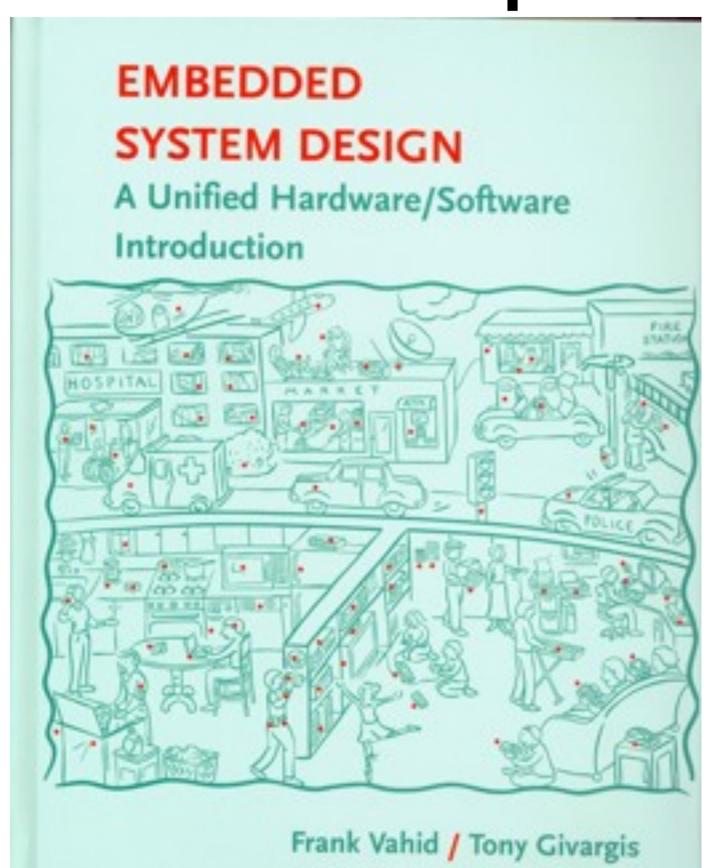
Introduction to control theory

Reference

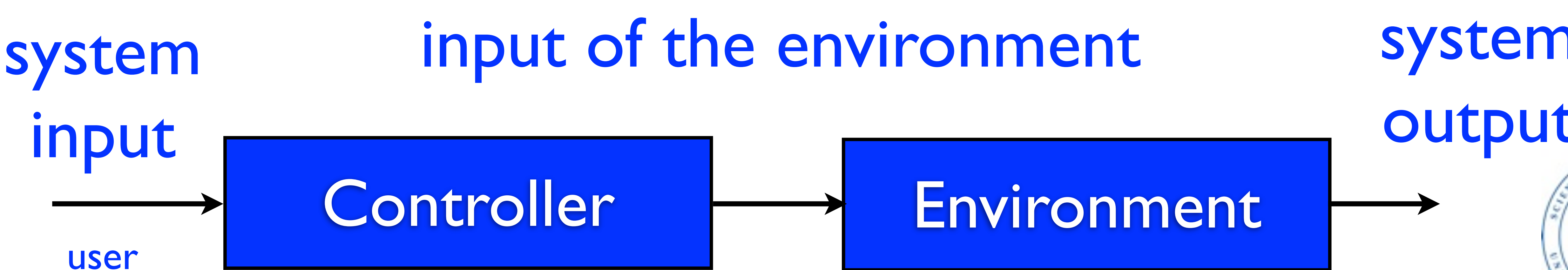
- These slides are freely adapted from the book *Embedded Systems design - A unified Hardware/Software introduction*, Frank Vahid, Tony Givargis

- Some figures are excerpt of the book



Controller

- A controller is a system acting on a physical device (called environment)
  - The controller has to ensure that the system output (system = environment + controller) respects a given property
  - The controller controls the input of the environment



Controller

- Example: a car "*cruise control*" system

Desired speed → Controller → Environment = car → Car Speed

Accelerator

The controller has to ensure that the car speed is as close as possible to the desired speed

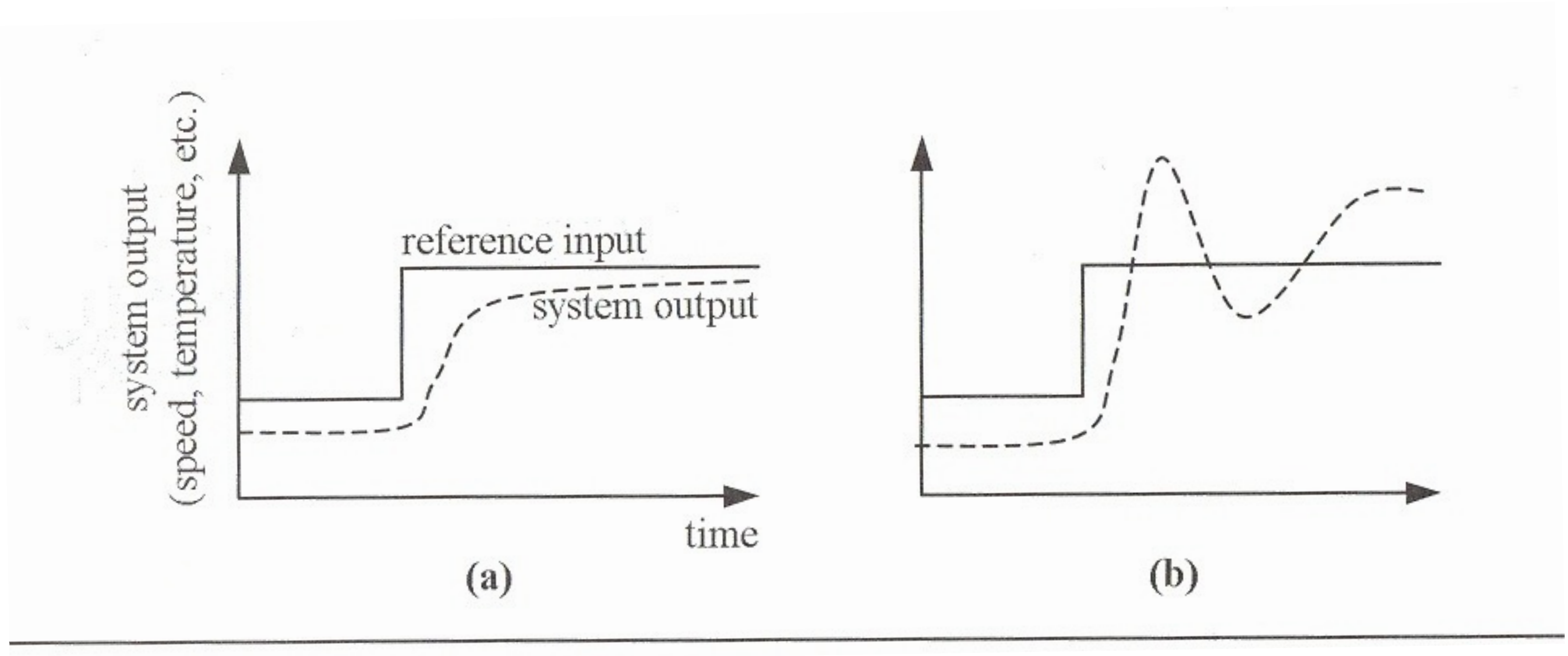
Controller quality

- What is a good controller ?
  - The definition we have given is vague.
    - A device that keeps the speed equal to 50 km per hour all the time is a controller...
  - We will allow the actual speed to be "more or less different" from the requested speed

Controller quality

- What is a good controller ?
  - In this chapter, we will not provide a formal definition of a "good controller" (in contrast with next chapter)
  - The quality will be assessed by criteria that are not strict, mainly by testing.
  - We will often need compromises...

Controller quality



"good" controller

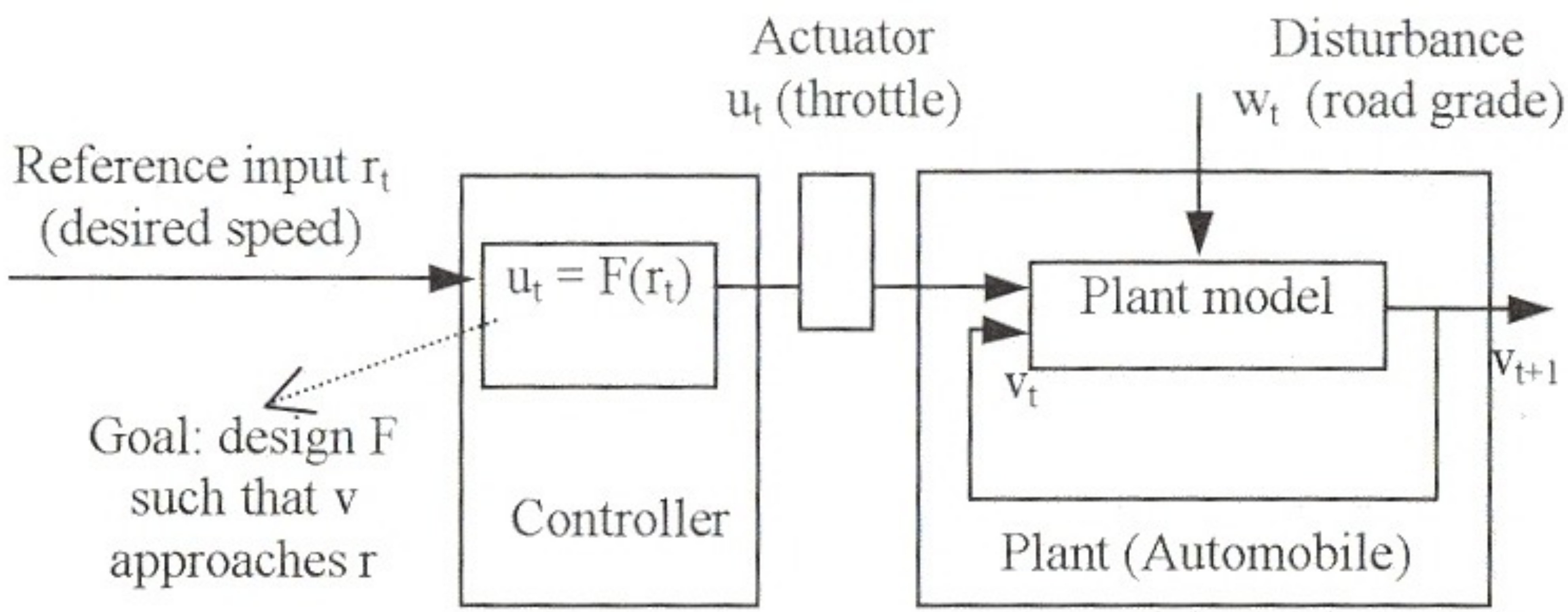
"acceptable" controller

Vocabulary

- The reference is the desired value for the output
- An actuator is a device that allows to control the input of the environment (e.g.: accelerator pedal)
- A disturbance is an uncontrolled parameter of the environment
  (e.g.: road grade)
- A sensor is a device to measure the output of the environment (e.g.: speedometer)

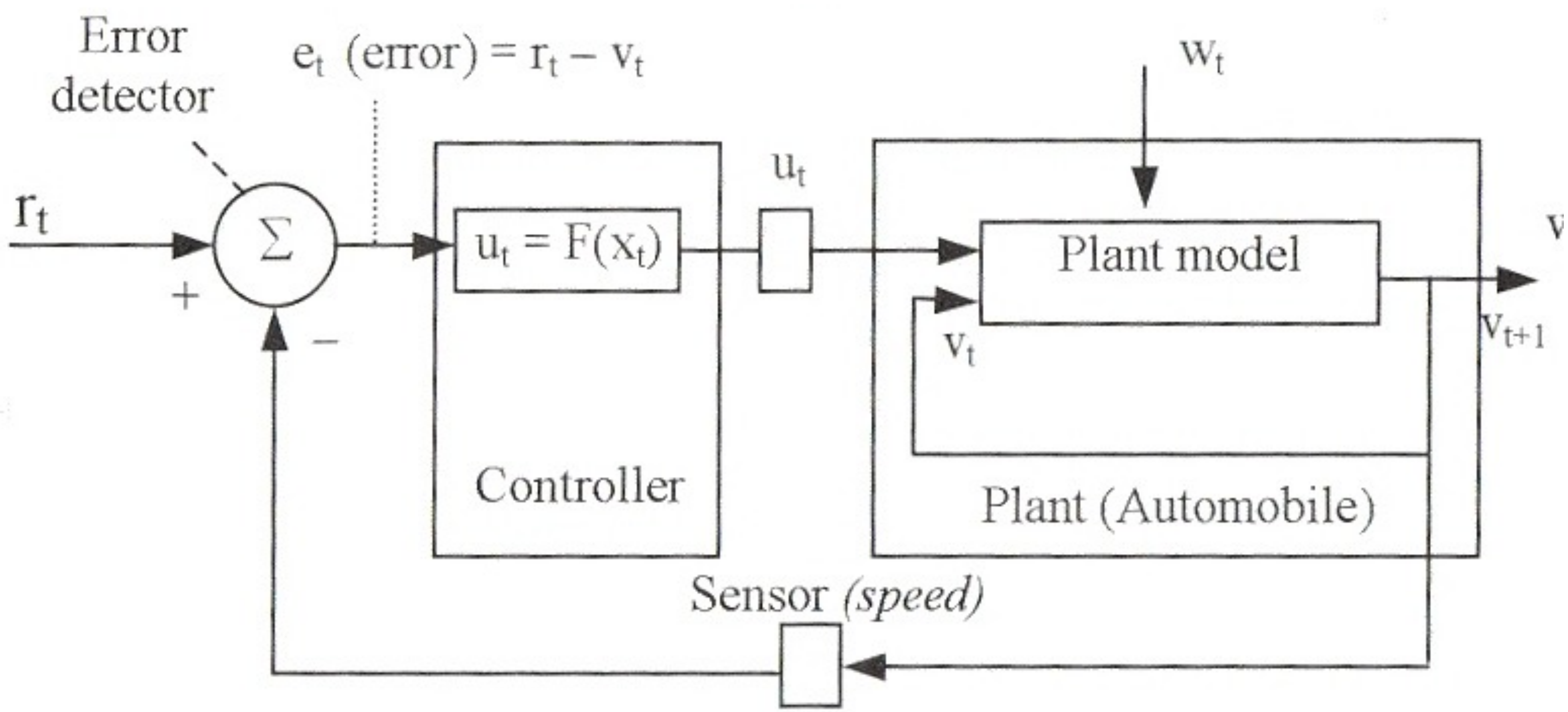
Open loop systems

Speed a time t+1 depends on the speed at time t, the input and the disturbance



The controller as no information about the car's actual speed

Closed loop systems



The controller knows the car's speed.
It can use this information to adapt its output.

Controller - flows

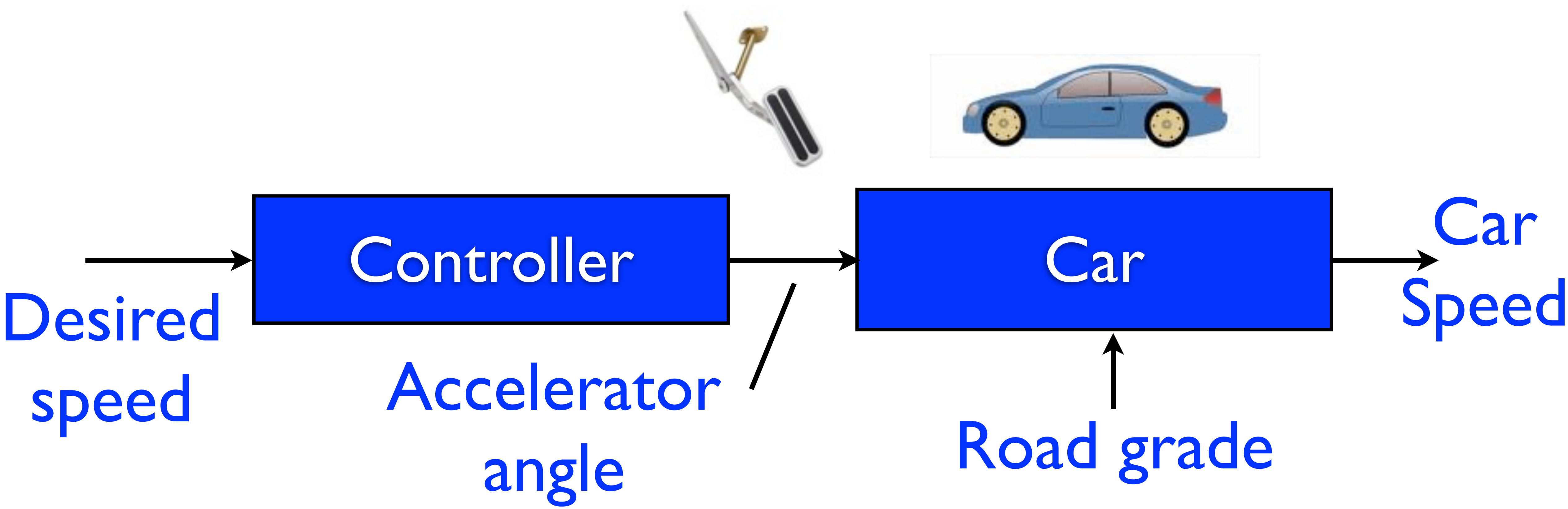
- In this chapter, we'll see the data as "à la Lustre" flows
- The controller and the environment are thus flow modifiers
- Example:
  - r = flow that defines the ref. speed
  - r_t = reference at time t

Controller - flows

- Some flows will be regarded as constant
- We'll abuse notation and use the flow name instead of its (constant) value
- Example:
  - the road grade will be regarded as a constant
  - We have $w_t = w$ for all t

Open loop controller

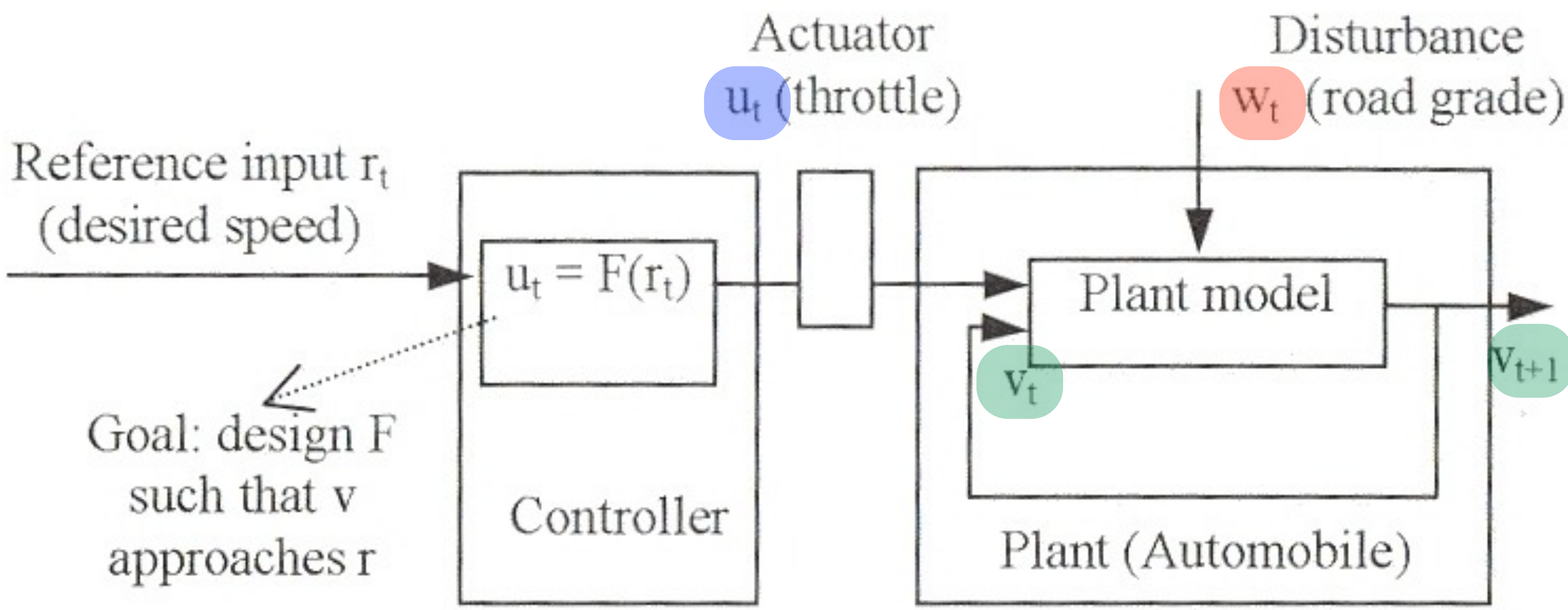
Running example



Flows at work in the example

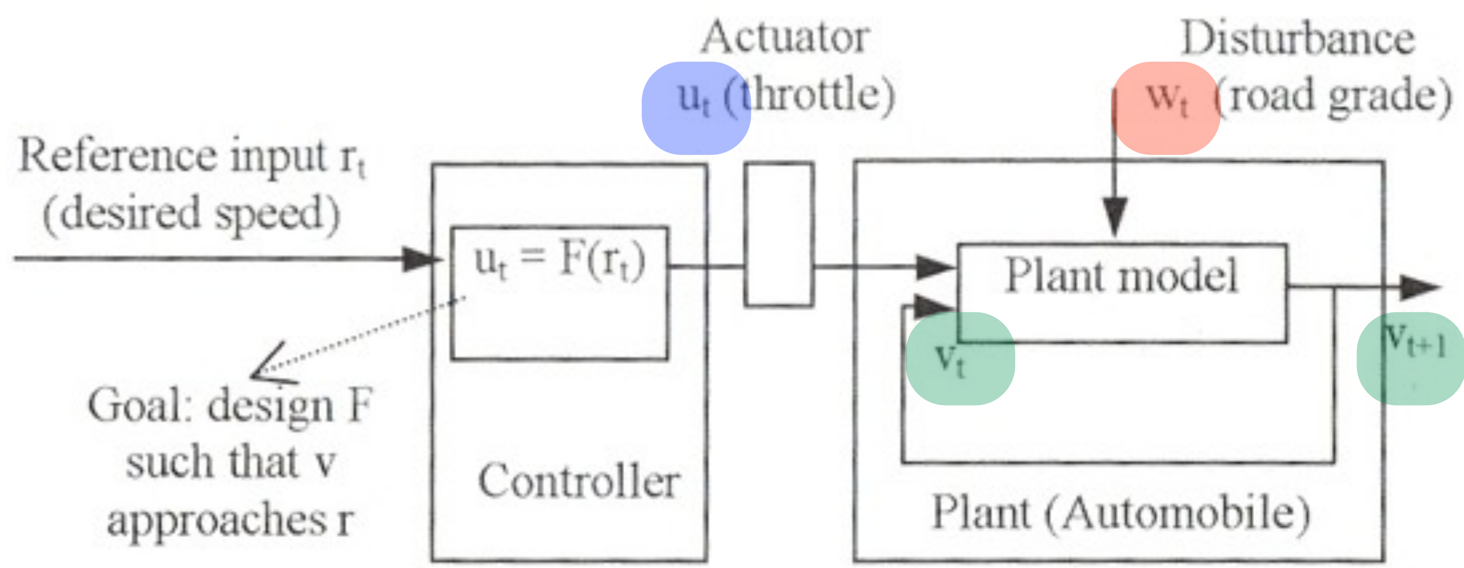
- r_t = reference speed (constant)
- u_t = controller output = input to the car = accelerator position
- w_t = road grade (constant)
- v_t = actual speed car
- e_t = error = $r_t - v_t$

Open loop



- We first need a model for the car
  - How is v_{t+1} computed from v_t, u_t and w_t?

Open loop



- Let us fix $v_{t+1} = 0{,}7v_t + 0{,}5u_t$
- We ignore disturbances for the moment

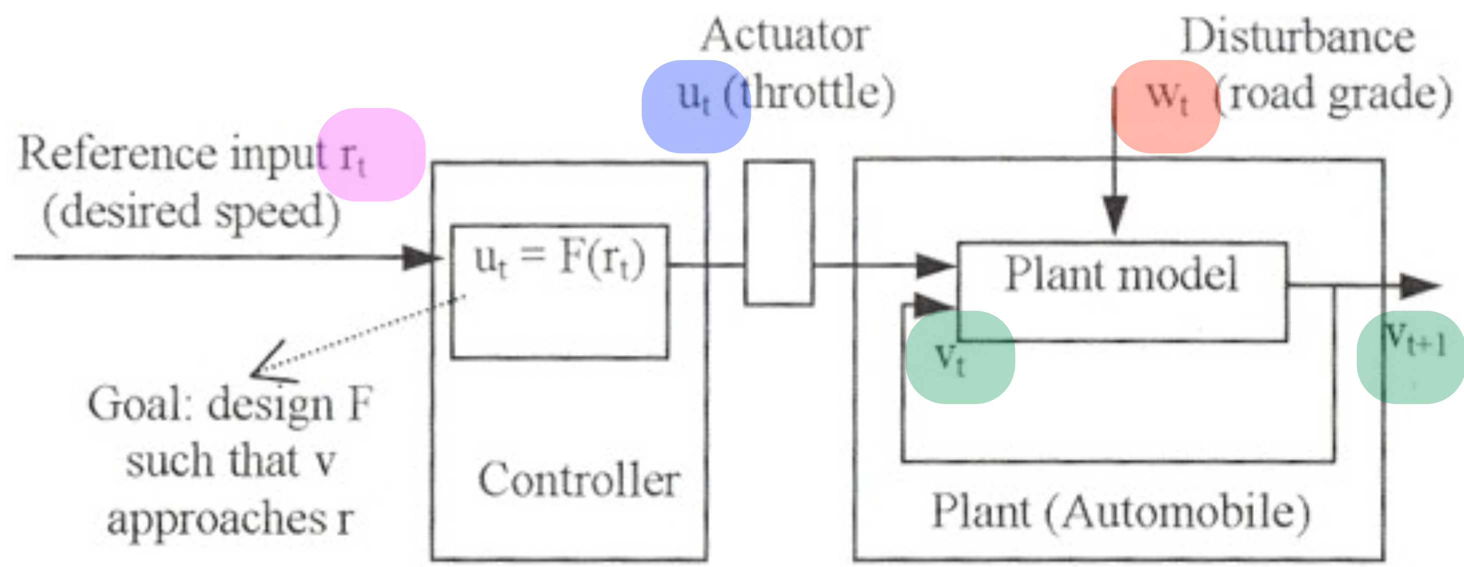
Car model

- $v_{t+1} = 0{,}7v_t + 0{,}5u_t$
  - At each time the speed depends on the speed at previous step and the position of the accelerator pedal
  - For each speed v, there exists a pedal position that allow to maintain v
  - e.g. $v_t = v_{t+1} = 50 \rightarrow u_t = 30$

Car model

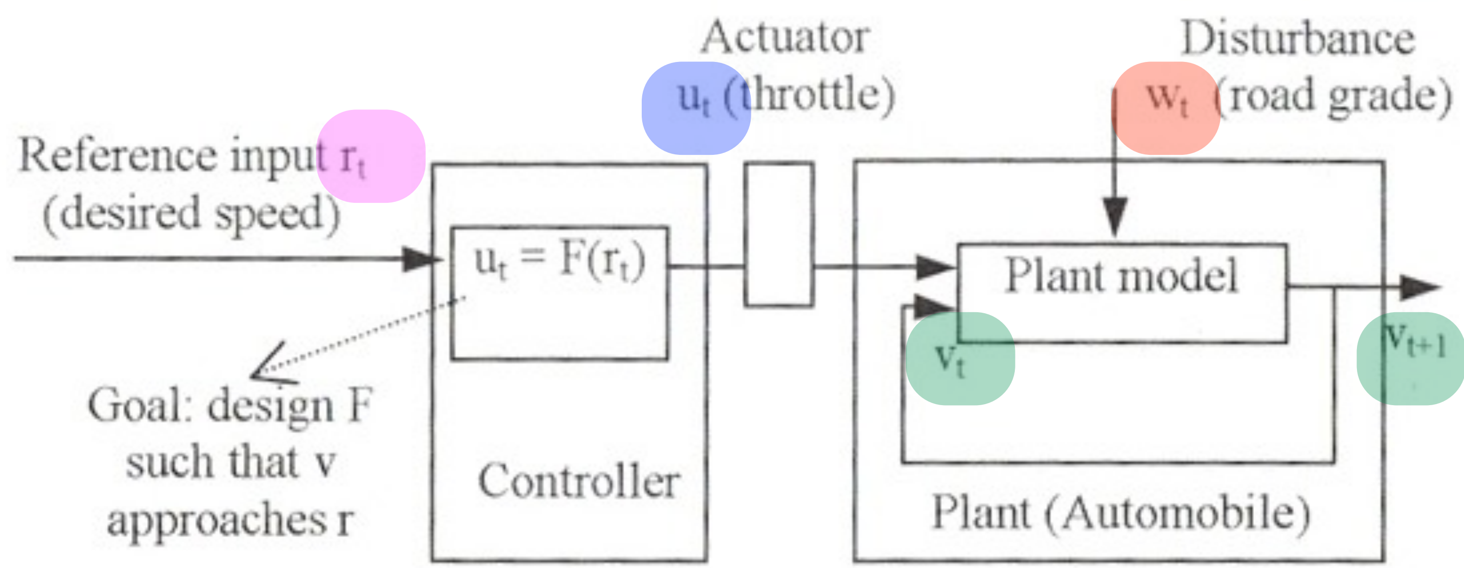
- $v_{t+1} = 0{,}7v_t + 0{,}5u_t$
  - This can be simulated in SCADE... (file modele-voiture.avi)

Open loop



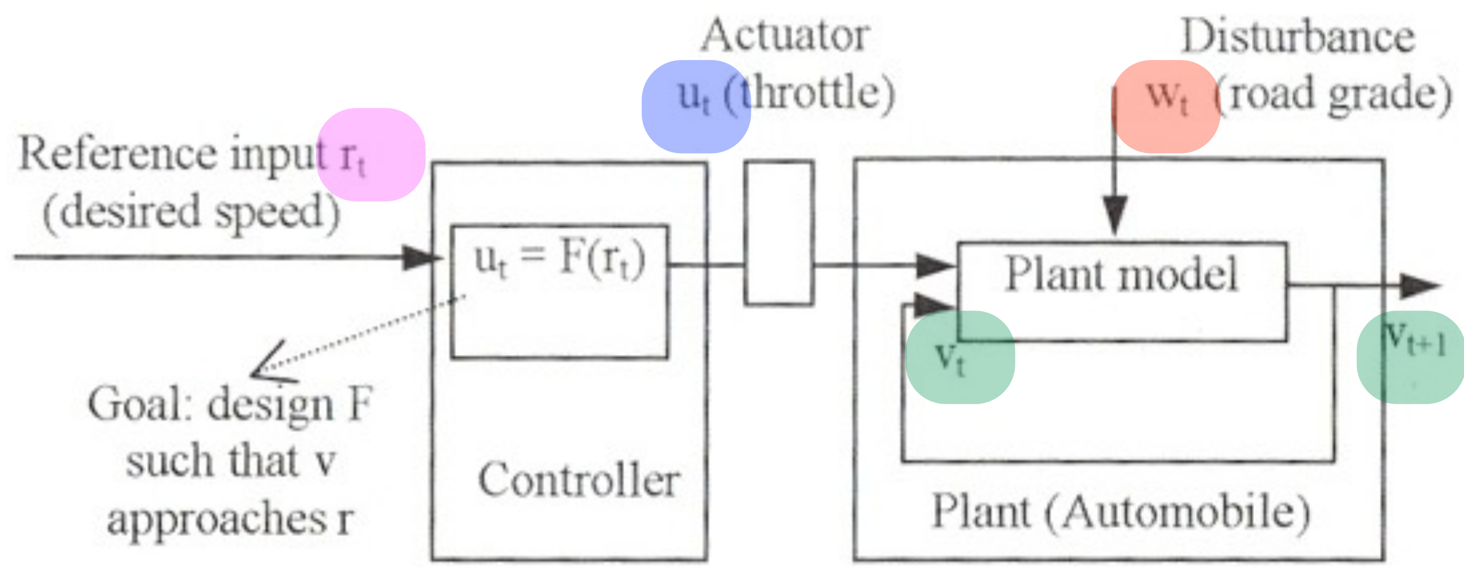
- Let us now design a controller $F(r_t)$, that, depending on r_t, returns u_t to be fed to the environment

