

Cognitive Modeling LC2

Harm Manders, 10677186

Opdracht 1

Wat is het Rescola-Wagner model?

De formule:

$$\Delta V = \alpha\beta(V_{max} - V)$$

α is de opvallendheid van de conditioned stimulus.

β is de sterkte van de unconditioned stimulus ofwel beloning, deze wordt meestal op 1 gezet.

V_{max} is de maximale associatieve sterkte.

V is de huidige associatieve sterkte.

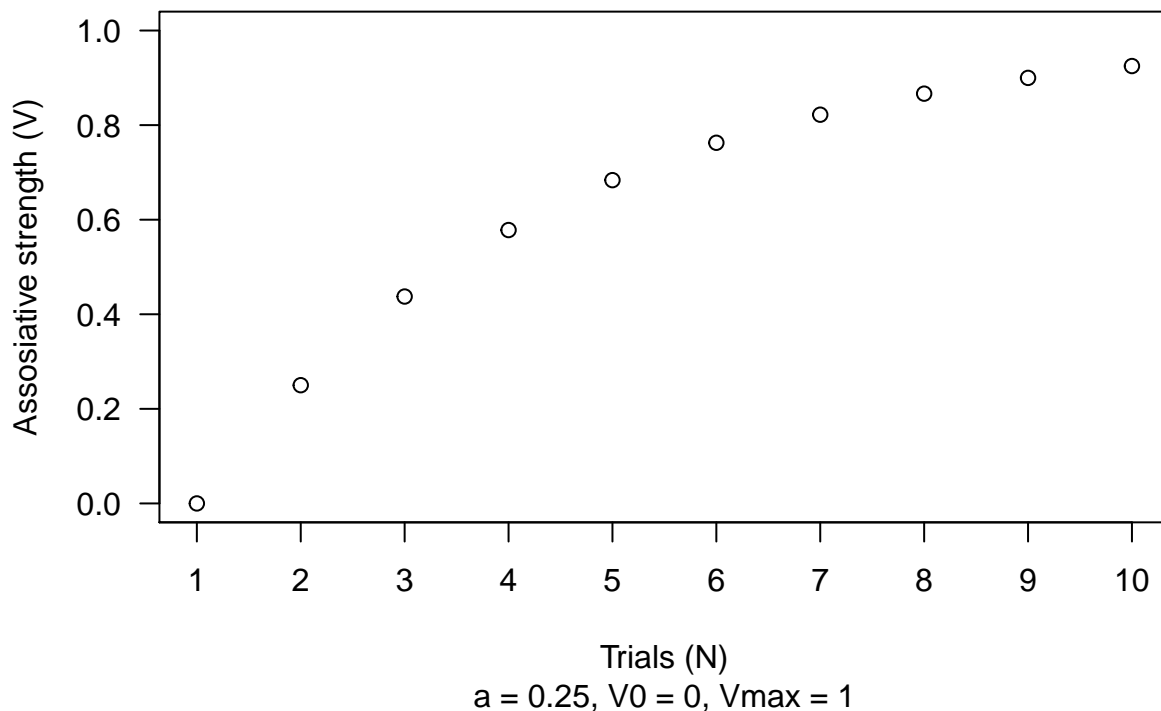
ΔV is het verschil in associatieve sterkte tussen trial t en $t+1$.

$(V_{max} - V)$ wordt ook wel de verrassings functie genoemd. Hoe hoger V is, hoe kleiner ΔV . Dit betekend dat gedurende de trials V steeds hoger wordt en de verrassing steeds lager wordt, de stimulus is dan dus sterker gekoppeld aan de beloning.

α representeerd de learning rate, hoe hoger deze is hoe sneller een stimulus can worden geconditioneerd.

```
v = list(y=rw(0.25,0,1,10)$v)
plt_rw(list(v), "Opdracht 1: Rescola wagner", "a = 0.25, V0 = 0, Vmax = 1")
```

Opdracht 1: Rescola wagner



Opdracht 2

Illustreer de betekenis van de parameters.

De parameter die voornamelijk invloed heeft op de acquisitie curve is α . Deze bepaald namelijk hoe snel de curve richting V_{max} stijgt.

