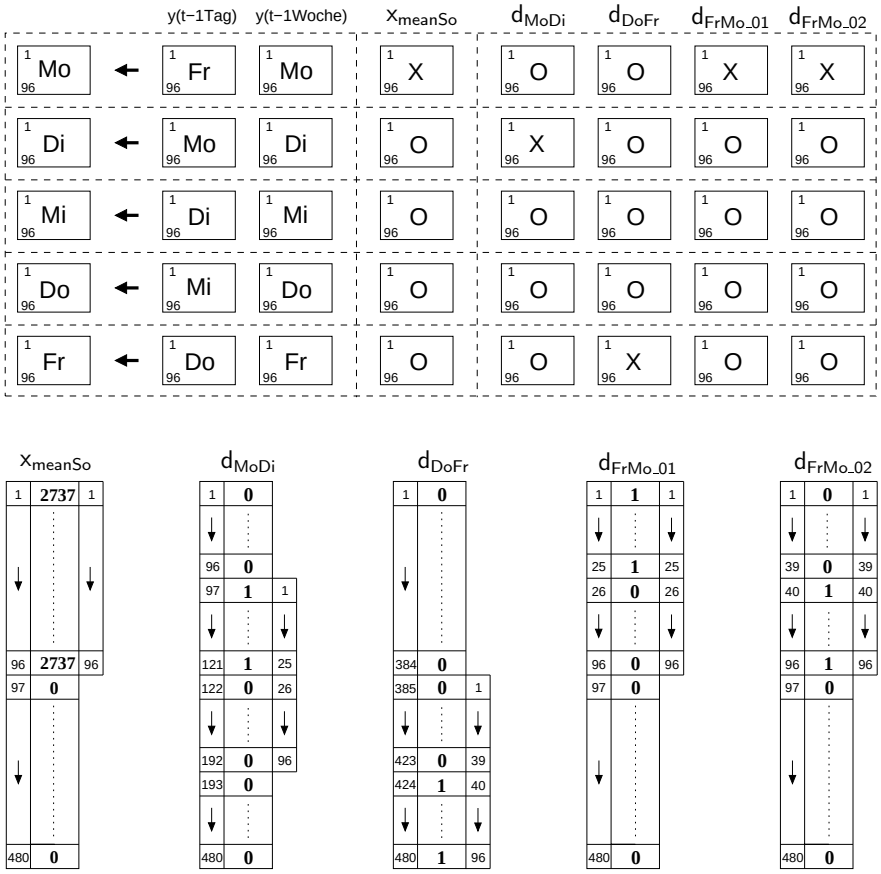

Abbildung 4.1.: Prinzip der Abbildungskonstanten.

grafische Auswertung der entsprechenden Tageslastverläufe, wie Abbildung 3.5 verdeutlicht.

In Abbildung 4.1 ist mit x_{meanSo} noch eine weitere neu eingeführte Abbildungskonstante dargestellt. Sie wirkt auf die Abbildung von Freitag auf Montag ein und ist wie folgt definiert:

$$x_{meanSo} = \begin{cases} \bar{y}_{Sonntag} & \text{Abbildung Freitag auf Montag} \\ 0 & \text{sonst} \end{cases} \tag{4.4}$$

Mit dieser Konstanten soll eine Niveauangleichung an den Sonntag ermöglicht werden, welche ja von sich aus nicht als autoregressive Größe in die Abbildung eingeht. Im Gegensatz zu den Dummys ist x_{meanSo} über den ganzen Bereich der entsprechenden Tagesabbildung aktiv.

Der Ansatz aus Gleichung 4.1, erweitert durch die genannten Punkte, lautet:

$$\hat{y}_n = \hat{\beta}_0 + \hat{\beta}_1 \cdot y_{(n-96)} + \hat{\beta}_2 \cdot y_{(n-480)} + \hat{\beta}_{meanSo} \cdot x_{meanSo} + \hat{\beta}_{MoDi} \cdot d_{MoDi} + \hat{\beta}_{DoFr} \cdot d_{DoFr} + \hat{\beta}_{FrMo\_01} \cdot d_{FrMo\_01} + \hat{\beta}_{FrMo\_02} \cdot d_{FrMo\_02} \quad (4.5)$$

Um die Dummys innerhalb von MATLAB^{TM} zugänglich zu machen, werden sie in Form einer Matrix mit der Bezeichnung *d_wt* abgespeichert. Für das Training kann diese Matrix, entsprechend den Anforderungen umgeformt und in das lineare Gleichungssystem, zur Berechnung der Parameter $\underline{\hat{\beta}}$, eingefügt werden. Für die Prognose hingegen ist nur das Auslesen der entsprechenden Zeile aus *d_wt* notwendig. Die genaue Funktionsweise der hierfür notwendigen Algorithmen ist dem Skript *prognoser* im Anhang B zu entnehmen.

Die neu eingeführten Abbildungskonstanten sind noch für einen weiteren Fall von Bedeutung. Tritt in einer Woche ein Feiertag auf, so hat dies Folgen für die Prognose der anschließenden Werktage. Hier können zwei Fälle unterschieden werden:

- Der Tag zuvor war ein Feiertag. Als Konsequenz verhält sich der zu prognostizierende Tag wie ein Montag, das heißt, das Minimum in der Nacht liegt wesentlich niedriger als normal. Um dies zu berücksichtigen, wird der Abbildungsdummy *FrMo_01* - und bei Bedarf auch *FrMo_02* - in die Prognosegleichung eingeführt. Zusätzlich wird die Konstante x_{meanSo} verwendet, nur dass in diesem Fall der Mittelwert des jeweiligen Feiertages eingesetzt wird.

- Der Tag zwei Tage zuvor war ein Feiertag. Wie oben erläutert, verhält sich ein Werktag, welcher einem Feiertag folgt, wie ein Montag. Für die Prognose des zweiten Tages nach einem Feiertag muß also der Abbildungsdummy *MoDi* mit verwendet werden.

Da alle Koeffinzieten für Dummys und für die Konstante x_{meanSo} bei jedem Trainingszyklus bestimmt werden, ist das geschilderte Verfahren ohne weiteres möglich. Alle denkbaren Konstellationen sind im Skript *prognoser* mittels Fallunterscheidung berücksichtigt.

Für Wochenendtage ist, wie schon erwähnt, ein eigener Ansatz vorgesehen. Dieser lautet:

$$\hat{y}_n = \hat{\beta}_0 + \hat{\beta}_1 \cdot y_{(n-672)} \ . \quad (4.6)$$

Die Daten für Training und Prognose zur Bestimmung eines Wochenendtages werden aus den Matrizen *data_tr* und *data_prog* ausgelesen, ein Lag von 672 Viertelstunden (entspricht sieben Tagen) berücksichtigt also den entsprechenden Wert eine Woche zuvor. Der Trainingszeitraum umfaßt auch hier vier Wochen, wobei jedoch nur die vier entsprechenden Samstage beziehungsweise Sonntage

zählen. Das Vorgehen zum Antrainieren der Koeffizienten und der anschließenden Prognose ist identisch zum Vorgehen bei einem Werktag. Die Prognose eines Feiertages erfolgt in Bezug auf Training und Prognose wie die eines Sonntages. Dies ist unter Berücksichtigung der in Kapitel 3 gewonnenen Erkenntnisse gerechtfertigt.

Bei der Prognose von Wochenend- und Feiertagen mit dem bis hier geschilderten Verfahren ergibt sich ein Problem. Es kommt zu Sprüngen in den Ergebnissen, das heißt, die Differenz zwischen dem Wert der ersten Viertelstunde des zu prognostizierenden Tages, zum Beispiel eines Samstages und der letzten Viertelstunde des vorherigen Tages, in diesem Fall eines Freitages, ist wesentlich zu groß. Ein Grund für diese Sprünge ist die Tatsache, dass ein Wochenendtag mit einem Tag eine Woche zuvor erklärt wird. Es fehlt also der Bezug zum vorherigen Tag. Der Versuch, durch eine Abbildungskonstante, ähnlich wie in Gleichung 4.4 definiert, einen Niveauangleich zu erreichen, erwies sich in Vorversuchen als untauglich. Anstelle dessen dient ein Differenzenverfahren zur Behebung des Effektes. Wie in Kapitel 2 dargelegt, werden bei einem Differenzenverfahren nicht die einzelnen Datenpunkte prognostiziert, sondern die Differenzbeträge zwischen ihnen. Um nach einer erfolgreichen Prognose den gewünschten Tageslastverlauf zu erhalten, müssen die einzelnen Differenzen wieder aufaddiert werden. Startwert für ein solches Aufaddieren ist der Wert der letzten Viertelstunde des vorherigen Tages (im Falle des oben genannten Beispieles wäre dies der Freitag). Durch dieses Anknüpfen an den letzten Wert des vorherigen Tages werden Sprünge in den Ergebnissen vermieden.

Der letzte im *ARX*-Ansatz enthaltene Punkt ist die Berücksichtgung des Einflusses der Temperatur. Innerhalb des Skriptes *prognoser* ist das Einbeziehen der Temperatur als Option vorgesehen. Folgende Punkte werden festgelegt:

- Der Einfluss der Temperatur wird nur für den Zeitraum zwischen 22.00 Uhr und 5.00 Uhr morgens berücksichtigt, und auch nur dann, wenn der Mittelwert der Tagestemperatur unter 15 $°C$ liegt.

- Als Werte gehen die Differenz des Mittelwertes der jeweiligen Viertelstunde zum Wert 15 $°C$ ein.

Die Vorgehensweise, den Einfluss der Temperatur nur in der kalten Jahreszeit in die Prognose mit einzubeziehen, geht aus den in Kapitel 3 gewonnenen Erkenntnissen hervor. Die genauen Werte für den Zeitraum und die Schwellentemperatur ergaben sich in einem Expertengespräch. Die oben genannten Punkte lassen sich mit folgendem Ausdruck in die Gleichungen 4.5 und 4.6 integrieren:

$$\cdots + \hat{\beta}_{temp} \cdot x_{\vartheta} \ . \tag{4.7}$$

Dabei gilt

$$x_{\vartheta} = d_{\vartheta} \cdot (\vartheta_n - 15 \ °C) \tag{4.8}$$

und

$$d_\vartheta = \begin{cases} 1 & \bar{\vartheta} < 15\ °C \vee n\epsilon[22.00, 5.00] \\ 0 & \text{sonst} \end{cases} \tag{4.9}$$

Die Temperaturdaten sind wie die Lastdaten in einer Matrix mit 96 Datenpunkten pro Tag und Zeile gespeichert. Die Bezeichnung dieser Matrix lautet *temp_wt*. Für Training und Prognose wird aus dieser Matrix ausgelesen und die Daten nach obigen Kriterien gefiltert und umgeformt.