Open loop



- We'll start with a simple linear function $F(r_t) = P \times r_t$

Open loop



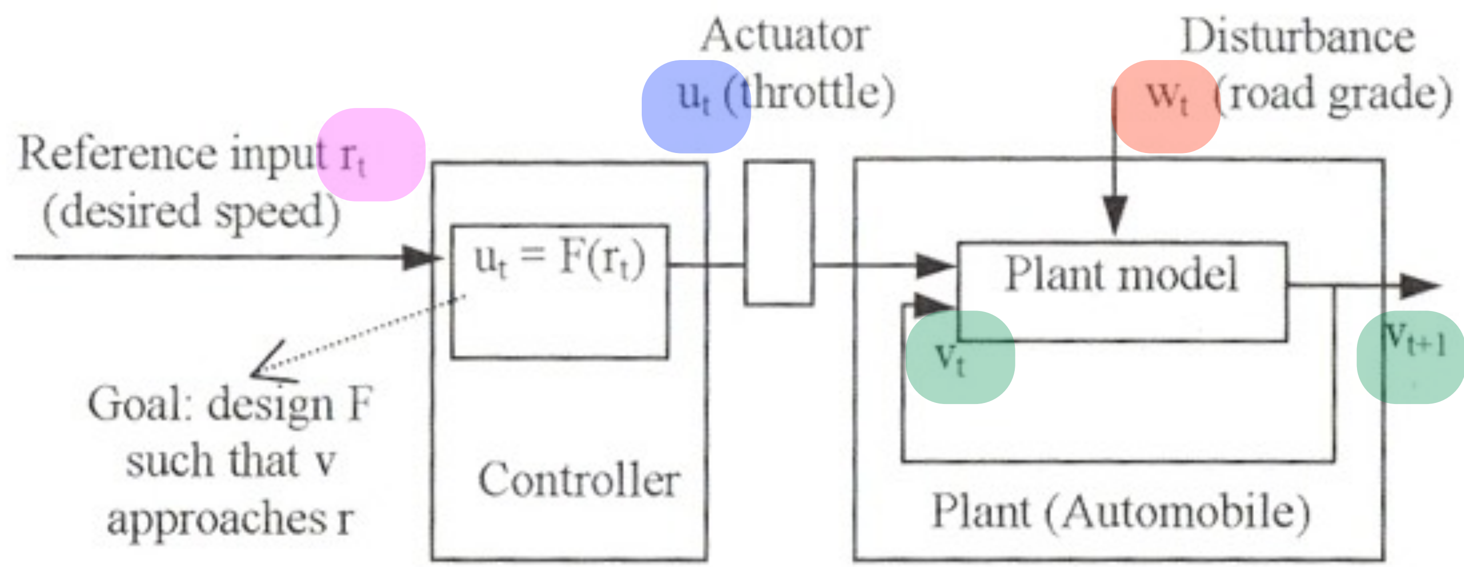
$$\left.\begin{array}{c} F(r_t) = P \times r_t \\ u_t = F(r_t) \end{array}\right\} \text{Controller}$$

$$\left. v_{t+1} = 0.7 \times v_t + 0.5 \times u_t \right\} \quad \text{Car}$$

Hence:

$$v_{t+1} = 0.7 \times v_t + 0.5 \times P \times r_t$$

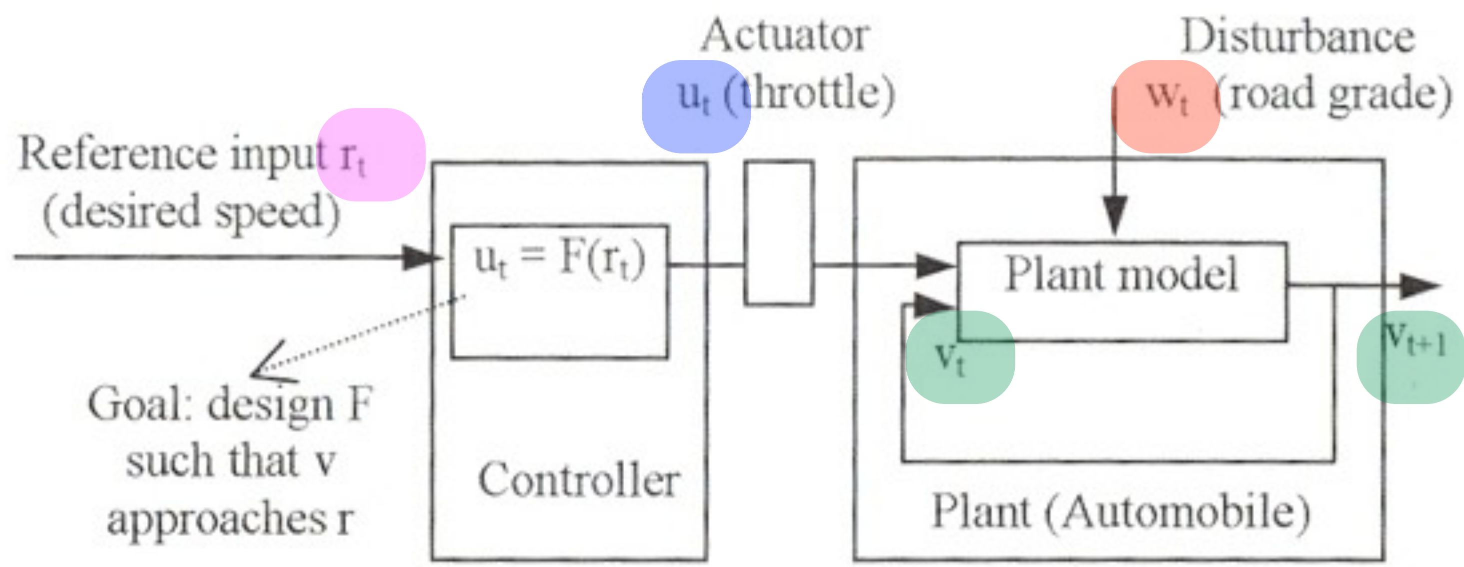
Open loop



$$v_{t+1} = 0.7 \times v_t + 0.5 \times P \times r_t$$

- Now, we have to fix a control objective
- We can't impose $v_{t+1} = r_t$ for any t, because the car takes time to accelerate or slow down

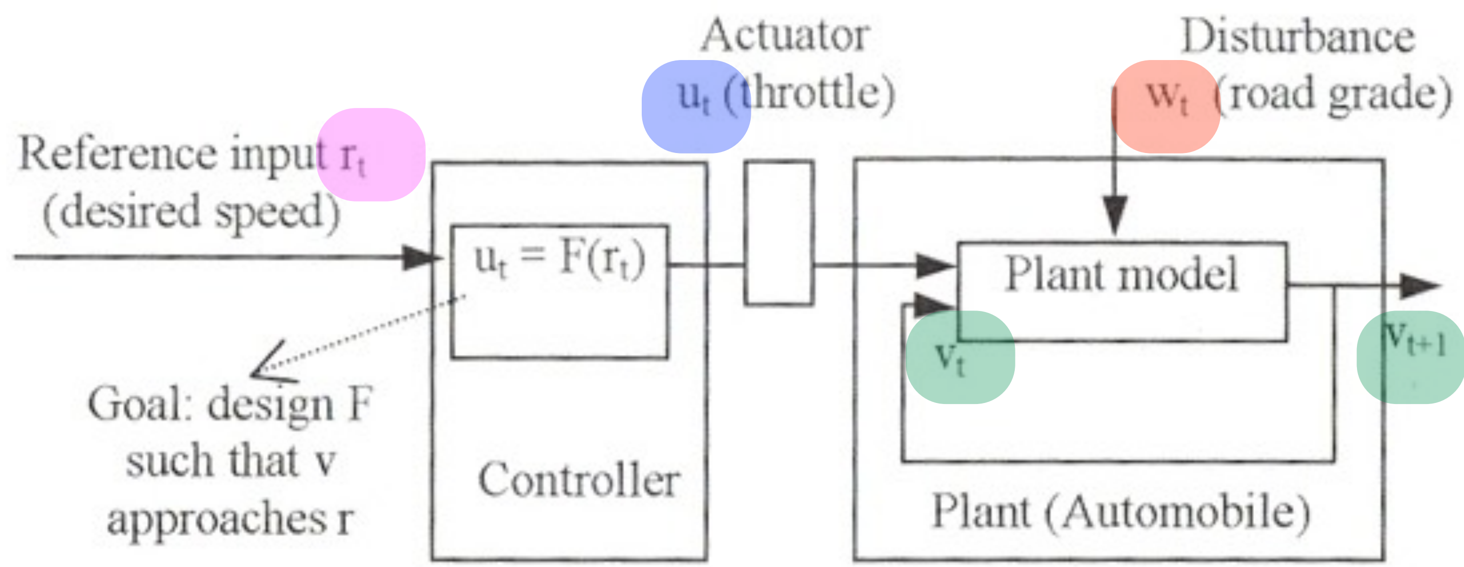
Open loop



$$v_{t+1} = 0.7 \times v_t + 0.5 \times P \times r_t$$

- Now, we have to fix a control objective
- Let us assume $r_t = r$ for any t
- The control objective will be: the output speed must be equal to the reference if the output is stable, that is if $v_t = v_{t+1} = v_{ss}$

Open loop



$$v_{t+1} = 0.7 \times v_t + 0.5 \times P \times r$$

$$v_{t+1} = v_t = v_{ss}$$

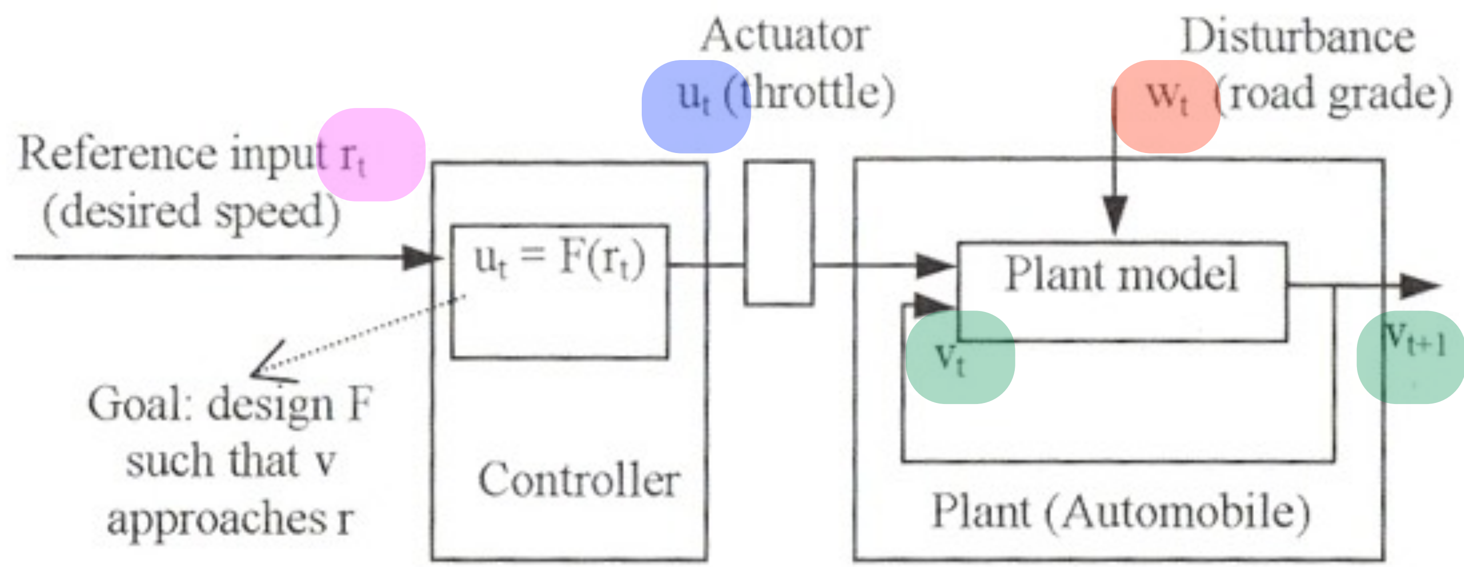
Hence:

$$v_{ss} = 0.7 \times v_{ss} + 0.5 \times P \times r$$

Hence:

$$v_{ss} = 1.67 \times P \times r$$

Open loop

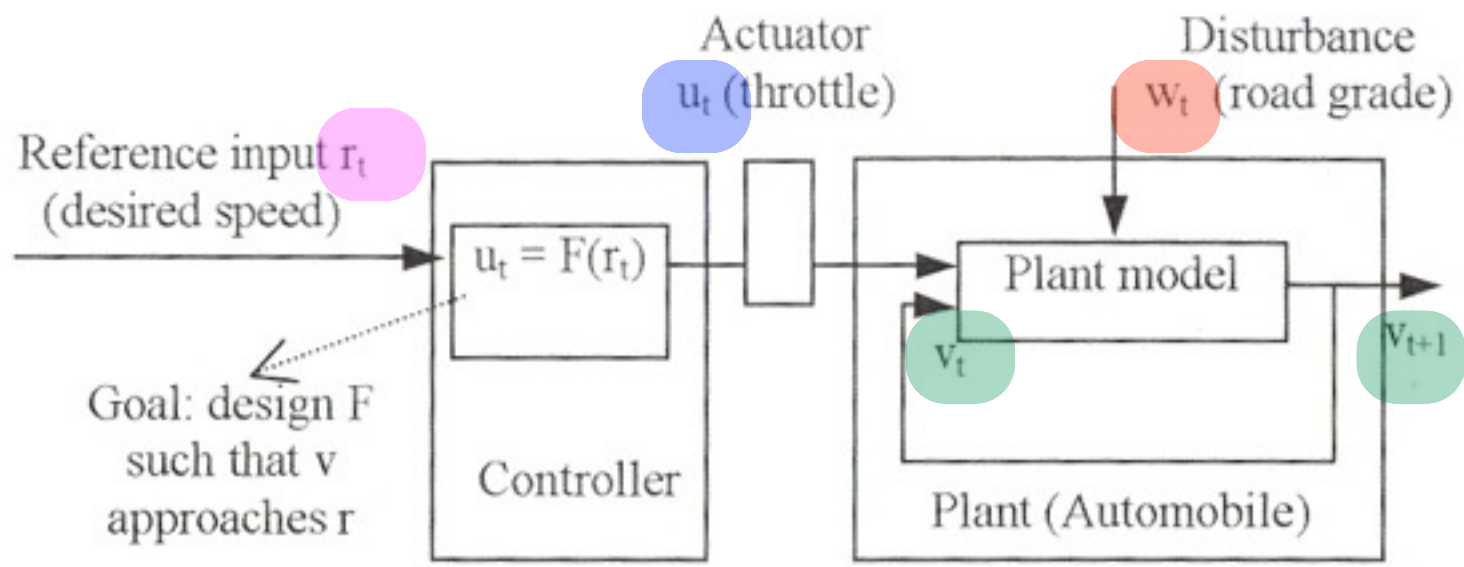


v_{ss}=1.67 × P × r

That equation describes all the speeds
for which we want
the output speed to be equal to the reference speed:

v_{ss} = r

Open loop



$v_{ss} = 1.67 \times P \times r$

We want: $v_{ss} = r$

Thus, we choose $P = 1/1.67 = 0.6$

Example

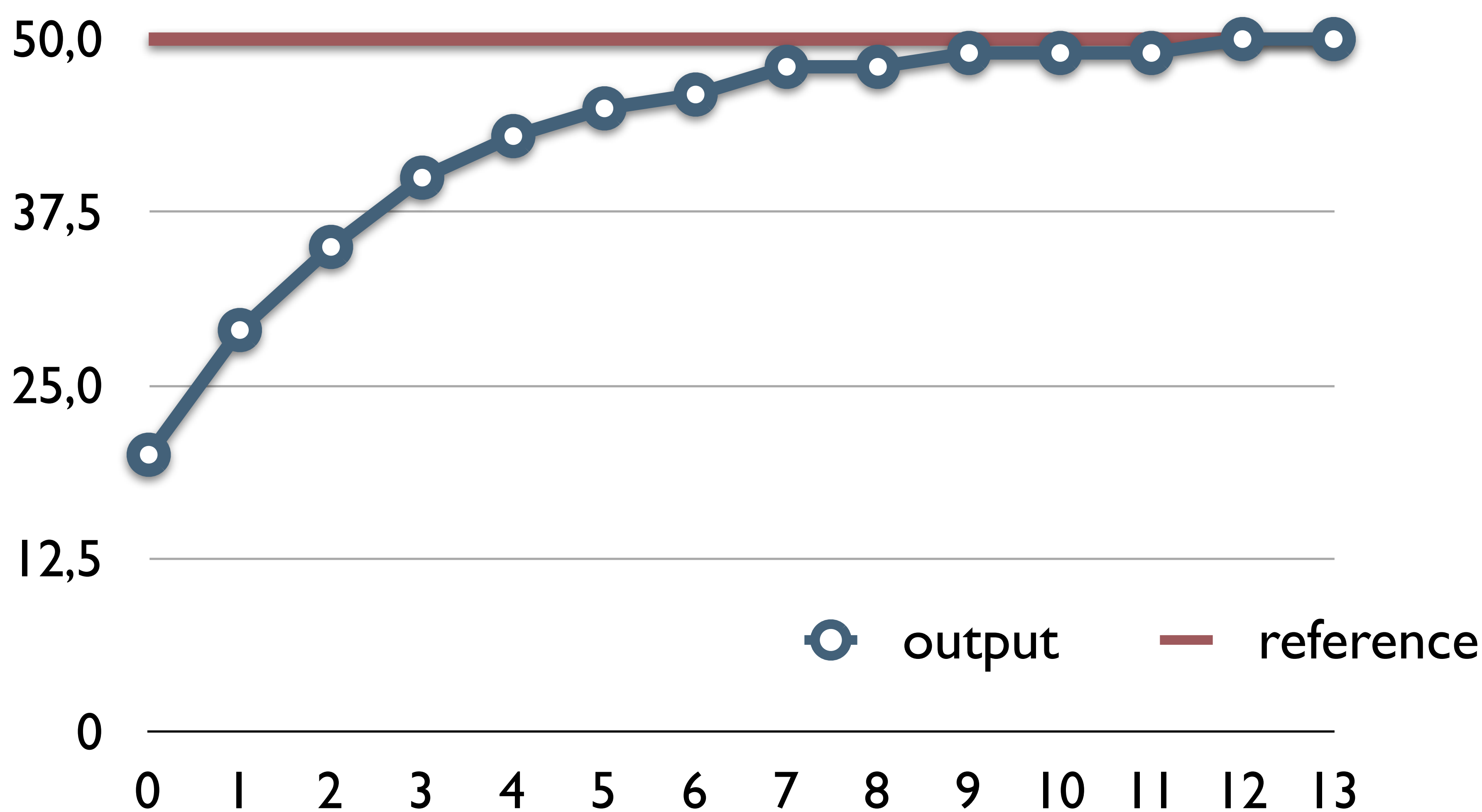
- Initial speed is 20 km/h, reference speed is r=50 km/h.

- $v_{t+1}=0.7 \times v_t+0.5 \times P \times r$ with P=0.6

t	v_t	t	vt
0	20	7	47,529371
1	29	8	48,27056
2	35,3	9	48,789392
3	39,71	10	49,152574
4	42,797	11	49,406802
5	44,9579	12	49,584761
6	46,47053	13	49,709333

Example

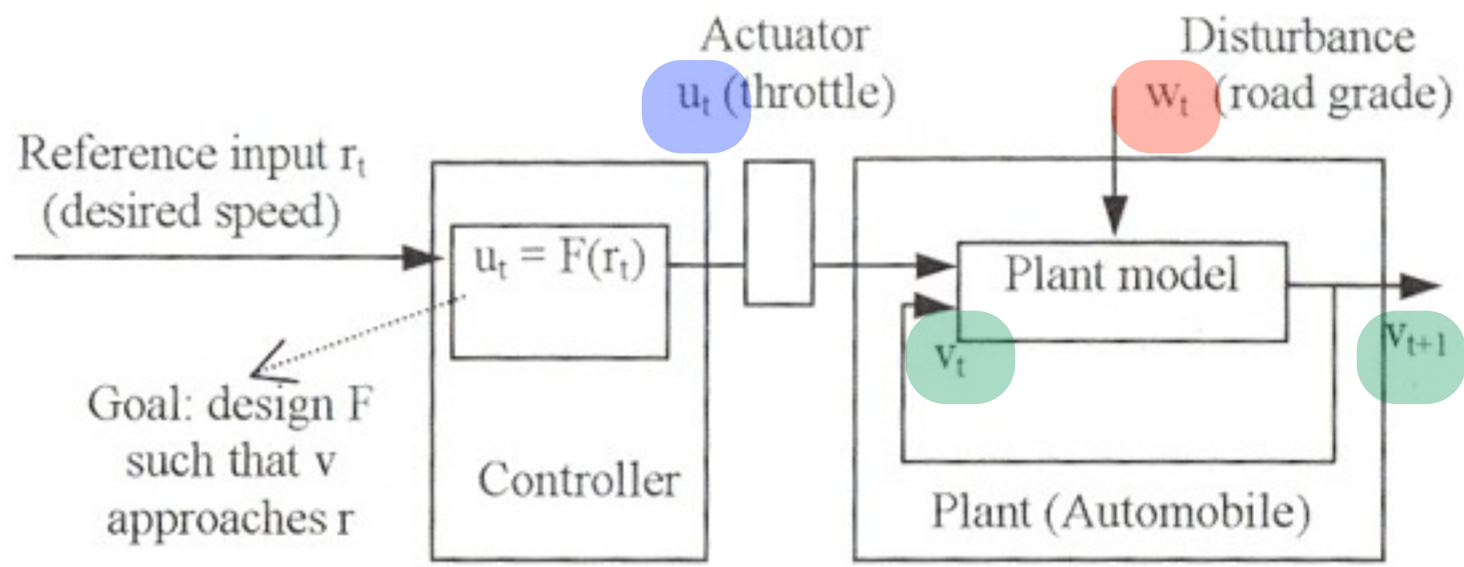
- Initial speed is 20 km/h, reference speed is r=50 km/h.
- $v_{t+1}=0.7 \times v_t+0.5 \times P \times r$ with P=0.6



Drawbacks

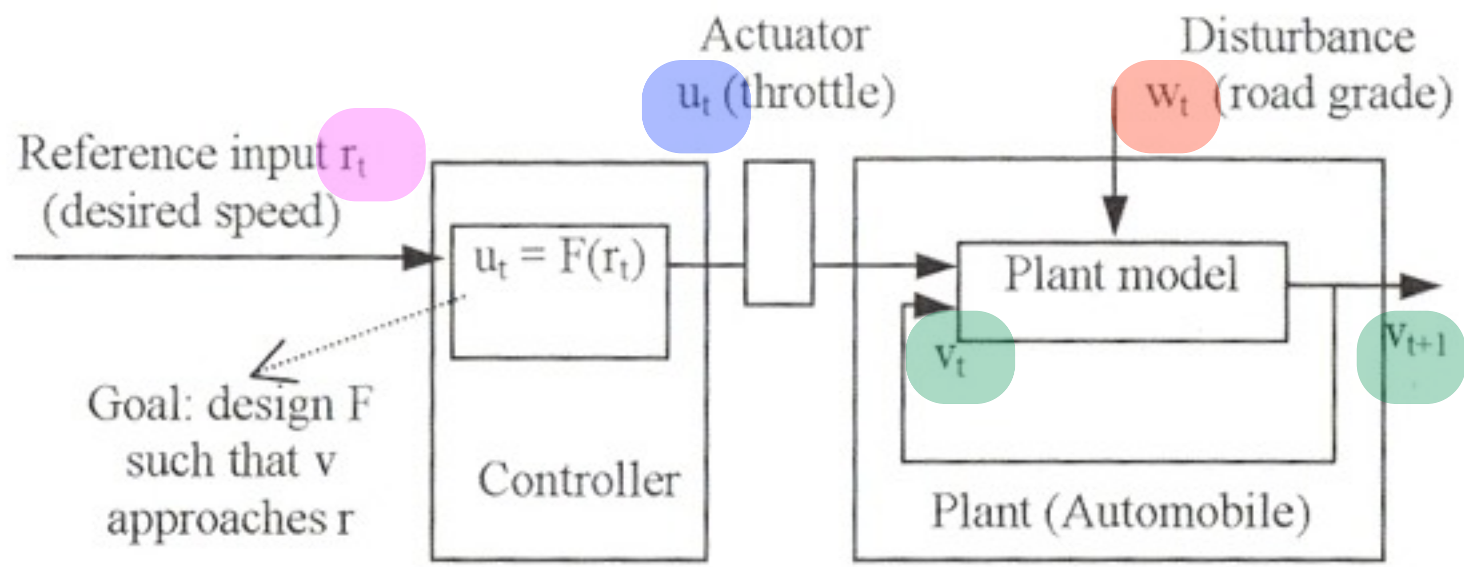
- That example works well because the car is highly predictable
  - The acceleration will be the same in every possible condition
  - In practice, this is not true !
- We will now refine the model to take disturbances into account

Open loop



- New model: $v_{t+1}=0.7v_t+0.5u_t - w_t$
- When the road grade is negative, the acceleration is higher

Open loop



- New model: $v_{t+1}=0.7v_t+0.5u_t - w_t$
- With the same controller we get: $v_{t+1}=0.7 \times v_t + 0.3 \times r_t - w_t$

Example

- Initial speed is 20 km/h, reference speed is r=50 km/h.

- $v_{t+1}=0.7 \times v_t + 0.3\, r - w$ for $w = \mathbf{5}$

t	v_t	t	vt
0	20	7	32,235276
1	24	8	32,564693
2	26,8	9	32,795285
3	28,76	10	32,9567
4	30,132	11	33,06969
5	31,0924	12	33,148783
6	31,76468	13	33,204148

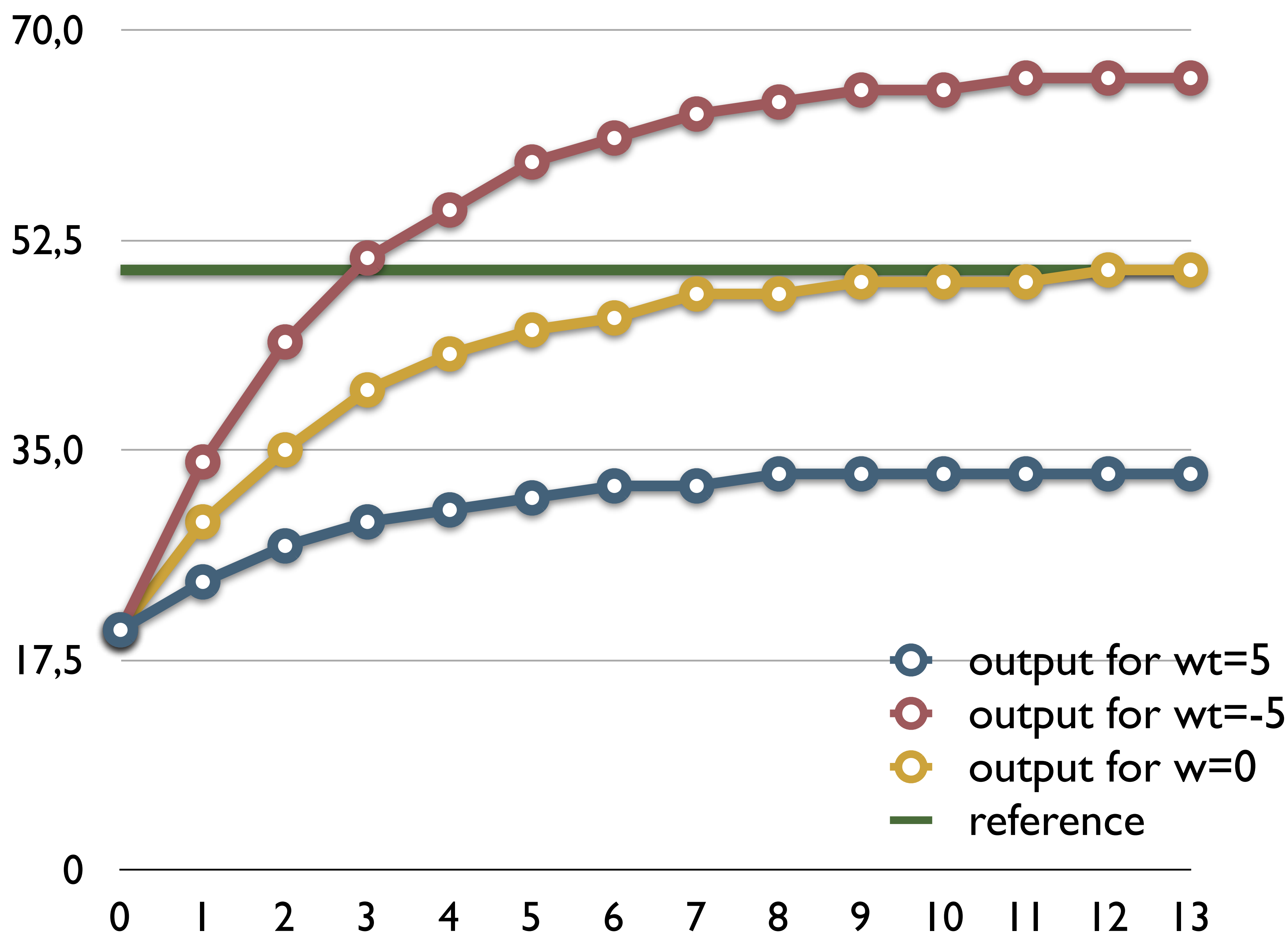
Example

- Initial speed is 20 km/h, reference speed is r=50 km/h.

