Topic 1

Task 2

b)

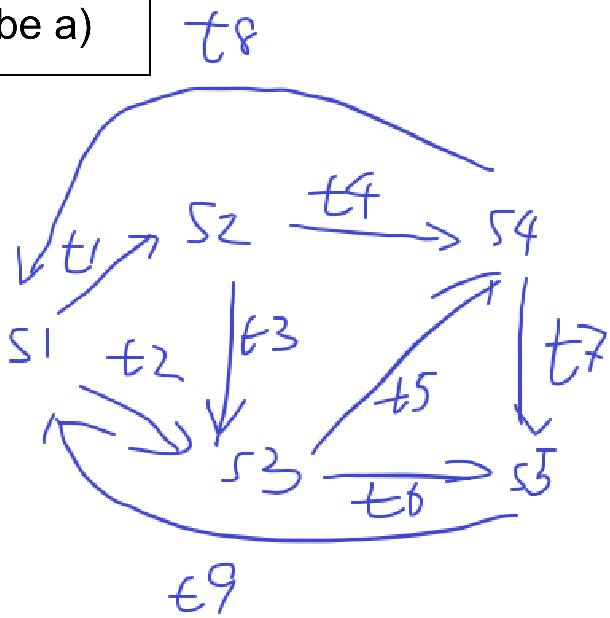
s1	s2	s3	s4	s5	t1	t2	t3
X	X		X		X		
		X	X			X	
				X			X

c)

s1	s2	s3	s4	s5	t1	t2	t3
X	X			X	X		X
		X		X			X
		X	X			X	
				X			X

Task 3

Aufgabe a)



reachability graph

Aufgabe b)

incidence matrix									
	t1	t2	t3	t4	t5	t6	t7	t8	t9
S1	-1	-1	0	0	0	0	0	1	1
S2	1	0	-1	-1	0	0	0	0	0
S3	0	1	1	0	-1	-1	0	0	0
S4	0	0	0	1	1	0	-1	-1	0
S5	0	0	0	0	0	1	1	0	-1

Aufgabe 3)_First switching sequence

```
% Day1 Task3 c) First switching sequence Matlab code

% initial marker vector

m0=[1;0;0;0;0];
```

```
% incidence matrix
C=[-1 -1 0 0 0 0 0 1 1; 1 0 -1 -1 0 0 0 0 0; 0 1 1 0 -1 -1 0 0 0; 0 0 0 1 1 0
-1 -1 0; 0 0 0 0 0 1 1 0 -1];
% parikh vector
v{1}=[1 0 0 0 0 0 0 0 0];
v{2}=[0 0 1 0 0 0 0 0 0];
v{3}=[0 0 0 0 1 0 0 0 0];
v{4}=[0 0 0 0 0 0 1 0 0];
v{5}=[0 0 0 0 0 0 0 0 1];
v{6}=[0 1 0 0 0 0 0 0 0];
v{7}=[0 0 0 0 0 1 0 0 0];
v{8}=[0 0 0 0 0 0 0 0 1];
v{9}=[1 0 0 0 0 0 0 0 0];
v{10}=[0 0 0 1 0 0 0 0 0];
% real in-between markt vector
m{1}=[0;1;0;0;0];
m{2}=[0;0;1;0;0];
m{3}=[0;0;0;1;0];
m{4}=[0;0;0;0;1];
m{5}=[1;0;0;0;0];
m{6}=[0;0;1;0;0];
m{7}=[0;0;0;0;1];
```

```
m{8}=[1;0;0;0;0];
m{9}=[0;1;0;0;0];
m{10}=[0;0;0;1;0];
% calculation of brand occupancy base on parikh vector
for i=1:10
    m_cal{i}=m0+C*transpose(v{i});
    m0=m_cal{i};
end
% compare calculated mark vectors with real mark vectors and print the
output
for i=1:10
    fprintf('%i-th calculated mark vector is [%i %i %i %i %i]\n', i,
transpose(m_cal{i}));
    fprintf('%i-th real mark vector is [%i %i %i %i %i]\n', i,
transpose(m{i}));
    if m_cal{i}==m{i}
        disp('true')
    else
        disp('false')
    end
    disp(' ')
end
```