4.2. AR(I)MAX-Ansatz

Der *MA*-Ansatz baut vollständig auf dem in Abschnitt 4.1 abgeleiteten *ARX*-Ansatz auf. Der zeitliche Lag q des *MA*-Anteils wird genauso wie der des *AR*-Anteils gewählt. Für die Prognose eines Wochentages ergibt sich somit

$$\hat{y}_n = \hat{\beta}_0 + \hat{\beta}_1 \cdot y_{(n-96)} + \hat{\beta}_2 \cdot y_{(n-480)} + \hat{\beta}_3 \cdot u_{(n-96)} + \hat{\beta}_4 \cdot u_{(n-480)} + \dots \tag{4.10}$$

für die eines Wochenendtages

$$\hat{y}_n = \hat{\beta}_0 + \hat{\beta}_1 \cdot y_{(n-672)} + \hat{\beta}_2 \cdot u_{(n-672)} + \dots\ . \tag{4.11}$$

Das Antrainieren der unbekannten Koeffizienten $\underline{\hat{\beta}}$ verläuft im Prinzip genau wie im Falle des *ARX*-Ansatzes, es muss aber ein anderes Berechnungsverfahren gewählt werden. Wie schon in Kapitel 2.2 erläutert, sind die Gleichungen 4.10 und 4.11 durch den im vorhinein unbekannten Prognosefehler u_n nicht linear. Das hierfür ebenfalls in Kapitel 2.2 beschriebene Verfahren der Minimierung des negativen Log-Likelyhood wird in MATLAB^{TM} mit Hilfe der internen Funktion *FMINSEARCH* gelöst. *FMINSEARCH* bestimmt die Parameter $\underline{\hat{\beta}}$ und $\underline{\mathbf{u}}$ mit Hilfe eines iterativen Suchverfahrens und unter Vorgabe von Startwerten derart, dass die Funktion $S(\underline{\hat{\beta}}, \underline{\mathbf{u}})$ (siehe Gleichung 2.24) minimal wird. Für die Prognose selbst kann der sich für den Traningsbereich ergebende Fehler $\underline{\mathbf{u}}$ als Eingang in den Prognosegleichungen 4.10 und 4.11 verwendet werden, diese Gleichungen sind somit auf herkömmlichen Wege lösbar. Beim Lösen der Trainingsgleichnungen zeigt sich, dass *FMINSEARCH* keine optimale Lösung findet. Die geschätzen Lastverläufe weichen drastisch von den Originalverläufen ab. Als Ursache wird vermutet, dass die vielen Nullen, welche auf Grund der Abbildungskonstanten im X-Anteil enthalten sind, dem Lösungsalgorithmus Probleme bereiten. Aus Zeitgründen ist dieser Aspekt aber nicht weiter verfolgt worden, die Diskussion von Ergebnissen eines *ARMAX*-Ansatzes kann im Rahmen dieser Arbeit somit nicht erfolgen.

Als Abschluss dieses Abschnitts ist das bisher beschriebene Vorgehen bei Training und Prognose, sowohl für den *ARX*- als auch für den *ARMAX*-Ansatz [1], in Abbildung 4.2 nochmals schematisch dargestellt.

[1] Die Berücksichtigung eines *MA*-Anteils ist neben der Berücksichtigung des Temperatureinflusses als weitere Option im Skript *prognoser* integriert.

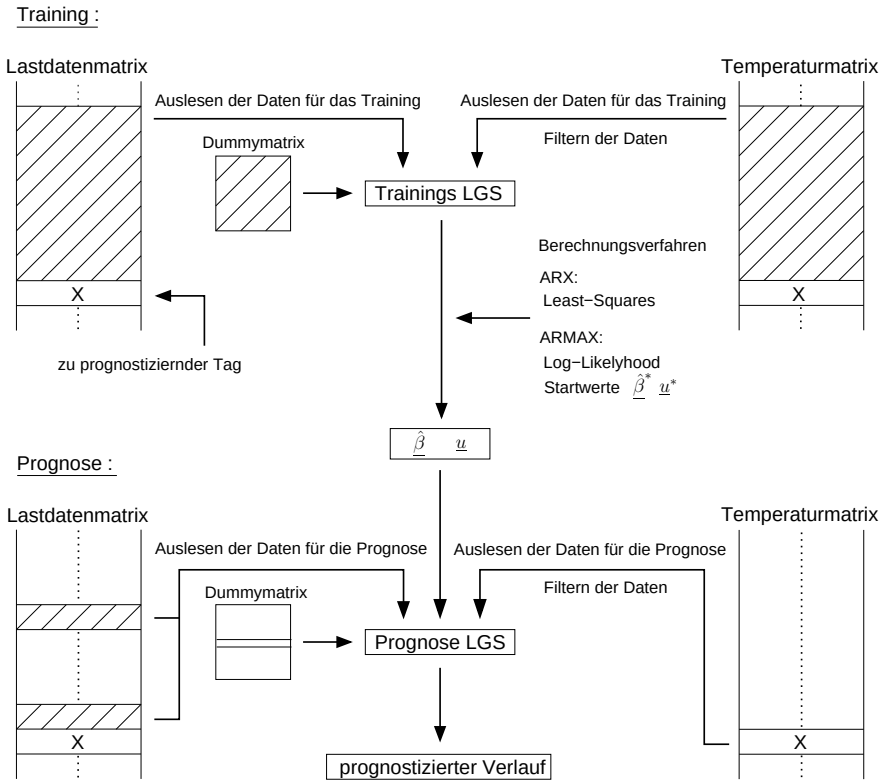

Abbildung 4.2.: Trainings- und Prognoseprinzip.

4.3. Erklärungen zum Programm prognoser

Die genaue Funktionsweise des schon erwähnten MATLAB^{TM} Skriptes *prognoser*, welches sowohl den ARX- als auch den $ARMAX$-Ansatz beinhaltet, soll in diesem Abschnitt erläutert werden. Das Skript ist in mehrere Teile untergliedert. Abbildung 4.3 gibt hierzu einen Überblick. Der Hauptteil, *prognoser_main*, übernimmt mit *zb* und *ze* die Zeilennummern der Datenmatrix *data_prog*, welche den zu prognostizierenden Bereich eingrenzen. *opt*1 legt fest, ob ein *MA*-Anteil, *opt*2, ob die Temperatur berücksichtigt werden soll. In einer Schleife wird nun die Prognose für jeden Tag des gewünschten Bereiches abgearbeitet. Dafür werden die Werte von *dy* und ft bestimmt, um Aussagen über den Tagtypen treffen zu können. In dem für den Tagtypen zuständigen Trainingsskript werden dann die Koeffizienten *beta* bestimmt. *prognoser_main* überprüft für jeden Tag, ob sich im Trainingsdaten-Bereich ein Feiertag oder Brückentag befindet. Ist dies der Fall, so wird dieser in Trainingslastdaten-Matrizen durch die Prognose eines normalen Werktages ersetzt. Im Falle eines $ARMAX$-Ansatzes übernimmt das Skript *arimax* die Berechnung von *beta*, wobei zusätzlich der Prognosefehler des Trainings *err* bestimmt wird. Jetzt kann in dem jeweiligen Prognoseskript der Lastverlauf fit des betrachteten Tages errechnet werden. Die einzelnen Verläufe werden

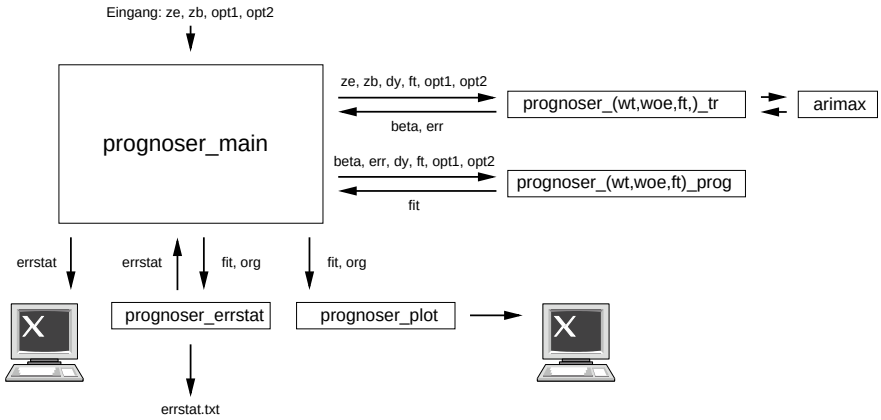

Abbildung 4.3.: Struktur des Skriptes *prognoser*.

sowohl zu einem Gesamtverlauf zusammengefügt, als auch für die einzelnen Tagtypen separat abgespeichert. Mit den Originaldaten wird ebenso verfahren. Auf diese Weise ist es möglich, in *prognoser_errstat* die statistischen Maßzahlen für den gesamten Bereich als auch nach den Tagtypen differenziert zu bestimmen. Die statistischen Maßzahlen werden zur Weiterverarbeitung als Textfile gespeichert. Das Skript *prognoser_plot* gibt den Verlauf von originalen und prognostizierten Daten grafisch am Bildschirm aus (Abbildung 4.4). Dabei wird zusätzlich eine Darstellung gewählt, bei der die Originaldaten über den geschätzten aufgetragen werden. Auf diese Weise lassen sich gut Abweichunugen in einem bestimmten Größenbereich der Daten erkennen.

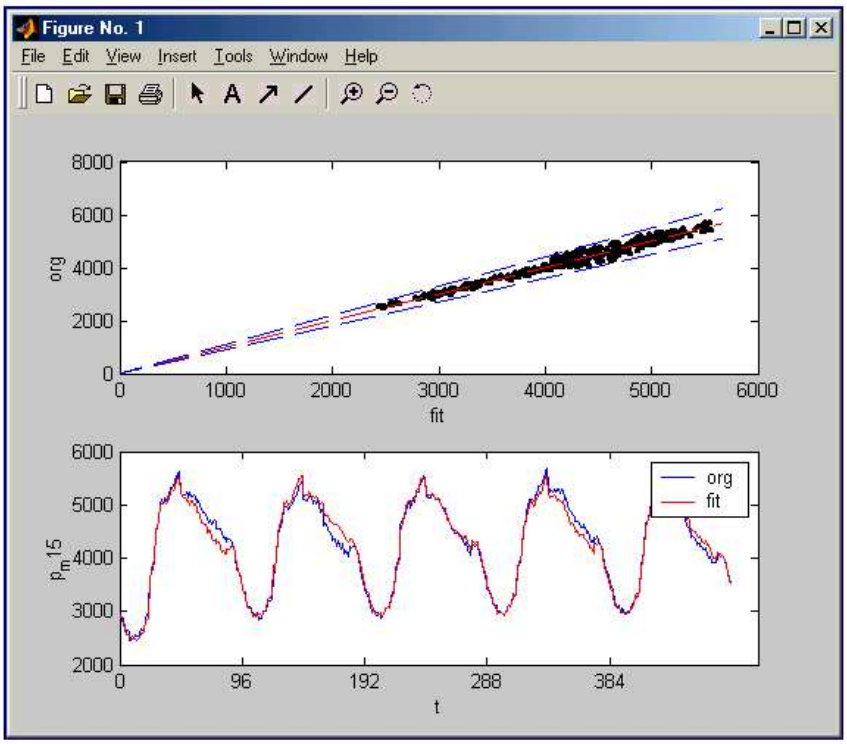

Abbildung 4.4.: Grafische Ausgabe von *prognoser_plot*.

5. Ergebnisse und Diskussion

In diesem Kapitel sollen die Prognoseergebnisse, welche anhand der verschiedenen Modelle ermittelt worden sind, miteinander verglichen und diskutiert werden. In den folgenden Diagrammen ist lediglich der *mape* als zu vergleichende statistische Maßzahl angegeben; er wird in der Praxis am häufigsten für Vergleiche verwendet, während die anderen Maßzahlen entweder keine anderen Aussagen liefern oder wie der *mbe* in den meisten Fällen als Null angenommen werden können. Eine vollständige Auflistung aller gewonnenen Ergebnisse ist in Anhang A gegeben.

Wie schon in Abschnitt 4.2 erwähnt, liegen für den $ARMAX$-Ansatz keine Ergebnisse vor. In Tabelle 5.1 sind alle für die Diskussion der Ergebnisse relevanten Modelle und deren Eigenschaften aufgelistet.

Tabelle 5.1.: Modellübersicht.

