State Space Control Coursework

Abstract

This study investigates the implementation of an active suspension system using state-space control to improve vehicle ride quality and passenger comfort. The performance of the active suspension was evaluated through stability analysis, bump response tests, and initial condition simulations. Compared to the passive suspension, the active system effectively eliminates a key resonance mode at 6 Hz and reduces the damping of a higher-frequency mode while maintaining system stability.

The active suspension demonstrates substantial improvements in ride dynamics, reducing body displacement and acceleration during road disturbances. In bump response tests, the system minimizes passenger-perceived displacement to less than 20% of the actual road disturbance. Similarly, initial condition simulations show a significant reduction in root mean squared (RMS) acceleration—over 70% for body disturbances and 96% for wheel disturbances—indicating enhanced disturbance recovery.

Introduction

The performance of vehicle suspension systems plays a critical role in ensuring passenger comfort, ride quality, and vehicle stability. Traditional passive suspension systems rely on fixed mechanical components, such as springs and dampers. While effective in many cases, these systems face inherent limitations, as their behaviour cannot adapt dynamically. An Active suspension incorporating sensors, actuators, and control algorithms to actively modify the suspension response in real time provide an opportunity to improve performance but also introduce additional complexity. This study investigates the applicability of an active suspension utilising a proportional derivative controller that would take in ride height data from a suitable sensor to dynamically adjust the system response with an actuator in parallel with a conventional passive shock system.

Aims

The aim of this project is to tune an active vehicle suspension using a mathematical model to improve ride quality and control. The vehicle will have independent suspension utilising an actuator in addition to the passive linear spring and damper at each wheel.

Background

State Space Representation

A **State Space Representation** is a standardised means of modelling complex MIMO dynamic systems using only linear equations in the time domain.
Below are explanations of all of the relevant terms and concepts referred to in this report.

Term	Description
State Space	The **State Space** is an n dimensional manifold with a dimension n for every **State Variable** needed to describe the system.
State Variable	**State Variable**s x_n are any measures of the systems internal dynamics necessary to determine the system state by first order transfer functions.
State Vector	The **State Vector** x is an n dimensional vector that represents the overall state of the system as a point in the **State Space** based on the values of all of the **State Variable**s.
Input Vector	The **Input Vector** u is a vector that represents the external inputs affecting the system.
State Equations	The sets of linear equations in matrix form A & B that transform the **State Vector** x and **Input Vector** u respectively to their resulting effect on the system dynamic $\dot{x}$.
Output Vector	The **Output Vector** y is an n dimensional vector that represents the outputs of interest of the system.
Output Equations	The sets of linear equations in matrix form C & D that transform the **State Vector** x and **Input Vector** u respectively to their resulting effect on the system outputs y.

Input Equation

The **Input Equation** gives the change in the system $\dot{x}$ at a given time t.

$$\dot{x}(t) = Ax(t) + Bu(t)$$

Where:

- $\dot{x}(t)$ is the **derivative** of the **State Vector** at time t
- $x(t)$ is the **State Vector** at time t
- $u(t)$ is the **Input Vector** at time t
- A & B are the **State Equations**:
  - A is a matrix of first order coefficients that relate the **State Variable**s to their effect on the system dynamic.
  - B is a matrix of first order coefficients that relate the system inputs to their effect on the system dynamic.

Output Equation

The **Output Equation** gives the output of the system at time t.

$$y(t) = Cx(t) + Du(t)$$

Where:

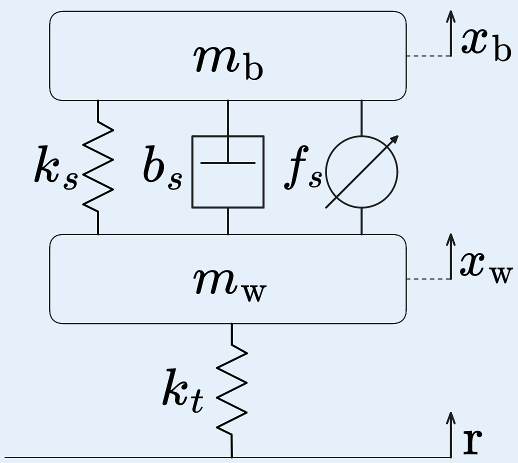
- $y(t)$ is the **Output Vector** at time t
- $x(t)$ is the **State Vector** at time t
- $u(t)$ is the **Input Vector** at time t
- C & D are the **Output Equations**:
  - C is a matrix of first order coefficients that relate the current system state to their impact on the system outputs.
  - D is a matrix of firs order coefficients that relate the inputs to the system to their effect on the system outputs.

Constructing a model

As the vehicle uses an independent suspension the behaviour will be simulated for a single wheel and a quarter of the cars mass, and will be referred to as a QC or quarter car.
The quarter car suspension model considers two masses; the body mass (m_{b}) and the wheel mass (m_{w}).
The wheel is connected to the ground via a spring (k_t) representing tire stiffness, while the body is supported by a passive spring-damper combination (k_s, b_s). The system is extended with an active suspension component, which includes an actuator (f_s) that applies a control force between the wheel and the body.

Free Body Diagrams

Figure 1: Quarter Car System diagram



This can be broken up into the following free body diagrams:

Figure 2: Car Body Free Body Diagram

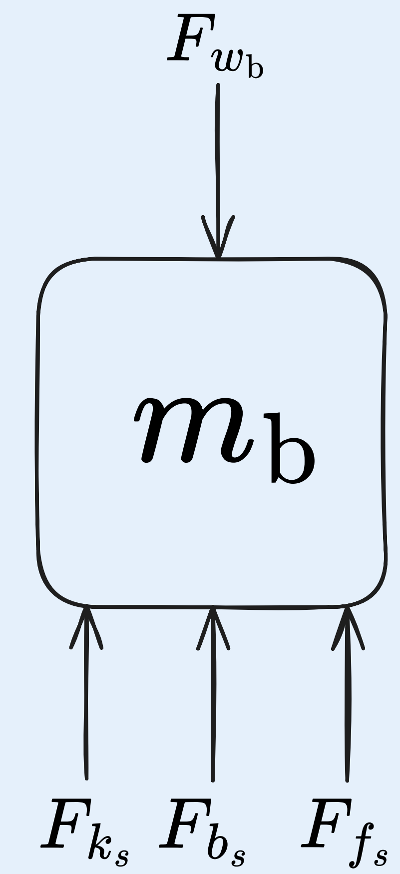
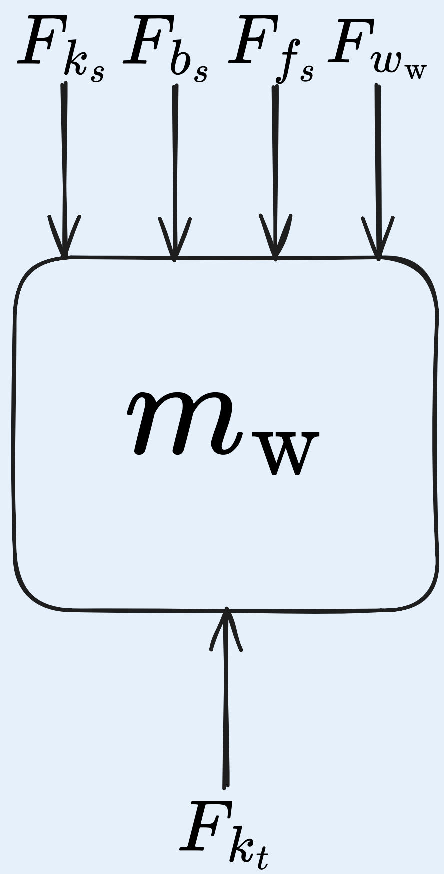


