# Bagatelle
# Minimum

Dr. Christoph Clauß, 23.4.2014

**gegeben:** $x(t), 0 \leq t \leq T, x \in R$

**gesucht:** $y(t), 0 \leq t \leq T, y \in R$

**mit der Eigenschaft:**

$$y(t) = \min_{0 \leq \tau \leq t} (x(\tau))$$

# Versuch 1  Simples Minimum

MODELICA Nutzergruppe Sachsen

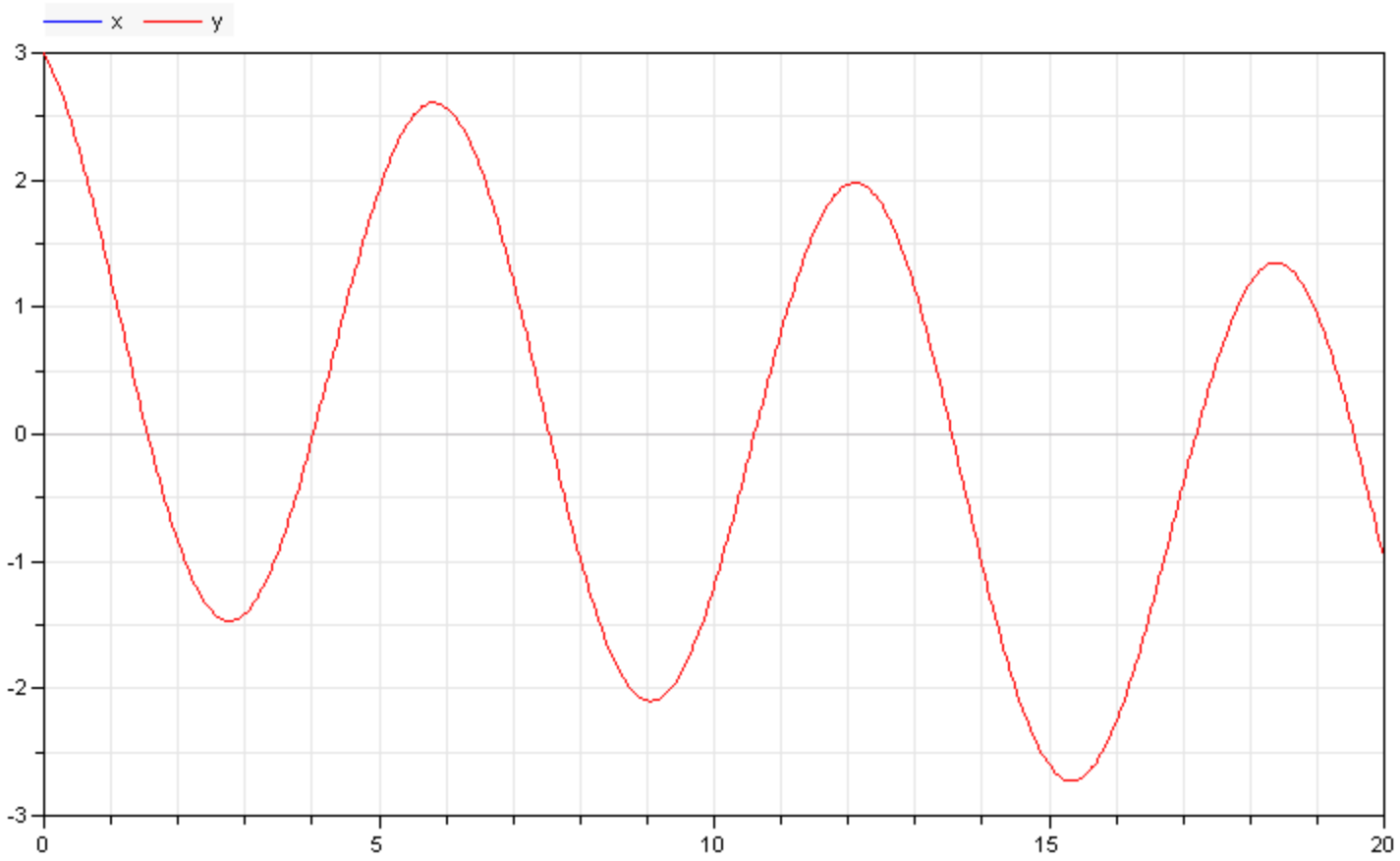
```
model versuch1
    Real x( start=3, fixed = true);
    Real derx( start=-1, fixed=true);
    Real y( start=3);
algorithm
  y := min(x, y);
equation
  der(x) = derx;
  der(derx) = - x + 1.0 - 0.1*time;
end versuch1;
```





