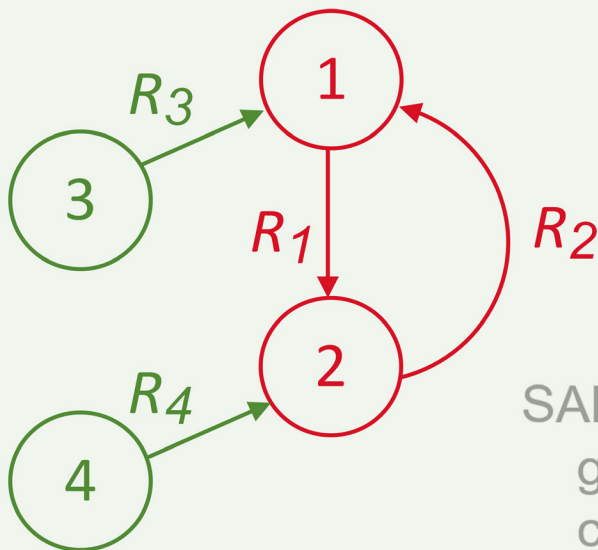


Engineering

# INSIGHTS



$$\mathcal{D} = \begin{pmatrix} p & p & z & n \\ p & p & p & n \\ p & p & p & n \\ n & z & p & p \end{pmatrix}$$

$$\nu = \begin{pmatrix} 2 & 1 & 1 & 2 \\ 0 & 3 & 3 & 2 \end{pmatrix}$$

SALS functions for signed systems  
 genAllCsv  
 csvFromCompVect  
 isSignControllable  
 isSignStabilizable

$$\mathbb{V} := \{z, p, n\}$$

Schriftenreihe der Fakultät Technik: 2/2020

## Komplexe Vorzeichenvektoren Forschungsbericht

Prof. Alfred Geisel

$$S_k = \sum_{j=1}^{k-1} \rho_j \pmod{4}$$

$$\kappa_l(l) = 4^{(l-1)} \sum_{k=1}^l (k-1)! \begin{Bmatrix} l \\ k \end{Bmatrix}$$



Professor Alfred Geisel lehrt und forscht im Fachbereich Elektrotechnik am Campus Horb der Dualen Hochschule Baden-Württemberg Stuttgart. Die Schwerpunkte liegen im Bereich der Regelungssysteme. Gemeinsam mit seinen Studierenden entwickelt er eine MATLAB-Toolbox zur strukturellen Analyse linearer Systeme.

Duale Hochschule Baden-Württemberg Stuttgart – Campus Horb  
Florianstraße 15  
72160 Horb am Neckar

E-Mail: [a.geisel@hb.dhbw-stuttgart.de](mailto:a.geisel@hb.dhbw-stuttgart.de)

## Zusammenfassung

Zur Untersuchung der Vorzeichen-Steuerbarkeit und der Vorzeichen-Stabilisierbarkeit von linearen zeitinvarianten Systemen der Form  $\dot{x}(t) = Ax(t) + Bu(t)$  mit Hilfe von algebraischen Methoden verwendet man Äquivalenzklassen komplexer Vektoren. Diese Äquivalenzklassen werden als komplexe Vorzeichenvektoren bezeichnet. In diesem Beitrag wird die Definition der komplexen Vorzeichenvektoren erläutert und es wird ausführlich beschrieben, wie man komplexe Vorzeichenvektoren systematisch konstruieren kann. Eine Funktion der MATLAB SALS Toolbox wird vorgestellt, die alle komplexen Vorzeichenvektoren der Ordnung  $n \leq 7$  erzeugen kann. Die so erzeugten komplexen Vorzeichenvektoren können von Funktionen zur Untersuchung der Vorzeichen-Steuerbarkeit und Vorzeichen-Stabilisierbarkeit verwendet werden.

## Abstract

To analyze sign controllability and sign stabilizability of linear time invariant systems of the form  $\dot{x}(t) = Ax(t) + Bu(t)$  by algebraic methods we make use of equivalence classes of complex vectors. These equivalence classes are called complex sign vectors. The definition of complex sign vectors is explained and it is demonstrated in detail how to construct them systematically. A function of the MATLAB SALS toolbox which constructs all complex sign vectors of orders  $n \leq 7$  is presented. Functions to test for sign controllability and sign stabilizability can make use of the constructed complex sign vectors.

## 1 Einleitung

Zur Überprüfung von linearen zeitinvarianten Systemen der Form  $\dot{x}(t) = Ax(t) + Bu(t)$  mit  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^r$ ,  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times r}$  und  $t \in \mathbb{R}$  auf die Eigenschaften Steuerbarkeit und Stabilisierbarkeit gibt es Methoden, die von den konkreten numerischen Parametern in den Systemmatrizen  $A$  und  $B$  unabhängig sind. Diese Methoden sind schon seit Jahrzehnten bekannt und sind in der wissenschaftlichen Literatur vielfältig behandelt worden, siehe beispielsweise [1–10].

In der Lehrbuch-Literatur finden die Methoden wenig Beachtung, eine Ausnahme bildet hier das Buch von Jan Lunze [11]. Die für MATLAB zur Verfügung stehende SALS Toolbox stellt Funktionen bereit, um von numerischen Parametern unabhängige Untersuchungen rechnergestützt durchführen zu können [12, 13]. Mit der Toolbox kön-

nen lineare zeitinvariante Systeme auf strukturelle Steuerbarkeit, streng strukturelle Steuerbarkeit, Vorzeichen-Steuerbarkeit und Vorzeichen-Stabilisierbarkeit untersucht werden.

In diesem Artikel stehen die zur Untersuchung von Systemen auf Vorzeichen-Steuerbarkeit und Vorzeichen-Stabilisierbarkeit benötigten komplexen Vorzeichenvektoren im Vordergrund. Diese Äquivalenzklassen komplexer Vektoren wurden in [9] eingeführt. Eine Einordnung von vorzeichensteuerbaren und vorzeichen-stabilisierbaren Systemen in der Klasse aller linearen zeitinvarianten Systeme ist in Abbildung 1 dargestellt.

In Abschnitt 2 wird der Begriff des komplexen Vorzeichenvektors vorgestellt. Verschiedene Darstellungsformen für komplexe Vorzeichenvektoren werden in Abschnitt 3 diskutiert. Abschnitt 4 betrachtet die Anzahl der komplexen Vorzeichenvektoren der Ordnung  $n$ . Wie man systematisch

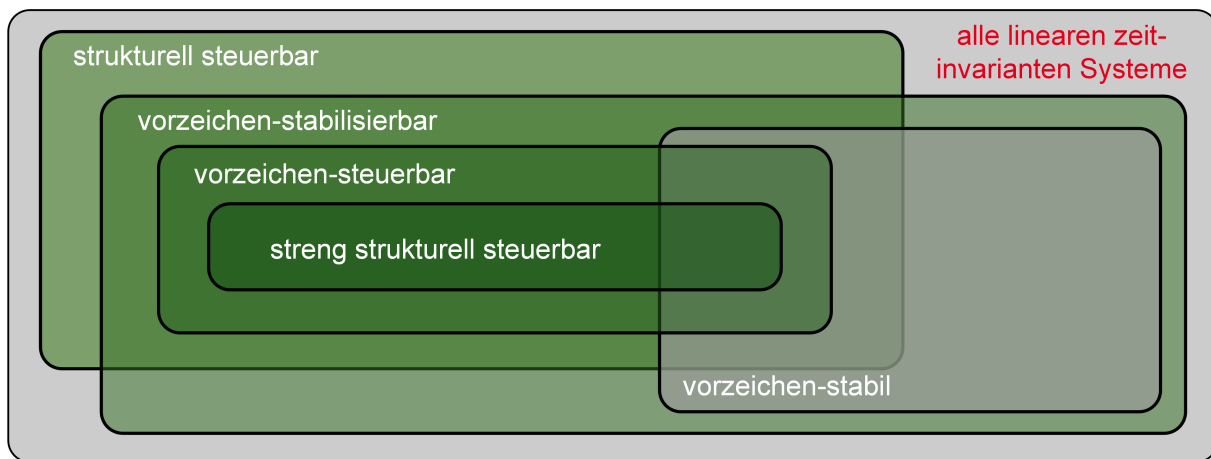


Abbildung 1: Klassendiagramm zu Eigenschaften von linearen zeitinvarianten Systemen nach [14]

komplexe Vorzeichenvektoren konstruiert, wird in Abschnitt 5 beschrieben. Hierbei ergibt sich eine weitere Darstellungsmöglichkeit für komplexe Vorzeichenvektoren. Die Prüfungen auf Vorzeichen-Steuerbarkeit und Vorzeichen-Stabilisierbarkeit werden in den Abschnitten 6 und 7 kurz skizziert. In Abschnitt 8 wird beschrieben, wie man die SALS Toolbox für die Erzeugung komplexer Vorzeichenvektoren verwendet. Einige weitere Funktionen im Zusammenhang mit komplexen Vorzeichenvektoren werden vorgestellt. Abschnitt 9 fasst die Ergebnisse zusammen und erläutert die Grenzen der Anwendbarkeit.

## 2 Komplexe Vorzeichenvektoren

Der Begriff des komplexen Vorzeichenvektors wurde von Christoph Hartung in seiner Dissertation [9] definiert und ausführlich charakterisiert. In Anlehnung daran werden nachfolgend die für das Verständnis wesentlichen Sachverhalte zusammengestellt.

Zur Untersuchung der Eigenschaften linearer Systeme ist in vielen Fällen die Bestimmung von Eigenwerten und Eigenvektoren von Matrizen erforderlich. Ist  $q \in \mathbb{C}^n$  ein Eigenvektor einer Matrix  $A \in \mathbb{R}^{n \times n}$  zum Eigenwert  $\lambda \in \mathbb{C}$ , so ist bekanntlich zu diesem Eigenwert auch jeder Vektor  $\alpha q$  mit  $\alpha \in \mathbb{C} \setminus \{0\}$  ein Eigenvektor.

Um die Vielzahl von Eigenvektoren zu einem Eigenwert beschreiben zu können, ist die Einführung von geeigneten Äquivalenzklassen hilfreich. Hierzu wurde in [9, S. 47] die folgende Relation angegeben:

**Definition 2.1.** Die binäre Relation  $\sim_c$  auf der Menge  $\mathbb{C}^n$  ist für zwei Vektoren  $x, y \in \mathbb{C}^n$  mit  $x = (r_1 e^{j\phi_1}, r_2 e^{j\phi_2}, \dots, r_n e^{j\phi_n})$  und  $y = (p_1 e^{j\theta_1}, p_2 e^{j\theta_2}, \dots, p_n e^{j\theta_n})$  gegeben durch

$$x \sim_c y \Leftrightarrow \begin{cases} \forall v, w \in \{1, \dots, n\} \text{ gilt:} \\ I: & \text{sgn}(r_v r_w \cos(\phi_v - \phi_w)) \\ & = \text{sgn}(p_v p_w \cos(\theta_v - \theta_w)) \\ II: & \text{sgn}(r_v r_w \sin(\phi_v - \phi_w)) \\ & = \text{sgn}(p_v p_w \sin(\theta_v - \theta_w)) \end{cases}$$

Die Relation  $\sim_c$  ist reflexiv, symmetrisch und transitiv und somit eine Äquivalenzrelation. Es gilt der folgende

**Satz 2.2.** Für einen komplexen Vektor  $q \in \mathbb{C}$  und eine komplexe Zahl  $\alpha \in \mathbb{C} \setminus \{0\}$  gilt:  $q \sim_c \alpha q$ .

Der kurze Beweis ist in [9, S.47] zu finden. Nach Satz 2.2 liegen alle zu einem bestimmten Eigenwert gehörigen Eigenvektoren in einer durch die Relation  $\sim_c$  bestimmten Äquivalenzklasse.

Die Menge  $\mathbb{C}^n$  der komplexen Vektoren der Ordnung  $n$  wird durch die Äquivalenzrelation  $\sim_c$  in disjunkte Äquivalenzklassen zerlegt. Diese Äquivalenzklassen werden nach Hartung als komplexe Vorzeichenvektoren bezeichnet [9, S. 48]:

**Definition 2.3.** Ein komplexer Vorzeichenvektor der Ordnung  $n$  ist eine Äquivalenzklasse  $[q] \in \mathbb{C}^n / \sim_c$  der Relation  $\sim_c$ .

## 3 Darstellungsformen für komplexe Vorzeichenvektoren

Dieser Abschnitt beschreibt einige der in [9] eingeführten Darstellungsformen für komplexe Vorzei-

chenvektoren und die zum Verständnis benötigten Grundlagen.

### 3.1 Komplexe Vorzeichenmuster

Mit Hilfe der Signum-Funktion kann man reelle Zahlen nach ihrem Vorzeichen klassifizieren. Für eine Zahl  $x \in \mathbb{R}$  gilt:

$$\operatorname{sgn}(x) := \begin{cases} -1 & \text{falls } x < 0 \\ 0 & \text{falls } x = 0 \\ 1 & \text{falls } x > 0 \end{cases} \quad (1)$$

In Anlehnung an [9, S. 11 ff.] werden zur Klassifizierung reeller Zahlen drei Mengen eingeführt:

$$p := \{x \in \mathbb{R} \mid x > 0\} \quad (2)$$

$$n := \{x \in \mathbb{R} \mid x < 0\} \quad (3)$$

$$z := \{0\} \quad (4)$$

Die Symbole  $p$ ,  $n$  und  $z$ <sup>1</sup> zur Bezeichnung dieser Mengen werden als Vorzeichen bezeichnet, und  $\mathbb{V} := \{z, p, n\}$  beschreibt die Menge der Vorzeichen. In [9] werden vier weitere Symbole zur Beschreibung von Teilmengen der reellen Zahlen eingeführt sowie der Begriff und die Menge  $\mathbb{U}$  der unsicheren Zahlen definiert. Unter Verwendung der Symbole  $zp$ ,  $zn$ ,  $pn$  und  $zpn$  notiert man  $\mathbb{U} := \{z, p, n, zp, zn, pn, zpn\}$ . Hier bedeuten  $zp$  die nichtnegativen reellen Zahlen,  $zn$  die nichtpositiven reellen Zahlen,  $pn$  die von null verschiedenen reellen Zahlen und  $zpn$  die reellen Zahlen. Die Menge der Vorzeichen  $\mathbb{V}$  ist eine Teilmenge der unsicheren Zahlen.

Für die unsicheren Zahlen kann man Rechenoperationen einführen. So werden in [9, S. 13 ff.] eine Addition und eine Multiplikation auf  $\mathbb{U}$  definiert. Die Regeln bezüglich der Multiplikation von Vorzeichen sind die mit den eingeführten Symbolen geschriebenen Vorzeichenregeln bei der Multiplikation reeller Zahlen. Beispielsweise gelten  $p \cdot n = n$ ,  $n \cdot n = p$  und  $p \cdot z = z$ . Bei der Multiplikation von Vorzeichen ist das Ergebnis wieder ein Vorzeichen, die Menge der Vorzeichen ist bezüglich der Multiplikation abgeschlossen. Das gilt allerdings nicht für die Addition: Addiert man eine positive und eine negative Zahl, so kann das Ergebnis positiv, negativ oder null sein. Mit den für die unsicheren Zahlen vereinbarten Symbolen kann dies wie folgt

<sup>1</sup> Die hier eingeführten Symbole unterscheiden sich von den in [9] verwendeten. Es werden solche Symbole verwendet, die auch in der SALS Toolbox darstellbar sind, vgl. [12].

notiert werden:  $p + n = zpn$ . Das Ergebnis der Addition von Vorzeichen kann somit nicht in jedem Fall durch ein Vorzeichen beschrieben werden. Dieser Sachverhalt ist wichtig bei der Überprüfung von Vorzeichen-Systemen auf Vorzeichen-Steuerbarkeit und Vorzeichen-Stabilisierbarkeit, spielt für die weiteren Betrachtungen im Zusammenhang mit der Konstruktion komplexer Vorzeichenvektoren allerdings keine Rolle. Die SALS Toolbox für MATLAB [13] stellt Funktionen zum Rechnen mit unsicheren Zahlen zur Verfügung.

Zur Klassifizierung einer komplexen Zahl  $c \in \mathbb{C}$  kann man die Vorzeichen des Realteils und des Imaginärteils betrachten. Offenbar gibt es neun Möglichkeiten, um komplexe Zahlen auf diese Weise zu klassifizieren: Die Vorzeichen der Real- und Imaginärteile sind für alle komplexen Zahlen eines der vier Quadranten der komplexen Zahlenebene gleich. Komplexe Zahlen, die auf einem der vier Achsenabschnitte der komplexen Ebene liegen, haben ebenfalls gleiche Vorzeichen der Real- und Imaginärteile. Somit können acht Äquivalenzklassen für komplexe Zahlen ungleich null angegeben werden. Für die Zahl Null kommt eine weitere Klasse hinzu.

Diese neun Möglichkeiten können durch Vorzeichenvektoren  $\gamma \in \mathbb{V}^2$  dargestellt werden. Die erste Komponente von  $\gamma$  soll das Vorzeichen des Imaginärteils und die zweite Komponente das Vorzeichen des Realteils darstellen. Der Vorzeichenvektor  $\gamma(c)$  beschreibt das komplexe Vorzeichenmuster einer komplexen Zahl  $c$ .

Eine komplexe Zahl  $c$  aus dem zweiten Quadranten der komplexen Zahlenebene wird beispielsweise durch den Vorzeichenvektor

$$\gamma(c) = \begin{pmatrix} p \\ n \end{pmatrix} \quad (5)$$

beschrieben, weil im zweiten Quadranten der Imaginärteil positiv und der Realteil negativ sind.

