An investigation into multi-dimensional prediction models to estimate the pose error of a quadcopter in a CSP plant setting

Jacobus C. Lock, Willie J. Smit, and Johann Treurnicht

Citation: AIP Conference Proceedings 1734, 130012 (2016); doi: 10.1063/1.4949222

View online: http://dx.doi.org/10.1063/1.4949222

View Table of Contents: http://scitation.aip.org/content/aip/proceeding/aipcp/1734?ver=pdfcov

Published by the AIP Publishing

Articles you may be interested in

The role of CSP in Brazil: A multi-model analysis
AIP Conf. Proc. 1734, 110004 (2016); 10.1063/1.4949201

Vlasov multi-dimensional model dispersion relation
Phys. Plasmas 21, 072103 (2014); 10.1063/1.4886122

Investigating Multi-Dimensional Effects in Single-Crystal Sapphire

Modeling DDT in granular explosives with a multi-dimensional hydrocode

Navigating Through Multi-Dimensional Data Sets: Dynamical User Interface Issues
Comput. Phys. 4, 650 (1990); 10.1063/1.4822958
An Investigation Into Multi-Dimensional Prediction Models to Estimate the Pose Error of a Quadcopter in a CSP Plant Setting

Jacobus C. Lock¹,a),d) , Willie J. Smit¹,b) and Johann Treurnicht²,c),e)

¹ Solar Thermal Energy Research Group, Stellenbosch University, South Africa
² Electronic Systems Laboratory, Stellenbosch University, South Africa

a) jaycee.lock@gmail.com
b) wjsmit@sun.ac.za
c) jtreurn@sun.ac.za
d) URL: sterg.sun.ac.za
e) URL: www.esl.sun.ac.za

Abstract. The Solar Thermal Energy Research Group (STERG) is investigating ways to make heliostats cheaper to reduce the total cost of a concentrating solar power (CSP) plant. One avenue of research is to use unmanned aerial vehicles (UAVs) to automate and assist with the heliostat calibration process. To do this, the pose estimation error of each UAV must be determined and integrated into a calibration procedure. A computer vision (CV) system is used to measure the pose of a quadcopter UAV. However, this CV system contains considerable measurement errors. Since this is a high-dimensional problem, a sophisticated prediction model must be used to estimate the measurement error of the CV system for any given pose measurement vector. This paper attempts to train and validate such a model with the aim of using it to determine the pose error of a quadcopter in a CSP plant setting.

INTRODUCTION

Concentrating solar power (CSP) is an attractive source of renewable energy which allows thermal energy to be stored for use at night time. However, a major hurdle to the construction of a CSP plant, in particular the central receiver configuration, is its hefty price tag. With CSP costing more per Watt than fossil fuels, as well as other renewable energy sources such as photovoltaic panels or wind turbines [1, fig. E.S.1], it’s difficult to convince financiers to invest in a central receiver CSP plant. Approximately 40% of the initial capital expenditure of a CSP plant are the thousands of heliostats placed around the receiver tower [2, p. 64]. To prevent mirror deformation, wind loads, impacts from foreign objects and ground settling from affecting the heliostat’s tracking path, each heliostat frame is designed to be heavy and sturdy and are placed on strong foundations. They also come equipped with expensive actuators and gearboxes that minimise error build-ups during a heliostat’s operation, contributing a significant portion to the total cost of each heliostat.

CSP as a concept is old and well documented, but the technologies used are still relatively new and widely researched. As such, CSP plant design has not yet reached full maturity in terms of efficiency and optimal design. Therefore, there are still ways to reduce the cost of CSP plants, such as improving the heliostat control scheme, optimising heliostat design and improving the thermal storage mechanisms, amongst others [3, p. 22].

Given the sheer size of a typical heliostat field, a small displacement in a mirror could influence the accuracy and effectiveness of that heliostat which is extremely undesirable given the lengthy calibration process required. Stellenbosch University (SU) and its Solar Thermal Energy Research Group (STERG) are attempting to make heliostats lighter, cheaper and reduce the need for expensive actuators and gearboxes, thereby allowing the heliostats to be placed on cheaper foundations with cheaper, less accurate actuators. This will lead to significant cost savings in the plant’s manufacturing stage. However, to achieve this without compromising the accuracy of the heliostats during operation,
the lengthy calibration procedure currently in use may have to be optimised and possibly redesigned to allow the heliostats to be calibrated more often on a monthly, or even weekly, basis.

The current calibration procedure uses the sun as a light source and a section of the receiver tower as a reference target. This constrains the process to take place in day time and only allows one heliostat to be calibrated at a time and since a CSP heliostat field typically contains several thousand heliostats, this procedure may become very time-consuming. See Figure 1a for a diagrammatic example of the calibration procedure.

