

A DATA MINING APPROACH TO USING POINCARÉ MAPS IN MULTI-BODY TRAJECTORY DESIGN STRATEGIES

Natasha Bosanac*

Poincaré maps representing two-dimensional data sets are a powerful tool for rapid trajectory design in multi-body systems. However, projections of higher-dimensional data sets onto a map are challenging to analyze. To reduce their complexity, a density-based clustering algorithm is employed to cluster map crossings by the geometry of their associated trajectories. By grouping the data into clusters and identifying representative trajectories in each cluster, a reduced data set is constructed. This smaller data set reduces the complexity of analysis. This data mining approach to leveraging Poincaré maps in the trajectory design process is explored in the circular restricted three-body problem.

INTRODUCTION

Rapid and informed trajectory design strategies for applications within multi-body systems often benefit from the use of Poincaré maps. Specifically, Poincaré mapping enables visualization of a large set of trajectories, generated in a given dynamical system, via their intersections with a hyperplane.¹ This technique simplifies representation of the fundamental motions in chaotic systems such as the Earth-Moon and Sun-Earth systems. Patterns or features on Poincaré maps reveal dynamical structures that may serve as mission orbits or the regions of existence of desired trajectory segments.² This insight is employed when selecting individual arcs directly from these maps to construct an initial guess for an end-to-end trajectory in a higher-fidelity dynamical model.^{3,4,5}

The ease of using Poincaré maps in the trajectory design process depends on the dimension and complexity of the data set. When the maps uniquely represent planar trajectories at a single energy level in an autonomous system on a two-dimensional map, they are straightforward to analyze and interpret.^{4,6,7,8} Quantities such as chaos indicators may be added to a map and represented in the color of each crossing to provide some insight into the relative behavior of two nearby solutions.⁹ However, they do not typically reflect any differences in the geometry of solutions across the map. In addition, Poincaré maps constructed for planar trajectories at a variety of energy levels, in a nonautonomous system, or for spatial motion in autonomous systems, are higher-dimensional.^{10,11,12} The higher dimensionality of the data creates numerous challenges for analysis via a two-dimensional or three-dimensional

*Assistant Professor, Colorado Center for Astrodynamics Research, Smead Department of Aerospace Engineering Sciences, University of Colorado Boulder, 431 UCB, Boulder, CO 80309.

projection – even for an expert astrodynamist. These challenges impede the human-in-the-loop data analysis tasks required for trajectory design. The focus of this work is to use data mining to both simplify and enhance the use of Poincaré maps in rapid and informed trajectory design strategies within chaotic, multi-body systems.

This work builds upon previous contributions to applying data mining, specifically clustering, to the Circular Restricted Three-Body Problem (CR3BP) and other chaotic dynamical models. For instance, Nakhjiri and Villac used k -means clustering to separate stable motion from chaos on a Fast Lyapunov Indicator (FLI) map in the planar CR3BP, with a focus on the specific region near distant retrograde orbits.¹³ This approach is also leveraged by Nakhjiri and Villac to govern automated map generation in this specific region. Hadjighasem, Karrasch, Teramoto and Haller apply spectral clustering to Lagrangian vortex detection in several generalized flow problems.¹⁴ They demonstrate that such an approach can effectively cluster trajectories based on their geometry, with similarity between two trajectories defined using a weighted sum of the distances between two particles sampled at regular times along the solutions. Another example of the use of clustering in trajectory design within multi-body dynamical systems includes the work of Villac, Anderson and Pini.¹⁵ These authors leveraged k -means clustering to organizing periodic orbits, computed in the vicinity of an irregular body, into sets that are analogous to families. Building upon these previous works, this investigation applies a different method for clustering, specifically Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) to a more general Poincaré map in the CR3BP that encompasses a wider region in the phase space.¹⁶ This density-based clustering algorithm is well-suited to the data associated with a general Poincaré map due to its ability to accommodate: a number of clusters that is not known a priori; clusters of various shapes; clusters of various density; and an unknown or constant value of the maximum separation between data points within a single cluster in a higher-dimensional space.¹⁷

A strategy is developed to efficiently and effectively visualize a higher-dimensional Poincaré map representing trajectories within the autonomous CR3BP via density-based clustering. First, the trajectories associated with each map crossing on a Poincaré map are summarized via their apsides over several revolutions to balance the fidelity-level of the geometrical representation with the dimensionality of the description.¹⁸ HDBSCAN is then used to group map crossings that correspond to trajectories with a similar geometry.¹⁶ Each cluster is summarized by a single representative trajectory or map crossing to produce a representative reduced data set. Crossings on the map associated with an arc of interest may then be identified from the representative reduced data set via a global view and then used to isolate the corresponding individual cluster of trajectories for further examination. This approach enables a trajectory designer to rapidly assess the design space and simultaneously gain insight into the sensitivity of a given arc and, therefore, the robustness of a solution. This reduction in the complexity of using Poincaré maps has the potential to: enhance trajectory design for upcoming missions or extensions; support rapid concept development for large spacecraft and SmallSats; support the incorporation of low-thrust-enabled arcs or nonpropulsive adjustments into the trajectory design process; and aid with the enhancement of trajectory design tools for multi-body dynamical systems.