% **Result: By running this program there is no problem with the mask vector result under the first switching sequence.**

Aufgabe 3)_Second switching sequence

```
% Day1 Task3 c) Second switching sequence Matlab code
% initial marker vector
m0=[1;0;0;0;0];
% incidence matrix
C=[-1 -1 0 0 0 0 0 1 1; 1 0 -1 -1 0 0 0 0 0; 0 1 1 0 -1 -1 0 0 0; 0 0 0 1 1 0 -1 -1 0; 0 0 0 0 0 1 1 0 -1];
% parikh vector
v{1}=[1 0 0 0 0 0 0 0 0];
v{2}=[0 0 1 0 0 0 0 0 0];
v{3}=[0 0 0 0 1 0 0 0 0];
v{4}=[0 0 0 0 0 1 0 0 0];
v{5}=[0 0 0 0 0 0 1 0 0];
v{6}=[0 0 0 0 0 0 0 0 1];
v{7}=[0 0 0 0 0 0 0 0 1];
v{8}=[0 1 0 0 0 0 0 0 0];
% real in-between mark vector
m{1}=[0;1;0;0;0];
m{2}=[0;0;1;0;0];
```

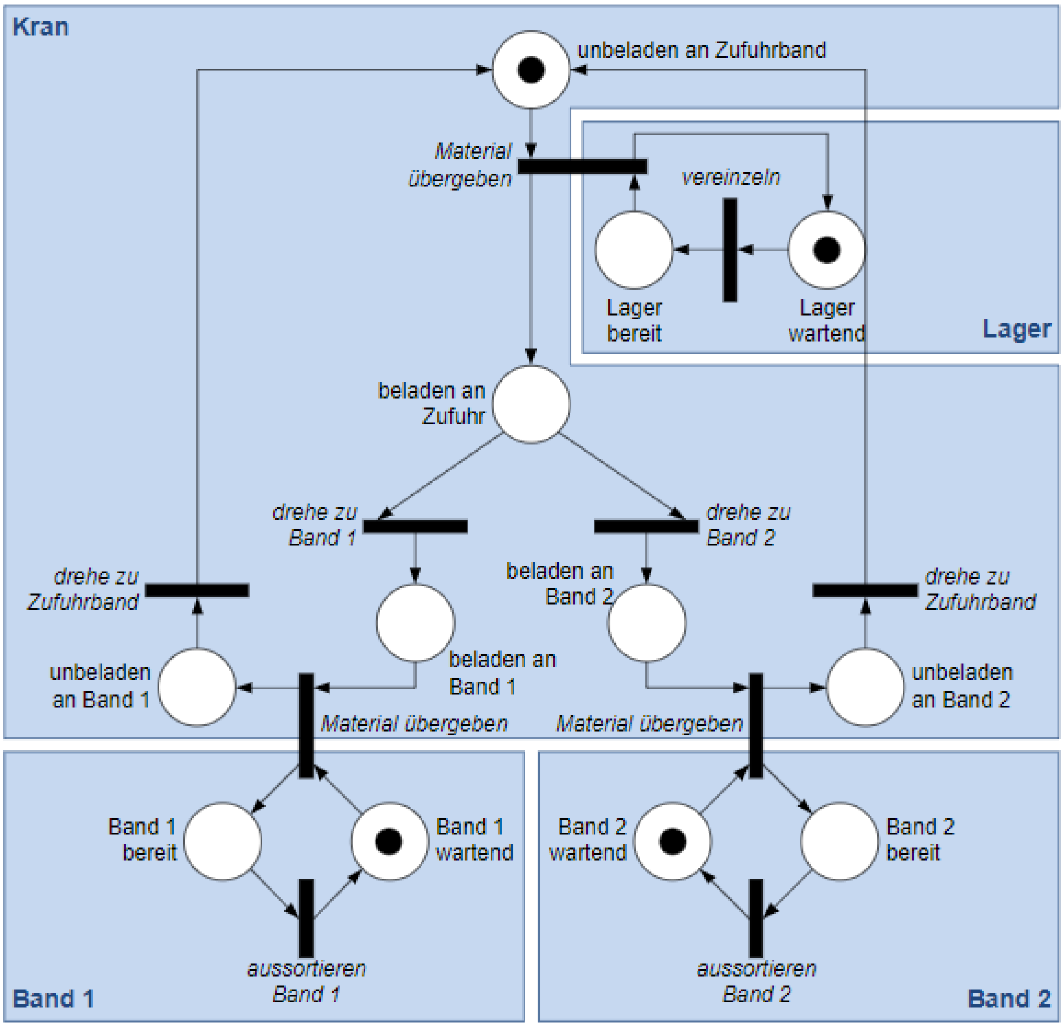
```
m{3}=[0;0;0;1;0];

m{4}=[0;0;0;1;0];

m{5}=[0;0;0;0;1];

m{6}=[1;0;0;0;0];

m{7}=[1;0;0;0;0];

m{8}=[0;0;1;0;0];

% calculation of brand occupancy base on parikh vector

for i=1:8

    m_cal{i}=m0+C*transpose(v{i});

    m0=m_cal{i};

end

% compare calculated mark vectors with real mark vectors and print the
output

for i=1:8

    fprintf('%i-th calculated mark vector is [%i %i %i %i %i]\n', i,
transpose(m_cal{i}));

    fprintf('%i-th real mark vector is [%i %i %i %i %i]\n', i,
transpose(m{i}));

    if m_cal{i}==m{i}

        disp('true')

    else

        disp('false')
```

```
    end
    disp(' ')
end
% Result: By running this programm a false value appears since the
4-th transition (...t6...). The value of calculated mark of s3 is at this
moment -1, what makes no sence. The reason for this error is that
there is only one s3 mark. It can't activate t5 and t6 repeatedly.
```

