Asset Management: Condition-Based Maintenance

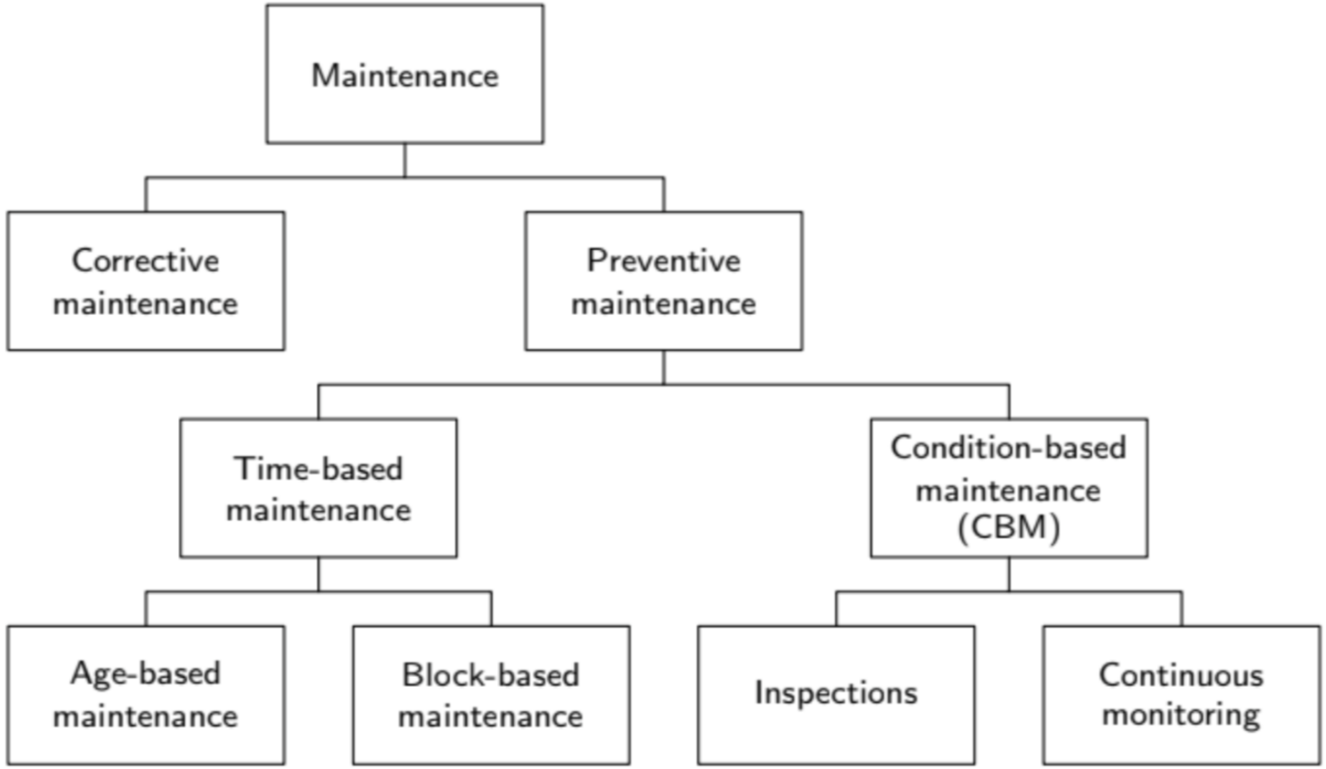
Luuk Pentinga MSc
l.pentinga@rug.nl

Contents

- Recap
- Condition monitoring
- Optimizing condition-based maintenance
- Challenges with CBM
- EQUANS

Recap

Maintenance strategies for single-component systems



Time-based maintenance

Plan preventive maintenance based on historical failure data.

Pros:

+ Data is fairly simple to collect.

+ Easy to plan, we always know when to maintain far in advance.

Cons:

- Relies on “averages”. We do not respond to slower or faster deterioration.

Condition-based maintenance

Using condition information may resolve these problems:

Inspections or sensor data may tell us something about the state of the machine.

If there is a lot of damage -> Maintain early.
If there is little damage -> Wait.