Figure 3: Wheel Free Body Diagram



Givens

The following are known values for the system that the model will be based off:

Description	Notation	Quantity	Units
Quarter Car Mass	m_{b}	150	kg
Wheel Mass	m_{w}	11	kg
Suspension Damping Constant	b_s	690	$\mathrm{N{\cdot}m^{-1}{\cdot}s}$
Suspension Spring Constant	k_s	6936	$\mathrm{N{\cdot}m^{-1}}$
Tire Effective Spring Constant	k_t	28712	$\mathrm{N{\cdot}m^{-1}}$

Assumptions and Approximations

As this is an independent suspension each wheel can be be assumed to behave similarly, and thus useful results can be found by only a quarter of the car and a single wheel.

Only linear vertical motion of the system will modelled under the assumption motion & rotation in other axis will be effectively constrained. The car body and wheel will be approximated as point masses within this single dimension with all forces acting in line with the centres of mass.

Factors such as air resistance will be ignored, only the forces directly mentioned and their implicit consequences such as inertia from newton's second law will be in effect.

The model will assume the system starts in a steady state and is already under compression. It will assume the wheel never looses contact with the road and thus a restoring force equal to the effective elasticity of the tire is always in effect.

Constituent forces

In order to determine the equations of motion the following constituent forces are considered.

Tire

$$F_{k_t} = k_t(r - x_w)$$

Spring

$$F_{k_s} = k_s(x_w - x_b)$$

Damper

$$F_{b_s} = b_s(\dot{x}_w - \dot{x}_b)$$

Actuator

$$F_{f_s} = f_s(k_1 x_b + k_2 \dot{x}_b + k_3 x_w + k_4 \dot{x}_w)$$

Equations of Motion

According to Newton's second law the acceleration of the bodies can be found by dividing the net force on the body by it's mass.

$$a = \frac{F}{m}$$

Car Body Acceleration

$$a_{\mathrm{b}} = \frac{F_{f_s} + F_{k_s} + F_{b_s}}{m_{\mathrm{b}}}$$

Wheel Acceleration

$$a_{\mathrm{w}} = \frac{F_{k_t} - F_{f_s} - F_{k_s} - F_{b_s}}{m_{\mathrm{w}}}$$

State Matrices

The standard form for the state equation is:

$$\dot{\mathrm{x}} = Ax + Bu$$

Where:

- $\dot{x}$ is the derivative of the State Vector
- x is the State Vector
- u is the Input Vector
- A is a matrix of first order coefficients that relate the State Variables to their effect on the system dynamic.
- B is a matrix of first order coefficients that relate the system inputs to their effect on the system dynamic.

The state vector x is comprised of the following state variables:

$$\mathrm{x} = \begin{bmatrix} x_1 := x_{\mathrm{b}} \\ x_2 := \dot{x}_{\mathrm{b}} \\ x_3 := x_{\mathrm{w}} \\ x_4 := \dot{x}_{\mathrm{w}} \end{bmatrix}$$

Where x and $\dot{x}$ refer to displacement and velocity respectively and b and w the body and wheel. Therefore the derivative of the state vector which gives the rate of change of the state vector is as follows:

$$\dot{\mathrm{x}} = \begin{bmatrix} \dot{x}_1 := \dot{x}_{\mathrm{b}} = x_2 \\ \dot{x}_2 := \ddot{x}_{\mathrm{b}} = a_{\mathrm{b}} \\ \dot{x}_3 := \dot{x}_{\mathrm{w}} = x_4 \\ \dot{x}_4 := \ddot{x}_{\mathrm{w}} = a_{\mathrm{w}} \end{bmatrix}$$

Thus A and B are:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_s}{m_{\mathrm{b}}} & -\frac{b_s}{m_{\mathrm{b}}} & \frac{k_s}{m_{\mathrm{b}}} & \frac{b_s}{m_{\mathrm{b}}} \\ 0 & 0 & 0 & 1 \\ \frac{k_s}{m_{\mathrm{w}}} & \frac{b_s}{m_{\mathrm{w}}} & \frac{-k_s-k_t}{m_{\mathrm{w}}} & -\frac{b_s}{m_{\mathrm{w}}} \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & \frac{f_s}{m_{\mathrm{b}}} \\ 0 & 0 \\ \frac{k_t}{m_{\mathrm{w}}} & -\frac{f_s}{m_{\mathrm{w}}} \end{bmatrix}$$

System Dynamics

A state space representation will be used to model the behaviour of the system where the state variables will be the displacement and velocity of the car body and wheel.

Implementation

Comparison of Simulink and Matlab implementation

Simulink is a visual block based user interface for MATLAB, the visual nature can help maintain a big picture intuition that can get lost in code. This can be especially helpful in larger organisations that need to collaborate and communicate with more stakeholders especially when communicating to non programmers.
That there are differences in the default workflows and as such some things are easier to do in one than the other. Most users will find a balance of the two and frequently use both in any given project. Functions from MATLAB scripts can easily be called in a Simulink `.slx` and so too outputs from Simulink simulations are accessible in scripts through MATLABs powerful shared "workspace" system.

The modelling for this project was done both in Simulink and MATLAB scripts. It's worth noting that MATLAB results can differ when running the same simulation in `Discrete-time` vs `Continous-time` and by default Simulink tends to handle simulations in continuous time. Similarly by default Simulink will handle time itself but in a tightly integrated project it can be important to use a specific shared time vector between sections handled in scripts and Simulink simulations. When this was taken into account the results achieved by both methods where Identical.

