Inversed-Based Youla Parameterization on Vehicle Steering with Feedforward Model Inversion

Thomas Chu
Department of Mechanical Engineering
University of Washington
Seattle, United States
tchu@uw.edu

***Abstract*—In this analysis, inverse-based Youla Kucera parameterization method is used to design a filter to reject wheel wobble disturbances on a vehicle steering system. The filter works best at rejecting sinusoidal disturbance at 80 radians per second, and the filter reduces the magnitude of the disturbance up to 180 radians per second. Additionally, adding feedforward model inversion with zero phase error tracking (ZPET) to the parameterized system significantly reduces the error of the output tracking position.**

Keywords*—*Youla, Kucera, parameterization, disturbance, steering, wobble, zero, phase, error, tracking, feedforward, model

I. INTRODUCTIONC

The dynamics of a vehicle steering is marginally stable without a control system. To stabilize the steering, modern-day vehicles have control systems to assist drivers and improve the safety and handling performance of the vehicles. The control system must be able to adapt to various road conditions that introduce disturbances which affect the steering. One common disturbance is wheel wobble, which is the uncontrolled oscillation of the steerable wheels occurring between 10 – 16 Hertz (100 – 160 radians per second). Possible solutions to mitigate wheel wobble include adding dampeners or modifying the steering geometry to counteract wheel wobble. Both solutions require physical modifications to the vehicle, and physical modifications are not always viable or economical. A third, less invasive option, is to install a digital filter to reject wheel wobble disturbances.

This analysis investigates the effectiveness of adding a digital filter to the vehicle control system, which has a proportional integral (PI) controller. Inverse-based Youla Kucera parameterization is used to design a disturbance rejection filter to attenuate periodic disturbances caused from wheel wobble. Additionally, model inversion with zero phase error tracking (ZPET) is added to improve the response of the PI controller.

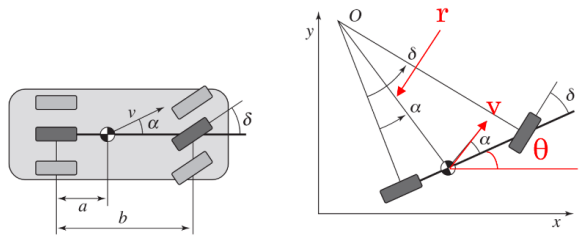


Fig. 1. Vehicle model on the left, bicycle model on the right

II. VEHICLE DYNAMICS

A. Physics

The vehicle model is simplified using the bicycle model in fig. 1. The following definitions are used for deriving the kinematics of the vehicle.

- a – length of the rear wheel to center of mass
- b – length of the wheel base
- b_w – width of the wheel base
- r – turning radius length from point O to the center of mass
- δ – steering angle direction
- δ_o – steering angle of the outer-front tire
- δ_i – steering angle of the inner-front tire
- α – slip angle
- V – vehicle velocity
- θ – vehicle direction relative to the X-axis
- ψ – heading of the vehcile

The bicycle model is used under the assumption that the steering angle is small. The inner and outer tires have different steering angles when turning since the inner and outer front wheels do not share the same axis to point O in fig 2. Using Ackerman's steering geometry, the inner and outer steering angle is approximated assuming the steering angle is small [1].

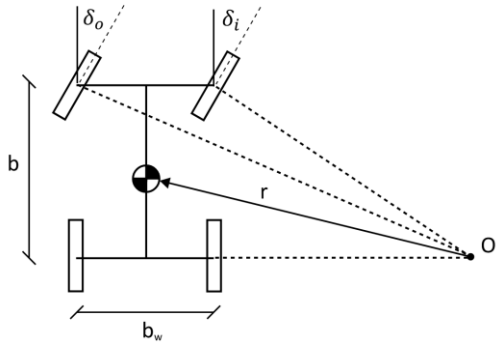


Fig. 2. Ackerman steering geometry

Assuming the wheel base, b_w, is small compared to the radius, r, then the distances from the front to point O are approximated as $r - 0.5b_w$ and $r + 0.5b_w$. Therefore, the steering angle of the inner and outer tires are approximated as

$$\delta_i = \frac{b}{r - \frac{b_w}{2}} \quad (1)$$

$$\delta_o = \frac{b}{r + \frac{b_w}{2}} \quad (2)$$

and the steering angle is approximated as

$$\delta = \frac{\delta_o + \delta_i}{2} \quad (3)$$

Adding (1) to (2) and dividing by two gets

$$\delta = \frac{\delta_o + \delta_i}{2} = \frac{bR}{(R^2 - \frac{b_w^2}{4})} \quad (4)$$