- $v_{t+1}=0.7 \times v_t + 0.3\ r - w$ for w = **-5**

t	v_t	t	vt
0	20	7	62,823466
1	34	8	63,976426
2	43,8	9	64,783498
3	50,66	10	65,348449
4	55,462	11	65,743914
5	58,8234	12	66,02074
6	61,17638	13	66,214518

Example



70,0
52,5
35,0
17,5
0
0
1
2
3
4
5
6
7
8
9
10
11
12
13
output for wt=5
output for wt=-5
output for w=0
reference

Remark on the model

$$v_{t+1} = 0.7 \times v_t + 0.3\, r - w$$

$$v_1 = 0.7 \times v_0 + 0.3\, r - w$$

$$v_2 = 0.7 \times (0.7 \times v_0 + 0.3\, r - w) + 0.3\, r - w$$

$$= 0.7^2 \times v_0 + (0.7 + 1) \times 0.3 \times r - (0.7 + 1) \times w$$

$$v_3 = 0.7 \times [\, 0.7^2 \times v_0 + (0.7 + 1) \times 0.3 \times r$$

$$- (0.7 + 1) \times w \,] + 0.3\, r - w$$

$$= 0.7^3 \times v_0 + (0.7^2 + 0.7 + 1) \times 0.3 \times r$$

$$- (0.7^2 + 0.7 + 1) \times w$$

Remark on the model

In general:

$$v_t = 0.7^t \times v_0 + (1 + \sum_{1 \leq i \leq t-1} 0.7^i) \times 0.3 \times r$$

$$- (1 + \sum_{1 \leq i \leq t-1} 0.7^i) \times w$$

At any time, the output depends on the initial speed
But this influence decreases quickly.

t	0	1	2	3	4	5	6
$0{,}7^t$	0,7	0,49	0,343	0,24	0,168	0,118	0,082

Remark on the model

$$v_{t+1} = 0.7^t \times v_0 + \ldots$$

- The 0,7 value is denoted α and called the decay rate of v_0
- The larger the α the greater the influence of the initial speed in subsequent speeds

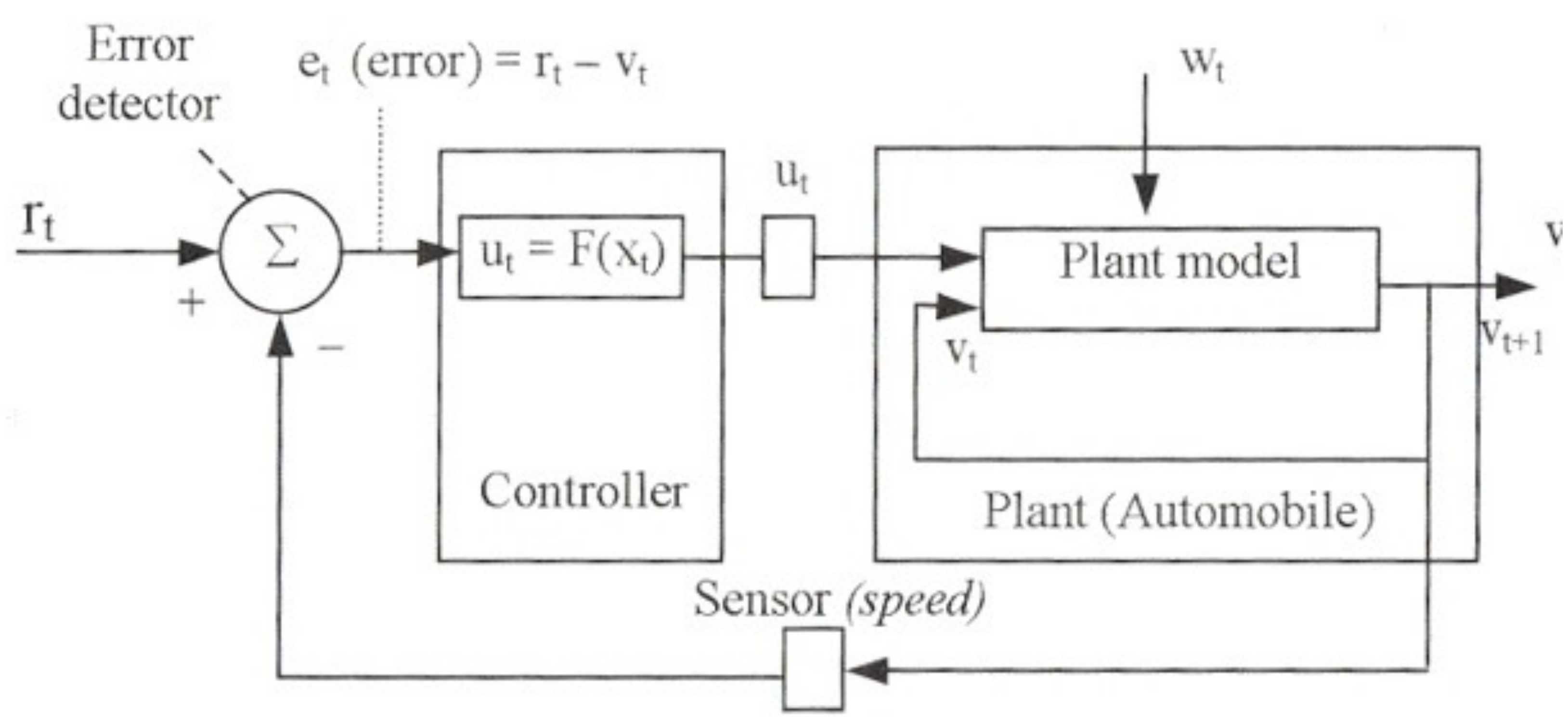
Remark on the model

$$v_{t+1} = 0.7^{t} \times v_{0} + \ldots$$

- If $\alpha > 1$, the output grows forever, and does not stabilise !
- If α is negative, the output will oscillate !
- This will be important when dealing with closed loop controllers

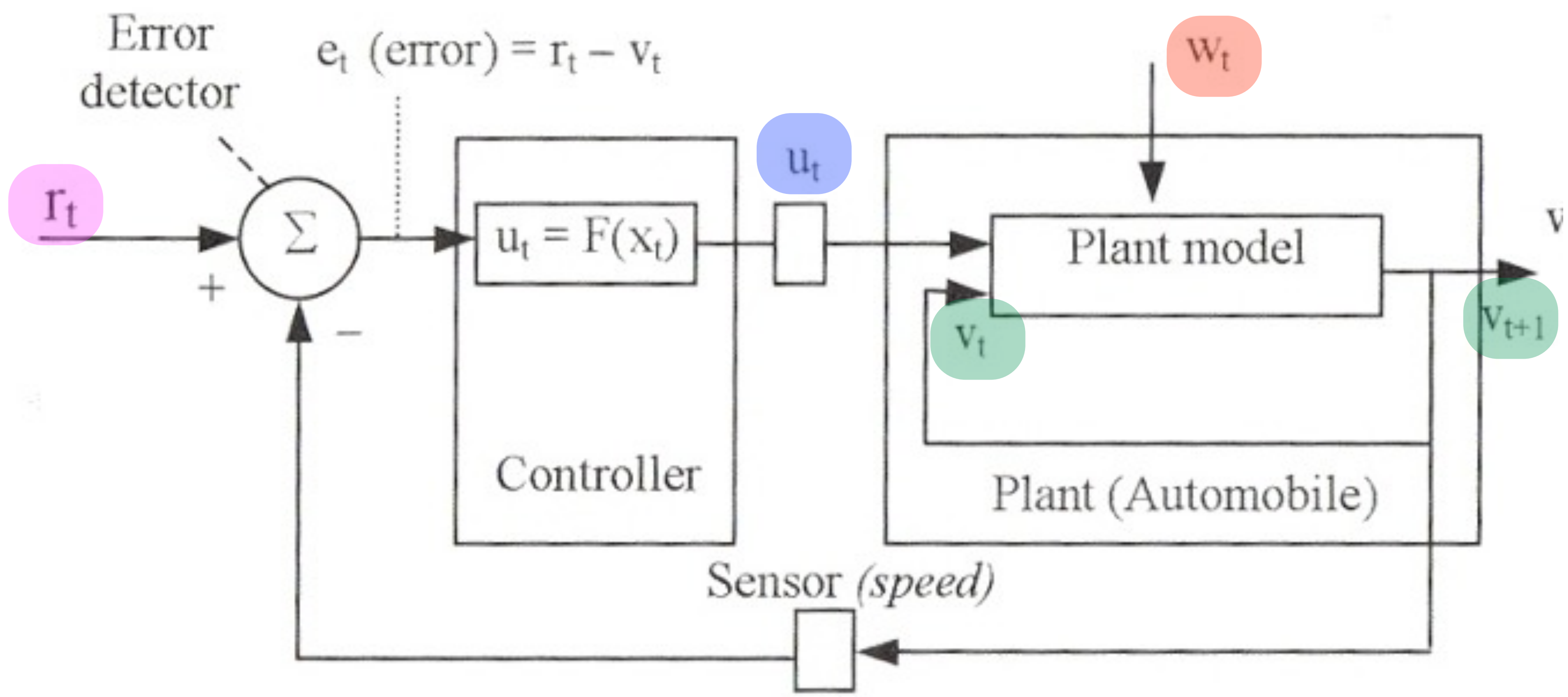
Drawbacks

- To obtain a more accurate controller, we'll introduce a feedback loop
- The controller will know the actual speed, which in turn depends on the road grade, and will able to adapt its policy to these disturbances



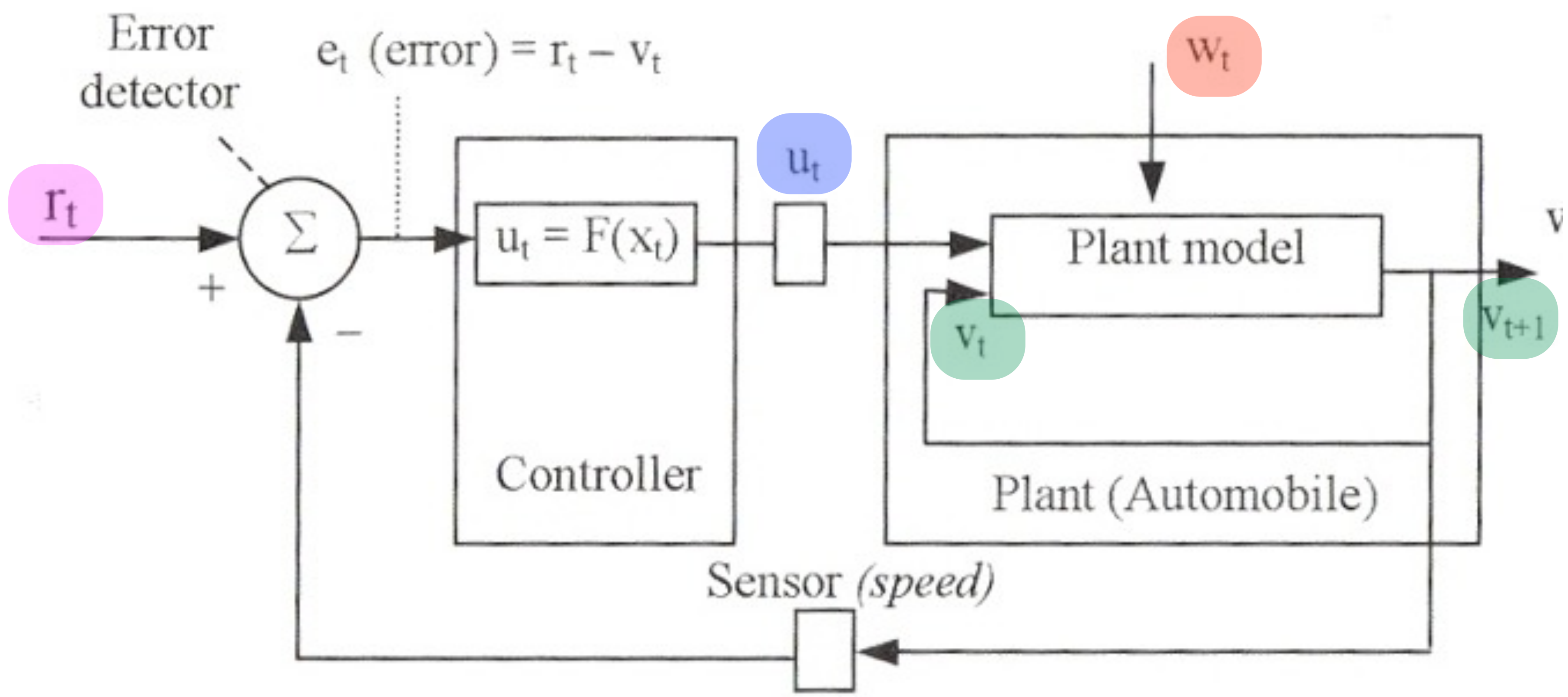
Closed loop controller

Closed loop



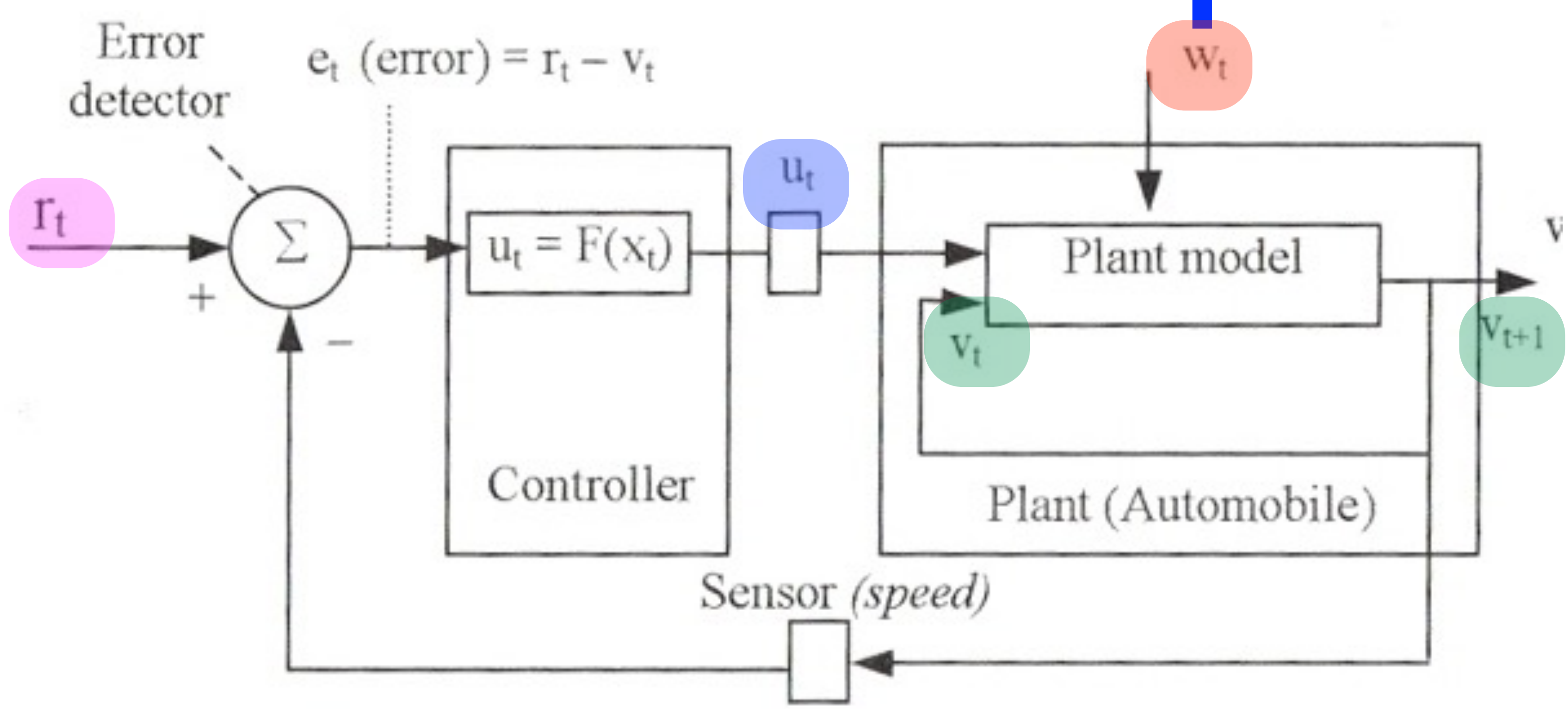
- Remark: The input to the controller is the difference between the desired speed and the current speed

Closed loop



- Let us assume once again that the controller is a linear function $F(r_t-v_t)= P \times (r_t-v_t)$
- We keep $v_{t+1}=0{,}7v_t+0{,}5u_t - w_t$ as a model for the car

Closed loop



$$\left.\begin{aligned} F(r_t - v_t) &= P \times (r_t - v_t) \\ u_t &= F(r_t - v_t) \end{aligned}\right\} \text{ Controller}$$

$$v_{t+1} = 0{,}7 v_t + 0{,}5 u_t - w_t \quad \} \quad \text{Car}$$

Hence:

$$v_{t+1} = 0{,}7 \times v_t + 0{,}5 \times P \times (r_t - v_t) - w_t$$

Closed loop

$$v_{t+1} = 0,7 \times v_t + 0,5 \times P \times (r_t - v_t) - w_t$$

$$= (0,7 - 0,5 \times P)\, v_t + 0,5 \times P \times r_t - w_t$$

- Now, the P parameter appears in the expression of v_0:
  $\alpha = (0,7 - 0,5 \times P)$
- We have to take that in to account to keep $0 \leq \alpha < 1$
- This means we want $-0,6 < P \leq 1,4$

Examples

$$v_{t+1} = (0{,}7 - 0{,}5 \times P)\, v_t + 0{,}5 \times P \times r_t - w_t$$

$$w_t = 0 \qquad P = -0{,}5 \qquad \alpha = (0{,}7 - 0{,}5 \times P) = 0{,}95$$

t	v_t	t	vt
0	20	7	-63,07519
1	6,5	8	-72,42143
2	-6,325	9	-81,30036
3	-18,50875	10	-89,73534
4	-30,08331	11	-97,74857
5	-41,07915	12	-105,3611
6	-51,52519	13	-112,5931

Examples

$$v_{t+1} = (0{,}7 - 0{,}5 \times P)\, v_t + 0{,}5 \times P \times r_t - w_t$$

$w_t = 0$ $\quad P = -0{,}5$ $\quad \alpha = (0{,}7 - 0{,}5 \times P) = 0{,}95$

				19
				43
				36
				34
				57
				11
				31

By choosing

$F(r_t - v_t) = -0{,}5 \times (r_t - v_t)$

we have a controller that decelerates when $r_t > v_t$...

Speeds <0 don't make any sense !

Examples

$$v_{t+1} = (0,7 - 0,5 \times P)\, v_t + 0,5 \times P \times r_t - w_t$$

$$w_t = 0 \qquad P = 1 \qquad \alpha = (0,7 - 0,5 \times 1) = 0,2$$

t	v_t	t	vt
0	20	7	31,249856
1	29	8	31,249971
2	30,8	9	31,249994
3	31,16	10	31,249999
4	31,232	11	31,25
5	31,2464	12	31,25
6	31,24928	13	31,25

Examples

$$v_{t+1} = (0{,}7 - 0{,}5 \times P)\ v_t + 0{,}5 \times P \times r_t - w_t$$

$w_t = 0$ $\quad P = 1{,}4$ $\quad \alpha = (0{,}7 - 0{,}5 \times 1{,}4) = 0$

t	v_t	t	vt
0	20	7	35
1	35	8	35
2	35	9	35
3	35	10	35
4	35	11	35
5	35	12	35
6	35	13	35

Examples

$$v_{t+1} = (0,7 - 0,5 \times P)\, v_t + 0,5 \times P \times r_t - w_t$$

$w_t = 0$ $P = 1,4$ $\alpha = (0,7 - 0,5 \times 1,4) = 0$

t	v_t	t	v_t
0			
2			
3			
4			
5			
6	35	13	35

That controller stabilises immediately at $0,7 \times r_t$

Examples

$$v_{t+1} = (0{,}7 - 0{,}5 \times P)\, v_t + 0{,}5 \times P \times r_t - w_t$$

$w_t = 0$ $\quad P = 3 \quad \alpha = (0{,}7 - 0{,}5 \times 3) = -0{,}8$

t	v_t	t	vt
0	20	7	46,210496
1	59	8	38,031603
2	27,8	9	44,574717
3	52,76	10	39,340226
4	32,792	11	43,527819
5	48,7664	12	40,177745
6	35,98688	13	42,857804

Examples

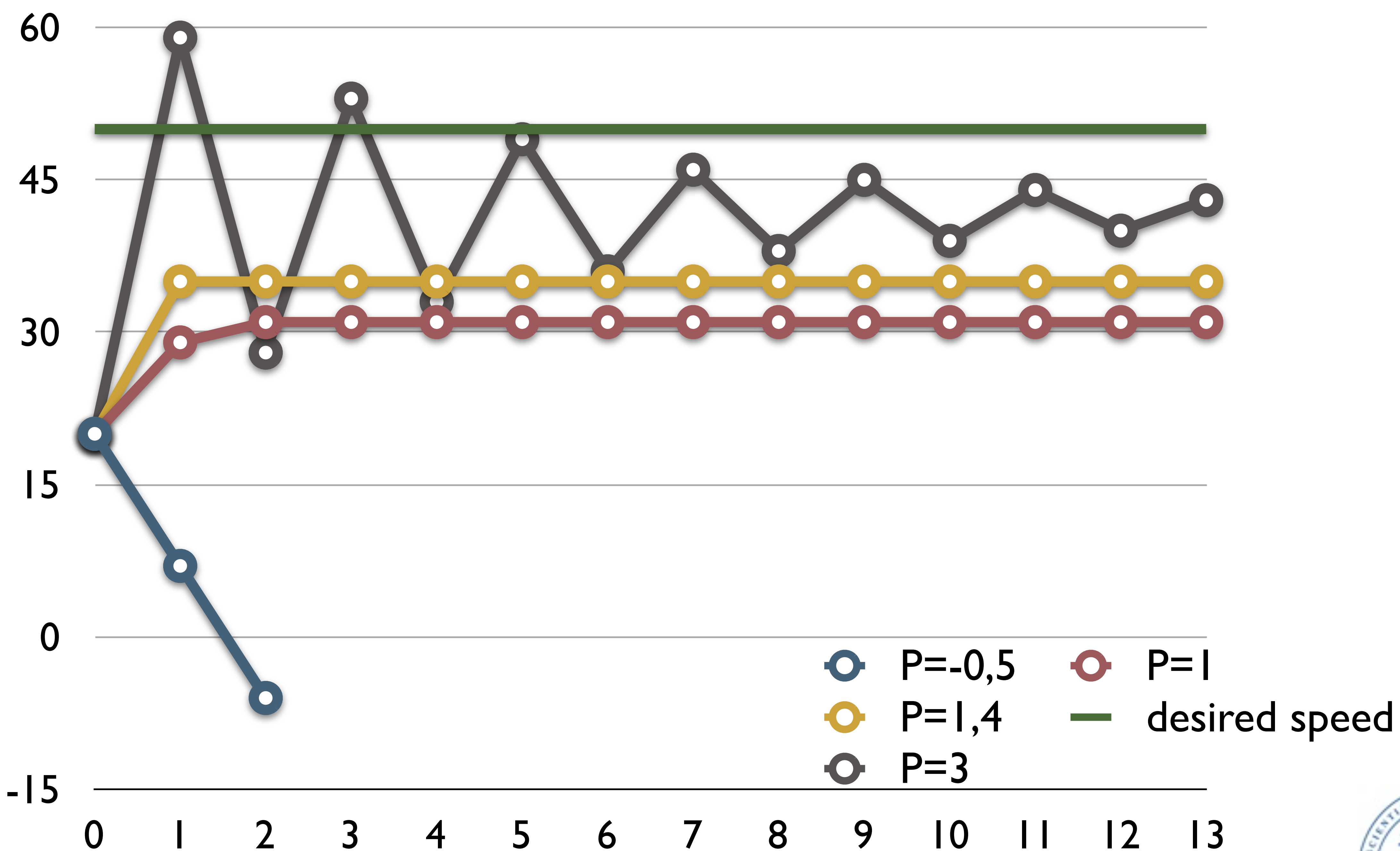
$$v_{t+1} = (0,7 - 0,5 \times P)\, v_t + 0,5 \times P \times r_t - w_t$$

$w_t = 0 \qquad P = 3 \qquad \alpha = (0,7 - 0,5 \times 3) = -0,8$

t	v_t	t	vt
0	20	7	46,210496
[illegible]	[illegible]	[illegible]	…603
[illegible]	[illegible]	[illegible]	…717
[illegible]	[illegible]	[illegible]	…226
[illegible]	[illegible]	[illegible]	…819
5	48,7664	12	40,177745
6	35,98688	13	42,857804

With a negative α
oscillations appear

Examples


60
45
30
15
0
-15
0
1
2
3
4
5
6
7
8
9
10
11
12
13
P=-0,5
P=1
P=1,4
desired speed
P=3

To sum up

- Up to now, we have only considered the effect of P on α
- If P≤-0,6, then α≥1 and the system does not converge
- If -0,6<P≤0, the controller is «reversed»
- If 0<P≤1,4: quick stability but large error, (15 km/h for P=1,4; 25 km/h for P=0,5)
- If 1,4<P<3,4 the α<0 and the system oscillates but the error is low
- If P≥3,4, then α≥1 and the system does not converge

Steady states

- Let us now fix a control objective: that the v speed be equal to the requested speed r in steady states
  - Steady sates are those where $v_{t+1} = v_t = v_{ss} = r$
- By replacing v_t, v_{t+1} by v_{ss} in $v_{t+1} = (0,7 - 0,5 \times P)\, v_t + 0,5 \times P \times r - w_0$ we get:

$$v_{ss} = \frac{0,5 \times P}{0,3 + 0,5 \times P} \times r - \frac{1}{0,3 + 0,5 \times P} \times w_0$$

Steady states

$$v_{ss} = \frac{0,5 \times P}{0,3 + 0,5 \times P} \times r - \frac{1}{0,3 + 0,5 \times P} \times w_0$$

- First observation: We now compute P by taking into account the perturbation w_0 (which is supposed constant)
- The best value for P thus depends upon w_0: we can't find a single value that allows perfect control in every situation !

Steady states

$$v_{ss} = \frac{0,5 \times P}{0,3 + 0,5 \times P} \times r - \frac{1}{0,3 + 0,5 \times P} \times w_0$$

- We'll try to minimise the influence of perturbation by having the 2nd term as small as possible
- We want to have $1/(0,3 + 0,5 \times P) < 1$
  - this implies $P > 1.4$
  - but then $\alpha \leq 0$, and we have oscillations !

Steady states

$$v_{ss} = \frac{0,5 \times P}{0,3 + 0,5 \times P} \times r - \frac{1}{0,3 + 0,5 \times P} \times w_0$$

- Moreover, we'd like v_{ss}=r
- To achieve this, the coefficient of the firs term should be as close to 1 as possible
- Thus P should be as large as possible

Steady states

$$v_{ss} = \frac{0,5 \times P}{0,3 + 0,5 \times P} \times r - \frac{1}{0,3 + 0,5 \times P} \times w_0$$

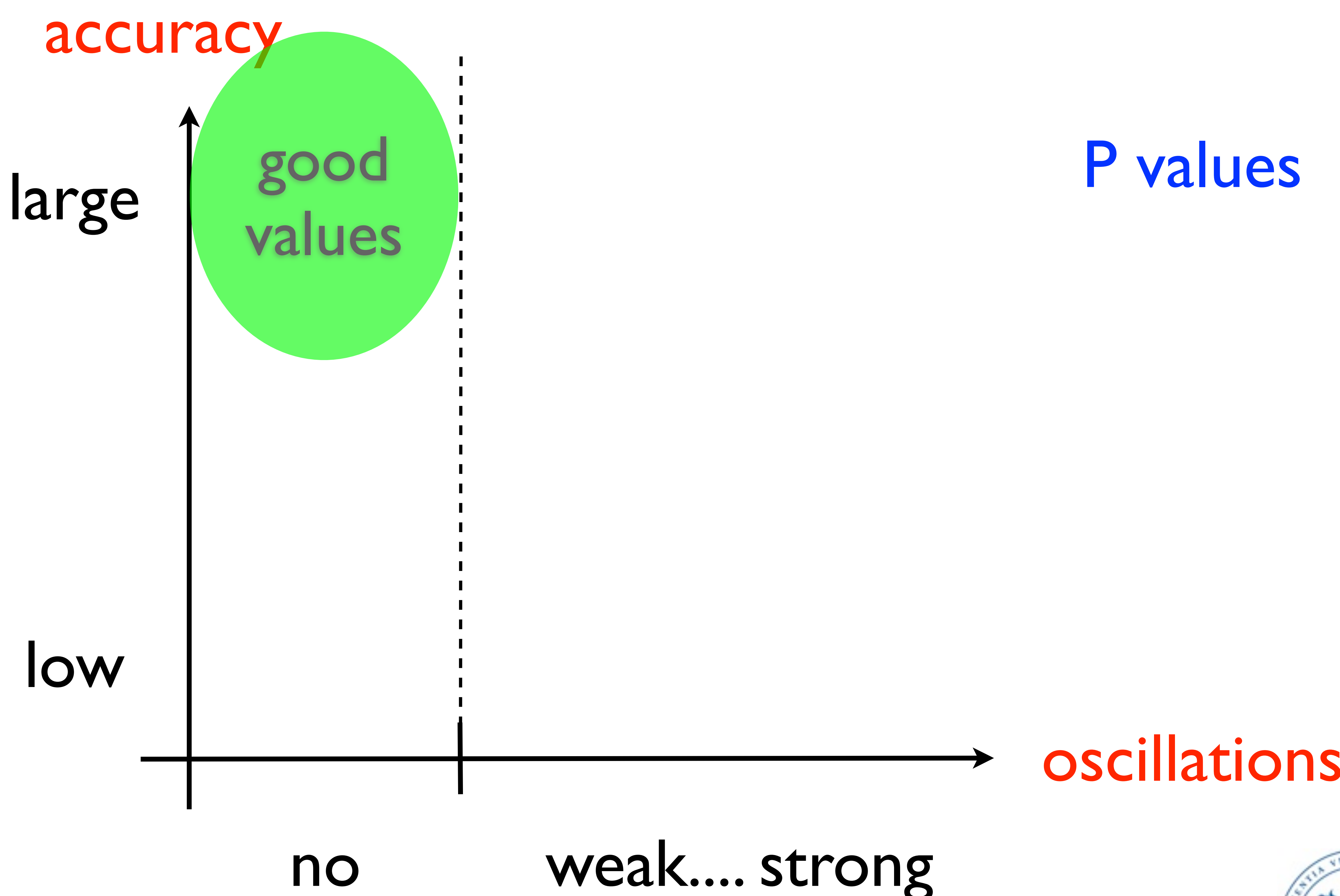
P	coeff. 1	coeff. 2
0,5	0,4545454545	1,8181818182
1	0,625	1,25
2	0,7692307692	0,7692307692
10	0,9433962264	0,1886792453
100	0,9940357853	0,0198807157
10000	0,9999400036	0,000199988

Larger values of P are better !