DYNAMICAL MODEL

To construct a Poincaré map that captures a sufficiently complex set of trajectories with a wide variety of geometries, the dynamics in the CR3BP are leveraged. This CR3BP reflects the motion of an assumed massless particle, P_3 , under the point mass gravitational interactions of two massive primaries, P_1 and P_2 .¹⁹ Each of these two primaries, with a mass M_i , where $i = 1, 2$, is assumed to follow circular orbits around their mutual barycenter. In addition, a nondimensionalization scheme is introduced to enable comparison between systems with similar relative masses and to reduce the potential for ill-conditioning. Typically, length quantities are normalized by the constant distance between the two primaries, while mass parameters are nondimensionalized by the total mass of the system. Then, μ is defined as the ratio of the mass of the smaller primary, P_2 , to the total mass of system. Finally, time quantities are normalized such that the mean motion of the primary system is equal to unity. Following nondimensionalization, a rotating frame, $\hat{x}\hat{y}\hat{z}$, is introduced to reduce the complexity of visualization and to enable construction of an autonomous dynamical system. This rotating frame is defined with the \hat{x} -axis directed from P_1 to P_2 , \hat{z} parallel to the orbital angular momentum vector of the primaries, and \hat{y} completing the right-handed triad. With these definitions, the state of P_3 is written in nondimensional coordinates relative to the system barycenter and in the rotating frame as $\bar{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]$. Then, the nondimensional equations of motion in the CR3BP, expressed in the rotating frame, are written as:

$$\ddot{x} - 2\dot{y} = \frac{\partial U}{\partial x}, \quad \ddot{y} + 2\dot{x} = \frac{\partial U}{\partial y}, \quad \ddot{z} = \frac{\partial U}{\partial z} \quad (1)$$

where $U = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{d} + \frac{\mu}{r}$ is a pseudo-potential function, while the distances between P_3 and the primaries are $d = \sqrt{(x + \mu)^2 + y^2 + z^2}$ and $r = \sqrt{(x - 1 + \mu)^2 + y^2 + z^2}$. This dynamical model is nonlinear, chaotic and autonomous. However, it admits a constant of motion, commonly labeled the Jacobi constant and equal to $C_J = 2U - \dot{x}^2 - \dot{y}^2 - \dot{z}^2$. At a single value of the Jacobi constant, a set of trajectories may exhibit a large variety of characteristics. These solutions tend to correspond to one of the following types of motion: equilibrium points, periodic orbits, quasi-periodic orbits, stable/unstable manifolds or chaos. While the first four types of solutions may be generated directly via dynamical systems techniques, it is often challenging to sufficiently capture the infinite variety of trajectories that exist in families – or even those that exist solely at a single value of the Jacobi constant. Furthermore, traditional techniques do not currently support comparison of the geometry of two solutions over finite time intervals. Accordingly, trajectory summarization techniques may be employed along with a useful visualization strategy to reflect the wide variety of geometries of solutions at a fixed value of the Jacobi constant.

POINCARÉ MAPPING

Poincaré mapping techniques offer a discrete-time representation of a continuous-time flow, reducing the complexity and dimensionality of visualizing motion within a chaotic dynamical system. To construct these maps, a hyperplane must first be defined transverse to the solutions of interest, requiring some a priori knowledge of the dynamics.²⁰ There

are numerous possibilities for defining a useful hyperplane, Σ , in the CR3BP including: a physically interpretable plane expressed in configuration space variables; a stroboscopic map that captures the flow at constant time intervals; or a known event that occurs along a trajectory, such as the locally minimum distance from a reference location (i.e., periapsis). Given an appropriately selected hyperplane, initial conditions are seeded within a desired region of the phase space. Each initial condition, \bar{x}_0 , is propagated forward or backward in time until its first intersection with the hyperplane in a desired direction. This intersection is labeled $P^1(\bar{x}_0)$, as illustrated in Figure 1. This process is repeated for a desired number of successive intersections with the hyperplane in a specified direction, with each crossing captured and represented on a lower-dimensional map. Patterns formed in the resulting one-sided map – or lack thereof – enable the detection of various types of fundamental motions, distinguishing order from chaos.

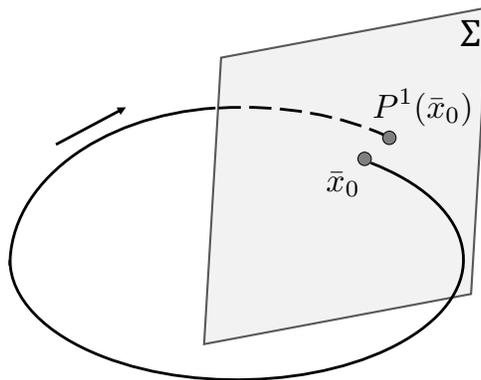


Figure 1. Poincaré mapping, capturing the intersections of a trajectory with a hyperplane, Σ .

To demonstrate the map construction process, consider planar motion near the Earth vicinity in the Sun-Earth CR3BP, described by a mass ratio of $\mu = 3.00348 \times 10^{-6}$. At a single value of the Jacobi constant, trajectories may potentially exhibit a large variety of geometries with behaviors including: captured motion near the Earth vicinity, impacting the Earth, or passing through either the L_1 or L_2 gateways to visit other regions of the Sun-Earth system.⁴ Such a diverse solution space in a chaotic flow regime may be visualized via Poincaré maps. First, a hyperplane is defined such that:

$$(x - 1 + \mu)\dot{x} + y\dot{y} + z\dot{z} = 0 \quad (2)$$

and

$$(x - 1 + \mu)\ddot{x} + y\ddot{y} + z\ddot{z} + \dot{x}^2 + \dot{y}^2 + \dot{z}^2 > 0 \quad (3)$$

capturing the perigees occurring along each trajectory.²¹ At a single value of the Jacobi constant, feasible initial conditions are seeded directly from this hyperplane, possessing the form $\bar{x}_C = [x, y, 0, \dot{x}, \dot{y}, 0]$ for N_x values of the x -coordinate between the L_1 and L_2 gateways and N_y values of the y -coordinate in the range $y = [y_{min}, y_{max}]$. At the perigee locatio along a trajectory, the velocity and position vectors, relative to the Earth, are perpendicular; thus, a unit vector aligned with the in-plane velocity vector is identified directly

