

Intro to Modern Control – Project (Part II)

Ajinkya Jain and Soovadeep Bakshi

April 28, 2017

Closed Loop Controllers

Linear Quadratic (LQ) Regulator (Infinite and Finite Horizon)

The Linear Quadratic Regulator is a closed loop controller whose formulation is derived in a similar fashion to the open loop controller discussed in Part I. The cost function for the infinite time LQR is defined in a similar fashion, as follows:

$$J = \frac{1}{2} \int_0^{\infty} (\ddot{z}_s + \rho_1 x_1^2 + \rho_2 x_2^2 + \rho_3 x_3^2 + \rho_4 x_4^2) dt \quad (1)$$

$$= \frac{1}{2} \int_0^{\infty} (\mathbf{x}^T Q \mathbf{x} + 2\mathbf{x}^T N F_a + F_a^T R F_a) dt \quad (2)$$

where $x_{1,2,3,4}$ are the states. The cost function puts weightage on the three performance parameters defined earlier (sprung mass acceleration, suspension deflection and tire deflection), as well as on the sprung and unsprung mass velocities. In the cost function, the three performance parameters (sprung mass acceleration, suspension deflection and tire deflection) are given more weightage, while the velocities are given smaller weights to ensure that the Q matrix is positive definite in equation 2. The highest weight is chosen for sprung mass acceleration, which is the most important parameter for ensuring passenger comfort. Thus, the chosen values of the weights are $\rho_1 = \rho_3 = 0.4$, $\rho_2 = \rho_4 = 0.04$. The matrices Q , N and R have been defined in Part I.

Therefore, the Pontryagin function, H , can be defined as follows:

$$H = \frac{1}{2} (\mathbf{x}^T Q \mathbf{x} + 2\mathbf{x}^T N F_a + F_a^T R F_a) + \boldsymbol{\lambda}^T (A \mathbf{x} + B F_a + L \dot{z}_r) \quad (3)$$

From equation 3, the state and co-state equations can be derived as:

$$\dot{\mathbf{x}} = \frac{\partial H}{\partial \boldsymbol{\lambda}} \quad (4)$$

$$= A\mathbf{x} + BF_a + L\dot{z}_r \quad (5)$$

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{x}} \quad (6)$$

$$= -Q\mathbf{x} - NF_a - A^T \boldsymbol{\lambda} \quad (7)$$

The stationarity equation is defined as:

$$0 = \frac{\partial H}{\partial F_a} \quad (8)$$

$$= RF_a + B^T \boldsymbol{\lambda} + N^T \mathbf{x} \quad (9)$$

$$\implies F_a = -R^{-1}B^T \boldsymbol{\lambda} - R^{-1}N^T \mathbf{x} \quad (10)$$

To make the controller closed loop, the sweep method has been used, as shown below:

$$\boldsymbol{\lambda} = S\mathbf{x} \quad (11)$$

Equation 11 can then be expanded and used to derive the matrix Riccati equation as follows:

$$\begin{aligned} \dot{S} &= -S(A - BR^{-1}N^T) - (A - BR^{-1}N^T)^T S - (Q - NR^{-1}N^T) \\ &\quad + SBR^{-1}B^T S \end{aligned} \quad (12)$$

For the matrix Riccati equation, the matrix satisfies the boundary condition $S(T) = 0$ for the chosen cost function. Therefore, this matrix differential equation can be integrated backwards. However, for the infinite time case, it is assumed that due to the infinite time horizon, the matrix S is a constant, i.e., $\dot{S} = 0$. The matrix differential equation therefore becomes an algebraic Riccati equation (ARE), which can then be solved easily.

The control law is then modified from equation 10 as follows:

$$F_a = -R^{-1}B^T S\mathbf{x} - R^{-1}N^T \mathbf{x} \quad (13)$$

$$= -R^{-1}(B^T S + N^T)\mathbf{x} \quad (14)$$

For the finite time LQR problem, the cost function used is the same, except for the fact that a final time, T , is specified. The cost function can therefore be written as:

$$J = \frac{1}{2} \int_0^T (\ddot{z}_s + \rho_1 x_1^2 + \rho_2 x_2^2 + \rho_3 x_3^2 + \rho_4 x_4^2) dt \quad (15)$$

$$= \frac{1}{2} \int_0^T (\mathbf{x}^T Q\mathbf{x} + 2\mathbf{x}^T NF_a + F_a^T RF_a) dt \quad (16)$$

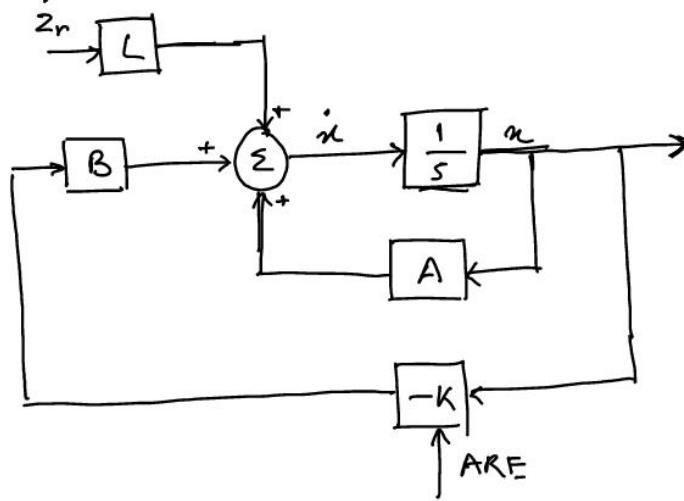


Figure 1: Block diagram for infinite time LQR control.

The closed loop control law for the finite time case is the same as the infinite time case (equation 14), derived using the Sweep method. The only difference is that the differential matrix Riccati is used to update the S matrix. Therefore, in the finite time case, the feedback gain used is not a constant value as in the infinite time case.

The block diagrams for the infinite and finite horizon LQR controllers are shown in Fig. 1–2.

The results for the infinite and finite time LQR controllers are shown in the following figures. Sprung mass deflection is chosen as an additional parameter for comparison because it is directly felt by the passengers. The comparison between the four parameters are shown in Fig. 3 – 7. The comparison of feedback gains is shown in Fig. 8, and the control inputs for the two cases are shown in Fig. 9. It can be seen that there is almost no difference between the finite time and infinite time cases, but the finite time LQR is a bit faster than the infinite time response. The effect of updating the Riccati matrix is seen in Fig. 8. The control inputs for both cases are also similar.

The results for the dynamic case are also presented in Fig. 10–13. It can be seen that the performances of the finite and infinite time controllers are almost indistinguishable because the chosen final time is large enough. The control inputs are also shown in Fig. 14.

The real parts of the closed loop eigenvalues are shown for the finite and infinite time cases in Fig. 15. It can be seen that for the finite case, the eigenvalues vary dynamically.

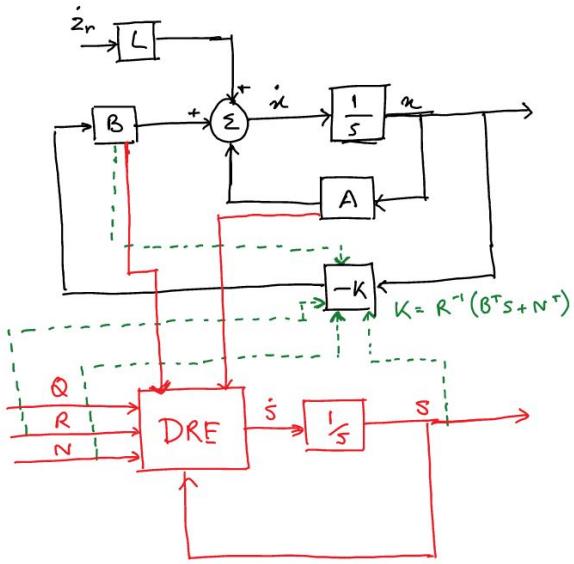


Figure 2: Block diagram for finite time LQR control.

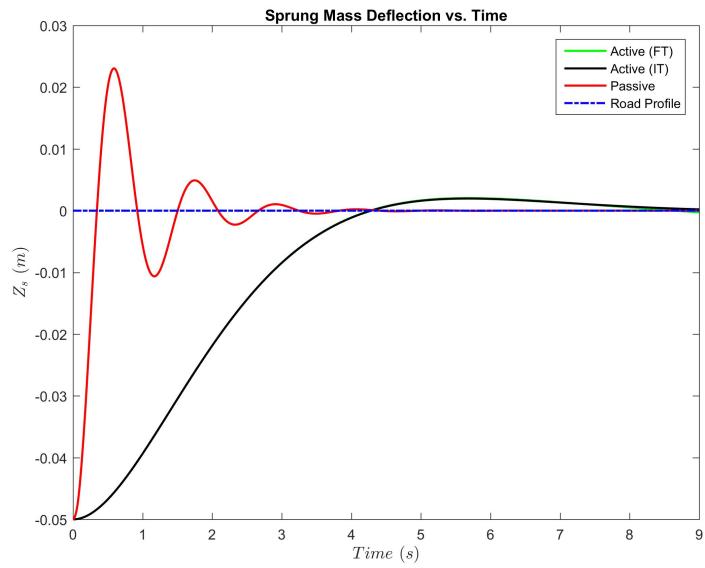


Figure 3: Sprung mass deflection for static case with active LQR control.

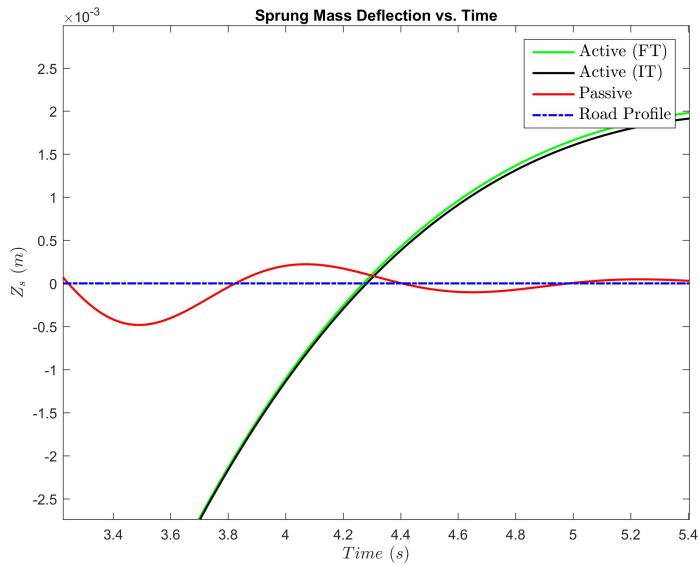


Figure 4: Difference between the infinite and finite time LQR controllers.

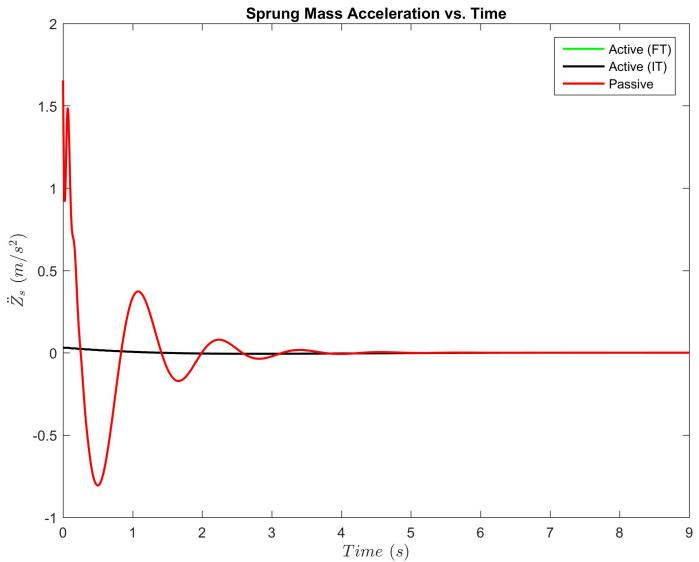


Figure 5: Sprung mass acceleration for static case with active LQR control.

