PRECISE PHASE PORTRAIT CONSTRUCTION FOR AN INVERTED PENDULUM USING CUBIC BEZIER CURVES

Gökhan DINDIŞ[1*], Abdurrahman KARAMANCIOĞLU[2]

[1]Eskişehir Osmangazi Üniversitesi, MMF, Elektrik Elektronik Müh. Böl. ,ESKİŞEHİR
ORCID No : https://orcid.org/0000-0001-5642-7212
[2] Eskişehir Osmangazi Üniversitesi, MMF, Elektrik Elektronik Müh. Böl. ,ESKİŞEHİR
ORCID No : https://orcid.org/0000-0003-2898-186X

Keywords

Cubic Bezier curve
inverted pendulum
phase portrait

Abstract

A cubic Bezier curve formulation is used for generating phase portrait of an inverted pendulum system. Assuming that the cart and pendulum masses, pendulum length, and friction coefficient values in the inverted pendulum model are unknown, its finite number of experimentally obtained trajectories are utilized as a basis for generating the experimentally unavailable trajectories. It is shown that both the experimentally obtained and the remaining computationally generated trajectories can be formulated as an overlapping sequence of cubic Bezier curves in a computationally efficient manner. In the case of having no model of system dynamics, cubic Bezier curve formulation accurately relates experimentally obtained trajectories to the computationally generated trajectories corresponding to different initial conditions. A case study presented in this manuscript verifies that the computational approach using a cubic Bezier curves formulation has far better approximation performance compared to the ones obtained by the linear interpolation.

BİR TERS SARKAÇ İÇİN KÜBİK BEZİER EĞRİLERİNİN KULLANILARAK YÜKSEK DOĞRULUKLU FAZ PORTRESİ ÇIKARIMI

Anahtar Kelimeler

Kübik Bezier eğrisi
ters sarkaç
faz portresi

Öz

Bir ters sarkaç sisteminin faz portresinin oluşturulmasında kübik Bezier eğrisi formülasyonu kullanılmıştır. Sonlu sayıda elde edilen deneysel yörünge verileri kullanılarak, araç ve sarkaç kütlelerinin, sarkaç boyunun ve sürtünme katsayı değerlerinin bilinmediği durumlarda deneysel olarak elde edilenlerin dışındaki yörüngelerin yüksek doğrulukla çıkarılabilmesi sağlanmıştır. Deney yapılmadan hesaplanan yörüngelerin elde edilmesinde içiçe geçmiş Bezier eğrileri kullanılarak verimli bir hesaplama yapılabildiği gösterilmiştir. Verilen keyfi başlangıç şartları için, system dinamiği modelinin bilinmemesi durumunda, deneysel verilerden kübik Bezier eğrisi formülasyonu ile elde edilen yörüngeler, model dinamiği bilgisi ile hesaplanan yörüngelerle yüksek doğrulukla uyuşmaktadır. Makalede sunulan örnek çalışma, kübik Bezier eğrileri ile hesaplama yaklaşımının doğrusal interpolasyon ile elde edilebilecek değerlere göre çok daha yüksek yaklaşıklık performansı gösterdiğini doğrulamaktadır.

Araştırma Makalesi		Research Article	
Başvuru Tarihi	: 13.05.2020	Submission Date	: 13.05.2020
Kabul Tarihi	: 26.08.2020	Accepted Date	: 26.08.2020

1. Introduction

Phase portrait is a graphical representation of the trajectories of a dynamical system in the phase plane. The dependent variables of a dynamic system form the axes of the phase plane. For the sake of visual perception, phase plane analysis primarily applies to systems described by second order differential equations. However, in case of a higher order system, it may be possible to decouple the system into low order parts and perform the phase plane analysis thereafter.

In this study we aim to construct the phase portrait for an inverted pendulum (IP) system which consists of a pendulum attached to a moving cart.

* Sorumlu yazar; e-posta : gdindis@ogu.edu.tr

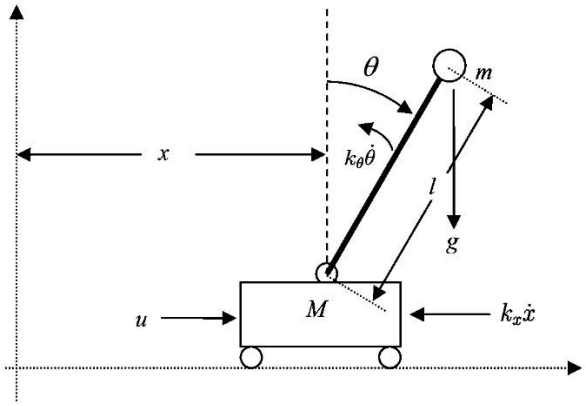

Figure 1 The inverted pendulum system

The inverted pendulum system has fourth order dynamics with the dependent variables pendulum angle θ, cart position x, and their derivatives $\dot{\theta}$, and $\dot{x}$ (Figure 1). Its nonlinear dynamics for an energy conserving configuration is derived in (Stefani, Shahian, Savant, and Hostetter, 2002). A generalization to an energy dissipating version can be obtained by incorporating the friction terms. For a clear presentation of the derivations one may refer to (Chalupa and Bobal, 2008). Thus, a differential equation model of an IP system accounting for the cart and pendulum frictions is

$$\begin{aligned} &(m+M)\ddot{x} + ml\ddot{\theta}\cos\theta + k_x\dot{x} - ml\dot{\theta}^2\sin\theta = u \\ &l\ddot{\theta} - g\sin\theta + k_\theta\dot{\theta} + \ddot{x}\cos\theta = 0 \end{aligned} \quad (1)$$