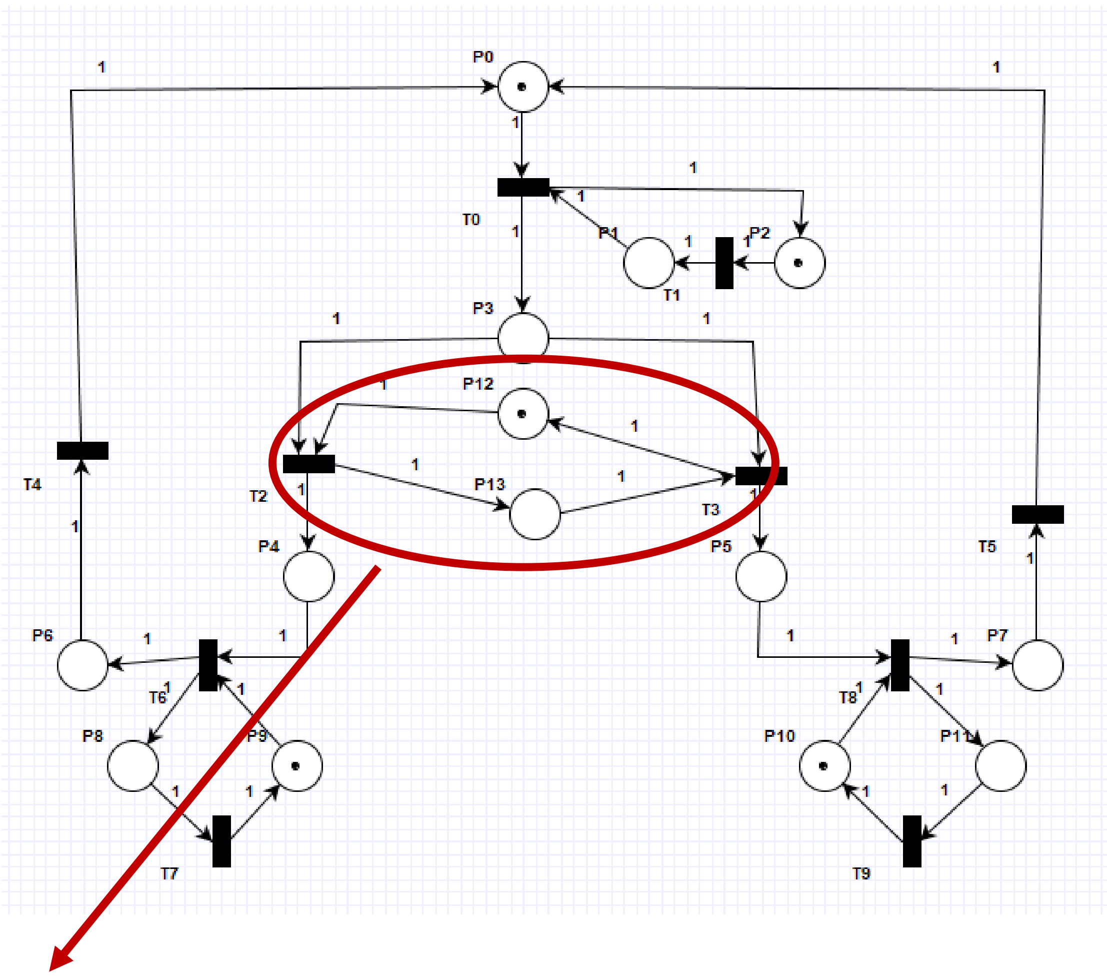
Aufgabe d)