Fraunhofer IIS

# Versuch 1  Simples Minimum

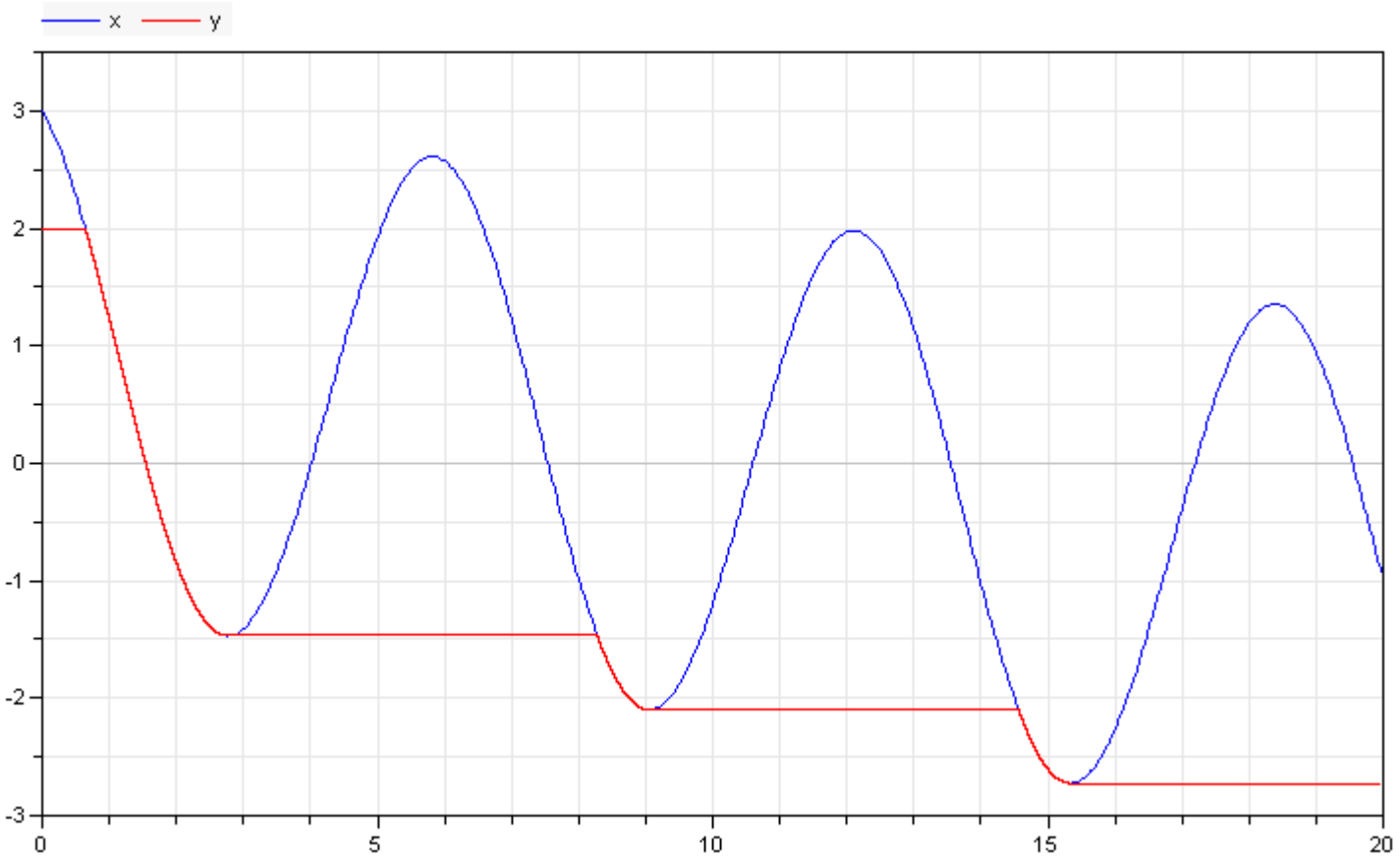



Dymola 2014 FD01

# Versuch 2 „Abtasten“

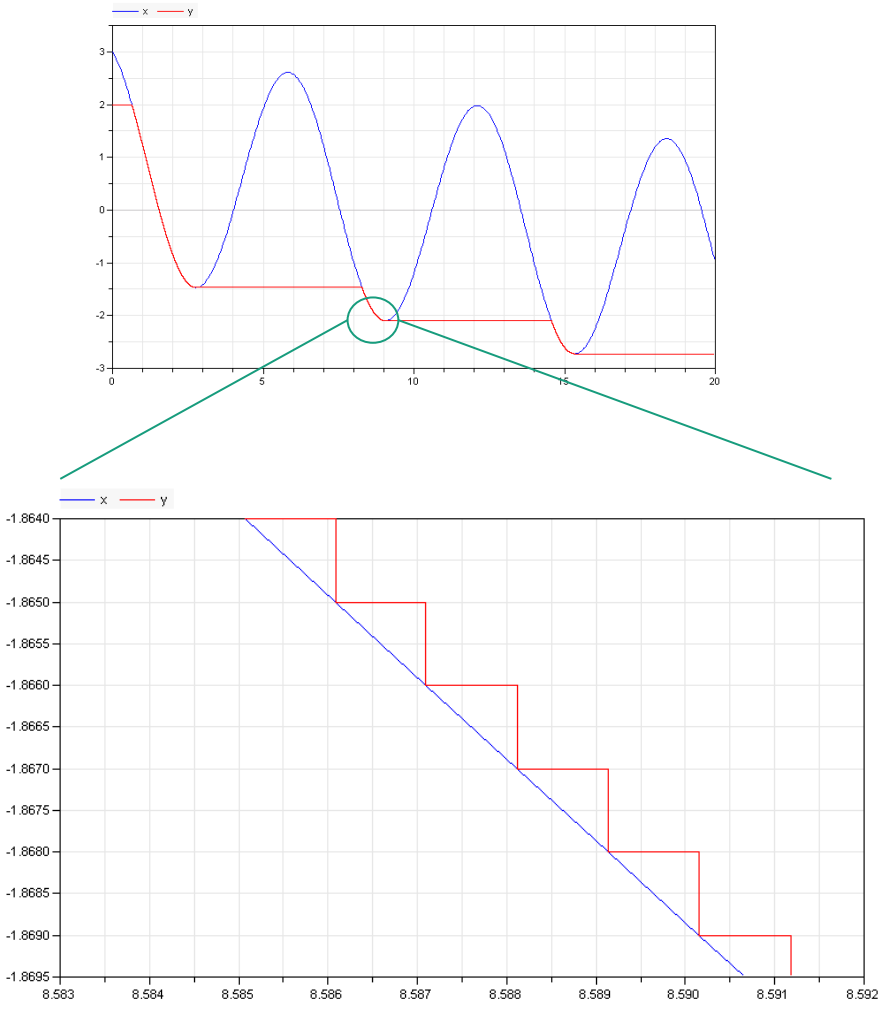
```
model versuch2
    Real x( start=3, fixed = true);
    Real derx( start=-1, fixed=true);
    Real y( start=3);
algorithm
  when x < y -0.001 then y := x;
  end when;
equation
  der(x) = derx;
  der(derx) = - x + 1.0 - 0.1*time;
end versuch2;
```

