

Zusammenfassung Automatisierungstechnik

Jonas Otto

Sommersemester 2022

Inhaltsverzeichnis

1	Ereignisdiskrete Systeme	3
1.1	Zustandsbeschreibung von Petri-Netzen	3
1.1.1	Netzmatrix	3
1.1.2	Zustandsgleichung	3
1.2	Analyse von Petri-Netzen	3
1.2.1	Zyklen: T-Invarianten	3
1.2.2	Lebendigkeit	4
1.2.3	Konstante Markensumme: S-Invarianten	4
1.3	Petri-Netz-Regler	4
2	Multi-Agenten-Systeme	6
2.1	Graphentheorie	6
2.2	Systemtheorie	7
2.2.1	Steuerbarkeit	7
2.2.2	Schur Zerlegung	7
2.2.3	Signalmodelle	7
2.2.4	Sonstige Sinnvolle Rechenregeln	8
2.3	Leader-Follower Zustandssynchronisierung	8
2.3.1	Folgefehler	8
2.3.2	Diffusive Kopplung	8
2.3.3	Fehlersystem mit diffusiver Kopplung	9
2.3.4	Simultanes Stabilisierungsproblem	9
2.3.5	Voraussetzungen Simultane Stabilisierung	10
2.3.6	Zustandstransformation	10
2.3.7	Entwurf Netzwerkregler	10
2.3.8	Sonderfälle	10
2.4	Leaderless Zustandssynchronisierung	11
2.4.1	Synchronisierungstrajektorie	11
2.4.2	Entwurf Netzwerkregler	12
2.5	Leader-Follower Ausgangssynchronisierung	13
2.5.1	Kooperativ angeregtes Internes Modell	13
2.5.2	Führungsbeobachter	13
2.6	Leaderless Ausgangssynchronisierung	14
2.6.1	Kooperativ angeregtes Internes Modell	14
2.6.2	Führungsbeobachter	16
2.7	Synchronisierung heterogener Multiagentensysteme	17
2.7.1	Kooperative Führungsbeobachter: Leader-Follower	17
2.8	Schaltende Netze	20

Kapitel 1

Ereignisdiskrete Systeme

1.1 Zustandsbeschreibung von Petri-Netzen

Der Zustand eines Petri-Netzes wird durch einen **Markierungsvektor** $m \in \{0, 1\}^{|S|}$ angegeben. Das ist bei uns ein binärer Vektor der angibt welche Stellen mit Marken belegt sind.

Ein Schaltvorgang wird mit einem **Transitionsvektor** $t \in \{0, 1\}^{|S|}$ angegeben. Der Transitionsvektor enthält einen 1-Eintrag für die schaltende Transition.

1.1.1 Netzmatrix

Die Netzmatrix beschreibt Spaltenweise den Markenfluss beim Schalten der Transitionen:

$$N = \begin{matrix} & \begin{matrix} t1 & t2 \end{matrix} \\ \begin{matrix} s1 \\ s2 \\ s3 \end{matrix} & \begin{pmatrix} -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \end{matrix} \quad (1.1)$$

1.1.2 Zustandsgleichung

Damit kann man dann die Zustandsgleichung aufstellen:

$$m(k+1) = m(k) + Nt(k) \quad (1.2)$$

Mit der wichtigen Nebenbedingung

$$t(k) \in \mathcal{T}_{\text{akt}}(m(k), u(k)) \quad (1.3)$$

welche fordert, dass die Transition aktiviert sein muss, also alle Prästellen markiert und alle Poststellen unmarkiert sind, und die Transition durch den Eingang $u(k)$ „freigeschaltet“ ist.

Die Anfangsmarkierung ist $m(0)$.

1.2 Analyse von Petri-Netzen

1.2.1 Zyklen: T-Invarianten

Ein Zyklus kann so definiert werden:

$$m(k_z) = m(0) + N \sum_{\kappa=0}^{k_z-1} t(\kappa) \stackrel{!}{=} m(0) \quad (1.4)$$

Oder äquivalent

$$N \sum_{\kappa=0}^{k_z-1} t(\kappa) = \boxed{Ni_T = 0} \quad (1.5)$$

Wenn jetzt also so ein passendes i_T (ganzzahlig, > 0) existiert, nennt man das **T-Invariante**. Wenn zusätzlich die T-Invariante als gültige Abfolge von Transition darstellbar ist, folgt daraus dass das Petri-Netz einen Zyklus besitzt.

1.2.2 Lebendigkeit

Eine **tote Markierung** ist eine Markierung, in welcher keine Marken mehr fließen. Wenn keine toten Markierungen auftreten, ist das Netz lebendig.

Jedes lebendige Petri-Netz besitzt mindestens einen Zyklus und damit eine T-Invariante.

1.2.3 Konstante Markensumme: S-Invarianten

Man fordert dass für jeden Zeitpunkt die gewichtete Markensumme konstant ist:

$$\sum_{i=1}^{|S|} i_{S,i} m_i(k) = i_S^T m(k) = \text{const} \quad \forall k \in \mathbb{N}_0, i_{S,i} \in \mathbb{N}_0 \quad (1.6)$$

mit Gewichtsvektor $i_S \in \mathbb{N}_0^{|S|}$.

Induktiv bedeutet das

$$\begin{aligned} i_S^T m(k+1) &= i_S^T m(k) + i_S^T N t(k) \stackrel{!}{=} i_S^T m(k) \\ \iff &\boxed{i_S^T N = 0} \end{aligned} \quad (1.7)$$

wobei $i_S \in \mathbb{N}_0^{|S|}$ die nichttriviale Lösung sein soll. Dann ist i_S eine **S-Invariante** des Petri-Netzes.

