We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

Deterministic Approaches to Transient Trajectory Generation

Matthew A. Cooper

IntechOpen

Abstract

This chapter studies a deterministic approach to transient trajectory generation and control as applied to the forced Van der Pol oscillatory system. This type of system tends towards a strongly nonlinear system, which can be considered chaotic. A classical tuning method, targeted exponential weighting, and isolated trajectory fractionalization trajectory generation methods are examined. Illustrating the given deterministic approach via the Van der Pol system highlights the potentially iterative nature of deterministic methods, and that traditional optimal linear time-invariant control techniques are unable to perform as desired whereas even an idealized nonlinear feedforward control significantly outperforms at the steady-state. It will be shown that utilizing a-priori knowledge of the system dynamics will enable the isolated trajectory fractionalization method to minimize the nonlinear transient effects due to miss-modeled or unmodeled plant dynamics, and that this benefit can be coupled with the targeted exponential weighting approach for greatly decreased trajectory tracking error on the order of a 92% reduction of the objective cost function in the presented case study based on the forced Van der Pol system.

Keywords: Van der Pol, trajectory generation, path planning, nonlinear transients, control systems, nonlinear dynamic system, aerospace engineering, non-stochastic, deterministic, autonomy, intelligent systems, feedforward, dynamic inversion, disturbance modeling, phase portrait tuning

IntechOpen

1. Introduction

Transient trajectory generation research is an interesting area, and often finds itself in a highly nonlinear environment. Similar to trajectory optimization problems [1–9], the goal of designing transient trajectories is to reach the desired steady-state trajectory more quickly while simultaneously minimizing and unwanted micro-transients along the way. Micro-transients can be considered as a subcategory of transient trajectories. Transient trajectories are smaller trajectories that connect two steady-state trajectories and can be quite volatile and full of nonlinear micro transients that may push the system towards instability. Section 1 will introduce the reader to basic control theory, and the Van der Pol oscillator which will be used to illustrate transient trajectory generation approaches. Section 2 will describe a classical tuning method, targeted exponential weighting, and isolated trajectory fractionalization trajectory generation methods and present the corresponding results of each method. Section 3 will finish off the chapter by summarizing the results in table format.

1.1 Basic control theory

An introduction to control theory will be conducted to set the understanding baseline in common controls. This refresher will start with state-space nomenclature and then quickly step through the most common feedback control, feedforward control, observers, and adaptive designs. This will be the ground work that is used to later illustrate to the trajectory generation techniques and the inherent complexity in nonlinear dynamical systems such as the Van der Pol oscillator. Any dynamical system can be represented in state-space matrix form [10] to provide a solution formed as shown in Eq. (1):

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{cases}, \tag{1}$$

where A is a matrix representing the system states (also referred to as the plant). B is an input matrix, C is the output matrix, and D is the feedback/feedforward control input matrix. A symbolic representation of a feedback control loop is illustrated in **Figure 1**.

The input to **Figure 1** is nominally a desired position (output) from a user which may be converted to a usable command in the generate trajectory block. For the initial output the input will be fed directly to the plant for a resulting position. If this is not the desired position, then the feedback controller will calculate that error between the desired and resulting position and produced an appropriately weighted control signal to adjust the input. This will happen cyclically until the error approaches zero, and is typically referred to as a zero-seeking negative-feedback approach [10]. A common way to represent the interaction between the input and output of a system is as a transfer function [11], and is usually presented in the Laplacian domain as shown as H in Eq. (2). Also known as the S-domain, this can be an easier format to mathematically manipulate the system variables primarily due to the fact that the system blocks interact in a convolutional way in the time-domain whereas in the S-domain one can just perform multiplication and achieve equivalent results [11].

$$H(s) = \frac{A(s)}{1 + D(s)A(s)} \tag{2}$$

A very common way to implement a feedback controller is through the use of a proportional-integral-derivative (PID) controller [11] as shown in **Figure 2**.

The PID controller is the staple of many feedback control systems [11], and can be formulated such that it is a zero-seeking architecture by calculating the difference between the measured output of the system and the desired output as shown in **Figure 2** and Eq. (4). The error signal is then fed into the proportional, integral, and derivative blocks, and summed together to provide an additional control input to the plant. The proportional block assigns a constant gain (K_p) to the error signal, the

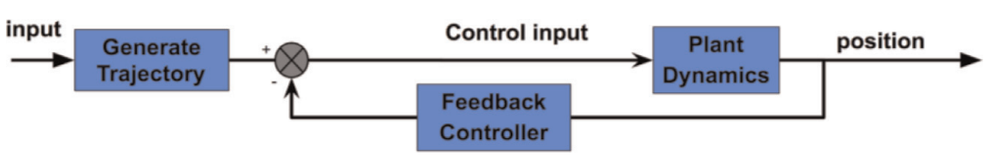


Figure 1.
Generic feedback control schematic.

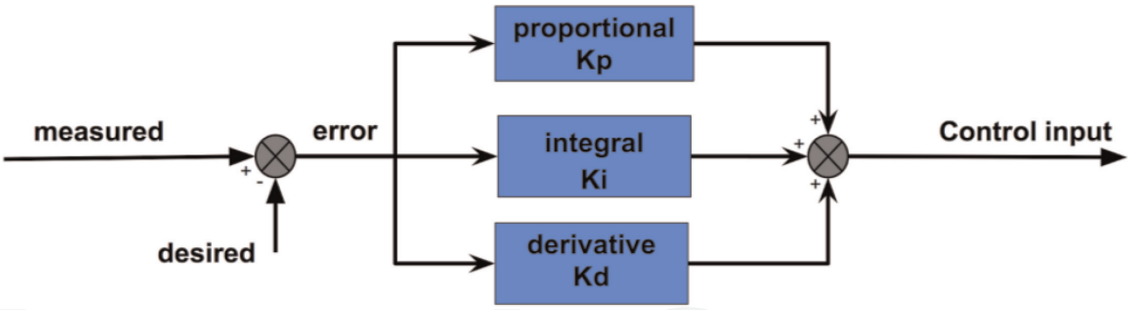


Figure 2.
Generalized PID feedback controller.

integral block calculates the integral of the error signal and applies a different gain (K_i). There are many different ways and variations of implementing a PID controller such as sometimes only utilizing one or two components as shown in Eq. (3) as a typical PI controller example.

$$u = (\tilde{q} - q)(K_p + \frac{K_i}{s}), \tag{3}$$

where q is the desired position and $\hat{\mathbf{q}}$ is the measured position and the error is defined as:

$$e = \hat{q} - q \tag{4}$$

The third and final component of the PID feedback controller is the derivative block which calculated the derivative of the error signal and assigns a corresponding gain (K_d) to it. If all three components are summed together as shown in **Figure 2**, it will result in Eq. (5) for the time-domain solution, and Eq. 6 for the S-domain. Eqs. (5) and (6) illustrate both the time-domain equation and the S-domain equation for ease of comparison.

$$u(t) = K_p e(t) + K_i \int e(\tau)\mathrm{d}\tau + K_d \frac{\mathrm{d}\, e(t)}{\mathrm{d}t} \tag{5}$$

$$u(S) = e\left(K_p + \frac{K_i}{S} + K_d S)\right) \tag{6}$$

Another common feedback control method extends into the realm of optimal control [1, 2] which is focused primarily at looking at control as a cost minimization problem. A linear quadratic regulator (LQR) is one way to treat a feedback control system as an optimal control problem [1]. One variation based in the continuous time-domain treats the problem with a focus on the infinite steady-state, the infinite-horizon continuous-time LQR. The cost function of this type of solution is identified in Eq. (7).

$$J = \int_0^\infty \left(x^T Q x + u^T R u + 2x^T N u\right) dt, \tag{7}$$

where Q is the state cost matrix, R is the input cost matrix, and N is the final state cost matrix with a feedback control law of:

$$u = -Kx \tag{8}$$

where:

$$K = R^{-1}(B^T P + N^T), \tag{9}$$

and P is found by solving the continuous time Riccati equation [11]:

$$A^T P + PA - (PB + N)R^{-1}(B^T P + N^T) + Q = 0 \tag{10}$$

For a linear system, with a linear response, an appropriately chosen cost function, J, can achieve optimal feedback control parameters and find the feedback control parameters that minimize the control cost J.

When full-state feedback is not present in a system, i.e. the outputs cannot be fully known, the control system will either need to be designed without the need for the missing information or those states will need to be estimated based on the sensed outputs as seen in **Figure 3**. In this case an estimator is also referred to as a state observer. A state observer's role is to estimate the internal states of a given system based on the current inputs and outputs. For a more thorough discussion the reader is referred to [10].

An example of taking the general formulation above and applying it to a specific system is illustrated in **Figure 4** for a spacecraft, more specifically a satellite. Here the system dynamics will also include control moment gyros, and other physical restraints/limitations of a real-world plant [12–20]. Additionally, the control variables here are in angle, angular rate, and angular acceleration in order to apply the correct torque on the motors to affect the desired outcome.

The key takeaway here is that the high-level depiction of the system identified in **Figure 4** is not very different from the general case, only that the details within the blocks encompass many more physical variables.