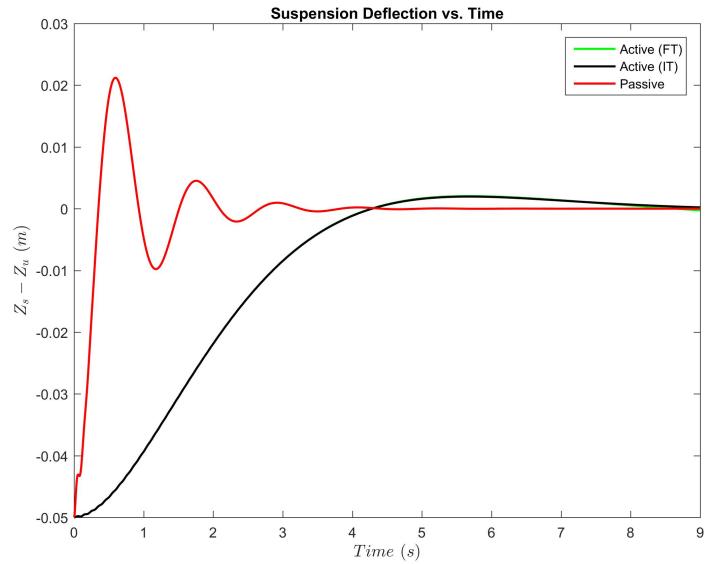


Figure 6: Suspension deflection for static case with active LQR control.

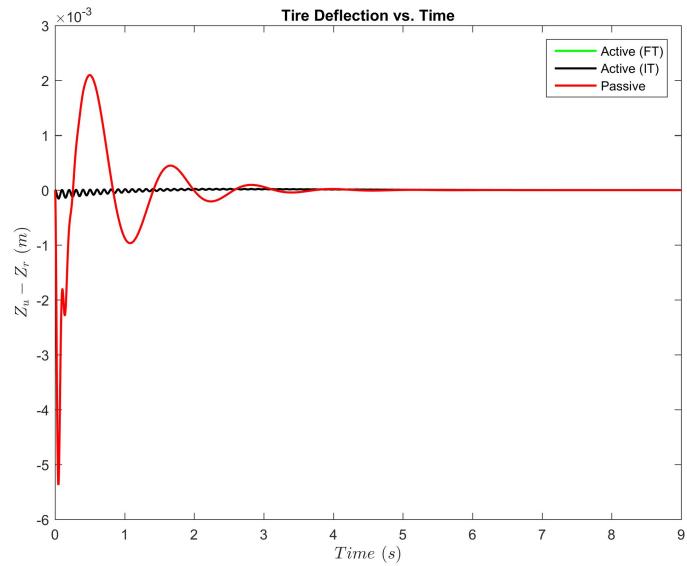


Figure 7: Tire deflection for static case with active LQR control.

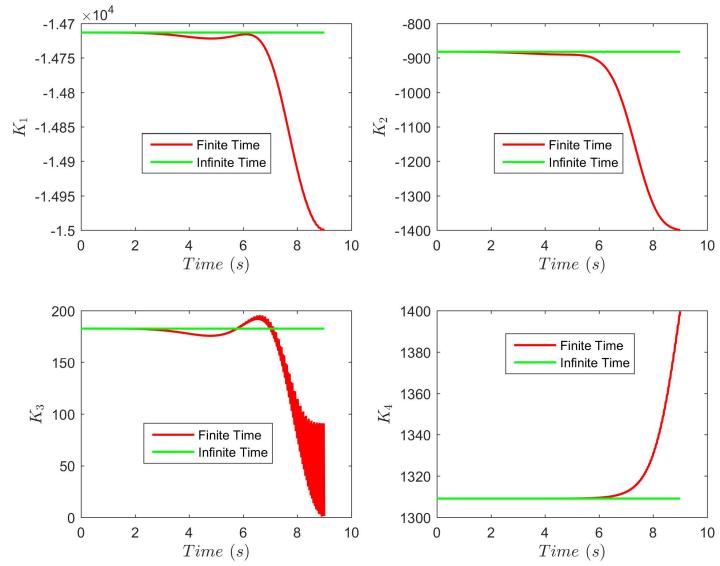


Figure 8: Feedback gains for finite vs. infinite time LQR controllers.

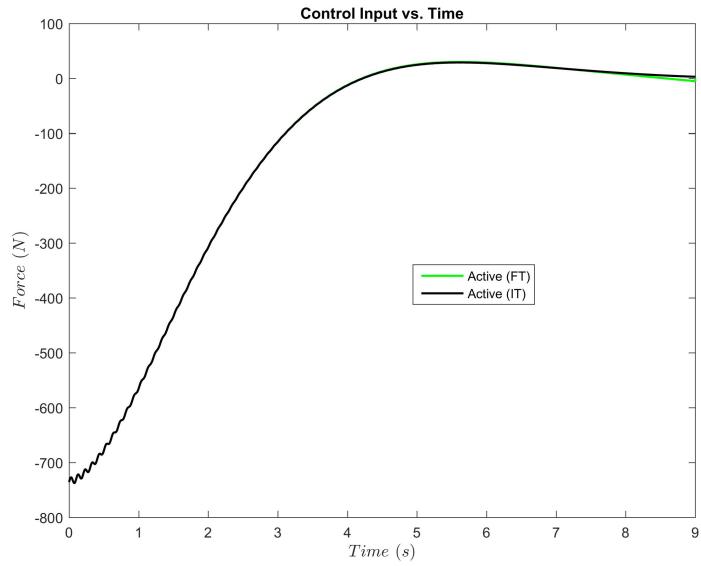


Figure 9: Control inputs for the two LQR controllers.

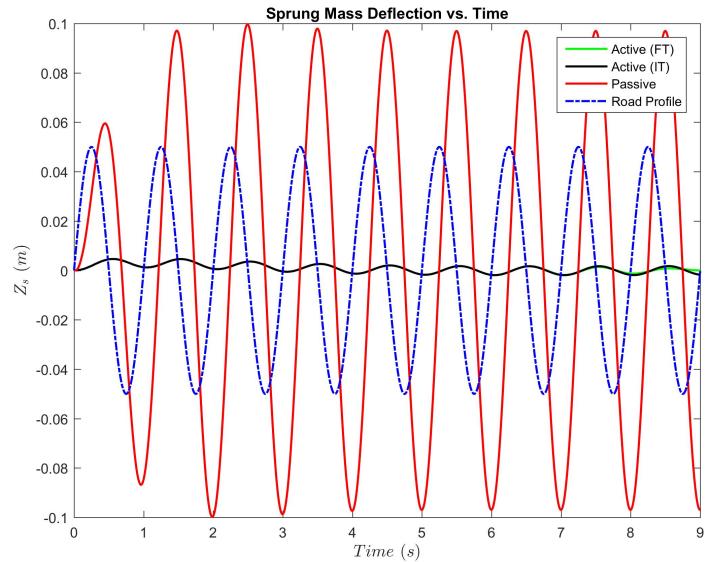


Figure 10: Sprung mass deflection for dynamic case with active LQR control.

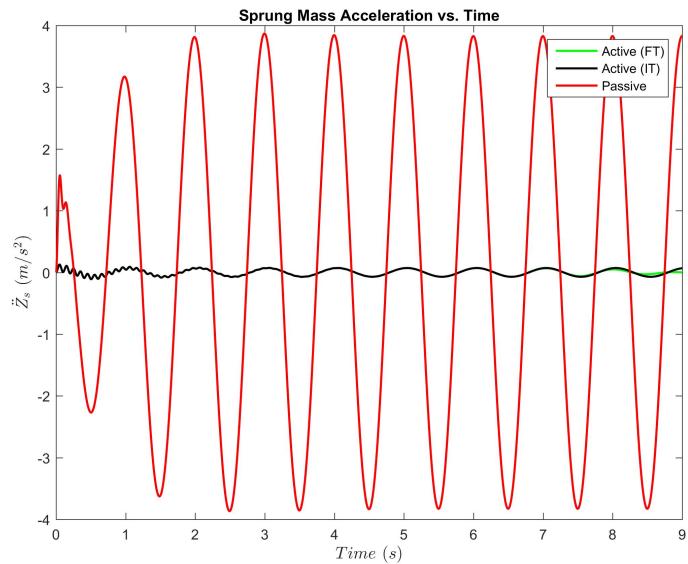


Figure 11: Sprung mass acceleration for dynamic case with active LQR control.

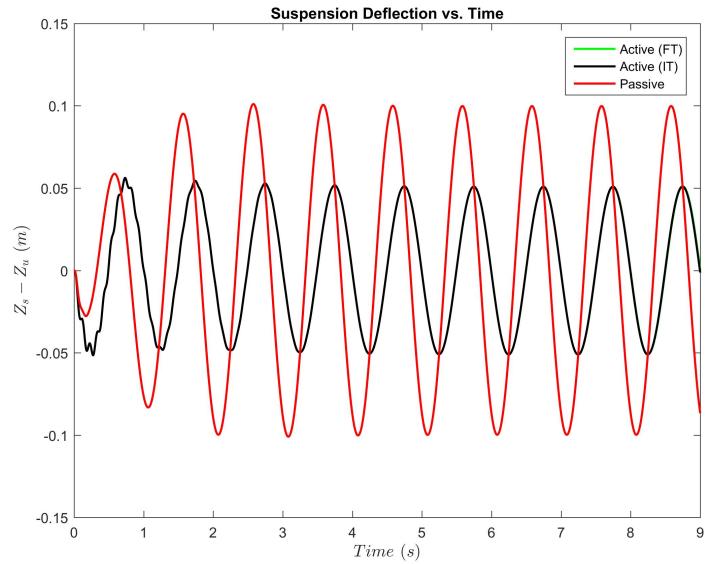


Figure 12: Suspension deflection for dynamic case with active LQR control.

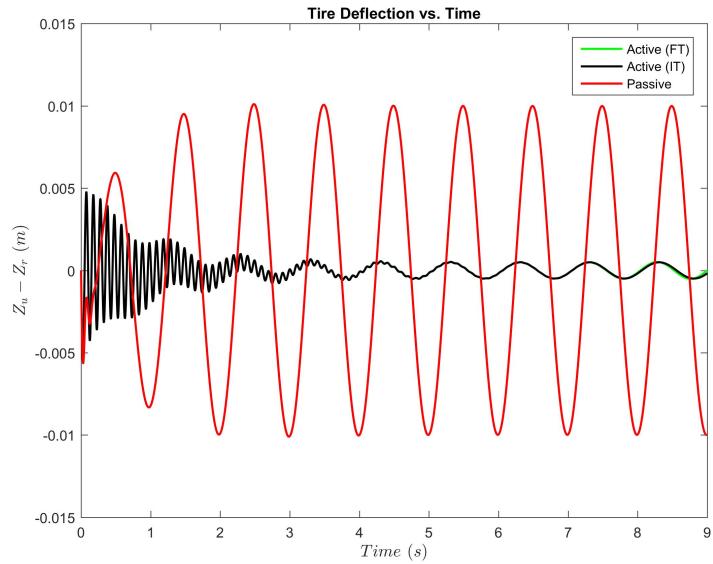


Figure 13: Tire deflection for dynamic case with active LQR control.

