Congestion control methods: A, B, C and D

`Method A`:

- if $Q(t) = Q^*$ then $\lambda(t+1) \leftarrow \lambda(t)$
- if $Q(t) < Q^*$ then $\lambda(t+1) \leftarrow \lambda(t) + a$
- if $Q(t) > Q^*$ then $\lambda(t+1) \leftarrow \lambda(t) - a$

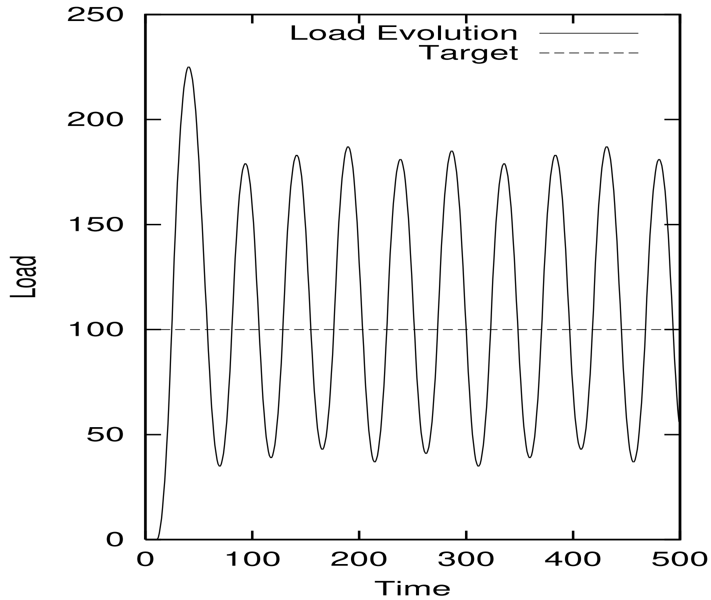
where $a > 0$ is a fixed parameter

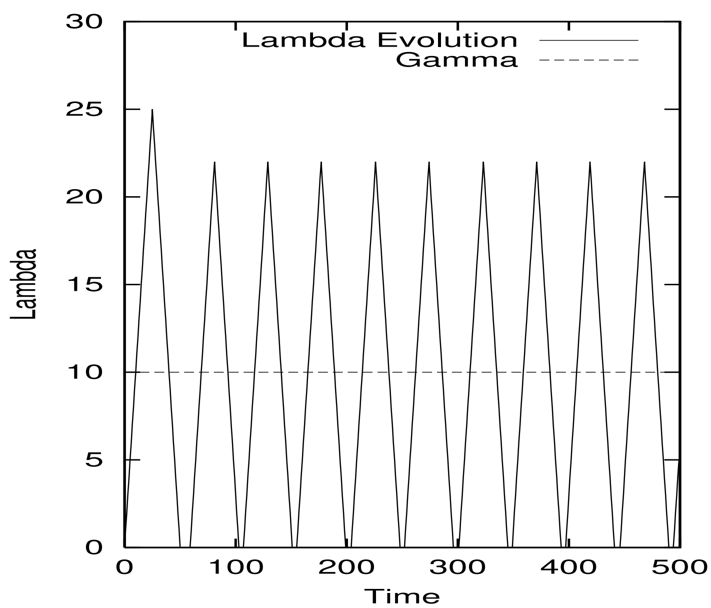
$\longrightarrow$ called linear increase/linear decrease

Question: how well does it work?

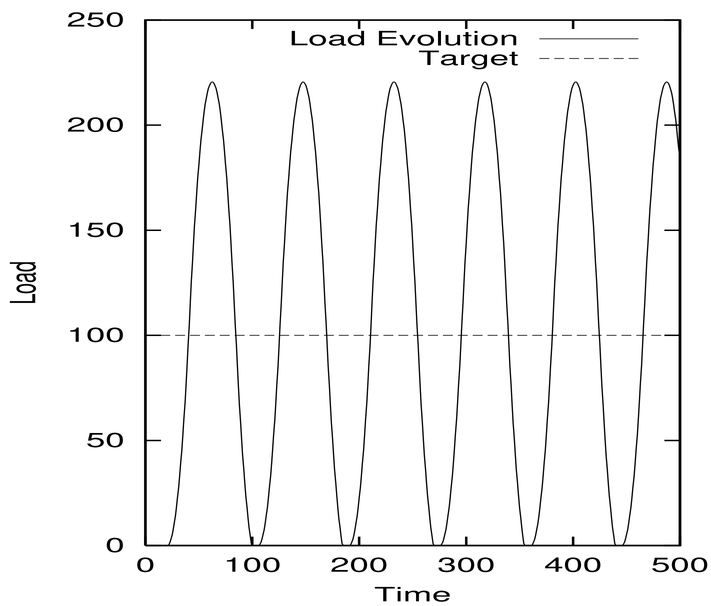
Example:

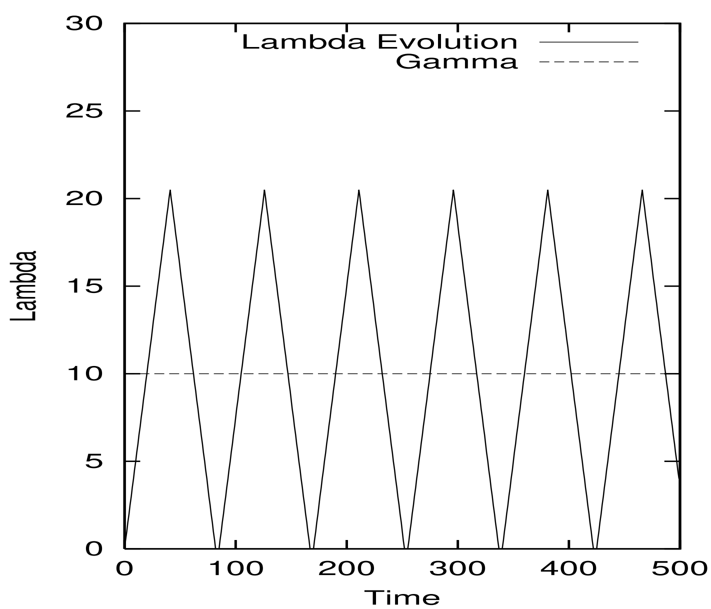
- $Q(0) = 0$
- $\lambda(0) = 0$
- $Q^* = 100$
- $\gamma = 10$
- $a = 1$