1.2 The Van der Pol oscillator

Balthasar van der Pol (1889–1959) was a Dutch physicist who became interested in the differential equations of coupled electrical systems, which formed into

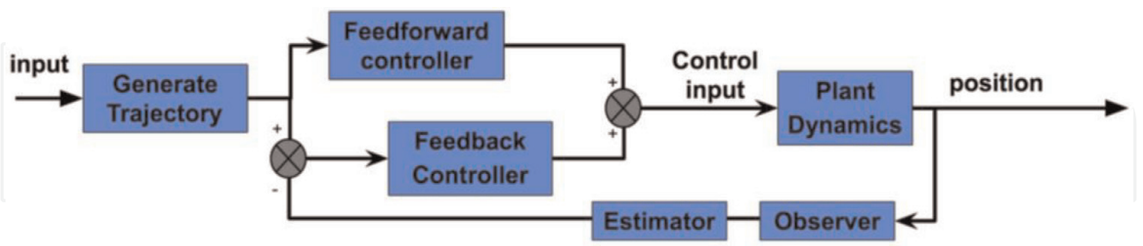


Figure 3.
Generic nonlinear feedforward/feedback control schematic.

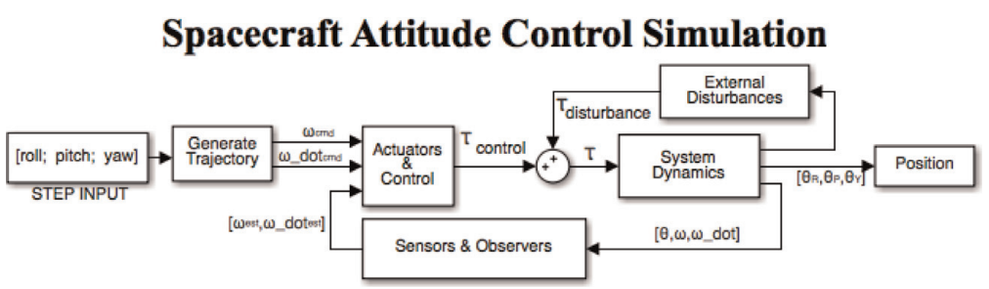


Figure 4.
Nonlinear feedforward/feedback spacecraft control schematic.

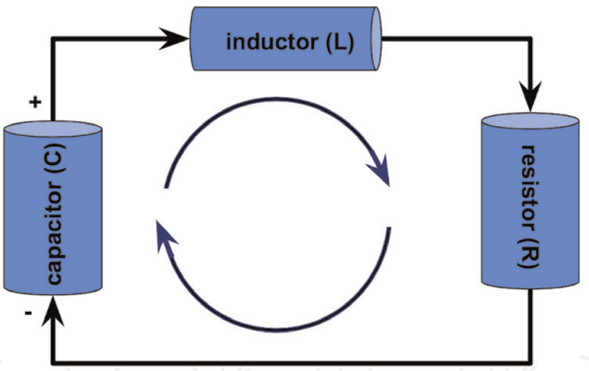


Figure 5.
RLC schematic for VDP derivation.

relaxation oscillators, what later became a description of what we now call limit cycles [22]. His initial investigation began with describing the human heartbeat, and later when turning towards the issues of radio communication [23–25] at the time (the result of deterministic chaos) and is illustrated with the resistor-inductor-capacitor (RLC) circuit shown in **Figure 5**.

The circuit in **Figure 5** represents a schematic for an RLC circuit comprised of a nonlinear resistor (R), an inductor (L), and a capacitor (C). This common circuit can be described by the differential equation [21] in Eq. (11):

$$\ddot{V} - \frac{1}{C}(\alpha - 3\gamma V^2)\dot{V} + \frac{1}{LC}V = 0, \tag{11}$$

where:

$$x = \sqrt{3\gamma/\alpha}V \text{ , and } \mu = \sqrt{L/C}\alpha, \text{ and } L/R \propto \mu\sqrt{LC}, \tag{12}$$

and where μ is the damping coefficient, and can be rearranged to get in the form of the differential equation that describes the Van der Pol equation for limit cycle oscillations in Eq. (13).

$$\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0, \tag{13}$$

By using the simplification assumption in Eq. (15), one can simulate the VPD as a plant to a controls simulation in MATLAB/Simulink structured around the model presented previously in **Figure 3**. The natural periodic oscillations in the phase portrait can be seen in **Figure 6**, which are shown to converge to the same limit cycle from any initial condition nearby (local). This model will be discussed in greater detail in Section 2.

Phase portraits are ways to represent the performance of a system, and to also analyze the expected stability within some local bounds usually determined by the anticipated operational bounds [10, 11, 23]. These portraits display the state of interest with respect to the derivative of that state. Ideally, the goal is such that the state is controllable enough so that the state can be forced to zero within a given time-constant. Sometimes, in a nonlinear system, just being able to prevent the state from growing unbounded (blowing up) and towards an arbitrary asymptotic limit is acceptable [11].

Taking the unforced VPD equation and adding a sinusoidal forcing function to Eq. (13) gives Eq. (14):

$$\frac{d^2x}{dt^2} - \mu(1-x^2)\frac{dx}{dt} + x - A\sin(\omega t) = 0, \tag{14}$$

Using the forced VDP equation, and simplifying it by using the outlined assumptions in Eq. (15) to achieve a perfectly circular limit cycle:

$$\mu = 1, \quad \omega = 1, \quad A = 1, \tag{15}$$

to get Eq. (16):

$$\ddot{x} - (1-x^2)\dot{x} + x - sin(t) = \ddot{x}_d - (1-x_d^2)\dot{x}_d + x_d, \tag{16}$$

In order to implement the forcing function onto the VDP system, it is needed to invert the dynamics such that the VDP equation is used as the feedforward control signal, which will feed into the plant. The steady-state results can be seen in **Figure 7**, with a severe amount to transient phenomenon before reaching the perfectly circular limit-cycle steady-state response.

1.3 What are transient trajectories?

What are transient trajectories? This is a good question, and one with issues and difficulties that the reader may understand without explicitly realizing it. Transient trajectories are those trajectories that transition between the intended steady-state trajectories, and are often short lived [1, 10, 11, 26–28]. The most common types of transient trajectories are of the variety if initial start-up of a system [11]. For example; let us say we have a satellite pointing at some location (x1, y1) on Earth, and now it is needed to point a different location (x2, y2). The desired control and corresponding control trajectory (if a solely feedback control loop is not implemented) will comprise the transient trajectory. It starts at (x1, y1), ends at (x2, y2), and incorporates every point in the resulting path between those two

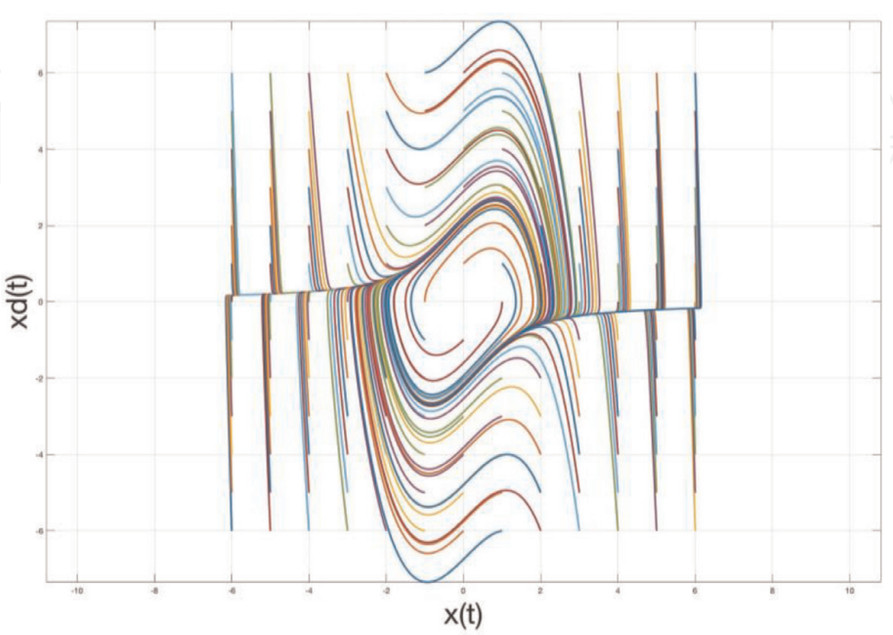


Figure 6.
Phase portrait illustration of unforced VDP system.

points that the platform takes to include the multiple controllable states of the system.

Figure 8 shows an example of a desired transient trajectory. In this example a satellite with an electro-optical telescope is being steered by directing the pointing vector of the staring sensor it employs. The sensor starts at a (0°, 0°, 0°) in the standard (roll, pitch, yaw) coordinate system, and then the user desires the sensor to move to (0°, 0°, 30°). The user inputs a command of 30° yaw to the system, and can be interpreted in many ways. The red, yellow, and blue lines represent different variations of the interpretation across the desired angle, angular rate, and angular acceleration. The trajectory in red represents a sinusoidal-based step function that takes on discontinuities in the angular rate and acceleration terms, whereas the trajectories in yellow and blue are entirely continuous.

The effects of transient trajectories can be illustrated via phase portraits like those presented in **Figures 6** and **7**, with the main concern in affecting the stability of a system of interest. If the transients couple with the potential variations in initial conditions to produce deleterious effects on the stability, it will be illustrated very clearly in a phase portrait as it will likely grow unbounded. Another method on evaluating the performance of a system when evaluating the effects of transients is through the use of an object cost function [10]. A common approach is by using the root-mean-square (RMS) value between the desired trajectory and the measured trajectory to gain an RMS error value as shown in Eq. (17).

$$RMS_{error} = \lim_{T\to\infty} \sqrt{\frac{1}{T}\int_0^T [q(t) - \hat{q}(t)]^2 \, dt}, \tag{17}$$

where $\hat{q}$ is the measured trajectory, and q is the desired trajectory.