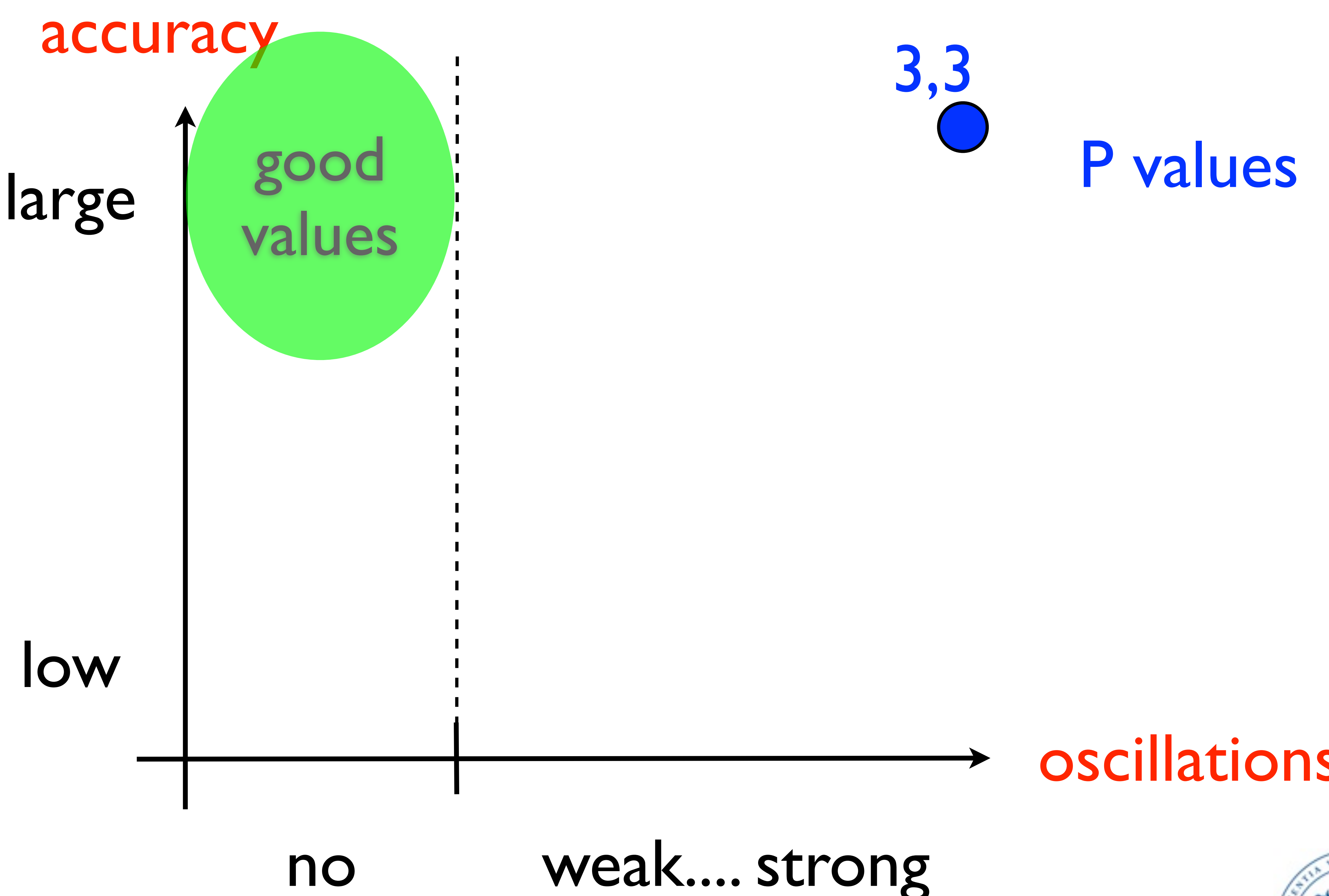
Steady states

- We could try to use a P as big as possible...
- ... however, with P>3,4 we get oscillations
- In this case, two control objectives are contradictory !
  - accuracy vs
  - smooth convergence (i.e. passenger comfort ;-)

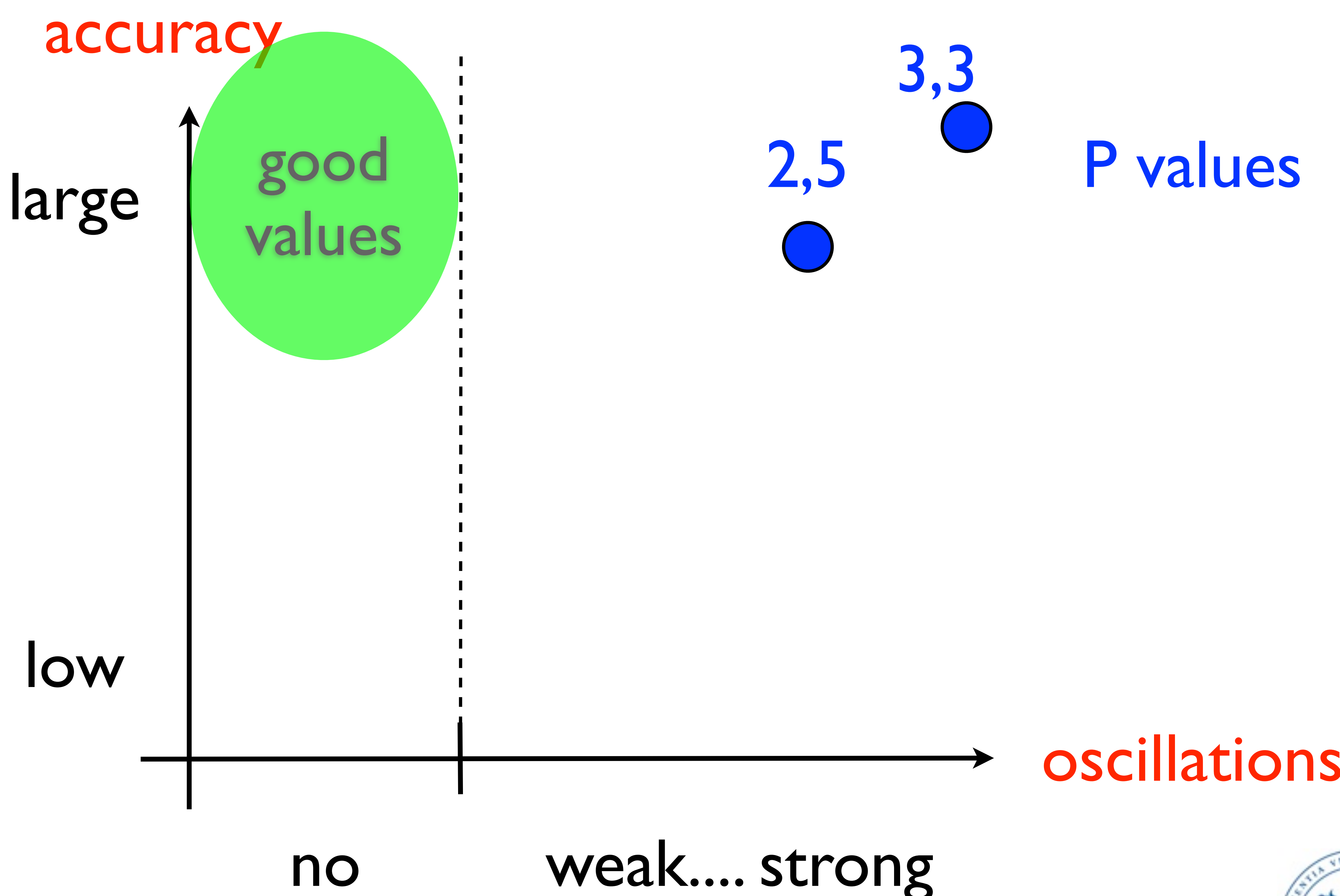
Control objectives



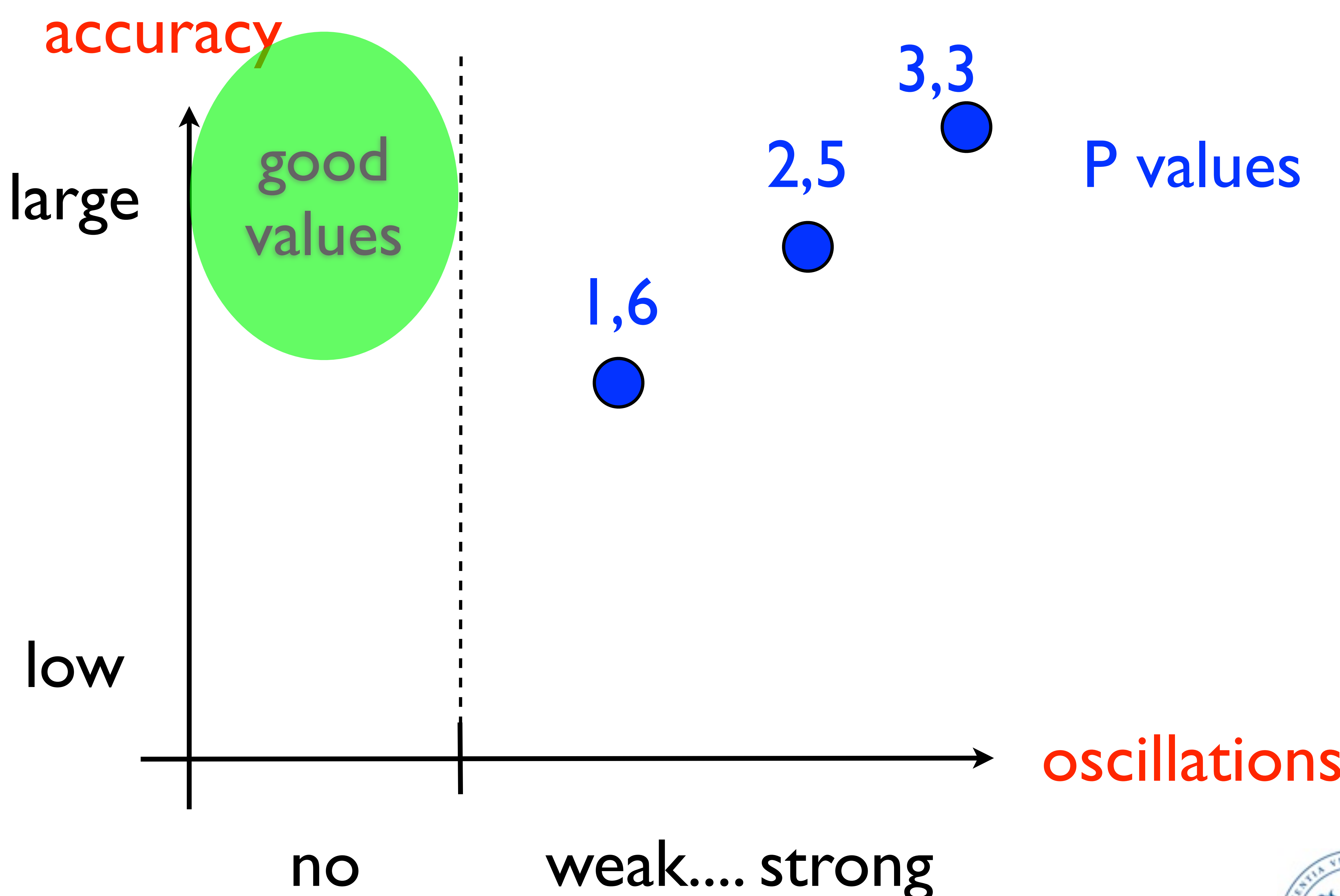
Control objectives



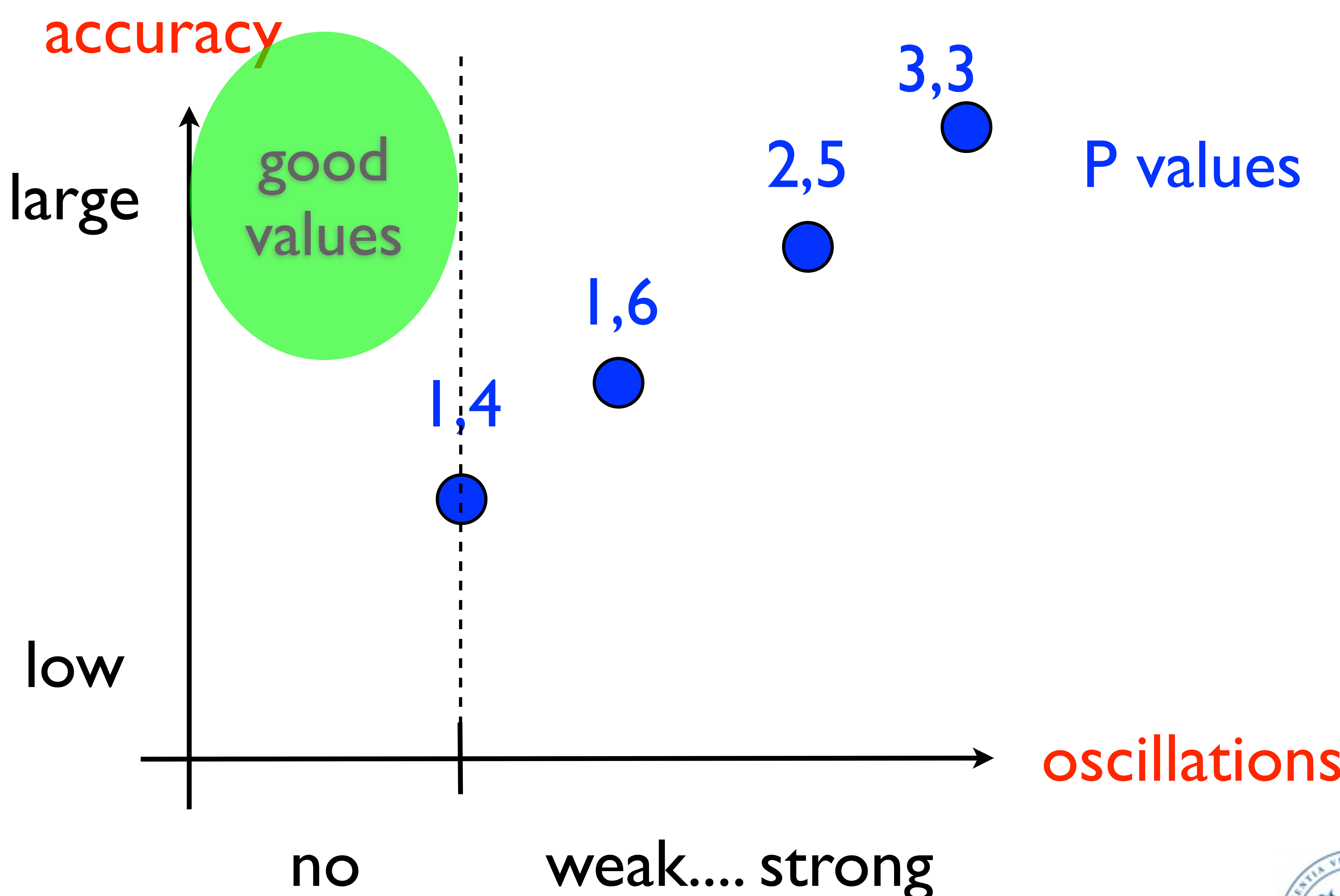
Control objectives



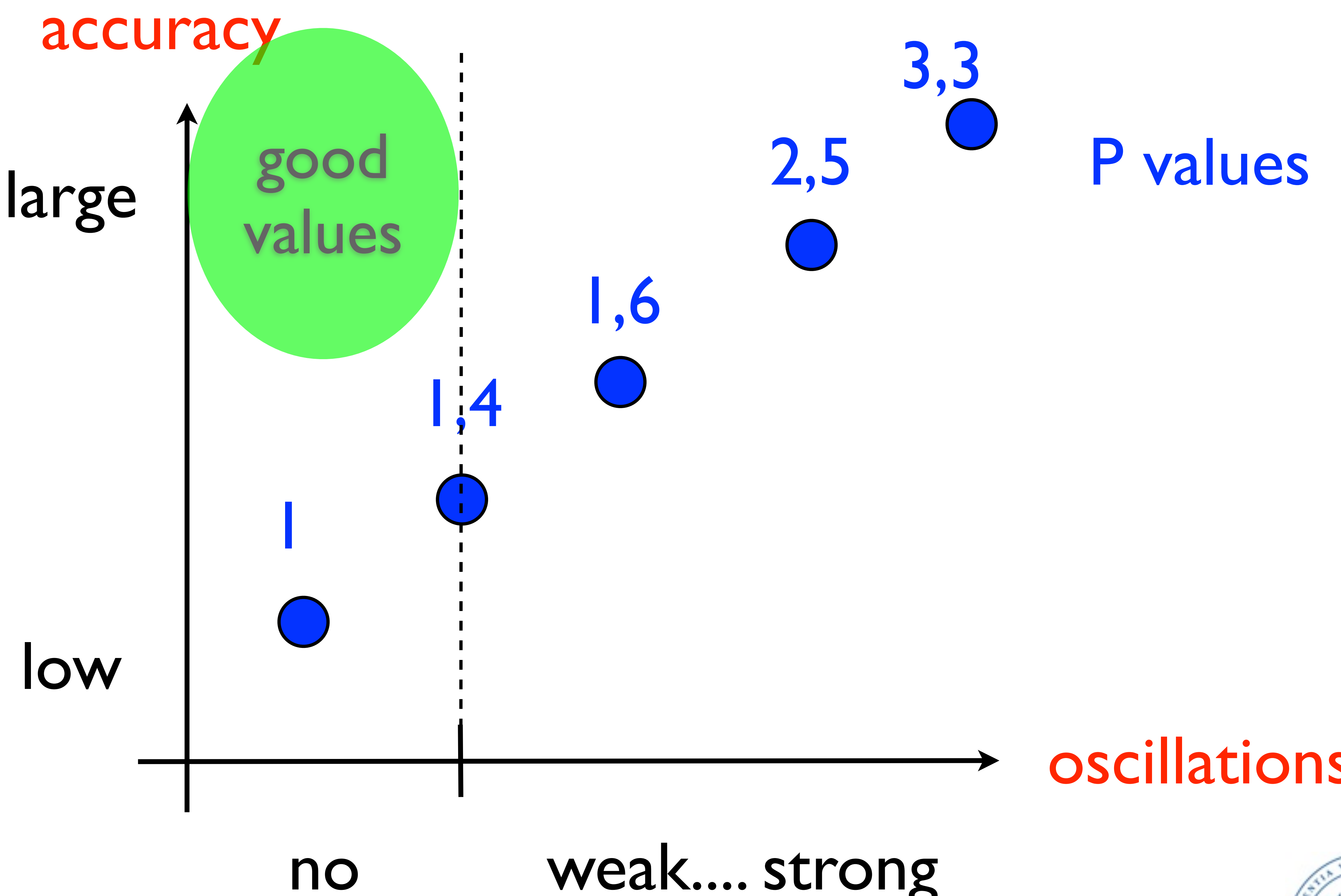
Control objectives



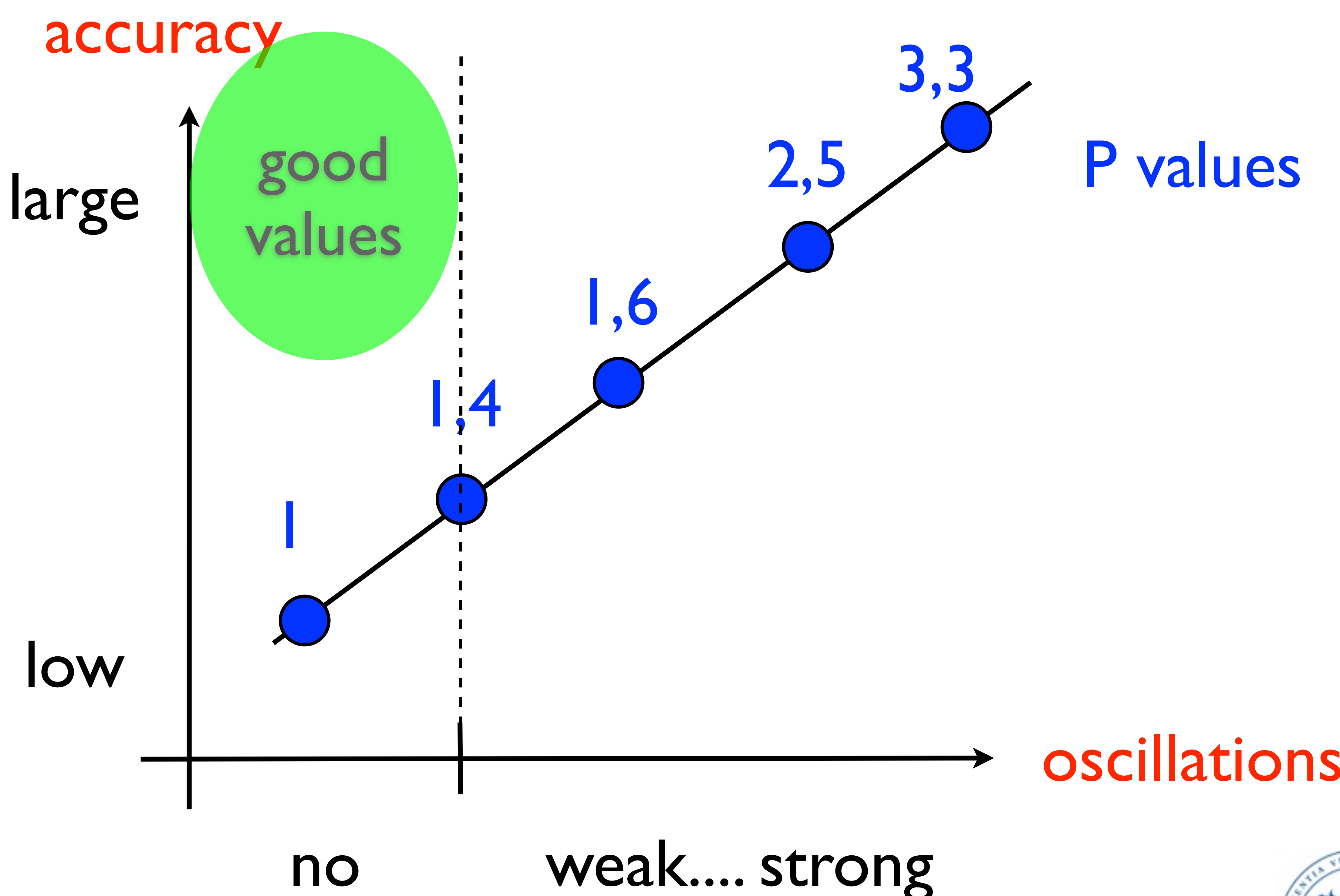
Control objectives



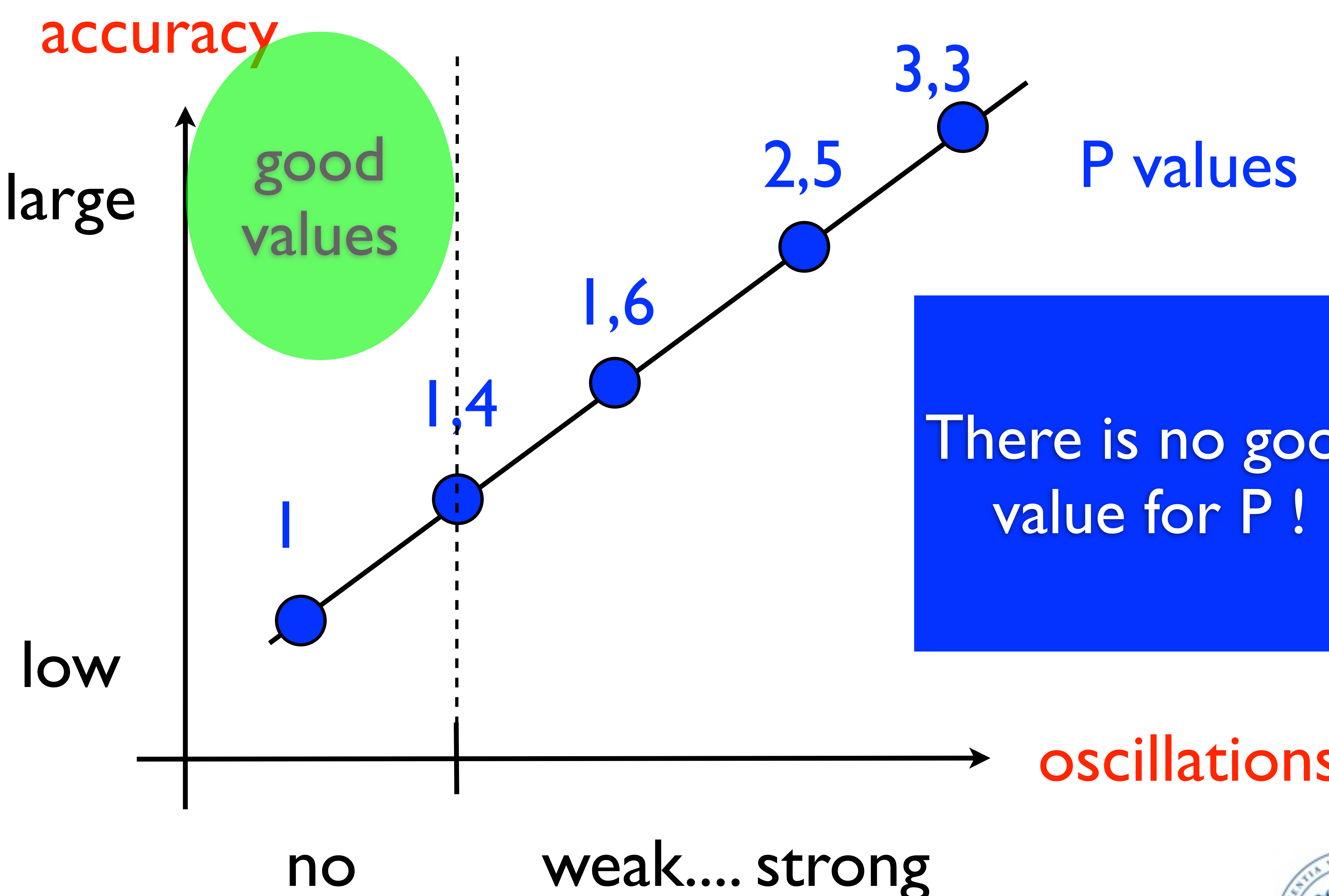
Control objectives



Control objectives



Control objectives



There is no good value for P !

Contradictory objectives

- We have to take decisions:
  - With a value such as P=3,3 we advantage the accuracy, but the oscillations will threaten the passengers' comfort
  - With a value such as P=1 there are no oscillations, but the speed stabilises way under the desired speed
- Nothing better can be achieved with such a simplistic controller

Disturbances

- Let us remind we had introduced closed loop controllers to cope correctly with disturbances
- It is the case ?

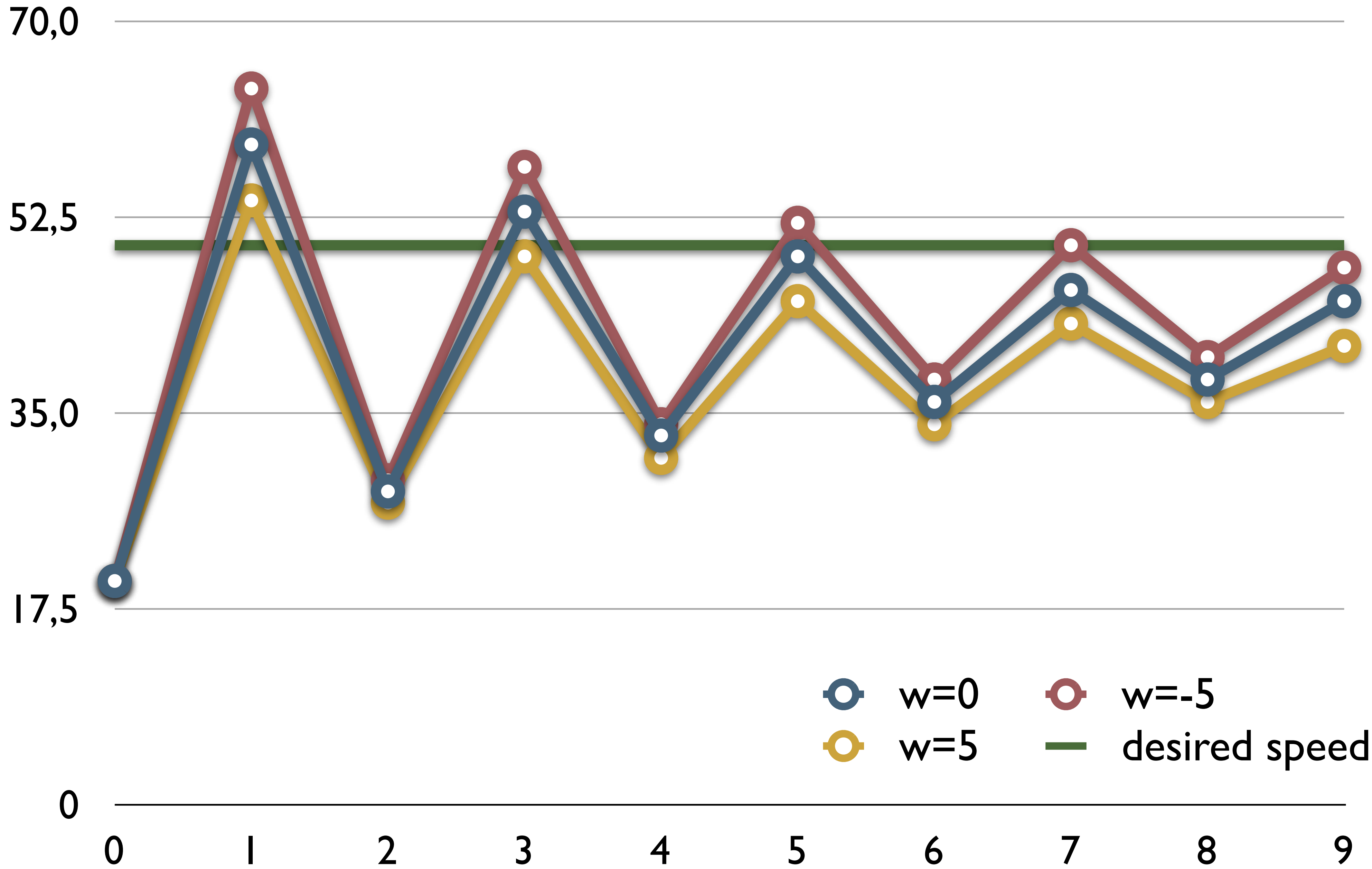
Disturbances

$P = 3 \quad r = 50 \quad v_0 = 20$

t	v for w=0	v for w=-5	v for w=5
0	20	20	20
1	59	64	54
2	28	29	27
3	53	57	49
4	33	34	31
5	49	52	45
6	36	38	34
7	46	50	43
8	38	40	36
9	45	48	41

Disturbances

$P = 3 \quad r = 50 \quad v_0 = 20$



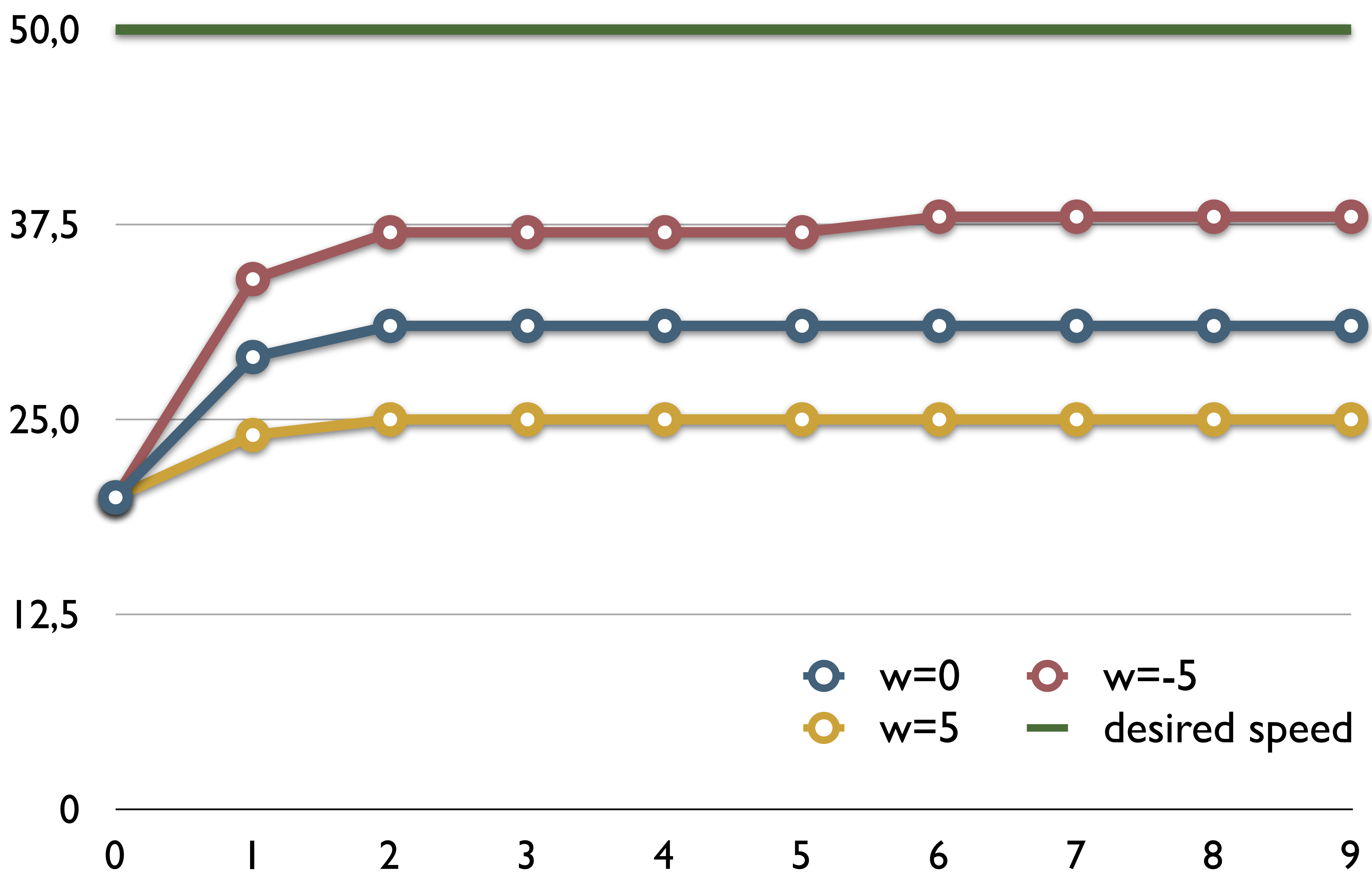
Disturbances

$P = 1 \qquad r = 50 \qquad v_0 = 20$

t	v for w=0	v for w=-5	v for w=5
0	20	20	20
1	29	34	24
2	31	37	25
3	31	37	25
4	31	37	25
5	31	37	25
6	31	38	25
7	31	38	25
8	31	38	25
9	31	38	25

Disturbances

$P = 1 \quad r = 50 \quad v_0 = 20$



Disturbances

- One can see that the plots for the different w values are indeed much closer
- In the P=3 case, they converge to very close values.

