Australian National University

DESIGN PROJECT REPORT COVER SHEET

ANU College of Engineering and Computer Science
Australian National University
Canberra ACT 0200 Australia
www.anu.edu.au

+61 2 6125 5254

Submission and assessment is anonymous where appropriate and possible. Please do not write your name on this coversheet.

This coversheet must be attached to the front of your assessment when submitted in hard copy. If you have elected to submit in hard copy rather than Turnitin, you must provide copies of all references included in the assessment item.

All assessment items submitted in hard copy are due at 5pm unless otherwise specified in the course outline.

Student ID For group assignments, list each student's ID	U 6366102		
Course Code	ENGN 6223		
Course Name	Control Systems		
Assignment number			
Assignment Topic	Design of Controller for Cruise Control		
Lecturer	Dr. Guodong Shi		
Tutor			
Tutorial (day and time)			
Word count		Due Date	17/09/2018
Date Submitted	16/09/18	Extension Granted	

I declare that this work:

- ✓ upholds the principles of academic integrity, as defined in the ANU Policy: Code of Practice for Student Academic Integrity;
- ✓ is original, except where collaboration (for example group work) has been authorised in writing by the course convener in the course outline and/or Wattle site;
- ✓ is produced for the purposes of this assessment task and has not been submitted for assessment in any other context, except where authorised in writing by the course convener;
- ✓ gives appropriate acknowledgement of the ideas, scholarship and intellectual property of others insofar as these have been used;
- ✓ in no part involves copying, cheating, collusion, fabrication, plagiarism or recycling

Table of Contents

Introduction

The Cruise control is one of the most used control systems in automobile industry to ensure the safety of the vehicle and cruise at any desired speed that can be achieved by a given vehicle. This report discusses the design of controllers for an average car, Audi A4 2.0 TDI, and BMW 520d.

The necessity of cruise control is to improve the driving experience, reduce fatigue, and advance the overall efficiency of the car. There are various techniques to attain a desirable solution for this problem. This report focuses on PD, PID based cruise controller design and Fuzzy P based controller design. The document also analyzes the advantages and disadvantages of each of the controllers and modelling of each of the above-mentioned cars and the performance of each of the controller in various scenarios that are experienced on roads.

Systems Modelling and Simulations

$$m\dot{v} + {}^{1}/_{2}\, A\rho C_d v^2 = u + d$$

Analyzing the given equation gives us an insight that the equation has non-linearity induced by the v^2 which in turn produces uncertainty over the system responses. The system is linearized over a specific point v^*. Here the system is linearized over the point 50 kmph (13.88 ms^{-1}).

The first step in achieving cruise control is to linearize and compute transfer function of the plant which generates the following equation,

$$\frac{\delta v(s)}{\delta u(s)} = \frac{1/m}{s + \left({}^{AC_d\rho v^*}/_{m}\right)}$$

The parameters chosen for each of the cars are tabulated below,

Cars / Parameters	AC_d m^2	v^* ms^{-1}	ρ	m kg
Average car	0.23	13.88	1.1839	1000
BMW 520d	0.52	13.88	1.1839	1743 [1]
Audi A4 2.0 TDI	0.51	13.88	1.1839	1650 [2]

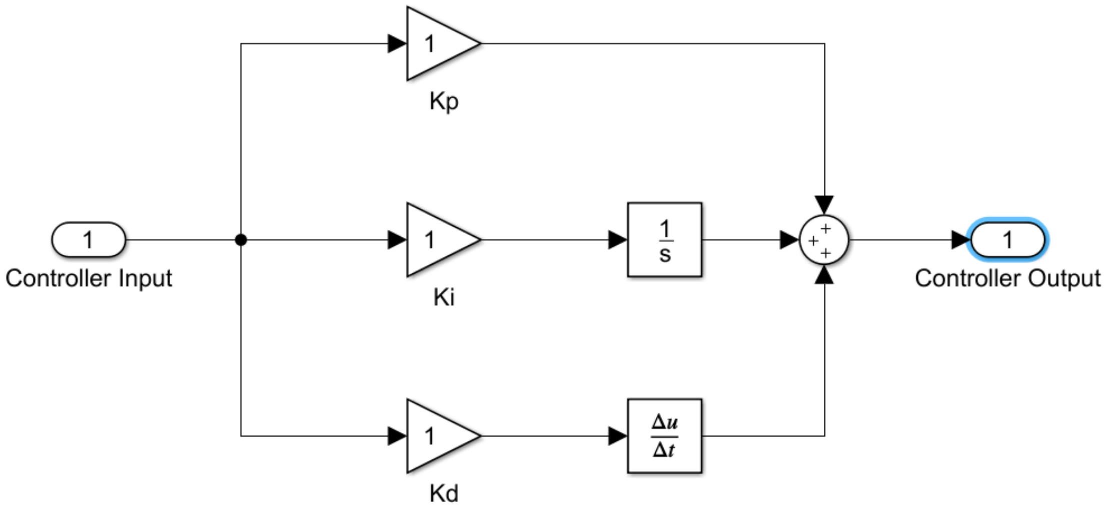
The systems modelling involves substitution of values to the transfer function to generate the plant transfer function. This function is substituted to the transfer function block of Simulink.

When the open-loop transfer function of the plant is simulated for unit step input, the plant response keeps rising as it does not have any reference value to settle. This confirms the need of controller design for this problem. Also, analyzing the transfer function also validates this stand as poles of the system remains close to the imaginary axis (in the order of 10^{-3}).

Controller Design and Validations

PID/PD controller design

The design of PID/ PD controller was one of the challenging tasks in arriving at a solution for this problem. The PID controller was manually designed using Ziegler – Nichols method [3] to understand the concepts behind the tuning of PID controllers. The controllers designed was also initially designed using the integrator, differentiator, and gain blocks to understand how a PID/ PD controller works. When designing the controller, the following block diagram was followed to implement the controller,



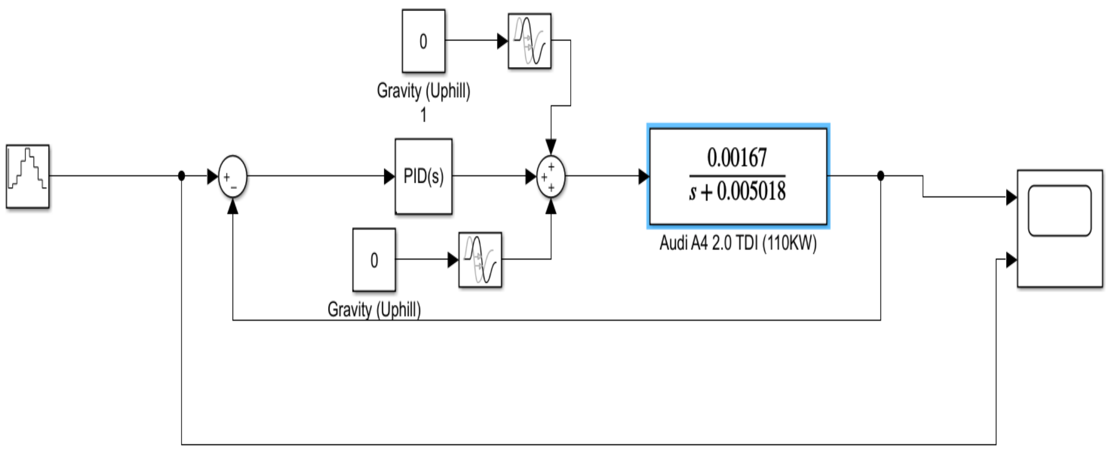
The Kp, Ki, Kd mentioned in the diagram are samples, the applied values are documented in the Appendix.

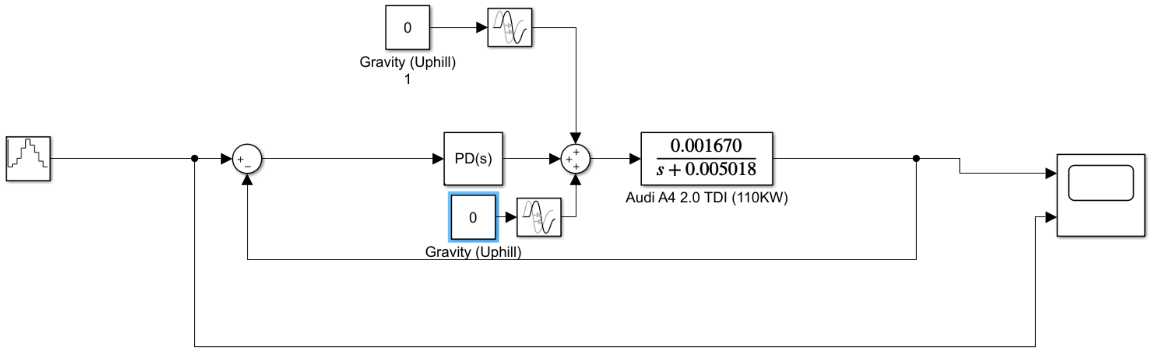
The initial open loop test conducted on the system resulted in a ramp response with slow settling time and large settling error. The open loop was made into a closed loop using proportional control alone. The system response was satisfactory, but the steady state error was about 5-7% on various instances like difference in mass, change of road conditions.

Later, after some research [4] I was able to confirm that whatever change in the gain of the proportional controller it won't affect the steady state of the system. This lead to testing of system with PD alone. The response of the system was better than the earlier proportional controller, yet the rise time was unrealistic. To understand the rise time of the car from 0-100 kmph, I was able to find some credible documents which provided enough details on the design specifications of the cars [1, 2]. When the controller was designed for a specific model, the controller was designed based on the rise time of each car from 0-100 kmph. BMW 520d and Audi A4 2.0 TDI has a rise time of 7.7s and 8.9s respectively. Updating the transfer function parameters, the controller gain was adjusted to the specified rise time. The controller was also validated using this criterion. The Simulink's PID block was also studied and the controller tuning methods were also studied which also resulted in similar gains as obtained through Ziegler-Nichols method.

To explore more on system performance, I tested the system with “Repeating stair sequence” in Simulink with 8s time space for each transition of signal over a period of 120s. On testing the system, I was able to comprehend the reality of applying gradient or sudden transition of speeds. Though PD controller performed satisfactorily on flat road scenario, when it came to mixed field scenario, both BMW 520d and Audi A4 2.0 TDI were unable to gradually reach the speed of 110 kmph due to the restrictions on engine power within the given timeframe. When the system was given enough time, it did settle in its set point value. Adjusting the parameters of the controller is out of scope because the gains of the controller were chosen based on the rise time specified by the company for the flat road scenario. Adjusting/ optimizing them would cause undesirable response in the flat road scenario. The average car is more like a *conceptual* car with the parameters based on the general design of 4-door sedan type cars. The controller response of the average car is considered as ideal response of cruise controller in the car. The changes in ρ does not cause any major deviations in the controller response. The weight tabulated earlier is kerb weight of the cars, when the car is operated in full load conditions, the controller responses are similar as of with a minimum load with slight variations in rise time in flat road (in order of 10^{-2}) and in mixed road, the controller reaches lower speeds with variation of about 3-5% of its value reached in minimum load conditions.

When I moved on to PID controllers in the thoughts of improving control accuracy [5], the integrator induced oscillations to the system response. As I moved from PD to PID for controller accuracy, it was clear that the differential term suppressed the overshoot that is induced by the integral of the controller. Increase of integral gain increased the overshoot in the system, so I intended to use the integral controller as a tool to reach the set point value with greater accuracy. When the controller was validated using the actual car parameters (BMW 520d and Audi A4 TDI), the controller response on the float road scenario had slight variation in settling value with 2 - 3% increase. A significant difference was seen in the mixed field scenario where the controller delivered a 3 - 5% increase in the settling value compared to the PD controller. The Simulink model of the PD and PID controllers implemented cruise control is,





The models show the implementation of cruise control for Audi A4 2.0 TDI using PD and PID controllers. Models of other cars and their explanations are documented in the supporting document.

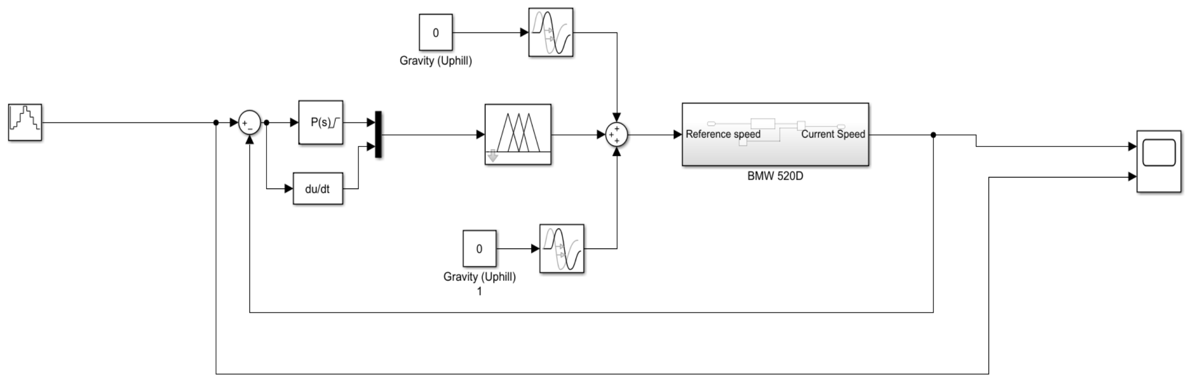
The gains of PD/ PID controller must be changed manually over every scenario to understand the performance of the controller. A car may be driven various regions in a country. So, when designing PD/ PID controller for cruise control of a car it is better to design controllers which are region specific to ensure the better satisfaction of the customers. Also, if the company management decides to design controller for various regions, it would be a wise choice to go with adaptive control where the gains of the controller can be changed on real time. Another feasible solution would be every car can have more than one controller which can be activated manually through switches for various regions of the country (like economy/ sport mode selection which stops/ kick starts turbochargers in engines). The only disadvantage of this solution is that it would shoot up the costs of the overall unit.

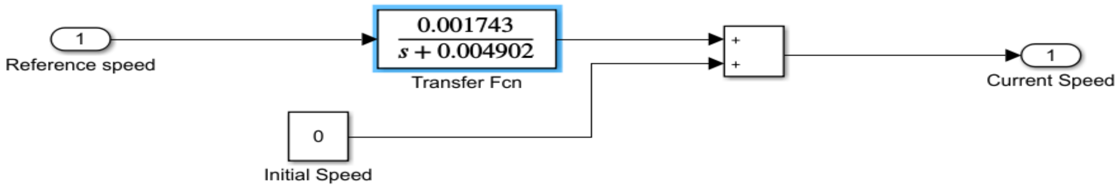
Fuzzy P controller design

To overcome this disadvantage, I implemented P controller alongside Fuzzy logic controller. Fuzzy is a rule-based algorithm which covers a wide area of errors and can be implemented for unpredictable systems. As mentioned earlier P controller reaches steady state value with up to 5% error from setpoint value. This can be rectified by coupling P controller with fuzzy logic controller. Fuzzy acts upon the rules to decide the output and fires it to the system. The rules of the controller are controlled using the membership functions, which ranges over a specific region. The rules are framed in the form of if-then. The controller output signals are almost similar to the PID and gives better control accuracy. Another reason for designing Fuzzy P controller is because it is easy to design and as it acts upon the rules alone [6], the designer just needs to frame the rules to give the output variable bounded for every bounded input which ensures that there is no unexpected behavior of the controller. The designer also needs to be extra careful while framing the rules because if one of the inputs doesn't have any rules then the controller output will be unpredictable.

To design Fuzzy controller in MATLab, one can use fuzzy toolboxes and call the variables to Simulink through Fuzzy block by exporting it to workspace. The rule viewer also helps to analyze the output for each possible value of error. I designed the controller considering a maximum error in the system as 100, so the driver of the car can go from 0-100 kmph in a single stretch and later adjust the car for higher speeds.

The controller can also be modified to accept error values (say 110), when error acceptance range is higher, the number of membership functions and rules increases to improve the accuracy of the controller output which increases the complexity of the system. This controller can also be coupled to perform other tasks like cruise control and braking which are highly related to each other and ensures the safety of the driver/ passengers. This can be done adding one more variable for braking and frame rules around error, error derivatives and cruising speed of the vehicle. The above controller is designed with future scope of the controller in mind to incorporate braking and many other features. The Fuzzy based controller model for BMW 520d is shown below,





The Fuzzy P controller was tested out for its responses for triangular and gaussian membership functions [7] for the inputs and outputs of the controller. The rules of the fuzzy controller were framed using the following tabulation,

Error/ Error derivative	HN	MN	Zero	MP	HP
LN	MP	HP	HP	VHP	VHP
MN	SP	MP	MP	HP	VHP
Zero	SN	SN	NO	SP	SP
MP	HN	HN	MN	MN	SN
LP	VHN	VHN	HN	HN	MN

HN / HP – High Negative / High Positive

MN / MP – Medium Negative / Medium Positive

SN / SP – Small Negative / Small Positive

VHN / VHP – Very High Negative / Very High Positive

Each of the categories in the table has a certain range of values depending on the car and there are 5 membership functions in each of the input and inputs outputs one of the 9 membership functions based on the tabulated rules. When the system over/undershoots, the output needs to be recalibrated so that over/undershoots are under acceptable levels.