In this regard, STERG is investigating the possibility of using quadcopter unmanned aerial vehicles (UAVs) to autonomously perform or assist with heliostat calibration by, for example, making one quadcopter the light source and another the receiver. This will allow as many heliostats to be calibrated simultaneously as there are pairs of quadcopters available. See Figure 1b for an illustration of the proposed calibration procedure. This approach requires an estimate of the quadcopter’s pose estimation error, where the pose is a six-dimensional vector containing position \((x, y, z)\) and orientation \((\text{roll}, \text{pitch}, \text{yaw})\) information. The pose estimation error refers to the difference between a quadcopter’s on-board pose estimate and its real pose. Currently, the only pose measurements of a quadcopter available are its own on-board estimates produced by a combination of readings from its sensor suite. This suite typically includes a gyroscope, accelerometer and GPS. The on-board data cannot be used here, since its error has not yet been quantified.

The pose error of indoor quadcopters have been determined before with indoor motion tracking and measurement systems, such as the Vicon motion tracking system which has been shown to have millimetre-levels of accuracy [4]. However, these results and measurement methods are not applicable in the outdoor case, since testing indoors would deny a quadcopter its GPS coordinates. An outdoor quadcopter’s GPS sensor readings provide an absolute position value, unlike its accelerometer, for example, whose readings are integrated twice to provide approximate position data, thereby making its readings prone to drift with time. Given the GPS sensor’s importance to an outdoor quadcopter’s localisation ability, it’s crucial that a quadcopter have access to its GPS readings if we are to properly determine how well it estimates its pose, making it necessary to perform the measurements in the outdoors.

Since laser trackers and radars are expensive and at the time of writing unavailable, a computer vision (CV) system was implemented and tested. It’s a simple system consisting of a single calibrated camera and a laptop running the object tracking software. It works by tracking the 2D coordinates of the corners on a chessboard pattern. These coordinates are then fed to a Perspective n-Points (PnP) problem solver, as implemented in the OpenCV library [5] [6] [7], which produces six dimensional pose data of the board. Attaching such a chessboard pattern to a quadcopter and tracking it with the CV software allows us to estimate the quadcopter’s position, enabling us to find the quadcopter’s pose error by comparing the CV system’s measurements with the quadcopter’s own estimates.

The main goal of this project is to use the CV system to determine the error in the pose estimation of a quadcopter in flight. Data of the CV system’s measurement error, i.e. the difference between the chessboard’s true position and
the CV system’s measured position, has been collected as part of previous work. However, subsequent field tests will be carried out under different circumstances and conditions where the measurement error will be unknown. Therefore, a reliable method of determining the CV system’s measurement error must be found and implemented. In this paper, we attempt to implement a reliable method of determining and drawing an error band around the CV system’s measurements. Once we are able to draw this error band, the CV system can be used to determine the pose error of a quadcopter, providing some insight into whether a UAV-based calibration procedure is feasible and how it can be implemented.

METHODOLOGY

In this paper, we attempt to create a model which we can use to predict the dimensional errors of the CV system’s measurements as accurately as possible. The CV system’s measurement data consists of six-dimensional translation and orientation data, providing a high-dimensional problem where previous field tests have shown that these dimensions are highly dependant on one another. This indicates that the error for subsequent measurements will need to be interpolated by some prediction model, since the CV system measurement error differs from sample to sample and test to test. This model’s error predictions will be crucial in later phases of the project when the CV system will be used in outdoor field tests.

This section discusses the decisions made and steps taken into creating the prediction model. First, the CV system is briefly discussed, followed by a discussion on the prediction model, as well as the process used to train it. Finally, the validation procedure and results are discussed.

CV Measurement System

The measurement system devised to measure the pose of a quadcopter is based on existing CV technology. Using the PnP problem solving algorithms packaged with the OpenCV library it’s possible to determine a single camera’s pose relative to a flat calibration pattern, such as a chessboard pattern. Conversely, by keeping the camera still and attaching the calibration pattern to the underside of a quadcopter, it’s possible to determine the quadcopter’s pose. The CV measurement system consists of a single camera (a Microsoft LifeCam HD-5000 in our case) capturing video data at a resolution of 640×480 pixels at 30 frames per second. It was found that recording video data in high-definition did not produce significantly better results and slowed the data extraction and pose estimation process down considerably. Pictures of the camera and chessboard with the Vicon system markers attached can be seen in Figure 2.
Before the CV system was used to make measurements, the true error of its measurements was first determined. This was achieved by comparing the CV system’s measurements with that of the state-of-the-art Vicon indoor motion tracking system located in the 3D motion tracking laboratory at SU’s Medicine campus at Tygerberg. The Vicon system has a reported accuracy of a few millimetres and its measurements were therefore taken as ground-truth values. The true measurement error of the CV system was then determined by comparing the CV and Vicon systems’ measurements. We select our training and validation data sets from the data gathered during this Vicon test. More detail on these data sets are provided in later sections of this paper.