Of course, it should be possible and economically feasible.

Condition monitoring

Condition monitoring

Condition information is gathered in one of two ways:

- Continuous monitoring:
  - Degradation information is gathered continuously through sensors. (Vibrations, heat, oil quality, …)
  - Signals are received at certain degradation levels.
- Inspections
  - Periodically or aperiodically.

Remember that monitoring systems need to be invested in early.

Degradation modelling

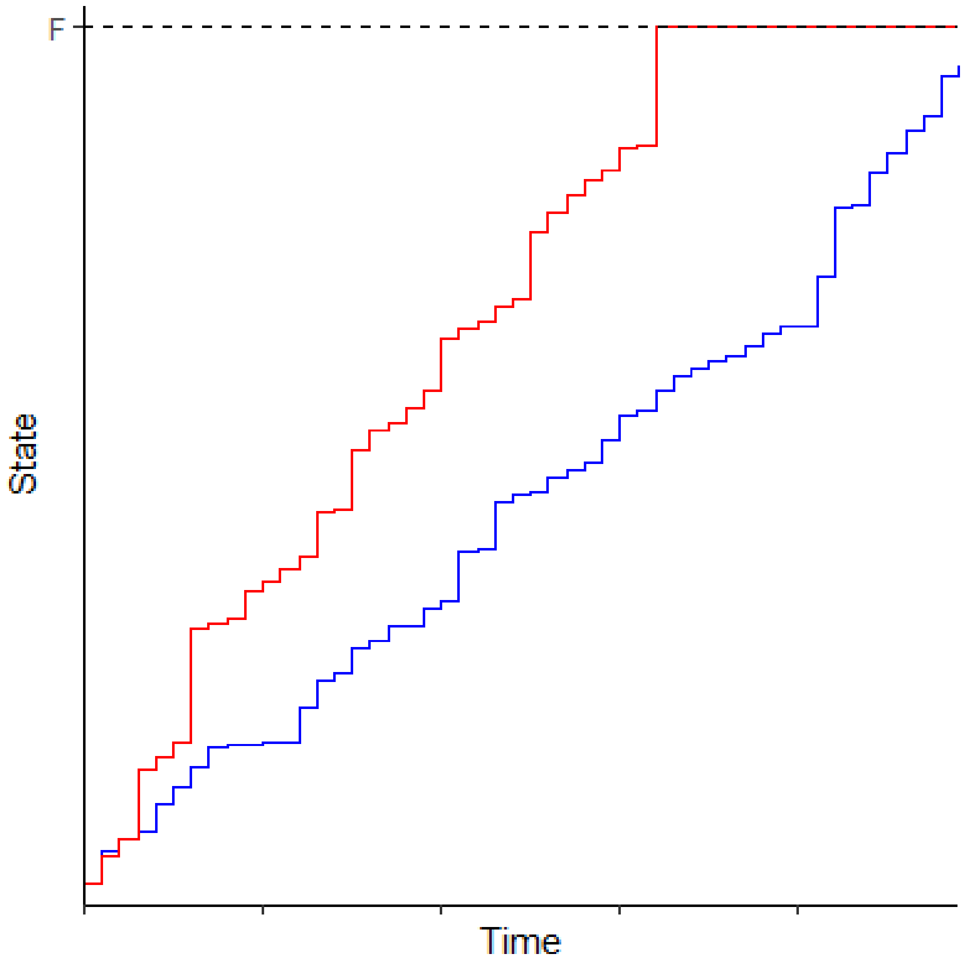
Degradation modelling often relies on two parts:

1. The **state** of the machine: a number explaining the condition of the machine, higher implies more damage.
2. The **failure level** of the machine: When the state is equal to this level, the machine is broken.

The state increases over time and will at some point hit the failure level.