# Versuch 2 „Abtasten“




Dymola 2014 FD01

# Versuch 2 „Abtasten“




Number of result points : 9949

Number of GRID points : 501

Number of (successful) steps : 28391

Number of F-evaluations : 52079

Number of H-evaluations : 65744

Number of Jacobian-evaluations: 23558

Number of (model) time events : 0

Number of (U) time events : 0

Number of state events : 4724

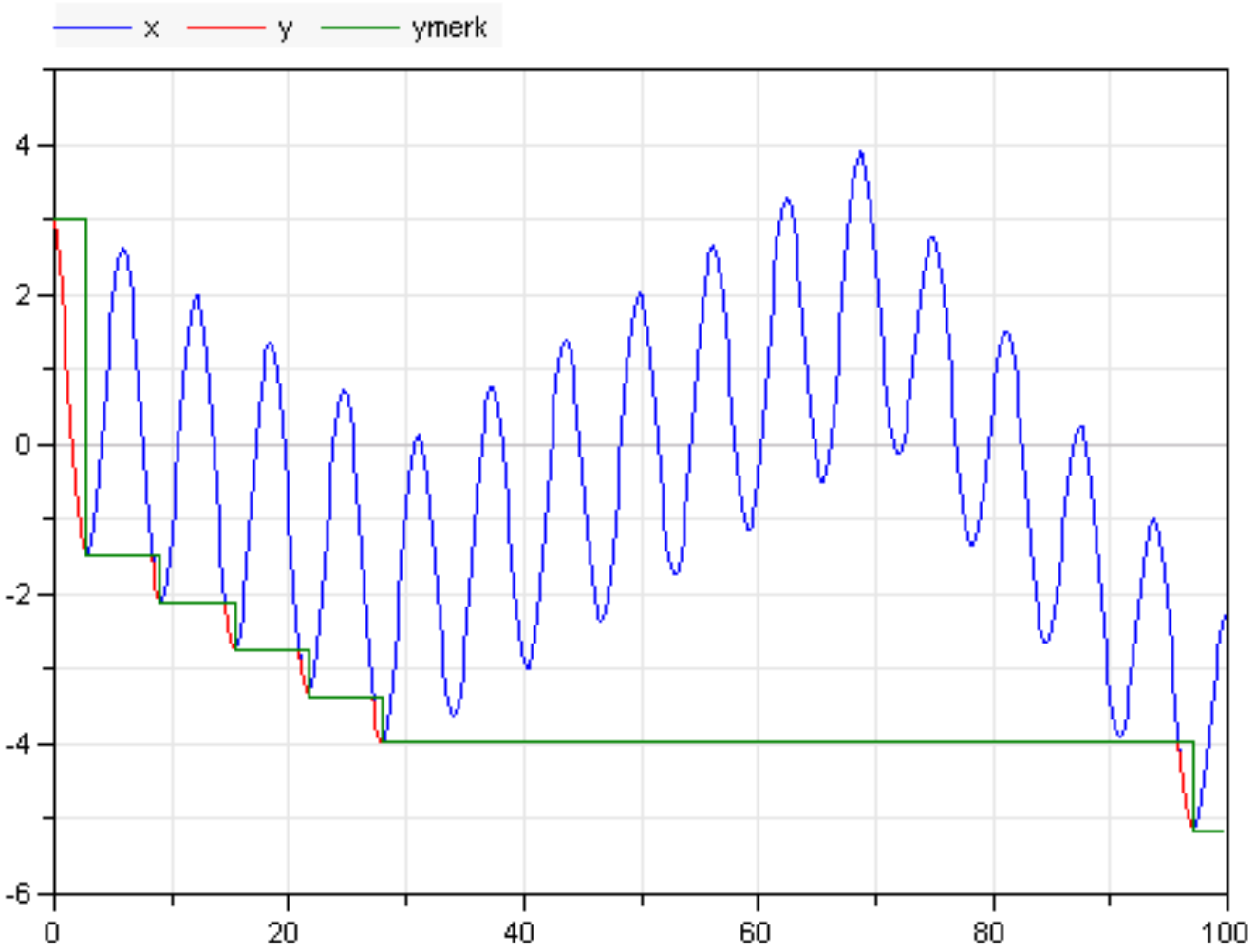
Number of step events : 0

Dymola 2014 FD01

# Versuch 3 „Auswertung Anstieg“

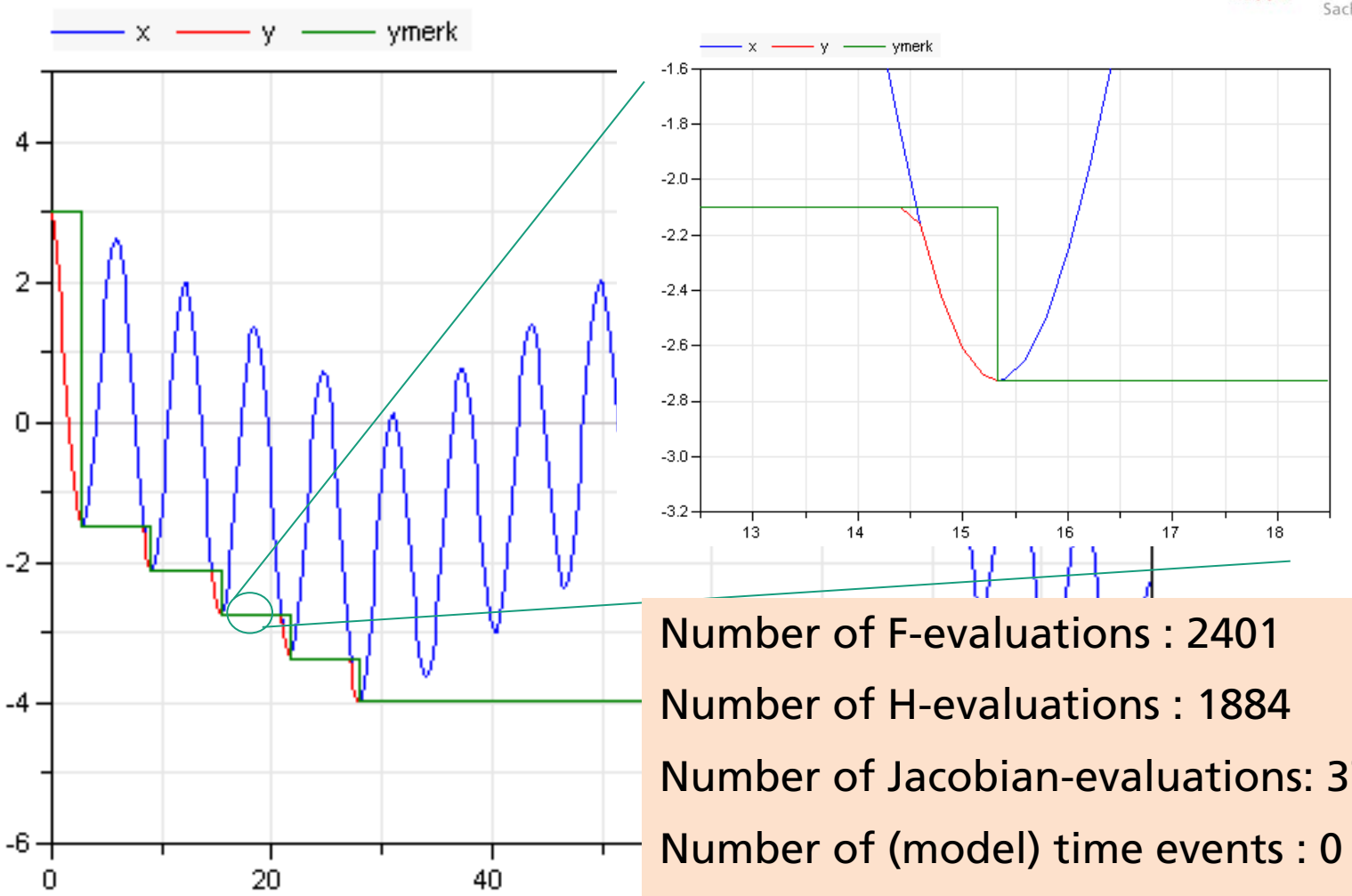
```
model versuch3
  Real x( start=3, fixed = true);  Real derx( start=-1, fixed=true);
  Real y( start=3);  discrete Real ymerk(start=3);
  Boolean sinking( start=false);
algorithm
  when derx > 0 then  ymerk:= min(x, pre(ymerk));  end when;
equation
  sinking = if derx > 0 then false else true;
  y = if sinking then min(x, ymerk) else ymerk;
  der(x) = derx;
  der(derx) = -x + 1 - 0.1*time + 0.2 *max(0,(time-30)) -
 0.3 *max(0,(time-70));
end versuch3;
```

