Coordinate Transform Definition

Owen Lu
Electroimpact Inc.
owenl@electroimpact.com

***Abstract*— In order to generate a transformation matrix efficiently, a method is devised using three angular parameters and three translation parameters. The method simply combines an angular parameter ψ with a unit vector generated by θ, ϕ using standard spherical coordinates which generates a description of axis-angle rotation. This is then concatenated with a translation vector and can be easily applied to a series of augmented vectors.**

I. Spherical Unit Vector Description

The projector model is made up with 5 main variables. These variables describe the input ray, the two galvanometer axes and the two mirror normal vectors. Thus, all five variables have a unit vector contained within them. To generate these unit vectors without constraints on the magnitude or further normalization calculations, spherical coordinates are used.

$$\bar{u} = [\sin(\phi)\cos(\theta) \quad \sin(\phi)\sin(\theta) \quad \cos(\phi)] \tag{1}$$

II. Quaternion Description

In general, an axis angle rotation can be described with unit vector $\bar{u}$ combined with magnitude of rotation ψ by using quaternions. Since $\bar{u}$ can be described in spherical coordinates with the radius $\rho = 1$ the quaternion q can be written as a function of 3 angles $q(\psi, \theta, \phi)$.

$$q = \left[\cos\left(\frac{\psi}{2}\right) \quad u_x \sin\left(\frac{\psi}{2}\right) \quad u_y \sin\left(\frac{\psi}{2}\right) \quad u_z \sin\left(\frac{\psi}{2}\right)\right] \tag{2}$$

III. Rotation matrix description

Ultimately, the quaternion is used to generate a rotation matrix.

$$q = [q_r \quad q_i \quad q_j \quad q_k] \tag{3}$$

The rotation can be applied to vector p such that p' is the product of the rotation matrix and original vector p. When writing the vectors as row vectors the rotation matrix which is a 3×3 must be applied via right multiplication. Note that most standard rotation matrices are left multiplied to a column vector.

$$p = [p_x \quad p_y \quad p_z] \tag{4}$$

$$p' = p\mathbf{R} \tag{5}$$

$$\mathrm{R} = \begin{bmatrix} 1 - 2(q_j^2 + qk^2) & 2(q_i q_j + q_k q_r) & 2(q_i q_k - q_j q_r) \\ 2(q_i q_j - q_k q_r) & 1 - 2(qi^2 + qk^2) & (2(q_j q_k + q_i q_r) \\ 2(q_i q_k + q_j q_r) & 2(q_j q_k - q_i q_r) & 1 - 2(qi^2 + qj^2) \end{bmatrix} \tag{6}$$

IV. Transformation Matrix

A translation described by a row vector can be added to the rotation matrix to describe a new transformed point. We will denote the new transformed vector p' which also combines translation.

$$p' = p\mathbf{R} + t \tag{7}$$

However, if an augmented vector $[p, 1]$ is used in the transform we can write a new matrix $\mathbf{T}$ when multiplied to $[p, 1]$ generates p'.

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} \\ t \end{bmatrix} \tag{8}$$

We can check that matrix multiplication is still well defined since $[p, 1]$ is 1×4 and $\mathbf{T}$ is 4×3.

$$p' = [p, 1]\mathbf{T} \tag{9}$$

The reason that this notation is used is that we can augment multiple rows each containing a vector p_i and apply the same transformation $\mathbf{T}$ in a single matrix multiplication.

$$\mathbf{P} = \begin{bmatrix} p_i & 1 \\ \vdots & \vdots \\ p_n & 1 \end{bmatrix} \tag{10}$$

$$\mathbf{P'} = \begin{bmatrix} p'_i \\ \vdots \\ p'_n \end{bmatrix} \tag{11}$$

$$\mathbf{P'} = \mathbf{PT} \tag{12}$$

To check the dimensionality we see that $\mathbf{P}$ is a $n \times 4$ matrix and $\mathbf{T}$ is a 4×3 matrix. Therefore, $\mathbf{P'}$ is $n \times 3$ as expected.

Laser Kinematics Derivation

Owen Lu
Electroimpact Inc.
owenl@electroimpact.com

***Abstract*—This document describes the method and assumptions used obtain the kinematic equations for the laser projector system. Notably, coordinates are taken within the projector frame.**

I. INTRODUCTION

This project has been undertaken in order to confirm an understanding of the laser project system and eventually develop better solving algorithms for the laser scanners. It has been historically noted that registration order of calibration points can create performance variations. This is not ideal as the input data set should result in one solution describing the location and orientation of the system.

II. COORDINATE FRAME DEFINTION

The construction of the laser projector allows good choices for the axes to be made in order to simplify calculations. The two mirror axes are assumed orthogonal to one another from one another.

We select the first stage mirror to have rotational axis along the Z-axis while the second stage mirror has axis along the X-axis. Thus, the first stage mirror rotates about the origin.

III. PLANE GEOMETRY

The construction of the system is such that an easy choice of a coordinate frame can be made.

Two rotational axes v_1, v_2 of the mirrors are orthogonal, to one another. The laser is also parallel to v_2.

Thus, we can select laser vector u to be along the x-axis

$$u = (1 \quad 0 \quad 0)^T \tag{1}$$

As a simplification the first plane is assumed to have no thickness, normal n_1 and rotates about the line L_1 which is parallel to the z-axis.

$$L_1\colon \begin{pmatrix} x_1 \\ y_1 \\ 0 \end{pmatrix} + \lambda_1 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = x_1 + \lambda_1 e_3 \tag{2}$$

In Figure 1 this virtual plane represents a mirror plane of zero thickness and will be corrected for later in the model. The normal to the plane can be written in terms of θ_1.

$$n_1 = \begin{pmatrix} -\sin(\theta_1) \\ \cos(\theta_1) \\ 0 \end{pmatrix} \tag{3}$$

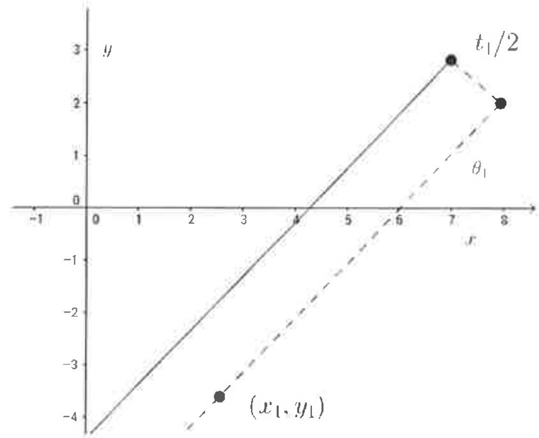

Figure 1 – Defined plane geometry in xy plane view

Therefore the mirror plane can be defined by a dot product expression.

$$x - x_1 \quad \cdot n_1 = 0 \tag{4}$$

Now that the expression for the zero thickness plane is developed, an offset plane can also easily formulated. This offset plane will be shifted by half the mirror thickness t_1.

This can be done easily by substituting x_1 for $x_1 + \frac{t_1}{2} n_1$. Distributive law can then be applied to the dot product.

$$x \cdot n_1 = x_1 \cdot n_1 + \frac{t_1}{2} \tag{5}$$

Similarly, the same method can be applied to describe the second mirror plane, with normal n_1 rotating about L_2 parallel to the x-axis. The diagram would be exactly the same as Figure 1 but with z replacing the ordinate and y replacing the abscissa.

$$L_2\colon \begin{pmatrix} 0 \\ y_2 \\ z_2 \end{pmatrix} + \lambda_2 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = x_2 + \lambda_2 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{6}$$

$$n_2 = \begin{pmatrix} 0 \\ -\sin(\theta_2) \\ \cos(\theta_2) \end{pmatrix} \tag{7}$$

$$x \cdot n_2 = x_2 \cdot n_2 + \frac{t_2}{2} \tag{8}$$

IV. RAY REFLECTION TRANSFORMATION

In order to find the ray emanating from the second mirror surface, the laser vector u must undergo two reflection transformations.

To find reflected vector r to input vector d reflected across plane of unit normal n, we can use the below equation.

$$r = d - 2(d \cdot n)n \tag{9}$$

Let u' denote the ray vector after the first planar reflection

$$u' = u - 2\ u \cdot n_1\ n_1 \tag{10}$$

Let u'' denote the ray vector after the second planar reflection

$$u'' = u' - 2\ u' \cdot n_2\ n_2 \tag{11}$$

At this point since u'' can be calculated, we only need point p the point where the laser intersects the second plane.