Eine auf der negativen imaginären Achse liegende komplexe Zahl  $c$  kann man durch den folgenden Vorzeichenvektor beschreiben:

$$\gamma(c) = \begin{pmatrix} n \\ z \end{pmatrix} \quad (6)$$

Bei von null verschiedenen rein reellen oder rein imaginären Zahlen ist immer genau eine der beiden Komponenten des Vorzeichenmusters  $z$ . Bei echt komplexen Zahlen sind die beiden Komponenten des Vorzeichenmusters  $p$  oder  $n$ . Für die Zahl Null sind beide Komponenten des Vorzeichenmusters  $z$ .

Ausgehend vom Vorzeichenmuster  $(p, p)^T$  im ersten Quadranten kann man mit Hilfe von vier Rotationsmatrizen aus  $\mathbb{V}^{2 \times 2}$  die Vorzeichenmuster aller vier Quadranten beschreiben. Die Menge dieser Rotationsmatrizen ist

$$R := \left\{ \begin{pmatrix} p & z \\ z & p \end{pmatrix}, \begin{pmatrix} z & p \\ n & z \end{pmatrix}, \begin{pmatrix} n & z \\ z & n \end{pmatrix}, \begin{pmatrix} z & n \\ p & z \end{pmatrix} \right\}, \quad (7)$$

vgl. [9, S. 54].

Die Matrizen der Menge  $R$  sollen in der Reihenfolge von links nach rechts mit  $R(0)$ ,  $R(1)$ ,  $R(2)$  und  $R(3)$  bezeichnet werden. Durch Angabe der Referenzzahlen  $\rho \in \{0, 1, 2, 3\}$  kann somit jede Rotationsmatrix  $R(\rho) \in R$  identifiziert werden. Wie man leicht erkennt, vermittelt die Rotationsmatrix  $R(\rho)$  eine Drehung um  $\rho \cdot \frac{\pi}{2}$ . Die Angabe der Referenzzahl  $\rho$  einer Rotationsmatrix ist damit in Bezug auf das komplexe Vorzeichenmuster äquivalent zur Angabe des entsprechenden Quadranten der komplexen Zahlenebene.

Das Vorzeichenmuster  $\gamma(c)$  einer komplexen Zahl  $c$  im Quadranten  $Q \in \{1, 2, 3, 4\}$  erhält man durch Multiplikation des Vorzeichenmusters  $(p, p)^T$  aus dem ersten Quadranten mit der Rotationsmatrix  $R(\rho)$  mit  $\rho = Q - 1$ .

Beispielsweise ergibt sich das Vorzeichenmuster im dritten Quadranten wie folgt:

$$R(2) \begin{pmatrix} p \\ p \end{pmatrix} = \begin{pmatrix} n & z \\ z & n \end{pmatrix} \begin{pmatrix} p \\ p \end{pmatrix} = \begin{pmatrix} n \\ p \end{pmatrix} \quad (8)$$

In gleicher Weise erhält man die Vorzeichenmuster der einzelnen Achsabschnitte der komplexen Zahlenebene. Geht man vom Vorzeichenmuster  $(z, p)^T$  für die positive reelle Achse aus, so erhält man beispielsweise das Vorzeichenmuster für die positive imaginäre Achse durch Multiplikation mit  $R(1)$ :

$$R(1) \begin{pmatrix} z \\ p \end{pmatrix} = \begin{pmatrix} z & p \\ n & z \end{pmatrix} \begin{pmatrix} z \\ p \end{pmatrix} = \begin{pmatrix} p \\ z \end{pmatrix} \quad (9)$$

### 3.2 Darstellung mit zwei Vorzeichenmatrizen $\mathcal{D}$ und $\mathcal{P}$

Die Äquivalenzklasse  $[q]$  eines komplexen Vektors  $q = (r_1 e^{j\phi_1}, r_2 e^{j\phi_2}, \dots, r_n e^{j\phi_n})$  ist nach Definition 2.1 durch die für  $v, w \in \{1, \dots, n\}$  gebildeten Ausdrücke

$$d_{v,w} := \operatorname{sgn}(r_v r_w \cos(\phi_v - \phi_w)) \quad (10)$$

und

$$p_{v,w} := \operatorname{sgn}(r_v r_w \sin(\phi_v - \phi_w)) \quad (11)$$

bestimmt. Diese  $2n^2$  Ausdrücke sind die charakteristischen Daten des komplexen Vorzeichenvektors  $[q]$ .

Die Ausdrücke  $d_{v,w}$  und  $p_{v,w}$  nehmen Werte aus der Bildmenge  $\{-1, 0, 1\}$  der Signum-Funktion an. Die Vorzeichen der Ausdrücke kann man gleichwertig mit Elementen der Menge  $\mathbb{V} := \{z, p, n\}$  beschreiben.

Bezeichnet man mit  $\mathcal{D}_{v,w} \in \mathbb{V}$  das Vorzeichen von  $d_{v,w}$  und mit  $\mathcal{P}_{v,w} \in \mathbb{V}$  das Vorzeichen von  $p_{v,w}$ , dann kann man die komplexen Vorzeichenvektoren auch durch Angabe aller Vorzeichen  $\mathcal{D}_{v,w}$  und  $\mathcal{P}_{v,w}$  für  $v, w \in \{1, \dots, n\}$  beschreiben.

Die charakteristischen Daten können mit zwei Matrizen  $\mathcal{D}, \mathcal{P} \in \mathbb{V}^{n \times n}$  dargestellt werden, wobei die  $\mathcal{D}_{v,w}$  und  $\mathcal{P}_{v,w}$  jeweils die Einträge in der Zeile  $v$  und der Spalte  $w$  der Matrizen sind. Somit kann man durch Angabe der beiden Vorzeichenmatrizen  $\mathcal{D}$  und  $\mathcal{P}$  einen komplexen Vorzeichenvektor charakterisieren.

Mit Hilfe der folgenden Überlegungen kann man für spezielle Situationen einige Einträge in den Matrizen  $\mathcal{D}$  und  $\mathcal{P}$  auf einfache Weise bestimmen. Man unterscheidet hierfür die beiden Fälle  $r_i \neq 0$  und  $r_i = 0$ . Nimmt man für den ersten Fall o.B.d.A.  $r_i > 0$  an, dann stehen in der Spalte  $w$  von  $\mathcal{D}$  die Vorzeichen von  $\cos(\phi_v - \phi_w)$  für  $v \in \{1, \dots, n\}$ . In der Spalte  $w$  von  $\mathcal{P}$  stehen die Vorzeichen von  $\sin(\phi_v - \phi_w)$  für  $v \in \{1, \dots, n\}$ . Bezeichnet man die  $w$ -ten Spalten jeweils mit  $\mathcal{D}_w$  und  $\mathcal{P}_w$ , so enthält die Spalte  $\mathcal{D}_w$  die Vorzeichen der Realteile und die Spalte  $\mathcal{P}_w$  die Vorzeichen der Imaginärteile der Komponenten eines auf die Komponente  $q_w$  normierten komplexen Vektors  $q$ . Für  $v = w$  sind wegen  $\cos(0) = 1$  und  $\sin(0) = 0$  die Einträge  $\mathcal{D}_{v,v} = p$  und  $\mathcal{P}_{v,v} = z$ .

Für jedes  $w \in \{1, \dots, n\}$  wird das Paar  $(\mathcal{D}_w, \mathcal{P}_w) \in \mathbb{V}^n \times \mathbb{V}^n$  als das  $w$ -te charakteristische Vorzeichenmuster bezeichnet, vgl. [9, S. 50].

Am Beispiel des komplexen Vektors

$$q = \begin{pmatrix} 7 e^{j\frac{2\pi}{3}} \\ 5 e^{j\frac{\pi}{3}} \end{pmatrix}$$

soll der Sachverhalt verdeutlicht werden.

Die Normierung auf die erste Komponente führt zum in der gleichen Äquivalenzklasse liegenden Vektor

$$\frac{q}{q_1} = \begin{pmatrix} 1 \\ \frac{5}{7} e^{-j\frac{\pi}{3}} \end{pmatrix}.$$



Die charakteristischen Daten  $d_{v,w}$  und  $p_{v,w}$  dieses Vektors für  $\phi_w = \phi_1 = 0$  und  $\phi_v = \phi_2 = -\frac{\pi}{3}$  sind:

$$\begin{aligned} d_{1,1} &= \operatorname{sgn}(\cos(0)) = 1 \\ p_{1,1} &= \operatorname{sgn}(\sin(0)) = 0 \\ d_{2,1} &= \operatorname{sgn}\left(\cos\left(-\frac{\pi}{3}\right)\right) = 1 \\ p_{2,1} &= \operatorname{sgn}\left(\sin\left(-\frac{\pi}{3}\right)\right) = -1 \end{aligned}$$

Das erste charakteristische Vorzeichenmuster  $(\mathcal{D}_1, \mathcal{P}_1)$  für den Vektor  $q$  ist somit gegeben durch

$$\begin{aligned} \mathcal{D}_1 &= \begin{pmatrix} \mathcal{D}_{1,1} \\ \mathcal{D}_{2,1} \end{pmatrix} = \begin{pmatrix} p \\ p \end{pmatrix} \\ \mathcal{P}_1 &= \begin{pmatrix} \mathcal{P}_{1,1} \\ \mathcal{P}_{2,1} \end{pmatrix} = \begin{pmatrix} z \\ n \end{pmatrix}. \end{aligned}$$

Diese Vorzeichenmuster bilden die erste Spalte in den Vorzeichenmatrizen  $\mathcal{D}$  und  $\mathcal{P}$ , die den komplexen Vorzeichenvektor  $[q]$  des komplexen Vektors  $q$  beschreiben.

Wegen der Symmetrieeigenschaften der Kosinusfunktion und der Sinusfunktion gilt  $\mathcal{D}_{1,2} = \mathcal{D}_{2,1} = p$  und  $\mathcal{P}_{1,2} = -\mathcal{P}_{2,1} = p$ . Die zweiten Spalten von  $\mathcal{D}$  und  $\mathcal{P}$  und damit die Vektoren des zweiten charakteristischen Vorzeichenmusters  $(\mathcal{D}_2, \mathcal{P}_2)$  sind

$$\begin{aligned} \mathcal{D}_2 &= \begin{pmatrix} \mathcal{D}_{1,2} \\ \mathcal{D}_{2,2} \end{pmatrix} = \begin{pmatrix} p \\ p \end{pmatrix} \\ \mathcal{P}_2 &= \begin{pmatrix} \mathcal{P}_{1,2} \\ \mathcal{P}_{2,2} \end{pmatrix} = \begin{pmatrix} p \\ z \end{pmatrix}. \end{aligned}$$

Im Fall  $r_i = 0$  ist die Komponente  $q_i$  des zu klassifizierenden Vektors  $q$  gleich null. Dann sind unter Berücksichtigung von (10) und (11) alle Ausdrücke  $d_{v,w}$  und  $p_{v,w}$  mit Beteiligung der Komponente  $q_i$  ebenfalls null. Demzufolge sind in den Matrizen  $\mathcal{D}$  und  $\mathcal{P}$  die Einträge in den Zeilen und Spalten  $i$  alle gleich  $z$ . Speziell für den Nullvektor sind sämtliche Einträge in den Matrizen  $\mathcal{D}$  und  $\mathcal{P}$  gleich  $z$ . Eine solche Vorzeichenmatrix wird als Nullmatrix bezeichnet.

### 3.3 Darstellung mit einer einzigen Vorzeichenmatrix $\mathfrak{D}$

Ein komplexer Vorzeichenvektor kann durch die  $2n^2$  Vorzeichen in den Matrizen  $\mathcal{D}$  und  $\mathcal{P}$  charakterisiert werden. Allerdings kann man wegen

der Symmetrie-Eigenschaften der Kosinusfunktion und der Sinusfunktion eine kompaktere Darstellungsform wählen. Da die Kosinusfunktion gerade und die Sinusfunktion ungerade sind, gilt  $d_{v,w} = d_{w,v}$  und  $p_{v,w} = -p_{w,v}$ . Zudem ist generell  $p_{v,v} = z$ . Somit reichen zur Beschreibung eines komplexen Vorzeichenvektors  $n^2$  Vorzeichen. Diese kann man beispielsweise in einer einzigen Vorzeichenmatrix  $\mathfrak{D} \in \mathbb{V}^{n \times n}$  darstellen. Diese Vorzeichenmatrix  $\mathfrak{D}$  repräsentiert in gleicher Weise den komplexen Vorzeichenvektor  $[q]$  wie die beiden Matrizen  $\mathcal{D}$  und  $\mathcal{P}$ . Der einfacheren Formulierung halber wird auch gelegentlich vom komplexen Vorzeichenvektor  $\mathfrak{D}$  gesprochen, obwohl  $\mathfrak{D}$  nur eine spezielle Darstellungsform der Äquivalenzklasse  $[q]$  ist.

Die Auswahl und Anordnung der Vorzeichen  $\mathcal{D}_{v,w}$  und  $\mathcal{P}_{v,w}$  in der Matrix  $\mathfrak{D}$  kann nach folgender Vorschrift vorgenommen werden [9, S. 48]:

$$\mathfrak{D}_{v,w} := \begin{cases} \mathcal{D}_{v,w} & \text{wenn } w \geq v \\ \mathcal{P}_{v,w} & \text{wenn } w < v \end{cases} \quad (12)$$

Die Vorzeichen unter Beteiligung der Kosinusfunktion stehen also oberhalb und auf der Hauptdiagonale der Matrix  $\mathfrak{D}$ , die Vorzeichen unter Beteiligung der Sinusfunktion stehen unterhalb der Hauptdiagonale.

Ist die Matrix-Repräsentation  $\mathfrak{D}$  eines komplexen Vorzeichenvektors gegeben, so kann man daraus die charakteristischen Vorzeichenmuster bestimmen. Die  $v$ -ten Komponenten des  $w$ -ten charakteristischen Vorzeichenmusters erhält man unter Berücksichtigung der Festlegung (12) mit Hilfe der folgenden Gleichungen [9, S. 50]:

$$(\mathcal{D}_w)_v := \begin{cases} \mathfrak{D}_{v,w} & \text{wenn } w \geq v \\ \mathfrak{D}_{w,v} & \text{wenn } w < v \end{cases} \quad (13)$$

$$(\mathcal{P}_w)_v := \begin{cases} -\mathfrak{D}_{w,v} & \text{wenn } w > v \\ z & \text{wenn } w = v \\ \mathfrak{D}_{v,w} & \text{wenn } w < v \end{cases} \quad (14)$$

Hierbei stehen  $(\mathcal{D}_w)_v$  und  $(\mathcal{P}_w)_v$  jeweils für die  $v$ -ten Komponenten der Spaltenvektoren  $\mathcal{D}_w$  bzw.  $\mathcal{P}_w$ . Als Einträge der Matrizen  $\mathcal{D}$  und  $\mathcal{P}$  geschrieben gilt:

$$(\mathcal{D}_w)_v = \mathcal{D}_{v,w} \quad (15)$$

$$(\mathcal{P}_w)_v = \mathcal{P}_{v,w} \quad (16)$$

### 4 Anzahl komplexer Vorzeichenvektoren

Die Anzahl unterschiedlicher komplexer Vorzeichenvektoren der Ordnung  $n \in \mathbb{N}$  ist durch

$$\kappa(n) = 1 + \sum_{l=1}^n 4^{l-1} \binom{n}{l} \sum_{k=1}^l (k-1)! \left\{ \begin{matrix} l \\ k \end{matrix} \right\} \quad (17)$$

gegeben [9, S. 96]. Hierin sind  $\binom{n}{l}$  der Binomialkoeffizient und  $\left\{ \begin{matrix} l \\ k \end{matrix} \right\} =: S(l, k)$  die Stirling-Zahl zweiter Art.

Die Stirling-Zahl zweiter Art  $S(l, k)$  gibt die Anzahl der Möglichkeiten an, eine Menge von  $l$  Elementen in genau  $k$  nicht leere disjunkte Teilmengen zu zerlegen, siehe z. B. [15, S. 285].

Die Anzahl komplexer Vorzeichenvektoren steigt mit zunehmender Systemordnung sehr schnell an. Tabelle 1 zeigt die Werte für  $\kappa(n)$  bis zur Ordnung  $n = 7$ .

$n$	$\kappa(n)$
1	2
2	11
3	124
4	2.101
5	47.766
6	1.365.375
7	46.987.088

**Tabelle 1:** Anzahl komplexer Vorzeichenvektoren der Ordnung  $n$  für  $n \leq 7$

### 5 Konstruktion komplexer Vorzeichenvektoren

Die in [9] beschriebenen Algorithmen zur Überprüfung der Vorzeichen-Steuerbarkeit und Vorzeichen-Stabilisierbarkeit machen es im allgemeinen Fall erforderlich, Tests mit allen komplexen Vorzeichenvektoren ungleich null durchzuführen. Hierzu müssen die komplexen Vorzeichenvektoren verfügbar sein.

Im Folgenden wird beschrieben, wie man systematisch alle komplexen Vorzeichenvektoren der Ordnung  $n$  generieren kann. Zum Verständnis der Vorgehensweise sind lediglich elementare mathematische Überlegungen erforderlich.

### 5.1 Komplexe Vorzeichenvektoren bis zur Ordnung 2

Zunächst werden die komplexen Vorzeichenvektoren der Ordnung 1 betrachtet. Ein komplexer Vektor der Ordnung 1 hat nur eine Komponente und wird somit durch eine einzige komplexe Zahl beschrieben:

$$q = q_1 = r_1 e^{j\phi_1} \quad (18)$$

Die darstellende Vorzeichenmatrix  $\mathfrak{D}$  hat bei  $n = 1$  nur einen Eintrag. Mit den Überlegungen aus Abschnitt 3 kann man die beiden möglichen komplexen Vorzeichenvektoren für diesen Fall angeben. Ist  $r_1 = 0$ , dann ist  $\mathfrak{D} = z$ . Ist  $r_1 \neq 0$ , dann ist  $\mathfrak{D} = \mathcal{D}_{1,1} = p$ .

