

SPACE OBJECT CLASSIFICATION USING MODEL DRIVEN AND DATA DRIVEN METHODS

Richard Linares*

University of Minnesota, Twin City Campus, Minneapolis, NM, 55403

John L. Crassidis†

University at Buffalo, State University of New York, Amherst, NY, 14260-4400

In recent years there has been an increase in the number of inactive and debris Space Objects (SOs). This work examines both data driven and model driven SO classification. The model driven approach investigated for this work is based on the Multiple Model Adaptive Estimation approach to extract SO characteristics from observations while estimating the probability the observations belonging to a given class of objects. The data driven methods are based on Principle Component Analysis and Convolutional Neural Network Classification approaches. The performance of these strategies for SO classification is demonstrated via simulated scenarios.

INTRODUCTION

SSA involves the collection, processing, fusion and assessment of data and information from many different sources and the dissemination of information to decision makers and various users. Due to the large number of space objects (SOs) and the limited number of sensors available to track them, it is difficult to maintain persistent surveillance, and, therefore, there is inherent uncertainty and latency in the knowledge of the SO population. Although the amount of light collected from these objects is small, information can still be extracted from photometric data which can be used to determine shapes and other properties. Light curve data are the time-varying sensor wavelength-dependent apparent magnitude of energy (e.g. photons) scattered (reflected) off of an object along the line-of-sight to an observer. Attitude estimation and extraction of other characteristic using light curve data has been demonstrated in Refs. 1–5.

This work studies the problem of classifying light curve measurements based on the Space Object (SO) class that it likely originates from using both model driven and data driven methods. The model driven approach uses the probability from a Multiple Model Adaptive Estimation (MMAE) process to determine the probability that a given SO falls in a given class. The MMAE approach has been applied to the shape estimation problem⁴ with success. This work uses the MMAE approach to also provide SO classification. Along with this approach data driven classification approaches are also considered. MMAE is a recursive algorithm that uses a bank of estimators, each dependent on a particular hypothesis, to determine an estimate based upon an unknown physical process under

* Assistant Professor, Department of Aerospace Engineering and Mechanics. Email: rlinares@umn.edu, Member AIAA.

† CUBRC Professor in Space Situational Awareness, Department of Mechanical & Aerospace Engineering. Email: johnc@buffalo.edu, Fellow AIAA, Fellow AAS.

consideration. In particular, the hypotheses can correspond to different mathematical models of the same physical process or of the same model but dependent upon different constants or model parameters. The classification approach used in the work in is based on tree-like structure. The first determination is made from the size of the shape models in the bank. For each model in the bank an aspect ratio is calculated for each size by calculating the length ratio of that side with respect to the largest side and if a given model has an aspect ratio less than 0.1 it is considered to be a fragment.

The second classification determination is made from the control states. For each model in the bank that is not a fragment or a rocket body additional models are created that are copies of that shape model but have different control profiles. The control profiles include uncontrolled, Sun pointing, spin stabilized, and nadir pointing. Therefore, for models that are not fragments of rocket bodies three additional modes are created with control. The control states are not limited to ones used in this work, and large lists of control states may be possible, but for this study these three are sufficient.

The classification is determined using the shape model, for example determining whether an object is intact or passive and whether it is a rocket body or a payload. The final classification further separates the SO into the type of control state and determining whether an object is uncontrolled or Sun pointing, etc. This method uses the MMAE probability to classify the four feature classes and results for this method are shown.

The MMAE classification approach uses models of the dynamics and physics to estimate model relevant parameters. In general, there are many unknown parameters that should be included in the estimation process but the dimensionality in this complex problem does not lend itself to computationally efficient solutions. Therefore, for this work data driven classification approaches are investigated that are not based on models but rather use the observations or data to detect relationships and use these relationships for classification. The two data driven classification approaches used for this work are the Principle Component Analysis⁶ (PCA) approach and the Convolutional Neural Network Classification⁷ (CNN) approach.

The PCA approach is remarkably simple and can be apply efficiently to large data sets to detect linear relationships in the data. This approach is very popular and is usually one of the first approaches scientists use for classification problems. The PCA approach provides a method to reduce the dimensionality of data set. This reduced representation is used as features in the classification problem. The PCA approach does have the shortcomings that it only learns linear relationships between the data, and that this approach does not naturally work well with data sets that have variable dimensionality (at least without special preprocessing). For example, each light curve observation sequence may have different sampling frequency, total observation time, and or measurement noise and the PCA approach may not handle this variability well.

The second data drive approach considered uses the CNN approach, which has the benefit of allowing for nonlinear relationships to be learned from the data. The CNN approach works by using the concept of Neural Networks (NNs) to learn convolution features for the data. NN are function approximations that are inspired by biological neural networks (in particular the human brain), and are used to estimate input output relationships. NN can be used for classification by estimating a relationship between data and class parameters. In other words, the NN classifier will map from a data vector with dimension $m \times 1$, (where m is the number of measurements) to a classification vector with dimension $n_c \times 1$ (where n_c is the number of classes) that indicated which class a measurement vector belongs to. The NN is then given training data to learn this mapping, and once

the mapping is learned the NN can be applied to real-world data sets.

The training data consist of measurements \mathbf{y}_i , and labels, \mathbf{c}_i for each measurement, where i denote the i^{th} training sample. The vectors \mathbf{y}_i are labeled by using an indicator, \mathbf{c}_i , for each class, where 1 indicates it belongs to a given class, and all other classes have an indicator of 0. The CNN basically convolves the data with convolutional kernels to generate features that can be used to map to classification states. This convolution step gives CNN the ability to be robust to feature location and input data size. If the features are moved within the data vector the CNN can still classify correctly and moreover determine the location of the feature.

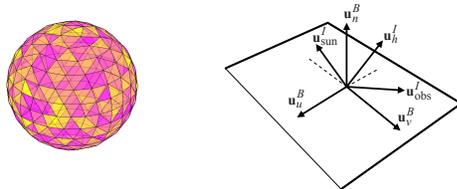
The organization of this paper is as follows. First, the Ashikmin-Shirley light curve model is shown, and the shape models used in this work are discussed. Next, the Unscented Kalman Filter approach used in this work is outlined. Following this, the MMAE approach used in this work is outlined. Then the MMAE classification approach is discussed. Following this both data driven approaches are discussed. Finally, results are shown for simulated examples, discussions, and conclusions are provided. This paper discusses the theory involved behind the proposed algorithms and results from a variety of simulation trials are shown.

LIGHT CURVE MODELING

There are a number of models used for simulating light curve measurements in the literature. Reference 8 provides a good summary of the most popular ones adopted for SO applications. These models differ in the physics that they represents and their level of complexity, but for SO applications the ability to model specular reflection and complex shapes while conserving energy is desirable. The Ashikhmin-Shirley⁹ model has all the desirable properties while producing realistic SO light curves. This model is based on the bidirectional reflectance distribution function (BRDF) which models light distribution scattered from the surface due to the incident light. The BRDF at any point on the surface is a function of two directions, the direction from which the light source originates and the direction from which the scattered light leaves the observed surface. The model in Ref. 9 decomposes the BRDF into a specular component and a diffuse component. The two terms sum to give the total BRDF:

$$f_r = (dR_d + sR_s) \quad (1)$$

which depends on the diffuse bidirectional reflectance (R_d), the specular bidirectional reflectance (R_s), and the fraction of each to the total (d and s respectively where $d + s = 1$), where i denotes the i^{th} facet of the Space Objects (SOs). Each facet contributes independently to the brightness and total brightness is the sum over each facet's contribution. The diffuse component represents light that is scattered equally in all directions (Lambertian) and the specular component represents light that is concentrated about some direction (mirror-like). Reference 9 develops a model for continuous arbitrary surfaces but simplifies for flat surfaces. This simplified model is employed in this work because shape models are considered to consist of a finite number of flat facets. Therefore the total observed brightness of an object becomes the sum of the contribution from each facet. In each model, however, $c = \mathbf{u}_{\text{obs}}^I T \mathbf{u}_h^I$ (seen in Figure 1(b)), ρ is the diffuse reflectance ($0 \leq \rho \leq 1$), and F_0 is the specular reflectance of the surface at normal incidence ($0 \leq F_0 \leq 1$). To be used as a prediction tool for brightness and radiation pressure calculations, an important aspect of the BRDF is energy conservation. For energy to be conserved, the integral of the BRDF times $\cos(\theta_r)$ over all