Where M, m, l, and g denote mass of the cart, mass of the pendulum, length of the pendulum, and gravitational acceleration respectively. The symbols kx and kθ denote cart and pendulum friction coefficients respectively. In the equation above, u denotes the force input applied to the cart. In control engineering community, due to its nonlinear and underactuated nature, developing control strategies to swing-up the pendulum is a challenging problem (Aström and Furuta, 2000). Its phase portrait is useful for the designers for driving the pendulum to its vertically upright state, $(\theta, \dot{\theta})$=(0,0), where its computational generation is of interest in this manuscript. In the case of knowing the system parameters M, m, l, kx, and $k\theta$ there are numerical methods, such as the method of isoclines, to construct the phase portrait in a straightforward way (Slotine and Li, 1991). However, unknowing these parameters, designers have to obtain the trajectories experimentally. A complete set of trajectories to form the phase portrait requires conducting too many experiments which is not practical. However, there are researches to identify certain features of the phase portrait using measurement data of a few number of experiments. For instance, in (Dındış and Karamancıoğlu, 2016) a finite number of experiments are conducted to predict trajectory features of an inverted pendulum that leads to a design of swing-up strategy. In this study we exploit cubic Bezier curves (CBC) for a numerically efficient generation of phase portrait of an IP system with unknown parameters utilizing a few experimental data to be explained shortly.

Bezier curves, introduced originally by P. F. de Casteljau, and then by P. Bezier, parametrize certain curves in a powerful way (Farouki, 2012). They are widely used in computer aided design due to its ability of manipulating geometric shapes in a computationally efficient way. Its detailed history and underlying philosophy are discussed in (Farouki, 2012; Yamaguchi, 1988).

Cubic Bezier curves have been shown to approximate circular arcs effectively. Even though it is impossible to draw an exact circle by one Bezier curve, a quarter of a circle (90° arc) can be approximated by a cubic Bezier curve with an error less than 1% in the radius (Yamaguchi, 1988; Riskus, 2006). This fact has motivated researchers, particularly in robotics area, to represent motion trajectories by a concatenated sequence of Bezier curves. Among numerous examples, one may see, for instance, (Zdesar, Skrjanc, and Klanjar, 2013). Also, in (Kim, 2020) a two-wheeled robot trajectory is constructed when only a few way-points are available to reach a nearby target state. The proposed method interpolates way-points by the combination of a straight line and a fifth-order Bezier curve along which the curvature varies continuously. In (Cimurs and Suh, 2020), a path from the start to goal position is obtained using a path planner algorithm. In their work, to find a locally time-optimal smooth trajectory between path planner output nodes, an optimization of Bezier curve control point positions is employed. In (Wang, Luo, Fang, and Yuan, 2018), it is stated that, the existence of the path dependent dynamic singularities limits the volume of available workspace of free-floating space robot and induces enormous joint velocities when such singularities are met. In order to overcome this problem, an optimal joint trajectory planning method using forward kinematics equations of free-floating space robot is presented. In the process, Bezier curves are applied to describe the joint trajectories.

In the present manuscript, we exploit the fact that IP angular motion trajectories are representable with high accuracy by concatenation of CBCs. Regarding this, we conduct few experiments to obtain free fall trajectories of an IP, and represent each of the so obtained trajectories as an overlapping sequence of CBCs. Using this data set we computationally generate the remaining trajectories in the phase portrait. We also present an algorithm for computing a trajectory corresponding to an arbitrary initial condition.

In the following section we briefly provide a background on the main tool, CBCs, of the manuscript. Following this, representing trajectories of an IP using overlapping CBCs is introduced. In the third section, a case study that verifies validity of our phase portrait construction approach through laboratory experiments using an inverted pendulum (See Figure 2) is presented. The concluding remarks will take place in the last section.

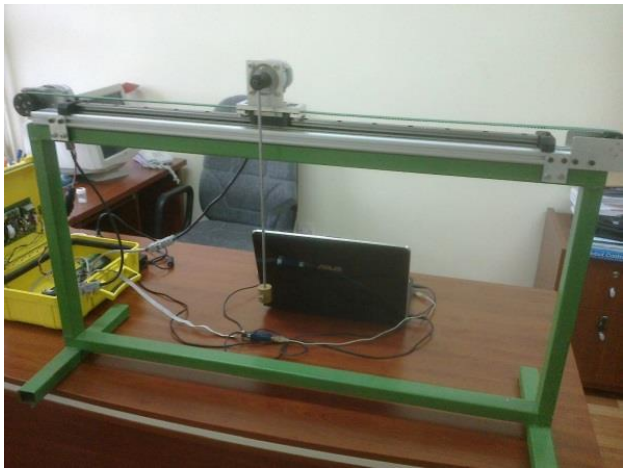

Figure 2 The experimental setup

2. Cubic Bezier Curves in Constructing Phase Portrait of an Inverted Pendulum

A cubic Bezier function $B(t) \triangleq (x(t), y(t))$ is a polynomial with restricted range defined for $0 \le t \le 1$. The function is completely specified with four control points (C_0, C_1, C_2, C_3) corresponding to the parameter values $(0, t_1, t_2, 1)$. The B values corresponding to these t parameters are denoted by $(0, B_1, B_2, B_3)$. The control points C_0 and C_3 equal B_0 and B_3 respectively. However, the control points C_1 and C_2 are not necessarily on the curve; they are on the tangent lines passing through B_0 and B_3 respectively, more explicitly, C_1 is the direction of the curve at B_0, and to B_3 the curve comes from the direction C_2. The cubic Bezier curve in terms of the control points is written as