Figure 3: Passive Suspension Simulink Diagram

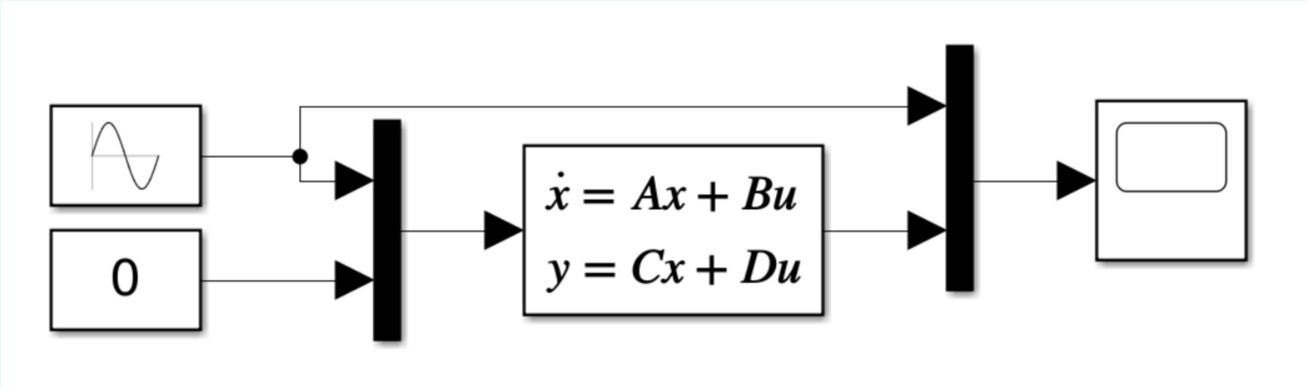
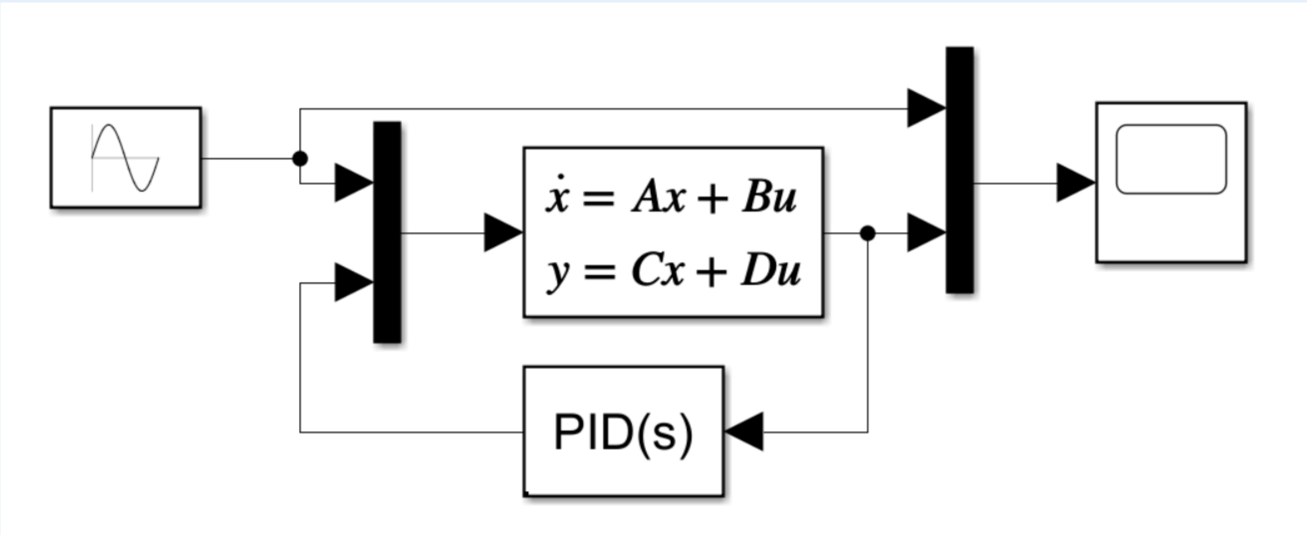


Figure 4: Active Suspension Simulink Diagram



Active Suspension Implementation

This section covers the implementation of the active suspension utilising the actuator and a PD controller in MATLAB.
The controller takes in the car body displacement state variable referred to previously as x_b / `x_b` as it's input, which may be collected in the real system with a ride height sensor. PD refers to proportional and derivative as the controller takes the form `controller = kp + kd * s;` where `kp` is a proportional gain and `kd` is a derivative gain on the car body position. The output of this controller is fed into the actuator input of the state space system where it applies it's force against the wheel and body.

Passive Suspension Performance

Stability

The poles of the system can be used to judge it's stability. The poles can be found as the eigenvalues of the feed forward transfer function and Matlab has built in methods `eig` or `pole` which can be used identify them.

```
>> pole(passive_suspension)
```

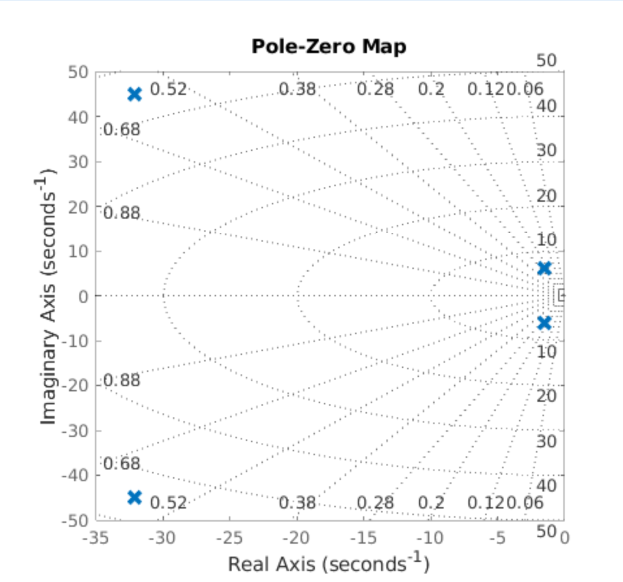
```
ans =

 -32.1112  +44.9092i
 -32.1112  -44.9092i
  -1.5525  + 6.0982i
  -1.5525  - 6.0982i
```

From these an intuition can be gained of the systems inherent stability and and of its modes of natural resonance. In this case they indicate 2 modes of damped oscillation.
The real component indicates the linear damping of the system; if it is positive the system is positively reinforcing and any disturbance will grow indefinitely, if it is negative it contributes negative feedback and the system will trend towards a steady state.
The imaginary component indicates oscillatory behaviour, it indicates the frequency(s) the system will tend to oscillate at when disturbed.