Assuming $R^2 \gg \frac{b_w^2}{4}$, then

$$\delta = \frac{\delta_o + \delta_i}{2} \approx \frac{b}{R} \quad (5)$$

The steering angle is approximately equal to the average of the inner and outer steering angle if $R^2 \gg b_w^2/4$.

B. Equations of Motion

The dynamics of the steering geometry is derived by applying the law of sines [2]. The following angle relations derived from the bicycle model are

$$\frac{\sin(\frac{\pi}{2} - \delta)}{r} = \frac{\sin(\delta - \alpha)}{b - a} \quad (6)$$

$$sin(\alpha) = \frac{a}{r} \quad (7)$$

$$\sin(\frac{\pi}{2} - \delta) = \cos(\delta) \quad (8)$$

$$\sin(\delta - \alpha) = \sin(\delta)\cos(\alpha) - \cos(\delta)\sin(\alpha) \quad (9)$$

Substituting (8) and (9) into (6) and adding (7) gets

$$\frac{b}{r} = \tan(\delta)\cos(\alpha) \quad (10)$$

Let $\psi = \alpha + \theta$ be the heading of the vehicle. Taking the derivative of the heading gets the angular velocity, defined as

$$\dot{\psi} = \frac{V}{r} \quad (11)$$

Rearranging (10) and substituting into (11) expresses angular velocity as a function of steering angle, δ, and slip angle, α in (12).

$$\dot{\psi} = \frac{V}{b}[\tan(\delta)\cos(\alpha)] \quad (12)$$

The relationship between δ and α is found by adding (6) to (10)

$$(b - a)\sin(\alpha) = a \cdot \tan(\delta)\cos(\alpha) - a \cdot \sin(\alpha) \quad (13)$$

which is rearranged into

$$\tan(\alpha) = \frac{a}{b}\tan(\delta) \quad (14)$$

Taking the arctangent of (14) defines α as a function of δ

$$\alpha(\delta) = \arctan(\frac{a}{b}\tan(\delta)) \quad (15)$$

Assuming α is small, then $\dot{\psi} \approx \dot{\theta}$. The equations of motion are defined in (16) – (18).

$$\frac{dX}{dt} = V[\cos(\alpha(\delta) + \theta)] \quad (16)$$

$$\frac{dY}{dt} = V[\cos(\alpha(\delta) + \theta)] \quad (17)$$

$$\dot{\theta} = \frac{V}{b}\left[\tan(\delta)\cos(\alpha(\delta))\right] \quad (18)$$

III. Modelling

The input to the vehicle model is the steering angle, δ, and the state variables are $x, y,$ and θ. Equations (16) – (18) are expressed as the following non-linear state space

$$\frac{d}{dt}\begin{bmatrix} X \\ Y \\ \theta \end{bmatrix} = \begin{bmatrix} V \cdot cos(\alpha(\delta) + \theta) \\ V \cdot sin(\alpha(\delta) + \theta) \\ \frac{V}{b} \cdot \cos(\alpha(\delta)) \cdot \tan(\delta) \end{bmatrix}$$