PID Controllers

PID

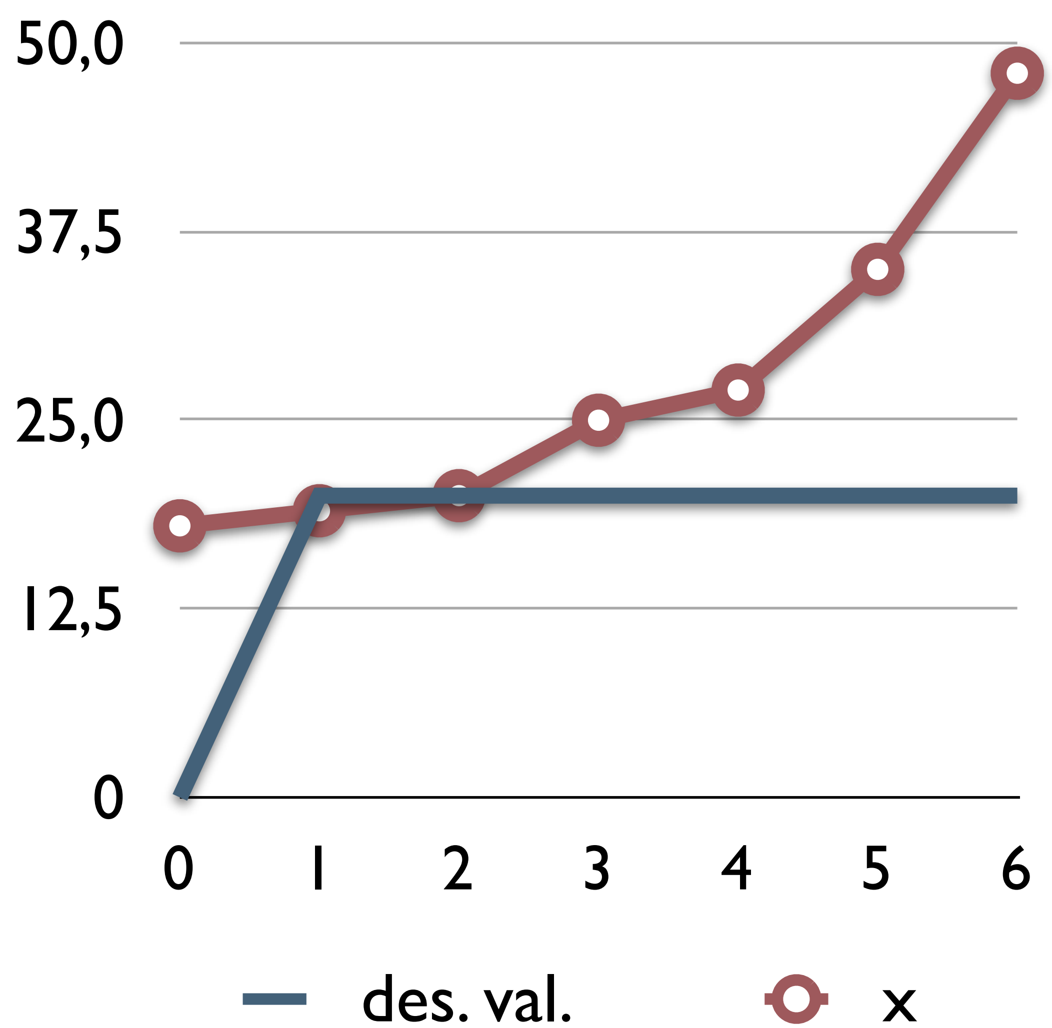
- Let us now study a more general form of closed loop controller: PID controller
- We will discuss with more details the control objectives.

Control objectives

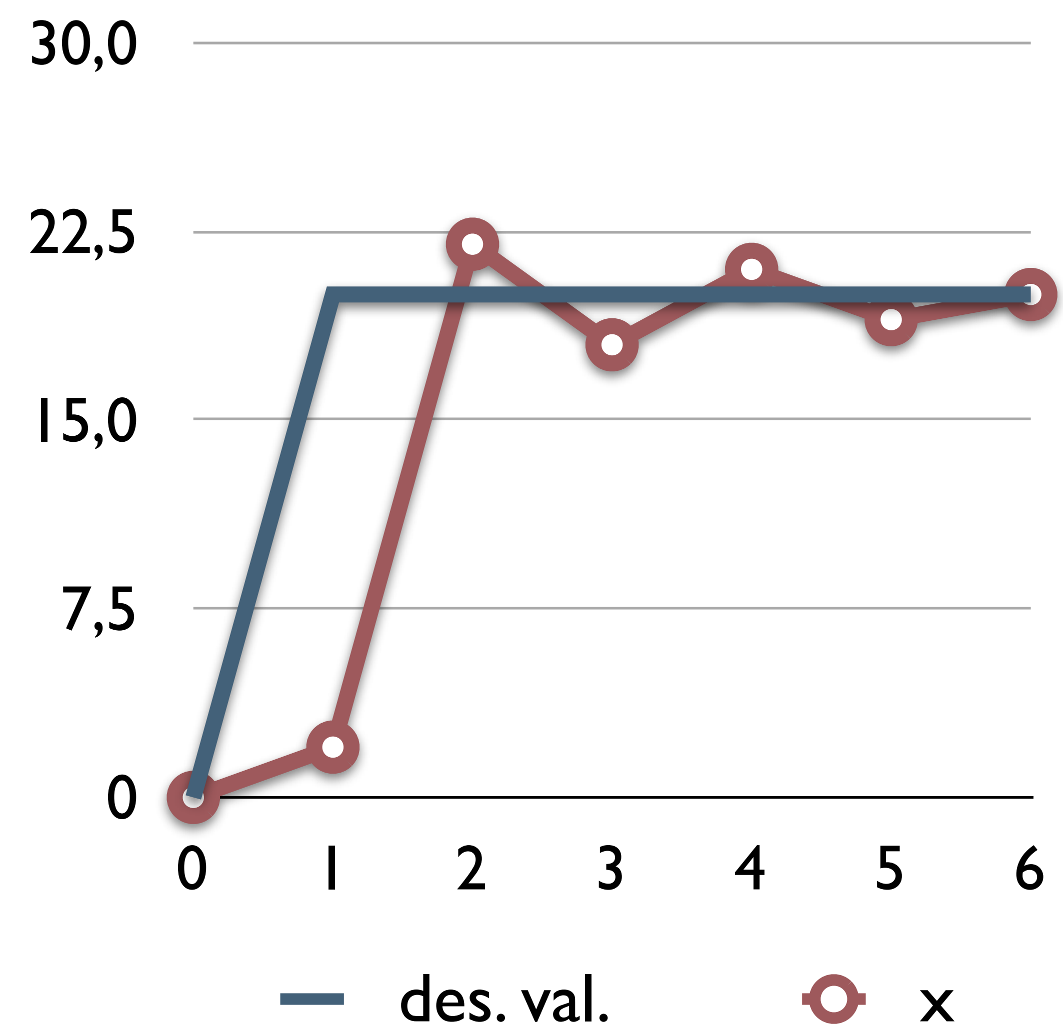
- Stability: All the system's variables (including error) must be bounded.
  - Ideally error variables should converge to 0
  - Stability is the main control objective
    - Without stability there is no actual control...

Control objective

Not stable



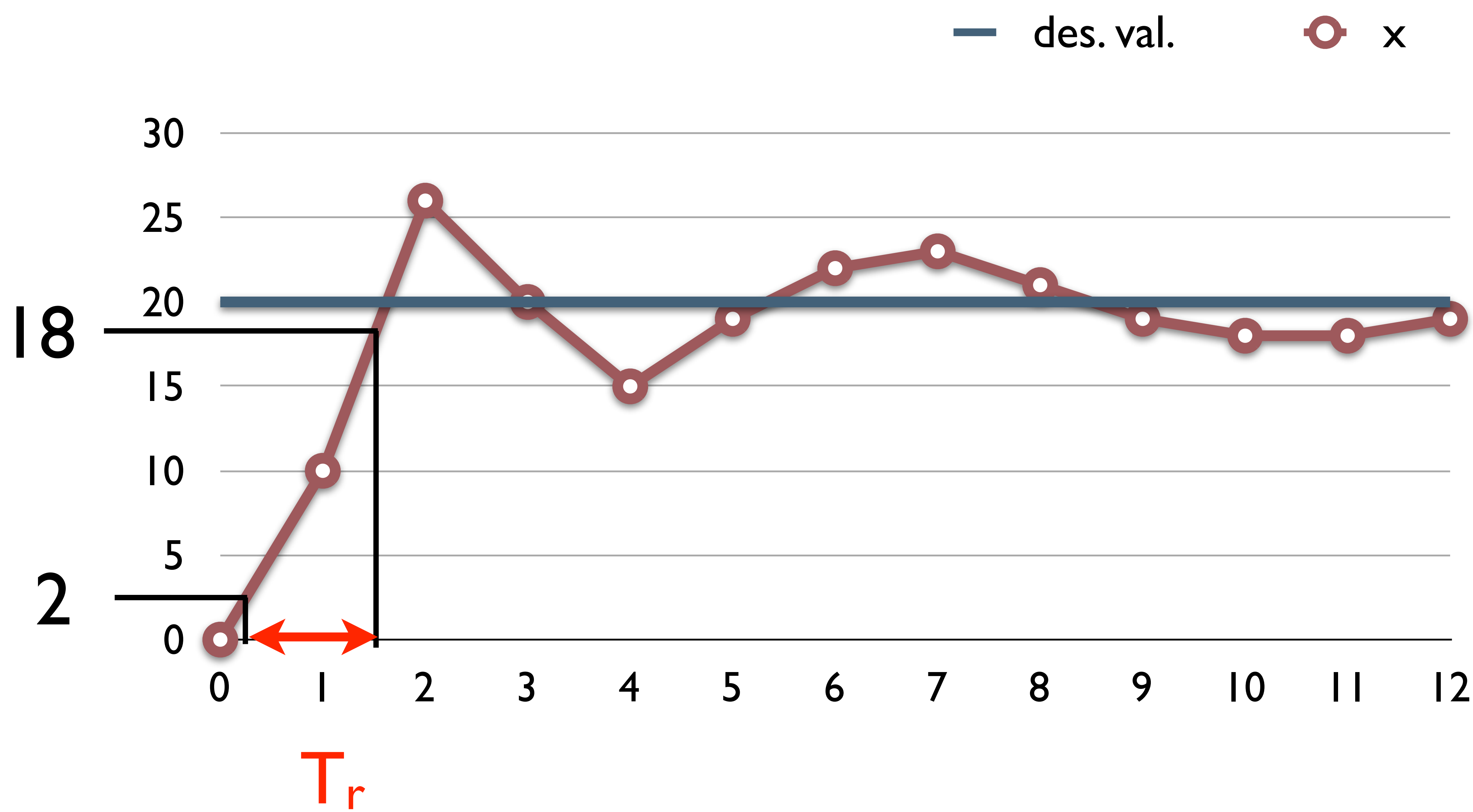
Stable



Control objectives

- Performance: In a stable system, performances measures, thanks to several parameters, the distance between the output and the desired value
  - Raising time T_r. Let d be the distance between the initial value i and the desired value r, then Tr is the time need to see the output change from i + 10% × d to i + 90% × d

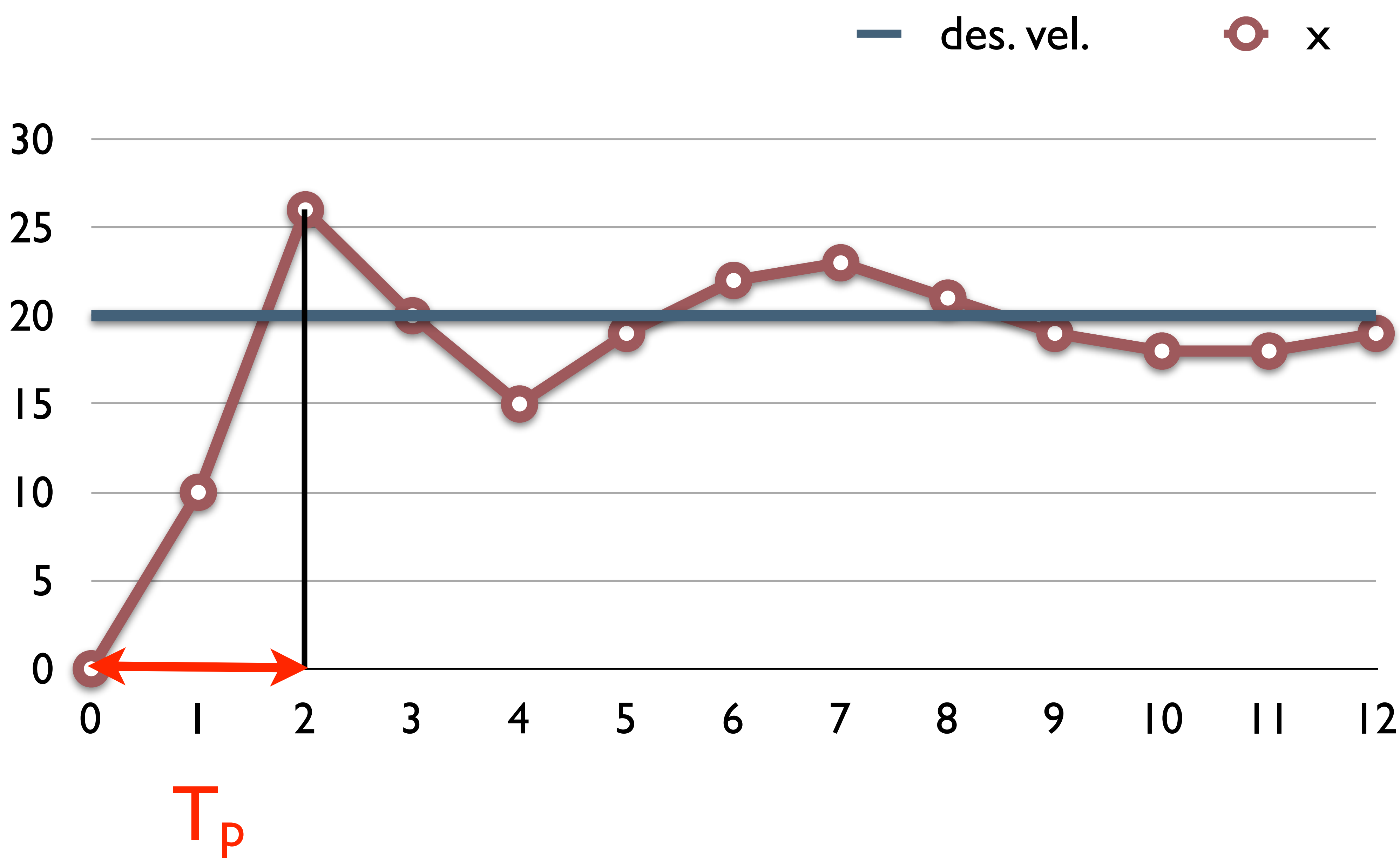
Control objectives



Control objectives

- Performance:
  - Peak time T_P: Time to reach the first peak in the output

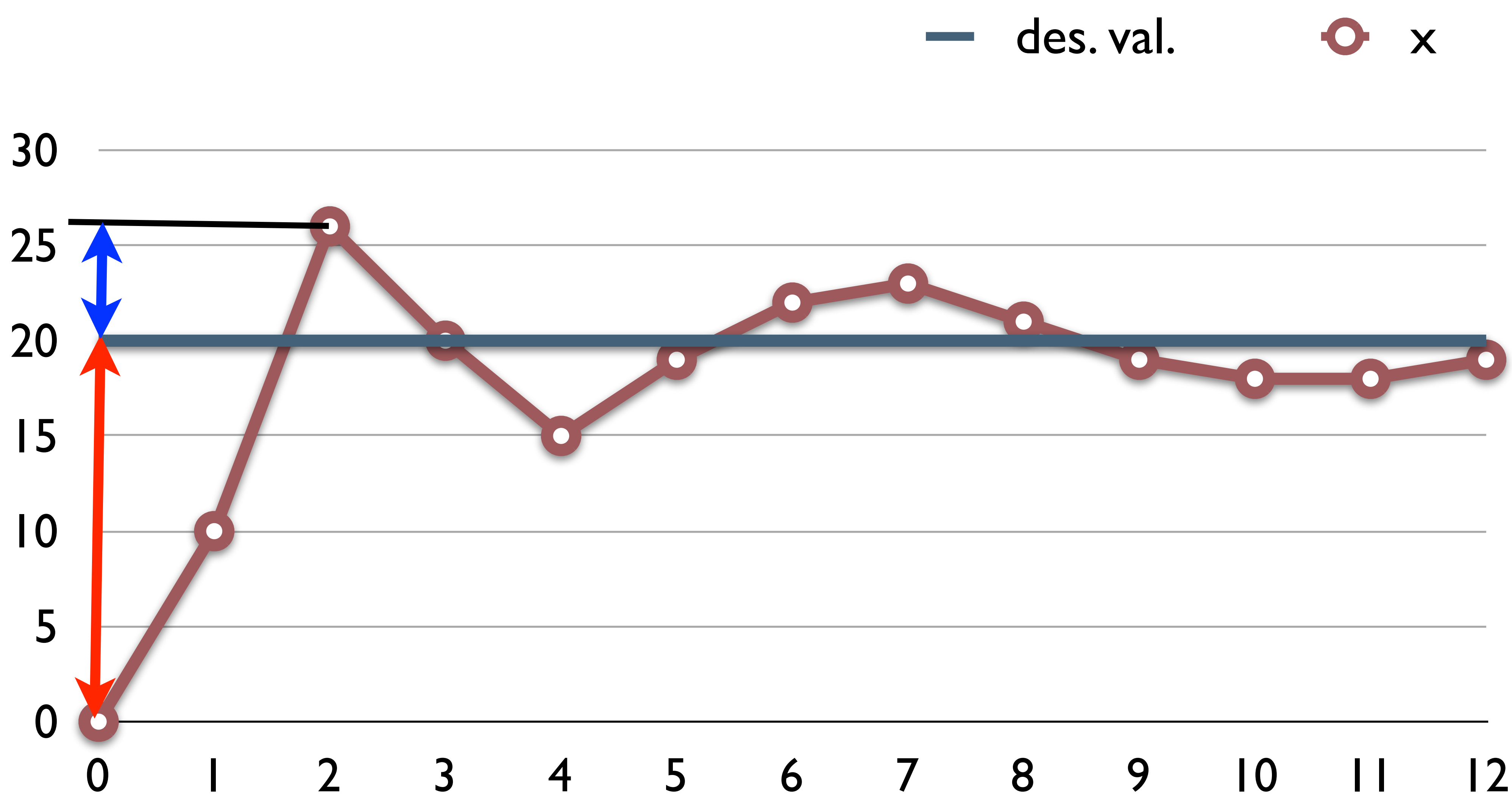
Control objectives



Control objectives

- Performance:
  - Overshoot M_p: Percentage amount by which the peak exceeds the desired value

Control objectives

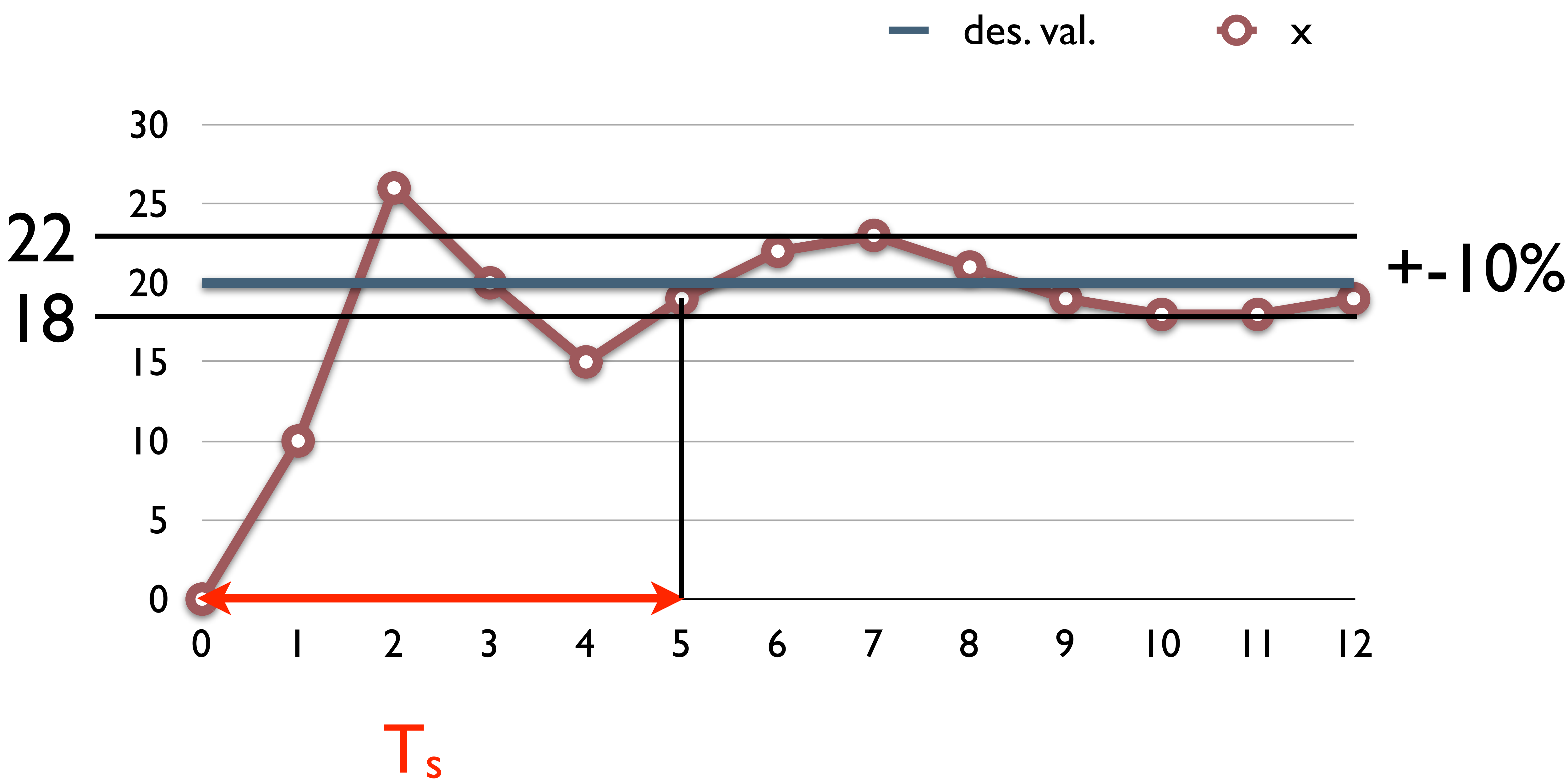


Mp= ———— (in %)

Control objectives

- Performance:
    - Settling time T_s: Time required for the system to settle down within an interval of x% centered on the desired value (x=1, 10,...)

Control objectives



Control objectives

- Disturbance rejection: the system should not be sensitive to disturbances
- Robustness: Remind that we are reasoning on a model of the environment. Slight changes in the behaviour of the environment should not affect significantly the controller.