(a) Space Object Shape Model (b) Reflection Geometry Model

Figure 1. Reflection Geometry and Space Object Shape Model

solid angles in the hemisphere with $\theta_r \leq 90$ needs to be less than unity, with

$$\int_0^{2\pi} \int_0^{\pi/2} f_r \cos(\theta_r) \sin(\theta_r) d\theta_r d\phi = R_d + R_s \quad (2)$$

For the BRDF given in Eq. (1), this corresponds to constant values of $R_d = d\rho$ and $R_s = sF_0$. The remaining energy not reflected by the surface is either transmitted or absorbed. In this paper it is assumed the transmitted energy is zero. The diffuse bidirectional reflectance is then calculated as follows:

$$R_d = \frac{28\rho}{23\pi} (1 - sF_0) \left(1 - \left(1 - \frac{\mathbf{u}_n^{I^T}(i)\mathbf{u}_{\text{sun}}^I}{2} \right)^5 \right) \left(1 - \left(1 - \frac{\mathbf{u}_n^{I^T}(i)\mathbf{u}_{\text{obs}}^I}{2} \right)^5 \right) \quad (3)$$

where

$$F = F_0 + \left(\frac{1}{s} - F_0 \right) (1 - c)^5 \quad (4)$$

In addition to d , ρ , and F_0 , the Ashikhmin-Shirley BRDF has two exponential factors (n_u , n_v) that define the reflectance properties of each surface. The Ashikhmin-Shirley diffuse and specular reflectivities are not constant, however, but rather complicated functions of illumination angle, exponential factor, and the diffuse and specular reflectances. In all cases, however, $R_d + R_s \leq 1$, so energy is conserved. The parameters of the Phong model that dictate the directional (locally horizontal or vertical) distribution of the specular terms are n_u and n_v . The specular bidirectional reflectance for the Ashikhmin-Shirley model is given by

$$R_s = \frac{\sqrt{(n_u + 1)(n_v + 1)}}{8\pi} \frac{F}{c \max[\mathbf{u}_n^{I^T}(i)\mathbf{u}_{\text{sun}}^I, \mathbf{u}_n^I(i)\mathbf{u}_{\text{obs}}^I]} (\cos(\alpha))^{n_u \cos^2(\beta) + n_v \sin^2(\beta)} \quad (5)$$

Flux Calculation

The apparent magnitude of an SO is the result of sunlight reflecting off of its surfaces along the line-of-sight to an observer. First, the fraction of visible sunlight that strikes an object (and is not absorbed) is computed by

$$F_{\text{sun}}(i) = C_{\text{sun,vis}} (\mathbf{u}_n^I(i) \cdot \mathbf{u}_{\text{sun}}^I) \quad (6)$$

where $C_{\text{sun,vis}} = 1062 \text{ W/m}^2$ is the power per square meter impinging on a given object due to visible light striking the surface. If either the angle between the surface normal and the observer's

direction or the angle between the surface normal and Sun direction is greater than $\pi/2$ then there is no light reflected toward the observer. If this is the case then the fraction of visible light is set to $F_{\text{sun}}(i) = 0$. Next, the fraction of sunlight that strikes an object that is reflected must be computed:

$$F_{\text{obs}}(i) = \frac{F_{\text{sun}}(i)\rho_{\text{total}}(i)\mathcal{A}(i)(\mathbf{u}_n^I(i) \cdot \mathbf{u}_{\text{obs}}^I)}{\|\mathbf{d}^I\|^2} \quad (7)$$

The reflected light of each facet is now used to compute the total photon flux, which is measured by an observer:

$$\tilde{F} = \left[\sum_{i=1}^N F_{\text{obs}}(i) \right] + v_{\text{CDD}} \quad (8)$$

where v_{CDD} is the measurement noise associated with flux measured by a Charge Coupled Device (CCD) sensor. The total photon flux is then used to compute the apparent brightness magnitude

$$m_{\text{app}} = -26.7 - 2.5 \log_{10} \left| \frac{\tilde{F}}{C_{\text{sun,vis}}} \right| \quad (9)$$

where -26.7 is the apparent magnitude of the Sun.

DYNAMICS OF SPACE OBJECTS

A number of parameterizations exist to specify attitude, including Euler angles, quaternions, and Rodrigues parameters.¹⁰ This paper uses the quaternion, which is based on the Euler angle/axis parametrization. The quaternion is defined as $\mathbf{q} \equiv [\boldsymbol{\rho}^T \ q_4]^T$ with $\boldsymbol{\rho} = \hat{\mathbf{e}} \sin(\nu/2)$, and $q_4 = \cos(\nu/2)$, where $\hat{\mathbf{e}}$ and ν are the Euler axis of rotation and rotation angle, respectively. Clearly, the quaternion must satisfy a unit norm constraint, $\mathbf{q}^T \mathbf{q} = 1$. In terms of the quaternion, the attitude matrix is given by

$$A(\mathbf{q}) = \Xi^T(\mathbf{q})\Psi(\mathbf{q}) \quad (10)$$

where

$$\Xi(\mathbf{q}) \equiv \begin{bmatrix} q_4 I_{3 \times 3} + [\boldsymbol{\rho} \times] \\ -\boldsymbol{\rho}^T \end{bmatrix} \quad (11a)$$

$$\Psi(\mathbf{q}) \equiv \begin{bmatrix} q_4 I_{3 \times 3} - [\boldsymbol{\rho} \times] \\ -\boldsymbol{\rho}^T \end{bmatrix} \quad (11b)$$

with

$$[\mathbf{g} \times] \equiv \begin{bmatrix} 0 & -g_3 & g_2 \\ g_3 & 0 & -g_1 \\ -g_2 & g_1 & 0 \end{bmatrix} \quad (12)$$

for any general 3×1 vector \mathbf{g} defined such that $[\mathbf{g} \times] \mathbf{b} = \mathbf{g} \times \mathbf{b}$.

The rotational dynamics are given by the coupled first order differential equations:

$$\dot{\mathbf{q}}_I^B = \frac{1}{2} \Xi(\mathbf{q}_I^B) \boldsymbol{\omega}_{B/I}^B \quad (13a)$$

$$\dot{\boldsymbol{\omega}}_{B/I}^B = J_{\text{RSO}}^{-1} \left(\mathbf{T} + \mathbf{T}_{\text{srp}}^B - \left[\boldsymbol{\omega}_{B/I}^B \times \right] J_{\text{RSO}} \boldsymbol{\omega}_{B/I}^B \right) \quad (13b)$$

where $\boldsymbol{\omega}_{B/I}^B$ is the angular velocity of the RSO with respect to the inertial frame, expressed in body coordinates, J_{RSO} is the inertia matrix of the RSO and $\mathbf{T}_{\text{srp}}^B$ and \mathbf{T} are the torques acting on the RSO due to SRP expressed in body coordinates and the control torque, respectively.