Example:

(F is the failure level)



The figure shows the **state,** increasing over time for two different runs to failure. We call these **deterioration paths.**

Policy

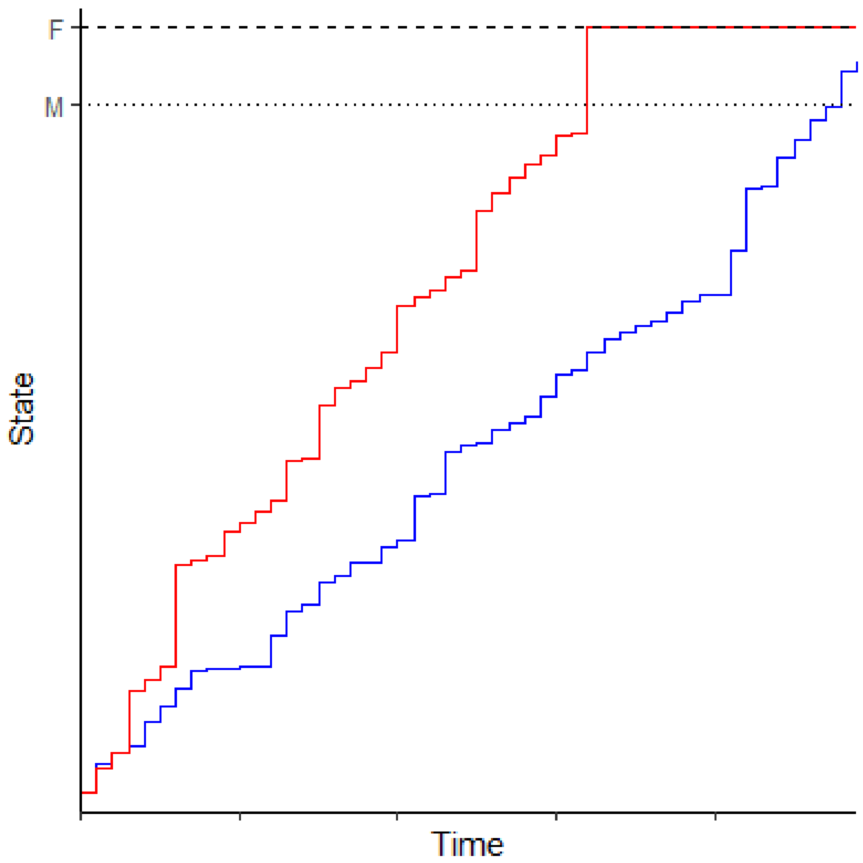
Condition-based maintenance is usually done through a **control-limit/threshold** policy.

If the state is above a **threshold** -> plan preventive maintenance.

If there is no planning time, then preventive maintenance is performed immediately. Otherwise, it will be performed immediately after the planning time (presuming no failure has occurred).

Condition-based maintenance in action

(M is the maintenance threshold)



Here, we successfully do preventive maintenance for blue.

The red deterioration path jumps immediately to F -> Still corrective maintenance.

Setting the maintenance threshold comes with similar a trade-off as setting a maintenance age.

Optimizing condition-based maintenance

Cost rate

Remember the formula for the cost rate?

$$\eta(T) = \frac{\text{Mean cost per cycle}}{\text{Mean cycle length}}$$

This is unchanged, except that we replace maintenance age T by maintenance threshold M.

$$\eta(M) = \frac{\text{Mean cost per cycle}}{\text{Mean cycle length}}$$

But, we calculate the mean cost per cycle and mean cycle length differently.

Evaluating a CBM policy

Last week we estimated the best Weibull distribution to explain our dataset.

We could do something similar, but:

- It is not always clear which distribution works the best.
- If we know the distribution, evaluating a policy is still not easy.

So we take a different approach!

Simulation

We can use simulation to evaluate a policy. The idea is to create large amounts of new deterioration paths from our dataset, and observe key metrics from these paths.

That is, we replace:

$$\text{Mean cost per cycle} = c_{\text{pm}} \times P(\text{cycle ends with PM}) + c_{\text{cm}} \times P(\text{cycle ends with failure})$$

with:

$$\text{Mean cost per cycle} = c_{\text{pm}} \frac{\text{\# cycles ending in PM}}{\text{\# simulated cycles}} + c_{\text{cm}} \frac{\text{\# cycles ending in failure}}{\text{\# simulated cycles}}$$

The mean cycle length is just the average length of our simulated deterioration paths.