A. Jacobian Linearization

The non-linear state space is linearized to simplify the design process of the controller, disturbance filter, and the feedward system. Jacobian linearization is used to linearize the state space. The main assumption for linearization is that changes in the state variables are small, and the system is linearized at the equilibrium points. Additionally, the vehicle travels in the x-axis in a constant velocity. This removes x as a state variable, which reduces the system to a second order system and simplifies the design process. The new state is defined as

$$z = x - x_e \quad v = u - u_e \quad w = y - h(x, u)$$

The new system is defined such that v is the input and w is the output, and x_e and u_e are the equilibrium points. The new state variables and input are defined as

$$x = \begin{bmatrix} x_1 = Y \\ x_2 = \theta \end{bmatrix} \qquad u = \delta$$

and the non-linear system is defined as

$$f(x, u) = \begin{bmatrix} \frac{V_0 \cdot sin(\alpha(u) + x_2)}{\cos(\alpha(u))} \\ \frac{V_0}{b} \cdot \tan(u) \end{bmatrix} \qquad h(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

The equilibrium points are $x_1 = 0$ meters, $x_2 = 0$ radians, and $u = 0$ radians. The state equations of the linearized system are defined as

$$\frac{dz}{dt} = Az + Bv$$

$$w = Cz + Dv$$

$$A = \frac{\partial f}{\partial x}\Big|_{\substack{x=x_e \\ u=u_e}}, B = \frac{\partial f}{\partial u}\Big|_{\substack{x=x_e \\ u=u_e}}, C = \frac{\partial h}{\partial x}\Big|_{\substack{x=x_e \\ u=u_e}}, D = \frac{\partial h}{\partial u}\Big|_{\substack{x=x_e \\ u=u_e}}$$

$$A = \begin{bmatrix} 0 & V_0 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{a}{b}V_0 \\ \frac{V_0}{b} \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The velocity, length of the vehicle, and distance from the center of mass to rear wheel are $V_0 = 30$ meters per second, $b = 5$ meters, and $a = 2.5$ meters respectively.

B. Plant – Vehicle Steering

The transfer functions for the linearized system are δ to y and δ to θ. In this report, the parameterized controller is designed to reject disturbance while following a prescribed path on the y position. The open loop transfer function and relative degree of the plant are

$$\delta \to Y: P_{1,OL}(s) = \frac{15(s+12)}{s^2}$$

$$m \triangleq relative\ degree\ of\ P(s) = 1$$

The open-loop plant is discretized using zero order hold (ZOH) with a sampling time ten times faster than the highest frequency of the disturbance.

$$T_s \triangleq sampling\ time = 160\ Hz = 0.00625\ s$$

$$P(z) = \frac{0.09727z - 0.09023}{z^2 - 2z + 1} \qquad (19)$$

C. Controller – Proportional Integral Design

The PI controller is designed using Ziegler Nichols tuning method. The PI controller increases the system type by one which allows a steady state error of zero for a step input. The PI controller is designed in continuous time and converted into a discrete time controller with a sampling time of $T_s = 0.00625$ second.

$$C(s) = 17.1\left(1 + \frac{6}{s}\right)$$

$$C(z) = \frac{17.1(z - 0.963)}{z - 1} \qquad (20)$$

The controller, $C(z)$, is stabilizing since the poles and zeros of the closed-loop system are located within the unit circle of the pole-zero map in fig. 3.

IV. Inverse-Based Youla Kucera Paramterization

A. Disturbance Signal

Disturbances due to road conditions and wheel imbalance can excite the steerable wheels and cause wobbling. Factors such as the stiffness of the tire and steering geometry influence the wobbling frequency; these wobbles have frequencies from $60 - 100$ radians per second at highway speeds. The disturbance of wheel wobble is modeled as a sinusoidal function

$$d(k) = \text{Asin}\,(\omega_w T_s k)$$

$$\omega_w \triangleq wobble\ frequency\ (\text{rad/s})$$

$$A \triangleq \text{amplitude (rad)}$$

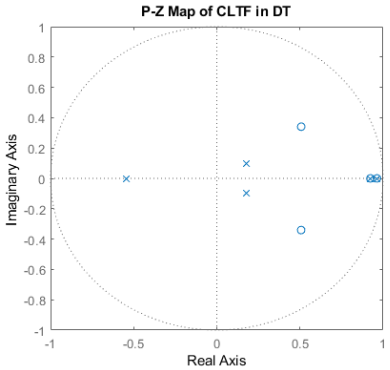


Fig. 3. Pole-zero mapping of the closed-loop system on the z-plane

B. Filter Design

The filter is designed using inverse-based Youla Kucera parameterization. The parameterized controller, $Q(z)$, is added to the system based on the block diagram in fig. 4. The inversion method takes an approximated inverse of $P(z)$. Since the relative degree of $P(z)$ is $m = 1$, the system is multiplied by z^{-1} to make the inverse proper. Therefore, the approximated inverse is