Types of controllers

- From now on we only consider closed loop controllers
  - The controller we have seen so far is a proportional controller (P)
  - Its control law is $P \times e_t$, where e_t is the error r_t-v_t at time t
  - Its output is proportional to the error

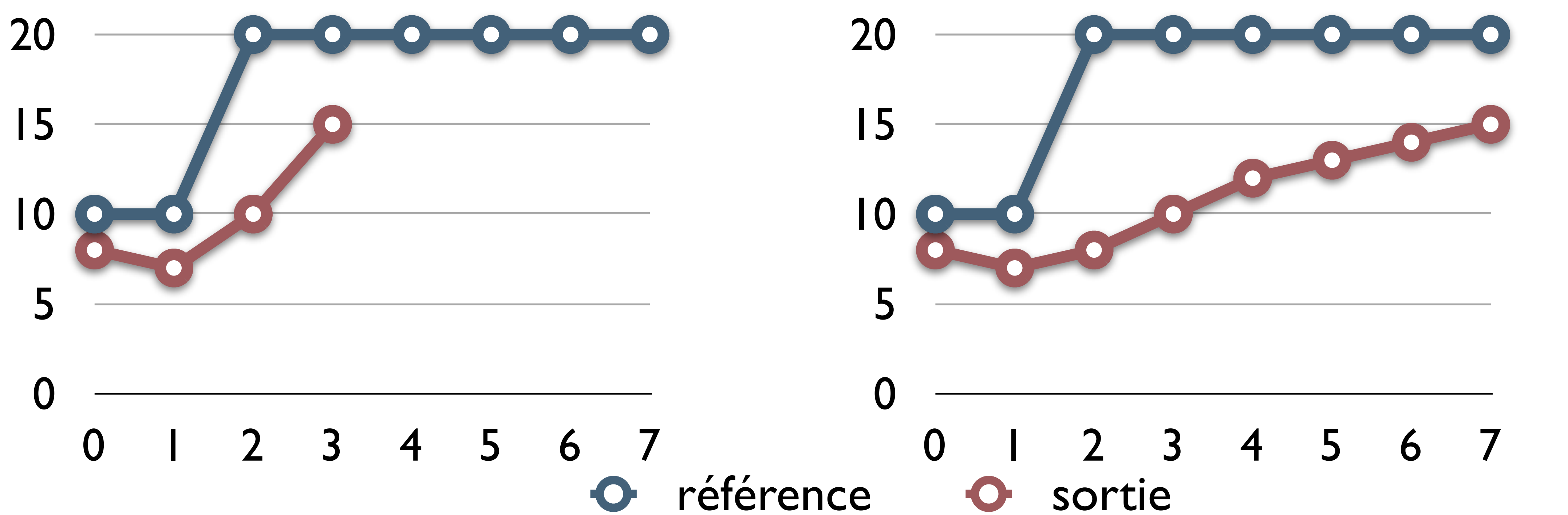
P controllers

$P \times e_t$

- A P controller, has access to the previous error only
- In many cases, control would be much easier to achieve if one could "predict future errors"

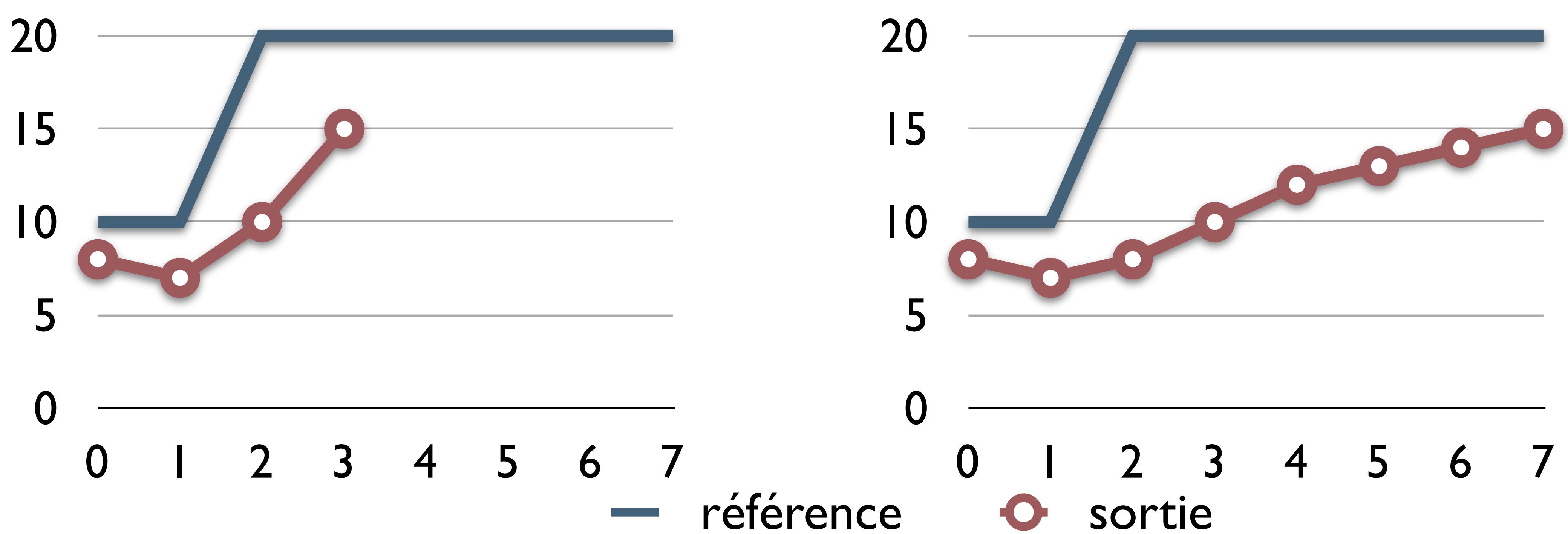
P controller

- At t=3, left and at t=7, right the errors are equal
- A P controller thus has the same reaction
- However, the ideal reaction is different



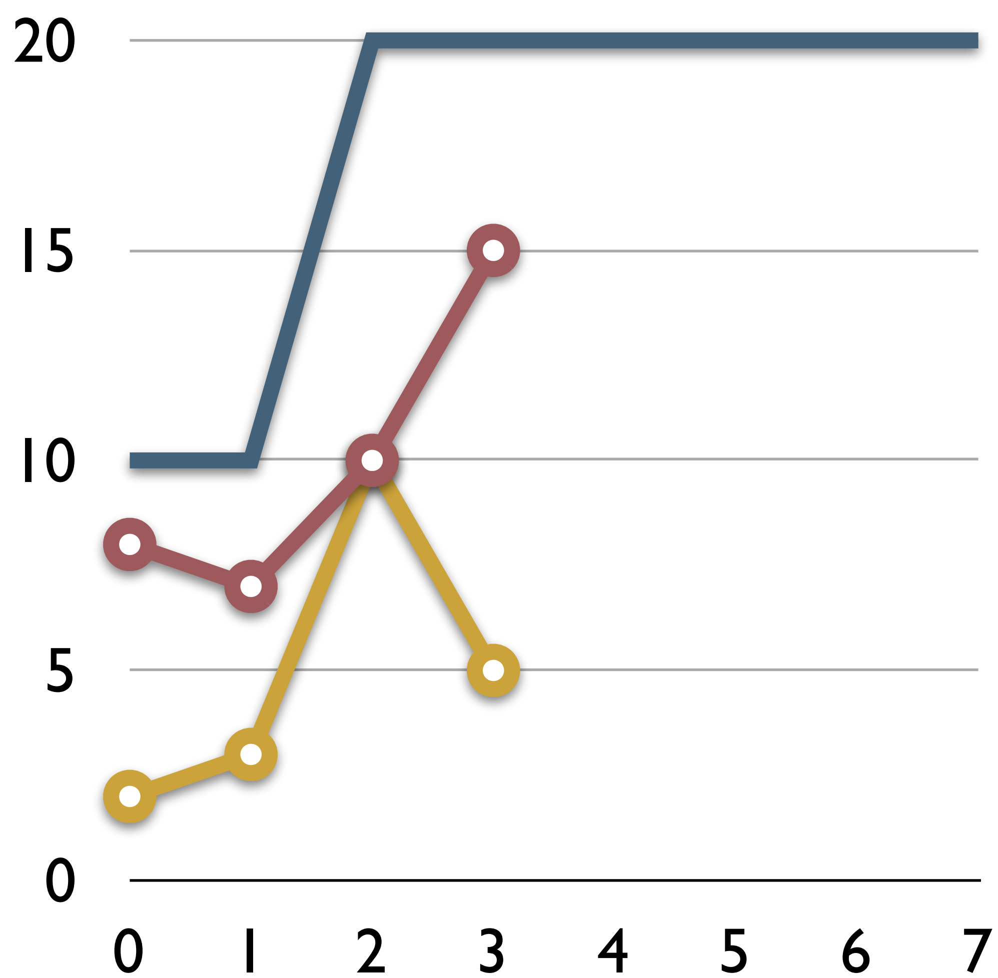
P controller

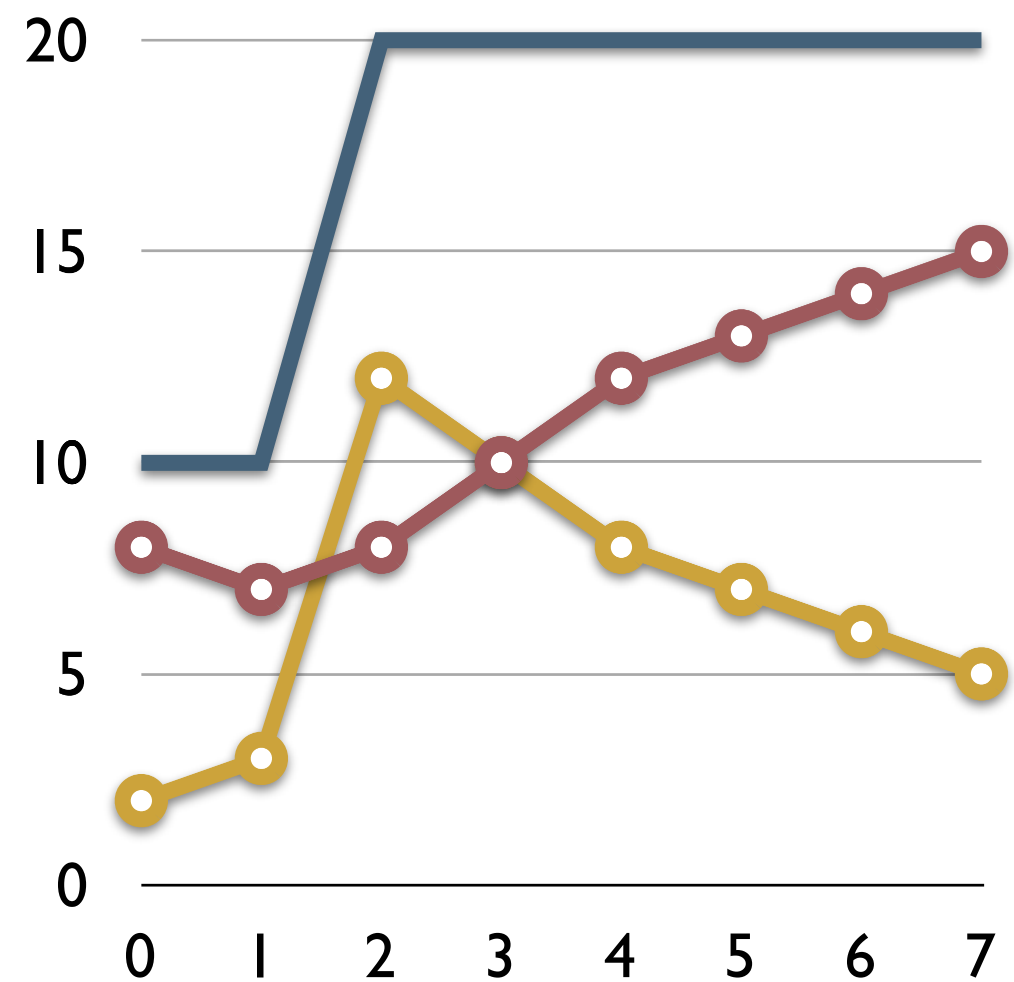
- To the left: one should slow down the grow rate to avoid overshooting the reference
- To the right: that rate should be accelerated



P controller

How the error behaves:





Quick decrease

ref. output error

Slow decrease

PD Controllers

- By taking into account the first derivative of the error, we can have a more precise controller
  - We'll use the derivative to predict how the error behaves
- Such controllers are called Proportional and Derivative (PD)
- Control law is of the form:
  $u_t = P \times e_t + D \times (e_t - e_{t-1})$
  - When D=0, we have a P controller

Example

- Let's go back to our example, considering r_t is constant and w_t is null.

$$v_{t+1} = 0{,}7v_t + 0{,}5u_t - w_t$$

$$u_t = P \times e_t + D \times (e_t - e_{t-1})$$

$$e_t = r - v_t$$

We get:

$$v_{t+1} = (0{,}7-0{,}5\times(P+D))\times v_t + 0{,}5\times D\times v_{t-1} + 0{,}5\times P\times r$$

Example

- Let's consider again the steady states: $v_{t+1} = v_t = v_{t-1} = v_{ss}$

$$v_{t+1} = (0{,}7-0{,}5\times(P+D))\times v_t + 0{,}5\times D\times v_{t-1} + 0{,}5\times P\times r$$

$$v_{t+1} = v_t = v_{t-1} = v_{ss}$$

We get:

$$v_{ss} = (0{,}5\times P/(0{,}3+0{,}5\times P)) \times r$$

Example

- Let's consider again the steady states: $v_{t+1} = v_t = v_{t-1} = v_{ss}$

$$v_{t+1} = (0,7-0,5\times(P+D))\times v_t + 0,5\times D\times v_{t-1} + 0,5\times P\times r$$

$$v_{t+1} = v_t = v_{t-1} = v_{ss}$$

We get:

$$v_{ss} = (0,5\times P/(0,3+0,5\times P)) \times r$$

It is the same result as with the P controller

Example

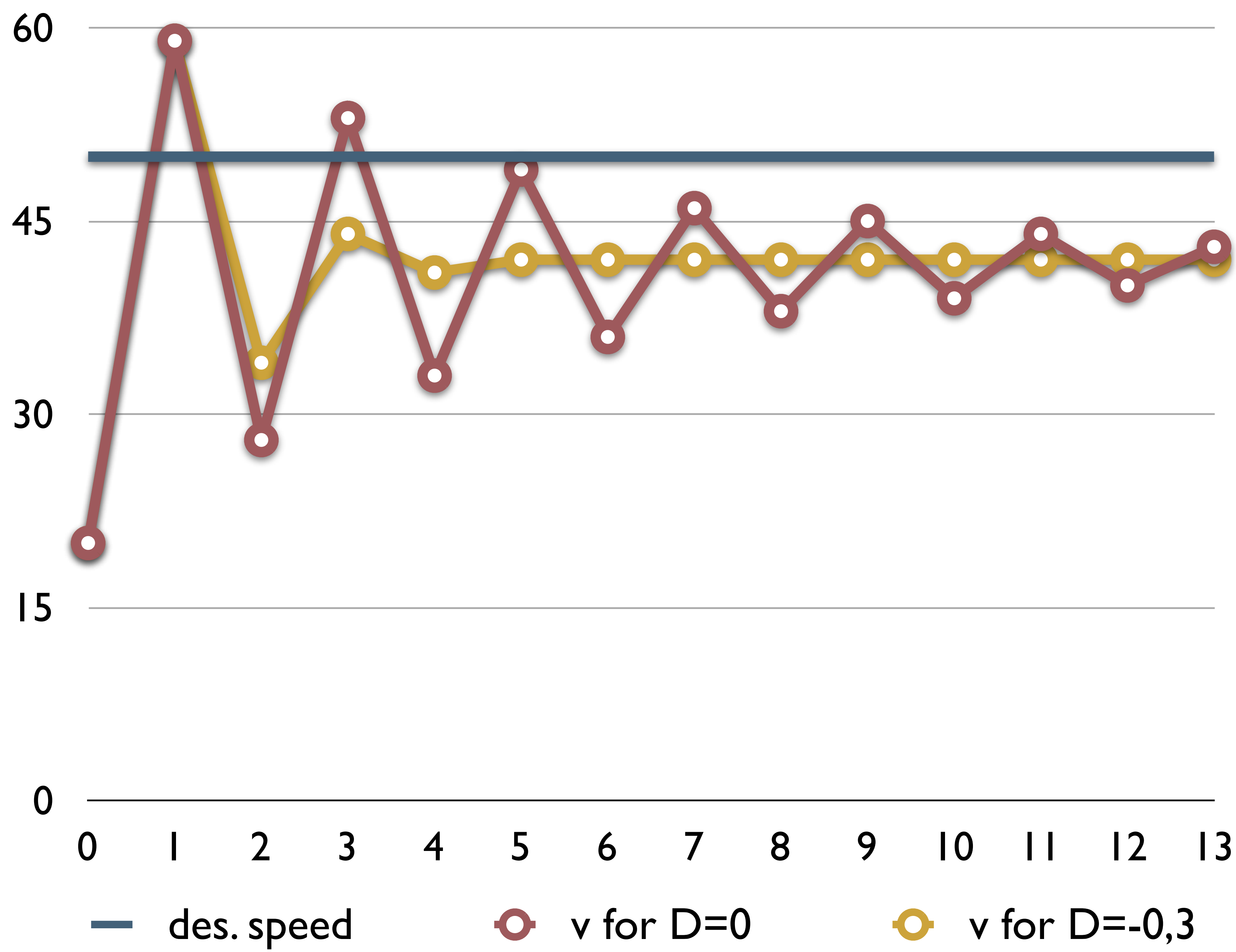
- The equation describing steady states is the same as for the P controller
- This is not (should not be ?) surprising. In the steady states, the output does not vary, neither the error !
- Adding a derivative term does not affect steady states
- But it allows more flexibility

Example

- Remember: With P controllers, when P was high, we had high accuracy but many oscillations
- We will adapt the value of D in the PD controller to avoid oscillations
- D will clearly have a negative value: if the error grows fast (first derivative highly positive) we have to reduce the input to the environment.

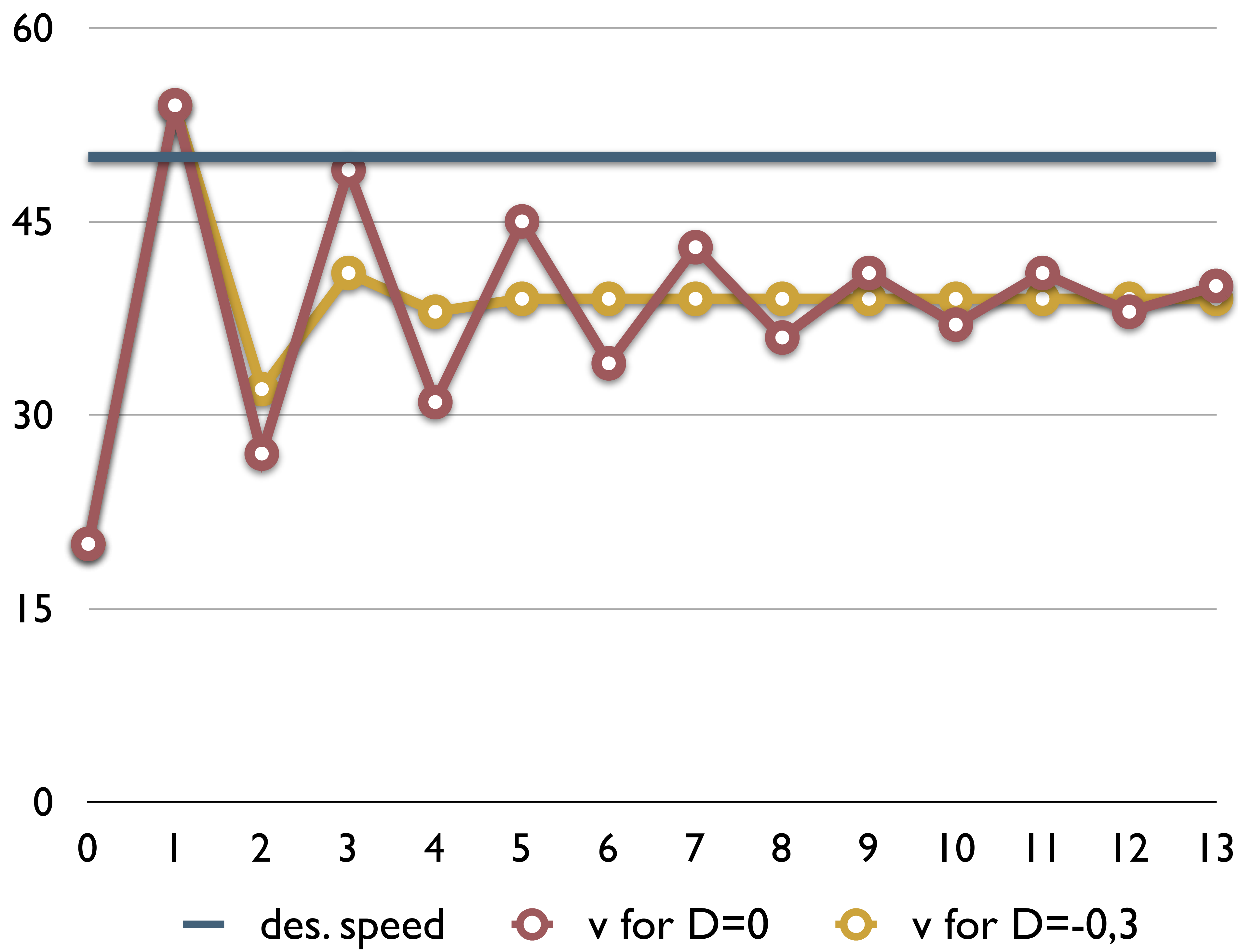
Examples

w = 0 r = 50 P = 3 $u_t = P \times e_t + D \times (e_t - e_{t-1})$



Examples

w = 5   r = 50   P = 3 $u_t = P \times e_t + D \times (e_t - e_{t-1})$

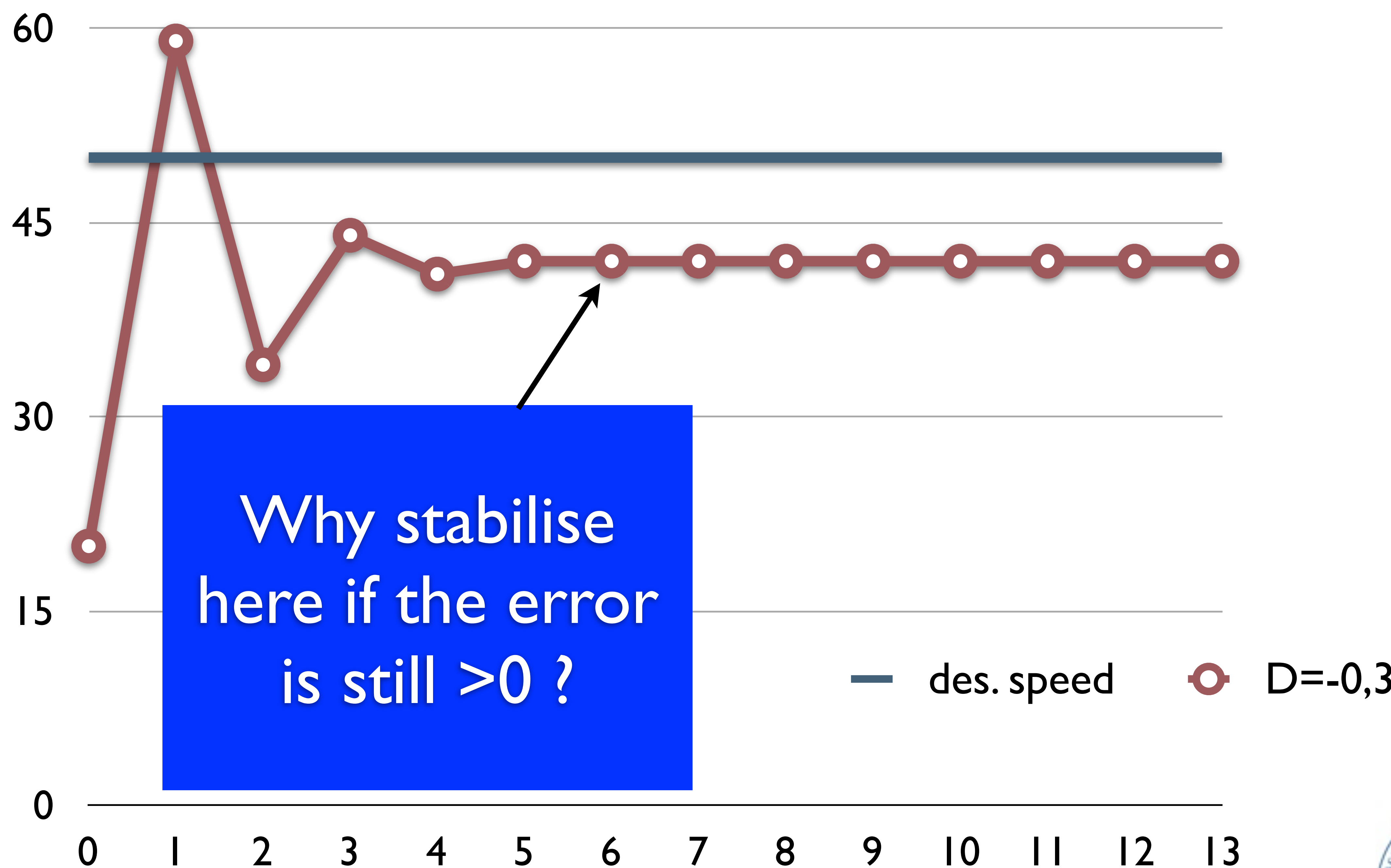


IP controller

- Adding the differential term has allowed to reduce oscillations
- Unfortunately, accuracy is still poor

Poor accuracy

w = 0   r = 50   P = 3



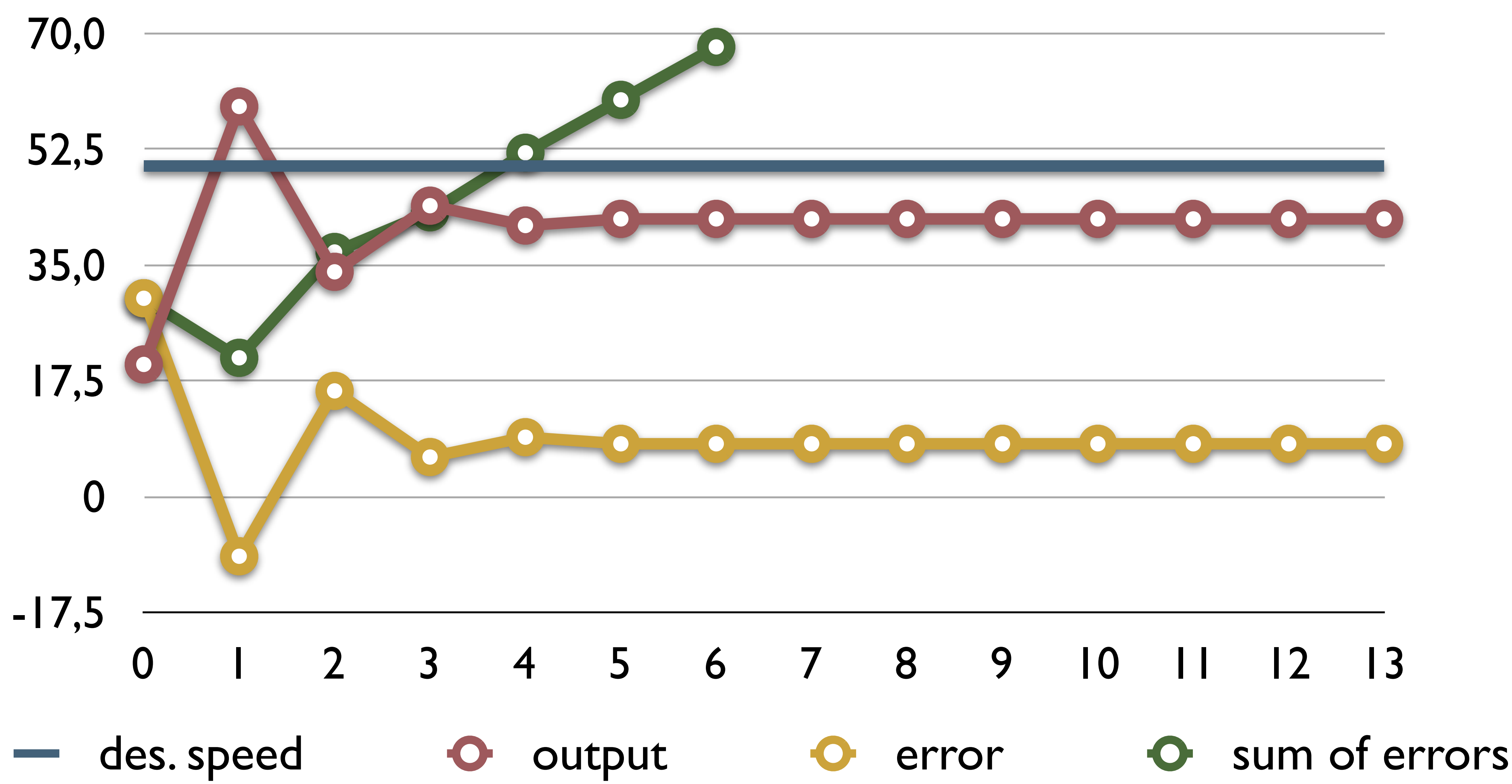
Poor accuracy

- We have seen that in steady states, the DP controller behaves like a P controller (for the same value of P)
- There is thus no reason that a PD controller converges with more accuracy than a P controller
- To obtain higher accuracy, we, should take a larger P, but then the system diverges...

IP controller

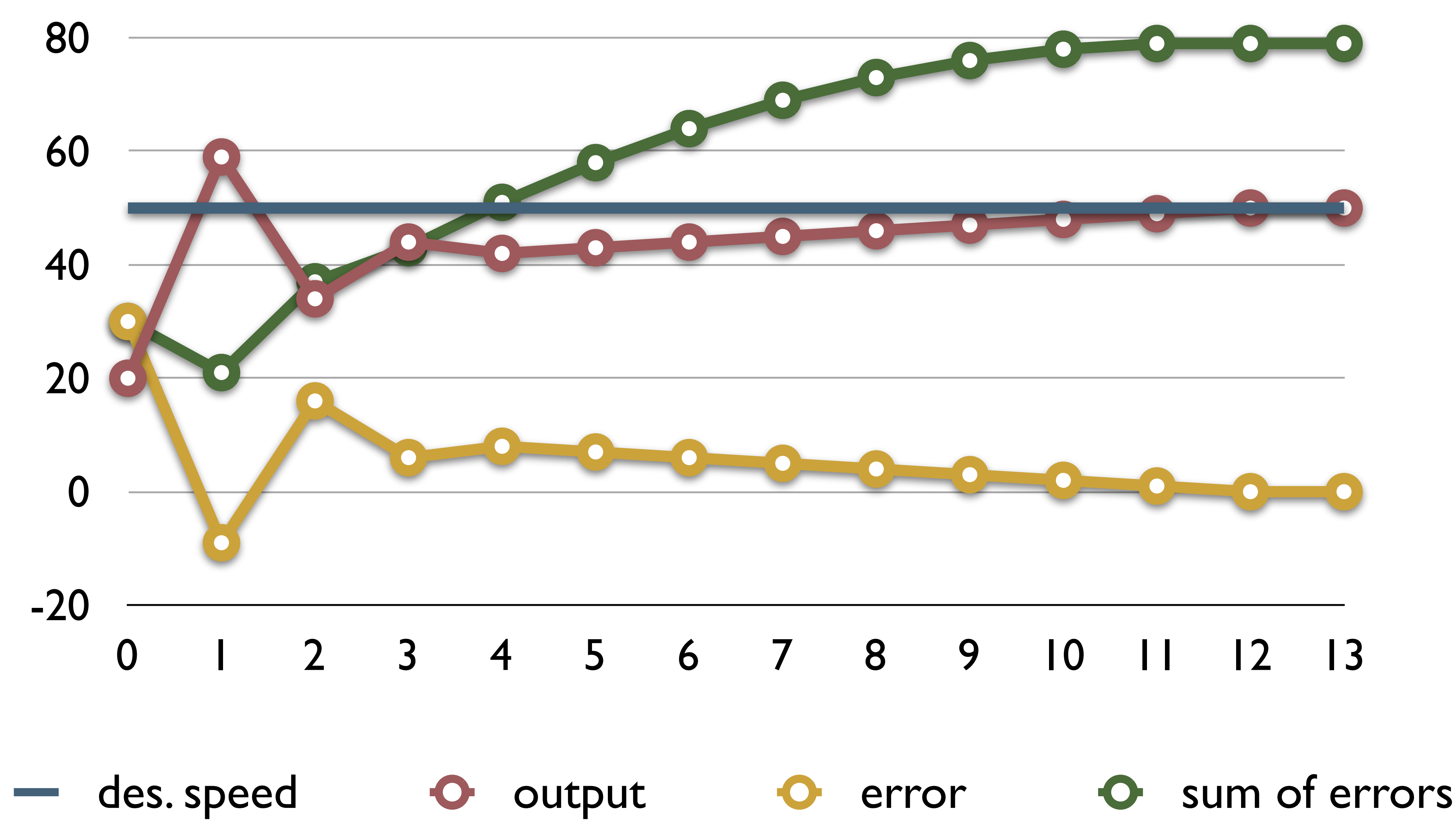
- A solution consists in taking into account not only the punctual error, but also the sum of all errors since the initial point.
- This is an Integral - Proportional (IP) controller

Convergence



In this example speed and error converge
but the sum of errors does not !

Convergence



If we manage to have the output converge to the reference, the sum of errors will converge too.