Labeled is Training Data

Labeled training data for the data driven approaches are generated using the light curve model discussed earlier and by sampling the parameters required to define the Ashikhmin-Shirley light curve model. The SO shape and surface parameter models are randomly generated and can be generically grouped into four categories: fragment, regular polygon prisms, rocket bodies, and rectangular cuboids. The regular polygon prisms are then further divided into equilateral triangular prisms, square prisms and regular hexagonal prisms. The regular polygon prisms are prisms whose ends (i.e. top and bottom) are regular shapes. The shape of a regular polygon prism is defined by the number of sides n , side length s and height h :

$$h_{\text{regular}} = (h_{\min} + 0.01) + (h_{\max} - h_{\min} - 0.01)\mathcal{U}[0, 1] \quad (14a)$$

$$s_{\text{regular}} = (s_{\min} + 0.01) + (s_{\max} - s_{\min} - 0.01)\mathcal{U}[0, 1] \quad (14b)$$

$$(14c)$$

where $\mathcal{U}[0, 1]$ is uniform random variable between 0 and 1. Assuming constant density throughout the shape model, the moment of inertia matrices for each of the regular polygon models are given by

$$J_{\text{triangle}} = m_{\text{SO}} \begin{bmatrix} \frac{s^2}{24} + \frac{h^2}{12} & 0 & 0 \\ 0 & \frac{s^2}{24} + \frac{h^2}{12} & 0 \\ 0 & 0 & \frac{s^2}{12} \end{bmatrix} \quad (15a)$$

$$J_{\text{square}} = m_{\text{SO}} \begin{bmatrix} \frac{s^2}{12} + \frac{h^2}{12} & 0 & 0 \\ 0 & \frac{s^2}{12} + \frac{h^2}{12} & 0 \\ 0 & 0 & \frac{s^2}{6} \end{bmatrix} \quad (15b)$$

$$J_{\text{hexagon}} = m_{\text{SO}} \begin{bmatrix} \frac{5s^2}{24} + \frac{h^2}{12} & 0 & 0 \\ 0 & \frac{5s^2}{24} + \frac{h^2}{12} & 0 \\ 0 & 0 & \frac{5s^2}{24} \end{bmatrix} \quad (15c)$$

where m_{SO} is the SO mass. The rectangular cuboids are prisms defined by two side lengths s_1 and s_2 as well as the height h . The moment of inertia matrix for the cuboids are given by

$$J_{\text{cuboid}} = \frac{m_{\text{RSO}}}{12} \begin{bmatrix} s_2^2 + h^2 & 0 & 0 \\ 0 & s_1^2 + h^2 & 0 \\ 0 & 0 & s_1^2 + s_2^2 \end{bmatrix} \quad (16)$$

Models are generated by sampling side lengths and heights from a uniform distribution on the interval $[0.01, 5]$ m. For the regular polygon prisms, the number of sides are also selected randomly on the interval $[3, 6]$, with all instances of 5 sides being set to 4 as pentagonal prism models are not included. In addition to the model geometry, the material properties also need be defined. For each model, all facets are assumed to have the following: $R_{\text{spec}} = 0.7$, $R_{\text{diff}} = 0.3$, $\epsilon = 0.5$. The Phong parameters n_u and n_v are each taken to be equal to 1000 for all facets of every model. The mass of the SO is randomly sampled using the following $m_{\text{SO}} = m_{\min} + (m_{\max} - m_{\min})\mathcal{U}[0, 1]$.

Rocket body models are generated using octant triangulation of a sphere discussed in Ref. 11 which divides the surface of a sphere into N facet normals. Then rocket body models are generated by connecting two hemisphere ends of radius r with a cylinder of height l . This model is not exact

for all rocket bodies but is close enough to approximate the types of light curves seen for rocket bodies.

$$\begin{aligned}
J_{\text{rocket}} = m_{\text{SO}} & \left\{ \frac{V_{\text{cyl}}}{V_{\text{tot}}} \text{diag} \left[\frac{1}{12}(3r^2 + l^2), \frac{1}{12}(3r^2 + l^2), \frac{r^2}{2} \right] \right. \\
& + \frac{V_{\text{top}}}{V_{\text{tot}}} \text{diag} \left[\frac{1}{12}(3r^2 + l^2), \frac{1}{12}(3r^2 + l^2), \frac{r^2}{2} \right] \\
& \left. + \left(\frac{V_{\text{top}}}{V_{\text{tot}}} \left(\frac{l}{2} + \frac{3r}{8} \right) + \frac{V_{\text{cyl}}}{V_{\text{tot}}} \left(\frac{l}{2} - \frac{3r}{8} \right) \right) (I_{3 \times 3} - \mathbf{e}\mathbf{e}^T) + 2 \frac{V_{\text{top}}}{V_{\text{tot}}} r^2 \text{diag} \left[\frac{83}{320}, \frac{83}{320}, \frac{2}{5} \right] \right\} \quad (17)
\end{aligned}$$

where $\mathbf{e} = [0, 0, 1]^T$ and the volume of the top hemisphere is given by $V_{\text{top}} = 2/3\pi r^3$. It is assumed the the bottom volume is $V_{\text{bot}} = V_{\text{top}}$. The volume of the cylinder is given by $V_{\text{cyl}} = \pi r^2 l$ and the total volume is $V_{\text{tot}} = V_{\text{top}} + V_{\text{bot}} + V_{\text{cyl}}$. Finally, the fragment shapes use the cuboid model but with much smaller aspect ratios than payload shapes.

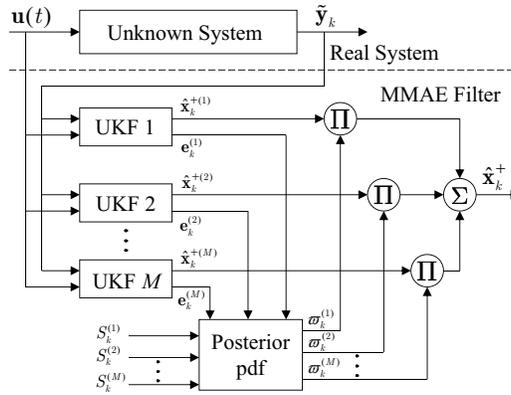


Figure 2. MMAE Process

Multiple Model Adaptive Estimation

In this section a review of MMAE is shown. More details can be found in Refs. 12 and 13. Figure 2 shows the MMAE process. Multiple-model adaptive estimation is a recursive estimator that uses a bank of filters that depend on models with different parameters, denoted by the vector \mathbf{p} , which is assumed to be constant (at least throughout the interval of adaptation). Note the stationary assumption for the state and/or output processes is not necessarily required though, i.e. time varying state and output matrices can be used. A set of distributed elements is generated from some known pdf of \mathbf{p} , denoted by $\Pr(\mathbf{p})$, to give $\{\mathbf{p}^{(\ell)}; \ell = 1, \dots, M\}$. The finite set of parameters can be the results of discretizing a continuous parameters space, selecting a set of values $\{\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots, \mathbf{p}^{(k)}\}$ dispersed throughout the region of reasonable parameter values.

The goal of the estimation process is to determine the conditional probability of the ℓ^{th} element, $\mathbf{p}^{(\ell)}$, given all the measurements. Application of Bayes' rule yields

$$\Pr(\mathbf{p}^{(\ell)} | \tilde{\mathbf{Y}}_k) = \frac{\Pr(\tilde{\mathbf{Y}}_k | \mathbf{p}^{(\ell)}) \Pr(\mathbf{p}^{(\ell)})}{\sum_{j=1}^M \Pr(\tilde{\mathbf{Y}}_k | \mathbf{p}^{(j)}) \Pr(\mathbf{p}^{(j)})} \quad (18)$$

where $\tilde{\mathbf{Y}}_k$ denotes the sequence $\{\tilde{\mathbf{y}}_0, \tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_k\}$. The conditional probability $\Pr(\mathbf{p}^{(\ell)}|\tilde{\mathbf{Y}}_k)$ will be the metric used to select the most likely model and or the most likely combination of shape models. The *a posteriori* probabilities can be computed through¹⁴

$$\begin{aligned}\Pr(\mathbf{p}^{(\ell)}|\tilde{\mathbf{Y}}_k) &= \frac{\Pr(\tilde{\mathbf{y}}_k, \mathbf{p}^{(\ell)}|\tilde{\mathbf{Y}}_{k-1})}{\Pr(\tilde{\mathbf{y}}_k|\tilde{\mathbf{Y}}_{k-1})} \\ &= \frac{\Pr(\tilde{\mathbf{y}}_k|\hat{\mathbf{x}}_k^{-(\ell)}) \Pr(\mathbf{p}^{(\ell)}|\tilde{\mathbf{Y}}_{k-1})}{\sum_{j=1}^M \left[\Pr(\tilde{\mathbf{y}}_k|\hat{\mathbf{x}}_k^{-(j)}) \Pr(\mathbf{p}^{(j)}|\tilde{\mathbf{Y}}_{k-1}) \right]}\end{aligned}\quad (19)$$