from this orthogonality condition.²¹ A consistent direction of motion is selected: either (1) prograde, with P_3 instantaneously possessing an angular momentum vector relative to the Earth that is aligned with the orbital angular momentum of the primaries; or (2) retrograde, with P_3 traveling in a clockwise direction around the Earth at that instant of time. This direction of motion is used to select the correct direction of the unit vector. Then, the Jacobi constant relationship is rearranged to provide the following expression for the velocity magnitude, v , as a function of the position and $C_{J,d}$, the desired Jacobi constant: $v = \sqrt{2U - C_{J,d}}$. If this speed corresponds to a real number at a given position relative to the Earth, the value of v is used to scale the velocity unit vector and directly recover the velocity components, \dot{x} and \dot{y} . This process is implemented for each initial condition, seeded along the map, and only apses that satisfy the periapsis conditions in Eq. (2) and (3) are retained as initial conditions. Each of the remaining initial conditions is then propagated forward in time until either: completing N_{ret} positive intersections of the map; passing within a distance of 10^{-6} nondimensional units to the Earth; or upon departure from the vicinity of the Earth as defined by the x -coordinate passing through the values associated with L_1 or L_2 . The collection of crossings of the hyperplane that satisfy these conditions are then displayed on a two-dimensional plot reflecting the x - and y -coordinates in the rotating frame to visualize the behavior of solutions in the Earth vicinity. Since each perigee occurs along a planar trajectory at a single value of the Jacobi constant, the values of x and y uniquely specify the full state vector.

Following the outlined procedure for implementation, a periapsis map is constructed for planar motion in the Sun-Earth system at a Jacobi constant of $C_J = 3.00088$. At this value of the Jacobi constant, both the L_1 and L_2 gateways are open. Thus, the solution space is comprised of trajectories that exhibit a wide variety of geometries, offering a complex data set for testing the proposed clustering approach. For this map, the initial conditions are seeded on the hyperplane via 200 x -coordinates between L_1 and L_2 and 200 y -coordinates in the range $y = [-0.1, 0.1]$. Only a subset of these initial conditions will produce viable state vectors that satisfy the periapsis condition and correspond to a real-valued velocity magnitude. Nevertheless, the feasible initial conditions are propagated forward in time in the CR3BP, with up to 20 returns to the map recorded. The resulting map is depicted in Figure 2 with each map crossing displayed via a black point and the x - and y -coordinates of the state at each hyperplane crossing are represented on the horizontal and vertical axes of the figure, respectively. Each of the two equilibrium points is located by red diamonds while the Earth is identified, not to scale, at the center of the figure via a dark green circle. In addition, the gray shaded regions represent ‘forbidden regions’ where, for a Jacobi constant of $C_J = 3.00088$, the speed of the spacecraft does not possess a real value and, therefore, the spacecraft cannot travel within the phase space. Since this map captures periapses along planar solutions, each crossing of the map uniquely defines the entire state at perigee. Yet, this map still captures a complex solution space.

The patterns that form on the sample Poincaré map, as well as the stable and unstable manifolds associated with L_1 and L_2 Lyapunov orbits, provide only preliminary insights into the behavior of trajectories in the Earth vicinity at this value of the Jacobi constant. First, the periapsis map in Figure 2 admits regions where the data aggregate with various

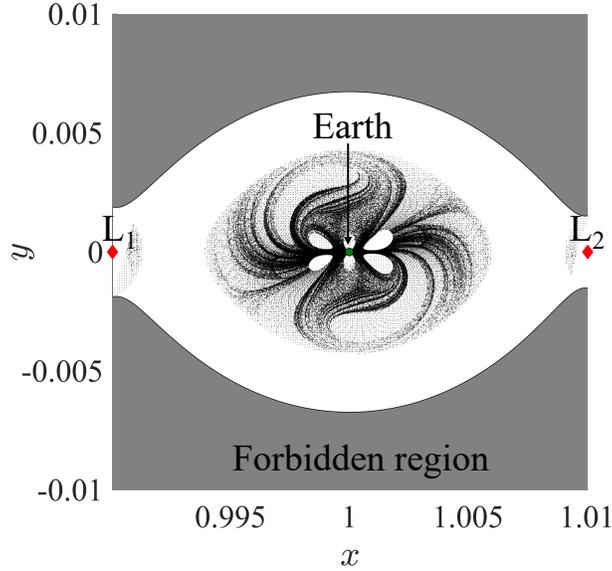


Figure 2. Periapsis map constructed in the Sun-Earth system at a Jacobi constant of $C_J = 3.00088$.

densities. Dark regions correspond to an aggregation of map crossings during subsequent returns to the map. Regions with a low density of data points, however, indicate that the map crossings do not return to the same region after subsequent revolutions. The boundaries of the low-density regions on the map in Figure 2 tend to correspond to the unstable manifolds associated with the L_1 and L_2 Lyapunov orbits at the same Jacobi constant of $C_J = 3.00088$.²¹ Accordingly, to possess periapses within this region at this energy level, trajectories must pass through the L_1 or L_2 gateways; thus, low-density regions correspond to solutions that have completed fewer revolutions around the Earth after passage through either gateway.