$$\hat{P}(z)^{-1} = \frac{z^{-m}}{P(z)}$$

$Q(z)$ is parameterized to reject disturbances with a frequency of 80 radians per second. The filter, $A(z^{-1})$, is defined as

$$A(z^{-1}) = 1 - z^{-m}Q(z) \qquad (21)$$

$$A(z^{-1})d(k) = 0 \qquad (22)$$

The following properties are used to solve (21) for any sinusoidal disturbances. Let $\omega_0 = \omega_w \cdot T_s$

$$d(k) = d_1(k) + d_2(k) = \frac{e^{j\omega_0 k} - e^{-j\omega_0 k}}{2j} \qquad (23)$$

$$A(z^{-1}) = A_1(z^{-1})A_2(z^{-1}) \qquad (24)$$

where

$$d_1(k) = \frac{e^{j\omega_0 k}}{2j}$$

$$d_2(k) = -\frac{e^{-j\omega_0 k}}{2j}$$

Substituting (23) and (24) into (22)

$$A_1(z^{-1})A_2(z^{-1})\big(d_1(k) + d_2(k)\big) = 0$$

$$A_2(z^{-1})\big(A_1(z^{-1})(d_1(k))\big) + (z^{-1})\left(A_2(z^{-1})\big(d_2(k)\big)\right) = 0$$

The solution to (22) is

$$A_1(z^{-1})\big(d_1(k)\big) = 0$$

$$A_2(z^{-1})\big(d_2(k)\big) = 0$$

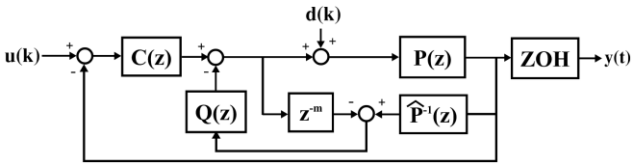


Fig. 4. Block diagram of the parameterized system with disturbance rejection. $d(k)$ is the disturbance input and $u(k)$ is the input for the desired path.

Solving for $A_1(z^{-1})$ and $A_2(z^{-1})$

$$A_1(z^{-1}) = (1 - e^{j\omega_0} z^{-1}) \quad (25)$$

$$A_2(z^{-1}) = (1 - e^{-j\omega_0} z^{-1}) \quad (26)$$

The solution to $A(z^{-1})$ and $Q(z)$ with $m = 1$ are

$$A(z^{-1}) = 1 - 2\cos(\omega_0)\, z^{-1} + z^{-2} \quad (27)$$

$$Q(z) = 2\cos(\omega_0) - z^{-1} \quad (28)$$

While the solution shown for $Q(z)$ is analytically correct, the bode plot of $Q(z)$ does not yield a desirable frequency response as there is amplification of input signals with frequencies lower than 80 radians per second. Instead, the desired filter is achieved by dividing the filter by $A(\alpha z^{-1})$, which produces the desired frequency response [3]. The desired $Q(z)$ is found using (29)

$$\frac{A(z^{-1})}{A(\alpha z^{-1})} = \frac{1-2\cos(\omega_0)z^{-1}+z^{-2}}{1-2\alpha\cos(\omega_0)z^{-1}+\alpha^2 z^{-2}} = 1 - z^{-m}Q(z) \quad (29)$$

where $\alpha < 1$ and $m = 1$

$$Q(z) = \frac{(2-2\alpha)\cos(\omega_0)+(\alpha^2-1)z^{-1}}{1-2\alpha\cos(\omega_0)z^{-1}+\alpha^2 z^{-2}} \quad (30)$$