	AR1	*AR2*	*ARX*	*ARXTe*
adaptives Training	Nein	Ja	Ja	Ja
Abbildungskonstanten	Nein	Nein	Ja	Ja
Wochenend-/Feiertagsprognose	Nein	Nein	Ja	Ja
Temperatureinfluss	Nein	Nein	Nein	Ja

Folgende Punkte sollen untersucht werden:

- Das adaptive Traingsprinzip
- Die Abbildungskonstanten
- Die Berücksichtigung der Temperatur
- Die Tagtypen

Zur Untersuchung der beiden ersten Punkte werden die Prognoseergebnisse von *AR1*-, *AR2*- und ARX-Ansatz miteinander verglichen. Prognostiziert wird ein Zeitraum vom 1. Juli bis 30. April. Dabei wurden nur Werktage berücksichtigt, da die ersten beiden Modelle keine Wochenend- und Feiertage prognostizieren können. Wie in Abbildung 5.1 deutlich zu erkennen ist, ergeben sich für

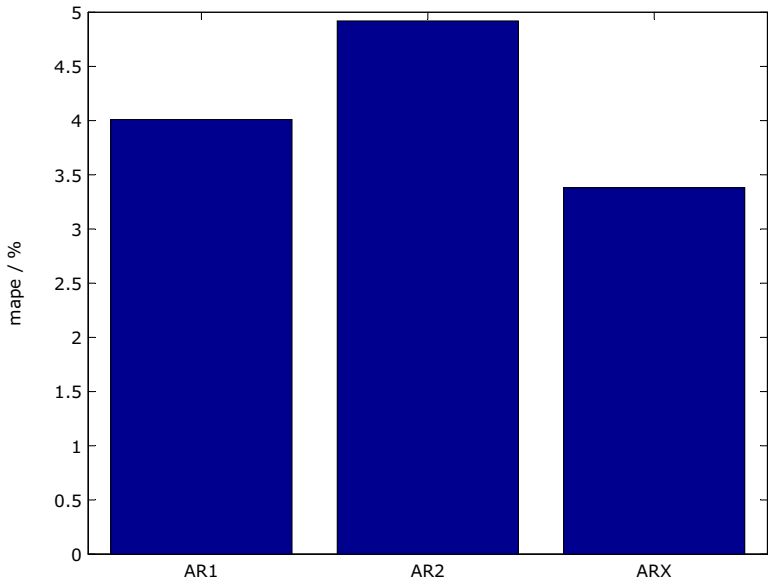

Abbildung 5.1.: Vergleich der Prognoseergebnisse der grundlegenden Modellansätze.

den *ARX*-Ansatz die besten Ergebnisse. Der direkte Vergleich der beiden adaptiven Modellansätze *AR2* und *ARX* zeigt, dass die Abbildungskonstanten des *X*-Anteils die Prognoseergebnisse wesentlich verbessern. Dass der nicht-adaptive *AR1*-Ansatz gegenüber dem adaptiven *AR2*-Ansatz deutlich bessere Resultate liefert, hat zwei Ursachen. Zum einen stehen zur Berechnung der Prognoseparameter im nicht-adaptiven Fall Daten eines Zeitraums von fast einem Jahr zur Verfügung und nicht nur, wie im adaptiven Fall, von vier Wochen, was sich wesentlich auf die Robustheit der Parameter auswirkt. Zum anderen wird im Falle des *AR1*-Ansatzes der Zeitraum, welcher zur Berechnung der Parameter verwendet wird, auch prognostiziert. Dieses Vorgehen ist jedoch unrealistisch, da in der Praxis die zu prognostizierenden Daten zum Zeitpunkt der Vorhersage noch unbekannt sind. Ob das adaptive Vorgehen im Fall der Kundenfluktuation, welche sich durch Sprünge in den Lastnachfragedaten bemerkbar macht, Vorteile bringt, kann nicht geklärt werden. Die hier verwendeten Lastdaten weisen keine solchen Sprünge auf.

Als nächstes soll der dritte Punkt untersucht werden. Abbildung 5.2 kann entnommen werden, inwiefern sich die Berücksichtigung des Temperatureinflusses auf die Prognoseergebnisse auswirkt. Dargestellt sind Ergebnisse, einmal mit (*ARXTe*-Ansatz), und einmal ohne Berücksichtigung der Temperatur (*ARX*-Ansatz). Prognostiziert wird jeweils ein Zeitraum der warmen Jahreszeit (1.Juli bis 31.August) und der kalten Jahreszeit (1.Februar bis 31.März). Für den Be-

reich der kalten Jahreszeit führt eine Berücksichtigung der Temperatur nur zu einer leichten Verbesserung in den Prognoseergebnissen. Für den der warmen Jahreszeit ist sogar eine geringfügige Verschlechterung der Resultate zu beobachten. Dies lässt sich damit begründen, dass in der warmen Jahreszeit die Kriterien für die Berücksichtigung der Temperatur, wie in Gleichung 4.9 formuliert, nur selten erfüllt sind. Infolgedessen besteht der Vektor mit den Werten der Temperatur hauptsächlich aus Nullen. Das lineare Gleichungssystem zur Berechnung der Prognoseparameter wird somit schlechter lösbar, da sich der Rang der Gleichungsmatrix verringert. MATLAB™ reagiert in diesem Fall mit der Ausgabe von Fehlerwarnungen. Vermieden werden kann dies, indem in Bereichen höherer Temperatur auf die optionale Temperaturberücksichtigung verzichtet wird. Ein modellinternes Abbruchkriterium, welches für die oben geschilderten Bedingungen automatisch das Einbeziehen der Temperatur in die Prognosegleichungen verweigert, ist für diese Arbeit nicht realisiert worden, lässt aber deutlich bessere Ergebnisse erwarten. An dieser Stelle ist darauf hinzuweisen, dass die Prognoseergebnisse für die kalte Jahreszeit im Vergleich zu denen der warmen Jahreszeit noch nicht zufriedenstellend sind, wobei davon auszugehen ist, dass in einem verfeinerten Verfahren zur Berücksichtigung der Temperatur noch weiteres Potential zur Verbesserung liegt. Denkbar ist, die Temperatur auf eine grundsätzlich andere Art in die Gleichungen einzufügen, Anregungen hierzu sind zum Beispiel in [4] gegeben. Zudem ist es, gerade in der kalten Jahreszeit, notwendig noch weitere äußere Einflussgrößen, wie zum Beispiel die Helligkeit, in den X-Anteil der Modells zu integrieren.

Zur Betrachtung des letzten Punktes, des Vergleichs der Prognoseergebnisse für die verschiedenen Tagtypen, dient Abbildung 5.3. Prognostiziert wird, der Bereich vom 1.Juni bis zum 31.Juli mit Hilfe des ARX-Ansatzes. Wie unzweifelhaft zu erkennen ist, sind die Ergebnisse für Wochenend- und Feiertage noch nicht zufriedenstellend sind. Die Ergebnisse für Werktage hingegen liegen deutlich unter 4% *mape*. Die Fehlerverteilung für die einzelnen Werktage in Abbildung 5.3 ist zufällig. Dies kann den detaillierten Ergebnisauflistungen in Anhang A entnommen werden. Grund für die schlechten Ergebnisse bei Wochenend- und Feiertagsprognose ist die adaptive Struktur des Modells. Zur Berechnung der Prognoseparameter stehen in diesem Fall nur Orginaldaten eines Zeitraums von vier Tagen zur Verfügung (vergleiche Abschnitt 4.1). Dies verringert die Robustheit der Prognoseparameter $\underline{\hat{\beta}}$. Um für die Wochenend- und Feiertagsprognose einen gleich großen Datenumfang wie bei der Prognose von Werktagen zur Berechnung der Prognoseparameter zur Verfügnung zu haben, müsste der adaptive Trainingszeitraum auf 20 Wochen ausgedehnt werden. Die Vorteile eines adaptiven Modells, sich an veränderliche Bedingungen anpassen zu können, sind in diesem Fall jedoch nicht mehr gegeben, da zu weit auf die Vergangenheit zurückgegriffen wird.

Zusätzlich zu den oben genannten Punkten soll abschließend die Prognose einiger Beispielwochen aus Kapitel 3 diskutiert werden. Abbildung 5.4 zeigt den originalen und den mit dem ARX-Ansatz geschätzten Verlauf der beiden Bei-

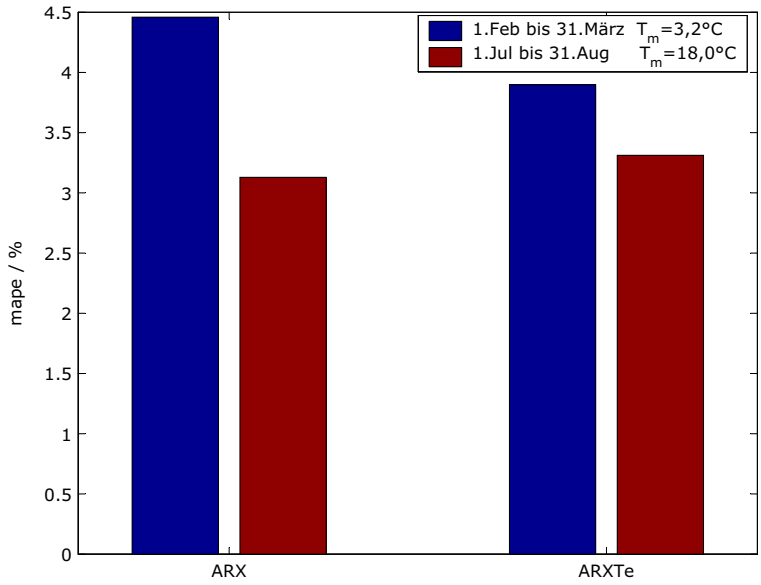

Abbildung 5.2.: Vergleich der Prognoseergebnisse einmal ohne und einmal mit Berücksichtigung der Temperatur.

spielwochen aus Abbildung 3.2. Deutlich zu sehen sind die großen Abweichungen für die Wochenendtage. Speziell für die Nächte zwischen Freitag und Samstag, beziehungsweise Samstag und Sonntag sowie für die Samstag Nachmittage wird nur eine geringe Übereinstimmung der Verläufe erreicht. Bei den Werktagen ist diese Übereinstimmung größer. Wesentliche Abweichungen sind für den Freitag und den Dienstag der zweiten Woche zu erkennen, welche in ihren Originalverläufen unerwartet niedrig sind. Zusätzlich sind Abweichnungen für die Nachmittagsverläufe und die Mittagsspitzen zu erkennen. Um die hier aufgezählten Abweichnungen zu eliminieren, ist es notwendig, die jeweiligen Ursachen, zum Beispiel für die oben erwähnten, unerwartet niedrigen Verläufe, zu ermitteln. Anschließend muss, möglichst über äußere Einflussgrößen, eine Berücksichtigung dieser Ursachen in die Prognosegleichungen integriert werden. Dies gilt insbesondere für den Bereich um Weihnachten herum. In Abbildung 5.5 ist die Prognose der beiden Wochen aus Abbildung 3.6 zu sehen. Wie schon in Kapitel 3 erwähnt, lässt sich dieser Bereich kaum prognostizieren. Wie bei der Problematik der Brückentage ist zur Erstellung spezieller Prognosemodelle für diesen Bereich die Analyse einer wesentlich umfangreicheren Datenmenge notwendig.

Abschließend ist zu sagen, dass das in dieser Arbeit vorgestellte ARX-Modell in Verbindung mit dem adaptiven Prognosevorgehen das größte Potential bietet, um im Rahmen weiterer Arbeiten ausgebaut zu werden.

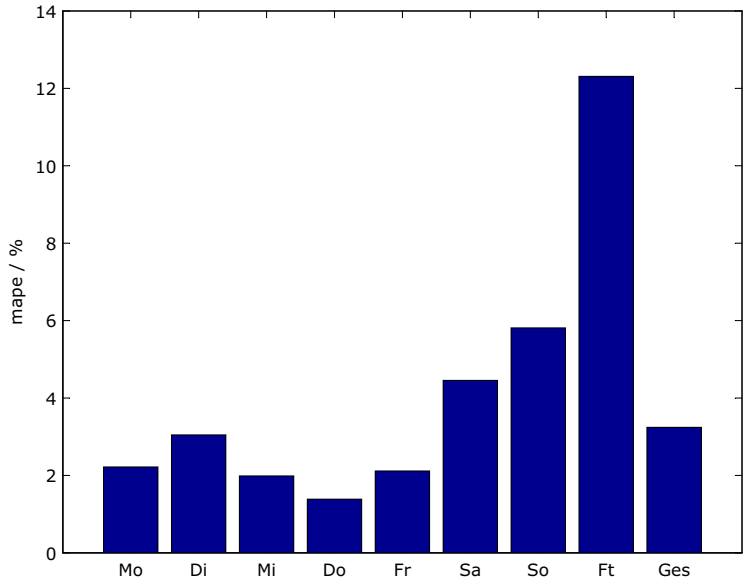

Abbildung 5.3.: Vergleich der Prognoseergebnisse für verschiedene Tagtypen, berechnet mit dem ARX-Ansatz.