Load Evolution
Target
Load
Time
0
50
100
150
200
250
0
100
200
300
400
500

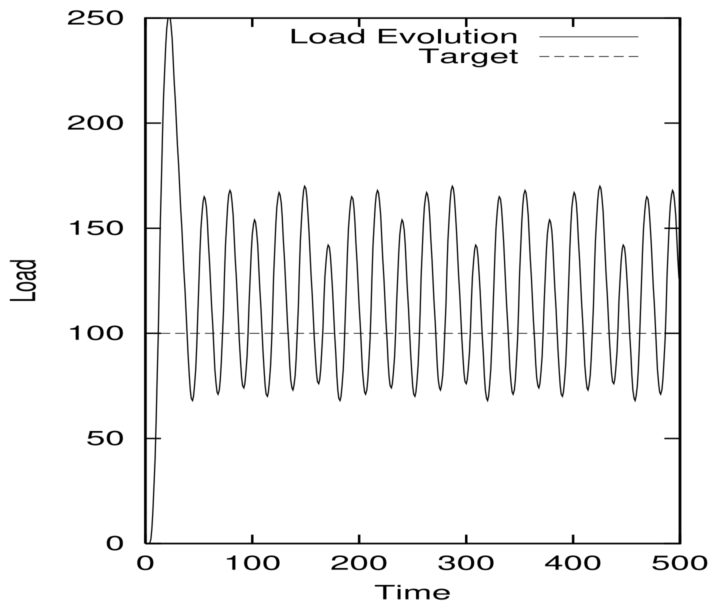

Lambda Evolution
Gamma
Lambda
Time
0
5
10
15
20
25
30
0
100
200
300
400
500

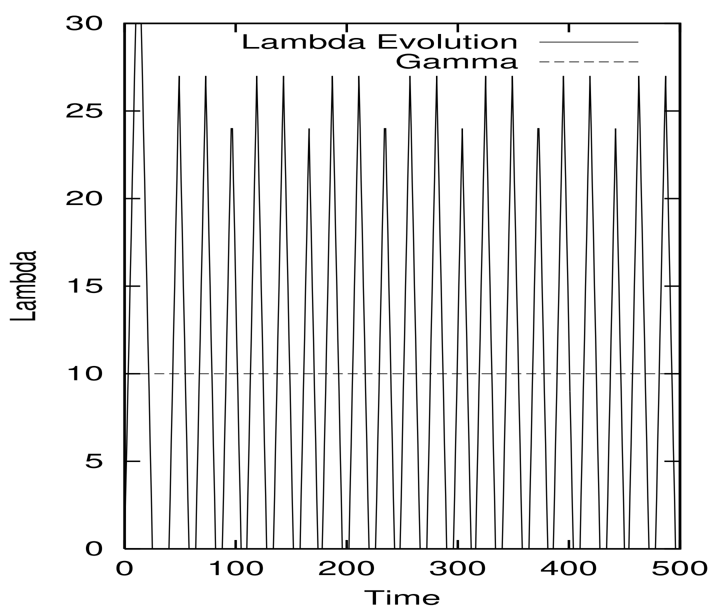
With $a = 0.5$:


Load Evolution
Target
250
200
150
100
50
0
Load
0
100
200
300
400
500
Time

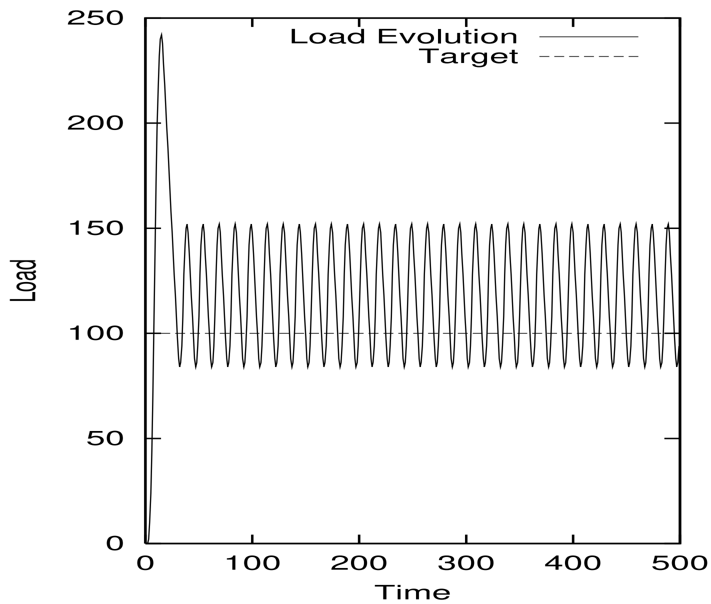

Lambda Evolution
Gamma
30
25
20
15
10
5
0
Lambda
0
100
200
300
400
500
Time

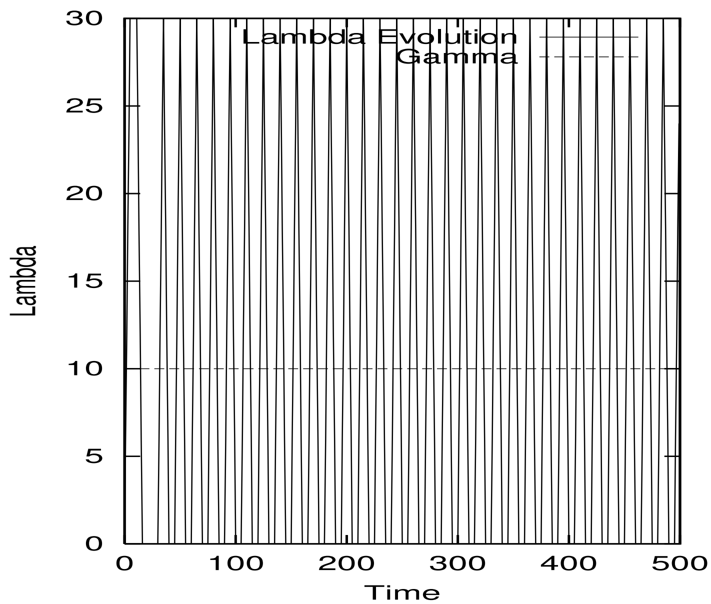
With $a = 3$:


250
200
150
100
50
0
Load
Load Evolution
Target
0
100
200
300
400
500
Time


30
25
20
15
10
5
0
Lambda
Lambda Evolution
Gamma
0
100
200
300
400
500
Time

With $a = 6$:


Load Evolution
Target
Load
Time
250
200
150
100
50
0
0
100
200
300
400
500


Lambda Evolution
Gamma
Lambda
Time
30
25
20
15
10
5
0
0
100
200
300
400
500

Remarks:

- Method A isn't that great no matter what a value is used

  $\rightarrow$ keeps oscillating

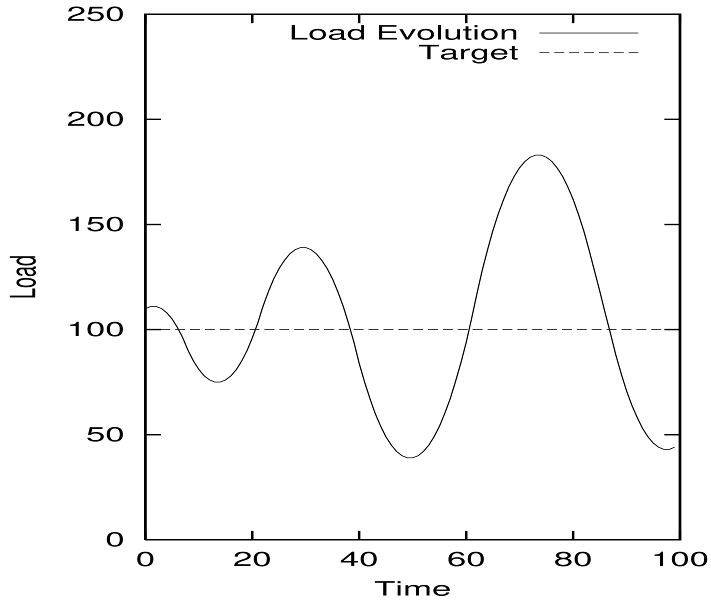
- Actually: would lead to unbounded oscillation if not for physical restriction $\lambda(t) \geq 0$ and $Q(t) \geq 0$

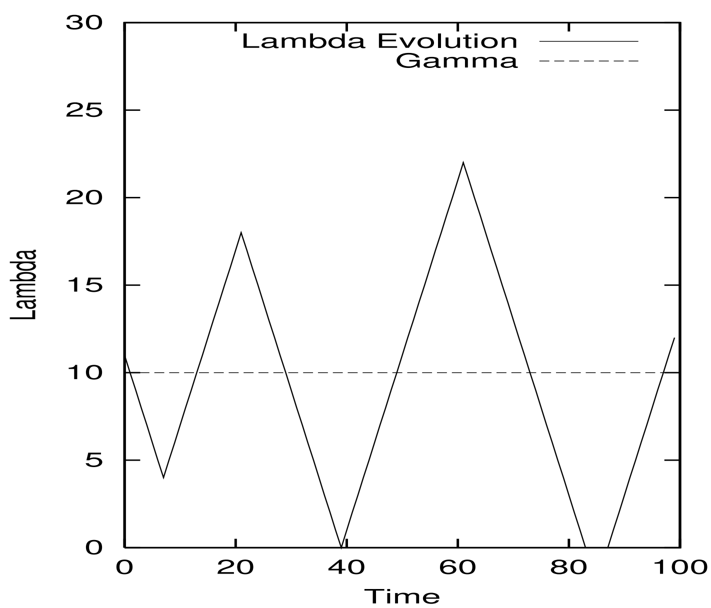
  $\longrightarrow$ i.e., bottoms out

  $\longrightarrow$ easily seen: start from non-zero buffer

  $\longrightarrow$ e.g., $Q(0) = 110$

With $a = 1$, $Q(0) = 110$, $\lambda(0) = 11$:





`Method B`:

- if $Q(t) = Q^*$ then $\lambda(t+1) \leftarrow \lambda(t)$
- if $Q(t) < Q^*$ then $\lambda(t+1) \leftarrow \lambda(t) + a$
- if $Q(t) > Q^*$ then $\lambda(t+1) \leftarrow \delta\, \lambda(t)$

where $a > 0$ and $0 < \delta < 1$ are fixed parameters

Note: only decrease part differs from `Method A`.

