Fault Diagnosis for a Three-Wheel Omidirectional Vehicle: A Geometric Approach

1st Adrian Lizarraga  
Dep. of Electrical Engineering  
CINVESTAV  
Guadalajara, Mexico  
Adrian.Lizarraga@cinvestav.mx

2nd Ofelia Begovich  
Dep. of Electrical Engineering  
CINVESTAV  
Guadalajara, Mexico  
ofelia.begovich@cinvestav.mx

3rd Antonio Ramírez  
Dep. of Electrical Engineering  
CINVESTAV  
Guadalajara, Mexico  
antonio.ramirez@cinvestav.mx

Abstract—This work presents a fault diagnoser for a three-wheel omnidirectional vehicle, which is based on the FDI (Fault Detection and Isolation) approach proposed by Massoumnia [1]. It is designed for a linearized model of the vehicle (mobile robot). More precisely, this diagnoser will detect and locate actuator's faults, when the velocities and orientation angle of the robot are considered constants. Then, the efficiency of the proposed method is evaluated on the nonlinear model of the vehicle. By virtue of the method, it is capable to locate both concurrent and non-concurrent faults.

Index Terms—Fault Diagnosis, Mobile Robots, Actuators, Nonlinear systems

I. INTRODUCTION

Mobile robots have been used in various fields such as planetary exploration [2], military (search and rescue), warehouses, etc. Besides, several works related to mobile robots and autonomous driving are mainly concerned with planning, control, perception and vehicle localization [2], [3]. The reliability and security are particularly important in mobile robots and autonomous driving applications to prevent human accidents or economical losses due to accidents. In order to improve the reliability and security in mobile robots, an early fault diagnosis is an important stage in these systems to reduce unnecessary risks and the time to fixing the faulty component.

There exist papers tackling the fault diagnosis in mobile robots as in [4], where the authors describe four model-based fault diagnosis approaches for mobile robots: the parity space relation, Hidden Markov Model (HMM), particle filter and Observable Operator Model (OOM). The OOM’s are a generalization of the HMM, where probabilistic and linear algebra tools are used to update information of a system by using its present observations. Some numerical results are shown for three of the approaches (HMM, particle filter and OOM) which are applied to a kinematic model of a four wheel omnidirectional mobile robot.

The Extended Kalman filter is also a common used algorithm for fault detection and isolation in Mobile Robots [5]–[8]. For example, in [5] is used a bank of Extended Kalman Filters (EKF) and the Fault isolation is achieved by using a probabilistic analysis of the residuals signals of each EKF. In [6] a mathematical model and an EKF are used to detect and isolate wheel actuators and encoders faults on a unicycle mobile robot. The work in [7] is similar to [5], but they can only isolate some faults combinations such as: right and left encoder faults; encoder and gyroscope faults and gyroscope and wheel actuator faults. In [8] an EKF and two particle filter algorithms are compared to evaluate their efficiency in detecting and isolating battery and encoder faults in an omnidirectional mobile robot. In their simulation experiments the Rao-Blackwellised particle filter has the best performance for detecting and isolating battery faults. In [9] a multiple particle filters method and a ruled based inference for sensors fault diagnosis are used. In that work, only certain faults are observable in certain movements of the mobile robot. Finally, it is important to mention that most of the aforementioned works are based on kinematic instead of dynamic models (which are more accurate). Besides, most of the referenced works use probabilistic or statistical analysis to isolate the faults and they do not mention if concurrent faults could be isolated.

In the literature there exist an elegant algorithm capable of isolating concurrent faults. This algorithm is the fault diagnoser proposed by Massoumnia [1]. However, at our knowledge the Massoumnia’s diagnoser has not been applied in mobile robots yet. Then, our interest is to study the feasibility of applying the Massoumnia’s method to isolate actuators faults (concurrent and nonconcurrent) in mobile robots.