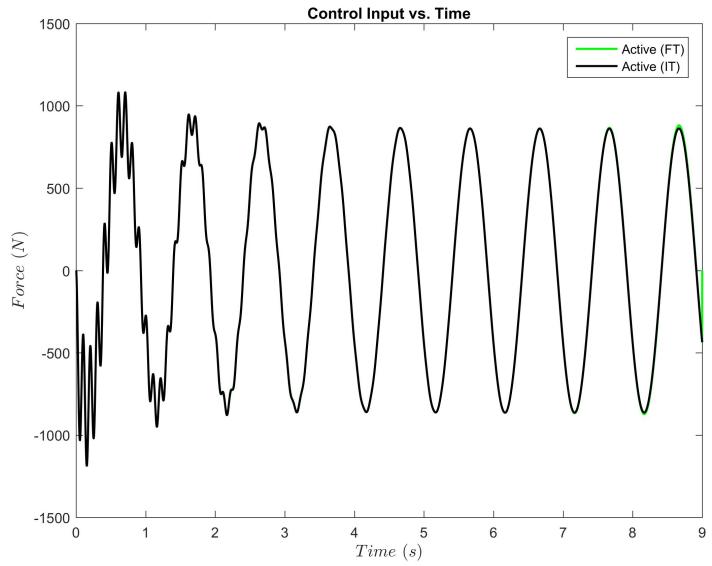


Figure 14: Control inputs for the two LQR controllers for the dynamic case.

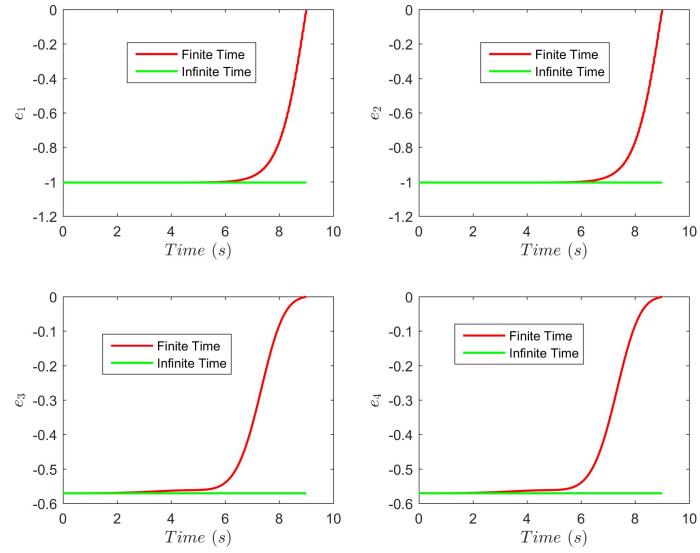


Figure 15: Real parts of the closed loop eigenvalues for finite vs. infinite time LQR controllers.

LQ Tracker (Finite and Infinite Horizon)

In automotive applications, although the most common applications of closed loop controllers would be the LQR, since the states are desired to be driven to zero, in some special cases, it may be required to drive a given state along a desired trajectory, or to a given non-zero set point. To ensure this, the cost function is modified as follows:

$$J = \frac{1}{2} \int_0^T \left((\mathbf{x} - \mathbf{r})^T Q (\mathbf{x} - \mathbf{r}) + 2(\mathbf{x} - \mathbf{r})^T N F_a + F_a^T R F_a \right) dt \quad (17)$$

In this case, \mathbf{r} is the reference trajectory to be tracked. For the simulations, the chosen reference trajectory is $[r_1(t), 0, 0, 0]^T$, i.e., the first state is expected to follow the trajectory r_1 . For this application, the weight of the first state in the cost function is increased to a large number ($\rho_1 = 100000$) to ensure tracking.

For the tracker, the state, co-state and stationarity equations are the same as derived in Part I. The matrix Riccati equation is also derived in similar fashion as in the LQR case,

$$\begin{aligned} \dot{S} = & -S(A - BR^{-1}N^T) - (A - BR^{-1}N^T)^T S - (Q - NR^{-1}N^T) \\ & + SBR^{-1}B^T S \end{aligned} \quad (18)$$

where $S(T) = 0$ for the chosen cost function. The differential matrix Riccati equation is used for the finite time horizon case.

The difference with the LQR case is in the Sweep method, where the sweep assumption is as follows:

$$\boldsymbol{\lambda} = S\mathbf{x} - \boldsymbol{\nu} \quad (19)$$

Substituting equation 19 in the co-state equation, we can derive a differential equation for the vector $\boldsymbol{\nu}$ as follows:

$$\begin{aligned} \dot{\boldsymbol{\nu}} = & SBR^{-1}B^T \boldsymbol{\nu} + SBR^{-1}N^T \mathbf{r} + SL\dot{z}_r - A^T \boldsymbol{\nu} - QR \\ & + NR^{-1}B^T \boldsymbol{\nu} + NR^{-1}N^T \mathbf{r} \end{aligned} \quad (20)$$

This differential equation is also integrated backwards, and for the given cost function, the boundary conditions are defined as $\boldsymbol{\nu} = 0$.

The control law for the tracker can then be written as:

$$F_a = -R^{-1}B^T(S\mathbf{x} - \boldsymbol{\nu}) - R^{-1}N^T \mathbf{x} \quad (21)$$

The block diagram for the finite time horizon LQT controller is shown in Fig. 16.

The variable of interest (suspension deflection) is shown against the trajectory for both the static and the dynamic cases in Fig. 17–18. It can be seen that in both cases the controller makes the suspension deflection track the desired trajectory. The control inputs for the two cases are shown in Fig. 19–20.

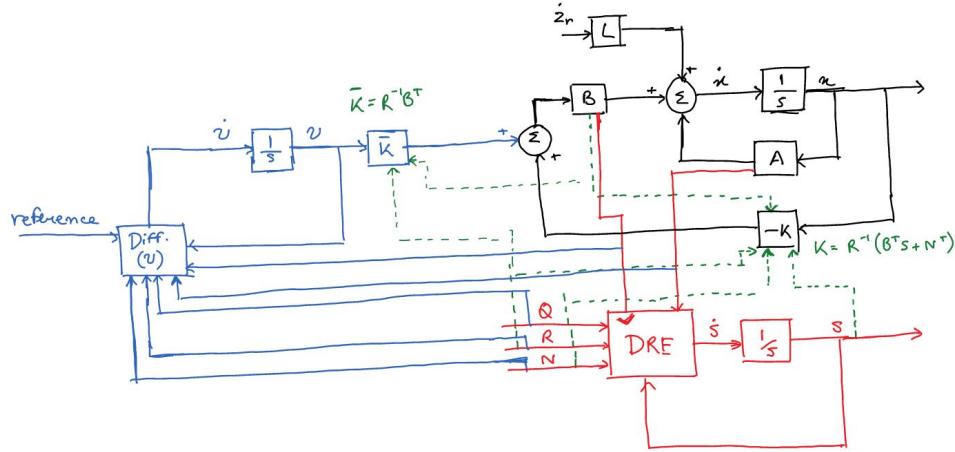


Figure 16: Block diagram for finite time LQT control.

The tracker equations can also be used to design the system to track a non-zero set point. In the case of an infinite time horizon controller, the Riccati equation and the equations for ν can be treated as algebraic equations. The performance of the finite and infinite horizon trackers with non-zero set points ($x_1 = 0.05$) are compared in Fig. 21–22. If the time horizon is short ($T = 1$), it can be seen that the finite time tracker is faster than the infinite time as expected. However, for a longer time horizon ($T = 3$), the two tracker performances are indistinguishable. The control plots are also shown in Fig. 23–24. The evolution of ν with time for both the cases are shown in Fig. 25–26. It can be seen that for the finite time case, the ν vector starts off at the same values as in the infinite case as expected.

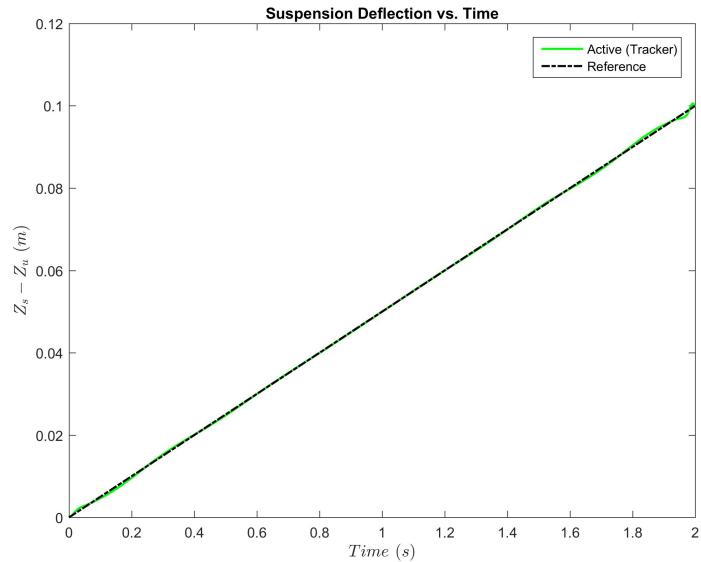


Figure 17: Suspension deflection for static case with active LQT control.

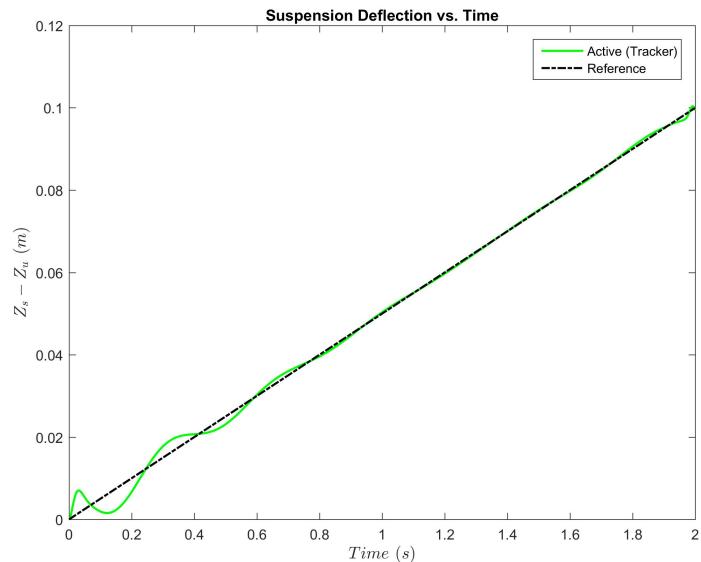


Figure 18: Suspension deflection for dynamic case with active LQT control.

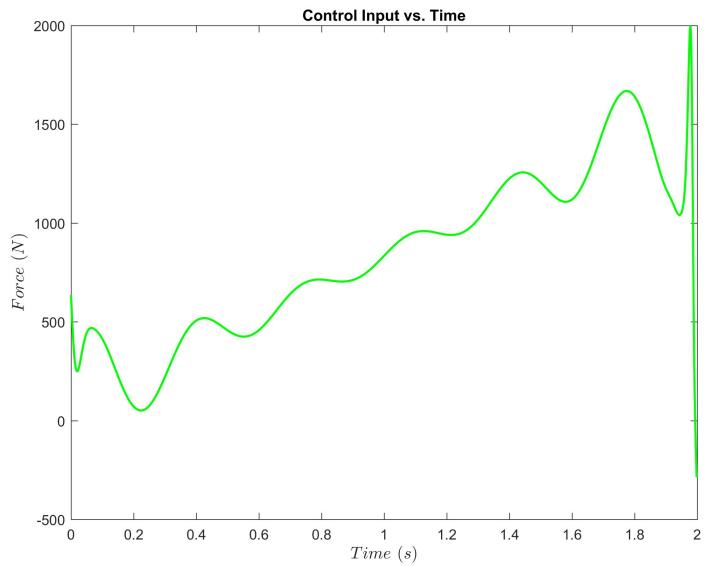


Figure 19: Control input for LQT controller in the static case.