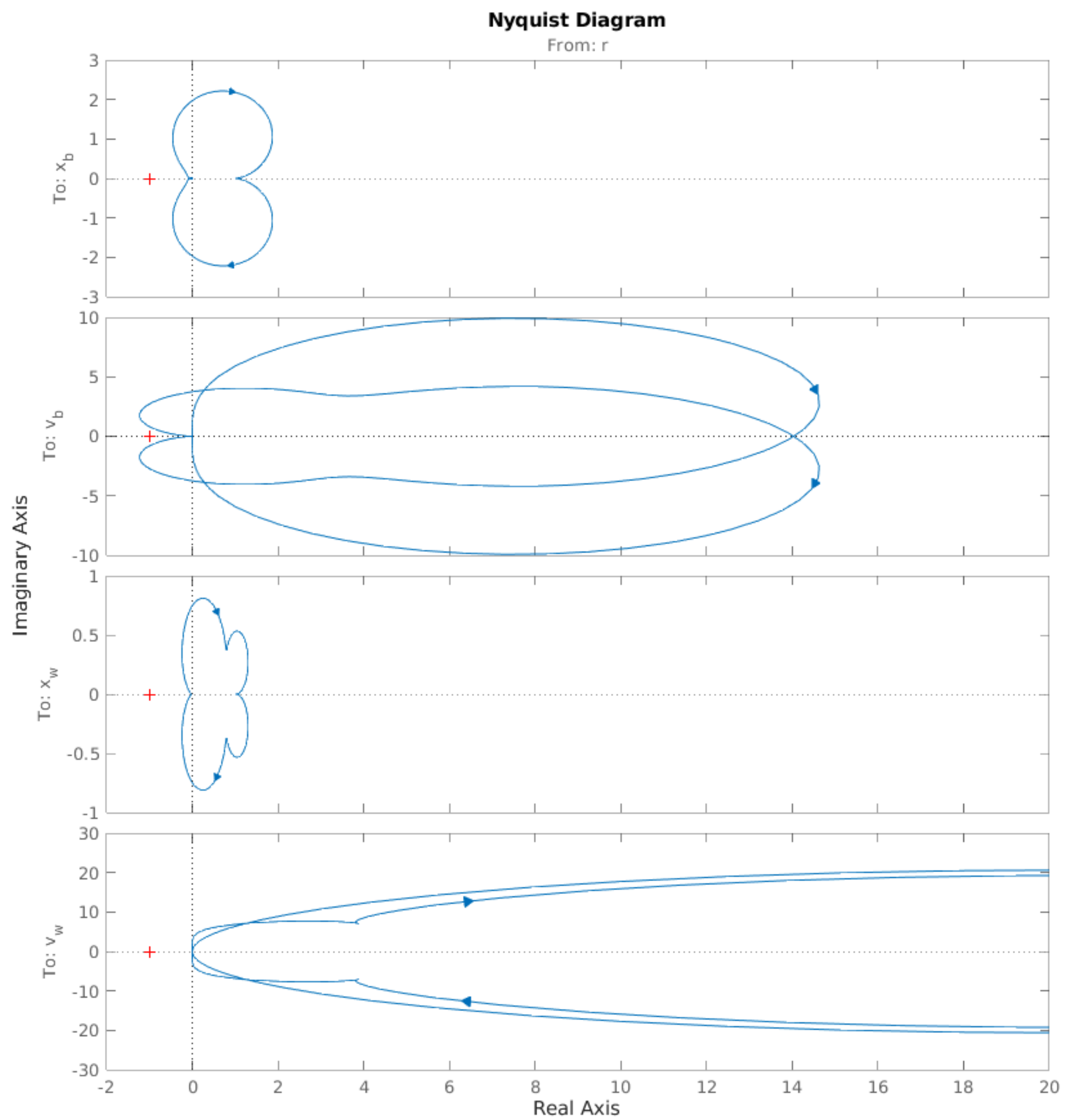
Faster oscillations with less damping combine to make a more energetic system, in the case of a suspension this is to be avoided as it makes the suspension feel too "bouncy". In contrast slower oscillation with less damping can make the car feel "floaty" removing the "road feel" that informs the drivers intuition. Damping ratio gives a measure of this behaviour where the first case is considered under damped indicated by a damping ratio closer to 0, and the second case would be over damped with a damping ratio closer to 1.

Figure 5: Passive Suspension Pole Zero Map



The radial gird lines in the figure above indicate lines of constant damping ratio and the blue crosses are the poles of the passive suspension system.
The mechanical properties of the spring, damper & tire in the passive suspension determine these poles and cause natural modes of oscillation. These present slightly differently differently in each state variable as can be visualised in the Nyquist plots below.

Figure 6: Passive Suspension Nyquist Plots



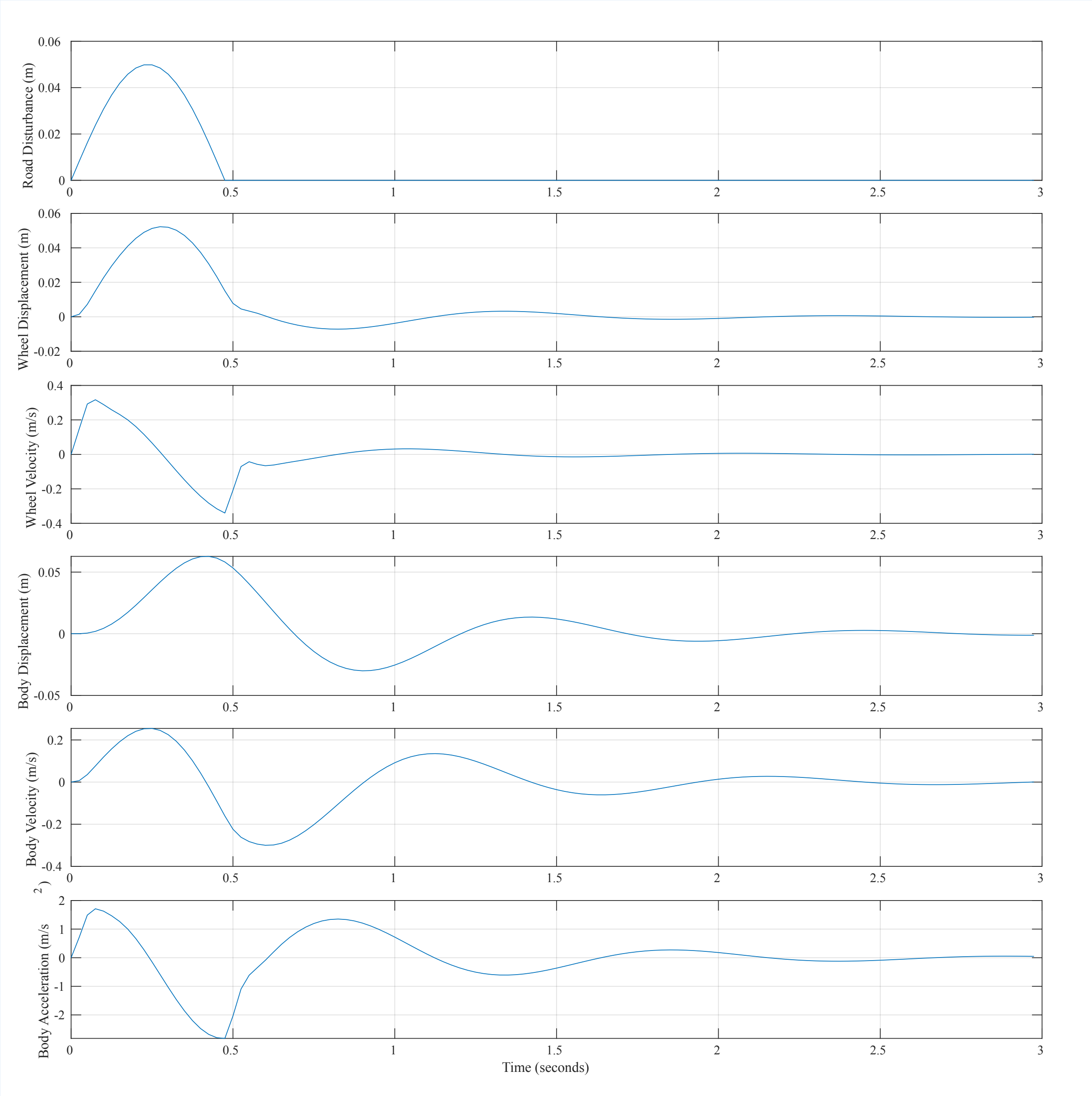
The wheel position displays two fairly distinguishable elliptical modes this is to be expected as it is connected by two springs of quite different elasticity. In contrast the car body position displays a more homogenised oscillation that can be interpreted as the superposition of 1) the natural frequency of the car body mass & suspension spring as it is deformed by 2) the wheel mass and tire elasticity.
The velocities are of course derivatives of the resonance of the position variables, displaying the same behaviour but as a rate of change. An intuition can be gleamed for visualising how the potential energy shifts from the car body mass to the tire and back by following around the "fish shape" in the resonance of the car body velocity. This effect is far more substantial than it's equivalent in the wheel's velocity as the car body has significantly more mass and thus deflects less.