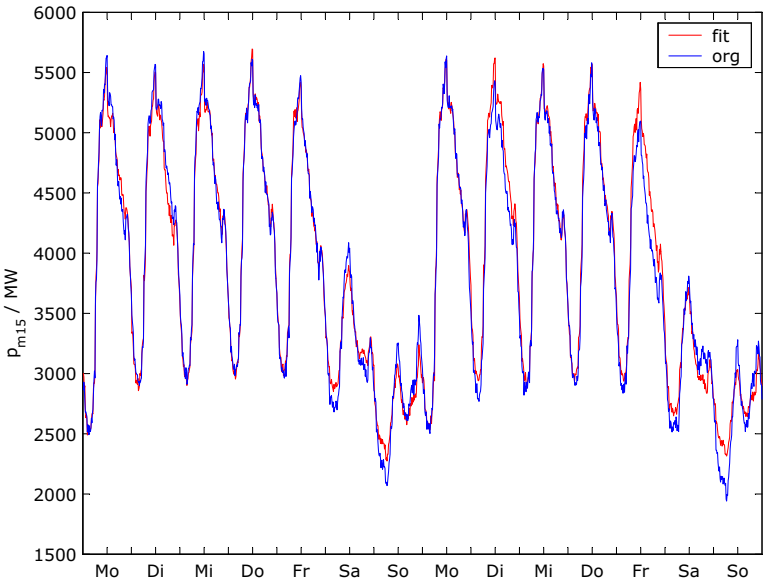

Abbildung 5.4.: Prognose der Beispielwochen aus Abbildung 3.2, berechnet mit dem ARX-Ansatz (mape: 2,82%).

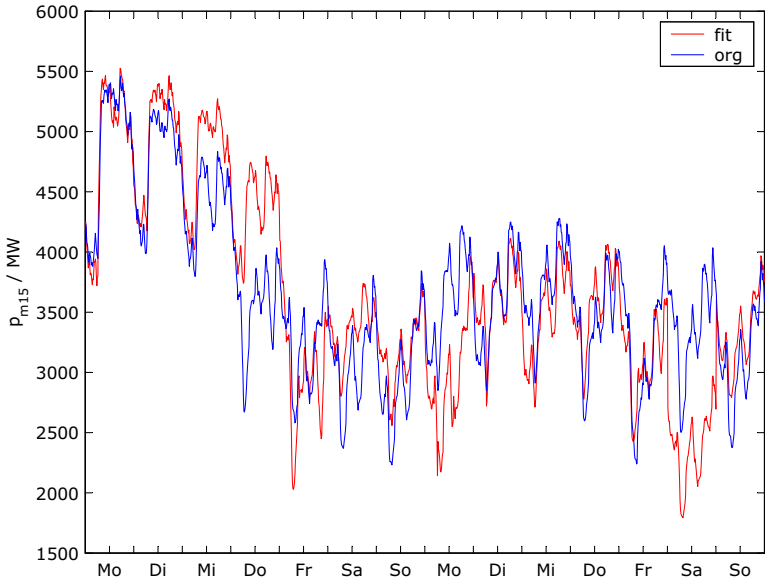

Abbildung 5.5.: Prognose der Beispielwochen aus Abbildung 3.6, berechnet mit dem *ARXTe*-Ansatz (mape: 10,36%).

6. Zusammenfassung und Ausblick

Um die elektrische Lastnachfrage seitens der Verbraucher vorhersagen zu können, werden in dieser Arbeit, auf der Grundlage des autoregressiven Ansatzes (AR-Ansatz), verschiedene Prognosemodelle abgeleitet. Um sich veränderlichen Bedingungen, zum Beispiel durch Kundenfluktuationen im liberalisierten Strommarkt verursacht, anpassen zu können, kommt in einigen der Modelle ein adaptives Prognosevorgehen zum Einsatz. Alle Modellansätze sind innerhalb der Programmierumgebung MATLAB$^{\text{TM}}$ als ausführbare Skripte realisiert.

Das Prinzip des autoregressiven Ansatzes ist es, den zukünftigen Verlauf einer gewünschten Zielgröße, hier der elektrischen Lastnachfrage, mit Hilfe des vergangenen Verlaufs dieser Größe zu bestimmen. Um festellen zu können, welche Werte der Vergangenheit wichtig für die Berechnung des zukünftigen Verlaufs sind, werden die Daten mit Hilfe von Analysefunktionen auf Autokorellation hin untersucht. Das Ergebnis dieser Datenanalyse geht ein in die Bestimmung der autoregressiven Struktur der Modelle. Dabei wird bewusst von der klassischen Vorgehensweise, alle vergangenen Werte bis hin zu einem maximalen zeitlichen Lag p_{max} in den Gleichungen zu berücksichtigen, abgesehen. Anstelle dessen werden nur einzelne, isolierte Werte zur Berechnung des zukünftigen Verlaufs verwendet. Um neben den autoregressiven noch weitere äußere Einflussgrößen berücksichtigen zu können, werden die Modelle um einen X-Anteil erweitert (ARX-Ansatz). Der X-Anteil beinhaltet sowohl die Temperatur als auch spezielle Abbildungskonstanten zur Berücksichtigung charakteristischer Tageslastverläufe. Ein integratives Verfahren (I-Anteil), welches die Daten vor der weiteren Verwendung differenziert, wird nur bei der Prognose von Wochenend- und Feiertagen verwendet. Die Erweiterung der Modelle um einen MA-Anteil, welcher Prognosefehler der Vergangenheit als weitere Eingangsgrößen in das Modell einführt, ist zwar als Modell realisiert, die nichtlinearen Gleichungen können jedoch mit dem hier verwendeten, MATLAB$^{\text{TM}}$-internen Lösungsalgorithmus nicht zufriedenstellend gelöst werden.
Die Ergebnisse machen folgendes deutlich:

- Die in dieser Arbeit eingeführten Abbildungskonstanten verbessern die Ergebnisse im Falle eines adaptiven Prognosevorgehens erheblich. Ob das adaptive Vorgehen speziell die oben angesprochene Kundenfluktuation abbilden kann, ist nicht geklärt, da sich in den hier verwendeten Daten keine

Anzeichen auf solche Fluktuationen, in Form von Sprüngen, feststellen lassen.

- Die Berücksichtigung der Temperatur führt bei der Prognose von Tagen der kalten Jahreszeit zu leicht verbesserten Prognoseergebnissen. Das Verfahren zur Berücksichtigung der Temperatur muss aber noch weiter verfeinert werden. Ebenfalls vorzunehemen ist eine Erweiterung des X-Anteils durch weitere Einflussgrößen, wie zum Beispiel die Helligkeit. Beide Punkte bieten noch viel Potential, die bisher noch nicht ausreichenden Prognoseergebnisse für die kalte Jahreszeit weiter zu verbessern.

- Werktage lassen sich mit dem hier vorgestellten ARX-Modell zufriedenstellend abbilden. Die Prognose von Wochenend- und Feiertagen muss noch verbessert werden. Grund für die schlechten Ergebnisse ist die adaptive Struktur des Modells, welche für die Berechnung des zukünftigen Verlaufs eine zu geringe Datenbasis liefert. Für Brückentage konnte kein Prognosemodell erstellt werden, da sich zu wenig Beispieltage in dem zur Verfügung stehenden Datensatz befanden.

Abschließend ist zu sagen, dass das in dieser Arbeit vorgestellte ARX-Modell, in Verbindung mit einem adaptiven Prognosevorgehen, im Vergleich zu den anderen Modellen die besten Ergebnisse liefert. Dieses Modell ist somit am besten geeignet, weiter ausgearbeitet zu werden.

A. Ergebnisse

	r^2 [%]	*mbe* [%]	*mape* [%]	*maxape* [%]	*rmse* [MW]	*maxae* [MW]
Prognoseergebnisse zu Abbildung 5.1						
AR1	91.59	$6.26 \cdot 10^{-18}$	4.01	63.23	241.61	1812.24
AR2	85.28	$-6.93 \cdot 10^{-3}$	4.92	119.41	331.70	2644.65
ARX	93.37	$2.03 \cdot 10^{-4}$	3.38	55.07	214.39	1579.70
Prognoseergebnisse zu Abbildung 5.2						
***ARX* (1.Februar bis 31.März)**						
Mo	78.52	$4.26 \cdot 10^{-2}$	5.38	20.37	314.37	849.09
Di	88.85	$2.88 \cdot 10^{-2}$	3.14	14.70	188.77	553.93
Mi	89.57	$-2.88 \cdot 10^{-1}$	2.44	13.67	177.52	638.94
Do	94.39	$-9.14 \cdot 10^{-1}$	1.72	9.73	117.63	469.36
Fr	82.43	-1.25	3.60	15.86	210.34	556.31
Sa	23.95	-3.77	7.46	31.64	354.05	1042.85
So	61.62	-4.95	7.78	27.94	313.00	755.32
Ft	0	0	0	0	0	0
Ges	89.59	-1.30	4.46	31.16	251.94	1042.85
***ARX* (1.Juli bis 31.August)**						
Mo	96.68	$5.09 \cdot 10^{-1}$	3.36	13.28	172.86	489.87
Di	97.35	$-6.43 \cdot 10^{-1}$	2.75	12.52	130.79	404.39
Mi	98.80	$4.95 \cdot 10^{-1}$	1.78	8.12	88.04	247.76
Do	98.94	$-3.48 \cdot 10^{-2}$	1.59	6.24	82.44	273.97
Fr	96.21	$-2.04 \cdot 10^{-1}$	2.64	10.70	143.33	483.57
Sa	84.24	$9.08 \cdot 10^{-2}$	4.03	13.83	145.83	446.82
So	77.81	-2.96	5.71	24.90	169.48	469.63
Ft	0	0	0	0	0	0
Ges	97.65	$-1.68 \cdot 10^{-1}$	3.13	24.90	137.46	489.90
***ARXTe* (1.Februar bis 31.März)**						
Mo	78.58	$1.48 \cdot 10^{-1}$	5.33	20.22	313.93	840.49
Di	88.90	$9.26 \cdot 10^{-2}$	3.09	14.92	188.36	558.21
Mi	89.33	$-2.25 \cdot 10^{-1}$	2.51	13.67	179.52	638.82
Do	94.65	$-8.51 \cdot 10^{-1}$	1.67	9.80	114.87	472.76
Fr	83.10	-1.78	3.56	15.40	206.27	540.21

	r^2 [%]	*mbe* [%]	*mape* [%]	*maxape* [%]	*rmse* [MW]	*maxae* [MW]
Sa	47.42	-1.24	6.12	25.06	294.39	825.98
So	81.45	$-7.05 \cdot 10^{-1}$	5.10	21.05	217.57	671.45
Ft	0	0	0	0	0	0
Ges	91.62	$-5.06 \cdot 10^{-1}$	3.90	25.06	226.03	840.50
***ARXTe* (1.Juli bis 31.August)**						
Mo	96.62	$6.11 \cdot 10^{-1}$	3.40	13.25	174.29	488.92
Di	96.95	$-2.80 \cdot 10^{-1}$	3.02	12.12	140.20	391.57
Mi	98.53	$6.82 \cdot 10^{-1}$	1.20	9.84	97.56	351.93
Do	98.24	$4.45 \cdot 10^{-1}$	2.11	11.55	106.47	334.65
Fr	96.01	$-7.44 \cdot 10^{-2}$	2.75	10.69	146.09	483.57
Sa	83.97	$9.68 \cdot 10^{-1}$	4.10	13.83	147.09	446.82
So	75.21	-1.66	5.82	25.15	179.14	556.90
Ft	0	0	0	0	0	0
Ges	97.41	$1.70 \cdot 10^{-1}$	3.31	25.15	144.46	556.90
Prognoseergebnisse zu Abbildung 5.3						
Mo	98.53	$-1.11 \cdot 10^{-1}$	2.22	10.12	119.47	375.39
Di	96.58	$-1.29 \cdot 10^{-1}$	3.04	22.02	156.66	739.56
Mi	97.96	$4.94 \cdot 10^{-1}$	1.99	16.64	117.24	520.30
Do	99.10	$2.23 \cdot 10^{-2}$	1.39	5.30	77.16	273.97
Fr	97.48	$-8.65 \cdot 10^{-1}$	2.11	10.70	120.95	449.90
Sa	81.12	-2.02	4.45	16.43	165.38	427.68
So	77.80	-2.36	5.81	26.20	178.05	502.04
Ft	-14.62	-10.41	12.31	34.81	411.13	784.80
Ges	97.44	$-7.54 \cdot 10^{-1}$	3.24	34.81	153.32	784.80
Prognoseergebnisse zu Abbildung 5.4						
Mo	99.24	$-2.85 \cdot 10^{-1}$	1.55	7.12	86.93	264.49
Di	96.93	$-9.07 \cdot 10^{-1}$	3.12	6.89	147.32	300.31
Mi	99.41	$-5.54 \cdot 10^{-2}$	1.24	6.34	64.27	178.74
Do	99.30	$-9.15 \cdot 10^{-1}$	1.28	4.24	68.35	188.39
Fr	93.57	-2.92	3.37	10.70	186.08	449.90
Sa	88.52	$-2.47 \cdot 10^{-1}$	3.57	10.80	127.55	291.65
So	80.83	$-9.60 \cdot 10^{-1}$	5.59	19.26	165.83	375.46
Ft	0	0	0	0	0	0
Ges	98.21	$-9.06 \cdot 10^{-1}$	2.82	19.26	128.98	449.90
Prognoseergebnisse zu Abbildung 5.5						
Mo	54.75	8.11	10.24	35.07	529.84	1215.63
Di	92.26	-2.29	4.12	11.57	196.18	376.19
Mi	36.82	-1.39	7.91	19.08	365.79	807.67
Do	-119.70	-12.76	14.32	55.07	638.75	1579.70
Fr	31.34	5.37	8.84	27.75	361.24	940.64