IP Controller

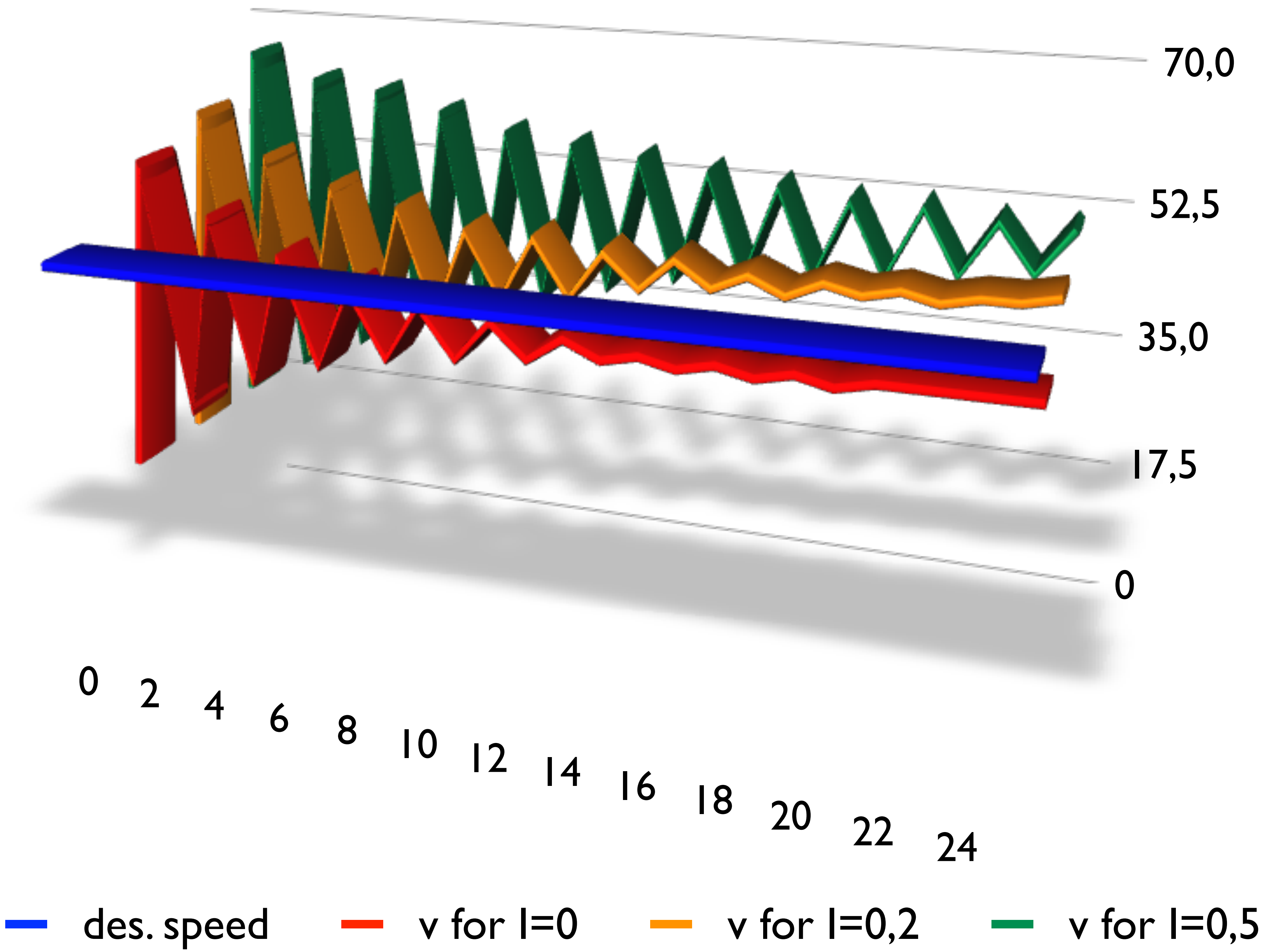
- The control law for IP controllers is of the form:
  $u_t = P \times e_t + I \times \sum_{0 \leq i \leq t} e_i$
- It contains two parameters: P and I
  - When I=0, we have a P controller
- The controller lets the error converge to 0 to avoid u_t diverging (i.e. growing infinitely)

Example

- Again, the closed loop controller
- We use P = 3 since it gave good results with the P and PD controllers

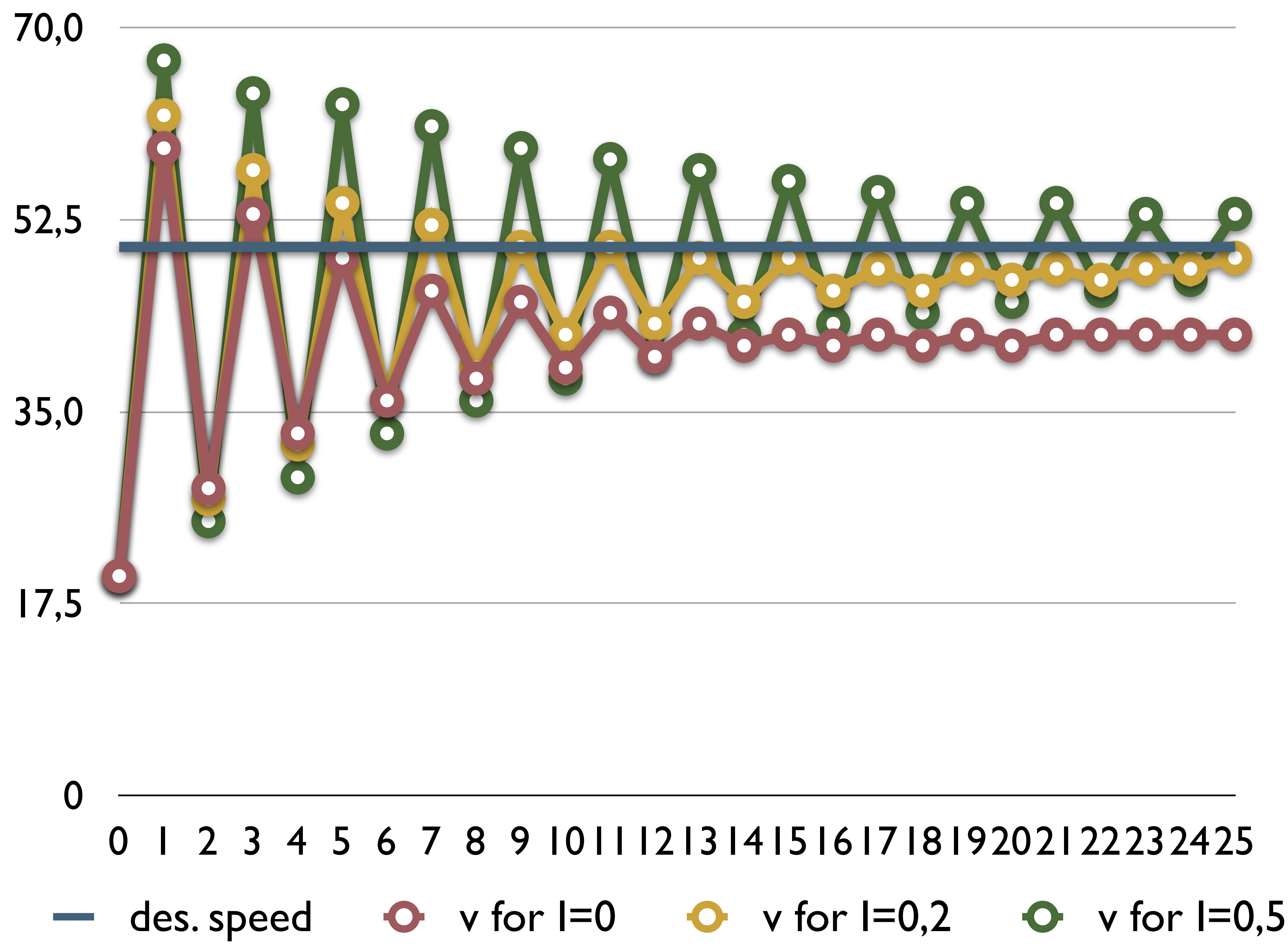
Examples

w = 0 r = 50 P = 3 $u_t = P \times e_t + I \times \sum_{0 \leq i \leq t} e_i$



Examples

w = 0   r = 50   P = 3 $u_t = P \times e_t + I \times \sum_{0 \leq i \leq t} e_i$

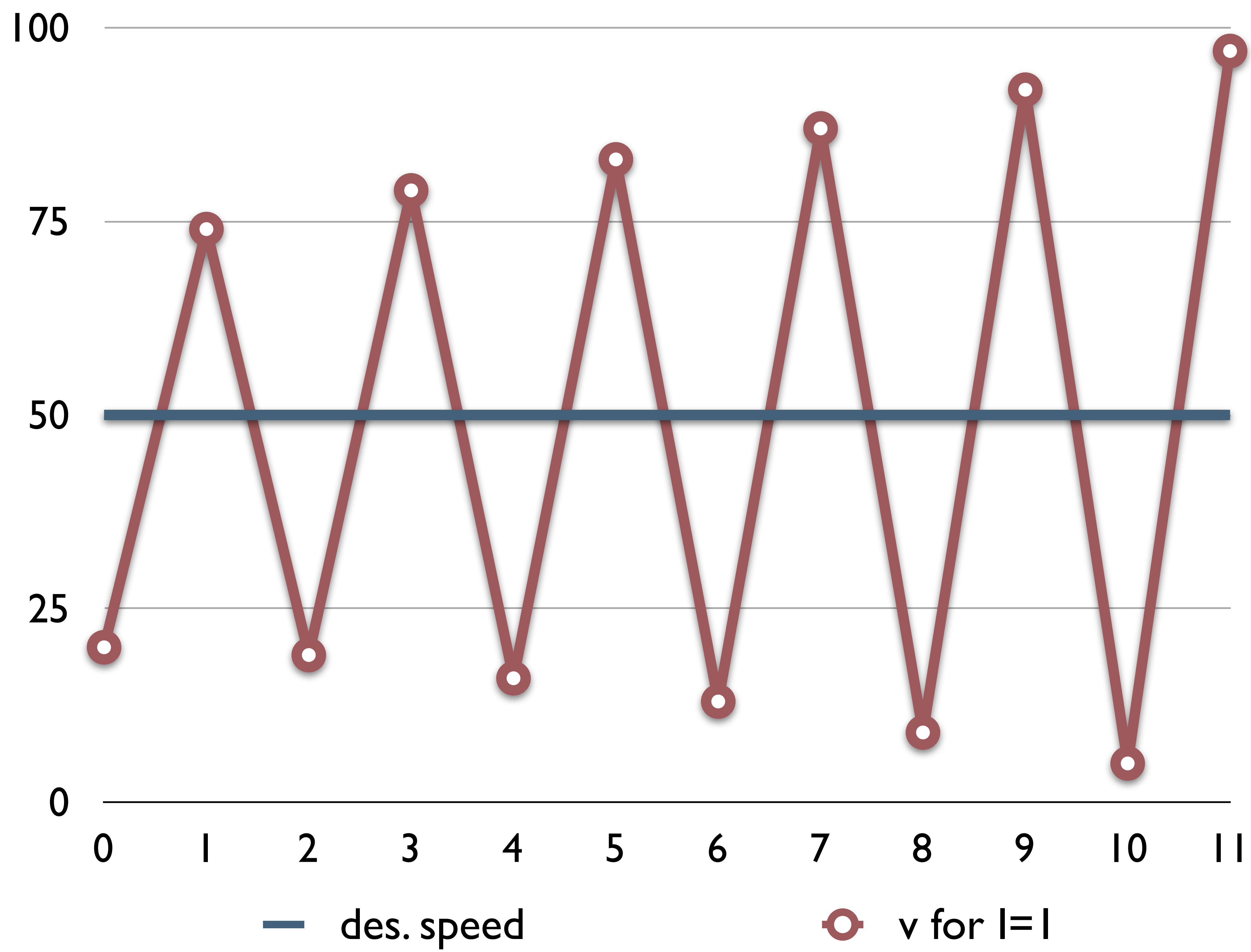


IP Controller

- With I > 0, we can see that the output converges to the reference value
- Unfortunately we have large oscillations
  - Taking a bigger I, exacerbates the phenomenon
  - For some values of I, the system diverges

Example - divergence

$w = 0 \quad r = 50 \quad P = 3 \quad u_t = P \times e_t + I \times \sum_{0 \leq i \leq t} e_i$



DP vs. IP

- To sum up:
  - The differential term allows to avoid oscillations, but does not increase accuracy
  - The integral term lets the controller converge to the reference value but exacerbates the oscillations that were already present in the P controller

PID Controller

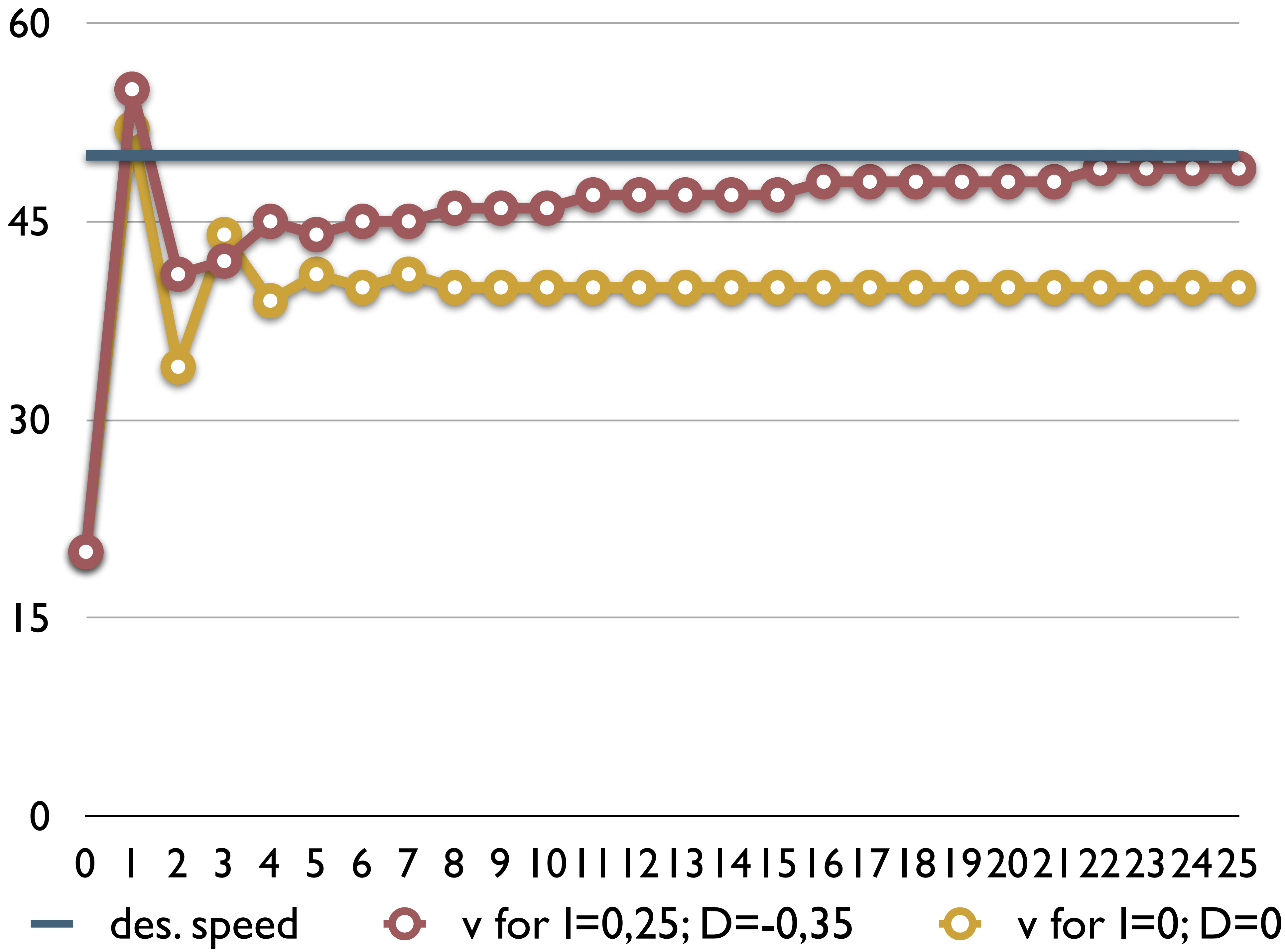
- We can combine DP and IP controllers to (hopefully) combine their advantages
- This a a Proportional - Integral - Derivative (PID) controller

PID Controller

- The control law is:
  $u_t = P \times e_t + I \times \sum_{0 \leq i \leq t} e_i + D \times (e_t - e_{t-1})$
- There are now 3 parameters, that one should tune finely
  - If $D = 0$ et $I \neq 0$, we get an IP controller
  - If $D \neq 0$ et $I = 0$, we get a DP controller
  - If $D = I = 0$, we get a P controller