Overlaying the stable manifolds of the L_1 and L_2 Lyapunov orbits at the associated value of the Jacobi constant provides further information that is not currently possible to extract from the original Poincaré map in Figure 2.^{5,21} To explore this additional insight, Figure 3 displays in blue the periapsis locations of the stable manifolds associated with each of the L_1 and L_2 Lyapunov orbits at a Jacobi constant of $C_J = 3.00088$, overlaid on the original map. Regions on the map corresponding to the first three crossings of the stable manifolds are also labeled with the number of the crossing indicated. When a periapsis occurs within the region defined by the i -th crossing of one of these stable manifold structures, the trajectory will pass through the corresponding gateway within approximately i revolutions around the Earth. Note that the region bounded by the first crossing of the stable manifolds of the L_1 and L_2 Lyapunov orbits is white; that is, no trajectories with a periapsis in that region produce a subsequent crossing of the map within the Earth vicinity, as expected. However, such insight cannot be gained from the original Poincaré map to locate regions inside subsequent crossings of the stable manifold due to the absence of any associated patterns in the data. Furthermore, neither the stable and unstable manifolds nor the original

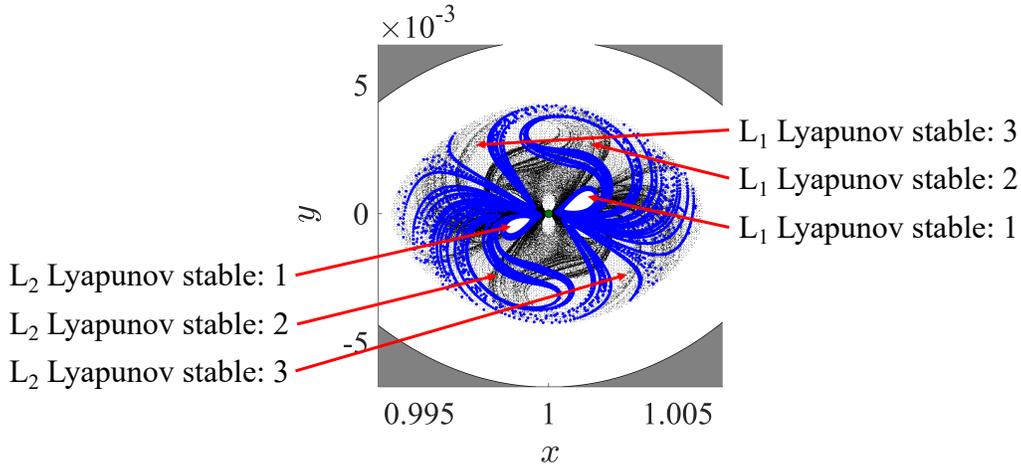


Figure 3. Periapsis map (black) constructed in the Sun-Earth system at a Jacobi constant of $C_J = 3.00088$ with stable manifolds (blue) associated with L_1 and L_2 Lyapunov orbits overlaid.

Poincaré map provide sufficient insight into the variation of the geometry of trajectories associated with crossings of the map. Such a limitation becomes increasingly significant when the map captures spatial motion, a higher-dimensional solution space or trajectories in a nonautonomous system – and serves as the motivation for this analysis.

DATA ANALYSIS VIA CLUSTERING

Clustering algorithms are a valuable tool for implementing an unsupervised grouping of data with similar properties. While there are a large variety of clustering algorithms available in the literature, the selected algorithm must be able to accommodate the properties of the data generated via Poincaré mapping: a number of clusters that is not known a priori; clusters of various shapes; clusters of various density; and an unknown or constant value of the maximum separation between data points within a single cluster in a higher-dimensional space. Given these properties of the data generated via Poincaré mapping, this preliminary investigation leverages an algorithm known as HDBSCAN, developed by Campello, Moulavi and Sander.¹⁶ This algorithm leverages a density-based approach to construct clusters each corresponding to data points that are densely located within the same neighborhood of a higher-dimensional space. These clusters are assigned hierarchically to capture the most significant clusters that possess a number of data points above a user-defined minimum threshold. This section offers a brief conceptual overview of the HDBSCAN algorithm outlined by Campello, Moulavi and Sander; for further details and proofs, see their original published article.¹⁶

Prior to implementing the clustering process, HDBSCAN leverages a quantity labeled the mutual reachability distance to compare data points within a set and locate regions of higher density. To explore the definition of this quantity, first consider a data set, $[S]$, consisting of N vectors such that $[S] = [\bar{s}_1, \dots, \bar{s}_N]$. Each component, \bar{s}_i , of this data set

is an M -dimensional vector that reflects the properties of the associated data. The core distance of the i -th data point is then defined as the distance between that data point and its $N_{min,core}$ -th nearest neighbor. The quantity $N_{min,core}$ is a user-selected parameter that defines the number of points required for a data point to be considered a *core* point, i.e., there is a sufficient number of points in its vicinity. Furthermore, the distance may be calculated via one of many possible distance metrics, e.g., a Euclidean distance, infinity-norm, l_1 -norm, etc. Using this quantity as a foundation, a mutual reachability distance between the i -th data point and the j -th data point is defined as the maximum of three distances: (1) the core distance of the i -th data point; (2) the core distance of the j -th data point; and (3) the distance between the i -th and j -th data points. The mutual reachability distance for each of the N data points to the remaining data in the set is then used to construct a distance matrix, $[D]$. Using this matrix, a mutual reachability graph may be constructed with each of the N data points serving as vertices and the edges weighted by the computed mutual reachability distance. This graph reflects the similarity between neighboring data points and supports cluster identification.