Inspection of the CV system’s error covariance matrix from the Vicon test and its large off-diagonal elements shows that there is strong interdimensional dependence. This means that the error in any dimension varies with the measurements made in the other 5 dimensions. This, along with the high-dimensionality of the measurement problem, means that there isn’t a constant error margin since it varies with the board’s pose relative to the camera. It’s therefore necessary to implement a sophisticated model to predict the CV system’s measurement errors for any set of pose measurements. The dimensional complexity of the error is demonstrated by the contour plots of Figure 3 where the absolute values of the error in each dimension are plotted over the $x-y$ plane. There are other plots for each dimension combination, but the $x$ and $y$ dimension plots are sufficient to illustrate the point.

From Figure 3 it can be seen that the error doesn’t remain constant or predictable. In some cases, as with the error in the yaw dimension of Figure 3f, there are multiple peaks which makes the problem even more complex.

**Error Prediction Model**

In this section we discuss our choice of prediction model to estimate the error of the CV system measurements. Then we discuss the model training strategy, as well as the validation procedure.

**Background**

The CV system’s measurement data is highly dimensional and dependant on the other dimensional values. Subsequently, the error measurements between tests will differ from one another due to differing conditions and poses, making it nigh impossible to predict the CV system’s resulting measurement error in a simple manner. Therefore, a sophisticated prediction model capable of interpolating irregular, multi-dimensional data is required.

A traditional neural network (NN) would be suitable to perform the prediction and can be trained to take a single input vector (the CV measurement data) and produce a single output (the expected error for the measurement). How-
ever, as demonstrated in Figure 3, the error data is irregular with multiple peaks and troughs and also has significant noise associated with it. A Radial Basis Function (RBF) network is better suited to handle such data [8].

The RBF neural network works similar to a traditional NN in that it can be trained to take a single measurement input and produce a corresponding expected error output. Details on how the RBF works is beyond the scope of this paper. However, the basic RBF function is given by

$$f(x_i) = \sum_{j=1}^{M} \lambda_j \phi(||x_i - x_j||).$$

(1)

as shown by [9]. Equation 1 means that for the data set $\{<x_i, f(x_i)>\}_{i=1}^{N}$, where $x_i$ is the domain coordinate, $f(x_i)$ the measured value, $M$ the number of RBF kernels generated by the training algorithm and $N$ the number of samples there exists a set of linear equations that best describes the data. The training algorithm isn’t described here, but the parameters $\lambda_j$ (the kernel weighting factor) and $x_j$ (the RBF kernel centre) are determined and optimised during the training process in a way that optimally covers and describes the training data set.

The resulting set of linear equations from Equation 1 can be used to interpolate new error data. However, the RBF needs to be trained with good quality data first.

**Model Training**

An RBF is good at interpolating complicated, high dimensional and irregular data and the training process is relatively straightforward. However, one of the RBF’s drawbacks is that it does not work well when interpolating data when the input does not roughly fall within the range of the training data set. This means that values that are fed into the RBF model must overlap with the training data set to a degree. It’s therefore important to train the RBF with data which one might expect to encounter in subsequent field tests.

For our training set, we select 50 random, normally distributed samples for each of the six dimensions from the Vicon test set, giving a diverse cloud of 300 measurement points. To verify that the training points are representative of future field test results, the training data is plotted against data from a typical field test that was conducted with a quadcopter in flight. The scatter plots of the position and orientation vectors are given in Figure 4.
Figure 4 shows that the training points are fairly representative of typical future tests since the test points fall largely within the training limits. The training data set can therefore be used to train the RBF.

Since the measurement vector contains position data in metres and orientation data in degrees, the training data is first normalised to a range of [-1, 1] before it’s used to train the RBF. This was done by simply dividing the measurement and error training data by each dimension’s absolute maximum value. These values will be used in the future to normalise the subsequent input data into the RBF and to denormalise the resulting output error vectors.

With the data verified as optimally spread and normalised, the training data set is now ready to be used to train the RBF. The RBF is implemented using Matlab’s Neural Network Toolbox. The `newrb()` function is fairly automated and does the training, optimisation and interpolation by itself. The function prototype for `newrb()` is given by

\[
\text{rbf} = \text{newrb}(
    \text{P}, \text{T}, \text{goal}, \text{spread}, \text{MN}, \text{DF})
\]

where \( P \) is a matrix of input vectors and \( T \) a matrix of target output vectors. Other parameters of interest here are the `spread` and `MN` parameters. The `spread` parameter determines the strictness of fit which is the extent to which the network will attempt to fit itself to the training data. \( MN \) determines the maximum number of neurons in the network. It’s important to find the right balance between the number of neurons and the strictness of fit parameters to ensure that overfitting doesn’t occur. Overfitting occurs when a network is trained to strictly, producing a non-generalised model which delivers poor performance when it’s presented with data that differs too much from the training data.