	r^2 [%]	mbe [%]	$mape$ [%]	$maxape$ [%]	$rmse$ [MW]	$maxae$ [MW]
Sa	−282.21	11.38	19.86	35.73	761.44	1381.41
So	66.11	−6.35	7.22	20.13	235.79	478.95
Ft	0	0	0	0	0	0
Ges	53.44	3.55	10.36	55.07	482.10	1579.70

B. Source Codes

B.1. checkdata

```
function checkdata(y);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Funktion berechnet die Autokorrelatonsfunktion ACF und die partielle Auto-
% korrelationsfunktion PACF für einen übergebenen Datenvektor y bis zum
% maximalen Lag von (n-1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = size(y,2);
my = mean(y);
s = zeros(n,n);
for j = 0:(n-1)
   for i = 1:(n-j)
      s (i,(j+1)) = (y(1,i) - my) .* (y(1,i+j) - my);
   end
end
acv = sum(s) ./ n; %AutoCoVariance(Function)
acf = acv ./ acv(1,1); %AutoCorrelationsFunction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Abspeichern der acf.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
save acf acf
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Einlesen der Sample AutoCoVarianceMatrix (ACV).
% [In der Literatur großes Tau]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ACV1 = zeros(n,n);
for i = 0:(n-1)
   dia(1,(i+1)) = acv(1,1);
   ACV1((i+1),(i+1):n) = acv(1,1:(n-i));
end
ACV2 = ACV1';
ACV3 = diag(dia);
ACV = ACV1 + ACV2 -ACV3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bildung der Sample AutoCorrelationsMatrix (ACF). [In der Literatur R]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ACF = ACV ./ acv(1,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Berechnung der "Yule-Walker estimators" [in der Literatur großes Phi] aus
% den Yule-Walker Gleichungen in Matrix Form. Die "Yule-Walker estimators"
% sind Schätzer für die partielle Autokorrelationsfunktion (PACF).
% [in der Literatur kleines Alpha]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
acfp(1,:) = acf(1,2:n);
acfp = acfp';
ACFp = ACF;
ACFp(:,n) = [];
ACFp(n,:) = [];
pacf = (inv(ACFp)*acfp)';
pacf = [1 pacf]; %Einfügen des Wertes für den Lag 0 (pacf(0))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Abspeichern der pacf.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
save pacf pacf
```

B.2. prognoser_pure01

```
prognoser_pure01
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Skript zur Prognose der elektrischen Tageslastnachfrage von Werktagen.
% Grundlegendes Modell ist ein AR-Ansatz. Modell ist nicht adaptiv.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Laden der Daten.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load data_wt_trai
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Einlesen der Daten.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y = reshape((data_wt_trai(6:260,3:98))',1,24480)';
x1 = reshape((data_wt_trai(5:259,3:98))',1,24480)';
x2 = reshape((data_wt_trai(1:255,3:98))',1,24480)';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Zusammenfassung des linearen und des autoregressiven Teils zus Matrix X.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
X = [ones(24480,1) x1 x2];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Regression.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
beta = X \ y;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Prognose.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x1 = reshape((data_wt_trai(20:259,3:98))',1,23040)';
x2 = reshape((data_wt_trai(16:255,3:98))',1,23040)';
X = [ones(23040,1) x1 x2];
fit = X * beta;
org = fit;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Speichern der Ergebnisse.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
save fit fit
save org org
```

B.3. prognoser_pure02

```
function prognoser_pure02(zb,ze)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Skript zur Prognose der elektrischen Tageslastnachfrage von Werktagen.
% Grundlegendes Modell ist ein AR-Ansatz.
%   Parameter:  ze: Begin Prognosebereich (Zeilennummer in data_prog.mat).
%               zb: Ende Prognosebereich (Zeilennummer in data_prog.mat).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Laden der benötigten Datenmatrizen.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load data_wt_tr;
load data_wt_prog;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Einlesen der Orginaldaten in Vektor org.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
org = data_wt_prog(zb:ze,3:98);
org = reshape(org',1,(((ze-zb)+1)*96));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Start der Prognose Schleife.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fit = [];
for i = zb : ze
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Berechnung des Koeffizientenvektors beta durch lineare Regression
    % in einem Datenzeitraum von vier Wochen.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    y = (reshape((data_wt_tr((i-15):(i-1),3:98))',1,1440))';
    x1 = (reshape((data_wt_tr((i-16):(i-2),3:98))',1,1440))';
    x2 = (reshape((data_wt_tr((i-20):(i-6),3:98))',1,1440))';
    X = [ones(1440,1) x1 x2];
    beta = X \ y;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Prognose eines Tages.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    x1 = (data_wt_tr((i-1),3:98))';
    x2 = (data_wt_tr((i-5),3:98))';
    X = [ones(96,1) x1 x2];
    fit1 = (X * beta)';
    fit = [fit fit1];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Speichern von fit und org.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
save fit fit
save org org
```