$$B(t) = (1-t)^3 C_0 + 3(1-t)^2 t C_1 + 3(1-t) t^2 C_2 + t^3 C_3, \quad t \in [0,1] \qquad (2)$$

where $t^n(1-t)^{3-n}, n = 0,1,2,3,$ are called Bernstein basis polynomials for a cubic Bezier function.

For a CBC, in the sequel, we use data points corresponding to equispaced parameter values $(0, \frac{1}{3}, \frac{2}{3}, 1)$, which may be viewed as a common sense when all the data are equally reliable. Regarding this, given the data points (B_0, B_1, B_2, B_3) on the curve corresponding to the parameter values $(0, \frac{1}{3}, \frac{2}{3}, 1)$, one can easily derive the CBC in terms of the data points as

$$\begin{aligned} B(t) = B_0 &+ (-5.5B_0 + 9B_1 - 4.5B_2 + B_3)t \\ &+ (9B_0 - 22.5B_1 + 18B_2 - 4.5B_3)t^2 \\ &+ (-4.5B_0 + 13.5B_1 - 13.5B_2 \\ &+ 4.5B_3)t^3 \end{aligned} \qquad (2)$$

Next, in this section, we use a Bezier curve formulation to represent trajectories of inverted pendulum obtained from free fall experiments. Let initial part of a trajectory corresponding to the free fall of an inverted pendulum from the initial condition $(\theta(0), \dot{\theta}(0)) = (\theta_0, 0)$ to the second time it reaches zero velocity be denoted by P (For a pictorial representation, see Figure 3-a). This partial trajectory corresponds to the initial angular cycle of the pendulum motion, there is no need to generate remainder of the trajectory P because it repeats the same pattern until it reaches the equilibrium state (0, 0). The repeated components of the pattern is already part of the phase portrait corresponding to some initial condition, and this initial condition coincides with a final condition of some other trajectory. Since the phase portrait consists of all P-like trajectories, any complete trajectory can be obtained by appending P-like trajectories of matching endpoints. Let a trajectory P take T_P seconds, and be represented by N points. We define the representative points P_i as $P_i := P\left(i \times \frac{T_P}{N}\right), i = 0,1,\dots,N-1$ (See Figure 3-b). We represent the trajectory P using a sequence of overlapping $N-3$ CBCs, P_{B0}, P_{B1},..., $P_{B(N-4)}$ (See Figure 4-a, for instance, for P_{B0}, P_{B1}, P_{B2} and P_{B3}). Next we parametrize P by a parameter u whose values lie between 0 and 1. In forming the trajectory P, $N-3$ CBCs are used. These curves are not end to end connected; each curve's last two segments overlap with the following one's first two segments, excluding the ending curve $P_{B(N-4)}$. Compared to the end to end connections of the curves, in this configuration only the middle segment of each CBC is used. The ends of P are excluded from this structure, because, the initial and final ends of P do not overlap with middle segments of any CBC. Below, the parametrization rule indicating which segments of P_{Bi} are used in representing the trajectory P for any value of the parameter u is given:

$$P = \begin{cases} P_{B0}, 0 \le u \le 2 \times \frac{1}{N-1} \\ P_{Bi}, (i+1) \times \frac{1}{N-1} \le u \le (i+2)\frac{1}{N-1}; \ i = 1,\dots,N-5 \\ P_{B(N-4)}, (N-3) \times \frac{1}{N-1} \le u \le 1 \end{cases} \qquad (3)$$

Using the parametrization procedure above with the points P_i, $i = 0,1,\dots,N-1$, one can represent the trajectory P by a sequence of overlapping CBCs. Notice that, excluding the first and last CBCs P_{B0} and $PB_{(N-4)}$, the parametrization (3) uses only the middle segment of each CBC P_{Bi}, $i = 0,1,\dots,N-5$. The segments of P_{Bi} used in parametric representation of P are shown in

Figure 4-b. Since middle segment of a CBC is influenced by the endpoints of the curve, which are overlapping with the preceding and succeeding CBCs, the so-formed overall curve P is smoother and does the approximation better. The CBCs P_{B0} and $P_{B(N-4)}$ are located at the ends of P , therefore, two segments of each one are used in the parametrization. In computational viewpoint, a vector containing representative points P_i, $i = 0,1, \dots, N-1$, and the rule (3) for parametrizing P suffice for generating the trajectory.

