Aufgabe: Sechs Flugzeuge mit unterschiedlichen Geschwindigkeiten und Abständen fliegen in einer Warteschleife im Kreis. Lösen Sie die Aufgabenstellung, daß alle Flugzeuge mit gleicher Geschwindigkeit und in gleichen Abständen zueinander im Kreis fliegen sollen, durch ein neuronales Netzwerk.

Lösung: Die Anordnung unserer Flugzeuge im *i*ten Zeitintervall sei beispielsweise die in Abb. 1 angegebene. Ohne Beschränkung der Allgemeinheit sei $\vec{r}_6 = R \cdot \vec{e}_x$, wobei R der Kreisradius ist.

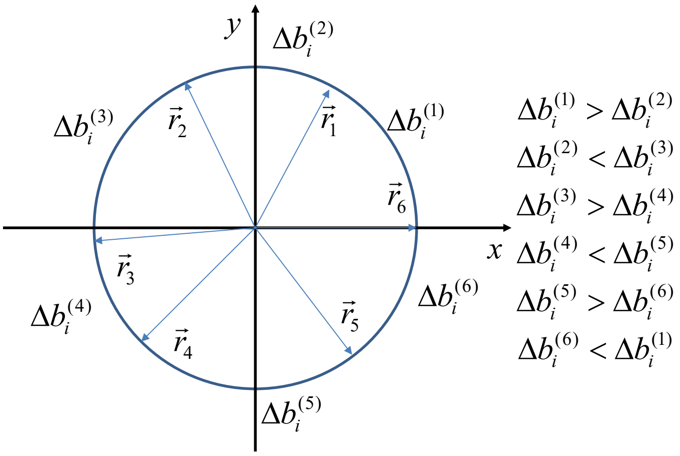


Abbildung 1. Sechs Flugzeuge in einer kreisförmigen Warteschleife zum Zeitpunkt *i*

Die Winkel $\Delta\varphi_i^{(n)}$ für $n \in \{1, 2, \ldots, 6\}$ sind mit den Bogenlängen $\Delta b_i^{(n)}$ durch folgende Beziehung verknüpft:

$$\Delta\varphi_i^{(n)} = \frac{\Delta b_i^{(n)}}{R}.$$

Aufgrund der Anfangsbedingungen ist $\varphi_i^{(7)} \equiv \varphi_i^{(1)}$. Sollen die Flugzeuge ihre Abstände zueinander beibehalten, müßten sie mit stets gleichbleibender Geschwindigkeit

$$v_i^{(n)} = \frac{\Delta b_i^{(n)}}{\Delta t}$$

fliegen, wobei Δt ein für alle Flugzeuge geltendes Zeitintervall ist, in dem die jeweiligen Bögen mit den entsprechenden Geschwindigkeiten durchflogen werden müssen. Dann wäre die mittlere Geschwindigkeit $\bar{v}$ gegeben durch

$$\bar{v} = \frac{1}{6}\sum_{n=1}^{6} v_i^{(n)} = \frac{1}{6\Delta t}\sum_{i=1}^{6} \Delta b_i^{(n)} = \frac{\pi}{3}\frac{R}{\Delta t}.$$

Da die Flugzeuge aber in der Regel entweder mit einer zu großen oder zu geringen Geschwindigkeit in die Warteschleife einfliegen, kann es zu gefährlichen Annäherungen kommen. Um diese unterschiedlichen Abstände auszugleichen, müssen die Geschwindigkeiten folglich systematisch an die mittlere Geschwindigkeit herangeführt werden und die Abstände auf einen mittleren Abstand $\bar{b} = \bar{v}\Delta t$ gebracht werden. Dazu sind zwei Bedingungen zu erfüllen. Der Abstand zum Vordermann muß gleich dem Abstand zum Hintermann werden, und für die Differenzgeschwindigkeit gilt das gleiche. Für die Abstände bedeutet das, daß gelten muß:

$$\Delta b_i^{(n+1)} = \Delta b_i^{(n)}.$$

Wenn für das nächste Meßintervall stets der Geschwindigkeitsmittelwert

$$v_{i+1}^{(n)} = \frac{1}{2}\left(v_i^{(n)} + v_i^{(n+1)}\right)$$

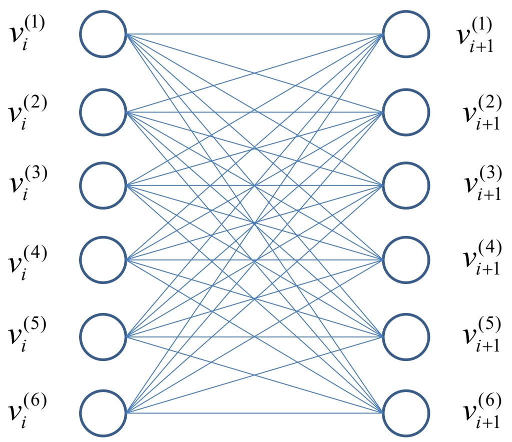
zweier aufeinanderfolgender Flugzeuge verwendet wird, nähern sich auch die Abstände der Flugzeuge untereinander dem mittleren Abstand zwischen zwei Flugzeugen an, d.h. wegen

$$\Delta\varphi_{i+1}^{(n)} = v_{i+1}^{(n)}\frac{\Delta t}{R} = \frac{1}{2}\left(v_i^{(n)} + v_i^{(n+1)}\right)\frac{\Delta t}{R}$$

ergibt sich der zur Geschwindigkeit analoge Ausdruck

$$\Delta\varphi_{i+1}^{(n)} = \frac{1}{2}\left(\Delta\varphi_i^{(n)} + \Delta\varphi_i^{(n+1)}\right).$$

Das dazu äquivalente Netzwerk ist in Abb. 2 dargestellt.



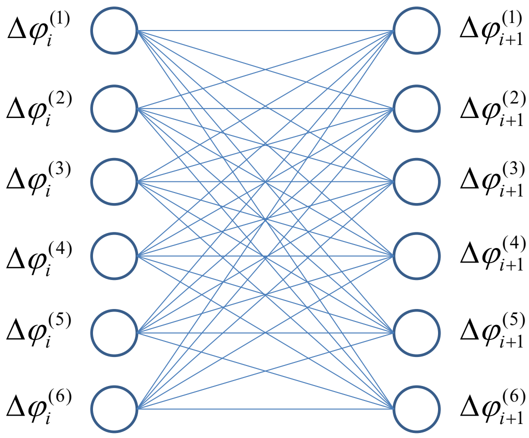
$$v_{i+1}^{(n)} = \frac{1}{2}\left(v_i^{(n)} + v_i^{(n+1)}\right)$$

Abbildung 2. Neuronales Netzwerk zur Regelung von 6 Geschwindigkeiten in einer Warteschleife

Die Ausgangsneuronen sollen nun direkt in die Geschwindigkeitsregelung der jeweiligen Flugzeuge eingreifen und ihre Werte rekursiv iterieren. Die Geschwindigkeitsberechnung obliegt dabei der zentralen Flugverkehrskontrollstelle, welche in Abständen die GPS-Positionen und Fluggeschwindigkeiten aller Teilnehmer erhält und umgekehrt die rekursiven Werte an die Eingangsneuronen der einzelnen Flugzeuge zurückgibt. Die Methode hat den Vorteil, daß nicht jedes teilnehmende Flugzeug mit einem eigenen Flugregler ausgestattet werden muß, sondern die Informationen für die Eingangsneuronen von zentraler Stelle aus kommen. Man sieht also, daß die Regelwerte

$$v_{i+1}^{(n)} = v_i^{(n)} + \frac{1}{2}\left(v_i^{(n+1)} - v_i^{(n)}\right)$$

sich immer weniger ändern, je mehr sich die Geschwindigkeiten aller Nachbarn untereinander angeglichen haben. Das setzt eine einwandfrei funktionierende Kommunikation voraus. Insgesamt haben wir jeweils 6 Ein- und Ausgangsneuronen, 64 verschiedene Bitmuster und 36 neuronale Verbindungen.