The bode plot of (29) is shown in fig. 5. There is large attenuation at 80 radians per second. The parameterized controller, $\tilde{C}(z)$, loop transfer function, $\tilde{L}(z)$, and sensitivity function, $\tilde{S}(z)$, are defined as

$$\tilde{C}(z) = \frac{C+z^{-m}\hat{P}^{-1}(z)Q(z)}{1-z^{-m}Q(z)} \quad (31)$$

$$\tilde{L}(z) = P(z)\tilde{C}(z) \quad (32)$$

$$\tilde{S}(z) = \frac{1}{P(z)\tilde{C}(z)} \quad (33)$$

C. Simulation

The parameterized system is built in Simulink and the input for the parameterized system is the desired trajectory of the vehicle. Fig. 6 shows the desired trajectory of the vehicle's y – position in meters, and fig. 7 shows the Simulink model. The output of the model is tracked at various disturbance frequencies, ω_w (60, 80, and 100 radians per second), and the amplitude is 1 radian. The filter is also tested at various α (0.2, 0.6, and 0.99) to observe the effects of α on the filter.

D. Analysis

1) Disturbance Rejection Frequency

The results of the three frequencies tested are shown in fig. 8(a) – (c). The parameterized system is compared to a control; the control only has a PI controller. The filter is fixed at $\alpha = 0.2$. Simulation shows that the filter works best at $\omega_w = 80$ radians per second. Fig. 8(b) shows that the oscillating behavior is gone, and the vehicle follows the trajectory well. However, at $\omega_w = 60$ and 100 radians per second, the oscillating is present, but the amplitude of the disturbance is reduced by 80 percent.

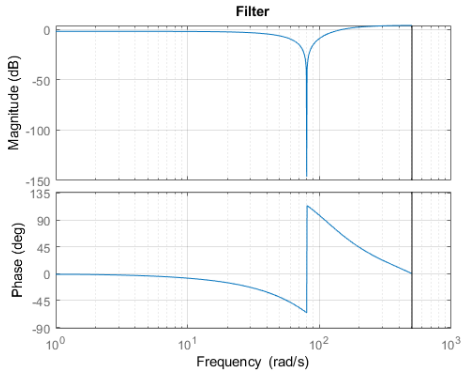


Fig. 5. Bode plot of the filter, $\frac{A(z^{-1})}{A(\alpha z^{-1})}$

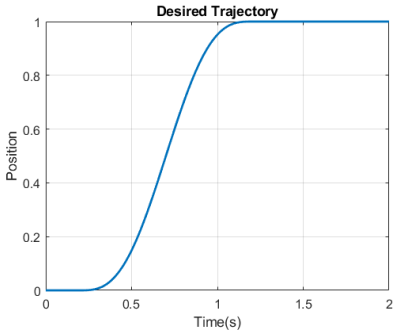


Fig. 6. Desired trajectory in the y – axis (meters)