Simulation

The approach is as follows:

1. Extract failure behaviour from the datasets.
2. Decide on a preventive maintenance threshold M.
3. Simulate the deterioration process.
4. Count the number of failures and calculate the average cycle length.
5. Calculate the cost rate.

Simulation

1. Extract failure behaviour from the datasets.

We make a list of the **deterioration increments:** the increases in the deterioration state over time. Decreases in the deterioration state should be filtered out (as these are due to maintenance).

Also, we need to find the failure level. Likely, this is the highest state in the dataset.

Simulation

Consider the following dataset.

Time	State
0	0
1	1.4
2	3.2
3	3.4
⋮	
7	9.6
8	10
8	0
9	1.2
⋮	

Inspections are planned at consistent intervals.

Maintenance interventions result in double entries.

Simulation

First, find the deterioration increments.

Time	State
0	0
1	1.4
2	3.2
3	3.4
⋮	
7	9.6
8	10
8	0
9	1.2
⋮	

→

Time	State	Increment
0	0	
1	1.4	1.4
2	3.2	1.8
3	3.4	0.2
⋮		
7	9.6	2.1
8	10	0.4
8	0	-10
9	1.2	1.2
⋮		

Simulation

Remove the first and any negative increments. (due to historic maintenance)

Time	State
0	0
1	1.4
2	3.2
3	3.4
⋮	
7	9.6
8	10
8	0
9	1.2
⋮	

→

Time	State	Increment
0	0	
1	1.4	1.4
2	3.2	1.8
3	3.4	0.2
⋮		
7	9.6	2.1
8	10	0.4
8	0	-10
9	1.2	1.2
⋮		

→

Time	State	Increment
~~0~~	~~0~~	
1	1.4	1.4
2	3.2	1.8
3	3.4	0.2
⋮		
7	9.6	2.1
8	10	0.4
~~8~~	~~0~~	~~-10~~
9	1.2	1.2
⋮		

Simulation

Next, we find the failure level.

Time	State
0	0
1	1.4
2	3.2
3	3.4
⋮	
7	9.6
8	10
8	0
9	1.2
⋮	

In this case, we assume that the highest state in the dataset is the failure level.

(F = 10)

Simulation

2. Decide on a preventive maintenance threshold M.

Remember, we will do preventive maintenance as soon as the state is above M (but below F).

So, this threshold needs to be somewhere between 0 and F.

Simulation

3. Simulate the deterioration process.

- We start with a as-good-as-new machine with state $x=0$.
- Pick a random increment Δx from our data and increase the state.
- Repeat until we do corrective ($x > F$) or preventive ($F > x > M$) maintenance.

Simulation (example)

(F=10, M = 8)

Start the deterioration path with x = 0

Time	x	Δx
0	0	

Increments
1.4
1.8
0.2
1.5
1.8
1.2
2.0
3.1
0.6
8.0
$\vdots$

Simulation (example)

(F=10, M = 8)

Randomly select an increment and increase the state:

Time	x	Δx
0	0	1.2
1	1.2	

Randomly pick Δx = 1.2 from the dataset.

Increments
1.4
1.8
0.2
1.5
1.8
1.2
2.0
3.1
0.6
8.0
⋮

Simulation (example)

(F=10, M = 8)

Randomly select an increment and increase the state:

Time	x	Δx
0	0	1.2
1	1.2	3.1
2	4.3	

Randomly pick Δx = 3.1 from the dataset.

Increments
1.4
1.8
0.2
1.5
1.8
1.2
2.0
3.1
0.6
8.0
$\vdots$

Simulation (example)

(F=10, M = 8)

Continue until either type of maintenance happens:

Time	x	Δx
0	0	1.2
1	1.2	3.1
2	4.3	0.2
3	4.5	2.0
4	6.5	1.8
5	8.3	

$M < x < F$ $(8 < 8.3 < 10)$
Preventive maintenance.

Simulation (example)

(F=10, M = 8)

Continue until either type of maintenance happens:

Time	x	Δx
0	0	1.2
1	1.2	3.1
2	4.3	0.2
3	4.5	2.0
4	6.5	8.0
5	14.5	

F < x (10 < 14.5)
Corrective maintenance.

We repeat this process for many cycles.