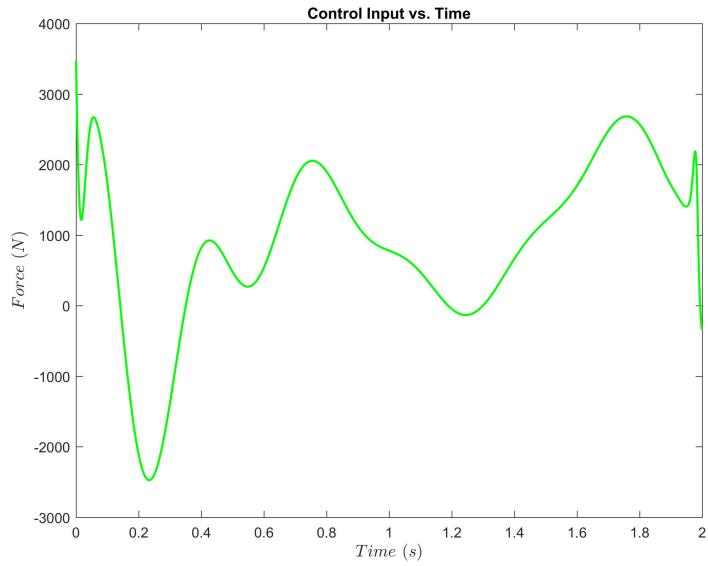


Figure 20: Control inputs for the LQT controller in the dynamic case.

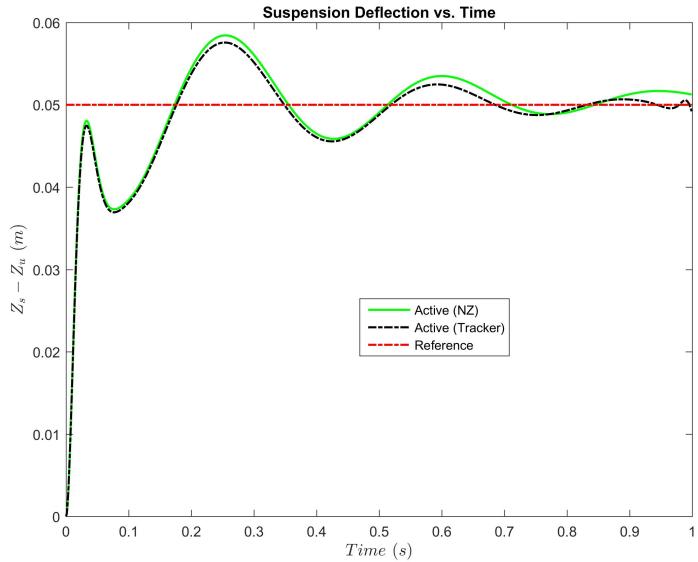


Figure 21: Suspension deflection for static case with active LQT control ($T = 1$).

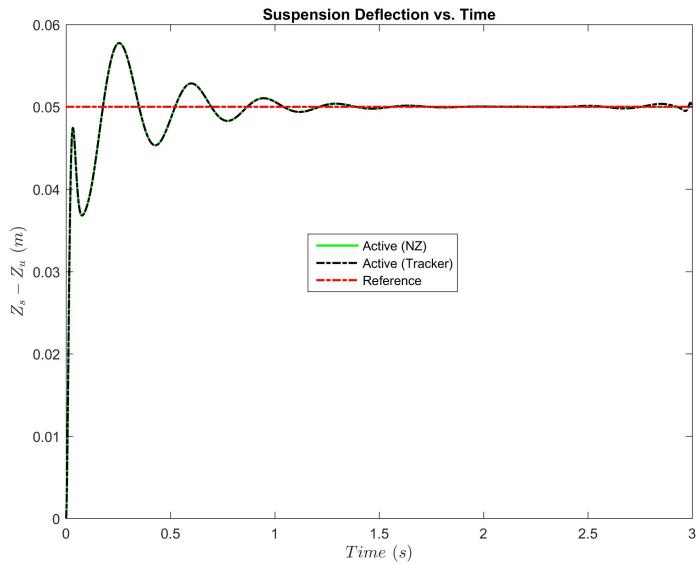


Figure 22: Suspension deflection for static case with active LQT control ($T = 3$).

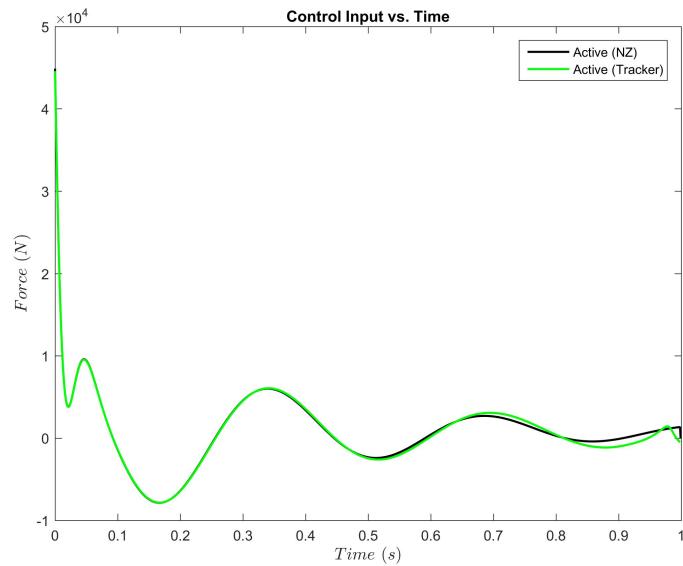


Figure 23: Control input for active LQT control ($T = 1$).

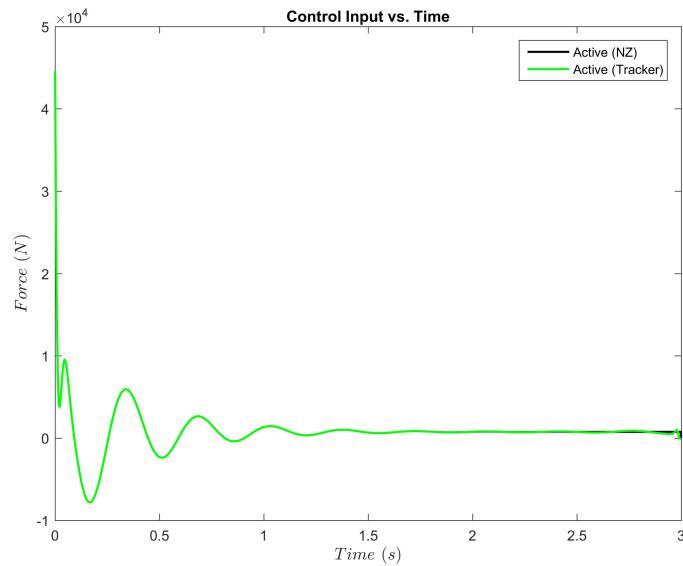


Figure 24: Control input for active LQT control ($T = 3$).

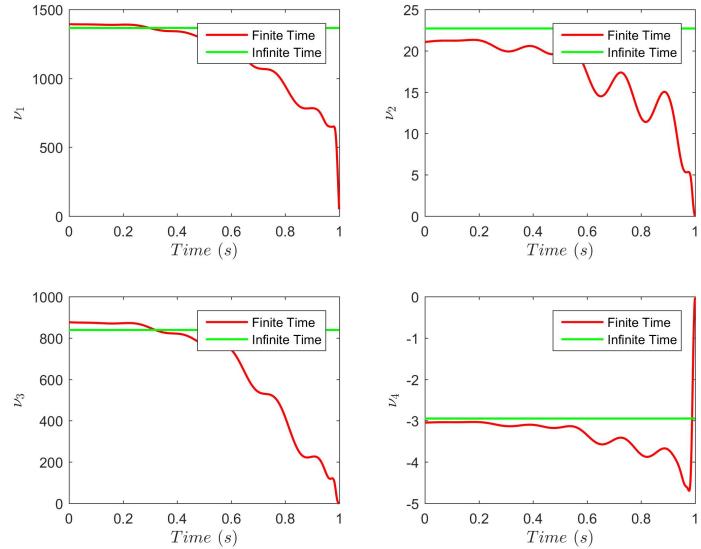


Figure 25: Evolution of ν with time for active LQT control ($T = 1$).

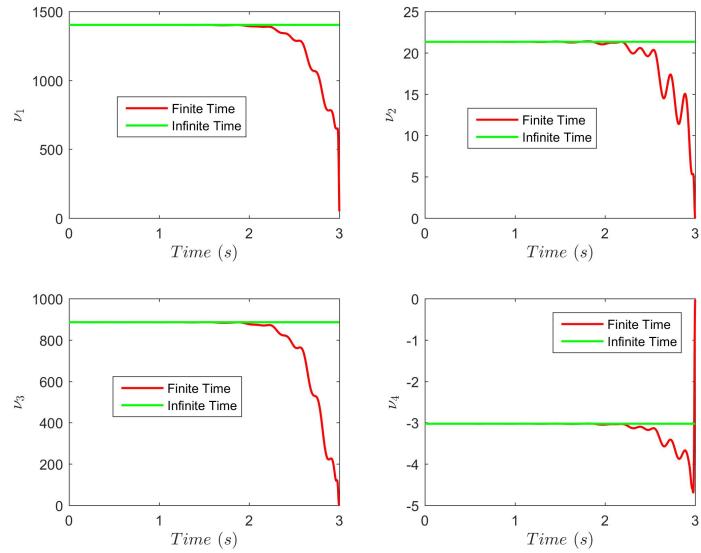


Figure 26: Evolution of ν with time for active LQT control ($T = 3$).

Regulator with Final State Fixed (Finite Horizon)

A regulator with final state fixed type controller is designed to drive some of the states of the system to predefined fixed values at the final time T, while the others to zero. An optimal regulator with final state fixed for the current system can be designed by optimizing it with respect to the same cost function defined earlier with an added constraint of reaching the fixed state at final time T. In the current work, the final state fixed constraint is assumed only for the first state. Cost function with the added constraint can be given as

$$J = \frac{1}{2} \int_0^T (\mathbf{x}^T Q \mathbf{x} + 2\mathbf{x}^T N F_a + F_a^T R F_a) dt \quad (22)$$

$$\psi(x(T), T) = C\mathbf{x}(T) - \mathbf{r}(T) \quad (23)$$

$$= 0 \quad (24)$$

where $\mathbf{r}(T) = [r(T), 0, 0, 0]^T$.

The augmented cost function can be defined as

$$J* = \nu^T \psi(\mathbf{x}(T), T) + \int_0^T H dt \quad (25)$$

where H being the Pontryagin function defined previously. Terminal conditions at the final time T are given as

$$\{\psi_x^T \nu - \lambda\}^T |_T d\mathbf{x}(T) + \{\psi_t^T \nu + H\}_T dT = 0 \quad (26)$$

This translates into the boundary conditions for the co-states as

$$\boldsymbol{\lambda} = C^T \nu \quad (27)$$

Using sweep method, co-states of the system can be defined as

$$\boldsymbol{\lambda} = S\mathbf{x} + V\nu \quad (28)$$

Solving for the states and co-states, the resultant matrix Riccati equations are given as:

$$\begin{aligned} \dot{S} &= -S(A - BR^{-1}N^T) - (A - BR^{-1}N^T)^T S - (Q - NR^{-1}N^T) \\ &\quad + SBR^{-1}B^T S \end{aligned} \quad (29)$$

$$\dot{V} = -(A - BK)^T V \quad (30)$$

where $K = R^{-1}(B^T S + N^T)$.

Lagrange multiplier for the fixed final state constraint ν can be obtained by assuming $\mathbf{r}(T)$ as a linear combination of $\mathbf{x}(t)$ and ν for all time, i.e.,

$$\mathbf{r}(T) = C\mathbf{x}(T) = U(t)\mathbf{x}(t) + P(t)\nu \quad (31)$$

where $U(t)$ and $P(t)$ can be obtained by:

$$\dot{U} = -U(A - BK) \quad U(T) = V^T(T) \quad (32)$$

$$\dot{P} = V^T BR^{-1}B^T V \quad P(T) = 0 \quad (33)$$

The value of ν can then be given as:

$$\nu = P^{-1}(r(T) - U(t)\mathbf{x}(\mathbf{t})) \quad (34)$$

Hence, the complete control input to the system is given as

$$u = -(K - R^{-1}B^T VP^{-1}V^T)\mathbf{x} - R^{-1}B^T VP^{-1}r(T) \quad (35)$$

Results for the LQR with final state fixed constraints are shown in the following (Fig. 27 – 29). Final time value for the first state is considered to be fixed at $x_1 = 0.05$. It can be seen from the plot (fig 27) that the controlled system tries to reach to a steady state towards the middle part of the plot (for time $\in [5 - 7]$ steps) as it should under a finite time LQR controller. But, due to the final state fixed constraint, the system is forced to reach the fixed point $x_1 = 0.05$ at the final time $T = 10$. A zoomed in version of the system behavior near the final time is shown in figure 28. Sprung mass acceleration reaches to a steady state quickly under a heavy cost weight, however, near the final time step it is forced to oscillate to make sure that the final state fixed constraint is satisfied (fig 29). The evolution of the parameters V and P with time are shown in Fig. 30 and Fig. 31 respectively.