V. Finding Laser-Mirror Intersection

Since the laser originates on the x-axis we can generate a line R_1.

$$R_1 : s_1 u \tag{12}$$

Substitution of $x = s_1 u$ allows us to solve s_1

$$s_1 = x_1 - y_1 \cot\ \theta_1\ + \frac{t_1}{2 \sin\ \theta_1} \tag{13}$$

Therefore, the point p' which is the intersection of the laser and the first mirror plane is given below.

$$p' = \begin{pmatrix} x_1 - y_1 \cot\ \theta_1\ - \dfrac{t_1}{2 \sin\ \theta_1} \\ 0 \\ 0 \end{pmatrix} \tag{14}$$

Then the line used to find the intersection of the laser line and the second mirror is using line R_2 and the second mirror plane.

$$R_2 : p' + s_2 u' \tag{15}$$

Substitute $x = p' + s_2 u'$ and solve for s_2

$$s_2 = \frac{y_2 - z_2 \cot\ \theta_2\ - t_2/2}{\sin\ 2\theta_1} \tag{16}$$

$$s_2 u' = \begin{pmatrix} \left(y_2 - z_2 \cot\ \theta_2\ - \dfrac{t_2}{2 \sin\ \theta_2}\right) \cot\ 2\theta_1 \\ \left(y_2 - z_2 \cot\ \theta_2\ - \dfrac{t_2}{2 \sin\ \theta_2}\right) \\ 0 \end{pmatrix} \tag{17}$$

$$p'' = p' + s_2 u' \tag{18}$$

The last point on the $z = k$ plane can be solved by using line R_3.

$$R_3 : p'' + s_3 u'' \tag{19}$$

Solving for s_3 such that $z = k$ allows for the final point to be found.

Since p'' has no z components, we simply use the equation

$$s_3 = \frac{k}{\sin\ 2\theta_1\ \sin\ 2\theta_2} \tag{20}$$

$$s_3 u'' = k \begin{pmatrix} \cot\ 2\theta_1\ / \sin\ 2\theta_2 \\ \cot\ 2\theta_2 \\ 1 \end{pmatrix} \tag{21}$$

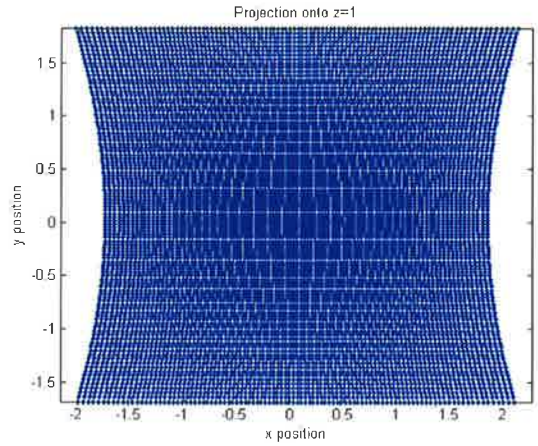

Figure 2 – Projection points using $30 < \theta_1, \theta_2 < 60$

Therefore, the final equation describing a projection on plane $z = k$ can be written below:

$$P = p' + s_2 u' + s_3 u'' \tag{22}$$

VI. Verification

Using Solidworks 3D sketches and work planes, the modelled system can be constructed to verify the calculations. Calculations using parameters matched by the Solidworks model have been verified to be exact matches.

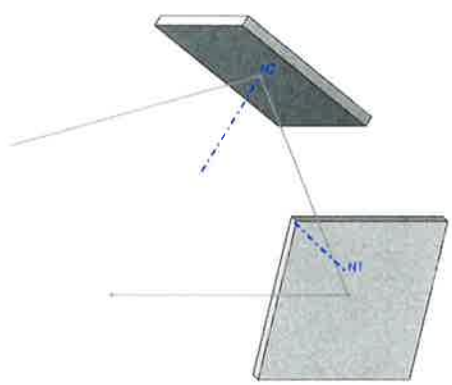

Figure 3 – Ray reflection solid model for verification

Optimization method to solving relative rotation and translation of galvonometer systems

Owen Lu
Electroimpact Inc.
owenl@electroimpact.com

***Abstract*—This document describes one method to solving for the rotation and translation of the galvanometer axes relative to a reference coordinate frame. Optimization methods are suggested due to the current algorithm returning different results based on the registration order of calibration points.**

I. INTRODUCTION

At the core of finding the rotation and translation of the laser reference frame in respect to the absolute reference frame, ray equations must be solved. Loosely speaking, the simplified problem has a set of 4 or more angle pairs θ_1, θ_2 describing mirror angles with corresponding calibration points x in 3 space for which a laser passes through. The ray, described by a mathematical line corresponding θ_1, θ_2 should then pass through all calibration points if no error exists.

In experience, the laser projectors solving method gives results that depend greatly on the order in which points are registered. This is not ideal since consistent results are unable to be realized. Furthermore, there should be a "best" solution that gives the least error possible, and the operator should not need to permute calibration points to find this best solution.

II. QUARTERNION ROTATION

Quaternion representation of rotation is well understood and resources can be found in many places online. However, a brief introduction here will be done for convenience and notation purposes.

Suppose that we wish to generate rotation of a vector around $\bar{u}$ by angle θ. This can be done using a quaternion q and generating a rotation matrix $\boldsymbol{R}$.

$$q = \left[\cos\left(\frac{\theta}{2}\right), u_x \sin\left(\frac{\theta}{2}\right), u_y \sin\left(\frac{\theta}{2}\right), y_z \sin\left(\frac{\theta}{2}\right)\right]'$$

$$q = [q_r, q_i, q_j, q_k]'$$

Where

$$\boldsymbol{R}(q) = \begin{bmatrix} 1-2q_j^2-2q_k^2 & 2(q_iq_j - q_kq_r) & 2(q_iq_k + q_jq_r) \\ 2(q_iq_j + q_kq_r) & 1-2q_i^2-2q_k^2 & 2(q_jq_k - q_iq_r) \\ 2(q_iq_k - q_jq_r) & 2(q_jq_k + q_iq_r) & 1-2q_i^2-2q_j^2 \end{bmatrix}$$

Then applying the rotation to general vector $\bar{v}$ to formulate rotated vector $\bar{v}'$ can be done with the equation following.

$$\bar{v}' = \boldsymbol{R}(q)\bar{v}$$

III. RAY FORMULATION

In a previous document ray equations in the galvanometer frame were derived depending on mirror angles. The results are reused below without derivation for convenience.

$$\bar{p}' = \begin{pmatrix} x_1 - y_1\cot(\theta_1) - \dfrac{t_1}{2\sin(\theta_1)} \\ 0 \\ 0 \end{pmatrix}$$

$$\bar{p}'' = \bar{p}' + \begin{pmatrix} \left(y_2 - z_2\cot(\theta_2) - \dfrac{t_2}{2\sin(\theta_2)}\right)\cot(2\theta_1) \\ y_2 - z_2\cot(\theta_2) - \dfrac{t_2}{2\sin(\theta_2)} \\ 0 \end{pmatrix}$$

$$\bar{u}'' = \begin{pmatrix} \cos(2\theta_1) \\ \sin(2\theta_1)\cos(2\theta_2) \\ \sin(2\theta_1)\sin(2\theta_2) \end{pmatrix}$$

The ray in galvanometer coordinates, defined as frame B is given below.

$$r_{i_B} = \bar{p}''(\theta_1, \theta_2) + \lambda\bar{u}''(\theta_1, \theta_2)$$

IV. RAY TRANSFORMATION

The ray equations must be converted to equations in the reference frame since calibration points are known in said frame.

Let L be the translation of B to A.

$$\bar{l} = [l_1, l_2, l_3]'$$

Let q be the quaternion describing the rotation of B to A

$$q = [q_r, q_i, q_j, q_k]'$$

Suppose a line r is defined in the equation below in the galvanometer frame B.

$$r_B\colon \bar{p}'' + \lambda\bar{u}'' \tag{1}$$

If we wish to find the same line equation within reference frame A we must know the quaternion q and translation $\bar{l}$ that when applied to A creates an equivalent frame with B.