$\longrightarrow$ linear increase with slope a

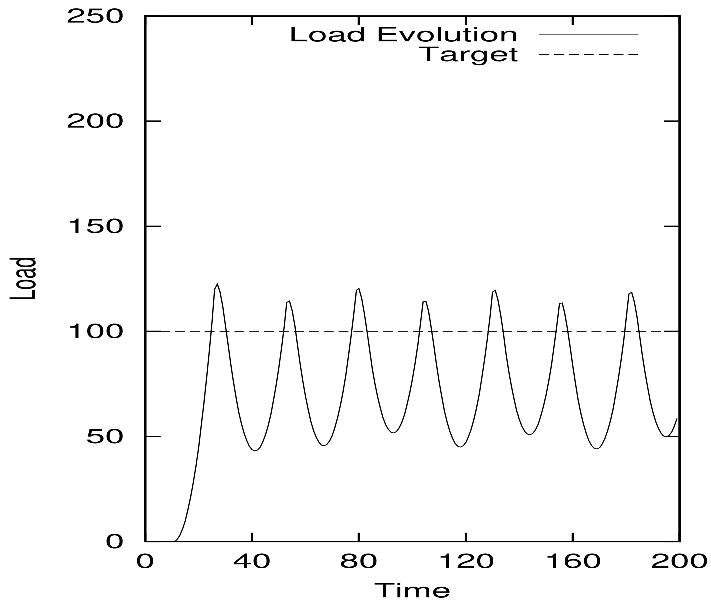
$\longrightarrow$ exponential decrease with backoff factor δ

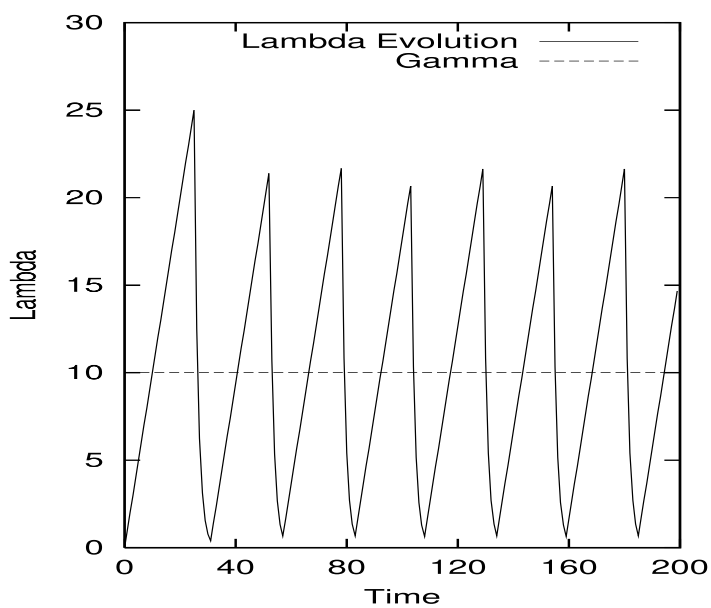
$\longrightarrow$ e.g., binary backoff in case $\delta = 1/2$

Similar to Ethernet and WLAN backoff

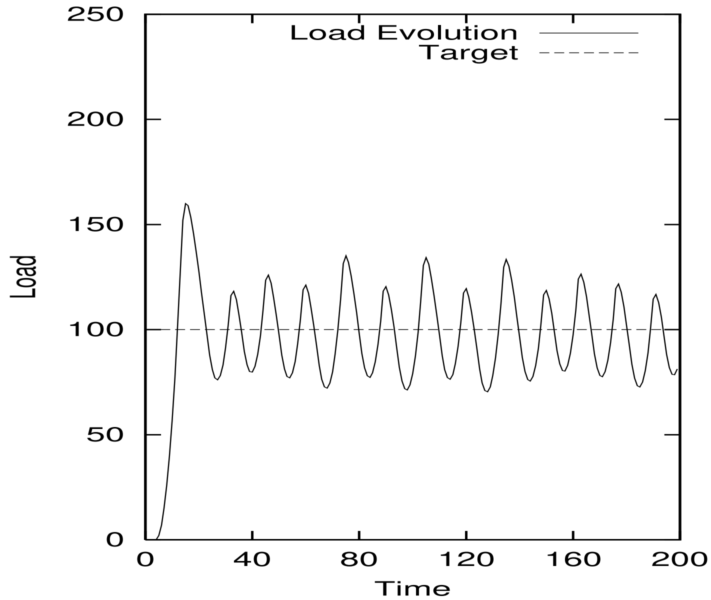
$\longrightarrow$ question: does it work?

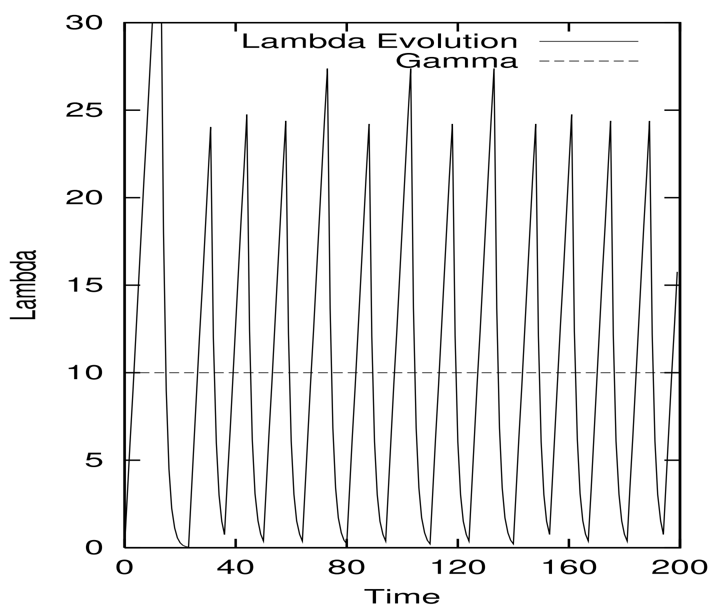
With $a = 1$, $\delta = 1/2$:



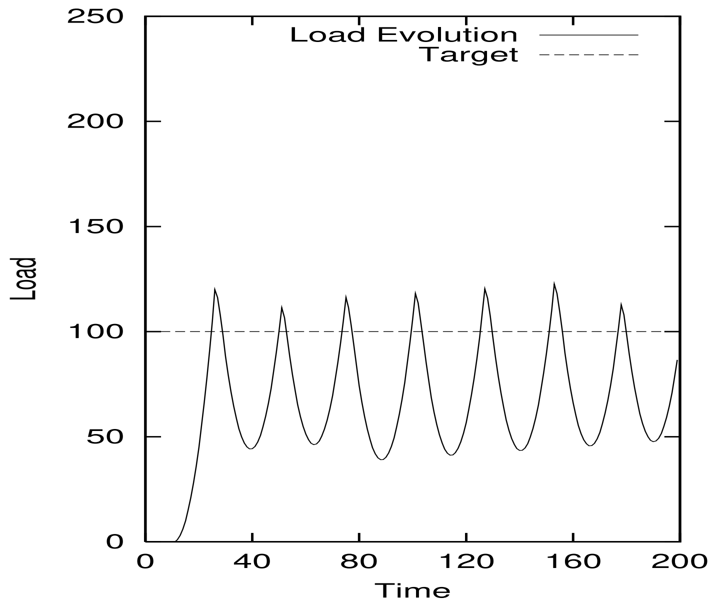


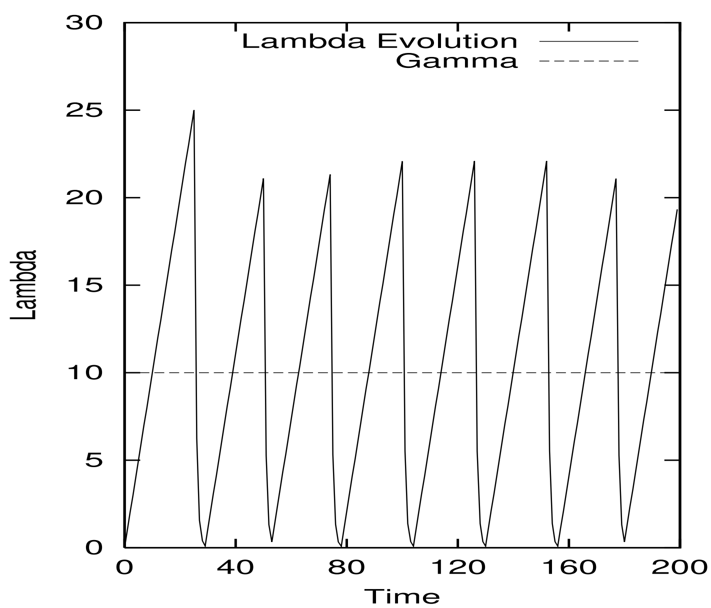
With $a = 3$, $\delta = 1/2$:



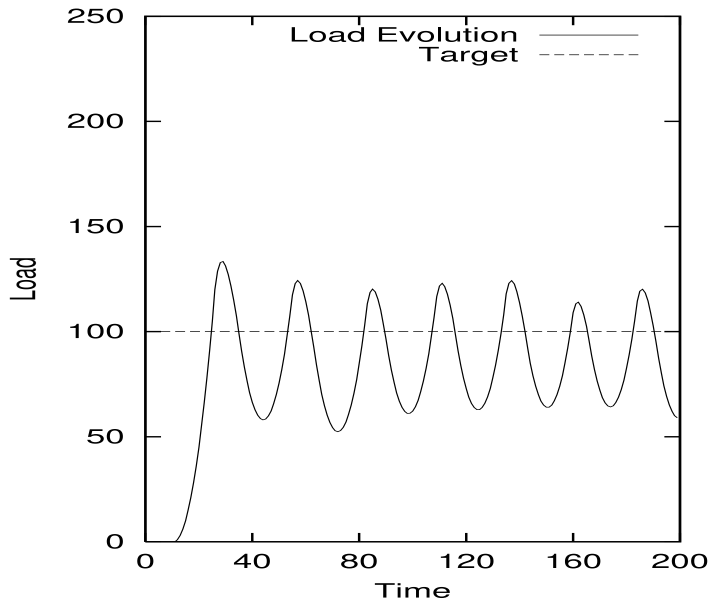


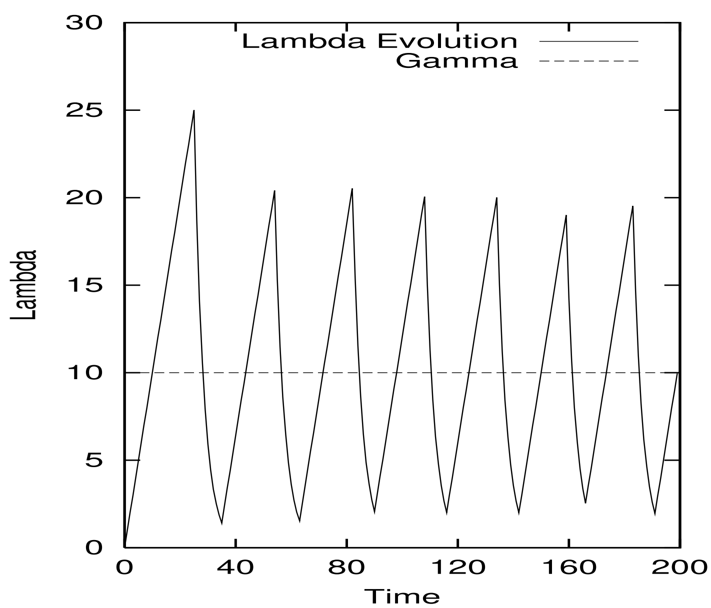
With $a = 1$, $\delta = 1/4$:





With $a = 1$, $\delta = 3/4$:





Note:

- Method B oscillates around target

  → does not converge

- Superior to Method A: unbounded oscillation

  → doesn't hit "rock bottom"

  → due to asymmetry in increase vs. decrease policy

  → we observe "sawtooth" pattern

- Method B is used by TCP

  → linear increase/exponential decrease

  → additive increase/multiplicative decrease (AIMD)

  → will discuss specifics separately

Question: can we do better?

⟶ what information have we not made use of?

`Method C`:

$$\lambda(t+1) \leftarrow \lambda(t) + \varepsilon(Q^* - Q(t))$$

where $\varepsilon > 0$ is a fixed parameter

Tries to adjust magnitude of change as a function of the gap $Q^* - Q(t)$

$\rightarrow$ if gap is big, change by a lot

$\rightarrow$ if gap is small, change by a little

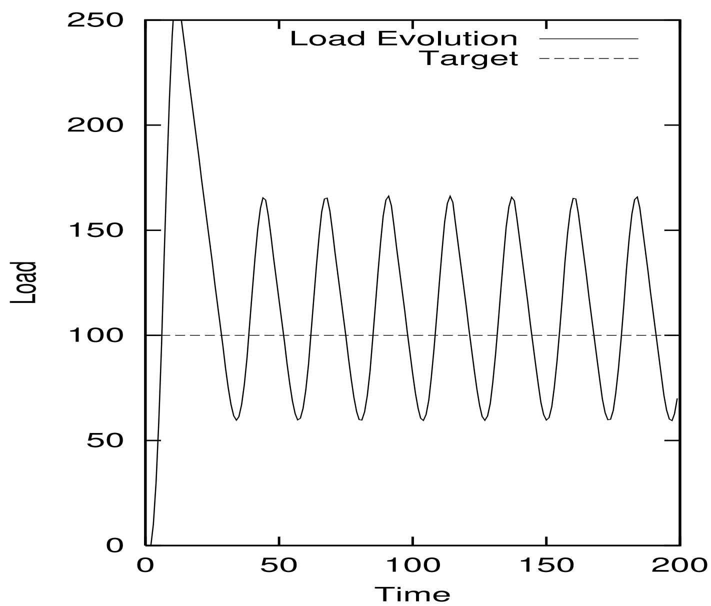
Thus:

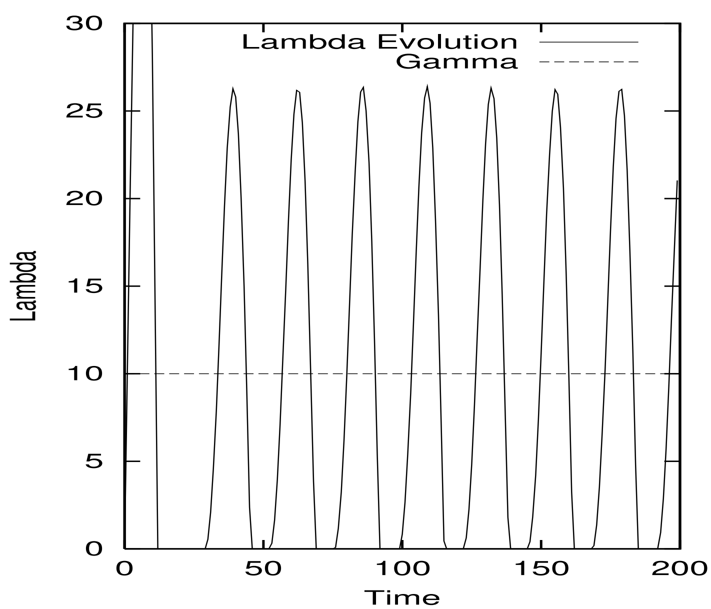
- if $Q^* - Q(t) > 0$, increase $\lambda(t)$ proportional to gap
- if $Q^* - Q(t) < 0$, decrease $\lambda(t)$ proportional to gap

Trying to be more clever...

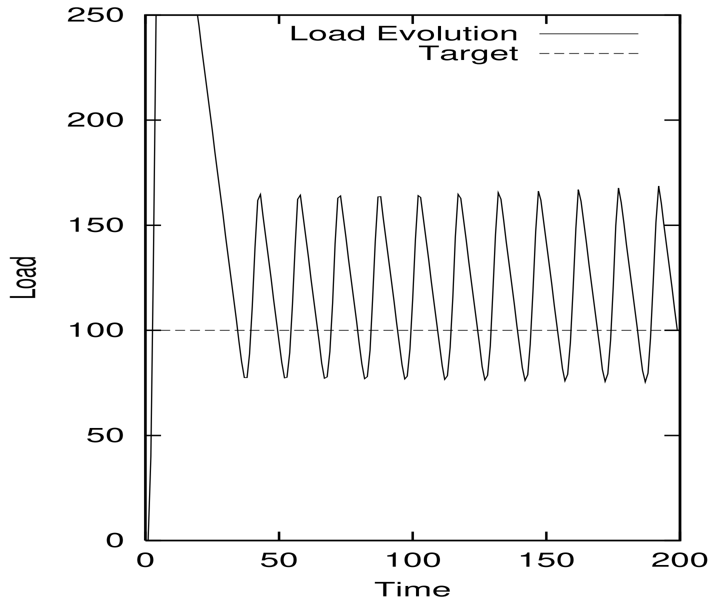
$\longrightarrow$ bottom line: is it any good?