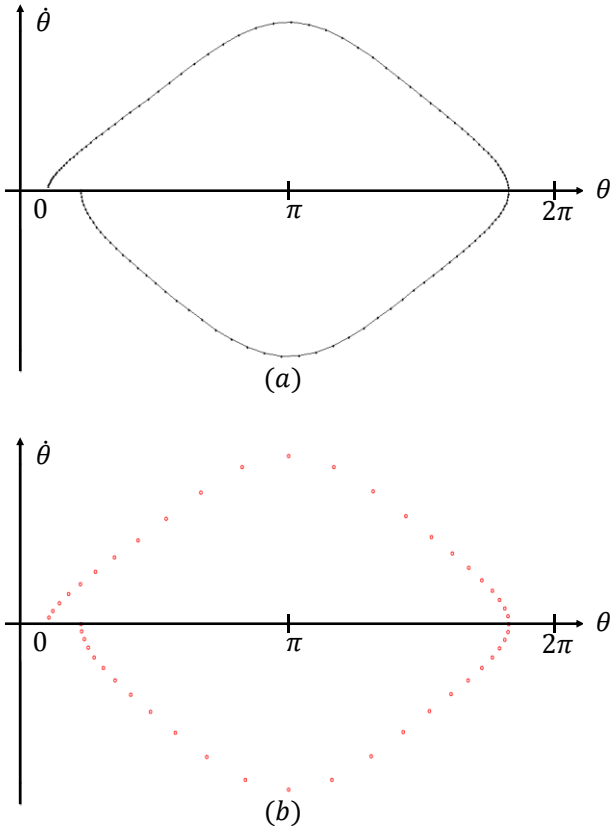

Figure 3 Descriptions of (a) P (b) P_i

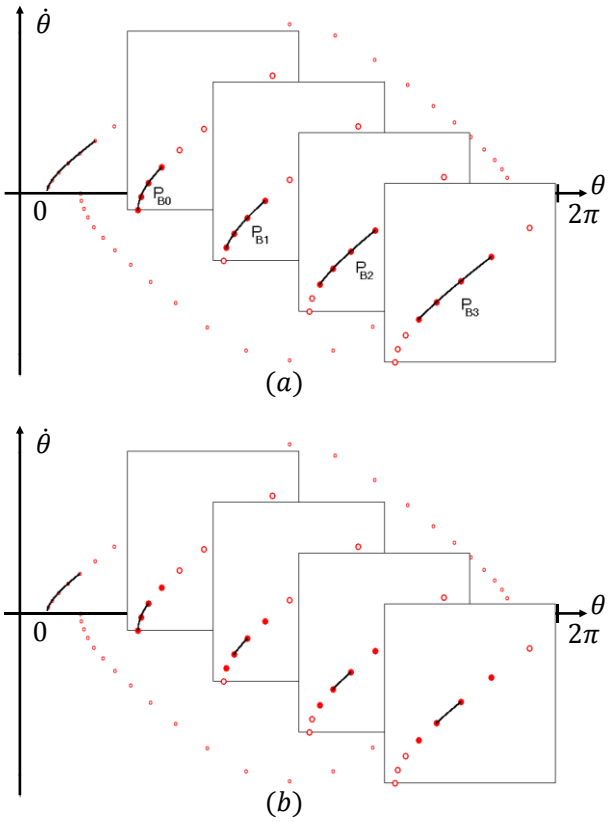

Figure 4 Descriptions of (a) P_{Bi} (b) segments of P_{Bi} used in parametric representation of P

We start with four experiments with equidistant initial conditions to form a practically useful basis for generating the remaining non-experimental trajectories. Of course, using more experimentally obtained trajectories improves approximation quality of the non-experimental trajectories. Likewise, fewer experimental trajectories or non-equidistant initial conditions cause degradation of the results. Let us denote the trajectories corresponding to the experiments with the initial conditions

$$\big(\theta(0), \dot{\theta}(0)\big) = \big((0.1\pi, 0), (0.3\pi, 0), (0.5\pi, 0), (0.7\pi, 0)\big) \quad (4)$$

by (P, Q, R, S). Let each experimental trajectory be presented by its samples P_i; $i = 0,1, \dots, N-1$. Sampled point trajectories (P, Q, R, S) resulting from the experiments are shown in Figure 5. It is shown in the sequel that the CBC formulation well fits to the trajectory description as a function of a continuous parameter in an efficient way.

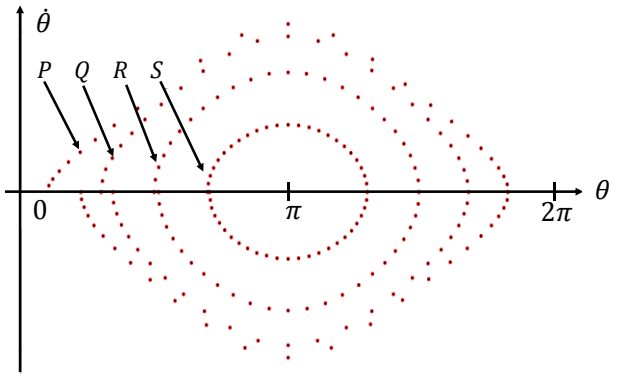

Figure 5 Sampled point trajectories (*P, Q, R, S*)