# Versuch 3 „Auswertung Anstieg“




Dymola 2014 FD01

# Versuch 3 „Auswertung Anstieg“




Number of F-evaluations : 2401

Number of H-evaluations : 1884

Number of Jacobian-evaluations: 378

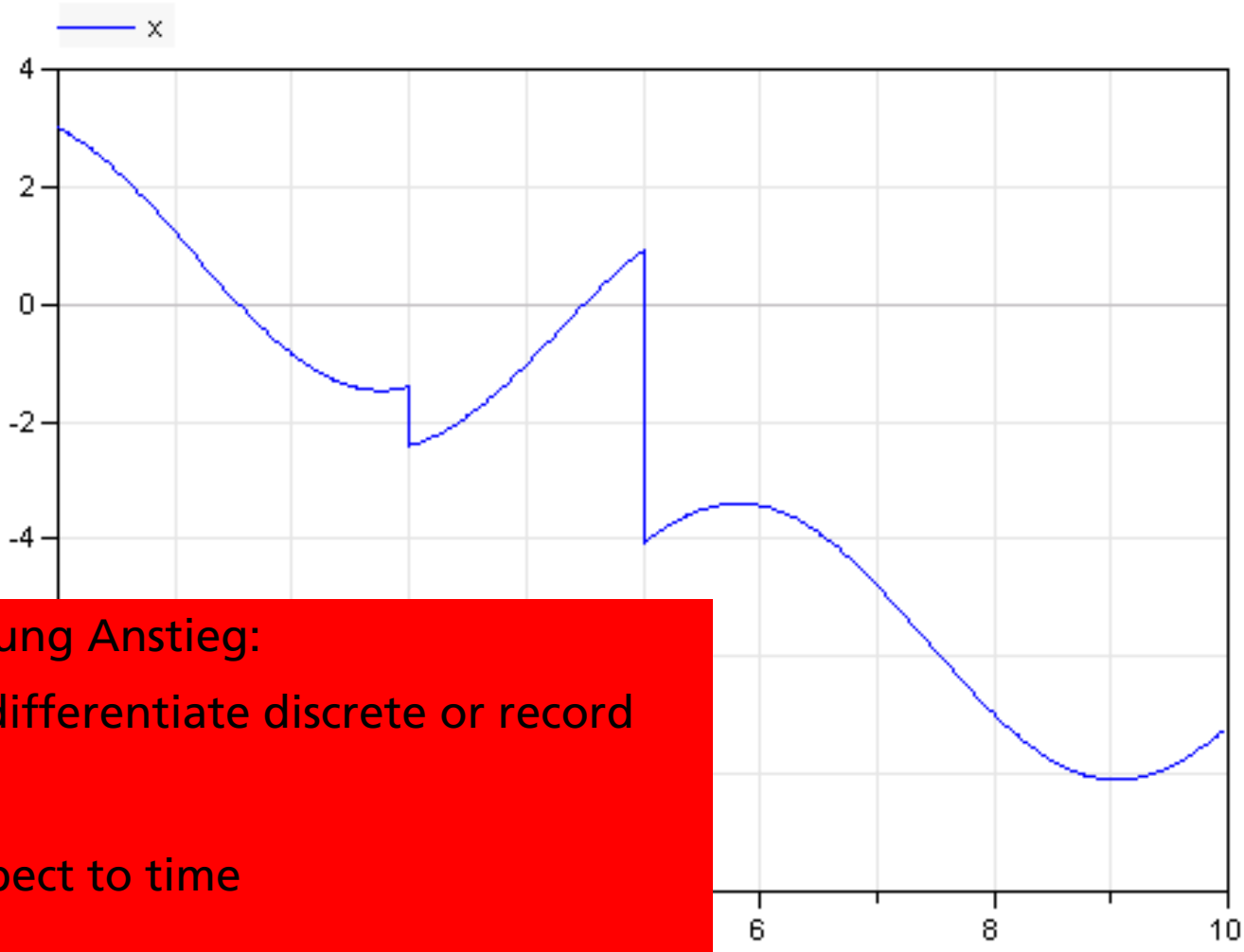
Number of (model) time events : 0

Number of (U) time events : 0

Number of state events : 32

Number of step events : 0

# Versuch 4  diskretes x, Auswertung Anstieg

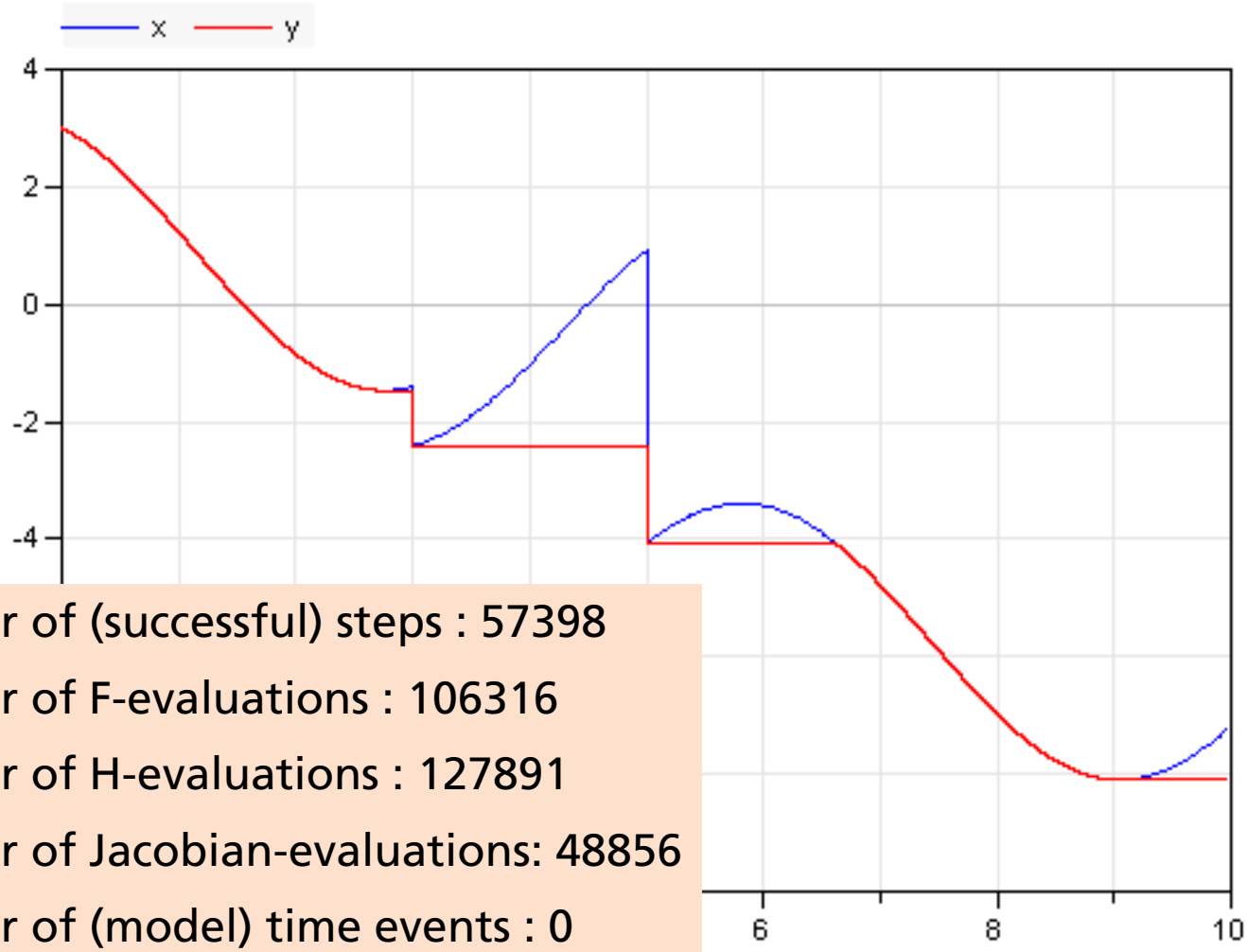



Auswertung Anstieg:

Cannot differentiate discrete or record variable:
z
with respect to time

Failed to differentiate the equation
x = w+z;