Bump tests

The behaviour of the system is simulated to observe the response to a bump in the road. The bump in the simulation below is a 5 cm tall half sinusoid with a duration of half a second.

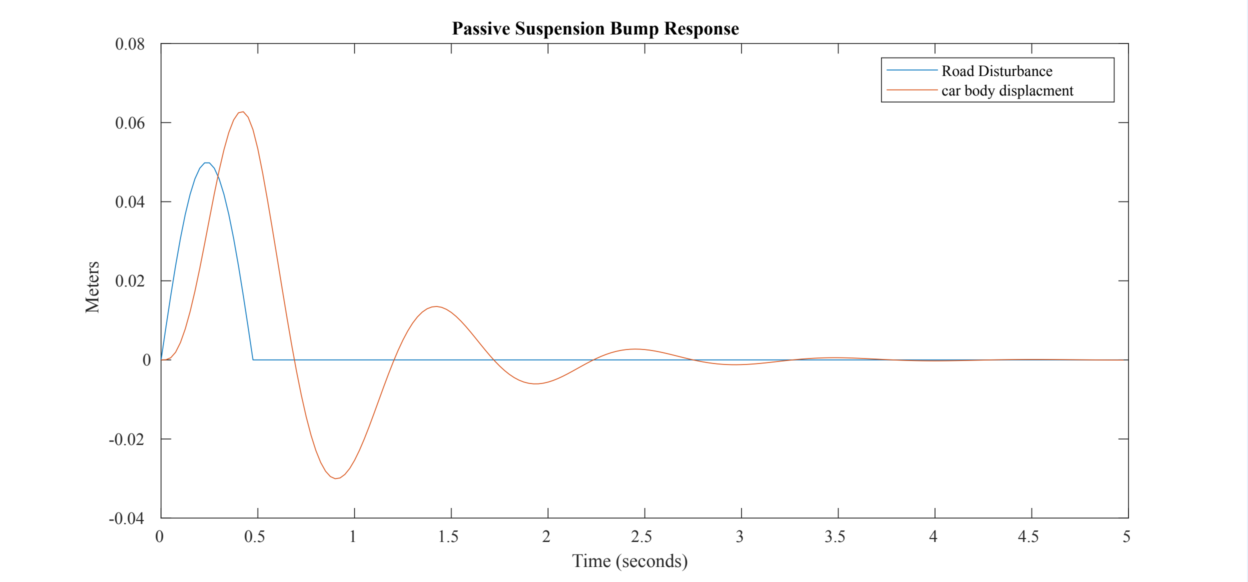
The simulations where created using the bump_response.m script that can be found here in the working directory.

Figure 7: Passive Suspension Bump Response



The passive suspension dose slightly reduce the impulse but results in a larger displacement than the original bump and oscillates several times.

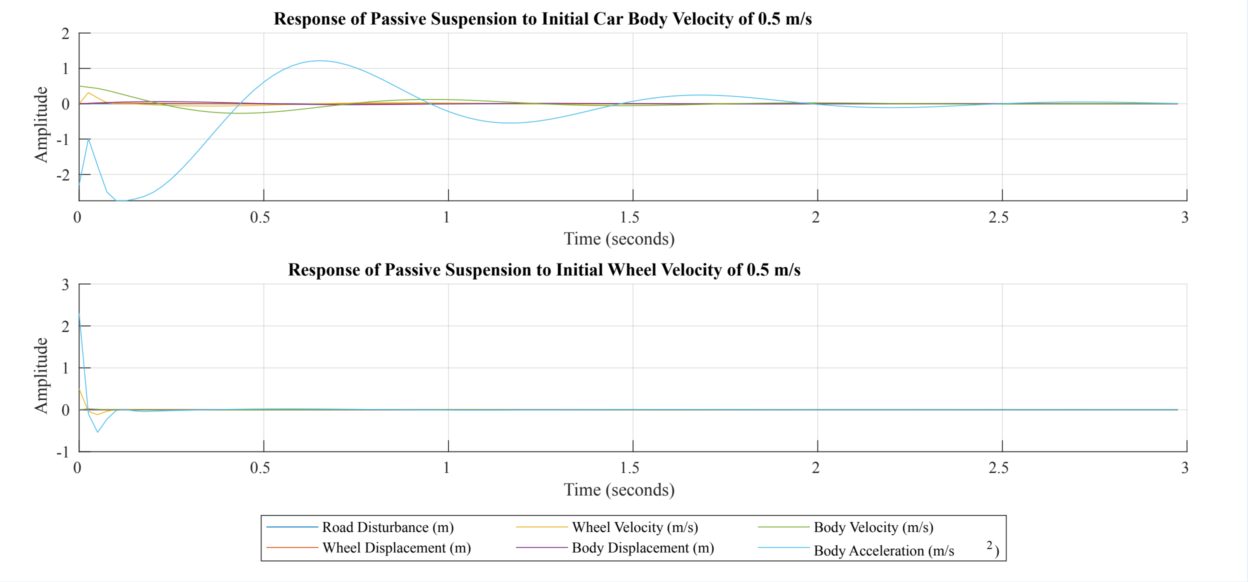
Figure 8: Passive Suspension bump against body displacement



Initial Condition tests

The system is simulated with an initial velocity in the body and then in the wheel to compare how it recovers from a disturbance in each case.
The simulations where created using the [initial_velocity_response.m]() script that can be found [here]() in the working directory.