Als Nächstes werden die komplexen Vorzeichenvektoren der Ordnung 2 betrachtet. Diese werden durch Vorzeichenmatrizen  $\mathfrak{D} \in \mathbb{V}^{2 \times 2}$  repräsentiert. Zum Nullvektor  $q = (0, 0)^T$  gehört die Nullmatrix

$$\mathfrak{D}_0 = \begin{pmatrix} z & z \\ z & z \end{pmatrix}.$$

Weiterhin gibt es komplexe Vektoren mit nur einer Komponente ungleich null. Diese werden durch die Vorzeichenmatrizen

$$\mathfrak{D}_1 = \begin{pmatrix} p & z \\ z & z \end{pmatrix} \text{ und } \mathfrak{D}_2 = \begin{pmatrix} z & z \\ z & p \end{pmatrix}$$

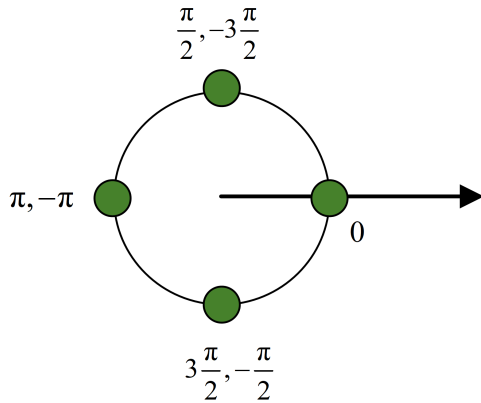
repräsentiert.

Weitere komplexe Vorzeichenvektoren klassifizieren komplexe Vektoren  $q$ , bei denen beide Komponenten  $q_1 = r_1 e^{j\phi_1}$  und  $q_2 = r_2 e^{j\phi_2}$  von null verschieden sind. Hierbei kann man folgende Fälle unterscheiden, vgl. [9, S. 52]: Die Differenzen der Argumente bzw. Winkel  $\phi_1$  und  $\phi_2$  der beiden Vektorkomponenten sind ganzzahlige Vielfache von  $\frac{\pi}{2}$ , d.h.  $\phi_1 - \phi_2 = \tilde{k} \frac{\pi}{2}$  mit  $\tilde{k} \in \mathbb{Z}$ . In diesem Fall werden die Komponenten  $q_1$  und  $q_2$  von  $q$  als voneinander abhängig bezeichnet. Ist die Winkeldifferenz  $\phi_1 - \phi_2 \neq \tilde{k} \frac{\pi}{2}$ , so bezeichnet man die Komponenten als unabhängig.

Zunächst wird der Fall der abhängigen Komponenten betrachtet. Da für die Klassifikation nach Definition 2.1 nur Winkeldifferenzen von Bedeutung sind, kann man o.B.d.A.  $\phi_1 = 0$  setzen. Das erreicht man durch eine geeignete Drehung oder durch Normieren von  $q$  auf die Komponente  $q_1$ , also Betrachtung von  $q q_1^{-1}$ . Die Komponente  $q_1$  wird als Bezugskomponente bezeichnet.



In Abbildung 2 ist der Fall abhängiger Komponenten schematisch dargestellt. Der nach rechts weisende Pfeil soll die in die reelle Achse gedrehte Komponente 1 darstellen. Wegen der Abhängigkeit der beiden Komponenten kann der Winkel der zweiten Komponente nur an den grün markierten Stellen liegen.



**Abbildung 2:** Mögliche Winkel der abhängigen Komponente bei  $\phi_1 = 0$

Die Winkel  $\phi_2$  geben die vier möglichen Verdrehungen der Komponente 2 relativ zur Bezugskomponente 1 an. Für die Winkeldifferenzen gilt:  $\phi_1 - \phi_2 = -\phi_2 = \tilde{k}\frac{\pi}{2}$ , wobei man o.B.d.A. für  $\tilde{k} \in \{-3, -2, \dots, 3\}$  annehmen darf. Mit den Rechenregeln für modulare Arithmetik kann man die negativen Werte für  $\tilde{k}$  in Werte aus  $\{0, 1, 2, 3\}$  abbilden: Es gilt allgemein  $(-a) \bmod b = (b - a) \bmod b$  für  $b > 0$ , sofern  $a$  kein Vielfaches von  $b$  ist, vgl. [16, S. 133]. In unserem Fall ergibt sich  $k = \tilde{k} \bmod 4$ . Die Verdrehung der abhängigen Komponente 2 zur Bezugskomponente 1 kann man somit durch Wahl eines Werts  $k \in \{0, 1, 2, 3\}$  eindeutig beschreiben. Anschaulich ist das in Abbildung 2 zu erkennen.

Es soll nun exemplarisch den Fall  $\phi_2 = \frac{\pi}{2}$  betrachtet werden. Die Winkeldifferenz  $\phi_1 - \phi_2$  ist somit  $-\frac{\pi}{2}$ , also  $\tilde{k} = -1$  bzw.  $k = 3$ . Zur Vereinfachung sollen die  $r_i$  positive reelle Zahlen sein. Wegen  $\cos(\phi_1 - \phi_2) = \cos(-\frac{\pi}{2}) = 0$  wird damit  $\mathcal{D}_{1,2} = z$ .  $\sin(\phi_1 - \phi_2) = \sin(-\frac{\pi}{2}) = -1$  führt zu  $\mathcal{P}_{1,2} = n$  bzw.  $\mathcal{P}_{2,1} = p$ . Dies bedeutet für die Einträge von  $\mathcal{D}$ :

$$\mathcal{D}_{1,2} = \mathcal{D}_{1,2} = z \quad \mathcal{D}_{2,1} = \mathcal{P}_{2,1} = p$$

Die Einträge in der Hauptdiagonale von  $\mathcal{D}$  sind von null verschiedene Vektorkomponenten nach den Überlegungen aus Abschnitt 3 immer gleich  $p$ . Damit liegt für dieses Beispiel die Vorzeichenmatrix für den komplexen Vorzeichenvektor fest:

$$\mathcal{D}_3 = \begin{pmatrix} p & z \\ p & p \end{pmatrix}$$

Anhand von Abbildung 2 ist leicht zu erkennen, dass auf diese Art drei weitere komplexe Vorzeichenvektoren erzeugt werden können. Tabelle 2 gibt an, wie in Abhängigkeit von  $k$  die beiden Einträge außerhalb der Hauptdiagonale, also  $\mathcal{D}_{1,w}$  und  $\mathcal{D}_{w,1}$  mit  $w = 2$ , gewählt werden müssen.

$k$	0	1	2	3
$\mathcal{D}_{1,w}$	p	z	n	z
$\mathcal{D}_{w,1}$	z	n	z	p

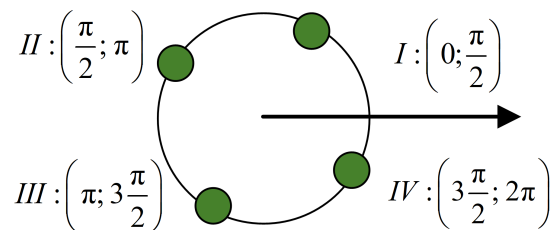
**Tabelle 2:** Einträge in der Matrix  $\mathcal{D}$  in Abhängigkeit von  $k$  bei abhängigen Komponenten

Die weiteren komplexen Vorzeichenvektoren mit abhängigen Komponenten werden durch die folgenden Vorzeichenmatrizen repräsentiert:

$$\mathcal{D}_4 = \begin{pmatrix} p & p \\ z & p \end{pmatrix} \quad \mathcal{D}_5 = \begin{pmatrix} p & z \\ n & p \end{pmatrix}$$

$$\mathcal{D}_6 = \begin{pmatrix} p & n \\ z & p \end{pmatrix}$$

Nun wird der Fall betrachtet, dass die beiden Komponenten des Vektors  $q$  unabhängig sind. Wenn wieder die erste Komponente die Bezugskomponente ist, dann kann der Winkel der zweiten Komponente in den vier in Abbildung 3 dargestellten offenen Intervallen liegen. Diese Intervalle repräsentieren die durch die römischen Zahlen I bis IV bezeichneten Quadranten der komplexen Zahlenebene. In jedem Quadranten ist beispielhaft eine mögliche Winkelposition für die zweite Komponente grün markiert.



**Abbildung 3:** Beispiele für Winkelbeziehungen bei unabhängigen Komponenten

Zur weiteren Erläuterung wird der Fall betrachtet, dass  $\phi_2$  in Bezug auf  $\phi_1$  im ersten Quadranten der komplexen Ebene liegt, dass also  $0 < \phi_2 < \frac{\pi}{2}$  bei  $\phi_1 = 0$  gilt. Unter einem Quadranten in der komplexen Ebene soll entsprechend

der Beschreibung durch offene Intervalle ein offener Quadrant verstanden werden. Die begrenzenden Achsen gehören somit nicht zu den Quadranten. Für den betrachteten Fall ist  $\mathcal{D}_{1,2} = p$ , da  $\cos(\phi_1 - \phi_2) = \cos(-\phi_2)$  im betrachteten Intervall immer positiv ist.  $\mathcal{P}_{1,2} = n$ , da  $\sin(\phi_1 - \phi_2) = \sin(-\phi_2)$  im betrachteten Intervall immer negativ ist. Wegen  $\mathcal{P}_{2,1} = -\mathcal{P}_{1,2}$  ist  $\mathcal{P}_{2,1} = p$ . Die Einträge in der Matrix  $\mathcal{D}$  außerhalb der Hauptdiagonale sind somit:

$$\mathcal{D}_{1,2} = \mathcal{D}_{1,2} = p \quad \mathcal{D}_{2,1} = \mathcal{P}_{2,1} = p$$

Der komplexe Vorzeichenvektor mit unabhängigen Komponenten, bei dem  $\phi_1 = 0$  und  $0 < \phi_2 < \frac{\pi}{2}$  gilt, wird also durch die Vorzeichenmatrix

$$\mathcal{D}_7 = \begin{pmatrix} p & p \\ p & p \end{pmatrix}$$

repräsentiert.

Offensichtlich gibt es drei weitere komplexe Vorzeichenvektoren in dieser Konstellation. Diese können mit Hilfe von Tabelle 3 mit  $w = 2$  konstruiert werden.

Quadrant von $\phi_w$	I	II	III	IV
$\mathcal{D}_{1,w}$	p	n	n	p
$\mathcal{D}_{w,1}$	p	p	n	n

**Tabelle 3:** Einträge in der Matrix  $\mathcal{D}$  in Abhängigkeit des Quadranten von  $\phi_w$

Die weiteren Vorzeichenmatrizen zur Beschreibung der komplexen Vorzeichenvektoren mit zwei unabhängigen Komponenten sind nachfolgend aufgelistet:

$$\mathcal{D}_8 = \begin{pmatrix} p & n \\ p & p \end{pmatrix} \quad \mathcal{D}_9 = \begin{pmatrix} p & n \\ n & p \end{pmatrix}$$

$$\mathcal{D}_{10} = \begin{pmatrix} p & p \\ n & p \end{pmatrix}$$

Damit sind alle elf komplexen Vorzeichenvektoren für komplexe Vektoren der Ordnung  $n = 2$  ermittelt.

### 5.2 Komplexe Vorzeichenvektoren höherer Ordnung

Bisher wurden die komplexen Vorzeichenvektoren für die Ordnungen 1 und 2 konstruiert und aufgeschrieben. Für die Ordnung  $n = 3$  müsste man gemäß Tabelle 1 bereits 124 komplexe Vorzeichenvektoren konstruieren und notieren. Ohne

Rechnerunterstützung ist das natürlich sehr mühselig. Zudem müssen für die Konstruktion komplexer Vorzeichenvektoren ab  $n = 3$  weitere Situationen berücksichtigt werden. Es gibt die bereits für  $n = 2$  diskutierten Fälle, dass entweder alle von null verschiedenen Komponenten unabhängig oder alle abhängig sind. Hinzu kommen die Fälle, bei denen zwei Komponenten als unabhängig und die dritte Komponente als abhängig zu einer der beiden anderen betrachtet werden müssen. Wenn mehr als zwei unabhängige Komponenten vorliegen, dann ist zusätzlich die Lage der Argumente der unabhängigen Komponenten zueinander zu berücksichtigen. Alle diese Fälle werden im Folgenden zunächst für die Ordnung  $n = 3$  erläutert.

Einige einfache Fälle von komplexen Vorzeichenvektoren für  $n = 3$  können sofort angegeben werden. Die Nullmatrix

$$\mathcal{D}_0 = \begin{pmatrix} z & z & z \\ z & z & z \\ z & z & z \end{pmatrix}$$

stellt den komplexen Vorzeichenvektor für den Nullvektor  $q = (0, 0, 0)^T$  dar.

Betrachtet man Vektoren der Ordnung 3 mit nur einer von null verschiedenen Komponente, so kann man auch dafür gleich die darstellenden Vorzeichenmatrizen angeben:

$$\mathcal{D}_1 = \begin{pmatrix} p & z & z \\ z & z & z \\ z & z & z \end{pmatrix} \quad \mathcal{D}_2 = \begin{pmatrix} z & z & z \\ z & p & z \\ z & z & z \end{pmatrix}$$

$$\mathcal{D}_3 = \begin{pmatrix} z & z & z \\ z & z & z \\ z & z & p \end{pmatrix}$$

Nun werden Vektoren mit zwei von null verschiedenen Komponenten betrachtet. Die acht Vorzeichenmatrizen für komplexe Vorzeichenvektoren der Ordnung  $n = 2$  mit zwei von null verschiedenen Komponenten sind bereits bekannt. Bekannt ist auch, dass für eine Komponente  $q_i = 0$  die Einträge in der Zeile  $i$  und Spalte  $i$  der Vorzeichenmatrix  $\mathcal{D}$  alle  $z$  sind. Es gibt drei Möglichkeiten, in einem Vektor der Ordnung  $n = 3$  genau zwei von null verschiedene Komponenten  $q_1$  und  $q_2$  zu platzieren:

$$\begin{pmatrix} q_1 \\ q_2 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0 \\ q_1 \\ q_2 \end{pmatrix} \quad \begin{pmatrix} q_1 \\ 0 \\ q_2 \end{pmatrix}$$

Für jede der Vorzeichenmatrizen der Ordnung  $n = 2$ , die einen komplexen Vorzeichenvektor mit zwei

von null verschiedenen Komponenten repräsentieren, kann man daher drei Vorzeichenmatrizen der Ordnung  $n = 3$  konstruieren. Das ist für die in Abschnitt 5.1 bestimmten Matrizen  $\mathfrak{D}_3, \dots, \mathfrak{D}_{10} \in \mathbb{V}^{2 \times 2}$  der Fall. Somit ergeben sich  $3 \cdot 8 = 24$  Vorzeichenmatrizen, die komplexe Vorzeichenvektoren der Ordnung  $n = 3$  mit zwei von null verschiedenen Komponenten repräsentieren.

Das folgende Beispiel soll zur Verdeutlichung dienen. Die Matrix

$$\mathfrak{D}_5 = \begin{pmatrix} p & z \\ n & p \end{pmatrix}$$

repräsentiert einen komplexen Vorzeichenvektor der Ordnung  $n = 2$  mit zwei von null verschiedenen Komponenten. Es wird untersucht, welche komplexen Vorzeichenvektoren der Ordnung  $n = 3$  daraus konstruiert werden können.

In den Matrizen für  $n = 3$  steht in den zu einer Nullkomponente im Vektor  $q$  gehörigen Zeilen und Spalten von  $\mathfrak{D}$  ausschließlich das Vorzeichen  $z$ . In die anderen Einträge werden die Werte aus der Matrix  $\mathfrak{D}_5 \in \mathbb{V}^{2 \times 2}$  übernommen. Das Ergebnis der so erzeugten Vorzeichenmatrizen  $\mathfrak{D}_i \in \mathbb{V}^{3 \times 3}$  ist nachfolgend aufgelistet:

$$\mathfrak{D}_4 = \begin{pmatrix} p & z & z \\ n & p & z \\ z & z & z \end{pmatrix} \quad \mathfrak{D}_5 = \begin{pmatrix} z & z & z \\ z & p & z \\ z & n & p \end{pmatrix}$$

$$\mathfrak{D}_6 = \begin{pmatrix} p & z & z \\ z & z & z \\ n & z & p \end{pmatrix}$$

Generiert man komplexe Vorzeichenvektoren der Ordnung  $n = 4$  mit zwei von null verschiedenen Elementen, dann hat man  $\binom{4}{2} = 6$  Möglichkeiten, die von null verschiedenen Komponenten im Vektor  $q$  zu platzieren. In diesem Fall gibt es  $6 \cdot 8 = 48$  Vorzeichenmatrizen für die komplexen Vorzeichenvektoren der Ordnung  $n = 4$  mit zwei von null verschiedenen Komponenten.

Allgemein kann man aus jedem komplexen Vorzeichenvektor der Ordnung  $k$  mit genau  $k$  von null verschiedenen Komponenten  $\binom{n}{k}$  verschiedene Vorzeichenvektoren der Ordnung  $n$  mit genau  $k$  von null verschiedenen Komponenten erzeugen.

### 5.3 Algorithmen zur Konstruktion der komplexen Vorzeichenmatrizen

Verallgemeinert man die bisherigen Überlegungen, so kann man den folgenden Algorithmus angeben:

**Algorithmus 1.** Bestimmen der darstellenden Vorzeichenmatrizen  $\mathfrak{D}$  aller komplexen Vorzeichenvektoren der Ordnung  $n$

1. Generiere die Nullmatrix als speziellen komplexen Vorzeichenvektor.
2. Initialisiere  $k := 0$ .
3. Setze  $k := k + 1$ .
4. Generiere alle darstellenden Matrizen  $\mathfrak{D} \in \mathbb{V}^{k \times k}$  für Vektoren der Ordnung  $k$  mit genau  $k$  von null verschiedenen Komponenten.
5. Konstruiere daraus die  $\binom{n}{k}$  Vorzeichenmatrizen  $\mathfrak{D} \in \mathbb{V}^{n \times n}$  der Ordnung  $n$  für Vektoren mit  $k$  von null verschiedenen Komponenten.
6. Wenn  $k < n$ , gehe zu 3.
7. Ende

Das Problem der Konstruktion von komplexen Vorzeichenvektoren ist damit zurückgeführt auf das Problem, die darstellenden Vorzeichenmatrizen  $\mathfrak{D}$  für Vektoren der Ordnung  $k$  mit genau  $k$  von null verschiedenen Komponenten zu finden. Im Algorithmus 1 ist dies der Schritt 4.