Reproducibility: Yes. The initial marker S1 is accessible from transition t8 or t9.

Task 4

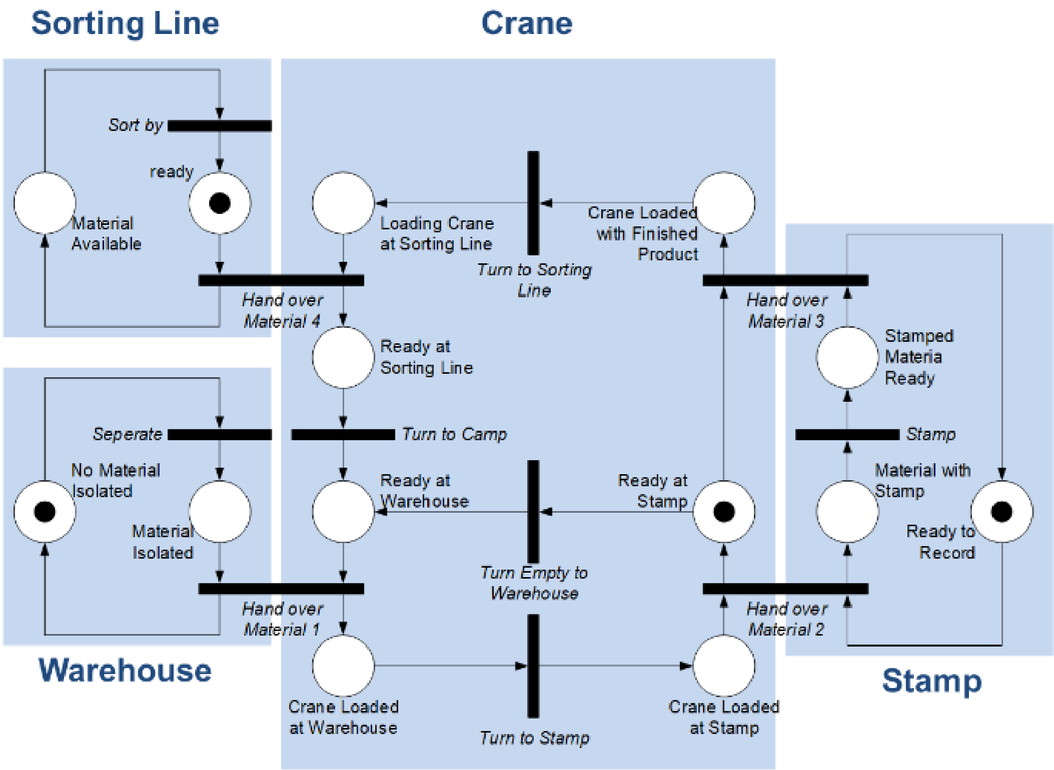


Error description: When transfering the mark from the state „beladen an Zufuhr“ to the following transition („drehe zu Band 1“ or „drehe zu Band 2“), the possibility of the mark being assigned to left transition (Band 1) and right transition (Band 2) is random, rather than alternate.