The conditional probability of the observations based on each hypothesis (likelihood), $p(\tilde{\mathbf{y}}_k|\hat{\mathbf{x}}_k^{-(\ell)})$ are given as

$$\Pr(\tilde{\mathbf{y}}_k|\hat{\mathbf{x}}_k^{-(\ell)}) = \frac{1}{\det(2\pi S_k^{(\ell)})^{1/2}} \exp\left\{-\frac{1}{2}\mathbf{e}_k^{(\ell)T} S_k^{(\ell)-1} \mathbf{e}_k^{(\ell)}\right\}\quad (20)$$

where measurement residual for the ℓ^{th} hypothesis (model) is given by

$$\mathbf{e}_k^{(\ell)} = \tilde{\mathbf{y}}_k - \mathbf{h}[\hat{\mathbf{x}}_k^-(\mathbf{p}^{(\ell)})]\quad (21)$$

and corresponding residual covariance matrix from the Unscented Kalman Filters (UKFs)

$$S_k^{(\ell)} = P_k^{vv(\ell)}\quad (22)$$

where $P_k^{vv(\ell)}$ is the innovation matrix for the ℓ^{th} filter.

Note that the denominator of Eq. (19) is just a normalizing factor. The recursion formula can now be cast into a set of defined weights $\varpi_k^{(\ell)}$, so that

$$\begin{aligned}\varpi_k^{(\ell)} &= \varpi_{k-1}^{(\ell)} \Pr(\tilde{\mathbf{y}}_{k-1}|\hat{\mathbf{x}}_{k-1}^{-(\ell)}) \\ \varpi_k^{(\ell)} &\leftarrow \frac{\varpi_k^{(\ell)}}{\sum_{j=1}^M \varpi_k^{(j)}}\end{aligned}\quad (23)$$

where $\varpi_k^{(\ell)} \equiv \Pr(\mathbf{p}^{(\ell)}|\tilde{\mathbf{Y}}_k)$. Note that only the current time measurement $\tilde{\mathbf{y}}_k$ is needed to update the weights. The weights at time t_0 are initialized to $\varpi_0^{(\ell)} = 1/M$ for $\ell = 1, 2, \dots, M$. The convergence properties of MMAE are shown in Ref. 14, which assumes ergodicity in the proof. The ergodicity assumptions can be relaxed to asymptotic stationarity and other assumptions are even possible for non-stationary situations.¹⁵

From Eq. (20) and Eq. (23) it is seen that models which have lower residuals will have probability that will increase; this will favor models that fit the observations better. Also from Eq. (20) it is seen that models which have small values for $\det(S_k^{(\ell)})$ will have probability that will grow. Assuming that all models have same measurement noise covariance matrix R_k , this will favor models that have smaller variance. Therefore the MMAE process will tend to select the maximum likelihood (minimum variance) model from the bank of models.

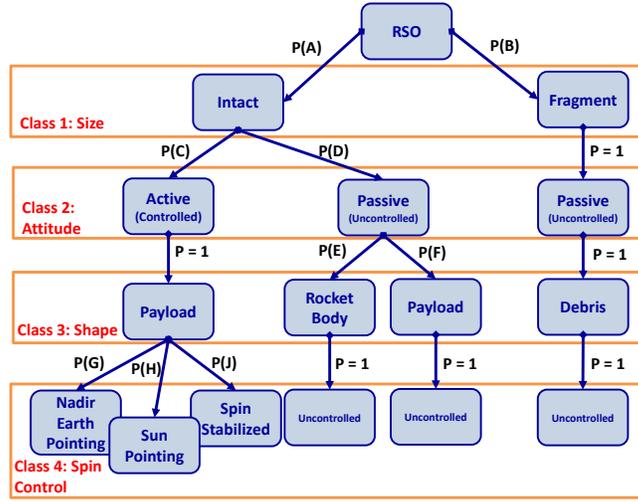


Figure 3. Overview of Classification Approach

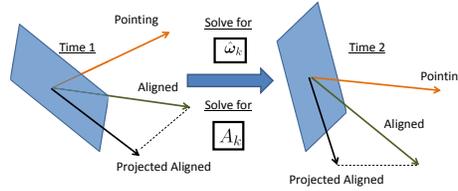


Figure 4. Overview of Angular Velocity Determination Approach

Angular Velocity Determination

When processing light curve observations it may not be valid to assume that the SO is uncontrolled, and therefore the possibility of controlled attitude states must be taken into account. Determining whether an SO has active control or not may also provide a feature state that may be used for classification. For example, a determination of whether an SO is passive or active can be made based on whether light curve observations indicate that the SO has active attitude control.

In this work, the attitude control is simulated by assuming control profiles, for example Sun pointing, Nadir pointing, and spin stabilized. Then for each control profile a desired angular velocity is determined which will allow the SO to track the relevant directions. The angular velocity profiles are used to calculate the control torque, \mathbf{T} , required to track this profile. This section discusses the attitude control approach used for calculating the desired angular velocity profile which is shown in Figure 4. The attitude control is designed to minimize the following error:

$$\mathbf{e} = \boldsymbol{\omega} - \boldsymbol{\omega}_d \quad (24)$$

Differentiating this equation with respect to time yields

$$\dot{\mathbf{e}} = \dot{\boldsymbol{\omega}} - \dot{\boldsymbol{\omega}}_d \quad (25)$$

It is desirable for the error dynamics to decay exponential over time, i.e. $\mathbf{e} \propto \exp\{-k_p t\}$, and therefore the error rate equation is desired to have the following form:

$$\dot{\mathbf{e}} = -k_p \mathbf{e} \quad (26)$$

Then using Euler's equation and assuming disturbance torques are negligible, Eq. (25) can be written as

$$\dot{\mathbf{e}} = J_{SO}^{-1} (\mathbf{T} - [\boldsymbol{\omega} \times] J_{SO} \boldsymbol{\omega}) - \dot{\boldsymbol{\omega}}_d \quad (27)$$

where T is the torque provide by the attitude actuator. Then for an exponentially decaying tracking error the desired torque expression becomes

$$\mathbf{T} = [\boldsymbol{\omega} \times] J_{SO} \boldsymbol{\omega} - k_p J_{SO} (\boldsymbol{\omega} - \boldsymbol{\omega}_d) \quad (28)$$

This expression is used to calculate the torque required to maintain the desired pointing profile.

In this section angular velocity determination approaches are discussed. Consider the following unit-vector measurement model at time t_k :

$$\tilde{\mathbf{b}}_{j_k} = A_k \mathbf{r}_j + \mathbf{v}_{j_k} \quad (29)$$

where $\tilde{\mathbf{b}}_{j_k}$ is the j^{th} pointing vector in the body frame and \mathbf{r}_j the same pointing vector in the inertial frame. The attitude matrix mapping from inertial to the body frame is denoted by A_k . Our goal is to determine the rate of change of this attitude matrix or the angular velocity. Taking the difference between successive measurements of Eq. (29) gives

$$\tilde{\mathbf{b}}_{j_{k+1}} - \tilde{\mathbf{b}}_{j_k} = [A_{k+1} - A_k] \mathbf{r}_j + \mathbf{v}_{j_{k+1}} - \mathbf{v}_{j_k} \quad (30)$$

It is assumed that the body angular velocity $\boldsymbol{\omega}$ is constant between t_k and t_{k+1} . So terms higher than first order in $\boldsymbol{\omega} \Delta t$ are ignored. With these assumptions the following first-order approximation can be used:¹⁰

$$A_{k+1} \approx [I_{3 \times 3} - \Delta t [\boldsymbol{\omega}_k \times]] A_k \quad (31)$$