At this point the line becomes a straight forward calculation.

$$r_A\colon \boldsymbol{R}(q)\bar{p}'' + \bar{l} + \lambda\boldsymbol{R}(q)\bar{u}''$$

Let

$$\boldsymbol{R}(q)\bar{u}'' = \bar{u}_A$$

$$\bar{x}_{0_A} = \boldsymbol{R}(q)\bar{p}'' + \bar{l}$$

By substitution

$$r_A = \bar{x}_{0_A} + \lambda \bar{v}_A$$

V. OBJECTIVE FORMULATION

Ideally the objective has a physical meaning so that the user can easily verify whether the optimization algorithm has converged to a reasonable solution. For this reason, the average distance from the ray to a calibration point is used.

The perpendicular distance between some point $\bar{x}_i$ and line r_i can be given by the equation below.

$$dist(r_i, x_i) = norm\left(\left(\bar{x}_i - \bar{x}_{0_A}\right) - \left(\left(\bar{x}_i - \bar{x}_{0_A}\right) \cdot \bar{v}_A \right) \bar{v}_A \right)$$

The average distance to a calibration M is given below.

$$M = \frac{1}{n} \sum_{i=1}^{n} dist\left(r_{i_A}\left(q, \bar{l}\right), \bar{x}_i\right)$$

Under constraint

$$\|q\| = 1$$

VI. NUMERICAL TESTING

To test whether this method can converge upon reasonable results, a sample data set was used assuming the following parameters.

$$k = 1$$

$$t_1 = 5mm$$

$$t_2 = 5mm$$

$$x_1 = 0$$

$$y_1 = 0$$

$$y_2 = 40mm$$

$$z_2 = 0$$

$\theta_1\ (deg)$	$\theta_2\ (deg)$	$x(mm)$	$y(mm)$	$z(mm)$
45	55	-3.53	-327.02	1000
45	65	-3.53	-801.85	1000
55	55	-403.82	-327.02	1000
55	65	-491.73	-801.85	1000
65	55	-926.71	-327.02	1000
65	65	-1129.37	-801.85	1000

The optimization function used was *fmincon* in MATLAB. Random seed values were used with to generate quaternions rotations within 10 degrees of the correct solution, while errors of 0.5 metres was used for the translational error.

A plot of the average error as a function of the seed angular error and the seed displacement error is given following.

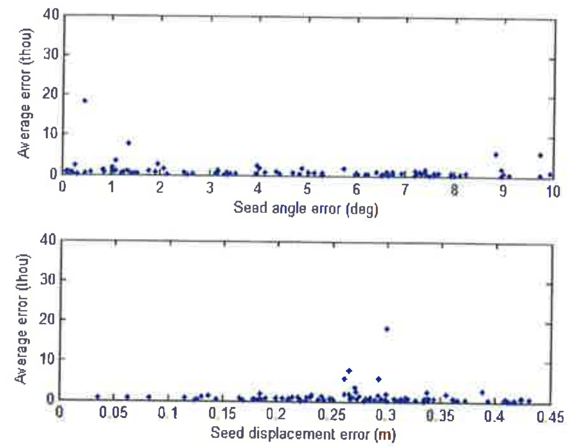

Figure 1 – Average distance error vs. Seed angle error (top); Average distance error vs. Seed displacement error (bottom

From the results, the error is small, under 20 thou if the initial seed can be estimated to reasonable accuracy. More tests are necessary, but initial feasibility has been validated.

*Note: A key setting in the MATLAB function is that TolFun = 0.000001.

Method of Virtual Image Planes for Dual Axis Galvanometer Pose Estimation

Owen Lu
Electroimpact Inc.
owenl@electroimpact.com

***Abstract*— A fundamental problem in computer vision is camera calibration and pose estimation. Due to the importance of the field of study, pose estimation is now a well understood topic and thus we look to leverage the same algorithms by converting the projector into a pinhole camera. After the conversion, the exact same techniques can be used to generate initial estimates of pose, shortcutting years of theoretical development0. A model based refinement technique can then be used to increase the accuracy.**

I. INTRODUCTION

Central to the model of the Dual-axis Galvanometer Model (DGM) is that rays are generated by mirror reflection about two planes in space. In principle, the output ray, defined by a point and unit vector, can be calculated in different mirror configurations. A simple form of the DGM assumes that rays emanate from a single point. This creates a situation that is identical to the pinhole camera. Since camera pixel coordinates are generated by projection to a plane, all that is left to do is to calculate the intersection of generated rays and a plane to obtain virtual pixels.

Electroimpact DGM EPnP implementation

1. Nominal projector model is generated
2. DAC xy pairs corresponding to XYZ coordinates are simulated in the model to generate rays
3. Rays are projected onto plane $Z = 1$ in a pinhole camera coordinate system with $A = I_3$
4. Projected xy coordinates are assumed to be pixels
5. EPnP algorithm is used in combination with Gauss-Newton (GN) Optimization
6. Rotation matrix R and translation vector t are obtained and converted back into AV coordinate system.

The second step to the method is a DGM reconstruction of all measured points combined with a Singular Value Decomposition (SVD) based rigid transform solver.

Electroimpact DGM Rigid Transform Refinement

1. Using transform from EPnP based estimation, points are brought into projector space
2. DAC xy pairs generate rays corresponding to XYZ positions
3. Calculate the nearest point on the ray to the XYZ coordinate point, or the point estimated by distance
4. Solve the rigid transform problem
5. Update the transformed points
6. Loop 4-5 until a mean error tolerance is achieved
7. Return R, t

A side note is the Efficient Perspective-n-Point Pose Estimation (EPnP) is a method that requires a calibrated camera model in order to work and can generate transforms without necessarily using points assumed to be coplanar. This fact is extremely important AV's use case, since calibration verification is done in plane, while registration in tool coordinates can have arbitrary arrangements and an arbitrary number of targets. Due to the convention used to generate the virtual image plane, assuming a plane on $Z = 1$, the camera intrinsic matrix, which is defined as a 3 by 3 matrix is the identity matrix.

II. EPNP VIA GAUSS-NEWTON

The algorithm is broken down into 8 detailed steps.

1. Define control points in world coordinates for the homogeneous barycentric system
2. Calculate the $\alpha_{j\in[1,2,3,4]}$ weights based on the control points for each n data points
3. For each of the n data points, generate 2 constraints on the control points described in the camera coordinates. Concatenate all equations and form M such that $\mathrm{M}x = 0$ where x is a 12 vector containing the 4 control points and their components within the camera system.
4. Solve for the right singular vectors of M to describe the general solution space of x.
5. Use Gauss-Newton optimization method to generate the best linear combination x_1, x_2, x_3, x_4 which are the right singular vectors associated with the 4 smallest singular values of M. Let x be this best linear combination from a least squares optimization.
6. Recover the control points in camera space from x.
7. Use control points in camera space and the previously calculated weights in step 2 to find all n points in camera space.
8. Solve the rigid body rotation problem R, t which defines the pose of the camera.

In general, since the model of the system is a linear model we attempt to describe as much of the model as possible as matrix operations.

First, the control points which are arbitrary are selected. Note they must be non-coplanar. We store these inside the 4×3 matrix C_w describing the control points in the world frame.

1. Define the control points

We simply pick the orthogonal basis vectors of the rectangular world system and add the origin. The choice is arbitrary, but this choice is well conditioned which results in numerical stability.

$$\mathbf{C}^w = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \\ \mathbf{c}_4 \end{bmatrix}$$

2. Calculate the weights for each point

Let $\mathbf{X}^w$ be the points in the world coordinate system where the ith in $\mathbf{X}^w$ is a 1×3 vector describing the x, y, z coordinates.

To solve the weights of the barycentric system we can augment the matrices $\mathbf{C}^{wT}$ and $\mathbf{X}^T$ by adding a row of ones.

Let:
The augmented $\mathbf{C}^{wT}$ be $\mathbf{C}_\star^{wT}$
The augmented $\mathbf{X}^{wT}$ be $\mathbf{X}_\star^{wT}$

$$\mathbf{C}_\star^{wT} \alpha = \mathbf{X}_\star^{wT}$$

Where α is a $4 \times n$ matrix where each column contains a vector of the control points weights for the ith point.

$$\alpha = \begin{bmatrix} \alpha_{1_i} & \cdots & \alpha_{1_n} \\ \alpha_{2_i} & \cdots & \alpha_{2_n} \\ \alpha_{3_i} & \cdots & \alpha_{3_n} \\ \alpha_{4_i} & \cdots & \alpha_{4_n} \end{bmatrix}$$

Since $\mathbf{C}_\star^{wT}$ is square we can simply invert it.

$$\alpha = inv(\mathbf{C}_\star^{wT})\mathbf{X}_\star^T$$

At this point, α has been calculated.