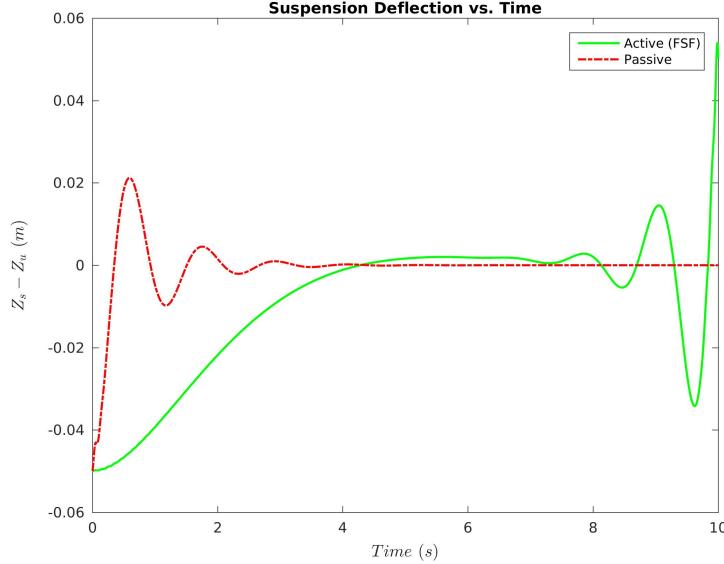


Figure 27: Suspension deflection for static case with first state $x_1 := z_s - z_u$ fixed at 0.05 at final time.

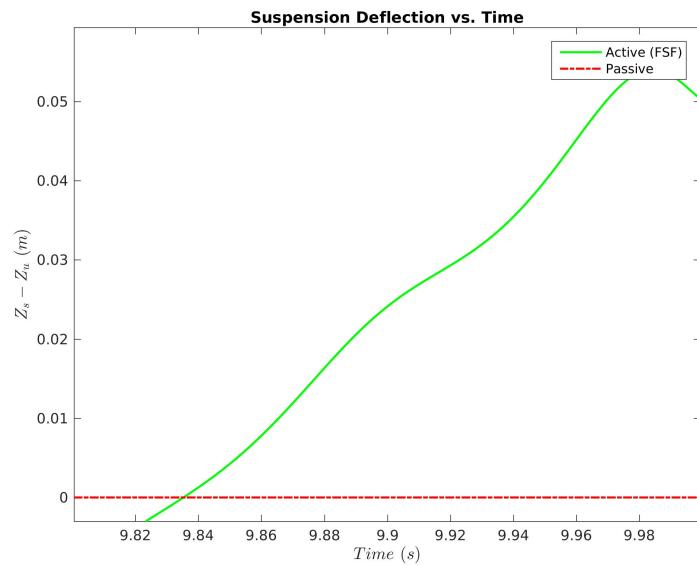


Figure 28: Zoomed plot to show convergence of first state at final time.

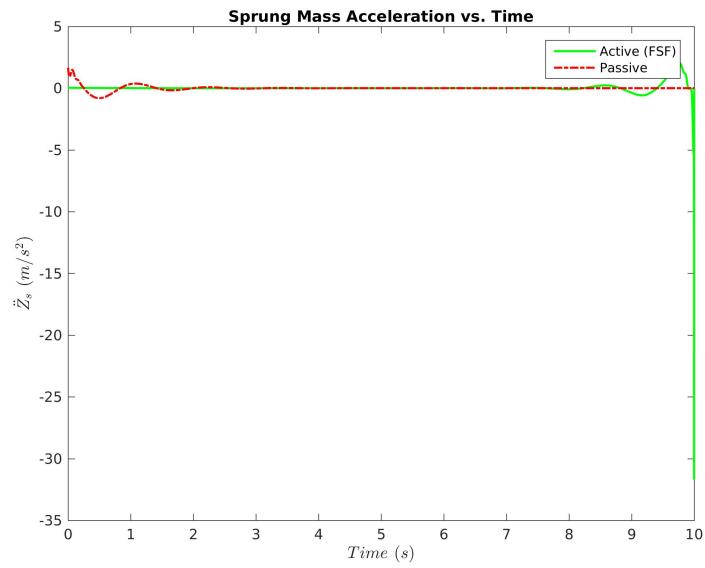


Figure 29: Sprung mass acceleration for static case with first state $x_1 := z_s - z_u$ fixed at 0.05 at final time.

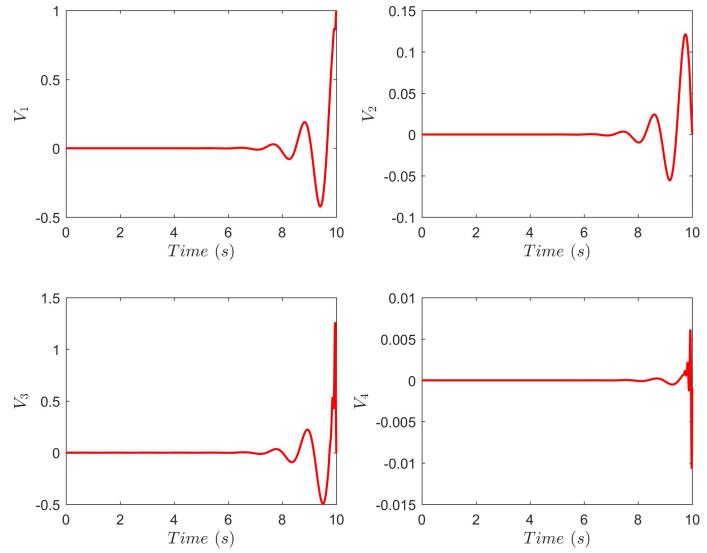


Figure 30: Evolution of V with time.

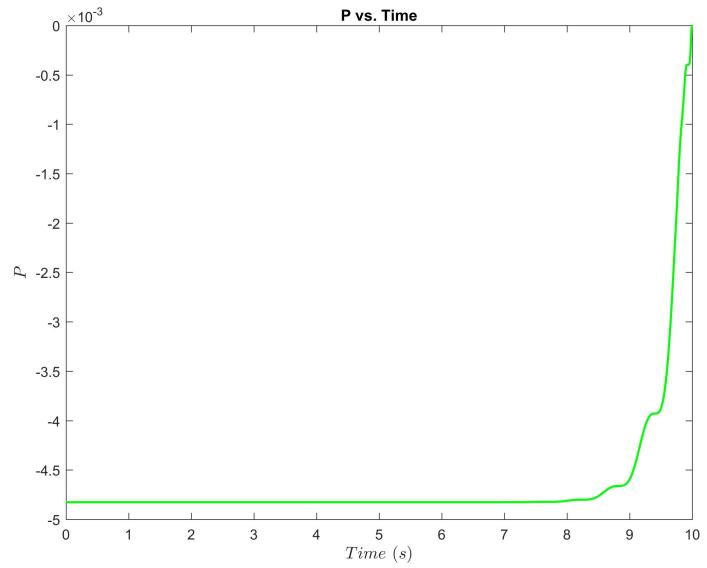


Figure 31: Evolution of P with time.

Reduced Order Observer

States of the system represent suspension deflection, sprung mass velocity, tire deflection and unsprung mass velocity. In a real system, out of these four states, only two of them, namely suspension deflection and sprung mass velocity $\{x_1, x_2\}$, are measurable. Hence, for designing a closed loop controller for the system, we need to estimate the other two states namely x_3 and x_4 . A reduced order observer has been designed to estimate these states. It is derived as following.

Observability matrix for the system is defined as

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

System dynamics matrices are segmented into two parts: one corresponding to the measured states and the other for the estimated states.

$$\dot{\mathbf{y}}_1 = A_{11}\mathbf{y}_1 + A_{12}\mathbf{y}_2 + B_1\mathbf{u} \quad (36)$$

$$\dot{\mathbf{y}}_2 = A_{21}\mathbf{y}_1 + A_{22}\mathbf{y}_2 + B_2\mathbf{u} + L_2\dot{z}_r \quad (37)$$

$$\mathbf{z} = C_1\mathbf{y}_1 \quad (38)$$

where the output vectors are $\mathbf{y}_1 = [y_1, y_2]^T$ and $\mathbf{y}_2 = [y_3, y_4]^T$ and the corresponding matrices are given as

$$A_{11} = \begin{bmatrix} 0 & 1 \\ -\frac{k_s}{m_s} & -\frac{b_s}{m_s} \end{bmatrix} \quad A_{12} = \begin{bmatrix} 0 & -1 \\ 0 & \frac{b_s}{m_s} \end{bmatrix} \quad (39)$$

$$A_{21} = \begin{bmatrix} 0 & 0 \\ \frac{k_s}{m_u} & \frac{b_s}{m_u} \end{bmatrix} \quad A_{22} = \begin{bmatrix} 0 & 1 \\ -\frac{k_t}{m_u} & -\frac{b_s}{m_u} \end{bmatrix} \quad (40)$$

$$B_1 = \begin{bmatrix} 0 & \frac{1}{m_s} \end{bmatrix}^T \quad B_2 = \begin{bmatrix} 0 & -\frac{1}{m_u} \end{bmatrix}^T \quad (41)$$

$$C_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (42)$$

$$L_2 = [-1 \quad 0]^T \quad (43)$$

The observer model is assumed as

$$\hat{\mathbf{y}}_2 = M\mathbf{z} + \mathbf{w} \quad (44)$$

where M is a square (2×2) matrix and

$$\dot{\mathbf{w}} = F\mathbf{w} + G\mathbf{z} + H\mathbf{u} \quad (45)$$

Defining estimator error as $\tilde{\mathbf{y}} = \mathbf{y} - \hat{\mathbf{y}}$, error propagation equation are given as

$$\dot{\tilde{\mathbf{y}}}_2 = \dot{\mathbf{y}}_2 - \dot{\hat{\mathbf{y}}}_2 = (A_{21}\mathbf{y}_1 + A_{22}\mathbf{y}_2 + B_2\mathbf{u} + L_2\dot{z}_r) - (M\dot{\mathbf{z}} + \dot{\mathbf{w}}) \quad (46)$$

Substituting \hat{y} , $\dot{\mathbf{y}}$, $\dot{\mathbf{z}}$ and $\dot{\mathbf{w}}$ using equations 36, 38, 44 and 45 in 46 and rearranging it to obtain

$$\begin{aligned}\dot{\tilde{\mathbf{y}}}_2 = & F\tilde{\mathbf{y}}_2 + (A_{21} - MC_1A_{11} + FMC_1 - GC_1)\mathbf{y}_1 + \\ & (A_{22} - MC_1A_{12} - F)\mathbf{y}_2 + (B_2 - MC_1B_1 - H)\mathbf{u} + L_2\dot{z}_r\end{aligned}$$

The error can be made independent of \mathbf{y}_1 , \mathbf{y}_2 and \mathbf{u} by making their coefficients to go to zeros.

$$y_2 : \quad F = A_{22} - MC_1A_{12} \quad (47)$$

$$u : \quad H = B_2 - MC_1B_1 \quad (48)$$

$$y_1 : \quad G = (A_{21} - MC_1A_{11})C_1^{-1} + FM \quad (49)$$

For observer design, we can consider input from the road profile \mathbf{z}_r as a known disturbance signal and cancel it by applying an additional constant input $u_\delta = -L\mathbf{z}_r$. Hence, the complete control input to the system is $u_{Total} = \mathbf{u} + u_\delta$

Hence, the net error propagation equation reduces to

$$\dot{\tilde{\mathbf{y}}}_2 = F\tilde{\mathbf{y}}_2 \quad (50)$$

A fast convergence of the state estimates $\hat{\mathbf{y}}_2$ to the actual measurements \mathbf{y} can be achieved by choosing appropriate eigenvalues for the error propagation matrix F . The eigenvalues of F are chosen to be 20 times the closed loop eigenvalues for the system, to make sure that the estimation errors go to zero 20 times faster than the system states. The closed loop eigenvalues for the system are $[-1.0048 + 62.3579i, -1.0048 - 62.3579i, -0.5707 + 0.5538i, -0.5707 - 0.5538i]^T$. Hence, the eigenvalues for F are chosen as $[-20.096, -20.096]$. A block diagram showing the system with the designed reduced order observer is shown in Fig. 32.