Next we computationally construct trajectories lying between the experimental ones P and S by Bezier interpolation of the experimental trajectories (*P, Q, R, S*). Let the computationally constructed trajectory corresponding to an initial condition $(\theta(0), \dot{\theta}(0)) = (\theta_A, 0)$ be denoted by *A*. Also, let us define an intercurve parametrization variable ϑ by associating its values $(\vartheta_P, \vartheta_Q, \vartheta_R, \vartheta_S) \triangleq (0, \frac{1}{3}, \frac{2}{3}, 1)$ to the experimental trajectories (*P*, *Q*, *R*, *S*). Thus, corresponding to a parameter $\vartheta_A \in [0,1]$, the computationally constructed trajectory A_i can be obtained by calculating its representative points as in (2):

$$A_i = P_i + (-5.5P_i + 9Q_i - 4.5R_i + S_i)\vartheta_A + (9P_i - 22.5Q_i + 18R_i - 4.5S_i)\vartheta_A^2 + (-4.5P_i + 13.5Q_i - 13.5R_i + 4.5S_i)\vartheta_A^3, \quad i = 0,1,\dots,N \qquad (5)$$

For a better intuition of parametrization variable ϑ_A, one may notice that $\vartheta_A = 0$ corresponds the representative points of the outermost curve *P* , and $\vartheta_A = 1$ corresponds that of the innermost curve *S*. The above obtained representative points of *A* can be used to form CBCs A_{Bi}, $i = 0, 1,..., N - 3$ in a similar way the P_{Bi}'s are formed . This sequence of CBCs with the parametrization rule (3) can be used to generate the trajectory *A*. The steps followed in constructing *A* are repeated for the initial conditions of interest to form any trajectory of the phase portrait.

Using the procedure above, Figure 6-a shows phase portrait corresponding to initial condition set (4). Also consider another initial condition set

$$(\theta(0), \dot{\theta}(0)) = ((0.05\pi), (0.35\pi), (0.65\pi), (0.95\pi)) \qquad (6)$$

which is not equally spaced as the set (4). Using the procedure described above, its phase portrait is shown in Figure 6-b. It will be further illustrated that the procedure above is applicable to experimental data obtained for different initial condition sets. Due to the intrinsic nature of the interpolation, in an ideal computational environment having no round off or truncation errors, interpolation with more, consequently spatially closer, experimental data always yield improved results. In the case of using more experimental trajectories, these trajectories can be grouped as overlapping four trajectories, and interpolation scheme used for the data points of a single trajectory can be extended to the interpolation of experimentally obtained trajectories.

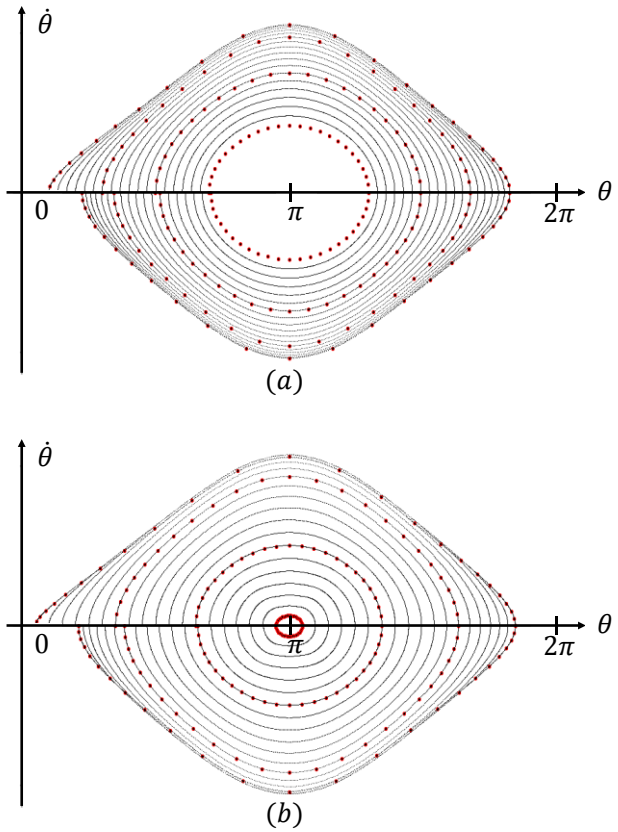

Figure 6 Phase portrait corresponding to (a) initial conditions set (4), (b) initial conditions set (6)

In phase portrait generation we plot the trajectories corresponding to the initial conditions whose velocity component is restricted to zero. This has allowed a simple calculation of ϑ_A . However, in constructing a trajectory for an arbitrary initial condition $(\theta_A, \dot{\theta}_A)$ the ϑ_A calculation needs more computational effort. The algorithm presented below calculates ϑ_A value for an arbitrary initial condition:

Initialization

If $\dot{\theta}_A > 0$ set s = 1; else set s = -1.

Set parameters $\vartheta_1 = \frac{1}{2};\ u_1 = \frac{1}{2} - s \times \frac{1}{4}$

Set step sizes $\vartheta_{0\Delta} = \frac{1}{2};\ u_{0\Delta} = \frac{1}{4}$.

Repeat the following for i = 1 to i = M

Set i-th step size $\vartheta_{i\Delta} = \frac{\vartheta_{(i-1)\Delta}}{2}$

Use $\vartheta_A = \vartheta_i$ in (5) to obtain trajectory *A*.