3. Generate the constraint matrix

Matrix M is derived from concatenating 2 constraint equations from each of the n points. We assume that for the ith point

$$s_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{A} \sum_{j=1}^{4} \alpha_{j_i} \mathbf{c}_j^T$$

It is assumed that matrix $\mathbf{A}$ describing the camera is skew-less.

$$\mathbf{A} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

This gives rise to two constraints, formed from the u_i, v_i pixel equations. This means that

$$\sum_{j=1}^{4} \alpha_{j_i} f_u x_j + \alpha_{j_i} \; u_0 - u_i \; z_j = 0$$

$$\sum_{j=1}^{4} \alpha_{j_i} f_v y_j + \alpha_{j_i} \; v_0 - v_i \; z_j = 0$$

Where $c_j = [x_j, y_j, z_j]$

The resultant vector is a $2n \times 12$ matrix which is solved for its right singular values. This is the same as solving for x in the following equation.

$$\mathrm{M}x = 0$$

4. Solve for the right singular values of M

In practice, finding the 4 eigenvectors of $\mathrm{M}^T\mathrm{M}$ that are associated with the smallest eigenvalues is more numerically stable than finding the null space of M since noise can make it such that the null space is empty for example.

The maximum number of null vectors is 4 which is the reason we find the 4 eigenvectors associated with the smallest eigenvalues. This basically means that the vectors are sent very close to the origin. Roughly satisfying our above equation.

At this point, we have four vectors x_1, x_2, x_3, x_4.

Assume that the corresponding eigenvalues have the following relationship: $\lambda_1 < \lambda_2 < \lambda_3 < \lambda_4$.

5. Gauss-Newton algorithm to linear combinations

Therefore, x_1 is the best single null vector approximation, to $\mathrm{M}x = 0$. It follows that x_2 is the second best approximation and so forth.

Thus, it makes sense to attempt to create a solution x which is a linear combination of $x_{i \in \{1,2,3,4\}}$.

We wish to solve the weights β_i that gives x such that the norms between corresponding points in world space and camera space are equal. This creates an equivalent scale between the two coordinate systems and also refines their directions.

In practice, it is helpful to use the 2-norm squared and compare the values since the expressions are simpler.

For any pair of points in camera space, the norms squared should be equal to that of the pair of points in world space.

$$\|p_a^c - p_b^c\|^2 = \|p_a^w - p_b^w\|^2$$

Since we have defined every point as a weighted sum of control points, there are only 4 points that we need to consider. Since there are only 6 ways to pair 2 points within a set of 4, we have only 6 equations to be summed for overall squared error.

We define each the 6 differences as such

Let

$$\begin{aligned}
\delta_1^w &= \mathbf{c}_1 - \mathbf{c}_2 \\
\delta_2^w &= \mathbf{c}_1 - \mathbf{c}_3 \\
\delta_3^w &= \mathbf{c}_1 - \mathbf{c}_4 \\
\delta_4^w &= \mathbf{c}_2 - \mathbf{c}_3 \\
\delta_5^w &= \mathbf{c}_2 - \mathbf{c}_4 \\
\delta_6^w &= \mathbf{c}_3 - \mathbf{c}_4
\end{aligned}$$

$$\sigma^2 = \sum_i^6 \left(\|\beta_1 \delta_{i\,1}^c + \beta_2 \delta_{i\,2}^c + \beta_3 \delta_{i\,3}^c + \beta_4 \delta_{i\,4}^c\|^2 - \|\delta_i^w\|^2\right)^2$$

We notice that squaring the norm squared yields quartic equations, which creates negative solutions. For example, if β was the vector of weights that minimized σ^2, then $-\beta$ would also be a solution to the minimum.

In practice, this means that the rotation and translation are negative, which results in the points within the camera system having a negative z coordinate. This is impossible due to camera coordinate conventions, therefore, the points are then multiplied by (-1) and the proper transform is found. This problem is avoided if the 2-norm squared residuals are used instead of the 2-norm squared-squared residuals. However, the derivative computation is elegantly expressed using the σ^2 expression from earlier which is helpful in using Gauss-Newton methods.

The problem is then state below:

Minimize σ^2 with respect to $(\beta_1, \beta_2, \beta_3, \beta_4)$

Since Gauss-Newton requires the derivatives of the residuals to be computed for each data point i

Let

$$r_i = \|\beta_1 \delta_{i\,1}^c + \beta_2 \delta_{i\,2}^c + \beta_3 \delta_{i\,3}^c + \beta_4 \delta_{i\,4}^c\|^2 - \|\delta_i^w\|^2$$

We formulate the Jacobian, which holds the derivative of r_i with respect to $\beta_{i \in [1,2,3,4]}$ in the columns, evaluated at (β) in its ith row. β is the vector containing all β_i. This is basically a derivative matrix calculated for each data point. Note that superscript s denotes the iteration count. This means that we must initialize β^0.

Let

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix}$$

$$r = \begin{bmatrix} r_1 \\ \vdots \\ r_6 \end{bmatrix}$$

$$\Delta_{ij} = \delta_i \cdot \delta_j$$

$$\Delta = [\Delta_1, \Delta_2, \Delta_3, \Delta_4]$$

$$\mathbf{J}_{ij} = \frac{\partial r_i(\beta^s)}{\partial \beta_j} = 2\beta^T \Delta_j$$

$$\beta^{s+1} = \beta^s - inv\ \mathbf{J}^T\mathbf{J}\ \mathbf{J}^T r(\beta^s)$$

This results in a general method to solve for β regardless of the number of approximate singular vectors x. For example, if we are only considering using vector x_1 to solve the system. Then β_1 is initialized and $\beta_{i \in [2,3,4]} = 0$.

Calculating the linear combination to find x is then the linear combination with the β values

$$x = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$$

Then the transformed control points in camera space can be calculated.

6. *Recover the camera control points*

Let

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_{12} \end{bmatrix}$$

Then the transformed control points can be recovered in camera space.

$$\begin{aligned}
\mathbf{c}_1^c &= [x_1, x_2, x_3] \\
\mathbf{c}_2^c &= [x_4, x_5, x_6] \\
\mathbf{c}_3^c &= [x_7, x_8, x_9] \\
\mathbf{c}_4^c &= [x_{10}, x_{11}, x_{12}]
\end{aligned}$$

$$\mathbf{C}^c = \begin{bmatrix} \mathbf{c}_1^c \\ \mathbf{c}_2^c \\ \mathbf{c}_3^c \\ \mathbf{c}_4^c \end{bmatrix}$$

7. *Calculate the coordinates in camera space*

At this point since the weights in α can be used to calculate all the points $\mathbf{X}^c$

$$\mathbf{X}^c = \alpha \mathbf{C}^c$$

Although scales should be very close at this point, we compute the scale factor to equate the norms again.

Let

$$d_i^c = \|\mathbf{X}_i^c\|$$

$$d_i^w = \|\mathbf{X}_i^w\|$$

Then we compute λ such that the squared error in norms r^2

$$r^2 = \sum_{i=1}^{n} \lambda d_i^c - d_i^w \;^2$$

Taking the derivative of r^2 with respect to lambda

$$\frac{dr^2}{d\lambda} = \sum_{i=1}^{n} 2 \;\lambda d_i^c - d_i^w \; d_i^c$$

Setting the derivative to zero allows us to express λ in closed form by using dot products, or equivalently, inner products.

$$\lambda = \frac{d^w \cdot d^c}{d^c \cdot d^c} = \frac{d^{w\,T} d^c}{d^{c\,T} d^c}$$

At this point, the scales are the closest possible using a single scaling factor with respect to a squared residual error.

At this point, we check if the z values contained in $\mathbf{X}^c$ are negative. If they are, then they are all negative, so we multiply $\mathbf{X}^c$ by (-1) to get the correct points.

