Beispiel für die Implementierung eines zeitdiskreten Zustandsraummodells in SIMULINK als Matlab Function oder Level 2 Matlab S-Function

T. Schauer

7. Mai 2020

1 Systembeschreibung

$$\begin{aligned} z_1(k+1) &= 0,4z_1(k) + 0,2z_2(k) \\ z_2(k+1) &= bz_1(k) + a(k)z_2(k) + u(k) \\ y(k) &= 0,6z_1(k) + 0,2z_2(k) + e(k) \end{aligned}$$

Simulation startet bei $k = 0 \rightarrow$ Initialwerte: $z_1(0)$ und $z_2(0)$; Parameter $a(k)$ (zeitvariant) und $b = 0,8$: Abtastperiode $\Delta = 1$.

2 init.m – Skript

```
b=0.8;
Delta=1;
z0=[0;0];
```

3 Zustandsraummodell als Level 2 MATLAB S-Function

```
function model_dsc(block)
% Level-2 MATLAB file S-Function
% Discrete-time state-space model with u and
% a (time-varying parameter) as inputs
  setup(block)

%endfunction

function setup(block)

% Register number of dialog parameters
  block.NumDialogPrms = 3; %Delta, z0, b
```

```
13    %% Register number of input and output ports
14    block.NumInputPorts  = 2;
15    block.NumOutputPorts = 2;
16
17    %% Setup functional port properties to dynamically
18    %% inherited.
19    block.SetPreCompInpPortInfoToDynamic;
20    block.SetPreCompOutPortInfoToDynamic;
21    %u(k)
22    block.InputPort(1).Dimensions        = 1;
23    block.InputPort(1).DirectFeedthrough = false;
24    block.InputPort(1).SamplingMode='Sample';
25    %a(k)
26    block.InputPort(2).Dimensions        = 1;
27    block.InputPort(2).DirectFeedthrough = false;
28    block.InputPort(2).SamplingMode='Sample';
29    %z(k)
30    block.OutputPort(1).Dimensions       = 2;
31    block.OutputPort(1).SamplingMode='Sample';
32    %y(k)
33    block.OutputPort(2).Dimensions       = 1;
34    block.OutputPort(2).SamplingMode='Sample';
35
36    %% Set block sample time to inherited
37    Delta=block.DialogPrm(1).data;
38    block.SampleTimes = [Delta 0];
39
40    %% Set the block simStateCompliance to default
41    %% (i.e., same as a built-in block)
42    block.SimStateCompliance = 'DefaultSimState';
43
44    %% Register methods
45    block.RegBlockMethod('PostPropagationSetup',    @DoPostPropSetup);
46    block.RegBlockMethod('InitializeConditions',    @InitConditions);
47    block.RegBlockMethod('Outputs',                 @Output);
48    block.RegBlockMethod('Update',                  @Update);
49
50  %endfunction
51
52  function DoPostPropSetup(block)
53
54    %% Setup Dwork
55    %state z
56    block.NumDworks = 1;
57    block.Dwork(1).Name = 'z';
58    block.Dwork(1).Dimensions      = 2;
59    block.Dwork(1).DatatypeID      = 0;
60    block.Dwork(1).Complexity      = 'Real';
61    block.Dwork(1).UsedAsDiscState = true;
```

```
62
63 %endfunction
64
65 function InitConditions(block)
66
67   %% Initialize Dwork
68   %set z(0)
69    z_0=block.DialogPrm(2).data;
70    block.Dwork(1).Data = z_0;
71
72 %endfunction
73
74 function Output(block)
75   %calculate y(k)
76    z1_k=block.Dwork(1).Data(1);
77    z2_k=block.Dwork(1).Data(2);
78    block.OutputPort(1).Data = block.Dwork(1).Data;
79    block.OutputPort(2).Data = 0.6*z1_k+0.2*z2_k+0.4*randn(1);
80
81 %endfunction
82
83 function Update(block)
84   %calculate z(k+1) and store it
85    z1_k=block.Dwork(1).Data(1);
86    z2_k=block.Dwork(1).Data(2);
87    b = block.DialogPrm(3).data;
88    u_k=block.InputPort(1).Data;
89    a_k=block.InputPort(2).Data;
90    z1_kp1=0.4*z1_k+0.2*z2_k;
91    z2_kp1=b*z1_k+a_k*z2_k+u_k;
92    block.Dwork(1).Data = [z1_kp1;z2_kp1];
```

4 Zustandsraummodell als MATLAB Function

```
1 function [z_k_out,y_k] = fcn(u_k, a_k, z0, b)
2
3 %initialize states (variables to be stored)
4 persistent  z_k
5 if isempty(z_k)
6     z_k= z0;
7 end
8 %Update states and calulate output
9 z_kp1=[0.4*z_k(1) + 0.2*z_k(2);
10        b*z_k(1) + a_k*z_k(2)+u_k];
11 y_k    = 0.6*z_k(1)+0.2*z_k(2)+0.4*randn(1);
12 z_k_out = z_k;
13 %Store states
```

```
14  z_k = z_kp1;
```

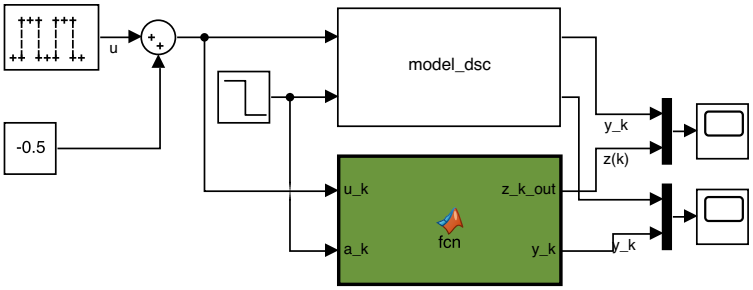

Abbildung 1: SIMULINK-Diagramm mit dem Modell (zwei Formen der Implementierung).