Hoe hoger α hoe sneller de curve bij V_{max} komt. $\Delta V_1 = \alpha$ bij $V_0 = 0$ en $V_{max} = 1$.

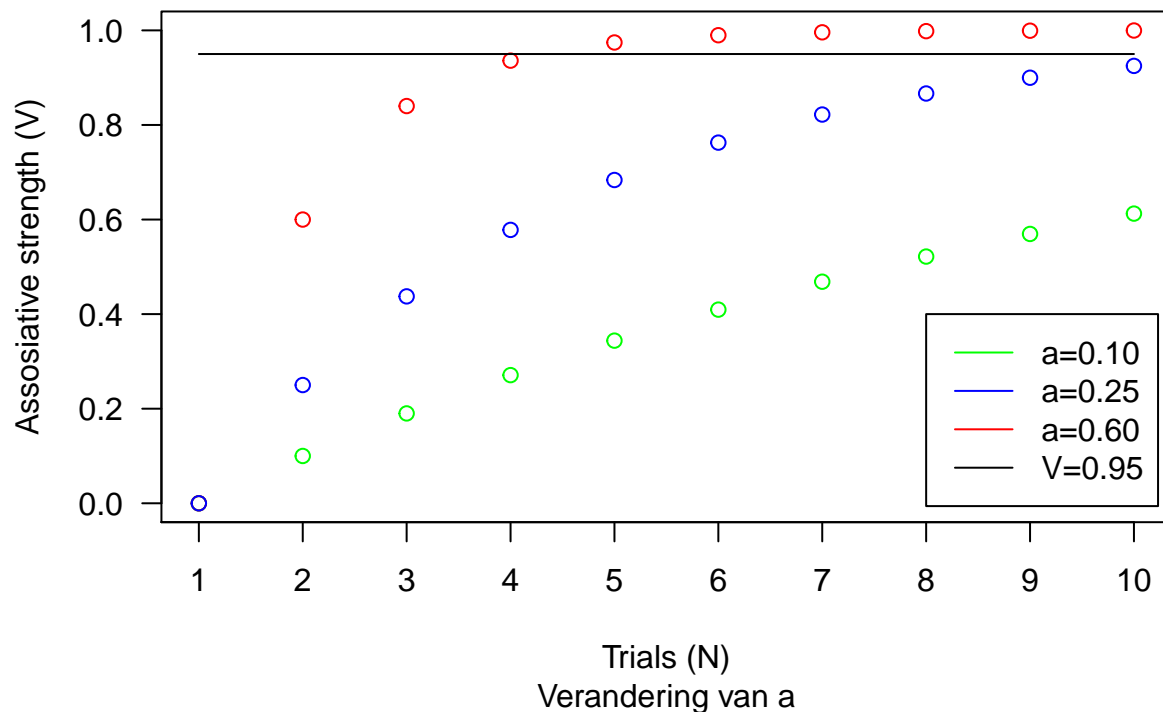
Bij de ΔV 's zie je dat hogere α 's zorgen voor een hogere initiele ΔV maar wel eentje die sneller richting 0 gaat.

$\alpha = 0.6$ gaat als enige binnen 10 trials over de 95% heen de rest van de α 's hebben meer trials nodig.

```
vA = list(y = rw(0.1,0,1,10)$v, col='green')
vB = list(y = rw(0.6,0,1,10)$v, col='red')
vC = list(y = rw(0.25,0,1,10)$v, col='blue')
l = list(y = rep(0.95,10), type='l')

plt_rw(list(vA, vB, vC, l), "Opdracht 2: Associatie sterktes V", "Verandering van a")
legend(8,0.4, legend=c("a=0.10", 'a=0.25', 'a=0.60', 'V=0.95'),
      col=c('green', 'blue', 'red', 'black'), lty = c(1,1,1,1))
```