In this case $\boldsymbol{\omega}_k$ is the *average* velocity, but this becomes less of a problem as the sampling interval decreases. Substituting Eq. (31) into Eq. (30) gives

$$\tilde{\mathbf{b}}_{j_{k+1}} - \tilde{\mathbf{b}}_{j_k} = -\Delta t [\boldsymbol{\omega}_k \times] A_k \mathbf{r}_j + \mathbf{v}_{j_{k+1}} - \mathbf{v}_{j_k} \quad (32)$$

Our goal is to determine an angular velocity estimate independent of attitude and the reference vectors. This is accomplished by solving Eq. (29) in terms of $A_k \mathbf{r}_i$ and substituting the resultant into Eq. (32), which yields

$$\frac{1}{\Delta t} [\tilde{\mathbf{b}}_{j_{k+1}} - \tilde{\mathbf{b}}_{j_k}] = [\tilde{\mathbf{b}}_{j_k} \times] \boldsymbol{\omega}_k + \mathbf{w}_{j_k} \quad (33)$$

where \mathbf{w}_{j_k} is the new effective measurement noise vector given by

$$\mathbf{w}_{j_k} \equiv [\boldsymbol{\omega}_k \times] \mathbf{v}_{j_k} + \frac{1}{\Delta t} [\mathbf{v}_{j_{k+1}} - \mathbf{v}_{j_k}] \quad (34)$$

Note that Δt will have finite values, since discrete-time measurements are assumed. Equation (33) can now be cast into a linear least-squares form for all measurement vectors, which leads to

$$\hat{\boldsymbol{\omega}}_k = \frac{1}{\Delta t} \left[\sum_{j=1}^{n_k} [\tilde{\mathbf{b}}_{j_k} \times]^T R_{j_k}^{-1} [\tilde{\mathbf{b}}_{j_k} \times] \right]^{-1} \sum_{j=1}^{n_k} [\tilde{\mathbf{b}}_{j_k} \times]^T R_{j_k}^{-1} (\tilde{\mathbf{b}}_{j_{k+1}} - \tilde{\mathbf{b}}_{j_k}) \quad (35)$$

where $\hat{\omega}_k$ is the estimate of ω_k . For small Δt the propagated true value of \mathbf{b}_j can be given using Eq. (31):

$$\mathbf{b}_{j_{k+1}} \approx \{I_{3 \times 3} - \Delta t [\boldsymbol{\omega}_k \times]\} \mathbf{b}_{j_k} \quad (36)$$

Substituting Eq. (36) into Eq. (31), left multiplying by $[\mathbf{b}_{j_k} \times]^T$ and right multiplying by $[\mathbf{b}_{j_k} \times]$ gives

$$[\mathbf{b}_{j_k} \times]^T R_{j_k}^{-1} [\mathbf{b}_{j_k} \times] = \bar{\sigma}_j^{-2} [\mathbf{b}_{j_k} \times]^T [\mathbf{b}_{j_k} \times] \quad (37)$$

where $\bar{\sigma}_j^2 \equiv 2\sigma_j^2/\Delta t^2$. Also, since $\mathbf{b}_{j_{k+1}}^T \mathbf{b}_{j_k} \approx 1$, it is easy to show that $[\mathbf{b}_{j_k} \times]^T R_{j_k}^{-1} (\mathbf{b}_{j_{k+1}} - \mathbf{b}_{j_k}) \approx \bar{\sigma}_j^{-2} [\mathbf{b}_{j_k} \times]^T \mathbf{b}_{j_{k+1}}$. Therefore, Eq. (35) is well approximated by

$$\hat{\omega}_k = \frac{1}{\Delta t} \left[\sum_{j=1}^{n_k} \bar{\sigma}_j^{-2} [\tilde{\mathbf{b}}_{j_k} \times]^T [\tilde{\mathbf{b}}_{j_k} \times] \right]^{-1} \sum_{j=1}^{n_k} \bar{\sigma}_j^{-2} [\tilde{\mathbf{b}}_{j_k} \times]^T \tilde{\mathbf{b}}_{j_{k+1}} \quad (38)$$

where the measurements have again been substituted in place of their true values. For this work perfect knowledge of the pointing direction in estimator is assumed and therefore the measurement error variances are irrelevant and thus set to $\bar{\sigma}_j = 1$. Then for the simulation of the training data Eq. (38) is used to determine the require angular velocity to track a particular attitude profile, and the control torque determined by Eq. (28) is in Eq. (13) used in simulate the rotational motion. The model driven approach therefore constrains each model by the assumed control profile, and the angular velocities are assumed to evolve under this assumed control law. If the data does not follow this control law this model will be assigned a low probability, and models that more closely match the actual control law will receive high probability weights. The data drive approach automatically learns how the light curve features change under different types of control.

PCA CLASSIFICATION

In this section the PCA and CNNC approaches are discussed. PCA is extensively used in feature extraction to reduce the dimensionality of the original data by a linear transformation. PCA extracts dominant features (principal components) from a set of multivariate data. The dominant features retain most of the information, both in the sense of maximum variance of the features and in the sense of minimum reconstruction error. To obtain the principal components or eigensignals, each data set from an observation is formed into a column vector, \mathbf{y}_i , with length m depending on the number of measurement per training sample. For N training samples the matrix Y (with size $m \times N$) can be defined given by

$$Y = [\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_N] \quad (39)$$

The mean measurement vector for the training set is then given by

$$\bar{\mathbf{y}} = \sum_{i=1}^N \mathbf{y}_i \quad (40)$$

Then a mean center representation of the data is given by

$$\mathbf{c}_i = \mathbf{y}_i - \bar{\mathbf{y}} \quad (41)$$

Next the principal components can be found by forming the sample covariance matrix

$$C = \frac{1}{N-1} \sum_{i=1}^N \mathbf{c}_i \mathbf{c}_i^T \quad (42)$$

Then a singular value decomposition of C can be used to write $C = V^T S U$ where $V = [\mathbf{v}_1, \dots, \mathbf{v}_m]$, $U = [\mathbf{u}_1, \dots, \mathbf{u}_m]$, $V \in m \times m$, $U \in N \times N$, and $S \in m \times N$. The singular values of C are given by $S = \text{diagonal} \{s_1, \dots, s_l, 0, \dots, 0\}$, where the number of nonzero singular values is given by rank of C . The principal components for any signal \mathbf{y}_i are given by

$$w_k = \sum_{i=1}^N \mathbf{v}_k^T (\mathbf{y}_i - \bar{\mathbf{y}}) \quad (43)$$

Then a PCA basis is selected by looking at the singular values s_i and truncating at some r where $C_r = V_r^T S_r U_r$ is a reduced representation of C . To choose the appropriate number of basis vectors r , there exist many criteria. A very popular graphical one is based on the screen plot, which exposes the eigenvalues in decreasing order. The index of the last component before the plot flattens is then selected. This is the approach used for this work.

Then the PCA classifier consists of a PCA basis which is used to calculate w_i , and a clustering approach is used which clusters based on the vector $\mathbf{w} = [w_1, \dots, w_r]$. The vector \mathbf{w}_i is then the projection of the mean centered measurement vector onto the PCA basis:

$$\mathbf{w}_i = V_r^T (\mathbf{y}_i - \bar{\mathbf{y}}) \quad (44)$$

Then the classifier is trained on training data to develop both the PCA basis representation and the clustering to map \mathbf{w}_i to a particular class. A simple nearest neighbor classifier is used to find a defined number of clusters given the PCA vector, \mathbf{w} .