During training, the `newrb()` function attempts to create a number of nodes, or kernels, that optimally describes the data. It does this by minimising the distance, \( r_{ij} \), between the data points and kernel centres, given by

\[
\phi(r_{ij}) = \phi(||x_i - x_j||),
\]

where \( \phi(r) \) is a Gaussian function given by

\[
\phi(r_{ij}) = -e^{-r_{ij}^2}.
\]

With the training data selected and with the `newrb()` function, the RBF is ready to be trained. During training, the `spread` and `MN` parameters are adjusted to produce a network which outputs the smallest mean square error (MSE) deviation from the true output of the training and validation sets. The network validation process is discussed next.

**Model Validation**

Before the newly trained RBF network can be used with new measurement data, its performance must first be evaluated and validated by a validation data set. This was done by testing the RBF with a validation test set selected from the Vicon test - the same test the training set was selected from. Since the training set was specifically selected to be normally distributed across the data spectrum, any number of random validation samples that aren’t part of the training set may be selected from the data pool.

With the validation data, consisting of the CV system measurement data and the corresponding ground-truth errors from the Vicon test, model validation can take place. This is done by feeding the pose data from the CV system’s validation set into the RBF network. The error estimate produced by the RBF is then compared to the validation set’s error data, providing an indication of how well the RBF predicts the true error. Taking the MSE values of the error for each dimension provides an indication of the RBF estimate’s average deviation from the true error.

Different combinations of `MN` and `spread` parameters were tested to find a combination that produces a good result. In this case, the optimal result would be an RBF which produces the smallest MSE in both the training and validation cases. A set of parameters that works well is a network with 8 nodes with a `spread` parameter of \( 9 \times 10^{-1} \).

A frequency histogram plotting the RBF’s error estimate deviation from the true error for the validation set is used as a measure of the network’s performance. An indication of a good parameter combination would be indicated by a narrow plot centred around 0, implying a zero mean and a small standard deviation. The training results and validation plots are discussed next.

**RESULTS**

The network was initialised with 8 nodes and a `spread` parameter of \( 9 \times 10^{-1} \). The MSE for the training set is \( 5.1 \times 10^{-3} \) and \( 1.9 \times 10^{-2} \) with the validation data set. It was found that this parameter combination gives a good model of the data without overfitting it.
\( \mu = 0.0724 \text{ m}, \sigma = 0.4174 \text{ m} \)

\( \mu = 0.1663 \text{ m}, \sigma = 0.1572 \text{ m} \)

\( \mu = 0.7731 \text{ m}, \sigma = 0.5484 \text{ m} \)

\( \mu = 17.62^\circ, \sigma = 8.158^\circ \)

\( \mu = 5.027^\circ, \sigma = 6.050^\circ \)

\( \mu = 9.405^\circ, \sigma = 10.67^\circ \)

**FIGURE 5.** Error deviation frequency histograms for the 6 different dimensions. The mean (\( \mu \)) and standard deviation (\( \sigma \)) for each plot is given in the plot’s caption. The error deviation is the difference between the true error and the RBF’s error output.
Figure 5 shows the resulting frequency histogram plots of the RBF’s error estimate deviation from the true error, along with a superimposed normal distribution graph based on the means and standard deviations of the data.

From Figure 5 it can be seen that the frequency plots look to be normally distributed about some mean value. In the z dimension the error deviation is centred around a relatively large value with a large standard deviation. However, since this is the depth dimension it does not come as a surprise that it produces the worst estimate, but the magnitude of the deviation is worrisome and should be kept in mind when using the network. Similarly, the roll dimension is also centred around a fairly large mean value, but with a standard deviation comparable to the other dimensions’. The means and standard deviations in the other dimensions are also quite large, but in itself they provide another measure of accuracy which can be used when the CV system is used for measurement.

Unfortunately, given the lack of literature for this specific application of a neural network, there are no other results to compare ours to. Regardless, there are ways that our network can be improved which are being investigated. The simplest of which are creating a new, more diverse set of training data and filtering out the noisy data points.

With these results, the RBF is ready to be used with real flight-test data which can then be used to determine the pose estimation accuracy of an outdoor quadcopter in flight.

CONCLUSION

With the RBF trained with a good parameter combination, it can be used to estimate the measurement error made by the CV system in subsequent measurement tests with a real quadcopter in flight. This data can then be used to find the pose estimation error of a quadcopter. With the pose estimation error known, it can be integrated into a heliostat calibration model that is currently under development which will allow a quadcopter to be used to autonomously calibrate a heliostat.

REFERENCES