Using the mutual reachability distance as a foundation, HDBSCAN produces a tree of all possible clusters that is hierarchically simplified to remove noise and retain only the significant clusters with a number of data points above a threshold value. The outlined procedure here summarizes the primary steps in *Algorithm 1* in Campello, Moulavi and Sander.¹⁶ First, a minimum spanning tree is constructed for the mutual reachability graph, retaining only a subset of the edges that produces the minimum total weight; use of a minimum spanning tree reduces the complexity of identifying clusters. This minimum spanning tree is then condensed to a summarized tree via the definition of a minimum cluster size, $N_{min,cluster}$, a user-selected parameter that significantly influences the number and membership of recovered clusters. Using this quantity, the tree is traversed to locate splits that correspond to new groupings of data being formed with a number of data points that is above the threshold $N_{min,cluster}$. Using this condensed tree, HDBSCAN identifies stable clusters as groups of data that persist over a large range of minimum threshold values in the mutual reachability distance. Points that are not assigned to a cluster are noise points.

Following the general procedure for HDBSCAN, several input and output parameters emerge. Input parameters that are provided by the user are the data set, $[S]$, the distance metric to be used as well as the quantities $N_{min,core}$ and $N_{min,cluster}$. The data set possesses a dimension of $N \times M$, and the M -dimensional data vectors must be constructed to sufficiently capture the properties of the data, while balancing the computational complexity and data storage requirements. In addition, Campello, Moulavi and Sander suggest selecting $N_{min,core} = N_{min,cluster}$ to reduce the number of parameters governing the performance of HDBSCAN.¹⁶ However, these quantities are often selected iteratively based on user-intuition on a test data set, the number of identified clusters and the number of noise points. The outputs of this clustering procedure are the number of clusters, $N_{clusters}$, and the labels, l_i , identifying the cluster that the i -th data point is assigned to: $0 \leq l_i \leq N_{cluster}$ if the data point belongs to a cluster or $l_i = -1$ if it is a noise point. As discussed by Campello, Moulavi and Sander, the computational complexity of this algorithm is $O(MN^2)$ in time and $O(MN)$ in memory storage when the algorithm is provided an $M \times N$ dimensional

data set.¹⁷ Depending on the user-defined properties and methods used at each step of the clustering algorithm, the computational complexity may be reduced even further. In this investigation, a fast and efficient implementation of HDBSCAN is accessed using the freely available *hdbscan* Clustering Library in Python.²²

APPLICATION OF CLUSTERING TO A POINCARÉ MAP

In this section, the HDBSCAN algorithm is leveraged to organizer the crossings on a Poincaré map, constructed in the planar CR3BP, into clusters based on the geometry of the associated trajectories. First, the trajectory summarization process employed in this investigation is outlined. Then, the process for generating map data is discussed, followed by the selection of input variables for clustering. While there may be several options for generating and summarizing the data, as well as selection of the clustering algorithm properties, this preliminary analysis is focused on a proof of concept in the CR3BP. Subsequent papers will detail comparisons of the clustering process for various options and with application to various dynamical models. Nevertheless, the approach presented here is applied to a Poincaré map constructed in the Sun-Earth planar CR3BP at the same Jacobi constant as in Figure 2, i.e., $C_J = 3.00088$.

Trajectory Summarization

The objective of trajectory summarization is to pre-process the data set by constructing a compressed description, \bar{T}_i , of the trajectory associated with the i -th map crossing. Each crossing on a discrete-time map is associated with a nonlinear trajectory that is described by a vector of reasonable dimensionality, while still capturing the solution geometry: this problem is commonly encountered in moving object database applications. In the absence of a closed form solution to a chaotic dynamical system, a trajectory could be represented with a high fidelity via discretization into a large time sequence of state vectors. However, such an approach would require prohibitively large data storage resources. Thus, reducing the dimensionality of each data point reduces the storage requirements for the entire data set, while pre-processing reduces the computation time during clustering.

In pursuit of a low-dimensional, yet representative, description of each data point, a curve-based representation is constructed via a subset of apses sampled along the trajectory for a finite number of subsequent crossings of the map. As a preliminary approximation of the entire nonlinear trajectory, reduction to a sequence of apses or turning points offers a low-dimensional summarization that captures the general shape of the solution. Furthermore, the integration time along the associated trajectory is limited to either three subsequent returns to the hyperplane (i.e., perigee), passage through the L_1 or L_2 gateways, or impact with the Earth – whichever event occurs first. Given the geometry of the solutions in the sample dynamical model, subsampling a continuous trajectory for a finite time interval at its apses will produce only a small number of states in a small computational time; for three returns to the periapsis map, up to 7 apses occur. Of course, increasing the maximum integration time will reveal further differences between trajectories and increase the number of different geometries exhibited by the solutions associated with the

map crossings. However, for this preliminary investigation, three returns to the map sufficiently differentiates the geometry of the solutions. The compressed description vector, \bar{T}_i , for the i -th trajectory is then formed using the map crossing and associated apsides. The j -th apse is described by a vector $\bar{R}_{i,j}$ that captures the time, state and direction information and is written as:

$$\bar{R}_{i,j} = \left[\tau_{i,j}, \quad x_{i,j}, \quad y_{i,j}, \quad \dot{x}_{i,j}, \quad \dot{y}_{i,j}, \quad \text{sign}(\hat{h}_{i,j} \cdot \hat{z})/7 \right] \quad (4)$$