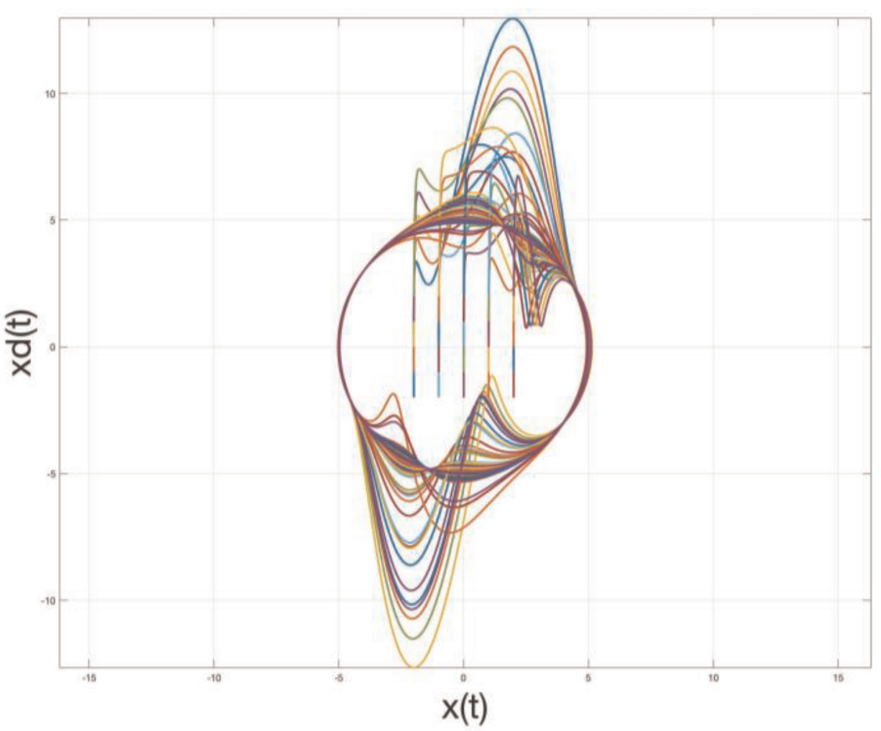


Figure 7.
Phase portrait illustration of forced VDP system.

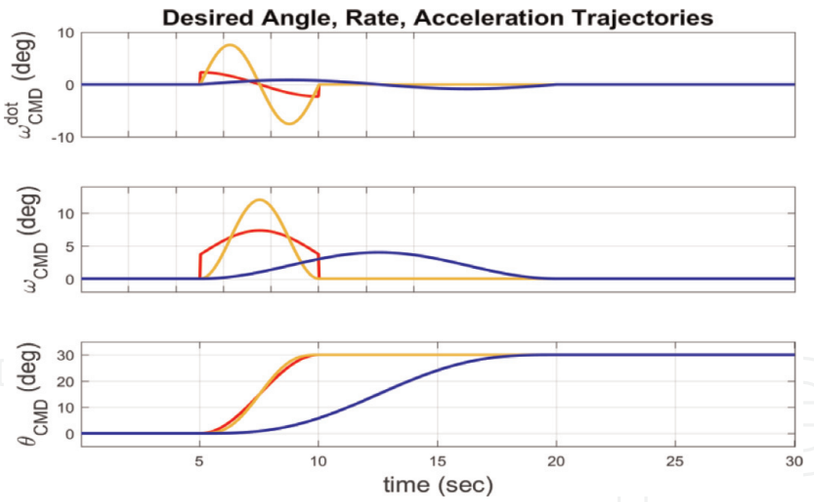


Figure 8.
Desired transient trajectory for a 30° yaw Maneuver of an optical pointing satellite vector.

2. Transient trajectory generation case study

At this point it is relevant to apply the basic components introduced in the previous section to the common nonlinear, and slightly chaotic VDP system to illustrate the benefits and highlight potential areas for further discussion.

Taking the Van der Pol oscillatory system as described by Eq. (16), and modeling it as the system plant we get the MATLAB/Simulink model in **Figure 9**. This model receives a control signal, and generates a new position. Inverting the dynamics result in a similarly defined model for a feedforward controller identified in **Figure 10**. Here, the model receives a defined trajectory vector comprised of the position, velocity, and acceleration ($x\dot{x}$, and $\ddot{x}$ respectively). Propagating those trajectories through the model creates an "ideal" control signal for the plant dynamics.

The plant dynamics and the feedforward controller form the heart of the simulation as the rest of the system parameters will change relative to the different approaches. Both of these models are incorporated in the high-level system diagram illustrated in **Figure 11**. The high-level diagram also shows the user input starting from the right-hand-side, which is fed in multiple trajectory generation functions.

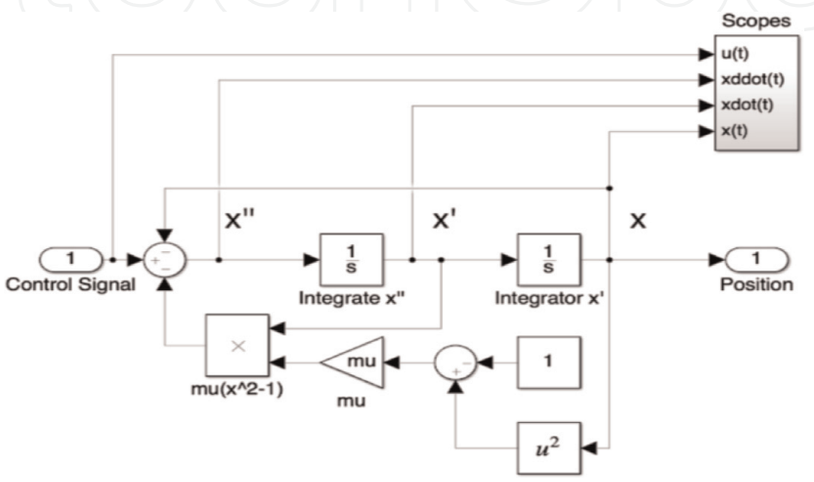


Figure 9.
Van der Pol plant dynamics.

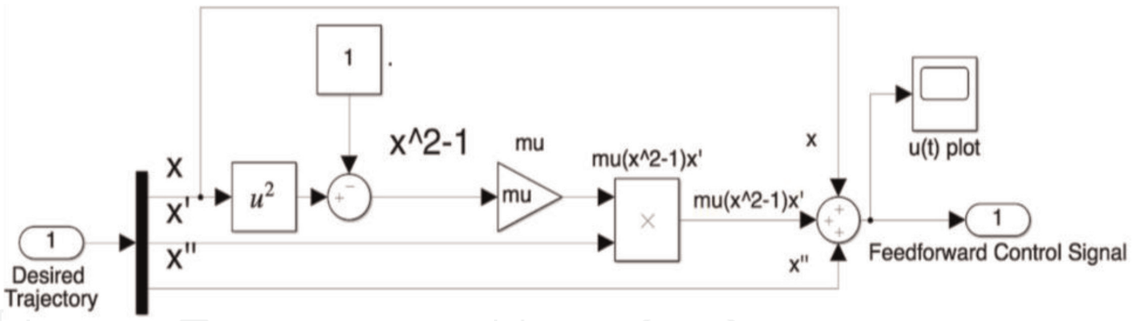


Figure 10.
Inverted Van der Pol system for feedforward controller.

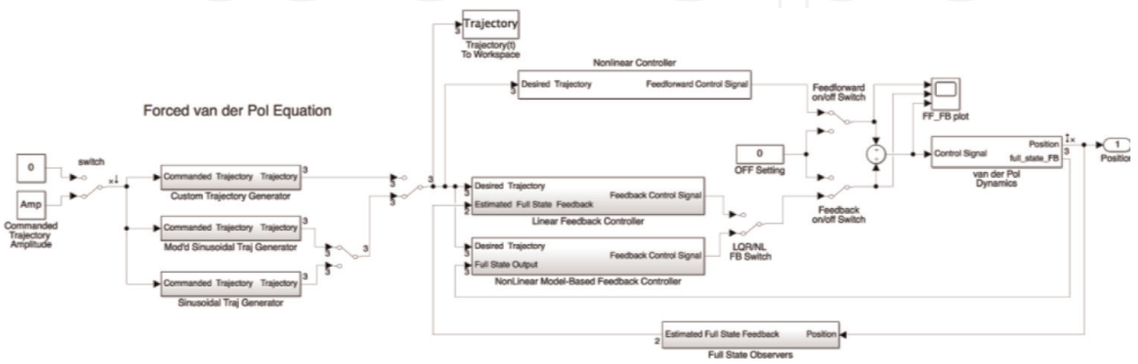


Figure 11.
High level system simulation in MATLAB/Simulink.

The output of the desired trajectory function is then fed into both the feedback and the feedforward controllers. Here, the output of the controllers can be turned on or off as desired to evaluate the performance of the controllers. For deterministic control algorithms it is necessary to assume either full-state feedback, or non-stochastic signals on the output.

