Application of Modern Control to Dynamic Maglev System

Isaac Tan[1], N9960210

***Abstract* — The purpose of this report is to investigate a dynamic maglev system and develop a complete output-feedback controller that drives the system to a set of desired equilibrium points. The report outlines how the system is modelled, linearised and controlled. The impact of LQR controllers is discussed and how adjusting the weight of constants effects the rest of the system. It also covers how a controller that was initially designed to regulate state-feedback was adapted to work in black-box environments where states are required to be estimated and adjusted according to error. The implementation of the optimised controller is then tested with different initial conditions over different time periods to test the robustness and validity of outputs.**

I. Introduction

As modern problems arise, society is constantly finding novel ways to implement technology and overcome challenges. One such problem is the industry of mass travel, which with increasing population and city area size requires a solution that can transport many people over a long distance at a high speed. One suggested solution to this is the use of magnetic levitation (maglev) trains. Maglev is a form of levitation in which an object is hovered using nothing but magnetic fields. Traditionally used in gyroscopes and accelerometers, it has been proposed for use in high-speed trains due to the reduced friction force, high acceleration and deceleration speed, and decreased maintenance from lessened wear on the track [1]. Fundamentally, the system works by passing an input current through a coil that creates a magnetic field and repels the train vertically from the track. The inductance of the coil and strength at which it repels is dependent on the magnitude of the input current.

Use of such technology in a high-speed train requires a very accurate and robust control system. This report outlines the use of a dynamic control system that mimics a rudimentary maglev train scenario. The aim of the system was to keep an object suspended at a chosen height while reducing the vertical velocity of the object and reducing the external input voltage. The system was described using state-space models which were then optimally controlled via state-space feedback, then with output feedback.

II. Modelling the System

The system model was reproduced from the system found in [2] where a ball of magnetic material was hovered using an electromagnet and the ball's velocity was optically measured. Fig. 1.1. shows a snippet taken from the source of the basic design of the system.

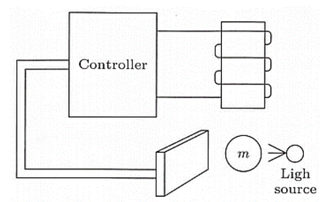

Fig. 1.1 High-level system design [2]

The system states were provided as:

$$\dot{x}_1 = x_2 \qquad (1.)$$

$$\dot{x}_2 = -bx_2 + 1 - \frac{4cx_3^2}{(1+x_1)^2} \qquad (2.)$$

$$\dot{x}_3 = \frac{1}{\beta[\alpha + \frac{1}{1+x_1}]}\left[-x_3 + u + \frac{\beta x_2 x_3}{(1+x_1)^2}\right] \qquad (3.)$$

$where:$

$$x_1 = \frac{y}{a}, \quad x_2 = \frac{1}{a\lambda}\frac{dy}{dt}, \quad x_3 = \frac{i}{I} \qquad (4.)$$

The values for the constants used in equations (1.), (2.), (3.) were provided or solved for using the provided values and equations. A table of constants used throughout this report is appended in appendix I.

To control the system to a desired equilibrium where the states are stationary, values of x were found such that equations (1.), (2.), (3.) would equal zero. This resulted in the vector:

$$\bar{x} = \begin{bmatrix} \overline{x_1} \\ \overline{x_2} \\ \overline{x_3} \end{bmatrix} = \begin{bmatrix} \overline{x_2} = 0 \\ -b\overline{x_2} + 1 - \dfrac{4c\overline{x_3^2}}{(1+\overline{x_1})^2} = 0 \\ \dfrac{1}{\beta\left[\alpha + \dfrac{1}{1+\overline{x_1}}\right]}\left[-\overline{x_3} + \bar{u} + \dfrac{\beta\overline{x_2 x_3}}{(1+\overline{x_1})^2}\right] = 0 \end{bmatrix}$$

For these states to hold, $\overline{x_1}$ must equal $2\bar{u} - 1$, $\overline{x_2}$ must equal 0, and $\overline{x_3}$ must equal $\bar{u}$. This gives the systems equilibrium points in terms of the states x and input u.