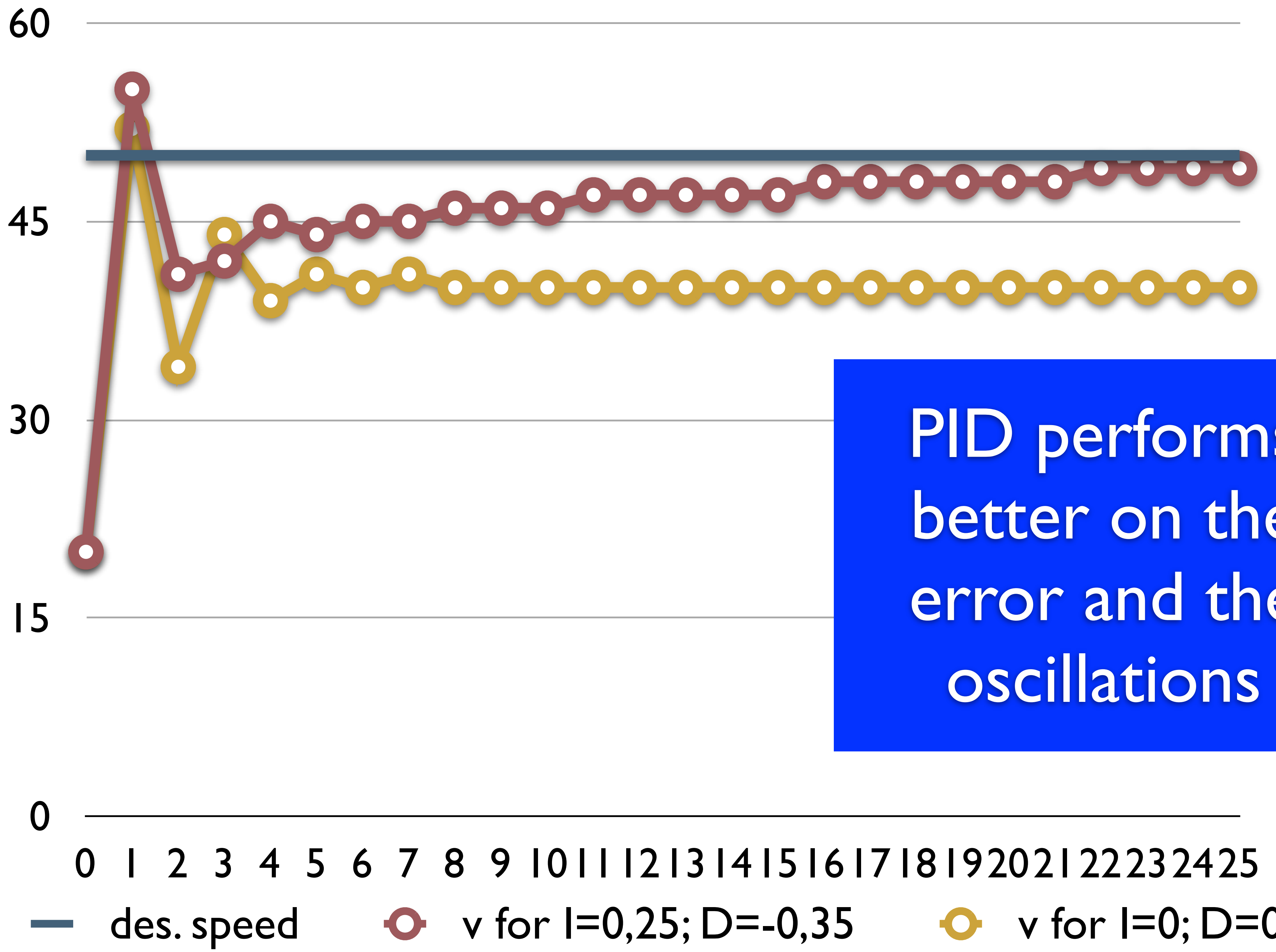
Example - PID vs P

w = 0 r = 50 P = 2,5 PID



Example - PID vs P

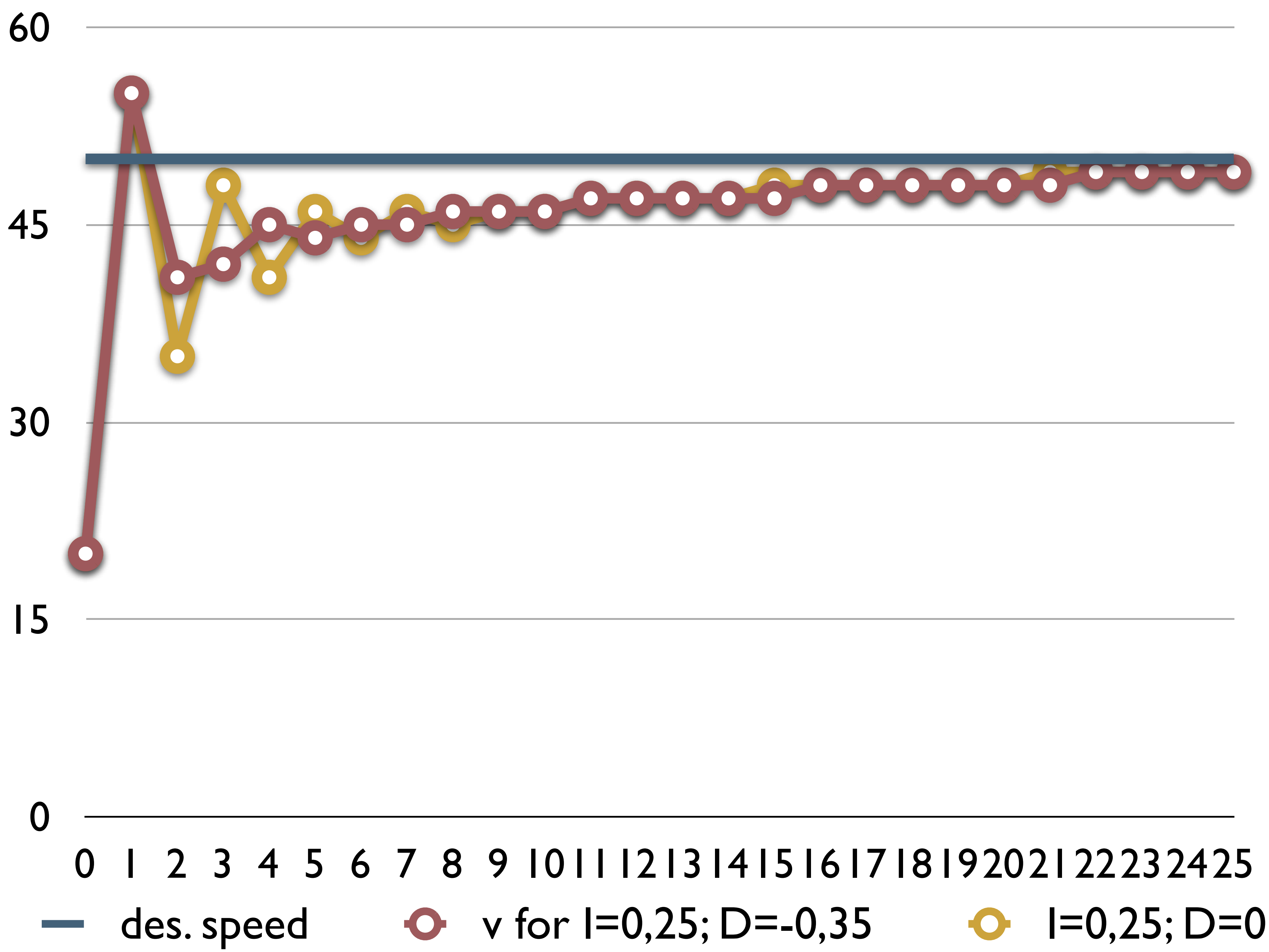
w = 0 r = 50 P = 2,5 PID



PID performs better on the error and the oscillations

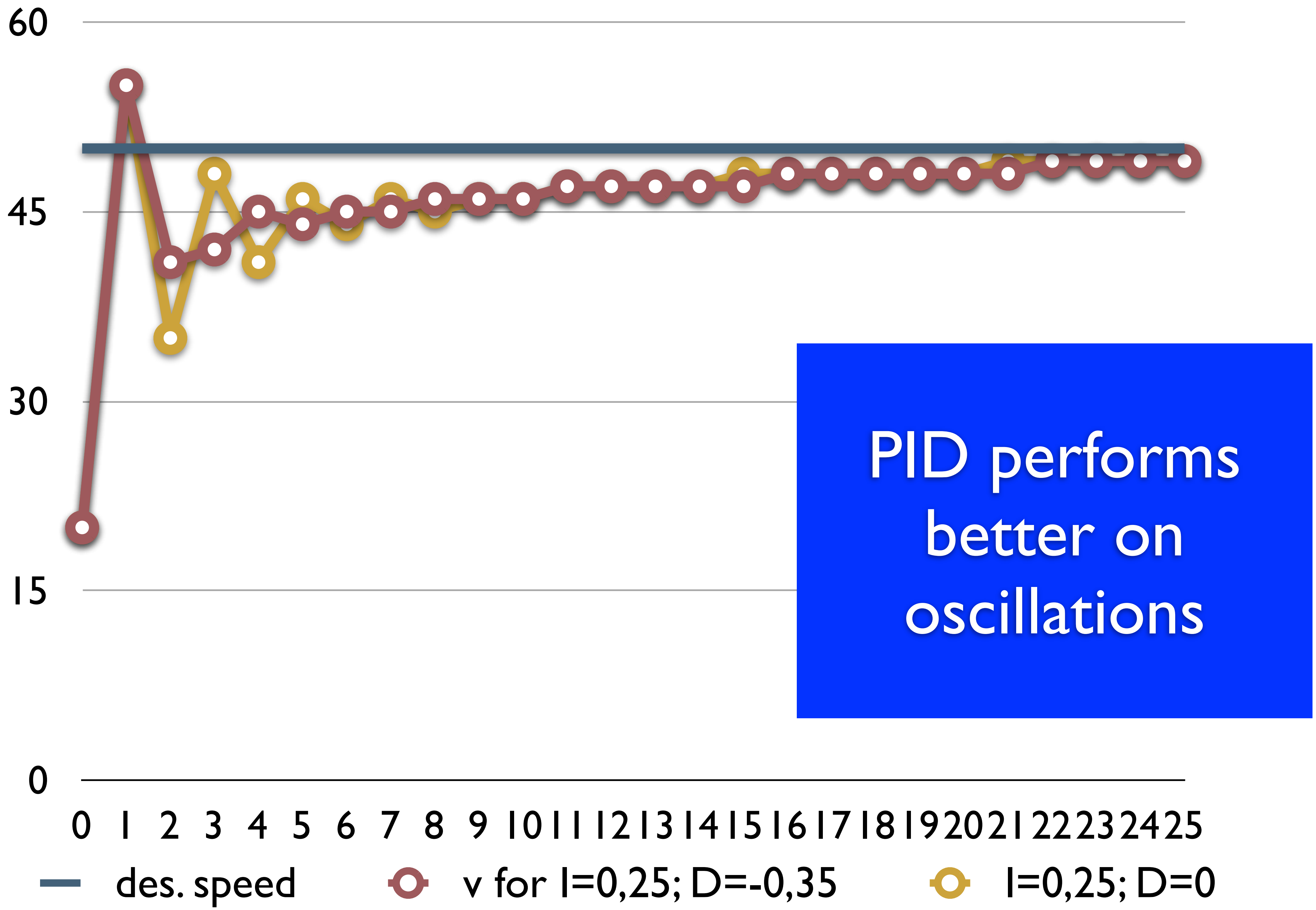
Examples PID vs IP

w = 0   r = 50   P = 2,5   PID



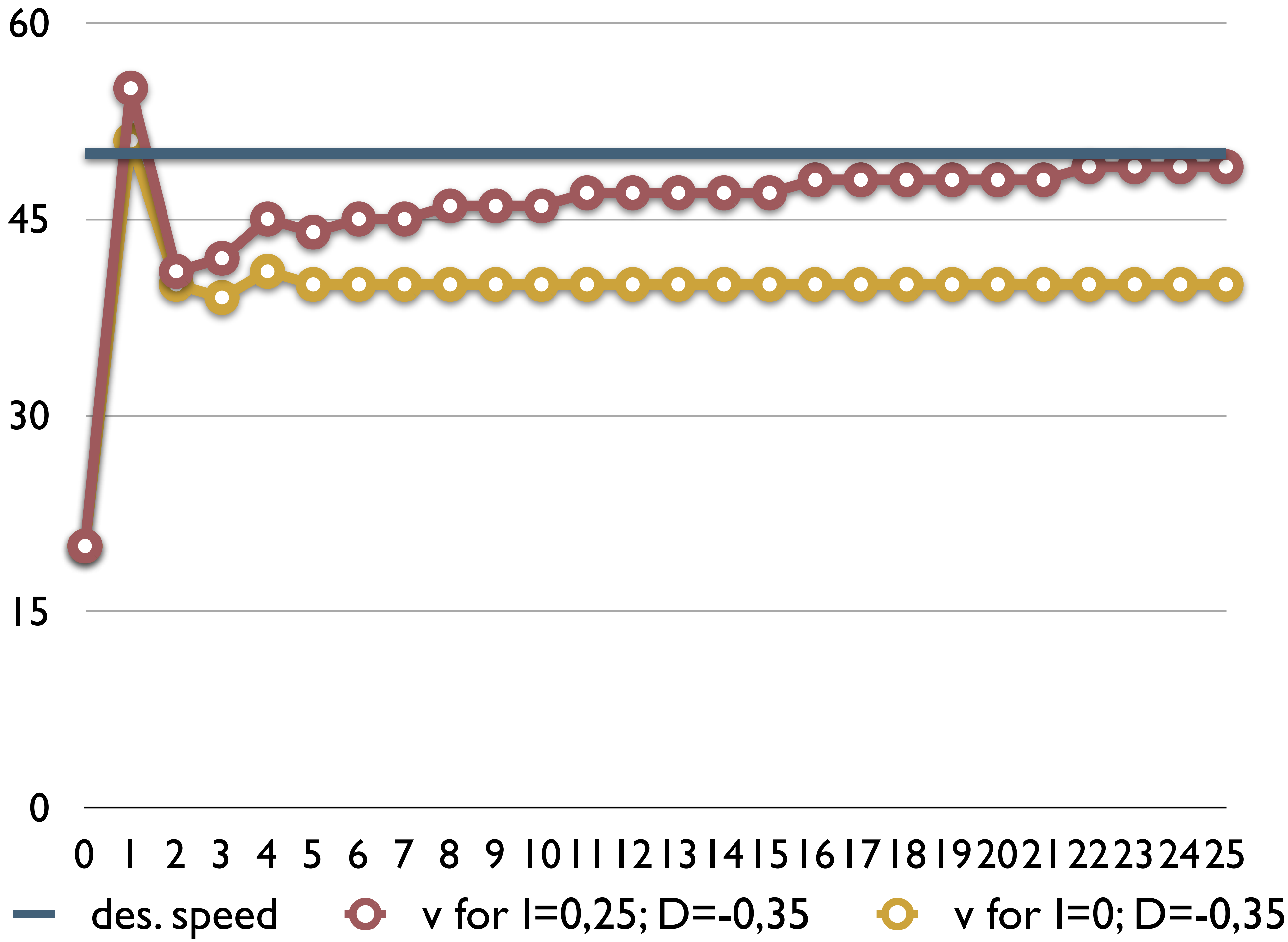
Examples PID vs IP

w = 0 r = 50 P = 2,5 PID



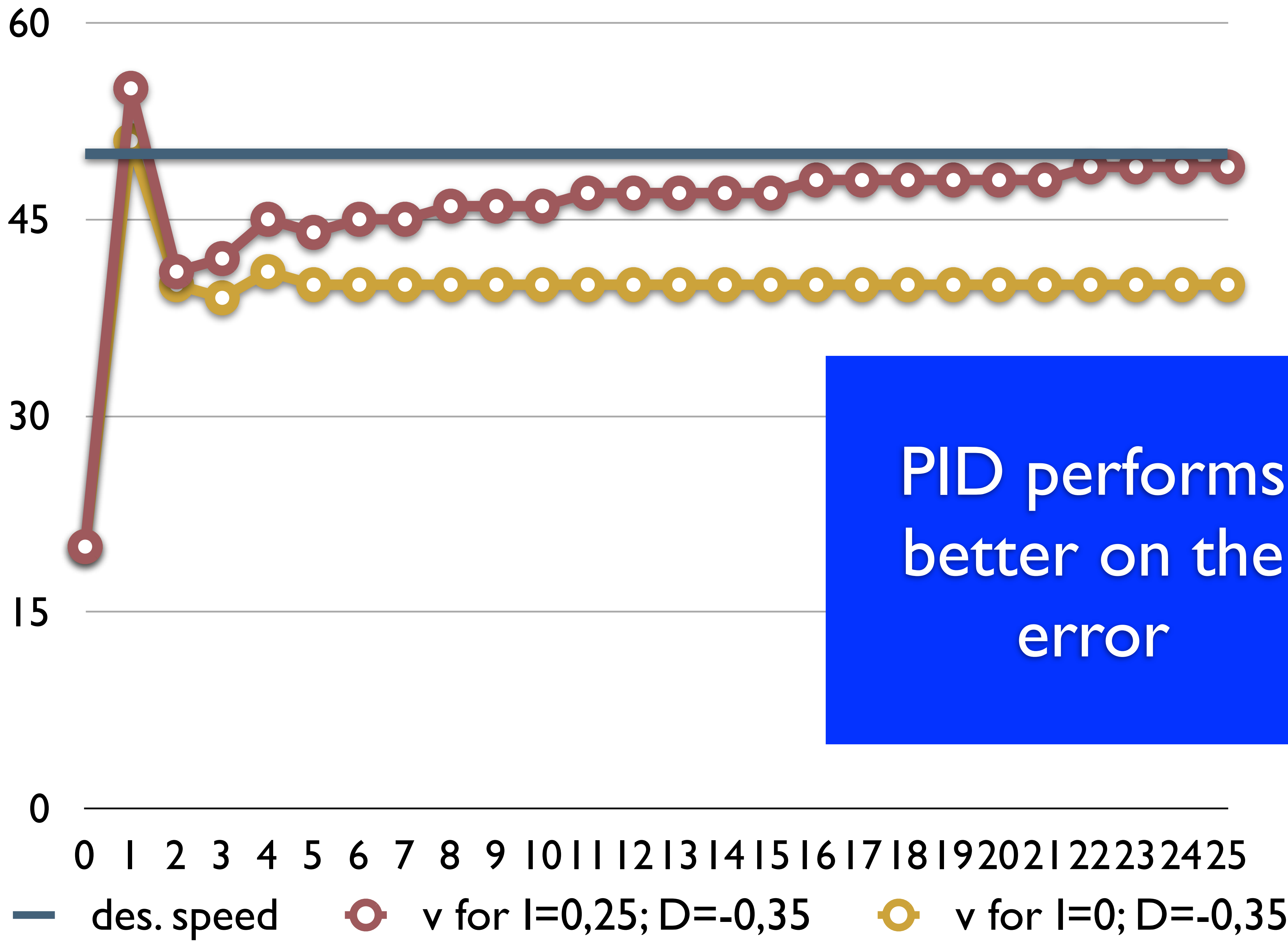
Example - PID vs DP

w = 0   r = 50   P = 2,5   PID



Example - PID vs DP

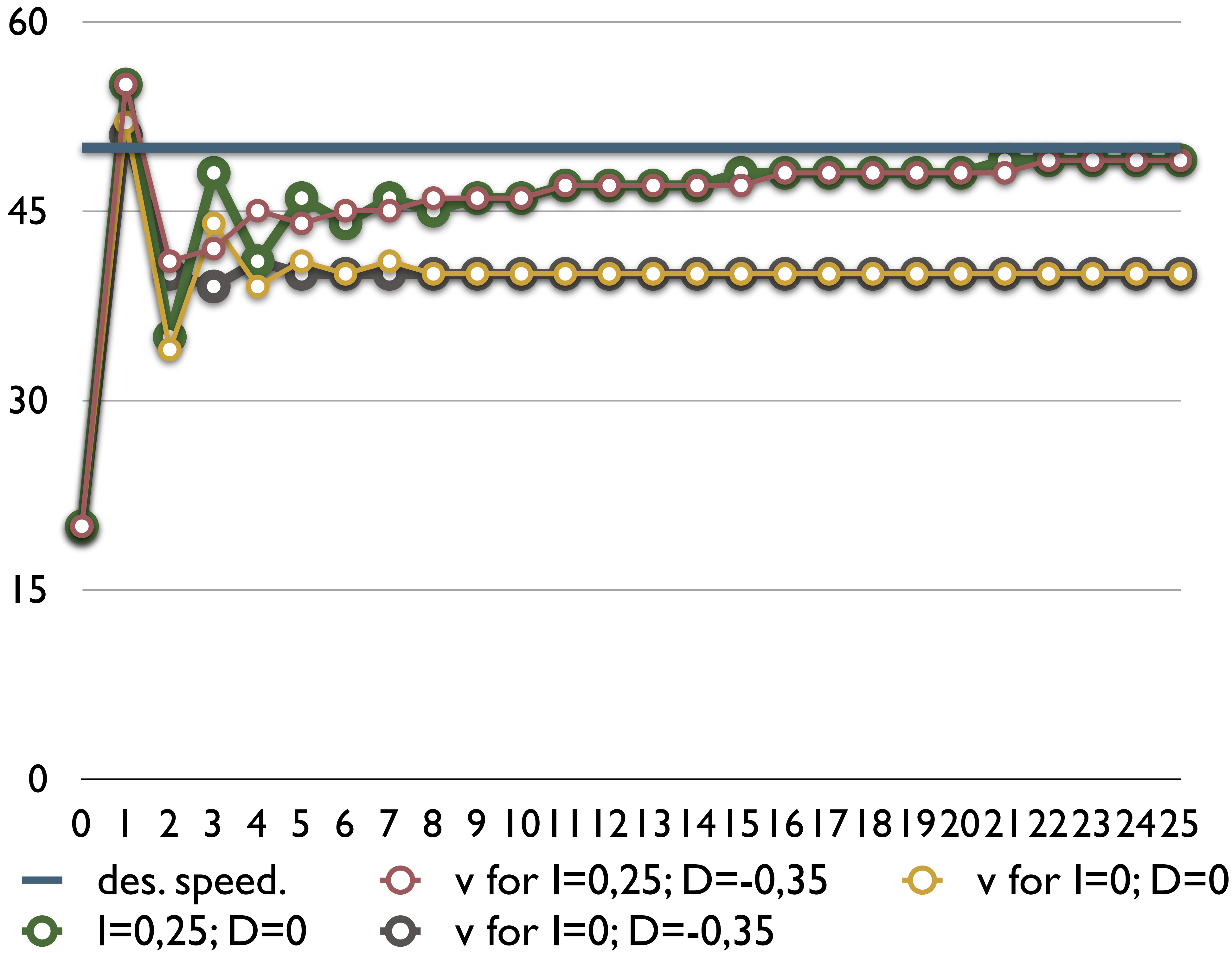
w = 0 r = 50 P = 2,5 PID



PID performs better on the error

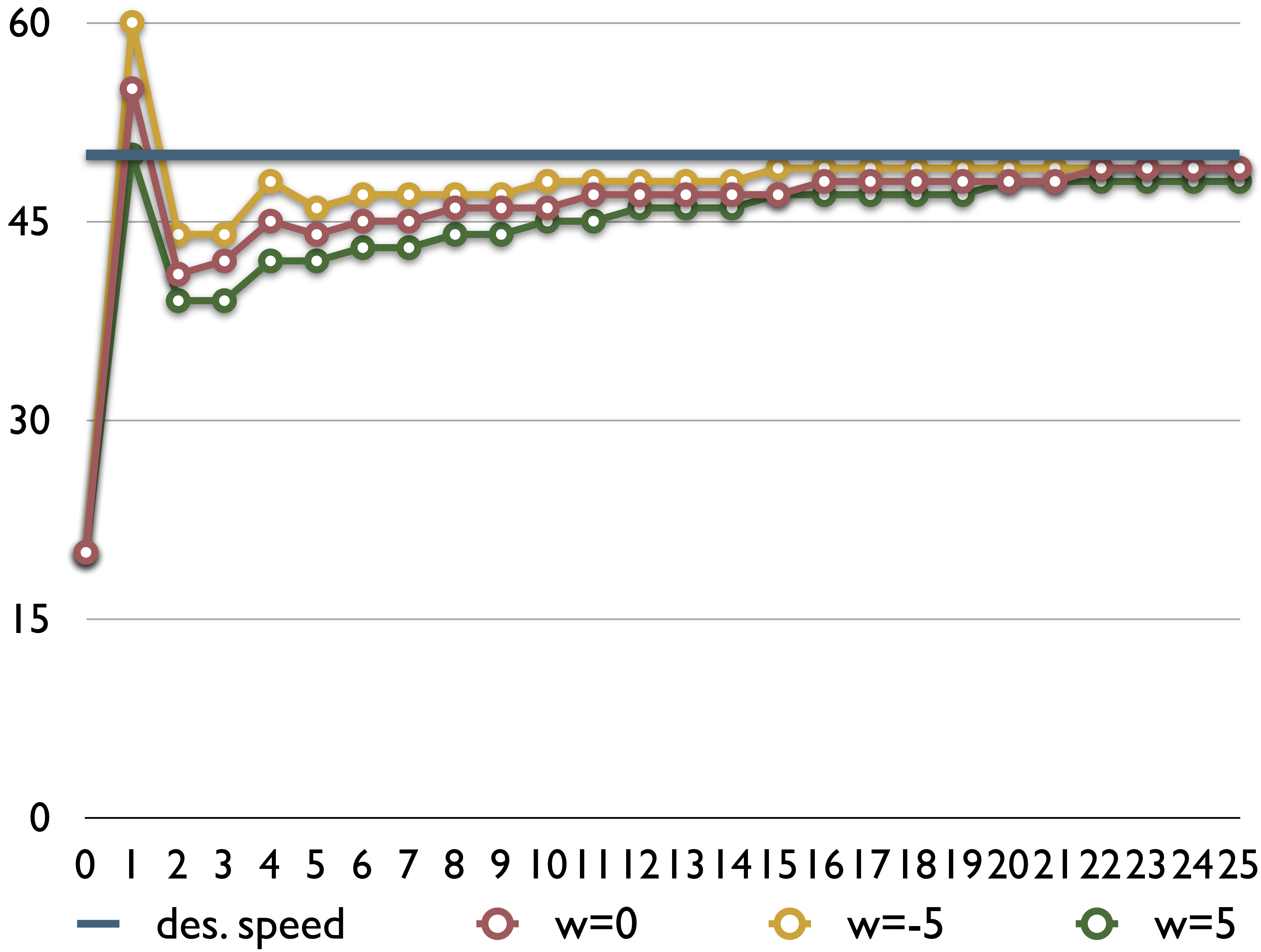
Examples

w = 0 r = 50 P = 2,5 PID



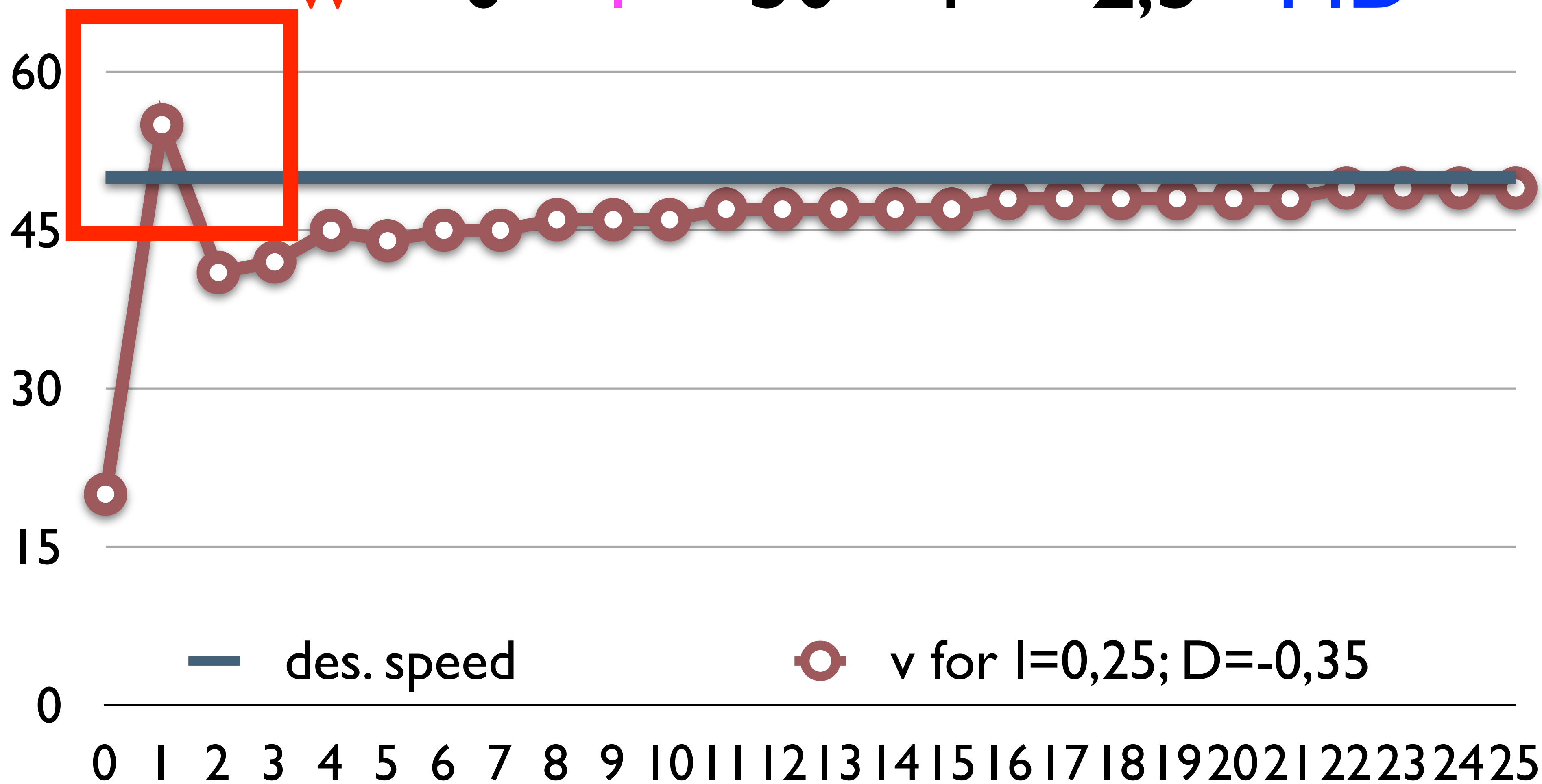
Disturbance rejection

r = 50 P = 2,5 I = 0,25 D = -0,35



Remark

w = 0   r = 50   P = 2,5   PID



The output still has a high peak value
This is due in part to the car model.
In fact, acceleration will be smoother !

Remark - Saturation

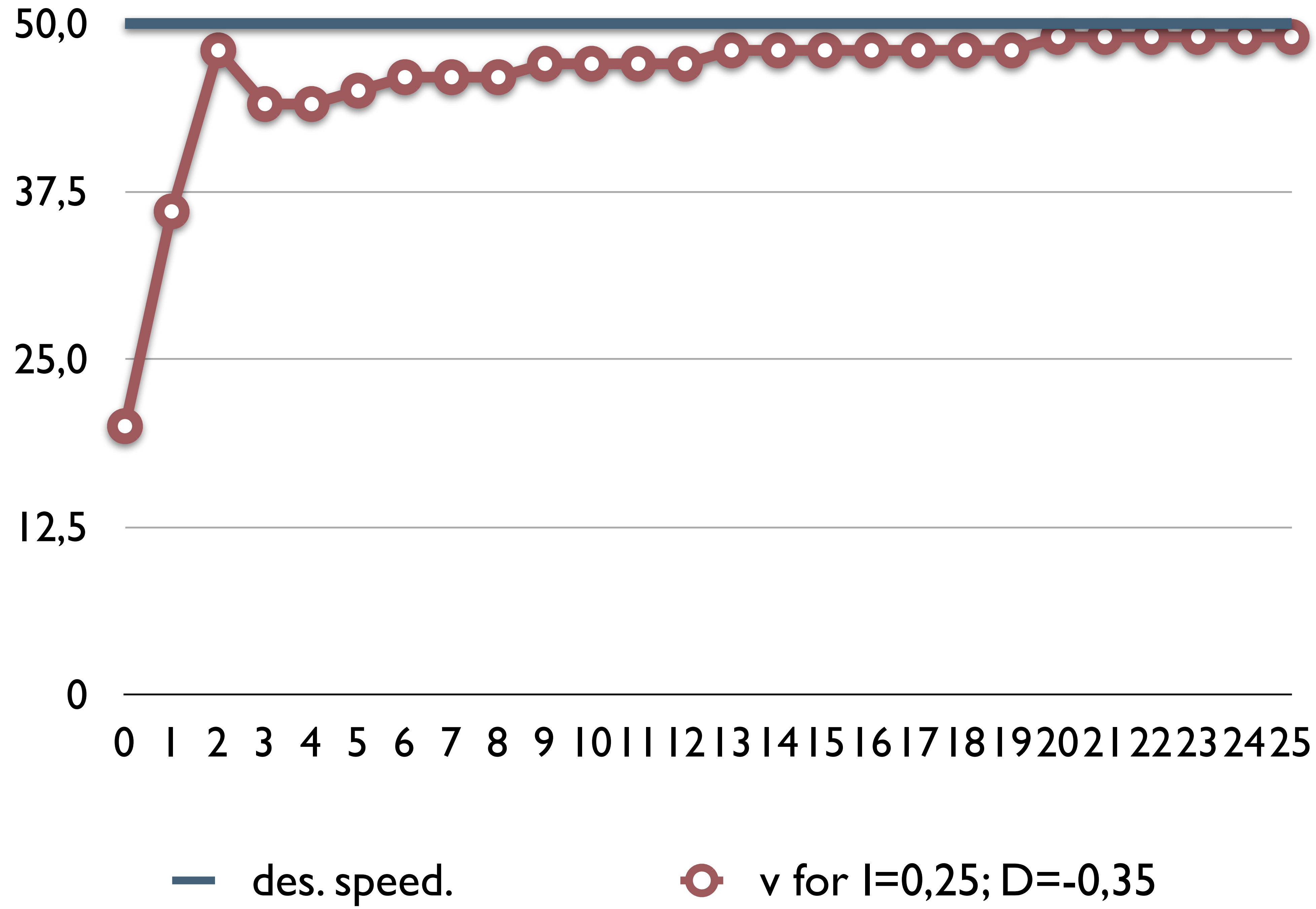
- In practice, in a realistic model, one should take into account the saturation of some variables
- For example, u_t (gas pedal) should always be between 0 and 45 !
- In our model, we haven't remarked that the controller set u_t to 90 at time 1, hence the peak

Remark - Saturation

- We can thus modify the controller:
  $u_t = P \times e_t + I \times \sum_{0 \leq i \leq t} e_i + D \times (e_t - e_{t-1})$
  as:
  $u_t = \min\{P \times e_t + I \times \sum_{0 \leq i \leq t} e_i + D \times (e_t - e_{t-1}), 45\}$
  to avoid overshooting
- The same can be done symmetrically (with a max) to avoid outputs <0

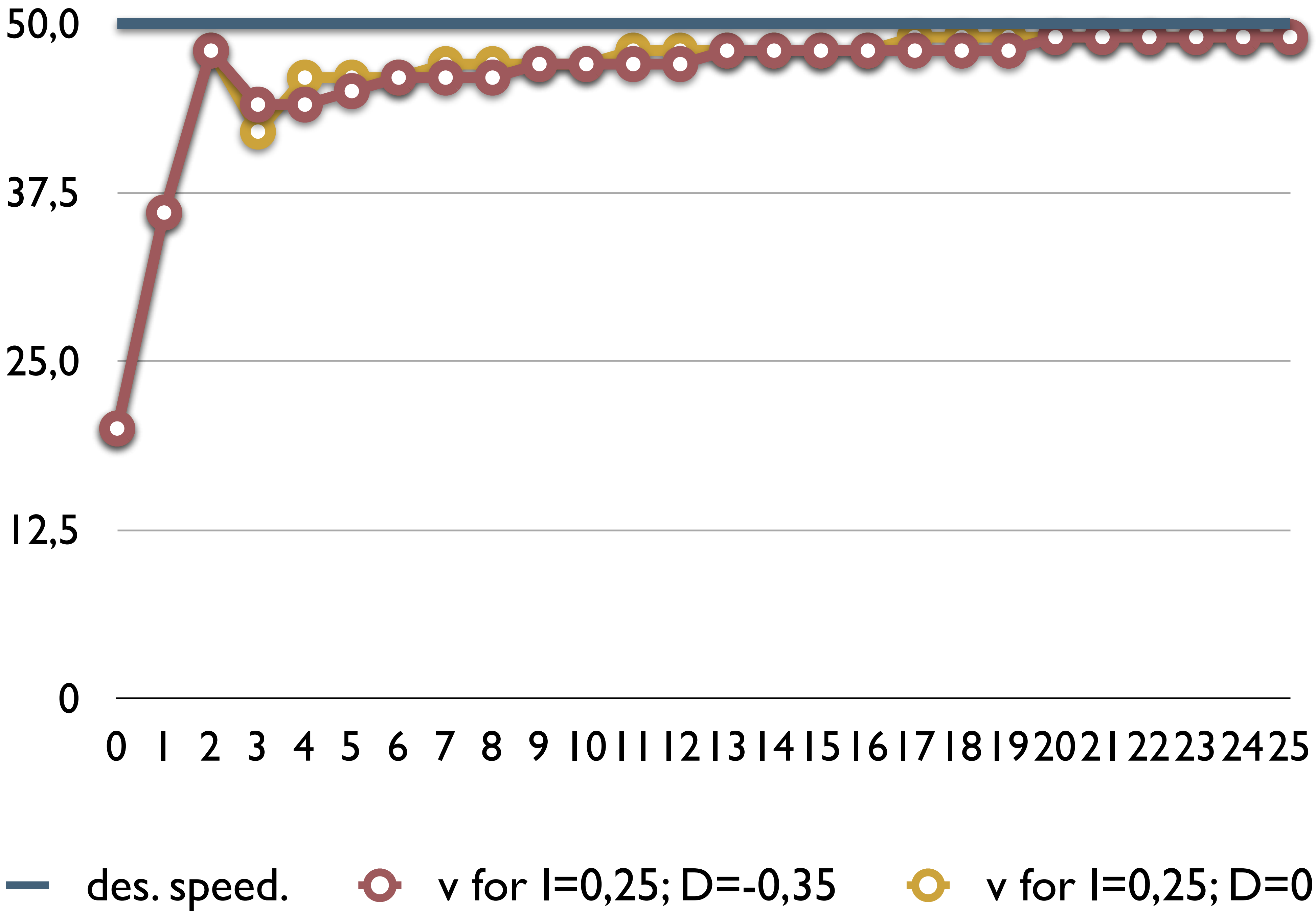
Example - saturation

w = 0   r = 50   P = 2,5   PID



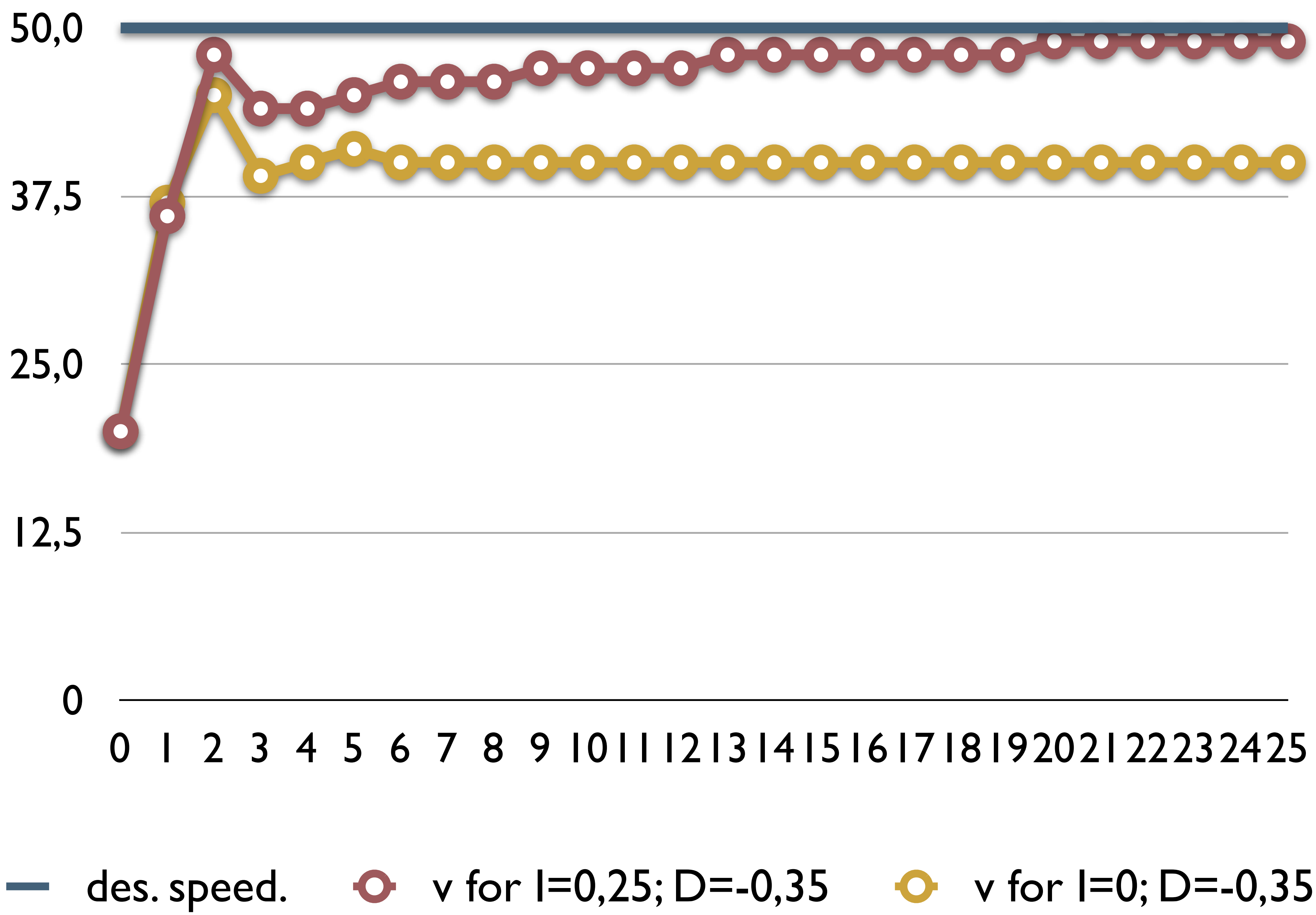
Saturation - PID vs IP

w = 0   r = 50   P = 2,5   PID



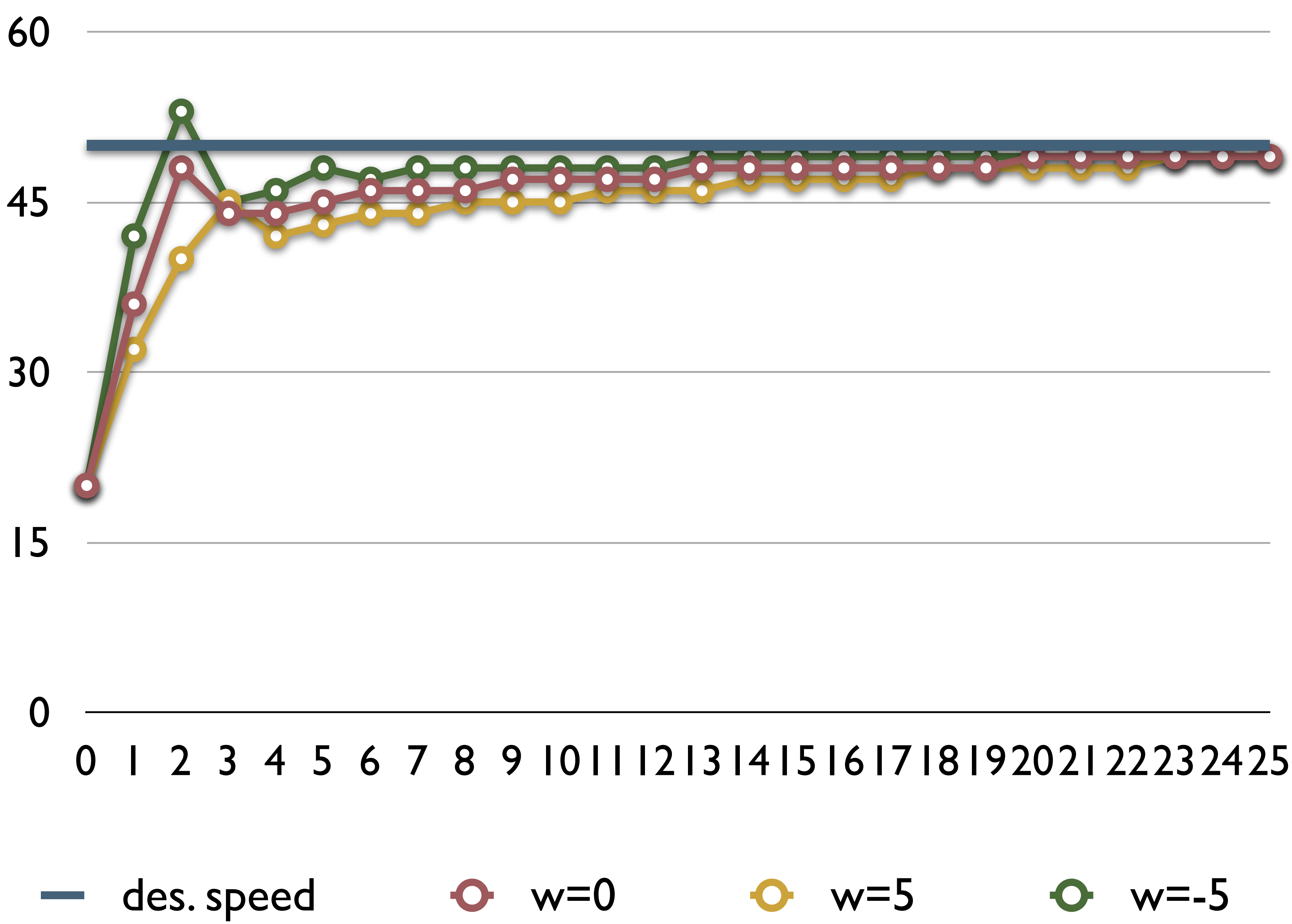
Saturation - PID vs DP

w = 0 r = 50 P = 2,5 PID



Saturation - disturbances

r = 50 P = 2,5 I=0,25 D=-0,35



Conclusion

Conclusion

- With PID controllers we have achieved:
  - fast convergence to the desired speed
  - few and small oscillations

Conclusion

- However, PID controllers are extremely simple:
  - few operations have to be computed at each step
  - trivial to implement !
  - Exist as ready-made IC !

Conclusion

- Those controllers are sufficient in many real- case applications
  - The "differential" and "integral" terms exist as operators in SCADE library