2.1 Classical tuning method

One can apply classical tuning methods to the feedback controllers such as PID, and LQR to treat this nonlinear system as a linear system. The PID controller in **Figure 2** can serve as a foundation for multiple variants of a linear feedback controller by modifying the gain coefficients K_i, K_d, and K_p. Due to the ineffectiveness of linear control algorithms on the highly nonlinear VDP plant, the LQR feedback controller will be the only approach illustrated. Calculating the LQR coefficients are somewhat trivial in MATLAB/Simulink if the system description is in a compatible form and the MATLAB function call can be used: [K, S,e] = LQR (A,B,Q,R,N) [29], which gives the relevant K_d, and K_p values. In this instance, the system is described in state-space as shown in Eq. (18):

$$A = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \end{bmatrix} \tag{18}$$

to get a resulting control law of:

$$\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = -K_d\left(\dot{x}_d - \dot{x}\right) - K_p\left(x_d - x\right), \tag{19}$$

or Eq. (22) when coupling the ideal forced VDP feedforward controller with LQR:

$$\ddot{x} - (1 - x^2)\dot{x} + x = \ddot{x}_d - (1 - x^2)\dot{x}_d + x_d - K_d\left(\dot{x}_d - \dot{x}\right) - K_p\left(x_d - x\right), \tag{20}$$

The resulting phase portrait with only incorporating the LQR gains implemented in Eq. (19) are displayed in the left side of **Figure 12**. It can be seen that the stability is affected such that the system steadily grows unbounded and not to an asymptotic limit which is desired for illustrating stability. Investigating the nearby initial conditions does not identify any condition that produces a stable system.

The right side of **Figure 12** illustrates similar results from coupling the ideal nonlinear feedforward controller with the LQR feedback gains. Again, it can be seen that the system grows unbounded with no identified initial conditions providing stability. It can also be seen that the trends in the trajectories are influenced by both the LQR feedback only system and the ideal feedforward controller, although the feedforward controller in not robust enough to bring the system back to an asymptotic limit cycle.

2.2 Isolated trajectory fractionalization

With the results identified in Section 2.1 using linear feedback methods to compensate for trajectory tracking error of the nonlinear VDP system, it is necessary to try something different. One way is to incorporate isolated trajectory fractionalization (ITF). This train of thought assumes the ability of the transient trajectory to be fractured into sub components which can be elegantly stitched together to form the desired transient trajectory. There are many ways to stitch trajectories utilizing the mathematical principles of spline interpolation, Chebyshev polynomials, Lagrange polynomials, and the Runge Kutta methods [30] for example. Here we will illustrate the micro-transient generation to get a better performing transient trajectory via the 8-term Fourier Series fitting method. Under the theory that any periodic signal can be broken up to an infinite series of orthogonal basis functions based on a combination of sines and cosines [30] as defined by Eq. (21):

$$T_n = a_0 + \sum_{i=1}^{n} a_i cos(i\omega t) + b_i sin(i\omega t), \tag{21}$$

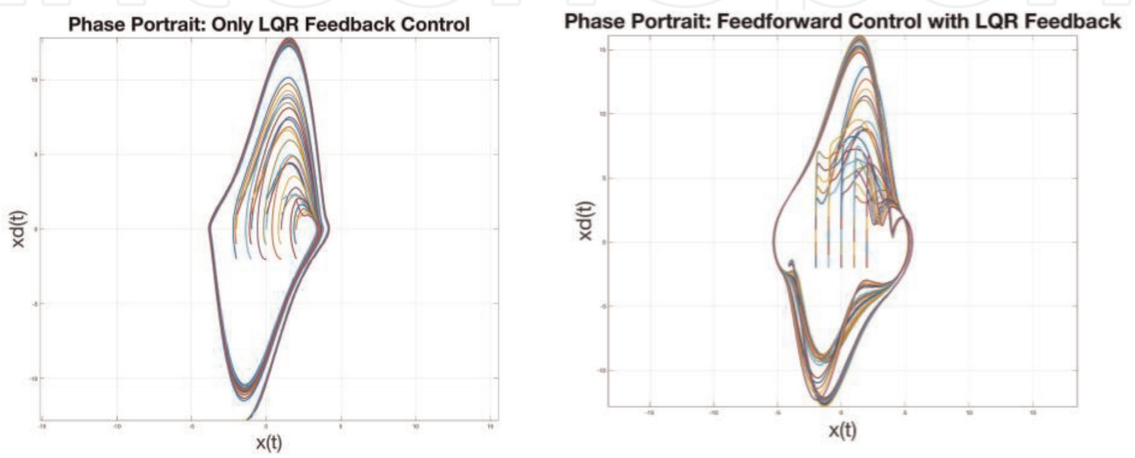


Figure 12.
Applying the optimal linear-quadratic-regulator feedback approach to the forced Van der Pol oscillator.

where T_n is the resulting sum of the functions used to replicate the originally desired trajectory. a_0 is a steady state offset, while a_i and b_i are the corresponding constant coefficients for each orthogonal basis set.

In the case of the sinusoidally forced VDP system illustrated in **Figure 7**, the unwanted transient response is primarily within the first few cycles of oscillations then after the sharp initial transients the tracking error slowly reduces over time until the trajectory is a perfect match to the desired trajectory. One way to split this desired trajectory is to heuristically describe the sinusoid function that will smoothly and quickly travel between the initial conditions and the desired steady state response. Taking advantage of the a-priori knowledge of how sinusoidal function performs with the VDP system a simple sine function with a phase shift is chosen as described by Eq. (22):

$$x_{01} = A_1 \sin\left(t - \frac{\pi}{2}\right), \tag{22}$$

where A_1 is a scaling coefficient proportional to the desired oscillatory radius in the phase plane. And the resulting derivative is:

$$\dot{x}_{01} = A_2 \cos\left(t - \frac{\pi}{2}\right), \tag{23}$$

resulting in a new scaling coefficient A_2. Using Eqs. (22) and (23) as the starting points, the steady state trajectory is then described by Eqs. (24) and (25) in order to match the initial conditions.

$$x_{23} = \frac{(A - 0.8)}{2} \sin\left(t - \frac{\pi}{2}\right) + \frac{\pi}{2} + 0.4, \tag{24}$$

$$\dot{x}_{23} = \frac{(A)}{2} \cos\left(t - \frac{\pi}{2}\right) \tag{25}$$

Next, by using the Fourier decomposition described in Eq. (21), the resulting trajectory can be planned as illustrated in **Figure 13**. This initial attempt is fairly close but still allows for large residual error. **Figure 13** displays the desired total trajectories for x_d and $\dot{x}_d$ with a Fourier fit line plotted for comparison. The Fourier fit line of the resulting trajectory using this transient trajectory fractionalization

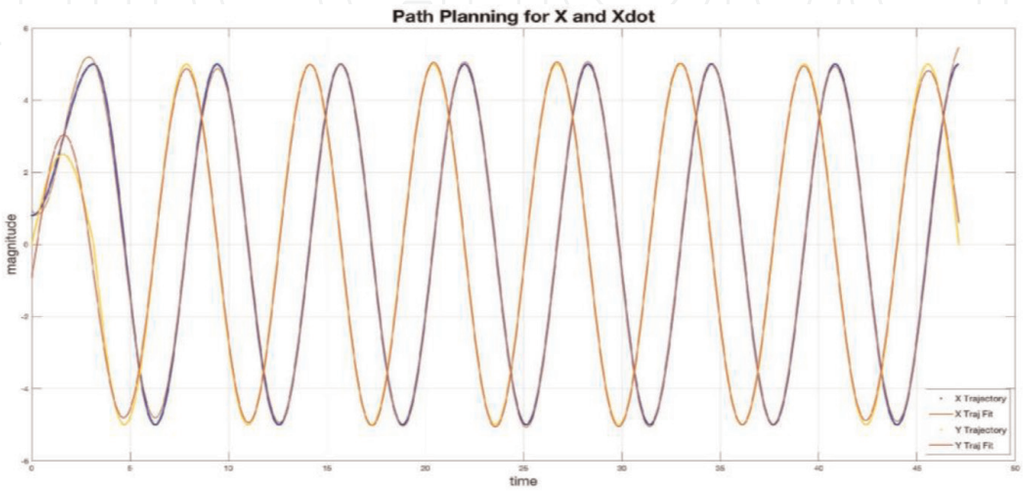


Figure 13.
Trajectory fractionalization via Fourier fitting.

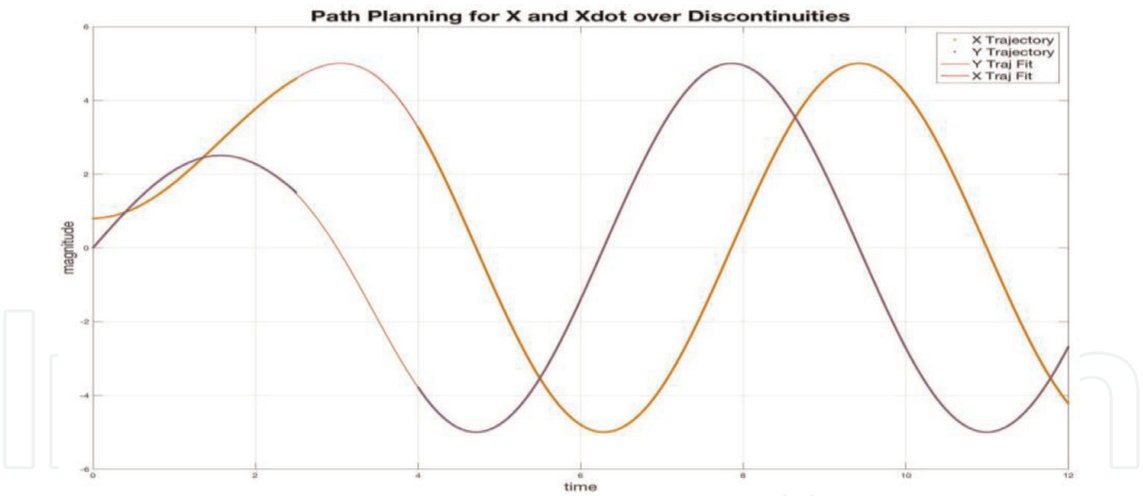


Figure 14.
Isolated trajectory fractionalization via Fourier fitting.

method. One way to mitigate this residual error is by given the fitting algorithm more "flexibility" to merge the trajectories.