Figure 9: Passive Suspension Initial Velocity Response



The system recovers from disturbances to the wheel much faster than a similar disturbance to the body as can be seen in [figure n](). However this is to be expected as at the same velocity the car body represents far more inertia and thus more energy for the system to dissipate.

Active Suspension Performance

Where the [Passive Suspension Testing]() section focused more on introducing and explaining of terms measures and intuitions. The performance of the active suspension system will be discussed largely in contrast with the baseline performance of the passive suspension system.
All of the same tests have been performed with the MATLAB implementation of the active suspension utilising the actuator and a PD controller. The controller takes in the car body displacement state variable referred to previously as x_b / `x_b` as it's input, which may be collected in the real system with a ride height sensor. PD refers to proportional and derivative as the controller takes the form `controller = kp + kd * s;` where `kp` is a proportional gain and `kd` is a derivative gain on the car body position. The

output of this controller is fed into the actuator input of the state space system where it applies it's force against the wheel and body.

Stability

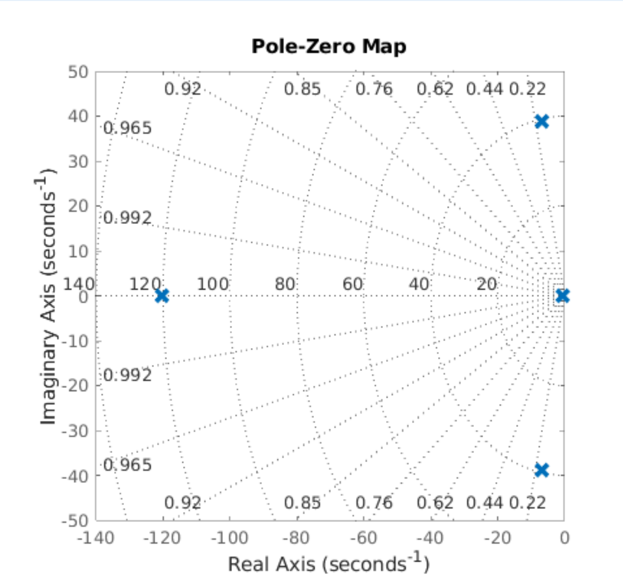
```
>> pole(tuned_suspension)

ans =

   -120.48  +      0i
   -6.8655  +38.709i
   -6.8655  -38.709i
  -0.64921  +      0i
```

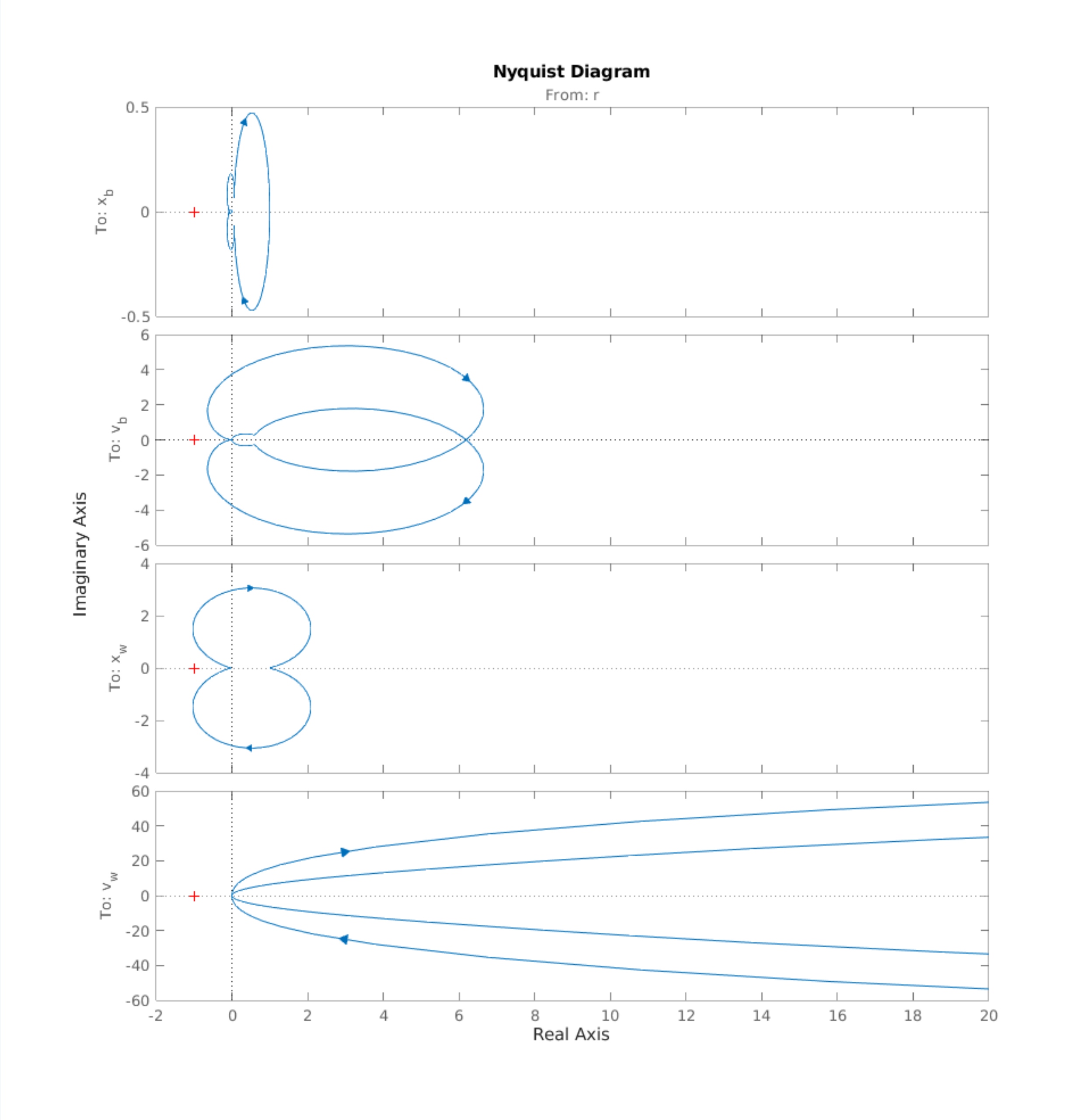
The active suspension eliminates one of the modes of resonance that was present in the Passive Suspension Testing at $6\ \mathrm{Hz}$. It slightly reduces the natural frequency of the other mode that was previously near $45\ \mathrm{Hz}$ and reduced its damping coefficient. Two additional poles have been added, these act as negative gains which can help the system as long as they aren't in a similar range to typical system inputs, in which case they can lead to instability.