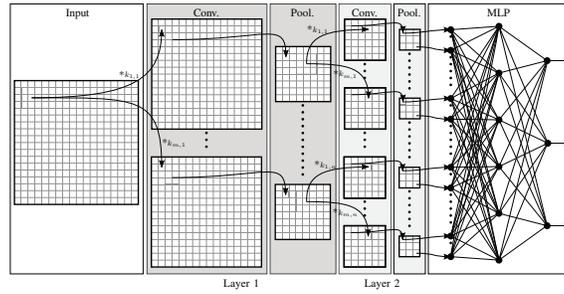


Figure 5. Convolutional Neural Network

CONVOLUTIONAL NEURAL NETWORK CLASSIFICATION

Convolutional Neural Networks (CNNs) learn both features and classification in a layered hierarchical architecture. This architecture consists of multiple layers of either convolutional filters, pooling, max pooling, and or subsampling layers. The convolutional layers are comprised of a set of convolutional kernels that learn to extract local features from data. In the CNN approach convolutional kernels are initialized randomly, and learning from data is done through an iterative process. Some CNN implementations use pretrained filters which can provide an improvement in performance. But in general it is desirable to learn these filters directly from the data. Each convolutional layer uses the feature outputs x_i with $i \in \{1, \dots, n\}$ from the previous layer as inputs. A convolutional kernel k_j with $j \in \{1, \dots, m\}$ in this layer consists of a filter kernel $k_{j,i}$ for every input x_i . These trainable kernels are convolved over a region of the input, so the kernel is always

completely within the input vector. With the convolutional operator $*$, the output y_j for the filter kernel $k_{j,i}$ and the input x is given by

$$y_j(x) = b_j + \sum_i k_{i,j} * x_i \quad (45)$$

The convolution of x_i with $k_{i,j}$ defines the output map $y_j(x)$. The number of output maps is determined by the number of convolution filters $k_{i,j}$, where $1 < j \leq m$ and m is the number of filter kernels. For the light curve problem the convolutions are one dimensional since the data is in a time series format. Then for input vector having size $(s_x, 1)$ the output vector is given by $(s_y, 1)$ for one dimensional data. The output vector size can then be calculated from the size of the kernel, s_k , and is given by

$$s_y = s_x - s_k + 1 \quad (46)$$

Then a CNN applies a series of these kernels $k_{i,j}$ in a layered fashion where each layer has a different size kernel that learns features on a given scale. To further reduce the number of output features in between convolution layers max-pooling or subsampling is used. In this work we use subsampling is used between each convolution layer. Finally, at the final layer a nonlinear function is applied to the output in a traditional neural network (shown in Figure 5). Then the CNN classification approach is trained by stochastic gradient descent by minimizing the error from the three outputs compared to the labelled data. LeCun⁷ showed that stochastic online learning is superior against the batch mode as it is faster and results in a better solutions. The weights for the output layer and the convolutional layer are updated using the following relationship:

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta \frac{\partial E}{\partial \mathbf{w}} \quad (47)$$

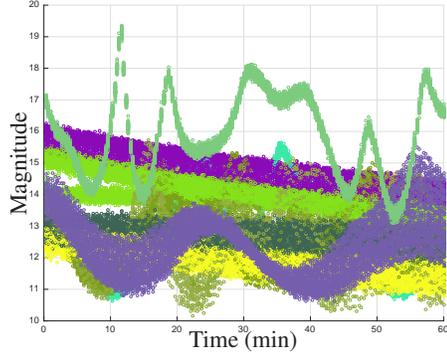
where t denotes the iteration step, and the convolutional kernel values and output neural network parameters are cased into the vector \mathbf{w} . The parameter E is the network classification error and η is the learning rate. The $\frac{\partial E}{\partial \mathbf{w}}$ is the gradient of the error with respect to the overall network parameters. This method of updating the parameters is referred to stochastic gradient descent, and the gradient is calculating with error back propagation.

SIMULATION RESULTS

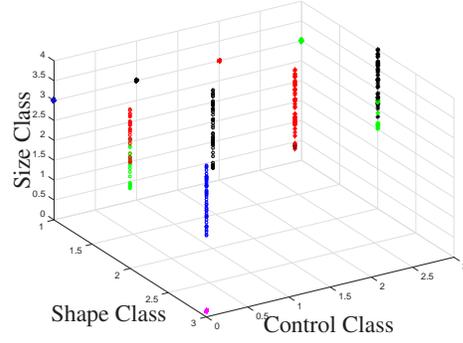
In the section the MMAE, PCA, and CNN classification approaches are tested using simulated data. The MMAE approach is evaluated against four different scenarios using the dynamics and measurement models discussed earlier. Both PCA and CNN approaches require training. These approaches are trained over 500 randomly generated scenarios. After training of the two approaches they are tested against data not used in the training set.

For all scenarios, an SO is in near geosynchronous orbit with orbital elements given by $a = 42,364.17$ km, $e = 2.429 \times 10^{-4}$, $i = 30$ deg, $\omega = \Omega = 0.0$ deg and $M_0 = 91.065$ deg. The simulation epoch is 15-March-2010 at 04:00:00 GST. The initial quaternion and angular rate of the SO are given by $\mathbf{q}_I^B \equiv [0.7041 \ 0.0199 \ 0.0896 \ 0.7041]^T$ and $\boldsymbol{\omega}_{B/I}^B = [206.26 \ 103.13 \ 540.41]^T$ deg/hr.

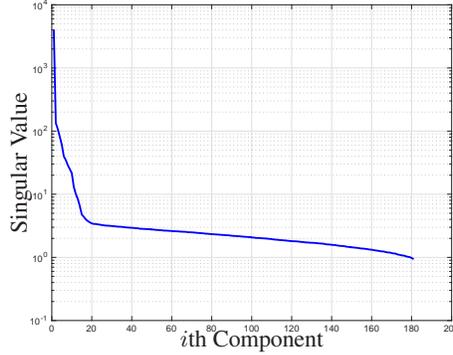
Brightness magnitude and angle observations are simulated using a ground station located at 20.71° North, 156.26° West longitude and 3,058.6 m altitude. Measurements constructed using instantaneous geometry are corrupted by zero-mean Gaussian white noise with standard deviations of 1 arc-seconds on the azimuth observation, 1 arc-seconds on the elevation observation and 0.1 for



(a) Measurement PCA Clusters



(b) Classification State and PCA Clusters



(c) Classification State and PCA Clusters

Figure 6. PCA Clustering Approach

the brightness magnitude.¹⁶ Observations are available every 5 seconds for one hour. The initial states for each filter are given by $\hat{\mathbf{q}}_I^B(t_0) = [0.7500 \ 0.0712 \ 0.0947 \ 0.6508]^T$ (a 10 degree attitude error), $\hat{\boldsymbol{\omega}}_{B/I}^B(t_0) = [220.26 \ 117.13 \ 554.41]^T$, $\hat{a}(t_0) = 42,364.148255 \text{ km}$, $\hat{e}(t_0) = 2.4290 \times 10^{-4}$, $\hat{i}(t_0) = 30.0083 \text{ deg}$, $\hat{\omega}(t_0) = -1.172 \text{ deg}$, $\hat{\Omega}(t_0) = 0.0 \text{ deg}$ and $\hat{M}_0(t_0) = 92.137 \text{ deg}$. Initial 3σ values are taken to be 20 deg for the attitude states, 72 (deg/hr) on the angular rates, 300 km on position and 3 (km/s) on velocity. The process noise for the estimation filters are taken as $Q_k^{(\ell)} = 0$ for this proof of concept simulation.