This paper is concerned with the application of the Massoumnia fault diagnoser [1], [10], which is applied to detect and isolate actuators faults, concurrents or non-concurrent, in a three-wheel omnidirectional mobile robot. First, a dynamic model for the vehicle is obtained and it is linearized at a certain operating point, where it is assumed that the vehicle’s velocities and its orientation angle are constants. Then, it is shown that the linearized model satisfies the existence conditions of the Massoumnia’s residual generator (fault diagnoser). Finally, the designed fault diagnoser is applied to the nonlinear model of the mobile robot. Although, the faults change the operating point, inducing some nonlinearities in the system, the designed fault diagnoser is able to detect and locate the faults. To prevent an accident, after the fault isolation, the vehicle is stopped. Simulation results demonstrate the effectiveness of the proposed diagnoser to isolate both concurrent and non-
concurrent faults. Then, the main contribution of this work is to show that it is possible to design a Massoumnia’s FDI diagnoser capable of isolate concurrent actuators faults in this kind of mobile robots under the imposed conditions.

This work is organized as follows: in Section II the dynamics of a three-wheel omnidirectional vehicle are described. Then, both the mathematical background needed for the understanding of the Geometric Fault diagnosis and the numeric algorithms to calculate the needed matrices for the diagnoser are given in Section III. Section IV showns the linearization of the dynamics of a three wheel omnidirectional vehicle. The results of the proposed fault diagnoser applied to the simulated faulty vehicle, are shown in Section V. Finally the conclusions are given in Section VI.

II. NONLINEAR MODEL

In this section, the dynamic nonlinear model of an omnidirectional mobile robot will be described.

Omnidirectional vehicles can have three, four, or more omnidirectional wheels (universal, mecanum or ball wheel). This work considers the three-wheeled vehicle with universal wheels type depicted in Fig. 1. An universal wheel allows a sideways motion and a forward motion due to rollers which are mounted on the outer diameter of the wheel.

![Fig. 1. Three-Wheel Omnidirectional mobile robot with universal wheels which allows a sideways motion.](image)

The dynamics equations of a three-wheel omnidirectional vehicle in global coordinate frame are given by [2]:

\[
\begin{align*}
\dot{X} &= V_X; \quad \dot{Y} = V_Y; \quad \dot{\phi} = \dot{\phi} \\
V_X &= a_2 V_Y - a_1 \cos(\phi) U_1 - \sqrt{3} a_1 \sin(\phi) U_2 \\
V_Y &= a_4 V_X V_\phi + a_2 V_Y - a_1 \sin(\phi) U_1 + \sqrt{3} a_1 \cos(\phi) U_2 \\
\dot{\phi} &= -b_2 V_\phi + b_1 U_3 \\
U_1 &= \tau_1 + \tau_2 - 2 \tau_3; \quad U_2 = \tau_1 - \tau_2; \quad U_3 = \tau_1 + \tau_2 + \tau_3 \\
a_1 &= \frac{K R_m}{2 R_o^2 M + 3 I_o}; \quad a_2 = \frac{-3 \beta}{2 R_o^2 M + 3 I_o}; \quad a_3 = \frac{2 R_o^2 M}{2 R_o^2 M + 3 I_o} \\
a_4 &= 1 - a_3; \quad b_1 = \frac{K R_m}{3 R_o^2 I_o + I_o R_m}; \quad b_2 = \frac{3 R_o^2 \beta}{3 R_o^2 I_o + I_o R_m}
\end{align*}
\]

where \( \dot{X} \) and \( \dot{Y} \) are the center of mass velocity of the vehicle along the \( X \) and \( Y \) axis, respectively. \( I_o \) is the moment of inertia about \( Q \), \( I_o \) is the moment of inertia of the wheels, \( \beta \) is the linear friction coefficient, \( K \) is the driving torque gain, \( R_o \) is the radius of the wheels, \( M \) is the vehicle mass and \( \tau_i \) is the driving input torque of the \( i^{th} \) wheel, where \( i = 1, 2, 3 \).