System response under full state feedback and the designed observer are compared for different types of controller designs namely infinite horizon LQR, finite horizon LQR and finite time LQR with fixed final state. System responses are shown for the two input signals: static and dynamic road profile. Results for the system performance using infinite LQR controllers are shown in Fig. 33 – 42, while the results with a finite controller are shown in Fig. 43 – 52. The results for the finite time LQR with fixed final states are shown in Fig. 53 – 58. Initially, due to the state estimation error, the system response with the observer is oscillatory. However, as it can be seen from the plots, the estimation error goes to zero much more faster than the convergence of system states to the desired behavior suggesting an effective observer design.

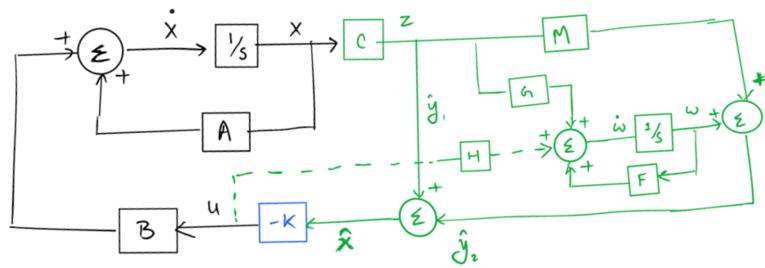


Figure 32: Block diagram showing the reduced order observer for the system

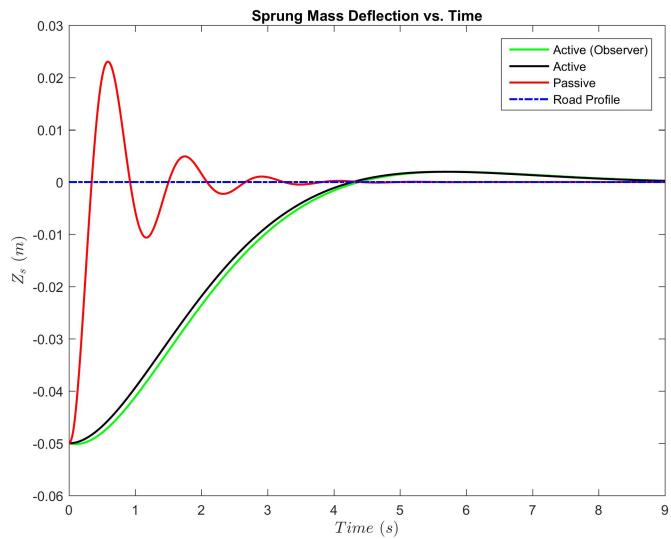


Figure 33: Sprung mass deflection for static case using reduced order observer with infinite time LQR.

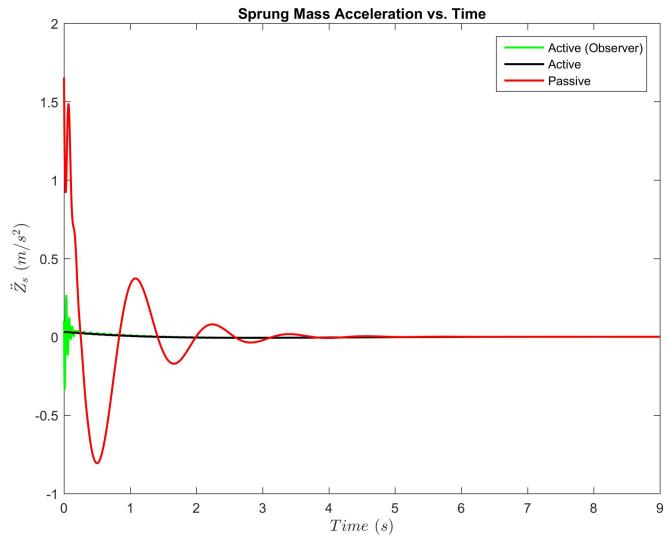


Figure 34: Sprung mass acceleration for static case using reduced order observer with infinite time LQR.

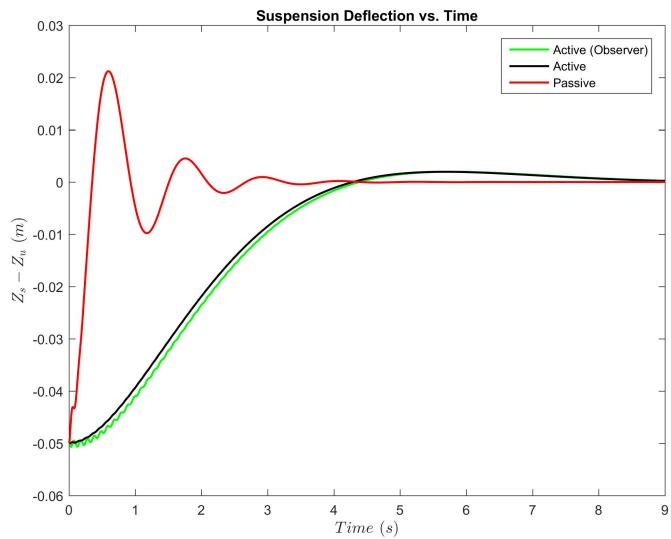


Figure 35: Suspension deflection for static case using reduced order observer with infinite time LQR.

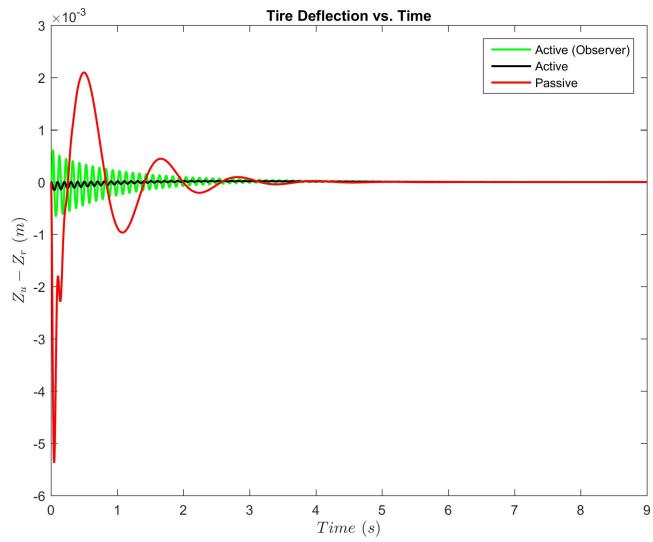


Figure 36: Tire deflection for static case using reduced order observer with infinite time LQR.

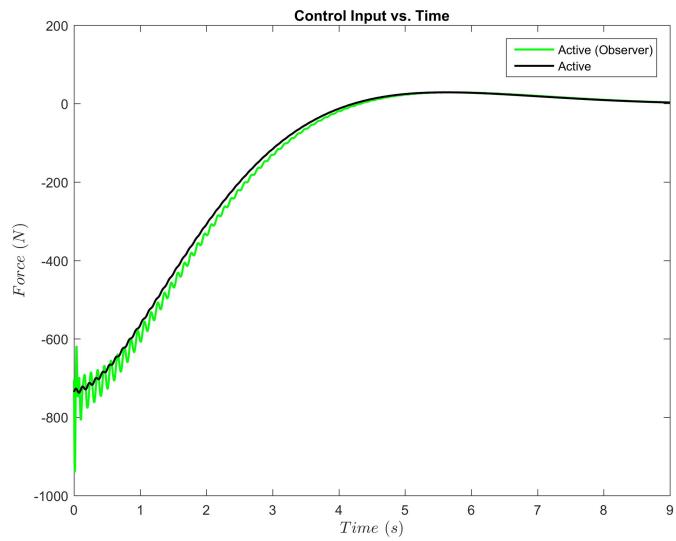


Figure 37: Control input for static case using reduced order observer with infinite time LQR.

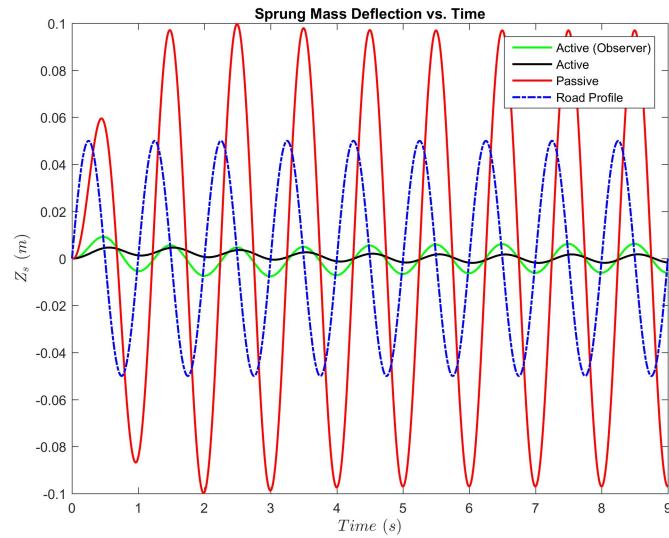


Figure 38: Sprung mass deflection for dynamic case using reduced order observer with infinite time LQR.

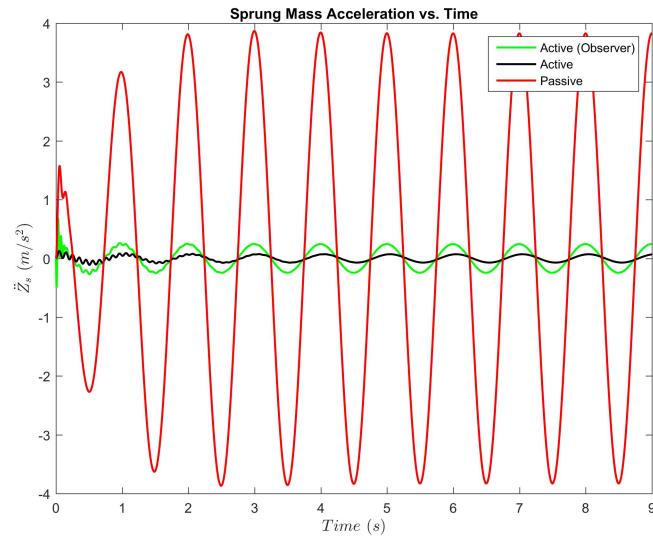


Figure 39: Sprung mass acceleration for dynamic case using reduced order observer with infinite time LQR.

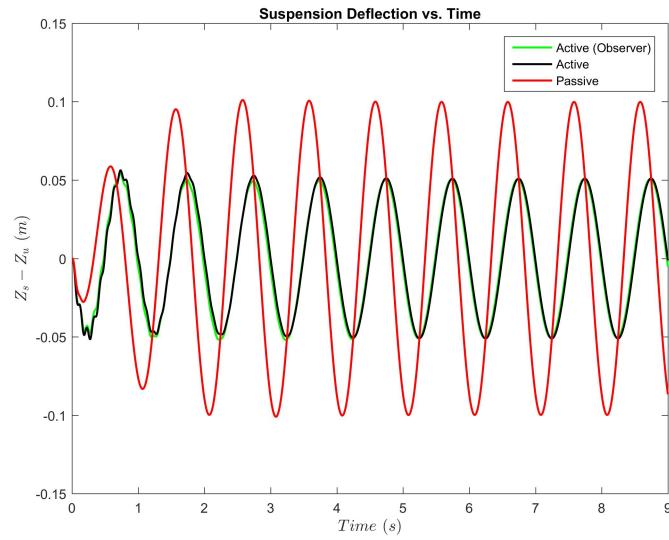


Figure 40: Suspension deflection for dynamic case using reduced order observer with infinite time LQR.