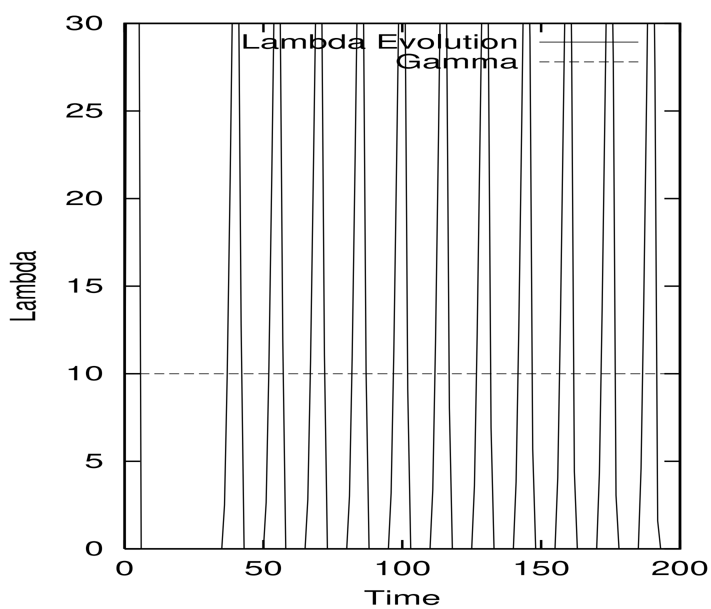
With $\varepsilon = 0.1$:





With $\varepsilon = 0.5$:





Answer: no

$\rightarrow$ control law looks good on the surface

$\rightarrow$ but looks can be deceiving

Time to try something new

$\longrightarrow$ any (crazy) ideas?

`Method D`:

$$\lambda(t+1) \leftarrow \lambda(t) + \varepsilon(Q^* - Q(t)) + \beta(\gamma - \lambda(t))$$

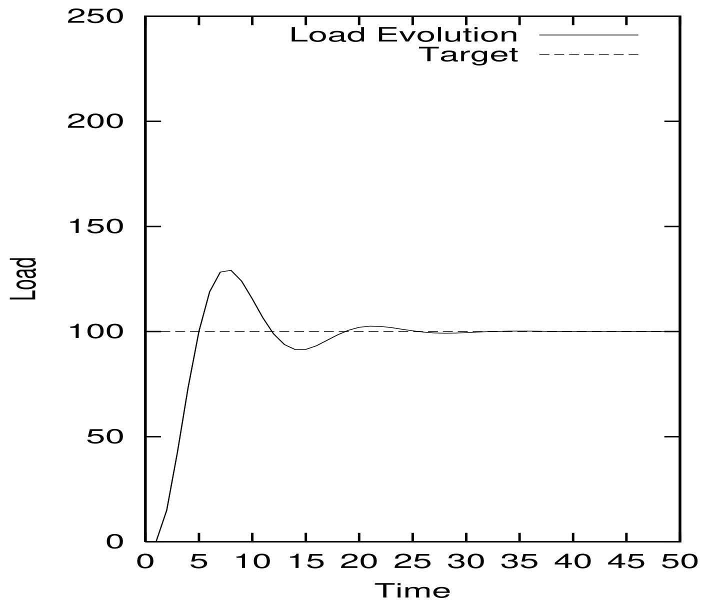
where $\varepsilon > 0$ and $\beta > 0$ are fixed parameters

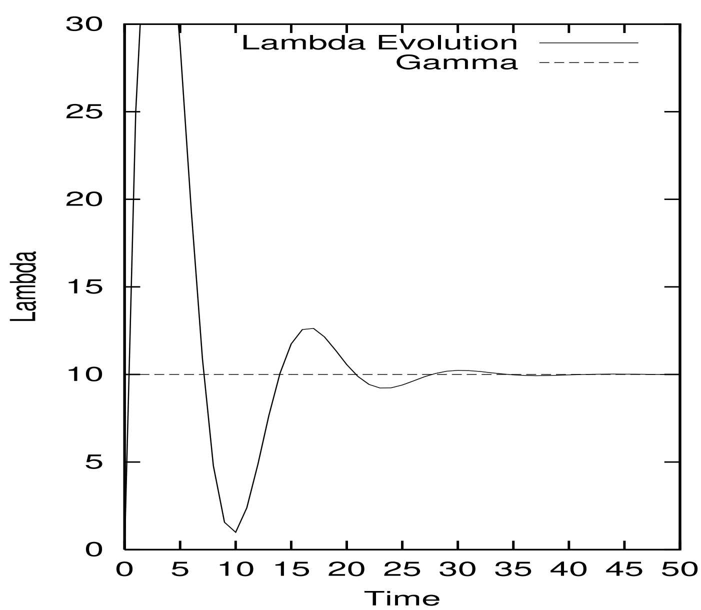
$\longrightarrow$ additional term $\beta(\gamma - \lambda(t))$

$\longrightarrow$ what's going on?