Für den Fall  $n = 3$  wurden bereits 28 Vorzeichenmatrizen für Vektoren mit weniger als drei von null verschiedenen Komponenten generiert. Somit sind noch 96 der 124 komplexen Vorzeichenvektoren zu konstruieren. Das sind die komplexen Vorzeichenvektoren der Ordnung  $n = 3$ , die genau drei von null verschiedene Komponenten haben. Zur Konstruktion der entsprechenden Vorzeichenmatrizen  $\mathfrak{D}$  wird nachfolgend eine Methode entwickelt, die für beliebige  $n > 3$  verallgemeinert und als Algorithmus formuliert werden kann.

Die folgenden drei bei der Konstruktion von komplexen Vorzeichenvektoren der Ordnung  $n \geq 3$  mit  $n$  von null verschiedenen Komponenten auftretenden Fälle werden unterschieden:

1. Alle Komponenten sind abhängig.
2. Alle Komponenten sind unabhängig.
3. Es gibt abhängige und unabhängige Komponenten.

Um alle unterschiedlichen Möglichkeiten benennen zu können, wird die Menge der Indizes  $I_3 := \{1, 2, 3\}$  aller Komponenten eines Vektors mit  $n = 3$  definiert. Zur Vereinfachung spricht man auch vom Element  $k$  eines Vektors, wenn dessen Komponente mit dem Index  $k$  gemeint ist. Die Menge

$I_3$  wird in eine, zwei und drei nichtleere Teilmengen zerlegt und die Partitionen werden wie folgt notiert:

$$\{\{1, 2, 3\}\}; \quad \{\{1, 2\}, \{3\}\}; \quad \{\{1, 3\}, \{2\}\}; \\ \{\{1\}, \{2, 3\}\}; \quad \{\{1\}, \{2\}, \{3\}\}$$

Für den betrachteten Fall  $n = 3$  ist die Anzahl der Partitionen noch sehr übersichtlich. Zu den beiden trivialen Zerlegungen in eine und drei Teilmengen kommen noch  $S(3, 2) = 3$  Zerlegungen in zwei Teilmengen hinzu. Für größere Ordnungen wächst die Anzahl der Partitionen schnell an. Zur systematischen Erzeugung der Partitionen sind Algorithmen bekannt, siehe z. B. [17].

Die Teilmengen innerhalb einer Partition werden als Gruppen bezeichnet. Die Elemente innerhalb einer Gruppe sind nach aufsteigender Reihenfolge geordnet. Alle Elemente in einer Gruppe betrachtet man als abhängig. O.B.d.A. kann das erste Element in einer Gruppe als das zu den Elementen anderer Gruppen unabhängige Element und als Bezugselement für die Gruppe betrachtet werden.

Damit sind der Fall 1 mit nur abhängigen Komponenten durch die Partition  $\{\{1, 2, 3\}\}$  und der Fall 2 mit nur unabhängigen Komponenten durch die Partition  $\{\{1\}, \{2\}, \{3\}\}$  beschrieben. Die anderen drei Partitionen beschreiben den Fall 3 mit sowohl abhängigen als auch unabhängigen Komponenten.

Zunächst soll der Fall 1 von abhängigen Elementen betrachtet werden. Es gibt  $4^2 = 16$  Möglichkeiten, die Winkel  $\phi_2$  und  $\phi_3$  in Relation zum Element 1 zu wählen.

Mit Hilfe von Abbildung 2 und Tabelle 2 können die Einträge in Spalte 1 und Zeile 1 von  $\mathcal{D}$  ermittelt werden. Es fehlen noch die Einträge  $\mathcal{D}_{2,3}$  und  $\mathcal{D}_{3,2}$ . Diese kann man mit Hilfe einer Verallgemeinerung von Tabelle 2 ermitteln. Die Winkel von Element 2 und Element 3 bei Normierung auf Element 1 sind wegen der Abhängigkeit durch  $k_2 \frac{\pi}{2}$  und  $k_3 \frac{\pi}{2}$  mit  $k_2, k_3 \in \mathbb{Z}$  gegeben. Man definiert als Distanz

$$d_k := (k_2 - k_3) \bmod 4$$

und erhält so für  $d_k$  Werte aus  $\{0, 1, 2, 3\}$ . Distanz 1 bedeutet beispielsweise, dass  $\phi_2 - \phi_3 \in \{\frac{\pi}{2}, -3\frac{\pi}{2}\}$  ist. Diese Überlegungen kann man wie folgt verallgemeinern.

Zur Ermittlung der Einträge  $\mathcal{D}_{v,w}$  und  $\mathcal{D}_{w,v}$  für  $v < w$  bildet man die Distanz

$$d_k := (k_v - k_w) \bmod 4. \quad (19)$$

In Tabelle 4 können unter Berücksichtigung von  $d_k$  die Werte der Einträge von  $\mathcal{D}$  für abhängige Elemente  $v, w$  mit  $v < w$  abgelesen werden.

$d_k$	0	1	2	3
$\mathcal{D}_{v,w}$	p	z	n	z
$\mathcal{D}_{w,v}$	z	n	z	p

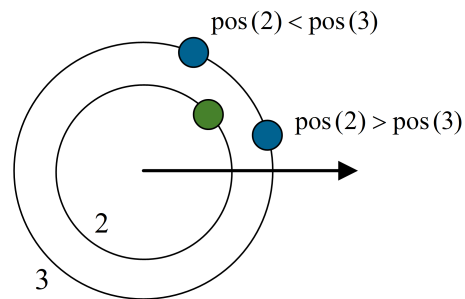
**Tabelle 4:** Einträge in der Matrix  $\mathcal{D}$  in Abhängigkeit von  $d_k$  bei abhängigen Elementen und  $v < w$

Tabelle 4 ist eine Verallgemeinerung von Tabelle 2 und kann daher generell für die Ermittlung der Einträge in  $\mathcal{D}$  bei abhängigen Elementen herangezogen werden. Die durch Tabelle 2 beschriebenen Festlegungen erhält man mit Tabelle 4 durch die Wahl von  $v = 1$  und  $k_v = 0$ .

Für die Betrachtung von Fall 2 ist die folgende Definition hilfreich:

**Definition 5.1.** Unter der Position  $\text{pos}(i)$  der Vektorkomponente mit dem Index  $i$  versteht man  $\text{pos}(i) := \phi_i \bmod \frac{\pi}{2}$ .

Im Fall 2 sind alle Komponenten in  $q$  unabhängig. Die Komponente 1 soll wieder als Bezugskomponente<sup>2</sup> betrachtet und o.B.d.A. der Winkel  $\phi_1 = 0$  gesetzt werden. Anhand von Abbildung 4 soll erläutert werden, dass bei mehr als zwei unabhängigen Elementen zusätzliche Überlegungen erforderlich sind.



**Abbildung 4:** Beispiel für die Lage der unabhängigen Elemente im gleichen Quadranten bei  $n = 3$

<sup>2</sup> Im allgemeinen Fall betrachtet man die erste Komponente eines Vektors, die ungleich null ist, als Bezugskomponente. deren Position ist 0. Alle anderen unabhängigen Komponenten  $k$  haben eine Position zwischen  $0 < \text{pos}(k) < \frac{\pi}{2}$ . Im betrachteten Beispiel wurde vorausgesetzt, dass alle Komponenten ungleich null sind. Somit ist die Komponente 1 die Bezugskomponente.

Für die Ermittlung der Einträge in der ersten Zeile und Spalte von  $\mathcal{D}$  ergibt sich kein Unterschied zum Fall  $n = 2$ . Man kann wieder die Tabelle 3 zur Ermittlung von  $\mathcal{D}_{1,w}$  und  $\mathcal{D}_{w,1}$  heranziehen. Wegen der Normierung auf Element 1 und der Unabhängigkeit der Elemente ist immer gewährleistet, dass  $\text{pos}(1) < \text{pos}(w)$  ist.

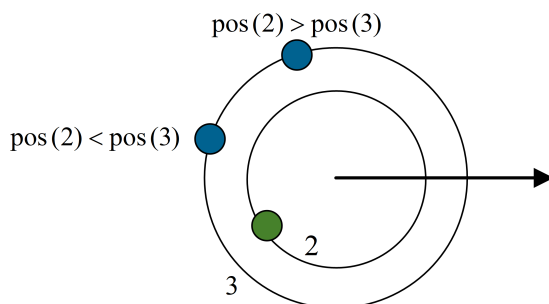
Abbildung 4 zeigt an einem Beispiel die Lage der Elemente 2 und 3 relativ zum Element 1. Die eingezeichnete Lage von Element 2 und die zwei unterschiedlichen Lagen für das Element 3 liegen alle im ersten Quadranten.

Die mit  $\text{pos}(2) > \text{pos}(3)$  beschriebene Situation führt dazu, dass die Differenz der Argumente  $\phi_2 - \phi_3$  im ersten Quadranten liegt, also im offenen Intervall  $(0, \frac{\pi}{2})$ . In diesem Intervall ist die Kosinusfunktion positiv, was zu  $\mathcal{D}_{2,3} = p$  führt. Die Sinusfunktion ist ebenfalls positiv, so dass  $\mathcal{P}_{2,3} = p$ . Dies führt zu den Einträgen  $\mathcal{D}_{2,3} = p$  und  $\mathcal{D}_{3,2} = n$ .

Anders sieht es für die mit  $\text{pos}(2) < \text{pos}(3)$  beschriebene Situation aus. Hier liegt die Differenz der Argumente  $\phi_2 - \phi_3$  im vierten Quadranten, also im offenen Intervall  $(-\frac{\pi}{2}, 0)$ . In diesem Intervall ist die Kosinusfunktion positiv und damit  $\mathcal{D}_{2,3} = p$ . Die Sinusfunktion liefert negative Werte und daher ist  $\mathcal{P}_{2,3} = n$ , was zu den Einträgen  $\mathcal{D}_{2,3} = p$  und  $\mathcal{D}_{3,2} = p$  führt.

Im Unterschied zu diesem Beispiel können im Allgemeinen die Elemente 2 und 3 in beliebigen Quadranten relativ zu 1 liegen. Daher müssen die Quadranten der unabhängigen Elemente ebenfalls zur Bestimmung der Einträge in  $\mathcal{D}$  berücksichtigt werden.

Zur Erläuterung wird ein zweites Beispiel analysiert, bei dem die Elemente 2 und 3 in unterschiedlichen Quadranten liegen. Eine solche Situation zeigt Abbildung 5.



**Abbildung 5:** Beispiel für die Lage der unabhängigen Elemente in unterschiedlichen Quadranten

Hier ist Element 2 im dritten Quadranten und die beiden Alternativen für das Element 3 liegen im zweiten Quadranten. Für  $\text{pos}(2) > \text{pos}(3)$  ist die Differenz der Argumente  $\phi_2 - \phi_3$  im zweiten Quadranten. Ist hingegen  $\text{pos}(2) < \text{pos}(3)$ , so liegt die Differenz der Argumente  $\phi_2 - \phi_3$  im ersten Quadranten. In beiden Fällen ist die Differenz der Quadranten gleich 1. Die für die Ermittlung der Vorzeichen zu betrachtenden Quadranten werden als effektive Quadranten bezeichnet. Diese hängen sowohl von der Differenz der Quadranten der betrachteten Komponenten als auch von deren Position ab.

Eine Analyse aller Kombinationen führt zu Tabelle 5. Sie gibt an, in welchem effektiven Quadranten die Differenz der Argumente  $\phi_v - \phi_w$  zu liegen kommt, wenn die Differenz der Quadranten

$$d_Q := [Q(v) - Q(w)] \bmod 4 \quad (20)$$

ist. Hier bedeuten  $Q(v)$  und  $Q(w)$  die Quadranten der Winkel  $\phi_v$  und  $\phi_w$ .

$d_Q$	$\text{pos}(v) < \text{pos}(w)$	$\text{pos}(v) > \text{pos}(w)$
0	IV	I
1	I	II
2	II	III
3	III	IV

**Tabelle 5:** Effektive Quadranten zur Ermittlung der Vorzeichen von  $\mathcal{D}_{v,w}$  und  $\mathcal{P}_{v,w}$  bei Differenz  $d_Q$  der Quadranten von  $v$  und  $w$

Anhand des so bestimmten effektiven Quadranten kann man die Einträge in  $\mathcal{D}$  ermitteln. Ist der effektive Quadrant beispielsweise der vierte Quadrant IV, so ist  $\mathcal{D}_{2,3} = p$  und  $\mathcal{P}_{2,3} = n$ . Daraus ergeben sich unter Berücksichtigung von (12) die Einträge in  $\mathcal{D}$ .

Damit ist die komplette Situation bei unabhängigen Elementen beschrieben. Es gibt  $4^2 = 16$  unterschiedliche Kombinationen der Quadranten für die Elemente 2 und 3. Außerdem gibt es  $(3-1)! = 2$  Möglichkeiten, die Positionen der Elemente 2 und 3 zu wählen. Dies führt also zu 32 komplexen Vorzeichenvektoren. Im allgemeinen Fall gibt es  $4^{n-1}$  unterschiedliche Kombinationen der Quadranten für die Elemente 2, ...,  $n$  und  $(n-1)!$  Möglichkeiten zur Wahl der Positionen dieser Elemente. Das führt zu  $4^{n-1}(n-1)!$  komplexen Vorzeichenvektoren mit  $n$  voneinander unabhängigen Elementen.



Es verbleibt noch, den Fall 3 zu diskutieren. Dieser liefert die noch ausstehenden 48 komplexen Vorzeichenvektoren. Bei den Partitionen

$$\{\{1, 2\}, \{3\}\}; \{\{1, 3\}, \{2\}\}; \{\{1\}, \{2, 3\}\}$$

gibt es jeweils eine Gruppe mit zwei Elementen und eine weitere mit nur einem Element.

Zunächst soll die Partition  $\{\{1, 2\}, \{3\}\}$  betrachtet werden. Die Elemente 1 und 3 sind unabhängig. Es gibt vier Möglichkeiten für die Wahl des Quadranten von Element 3 bezüglich Element 1. Weiterhin gibt es vier Möglichkeiten zur Wahl der Verdrehung von Element 2 relativ zum Element 1. Somit liefert diese Partition 16 komplexe Vorzeichenvektoren. In gleicher Weise kann man für die anderen beiden Partitionen argumentieren und kommt so insgesamt auf 48 komplexe Vorzeichenvektoren.

Für die Bestimmung der Einträge  $\mathcal{D}_{1,3}$  und  $\mathcal{D}_{3,1}$  für die unabhängigen Elemente 1 und 3 verwendet man Tabelle 3. Für die Bestimmung der Einträge  $\mathcal{D}_{1,2}$  und  $\mathcal{D}_{2,1}$  der zueinander abhängigen Elemente greift man auf Tabelle 4 zurück.

Es fehlen jetzt noch die Einträge in  $\mathcal{D}$ , welche die Relation zwischen dem zu 1 abhängigen Element 2 und dem unabhängigen Element 3 repräsentieren. Für die Bestimmung dieser Einträge sind die Quadranten der unabhängigen Elemente 1 und 3 sowie die Verdrehung  $k_2 \in \{0, \dots, 3\}$  des abhängigen Elements 2 zu seinem Bezugselement 1 von Relevanz. Der Quadrant eines abhängigen Elements ergibt sich als Summe aus dem Quadranten des unabhängigen Elements in der Gruppe und dem  $k$ -Wert des abhängigen Elements. Im Beispiel ist der Quadrant von Element 2 also  $Q(2) = Q(1) + k_2$ . Für  $v = 2$  und  $w = 3$  ist damit die Differenz  $d_Q = [Q(2) - Q(3)] \bmod 4$  zu ermitteln. Mit Tabelle 5 findet man die effektiven Quadranten und kann damit die Einträge in den Matrizen  $\mathcal{D}$  und  $\mathcal{P}$  und somit in  $\mathcal{D}$  bestimmen. Hierbei ist zu berücksichtigen, dass  $\text{pos}(3) > \text{pos}(2)$  ist, da das Element 2 zu 1 abhängig ist und  $\text{pos}(3) > \text{pos}(1)$  ist.

Für diese Partition ist somit geklärt, wie die charakteristischen Daten des komplexen Vorzeichenvektors ermittelt werden.

Für  $n = 3$  kommt es nicht vor, dass mehrere Gruppen mit mehr als einem Element auftreten können. Um eine solche Situation zu analysieren, soll die folgende zu einem Vektor der Ordnung  $n = 5$  gehörigen Partition betrachtet werden:

$$\{\{1\}; \{2, 3\}; \{4, 5\}\}$$

Diese Partition besteht aus drei Gruppen. Die Elemente 1, 2 und 4 werden als unabhängige Elemente betrachtet. Die Einträge  $\mathcal{D}_{1,2}$ ,  $\mathcal{D}_{2,1}$ ,  $\mathcal{D}_{1,4}$  und  $\mathcal{D}_{4,1}$  sind somit durch die Quadranten gegeben, in denen die Winkel von 2 und 4 relativ zu 1 liegen. Diese Einträge bestimmt man mit Tabelle 3. Die Einträge  $\mathcal{D}_{2,3}$ ,  $\mathcal{D}_{3,2}$  sowie  $\mathcal{D}_{4,5}$  und  $\mathcal{D}_{5,4}$  können mit Hilfe der Tabelle 4 bestimmt werden. Zur Bestimmung der Einträge  $\mathcal{D}_{2,4}$  und  $\mathcal{D}_{4,2}$  sind auch die Positionen von 2 und 4 relevant, vgl. Fall 2. Mit Hilfe von Tabelle 5 findet man die zur Berechnung der Einträge benötigten effektiven Quadranten.