$$\Delta\varphi_{i+1}^{(n)} = \frac{1}{2}\left(\Delta\varphi_i^{(n)} + \Delta\varphi_i^{(n+1)}\right)$$

Abbildung 3. Neuronales Netzwerk zur Regelung von 6 Abständen in einer Warteschleife

Das rekursive neuronale Netzwerk können wir entweder in Matrixdarstellung angeben:

$$\begin{pmatrix} v_{i+1}^{(1)} \\ v_{i+1}^{(2)} \\ v_{i+1}^{(2)} \\ v_{i+1}^{(4)} \\ v_{i+1}^{(5)} \\ v_{i+1}^{(6)} \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 1/2 & 0 & 0 & 0 & 0 & 1/2 \end{pmatrix} \begin{pmatrix} v_i^{(1)} \\ v_i^{(2)} \\ v_i^{(3)} \\ v_i^{(4)} \\ v_i^{(5)} \\ v_i^{(6)} \end{pmatrix},$$

wobei die Gewichte bereits festgelegt sind, oder wir beschreiben es als selbstorganisierende Karte der Form

$$\begin{pmatrix} \Delta\varphi_{i+1}^{(1)} \\ \Delta\varphi_{i+1}^{(2)} \\ \Delta\varphi_{i+1}^{(3)} \\ \Delta\varphi_{i+1}^{(4)} \\ \Delta\varphi_{i+1}^{(5)} \\ \Delta\varphi_{i+1}^{(6)} \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 1/2 & 0 & 0 & 0 & 0 & 1/2 \end{pmatrix} \begin{pmatrix} \Delta\varphi_i^{(1)} \\ \Delta\varphi_i^{(2)} \\ \Delta\varphi_i^{(3)} \\ \Delta\varphi_i^{(4)} \\ \Delta\varphi_i^{(5)} \\ \Delta\varphi_i^{(6)} \end{pmatrix}.$$

Dabei ist

$$\mathbf{W} = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 1/2 & 0 & 0 & 0 & 0 & 1/2 \end{pmatrix}$$

unsere Gewichtsmatrix. Die von null verschiedenen Gewichte in diesem neuronalen Netz sind besonders einfach und haben alle den Wert 0,5.

Zum Abschluß geben wir noch die Softwarelösung für dieses Netzwerk an. Wir konnten somit zeigen, daß ein einfaches Mittelungsverfahren zu einer Selbstorganisation fähig ist. Unwillkürlich machen die Piloten genau das gleiche, sie halten die Abstände per Augenmaß, um nicht zu kollidieren. Wenn wir es allerdings mit autonomen oder unbemannten Flugzeugen zu tun haben, ist ein entsprechender Algorithmus, der die Organisation stellvertretend übernimmt, unverzichtbar.