Repeat the following for j =1 to j = M

Set j-th step size $u_{j\Delta} = \frac{u_{(j-1)\Delta}}{2}$

If θ_A is to the right of $A(u_j)$ then

$u_{j+1} \leftarrow u_j + s \times u_{j\Delta}$; else $u_{j+1} \leftarrow u_j - s \times u_{j\Delta}$

End

If $\dot{\theta}_A$ is above the $A(u_M)$ then

$\vartheta_{i+1} \leftarrow \vartheta_i - s \times \vartheta_{i\Delta}$; else $\vartheta_{i+1} \leftarrow \vartheta_i + s \times \vartheta_{i\Delta}$

End

The outline of how the above algorithm functions is as follows: The flag *s* let the program know which half of the phase portrait contains the initial condition. Then outer loop of the program sets a trajectory *A* as a Bezier interpolation of (*P*, *Q*, *R*, *S*). Upon setting the trajectory *A*, the inner loop finds its closest point to the first component of the given initial condition in a geometrically converging manner. After finding the point, the outer loop checks whether it is above or below the second component of the given initial condition. Then the parameter ϑ is adjusted in a geometrically converging manner to try another trajectory *A*. In rough terms, the above algorithm tries u and ϑ values in a manner used by the well-known bisection algorithm so that a geometrically improving result, at every step, is obtained.

In the following section we present computationally constructed phase portraits using a limited number of experimental trajectories. Also we generate a trajectory corresponding to an arbitrary initial condition and, for verification purpose, compare this to the experimentally generated one. Scientific and publication ethics guidelines are followed in all experimental work.

3. A Case Sudy

The experimental data used in this section are obtained from an inverted pendulum system shown in Figure 2. Every trajectory is represented by *N* = 49 points. In the ϑ_A - finding algorithm, *M* = 8 steps is used in both loops. The data is filtered by a low pass filter to eliminate unmodelled dynamics and noise effects. The phase portraits generated for two different sets of initial conditions have already been presented in Figure 6 of the previous section. Their linearly interpolated counterparts are given in Figure 7 for a comparison. In this figure, trajectories generated by the linear interpolation are constructed as linear combinations of the neighboring experimental trajectories. Besides, every data within a trajectory is connected by using straight line segments.

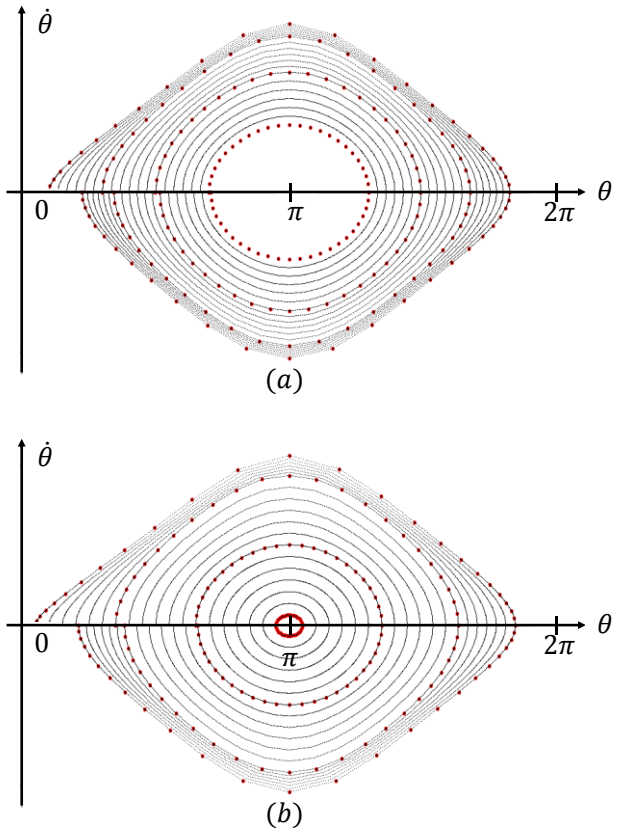

Figure 7 Linearly interpolated phase portrait corresponding to (a) initial conditions set (4), (b) initial conditions set (6)

We below compare the trajectories generated by using CBC formulation and those generated by linear interpolation to the experimental trajectories. For a quantitative analysis, we pick representative

trajectories corresponding to interpolation parameters $\vartheta = 0.25$, $\vartheta = 0.50$, $\vartheta = 0.75$. For each of these trajectories we show the deviations from the experimental trajectories throughout the range of parameter u in Figure 8.