Wenn eine S-Invariante keinen 0-Eintrag enthält, bedeutet das, dass die Anzahl der Markierungen im Petri-Netz konstant ist.

1.3 Petri-Netz-Regler

Die Spezifikation des Reglers wird mit einer Ungleichungsbedingung formuliert:

$$Lm(k) \leq b, \quad k \in \mathbb{N}_0, \quad L \in \mathbb{N}_0^{n_R \times |S|}, \quad b \in \mathbb{N}_0^{n_R} \quad (1.8)$$

mit n_R der Anzahl an Bedingungen (Ordnung des Reglers), und \leq Elementweise.

Der ereignisdiskrete Regler ist selbst ein Petri-Netz, mit Zustandsdarstellung

$$m_R(k+1) = m_R(k) + N_R t(k), \quad m_R \in \mathbb{N}_0^{n_R} \quad (1.9)$$

, das subscript $_R$ steht für „Regler“.

Achtung: $t(k)$ ist auch die Transition der Strecke!

Es muss gelten $N_R \in \mathbb{Z}^{n_R \times |\mathcal{T}|}$ mit $|\mathcal{T}|$ der Anzahl an Transitionen der Strecke.

Zum Reglerentwurf wird die Ungleichungsbedingung auf S-Invarianten-Vorgabe zurückgeführt.

Man setzt

$$Lm(k) + m_R(k) = b \quad (1.10)$$

(das geht, weil $m_R(k) \in \mathbb{N}_0^{n_R}$), und das Regler und Strecke zusammen:

$$\begin{bmatrix} m(k+1) \\ m_R(k+1) \end{bmatrix} = \begin{bmatrix} m(k) \\ m_R(k) \end{bmatrix} + \begin{bmatrix} N \\ N_R \end{bmatrix} t(k) \quad (1.11)$$

Was dann erlaubt die Bedingung umzuschreiben zu

$$\begin{bmatrix} L & I \end{bmatrix} \begin{bmatrix} m(k) \\ m_R(k) \end{bmatrix} = b \quad (1.12)$$

Was bedeutet dass die Zeilen von $\begin{bmatrix} L & I \end{bmatrix}$ gerade S-Invarianten des geregelten Systems sein müssen.

Für $k = 0$ ergibt sich durch Einsetzen

$$\boxed{m_R(0) = b - Lm(0)} \quad (1.13)$$

(Was dann auch b garantiert, ohne die Anfangsbedingung würden die S-Invarianten zwar konstantes, aber beliebiges b fordern.)

Um nun N_R zu konstruieren wird die Definition der S-Invarianten genutzt (für jede Zeile, rechts steht jetzt ein Nullvektor statt Skalar):

$$\begin{aligned} \begin{bmatrix} L & I \end{bmatrix} \begin{bmatrix} N \\ N_R \end{bmatrix} &= 0 \\ \iff LN + N_R &= 0 \\ \boxed{\iff N_R} &= -LN \end{aligned} \quad (1.14)$$

Kapitel 2

Multi-Agenten-Systeme

2.1 Graphentheorie

Gradmatrix Die Gradmatrix D_G enthält auf der Hauptdiagonalen den Eingangsgrad des jeweiligen Knotens.

Adjazenzmatrix Die Adjazenzmatrix A_G enthält im Element a_{ij} das Kantengewicht der Kante $(j \rightarrow i)$.

Die Adjazenzmatrix eines nicht stark zusammenhängenden Graphen ist reduzibel. Eine Matrix ist reduzibel, wenn mittels Permutationsmatrix P gilt

$$PMP^T = \begin{bmatrix} M_{11} & 0 \\ M_{12} & M_{22} \end{bmatrix} \quad (2.1)$$

mit quadratischem M_{11} . Permutationsmatrix bedeutet dass PM Zeilen tauscht, und MP^T Spalten tauscht.

Wenn PA_GP^T eine untere Blockdreiecksmatrix ist mit irreduziblen Diagonalblockmatrizen, entspricht das der **Frobeniusform**. Die Diagonalblockmatrizen beschreiben dann stark zusammenhängende Teilgraphen.

Laplacematrix

$$L_G = D_G - A_G \quad (\text{Laplacematrix})$$

Die Laplacematrix berechnet sich aus Gradmatrix D und Adjazenzmatrix A .

- Ein Eigenwert von L_G bei 0 ist einfach \iff Graph zusammenhängend
- $\text{rang } L_G = N - 1$ (N Knoten) \iff Graph zusammenhängend
- Die Aussagen bzgl. Reduktion der Adjazenzmatrix gelten auch für die Laplacematrix.
- Das Laplacespektrum ist die Menge der Eigenwerte von L_G .

Inzidenzmatrix Elemente n_{ij} der Inzidenzmatrix $N_G \in \{-1, 0, 1\}^{N \times M}$ eines Graphen mit N Knoten und M Kanten:

$$n_{ij} = \begin{cases} -1, & \text{wenn Kante } j \text{ im Knoten } i \text{ startet} \\ 1, & \text{wenn Kante } j \text{ im Knoten } i \text{ endet} \\ 0, & \text{sonst} \end{cases} \quad (2.2)$$

Jede Spalte stellt dabei eine Kante dar.

Zusammenhängend Ein Wurzelknoten existiert.

- \iff Adjazenzmatrix A_G irreduzibel.
- \iff Ein Eigenwert von L_G bei 0 ist einfach

Stark Zusammenhängend Jeder Knoten ist Wurzelknoten, es gibt von jedem Knoten einen Pfad zu jedem anderen Knoten. \iff Laplacematrix L_G irreduzibel.