Man muss nun noch die Einträge in  $\mathcal{D}$  für jedes unabhängige Element und die abhängigen Elementen der nachfolgenden Gruppen bestimmen. Das wurde für  $n = 3$  bereits ausgeführt. Die dazu analoge Vorgehensweise für diese Partition ist nachfolgend der Vollständigkeit halber beschrieben. Betroffen sind die Einträge  $\mathcal{D}_{1,3}$ ,  $\mathcal{D}_{3,1}$ ,  $\mathcal{D}_{1,5}$ ,  $\mathcal{D}_{5,1}$ ,  $\mathcal{D}_{2,5}$  und  $\mathcal{D}_{5,2}$ . Hierzu benötigt man die Quadranten und Positionen der unabhängigen Elemente sowie die Verdrehung  $k$  des abhängigen Elements. Der Quadrant des abhängigen Elements ist die Summe aus dem Quadranten des unabhängigen Elements in der Gruppe und dem  $k$ -Wert des abhängigen Elements. Man bildet die Differenz aus dem Quadranten des unabhängigen Elements und dem Quadranten des abhängigen Elements der Folgegruppe. Damit bestimmt man nach Formel (20) den Wert  $d_Q$ . Mit den Positionen der beiden unabhängigen Elemente und dem  $d_Q$ -Wert kann man aus Tabelle 5 den effektiven Quadranten bestimmen.

Schließlich bestimmt man noch die Einträge für jedes abhängige Element in einer Gruppe mit allen Elementen aller folgenden Gruppen. Für die betrachtete Partition entspricht dies den Einträgen  $\mathcal{D}_{3,4}$ ,  $\mathcal{D}_{4,3}$ ,  $\mathcal{D}_{3,5}$  und  $\mathcal{D}_{5,3}$ .

Hierzu benötigt man die Quadranten und Positionen der unabhängigen Elemente sowie die Verdrehungen  $k$  der abhängigen Elemente. Die Vorgehensweise ist auch hier wie zuvor beschrieben. Man bildet die Differenz  $d_Q$  aus den Quadranten der beteiligten Elemente und ermittelt mit Hilfe von Tabelle 5 den effektiven Quadranten.

Die bisher anhand der Betrachtung von Beispielen gewonnenen Erkenntnisse sind in Algorithmus 2 in Form eines Struktogramms zusammengefasst. Die einzelnen Unterprogramme sind mit dem Buchstaben  $U$  und die Schleifen mit dem Buchstaben  $L$ , jeweils ergänzt durch eine Nummer, gekennzeichnet.



**Algorithmus 2.** Bestimmung aller Vorzeichenmatrizen  $\mathcal{D}$  für die komplexen Vorzeichenvektoren der Ordnung  $n$  mit genau  $n$  von null verschiedenen Elementen

<p><b>U1:</b> Erstelle eine Liste aller <math>\sum_{k=1}^n \left\{ \begin{matrix} n \\ k \end{matrix} \right\}</math> Partitionen der Elemente aus <math>I_n := \{1, \dots, n\}</math>.</p>	
<p><b>L1:</b> Für jede Partition in der Liste</p>	
<p><b>U2:</b> Generiere alle <math>4^{k-1}</math> Kombinationen von Quadranten aus <math>Q := \{1, 2, 3, 4\}</math> für die <math>(k-1)</math> unabhängigen Elemente größer als 1 aus den <math>k</math> Gruppen der Partition.</p>	
<p><b>U3:</b> Generiere alle <math>(k-1)!</math> Sequenzen der Positionen der <math>(k-1)</math> unabhängigen Elemente der <math>k</math> Gruppen.</p>	
<p><b>U4:</b> Generiere alle <math>4^{n-k}</math> <math>(n-k)</math>-stelligen Kombinationen aus <math>\{0, 1, 2, 3\}</math> für die Verdrehungen der <math>(n-k)</math> abhängigen Elemente.</p>	
<p><b>L2:</b> Für jede Kombination der Quadranten</p>	
<p><b>U5:</b> Bestimme die Einträge <math>\mathcal{D}_{1,w}</math> und <math>\mathcal{D}_{w,1}</math> in den Zeilen und Spalten 1 für die unabhängigen Elemente <math>w</math> nach Tabelle 3.</p>	
<p><b>L3:</b> Für alle Sequenzen</p>	
<p><b>U6:</b> Bestimme die Einträge <math>\mathcal{D}_{v,w}</math> und <math>\mathcal{D}_{w,v}</math> für unabhängige Elemente <math>v, w &gt; 1</math> nach Tabelle 5.</p>	
<p><b>L4:</b> Für jede Kombination der Verdrehungen</p>	
<p><b>L5:</b> Für jede der <math>k</math> Gruppen in der Partition</p>	
<p><b>U7:</b> Bestimme die Einträge <math>\mathcal{D}_{v,w}</math> und <math>\mathcal{D}_{w,v}</math> für die abhängigen Elemente <math>v, w</math> nach Tabelle 4.</p>	
<p><b>L6:</b> Für alle <math>k</math> unabhängigen Elemente</p>	
<p><b>U8:</b> Generiere die Einträge <math>\mathcal{D}_{v,w}</math> und <math>\mathcal{D}_{w,v}</math> der unabhängigen Elemente <math>v</math> zu den abhängigen Elementen <math>w</math> der Folgegruppen nach Tabelle 5.</p>	
<p><b>L7:</b> Für alle <math>(n-k)</math> abhängigen Elemente</p>	
<p><b>U9:</b> Generiere die Einträge <math>\mathcal{D}_{v,w}</math> und <math>\mathcal{D}_{w,v}</math> der abhängigen Elemente <math>v</math> zu allen Elementen <math>w</math> in den Folgegruppen nach Tabelle 5.</p>	
<p><b>U10:</b> Schreibe die Vorzeichenmatrix <math>\mathcal{D}</math> in die Ergebnisliste.</p>	

Mit der im Struktogramm zusammengefassten Vorgehensweise kann man alle komplexen Vorzeichenvektoren der Ordnung  $k$  mit genau  $k$  von null verschiedenen Elementen bestimmen. Damit ist der Schritt 4 im Algorithmus 1 spezifiziert.

Basierend auf den Algorithmen 1 und 2 kann die Gleichung (17) zur Ermittlung der Anzahl der komplexen Vorzeichenvektoren hergeleitet werden. Hierzu betrachtet man die komplexen Vorzeichenvektoren der Ordnung  $l$  mit genau  $l$  Elementen ungleich null. Es gibt  $S(l, k)$  Partitionen der  $l$  von null verschiedenen Elemente in  $k$  Gruppen.

Für jede Partition mit  $k$  Gruppen gibt es  $(k-1)!$

Sequenzen der Positionen sowie  $4^{k-1}$  Kombinationen der Quadranten der unabhängigen Elemente. Weiterhin gibt es  $4^{l-k}$  Verdrehungskombinationen der  $(l-k)$  abhängigen Elemente. Die Partitionierung kann in  $k = 1, \dots, l$  Gruppen vorgenommen werden.

Die Anzahl der komplexen Vorzeichenvektoren der Ordnung  $l$  mit genau  $l$  von null verschiedenen Elementen ist somit

$$\kappa_l(l) := \sum_{k=1}^l \left\{ \begin{matrix} l \\ k \end{matrix} \right\} (k-1)! 4^{(k-1)} 4^{(l-k)} \quad (21)$$

und nach einer Zusammenfassung der Terme

$$\kappa_l(l) = 4^{(l-1)} \sum_{k=1}^l (k-1)! \binom{l}{k}. \quad (22)$$

Für die  $\kappa_l$  auf diese Weise bestimmten komplexen Vorzeichenvektoren der Ordnung  $l$  gibt es jeweils  $\binom{n}{l}$  Möglichkeiten, einen komplexen Vorzeichenvektor der Ordnung  $n$  mit  $l$  Elementen ungleich null zu bilden. Hinzu kommt der Nullvektor. Damit ergibt sich für die Anzahl der komplexen Vorzeichenvektoren

$$\kappa(n) = 1 + \sum_{l=1}^n \binom{n}{l} \kappa_l(l). \quad (23)$$

Setzt man Gleichung (22) in (23) ein, kommt man auf die in Abschnitt 4 angegebene Formel (17).

### 5.4 Eine weitere Darstellungsform

Mit den Algorithmen 1 und 2 kann man alle komplexen Vorzeichenvektoren der Ordnung  $n$  in der Darstellung mit der Vorzeichenmatrix  $\mathcal{D}$  bestimmen. Es besteht auch die Möglichkeit, lediglich die zur Konstruktion der Vorzeichenmatrix  $\mathcal{D}$  herangezogenen Daten in geeigneter Weise zu notieren. Man kann aus diesen Daten bei Bedarf die Matrizen  $\mathcal{D}$  und  $\mathcal{P}$  bzw.  $\mathcal{Q}$  erzeugen. Damit ist eine weitere Darstellungsform für komplexe Vorzeichenvektoren gegeben.

Wesentlich für die Konstruktion der Vorzeichenmatrizen sind die durch die Partitionen dargestellten Abhängigkeiten (siehe  $U1$  in Algorithmus 2), die Quadranten der unabhängigen Elemente ( $U2$ ), die Sequenz der Positionen der unabhängigen Elemente ( $U3$ ) sowie die Verdrehungen der abhängigen Elemente relativ zu ihrem Bezugselement ( $U4$ ).

In [9, S. 82] wurde eine auf diesen Daten beruhende Darstellungsform formal definiert. Diese alternative Darstellungsform soll anhand eines Beispiels aufbauend auf die Ausführungen der vorigen Unterabschnitte erläutert werden.

Es soll für  $n = 4$  die Partition  $\{\{1, 3\}, \{2, 4\}\}$  betrachtet werden. Die Elemente 1 und 2 sind in dieser Partition unabhängig. Das Argument von Element 2 soll bei Normierung auf Element 1 im ersten Quadranten liegen. Das Argument des abhängigen Elements 3 soll relativ zum Argument seines Bezugselements 1 um  $\frac{\pi}{2}$  verdreht sein. Weiterhin

soll das Argument des abhängigen Elements 4 relativ zum Argument seines Bezugselements 2 um  $\pi$  verdreht sein.

Wegen der auf Element 1 vorgenommenen Normierung und der Unabhängigkeit von 2 gilt für die Positionen dieser beiden Elemente  $\text{pos}(2) > \text{pos}(1)$ .

In Abbildung 6 sind die Zusammenhänge für dieses Beispiel in Form eines gerichteten Graphen dargestellt. Die Knoten im Graphen entsprechen den von null verschiedenen Elementen des komplexen Vektors. Die vom Knoten  $i$  ausgehenden Kanten sind mit  $R_i$  beschriftet. Diese Kantengewichte bezeichnen Rotationsmatrizen zur Beschreibung der Verhältnisse der Argumente der Elemente.

Die unabhängigen Knoten 1 und 2 und die abhängigen Knoten 3 und 4 sind in unterschiedlichen Farben dargestellt.

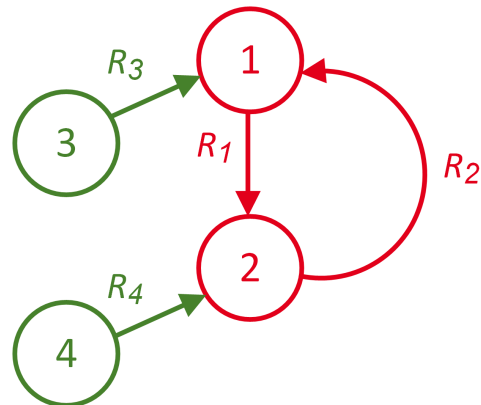


Abbildung 6: Graph zur Darstellung der Situation des Beispiels

Die von den unabhängigen Knoten ausgehenden Kanten bilden im Graphen einen gerichteten Zyklus. Die Knoten sind im Zyklus beginnend mit Knoten 1 nach aufsteigenden Positionen angeordnet. Vom letzten Knoten führt eine Kante zum Knoten 1, um den Zyklus zu schließen. Die abhängigen Knoten haben eine jeweils zu ihrem Bezugsknoten gerichtete Kante.

Die Kantengewichte  $R_3$  und  $R_4$  geben an, wie die Argumente der Bezugsknoten relativ zu den Argumenten der abhängigen Knoten verdreht sind. Der Bezugsknoten von 3 ist Knoten 1. Im Beispiel ist 3 gegen 1 um  $\frac{\pi}{2}$  verdreht, was einer Verdrehung von 1 relativ zu 3 um  $-\frac{\pi}{2}$  entspricht. Daher ist  $R_3 = R(3)$ , vgl. Abschnitt 3.1. Der Bezugsknoten von 4 ist Knoten 2. Im Beispiel ist 4 gegen 2

um  $\pi$  verdreht, was einer Verdrehung von 2 relativ zu 4 um  $-\pi$  entspricht. Daher ist  $R_4 = R(2)$ .

Im Zyklus der unabhängigen Knoten geben die Kantengewichte an, in welchem Quadranten der Nachfolgerknoten bezüglich seines Vorgängers liegt. Im Beispiel ist Knoten 2 der Nachfolger von Knoten 1 und Knoten 1 der Nachfolger von Knoten 2. Das Argument von Knoten 2 liegt relativ zu Knoten 1 im ersten Quadranten. Das wird durch  $R_1 = R(0)$  beschrieben. Wegen  $\text{pos}(2) > \text{pos}(1)$  liegt die Differenz  $\phi_2 - \phi_1$  im ersten Quadranten. Bezüglich Knoten 2 ist sein Nachfolger 1 daher im vierten Quadranten, was durch  $R_2 = R(3)$  ausgedrückt wird.

Graph und Kantengewichte beinhalten alle Informationen zur Bestimmung des komplexen Vorzeichenvektors. Die Informationen im Graph können gleichwertig mit Hilfe einer Matrix  $\nu$  dargestellt werden, vgl. [9, S. 89 ff.]. Diese Matrix soll im Folgenden als Konstruktionsdatenmatrix  $\nu$  bezeichnet werden. Die Matrix  $\nu$  besteht in unserem Beispiel aus zwei Zeilen und vier Spalten. Im allgemeinen Fall besteht die Konstruktionsdatenmatrix aus zwei Zeilen und  $n$  Spalten. Mit  $\nu(i, j)$  bezeichnet man den Eintrag in der Zeile  $i$  und der Spalte  $j$  der Konstruktionsdatenmatrix.

Die Spalte  $k$  der Matrix beschreibt die Situation von Knoten  $k$ . Ist  $k$  ein unabhängiger Knoten, so steht in  $\nu(1, k)$  der Nachfolger-Knoten von  $k$  im Zyklus<sup>3</sup>. In  $\nu(2, k)$  steht die Nummer der Rotationsmatrix, die den Quadranten des Nachfolger-Knotens relativ zu  $k$  bestimmt. Ist  $k$  ein abhängiger Knoten, so steht in  $\nu(1, k)$  die Nummer seines Bezugsknotens, also des Knotens, von dem  $k$  abhängig ist. In  $\nu(2, k)$  steht die Nummer der Rotationsmatrix zur Beschreibung der Verdrehung des Bezugsknotens  $\nu(1, k)$  relativ zum abhängigen Knoten  $k$ . Ist die Komponente  $k$  eines Vektors gleich null, so ist die entsprechende Spalte  $k$  der Konstruktionsdatenmatrix eine Nullspalte. Somit stehen in der ersten Zeile der Konstruktionsdatenmatrix nur die Indizes der unabhängigen Knoten oder die Null.

Für unser Beispiel hat die Konstruktionsdatenmatrix den folgenden Aufbau:

$$\nu = \begin{pmatrix} 2 & 1 & 1 & 2 \\ 0 & 3 & 3 & 2 \end{pmatrix} \quad (24)$$

Die einzelnen Rotationsmatrizen sind:

<sup>3</sup> Der Spezialfall, dass nur ein einziger unabhängiger Knoten vorhanden ist, wird am Ende dieses Abschnitts betrachtet.

$$R_1 = R(0); R_2 = R(3); R_3 = R(3); R_4 = R(2)$$

Nachfolgend werden die Einträge in der Matrix  $\nu$  als Ergebnis der Vorgaben im Beispiel erläutert. Da die Konstruktionsdatenmatrix lediglich eine algebraische Repräsentation des ausführlich erläuterten Graphen darstellt, sind die Erläuterungen an dieser Stelle etwas knapper ausgeführt.

Spalte 1 beschreibt die Situation für den Knoten 1. Der Nachfolger von Knoten 1 ist Knoten 2, daher ist  $\nu(1, 1) = 2$ . Gemäß der Vorgabe im Beispiel liegt das Argument von 2 relativ zum Argument von 1 im ersten Quadranten. Daher ist die Rotationsmatrix  $R_1 = R(0)$  und der Eintrag  $\nu(2, 1) = 0$ .

Spalte 2 beschreibt die Situation für den Knoten 2. Der Nachfolger von 2 ist in diesem Beispiel gleich dem Vorgänger 1, wie es im Graph auch zu sehen ist. Somit ist  $\nu(1, 2) = 1$ . Relativ zu 2 liegt wegen  $\text{pos}(2) > \text{pos}(1)$  der Nachfolger 1 im vierten Quadranten, daher ist die Rotationsmatrix  $R_2 = R(3)$  und der Eintrag  $\nu(2, 2) = 3$ .

Spalte 3 beschreibt die Situation für den abhängigen Knoten 3. Dieser hängt von Knoten 1 ab, daher ist der Eintrag  $\nu(1, 3) = 1$ . Der Bezugsknoten 1 ist gegen 3 um  $-\frac{\pi}{2}$  verdreht, daher ist die Rotationsmatrix  $R_3 = R(3)$  und der Eintrag  $\nu(2, 3) = 3$ .