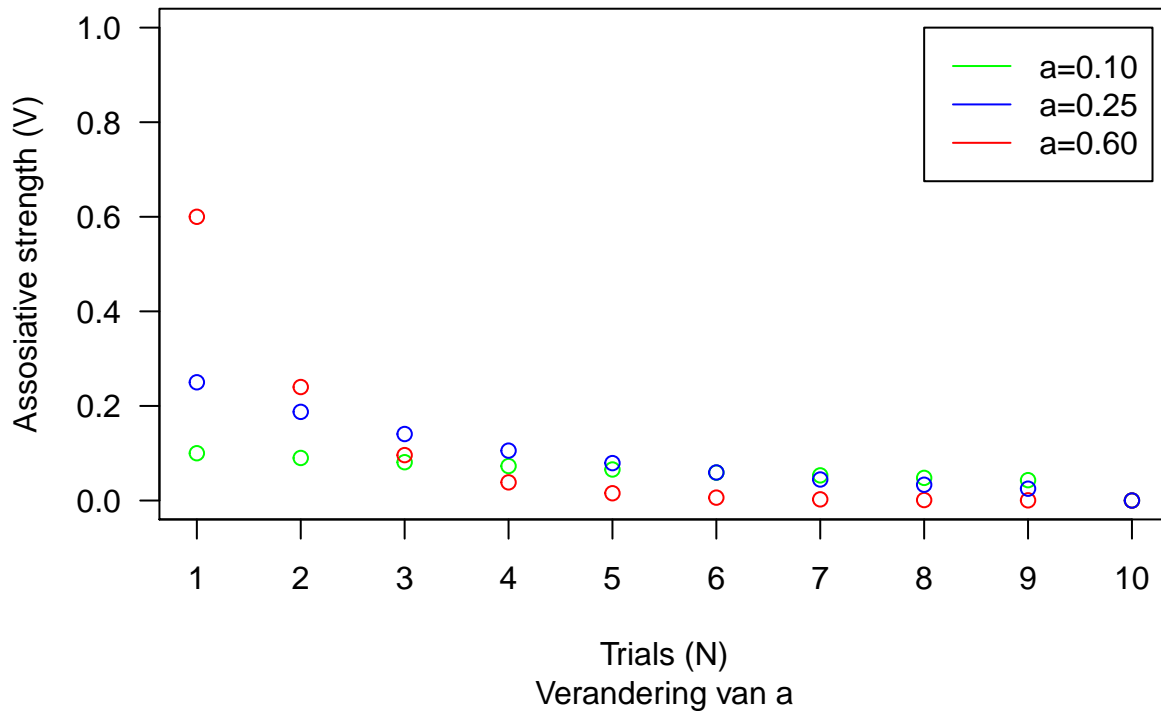
Opdracht 2: Associatie sterktes V



```
vA = list(y = rw(0.1,0,1,10)$dV, col='green')
vB = list(y = rw(0.6,0,1,10)$dV, col='red')
vC = list(y = rw(0.25,0,1,10)$dV, col='blue')
```

```
plt_rw(list(vA, vB, vC), "Opdracht 2: delta V's", "Verandering van a")
legend(8,1, legend=c("a=0.10", 'a=0.25', 'a=0.60'), col=c('green','blue','red'),
      lty = c(1,1,1))
```