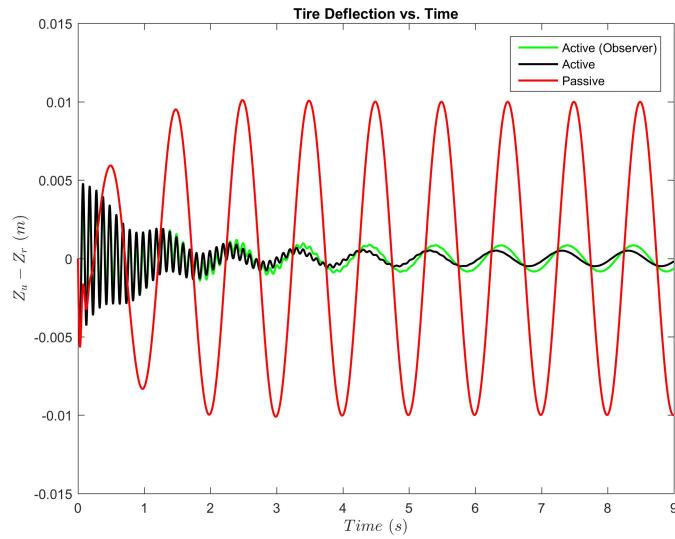


Figure 41: Tire deflection for dynamic case using reduced order observer with infinite time LQR.

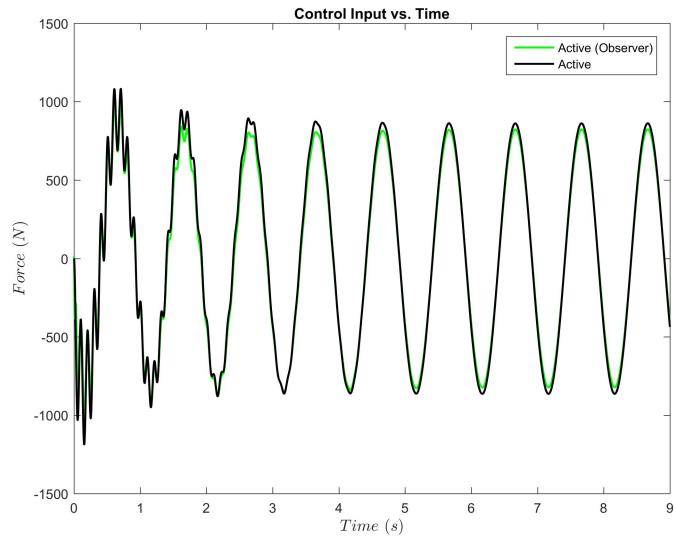


Figure 42: Control input for dynamic case using reduced order observer with infinite time LQR.

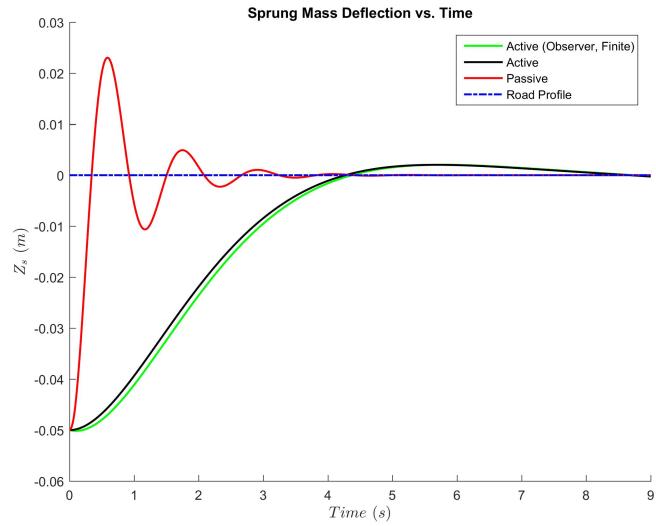


Figure 43: Sprung mass deflection for static case using reduced order observer with finite time LQR.

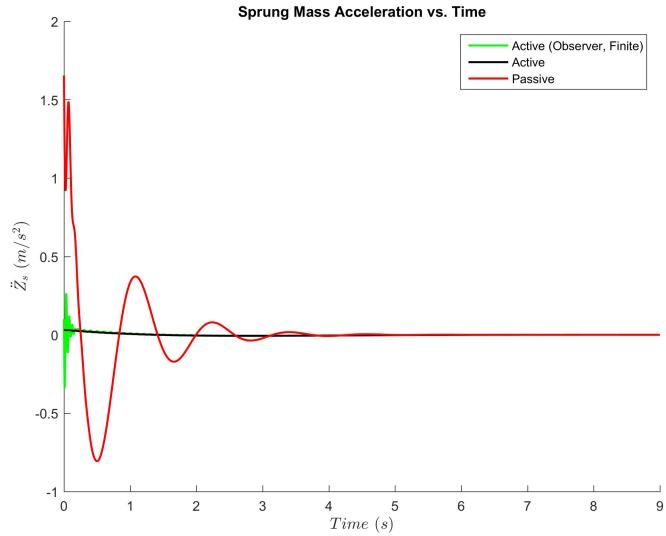


Figure 44: Sprung mass acceleration for static case using reduced order observer with finite time LQR.

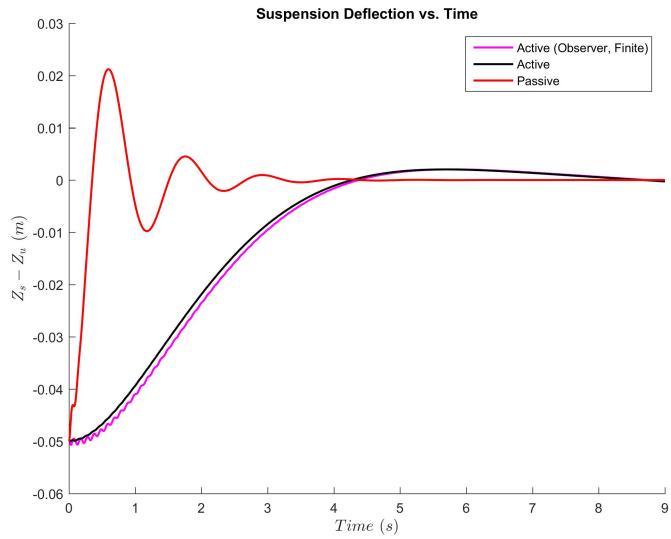


Figure 45: Suspension deflection for static case using reduced order observer with finite time LQR.

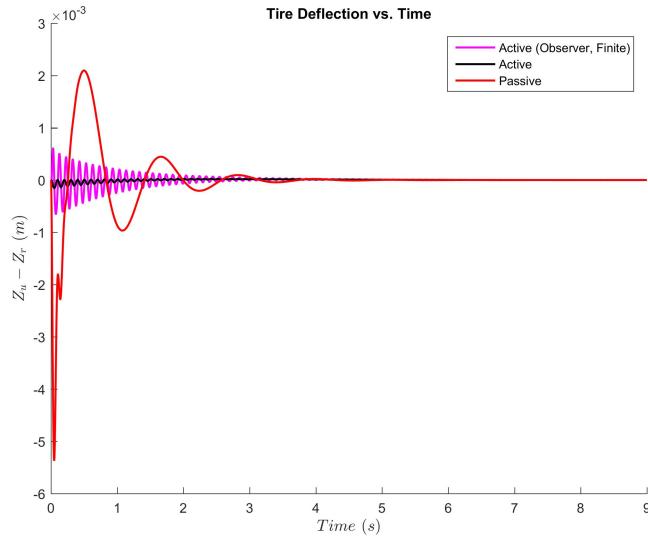


Figure 46: Tire deflection for static case using reduced order observer with finite time LQR.

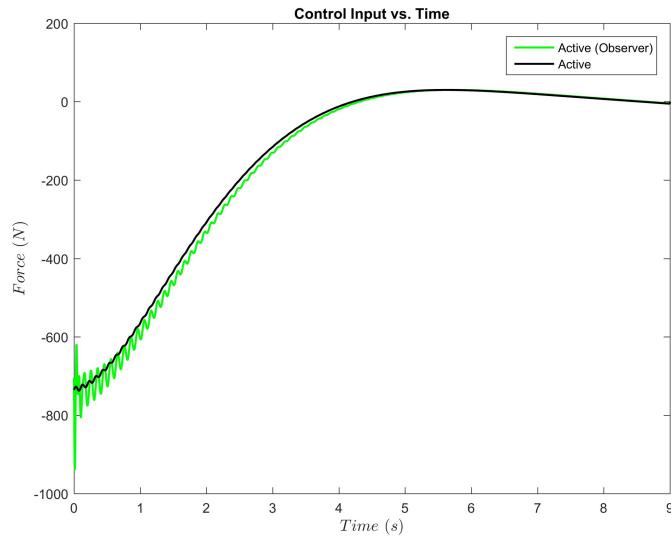


Figure 47: Control input for static case using reduced order observer with finite time LQR.

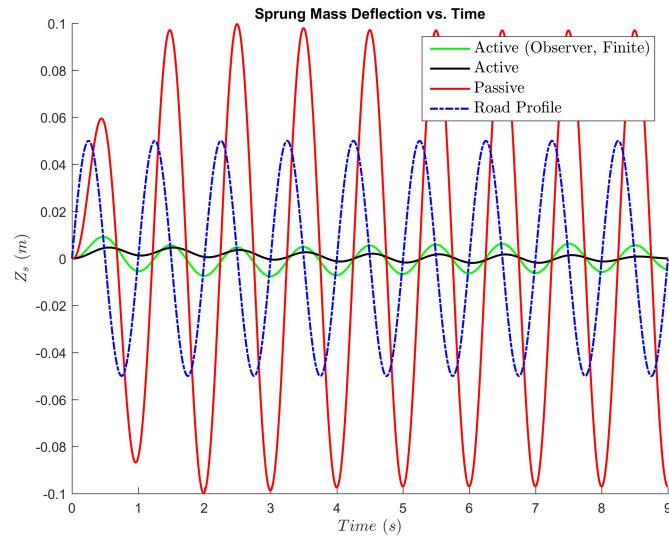


Figure 48: Sprung mass deflection for dynamic case using reduced order observer with finite time LQR.

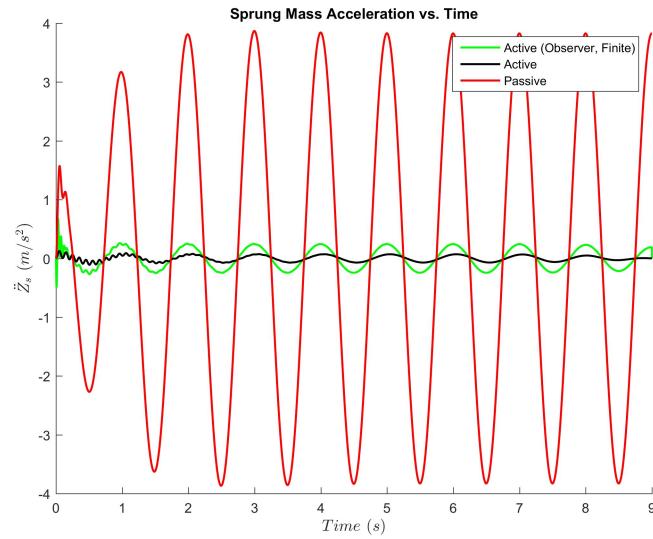


Figure 49: Sprung mass acceleration for dynamic case using reduced order observer with finite time LQR.