8. *Solve the rigid body transformation problem*

Now we have two matrices with the 3D points in camera space and world space. The last step is to solve the rigid rotation problem. Start by calculating the centroids.

$$x^c = mean \;\mathbf{X}^c$$
$$x^w = mean(\mathbf{X}^w)$$

The by abuse of notation matrix $\mathbf{X}$ subtracted by the centroid x denotes a point by point subtraction.

$$\delta\mathbf{X}^c = \mathbf{X}^c - x^c$$

$$\delta\mathbf{X}^w = \mathbf{X}^w - x^w$$

Let

$$\mathbf{S} = \delta\mathbf{X}^w diag \;1, \dots, 1_n \;\delta\mathbf{X}^{c\,T}$$

Compute the Singular Value Decomposition (SVD) of S

$$\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

Finally, the rotation and translation of the camera with respect to the reference coordinate frame is calculated.

$$\mathbf{R} = \mathbf{V} \begin{pmatrix} 1 & & \\ & \ddots & \\ & & \det(\mathbf{V}\mathbf{U}^T) \end{pmatrix} \mathbf{U}^T$$

$$\mathbf{t} = x^c - \mathbf{R}x^w$$

In practice, the process is completed 4 times to solve 4 different transforms $\mathbf{R}, \mathbf{t}$ using different numbers of null vectors. For example, we compute the linear weights β for 1 vector, 2 vectors, 3 vectors, and finally 4 vectors. We always keep the smallest null vectors first, and add the next smallest to compute the transformed control points. Finally, we compare the reprojection error of all 4 sets in order to pick out the best transform.

III. RIGID TRANSFORM REFINEMENT

In simplified versions of the DGM projection rays emanate from the origin in AV conventions. This is the same as a pinhole camera model. However, in pinhole camera models the Z-axis is along the direction of projection. Overall the AV coordinate convention is rotated 180° from the standard intrinsic camera coordinate convention which makes coordinate conversion easy.

Figure 1 is shown to explain the idea of ray projection onto an image plane, creating the virtual camera pixels. The only difference is that the rays are not centered at the origin in the general DGM.

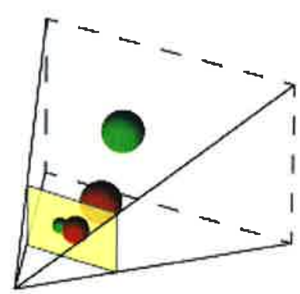

Figure 1 - XYZ projection on to the image plane

The equation which must be solved for each point

$$x(x_d, y_d) + \lambda u(x_d, y_d) = \overline{X}$$

Since u is a unit vector

$$\lambda = \|\overline{X} - x\|$$

Assuming that points in the projector space $\overline{X}_p$ undergo a rigid transform from tool coordinates allows us to substitute.

$$x(x_d, y_d) + \lambda u(x_d, y_d) = \mathbf{R}\overline{X}_w + t$$

This equation should hold true for all registration points. The method to refine transform estimate $\mathbf{R}, t$ is an iterative method that uses a model estimate to estimate the left side, to generate a point could which can be solved exactly for $\mathbf{R}, t$.

Assume that $\mathbf{R}_0, \overline{t_0}$ is estimated the by the EPnP method. Then the ith registration point is below.

$$\overline{X}_0^i = \mathbf{R}_0\overline{X}_w^i + t_0$$

Estimate λ^i

$$\lambda^i = \|\mathbf{R}_0\overline{X}_w^i + t_0 - x^i\|$$

Calculate the estimated point in projector space

$$x^i + \lambda^i u^i = Y^i$$

Solve the rigid rotation problem that minimizes the least squares distances. This is a problem with a known solution via SVD and also has known solutions for linear weights.

$$\sigma^2(\mathbf{R}_k, t_k) = \sum_{i=1}^{m}\|\mathbf{R}_k\overline{X}_w^i + t_k - Y_{k-1}^i\|$$

Adjust the points in projector space.

$$\overline{X}_{k+1}^i = \mathbf{R}_k\overline{X}_w^i + t_k$$

Recalculate λ^i for all points and loop until a tolerance ϵ is reached on the vector norm of the mean error.

$$\delta x = \left\|\frac{1}{m}\sum_{i=1}^{m}\mathbf{R}_k\overline{X}_w^i + t_k - Y_{k-1}^i\right\| < \epsilon$$

Note that this method estimates pose more accurately on calibrated projectors, although nominal parameters can still generate estimates.

Explicit method to solving angle calibration functions of dual mirror laser galvonometer

Owen Lu
Electroimpact Inc.

owenl@electroimpact.com

***Abstract*— It was thought that galvanometer angles within AGS laser projects were known by measurement however, it has been revealed that this is not the case. Thus, although the galvanometer relationship between digital command rotation is approximately linear, the angle must be inferred via calculation and rays generated using a mathematical model. This document outlines how such calibration functions can be inferred in the absence of measurement.**

I. Introduction

The problem, which has yet to be proven in a mathematical sense, is summarized in this basic statement:

Knowing the projector geometry you can solve for the angle pair that generates a ray r through some point x_0. Knowing the angle pair that generates r through some point x_0 you can solve for the geometry of the projector. Knowing neither the geometry nor the angle pair, there are infinite solutions for both the angle pair and projector geometry that produces r that passes through x_0.

However, what this statement does not include is the assumption that the DAC angles can be converted to angles perfectly through a linear relationship. This is the most important overarching assumption in this model.

II. Forward kinematic assumption

Previously, laser kinematic equations have been derived to describe the ray output of the second mirror. Thus, assuming some projector geometry the ray is a function of only two variables (θ_1, θ_2). The geometry is the exact geometry used as reference in the projector design drawings and is referred to as nominal dimensions.

The ray r is described with the equations below.

$$\bar{p}' = \begin{pmatrix} x_1 - y_1 \cot(\theta_1) - \dfrac{t_1}{2\sin(\theta_1)} \\ 0 \\ 0 \end{pmatrix} \tag{1}$$

$$\bar{p}'' = \bar{p}' + \begin{pmatrix} \left(y_2 - z_2 \cot(\theta_2) - \dfrac{t_2}{2\sin(\theta_2)}\right) \cot(2\theta_1) \\ y_2 - z_2 \cot(\theta_2) - \dfrac{t_2}{2\sin(\theta_2)} \\ 0 \end{pmatrix} \tag{2}$$

$$\bar{u}'' = \begin{pmatrix} \cos(2\theta_1) \\ \sin(2\theta_1)\cos(2\theta_2) \\ \sin(2\theta_1)\sin(2\theta_2) \end{pmatrix} \tag{3}$$

The ray in galvanometer coordinates, defined as frame B is given below.

$$r = \bar{p}''(\theta_1, \theta_2) + \lambda \bar{u}''(\theta_1, \theta_2) \tag{4}$$

Thus, solving the system of 3 equations below will generate the inferred angle pair, with corresponding ray length λ.

$$\bar{p}''(\theta_1, \theta_2) + \lambda \bar{u}''(\theta_1, \theta_2) - x_0 = 0 \tag{5}$$

III. Linearity assumption

In order to convert the DAC readings to mirror angles a simple linear formula is assumed where mirror angle.

$$\theta_1 = k_1 DAC_1 + b_1 \tag{6}$$

$$\theta_2 = k_2 DAC_2 + b_2 \tag{7}$$

Constants k_1, b_1, k_2, b_2 are all constructed by solving the system for (θ_1, θ_2) and curve fitting the relationship vs. DAC readings.

IV. Error calculation

The error in the angles is calculated with the formula below.

$$\delta\theta_1 = \theta_{1_i} - k_1 DAC_{1_i} - b_1 \tag{8}$$

$$\delta\theta_2 = \theta_{2_i} - k_2 DAC_{2_i} - b_2 \tag{9}$$

Ideally random error would be the cause for $\delta\theta$ disagreements. However, projector geometry is not perfect, and the main factor, which is mirror misalignment causes $\delta\theta_1$ and $\delta\theta_2$ to be functions of both DAC readings.

$$\delta\theta_1 = f(DAC_1, DAC_2) \tag{10}$$

$$\delta\theta_2 = g(DAC_1, DAC_2) \tag{11}$$

This phenomenon must be further studied as understanding these plots in relation to the mirror misalignment may yield vastly more accurate theoretical results. Possible changes in the experimental procedure may be necessary to determine such relationships.

V. Test results

AGS has provided 10ft calibration data along with their own algorithm error report. In general, the projector is calibrated at one distance and the error is reported as the RMS difference between the point in space the laser intersection on a corresponding z-plane. It has been discussed to instead use ray to point perpendicular distance instead of constraining error to be in-plane, as this is an arbitrary constraint that inflates the error.

However, for purpose of comparison, the same AGS error algorithm is used.