Vollständig Zusammenhängend = fully connected.

2.2 Systemtheorie

2.2.1 Steuerbarkeit

Der Eigentwert λ_i von A ist steuerbar wenn gilt

$$\text{rang} \begin{bmatrix} A - \lambda_i I & B \end{bmatrix} = n \quad (2.3)$$

Wobei $A \in \mathbb{R}^{n \times n}$. Sind alle Eigenwerte in der rechten Halbebene steuerbar, ist (A, B) steuerbar bzw stabilisierbar.

2.2.2 Schur Zerlegung

Quadratische Matrix A :

$$R = U^H A U = \quad (2.4)$$

- R ist obere Dreiecksmatrix (\implies EW auf Hauptdiagonalen)
- Die Matrizen A und R sind Ähnlich, haben also die gleichen Eigenwerte

Mit Blockmatrizen/Kronecker-Produkten funktioniert das auch.

2.2.3 Signalmodelle

Konstant

$$\dot{v} = 0 \quad (2.5)$$

$$v(t) = v(0) = c \quad (2.6)$$

$$\sigma = \{0\} \quad (2.7)$$

Rampe

$$\dot{v} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} v \quad (2.8)$$

$$v(0) = \begin{bmatrix} a \\ b \end{bmatrix} \quad (2.9)$$

$$v(t) = \begin{bmatrix} b \cdot t + a \\ b \end{bmatrix} \quad (2.10)$$

$$\sigma = \{0, 0\} \quad (2.11)$$

Sinus

$$\dot{v} = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} v \quad (2.12)$$

$$v(0) = \begin{bmatrix} \sin(\varphi_1) \\ \cos(\varphi_2) \end{bmatrix} \quad (2.13)$$

$$v(t) = \begin{bmatrix} \sin(\omega t + \varphi_1) \\ \cos(\omega t + \varphi_2) \end{bmatrix} \quad (2.14)$$

$$\sigma = \{+j\omega, -j\omega\} \quad (2.15)$$

2.2.4 Sonstige Sinnvolle Rechenregeln

- $\det(\text{Block-Diag-Matrix}) = \prod_i \det(\text{Block}_i)$

2.3 Leader-Follower Zustandssynchronisierung

Leader-Follower-Matrix

$$H = L_G + L_0 \in \mathbb{R}^{N \times N} \quad (\text{Leader-Follower-Matrix})$$

L_0 ist dabei die Diagonalmatrix mit $L_{0,ii} = a_{i0}$. L_G enthält den Führungsagenten nicht!

Es gilt $\det(H) \neq 0$ und $\text{Re } \lambda_i(H) > 0$, wenn der erweiterte Graph einen Spannbaum mit dem Führungsagenten als Wurzel besitzt.

2.3.1 Folgefehler

Der Folgefehler ist definiert als:

$$\varepsilon = \begin{bmatrix} x_1 - x_0 \\ \vdots \\ x_N - x_0 \end{bmatrix} = x - 1_N \otimes x_0 \quad (2.16)$$

wobei x_i der Zustandsvektor des Agenten i ist. Mit ein bisschen einsetzen von $\dot{x} = (I_N \otimes A)x + (I_N \otimes B)u$ und Umformen kommt man daraus auf die **Folgefehlerdynamik**

$$\dot{\varepsilon} = (I_N \otimes A)\varepsilon + (I_N \otimes B)u \quad (2.17)$$

Mit einem Regler $u = -(I_N \otimes K)\varepsilon$ ergibt sich das **geregelte Folgefehlersystem**

$$\dot{\varepsilon} = (I_N \otimes (A - BK))\varepsilon \quad (2.18)$$

Aufgabe ist jetzt K so zu bestimmen dass das geregelte Fehlersystem unseren Anforderungen (exponentielle Stabilität) entspricht.

Dies beachtet allerdings noch nicht die Netzwerktopologie. Das macht im nächsten Abschnitt die

2.3.2 Diffusive Kopplung

Im Hinblick auf die Netzwerktopologie findet sich der Folgefehler in der diffusiven Kopplung wieder:

$$\bar{u}_i = \sum_{j=1}^N a_{ij}(x_i - x_j) + a_{i0}(x_i - x_0) \quad (2.19)$$

$$= \sum_{j=1}^N a_{ij}(\varepsilon_i - \varepsilon_j) + a_{i0}\varepsilon_i \quad (2.20)$$

$$= \left(e_i^T \underbrace{(L_G + L_0)}_H \otimes I_n \right) \varepsilon \quad (2.21)$$

Wobei e_i^T ein Zeilenvektor ist mit 1 an Stelle i , sonst 0.

Aggregieren der \bar{u}_i Elemente ergibt

$$\bar{u} = (H \otimes I_n) \varepsilon \quad (2.22)$$

2.3.3 Fehlersystem mit diffusiver Kopplung

Nun lässt sich eine Rückführung definieren:

$$u = -\tilde{K}(H \otimes I_n) \varepsilon \quad (2.23)$$

wobei \tilde{K} eine Blockmatrix ist mit den Rückführmatrizen K_i der einzelnen Agenten auf der Hauptdiagonalen.