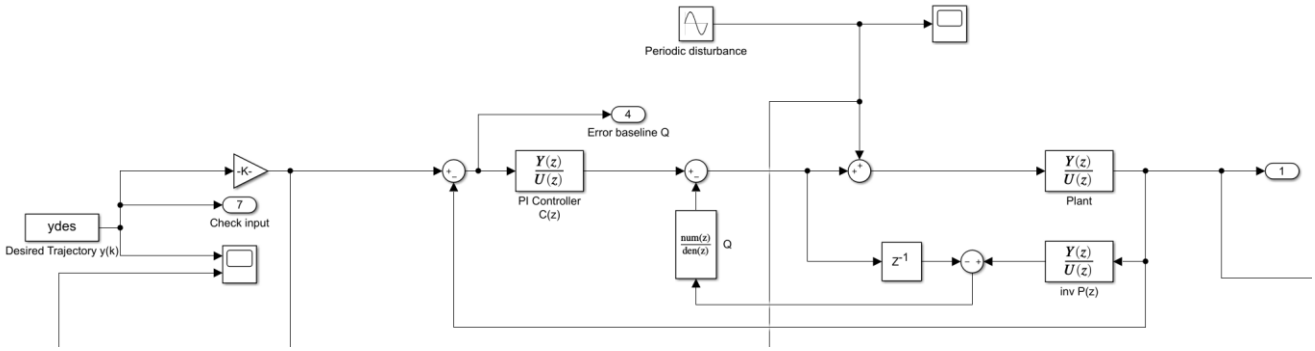
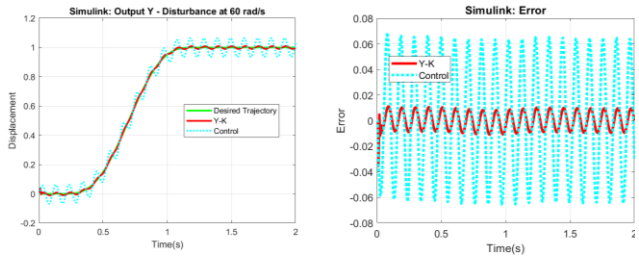
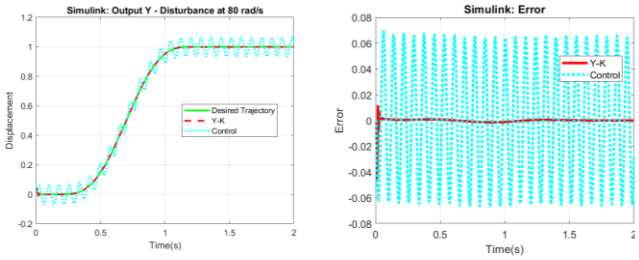


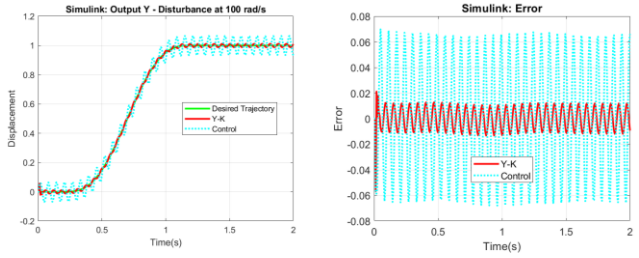
Fig. 7. Simulink model of the parameterized system



(a) $\omega_w = 60\ (rad/s)$



(b) $\omega_w = 80\ (rad/s)$



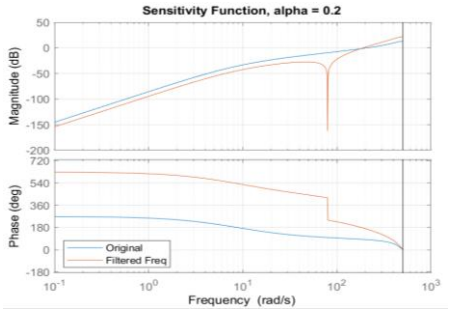
(c) $\omega_w = 100\ (rad/s)$

Fig. 8. Output tracking of the Simulink model ($\alpha = 0.2$) at various disturbance frequencies

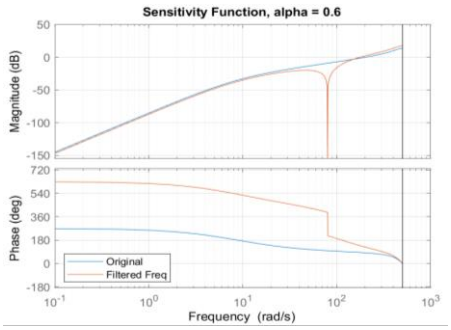
The results show that the filter increases the disturbance attenuation magnitude across a range of frequencies. Fig. 9(a) – (c) show the sensitivity functions of the parameterized system; a filter with lower α increases the magnitude of the disturbance rejection region up to 180 radians per second, with the best rejection at 80 rad/s.