Simulation

4. Count the number of failures and calculate the average cycle length.

With all simulations done, we can now easily find:

1. #cycles ending in failure
2. #cycles ending in PM
3. Average length of simulated cycles

Simulation

5. Calculate the cost rate.

We can now calculate

$$\eta(M) = \frac{\text{Mean cost per cycle}}{\text{Mean cycle length}}$$

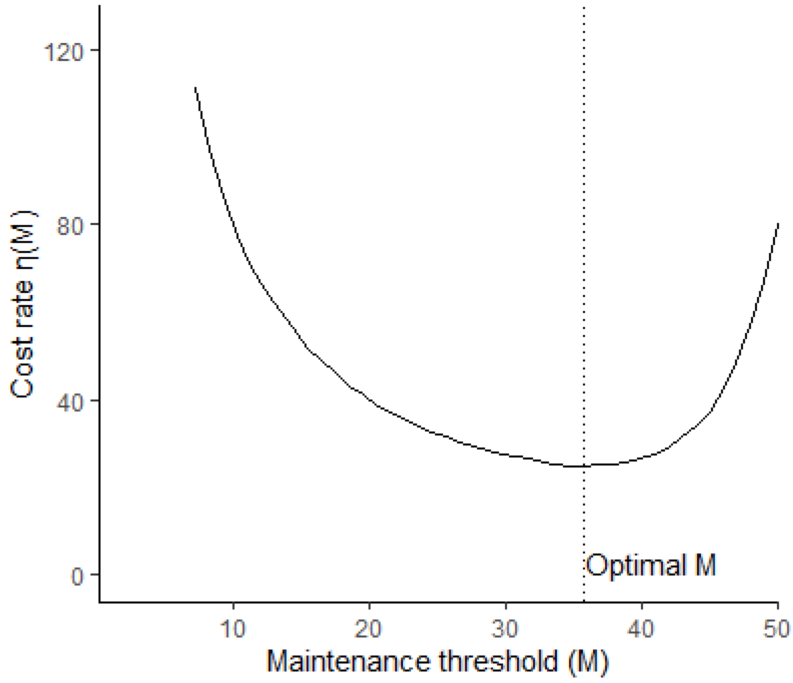
Where

$$\text{Mean cost per cycle} = c_{\text{pm}} \frac{\text{\# cycles ending in PM}}{\text{\# simulated cycles}} + c_{\text{cm}} \frac{\text{\# cycles ending in failure}}{\text{\# simulated cycles}}$$

$$\text{Mean cycle length} = \text{Average length of simulated cycles}$$

Cost rate

We can solve η(M) for the best value of M, but we can also solve it for all possible M and create a continuous figure:



The optimal threshold is at the minimum of the curve.

Found optimum:

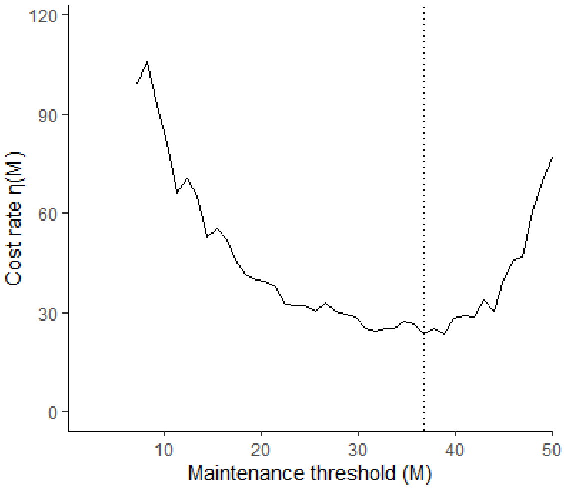
M = 37.78

η(M) = 23.36

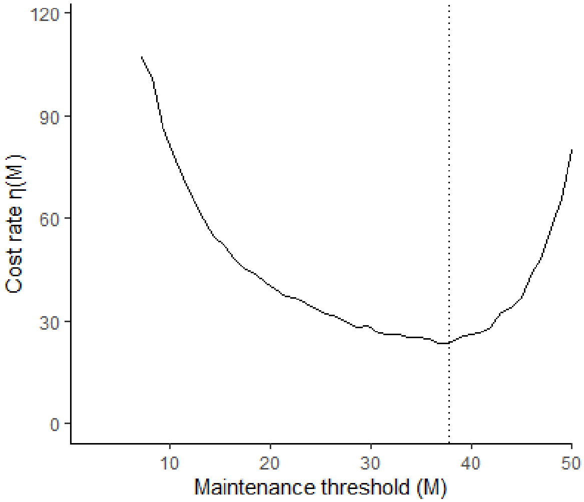
Impact of number of simulations

You may need to do many simulations per threshold to get a smooth curve and accurate cost rates. This comes with increased computational times.

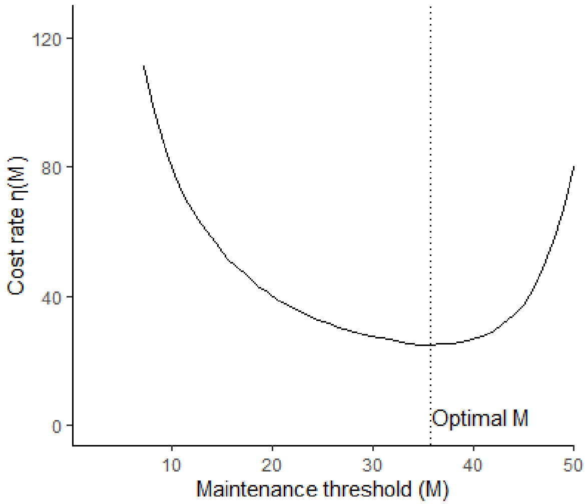
100 simulations



1,000 simulations



10,000 simulations



Challenges with CBM

Challenges with CBM

In practice, you will never have such accurate condition information.

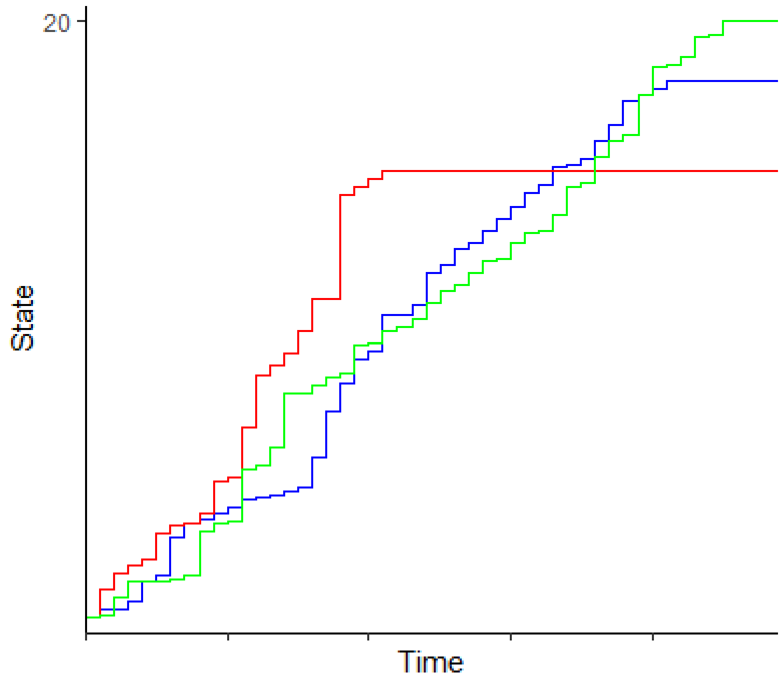
Some problems you may have:

- Uncertain failure levels
- Imperfect maintenance
- Nonstationary deterioration
- Few measurable states

And many more!