Das geregelte Fehlersystem ist nun

$$\dot{\varepsilon} = (I_N \otimes A - (I_N \otimes B)\tilde{K}(H \otimes I_n)) \varepsilon \quad (2.24)$$

2.3.4 Simultanes Stabilisierungsproblem

Im folgenden wird für jeden Folgeagenten die Gleiche Rückführmatrix K verwendet. Das bedeutet, dass diese Matrix **simultan** jeden Folgeagenten stabilisieren muss:

$$\tilde{K} = I_N \otimes K \quad (2.25)$$

Das Stellgesetz lautet nun

$$u = -(I_N \otimes K)(H \otimes I_n) \varepsilon = -(H \otimes K) \varepsilon \quad (2.26)$$

Mit dem Ziel, die leader-follower Matrix H auf Schursche Normalform zu bringen, wird die Transformation

$$\tilde{\varepsilon} = (T^H \otimes I_n) \varepsilon \quad (2.27)$$

$$\varepsilon = (T \otimes I_n) \tilde{\varepsilon} \quad (2.28)$$

eingeführt.

Eingesetzt in die Gleichung des geregelten Fehlersystems 2.24 ergibt sich:

$$\dot{\tilde{\varepsilon}} = ((T^H \otimes I_n)(I_N \otimes A)(T \otimes I_n) - (T^H \otimes I_n)(I_N \otimes B)\tilde{K}(H \otimes I_n)(T \otimes I_n)) \tilde{\varepsilon} \quad (2.29)$$

$$= (I_N \otimes A - H \otimes BK) \tilde{\varepsilon} \quad (2.30)$$

$$= \tilde{A} \tilde{\varepsilon} \quad (2.31)$$

mit $H \triangleright = T^H H T$.

Da die Diagonalelemente von $H \triangleright$ gerade dessen Eigenwerte, und damit auch die Eigenwerte von H sind (Matrizen sind Ähnlich), lassen sich die Blockdiagonalelemente von \tilde{A} darstellen als:

$$\tilde{A}_{ii} = A - \lambda(H)BK \quad (2.32)$$

K muss nun also so bestimmt werden, dass diese N unterschiedlichen Matrizen simultan zu Hurwitz-Matrizen ($\text{Re } \lambda < 0$) macht.

Notwendige Bedingung dafür ist $\det H \neq 0$ (Siehe 2.3, das kann man evtl am Graph sehen).

Der Riccati-Entwurf kann die simultane Stabilisierung in diesem Fall immer lösen, was das ganze zur hinreichenden Bedingung macht.

2.3.5 Voraussetzungen Simultane Stabilisierung

- (A, b) der Agenten muss stabilisierbar sein: Siehe 2.2.1
- Führungsagent ist Wurzelknoten des erweiterten Graphens: Angeben eines Spannbaums oder zeigen dass $\det H \neq 0$ für die Leader-Follower Matrix.

2.3.6 Zustandstransformation

Wenn z.B. der Abstand von Agenten untereinander geregelt wird, bietet sich eine Zustandstransformation an die das berücksichtigt. Wenn in der ersten Komponente z.B. die Position steht und jeder Agent i den Abstand Δx_i haben soll:

$$\bar{x}_i = x_i - \begin{bmatrix} \Delta x_i \\ 0 \end{bmatrix} \quad (2.33)$$

Die A Matrix etc muss dann für den neuen Zustand bestimmt werden, und im Folgenden wird damit dann gerechnet.

2.3.7 Entwurf Netzwerkregler

Ziel ist ein Netzwerkregler der Form

$$u_i = - \sum_{j=1}^N a_{ij} (Kx_i - Kx_j) - a_{i0} (Kx_i - Kx_0) \quad (2.34)$$

$$= -K(e_i^T H \otimes I_n) \varepsilon \quad (2.35)$$

Die Rückführmatrix K bestimmt sich durch

$$K = cR^{-1}B^T P \quad (2.36)$$

mit $P \in \mathbb{R}^{n \times n}$ (symmetrisch) Lösung der algebraischen Riccati-Gleichung

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (\text{ARE})$$

wobei $Q \in \mathbb{R}^{n \times n}$ und $R \in \mathbb{R}^{p \times p}$ positiv definit, mit n Dimension des Zustandes und p Dimension des Inputs.

Für den Parameter c muss gelten

$$c \geq \frac{1}{2 \cdot \min_{i=1, \dots, N} \text{Re } \lambda_i(H)} \quad (2.37)$$

2.3.8 Sonderfälle

Sonderfall: Reduzible Kommunikationstopologie

Wenn die Leader-Follower-Matrix als untere $(r \times r)$ Blockdiagonalmatrix dargestellt werden kann (evtl nach Ändern der Nummerierung der Agenten), können r unabhängige Stabilisierungsprobleme mit r unterschiedlichen Rückführmatrizen gelöst werden.

Dabei ist zu beachten dass sich im Vergleich zum allgemeinen Fall in der Gleichung für K nur c ändert, P und B sind nicht vom Graph abhängig.

Sonderfall: Kreisfreie Kommunikationstopologie

Eine kreisfreie Topologie erlaubt umnummerierung sodass die Leader-Follower-Matrix eine untere $(N \times N)$ Dreiecksmatrix ist.

Dann können N unterschiedliche Rückführmatrizen bestimmt werden:

Das Fehlersystem ist jetzt

$$\dot{\varepsilon} = (I_N \otimes A - (I_N \otimes B)\tilde{K}(H \otimes I_n))\varepsilon \quad (2.38)$$

$$= \begin{bmatrix} A - h_{11}BK_1 & & \\ \vdots & \ddots & \\ -h_NBK_N & \dots & A - H_{NN}BK_N \end{bmatrix} \varepsilon \quad (2.39)$$

Daraus folgt, dass die Stabilisierung mit unterschiedlichen Rückführmatrizen K_i gelöst werden kann, indem jeweils $(A - h_{ii}BK_i)$ stabilisiert wird. Das haben wir in der Übung mit Eigenwertvorgabe (Koeffizientenvergleich) gemacht.