where $\tau_{i,j}$ is the time at each apse after the initial condition for the i -th trajectory, normalized by the total integration time along the solution; $x_{i,j}, y_{i,j}, \dot{x}_{i,j}, \dot{y}_{i,j}$ are the state vector components at periapsis or apoapsis; and $\text{sign}(\hat{h}_{i,j} \cdot \hat{z})/7$ indicates the direction of motion at that apse, captured by the sign of the component of the orbital angular momentum unit vector in the \hat{z} -direction and normalized by the maximum number of apsides, i.e., 7. While the final component of $\bar{R}_{i,j}$ is not independent to other parameters, it does help to further separate the data during clustering. If the solution passes through either of the L_1 or L_2 gateways or impacts the Earth after the k -th apse, $\bar{R}_{i,j}$ is assigned an arbitrary value, e.g., $\bar{R}_{i,j} = [10, 0, 0, 0, 0, 0]$, for $j > k$. Then, the complete compressed description vector for the i -th trajectory is formed as:

$$\bar{T}_i = [\bar{R}_{i,1}, \quad \dots, \quad \bar{R}_{i,\tau}] \quad (5)$$

The data descriptions, \bar{T}_i , recovered by this trajectory compression strategy for all map crossings are combined to form the rows of the data set, $[\mathbf{D}]$. This information is input to the clustering algorithm and ensures a reasonable dimensionality for the data point representing each map crossing.

Map Data Generation

The data set, $[\mathbf{D}]$, is constructed to consist of the map crossings associated with the prograde periapses that produce at least one crossing of the hyperplane along with a summary of the associated trajectory integrated for three subsequent returns via its apsides. Since data is only generated for crossings that produce at least one crossing of the hyperplane, up to 400 values of the x -coordinate between L_1 and L_2 as well as up to 400 values of the y -coordinate between -0.01 and 0.01 are used. From all possible combinations of x and y that produce feasible perigees at a Jacobi constant of $C_J = 3.00088$, 31,500 initial perigees are constructed. Then, the map and associated trajectories are generated in MATLAB to produce a data set of dimension $31,500 \times 42$ that is written to a text file. This text file is then read into a Python script that implements the clustering process via the hdbscan library.²² Following iterative adjustment of the parameters governing the clustering algorithm, $N_{min,core}$ and $N_{min,cluster}$ are set to 5 and 200, respectively. These values are observed to provide a reasonable number of clusters while also limiting the number of noise points. In addition, the HDBSCAN algorithm is customized in this preliminary investigation to use the Euclidean distance metric to compare the vectors describing each data point in $[\mathbf{D}]$. Together, these parameter selections are leveraged by HDBSCAN to discover groupings in the data and identify similarity in the solutions associated with nearby crossings on the map.

Results and Analysis

The outlined approach is employed to cluster prograde periapses in the Earth vicinity for the Sun-Earth planar CR3BP at a Jacobi constant of $C_J = 3.00088$, based on the geometry of their associated trajectories. As a result, the algorithm, with the selected parameterization and trajectory summarization strategy, uncovers 34 clusters with only 1857 of the 31,500 data points labeled as noise in 6.33 seconds on a computer with a 1.2 GHz Intel Core m5 processor. The result of this clustering procedure is displayed in Figure 4 with map crossings uniquely colored by their cluster: clusters with an even-numbered label are colored by shades of red, while those with an odd-numbered label are colored in shades of blue. Noise points are colored white. Note that this view is zoomed-in to the main region of crossings for clarity and does not capture the crossings near the L_1 and L_2 gateways. Analysis of this figure, and comparison to Figure 3 reveals that, at a minimum, the clustering process can separate the regions inside subsequent crossings of the stable manifolds of the L_1 and L_2 Lyapunov orbits – without knowledge of these manifold structures. Furthermore, additional differentiation between trajectories and their geometry appears: some separation reflects the intersection between the regions defined by the map crossings associated with the stable and unstable manifolds, and other subdivisions occur within these regions.

For further insight into the ability of this clustering approach to differentiate trajectories of various geometries on the periapsis map, an analysis of the solutions in each cluster is useful. To aid visualization and interpretation, Figure 5 displays a translucent version of the map, colored by clusters, with each cluster labeled using integers from 0 to 33. One representative solution from each cluster is then plotted in Figure 6 with the number indicating the associated cluster label. All solutions in this grid generally revolve around

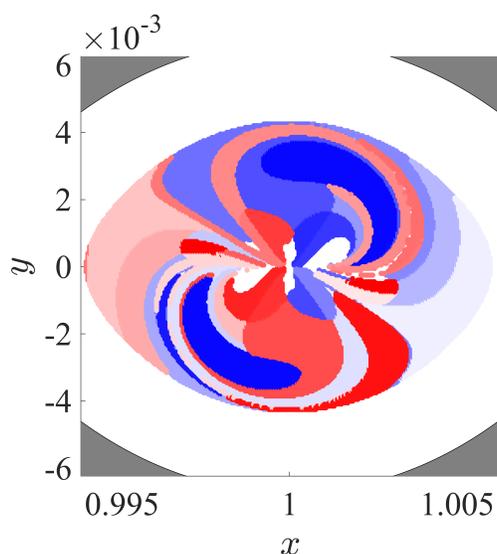


Figure 4. Zoomed-in view of the periapsis map constructed in the Sun-Earth system at a Jacobi constant of $C_J = 3.00088$ with 31,500 data points organized into 34 clusters with 1857 noise points indicated via white. Even-numbered clusters are colored using shades of red while odd-numbered clusters are indicated by shades of blue.