Spalte 4 beschreibt die Situation für den von 2 abhängigen Knoten 4. Somit ist der Eintrag  $\nu(1, 4) = 2$ . Der Bezugsknoten 2 ist gegen 4 um  $-\pi$  verdreht, was durch die Rotationsmatrix  $R_4 = R(2)$  beschrieben wird. Damit ist der Eintrag  $\nu(2, 4) = 2$ .

Nun wird ausführlich gezeigt, wie man aus dieser Konstruktionsdatenmatrix  $\nu$  die zugehörige Vorzeichenmatrix  $\mathfrak{D}$  erzeugen kann. Mit Hilfe der Einträge in der Spalte  $k$  der Konstruktionsdatenmatrix kann man die Vorzeichen  $(\mathcal{P}_k)_{\nu(1,k)}$  und  $(\mathcal{D}_k)_{\nu(1,k)}$  ermitteln. Die  $k$ -te Spalte der Konstruktionsdatenmatrix liefert die  $\nu(1, k)$ -ten Einträge der  $k$ -ten charakteristischen Vorzeichenmuster  $(\mathcal{D}_k, \mathcal{P}_k)$  und damit die entsprechenden Einträge in den Matrizen  $\mathcal{D}$  und  $\mathcal{P}$ :

$$\begin{pmatrix} (\mathcal{P}_k)_{\nu(1,k)} \\ (\mathcal{D}_k)_{\nu(1,k)} \end{pmatrix} = \begin{pmatrix} \mathcal{P}_{\nu(1,k),k} \\ \mathcal{D}_{\nu(1,k),k} \end{pmatrix} \quad (25)$$

Sind die beteiligten Knoten unabhängig, so ergeben sich die Vorzeichen mit Hilfe der Beziehung

$$\begin{pmatrix} (\mathcal{P}_k)_{\nu(1,k)} \\ (\mathcal{D}_k)_{\nu(1,k)} \end{pmatrix} = R(\nu(2, k)) \begin{pmatrix} \mathcal{P} \\ \mathcal{D} \end{pmatrix}. \quad (26)$$

Sind die beteiligten Knoten abhängig, so erhält man die Vorzeichen mit Hilfe der Beziehung

$$\begin{pmatrix} (\mathcal{P}_k)_{\nu(1,k)} \\ (\mathcal{D}_k)_{\nu(1,k)} \end{pmatrix} = R(\nu(2,k)) \begin{pmatrix} z \\ p \end{pmatrix}. \quad (27)$$

Die erste Spalte der Konstruktionsdatenmatrix aus (24) liefert den zweiten Eintrag der ersten charakteristischen Vorzeichenmuster:

$$\begin{pmatrix} (\mathcal{P}_1)_2 \\ (\mathcal{D}_1)_2 \end{pmatrix} = R(0) \begin{pmatrix} p \\ p \end{pmatrix} = \begin{pmatrix} p & z \\ z & p \end{pmatrix} \begin{pmatrix} p \\ p \end{pmatrix} = \begin{pmatrix} p \\ p \end{pmatrix}$$

Wegen

$$\begin{pmatrix} (\mathcal{P}_1)_2 \\ (\mathcal{D}_1)_2 \end{pmatrix} = \begin{pmatrix} \mathcal{P}_{2,1} \\ \mathcal{D}_{2,1} \end{pmatrix} = \begin{pmatrix} \mathcal{D}_{2,1} \\ \mathcal{D}_{1,2} \end{pmatrix}$$

gilt also

$$\begin{pmatrix} \mathcal{D}_{2,1} \\ \mathcal{D}_{1,2} \end{pmatrix} = \begin{pmatrix} p \\ p \end{pmatrix}.$$

Eine Auswertung der zweiten Spalte führt zum gleichen Ergebnis und soll hier nicht vorgenommen werden.

Durch Auswertung der dritten Spalte erhält man den ersten Eintrag im dritten charakteristischen Vorzeichenmuster. Da es sich hier um abhängige Knoten handelt, ist das Ausgangsvorzeichenmuster für die Anwendung der Rotationsmatrix  $(z, p)^T$ , siehe Gleichung (27).

$$\begin{pmatrix} (\mathcal{P}_3)_1 \\ (\mathcal{D}_3)_1 \end{pmatrix} = R(3) \begin{pmatrix} z \\ p \end{pmatrix} = \begin{pmatrix} z & n \\ p & z \end{pmatrix} \begin{pmatrix} z \\ p \end{pmatrix} = \begin{pmatrix} n \\ z \end{pmatrix}$$

Wegen

$$\begin{pmatrix} (\mathcal{P}_3)_1 \\ (\mathcal{D}_3)_1 \end{pmatrix} = \begin{pmatrix} \mathcal{P}_{1,3} \\ \mathcal{D}_{1,3} \end{pmatrix} = \begin{pmatrix} -\mathcal{D}_{3,1} \\ \mathcal{D}_{1,3} \end{pmatrix}$$

gilt also

$$\begin{pmatrix} \mathcal{D}_{3,1} \\ \mathcal{D}_{1,3} \end{pmatrix} = \begin{pmatrix} p \\ z \end{pmatrix}.$$

Die Auswertung der vierten Spalte für den abhängigen Knoten 4 ergibt:

$$\begin{pmatrix} (\mathcal{P}_4)_2 \\ (\mathcal{D}_4)_2 \end{pmatrix} = R(2) \begin{pmatrix} z \\ p \end{pmatrix} = \begin{pmatrix} n & z \\ z & n \end{pmatrix} \begin{pmatrix} z \\ p \end{pmatrix} = \begin{pmatrix} z \\ n \end{pmatrix}$$

und damit

$$\begin{pmatrix} (\mathcal{P}_4)_2 \\ (\mathcal{D}_4)_2 \end{pmatrix} = \begin{pmatrix} \mathcal{P}_{2,4} \\ \mathcal{D}_{2,4} \end{pmatrix} = \begin{pmatrix} -\mathcal{D}_{4,2} \\ \mathcal{D}_{2,4} \end{pmatrix} = \begin{pmatrix} z \\ n \end{pmatrix}.$$

Alle Spalten sind nun ausgewertet. Zur Ermittlung weiterer Einträge in den Matrizen  $\mathcal{D}$  und  $\mathcal{P}$  und damit in der Matrix  $\mathfrak{D}$  ist es hilfreich, den zugehörigen Graphen in Abbildung 6 heranzuziehen. Es müssen noch die Einträge in  $\mathfrak{D}$  für Knoten ermittelt werden, die im Graph nicht direkt benachbart sind. Zur Ermittlung der Einträge  $\mathcal{D}_{i,j}$  und  $\mathcal{P}_{i,j}$  betrachtet man den Pfad von Knoten  $j$  zu Knoten  $i$  im Graph. Für die Auswertung wertet man die zum Pfad gehörigen Kanten aus und berücksichtigt, ob die Knoten abhängig zueinander sind oder nicht.

Beginnt man im Graph mit dem Knoten 3, so kommt man über die Kanten mit  $R_3$  und  $R_1$  zum Knoten 2. Man dreht somit mit  $R_3 = R(3)$  um  $3\frac{\pi}{2}$  und mit  $R_1 = R(0)$  um  $0\frac{\pi}{2}$ . Insgesamt dreht man also um  $3\frac{\pi}{2}$ , was durch die Anwendung der Rotationsmatrix  $R(3)$  umgesetzt wird. Da Knoten 3 keine Abhängigkeit zu Knoten 2 hat, muss man hier wie in Gleichung (26) die Rotationsmatrix auf das Vorzeichenmuster  $(p, p)^T$  anwenden. Es ergibt sich

$$\begin{pmatrix} (\mathcal{P}_3)_2 \\ (\mathcal{D}_3)_2 \end{pmatrix} = R(3) \begin{pmatrix} p \\ p \end{pmatrix} = \begin{pmatrix} z & n \\ p & z \end{pmatrix} \begin{pmatrix} p \\ p \end{pmatrix} = \begin{pmatrix} n \\ p \end{pmatrix}$$

und damit

$$\begin{pmatrix} (\mathcal{P}_3)_2 \\ (\mathcal{D}_3)_2 \end{pmatrix} = \begin{pmatrix} \mathcal{P}_{2,3} \\ \mathcal{D}_{2,3} \end{pmatrix} = \begin{pmatrix} -\mathcal{D}_{3,2} \\ \mathcal{D}_{2,3} \end{pmatrix} = \begin{pmatrix} n \\ p \end{pmatrix}.$$

Beginnt man im Graph in gleicher Weise mit dem Knoten 4, so kommt man über die Kanten mit  $R_4$  und  $R_2$  zum Knoten 1. Man dreht somit mit  $R_4 = R(2)$  um  $2\frac{\pi}{2}$  und mit  $R_2 = R(3)$  um  $3\frac{\pi}{2}$ , also insgesamt um  $5\frac{\pi}{2}$ , was einer Drehung mit der Rotationsmatrix  $R(1)$  entspricht. Es ergibt sich

$$\begin{pmatrix} (\mathcal{P}_4)_1 \\ (\mathcal{D}_4)_1 \end{pmatrix} = R(1) \begin{pmatrix} p \\ p \end{pmatrix} = \begin{pmatrix} z & p \\ n & z \end{pmatrix} \begin{pmatrix} p \\ p \end{pmatrix} = \begin{pmatrix} p \\ n \end{pmatrix}$$

und damit

$$\begin{pmatrix} (\mathcal{P}_4)_1 \\ (\mathcal{D}_4)_1 \end{pmatrix} = \begin{pmatrix} \mathcal{P}_{1,4} \\ \mathcal{D}_{1,4} \end{pmatrix} = \begin{pmatrix} -\mathcal{D}_{4,1} \\ \mathcal{D}_{1,4} \end{pmatrix} = \begin{pmatrix} p \\ n \end{pmatrix}.$$

Die Knoten 3 und 4 sind von unterschiedlichen Knoten abhängig und weisen daher zueinander keine Abhängigkeit auf. Für die Ermittlung der Einträge bezüglich dieser beiden Knoten beginnt man beispielsweise im Knoten 3. Mit  $R_3 = R(3)$  dreht man um  $3\frac{\pi}{2}$ , dann mit  $R_1 = R(0)$  um  $0\frac{\pi}{2}$ .  $R_4$  entspricht einer Drehung um  $2\frac{\pi}{2}$ , allerdings wird die Kante entgegen ihrer Richtung durchlaufen. Somit entspricht dies auf dem Weg von 3 nach 4 einer Drehung um  $-2\frac{\pi}{2}$ . Die Gesamtdrehung ist somit

$(3 + 0 - 2) \frac{\pi}{2} = \frac{\pi}{2}$ , was der Matrix  $R(1)$  entspricht. Es ergibt sich:

$$\begin{pmatrix} (\mathcal{P}_3)_4 \\ (\mathcal{D}_3)_4 \end{pmatrix} = R(1) \begin{pmatrix} p \\ p \end{pmatrix} = \begin{pmatrix} z & p \\ n & z \end{pmatrix} \begin{pmatrix} p \\ p \end{pmatrix} = \begin{pmatrix} p \\ n \end{pmatrix}$$

und damit

$$\begin{pmatrix} (\mathcal{P}_3)_4 \\ (\mathcal{D}_3)_4 \end{pmatrix} = \begin{pmatrix} \mathcal{P}_{4,3} \\ \mathcal{D}_{4,3} \end{pmatrix} = \begin{pmatrix} \mathfrak{D}_{4,3} \\ \mathfrak{D}_{3,4} \end{pmatrix} = \begin{pmatrix} p \\ n \end{pmatrix}.$$

Die Darstellung des Vorzeichenvektors mit der Vorzeichenmatrix  $\mathfrak{D}$  ist somit berechnet:

$$\mathfrak{D} = \begin{pmatrix} p & p & z & n \\ p & p & p & n \\ p & p & p & n \\ n & z & p & p \end{pmatrix}$$

Für den allgemeinen Fall ist die Situation in Abbildung 7 dargestellt. Es sind  $k$  unabhängige Knoten  $U_i$  dargestellt, die einen Zyklus bilden. Weiterhin sind exemplarisch einige abhängige Knoten eingezeichnet. Ein unabhängiger Knoten muss keine abhängigen, kann aber auch mehrere abhängige Knoten haben.

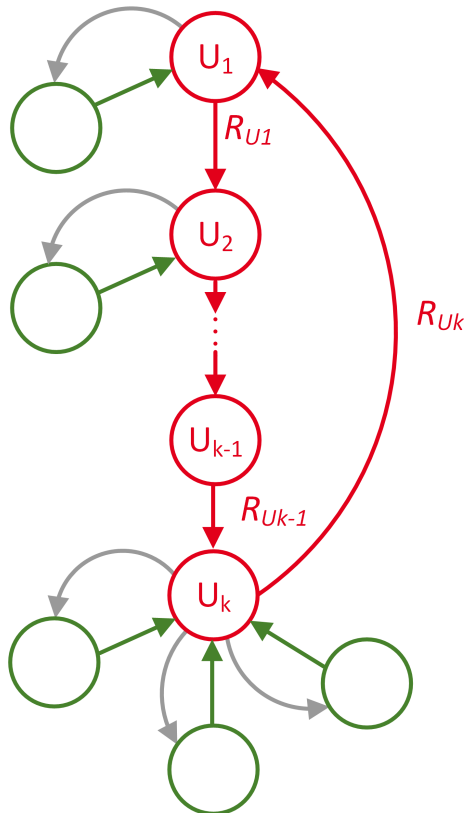


Abbildung 7: Erweiterter Graph zur Darstellung der allgemeinen Situation

Der Graph ist im Vergleich zum Graphen des Beispiels in Abbildung 6 insofern erweitert worden, dass auch Kanten vom unabhängigen Knoten zu seinen abhängigen eingezeichnet sind. Das Kantengewicht dieser hinzugekommenen Kanten ist nicht in den Daten der Konstruktionsdatenmatrix enthalten. Es kann aber mit der folgenden Beziehung einfach ermittelt werden: Ist  $\rho$  die Referenznummer der Rotationsmatrix des abhängigen Knotens  $k$ , gilt also  $R_k = R(\rho)$ , so ist die Referenznummer  $\rho_i$  der Rotationsmatrix für die entgegengesetzte Kante

$$\rho_i = (-\rho) \bmod 4. \tag{28}$$

In den Graph zeichnet man der Übersichtlichkeit halber lediglich die Knoten ein, die für Nicht-Null-Einträge stehen. Die Graphen haben somit maximal  $n$  Knoten. Sind  $k$  unabhängige Knoten vorhanden, dann umfasst der Zyklus diese  $k$  Knoten. Hinzu kommen noch maximal  $(n - k)$  abhängige Knoten.

Es soll nun der Spezialfall betrachtet werden, dass nur ein einziger unabhängiger Knoten vorhanden ist. Eine mit dem allgemeinen Fall konsistente Definition ist, dass in einem solchen Fall der Knoten selber als sein Nachfolger gelten soll. Der Zyklus im Graphen besteht nur aus dem unabhängigen Knoten und einer Selbstschleife.

Ist  $k$  dieser einzige unabhängige Knoten, so ist mit der getroffenen Festlegung der Eintrag in der Konstruktionsdatenmatrix in der ersten Zeile der Spalte  $k$  gleich  $k$ . Es fehlt allerdings ein Kriterium, um die Rotationsmatrix  $R_k$  festlegen zu können. In diesem Zusammenhang ist die folgende Notiz interessant:

**Notiz 5.2.** Sind mindestens zwei unabhängige Knoten vorhanden, so ist die Summe modulo 4 der Referenzzahlen der Rotationsmatrizen aller unabhängigen Knoten immer 3.

**Beweis:** O.B.d.A. kann angenommen werden, dass die nach zunehmender Position geordnete Sequenz der unabhängigen Knoten  $1, 2, \dots, k$  ist. Dies vereinfacht die Schreibweise der Summanden in den folgenden Beweisschritten. Es gilt dann  $\text{pos}(i) < \text{pos}(j)$  für  $i < j$ . Die Rotationsmatrix des Knotens  $j$  gibt an, in welchem Quadranten der Nachfolger  $\nu(1, j)$  in Bezug auf  $j$  liegt. Bezeichnet man mit  $Q(j)$  den Quadranten von Knoten  $j$ , so ist  $\rho_j = (Q(\nu(1, j)) - Q(j)) \bmod 4$ .

Es ist zu beweisen, dass bei einem Zyklus mit  $k$



unabhängigen Knoten gilt:

$$S_k := \sum_{j=1}^k \rho_j \bmod 4 = 3 \quad (29)$$

Für die ersten  $(k - 1)$  unabhängigen Knoten im Zyklus gilt:

$$\sum_{j=1}^{k-1} \rho_j \bmod 4 = \sum_{j=1}^{k-1} (Q(\nu(1, j)) - Q(j)) \bmod 4 \quad (30)$$

Da für  $k > 2$  der Nachfolger von Knoten  $j$  der Vorgänger von Knoten  $j + 2$  ist, heben sich in (30) alle Summanden bis auf zwei auf und es gilt generell für  $k > 1$ :

$$\sum_{j=1}^{k-1} \rho_j \bmod 4 = (Q(k) - Q(1)) \bmod 4 \quad (31)$$

In der Gesamtsumme fehlt noch die Referenzzahl  $\rho_k$  für die Rotationsmatrix  $R_k$ . Für die Bestimmung von  $\rho_k$  ist zu berücksichtigen, dass anders als in den ersten  $(k - 1)$  Fällen die Position des Nachfolgerknotens kleiner als die des Vorgängers ist. Der Nachfolger von Knoten  $k$  ist  $\nu(1, k) = 1$ . Weil Knoten 1 nach unserer Annahme der erste in der Sequenz ist, gilt  $\text{pos}(1) < \text{pos}(k)$ .