2.4 Leaderless Zustandssynchronisierung

Folgefehlerdynamik Es gibt zwar keinen Führungsagenten, aber wir können einen Folgefehler bezüglich eines ausgewählten Agentens (x_1) definieren. Damit ergeben sich die **Synchronisierungskoordinaten**:

$$\tilde{x} = \begin{bmatrix} x_1 \\ \varepsilon \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 - x_1 \\ \vdots \\ x_N - x_1 \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 \\ -1_{N-1} & I_{N-1} \end{bmatrix} \otimes I_n \right) x = (\Theta \otimes I_n)x \quad (2.40)$$

(Die Einträge 1, 0, und -1_{N-1} in Θ sind hierbei als skalar bzw Zeilen- und Spaltenvektor zu verstehen.)

$$\Theta^{-1} = \begin{bmatrix} 1 & 0 \\ 1_{N-1} & I_{N-1} \end{bmatrix} \quad (2.41)$$

Aus der Systemdynamik

$$\dot{x} = (I_N \otimes A)x + (I_N \otimes B)u \quad (2.42)$$

folgt dann nach Transformation auf Synchronisierungskoordinaten die **Folgefehlerdynamik**:

$$\dot{\tilde{x}} = (I_N \otimes A)\tilde{x} + (\Theta \otimes B)u \quad (2.43)$$

Stellgesetz

$$u_i = -K \sum_{j=1}^N a_{ij}(x_i - x_j) \quad (2.44)$$

2.4.1 Synchronisierungstrajektorie

Der Netzwerkregler stellt sicher dass die Agenten gegen die Synchronisierungstrajektorie konvergieren (auch „virtueller Führungsagent“):

$$\dot{x}_s = Ax_s \quad (2.45)$$

$$x_s(0) = (l^T \otimes I_n)x(0) \quad (2.46)$$

Der Vektor $l \in \mathbb{R}^N$ gewichtet dabei die Anfangszustände der einzelnen Agenten. Zur Bestimmung genügen die constraints

$$l^T 1_N = 1 \quad (2.47)$$

$$l^T L_G = 0 \quad (2.48)$$

Dies folgt von der Herleitung über die **invariante Trajektorie**, für die Unabhängigkeit von K gefordert wird.

Nur **Wurzelagenten** können einen Beitrag zur Synchronisierungstrajektorie leisten!

2.4.2 Entwurf Netzwerkregler

Bedingung

Damit sich eine Rückführmatrix K bestimmen lässt, muss gelten

$$\operatorname{Re} \lambda_i(L_{22}) > 0 \quad (2.49)$$

bzw.

$$\det L_{22} \neq 0 \quad (2.50)$$

wobei L_{22} folgt aus

$$\Theta L_G \Theta^{-1} = \begin{bmatrix} 0 & \Theta L_G M \\ 0_{N-1} & L_{22} \end{bmatrix} \quad (2.51)$$

$\det L_{22} \neq 0$ folgt auch direkt daraus, dass der Graph zusammenhängend ist.

Außerdem muss (A, b) steuerbar und stabilisierbar sein.

Entwurf

Bestimmung von K wie in 2.3.7, aber mit L_{22} statt H .

2.5 Leader-Follower Ausgangssynchronisierung

2.5.1 Kooperativ angeregtes Internes Modell

Das kann man so machen wie Leaderless mit internem Modell (2.6.1). Beim Riccati Entwurf muss für die Berechnung H statt L_{22} verwendet werden.

2.5.2 Führungsbeobachter

In 2.7.1 ist angegeben wie das mit heterogenen Agenten funktioniert, das geht natürlich auch hier.

2.6 Leaderless Ausgangssynchronisierung

2.6.1 Kooperativ angeregtes Internes Modell

Hier gibt es statt Führungsagent ein Führungsmodell:

$$\dot{v}_w = S_w v_w \quad (2.52)$$

$$w = Q_w v_w \quad (2.53)$$

Und bei Bedarf Störmodelle S_{d_i} .

Der Netzwerkregler für die Ausgangssynchronisierung beinhaltet die Dynamik des Führungsmodells und aller Störmodelle. Dafür baut man das kooperativ angeregte interne Modell:

$$\dot{v}_i = S^* \bar{v}_i + B_y \sum_{j=1}^N a_{ij} (y_i - y_j) \quad (2.54)$$

$$S^* = I_p \otimes S \quad (2.55)$$

S^* ist die $p = \dim y$ - fache Kopie von S , wobei S die Zusammenfassung des Führungssignalmodells und Störmodells ist. Der Zyklische Anteil bedeutet, dass jede Signalform nur ein mal enthalten sein soll.

$$S = \text{zykl}(\text{diag}(S_w, S_{d1}, \dots, S_{dN})) \quad (2.56)$$

$B_y = \text{diag}(b_1, \dots, b_p)$ mit Spaltenvektoren b_i muss weiterhin so gewählt werden dass (S, b_i) steuerbar ist. Für Sinus- und Rampensignale kann üblicherweise $[0 \ 1]^T$ gewählt werden.

Die **Voraussetzungen** für erfolgreiche Ausgangssynchronisierung mit internem-Modell-Prinzip sind:

- Graph \mathcal{G} zusammenhängend
- (A, B) stabilisierbar
- Kein EW von S ist invariante Nullstelle der Agenten

Invariante Nullstellen η werden ermittelt durch

$$\det \begin{bmatrix} A - \eta I & B \\ C & 0 \end{bmatrix} = 0 \quad (2.57)$$

Das System kann auch keine oder mehrere invariante Nullstellen haben.