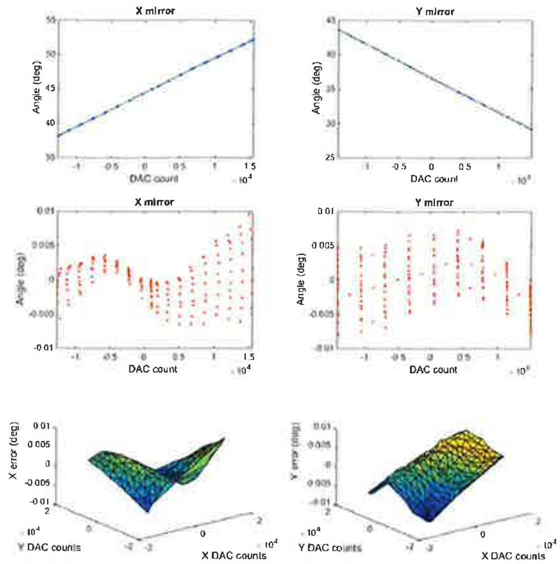

Figure 1 – (top) Linear curve fit to explicit solves;
(middle) Error between fit and solved angles
(bottom) Error between fit and solved angles vs. DAC readings

Using a LOESS quadratic fit in two variables, a fit was constructed to the error in order to correct the angles. The overall performance of the algorithm to predict the location of the laser ray outperformed the AGS algorithm by 37% for RMS error and 30% standard deviation.

VI. Conclusion

Currently the method to solve angles and use curve fits to correct the error can generate good results at set distances. However, this method noticeably reduces in accuracy when using the same projector at different distances. In theory, once the projector model parameters are defined, different distances should not affect the error plots. Furthermore, the error plots look smooth and have obvious grouping along meridian data. This implies that systematic error that is unmodelled must be added. More study in different methods and important model parameters to be added will be conducted to improve understanding of the system. The wish is to be able to construct a mathematical projector model that works well at multiple distances, resulting in more consistent performance.

Optimization method to solving angle calibration functions and projector geometry

Owen Lu
Electroimpact Inc.
owenl@electroimpact.com

***Abstract*— Using nominal projector dimensions and solving for mirror angles, a preliminary curve fit can be created to map both DAC angles to real angles. This allows optimization to begin near a feasible region. Ideally, all parameters that describe the geometry of the projector could be solved on in order to get the real dimensions of the projector. However, measurement error and redundancy in the effects of parameters on the objective function may prevent this from being possible. This document describes the main parameters that will be solved on to generate projector structures to be used for ray angle calculations.**

I. INTRODUCTION

In order to account for errors in construction a more general model is necessary to account for variations. Fortunately although the explicit equations to generate the output ray are very long expressions, the core of the problem is simple. There are two mirrors, described as planes in space, with a ray input that is reflected off both mirrors. It is known that any deviation from the mirror normal from the ideal orientation can greatly affect the ray output. Thus, the optimization method is used to solve on 8 scalar values. The first 4 describe the linear calibration constants to convert DAC to angles, the last 4 describe the initial configuration of the normal. All other dimensions that describe the projector are assumed to be nominal

II. UNIT VECTOR GENERATION

To generate unit vectors spherical coordinates are used. This allows the optimization to be done without constraints on vector length.

$$u_x = \sin(\phi)\cos(\theta) \tag{1}$$

$$u_y = \sin(\phi)\sin(\theta) \tag{2}$$

$$u_z = \cos(\phi) \tag{3}$$

Therefore, two angles can describe any unit vector in 3 space.

III. RAY-PLANE INTERSECTION

Laser ray r is defined by a point and a vector for some interval of λ. For some interval $\lambda \in (a, b)$ the finite line describes the light ray in space.

$$r\colon \bar{x}_0 + \lambda\bar{v} \tag{4}$$

The mirror plane is defined also with a point and vector by the below equation.

$$m\colon (\bar{x} - \bar{p}_0)\cdot\bar{n} = 0 \tag{5}$$

The intersection between the line and the plane $\bar{y}$ is then given by (7).

$$s = \frac{\bar{p}_0\cdot\bar{n} - \bar{x}_0\cdot\bar{n}}{\bar{v}\cdot\bar{n}} \tag{6}$$

$$\bar{x}' = \bar{x}_0 + s\bar{v} \tag{7}$$

IV. RAY REFLECTION

The final equation to generate the ray output from a mirror is calculated using the ray orientation and the normal vector. The calculation is to reflect $\bar{v}$ across $\bar{n}$ shown in (8).

$$\bar{v}' = \bar{v} - 2(\bar{v}\cdot\bar{n})\bar{n} \tag{8}$$

V. SERIAL RAY GENERATION

Using a ray and a plane, the output ray can be generated. This process can repeated from multiple mirrors. In this case there are only two mirrors defined by $\bar{n}_1, \bar{p}_1, \bar{n}_2, \bar{p}_2$. If the ray input is defined by $\bar{x}_0, \bar{v}_0$ then the following calculations will generate the output of the projector.

The output of m_1, r_1 is defined by the ray-plane intersection and ray reflection equations.

$$\bar{x}_1 = \bar{x}'(\bar{x}_0, \bar{v}_0, \bar{p}_1, \bar{n}_1) \tag{9}$$

$$\bar{v}_1 = \bar{v}'(\bar{v}_0, \bar{n}_1) \tag{10}$$

$$r_1\colon \bar{x}_1 + \lambda\bar{v}_1 \tag{11}$$

Applying the same equations to r_1 using m_2 generates r_2 in (14).

$$\bar{x}_2 = \bar{x}'(\bar{x}_1, \bar{v}_1, \bar{p}_2, \bar{n}_2) \tag{12}$$

$$\bar{v}_2 = \bar{v}'(\bar{v}_1, \bar{n}_2) \tag{13}$$

$$r_2\colon \bar{x}_2 + \lambda\bar{v}_2 \tag{14}$$

VI. Objective Formulation

The objective used is the average perpendicular distance from the point in space to the corresponding ray generated by the project with corresponding angle pairs.

The perpendicular distance between $\bar{x}_i$ and line r_i is given by (15).

$$d = norm\big((\bar{x}_i - \bar{x}_0) - ((\bar{x}_i - \bar{x}_0)\cdot\bar{v})\bar{v}\big) \tag{15}$$

The objective is then the average of d over all data points. Currently, the first optimization algorithm takes into account only 8 variables describing the projector. Until dimensions can be verified, these will be the only ones used.

$$M = \frac{1}{n}\sum_{i=1}^{n} d(k_1, b_1, k_2, b_2, \phi_1, \theta_1, \phi_2, \theta_2) \tag{16}$$

VII. Initial conditions

Using an initial conditions generated from the "Explicit method to solving angle calibration functions to dual mirror galvonometers" the seed values for k_1, b_1, k_2, b_2

Description of Objective Functions

Owen Lu
Electroimpact Inc.
owenl@electroimpact.com

***Abstract*— From discussion with AGS, the accuracy of the inverse algorithm to compute DAC commands from 3D coordinates is of critical importance. Thus, ensuring that the objective is created properly to result in good inverse performance and precision is necessary. However, calculation speed is also an important consideration as implementations may become impractical if this important element is neglected. The methods in this paper are all in different regions of the speed-precision spectrum.**

I. Introduction

The underlying problem of generating an objective that uses errors in DAC commands upon inversion of coordinates is the speed. The methods that are described all have different accuracy and speed characteristics when used as an objective and are used in different stages in the solving process.

II. Ray-Distance Method

This was the first objective used when building the projector model. The perpendicular distance between a modeled ray and a point was used to quantify the model performance. The DAC inputs would be used to generate the ray, and the point is measured so the distance can be computed.

The main advantage of this technique is that computing the resultant ray from an input-ray and two reflections are extremely fast. However, using the resultant model to generate DAC commands was not consistent. The root of the problem results from the fact that the ray position and orientation cannot be measured via the current procedure. Thus, simply because a mathematical ray is close to a point, does not mean the inverse function generates very accurate DAC commands

$$r = p\ DAC_1, DAC_2\ + \lambda u\ DAC_1, DAC_2 \qquad \lambda \in (0, \infty) \tag{1}$$

III. Ray-Equation Method

In order to generate two DAC commands with XYZ coordinates, three equations with two unknowns must be solved. This is summarized by the vector equation in Eq. 2 which is equivalent to Eq. 3.

$$p + \|x - p\|u = x \tag{2}$$

$$f = p + \|x - p\|u - x = \mathbf{0} \tag{3}$$

Instead of varying DAC commands, model commands are varied in attempts to solve the equation. The computed vector norm of f during optimization measures how well the model fits the data. The number also represents the Euclidean distance between the model point and the measured point.