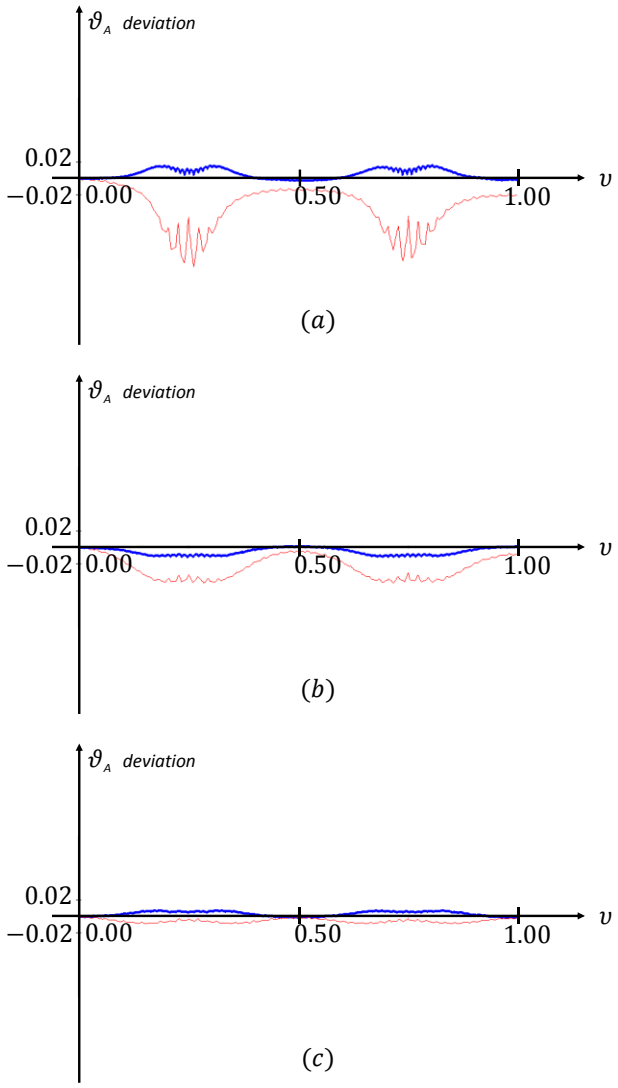

Figure 8 Deviation of CBC formulated (bold lines) and linearly interpolated trajectories (thin lines) from the experimental trajectories when (a) $\vartheta = 0.25$, (b) $\vartheta = 0.50$, (c) $\vartheta = 0.75$

The deviation graphics in Figure 8 show that for significantly long sections of the graphics, compared to the Bezier interpolated trajectories, linearly interpolated trajectories are more distant to the actual trajectories. The performance of the CBC approximation is obviously far better than the linearly interpolated ones.

The second group of tests are intended to show performance of the ϑ_A - finding algorithm. For this, approximation performances of trajectories generated for arbitrarily chosen initial conditions are tested. We ran the ϑ_A-finding algorithm for the experimental trajectories (P, Q, R, S) obtained for the initial condition sets (4) and (6). Using each experimental trajectory set we find the trajectories passing through the arbitrarily chosen test points (a) $(\theta_A, \dot{\theta}_A) = (0.840\pi, 15.174)$ and (b) $(\theta_A, \dot{\theta}_A) = (1.504\pi, 4.719)$. The resulting trajectories are shown in Figures 9 and 10. The approximation performance for each case is shown by plotting the generated and experimental trajectories on the same graphics. The graphics show that the deviations of the generated trajectories from their experimental counterparts are small enough to be detected at this scale.

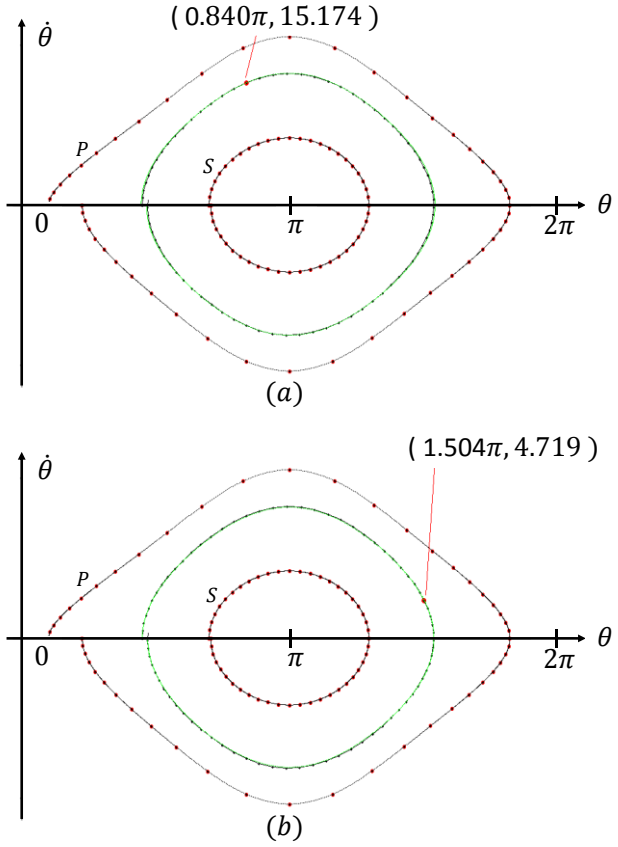

Figure 9 Trajectories passing through the test points (a) $(0.840\pi, 15.174)$ and

(b) $(1.504\pi, 4.719)$ when using experimental trajectories with initial conditions (4)