Uncertain failure levels

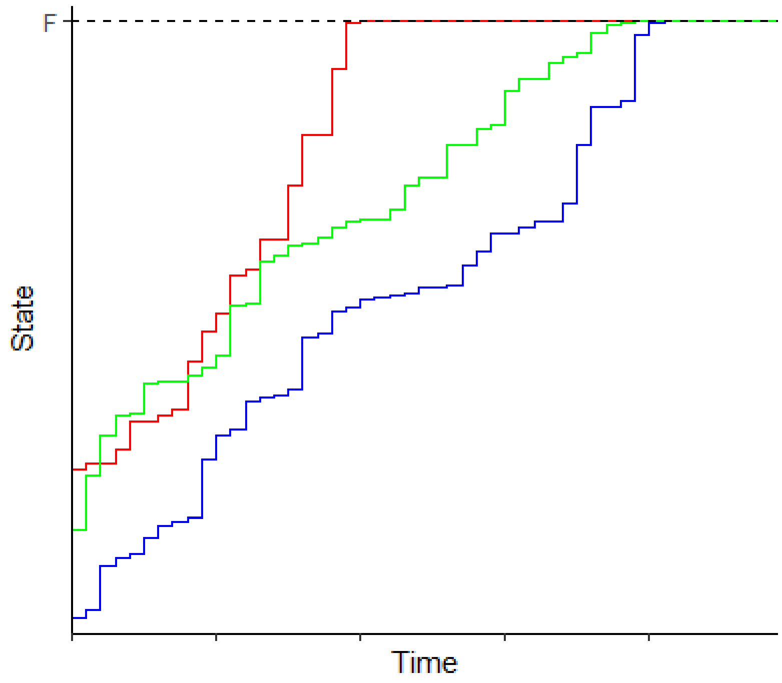
So far, we assumed machines always failed at an exact deterioration level. This does not need to be the case:



In this example the failure level is not constant: we observed failures at x = 15, x=18 and x=20.

Imperfect maintenance

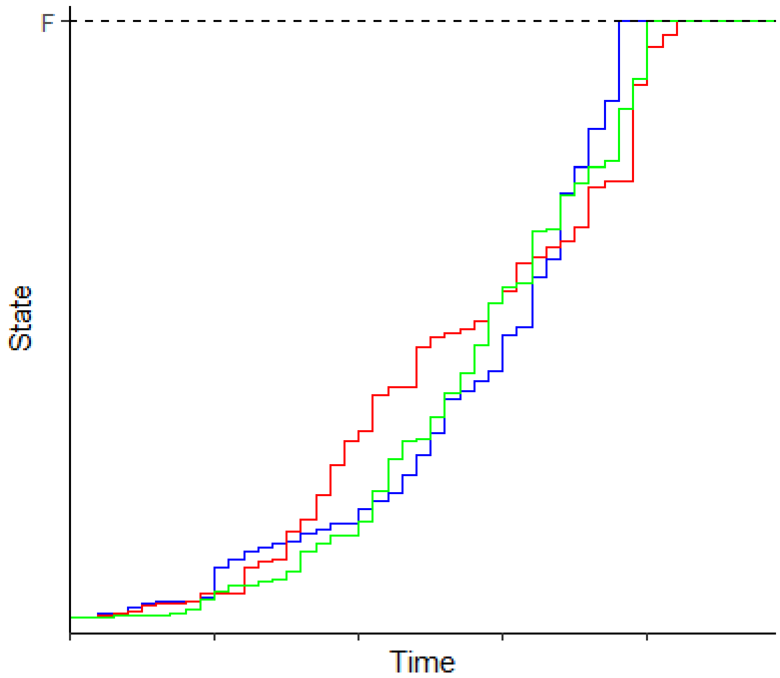
Maintenance is not always a replacement! A machine does not need to go to an as-good-as-new state after maintenance.



Each deterioration path starts at a different state -> Maintenance is not perfect.

Nonstationary degradation

Deterioration is not always at the same speed. Deterioration processes can be **nonstationary.**



Observe that the deterioration paths become steeper with more wear. Deterioration speeds increase as machines degrade.

As a result, our simulations are invalid.

Few measurable states

But really, the most common problem is that we can not measure many states at all.

In practice, differentiating between a new and old machine based on sensors is difficult enough!

Often, condition-based maintenance amounts to **detecting a fault/anomaly**. When a sensor consistently gives abnormal readings, we take action.

Conclusion

Condition-based maintenance is a great tool that can result in a real improvement over age-based maintenance.

But:

- It requires significant investment.
- It is far more difficult to do well.

This Thursday: More on simulating. We will also discuss methods to resolve some of these problems.