The following table summarizes the controller performance under various testing conditions.

Controller	Constant disturbances/ noise	Dynamic speed performance	Ground to steep hill transition	Uncertainty due to mass
PD	Reaches the final value after a long time (averagely in 45 seconds for various cars)	Good during ideal conditions, steady state error is about 7%	Couldn’t reach desired value, but the system remains stable	Variations are about 3-5% of final value in ideal and non-ideal situations
PID	Reaches the final value averagely in 30 seconds for various cars	Better than PD controller with steady state error less than 5%	Reached the desired value when enough time is given, system is stable throughout the process	Variations are about 2-3% of final value in ideal/ non-ideal conditions
Fuzzy P	Reaches the final value in about 15 seconds for various cars	Better than PID controller with steady state error less than 3%	Reached the final value within shorter time about 15 seconds for different gradients	Variations are about 2-3% of final value in ideal/ non-ideal conditions

From the table it is clear that Fuzzy P controller outperforms other controllers in various categories. So, I recommend using Fuzzy P controller for great controller accuracy at an economical cost. I also wouldn’t say other controllers are bad, those controllers can also be improved by adding a few parts (like adaptiveness) to rectify their disadvantages.

Discussions and Conclusions

To begin with the discussions, all the designed controllers show promising results. The results in Appendix and GitHub shows designed models and their implementation. Here, I would like to present the discussion on various criteria that are important when designing a system.

Stability

Any system's top priority is stability. The controllers designed to help with improving the stability of the system is to be appreciated. The PD and PID controllers provide a wide stability range. Generally, cruise control can be enabled only at an optimal speed of about 40-50 kmph. Forcing the controller to cruise at lower speed might be harmful to the engines as they would most likely operate in low speed high gear scenario which will affect the engine over time. I tested PD/ PID controllers at lower speed to understand the system response. High gradient slopes make the system to respond sluggishly in lower speeds compared to higher speeds. Measurement noises affects it even more to respond sluggishly and speeds wobble between desired and actual speed. So, it is evident that cruise control cannot be activated until the vehicle reaches a particular speed. The PID controller for BMW 520d experiences a lower rise time in lower speeds (0-12 kmph) which in the real-world scenario will happen until 14 kmph which adds up to the conclusion cruise control won't be efficient in lower speeds. Both, PD and PID controllers were designed in parallel form, so change of gains will affect the controller phase which can lead to oscillations. Hence, it is slightly difficult to intuitively understand the behavior of the controller.

When the controller was tested in mixed field scenario, it delivered expected output only up to certain extent of change in gradient. When the gradient was high, the controller introduced a negative trend to the process variable, yet it settled to the set point when ample time was given (15 seconds). During the overall mixed field scenario, the system was stable.

In Fuzzy based controllers, the system remains stable. The timings at which rules are fired and output values play a major role in stability. Framing rules is simpler for smaller systems but as the number of inputs/ process variables increases complexity rises as the rules has to be framed accordingly so that they do not conflict with other outputs.

The controller performed very well in flat road scenario and mixed field scenario both in minimum as well as maximum load conditions. Controller shows slight oscillations which is caused as the controller is designed using the P controller alone. These oscillations can be rectified if the P controller can be coupled along with other controllers. Moreover, as the oscillations are damped oscillations with low steady-state error the controller is expected to perform as desired.

Robustness

Unexpected noise will generally drive the system to instability, any designer should design the system robust to those noise i.e., stable even when there's unexpected scenario/noise.

PD/ PID controllers are robust to small changes giving a better performance in flat road scenario making it highly suited for such regions. From the graphs, it is clear that the controller is not as robust as flat roads in extremely hilly regions.

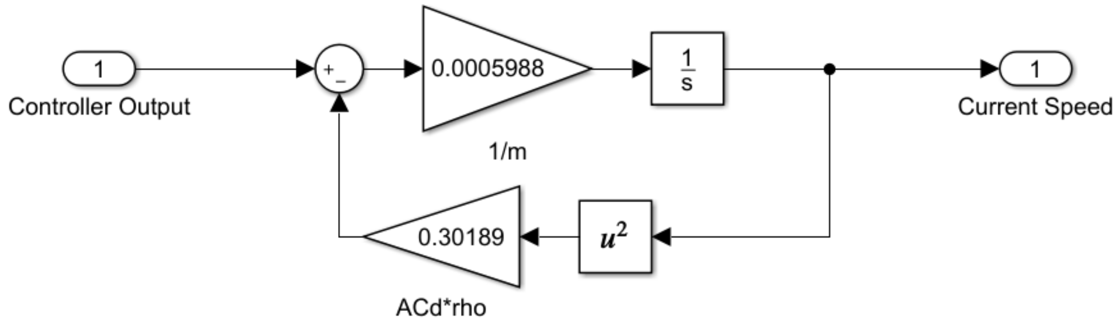
The robustness of Fuzzy P controller is good for various scenarios considered like flat road, mixed road, minimal load, maximum load conditions. Though it has slight oscillations, it eventually dies down over time. If there are more rules introduced, it improves the robustness of the system.

Though the discussion gives a general performance of the controllers, their performances for the selected cars are listed in the appendix which will help in gaining insights on how the actually performs on a real car under various situations.

Understanding company's motives and target customers and their region, the company has to decide upon what controller to use on its cars. If the target audience is having the purchasing power to buy expensive cars, the car must be facilitated with Fuzzy P controllers to give them a better experience. I recommend using Fuzzy P controllers for other customers as well without compromising the control accuracy/ quality of the car. The reason would be it outperforms the other controllers at least by a margin so I also believe overall customer satisfaction is important and the controller also gives satisfactory output when influenced by signal noises.

Non-linearity

The system was also studied using nonlinear transfer function of the system to ensure the proper functionality of the system. When the system is using PD/PID controllers the system behaved well and controllable with slight oscillations. To implement non-linearity, I built a block which generates velocity through just simplifying the given first order differential equation. When disturbances are added to the non-linear system with PD/PID controllers, it oscillates above the permitted limits which can be brought down by designing an adaptive controller to perform the complete functionality. The non-linear block implemented using Simulink is shown below,



When the same non-linear system is implemented using the Fuzzy P control the non-linearity parameter affected only the risetime and demanded more powerful output from the controller, within the power limits of the engine.

Mathematical Calculations

$$mv' + {}^{1}/_{2}\, AC_d\rho v^2 = u$$

m – mass of the car

A – Area of cross section of the car

C_d – Drag coefficient

ρ – air density

v – velocity of the car

u – engine force/ input

$$v' = {}^{1}/_{m}\left(u - {}^{1}/_{2}\, AC_d\rho v^2\right)$$

$$\delta v = v - v^*$$

$$\delta u = u - u^*$$

$$\delta v' = v' - v^{*\prime}$$

$$f(u, v) = v'$$

$$f(u, v) = f(u^*, v^*) + \frac{\partial f(u, v)}{\partial u}(u - u^*) + \frac{\partial f(u, v)}{\partial v}(v - v^*)$$

Computing Taylor series,

$$m\delta v' = (\delta u - AC_d\rho v^*\delta v)$$

Taking Laplace Transforms,

$$ms\delta v(s) = \delta u(s) - AC_d\rho v^*\delta v(s)$$

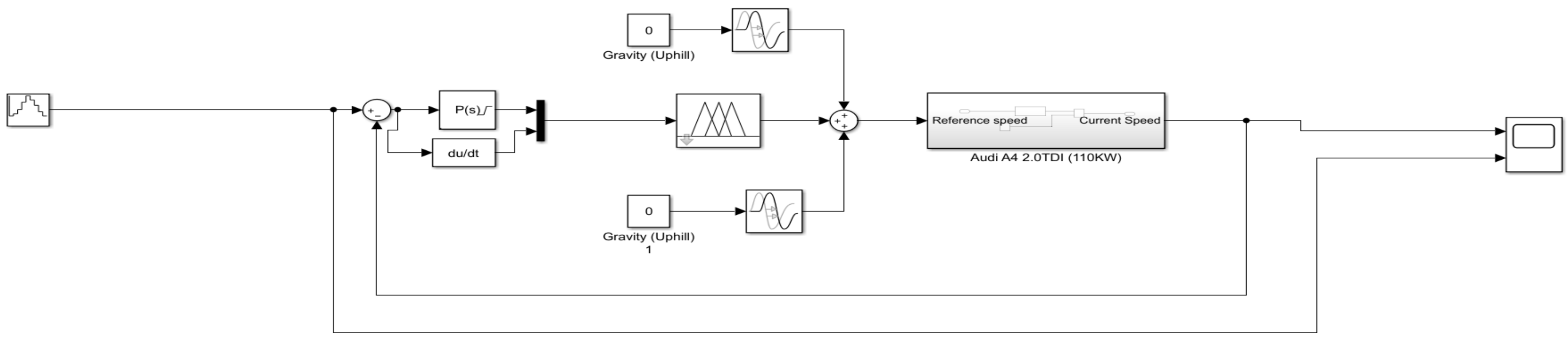
$$(ms + AC_d\rho v^*)\,\delta v(s) = \delta u(s)$$

$$\frac{\delta v(s)}{\delta u(s)} = \frac{1/m}{s + \left({}^{AC_d\rho v^*}/_{m}\right)}$$

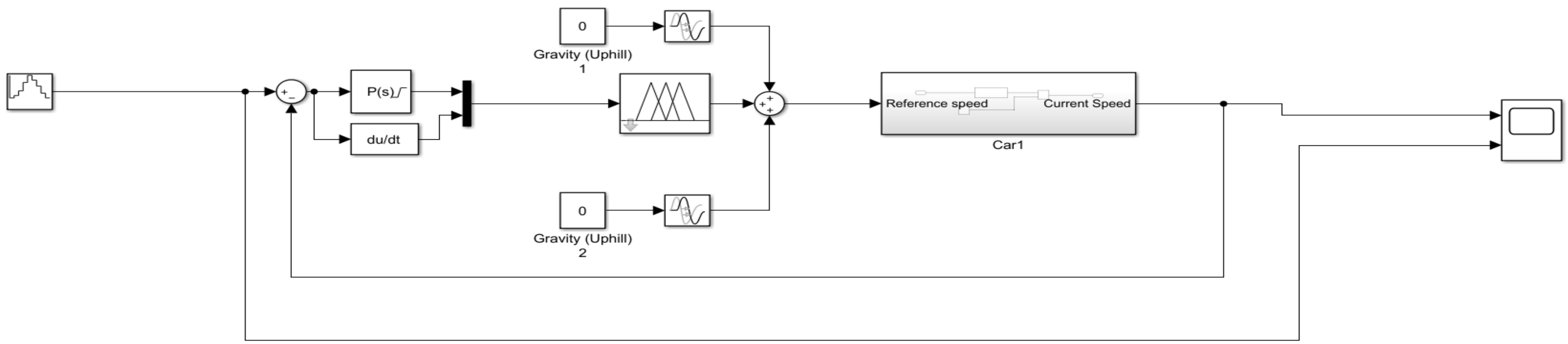
Get back to page 1

Appendix: Models and Graphs

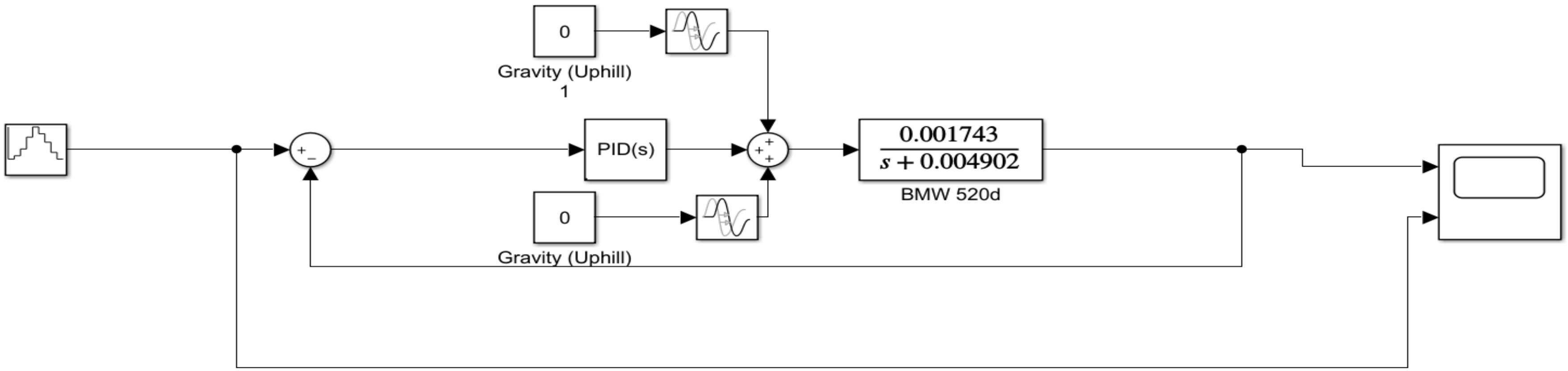
Fuzzy P cruise controller system model – Audi A4 2.0 TDI



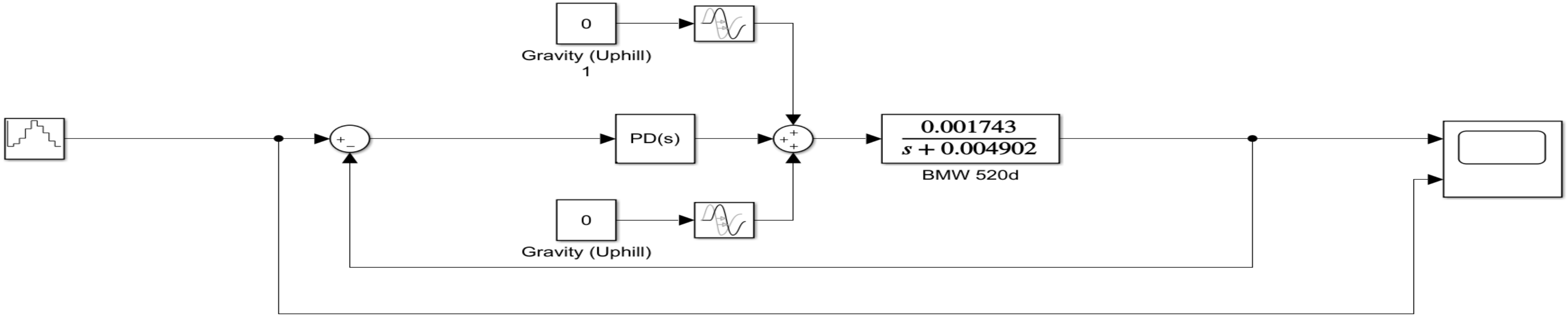
Fuzzy P cruise controller system model – Average car



PID cruise controller system model – BMW 520d



PD cruise controller system model – BMW 520d



Get back to page 4

In the following pages, the responses of the considered cars in various scenarios are shown.

Uphill scenario – the car/ model is continuously subjected to gravitational disturbance throughout the timespan.