PCA CLASSIFICATION RESULTS

The PCA approach is the simplest of the approaches considered in this paper. The training data is used to form a matrix of size $m \times N$, where m is the number of observation per each training sample and N denotes the total number of training samples. Then a singular value decomposition is taken of this matrix. Only basis vectors with greater than a user defined cut-off are retained. The results for the PCA classification approach are shown in Figure 6. For this work, the first 20 singular values are considered. From Figure 6(c) it can be that the singular values flatten at around the 20th component. Then only using the basis vectors for the first 20 singular values, the measurement vector is projected onto the PCA these basis vectors using Eq. (44) to produce the classification features. Once each measurement is used to calculate \mathbf{w} , a nearest neighbor classifier is used to

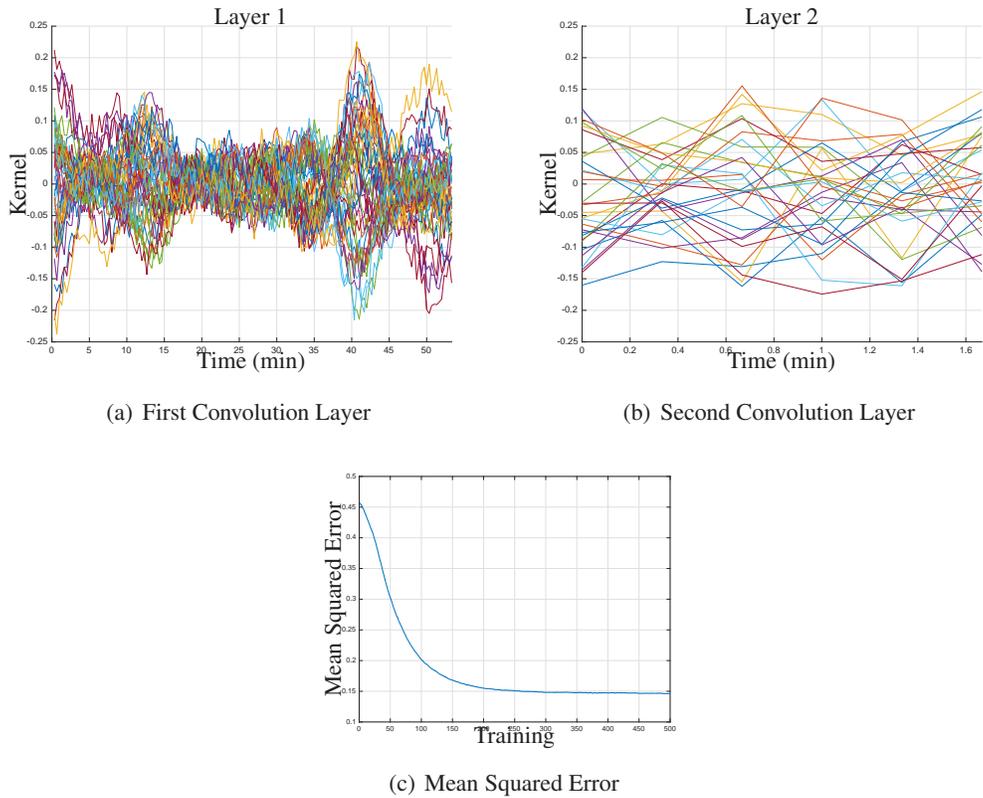
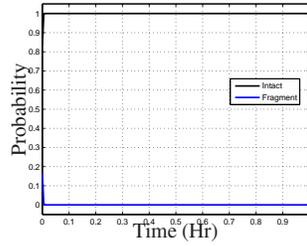


Figure 7. CNN Classifications Results

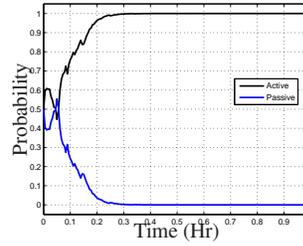
find the given number of clusters using the feature vector w . Then this clustering approach can be used on data not considered in the training set. In this work 9 clusters are looked for in the data, and these clusters are compared to the class of each model. Figures 6(a) and 6(b) show the cluster measurements and the clusters determined from the data label on the actual classes. From Figure 6(a) it can be seen that the clustering approach clearly finds meaningful clusters of different types of light curves. Figure 6(b) shows how the clusters relate to the classes, where the figure axes are the values of each class. Each training sample is shown in Figure 6(b) labeled by a color corresponding to the different classes. From this figure it can be seen that the main classes are identified, although there were some errors. Given the simplicity of the PCA classification approach it did remarkably well at classifying the training set.

CNN CLASSIFICATION RESULTS

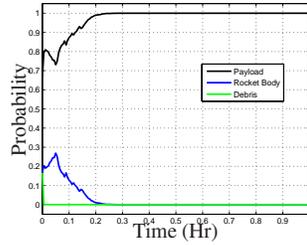
The CNN classification approach is more complex than the PCA approach but it is expected that this approach can perform better while being able to handle data sets of varying size and structure. The CNN used in the work uses a four layer structure, where the order is convolution, subsampling, convolution, the subsampling. The first and third layer used a 160 and 6 unit size kernel, respectively. Both subsampling layers subsample the output by a factor of 2. Then the training data consists of simulated light curve measurements as inputs and class states as outputs. For this example only shape classes are considered, but other classes can be added in the same CNN or with independent CNNs for each class. The training set used here is the same as the PCA simulation



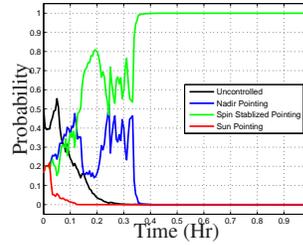
(a) Class 1: SO Size Features



(b) Class 2: Attitude Control

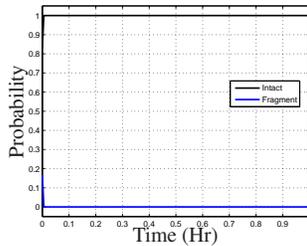


(c) Class 3: SO Type

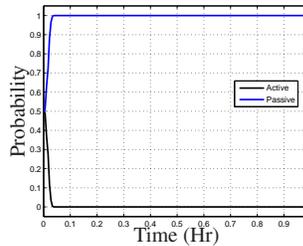


(d) Class 4: Spin Control State

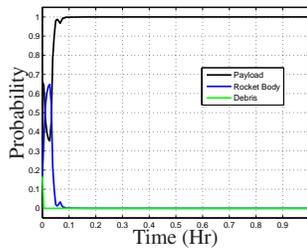
Figure 8. Spin stabilized Bus Example



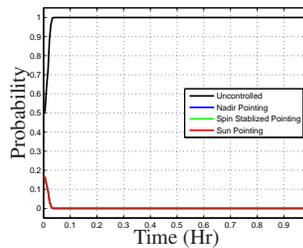
(a) Class 1: SO Size Features



(b) Class 2: Attitude Control



(c) Class 3: SO Type



(d) Class 4: Spin Control State

Figure 9. Uncontrolled Bus Example

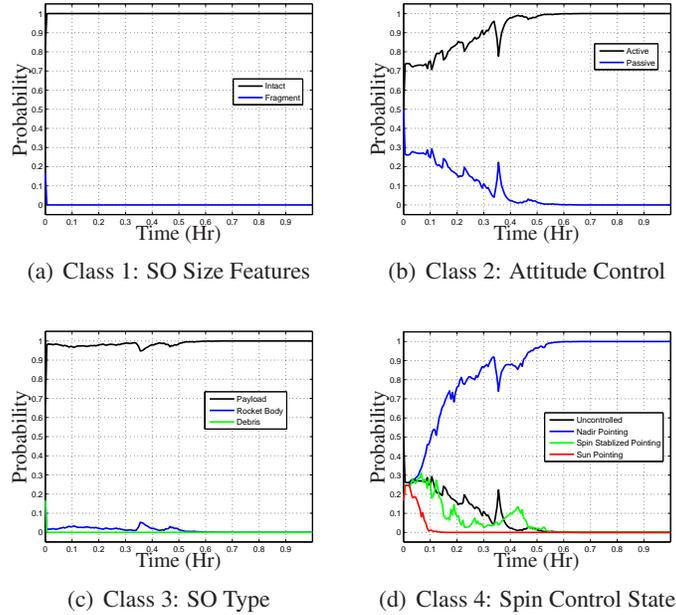


Figure 10. Nadir Pointing Bus Example

example shown in the last section. Figure 7 shows the CNN classification kernel features estimated during the training stage. From Figure 7(a) it can be seen that the CNN approach learns light curve features that are relevant to the classification of shape. The CNN approach reaches an overall accuracy of 85.4% correct classification. Figure 7(c) shows the mean squared error for classification of shape with the CNN as a function of training data point. From this figure it can be seen that for the size of kernels using the error has converged to a steady state.