III. GEOMETRIC APPROACH

This section briefly resumes the method presented in [1], [10] and how this method can be applied to our mobile robot. Consider the following Linear Time Invariant system (LTI):

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) + \sum_{i=1}^{k} L_i m_i(t) \\
y(t) &= Cx(t)
\end{align*}
\]

where \( x(t) \in \mathbb{R}^n \) is the state vector, \( u(t) \) are input signals with dimension \( k \), \( y(t) \) are output signals with dimension \( p \), \( A, B \) and \( C \) are matrices of known dimensions, \( L_i \) is a fault signature and \( m_i \) is a fault mode [11]. In order to model a fault in the \( i^{th} \) actuator, \( L_i \) is chosen as the \( i^{th} \) column of the matrix \( B \) and \( m_i \) is chosen to model a fault type [12]. In this work, we will design a residuals generator in which the residual \( r_i \) will only be affected by the fault on the \( i^{th} \) actuator, then the others actuators faults will not be observed in this residual.
When there is no fault, all the residuals will be closed to zero. On the other hand, if a fault is present on the $i^{th}$ actuator, then the $r_i$ residual will be greater than a preset threshold. In the case of our three-wheel omnidirectional vehicle $i = 1, 2, 3$.

In order to obtain a residual $r_i$ which can only be affected by the fault $L_i m_i$, the following Isolability Condition is necessary and sufficient [10]:

$$S_i^* \cap L_i = 0 \quad (3)$$

where $S_i^*$ is the infimal unobservability subspace and $L_i = \text{Im}(L_i)$, where the fault $L_i$ is the fault that we want to isolate and therefore must not be contained in $S_i^*$, whereas the faults that we do not want to observe in our residual $r_i$, say $L_j$ ($j \neq i$), are contained in the subspace $S_j^*$. A family of fault signatures that satisfied (3) are called strongly identifiable and from (3) follows that the fault signatures must necessarily be independent [10], [11], i.e.

$$\text{Rank}\{L_1, L_2, ..., L_k\} = k \quad (4)$$

where $k = 3$ for our mobile robot. In order to calculate the $S_i^*$ subspace, first the infimal (C,A) invariant subspace $W_i^*$ must be found. Such subspaces can be calculated in terms of the matrix algorithms cited in [10]. When $S_i^*$ is calculated, a matrix $P_i$ will be found too and such matrix is known as the canonical projection of $X$ on the quotient space $X/S_i^*$, where $x(t) \in X$.

Once (3) is satisfied, the fault diagnoser can be designed. Let $D_0i$ be a solution of $P_i(A + D_0iC)S_i^* = 0$; $H_i$ be a solution of $\text{Ker}(H,C) = S_i^* + \text{Ker}(C)$ and $M_i$ be a unique solution of $M_iP_i = H_iC$. Let $A_{0i} = (A + D_0iC : X/S_i^*)$ where $(A + D_0iC) : X/S_i^*$ denotes the insertion map of $(A + D_0iC)$ on $X/S_i^*$. By construction the pair $(M_i, A_{0i})$ is observable, hence exist a matrix $D_1i$ such that the spectrum of $(A_{0i} + D_1iC_0)$ can be assigned arbitrarily, i.e. $\sigma(A_{0i} + D_1iC_0) = \Lambda$ where $C_0i = M_i$, $\Lambda$ is an arbitrary symmetric set and the gain matrix is $D_i = D_0i + P_1i^* D_1i H_i$. Finally, let $E_i = P_i D_i$ and $G_i = P_i B$. Thus, the residual filter which will be only affected by the fault $L_i m_i$ is given by:

$$\dot{w}_i(t) = F_i w(t) - E_i y(t) + G_i u(t) \quad (5)$$

$$r_i(t) = M_i w(t) - H_i y(t) + K_i u(t)$$

where $w_i$ is the state of the residual generator and $r_i$ is the residue. The matrices of (5) can be obtained as follows:

$$F_i = P_i(A + D_i C) P_i^{-T}; \quad G_i = P_i B; \quad E_i = P_i D_i \quad (6)$$

$$M_i = H_i C P_i^{-T}; \quad K_i = 0$$

where $P_i^{-T}$ denotes the right inverse of $P_i$. Based on (5) it can be seen that the residual generator only needs as inputs the output and the input of the system, $y(t)$ and $u(t)$, respectively.

### IV. Vehicle Model Linearization

In this section is described the linearization of the nonlinear model of the three-wheel omnidirectional vehicle. The linearized model is necessary to design the Massoumnia’s fault diagnoser.

The linearization is done under the following assumptions:

1) The vehicle’s orientation angle must be constant.
2) The vehicle’s linear velocities must be constants.