2) α and Disturbance Rejection

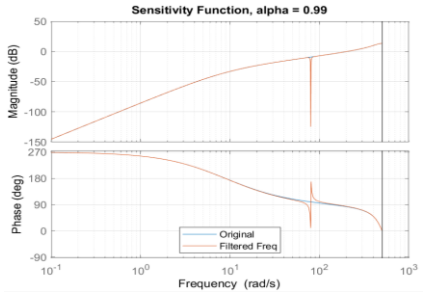
Changing α changes the response time of the disturbance rejection and the attenuation range of the filter. Decreasing α increases the amplitude of the disturbance rejection region, as seen in the bode plot when $\alpha = 0.2$ in fig. 9(a). Increasing α reduces the attenuation range, and the magnitude of the rejection region is lower over the same range, as shown in the bode plots in fig. 9(b) and (c). Additionally, the response time of disturbance rejection increases as α approaches 1. Therefore, reducing α is favorable for disturbance rejection if the system does not have any input frequencies higher than the 180 radians per second. The region with frequencies above 180 radians per second is the disturbance enhancement region, which has a positive magnitude. The magnitude of the parameterized system is larger in the disturbance enchantment region when α is decreased, which is proven with Bode's Integral Theorem [4]. Bode's Integral Theorem states



(a) $\alpha = 0.2$



(b) $\alpha = 0.6$



(c) $\alpha = 0.99$

Fig. 9. Bode plot of the sensitivity function, $\tilde{S}(z)$ at various α

$$\frac{1}{\pi}\int_0^{\infty} ln\left|\tilde{S}(j\omega)\right| d\omega = -\frac{1}{2}K_s$$

$$k_s = \lim_{s\to\infty} s\tilde{L}(s)$$

which shows the integral of the sensitivity function is constant if the loop transfer function, $\tilde{L}(s)$, and sensitivity function, $\tilde{S}(s)$, are rational and stable. Therefore, increasing the area of the disturbance rejection region also increases the area of the disturbance enhancement region equally, which is known as the waterbed effect. The waterbed effect is seen in fig. 9(a) and (b), where the disturbance enhancement region of the parameterized system has a larger magnitude than the nonparameterized system.

V. Feedforward with Zero Phase Error Tracking

The PI controller has good tracking, and the error of the output tracking is improved by adding a feedforward system

with ZPET. The closed-loop system with the parameterized controller is inverted and multiplied by z^{-1} to make the system proper. The closed-loop transfer function of the parameterized system is defined in (34), and the approximated inverse model is defined in (35).

$$G(z) = \frac{P(z)\cdot\tilde{C}(z)}{1+P(z)\cdot\tilde{C}(z)} \quad (34)$$

$$\hat{G}^{-1}(z) = \frac{z^{-1}}{G(z)} \quad (35)$$

The input for the feedforward system is the desired trajectory in fig. 6. To remove phase lag, the desired trajectory is shifted one step forward since the system has a relative degree $m = 1$ [5].

$$u(k) = y_d(k+1)$$

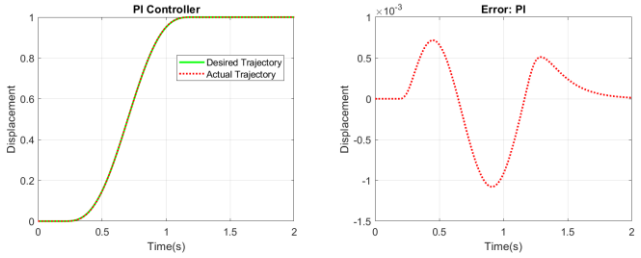
$$\text{where } y_d(k) \triangleq desired\ trajectory$$

A. MATLAB Simulation

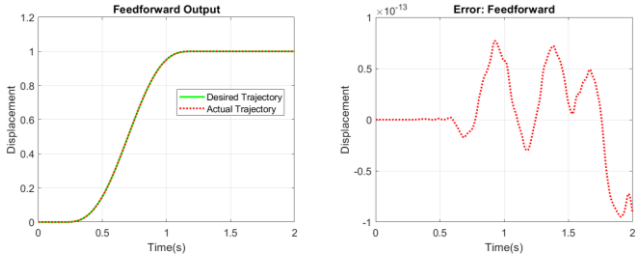
The feedforward inverse is the input for the closed loop system. The parameterized system with feedforward is simulated using LSIM() in MATLAB. Output tracking and error are shown in fig. 10(a) and (b). The error of the system without feedforward has a small error with a magnitude of 10^{-3}; however, adding feedforward with ZPET reduces the error to a magnitude of 10^{-13}.