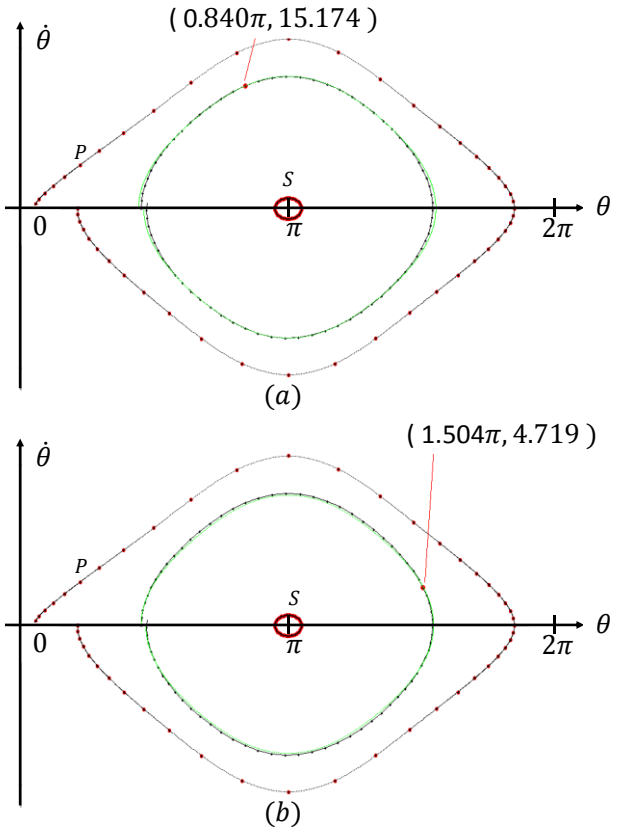

Figure 10 Trajectories passing through the test points (a) $(0.840\pi, 15.174)$ and

(b) $(1.504\pi, 4.719)$ when using experimental trajectories with initial conditions (6)

4. Conclusions

For the purpose of generating a phase portrait of an inverted pendulum system, a finite number of free fall experiments are conducted. The obtained experimental trajectories are modelled by using concatenation of overlapping cubic Bezier curves. Cubic Bezier curve formulation, a commonly used computer graphics tool, is used for both finding the representative points of the non-experimental new trajectories and interpolating their representative points. Besides generating phase portrait trajectories using ordered initial conditions on the θ axis, we also presented an algorithm to generate a trajectory for an arbitrary initial condition within the phase portrait. The trajectories generated by using both CBC formulation and a linear interpolation method are compared to the actual trajectories. It has been shown that trajectories generated by our approach successfully approximate the actual trajectories and have far better accuracy than the ones generated by the linear interpolation method.

Author Contributions

Gökhan DINDIŞ contributed to concept study, experiment design, experimental work, preparation of figures, Abdurrahman KARAMANCIOĞLU contributed to concept study, literature search, manuscript layout, preparation of figures. Scientific and publication ethics guidelines are followed in all phases of this study.

Conflict of Interest

There is no conflict of interest.

References

Aström, K.J. & Furuta, K. (2000). Swinging up a pendulum by energy control. *Automatica*, 36, 287-295. Doi : https://doi.org/10.1016/S1474-6670(17)57951-3

Chalupa, P. & Bobal, V. (2008). Modelling and predictive control of inverted pendulum. *22nd European Conference on Modelling and Simulation*, 3-6 June, Nicosia, Cyprus. 531-537. Doi : https://doi.org/10.1016/j.cagd.2012.03.001

Cimurs, R. & Suh, I.H. (2020). Time-optimized 3D Path Smoothing with Kinematic Constraints. *International Journal of Control Automation and Systems*, 18, 1277-1287. Doi : https://doi.org/10.1007/s12555-019-0420-x

Dındış, G. & Karamancıoğlu, A. (2016). A Self-Learning Robust Swing-up Algorithm. *Transactions of the Institute of Measurement and Control*, 38, 395-401. Doi : https://doi.org/10.1177/0142331215584420

Farouki, R.T. (2012). The Bernstein polynomial basis: A centennial retrospective. *Comput Aided Geom D* 29, 379-419. Doi : https://doi.org/10.1016/j.cagd.2012.03.001

Kim J. (2020). Trajectory Generation of a Two-Wheeled Mobile Robot in an Uncertain Environment. *IEEE Transactions on Industrial Electronics*, 67, 5586-5594. Doi : https://doi.org/10.1109/TIE.2019.2931506

Riskus A. (2006). Approximation of a cubic Bezier curve by a circular arcs and vice versa. *Inf Technol Control*, 35, 371-378. Doi : https://doi.org/10.5755/j01.itc.35.4.11812

Slotine, J.J.E. & Li W. (1991), Applied Nonlinear Control. *Prentice Hall*.

Stefani, R.T,. Shahian. B,. Savant. C.J, & Hostetter, G.H. (2002). Design of Feedback Control Systems. *4th ed. Oxford University Press*.

Wang, M.M., Luo, J.J., Fang, J, & Yuan, J.P. (2018). Optimal trajectory planning of free-floating space manipulator using differential evolution algorithm.

Advances in Space Research, 61, 1525-1536. Doi : https://doi.org/10.1016/j.asr.2018.01.011

Yamaguchi, F. (1988). Curves and Surfaces in Computer Aided Geometric Design. *5th ed. Springer Verlag*. Doi : https://doi.org/10.1007/978-3-642-48952-5

Zdesar, A., Skrjanc, I, & Klanjar, G. (2013). Visual trajectory-tracking model-based control for mobile robots. *Int J Adv Robot Syst* 10, 323. Doi : https://doi.org/10.5772/56757