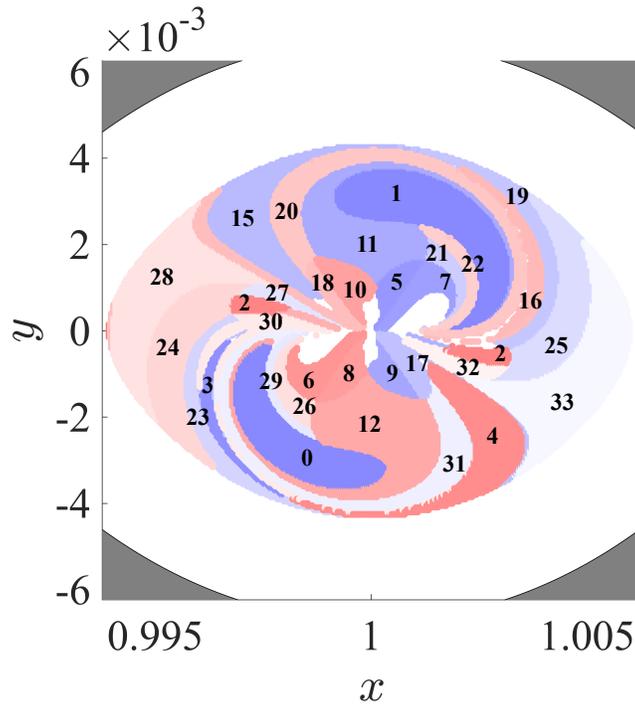


Figure 5. Zoomed-in view of the periaapsis map constructed in the Sun-Earth system at a Jacobi constant of $C_J = 3.00088$ with 34 clusters numbered.

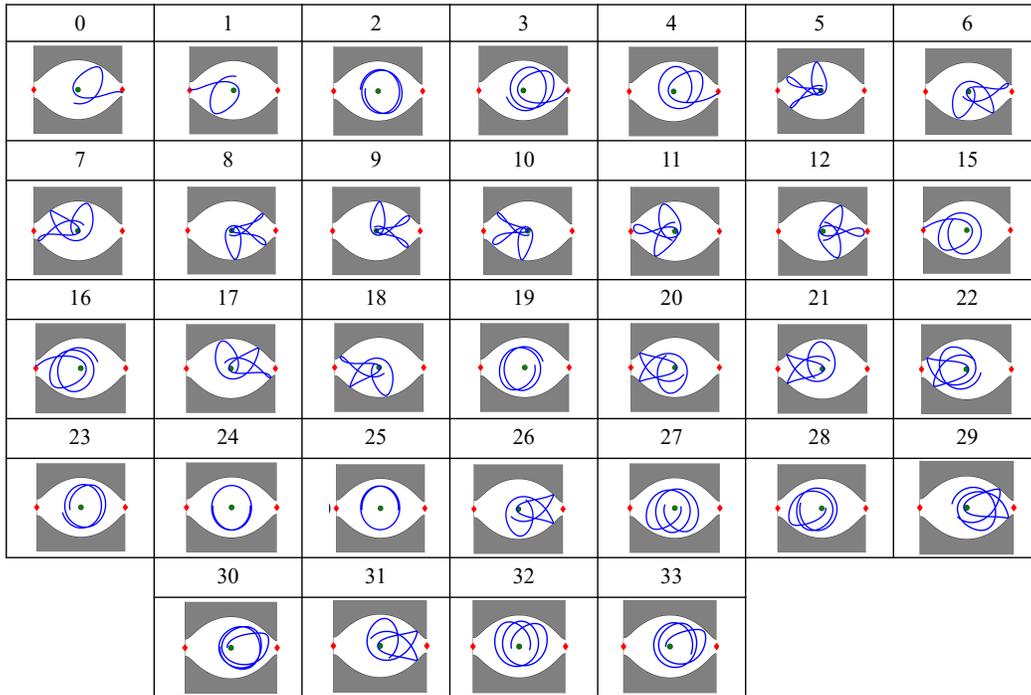


Figure 6. Key for cluster numbers, with sample trajectories from each cluster plotted in the rotating frame for up to 3 revolutions around the Earth.

the Earth in a prograde manner since the initial conditions are defined as prograde; along some solutions, the direction of motion does change temporarily. Nevertheless, analysis of Figures 5 and 6 reveals that, in general, solutions of various geometry – as defined by their apsides for three crossings of the hyperplane, assembled chronologically into a sequence – are separated to distinctly different clusters. For solutions with a similar geometry, such as those in clusters 24 and 25, the clusters are considered separate and distinct based on the location of the first and last periapses. Of course, based on the use of a Euclidean distance to differentiate between two trajectories via chronological time sequences of apsides, such a result is expected. Future work will explore the use of a more general distance metric capturing differences between the entire solutions, regardless of start time. In any case, the results in Figures 5 and 6 suggest that density-based clustering approaches can organize map crossings into clusters based on the geometry of the associated trajectories. Some of these clusters can be used to locate the regions within the stable manifold structures associated with L_1 and L_2 Lyapunov orbits at the same Jacobi constant. These results suggest that a density-based clustering procedure has the potential to serve as a valuable tool in assisting the trajectory designer in interpreting a Poincaré map, both in the planar CR3BP and in more complex or higher-dimensional dynamical systems.

CONCLUSION

A density-based clustering algorithm, HDBSCAN, is employed to implement an automated clustering of map crossings on a general Poincaré map based on the geometry of the associated solution. As opposed to other clustering algorithms, this particular method is selected due its ability to accommodate the properties of the data generated via Poincaré mapping: a number of clusters that is not known a priori; clusters of various shapes; clusters of various density; and an unknown or constant value of the maximum separation between data points within a single cluster in a higher-dimensional space. Furthermore, data is generated by associating each map crossing with its trajectory integrated for a few revolutions around the Earth and summarizing the solution via the location of each periapsis and apoapsis, as well as the associated epoch and direction of motion. Solutions are considered similar if the vectors describing the finite set of apsides are close, as calculated by the Euclidean distance metric. HDBSCAN, implemented in Python via the `hdbscan` library, is used to cluster the map crossings based on the geometry of the solutions. The result of this preliminary analysis is a map with individual clusters indicated by distinct colors, and each cluster sufficiently capturing only solutions of similar geometry as defined by a chronological sequence of apsides. The results of this analysis motivate future work to update the distance metric input to HDBSCAN and to analyze the impact of various trajectory summarization approaches. Furthermore, this investigation provides a foundation for future applications to maps of higher dimensional data and for dynamical models of increased complexity.