Opdracht 2: delta V's

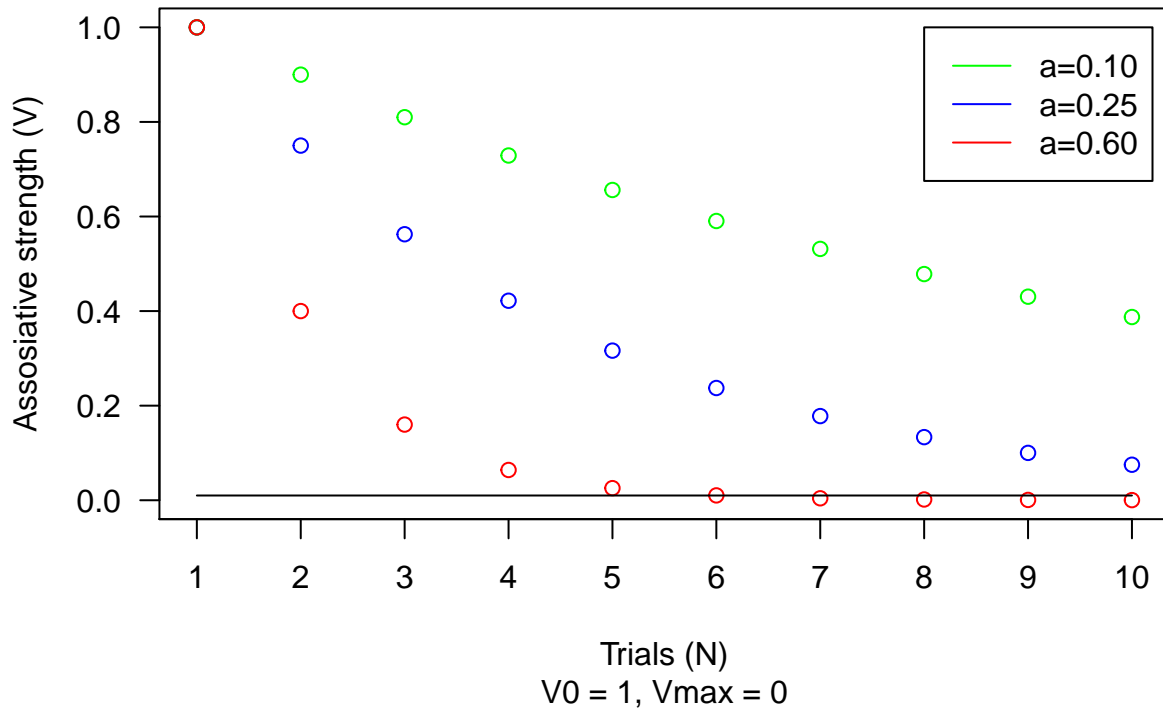


Opdracht 3: Extinctie

Om extinctie te modelleren met het Rescorla-Wagner Model kan je er vanuit gaan dat de stimulus maximaal geconditioneerd is en er geen US meer is en dus dat de curve richting 0 gaat. Bij een extinctie curves geldt dus $V_0 = 1$ en $V_{max} = 0$. Hierdoor is de verrassings functie ($V_{max} - V_t$) negatief en daalt de curve richting 0. De eerste waarde van V is $V_1 = 1 - \alpha$. Hogere α 's gaan sneller naar 0. Alleen $\alpha = 0.6$ komt onder de 1% associatie sterkte na 10 trials, de rest heeft meer trials nodig.

```
ext2 = list(y=rw(0.1,1,0,10)$v, col='green')
ext1 = list(y=rw(0.25,1,0,10)$v, col='blue')
ext3 = list(y=rw(0.6,1,0,10)$v, col='red')
l = list(y = rep(0.01,10), type='l')
plt_rw(list(ext1,ext2,ext3, l), "Opdracht 3: Extinctie curve", "V0 = 1, Vmax = 0")
legend(8,1, legend=c("a=0.10", 'a=0.25', 'a=0.60'), col=c('green','blue', 'red'),
      lty = c(1,1,1))
```