$$\bar{x} = \begin{bmatrix} 2\bar{u} - 1 \\ 0 \\ \bar{u} \end{bmatrix} \qquad (5.)$$

[1] **Name** is with Faculty of Engineering, Queensland University of Technology (QUT), Australia. This report is in partial fulfilment of EGH445 unit assessment requirements and **submitted on 23/05/2022.** **isaac.tan@connect.edu.au**

To design a state-feedback controller for the system it first had to be linearised. The linearised system can be denoted as $\dot{\tilde{x}} = A\tilde{x} + B\tilde{u}$ where A is a square matrix the length of x and B is a column vector the height of x. The A and B matrices are calculated using the Jacobian of the state space equations f with respect to the state x and the input u respectively. This can be denoted as:

$$A = \left(\frac{\partial f}{\partial x}\right)^T_{|x=\bar{x},F=\bar{F}}, B = \left(\frac{\partial f}{\partial F}\right)^T_{|x=\bar{x},F=\bar{F}}$$

which can be expanded to:

$$A = \begin{bmatrix} \frac{\partial \dot{x_1}}{\partial \overline{x_1}} & \frac{\partial \dot{x_1}}{\partial \overline{x_2}} & \frac{\partial \dot{x_1}}{\partial \overline{x_3}} \\ \frac{\partial \dot{x_2}}{\partial \overline{x_1}} & \frac{\partial \dot{x_2}}{\partial \overline{x_2}} & \frac{\partial \dot{x_2}}{\partial \overline{x_3}} \\ \frac{\partial \dot{x_3}}{\partial \overline{x_1}} & \frac{\partial \dot{x_3}}{\partial \overline{x_2}} & \frac{\partial \dot{x_3}}{\partial \overline{x_3}} \end{bmatrix}, B = \begin{bmatrix} \frac{\partial \dot{x_1}}{\partial \overline{u_1}} \\ \frac{\partial \dot{x_2}}{\partial \overline{u_2}} \\ \frac{\partial \dot{x_3}}{\partial \overline{u_3}} \end{bmatrix}$$

To solve these partial differential equations such that they would equal a value with no variables, an arbitrary $\bar{u}$ was selected. This was chosen as 1. All other variables were either ignored due to partial differentials or substituted using the x values in equation (5.). The resulting matrices were calculated as:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -3.194 & -2 \\ 0 & 0.1667 & -0.2129 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 0.1596 \end{bmatrix} \qquad (6.)$$

To analyse the stability of the system the eigenvalues of the A matrix were calculated.

$$eig(A) = \begin{cases} -3.3832 \\ 0.2399 \\ -0.2623 \end{cases}$$

The three eigenvalues of A indicate that the system is stable at equilibrium points 1 and 3 as the poles are negative in the real plane. However, the system is unstable at equilibrium point 2 where the s placement is positive in the real dimension.

The output of the system can also be written as a function of x and u. It was previously stated that the output of the system optically measured magnetic balls velocity $\dot{y}(x,i)$, however for the purpose of this control system, it seemed more appropriate to measure the displacement of the ball $y(x,i)$. This is because the position gives more practical feedback and allows for quick analysis as to whether the ball is floating and by how much. Equation (4.) was rearranged to give $y(x,i) = x_1$.

This was used to develop C and D matrices such that $\tilde{y} = C\tilde{x} + D\tilde{u}$

$$C = [1 \quad 0 \quad 0], D = 0$$

III. State-Feedback Controller

A controller can be designed such that when attached to a state-space model, it will drive the system towards some desirable state and make the system stable.