To allow the fitting algorithm more flexibility, a large section isolated between the desired trajectories is introduced. The missing data is strategically placed such that the points where the 2nd derivative is zero. This allows for an extremely smooth function across the 1st derivative. Any sharp discontinuities can introduce large unwanted transients. The next iteration in the isolated trajectory fractionalization routine is illustrated in **Figure 14**.

The Fourier fit in **Figure 14** was built with the following parameters:

$$\delta_t = 0.001, t_0 = 0, t_1 = 2500, t_2 = 4000, t_3 = 12000 \tag{26}$$

where δ_t is the time step, t_0 is the initial time, t_1 is the end time of the first trajectory fraction, t_2 is the start of the 2nd trajectory fraction, and t_3 is the final time. The resulting parameters allow the fitting algorithm 1500 (1.5 second) time steps to find a sm0oth fit. The resulting trajectories form the desired phase plane trajectory being fed into the feedforward controller is illustrated on the left in **Figure 15**. The right side of **Figure 15** illustrates the actual results on the output of the VDP system.

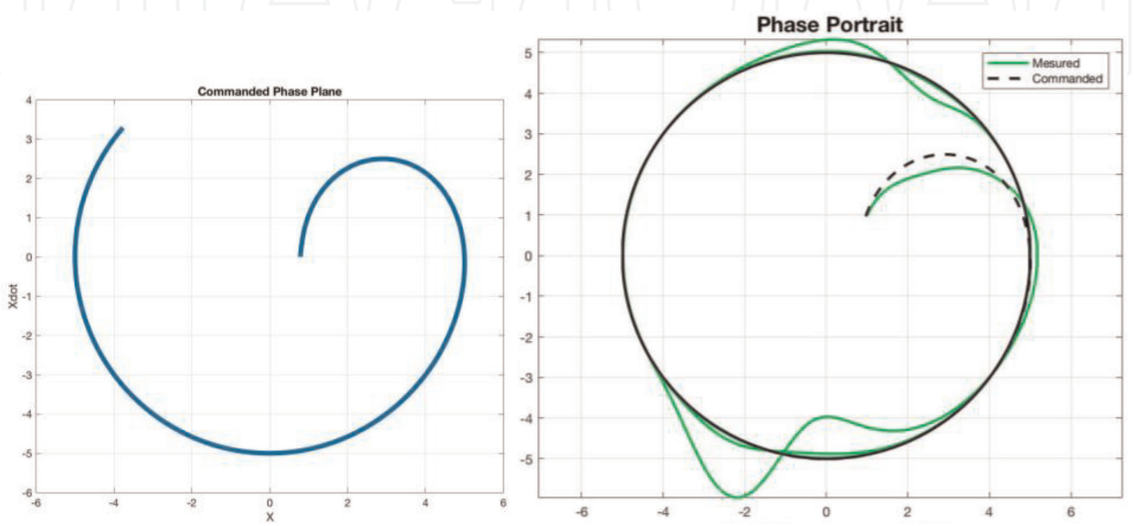


Figure 15.
Phase plane error in forced VDP system using isolated trajectory fractionalization via Fourier fitting.

The corresponding x, $\dot{x}$, and $\ddot{x}$ tracking error is illustrated in **Figure 16**. It's interesting to note that the large disturbance in the trajectories seen if **Figure 15** occur during the transition to the steady-state trajectory and that a spike in the $\ddot{x}$ error happens about 1 second later. The results shown here highlight the need to ensure that multiple derivative states are smooth when implementing the fitting algorithms, due to unforeseen perturbations. Even though the desired phase plane in **Figure 15** is perfectly smooth, the unidentified perturbations in the second derivative are affecting the results.

Utilizing the tracking error values along the entire transient trajectory an objective cost function can be defined and evaluated. In the case of the VDP system, the goal is to match the actual trajectory with the desired trajectory with respect to the phase plane. Therefore the parameters of interest are x, $\dot{x}$, and $\ddot{x}$ along the entire trajectory resulting in the objective cost function, J, in Eq. (27).

$$J = \sqrt{RMSe_x^2 + RMSe_{\dot{x}}^2 + RMSe_{\ddot{x}}^2}, \tag{27}$$

where RMSe is the RMS value of the error along the entire trajectory on each component. This provides a single metric to evaluate how well the actual trajectory fits to the desired trajectory. Once evaluated, a value of J = 1.0604 is achieved using the isolated trajectory fractionalization technique, whereas the objective cost function for the sinusoidal forced VDP system is J = 4.9603. This shows a 78% reduction in trajectory tracking error via this technique.

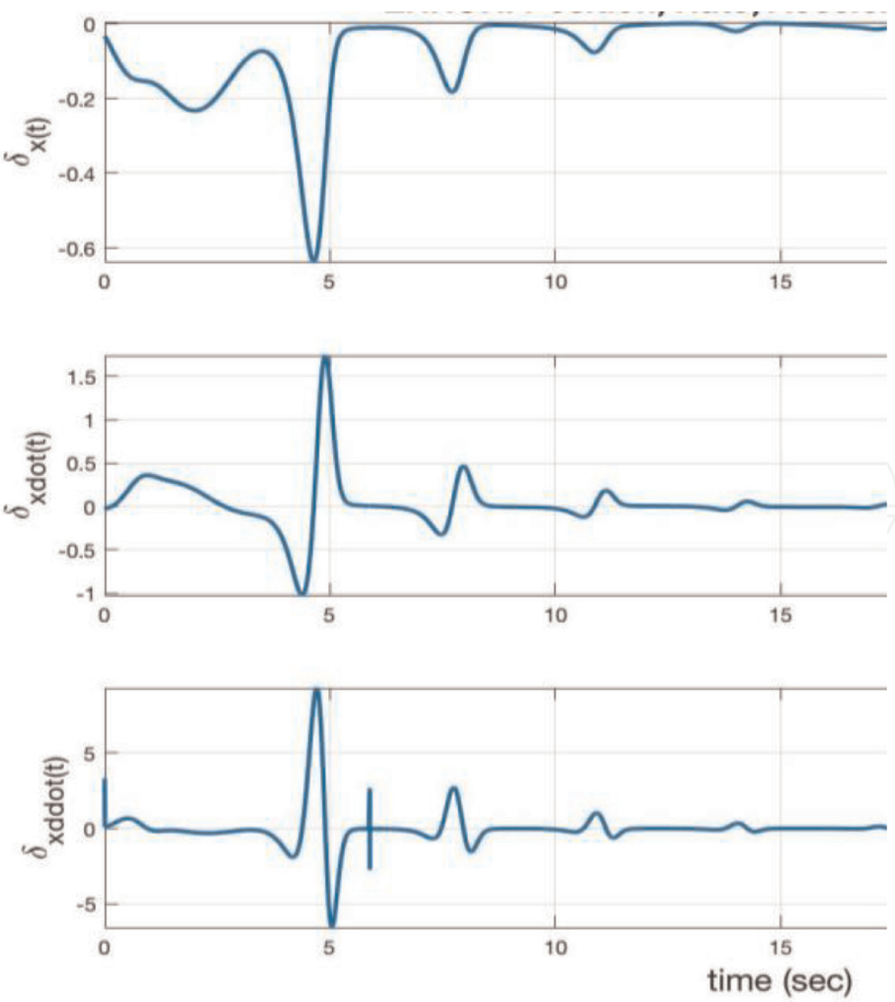


Figure 16.
Trajectory tracking error using ITF.

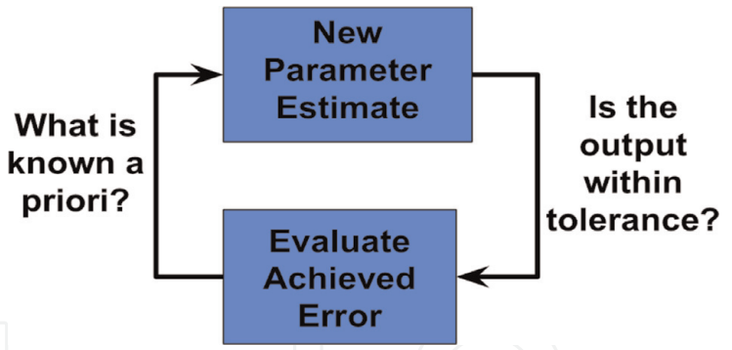


Figure 17.
Cyclical process of deterministic parameter tuning.