Opdracht 3: Extinctie curve



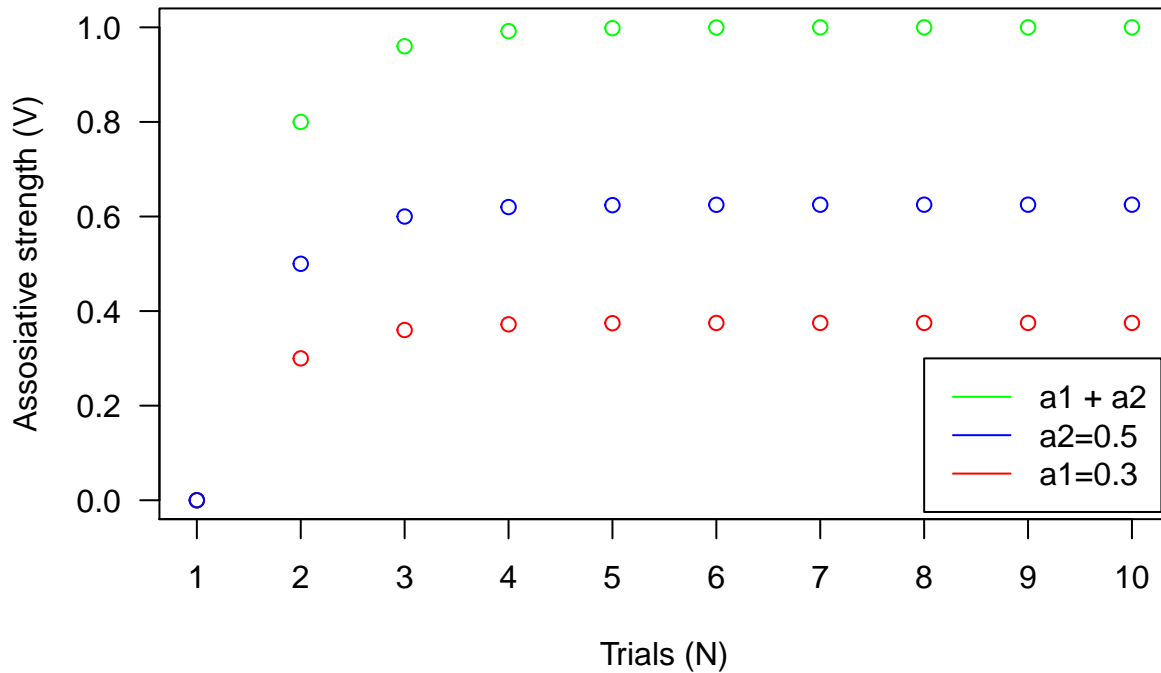
Opdracht 4: Compound stimuli

Bij twee geconditioneerde stimuli die samen resulteren in een geconditioneerde respons, kunnen deze stimuli individueel geplott worden in het Rescorla-Wagner Model. De V_{max} van de individuele stimuli zijn afhankelijk van elkaar en combineren bij een compleet geconditioneerde respons tot 1. De α per individuele stimulus is wel eigen.

De gecombineerde curve is gelijk aan een curve met $\alpha = \alpha_1 + \alpha_2$

```
compd = compound(c(0.3,0.5), 0, 1, 10)
v = list(y = compd$v, col = 'green')
v1 = list(y = compd$v1, col = 'red')
v2 = list(y = compd$v2, col = 'blue')
plt_rw(list(v, v1, v2), "Opdracht 4: Compound stimuli", '')
legend(8,0.3, legend=c('a1 + a2','a2=0.5', 'a1=0.3'), col=c('green','blue','red'),
      lty = c(1,1,1))
```

Opdracht 4: Compound stimuli



Opdracht 5: Blocking

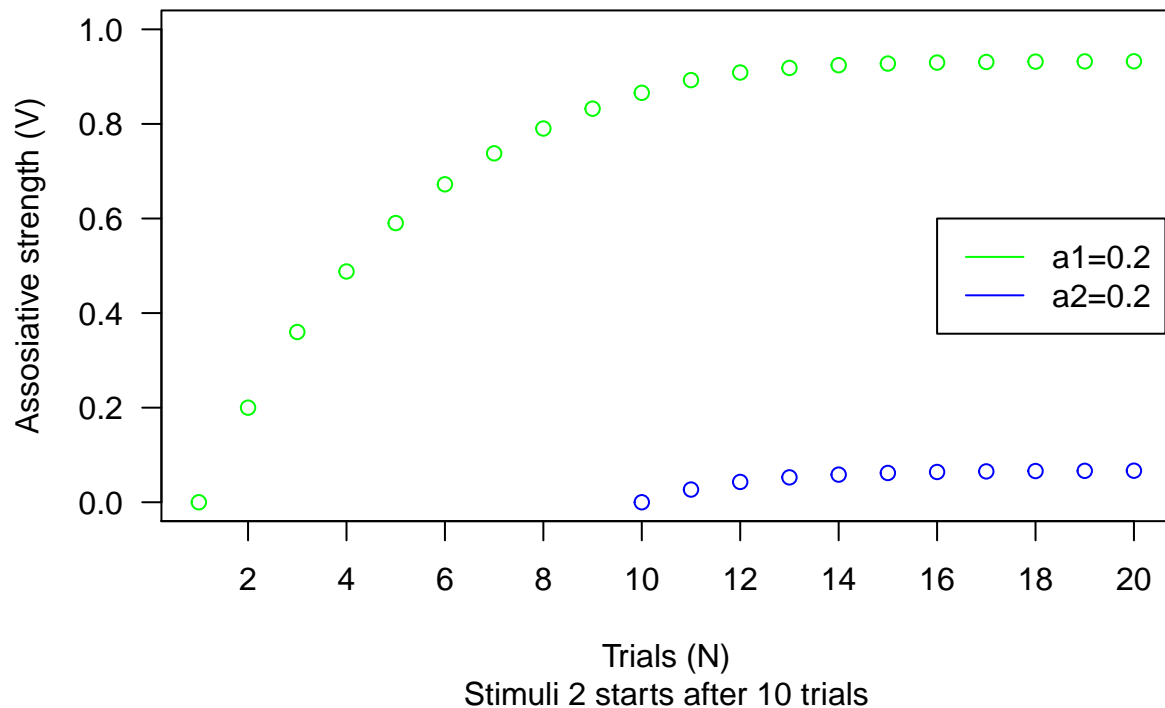
Bij blocking wordt eerst de eerste stimulus geconditioneerd en na een bepaalde tijd pas een tweede. Omdat beide stimuli nooit boven V_{max} uit kunnen komen en de eerste stimulus al tegen V_{max} aan zit zal de tweede stimulus nauwelijks geconditioneerd worden.