Not all systems are controllable. To assess whether the system could be controlled, the determinant of the controllability matrix was investigated. The controllability matrix can be denoted by $C_{AB} = [\ B\ AB\ A^2B\ ...\ A^{n-1}B]$. Since matrix A is a 3x3 matrix the controllability matrix for this system is $C_{AB} = [\ B\ AB\ A^2B]$. Substituting the matrices, A and B from equation (6.) into this results in the controllability matrix:

$$C_{AB} = \begin{bmatrix} 0 & 0 & -0.3194 \\ 0 & -0.3194 & 1.0873 \\ 0.1596 & -0.0340 & -0.0460 \end{bmatrix}$$

The determinant of this matrix was then calculated as -0.0163. As the determinant was non-zero, this controllability matrix is full rank and therefore controllable.

To control this system, optimal control was used in the form of linear quadratic regulator (LQR) to minimise the cost function $J = \int (x^T Qx + u^T Ru)dt$. In this cost function, the square matrix Q penalises the state error while the scalar R penalises the actuation error. These values of Q and R must be positive semi-definite and positive definite respectively. As such, given we want to heavily penalise the state error in the system, a scalar R of 1 and a Q matrix of $20 \times I_3$ were chosen, where I_3 is a 3x3 identity matrix. These were then used to solve the algebraic Riccati equation with the A and B matrices calculated in section II. The algebraic Riccati equation is denoted as:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \qquad (7.)$$

Solving this equation for the matrix P allows for the calculation of LQR gain vector K given by the equation:

$$K = R^{-}B^T P \qquad (8.)$$

$$K = [-9.7510, -2.6673, 8.0797]$$

This 1x3 K vector was then applied to x such that $u = -Kx$. This controls the system towards the equilibrium points using state-feedback x.

IV. Output-Feedback Controller

The state-feedback controller detailed above is shown to move the system towards the desired location. However, the states are not always available to use when designing a controller. To control a system in a black-box scenario where not all the actual states can be obtained, an observer is designed to estimate the states based upon measurable outputs from the system. These estimated states are then used to feed into the controller like in the state-feedback controller.

Similar to controllability, not all systems are completely observable. The observability of a system can be dictated by the determinant of the observability matrix:

$$O_{AC} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

which for the 3x3 A matrix and 3x1 C matrix gives:

$$O_{AC} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & -3.194 & -2 \end{bmatrix}$$

The determinant of this matrix was calculated as -2 which is non-zero and thus makes the system fully observable.

The observer works by estimating a measured output, comparing the measured output and the estimated output to find an error. This error is then scaled by a gain that adjusts the other state estimates. For this reason, the observer needs to act faster than the controller so that the controller is controlling adequate estimates of the system. To ensure the observer has a faster response than the controller, the gain L used in the observer is calculated using pole placement that is significantly greater than the pole placement used to calculate the gain K of the controller. For the system in this report, the poles were distanced by a magnitude of 10.

L is found by letting the eigenvalues of the closed loop system $A^T - C^T L^T$ equal to the magnified poles such that $char_{A^T - C^T L^T} = |sI - (A^T - C^T L^T)|$

$$L = \begin{bmatrix} 43.5492 \\ 349.9527 \\ -746.0502 \end{bmatrix}$$

Applying this gain L to the system caused the estimated state $\hat{x}$ to converge with state-feedback x thus validating this estimate. This is covered later in section V.

The implementation of L and the observer can be seen below in Fig 4.1 which shows how the two known states of u and $y = x_1$ are used to estimate $\hat{x}$. Fig 4.2 shows how the observer interacts with the other subsystems in the non-linear system. It should be noted that the system synthesis block is primarily used for debugging and is not vital to the implementation of the control system.