B.4. prognoser

```
function prognoser_main(zb,ze,opt1,opt2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Skript zur Prognose der elektrischen Tageslastnachfrage. Grundlegendes
% Modell ist ein ARX-Ansatz. Erweiterung über opt-Parameter.
%   Parameter:  ze: Begin Prognosebereich (Zeilennummer in data_prog.mat).
%               zb: Ende Prognosebereich (Zeilennummer in data_prog.mat).
%               opt1: Berücksichtigung eines MA-Anteiles (1=nein ; 2=ja).
%               opt2: Berücksichtigung der Temperatur (1=nein ; 2=ja).
% Die Ergebnisausgabe erfolgt am Bildschirm. Ausgegeben werden statistische
% Masszahlen und ein Plot der Daten. Alle masszahlen werden in Textfile
% errstat.txt gespeichert, die Prognose- und die Originldaten jeweils in
% deb Datenvektoren fit.mat und org.mat.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Laden der orginal Daten aus data_prog.mat in Vektor org.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load data_prog.mat;
org = data_prog(zb:ze,3:98);
org = reshape(org',1,((ze-zb)+1)*96);
orgMo = [];
orgDi = [];
orgMi = [];
orgDo = [];
orgFr = [];
orgSa = [];
orgSo = [];
orgFt = [];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Start der Prognoseschleife.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fit = [];
fitMo = [];
fitDi = [];
fitMi = [];
fitDo = [];
fitFr = [];
fitSa = [];
fitSo = [];
fitFt = [];
for i = zb : ze
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Ermittlung der Zeilennummer in data_wt_prog.mat (bzw. _tr.mat) und
    % err_wt.mat für den ARMAX Ansatz.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    z = (floor(i/7) * 5) + (i - (floor(i/7) * 7));
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Test ob im Trainingszeitraum ein Feiertag/Sondertag vorliegt, und
    % Beseitigung dieses Feiertages in den Trainingsmatrizen
    % data_wt_tr.mat und data_tr durch eine fiktive Wochentagsprognose.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    load data_tr;
    load data_wt_tr;
    fixtest = sum(data_tr((i-28):(i-1),2));
    if fixtest > 0
        k = find(data_tr((i-28):(i-1),2));
        isave = i;
        for j = 1 : (size(k,1))
            i = isave - (28 - (k(j,1) - 1));
            dy = data_tr(i,1);
            [beta] = prognoser_wt_trai(dy,i,z,opt1,opt2);
            [fit1] = prognoser_wt_prog(beta,dy,i,z,opt1,opt2);
            data_tr(i,3:98) = fit1;
            data_tr(i,2) = 0;
            data_wt_tr(z,3:98) = fit1;
            data_wt_tr(z,2) = 0;
            save data_tr data_tr;
            save data_wt_tr data_wt_tr;
        end
        i = isave;
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Überprüfung Wochentag <=> Wochenende, Feiertag <=> kein Feiertag
    % anhand der Werte dy und ft in Datenmatrix data_prog.mat.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    dy = data_prog(i,1);
    ft = data_prog(i,2);
    switch (ft+dy)
```

```
 78:     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 79:     % Wochentag, kein Feiertag
 80:     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 81:     case {0,1,2,3,4}
 82:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 83:         % Ermittlung des Koeffizentenvektors beta für die Prognose von
 84:         % Wochentagen.
 85:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 86:         [beta,err] = prognoser_wt_trai(dy,i,z,opt1,opt2);
 87:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 88:         % Start der Prognose für einen Wochentag.
 89:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 90:         [fit1] = prognoser_wt_prog(beta,err,dy,i,z,opt1,opt2);
 91:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 92:         % Einlesen der Daten in fit* und org* in Abhängigkeit vom Tagwert.
 93:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 94:         switch dy
 95:         case 0
 96:             orgMo = [orgMo data_prog(i,3:98)];
 97:             fitMo = [fitMo fit1];
 98:         case 1
 99:             orgDi = [orgDi data_prog(i,3:98)];
100:             fitDi = [fitDi fit1];
101:         case 2
102:             orgMi = [orgMi data_prog(i,3:98)];
103:             fitMi = [fitMi fit1];
104:         case 3
105:             orgDo = [orgDo data_prog(i,3:98)];
106:             fitDo = [fitDo fit1];
107:         case 4
108:             orgFr = [orgFr data_prog(i,3:98)];
109:             fitFr = [fitFr fit1];
110:         end
111:     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
112:     % Wochenende
113:     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
114:     case {5,6}
115:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
116:         % Ermittlung des Koeffizentenvektors beta für die Prognose von
117:         % Wochenenden.
118:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
119:         [beta,err] = prognoser_woe_trai(i,opt1,opt2);
120:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
121:         % Start der Prognose für Samstag bzw. Sonntag.
122:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
123:         [fit1] = prognoser_woe_prog(beta,err,i,opt1,opt2);
124:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
125:         % Einlesen der Daten in fit* und org* in Abhängigkeit vom Tagwert.
126:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
127:         switch dy
128:         case 5
129:             orgSa = [orgSa data_prog(i,3:98)];
130:             fitSa = [fitSa fit1];
131:         case 6
132:             orgSo = [orgSo data_prog(i,3:98)];
133:             fitSo = [fitSo fit1];
134:         end
135:     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
136:     % Feiertag
137:     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
138:     case {7,8,9,10,11}
139:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
140:         % Ermittlung des Koeffizentenvektors beta für die Prognose eines
141:         % Feiertages.
142:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
143:         [beta,err] = prognoser_ft_trai(dy,i,opt1,opt2);
144:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
145:         % Start der Prognose für einen Feiertag.
146:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
147:         [fit1] = prognoser_ft_prog(beta,err,dy,i,opt1,opt2);
148:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
149:         % Einlesen der Daten in fitFt und orgFt.
150:         %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
151:         orgFt = [orgFt data_prog(i,3:98)];
152:         fitFt = [fitFt fit1];
153:     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
154:     % Brückentag/Sondertag
```

```
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    case {12,13,14,15,16}
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Da zu wenige Brückentage in den Daten vorhanden waren, wurde deren
        % Prognose ausgenommen.
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        fit1 = data_prog(i,3:98);
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Schreiben der Prognosedaten in den Vektor fit.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    fit = [fit fit1];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Übergabe von fit und org an die Plotausgabe.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
save fit fit;
save org org;
prognoser_plot(org,fit);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Berechnung und Ausgabe der differenzierten Fehlerstatistik.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[errstat] = prognoser_errstat(org,fit);
errstatGes = errstat;
disp ('________________________________________________')
disp ('________________________________________________')
if isempty(orgMo) == 0
    [errstat] = prognoser_errstat(orgMo,fitMo);
    errstatMo = errstat;
    disp ('Fehlerstatistik für Montage:')
    disp ('________________________________________________')
    disp ('    r2        mbe       mape       maxape      ')
    disp (errstatMo(1,1:4))
    disp ('     rmse        maxae')
    disp (errstatMo(1,5:6))
    disp ('________________________________________________')
else
    errstatMo = zeros(1,6);
end
if isempty(orgDi) == 0
    [errstat] = prognoser_errstat(orgDi,fitDi);
    errstatDi = errstat;
    disp ('Fehlerstatistik für Dienstage:')
    disp ('________________________________________________')
    disp ('    r2        mbe       mape       maxape      ')
    disp (errstatDi(1,1:4))
    disp ('     rmse        maxae')
    disp (errstatDi(1,5:6))
    disp ('________________________________________________')
else
    errstatDi = zeros(1,6);
end
if isempty(orgMi) == 0
    [errstat] = prognoser_errstat(orgMi,fitMi);
    errstatMi = errstat;
    disp ('Fehlerstatistik für Mittwoche:')
    disp ('________________________________________________')
    disp ('    r2        mbe       mape       maxape      ')
    disp (errstatMi(1,1:4))
    disp ('     rmse        maxae')
    disp (errstatMi(1,5:6))
    disp ('________________________________________________')
else
    errstatMi = zeros(1,6);
end
if isempty(orgDo) == 0
    [errstat] = prognoser_errstat(orgDo,fitDo);
    errstatDo = errstat;
    disp ('Fehlerstatistik für Donnerstage:')
    disp ('________________________________________________')
    disp ('    r2        mbe       mape       maxape      ')
    disp (errstatDo(1,1:4))
    disp ('     rmse        maxae')
    disp (errstatDo(1,5:6))
    disp ('________________________________________________')
else
    errstatDo = zeros(1,6);
```

```
end
if isempty(orgFr) == 0
    [errstat] = prognoser_errstat(orgFr,fitFr);
    errstatFr = errstat;
    disp ('Fehlerstatistik für Freitage:')
    disp ('_______________________________________________')
    disp ('    r2        mbe       mape       maxape      ')
    disp (errstatFr(1,1:4))
    disp ('     rmse        maxae')
    disp (errstatFr(1,5:6))
    disp ('_______________________________________________')
else
    errstatFr = zeros(1,6);
end
if isempty(orgSa) == 0
    [errstat] = prognoser_errstat(orgSa,fitSa);
    errstatSa = errstat;
    disp ('Fehlerstatistik für Samstage:')
    disp ('_______________________________________________')
    disp ('    r2        mbe       mape       maxape      ')
    disp (errstatSa(1,1:4))
    disp ('     rmse        maxae')
    disp (errstatSa(1,5:6))
    disp ('_______________________________________________')
else
    errstatSa = zeros(1,6);
end
if isempty(orgSo) == 0
    [errstat] = prognoser_errstat(orgSo,fitSo);
    errstatSo = errstat;
    disp ('Fehlerstatistik für Sonntage:')
    disp ('_______________________________________________')
    disp ('    r2        mbe       mape       maxape      ')
    disp (errstatSo(1,1:4))
    disp ('     rmse        maxae')
    disp (errstatSo(1,5:6))
    disp ('_______________________________________________')
else
    errstatSo = zeros(1,6);
end
if isempty(orgFt) == 0
    [errstat] = prognoser_errstat(orgFt,fitFt);
    errstatFt = errstat;
    disp ('Fehlerstatistik für Feiertage:')
    disp ('_______________________________________________')
    disp ('    r2        mbe       mape       maxape      ')
    disp (errstatFt(1,1:4))
    disp ('     rmse        maxae')
    disp (errstatFt(1,5:6))
    disp ('_______________________________________________')
else
    errstatFt = zeros(1,6);
end
disp ('Gesamte Fehlerstatistik')
disp ('_______________________________________________')
disp ('    r2        mbe       mape       maxape      ')
disp (errstatGes(1,1:4))
disp ('     rmse        maxae')
disp (errstatGes(1,5:6))
disp ('_______________________________________________')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Schreiben der errstat Daten in ein Textfile.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
errstat = [errstatMo' errstatDi' errstatMi' errstatDo' errstatFr' errstatSa' errstatSo' errstatFt' errstatGes']';
dlmwrite('errstat.txt',errstat,'\t');
```

```
function [beta,err] = prognoser_wt_trai(dy,i,z,opt1,opt2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Skript zum Antrainieren der Prognoseparameter beta für die Prognose eines
% Wochentages.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Einlesen der Datenmatrizen.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load data_wt_tr.mat
load d_wt.mat
load data_tr.mat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Einlesen der Lastwerte.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y = (reshape(data_wt_tr((z-15):(z-1),3:98)',1,1440))';
x1 = (reshape(data_wt_tr((z-16):(z-2),3:98)',1,1440))';
x2 = (reshape(data_wt_tr((z-20):(z-6),3:98)',1,1440))';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Einlesen der Durchschnittswerte für Sonntage (exogene Grösse).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xmeanSo = zeros(1440,1);
if dy == 0
    xmeanSo(1:96,1) = mean(data_tr((i-22),3:98));
    xmeanSo(481:576,1) = mean(data_tr((i-15),3:98));
    xmeanSo(961:1056,1) = mean(data_tr((i-8),3:98));
else
    xmeanSo((((5-dy)*96)+1):((6-dy)*96),1) = mean(data_tr((i-(dy+15)),3:98));
    xmeanSo((((10-dy)*96)+1):((11-dy)*96),1) = mean(data_tr((i-(dy+8)),3:98));
    xmeanSo((((15-dy)*96)+1):((16-dy)*96),1) = mean(data_tr((i-(dy+1)),3:89));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Einlesen der Abbildungsdummys (exogene Grössen).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
d_wt_tr = [d_wt(((dy*96)+1):1440,:)' d_wt(1:(dy*96),:)']';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Zusammenfügen der X Matrix mit linearem Teil, autoregresivem Teil und den exogenen
% Grössen.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
X = [ones(1440,1) x1 x2 xmeanSo d_wt_tr];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fallunterscheidung ob die Temperatur als weitere exogene Grösse berücksichtigt
% werden soll (Variable opt2). Die Temperatur wird nur berücksichtigt, wenn die
% mittlere Tagestemperautr unter 15°C lag. Als exogene Grösse geht die Differenz der
% Temperatur zu 15°C in den Nacht- und morgenstunde ein.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if opt2 == 2
    load temp_wt.mat
    xtemp = [];
    for j = (z-15):(z-1)
        if temp_wt(j,97) < 15
            xtemp1 = (temp_wt(j,1:96) - 15);
            xtemp1(1,21:87) = zeros;
        else
            xtemp1 = zeros(1,96);
        end
        xtemp = [xtemp xtemp1];
    end
    xtemp = xtemp';
    X = [X xtemp];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Definition der Variablen err welche bei der Prognose mit MA Ansatz benötigt wird.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
err = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fallunterscheidung ob beta mittels eines ARX oder eines ARMAX Ansatzes ermittelt
% werden soll (Variable opt1).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
switch opt1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ARX Ansatz
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 1
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Lösung des Gleichungssystemes y = X * beta mit der nichtquadratischen Matrix X
    % für die lineare Regression.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    beta = X \ y;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ARMAX Ansatz
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 2
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Lösung des Ansatzes mit Hilfe des ARIMAX Skriptes von Jan Derk Swider.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Neue Datenzuweisung.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    x = X;
    p = []; % AR-Anteil in x enthalten!
    d = 0;
    q = [96 480];
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Definition der Startparameter (c = 1).
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    parameter_0 = 1*ones((size(p,2))+(size(q,2))+(size(x,2))+1,1);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Aufruf der Funktion ARIMAX.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    [parameter,err] = arimax(p,d,q,parameter_0,y,x);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Umformung parameter -> beta.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    beta = [parameter(4:6,1)' parameter(2:3,1)' parameter(7:(size(parameter,1)),1)']';
    parameter
end
```

```
function [fit1] = prognoser_wt_prog(beta,err,dy,i,z,opt1,opt2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Skript zur Prognose eines Wochentages.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Laden der Datenmatrizen.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load data_wt_prog.mat
load data_wt_tr.mat
load data_prog
load d_wt.mat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Einlesen der Variablen mit Untersuchung ob am Tag zuvor ober zwei Tage zuvor ein
% Feiertag vorgelegen hat. Unterscheidung ob ARX ober ARMAX Modell vorliegt mit hilfe
% der Variablen opt1.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if dy == 0 % Fall Montag
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if data_wt_prog((z-1),2) == 7 % Freitag war Feiertag
        o = zeros(96,1);
        d_wt_prog = [o o d_wt(1:96,3) o];
        xmeanSo = zeros(96,1);
        xmeanSo(:,1) = mean(data_prog((i-1),3:98)); % Mittelwert Sonntag
        x1 = data_wt_tr((z-2),3:98)';
        x2 = data_wt_tr((z-5),3:98)';
        X = [ones(96,1) x1 x2 xmeanSo d_wt_prog];
        if opt1 == 2
            xerr1 = err(1345:1440,1);
            xerr2 = err(1249:1344,1);
            X = [X(:,1:3) xerr1 xerr2 X(:,4:8)];
        end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    elseif data_wt_prog((z-2),2) == 7 % Donnerstag war Feiertag
        o = zeros(96,1);
        d_wt_prog = [o o d_wt(1:96,3) o];
        xmeanSo = zeros(96,1);
        xmeanSo(:,1) = mean(data_prog((i-1),3:98)); % Mittelwert Sonntag
        x1 = data_wt_tr((z-3),3:98)';
        x2 = data_wt_tr((z-5),3:98)';
        X = [ones(96,1) x1 x2 xmeanSo d_wt_prog];
        if opt1 == 2
            xerr1 = err(1153:1248,1);
            xerr2 = err(961:1056,1);
            X = [X(:,1:3) xerr1 xerr2 X(:,4:8)];
        end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    else % Keine Feiertagsproblematik
        d_wt_prog = d_wt(1:96,:);
        xmeanSo = zeros(96,1);
        xmeanSo(:,1) = mean(data_prog((i-1),3:98)); % Mittelwert Sonntag
        x1 = data_wt_tr((z-1),3:98)';
        x2 = data_wt_tr((z-5),3:98)';
        X = [ones(96,1) x1 x2 xmeanSo d_wt_prog];
        if opt1 == 2
            xerr1 = err(1345:1440,1);
            xerr2 = err(961:1056,1);
            X = [X(:,1:3) xerr1 xerr2 X(:,4:8)];
        end
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif dy == 1 % Fall Dienstag
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if data_wt_prog((z-1),2) == 7 % Montag war Feiertag
        d_wt_prog = d_wt(1:96,:);
        xmeanSo = zeros(96,1);
        xmeanSo(:,1) = mean(data_prog((i-1),3:98)); % Mittelwert des Feiertages Montag
        x1 = data_wt_tr((z-2),3:98)';
        x2 = data_wt_tr((z-6),3:98)';
        X = [ones(96,1) x1 x2 xmeanSo d_wt_prog];
        if opt1 == 2
            xerr1 = err(1249:1344,1);
            xerr2 = err(865:960,1);
            X = [X(:,1:3) xerr1 xerr2 X(:,4:8)];
        end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    else % Keine Feiertagsproblematik
        d_wt_prog = d_wt(97:192,:);
        xmeanSo = zeros(96,1);
        x1 = data_wt_tr((z-1),3:98)';
        x2 = data_wt_tr((z-5),3:98)';
        X = [ones(96,1) x1 x2 xmeanSo d_wt_prog];
        if opt1 == 2
            xerr1 = err(1345:1440,1);
            xerr2 = err(961:1056,1);
            X = [X(:,1:3) xerr1 xerr2 X(:,4:8)];
```

```
        end
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif dy == 2 % Fall Mittwoch
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if data_wt_prog((z-1),2) == 7 % Dienstag war Feiertag (Spekulation)
        d_wt_prog = zeros(96,4);
        xmeanSo = zeros(96,1);
        xmeanSo(:,1) = mean(data_prog(i-1,3:98)); % Mittelwert des Feiertages Dienstag
        x1 = data_wt_tr((z-2),3:98)';
        x2 = zeros(96,1);
        X = [ones(96,1) x1 x2 xmeanSo d_wt_prog];
        if opt1 == 2
            xerr1 = err(1249:1344,1);
            xerr2 = zeros;
            X = [X(:,1:3) xerr1 xerr2 X(:,4:8)];
        end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    elseif data_wt_prog((z-2),2) == 7 % Montag war Feiertag
        d_wt_prog = d_wt(97:192,:);
        xmeanSo = zeros(96,1);
        x1 = data_wt_tr((z-1),3:98)';
        x2 = data_wt_tr((z-5),3:98)';
        X = [ones(96,1) x1 x2 xmeanSo d_wt_prog];
        if opt1 == 2
            xerr1 = err(1345:1440,1);
            xerr2 = err(961:1056,1);
            X = [X(:,1:3) xerr1 xerr2 X(:,4:8)];
        end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    else % Keine Feiertagsproblematik
        d_wt_prog = zeros(96,4);
        xmeanSo = zeros(96,1);
        x1 = data_wt_tr((z-1),3:98)';
        x2 = data_wt_tr((z-5),3:98)';
        X = [ones(96,1) x1 x2 xmeanSo d_wt_prog];
        if opt1 == 2
            xerr1 = err(1345:1440,1);
            xerr2 = err(961:1056,1);
            X = [X(:,1:3) xerr1 xerr2 X(:,4:8)];
        end
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif dy == 3 % Fall Donnerstag
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if data_wt_prog((z-1),2) == 7 % Mittwoch war Feiertag
        o = zeros(96,1);
        d_wt_prog = [o o d_wt(1:96,3) o];
        xmeanSo = zeros(96,1);
        xmeanSo(:,1) = mean(data_prog(i-1,3:98)); % Mittelwert des Feiertages Mittwoch
        x1 = data_wt_tr((z-2),3:98)';
        x2 = zeros(96,1);
        X = [ones(96,1) x1 x2 xmeanSo d_wt_prog];
        if opt1 == 2
            xerr1 = err(1249:1344,1);
            xerr2 = zeros(96,1);
            X = [X(:,1:3) xerr1 xerr2 X(:,4:8)];
        end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    elseif data_wt_prog((z-2),2) == 7 % Dienstag war Feiertag (Spekulation)
        d_wt_prog = d_wt(97:192,:);
        xmeanSo = zeros(96,1);
        x1 = data_wt_tr((z-1),3:98)';
        x2 = data_wt_tr((z-5),3:98)';
        X = [ones(96,1) x1 x2 xmeanSo d_wt_prog];
        if opt1 == 2
            xerr1 = err(1345:1440,1);
            xerr2 = err(961:1056,1);
            X = [X(:,1:3) xerr1 xerr2 X(:,4:8)];
        end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    else % Keine Feiertagsproblematik
        d_wt_prog = zeros(96,4);
        xmeanSo = zeros(96,1);
        x1 = data_wt_tr((z-1),3:98)';
        x2 = data_wt_tr((z-5),3:98)';
        X = [ones(96,1) x1 x2 xmeanSo d_wt_prog];
        if opt1 == 2
            xerr1 = err(1345:1440,1);
            xerr2 = err(961:1056,1);
            X = [X(:,1:3) xerr1 xerr2 X(:,4:8)];
        end
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif dy == 4 % Fall Freitag
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if data_wt_prog((z-2),2) == 7 % Mittwoch war Feiertag
        o = zeros(96,1);
        d_wt_prog = [d_wt(385:480,1) d_wt(97:192,2) o o];
        xmeanSo = zeros(96,1);
        x1 = data_wt_tr((z-1),3:98)';
        x2 = data_wt_tr((z-5),3:98)';
        X = [ones(96,1) x1 x2 xmeanSo d_wt_prog];
        if opt1 == 2
            xerr1 = err(1345:1440,1);
            xerr2 = err(961:1056,1);
            X = [X(:,1:3) xerr1 xerr2 X(:,4:8)];
        end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    else % Keine Feiertagsproblematik
        d_wt_prog = d_wt(385:480,:);
        xmeanSo = zeros(96,1);
        x1 = data_wt_tr((z-1),3:98)';
        x2 = data_wt_tr((z-5),3:98)';
        X = [ones(96,1) x1 x2 xmeanSo d_wt_prog];
        if opt1 == 2
            xerr1 = err(1345:1440,1);
            xerr2 = err(961:1056,1);
            X = [X(:,1:3) xerr1 xerr2 X(:,4:8)];
        end
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fallunterscheidung ob die Temperatur als weitere exogene Grösse berücksichtigt
% werden soll (Variable opt2).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if opt2 == 2
    load temp_wt.mat
    if temp_wt(z,97) < 15
        xtemp = (temp_wt(z,1:96))';
        xtemp(21:87,1) = zeros;
    else
        xtemp = zeros(96,1);
    end
    X = [X xtemp];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Prognose
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fit1 = (X * beta)';
```

```
function [beta,err] = prognoser_woe_trai(i,opt1,opt2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Skript zum Antrainieren der Prognoseparameter beta für die Prognose eines
% Wochenendtages.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Laden der Datenmatrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load data_tr.mat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Einlesen der Lastwerte.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y = (reshape((data_tr((i-21):7:(i-7),3:98))',1,288))';
x1 = (reshape((data_tr((i-28):7:(i-14),3:98))',1,288))';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fallunterscheidung ob die Temperatur als weitere exogene Grösse berücksichtigt
% werden soll (Variable opt2).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if opt2 == 2
    load temp.mat
    xtemp = [];
    for j = (i-21):7:(i-7)
        if temp(j,97) < 15
            xtemp1 = (temp(j,1:96) - 15);
            xtemp1(1,21:87) = zeros;
        else
            xtemp1 = zeros(1,96);
        end
        xtemp = [xtemp xtemp1];
    end
    xtemp = xtemp';
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Definition der Variablen err welche bei der Prognose mit MA Ansatz benötigt wird.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
err = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fallunterscheidung ob ein ARIX oder ein ARIMAX Ansatz verwendet werden soll
% (Variable opt1).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
switch opt1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fall ARIX.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 1
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Neuzuweisung der Daten.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    y = [data_tr(i-22,98) y'];
    y = (diff(y))';
    x1 = [data_tr(i-29,98) x1'];
    x1 = (diff(x1))';
    X = [x1];
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Anwendung eines modifizierten Temperaturverfahrens (Variable opt2).
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if opt2 == 2
        if isequal(xtemp(1,1),0) == 0
            xtemp = [(temp((i-22),96) - 15) xtemp(1:287,1)']';
        else
            xtemp = [0 xtemp(1:287,1)']';
        end
    X = [X xtemp];
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Lösung des Gleichungssystemes.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    beta = X \ y;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fall ARIMAX. Allgemein Lösung des Ansatzes mit dem ARIMAX Skript
% von Derk Jan Swider.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 2
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Neuzuweisung der Daten.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    y = [data_tr((i-22),98) y'];
    y = (diff(y))';
    x = [data_tr((i-29),98) x1'];
    x = (diff(x))';
    p = []; % AR-Anteil in x!
    d = 0; % diff schon erledigt!
    q = [96];
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Anwendung eines modifizierten Temperaturverfahrens (Variable opt2).
```

```
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if opt2 == 2
        if isequal(xtemp(1,1),0) == 0
            xtemp = [(temp((i-22),96) - 15) xtemp(1:287,1)'];
        else
            xtemp = [0 xtemp(1:287,1)'];
        end
        x = [x xtemp'];
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Definition der Startparameter (c = 1).
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    parameter_0 = 1*ones((size(p,2))+(size(q,2))+(size(x,2))+1,1);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Aufruf der Funktion ARIMAX.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    [parameter,err] = arimax(p,d,q,parameter_0,y,x);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Datenzuweisung.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    beta = [parameter(3,1) parameter(2,1)]';
    if opt2 == 2
        beta = [beta' parameter(4,1)]';
    end
end
```

```
function [fit1] = prognoser_woe_prog(beta,err,i,opt1,opt2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Skript zur Prognose eines Wochenendtages.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Laden der Datenmatrizen
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load data_prog.mat
load data_tr.mat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Differenzenverfahren.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x1 = [data_tr((i-8),98) data_prog((i-7),3:98)];
x1 = (diff(x1))';
X = [x1];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Überprüfung ob ein ARIMAX Modell verwendet wird (Variable opt1).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if opt1 == 2
    xerr1 = err(193:288,1);
    X = [x1 xerr1];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Überprüfung ob die Temperatur berücksichtigt wird
% (modifiziertes Verfahren/Variable opt2).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if opt2 == 2
    load temp.mat
    if temp(i,97) < 15
        xtemp = [(temp((i-1),96) - 15) (temp(i,1:95) - 15)]';
        xtemp(22:88,1) = zeros;
    else
        xtemp = zeros(96,1);
    end
    X = [X xtemp];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Prognose.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
delta = (X * beta)';
fit1 = zeros(1,96);
a = data_tr((i-1),98);
fit1(1,1) = a + delta(1,1);
for j = 2:96
    fit1(1,j) = fit1(1,(j-1)) + delta(1,j);
end
```

```
function [beta,err] = prognoser_ft_trai(dy,i,opt1,opt2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Skript zum Antrainieren der Prognoseparameter beta für die Prognose eines
% Feiertages.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Einlesen der benötigten Datenfiles.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load data_tr.mat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Einlesen und Differenzierung der Lastwerte.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y = reshape(data_tr((i-(dy+15)):7:(i-(dy+1)),3:98)',1,288);
y = [data_tr((i-(dy+16)),98) y]';
y = diff(y);
x1 = reshape(data_tr((i-(dy+22)):7:(i-(dy+8)),3:98)',1,288);
x1 = [data_tr((i-(dy+23)),98) x1]';
x1 = diff(x1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Zusammenfassung der Matrix X mit autoregressivem Teil.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
X = x1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Überprüfung ob die Temperatur als weitere exogene Grösse berücksichtigt werden
% soll (modifiziertes Verfahren / Variable opt2).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if opt2 == 2
    load temp.mat
    xtemp = [];
    for j = (i-(dy+15)):7:(i-(dy+1))
        if temp(j,97) < 15
            xtemp1 = (temp(j,1:96) - 15);
            xtemp1(1,21:87) = zeros;
        else
            xtemp1 = zeros(1,96);
        end
        xtemp = [xtemp xtemp1];
    end
    if isequal(xtemp(1,1),0) == 0
        xtemp = [(temp((i-(dy+16)),96) - 15) xtemp(1,1:287)]';
    else
        xtemp = [0 xtemp(1,1:287)]';
    end
    X = [X xtemp];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Definition der Variablen err welche bei der Prognose mit MA Ansatz benötigt wird.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
err = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fallunterscheidung ob ein ARIX oder ein ARIMAX Ansatz verwendet werden soll
% (Variable opt1).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
switch opt1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fall ARIX.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 1
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Lösung des linearen Gleichungssystemes y = X * beta mit der
    % nichtquadratischen Matrix X.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    beta = X \ y;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Fall ARIMAX. Allgemein Lösung des Ansatzes mit dem ARIMAX Skript von
% Derk Jan Swider.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 2
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Neuzuweisung der Daten.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    x = X;
    p = []; % AR-Anteil in x!
    d = 0; % diff schon erledigt!
    q = [96];
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Definition der Startparameter (c = 1).
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    parameter_0 = 1*ones((size(p,2))+(size(q,2))+(size(x,2))+1,1);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Aufruf der Funktion ARIMAX.
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    [parameter,err] = arimax(p,d,q,parameter_0,y,x);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Datenzuweisung.
```

```
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    beta = [parameter(3,1) parameter(2,1)]'
    if opt2 == 2
        beta = [beta' parameter(4,1)]';
    end
end
```

```
function [fit1] = prognoser_ft_prog(beta,err,dy,i,opt1,opt2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Skript zur Prognose eines Feiertages.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Laden der Datenmatrizen
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load data_prog.mat
load data_tr.mat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Einlesen der Variablen in Matrix X.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x1 = [data_prog((i-(dy+2)),98) data_prog((i-(dy+1)),3:98)];
x1 = diff(x1)';
X = x1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Überprüfung ob ein ARIMAX Ansatz verwendet wird (Variable opt1).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if opt1 == 2
    xerr1 = err(193:288,1);
    X = [X xerr1];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Überprüfung ob die Temperatur berücksichtigt wird
% (modifiziertes Verfahren/Variable opt2).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if opt2 == 2
    load temp.mat
    if temp(i,97) < 15
        xtemp = [(temp((i-1),96) - 15) (temp(i,1:95) - 15)]';
        xtemp(22:88,1) = zeros;
    else
        xtemp = zeros(96,1);
    end
    X = [X xtemp];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Prognose.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
delta = (X * beta)';
fit1 = zeros(1,96);
a = data_tr((i-1),98);
fit1(1,1) = a + delta(1,1);
for j = 2:96
    fit1(1,j) = fit1(1,(j-1)) + delta(1,j);
end
```

```
function [parameter,fit,err] = arimax(p,d,q,parameter_0,y,x)
%
%   DESCRIPTION
%       The function ARIMAX is able to perform any regression with or
%       without inclusion of autoregression, integration, moving-average or
%       external inputs by minimizing the negative log-likelihood function
%       (given by Box-Jenkins, pp.224ff, chapter 7).
%
%       Within ARIMAX the following user defined functions are called:
%       -   ARIMAXFUN defines the log-likelihood function.
%       -   ARIMAXMOD defines the model and the iterative solution
%           algorithm.
%       -   ERR_STAT_DISP and ERR_STAT define the error statistics.
%
%       ARIMAX needs the following inputs:
%       -   p as a vector for the autoregressive time-lag, like
%               p=[1:5];
%           if you want to include the last 5 values of y (the original data
%           vector) or like
%               p=[1 5];
%           if you want to include the y-value with time-lag 1 and 5 only.
%       -   d as an integer defining the integrative part. If you do not
%           want any integrative part you chose d=0. Normally a value of
%           d=1 or d=2 is sufficient if an integrative part is considered.
%       -   q as a vector for the moving-average time-lag, see description
%           for p.
%       -   parameter_0 as a vector of the starting parameters of the model.
%           This vector is normally defined as
%               parameter_0=c*ones(1+size(p,2)+size(q,2)+size(x,2),1);
%           where c is chosen to be 1.
%       -   y as the original data vector with n+offset+d columns and 1 row.
%           The offset is defined by the maximum in p or q. So if p=[1 5]
%           and q=[2 6] and d=2 the offset equals 6 and d equals 2 so y
%           needs to have 8 data points more than x
%       -   x as the external input data vector with n columns and several
%           rows for e.g. temperature data. Note that so far no constant
%           (intercept) is included in the model. If you want a constant,
%           include
%               x=[ones(size(y,1)-offset-d,1) x];
%           in the file calling ARIMAX.
%
%       ARIMAX gives the following outputs:
%       -   parameter as the vector with the obtained parameters.
%       -   fit as the vector with the fitted values for y.
%       -   err as the vector defined by the difference of the original
%           y-values and the fitted ones.
%
%   COPYRIGHT
%       (c) Derk Jan Swider (June, 28th 2002)

%check if any external inputs
if nargin == 5
   x=[];
end

%check if y is sufficiently larger than x
%define the offset by considering the time-lags
if isempty(p) == 1
    max_p = 0;
else
    max_p = max(p);
end
offset=max_p;
a=num2str(size(y,1));
b=num2str(size(x,1)+offset+d);
%now check if OK
if size(y,1) ~= size(x,1)+offset+d
    disp('Problem mit der Definition des Eingangsvektors y')
    str = ['Statt y=',a,' muß gelten y=',b];
    disp(str)
end

%calculate the integrative part
if d > 0
    for dz = 1 : d
    y=diff(y);
   end
end

%define some size-values
size2_p=size(p,2);
size2_q=size(q,2);
size2_x=size(x,2);
size1_y=size(y,1);