2.3 Targeted exponential weighting

2.3.1 1-term targeted exponential weighting

The targeted exponential weighting method is another technique to deterministically modify transient trajectories, and is ideally applied to the extrema in the transients via a-priori knowledge of the system characteristics. By taking advantage of the mathematical approach of exponential decay (or growth), one can modify an existing control trajectory via the cyclical parameter tuning process shown in **Figure 17**.

The parameter tuning process outlined in **Figure 17** starts with a-priori information of the system's characteristic behavior, this is then used to tailor a trajectory for the system. The trajectory is fed through the feedforward controller/plant. The output is then evaluated along the trajectory and compared to the desired. If the error is greater than a given tolerance value then the new information gained from the input parameters are compared to what was previously known and a new set of parameters are generated for the next evaluation. This is repeated until the error is within the given tolerance or until a pseudo-global minimum is found.

Considering the forced Van der Pol system, again it is now known that the initial conditions give rise to the largest perturbations to the ideal trajectory of a perfect circle in the phase plane. With this in mind the targeted exponential weighting technique will be targeted for those initial transient behaviors in order to minimize their impact.

For the forced VDP system under investigation, the initial configuration will take the original forcing function, $x_d = A\sin(t)$, as the ideal trajectory for the position. This will give the desired steady state response. Next, the derivative will be evaluated to get $\dot{x}_d$, and here we need to look at the trajectory error from the baseline forced VDP case to decide the next course of action. The tracking error can be found in **Figure 18**.

$$x_d = A\sin(t), \dot{x}_d = A\left(1 - e^{-10t}\right)\cos(t), \ddot{x}_d = \frac{dx}{dt}\dot{x}_d \tag{28}$$

It can be seen in **Figure 18** that the most severe trajectory transients are in the initial states of x, and $\dot{x}$, in addition to the transients on $\ddot{x}$ beyond t = 1 second. The first focus area will be on $\dot{x}$ because there is no desire to reduce x as that will help preserve the final desired trajectory. Additionally, the large transients are periodic in nature and therefore if the initial spike can be minimized then the following periodic spikes should be greatly reduced. In order to minimize the initial transients in $\dot{x}$ an increasing exponential will be used to force the derivative to zero at time = 0. The desired trajectory response will grow exponentially to the final state and the exponential rate will be tuned deterministically to get the resulting trajectory equations in Eq. (28). Taking the derivative of the tuned exponentially increase

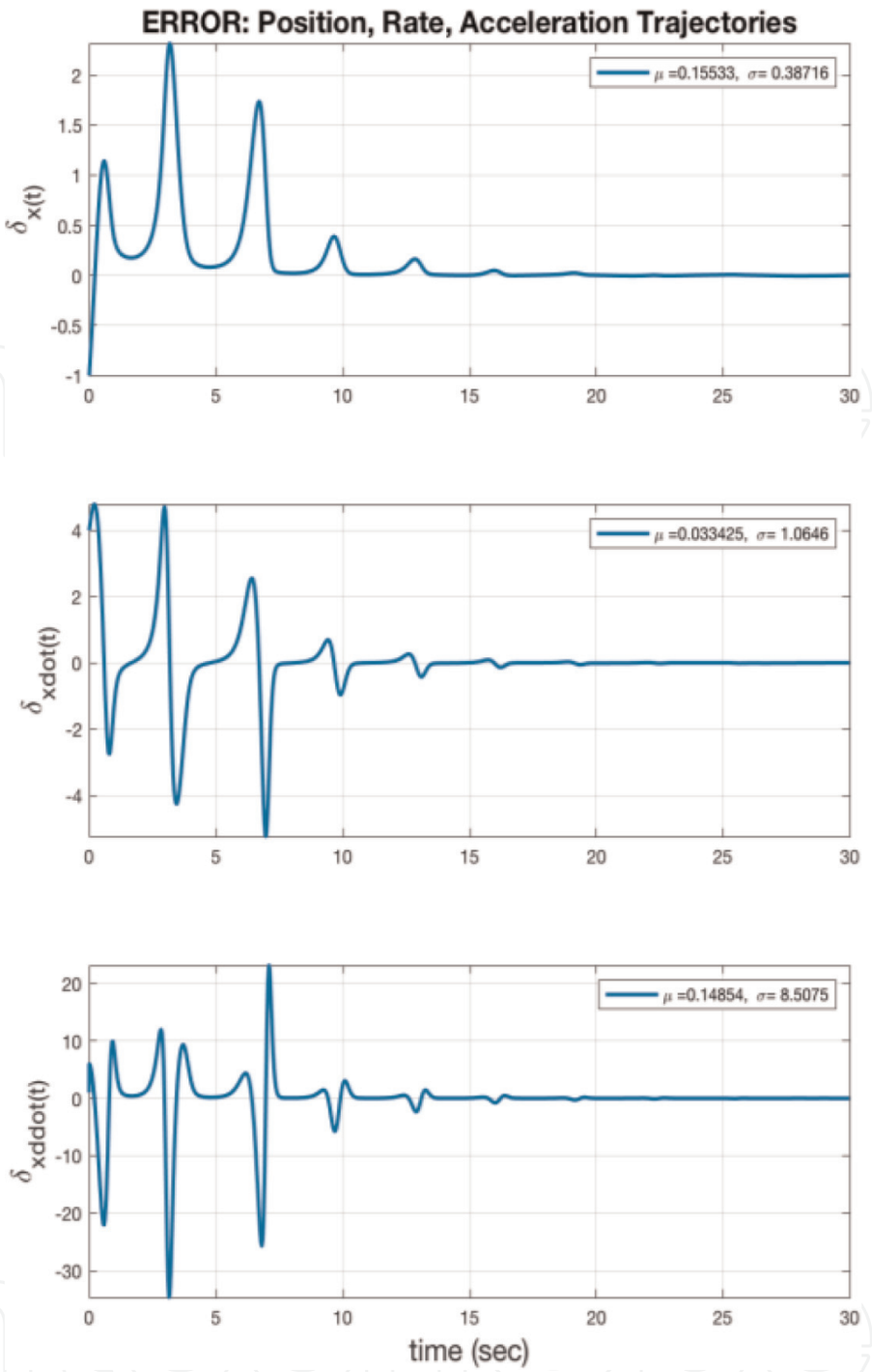


Figure 18.
Trajectory tracking error in forced VDP system.

trajectory will provide $\ddot{x}$. where A = 5, for an arbitrary circle radius of 5 in the phase plane.

This approach takes the ideal forcing function for the Van der Pol oscillator, Asin (ω t), and by modifying the nominal derivative, ω Acos(ω t), via an exponential dithering method $(1 - e^{-Bt})$ manipulates the targeted trajectory section (in this case where t < 2) in order to attempt to minimize the unwanted micro-transients. Generating the three trajectories in x, $\dot{x}$, and $\ddot{x}$ via the expressions in Eq. (28) result in a output phase portrait as illustrated in **Figure 19**, with the residual tracking error on each trajectory component in **Figure 20**.

The benefits by utilizing the targeted exponential weighting method are immediately noticeable when comparing the original phase portrait of the forced VDP system in **Figure** 7, and the results using the isolated trajectory fractionalization

method in **Figure 15**. Additionally, the objective cost function, J, decreases from J = 4.9603, to J = 1.0604, to J = 0.6328 for an overall reduction of 87% from implementing target exponential weighting algorithm. The next step will be to implement the same methodology on $\ddot{x}$ to gain further improvement.

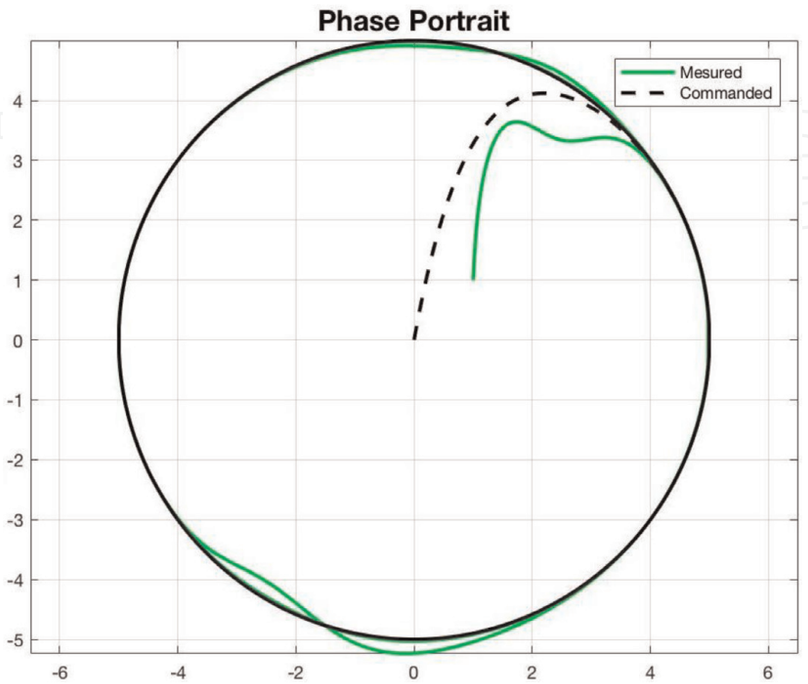


Figure 19.
Phase plane error in forced VDP system using 1-term targeted exponential weighting.