# Versuch 5 diskretes x, Abtasten




Number of (successful) steps : 57398

Number of F-evaluations : 106316

Number of H-evaluations : 127891

Number of Jacobian-evaluations: 48856

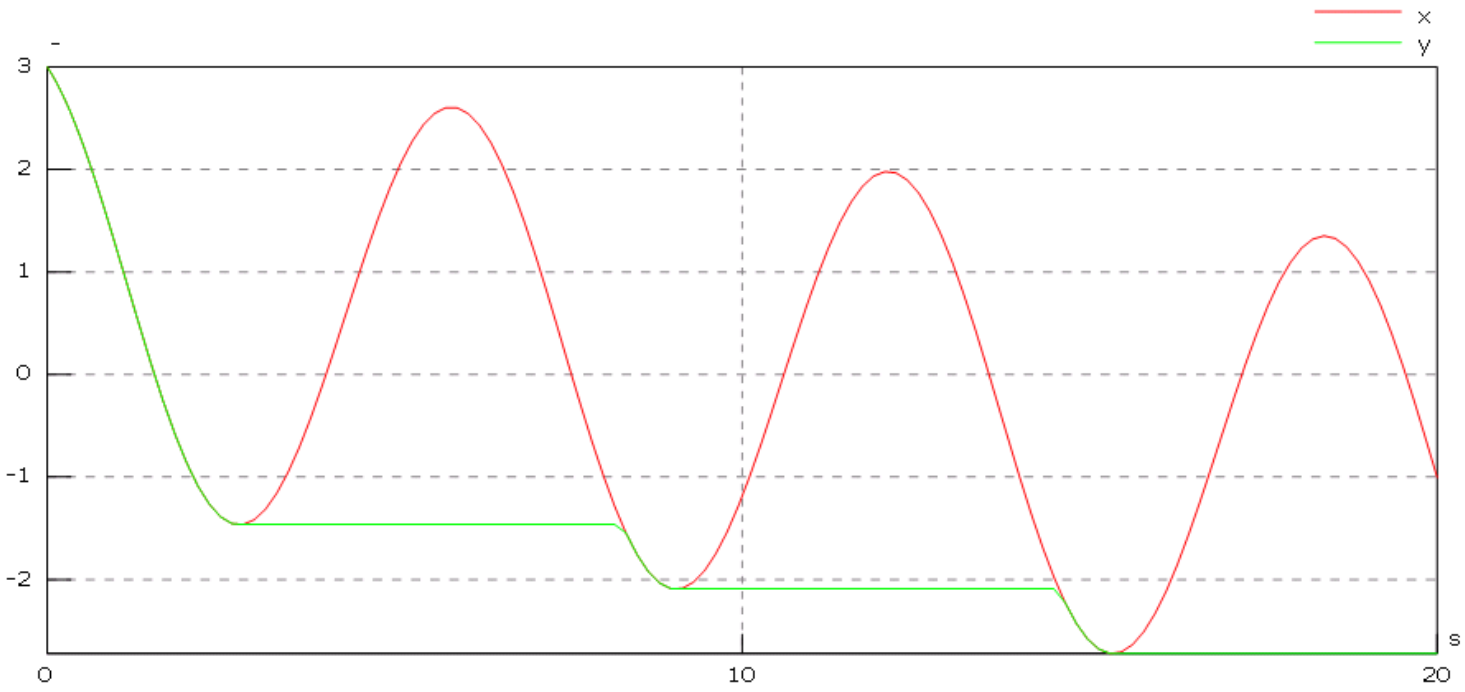
Number of (model) time events : 0

Number of (U) time events : 0

Number of state events : 8495

# Versuch 1

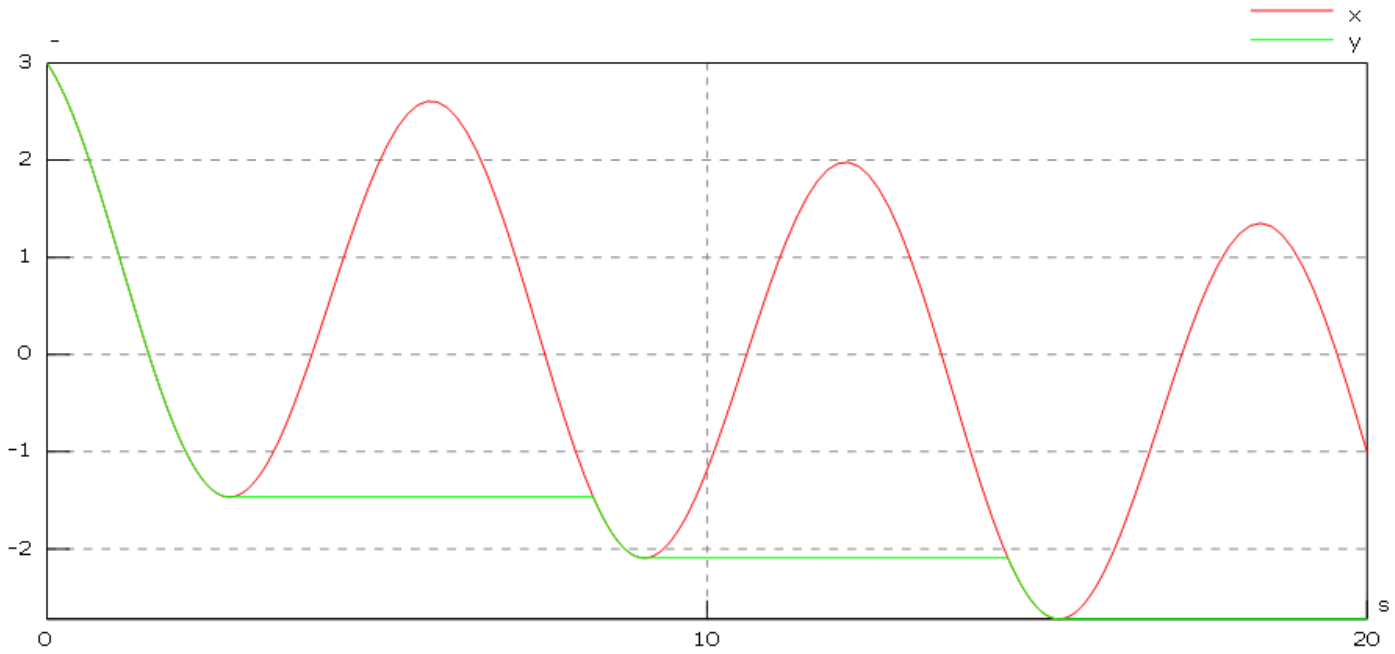
Event-Iterationen: 2




SimulationX 3.6.1

# Versuch 2 „Abtasten“

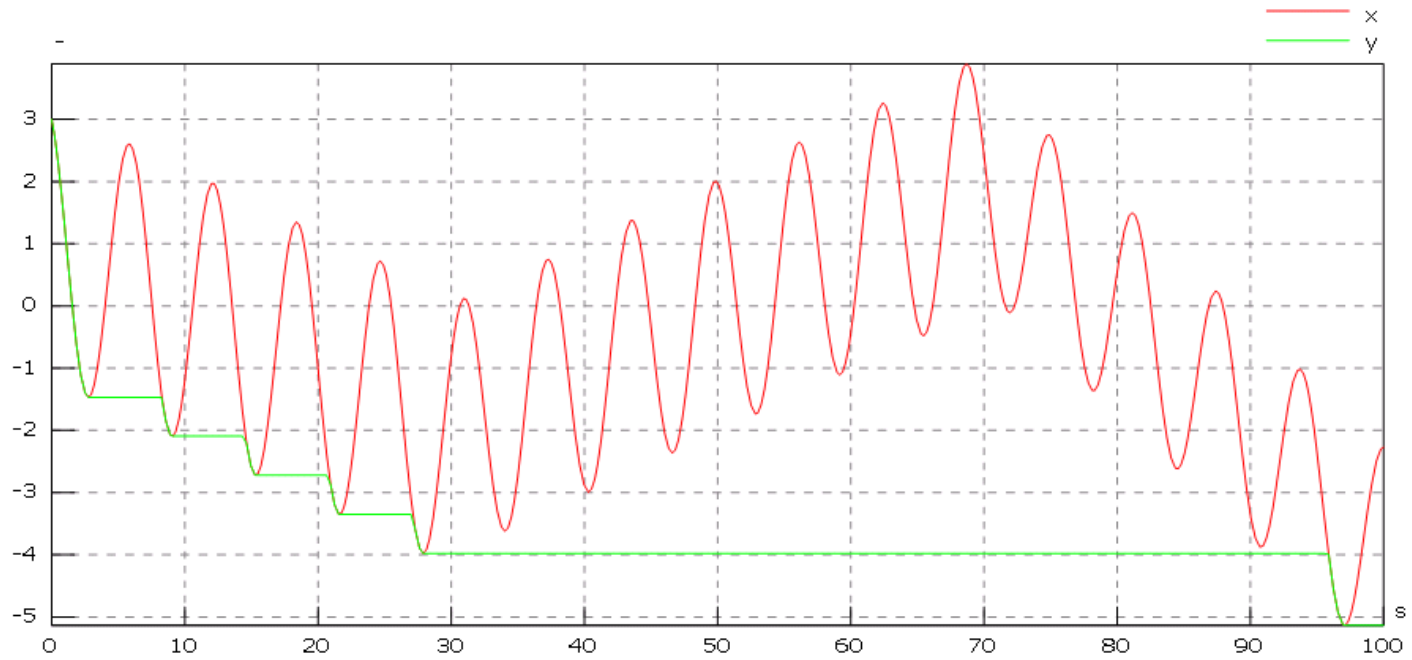