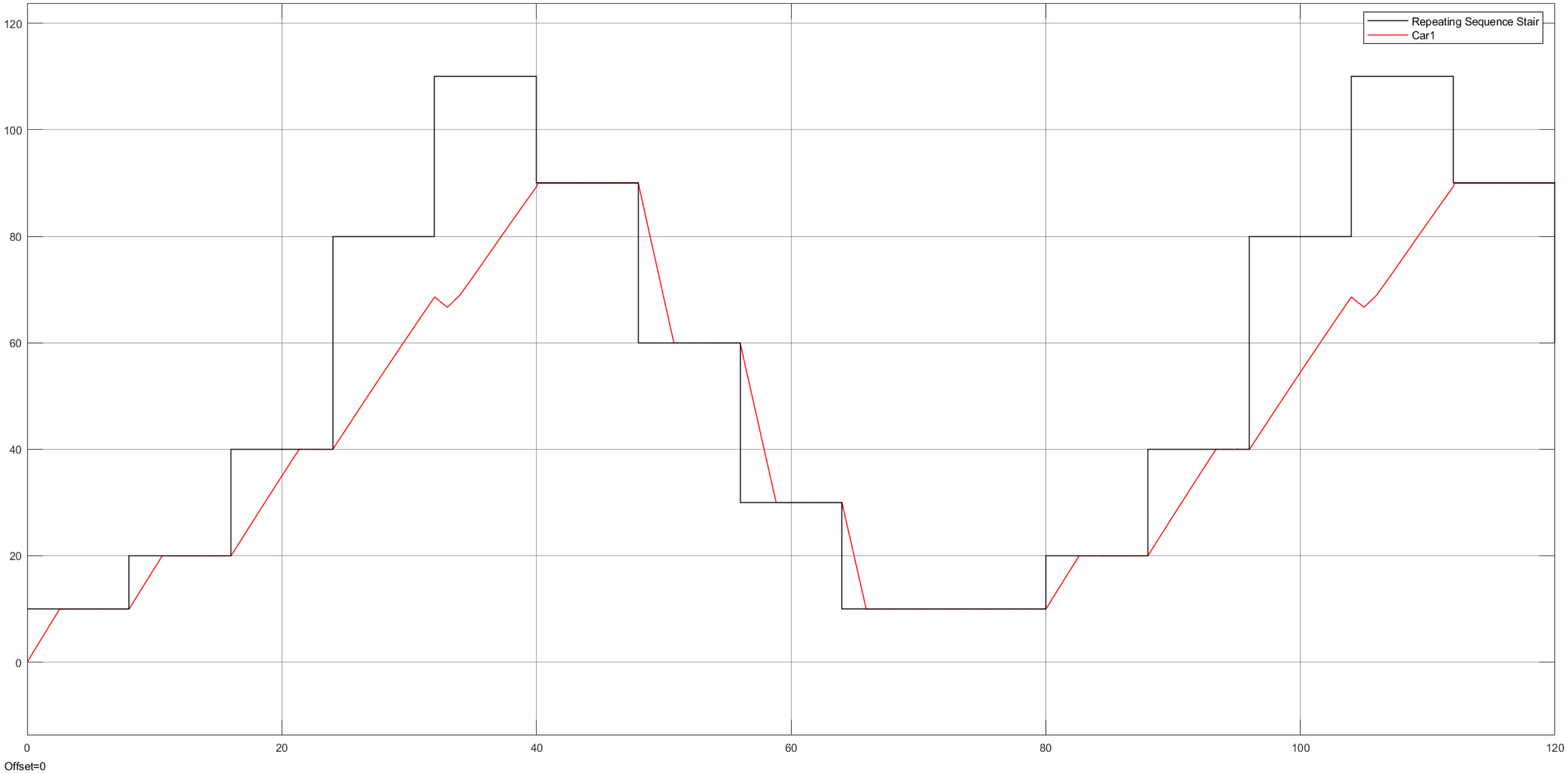
Flat road scenario – the car/ model is not subjected to any kind of disturbance (ideal case)

Mixed road scenario – the car / model is subjected to disturbances, noises and ideal situation which is achieved through delay set throughout the time span.

The car goes in ideal situation for first 24s, after that gravitational disturbance is applied to the model and later after 48s of total time span, measurement noise is added to the model through signal generator block in Simulink. It continues the same until end of sampling time.

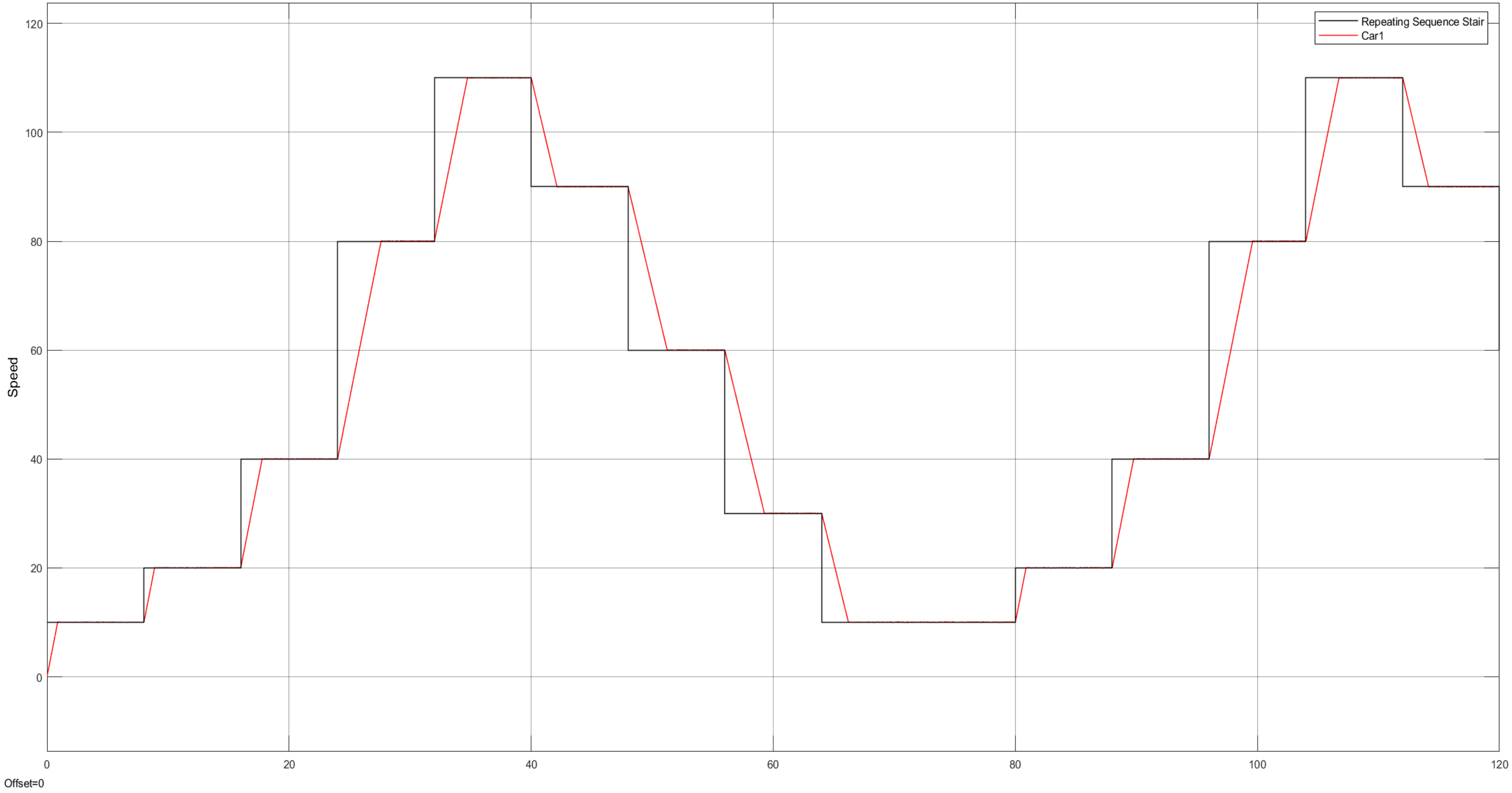
Fuzzy P Controller design for an average car

Uphill scenario

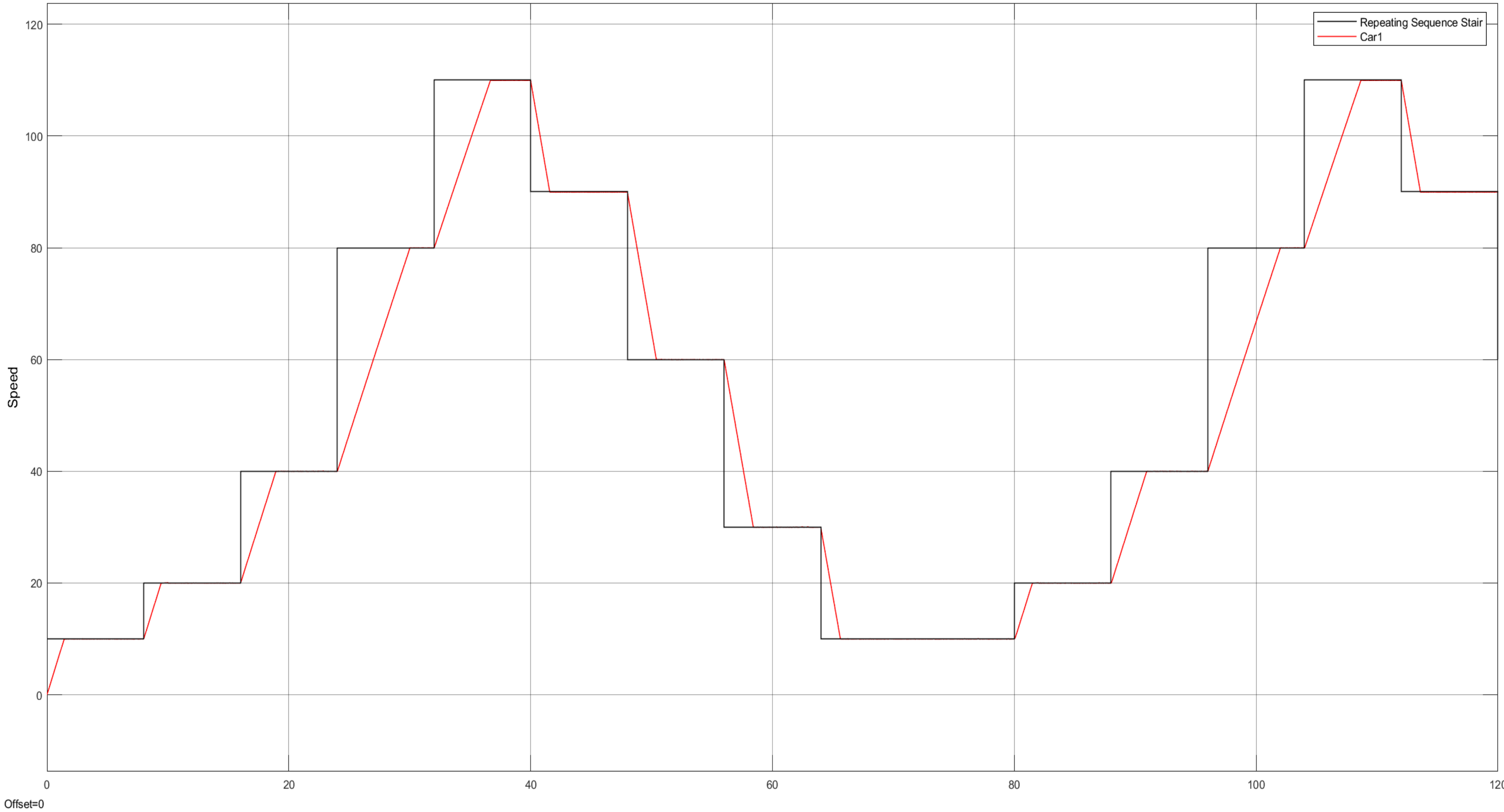


Fuzzy P Controller design for an average car - Optimized (Triangular Membership functions)

Flat road scenario

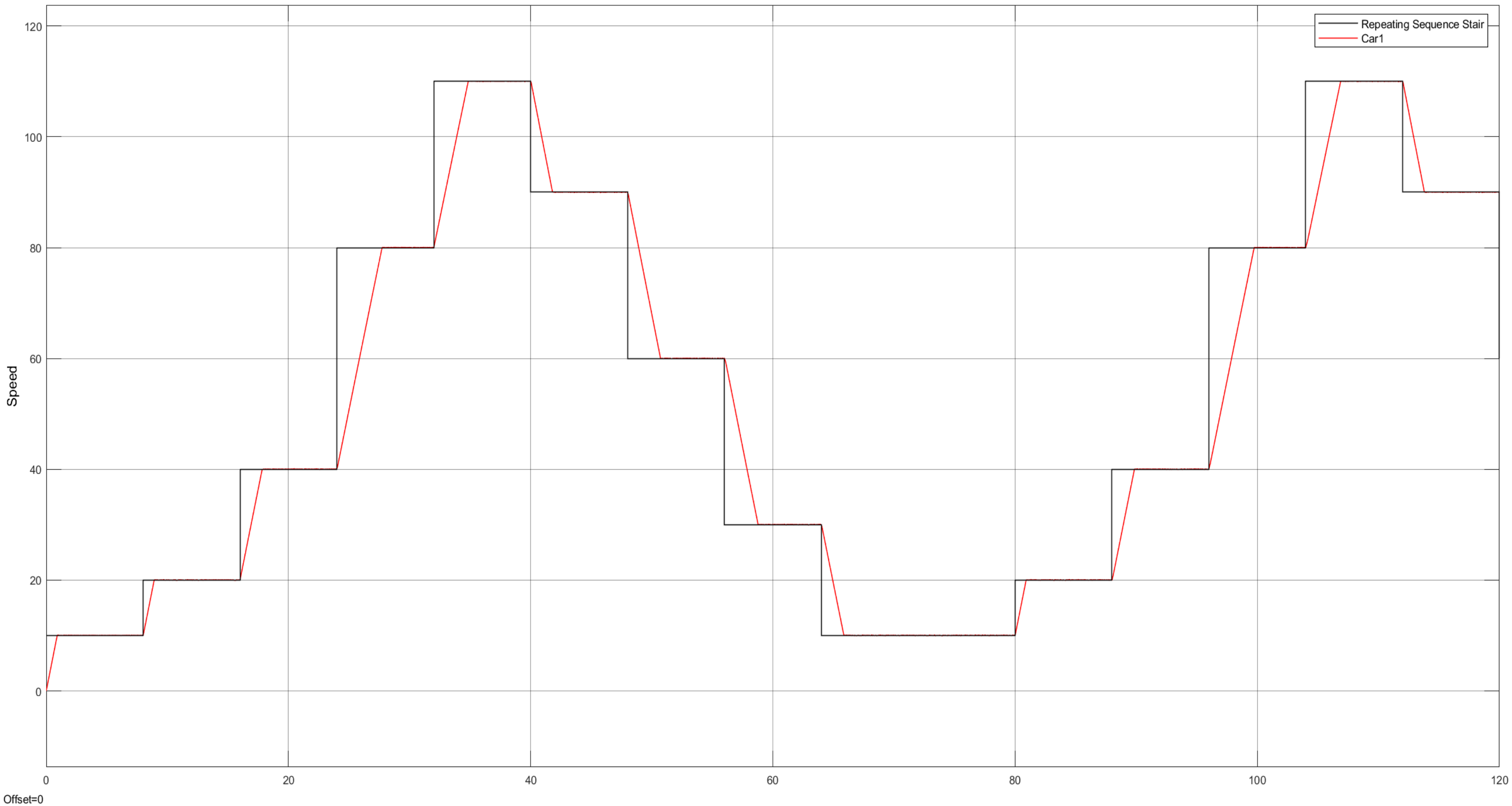


Uphill scenario

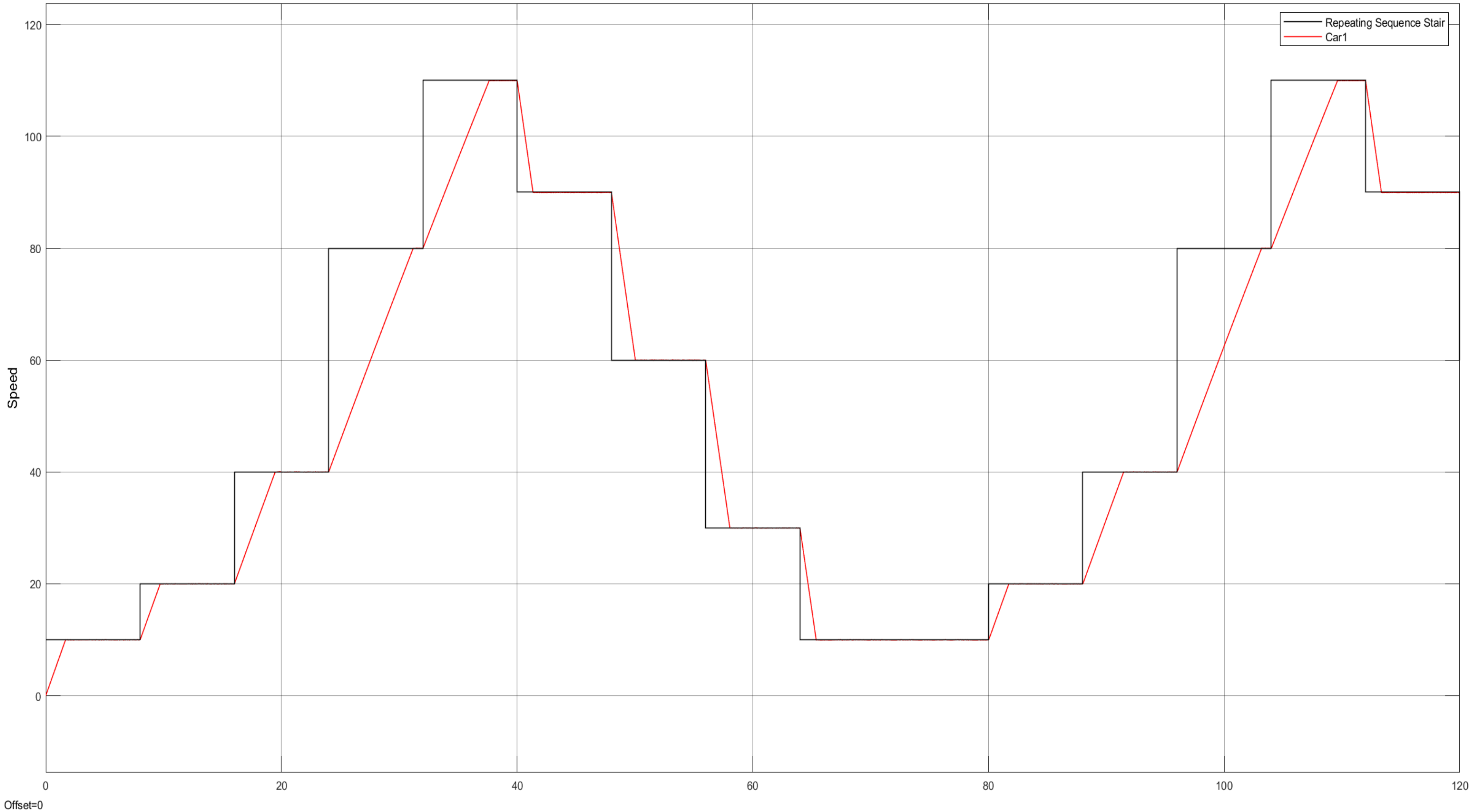


Fuzzy P Controller design for an average car (Gaussian Membership functions)