```

```
 86: %define the y_ar-matrix
 87: if isempty(p) == 0
 88:     for pz = 1 : size2_p
 89:         if pz == 1
 90:             y_ar=[y(offset+1-p(pz):size1_y-p(pz),1)];
 91:         elseif pz > 1
 92:             y_ar=[y_ar y(offset+1-p(pz):size1_y-p(pz),1)];
 93:         end
 94:     end
 95: else
 96:     y_ar=[];
 97: end
 98:
 99: %define the new y-vector
100: y=y(offset+d+1:size1_y);
101:
102: %define some size-values
103: size1_y=size(y,1);
104: size1_y_ar=size(y_ar,1);
105:
106: %define all errors to be zero initially
107: err=zeros(size1_y,1);
108:
109: %define the options for the optimization
110: options=optimset('TolX',1e-10,'TolFun',1e-10','MaxIter',10000,'MaxFunEvals',10000);
111:
112: %start ticking the time
113: tic
114:
115: %start minimizing the negativ log-likelihood (FMINSEARCH, ARIMAXFUN and ARIMAXMOD)
116: [parameter,fval,exitflag,output] = fminsearch('arimaxfun',parameter_0,options,y,y_ar,x,err,
     p,d,q);
117:
118: %stop ticking the time
119: toc
120:
121: %value of log-likelihood before minimization (ARIMAXFUN and ARIMAXMOD)
122: LL0 = arimaxfun(parameter_0,y,y_ar,x,err,p,d,q);
123:
124: %value of log-likelihood after minimization (ARIMAXFUN and ARIMAXMOD)
125: LL = arimaxfun(parameter,y,y_ar,x,err,p,d,q);
126:
127: %calculate the final fitted and error values (ARIMAXMOD)
128: [fit,err] = arimaxmod(parameter,y,y_ar,x,err,p,d,q);
```

```
 1: function L = arimaxfun(parameter_0,y,y_ar,x,err,p,d,q)
 2: %
 3: %   DESCRIPTION
 4: %       The function ARIMAXFUN is automatically called by the function
 5: %       ARIMAX and again calls the function ARIMAXMOD and defines the
 6: %       log-likelihood function. Two possible functions are included, one
 7: %       defined by a colleague of mine the other one defined by Box-Jenkins
 8: %       in their book Time Series Analysis Forecasting and Control on page
 9: %       224ff in the 3rd edition.
10: %
11: %   COPYRIGHT
12: %       (c) Derk Jan Swider (June, 28th 2002)
13:
14: %define the log-likelihood parameter sigma
15: sigma=parameter_0(1,1);
16:
17: %calculate the fitted and error values (ARIMAXMOD)
18: [fit,err] = arimaxmod(parameter_0,y,y_ar,x,err,p,d,q);
19:
20: %minimize the negative log-likelihood defined by Weber
21: %L=-1/2*sum(-log(2*pi*sigma^2)-(err(:,1)./sigma).^2);
22:
23: %minimize the negative log-likelihood defined by Box-Jenkins
24: L=-1/2*sum(-2*log(sigma)-(err(:,1)./sigma).^2);
25:
```

```
function [fit,err] = arimaxmod(parameter_0,y,y_ar,x,err,p,d,q)
%
%   DESCRIPTION
%       The function ARIMAXMOD is automatically called by the function
%       ARIMAX and ARIMAXFUN and defines the considered model with the
%       relevant parameters. The AR and X parts are calculated within one
%       step (saves computation time) and only the MA part is calculated
%       recursive as it needs the errors previously calculated. This part
%       is highly computation time consuming.
%
%   COPYRIGHT
%       (c) Derk Jan Swider (June, 28th 2002)

%define some size-values
size2_p=size(p,2);
size2_q=size(q,2);
size2_x=size(x,2);
size1_y=size(y,1);
size1_y_ar=size(y_ar,1);

%define the parameters
sigma=parameter_0(1,1);
alpha=parameter_0(2:size2_p+1,1);
beta=parameter_0(size2_p+2:size2_q+size2_p+1,1);
gamma=parameter_0(size2_q+size2_p+2:size(x,2)+size2_q+size2_p+1,1);

%define the fit-vector to have zero values initially
fit=zeros(size1_y,1);

%check if the AR part needs to be calculated and do the job in one step
%each
if isempty(p) == 0
    ar = y_ar * alpha;
    fit = fit + ar;
end

%check if the X part needs to be calculated and do the job in one step
%each
if isempty(x) == 0
    xi = x * gamma;
    fit = fit + xi;
end

%check if the MA part needs to be calculated and do the job recursive
if isempty(q) == 0
    %redefine err to consider the offset (defined by max of q)
    offset = max(q);
    err = [zeros(offset,1);err];
    ma = zeros(size1_y,1);
    for n = 1 + offset : size1_y
        for qz = 1 : size2_q
            ma(n-offset,1)=ma(n-offset,1)+beta(qz,1).*err(n-q(qz),1);
        end
        fit(n-offset,1)=fit(n-offset,1)+ma(n-offset,1);
        err(n,1)=y(n-offset,1)-fit(n-offset,1);
    end
end

%calculate the final error
err=y-fit;