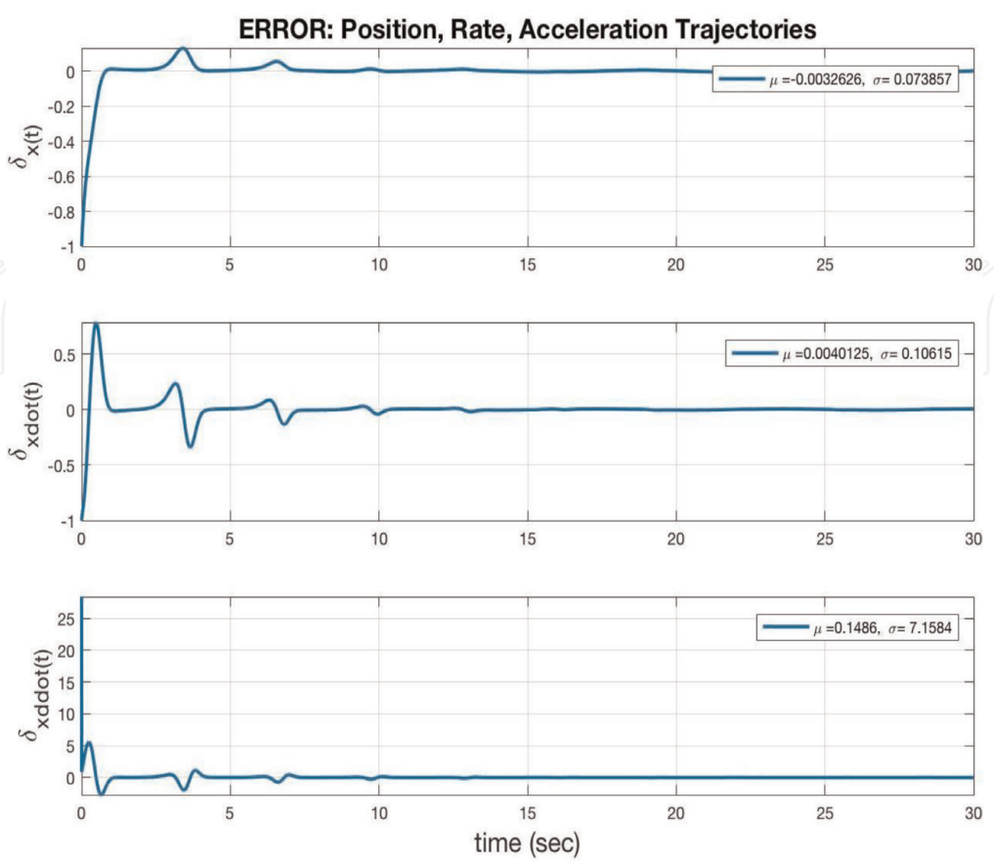


Figure 20.
Trajectory tracking error in forced VDP system using 1-term targeted exponential weighting.

2.3.2 2-term targeted exponential weighting

Using the same methodology in the 1-term method previously described, the exponential dithering will be applied to the 2nd derivative, $\ddot{x}_d$. Starting with Eq. (29) where x_d is of the form: $A(1 - e^{-Bt})\cos(t)$. The direct derivative is of the

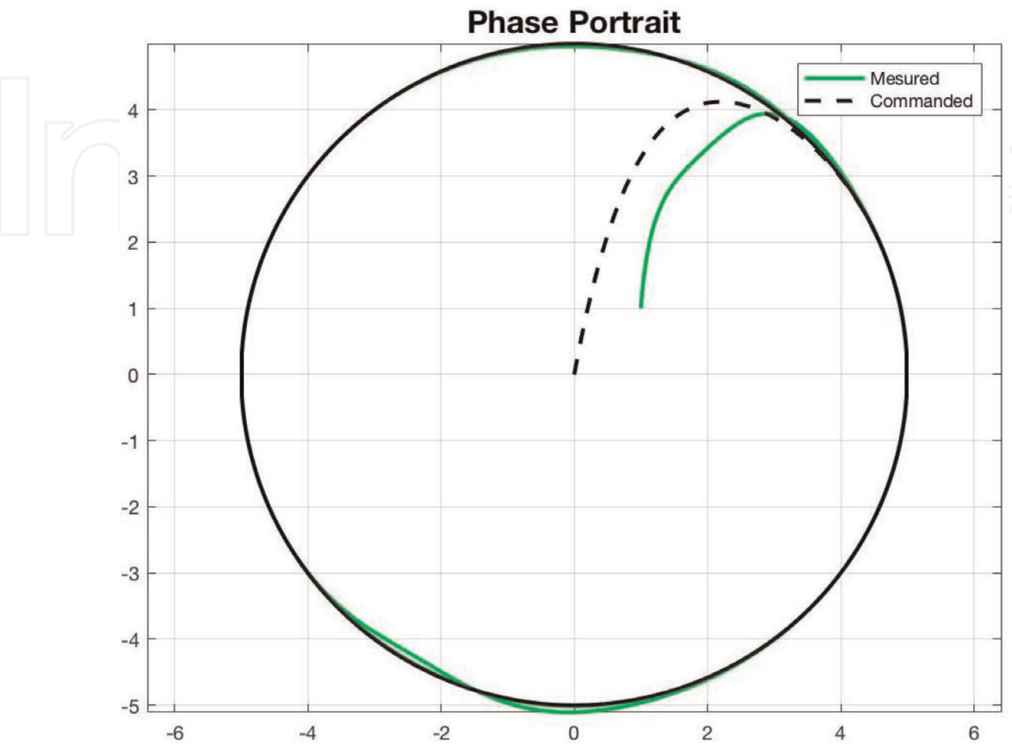


Figure 21.
Phase plane error in forced VDP system using 2-term targeted exponential weighting.

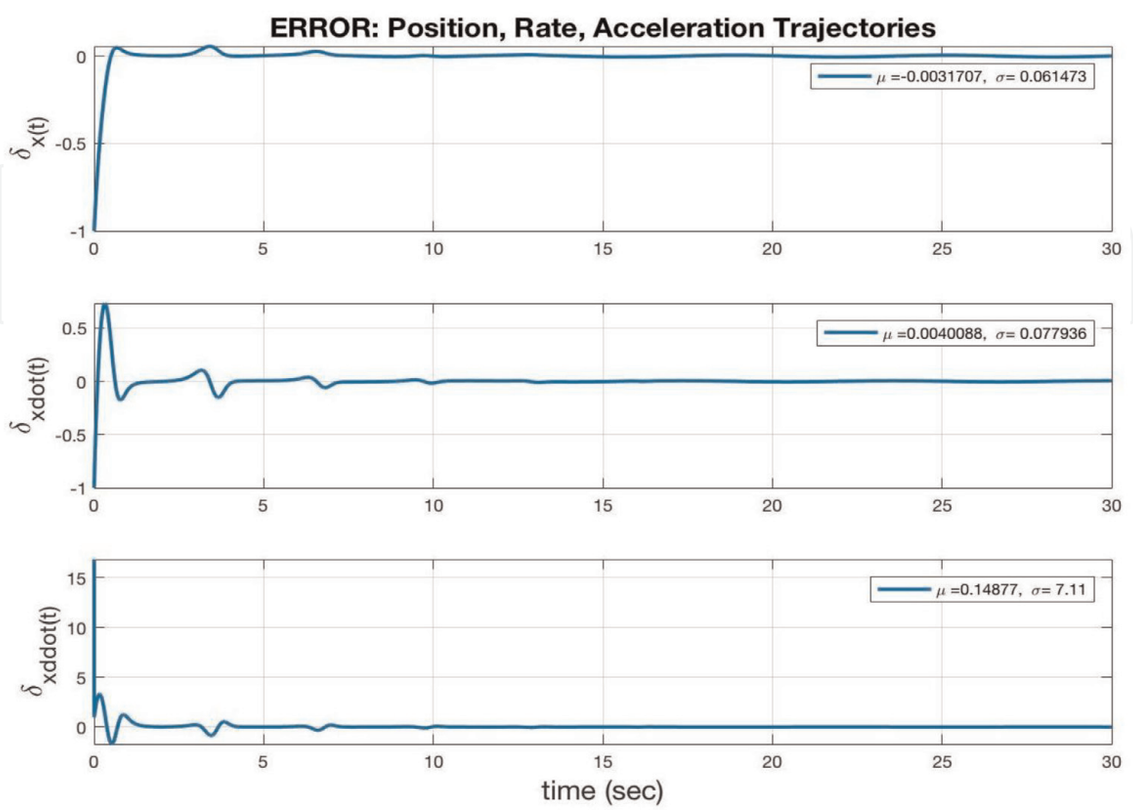


Figure 22.
Trajectory tracking error in forced VDP system using 2-term targeted exponential weighting.

form: $-\mathrm{Ae}^{-\mathrm{Bt}}((\mathrm{e}^{-\mathrm{Bt}}-1)\sin(\mathrm{t})-\mathrm{Bcos}(\mathrm{t}))$. And making the assumption from a heuristic approach that $e^{-Bt} \approx Bt$ in the dithering sine term to get Eq. (30).

$$X_d = Asin(t), \dot{x}_d = A\left(1 - e^{-5.5t}\right)\cos(t), \tag{29}$$

$$\ddot{x}_d = -Ae^{-3.17t}((3.17t - 1)\sin(t) - 3.17\cos(t)) \tag{30}$$

Generating the three trajectories in $x\dot{x}$, and $\ddot{x}$ via the expressions in Eqs. (29) and (30) result in a output phase portrait as illustrated in **Figure 21**, with the residual tracking error on each trajectory component in **Figure 22**.