Flat road scenario

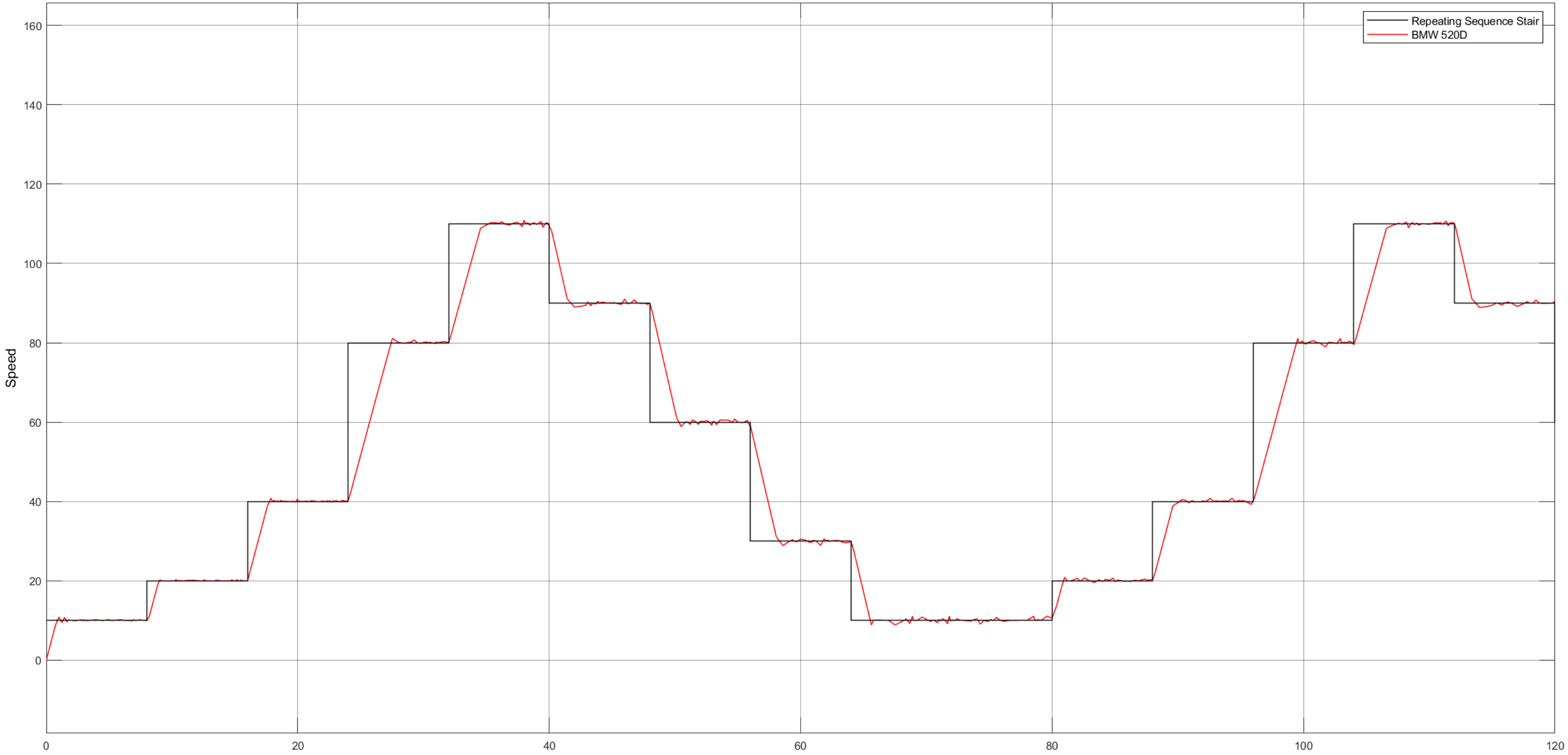


Uphill scenario

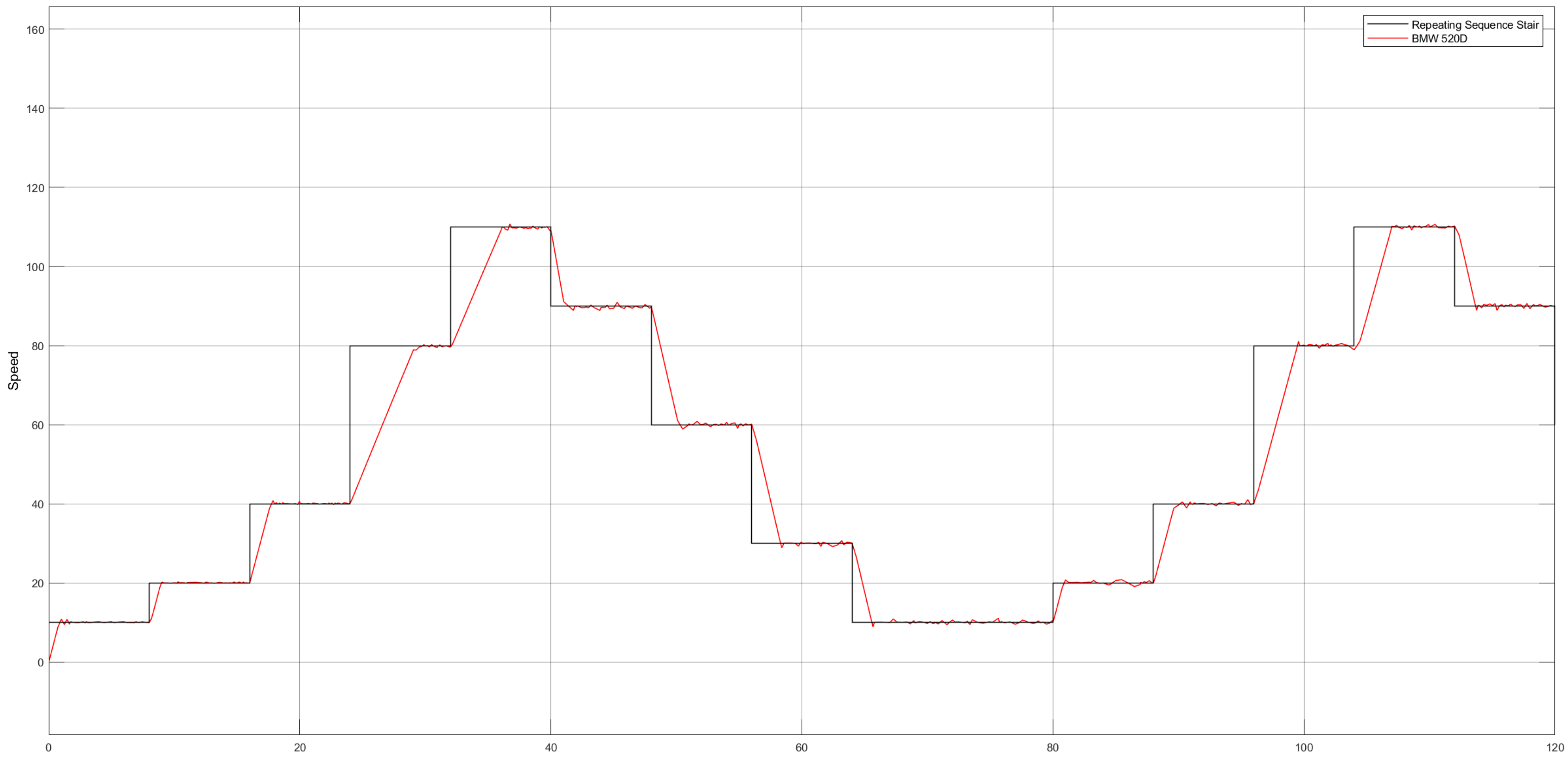


Fuzzy P Controller design for BMW 520D (Gaussian Membership functions)

Flat road scenario

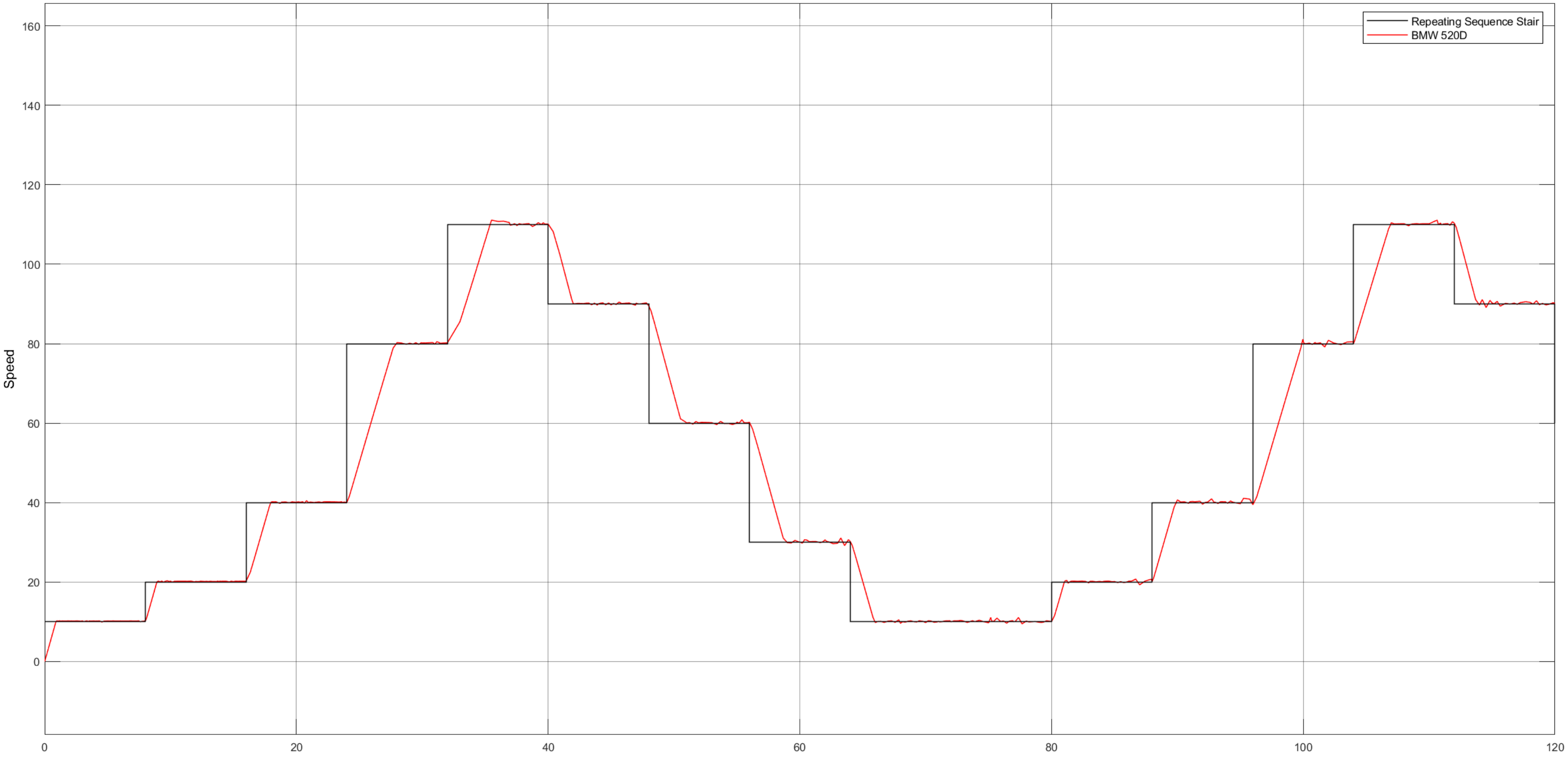


Uphill scenario

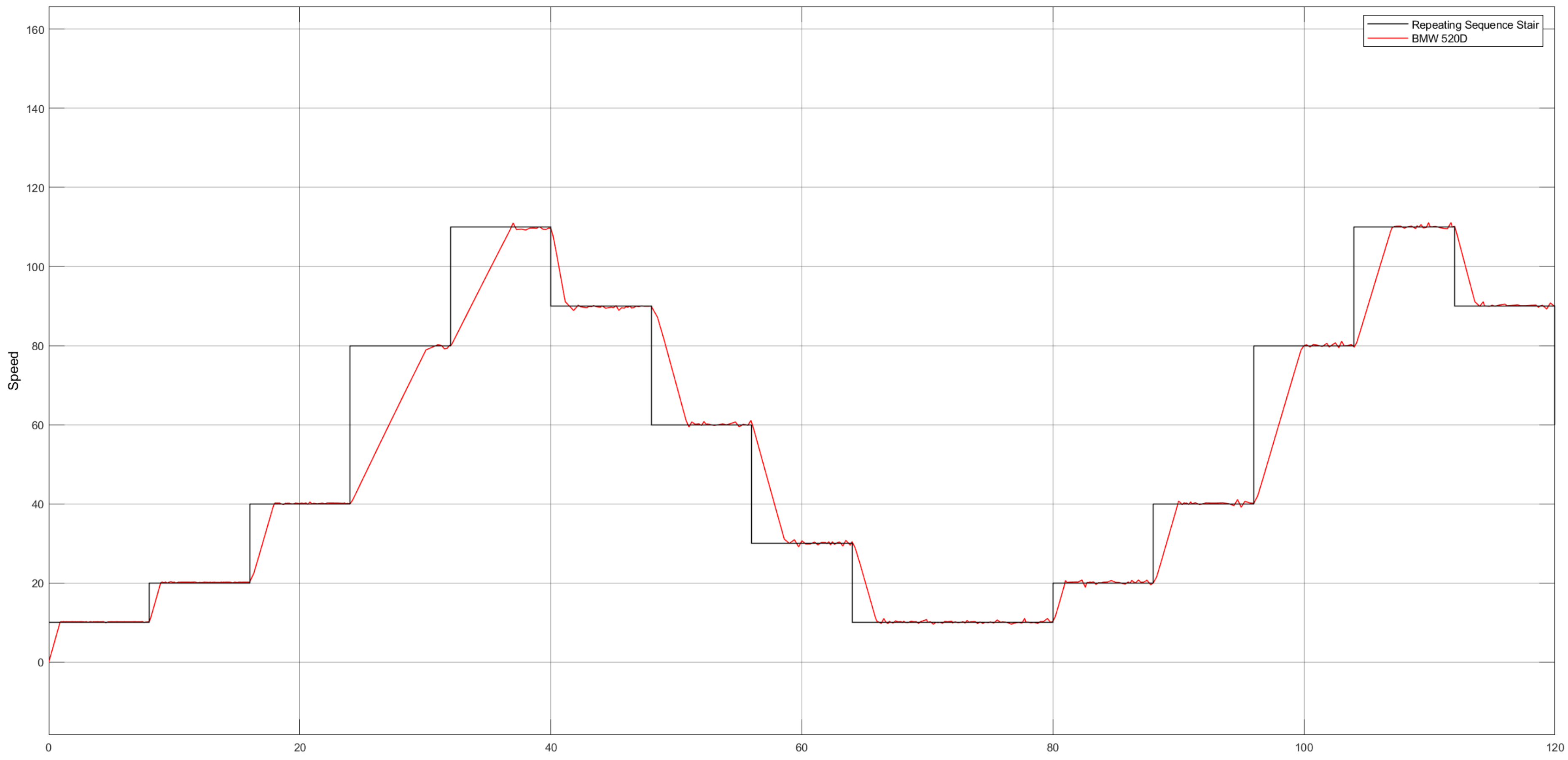


Fuzzy P Controller design for BMW 520D (Triangular Membership functions)