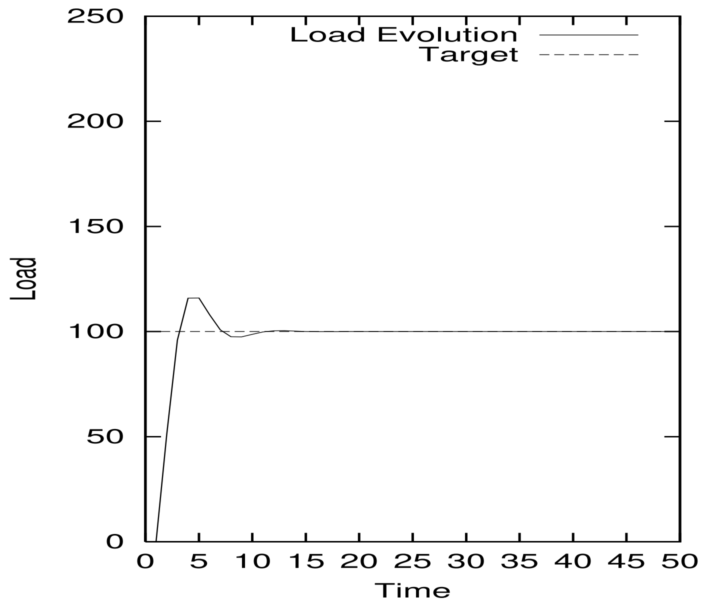
$\longrightarrow$ does it work?

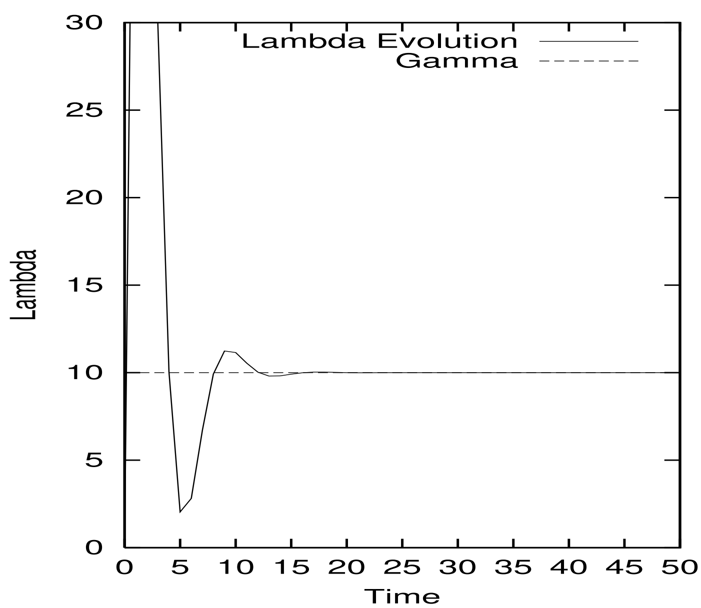
With $\varepsilon = 0.2$ and $\beta = 0.5$:



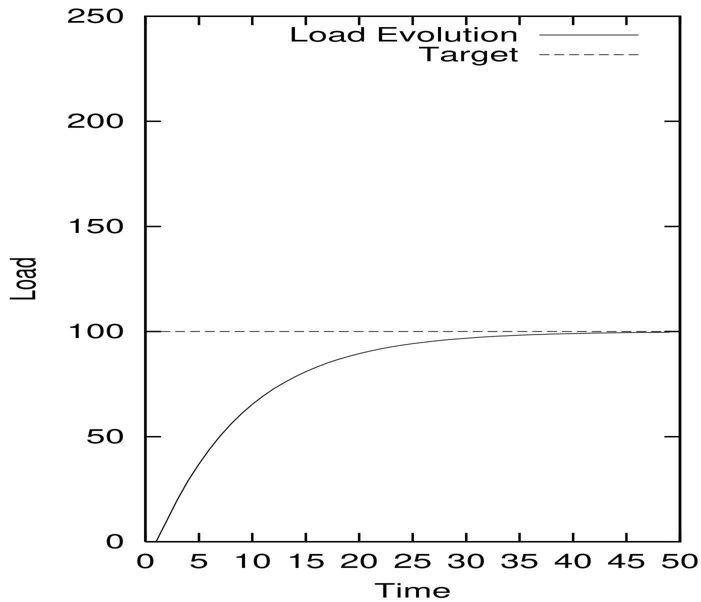


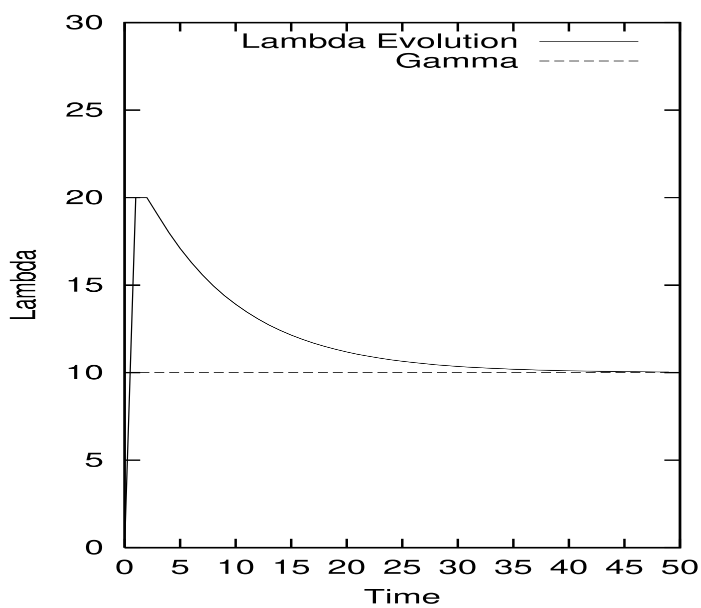
With $\varepsilon = 0.5$ and $\beta = 1.1$:





With $\varepsilon = 0.1$ and $\beta = 1.0$:





Remarks:

- Method D has desired behavior
- Superior to Methods A, B, and C,
- No unbounded oscillation
- Convergence to desired state

  $\rightarrow$ target operating point (Q^*, γ)

  $\rightarrow$ asymptotically stable