The 2-term targeted exponential weighting method does indeed show an improvement over the 1-term variation. The resulting improvement based on the objective cost function shows an extra 5% decrease in trajectory tracking error to get J = 0.3758, which results in an overall reduction in tracking error on the order of 92% over the baseline forced Van der Pol system.

3. Conclusion

This chapter presented a small set of deterministic approaches to transient trajectory generation with particular interest in minimizing unwanted micro-transients that may cause havoc on a control system performance. Beginning with a brief introduction to control theory terminology which introduced state-space notation, transfer functions, feedback controllers, and the concept of observers for estimating unknown system states. From there the Van der Pol oscillatory equation was introduced and presented as a system under test to apply the deterministic trajectory generation techniques to.

Using the Van der Pol oscillator, feedforward controllers were introduced through the forced Van der Pol system and initial results of system performance were discussed and evaluated through the use of phase portraits. The micro-transients in the baseline case, the forced Van der Pol system using a forcing function in the form of Asin(t), were illustrated and compared to three different approaches. The first approach utilized a common feedback technique for linear system, the linear quadratic regulator (LQR). This was shown to be unstable.

	Sinusoidal (base)	**LQR FB**	**LQR w/ FF**	**Custom FF trajectory**	**Exp Xdot w/FF**	**Exp Xddot w/FF**
RMSe on x	0.4171	1.9181	0.9624	0.1044	0.0739	0.0616
RMSe on $\dot{x}$	1.065	3.1803	2.2486	0.2206	0.1062	0.078
RMSe on $\ddot{x}$	4.8266	13.8416	11.7283	1.0319	0.6194	0.3624
K_p		−2.6818	−2.6818			
K_d		−0.4142	−0.4142			
K_i		−0.2682	−0.2682			
J(x, $\dot{x}$)	1.1438	3.7139	2.4459	0.2441	0.1294	0.0994
J($x\dot{x}\ddot{x}$)	4.9603	14.3312	11.9806	1.0604	0.6328	0.3758
% error reduction	0	n/a	n/a	78%	87%	92%

Table 1.
Summary of case study results.

The next approach utilized the isolated trajectory fractionalization technique which segmented the desired trajectory into sub-trajectories, and isolated the transition points and applying a Fourier fitting routine to stich the desired trajectories back together. This approach led to a reduction in trajectory error (as evaluated by an objective cost function) by 78%.

The final method was split into two slightly different variants. The first one presented was the 1-term targeted exponential weighting method. This method utilized a-priori knowledge of where the largest micro-transient response and applied dithering techniques to minimize those unwanted transients in the derivative of the desired trajectory. The second variation, the 2-term targeted exponential weighting method, applied a similar approach to the 2nd derivative of the desired trajectory. The resulting improvement over the baseline case was an 87, and 92% trajectory tracking error reduction.

For ease of comparison, the results are summarized in **Table 1**.

Acknowledgements

The author would like to acknowledge the technical guidance and leadership of Dr. Tim Sands, and the support of Air Force Research Laboratory during the research and writing of this chapter.

Conflict of interest

The authors declare no conflict of interest. The views expressed in this paper are those of the author, and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

Author details

Matthew A. Cooper
Air Force Research Laboratory, Albuquerque, New Mexico, USA

*Address all correspondence to: matthew.cooper.17@us.af.mil

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. Distributed under the terms of the Creative Commons Attribution - NonCommercial 4.0 License (https://creativecommons.org/licenses/by-nc/4.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited. BY–NC

References

[1] Huibert K, Raphael S. Linear Optimal Control Systems. 1st ed. New York, NY, USA: Wiley-Interscience; 1972. ISBN 0-471-51110-2

[2] Sands T. Fine Pointing of Military Spacecraft [Ph.D. dissertation]. Monterey, CA, USA: Naval Postgraduate School; 2007

[3] Abdessemed F, Djebrani S. Trajectory generation for mobile manipulators. In: Jimenez A, Al Hadithi BM, editors. Robot Manipulators. Rijeka, Croatia: IntechOpen; 2010. DOI: 10.5772/9200

[4] Barrientos A, Gutierrez P, Colorado J. Advanced UAV trajectory generation: Planning and guidance. In: Lam TM, editor. Aerial Vehicles. Rijeka, Croatia: IntechOpen; 2009. DOI: 10.5772/6467

[5] Lim SJ, Yu SN, Han CS, Kang MK. Palletizing simulator using optimized pattern and trajectory generation algorithm. In: Di Paola AMD, Cicirelli G, editors. Mechatronic Systems. Rijeka, Croatia: IntechOpen; 2010. DOI: 10.5772/8929

[6] Jin TS, Hashimoto H. Trajectory generation and object tracking of mobile robot using multiple image fusion. In: Milisavljevic N, editor. Sensor and Data Fusion. Rijeka, Croatia: IntechOpen; 2009. DOI: 10.5772/6576

[7] Nemec B, Zlajpah L. Automatic trajectory generation using redundancy resolution scheme based on virtual mechanism. In: Rodi AD, editor. Contemporary Robotics. Rijeka, Croatia: IntechOpen; 2009. DOI: 10.5772/7809

[8] Ata AA, Johar H. Trajectory planning of a constrained flexible manipulator. In: Kordic V, Lazinica A, Merdan M, editors. Cutting Edge Robotics. Rijeka, Croatia: IntechOpen; 2005. DOI: 10.5772/4671

[9] Zhu ZH, Shi G. Piecewise parallel optimal algorithm. In: Valdman J, editor. Optimization Algorithms. Rijeka, Croatia: IntechOpen; 2018. DOI: 10.5772/intechopen.76625

[10] Slotine J. Applied Nonlinear Control, Chapter 9. Englewood Cliffs, NJ, USA: Prantice-Hall; 1991

[11] Ogata K. Modern Control Engineering. 5th ed. Upper Saddle River, NJ, USA: Prentice Hall PTR; 2009

[12] Nakatani S, Sands T. Autonomous damage recovery in space. International Journal of Control, Automation and Systems. 2016;**2**:31

[13] Nakatani S, Sands T. Simulation of spacecraft damage tolerance and adaptive controls. In: Proceedings of the 2014 IEEE Aerospace Proceedings, Big Sky; 1-8 March 2014; MT, USA. 2014

[14] Sands T. Physics-based control methods. In: Rushi G, editor. Advances in Spacecraft Systems and Orbit Determination. Rijeka, Croatia: In-Tech Publishers; 2012. pp. 29-54

[15] Cooper MA, Heidlauf PT. Nonlinear feed forward control of a perturbed satellite using extended least squares adaptation and a Luenberger observer. Journal of Aeronautics and Aerospace Engineering. 2018;**7**:205. DOI: 10.4172/2168-9792.1000205

[16] Sands T, Lorenz R. Physics-based automated control of spacecraft. In: Proceedings of the AIAA Space 2009 Conference and Exposition; 14-17 September 2009; Pasadena, CA, USA. 2009

[17] Sands T. Improved magnetic levitation via online disturbance decoupling. Physik Journal. 2015;**1**: 272-280

[18] Smeresky B, Rizzo A, Sands T. Kinematics in the information age. Mathematical Engineering, A Special Issue of Mathematics. 2018;**6**(9):148. DOI: 10.3390/math6090148

[19] Sands T, Bollino K, Kaminer I, Healey A. Autonomous minimum safe distance maintenance from submersed obstacles in ocean currents. Journal of Marine Science and Engineering. 2018; **6**(3):98

[20] Sands T, Bollino K. Autonomous underwater vehicle guidance, navigation, and control. In: Autonomous Vehicles. 2018. DOI: 10.5772/intechopen.80316. Available online at: https://www.intechopen.com/online-first/autonomous-underwater-vehicle-guidance-navigation-and-control [Accessed: 28 December 2018]

[21] Van der Pol B. A theory of the amplitude of free and forced triode vibrations. Radio Review. 1920;**1**: 701-710

[22] Van der Pol B. On "relaxation oscillations" I. Philosophical Magazine. 1926;**2**:978-992

[23] Van der Pol B. The nonlinear theory of electric oscillations. Proceedings of the IRE. 1934;**22**:1051-1086

[24] Van der Pol B, van der Mark J. Frequency de-multiplication. Nature. 1927;**120**:363-364

[25] Van der Pol B. Over relaxatie-trillingen. In: Tijdschrift van het Nederlandsch Radiogenootschap. Vol. 3. 1926. pp. 25-40

[26] Cooper M, Heidlauf P, Sands T. Controlling chaos—Forced van der Pol equation. Mathematics. 2017;**5**:70-80. DOI: 10.3390/math5040070

[27] Baker K, Cooper M, Heidlauf P, Sands T. Autonomous trajectory generation for deterministic artificial intelligence. Electrical and Electronic Engineering. 2018;**8**(3):59-68. DOI: 10.5923/j.eee.20180803.01

[28] Lobo K, Lang J, Starks A, Sands T. Analysis of deterministic artificial intelligence for inertia modifications and orbital disturbances. International Journal of Control Science and Engineering. 2018;**8**(3):53-62. DOI: 10.5923/j.control.20180803.01

[29] Linear-Quadratic Regulator (LQR) Design. Mathworks. Available from: https://www.mathworks.com/help/control/ref/lqr.html [Accessed: 27 December 2018]

[30] Trefethen LN. Approximation Theory and Approximation Practice (Other Titles in Applied Mathematics). Philadelphia, PA, USA: Soc. for Industrial and Applied Math; 2012