Flat road scenario



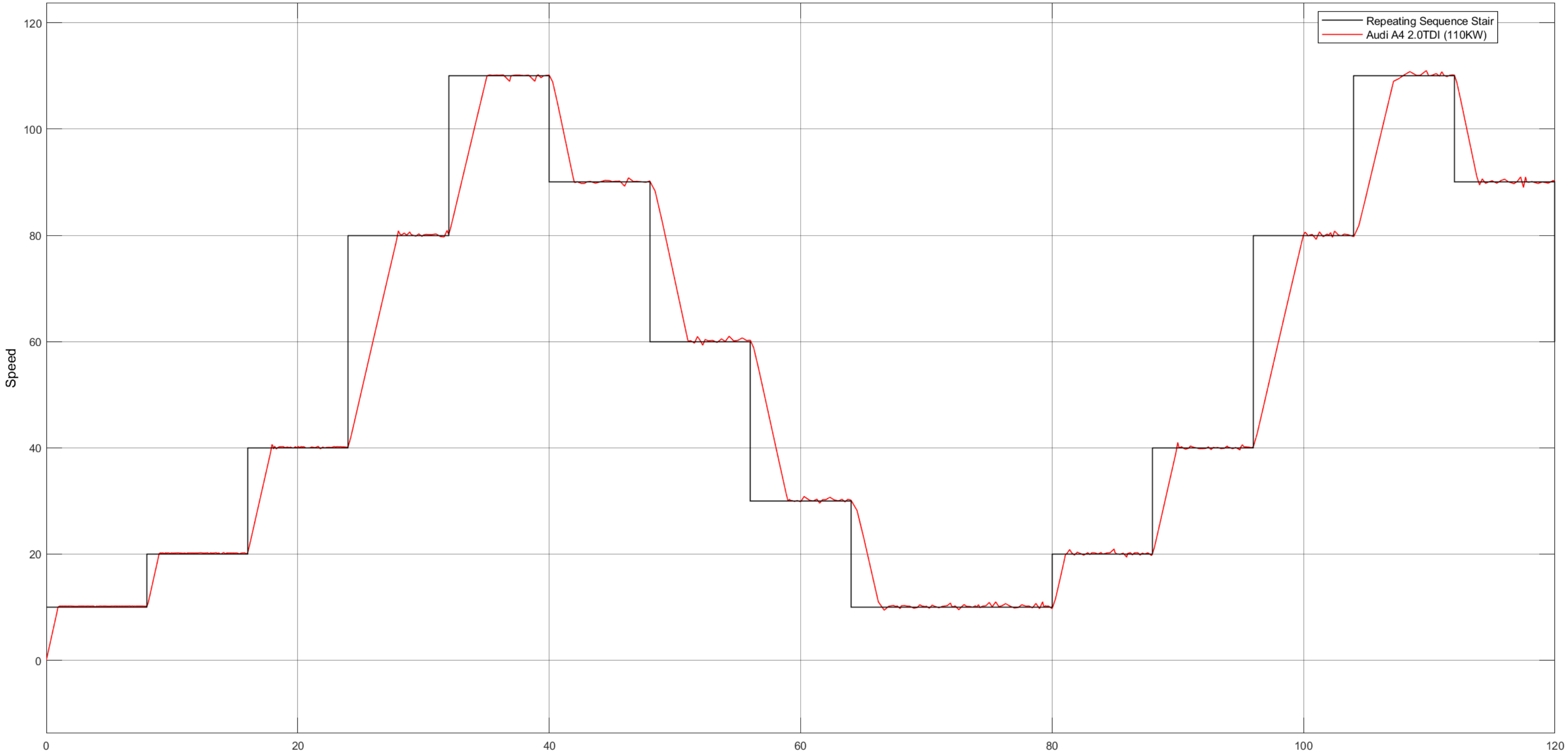
Mixed road scenario



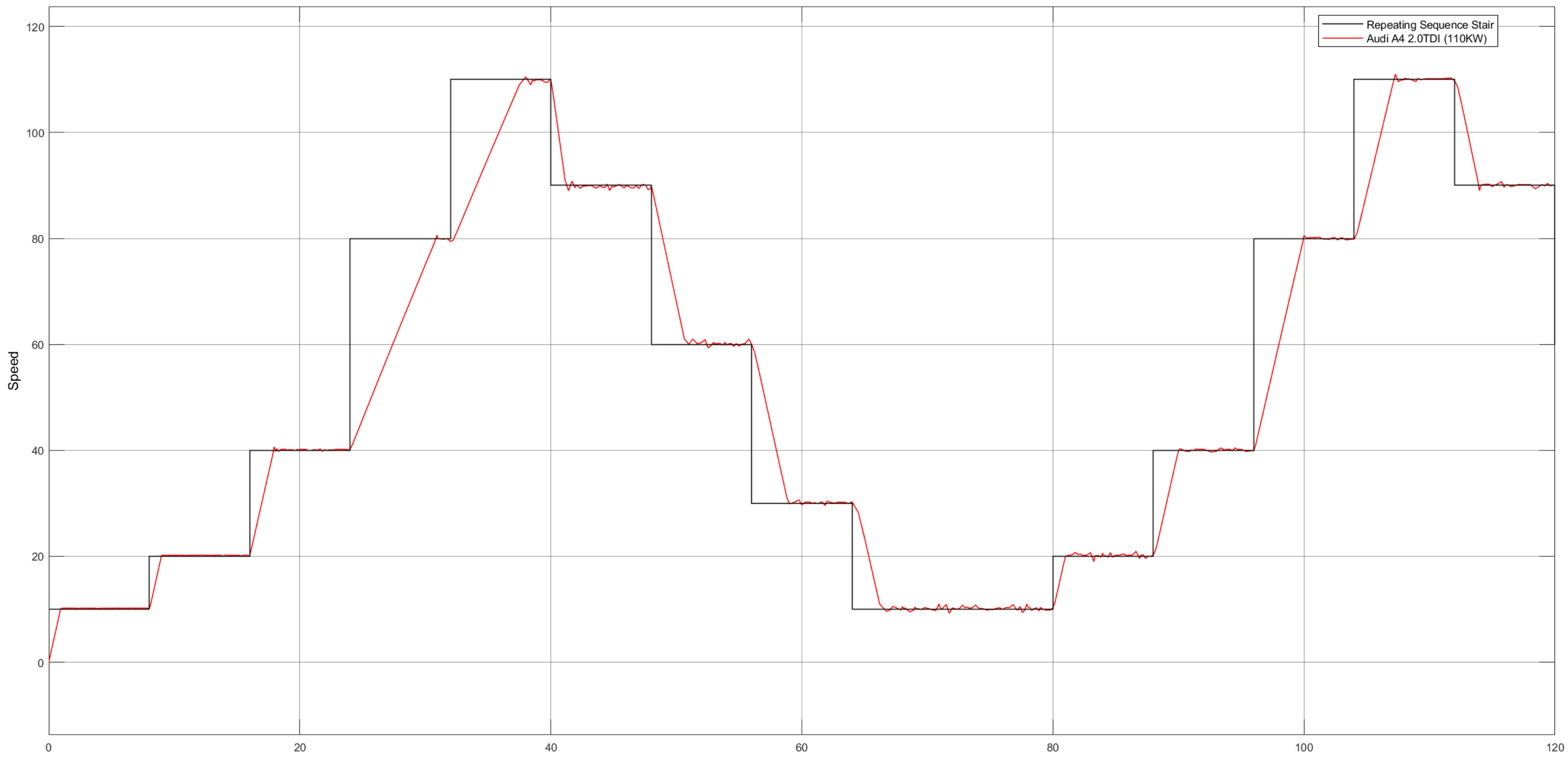
Get back to page 5

Fuzzy P Controller design for Audi A4 2.0 TDI (Triangular Membership functions)

Flat road scenario

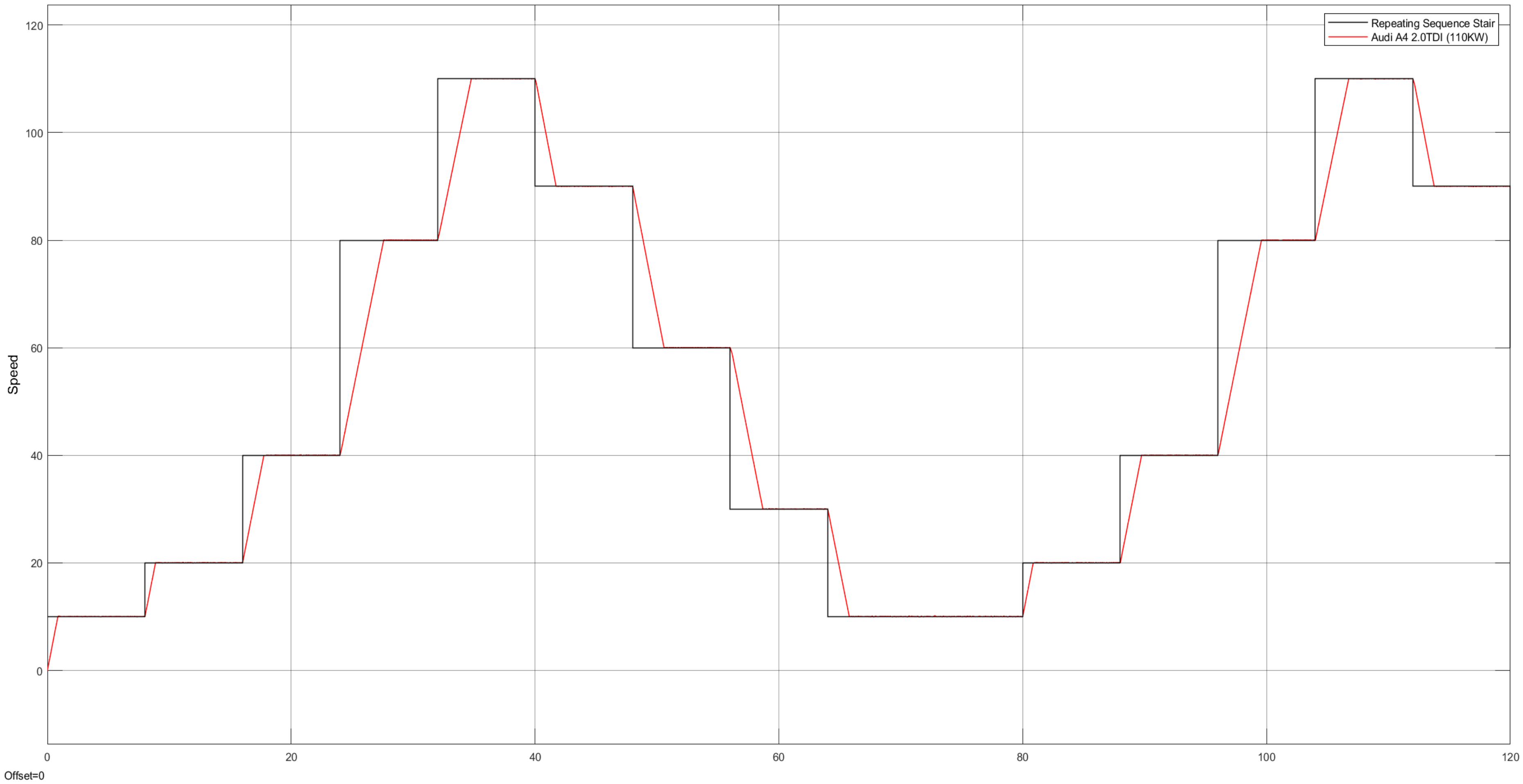


Mixed road scenario

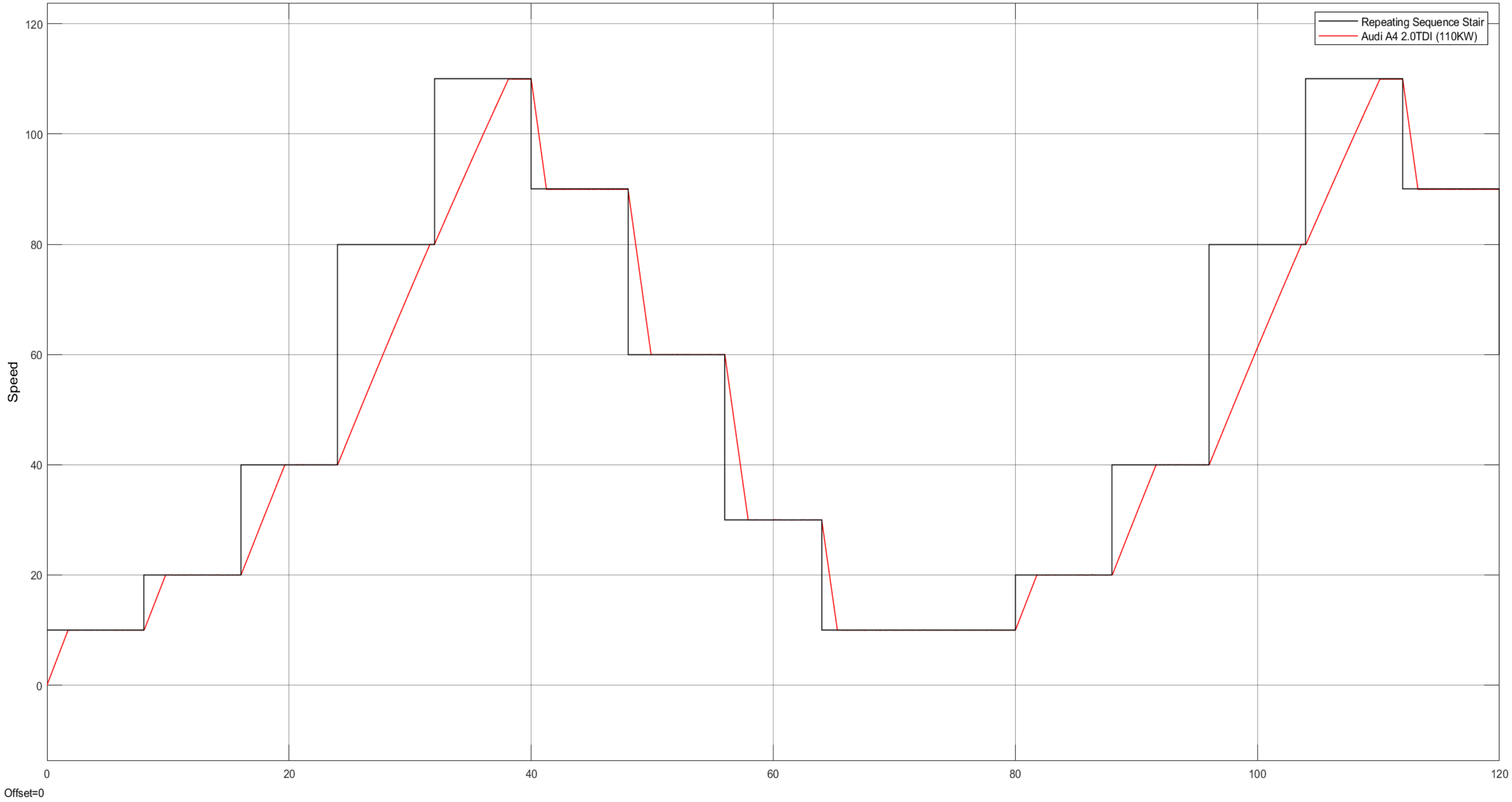


Fuzzy P Controller design for Audi A4 2.0 TDI (Gaussian Membership functions)