Ist  $v = \nu(1, j)$  der Nachfolger und  $w = j$  der Vorgänger, so erkennt man in Tabelle 5, dass für  $\text{pos}(\nu(1, j)) > \text{pos}(j)$  die Differenz der Quadranten  $d_Q$  gerade der Referenzzahl der Rotationsmatrix entspricht<sup>4</sup>.

Im Falle  $\text{pos}(\nu(1, j)) < \text{pos}(j)$  ist hingegen  $\rho_j = (d_Q - 1) \bmod 4$ . Daher gilt:

$$\rho_k = (Q(1) - Q(k) - 1) \bmod 4 \quad (32)$$

Mit (31) ergibt sich somit für die Summe der Referenzzahlen der Rotationsmatrizen aller unabhängigen Knoten

$$\begin{aligned} S_k &= \sum_{j=1}^k \rho_j \bmod 4 \\ &= (Q(k) - Q(1) + Q(1) - Q(k) - 1) \bmod 4 \\ &= -1 \bmod 4 = 3, \end{aligned} \quad (33)$$

wie zu beweisen war.

Es ist daher naheliegend, auch für den Fall eines einzigen unabhängigen Knotens  $k$  für die Referenzzahl der Rotationsmatrix  $\rho_k = 3$  zu wählen.

<sup>4</sup> In Tabelle 5 werden die effektiven Quadranten angegeben. Der effektive Quadrant  $Q$  kann mit der Referenzzahl  $\rho = Q - 1$  beschrieben werden, siehe Abschnitt 3.1.

## 5.5 Algorithmus zur Erzeugung der Konstruktionsdatenmatrix

Der Algorithmus 2 beschreibt die Erzeugung aller Vorzeichenmatrizen  $\mathfrak{D}$  für komplexe Vorzeichenvektoren der Ordnung  $n$  mit genau  $n$  von null verschiedenen Komponenten. Mit diesem Algorithmus werden die  $n^2$  Einträge von  $\mathfrak{D}$  ermittelt. Naturgemäß sieht der Algorithmus zur Erzeugung der Konstruktionsdatenmatrix mit lediglich  $2n$  Einträgen einfacher aus. Die Vorgehensweise hierfür wird durch den auf der nächsten Seite dargestellten Algorithmus 3 beschrieben.

Durch Vergleich der beiden Struktogramme sind die Parallelen in beiden Algorithmen leicht zu erkennen. So sind die Unterprogramme U1 bis U4, welche die unterschiedlichen Fälle generieren, in beiden Algorithmen dieselben. Auch bei den Schleifen findet man Übereinstimmungen. In beiden Algorithmen treten Iterationen über alle Partitionen, über alle Kombinationen der Quadranten der unabhängigen Elemente, über alle Sequenzen der unabhängigen Elemente und über alle Verdrehungs-Kombinationen der abhängigen Elemente auf. Lediglich die Nummerierung der Schleifen unterscheidet sich.

Die Konstruktionsdatenmatrizen der komplexen Vorzeichenvektoren der Ordnung  $k \leq n$  mit genau  $k$  von null verschiedenen Elementen bilden die Basis für die Erstellung der Konstruktionsdatenmatrizen der Ordnung  $n$ .

Um von einer Konstruktionsdatenmatrix der Ordnung  $k$  auf eine der Ordnung  $n$  zu kommen, wird für jedes Null-Element  $j$  des Vektors  $q$  die Spalte  $j$  in der Konstruktionsdatenmatrix der Ordnung  $n$  zu einer Null-Spalte. Ausgehend von den Matrizen mit  $k$  Spalten müssen deswegen die Einträge in der ersten Zeile passend umnummeriert werden.

Das folgende Beispiel verdeutlicht die Vorgehensweise. Die Konstruktionsdatenmatrix  $\nu_{I3}$  mit  $k = 3$  Elementen ungleich null soll als Basis für einen komplexen Vorzeichenvektor der Ordnung  $n = 5$  verwendet werden.

$$\nu_{I3} = \begin{pmatrix} 2 & 1 & 1 \\ 3 & 0 & 2 \end{pmatrix}$$

Es gibt  $\binom{5}{2} = 10$  Möglichkeiten, die beiden Null-Komponenten im Vektor der Ordnung  $n = 5$  zu platzieren. Wählt man für das Beispiel die erste und vierte Komponente gleich null, so erhält man die folgende aus  $\nu_{I3}$  erzeugte Konstruktionsdatenmatrix:



**Algorithmus 3.** Bestimmung aller Konstruktionsdatenmatrizen  $\nu$  für die komplexen Vorzeichenvektoren der Ordnung  $n$  mit genau  $n$  von null verschiedenen Elementen

<p><b>U1:</b> Erstelle eine Liste aller <math>\sum_{k=1}^n \binom{n}{k}</math> Partitionen der Elemente aus <math>I_n := \{1, \dots, n\}</math>.</p>	
<p><b>L1:</b> Für jede Partition in der Liste</p>	
<p><b>U2:</b> Generiere alle <math>4^{k-1}</math> Kombinationen von Quadranten aus <math>Q := \{1, 2, 3, 4\}</math> für die <math>(k-1)</math> unabhängigen Elemente größer als 1 aus den <math>k</math> Gruppen der Partition.</p>	
<p><b>U3:</b> Generiere alle <math>(k-1)!</math> Sequenzen der Positionen der <math>(k-1)</math> unabhängigen Elemente der <math>k</math> Gruppen.</p>	
<p><b>U4:</b> Generiere alle <math>4^{n-k} (n-k)</math>-stelligen Kombinationen aus <math>\{0, 1, 2, 3\}</math> für die Verdrehungen der <math>(n-k)</math> abhängigen Elemente.</p>	
<p><b>L2:</b> Für jede Kombination der Quadranten</p>	
<p><b>L3:</b> Für alle Sequenzen</p>	
<p><b>L4:</b> Für alle <math>k</math> unabhängigen Elemente</p>	
<p><b>U5:</b> Trage den Nachfolger in Zeile 1 und die Referenzzahl der Rotationsmatrix in Zeile 2 der Spalte des unabhängigen Elements ein.</p>	
<p><b>L5:</b> Für jede Kombination der Verdrehungen</p>	
<p><b>L6:</b> Für alle <math>(n-k)</math> abhängigen Elemente</p>	
<p><b>U6:</b> Trage das unabhängige Element in Zeile 1 und die Rotationsmatrix in Zeile 2 der Spalte des abhängigen Elements ein.</p>	
<p><b>U7:</b> Schreibe die Konstruktionsdatenmatrix <math>\nu</math> in die Liste.</p>	

$$\nu_{I5} = \begin{pmatrix} 0 & 3 & 2 & 0 & 2 \\ 0 & 3 & 0 & 0 & 2 \end{pmatrix}$$

Aus Knoten 1 für  $n = 3$  wird Knoten 2 für  $n = 5$ , aus Knoten 2 wird Knoten 3 und aus Knoten 3 wird Knoten 5. Die zueinander gehörigen Spalten der Matrizen  $\nu_{I3}$  und  $\nu_{I5}$  sind zur Verdeutlichung in gleichen Farben abgedruckt. Die beiden Null-Knoten 1 und 4 für  $n = 5$  werden in  $\nu_{I5}$  jeweils durch eine Null-Spalte repräsentiert.

Zur Ermittlung aller Konstruktionsdatenmatrizen für komplexe Vorzeichenvektoren der Ordnung  $n$  kann Algorithmus 1 in naheliegender Weise modifiziert werden. In den Schritten 4 und 5 erzeugt man an Stelle der Vorzeichenmatrizen  $\mathcal{D}$  die entsprechenden Konstruktionsdatenmatrizen  $\nu$ .

Die in Abschnitt 8 beschriebene Funktion zum Generieren aller komplexen Vorzeichenvektoren nutzt die weitgehend parallelen Schritte aus und erzeugt in einem Durchlauf sowohl die Vorzeichenmatrizen  $\mathcal{D}$  als auch die Konstruktionsdatenmatrizen  $\nu$ .

## 6 Prüfung auf Vorzeichen-Steuerbarkeit

Anhand von Abbildung 1 kann man erkennen, dass die vorzeichensteuerbaren Systeme eine Teilklasse der strukturell steuerbaren Systeme bilden. Die streng strukturell steuerbaren Systeme bilden eine Teilklasse der vorzeichensteuerbaren Systeme.

In [9, S. 100] ist ein Algorithmus mit der Bezeichnung *Algorithm 2* zum Test auf Vorzeichen-Steuerbarkeit angegeben. Die Überprüfung eines Systems auf strenge strukturelle Steuerbarkeit kann sehr effizient ausgeführt werden, siehe [18]. Daher wird in *Algorithm 2* zunächst überprüft, ob das System streng strukturell steuerbar ist. Ist dies der Fall, so ist es auch vorzeichensteuerbar. Für nicht streng strukturell steuerbare Systeme wird auf strukturelle Steuerbarkeit geprüft, da diese eine notwendige Bedingung für Vorzeichen-Steuerbarkeit ist. Ist das System strukturell steuerbar, dann müssen für jeden komplexen Vorzeichenvektor der Ordnung  $n$  einige Bedingungen

überprüft werden, die in einem weiteren Algorithmus mit der Bezeichnung *Algorithm 1* formuliert sind [9, S. 98]. Hierzu benötigt man eine Liste aller komplexen Vorzeichenvektoren, wie sie mit den Algorithmen aus Abschnitt 5 erzeugt werden können.

Aufgrund der bereits für niedrige Ordnungen hohen Zahl von komplexen Vorzeichenvektoren ist diese Überprüfung in der Praxis nur für Systeme niedriger Ordnung effizient durchführbar. Das Entscheidungsproblem, ob ein System vorzeichensteuerbar ist, gehört zu den NP-vollständigen Problemen, vgl. [9, S. 95].

In [9, S. 99] wird in Satz 5.8 gezeigt, dass man in bestimmten Fällen ein System der Ordnung  $(n-1)$  an Stelle des ursprünglichen Systems der Ordnung  $n$  auf Vorzeichen-Steuerebarkeit untersuchen kann. *Algorithm 2* ist daher rekursiv angelegt und führt die Prüfung eines Systems der Ordnung  $n$  so oft wie möglich auf ein System der Ordnung  $(n-1)$  zurück, auf das der Algorithmus dann angewandt wird.

## 7 Prüfung auf Vorzeichen-Stabilisierbarkeit

Anhand von Abbildung 1 kann man erkennen, dass vorzeichen-steuerbare Systeme und damit auch streng strukturell steuerbare Systeme eine Teilklasse der vorzeichen-stabilisierbaren Systeme bilden.

Man kann die beiden in Abschnitt 6 skizzierten Algorithmen leicht abändern, um ein System auf Vorzeichen-Stabilisierbarkeit zu untersuchen. Die Änderungen in *Algorithm 1* wurden in [9, S. 98] bereits explizit genannt.

In *Algorithm 2* muss man im Vergleich zur Überprüfung auf Vorzeichen-Steuerebarkeit die Prüfung auf strukturelle Steuerbarkeit weglassen, da strukturelle Steuerbarkeit keine notwendige Bedingung für Vorzeichen-Stabilisierbarkeit ist, siehe Abbildung 1.

Auch der auf diese Weise für die Vorzeichen-Stabilisierbarkeit angepasste *Algorithm 2* kann rekursiv konstruiert werden, da für die Vorzeichen-Stabilisierbarkeit die Reduktionsmöglichkeit aus Hartungs Satz 5.8 [9, S. 99] ebenfalls anwendbar ist.

## 8 Funktionen in der SALS Toolbox

Die SALS Toolbox ist eine MATLAB-Toolbox zur Analyse linearer Systeme mit Hilfe von Methoden, die von den konkreten numerischen Parametern der Systemmatrizen unabhängig sind [12]. Sie kann über den MathWorks File-Exchange-Server heruntergeladen werden. Es stehen Funktionen zur Verfügung, mit denen man alle komplexen Vorzeichenvektoren für Ordnungen  $n \leq 7$  generieren kann. Die ebenfalls bereitgestellten Funktionen zur Überprüfung von Systemen auf Vorzeichen-Steuerebarkeit und Vorzeichen-Stabilisierbarkeit greifen auf die Funktionen zur Erzeugung der komplexen Vorzeichenvektoren zurück.

Nachfolgend werden einzelne Funktionen der SALS Toolbox im Zusammenhang mit der Erzeugung der komplexen Vorzeichenvektoren vorgestellt. Die Beschreibung bezieht sich auf die Version 1.3.1 der Toolbox, die für MATLAB-Versionen ab R2018b lauffähig ist.

### 8.1 Funktion `genAllCsv()`

Die Funktion `genAllCsv()` ist eine MATLAB-Implementierung von Algorithmus 1. Mit ihr können alle komplexen Vorzeichenvektoren der Ordnungen  $n = 1, \dots, 7$  in Form der Vorzeichenmatrizen  $\mathcal{D}$  und in Form der Konstruktionsdatenmatrizen  $\nu$  erzeugt werden. Die Syntax zum Aufruf der Funktion lautet:

```
[csvList,cdmList] = genAllCsv(n)
```

`csvList` ist ein Vektor aller Vorzeichenmatrizen  $\mathcal{D}$  aus  $\mathbb{V}^{n \times n}$ , die komplexe Vorzeichenvektoren  $[q]$  der Ordnung  $n$  repräsentieren. Die zugehörigen Konstruktionsdatenmatrizen  $\nu$  stehen in `cdmList`. Die Einträge der gespeicherten Vorzeichenmatrizen sind zur effizienten Nutzung des Speicherplatzes vom Datentyp `uint8`. `csvList` hat die Struktur einer  $(n \times n \times \text{num})$ -Matrix, wobei `num` die mit Gleichung (17) bestimmte Zahl der komplexen Vorzeichenvektoren ist. `cdmList` hat die Struktur einer  $(2 \times n \times \text{num})$ -Matrix.

Die Vorzeichen in der `csvList` sind aus Effizienzgründen in der für die SALS Toolbox definierten Integer-Notation gespeichert. Zur besseren Lesbarkeit können die für die Vorzeichen verwendeten Symbole `z`, `p` und `n` mit Hilfe von MATLAB-Strings dargestellt werden. Es gelten folgende

Vereinbarungen:

$$z \hat{=} 0 \quad p \hat{=} 1 \quad n \hat{=} 2 \quad (34)$$

Mit Hilfe der Toolbox-Funktion `intf_int2str()` kann man eine Darstellung unter Verwendung der definierten Symbole erzeugen. Listing 1 zeigt die Erzeugung der komplexen Vorzeichenvektoren und die Wandlung in die String-Darstellung.

```
% Erzeugen aller komplexen Vor-
% zeichenvektoren der Ordnung 4
[csvList,cdmList] = genAllCsv(4);
% Auswahl Element 400 aus der Liste
D_int = csvList(:, :, 400);
% Wandeln in string-Darstellung
D_str = intf_int2str(D_int);
```

**Listing 1:** Beispiel zur Erzeugung komplexer Vorzeichenvektoren und Wandlung in String-Darstellung

Wegen der großen Anzahl der komplexen Vorzeichenvektoren ist der Speicherbedarf bereits für niedrige Ordnungen groß. Die Liste aller Vorzeichenmatrizen für die Ordnung  $n = 7$  belegt Speicherplatz von rund 2,15 GB. Für die Ordnung  $n = 8$  würde man rund 112,6 GB Speicher benötigen. Dieser Wert übersteigt die von MATLAB auf einem Windows-Rechner mit 16 GB Arbeitsspeicher mögliche Matrix-Größe um ein Vielfaches. Auch wenn man nur die Konstruktionsdatenmatrizen abspeichern würde, bräuchte man dafür bei  $n = 8$  bereits rund 28,2 GB Arbeitsspeicher.

## 8.2 Funktion `intf_genAllCsvNz()`

Mit der Funktion `intf_genAllCsvNz()` werden die Algorithmen 2 und 3 aus Abschnitt 5 umgesetzt. Die Funktion wird von `genAllCsv()` aufgerufen. Sie generiert alle komplexen Vorzeichenvektoren der Ordnung  $n$  mit genau  $n$  von null verschiedenen Elementen.

Die Syntax zum Aufruf der Funktion lautet

```
[csvList, cdmList] = genAllCsvNz(n).
```

`csvList` ist ein Vektor aller Vorzeichenmatrizen aus  $\mathbb{V}^{n \times n}$ , die komplexe Vorzeichenvektoren  $[q]$  der Ordnung  $n$  mit genau  $n$  Elementen ungleich null repräsentieren. Entsprechend ist `cdmList` der zugehörige Vektor der Konstruktionsdatenmatrizen.

## 8.3 Funktion `intf_stirling2()`

Mit der Funktion `intf_stirling2()` kann die Stirling-Zahl zweiter Art berechnet werden. Die Syntax lautet:

```
s2 = intf_stirling2(n,k)
```

Hierbei ist  $n$  die Anzahl der Elemente der Menge, die in  $k$  nichtleere und disjunkte Teilmengen partitioniert wird.  $s2$  ist die Stirling-Zahl zweiter Art. In der Funktion wird eine Formel aus [15, S. 287] verwendet.

## 8.4 Funktion `intf_getNumCsvAll()`

Die Anzahl aller komplexen Vorzeichenvektoren der Ordnung  $n$  kann mit der Funktion `intf_getNumCsvAll()` bestimmt werden. Die Syntax lautet:

```
numCsv = intf_getNumCsvAll(n)
```

Hierbei ist `numCsv` die Anzahl der komplexen Vorzeichenvektoren der Ordnung  $n$  nach Formel (17).

## 8.5 Funktion `intf_getSignPattern()`

Mit der Funktion `intf_getSignPattern()` können die charakteristischen Vorzeichenmuster eines komplexen Vorzeichenvektors  $[q]$  erzeugt werden.