Für die Stabilisierung setzt man das **Stellgesetz** an in der Form

$$u = - \underbrace{(I_N \otimes R_{\bar{v}})}_{\text{Lokal: } u^l} \bar{v} - \underbrace{(L_{\mathcal{G}} \otimes K_x)}_{\text{Kooperativ: } u^k} x \quad (2.58)$$

$$u_i = -R_{\bar{v}} \bar{v}_i - \sum_{j=1}^N a_{ij} K_x (x_i - x_j) \quad (2.59)$$

Eingesetzt und aggregiert ergibt das folgende **Dynamik für das geregelte MAS** und das **kooperative interne Modell**:

$$\dot{x} = (I_N \otimes A - L_{\mathcal{G}} \otimes B K_x) x - (I_N \otimes B R_{\bar{v}}) \bar{v} \quad (2.60)$$

$$\dot{\bar{v}} = (I_N \otimes S^*) \bar{v} + (L_{\mathcal{G}} \otimes B_y C) x \quad (2.61)$$

Die Transformation auf Synchronisierungskoordinaten

$$\tilde{x} = \begin{bmatrix} x_1 \\ \varepsilon_x \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 - x_1 \\ \vdots \\ x_N - x_1 \end{bmatrix} \quad (2.62)$$

erfolgt mittels $\tilde{x} = (\Theta \otimes I_n)x$, analog für \tilde{v} .

Das System kann man nun aufteilen in einen Anteil der instabil bleibt, und die **Synchronisierungsdynamik** vorgibt:

$$\dot{x}_1 = Ax_1 - (l_{12}^T \otimes BK_x)\varepsilon_x - BR_{\bar{v}}\bar{v}_1 \quad (2.63)$$

$$\dot{\bar{v}}_1 = S^*\bar{v}_1 + (l_{12}^T \otimes B_y C)\varepsilon_x \quad (2.64)$$

Und die Fehlerzustände, welche mittels simultaner Stabilisierung zu null geregelt werden sollen:

$$\dot{\varepsilon}_x = (I_{N-1} \otimes A - L_{22} \otimes BK_x)\varepsilon_x - (I_{N-1} \otimes BR_{\bar{v}})\varepsilon_{\bar{v}} \quad (2.65)$$

$$\dot{\varepsilon}_{\bar{v}} = (I_{N-1} \otimes S^*)\varepsilon_{\bar{v}} + (L_{22} \otimes B_y C)\varepsilon_x \quad (2.66)$$

Das simultane Stabilisierungsproblem ist dann

$$\check{A}_{\gamma,ii} = \tilde{A} - \lambda_i(L_{22})\tilde{B} \begin{bmatrix} K_x & R_{\bar{v}} \end{bmatrix} \quad (2.67)$$

Wobei $\check{A}_{\gamma,ii}$ Hurwitz-Matrix (alle EW links der Imaginärachse) sein soll für alle $\lambda_i(L_{22})$, mit

$$\tilde{A} = \begin{bmatrix} A & 0 \\ B_y C & S^* \end{bmatrix} \quad (2.68)$$

$$\tilde{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad (2.69)$$

Das kann man mit ARE-Entwurf machen für das $(\varepsilon_x, \varepsilon_{\bar{v}})$ -System:

$$\boxed{\begin{bmatrix} K_x & R_{\bar{v}} \end{bmatrix} = cR^{-1}\tilde{B}^T P} \quad (2.70)$$

Mit P (symmetrisch) Lösung der ARE

$$\tilde{A}^T P + P\tilde{A} - P\tilde{B}R^{-1}\tilde{B}^T P + Q = 0 \quad (2.71)$$

und

$$c \geq \frac{1}{2 \min(\operatorname{Re} \lambda_i(L_{22}))} \quad (2.72)$$

Die Synchronisierungstrajektorie ergibt sich dann als

$$\begin{bmatrix} \dot{x}_s \\ \dot{\bar{v}}_s \end{bmatrix} = \begin{bmatrix} A & -BR_{\bar{v}} \\ 0 & S^* \end{bmatrix} \begin{bmatrix} x_s \\ \bar{v}_s \end{bmatrix} \quad (2.73)$$

Da in \dot{x}_s die (eventuell instabile) Agentendynamik A enthalten ist, kann es sinnvoll sein eine lokale Stabilisierung zu machen:

$$u_i = -K_l x_i + \bar{u}_i \quad (2.74)$$

$$\dot{x}_i = \underbrace{(A - BK_l)}_{\text{Neue Dynamikmatrix } \tilde{A}_l} x_i + B\bar{u}_i \quad (2.75)$$

Was dann in der Synchronisierungstrajektorie

$$\begin{bmatrix} \dot{x}_s \\ \dot{\bar{v}}_s \end{bmatrix} = \begin{bmatrix} \tilde{A}_l & -BR_{\bar{v}} \\ 0 & S^* \end{bmatrix} \begin{bmatrix} x_s \\ \bar{v}_s \end{bmatrix} \quad (2.76)$$

resultiert. Das \tilde{A}_l muss aber auch im Entwurf berücksichtigt werden!

2.6.2 Führungsbeobachter

Siehe 2.7.1.