Flat road scenario



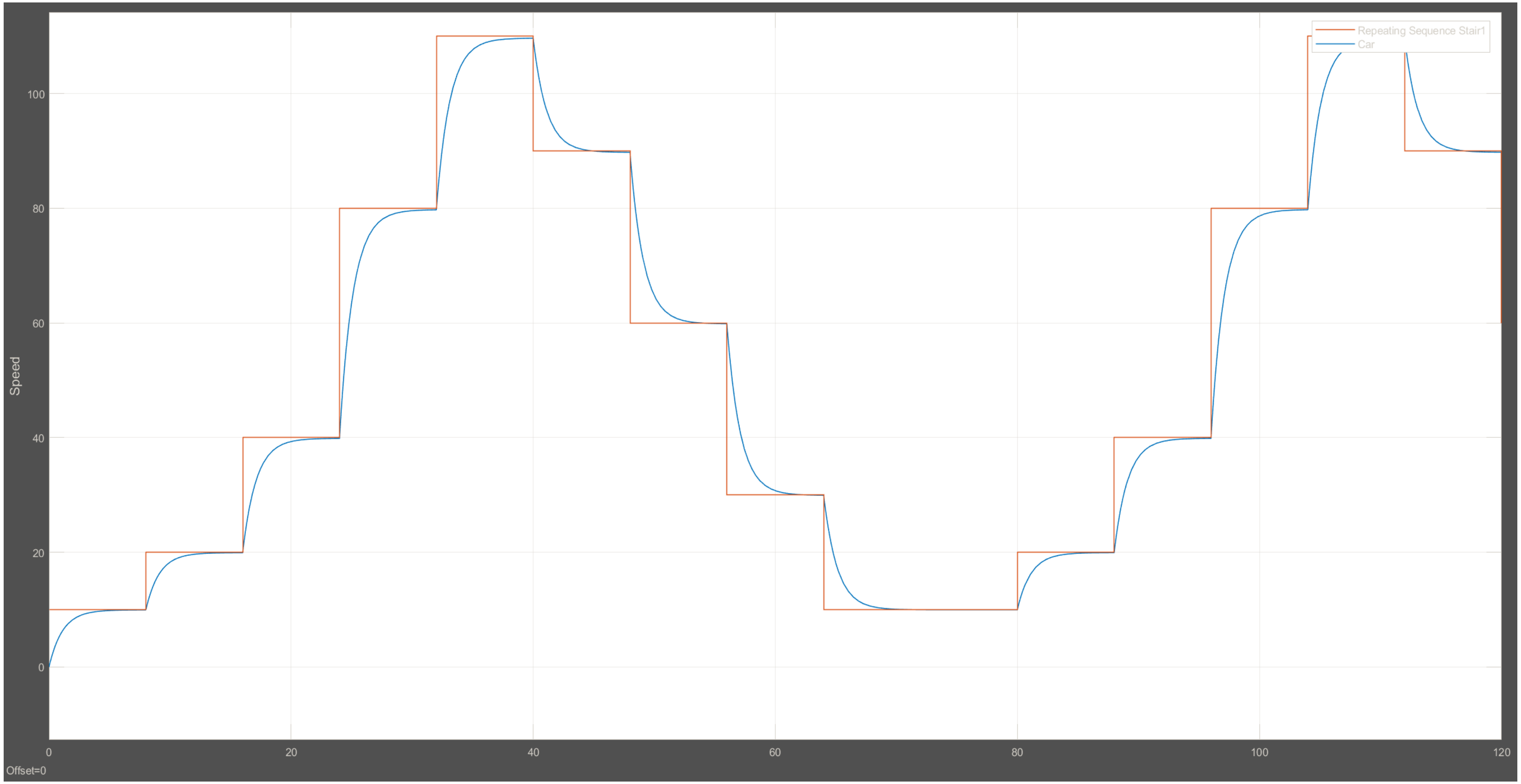
Uphill scenario



PID Controller in average car

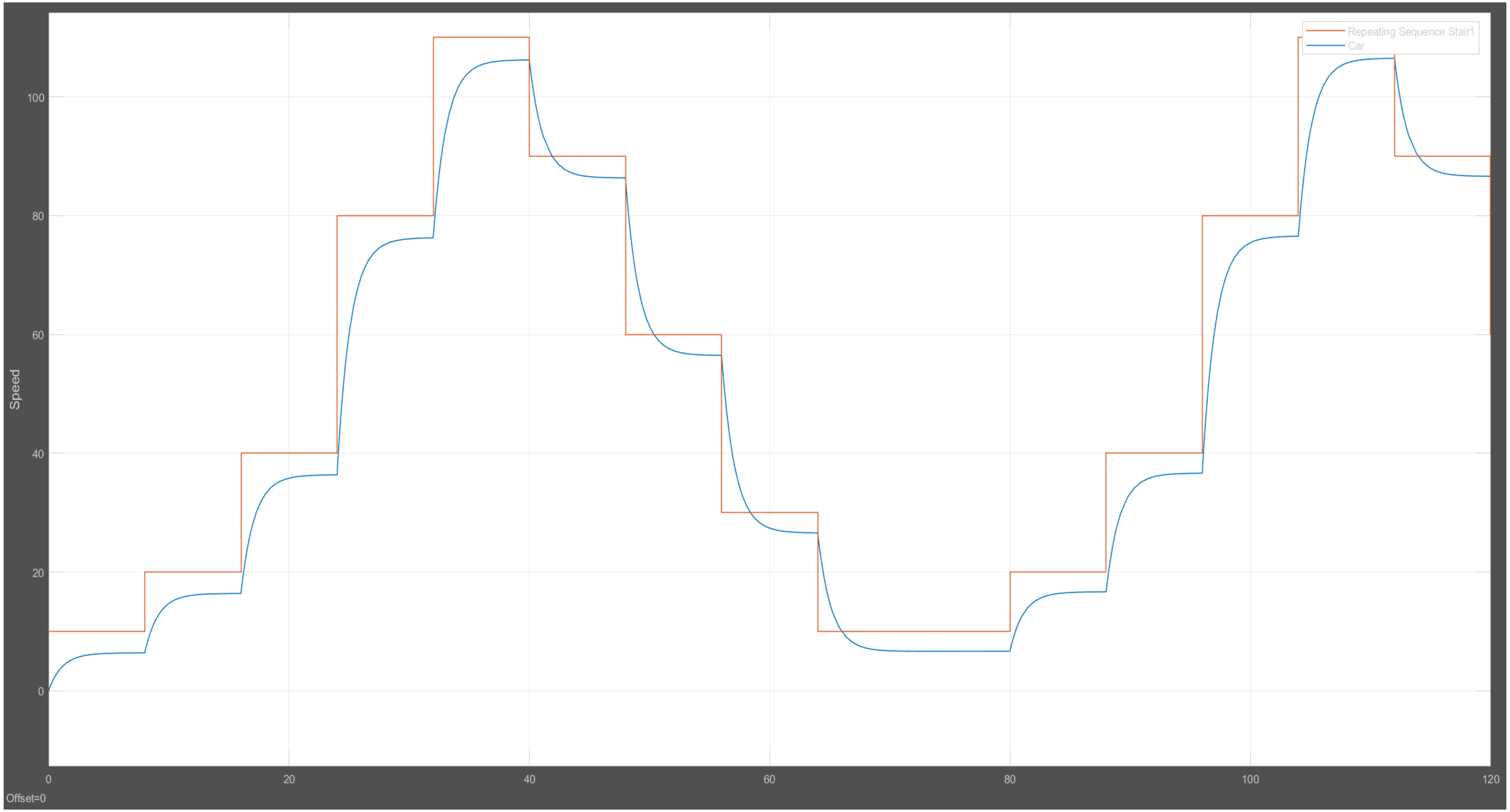
Flat road scenario

[Kp, Ki, Kd] = [900, 1, 5]

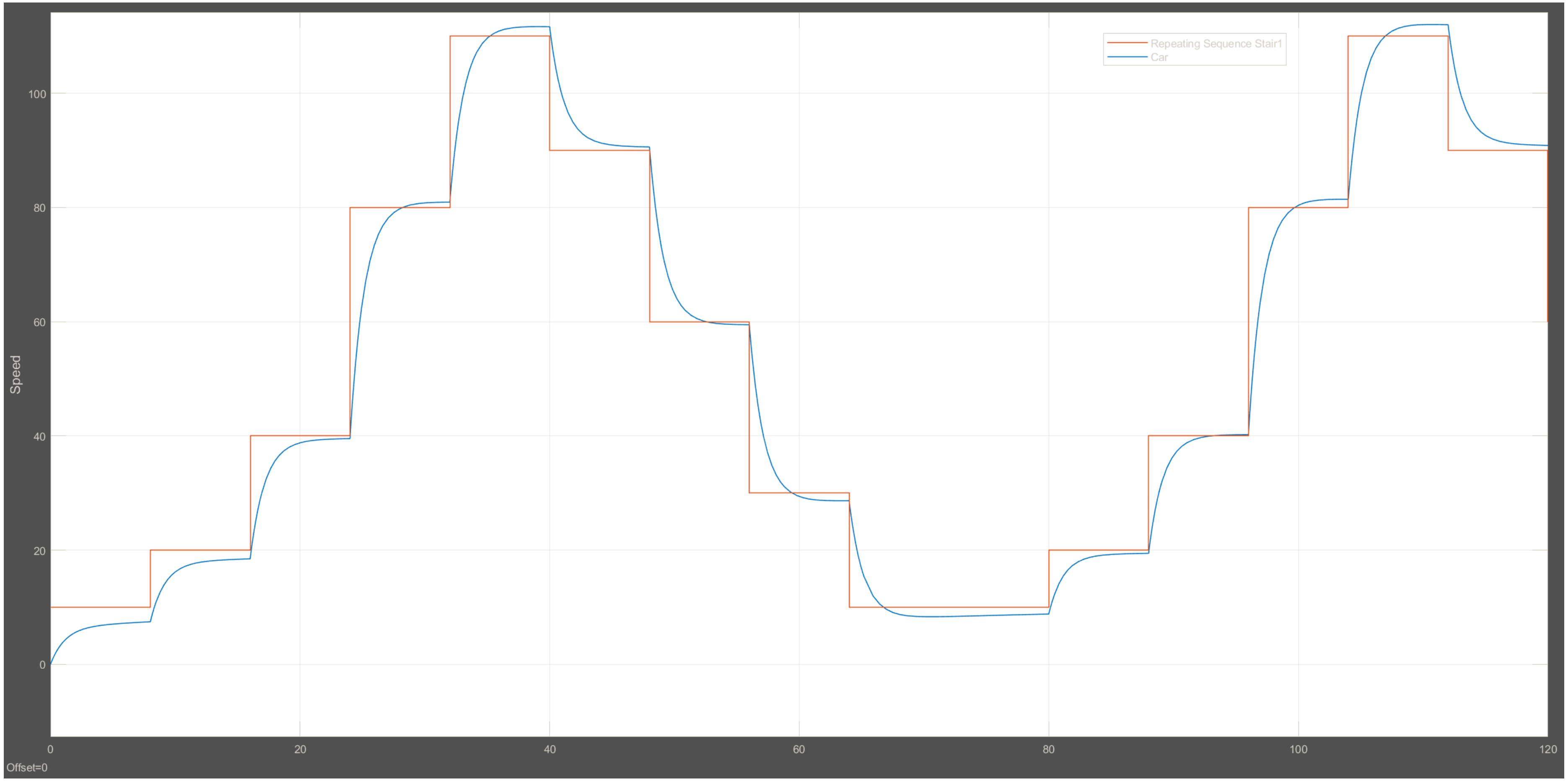


Uphill Scenario

[Kp, Ki, Kd] = [900, 1, 5]



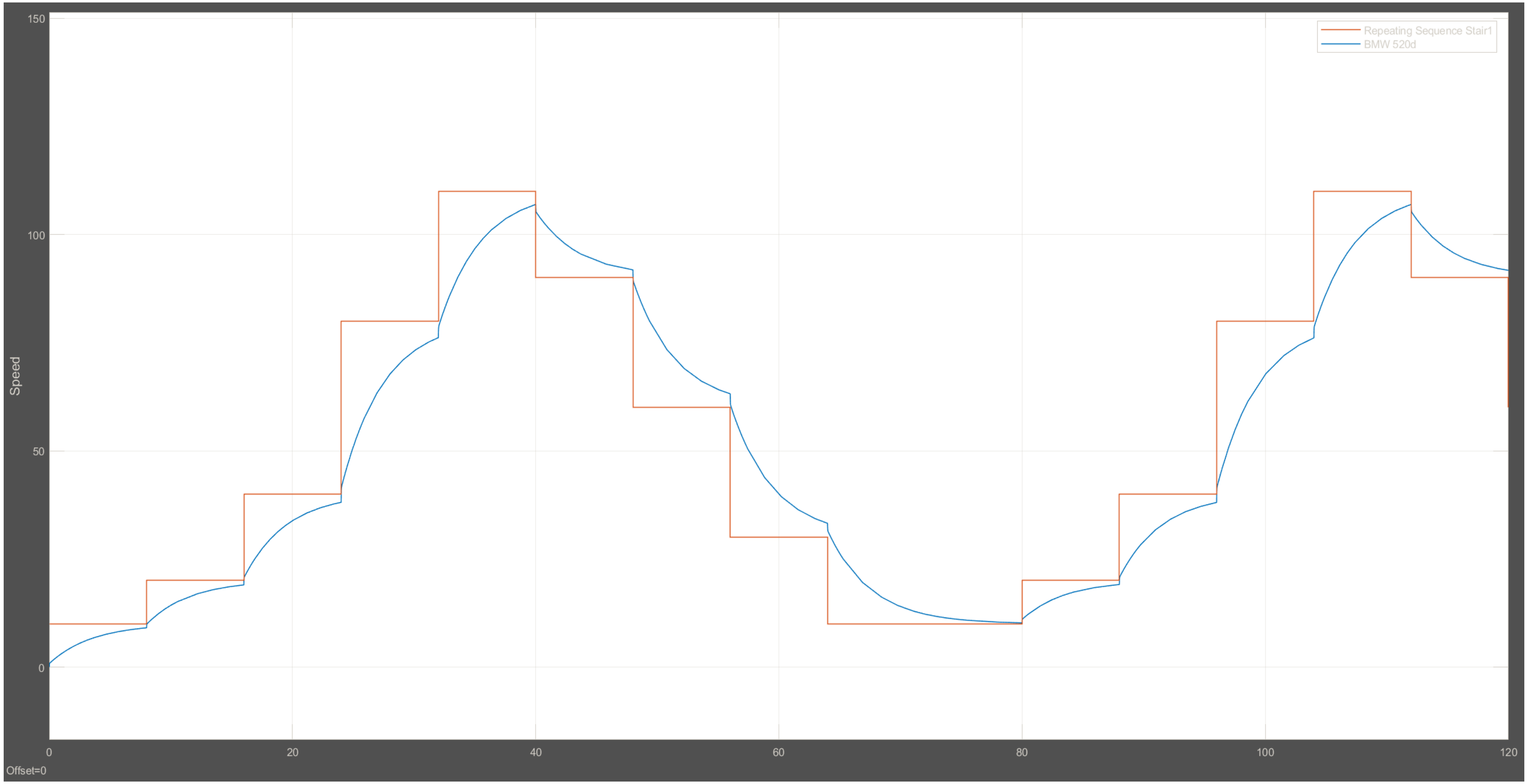
[Kp, Ki, Kd] = [900, 35, 5]



PID Controller design for BMW 520D

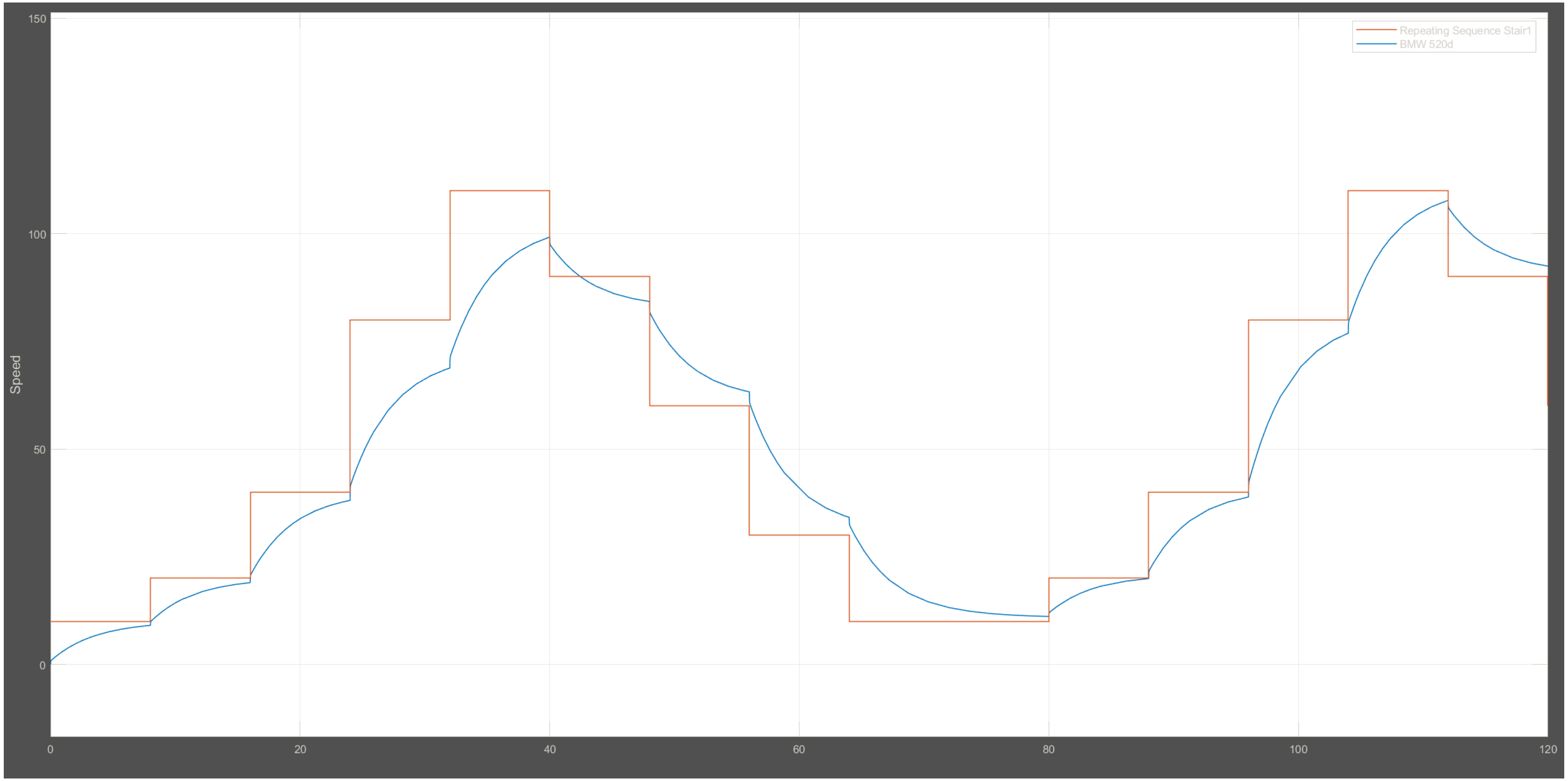
Flat road scenario

[Kp, Ki, Kd] = [178, 1, 50]