Event-Iterationen: 5723




SimulationX 3.6.1

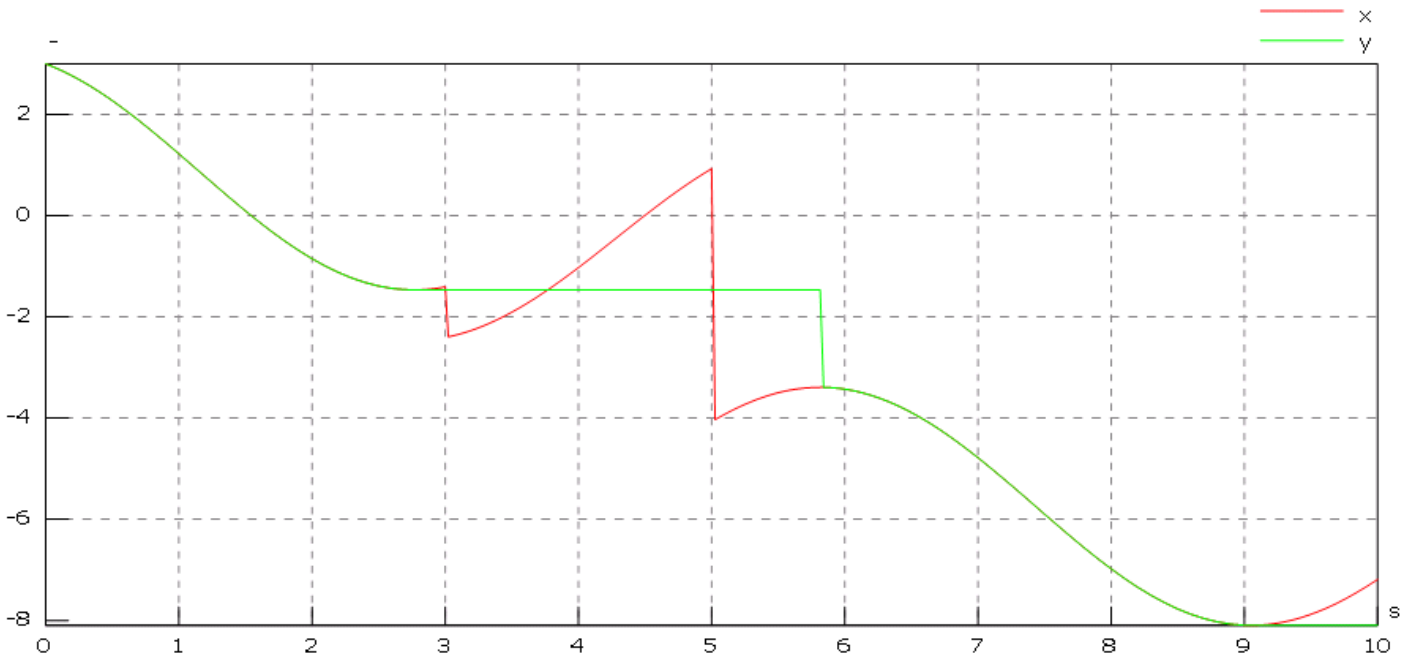
# Versuch 3 „Auswertung Anstieg“

Event-Iterationen: 34




SimulationX 3.6.1

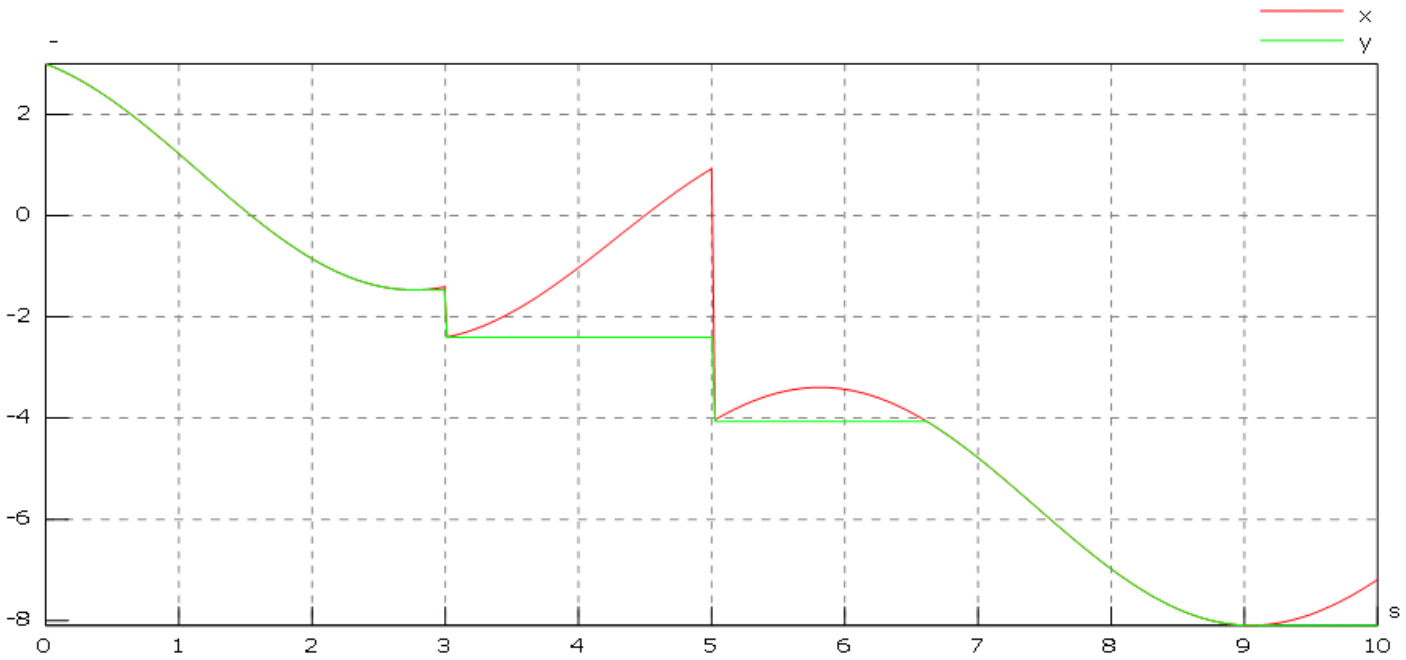
# Versuch 4  diskretes x, Auswertung Anstieg




SimulationX 3.6.1

# Versuch 5 diskretes x, Abtasten

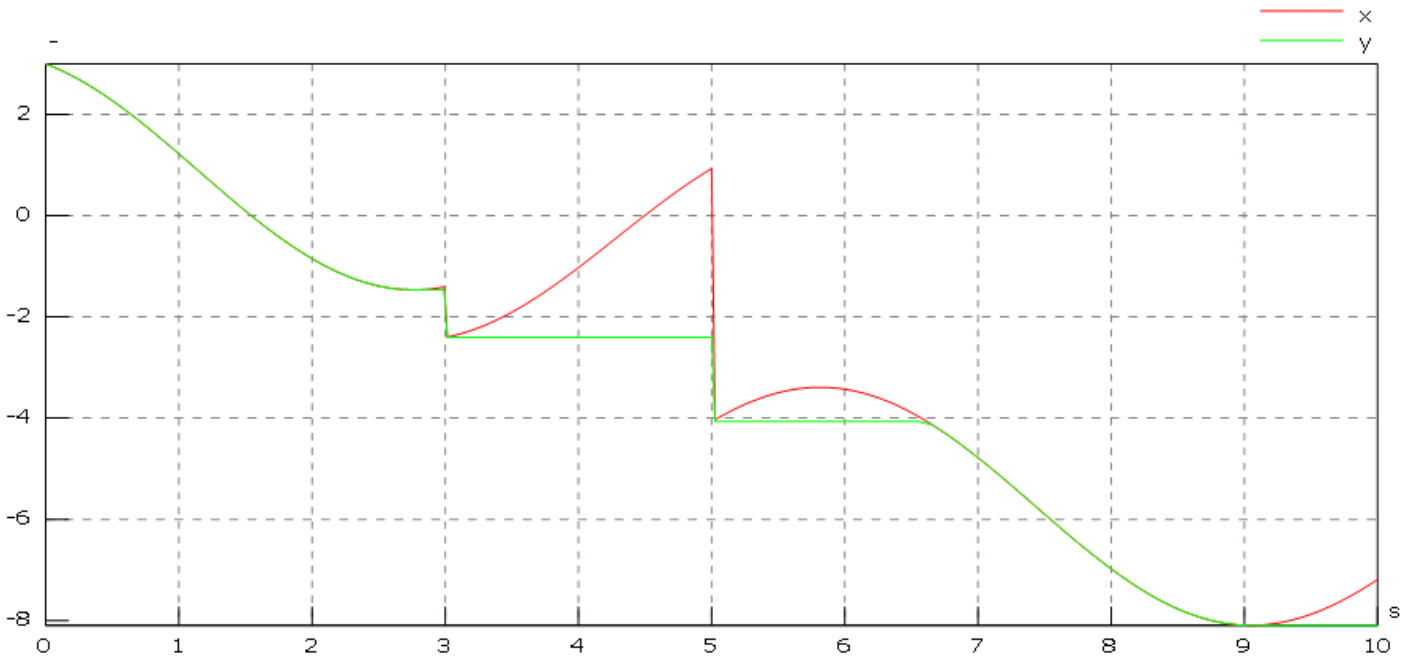
Event-Iterationen: 8493




SimulationX 3.6.1

# Versuch 6  diskretes x, simples Minimum

Event-Iterationen: 4




SimulationX 3.6.1

# Versuch 1  Simples Minimum

```
model versuch1
    Real x( start=3, fixed = true);
    Real derx( start=-1, fixed=true);
    Real y( start=3);
algorithm
  y := min(x, y);
equation
  der(x) = derx;
  der(derx) = - x + 1.0 - 0.1*time;
end versuch1;
```

```
equation
  y = min(x, y);
```

**System singulär**

SimulationX 3.6.1

Fazit:

- Ist diese einfache Form (y=min(x,y)), die in SimX geht, Modelica-konform?
- Eine Art Funktion für die genannte Problemstellung ist in Modelica erforderlich