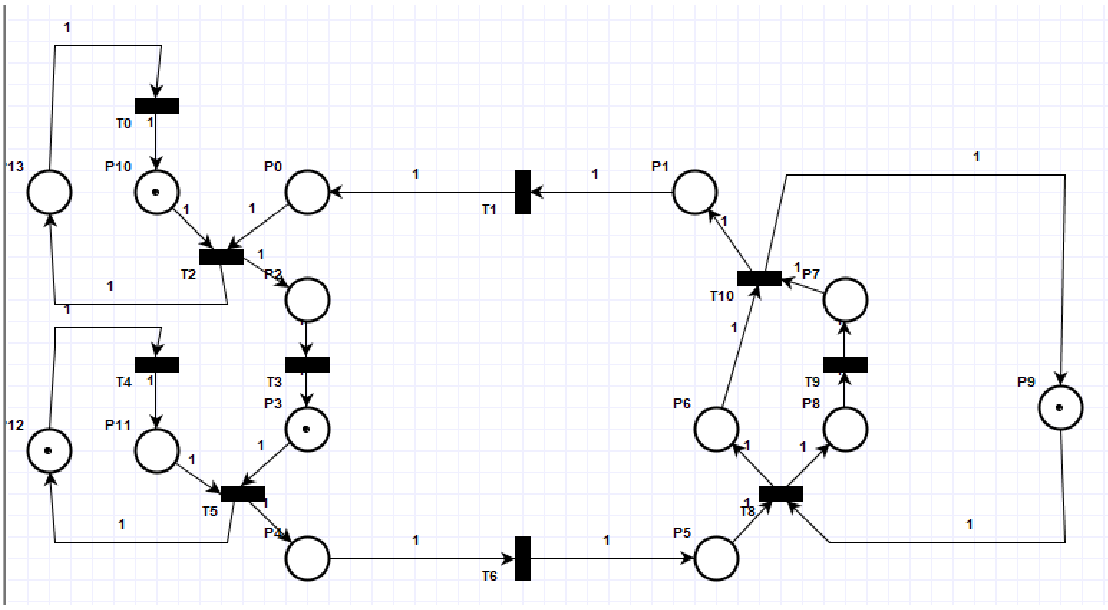
<u>Modification proposal</u>: In order to enable transitions „drehe zu Band 1“ and „drehe zu Band 2“ to be activated alternately, two control states (p12 and p13) are added between these two transitions. The two control states share a mark. When the transition corresponding to the control state with mark is activated, the mark will be transfered to another control state to make sure that, the another transition will be activated at the next loop.

Task 5



Error discription: From the given initial state of the whole petri net a serie of processes at Crane will be implemented: „Ready at Stamp“→“Turn Empty to Warehouse“ →“Ready at Warenhouse“ →“Hand over Material 1“ →“Crane Loaded at Warehouse“ →“Turn to Stamp“ →“Crane Loaded at Stamp“. After this process is implemented twice, the situation in the above screenshot will appear. Petri Net at this situation is dead und cannot be implemented continued.

Physical significance: When the mark in the crane returns to the „ready at stamp“ state for the first time, after the material is stamped, crane should remain in the current position until the material is pushed to the sorting line and then return to the warehouse during the transition „Turn to Camp“, but under the current petri net the crane will return to the warehouse directly during the transition “Turn Empty to Warenhouse“, what cause the stamped materials to fail to be transfered to the sorting line.



Modification proposal: After the stamping („Ready at Stamp“) the material should always immediately following waiting tob e pushed to sorting line. So the transition „Turn Empty to Warehouse“ makes actually no sense. And the initial mark in the Crane should be at the state „Ready at Warehouse“. The first thing that the crane should do is to get Material from warehouse.

Task 6

a)

Combined incidence matrix *I*

	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9
P0	-1	0	-1	0	0	0	0	0	0	0
P1	1	-1	0	0	0	0	0	0	0	0
P2	0	1	0	1	-1	-1	0	0	0	0
P3	0	0	0	0	0	1	0	1	0	0
P4	0	0	0	0	0	0	1	-1	0	0
P5	0	0	1	-1	1	0	-1	0	1	-1
P6	0	0	0	0	0	0	0	0	-1	1

b)

cost vector = [0.5 0.8 0.6 0.25 0.25 0.8 0.6 0.5 0.25 0.25]

c)

start markers m0 = [1; 0; 0; 0; 0; 0; 0]