Uphill Scenario

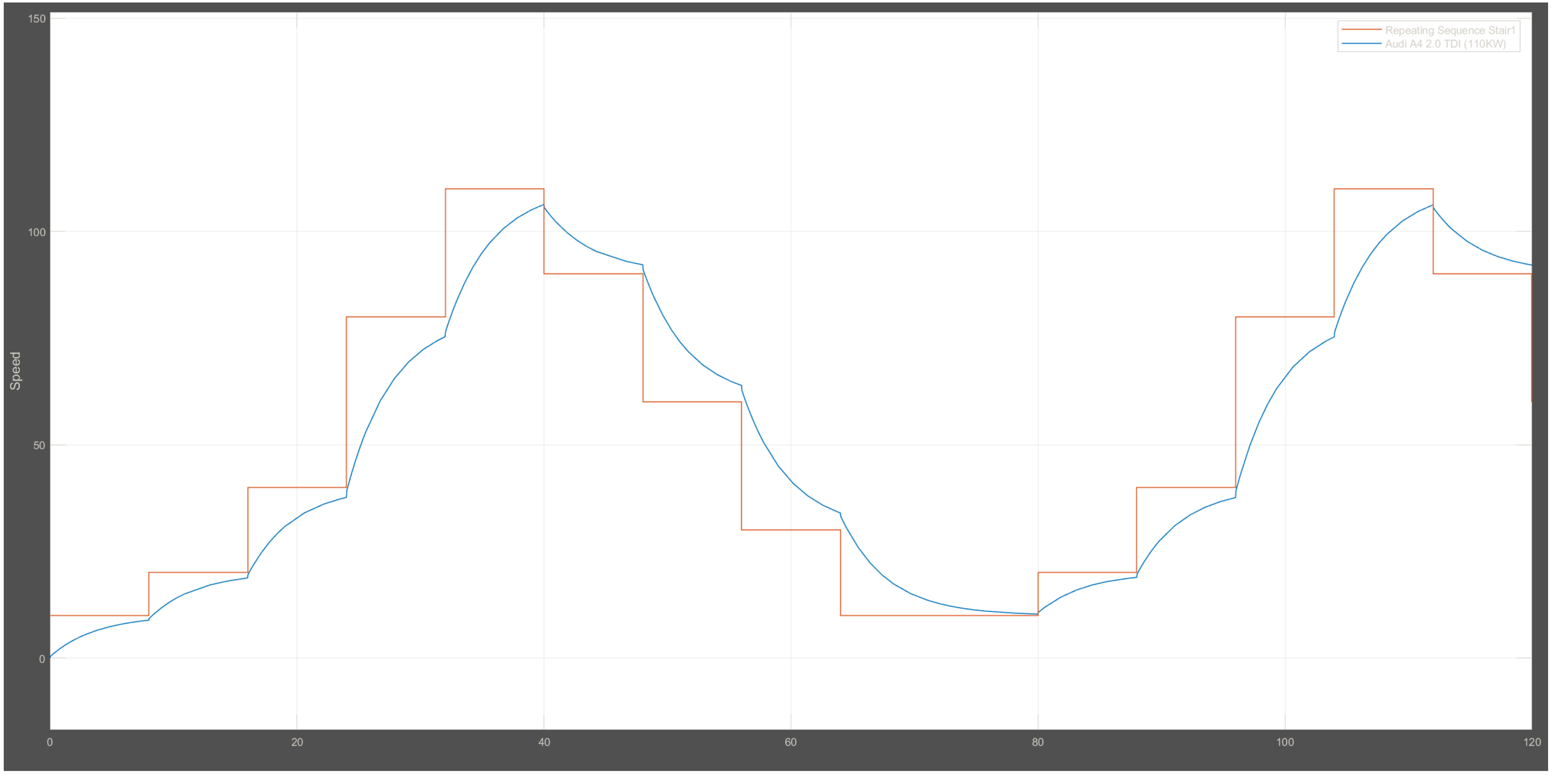
[Kp, Ki, Kd] = [178, 1, 50]



PID Controller design for Audi A4 2.0 TDI

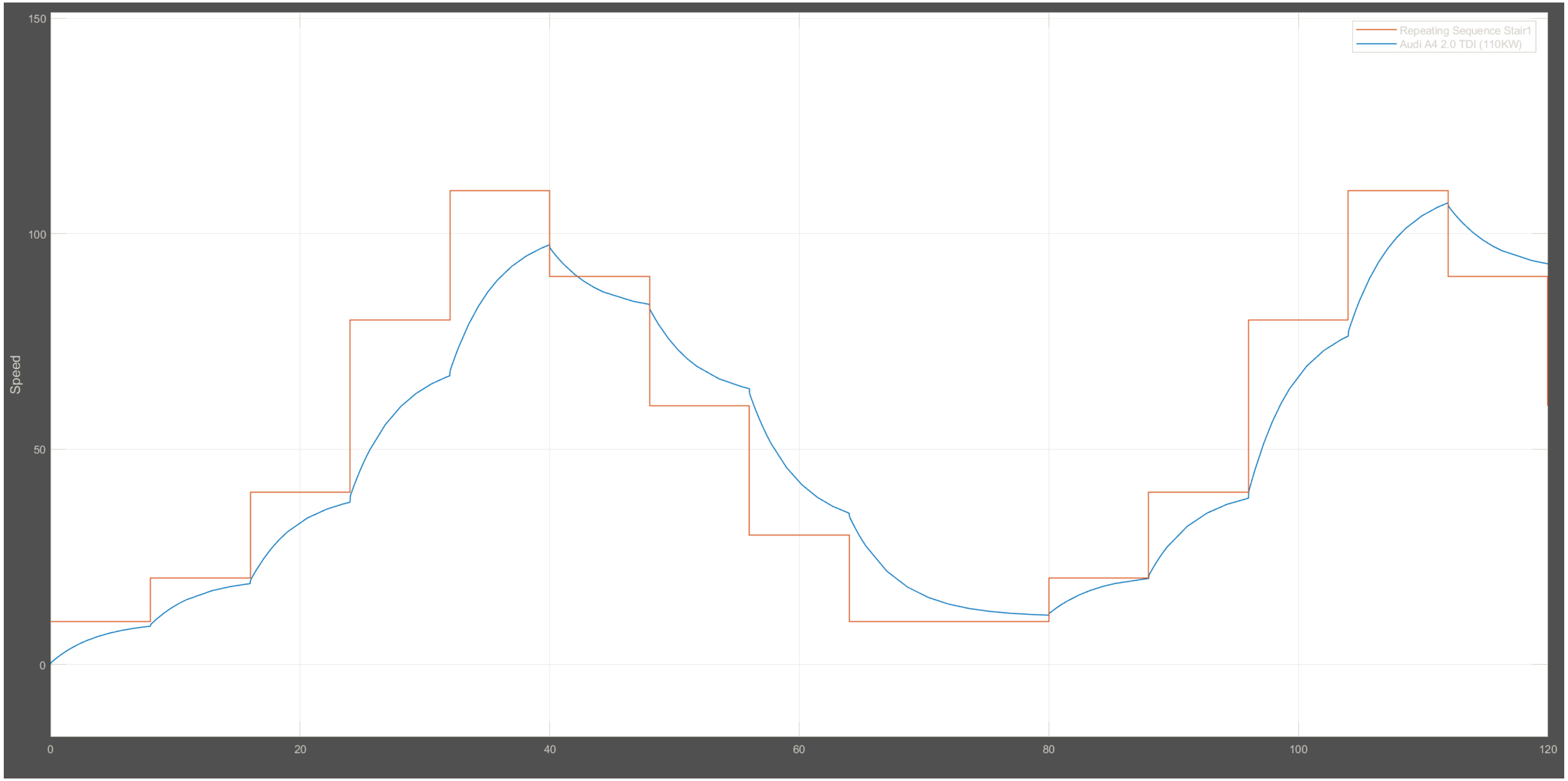
Flat road scenario

[Kp, Ki, Kd] = [147, 1, 5]



Mixed road Scenario

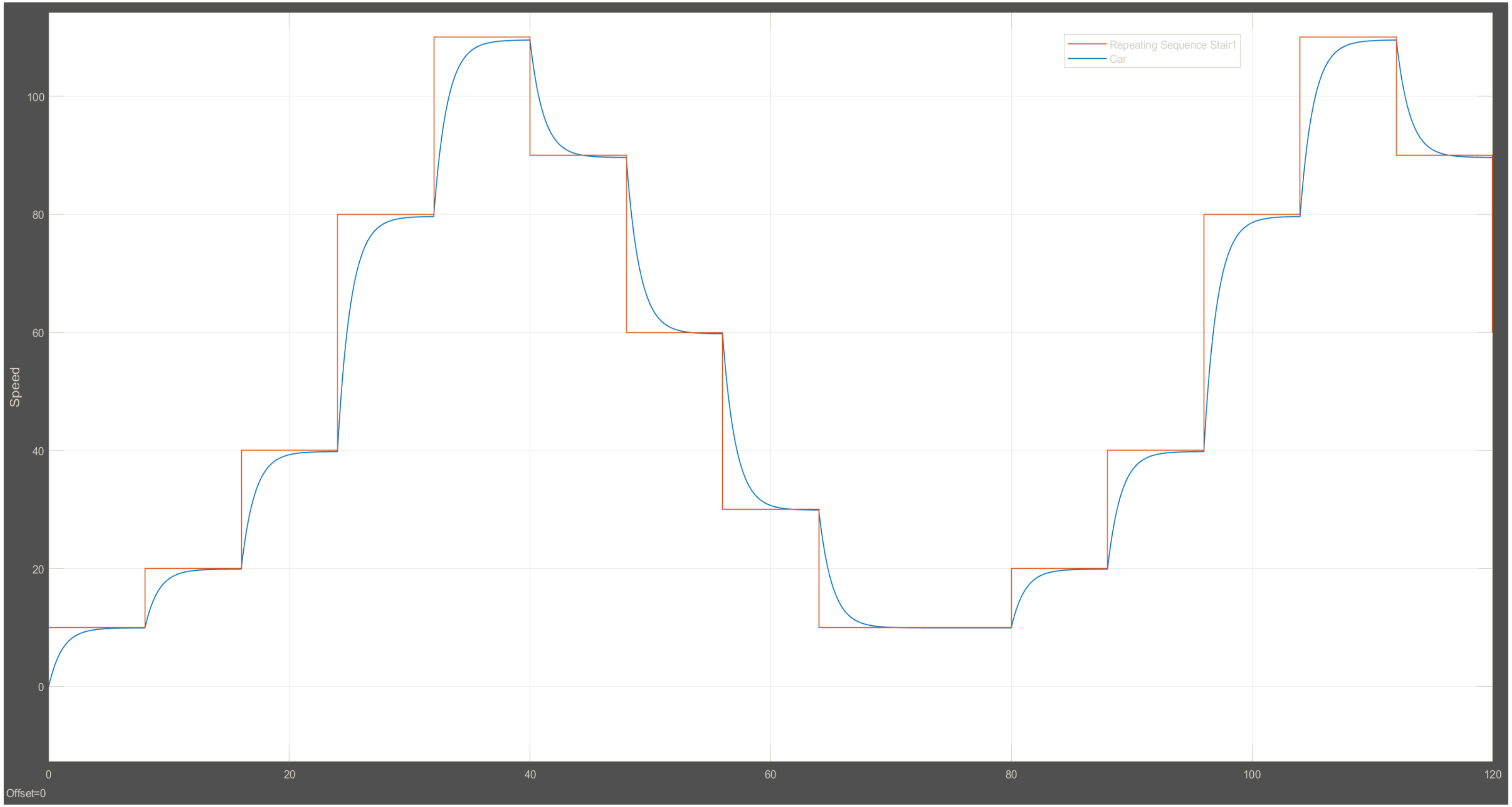
[Kp, Ki, Kd] = [147, 1, 5]



[Get back to page 3](#)