Figure 10: Active Suspension Pole Zero Map



Comparing the Nyquist plots below with those of the Passive Suspension shows how the controller has effected the natural modes of resonance in the system. The modes of the body position and acceleration are substantially smaller in both their real and imaginary components whereas those of the wheel are larger. The system still has to absorb and dissipate the same amount of energy but the active suspension shifts much of the systems energy into the wheel mass.

Figure 11: Active Suspension Nyquist Diagrams

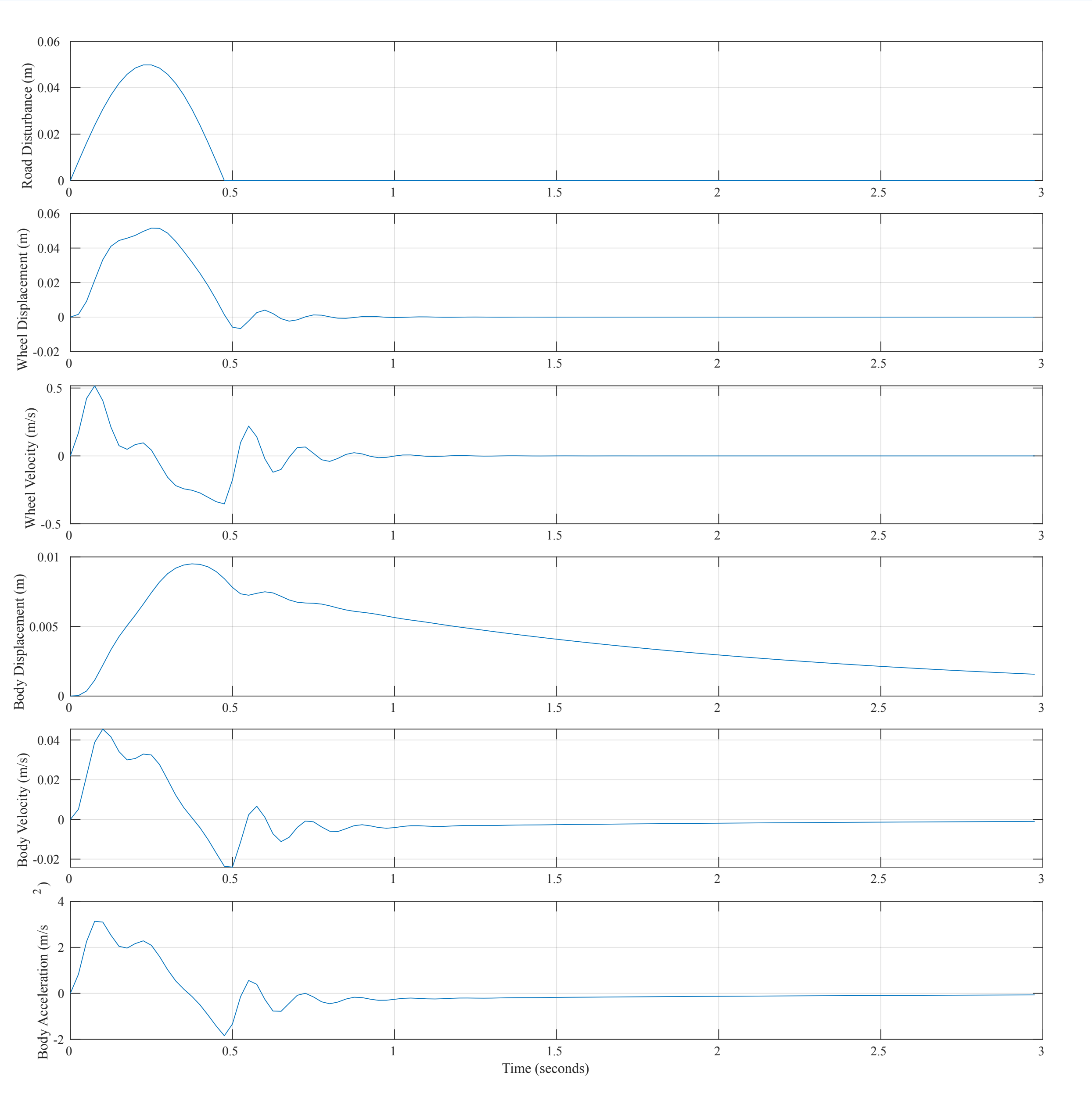


Bump Test

The behaviour of the system is simulated to observe the response to a bump in the road. The bump in the simulation below is a 5 cm tall half sinusoid with a duration of half a second.

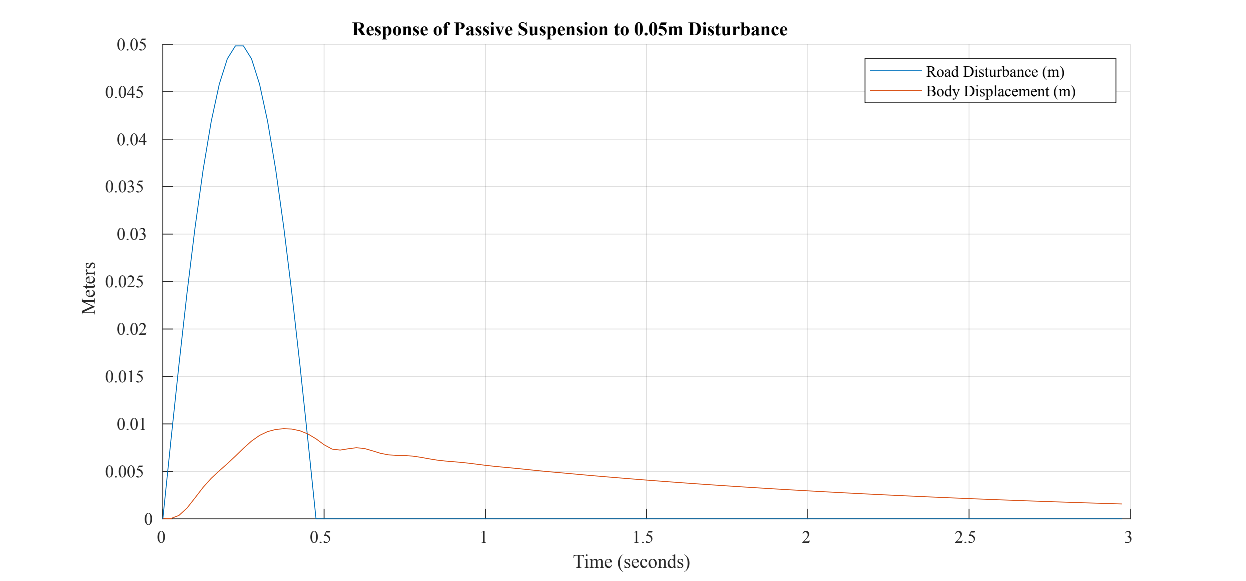
The simulations where created using the bump_response.m script that can be found here in the working directory.