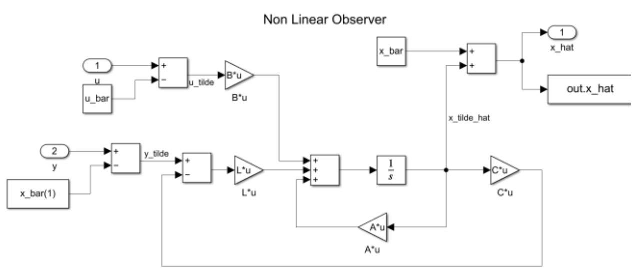

Fig. 4.1 *System Model of Non-Linear Observer*

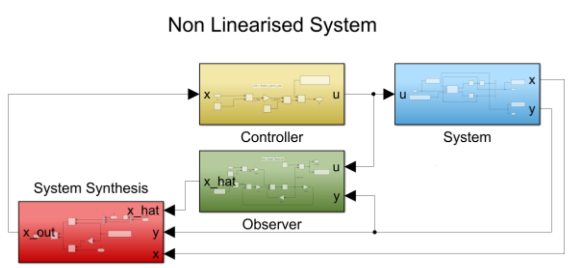

Fig. 4.2 *Complete Non-Linearised system model*

V. Performance Analysis

To analyse the performance and limitations of the state-feedback controller and output-feedback controller, both systems were simulated in MATLAB Simulink. These were modelled with a linearised model - using the calculated A, B, and C matrices to form the system plant, and a non-linear model using the system-state equations provided to construct the system plant.

To isolate the state-feedback controller so that it acted independently from the observed state $\hat{x}$, x was output directly from the system plant and into the controller system. The system response for this can be seen plotted below in Fig 5.1.

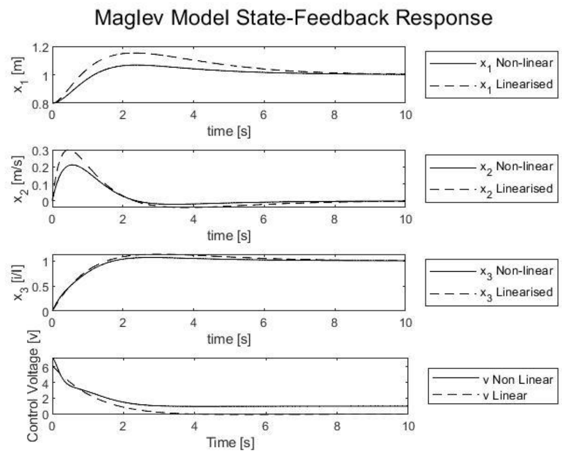

Fig. 5.1 *System response with state-feedback controller implemented*

The response from the state-feedback shows the ball displacement and velocity, and the current and voltages approaching equilibrium at the desired values $\bar{x}$. It should be noted that neither the current x_3 nor the control voltage have been scaled by $\frac{i}{I}$ or $\frac{v}{V}$ in these plots.

Now that the controller was known to work with state-feedback, it was tested using the estimations from the observer. The system response plotted in Fig. 5.2. shows x_1, x_2, x_3 all approaching the desired $\bar{x}$ values described in section II equation (5). Furthermore once x approaches $\bar{x}$, the system appears to stabilise at these values. This indicates stable pole placement and appropriate controller and observer gains. Additionally, input voltage v approaches $\overline{x_3}$ as x stabilises proving the relationship between $\bar{u}$ and $\overline{x_3}$ outlined in section II. It should be noted that the plot shows v linearised approaching 0, this is because the linearised plot is relative to the origin and does not include the addition of $\bar{u}$ as the non-linear does.

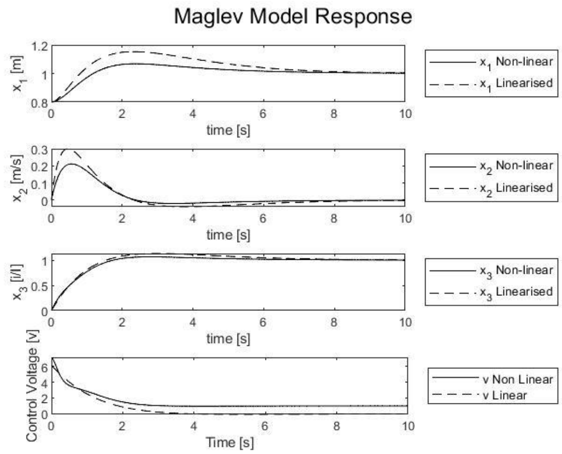

Fig. 5.2 *System response with output-feedback controller implemented*

It is evident when comparing Fig 5.1 and Fig 5.2 that the observer is behaving as expected as both plots appear the same even though Fig 5.2 is using an estimated state of x. This can be further proved by plotting the output of $\hat{x}$ against state-feedback x. This can be seen below in Fig 5.3 where the two different methods of finding x for both linear and non-linear converge into the same points.