Je ziet in de plot dat de tweede stimulus een klein beetje stijgt maar nauwelijks boven de 0.1 uit komt terwijl beiden stimuli een gelijkwaardige α hebben.

Als de tweede stimulus wordt geïntroduceerd na 5 trials zal deze een grotere uiteindelijke sterkte hebben maar nog steeds onder de eerste stimulus blijven.

```
blk = blocking(c(0.2,0.2), 0, 1, 20, 10)
v1 = list(y=blk$v1, x=blk$x1, col='green')
v2 = list(y=blk$v2, x=blk$x2, col='blue')
plt_rw(list(v1, v2), "Opdracht 5: Blocking", "Stimuli 2 starts after 10 trials")
legend(16,0.6, legend=c('a1=0.2', 'a2=0.2'), col=c('green','blue'),
      lty = c(1,1))
```

Opdracht 5: Blocking



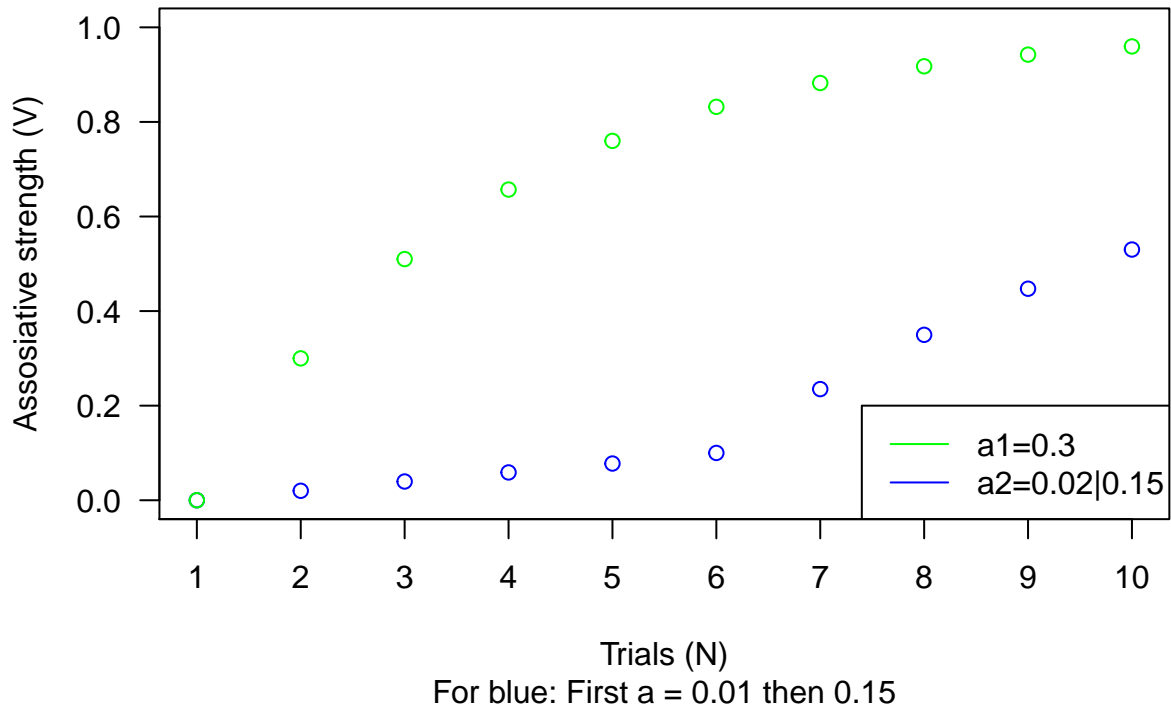
Opdracht 6: Latent inhibition

Bij latent inhibition is de CS al bekend bij de proefpersoon. Hierdoor duurt het langer voordat de proefpersoon de CS linkt aan de US. Dit zou modelleerbaar zijn door de α van de proefpersoon gedurende de trial langzaam omhoog te doen.

```
cont = list(y=rw(0.3, 0, 1, 10)$v, col='green')
vli = list(y=append(rw(0.02, 0, 1, 5)$v,rw(0.15,0.1,1, 5)$v), col='blue')

plt_rw(list(vli, cont), "Latent inhibition","For blue: First a = 0.01 then 0.15")
legend(7.4,0.2, legend=c('a1=0.3', 'a2=0.02|0.15'), col=c('green','blue'),
      lty = c(1,1))
```

Latent inhibition



Appendix I: Functions used

```
# Basic Rescola wagner function
rw <- function(a, v0, vM, trials) {
  v = c(v0)
  dV = c()

  for (t in 1:(trials-1)) {
    dV[t] = a * (vM - v[t])
    v[t+1] = v[t] + dV[t]
  }
  dV = append(dV, 0)
  return(data.frame(v, dV))
}