Figure 12: Active Suspension Bump Response



The Active suspension provides a substantial improvement in every respect. It eliminates the oscillations seen in the Passive Suspension Testing > Bump tests and reduces the magnitude of the body displacement, velocity and acceleration.

Figure 13: Active Suspension Bump against Body Displacment

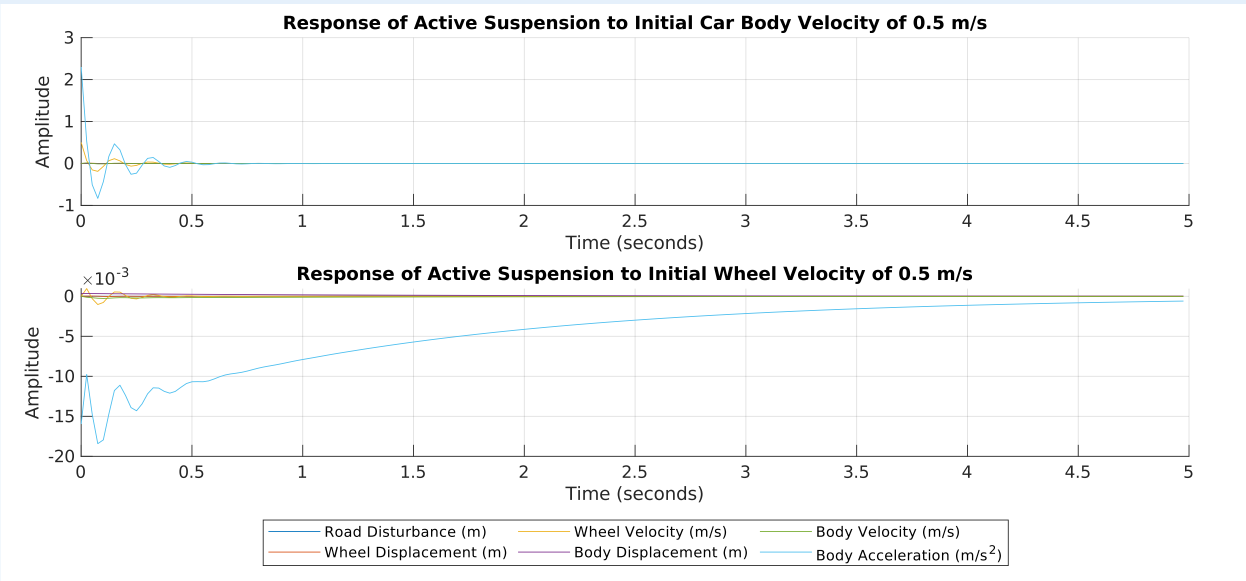


The effect of the bump is diminished significantly from the perspective of the passenger with the displacement felt being reduced to less than 20% of the actual disturbance.

Initial Condition Test

The system is simulated with an initial velocity in the body and then in the wheel to compare how it recovers from a disturbance in each case.
The simulations where created using the initial_velocity_response.m script that can be found here in the working directory.

Figure 14: Active Suspension Initial Velocity Response



In the initial car body velocity simulation the peak car body acceleration is actually 50% higher with this controller implementation than it was in the Passive Suspension Testing > Initial Condition tests. However, the benefit is made clear by comparing the root mean squared acceleration across the same 5 second simulation with $0.5\ m/s$ of initial body velocity, the controller reduces `rms(a_b)` down from $0.6677\ m/s$ to $0.1903\ m/s$, less than 30% of that of the passive suspension.
The initial wheel velocity simulation sees significant improvement by all measures, most importantly peak acceleration similarly reduced by 2 orders of magnitude with a 96% reduction in root mean squared acceleration `rms(a_b)`.

Discussion

The results of the simulations are promising, but the results must be considered in context. These results rely on the assumptions and approximations discussed previously several of which do not hold in reality.

For example the damping coefficient is assumed to be linear in terms of force/velocity, this simplifies the calculations and is likely how a predictive controller might model the system for processing speed but likely deviates from reality. The behaviour of dampers depends heavily on the type of damper used in the vehicles shocks and can change over the course a single use and over its useful lifespan.[1] It's a similar case with the approximations of elasticity for the tire and the spring but this provides an illustration of the trade off's that must be made between accuracy and practicality. The usefulness of a model is not in it's perfection but in it's ease of application.
It should be considered that the system would behave quite differently if the controller introduced any significant latency, in which case additional modes of resonance emerge that can cause instability.
In addition the physical constraints of the actuator should be accounted for in future simulations should this project be carried forward. These results may depend on unreasonable force, response time, travel speed, accuracy, ect. and taking these limitations into account may require a different tuning.
It would also be wise to utilise some more advanced tuning methods such as the H_∞ synthesis methods provided by the `control systems` package that could optimise weights across a range of uncertainties through the system. [2]

Conclusion

The simulations show the use of an actuator and tuned PD controller can provide significant improvements in system performance as can be measured in the significant reduction in both peak magnitude and root mean square acceleration of the car body in every test.

Overall, these simulations provide a strong foundation for moving forward with the active suspension design and demonstrate the potential benefits of this approach for enhancing vehicle ride quality and passenger comfort.

---

1. T. P. Waters, Y. Hyun, and M. J. Brennan, “The Effect of Dual-Rate Suspension Damping on Vehicle Response to Transient Road Inputs,” *Journal of Vibration and Acoustics*, vol. 131, no. 1, Jan. 2009, doi: https://doi.org/10.1115/1.2980370. read online ↩
2. “control.hinfsyn — MATLAB Control Systems Library documentation.” [Online]. Available: https://www.mathworks.com/help/robust/ref/dynamicsystem.hinfsyn.html. [Accessed: 19-Dec-2024]. ↩