ACKNOWLEDGMENT

This work is being completed at the University of Colorado Boulder under NASA Grant 80NSSC18K1536.

REFERENCES

- [1] Paskowitz, M.E.; Scheeres, D.J., “Robust Capture and Transfer Trajectories for Planetary Satellite Orbiters,” *Journal of Guidance, Control and Dynamics*, Vol. 29, No. 2, pp. 342-353, 2006.
- [2] Craig Davis, D.; Phillips, S.M.; McCarthy, B.P., “Trajectory Design for Saturnian Ocean Worlds Orbiters Using Multidimensional Poincaré Maps,” *Acta Astronautica*, Vol. 143, pp. 16-28, 2018.
- [3] Haapala, A.F., “Trajectory Design in the Spatial Circular Restricted Three-Body Problem Exploiting Higher-Dimensional Poincaré Maps,” PhD Dissertation, School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, 2014.
- [4] Bosanac, N, “Leveraging Natural Dynamical Structures to Explore Multi-Body Systems,” PhD Dissertation, School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, 2016.
- [5] Howell, K.C.; Craig Davis, D.; Haapala, A.F., “Application of Periapse Maps for the Design of Trajectories Near the Smaller Primary in Multi-Body Regimes,” *Mathematical Problems in Engineering*, Vol. 2012, 2011.
- [6] Contopoulos, G, *Order and Chaos in Dynamical Astronomy*, Germany: Springer-Verlag, 2002.
- [7] Schlei, W; Howell, K.C.; Tricoche, X.M.; Garth, C, “Enhanced Visualization and Autonomous Extraction of Poincaré Map Topology,” *The Journal of the Astronautical Sciences*, Vol. 61, pp. 170-197, 2014.
- [8] Craig Davis, D.E., “Multi-Body Trajectory Design Strategies Based on Periapsis Poincaré Maps,” PhD Dissertation, School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, 2011.
- [9] Short, C.; Howell, K.C., “Lagrangian Coherent Structures in Various Map Representations for Application to Multi-body Gravitational Regimes,” *Acta Astronautica*, Vol. 94, No. 2, pp. 592-607, 2014.
- [10] Haapala, A.F.; Howell, K.C., “Trajectory Design Strategies Applied to Temporary Comet Capture Including Poincaré Maps and Invariant Manifolds,” *Celestial Mechanics and Dynamical Astronomy*, Vol. 116, pp. 299-323, 2013.
- [11] Gómez, G; Koon, W.S.; Lo, M.W.; Marsden, J.E.; Masdemont, J; Ross, S.D., “Connecting Orbits and Invariant Manifolds in the Spatial Restricted Three-Body Problem,” *Nonlinearity*, Vol. 17, pp. 1571-1606, 2004.
- [12] Patsis, P.A., Zachilas, L, “A Method for Visualizing the 4-Dimensional Space of Section in 3-D Hamiltonian Systems.” In: Seimenis, J. (eds) *Hamiltonian Mechanics, NATO ASI Series (Series B: Physics)* Vol. 331, Boston: Springer, 1994.
- [13] Nakhjiri, N; Villac, B.F., “Automated Stable Region Generation, Detection, and Representation for Applications to Mission Design,” *Celestial Mechanics and Dynamical Astronomy*, Vol. 123, No. 1, pp. 63-83, 2015.
- [14] Hadjighasem, A; Karrasch, D; Teramoto, H; Haller, G, “Spectral-Clustering Approach to Lagrangian Vortex Detection,” *Physical Review E*, Vol. 93, No. 6, 2016.
- [15] Villac, B.F.; Anderson, R.L.; Pini, A.J., “Organizing Ballistic Orbit Classes Around Small Bodies,” *The Journal of the Astronautical Sciences*, Vol. 63, No. 3, pp. 175-205, 2016.
- [16] Campello, R.J.G.B.; Moulavi, D; Sander, J, “Density-Based Clustering Based on Hierarchical Density Estimates,” In: Pei J., Tseng V.S., Cao L., Motoda H., Xu G. (eds) *Advances in Knowledge Discovery and Data Mining. PAKDD 2013. Lecture Notes in Computer Science*, Vol. 7819, Springer, Berlin, Heidelberg.
- [17] Campello, R.J.G.B.; Moulavi, D; Zimek, A; Sander, J, “Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection,” *ACM Transactions on Knowledge Discovery from Data* Vol. 10, No. 1, 2015.
- [18] Zheng, Y, “Trajectory Data Mining: An Overview,” *ACM Transactions on Intelligent Systems and Technology*, Vol. 6, No. 3, Article 29, 2015.
- [19] Szebehely, V, *Theory of Orbits: The Restricted Problem of Three Bodies*, London, UK: Academic Press, 1967.
- [20] Perko, L, *Differential Equations and Dynamical Systems, Second Edition* New York: Springer-Verlag, 1996.
- [21] Haapala, A.F., “Trajectory Design Using Periapse Maps and Invariant Manifolds,” MS Thesis, School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, 2010.
- [22] McInnes, L; Healy, J; Astels, S; “hdbscan: Hierarchical Density Based Clustering,” *Journal of Open Source Software, The Open Journal*, Vol. 2, No. 11, 2017.