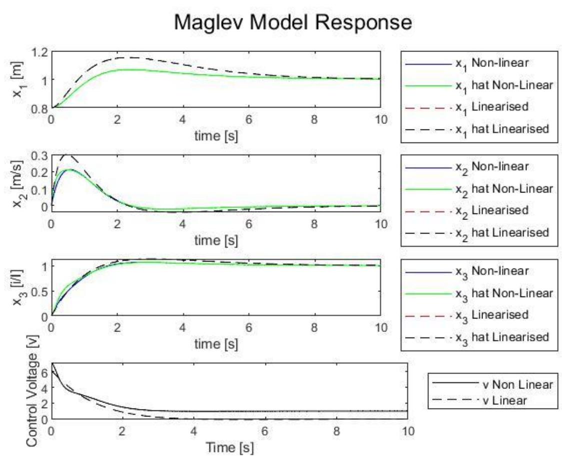

Fig. 5.3 *System response with $\hat{x}$ plotted against x*

To test the robustness of the system, different initial conditions were trialled. Fig. 5.4. shows the response of the system when the initial conditions are $x(0) = [0, \quad 0, \quad 0]$ or when the object is touching the electromagnet at $t = 0$.

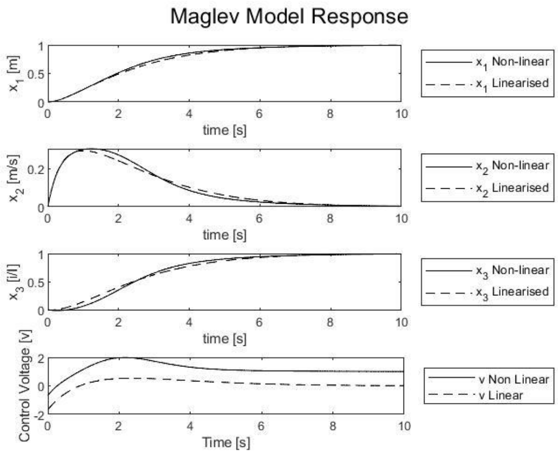

Fig. 5.4 *System response with initial condition x(0) = [0, 0, 0]*

The system response in Fig. 5.4 shows a decreased voltage input and slower application of current than the system when starting at $x(0) = [0.8, 0, 0]$. This makes sense as the gravitational force components attributed in the β components of the state-space equations pull the mass away from the electromagnet above it and towards the equilibrium point $\bar{x}$ 1 meter below the starting position. Additionally, when the mass is closer to the electromagnet than the equilibrium point, the electromagnet will try to repel instead of attracting. This change in directional force is done by reversing the direction of current. This is seen in the plot of the control voltage where it starts negative.

Conversely, if the system is initialised at $x(0) = [10, 0, 0]$ where the object is 10 meters from the electromagnet and 9 meters from the equilibrium point, the system will require a much larger input voltage to move the mass towards equilibrium quickly. This can be seen in Fig. 5.5. where the control voltage initialises at 100 volts.

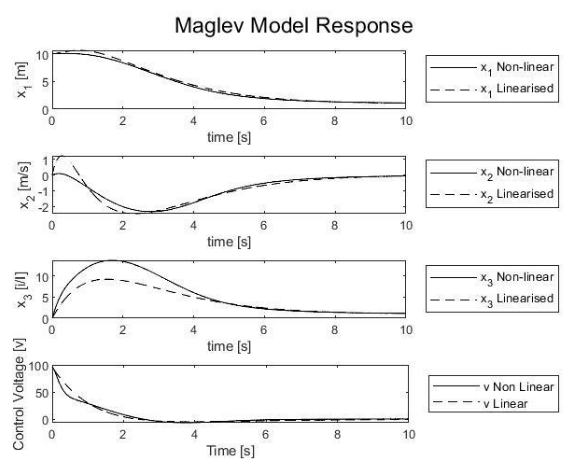

Fig. 5.5 *System response with initial condition x(0) = [10, 0, 0]*

In the application of a maglev train, this scenario where the mass starts 10 meters from the electromagnet is almost impossible, and as such the high voltage seen in Fig 5.5 is not alarming in a real-world application.