2.7 Synchronisierung heterogener Multiagentensysteme

Um das heterogene Multiagentensystem zu synchronisieren, müssen die Agenten das **kooperative Interne-Modell-Prinzip** erfüllen:

Gemeinsame Teildynamik A_s mit Ausgang C_s .

$$T_i A_i T_i^{-1} = \begin{bmatrix} A_s & * \\ 0 & * \end{bmatrix} \quad (2.77)$$

$$C_i T_i^{-1} = [C_s \quad *] \quad (2.78)$$

Bemerkung: Notwendig (aber nicht hinreichend) dafür ist, dass die Agenten gleiche Eigenwerte haben.

Wenn das nicht erfüllt ist, gibt es zwei Lösungsmöglichkeiten:

1. Dynamik mit Regler mitberücksichtigen (kooperativ angeregtes internes Modell)
2. Kooperative Führungsbeobachter

2.7.1 Kooperative Führungsbeobachter: Leader-Follower

Heterogene Agenten:

$$\dot{x}_i = A_i x_i + B_i u_i \quad (2.79)$$

$$y_i = C_i x_i \quad (2.80)$$

Die Synchronisierungstrajektorie wird vom Führungsagenten vorgegeben, beschrieben durch Signalmodell:

$$\dot{v}_0 = S_w v_0 \quad (2.81)$$

$$w = Q_w v_0 \quad (2.82)$$

(Anforderung: (Q_w, S_w) beobachtbar)

Daraus resultiert das Führungssignal w . Diesem Signal sollen die Ausgänge der Agenten folgen.

Dafür werden an den Agenten Führungsbeobachter entworfen, die den Zustand des Führungsagenten rekonstruieren (\hat{v}_i).

Dann wird eine Führungsgrößenaufschaltung konstruiert, die eine Zustandsrückführung der Agentenzustände und eine Aufschaltung des rekonstruierten Führungsagentenzustands enthält.

Informierte Agenten

Bei informierten Agenten ist das ein klassischer Beobachter:

$$\dot{\hat{v}}_i = S_w \hat{v}_i + L_i a_{i0} (w - \hat{w}_i) \quad (2.83)$$

(Die Beobachterrückführmatrizen L_i lassen sich für jeden Agenten unterschiedlich wählen.)

Hier wird ein lokaler Führungsbeobachter entworfen. Dafür betrachtet man den globalen Beobachtungsfehler $e_i = v_0 - \hat{v}_i$:

$$\dot{e}_i = (S_w - L_i a_{i0} Q_w) e_i \quad (2.84)$$

Daraus dass (Q_w, S_w) beobachtbar ist, kann immer ein passendes L_i bestimmt werden, um die Beobachterfehlerdynamik zu stabilisieren.

Das kann z.B. mittels Eigenwertvorgabe für $S_w - L_i a_{i0} Q_w$ und Koeffizientenvergleich gemacht werden.

Nichtinformierte Agenten

Bei nichtinformierten Agenten wird ein kooperativer Führungsbeobachter konstruiert, der dann durch diffusive Kopplung mit den anderen Agenten kommuniziert:

$$\dot{\hat{v}}_i = S_w \hat{v}_i + L \sum_{j=1}^N a_{ij} (\hat{w}_j - \hat{w}_i) \quad (2.85)$$

(Gemeinsame Beobachterrückführmatrix L .)

Hierfür wird der kooperative Führungsbeobachter entworfen. Dafür betrachtet man den globalen Beobachtungsfehler $e_i = v_0 - \hat{v}_i$:

$$\dot{e}_i = S_w e_i - \sum_{j=1}^N a_{ij} L Q_w (\hat{v}_j - \hat{v}_i) \quad (2.86)$$

$$= S_w e_i - \sum_{j=1}^N a_{ij} L Q_w (e_i - e_j) \quad (2.87)$$

Zur Bestimmung von L bildet man die Laplacematrix des erweiterten Graphen in Blockmatrixform:

$$L_{\bar{\mathcal{G}}} = \begin{bmatrix} 0 & 0_{1 \times M} & 0_{1 \times N-M} \\ L_{21} & L_{22} & 0_{M \times N-M} \\ 0_{N-M \times 1} & L_{32} & \mathbf{L}_{33} \end{bmatrix} \quad (2.88)$$

Dabei ist M die Anzahl informierter Agenten, N die Anzahl Agenten, und $N - M$ die Anzahl Nichtinformierter Agenten. Achtung: Das erfordert im Allgemeinen zuerst ein Umsortieren der Agenten, sodass an Stelle 0 der Führungsagent, dann die informierten Agenten und dann die nichtinformierten Agenten in der Reihenfolge vorkommen.

- Nullzeile: Keine Verbindungen zum Führungsagenten
- L_{21} : Verbindungen Führungsagent \rightarrow Informierte Agenten
- L_{22} : Verbindungen zwischen informierten Agenten
- $0_{M \times N-M}$: Keine Verbindungen Nichtinformierte Agenten \rightarrow Informierte Agenten
- $0_{N-M \times 1}$: Keine Verbindungen Führungsagent \rightarrow Nichtinformierte Agenten
- L_{32} : Verbindungen Informierte Agenten \rightarrow Nichtinformierte Agenten
- \mathbf{L}_{33} : Verbindungen zwischen nichtinformierten Agenten

Die Rückführmatrix L bestimmt sich dann als

$$L = \bar{c} \bar{P} Q_w^T \bar{R}^{-1} \quad (2.89)$$

mit

$$\bar{c} \geq \frac{1}{2 \min_i \operatorname{Re} \lambda_i(L_{33})} \quad (2.90)$$

und \bar{P} (symmetrisch) als Lösung der ARE

$$S_w \bar{P} + \bar{P} S_w^T - \bar{P} Q_w^T \bar{R}^{-1} Q_w \bar{P} + \bar{Q} = 0 \quad (2.91)$$

Wenn $\bar{\mathcal{G}}$ den Führungsagenten als Wurzelknoten enthält.