Thus, minimizing the sum of squared norms with respect to projector parameters results in better DAC agreement between the data and model.

$$\min \sum_i^n \|p_i + \|x_i - p_i\|u_i - x_i\|^2 \tag{4}$$

IV. DAC Error Method

This is the most direct method of optimization with respect to an important objective. This objective uses the projector model parameters to invert positions to DAC commands and compares the sum of squared DAC errors. Using a linearization of the projector model with respect to the DAC commands the over determined system can be solved.

Thus we solve the vector equation in Eq. 5.

$$f_i = p_i + \|x_i - p_i\|u_i - x_i = 0 \tag{5}$$

The solution is represented by Θ_i.

$$\Theta_i = (DAC_{1s_i}, DAC_{2s_i}) \tag{6}$$

Let the DAC commands from the calibration data be represented by Γ_i.

$$\Gamma_i = (DAC_{1_i}, DAC_{2_i}) \tag{7}$$

Thus, we select the parameters to solve Eq. 8.

$$\min \sum_i^n \|\Theta_i - \Gamma_i\|^2 \tag{8}$$

V. Conclusion

The three objectives all have the inherent tradeoff between accuracy and speed. The Ray-Distance method is the fastest and the least accurate, while the DAC Error Method is the slowest and most accurate. In practice, it is helpful to start with the lower accuracy methods and quickly switch to the higher accuracy methods. This results in good error performance overall and reasonable solve time.

Ray equation solver

Owen Lu
Electroimpact Inc.
owenl@electroimpact.com

***Abstract*— In order to generate DAC commands for the projector three equations are solved for two unknowns. In this implementation, Newton's method is used.**

I. Linearization

The projector model generates the point on the second mirror from which the ray exits the projector as well as the direction of the ray.

This gives rise to three equations when solved generate a theoretical intersection of the ray and the point. Below it is represented by a vector equation.

$$f(x_{dac}, y_{dac}) = x + \|x_i - x\|u - x_i = 0 \tag{1}$$

$$f = [f_1 \quad f_2 \quad f_3]^T \tag{2}$$

Both x and u are both functions of the projector geometry and the DAC angles. We wish to find the DAC commands that best satisfies Equation 1.

Linearizing with respect to the DAC angles allows us to calculate the solution very quickly. In practice, numerically differentiating using a forward difference with a step of 1 DAC count is sufficient. In short, we wish to find the derivative matrix $\boldsymbol{D}$ to calculate the solution of Equation 4.

$$\theta_{dac} = \left[x_{dac_0}, y_{dac_0}\right]^T \tag{3}$$

$$f(\theta_{dac_0}) + \boldsymbol{D}(\theta_{dac_0})(\theta_{dac_1} - \theta_{dac_0}) = 0 \tag{4}$$

$$\boldsymbol{D} = \begin{bmatrix} \frac{\partial f_1}{\partial x_{dac}} & \frac{\partial f_1}{\partial y_{dac}} \\ \frac{\partial f_2}{\partial x_{dac}} & \frac{\partial f_2}{\partial y_{dac}} \\ \frac{\partial f_2}{\partial x_{dac}} & \frac{\partial f_3}{\partial y_{dac}} \end{bmatrix} \tag{5}$$

II. Numerical calculation

To calculate the derivative matrix with respect to x_{dac}, y_{dac} we simplify evaluate the function f three times.

Function evaluations:

$$f(x_{dac_0}, y_{dac_0})$$
$$f(x_{dac_0} + 1, y_{dac_0})$$
$$f(x_{dac_0}, y_{dac_0} + 1)$$

$$D_x = f(x_{dac_0} + 1, y_{dac_0}) - f(x_{dac_0}, y_{dac_0}) \tag{6}$$

$$D_y = f(x_{dac_0}, y_{dac_0} + 1) - f(x_{dac_0}, y_{dac_0}) \tag{7}$$

$$\boldsymbol{D}(\theta_{dac_0}) \approx [D_x \quad D_y] \tag{8}$$

Then we simply solve the overdetermined system for the increment.

Let

$$\boldsymbol{D}(\theta_{dac_0}) \approx [D_x \quad D_y] \tag{9}$$

$$d\theta = (\theta_{dac_1} - \theta_{dac_0})$$

Solve

$$\boldsymbol{D}(\theta_{dac_0}) \approx [D_x \quad D_y] \tag{10}$$

$$\boldsymbol{D}(\theta_{dac_0})d\theta = -f(\theta_{dac_0})$$

Iterate

$$\boldsymbol{D}(\theta_{dac_0}) \approx [D_x \quad D_y] \tag{11}$$

$$\theta_1 = \theta_0 + d\theta$$

In practice, this takes 1 iteration to converge for the nominal projector model parameters within 1 DAC count. However, 2 or 3 iterations is recommended.

Optimization method to solving angle calibration functions and projector geometry

Owen Lu
Electroimpact Inc.
owenl@electroimpact.com

***Abstract*— Using nominal projector dimensions and solving for mirror angles, a preliminary curve fit can be created to map both DAC angles to real angles. This allows optimization to begin near a feasible region. Ideally, all parameters that describe the geometry of the projector could be solved on in order to get the real dimensions of the projector. However, measurement error and redundancy in the effects of parameters on the objective function may prevent this from being possible. This document describes the main parameters that will be solved on to generate projector structures to be used for ray angle calculations.**

I. INTRODUCTION

In order to account for errors in construction a more general model is necessary to account for variations. Fortunately although the explicit equations to generate the output ray are very long expressions, the core of the problem is simple. There are two mirrors, described as planes in space, with a ray input that is reflected off both mirrors. It is known that any deviation from the mirror normal from the ideal orientation can greatly affect the ray output. Thus, the optimization method is used to solve on 8 scalar values. The first 4 describe the linear calibration constants to convert DAC to angles, the last 4 describe the initial configuration of the normal. All other dimensions that describe the projector are assumed to be nominal

II. UNIT VECTOR GENERATION

To generate unit vectors spherical coordinates are used. This allows the optimization to be done without constraints on vector length.

$$u_x = \sin(\phi)\cos(\theta) \tag{1}$$

$$u_y = \sin(\phi)\sin(\theta) \tag{2}$$

$$u_z = \cos(\phi) \tag{3}$$

Therefore, two angles can describe any unit vector in 3 space.

III. RAY-PLANE INTERSECTION

Laser ray r is defined by a point and a vector for some interval of λ. For some interval $\lambda \in (a, b)$ the finite line describes the light ray in space.

$$r\colon \bar{x}_0 + \lambda\bar{v} \tag{4}$$

The mirror plane is defined also with a point and vector by the below equation.

$$m\colon (\bar{x} - \bar{p}_0)\cdot\bar{n} = 0 \tag{5}$$

The intersection between the line and the plane $\bar{y}$ is then given by (7).

$$s = \frac{\bar{p}_0\cdot\bar{n} - \bar{x}_0\cdot\bar{n}}{\bar{v}\cdot\bar{n}} \tag{6}$$

$$\bar{x}' = \bar{x}_0 + s\bar{v} \tag{7}$$

IV. RAY REFLECTION

The final equation to generate the ray output from a mirror is calculated using the ray orientation and the normal vector. The calculation is to reflect $\bar{v}$ across $\bar{n}$ shown in (8).

$$\bar{v}' = \bar{v} - 2(\bar{v}\cdot\bar{n})\bar{n} \tag{8}$$

V. SERIAL RAY GENERATION

Using a ray and a plane, the output ray can be generated. This process can repeated from multiple mirrors. In this case there are only two mirrors defined by $\bar{n}_1, \bar{p}_1, \bar{n}_2, \bar{p}_2$. If the ray input is defined by $\bar{x}_0, \bar{v}_0$ then the following calculations will generate the output of the projector.

The output of m_1, r_1 is defined by the ray-plane intersection and ray reflection equations.

$$\bar{x}_1 = \bar{x}'(\bar{x}_0, \bar{v}_0, \bar{p}_1, \bar{n}_1) \tag{9}$$

$$\bar{v}_1 = \bar{v}'(\bar{v}_0, \bar{n}_1) \tag{10}$$

$$r_1\colon \bar{x}_1 + \lambda\bar{v}_1 \tag{11}$$

Applying the same equations to r_1 using m_2 generates r_2 in (14).

$$\bar{x}_2 = \bar{x}'(\bar{x}_1, \bar{v}_1, \bar{p}_2, \bar{n}_2) \tag{12}$$

$$\bar{v}_2 = \bar{v}'(\bar{v}_1, \bar{n}_2) \tag{13}$$

$$r_2\colon \bar{x}_2 + \lambda\bar{v}_2 \tag{14}$$

VI. Objective Formulation

The objective used is the average perpendicular distance from the point in space to the corresponding ray generated by the project with corresponding angle pairs.