# Plot multiple data sets
plt_rw <- function(plts, title, sub) {
  x1 = length(plts[[1]]$y)
  plot(0, main=title, sub=sub,
       xlab = "Trials (N)",
       ylab="Assosiative strength (V)",
       xlim = c(1,x1), ylim=c(0,1),
       xaxt="n", yaxt="n", type='l', lty=0)
```

```

for (plt in plts) {
  if (is.null(plt$col)) {
    plt$col = 'black'
  }
  if (is.null(plt$type)) {
    plt$type = 'p'
  }
  if (is.null(plt$x)) {
    plt$x = 1:xl
  }
  lines(plt$x, plt$y, type=plt$type, col=plt$col)
}
if (xl > 10) {
  axis(1, at=seq(0, xl, 2))
}
else {
  axis(1, at=seq(0,xl,1))
}
axis(2, at=seq(0,1,0.2), las=2)
}

```

Compound stimulus

```

compound <- function(as, v0, vM, trials) {
  v = c(v0); v1 = c(v0); v2 = c(v0)
  dV = c(); dV1 = c(); dV2 = c()

  for (t in 1:(trials-1)) {
    dV1[t] = as[1] * (vM - v[t])
    dV2[t] = as[2] * (vM - v[t])
    dV[t] = dV1[t] + dV2[t]
    v[t+1] = v[t] + dV[t]
    v1[t+1] = v1[t] + dV1[t]
    v2[t+1] = v2[t] + dV2[t]
  }
  return(data.frame(v,v1,v2))
}

```

Blocking

```

blocking <- function(as, v0, vM, trials, startB) {
  x1 = 1:trials
  x2 = startB:trials
  v = c(v0); v1 = c(v0); v2 = c(v0)
  dV = c(); dV1 = c(); dV2 = c()
  for (t in 1:(trials-1)) {
    dV1[t] = as[1] * (vM - v[t])
    if (t >= startB) {
      dV2[t] = as[2] * (vM - v[t])
    }
    else {
      dV2[t] = 0
    }
  }
  dV[t] = dV1[t] + dV2[t]
}

```



```

v[t+1] = v[t] + dV[t]
v1[t+1] = v1[t] + dV1[t]
if (t >= startB) {
  v2[t+1] = v2[t] + dV2[t]
}
else {
  v2[t+1] = 0
}
}
return(list(x1=x1, x2=x2, v1=v1, v2=v2[startB:trials]))
}

```