Die charakteristischen Vorzeichenmuster wurden in Abschnitt 3 eingeführt. Für jeden komplexen Vorzeichenvektor  $[q]$  gibt es  $n$  Paare von Spaltenvektoren  $(\mathcal{D}_w, \mathcal{P}_w)$  mit  $w \in \{1, \dots, n\}$ . Diese werden in Form zweier Vorzeichenmatrizen  $\mathcal{D}$  und  $\mathcal{P}$  aus  $\mathbb{V}^{n \times n}$  dargestellt. Die Syntax lautet:

```
(dPat,pPat) = intf_getSignPattern(D)
```

Hierin sind `dPat` und `pPat` die beiden Vorzeichenmatrizen  $\mathcal{D}$  und  $\mathcal{P}$ , deren  $w$ -te Spalten jeweils das  $w$ -te charakteristische Vorzeichenmuster  $(\mathcal{D}_w, \mathcal{P}_w)$  bilden. Der Eingangsparameter `D` ist die Vorzeichenmatrix  $\mathcal{D}$  des komplexen Vorzeichenvektors.

### 8.6 Funktion `intf_chkSignPattern()`

Nicht jede Vorzeichenmatrix  $\mathfrak{X} \in \mathbb{V}^{n \times n}$  stellt einen komplexen Vorzeichenvektor dar. Zur Überprüfung bestimmt man zunächst die charakteristischen Vorzeichenmuster von  $\mathfrak{X}$ . Die Funktion `intf_chkSignPattern()` überprüft, ob die charakteristischen Vorzeichenmuster einen komplexen Vorzeichenvektor repräsentieren. Die Syntax für diese Funktion lautet:

```
isCsv = intf_chkSignPattern(dPat,pPat)
```

`dPat` und `pPat` sind die Vorzeichenmatrizen mit den charakteristischen Vorzeichenmustern, wie sie von Funktion `intf_getSignPattern()` bestimmt werden. Falls diese einen komplexen Vorzeichenvektor darstellen, ist das Ergebnis `isCsv` gleich eins, andernfalls null. Listing 2 zeigt die Anwendung dieser Funktion.

```
% X ist eine Vorzeichenmatrix
% in beliebiger Darstellung
[dPpat, pPat] = intf_getSignPattern(
    X);
isCsv = intf_chkSignPattern(...
    dPat, pPat);
```

**Listing 2:** Beispiel zur Prüfung, ob eine Vorzeichenmatrix einen komplexen Vorzeichenvektor darstellt

### 8.7 Funktion `intf_signPatt2Csv()`

Mit der Funktion `intf_signPatt2Csv()` kann aus den Matrizen mit den komplexen Vorzeichenmustern die Darstellung des komplexen Vorzeichenvektors  $[q]$  mit Hilfe der Vorzeichenmatrix  $\mathcal{D}$  erzeugt werden. Die Syntax lautet:

```
D = intf_signPatt2Csv(dPat,pPat)
```

Hierin sind `dPat` und `pPat` die beiden Vorzeichenmatrizen  $\mathcal{D}$  und  $\mathcal{P}$ , deren  $w$ -te Spalten jeweils die  $w$ -ten charakteristischen Vorzeichenmuster  $\mathcal{D}_w$  bzw.  $\mathcal{P}_w$  enthalten. Der Rückgabewert `D` ist die Vorzeichenmatrix  $\mathcal{D}$  des komplexen Vorzeichenvektors.

### 8.8 Funktion `intf_cdm2csv()`

Mit der Funktion `intf_cdm2csv()` kann man aus der Darstellung mit der Konstruktionsdatenmatrix

$\nu$  die Darstellung mit der Vorzeichenmatrix  $\mathcal{D}$  erzeugen. Die Syntax lautet:

```
csvD = intf_cdm2csv(cdm)
```

Hierin ist `csvD` die Darstellung des komplexen Vorzeichenvektors als Vorzeichenmatrix  $\mathcal{D}$  und `cdm` die Darstellung als Konstruktionsdatenmatrix  $\nu$ .

### 8.9 Funktion `intf_csv2cdm()`

Mit der Funktion `intf_csv2cdm()` kann man aus der Darstellung eines komplexen Vorzeichenvektors als Vorzeichenmatrix  $\mathcal{D}$  die Darstellung als Konstruktionsdatenmatrix  $\nu$  erzeugen. Die Syntax lautet:

```
cdm = intf_csv2cdm(D)
```

Hierin ist `cdm` die Darstellung des komplexen Vorzeichenvektors als Konstruktionsdatenmatrix  $\nu$  und `D` die Darstellung als Vorzeichenmatrix  $\mathcal{D}$ .

### 8.10 Funktion `csvFromComplVect()`

Mit der Funktion `csvFromComplVect()` kann man für einen komplexen Vektor  $q$  den komplexen Vorzeichenvektor in unterschiedlichen Darstellungen erzeugen. Die Syntax der drei Aufrufoptionen lautet:

```
D = csvFromComplVect(q,ctrl)
```

```
[Dpat,Ppat] = csvFromComplVect(q,ctrl)
```

```
[D,Dpat,Ppat] = csvFromComplVect(q,ctrl)
```

Hierin stehen  $q$  für den zu klassifizierenden komplexen Vektor,  $D$  für die Darstellung des komplexen Vorzeichenvektors mit der Vorzeichenmatrix  $\mathcal{D}$ ,  $Dpat$  und  $Ppat$  für die Vorzeichenmatrizen  $\mathcal{D}$  und  $\mathcal{P}$  mit den komplexen Vorzeichenmustern. Über den Parameter `ctrl` kann gewählt werden, ob die Darstellung der Vorzeichen mit den vereinbarten Symbolen oder mit Hilfe von Integer-Werten erfolgen soll.

## 9 Ergebnisse und Ausblick

In diesem Beitrag wurden der Begriff des komplexen Vorzeichenvektors und mehrere Darstellungsmöglichkeiten für komplexe Vorzeichenvektoren erläutert. Es wurde eine Methode entwickelt, rechnergestützt alle komplexen Vorzeichenvektoren der Ordnung  $n$  zu erzeugen. Der hierbei

entwickelte Algorithmus wurde in der MATLAB-Funktion `genAllCsv()` als Bestandteil der SALS Toolbox implementiert. Man kann damit alle komplexen Vorzeichenvektoren bis einschließlich der Ordnung  $n = 7$  als Vorzeichenmatrix und Konstruktionsdatenmatrix generieren. Weitere Funktionen im Zusammenhang mit komplexen Vorzeichenvektoren wurden ebenfalls realisiert und beschrieben. Mit den in der SALS Toolbox verfügbaren Funktionen ist ein rechnergestützter Umgang mit komplexen Vorzeichenvektoren sowie die Prüfung von Systemen auf Vorzeichen-Steuerbarkeit und Vorzeichen-Stabilisierbarkeit möglich.

Prinzipiell ist es möglich, auf die beschriebene Weise auch komplexe Vorzeichenvektoren für Ordnungen  $n > 7$  zu erzeugen. Allerdings ist in der Praxis der zur Verfügung stehende Speicherplatz begrenzt. Man könnte auch auf die Speicherung der komplexen Vorzeichenvektoren verzichten und direkt nach der Erzeugung des jeweiligen komplexen Vorzeichenvektors die gewünschten Prüfungen durchführen. Wegen der großen Anzahl komplexer Vorzeichenvektoren und der entsprechend großen Rechenzeiten ist dieser Ansatz allerdings auch nicht zielführend. Für die Praxis wird man sich daher auf solche Systeme beschränken müssen, deren Ordnung maximal 7 ist oder die auf ein System mit einer Ordnung von  $n \leq 7$  zurückgeführt werden können. ▣

### Symbolverzeichnis

$\sim_c$	Äquivalenzrelation zur Klassifikation von komplexen Vektoren
$\mathbb{C}$	Menge der komplexen Zahlen
$\mathbb{R}$	Menge der reellen Zahlen
$\mathbb{U}$	Menge der unsicheren Zahlen, $\mathbb{U} := \{z, p, n, zp, zn, pn, zpn\}$
$\mathbb{V}$	Menge der Vorzeichen, $\mathbb{V} := \{z, p, n\}$
$\mathcal{D}$	Vorzeichenmatrix mit den charakteristischen Vorzeichenmustern der Realteile eines komplexen Vorzeichenvektors, $\mathcal{D} \in \mathbb{V}^{n \times n}$
$\mathcal{D}_w$	$w$ -te Spalte von $\mathcal{D}$ , $\mathcal{D}_w \in \mathbb{V}^n$ . Der Vektor enthält die Vorzeichen der Realteile eines auf den $w$ -ten Eintrag normierten komplexen Vektors.
$(\mathcal{D}_w)_v$	$v$ -te Komponente in $\mathcal{D}_w$ bzw. Eintrag $\mathcal{D}_{v,w}$ in Zeile $v$ und Spalte $w$ in $\mathcal{D}$
$\mathcal{P}$	Vorzeichenmatrix mit den charakteristischen Vorzeichenmustern der Imaginärteile eines komplexen Vorzeichenvektors, $\mathcal{P} \in \mathbb{V}^{n \times n}$

$\mathcal{P}_w$	$w$ -te Spalte von $\mathcal{P}$ , $\mathcal{P}_w \in \mathbb{V}^n$ . Der Vektor enthält die Vorzeichen der Imaginärteile eines auf den $w$ -ten Eintrag normierten komplexen Vektors.
$(\mathcal{P}_w)_v$	$v$ -te Komponente in $\mathcal{P}_w$ bzw. Eintrag $\mathcal{P}_{v,w}$ in Zeile $v$ und Spalte $w$ in $\mathcal{P}$
$\mathfrak{D}$	Darstellungsform für einen komplexen Vorzeichenvektor mit Hilfe einer Vorzeichenmatrix, $\mathfrak{D} \in \mathbb{V}^{n \times n}$
$\kappa$	Anzahl aller komplexer Vorzeichenvektoren der Ordnung $n$
$\kappa_l$	Anzahl komplexer Vorzeichenvektoren der Ordnung $l$ für Vektoren mit genau $l$ von null verschiedenen Komponenten
$\nu$	Konstruktionsdatenmatrix zur Darstellung eines komplexen Vorzeichenvektors, $\nu \in \mathbb{N}_0^{2 \times n}$
$\gamma$	Vorzeichenvektor zur Charakterisierung der Vorzeichen von Real- und Imaginärteil einer komplexen Zahl, $\gamma \in \mathbb{V}^2$ . Die erste Komponente enthält das Vorzeichen des Imaginärteils.
$\phi_i$	Argument bzw. Winkel der $i$ -ten Komponente eines komplexen Vektors
$\rho$	Referenznummer zur Beschreibung einer der vier Rotationsmatrizen, $\rho \in \{0, \dots, 3\}$
$A$	Systemmatrix eines linearen zeitinvarianten Systems, $A \in \mathbb{R}^{n \times n}$
$B$	Eingangs- oder Steuermatrix eines linearen zeitinvarianten Systems, $B \in \mathbb{R}^{n \times r}$
$Q$	Quadrant der komplexen Zahlenebene, $Q \in \{1, \dots, 4\}$ . Die Quadranten 1 bis 4 werden auch mit den römischen Zahlen $I$ bis $IV$ bezeichnet.
$R$	Menge der Rotationsmatrizen zur Rotation von Vorzeichenmustern
$S$	Stirling-Zahl 2. Art, $S(l, k) = \begin{Bmatrix} l \\ k \end{Bmatrix}$
$d_Q$	Differenz von Quadranten zweier komplexen Zahlen modulo 4, $d_Q \in \{0, \dots, 3\}$
$d_{v,w}$	Element aus den $2n^2$ charakteristischen Daten für einen komplexen Vorzeichenvektor, $d_{v,w} := \text{sgn}(r_v r_w \cos(\phi_v - \phi_w))$
$n$	Ordnung eines Systems bzw. eines komplexen Vorzeichenvektors
$n$	Menge aller negativen reellen Zahlen
$p$	Menge aller positiven reellen Zahlen
$pn$	Menge der reellen Zahlen ungleich null
$p_{v,w}$	Element aus den $2n^2$ charakteristischen Daten für einen komplexen Vorzeichenvektor, $p_{v,w} := \text{sgn}(r_v r_w \sin(\phi_v - \phi_w))$



$q$	Komplexer Vektor, $q \in \mathbb{C}$
$[q]$	Komplexer Vorzeichenvektor zum komplexen Vektor $q$ , $[q] \in \mathbb{C}^n / \sim_c$
$r_i$	Betrag der $i$ -ten Komponente eines komplexen Vektors
$t$	Zeitvariable, $t \in \mathbb{R}$
$u$	Eingangs- oder Steuervektor eines linearen zeitinvarianten Systems, $u \in \mathbb{R}^r$
$x$	Zustandsvektor eines linearen zeitinvarianten Systems, $x \in \mathbb{R}^n$
$z$	Menge mit der reellen Zahl 0
$z_{\text{n}}$	Menge der nichtpositiven reellen Zahlen
$z_{\text{p}}$	Menge der nichtnegativen reellen Zahlen
$z_{\text{pn}}$	Menge der reellen Zahlen

## Literatur

- [1] Chin-Tai Lin. „Structural Controllability“. In: *IEEE Transactions on Automatic Control* AC-19.3 (Juni 1974), S. 201–208.
- [2] R. Shields und J. Pearson. „Structural controllability of multiinput linear systems“. In: *IEEE Transactions on Automatic Control* 21.2 (Apr. 1976), S. 203–212. ISSN: 0018-9286.
- [3] Hirokazu Mayeda und Takahashi Yamada. „Strong structural controllability“. In: *SIAM Journal Control and Optimization* 17 (1979), S. 123–138.
- [4] K. J. Reinschke, F. Svaricek und H. D. Wend. „On strong structural controllability of linear systems“. In: *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*. 1992, 203–208 vol.1.
- [5] Ferdinand Svaricek. *Zuverlässige numerische Analyse linearer Regelungssysteme*. Vieweg-Teubner Verlag Wiesbaden, 1. Feb. 1995. ISBN: 978-3-519-06175-5.
- [6] Jean-Michel Dion, Christian Commault und Jacob van der Woude. „Generic properties and control of linear structured systems: a survey“. In: *Automatica* 39.7 (2003), S. 1125–1144. ISSN: 0005-1098.
- [7] J. C. Jarczyk, F. Svaricek und B. Alt. „Strong structural controllability of linear systems revisited“. In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. Dez. 2011, S. 1213–1218.
- [8] Yang-Yu Liu, Jean-Jacques Slotine und Albert-László Barabási. „Controllability of complex networks“. In: *Nature* 473.7346 (Mai 2011), S. 167–173.
- [9] Christoph Hartung. *Zur algebraischen Untersuchung der Steuerbarkeit und der Stabilisierbarkeit linearer zeitinvarianter Systeme*. Fortschritt-Berichte VDI Reihe 8 Nr.1247, 2016.
- [10] Jiajia Jia u. a. „A Unifying Framework for Strong Structural Controllability“. In: <https://arxiv.org/abs/1903.03353> (2019).
- [11] Jan Lunze. *Regelungstechnik 2 : Mehrgrößensysteme, Digitale Regelung*. 9. Aufl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. ISBN: 9783662526750.
- [12] Alfred Geisel und Ferdinand Svaricek. „A MATLAB Toolbox for Structural Analysis of Linear Systems“. In: *IFAC-PapersOnLine* 52.17 (2019), S. 7–12.
- [13] Alfred Geisel u. a. *SALS Toolbox Version 1.3.1 MATLAB Central File Exchange*. Hrsg. von Alfred Geisel. 2020.
- [14] Christoph Hartung und Ferdinand Svaricek. „Sign Stabilizability“. In: *22nd Mediterranean Conference on Control and Automation (MED) University of Palermo. June 16-19. 2014*.
- [15] Richard A. Brualdi. *Introductory Combinatorics*. 5. Aufl. Pearson Education, 2010.
- [16] Raymond T. Boute. „The Euclidean definitions of the functions div and mod“. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 14.2 (1992), S. 127–144.
- [17] G. Hutchinson. „Partitioning Algorithms for Finite Sets“. In: *Communications of the ACM* 6.10 (1963), S. 613–614.
- [18] Alexander Weber, Gunther Reissig und Ferdinand Svaricek. „A linear time algorithm to verify strong structural controllability“. In: *53rd IEEE Conference on Decision and Control, December 15-17. Los Angeles, California, USA. 2014*.



# IMPRESSUM

**Schriftenreihe INSIGHTS**  
**Themenreihe Engineering INSIGHTS**

**Herausgeber:**

Fakultät Technik der  
Dualen Hochschule Baden-Württemberg Stuttgart  
Postfach 10 05 63, 70004 Stuttgart

**Prof. Dr.-Ing. Harald Mandel, stellvertretender Dekan der Fakultät Technik**  
Jägerstraße 56, 70174 Stuttgart

E-Mail: [harald.mandel@dhbw-stuttgart.de](mailto:harald.mandel@dhbw-stuttgart.de)  
Tel.: 0711/1849-605  
Fax: 0711/1849-719

[www.dhbw-stuttgart.de/technik/insights](http://www.dhbw-stuttgart.de/technik/insights)

**Satz und Gestaltung:** Prof. Alfred Geisel mit Unterstützung von PeiTeX TeXnical Solutions  
**Umschlaggestaltung und Korrektorat:** Inna Avrutina

**Bildnachweis:** Prof. Alfred Geisel

**ISSN 2193-9098**

© Prof. Alfred Geisel, 2020

Alle Rechte vorbehalten. Der Inhalt dieser Publikation unterliegt dem deutschen Urheberrecht. Die Vervielfältigung, Bearbeitung, Verbreitung und jede Art der Verwertung außerhalb der Grenzen des Urheberrechtes bedürfen der schriftlichen Zustimmung der Autoren und des Herausgebers.

Der Inhalt der Publikation wurde mit größter Sorgfalt erstellt. Für die Richtigkeit, Vollständigkeit und Aktualität des Inhalts übernimmt der Herausgeber keine Gewähr.

ISSN 2193-9098

[www.dhbw-stuttgart.de/technik/insights](http://www.dhbw-stuttgart.de/technik/insights)