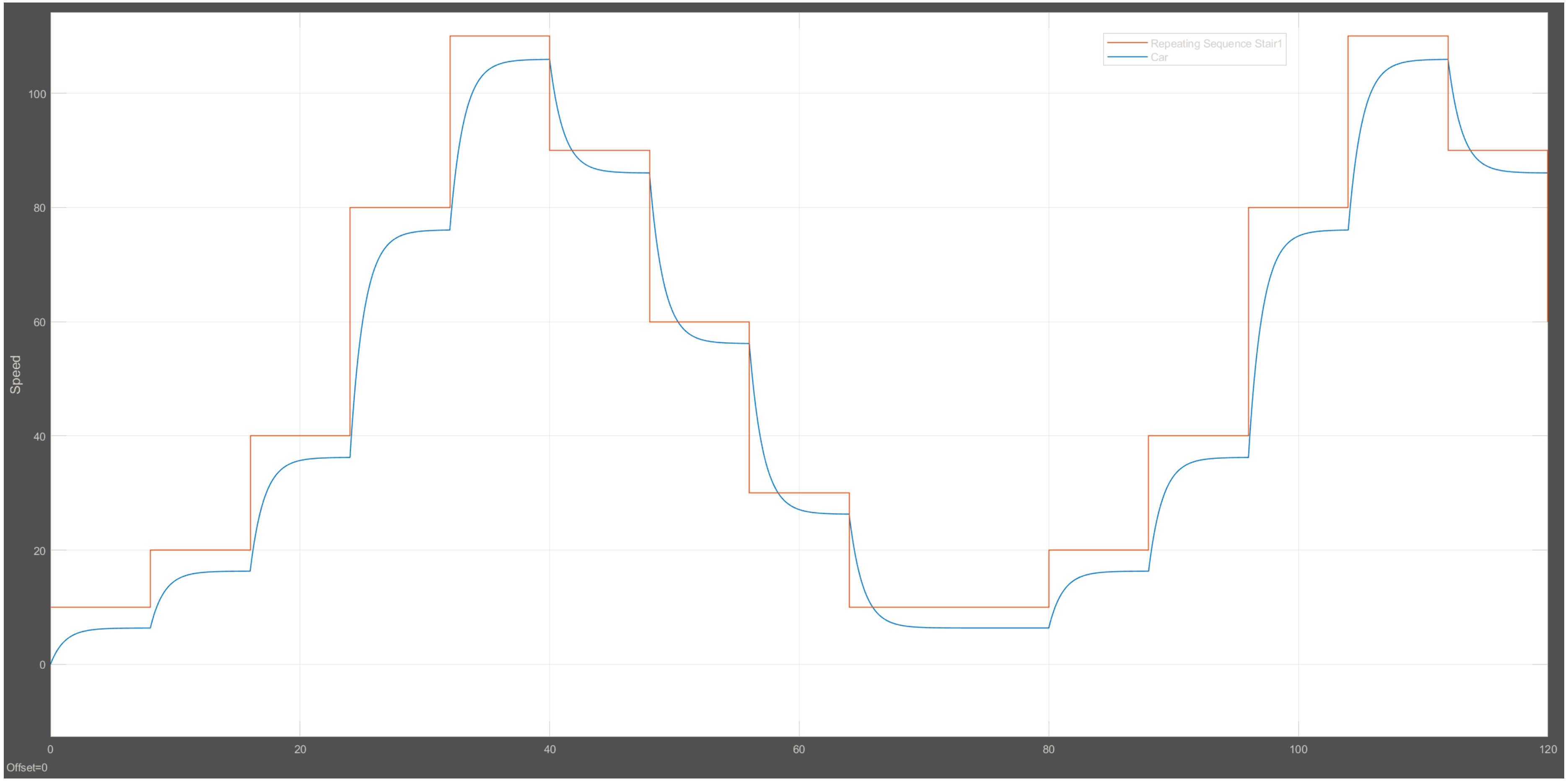
PD Controller design for an average car

Flat road scenario



Uphill scenario

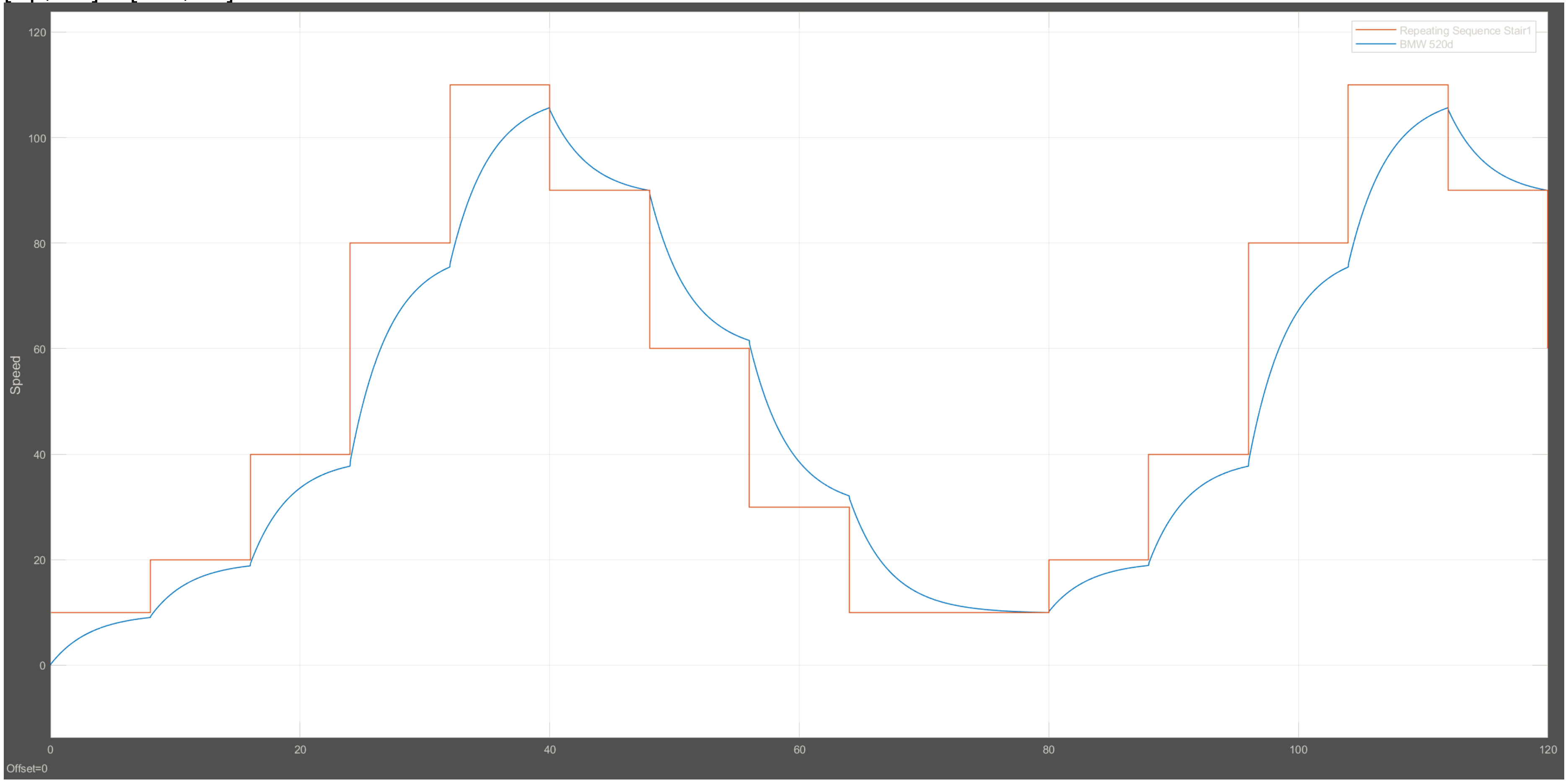
[Kp, Kd] = [900, 5]



PD Controller design for BMW 520d

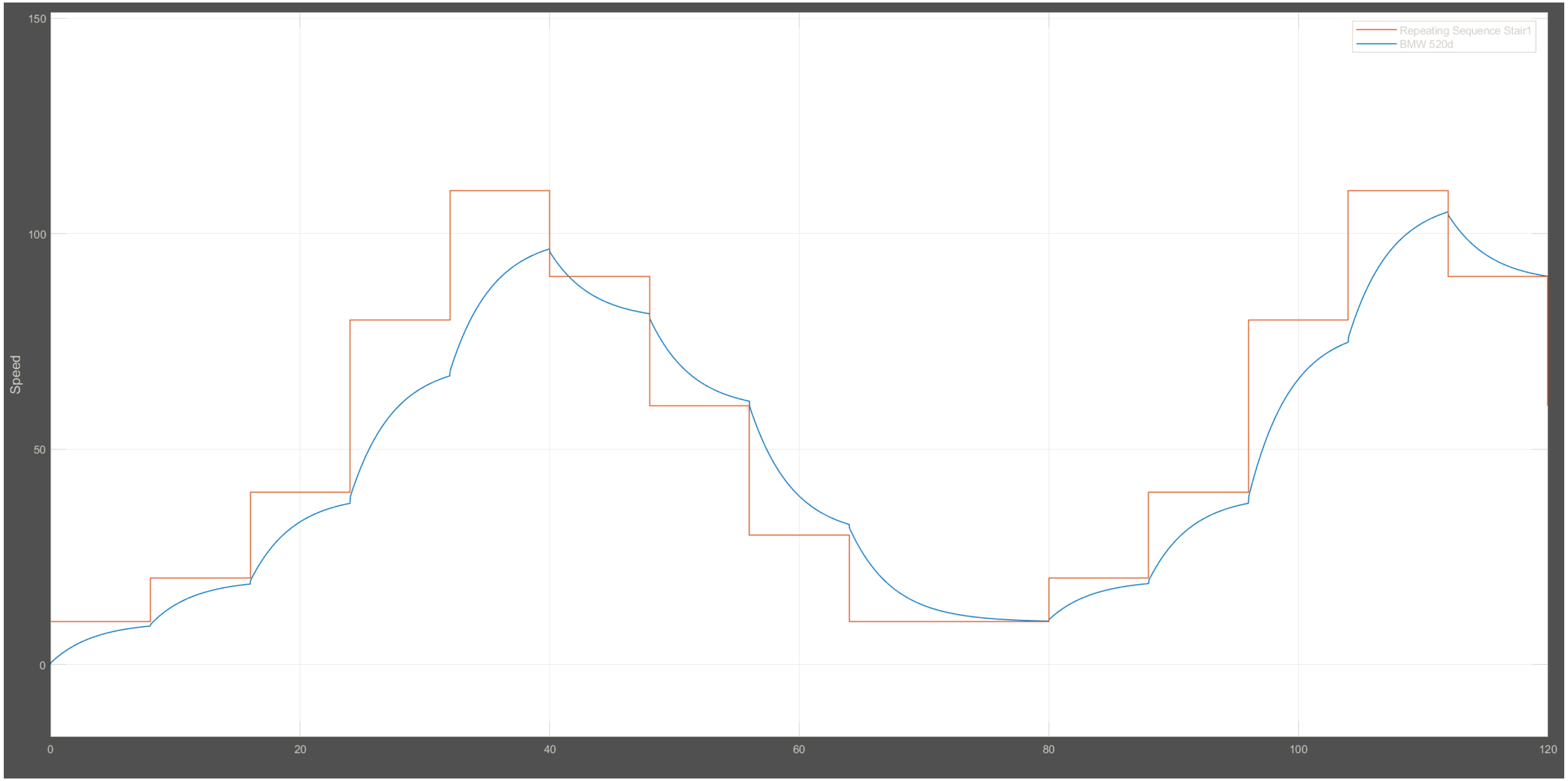
Flat road scenario

[Kp, Kd] = [180, 10]



Mixed road scenario

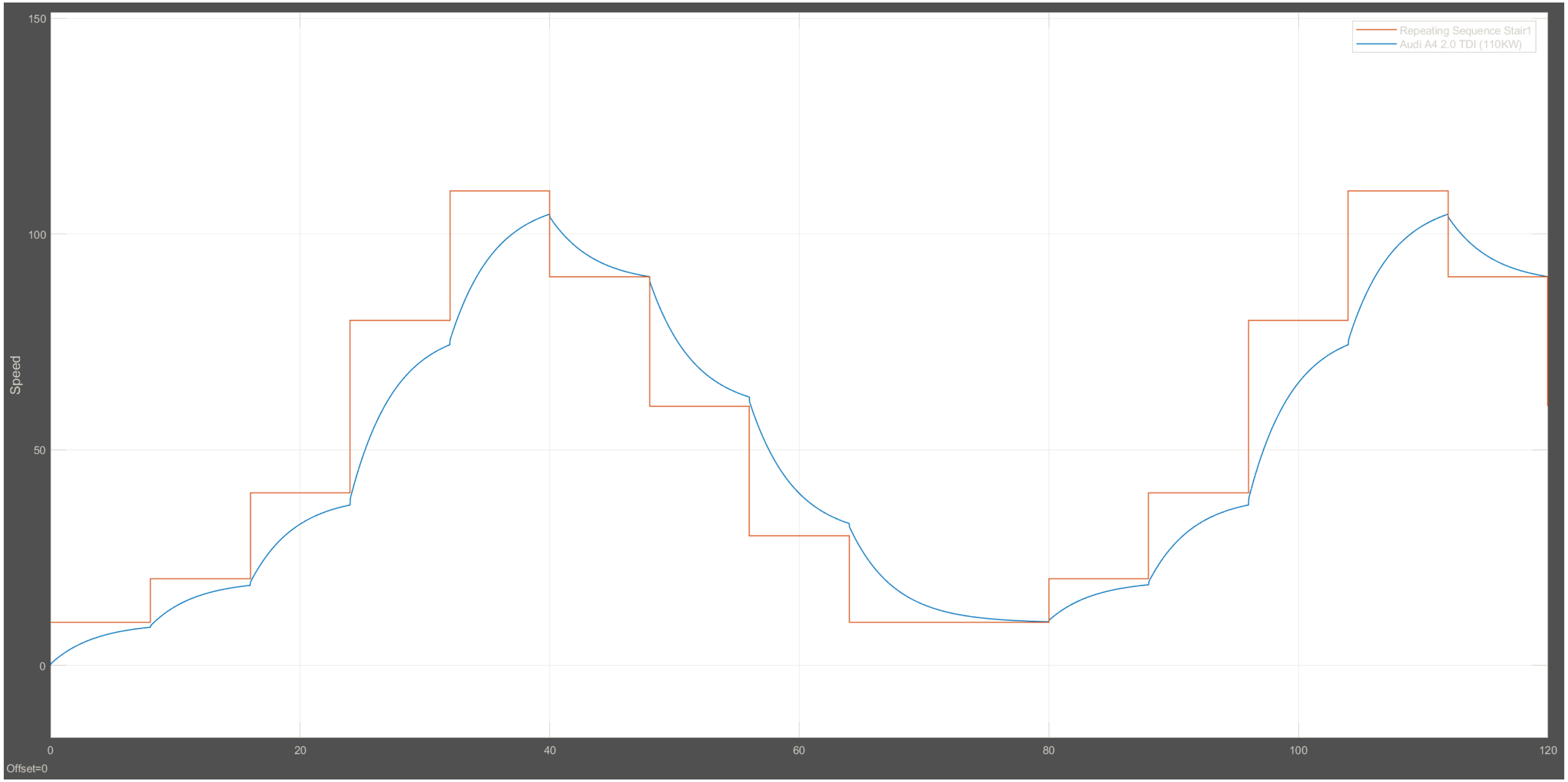
[Kp, Kd] = [180, 10]



PD Controller design for Audi A4 2.0 TDI

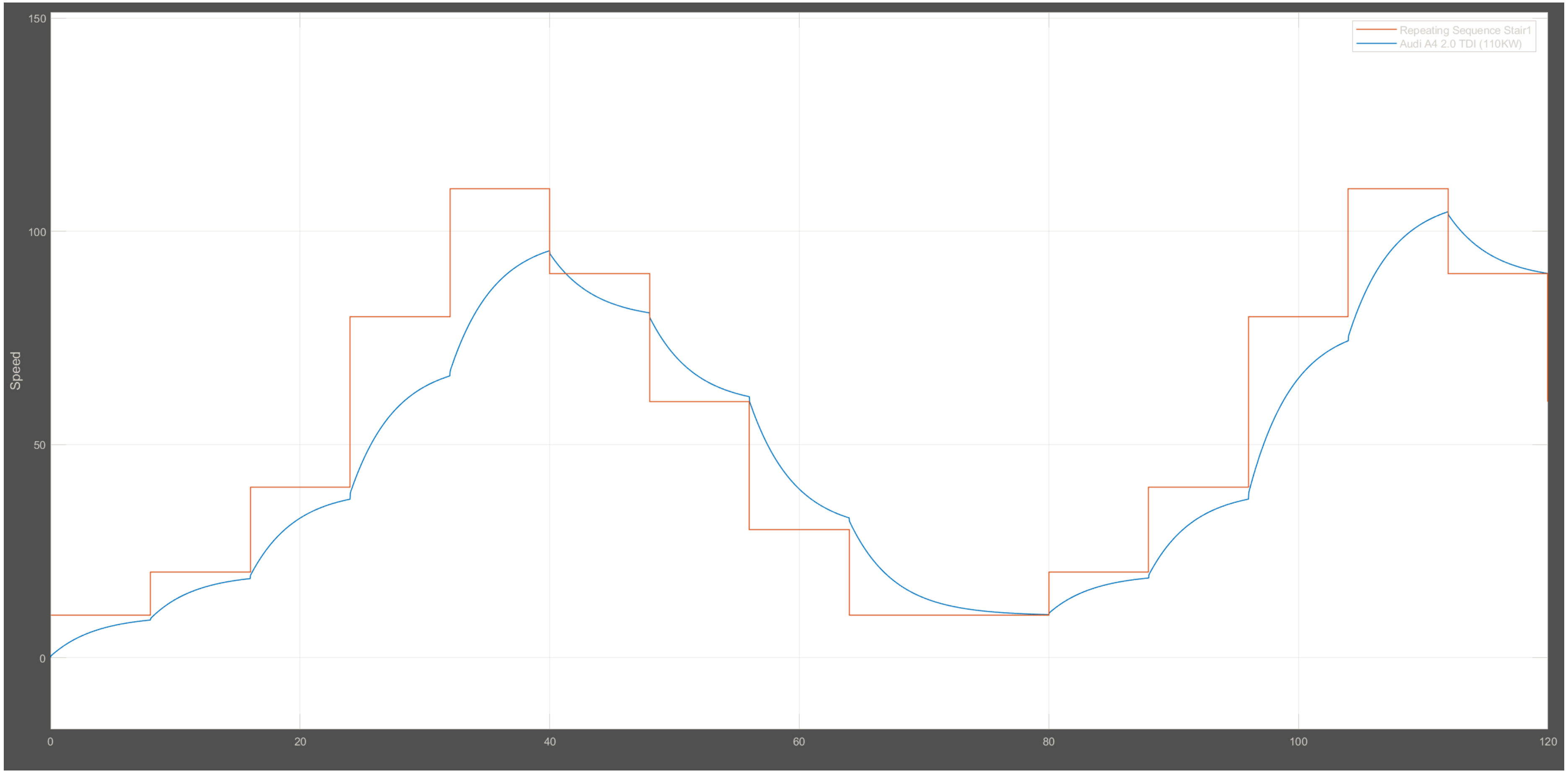
Flat road scenario

[Kp, Kd] = [170, 20]



Mixed road scenario

[Kp, Kd] = [170, 20]



Get back to page 3

References

[1] BMW, "The BMW Series 5 Sedan - Specifications," ed, 2017.

[2] Audi, "Audi A4 Specifications guide," ed, 2017.

[3] J. Smuts, "Ziegler-Nichols Closed-Loop Tuning Method ", ed, 2010.

[4] Karl J. Åström, Richard M. Murray, *Feedback Systems -- An Introduction for Scientists and Engineers* Second ed. 2008.

[5] C. Hardy, "The Basics of Tuning PID Loops ", ed, 2014.

[6] C. L. Hassan Asere, Ruting Jia, "Cruise Control Design Using Fuzzy Logic Controller," presented at the IEEE International Conference on Systems, Man, and Cybernetics, 2015. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7379518

[7] MathWorks. (2018). *Matlab Documentation*. Available: https://au.mathworks.com/help/

[8] https://github.com/harish-kp/cruise-control-design