Anhang

```
% Kreisstaffelung
clear all

% Kreisradius in m
R = 13200;

% Anfangswinkel phi(n,i) zwischen zwei aufeinanderfolgenden Flugzeugen
phi(1,1) = 65;
phi(2,1) = 59;
phi(3,1) = 60;
phi(4,1) = 56;
phi(5,1) = 63;
phi(6,1) = 57;

umfang = phi(1,1) + phi(2,1) + phi(3,1) + phi(4,1) + phi(5,1) + phi(6,1);

% Umrechnung ins Bogenmaß
for n=1:6
    x(n,1) = phi(n,1)/180*pi;
end

kreisumfang = x(1,1) + x(2,1) + x(3,1) + x(4,1) + x(5,1) + x(6,1);

% Bogenabstand zwischen zwei Flugzeugen in m
for n=1:6
    b(n,1) = R*x(n,1);
end

bogenumfang = b(1,1) + b(2,1) + b(3,1) + b(4,1) + b(5,1) + b(6,1);

% Anfangsgeschwindigkeiten der Flugzeuge in m/s
v(1,1) = b(6,1)/60;
v(2,1) = b(5,1)/60;
v(3,1) = b(4,1)/60;
v(4,1) = b(3,1)/60;
v(5,1) = b(2,1)/60;
v(6,1) = b(1,1)/60;
v(7,1) = v(1,1);

v_m = (v(1,1) + v(2,1) + v(3,1) + v(4,1) + v(5,1) + v(6,1))/6

m = 60;

for i=1:60
    v(7,i) = v(1,i);
    for n=1:6
        v(n,i+1) = (v(n,i) + v(n+1,i))/2;
        b(n,i+1) = v(n,i+1)*60;
        x(n,i+1) = b(n,i+1)/R;
        phi(n,i+1) = x(n,i+1)/pi*180;
    end
end

i = 17;
disp('i = ')
disp(i)
```



```
phi_1 = phi(1,i);
v1 = v(1,i);
b1 = b(1,i);
x1 = x(1,i);
q1 = (phi_1-60)^2;
qv1 = (v1-v_m)^2;

disp('   phi_1       x1        b61           v1')
A1 = [phi_1, x1, b1/1000, v1];
disp(A1)

phi_2 = phi(2,i);
v2 = v(2,i);
b2 = b(2,i);
x2 = x(2,i);
q2 = (phi_2-60)^2;
qv2 = (v2-v_m)^2;

disp('   phi_2       x2        b12           v2')
A2 = [phi_2, x2, b2/1000, v2];
disp(A2)

phi_3 = phi(3,i);
v3 = v(3,i);
b3 = b(3,i);
x3 = x(3,i);
q3 = (phi_3-60)^2;
qv3 = (v3-v_m)^2;

disp('   phi_3       x3        b23           v3')
A3 = [phi_3, x3, b3/1000, v3];
disp(A3)

phi_4 = phi(4,i);
v4 = v(4,i);
b4 = b(4,i);
x4 = x(4,i);
q4 = (phi_4-60)^2;
qv4 = (v4-v_m)^2;

disp('   phi_4       x4        b34           v4')
A4 = [phi_4, x4, b4/1000, v4];
disp(A4)

phi_5 = phi(5,i);
v5 = v(5,i);
b5 = b(5,i);
x5 = x(5,i);
q5 = (phi_5-60)^2;
qv5 = (v5-v_m)^2;

disp('   phi_5       x5        b45           v5')
A5 = [phi_5, x5, b5/1000, v5];
disp(A5)

phi_6 = phi(6,i);
v6 = v(6,i);
b6 = b(6,i);
x6 = x(6,i);
```



```
q6 = (phi_6-60)^2;
qv6 = (v6-v_m)^2;

disp('   phi_6      x6       b56          v6')
A6 = [phi_6, x5, b6/1000, v6];
disp(A6)

q = q1 + q2 + q3 + q4 + q5 + q6;
rms_phi = sqrt((q1 + q2 + q3 + q4 + q5 + q6)/6)

qv = qv1 + qv2 + qv3 + qv4 + qv5 + qv6;
rms_v = sqrt(qv/6)



>> kreisstaffelung

v_m =

  230.3835

i =

    60

   phi_1    x1          b61          v1

   60.0004    1.0472   13.8231  230.3848

   phi_2    x2          b12          v2

   60.0001    1.0472   13.8230  230.3837

   phi_3    x3          b23          v3

   59.9997    1.0472   13.8229  230.3823

   phi_4    x4          b34          v4

   59.9996    1.0472   13.8229  230.3821

   phi_5    x5          b45          v5

   59.9999    1.0472   13.8230  230.3832

   phi_6    x6          b56          v6

   60.0003    1.0472   13.8231  230.3846

rms_phi =

   2.7061e-04

rms_v =

    0.0010
```