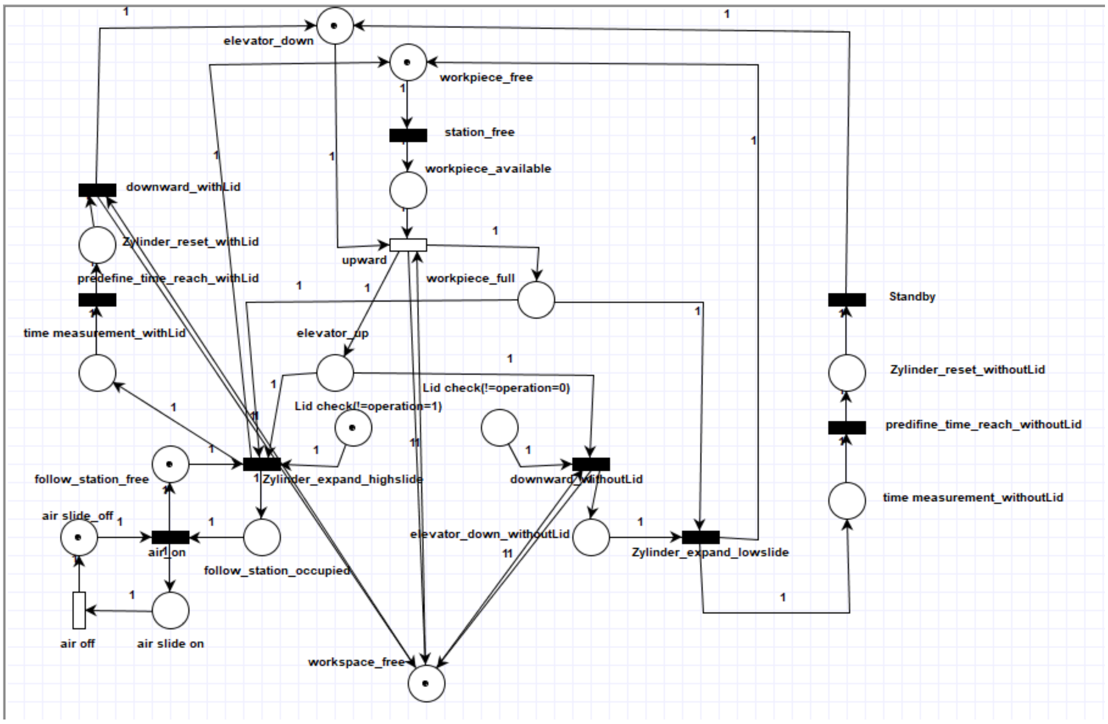
end markers m' = [0; 0; 0; 1; 0; 0; 0]

d)

optimal sequence t_opt = [0;0;1;1;0;1;0;0;0;0]

cost in total: cost vector .* t_opt = 1.65

Practice block



Petri net state space analysis results

Bounded	true
Safe	true
Deadlock	true

Shortest path to deadlock: station_free upward Zylinder_expand_highslide air_on predefine_time_reach_withLid downward_withLid station_free air off upward

Characteristics analysis: By using the analysis tool in PIPE, it can be seen that the petri net of Festo system that I designed by myself has deadlock. The reason for the deadlock is obvious: the two states in the middle part of the figure (Lid check(!=operation=0/1)) are pre-assigned and can only be executed once, what makes the entire petri net weak alive.

The reason for this design is that, in the current simulation, the actual properties of the real workpieces (black/non-black, height) are not yet known, so i decided to introduce ahypothetical measurement results (Lid check(!=operation=0 or 1)) in these two states.

Pros and cons of the Petrinet-Modeling:

Pros: Compared with Code programming, Petrinet's graphical design language can make the structure of the entire system more intuitive. After the model design is completed, people can also quickly analyze the defects of the current design through the analysis tool in PIPE. By using the „Animation Model" the understanding of the entire simulation process is also much faster and more interesting than reading the codes.

Cons: When there are too many components in the entire system, especially when the relationship between the components is particularly complicated, there will be many arrows overlapping each other, which may make the structure of the entire system no longer intuitive. In addition, if there are many subsystems or components with the same function in the system, they should be able to be combined to make the entire structure more concise.