MMAE Classification Results

Four simulation scenarios are presented to show the performance of the MMAE based classification approach to classify an SO from magnitude and angles observations. In each scenario a different object is selected that falls into a different class. The objects selected are spin stabilized bus, uncontrolled bus, nadir pointing bus, and uncontrolled rocket body.

Figure 8 shows the classification results for a spin stabilized bus. The simulation results use the same initial conditions as the MMAE examples, and the true model is in the bank. In this case the bus models are considered to be regular cuboids with aspect ratio larger than 0.1. As discussed in the classification section there are a number of classes the classification approach determines, whether the SO belongs to these classes or not. The first class is whether the SO is intact or a fragment. In this case the true model is an intact bus. From Figure 8 it can be seen that this determination is made relatively quickly. The second determination is whether the SO is Active or Passive, which is also shown in Figure 8. The Active or Passive decision is made using the probability of all Active and Passive models in the bank. From Figure 8 shows that the Active model has large probabilities after 0.2 hours.

Additional examples are shown for uncontrolled bus (Figure 9), nadir pointing bus (Figure 10), and uncontrolled rocket body (Figure 11). The approach shows good performance for these exam-

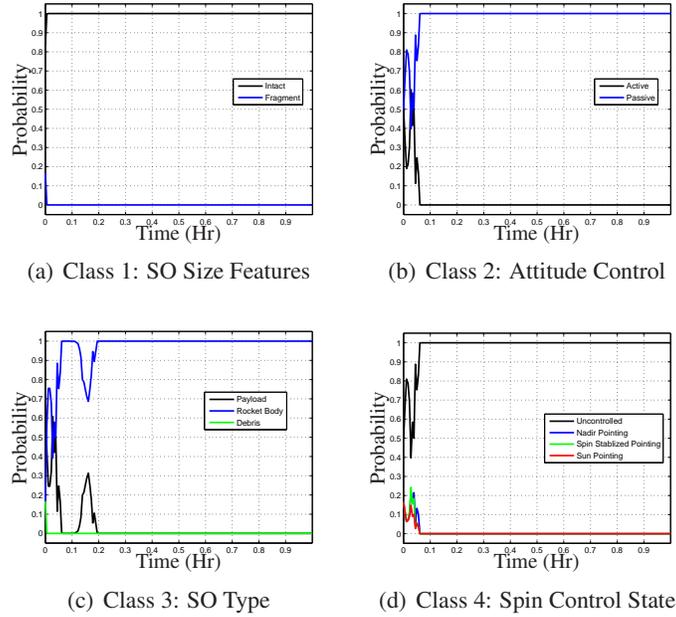


Figure 11. Rocket Body Example

ples at determining the correct class. From the figures it can be seen that some objects take longer to classify. This is due to the fact that for some spin states the light curves are similar, but for uncontrolled spin states the light curves differ significantly. This can be seen from nadir pointing bus (Figure 10), which takes the longest to converge to its classification, and from uncontrolled rocket body (Figure 11), which converges the fastest to its classification.

CONCLUSION

In this paper a model driven classification approach based on the MMAE scheme is used for space object classification using light curves. Alternatively, two data driven classification approaches were also used, both the Convolutional Neural Network and PCA classification approaches are considered. The MMAE approach used a filter to reduce brightness magnitude and angle data, and is able to determine the probability of each model. An approach is presented that uses the probability from MMAE process to determine the probability that a given SO falls in a given class. The classification approach determines whether the SO is intact or fragment, its control states, the type of control state, and whether it is rocket body, payload, or debris. Simulation results are shown for a number of examples and good results for classification are shown.

The data driven approaches use training data to build up a classification mapping. The PCA approach learns a PCA basis which is used to map observation vectors into a feature space. Then in the feature space simple clustering is applied to find relationships between the data. These clusters are then used to define the classes. This work considers only the first 20 singular values for the PCA basis and looked for 9 clusters in the data. These clusters are compared to the class of each model. It was found that the clustering approach clearly found meaningful clusters of different types of light curves. The PCA classification approach did remarkably well on classifying the training set.

The second data drive approach applied was the CNN classification approach. CNN is a feature learning and classification architecture. The convolutional layer is built from a set of filters to extract local features. This work used two convolutional layers with 160 and 6 unit size kernels, respectively. For the CNN classification, this work only considered shape classes for class state. On the shape classification problem the CNN approach reached a overall accuracy of 85% correct classification. There is still room for improvement in the data drive classification and high accuracy has been seen in other applications, therefore this is goal for this application. The data driven approach has the benefit of having simpler implementation requiring no modeling but the classification results are still not as good as a model based approaches (i.e. the MMAE approach).

REFERENCES

- [1] Linares, R., Jah, M. K., Leve, F. A., Crassidis, J. L., and Kececy, T., "Astrometric and Photometric Data Fusion For Inactive Space Object Feature Estimation," *Proceedings of the International Astronautical Federation*, Cape Town, South Africa, Sept. 2011, Paper ID: 11340.
- [2] Linares, R., Jah, M. K., and Crassidis, J. L., "Inactive Space Object Shape Estimation via Astrometric And Photometric Data Fusion," *AAS/AIAA Space Flight Mechanics Meeting*, No. AAS Paper 2012-117, Charleston, SC, Jan.-Feb 2012.
- [3] Linares, R., Jah, M. K., and Crassidis, J. L., "Space Object Area-To-Mass Ratio Estimation Using Multiple Model Approaches," *Advances in the Astronautical Sciences*, Vol. 144, 2012, pp. 55–72.
- [4] Linares, R., Jah, M. K., Crassidis, J. L., and Nebelecky, C. K., "Space Object Shape Characterization and Tracking Using Light Curve and Angles Data," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 1, 2013, pp. 13–25.
- [5] Hinks, J. C., Linares, R., and Crassidis, J. L., "Attitude Observability from Light Curve Measurements," *AIAA Guidance, Navigation, and Control (GNC) Conference*, No. 10.2514/6.2013-5005, AIAA, Boston, MA, August 2013.
- [6] Hotelling, H., "Analysis of a Complex of Statistical Variables into Principal Components." *Journal of educational psychology*, Vol. 24, No. 6, 1933, pp. 417.
- [7] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, Vol. 86, No. 11, 1998, pp. 2278–2324.
- [8] Wetterer, C. J., Linares, R., Crassidis, J. L., Kececy, T. M., Ziebart, M. K., Jah, M. K., and Cefola, P. J., "Refining Space Object Radiation Pressure Modeling with Bidirectional Reflectance Distribution Functions," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 1, 2013, pp. 185–196.
- [9] Ashikmin, M. and Shirley, P., "An Anisotropic Phong Light Reflection Model," Tech. Rep. UUCS-00-014, University of Utah, Salt Lake City, UT, 2000.
- [10] Shuster, M. D., "A Survey of Attitude Representations," *Journal of the Astronautical Sciences*, Vol. 41, No. 4, Oct.-Dec. 1993, pp. 439–517.
- [11] Kaasalainen, M. and Torppa, J., "Optimization Methods for Asteroid Lightcurve Inversion: I. Shape Determination," *Icarus*, Vol. 153, No. 1, 2001, pp. 24–36.
- [12] Brown, R. G. and Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, New York, NY, 3rd ed., 1997, pp. 353–361.
- [13] Stengel, R. F., *Optimal Control and Estimation*, Dover Publications, New York, NY, 1994, pp. 402–407.
- [14] Anderson, B. D. O. and Moore, J. B., *Optimal Filtering*, chap. 10.1, Dover Publications, Mineola, NY, 2005.
- [15] Anderson, B. D. O., Moore, J. B., and Hawkes, R. M., "Model Approximations via Prediction Error Identification," *Automatica*, Vol. 14, No. 6, Nov. 1978, pp. 615–622.
- [16] Hall, D. T., Africano, J. L., Lambert, J. V., and Kervin, P. W., "Time-Resolved I-Band Photometry of Calibration Spheres and NaK Droplets," *Journal of Spacecraft and Rockets*, Vol. 44, No. 4, July 2007, pp. 910–919.