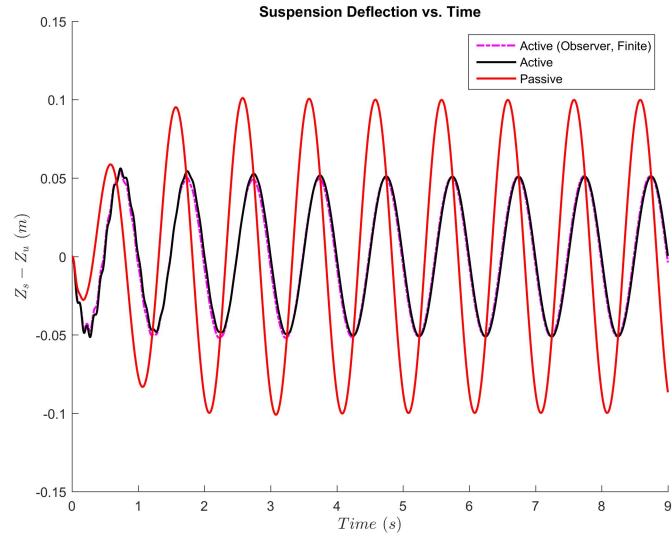


Figure 50: Suspension deflection for dynamic case using reduced order observer with finite time LQR.

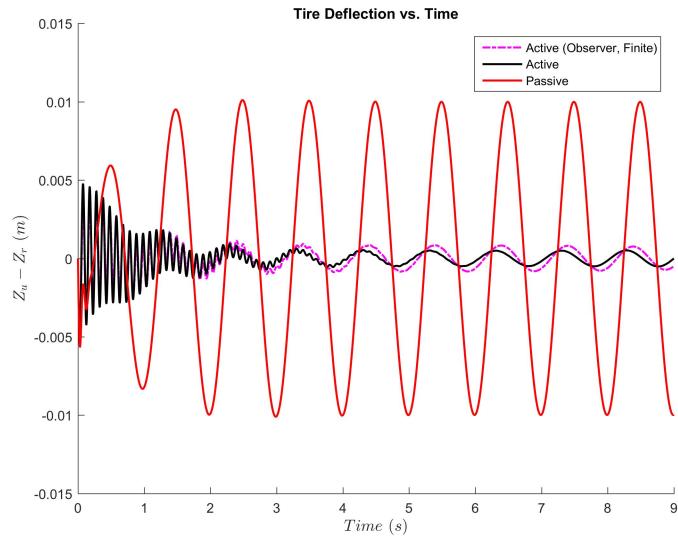


Figure 51: Tire deflection for dynamic case using reduced order observer with finite time LQR.

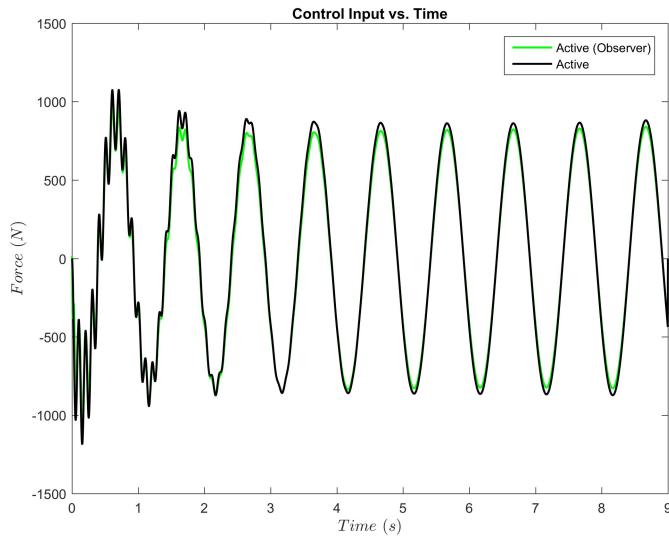


Figure 52: Control input for dynamic case using reduced order observer with finite time LQR.

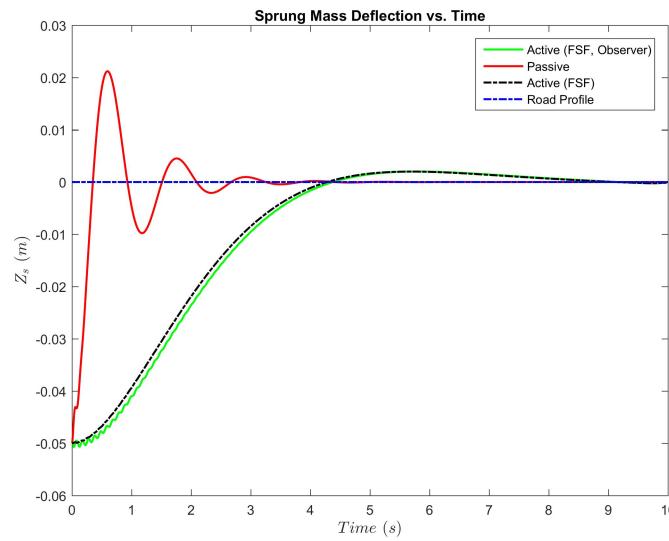


Figure 53: Sprung mass deflection for static case using reduced order observer with final state fixed LQR.

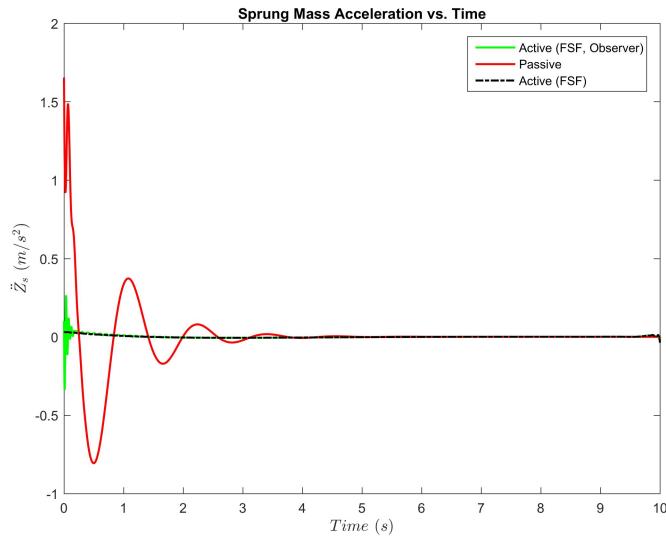


Figure 54: Sprung mass acceleration for static case using reduced order observer with final state fixed LQR.

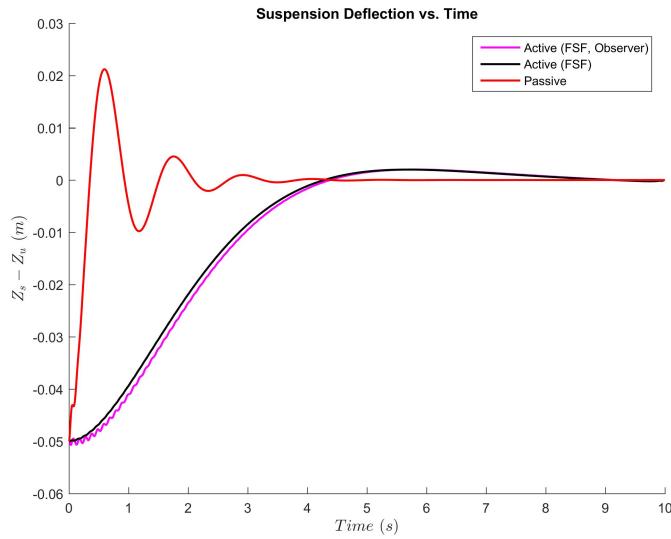


Figure 55: Suspension deflection for static case using reduced order observer with final state fixed LQR.

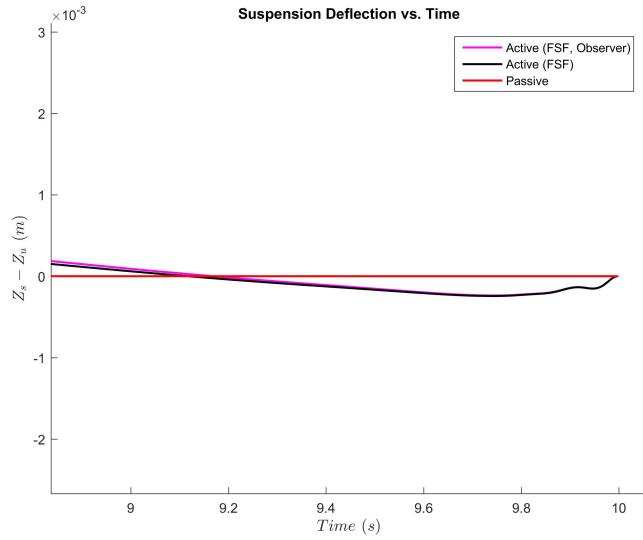


Figure 56: Zoomed in on the suspension deflection plot for static case using reduced order observer with final state fixed LQR.

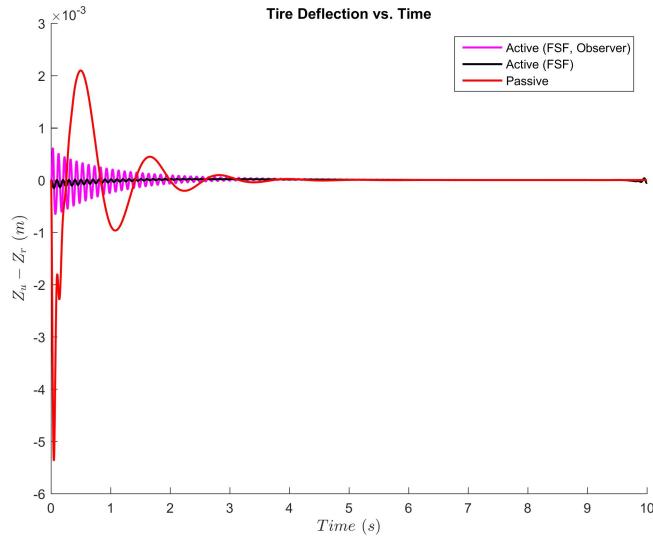


Figure 57: Tire deflection for static case using reduced order observer with final state fixed LQR.

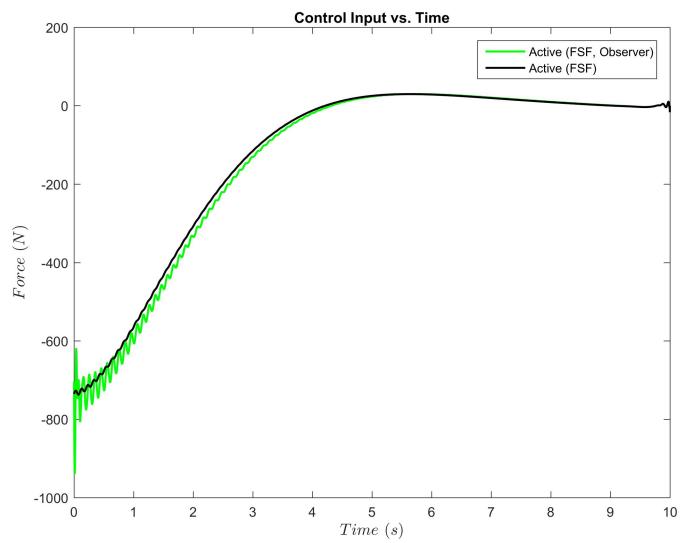


Figure 58: Control input for static case using reduced order observer with final state fixed LQR.

References

- [1] Rajamani, Rajesh, *Vehicle Dynamics and Control*. Mechanical Engineering Series, Springer Science & Business Media, 2011.
- [2] Wong, J. Y., *Theory of Ground Vehicles*. John Wiley & Sons, 2001 (3rd edition).
- [3] Lewis, F. L. and Syrmos, V. L., *Optimal Control*. John Wiley & Sons, 1995.