B. Simulink Simulation

The feedforward system is built in Simulink based on the block diagram in fig. 11. Output tracking and error are shown in fig. 12. The error of the system without feedforward is similar to the MATLAB results, but the system with feedforward has larger error than the feedforward system in MATLAB. There is an improvement by adding a feedforward system, but the feedforward system in Simulink does not provide perfect tracking.



(a) Without feedforward



(b) Feedforward with ZPET

Fig. 10. MATLAB simulation of the output trajectory

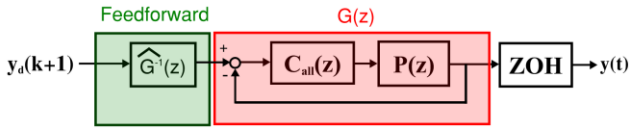


Fig. 11. Block diagram of the feedforward system with ZPET

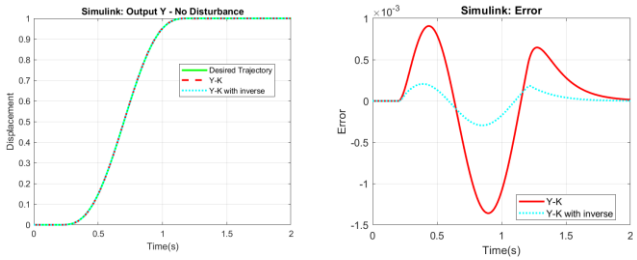


Fig. 12. Simulink model of the output trajectory

C. Analysis

The feedforward model with ZPET reduces the error of the output tracking. The results from MATLAB show that applying feedforward inverse with ZPET reduces the error to 10^{-13}. This result shows near perfect tracking. The Simulink results show that feedforward inverse with ZPET does reduce the error compared to the PI controller, but the feedforward tracking does not provide prefect tracking. Adding a feedforward system also requires the known path trajectory, which may not be viable when taking the vehicle to on an unknown destination.

VI. Conclusion

The results from Simulink show that it is possible to design a filter to reject periodic disturbances. The parameterized controller, Q(z), rejects disturbance best at 80 radians per second. Additionally, the filter reduces the amplitude of disturbances near the design frequency of the filter. Reducing α in the filter does increase the range and magnitude of disturbance attenuation, but it also increases the magnitude of the disturbance enhancement region due to the water bed effect.

Adding a feedforward system with ZPET reduces output error and improve output tracking. The implementation of feedforward systems is best for systems that have large phase lags. However, knowledge of the path of travel is required, so if the vehicle is traveling on an unknown path, feedforward cannot be used. The PI controller on this vehicle steering system provides adequate tracking, so feedforward is not necessary.

References

[1] Rajamani, R. (2005). Vehicle Dynamics and Control. Minnesota: Springer Science and Business Media

[2] Åström, K. J., & Murray, R. M. (2012). Feedback Systems. Oxfordshire: Princeton University Press

[3] X. Chen and M. Tomizuka, "Overview and New Results in Disturbance Observer based Adaptive Vibration Rejection With Application to Advanced Manufacturing," Int. J Adapt. Control Signal Process. 2015; 00:1-16

[4] X. Chen, T. Jiang, and M. Tomizuka, "Pseudo Youla-Kucera Parameterization with Control of the Waterbed Effect for Local Loop Shaping," Automatica, vol. 62, 2015, pp.177-183

[5] M. Tomizuka, "Zero Phase Error Tracking Algorithm for Digital Control," ASME Journal of Dynamic Systems, Measurements, and Control, vol. 109, 1987, no. 1, pp. 65–68, 198