```

```
function [errstat] = prognoser_errstat(org,fit)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Berechnung verschiedener statistischer Masszahlen.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Berechnung der benötigten Parameter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
err = org - fit;
n = size(org,2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Bestimmtheitsmass (r2).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
r2 = 1-sum(err.^2)/sum((org-sum(org)/n).^2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Root Mean Sqare Error (rmse).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
rmse = sqrt((err * err') / n);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mean Bias Error (mbe).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mbe = sum(err) / sum(org);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mean Absolut Percentage Error (mape).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mape = mean(abs(err) ./ org);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Maximum absolut Error (maxae).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
maxae = max(abs(err));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Maximum absolut Percentage Error (maxape).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
maxape = max(abs(err)./org);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Zusammenfassung in errstat.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
errstat = [r2 mbe mape maxape rmse maxae];
save errstat errstat
```

```
function prognoser_plot(org,fit)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Skript zur Plotausgabe.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(2,1,1)
m = 0:(max(max([org' fit'])));
o = 1.1 .* m;
u = 0.9 .* m;
plot(fit,org,'.k',m,m,'--r',m,o,'--b',m,u,'--b');
xlabel('fit')
ylabel('org')
subplot(2,1,2)
ind = 0:(size(org,2)-1);
plot(ind,org,'b',ind,fit,'r')
legend('org','fit')
xlabel('t')
ylabel('p_m15')
set(gca,'xtick',[0:96:(size(org,2)-1)])
```

Literaturverzeichnis

[1] **Box, G.E.P. ; Jenkins, G.M**: *Time Series Analysis: Forecasting and Controll*; San Francisco: Holden-Day, 1976.

[2] **Leiner, Bernd**: *Einführung in die Zeitreihenanalyse, 2.Auflage*; München / Wien: Oldenbourg, 1989.

[3] **Amjady, Nima**: *Short-Term Hourly Forecasting Using Time-Series with Peak Load Estimation with Peak Load Estimation Capability*; In: IEEE Transactions on Power Systems, Vol. 16, Nr. 3, August 2001 .

[4] **Hufendiek, Kai**: *Systematische Entwicklung von Lastprognosesystemen aus Basis neuronaler Netze*; In: Fortschritt-Berichte VDI, Reihe 6 , Nr. 455, Düsseldorf, 2001.