The perpendicular distance between $\bar{x}_i$ and line r_i is given by (15).

$$d = norm\left((\bar{x}_i - \bar{x}_0) - ((\bar{x}_i - \bar{x}_0) \cdot \bar{v})\bar{v}\right) \qquad (15)$$

The objective is then the average of d over all data points. Currently, the first optimization algorithm takes into account only 8 variables describing the projector. Until dimensions can be verified, these will be the only ones used.

$$M = \frac{1}{n}\sum_{i=1}^{n} d(k_1, b_1, k_2, b_2, \phi_1, \theta_1, \phi_2, \theta_2) \qquad (16)$$

VII. Initial conditions

Using an initial conditions generated from the "Explicit method to solving angle calibration functions to dual mirror galvonometers" the seed values for k_1, b_1, k_2, b_2

Optimization method to solving relative rotation and translation of galvonometer systems

Owen Lu
Electroimpact Inc.
owenl@electroimpact.com

***Abstract*—This document describes one method to solving for the rotation and translation of the galvanometer axes relative to a reference coordinate frame. Optimization methods are suggested due to the current algorithm returning different results based on the registration order of calibration points.**

I. Introduction

At the core of finding the rotation and translation of the laser reference frame in respect to the absolute reference frame, ray equations must be solved. Loosely speaking, the simplified problem has a set of 4 or more angle pairs θ_1, θ_2 describing mirror angles with corresponding calibration points x in 3 space for which a laser passes through. The ray, described by a mathematical line corresponding θ_1, θ_2 should then pass through all calibration points if no error exists.

In experience, the laser projectors solving method gives results that depend greatly on the order in which points are registered. This is not ideal since consistent results are unable to be realized. Furthermore, there should be a "best" solution that gives the least error possible, and the operator should not need to permute calibration points to find this best solution.

II. Quarternion Rotation

Quaternion representation of rotation is well understood and resources can be found in many places online. However, a brief introduction here will be done for convenience and notation purposes.

Suppose that we wish to generate rotation of a vector around $\bar{u}$ by angle θ. This can be done using a quaternion q and generating a rotation matrix $\boldsymbol{R}$.

$$q = \left[\cos\left(\frac{\theta}{2}\right), u_x \sin\left(\frac{\theta}{2}\right), u_y \sin\left(\frac{\theta}{2}\right), y_z \sin\left(\frac{\theta}{2}\right)\right]'$$

$$q = [q_r, q_i, q_j, q_k]'$$

Where

$$\boldsymbol{R}(q) = \begin{bmatrix} 1-2q_j^2-2q_k^2 & 2(q_iq_j - q_kq_r) & 2(q_iq_k + q_jq_r) \\ 2(q_iq_j + q_kq_r) & 1-2q_i^2-2q_k^2 & 2(q_jq_k - q_iq_r) \\ 2(q_iq_k - q_jq_r) & 2(q_jq_k + q_iq_r) & 1-2q_i^2-2q_j^2 \end{bmatrix}$$

Then applying the rotation to general vector $\bar{v}$ to formulate rotated vector $\bar{v}'$ can be done with the equation following.

$$\bar{v}' = \boldsymbol{R}(q)\bar{v}$$

III. Ray Formulation

In a previous document ray equations in the galvanometer frame were derived depending on mirror angles. The results are reused below without derivation for convenience.

$$\bar{p}' = \begin{pmatrix} x_1 - y_1\cot(\theta_1) - \dfrac{t_1}{2\sin(\theta_1)} \\ 0 \\ 0 \end{pmatrix}$$

$$\bar{p}'' = \bar{p}' + \begin{pmatrix} \left(y_2 - z_2\cot(\theta_2) - \dfrac{t_2}{2\sin(\theta_2)}\right)\cot(2\theta_1) \\ y_2 - z_2\cot(\theta_2) - \dfrac{t_2}{2\sin(\theta_2)} \\ 0 \end{pmatrix}$$

$$\bar{u}'' = \begin{pmatrix} \cos(2\theta_1) \\ \sin(2\theta_1)\cos(2\theta_2) \\ \sin(2\theta_1)\sin(2\theta_2) \end{pmatrix}$$

The ray in galvanometer coordinates, defined as frame B is given below.

$$r_{i_B} = \bar{p}''(\theta_1, \theta_2) + \lambda\bar{u}''(\theta_1, \theta_2)$$

IV. Ray Transformation

The ray equations must be converted to equations in the reference frame since calibration points are known in said frame.

Let L be the translation of B to A.

$$\bar{l} = [l_1, l_2, l_3]'$$

Let q be the quaternion describing the rotation of B to A

$$q = [q_r, q_i, q_j, q_k]'$$

Suppose a line r is defined in the equation below in the galvanometer frame B.

$$r_B\colon \bar{p}'' + \lambda\bar{u}'' \tag{1}$$

If we wish to find the same line equation within reference frame A we must know the quaternion q and translation $\bar{l}$ that when applied to A creates an equivalent frame with B.

At this point the line becomes a straight forward calculation.

$$r_A\colon \boldsymbol{R}(q)\bar{p}'' + \bar{l} + \lambda\boldsymbol{R}(q)\bar{u}''$$

Let

$$\boldsymbol{R}(q)\bar{u}'' = \bar{u}_A$$

$$\bar{x}_{0_A} = \boldsymbol{R}(q)\bar{p}'' + \bar{l}$$

By substitution

$$r_A = \bar{x}_{0_A} + \lambda \bar{v}_A$$

V. Objective Formulation

Ideally the objective has a physical meaning so that the user can easily verify whether the optimization algorithm has converged to a reasonable solution. For this reason, the average distance from the ray to a calibration point is used.

The perpendicular distance between some point $\bar{x}_i$ and line r_i can be given by the equation below.

$$dist(r_i, x_i) = norm\left(\left(\bar{x}_i - \bar{x}_{0_A}\right) - \left(\left(\bar{x}_i - \bar{x}_{0_A}\right)\cdot \bar{v}_A\right)\bar{v}_A\right)$$

The average distance to a calibration M is given below.

$$M = \frac{1}{n}\sum_{i=1}^{n} dist(r_{i_A}(q, \bar{l}), \bar{x}_i)$$

Under constraint

$$\|q\| = 1$$

VI. Numerical Testing

To test whether this method can converge upon reasonable results, a sample data set was used assuming the following parameters.

$$k = 1$$

$$t_1 = 5mm$$

$$t_2 = 5mm$$

$$x_1 = 0$$

$$y_1 = 0$$

$$y_2 = 40mm$$

$$z_2 = 0$$

θ_1 (deg)	θ_2 (deg)	$x(mm)$	$y(mm)$	$z(mm)$
45	55	-3.53	-327.02	1000
45	65	-3.53	-801.85	1000
55	55	-403.82	-327.02	1000
55	65	-491.73	-801.85	1000
65	55	-926.71	-327.02	1000
65	65	-1129.37	-801.85	1000

The optimization function used was *fmincon* in MATLAB. Random seed values were used with to generate quaternions rotations within 10 degrees of the correct solution, while errors of 0.5 metres was used for the translational error.

A plot of the average error as a function of the seed angular error and the seed displacement error is given following.

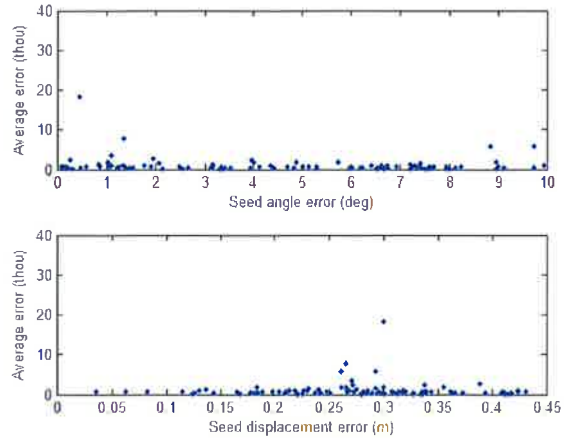

Figure 1 – Average distance error vs. Seed angle error (top); Average distance error vs. Seed displacement error (bottom

From the results, the error is small, under 20 thou if the initial seed can be estimated to reasonable accuracy. More tests are necessary, but initial feasibility has been validated.

*Note: A key setting in the MATLAB function is that TolFun = 0.000001.