Fehlerzustandsrückführung mit Führungsgrößenaufschaltung

Dann wird eine Führungsgrößenaufschaltung konstruiert, die eine Zustandsrückführung K der Agentenzustände und eine Aufschaltung M des rekonstruierten Führungsagentenzustands enthält:

$$u_i = \underbrace{-K_i(x_i - \Pi_i v_0)}_{\text{Folgeregler}} + \underbrace{\Gamma_i v_0}_{\text{Steuerung}} \quad (2.92)$$

$$= -K_i x_i + (K_i \Pi_i + \Gamma_i) v_0 \quad (2.93)$$

$$= -K_i x_i + M_i v_0 \quad (2.94)$$

Dabei muss $A_i - B_i K_i$ Hurwitz Matrix sein (z.B. Eigenwertvorgabe) und es muss gelten:

$$\Pi_i S_w - A_i \Pi_i = B_i \Gamma_i \quad (2.95)$$

$$C_i \Pi_i = Q_w \quad (2.96)$$

Für die eindeutige Lösbarkeit davon ist Voraussetzung, dass keine „invariante Nullstelle“ von (C_i, A_i, B_i) mit einem Eigenwert von S_w übereinstimmt: Für alle Eigenwerte s von S_w muss gelten:

$$\det \begin{bmatrix} A_i - sI & B_i \\ C_i & 0 \end{bmatrix} \neq 0 \quad (2.97)$$

2.8 Schaltende Netze

Schaltende Netzze modellieren wir dadurch dass wir nicht den konstanten Graphen \mathcal{G} betrachten sondern den Graphen $\mathcal{G}_{\sigma(t)}$, wobei $\sigma(t)$ zu jedem Zeitpunkt den momentan aktiven Graphen aus einer endlichen Menge an Graphen auswählt.

Hier schränken wir $\sigma(t)$ noch soweit ein dass ein Graph mindestens für die Verweildauer τ aktiv bleibt, in dieser Zeit ändert sich $\sigma(t)$ also nicht. Das Schaltsignal $\sigma(t)$ wird als bekannt und deterministisch angenommen.

Der **Vereinigungsgraph** $\tilde{\mathcal{G}}$ beinhaltet die Kanten aller Graphen in der Graphenmenge. Die Kantengewichte werden dabei addiert.

Zur Synchronisierung wird ein kooperativer Führungsbeobachter genutzt, das Problem wird auf die leader-follower Zustandssynchronisierung des homogenen Führungsmodells zurückgeführt.

$$\dot{\hat{v}}_i = S_w \hat{v}_i + \mu \left(\sum_{j=1}^N a_{ij}(t)(\hat{v}_j - \hat{v}_i) + a_{i0}(t)(v_0 - \hat{v}_i) \right) \quad (2.98)$$

Hier wird eine gemeinsame Verstärkung $\mu > 0$ gewählt und informierte und Nichtinformierte Agenten gemeinsam betrachtet.

Für den Folgefehler $e_i = \hat{v}_i - v_0$ ergibt sich nach Aggregation

$$\dot{e} = (I_N \otimes S_w - \mu H_{\sigma(t)} \otimes I_{n_w})e \quad (2.99)$$

Die Folgefehlerdynamik ist stabil für alle $\mu > 0$ wenn der Vereinigungsgraph mit Agent 0 als einzigem Wurzelknoten zusammenhängend ist und $\text{Re } \lambda(S_w) \leq 0$.

Für die Bestimmung von μ kann z.B. Eigenwertvorgabe für die Beobachterfehlerdynamik gemacht werden, sodass eine bestimmte Stabilitätsreserve eingehalten wird. In der Übung wurde hier dann das Statische Kommunikationsnetz mit \tilde{H} betrachtet.

Die Eigenwerte sollten über die Schur-Zerlegung von \tilde{H} bestimmt werden:

$$T^{-1} \tilde{H} T = \begin{bmatrix} \lambda_1(\tilde{H}) & & * \\ & \ddots & \\ 0 & & \lambda_N(\tilde{H}) \end{bmatrix} = Q \quad (2.100)$$

Dafür wird eine Ähnlichkeitstrafo mit $(T \otimes I_{n_w})$ gemacht:

$$(T^{-1} \otimes I_{n_w})(I_N \otimes S_w - \mu \tilde{H} \otimes I_{n_w})(T \otimes I_{n_w}) \quad (2.101)$$

$$= I_N \otimes S_w - \mu Q \otimes I_{n_w} \quad (2.102)$$

$$= \begin{bmatrix} S_w - \mu \lambda_1(\tilde{H}) I_{n_w} & & * \\ & \ddots & \\ 0 & & S_w - \mu \lambda_N(\tilde{H}) I_{n_w} \end{bmatrix} \quad (2.103)$$

Mit ein paar Rechenregeln für Determinanten und scharfem Nachdenken stellt man nun fest dass die kombinierten Eigenwerte der Diagonalelemente der Eigenwerte der ursprünglichen Beobachterfehlerdynamik entsprechen.

Das eben bestimmte μ kann nun auch für den schaltenden Graphen verwendet werden. Die Stabilitätsreserve wird aber nicht unbedingt eingehalten, da bei der Bestimmung eine abweichende Topologie genutzt wurde. Exponentielle Stabilität und positive Stabilitätsreserve ist aber dennoch auch für den schaltenden Graphen garantiert.