The system response was analysed when $x(0)$ is 0.5 meters from the equilibrium point which is a much more realistic worst case scenario for a maglev train. Even under this initial condition, the required input voltage to move the mass to equilibrium was only 12V which can be seen in Fig. 5.6.

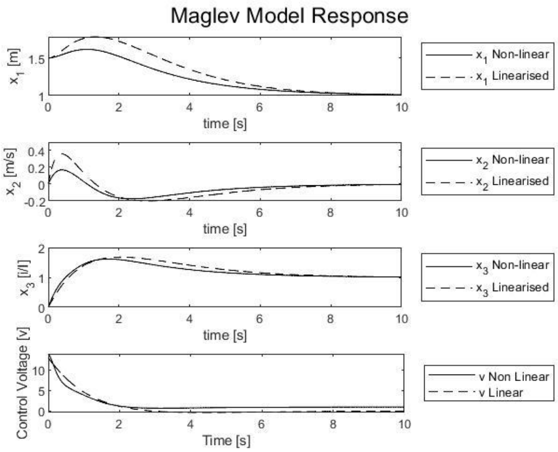

Fig. 5.6 *System response with initial condition x(0) = [1.5, 0, 0]*

The final test of robustness of the system was modelling the response over a much longer time to ensure the system behaviour continued as expected. Fig 5.7 shows the system modelled up to 10000 seconds.

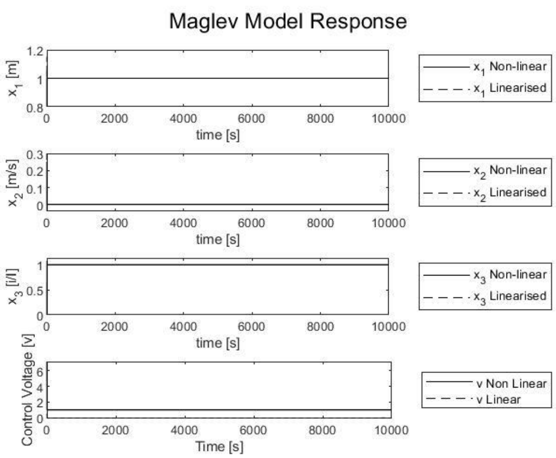

Fig. 5.7 *System response modelled to 10000s*

VI. CONCLUSION

In summary, this report has covered a method of modelling a system of state equations and designing both a state-feedback and output-feedback controller to take the system to a set of equilibrium points. Mathematical concepts such as linearisation were used to transform the state equations into matrix form so that the software MATLAB Simulink could be used to simulate the system. The effects of pole placement and changing the magnitude of Q and R components of LQR optimised control was discussed. Finally, the controllers were analysed and critiqued on validity and robustness.

APPENDIX

I. CONSTANT VALUES

Table 1.1 *Constant values and formulas*

Constant	Value	Formula
m	0.1	-
m_0	0.1	-
k	1	-
g	9.81	-
a	1	-
α	1	-
L_0	1	-
R	1	-
λ	3.1321	$\sqrt{\left(\frac{g}{a}\right)}$
β	3.1321	$\frac{\lambda L_0}{R}$
c	1	$\frac{m_0}{m}$
I	2.8014	$\sqrt{\frac{8\alpha m_0 g}{L_0}}$
b	3.1928	$\frac{1}{\lambda m}$

REFERENCES

[1] H.-W. Lee, K.-C. Kim and J. Lee, "Review of maglev train technologies," *IEEE Transactions on Magnetics,* vol. 42, no. 7, pp. 1917-1925, July 2006.

[2] H. K. Khalil, "Nonlinear Control," *Pearson Education Limited,* 2015.