Note that the three-wheel omnidirectional vehicle can accomplish the above assumptions due to its geometry (see Fig. 2).

Once (1) is linearized, a state space model is derived, with the state vector defined as $x = [X_Q, Y_Q, \phi, V_x, V_y, V_\alpha]^T = [x_1, x_2, x_3, x_4, x_5, x_6]^T$, then the following matrices of the linear system are obtained:

$$A = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & \alpha & a_2 & -a_4 x_6 & -a_4 x_5^* \\
0 & \gamma & a_4 x_6^* & a_2 & a_4 x_4^* \\
0 & 0 & 0 & 0 & -b_2 & 0
\end{bmatrix} \quad (7a)$$

$$B = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
- a_1 \lambda_1 & - a_1 \lambda_2 & 2 a_1 \cos(x_4^*) \\
- a_1 \lambda_3 & - a_1 \lambda_4 & 2 a_1 \sin(x_3^*) \\
0 & 0 & 0 \\
b_1 & b_1 & b_1
\end{bmatrix} \quad (7b)$$

$$C = I^{6 \times 6} \quad (7c)$$

where $I$ is an identity matrix (all the vehicle’s states can be measured using sensors), $\alpha = a_1 \sin(x_4^*) U_1 - a_1 \sqrt{3} \cos(x_3^*) U_2$, $\gamma = -a_1 \cos(x_3^*) U_1 - a_1 \sqrt{3} \sin(x_3^*) U_2$, $\lambda_1 = \cos(x_3^*) + \sqrt{3} \sin(x_4^*)$, $\lambda_2^* = \cos(x_3^*) - \sqrt{3} \sin(x_3^*)$, $\lambda_3 = \sin(x_4^*) - \sqrt{3} \cos(x_3^*)$, $\lambda_4 = \sin(x_4^*) + \sqrt{3} \cos(x_3^*)$ and $x_4^*$, $x_5^*$, $x_6^*$ and $x_*^*$ are the vehicle’s constant velocities and its orientation angle at the chosen operating point.

It can be proved that the columns of $B$ in (7b) are linearly independent and since the fault signatures $(L_1, L_2, L_3)$ are chosen as the columns of $B$, they satisfy the fault diagnoser necessary existence condition (4). Then, from “The Special Case of C Monic”, cited in [10], it follows that if the condition (4) is satisfied and the $C$ matrix is monic (it has full rank columns, i.e. $\text{Ker}(C) = 0$), then the condition (3) is satisfied. In our case the $C$ matrix in (7c) is monic and the condition (4) is satisfied. Thus, a fault diagnoser for the linear state model (7) can be designed, since condition (3) is satisfied.

The numeric values of the designed fault diagnoser’s matrices are shown in the appendix, where it is assumed that the vehicle is moving along the $X$ axis, with $x_3^* = 0$ rad, $x_4^* = 0.02$ m/s, $x_5^* = 0$ m/s and $x_6^* = 0$ rad/s.

### V. Simulation Results

In this section, simulation results are presented for the designed FDI algorithm which is applied to the nonlinear model of our three-wheel omnidirectional vehicle which is
shown in Fig. 1. This mobile robot was built in our laboratory and it has three brushless motors as actuators for the wheels. The parameters of the model are taken from our mobile robot and they are shown in Table I. In the case of the velocity of the mobile robots, others papers report velocities from 0.04 m/s up to 0.6 m/s [13]–[15]. Since, the vehicle in [13] is similar to our mobile robot, in this paper a similar linear velocity is chosen. The simulations are done using Simulink®. The scenario of our experiments are the following: the faults occur at the 25 seconds of the simulation start and the simulated faults are \( m_i = -u_i(t) \), \( i = 1, 2, 3 \), i.e. the complete failure of the \( i^{th} \) actuator. In addition, at the output of the system is added some gaussian noise with mean zero and variance \( 1 \times 10^{-6} \). Besides, it is used a moving average filter on the residuals \( r_i \) of the designed fault diagnoser. Finally, the threshold for detecting faults is calculated using the RMS algorithm [16] on the fault diagnoser’s residuals and it is equal to \( 8 \times 10^{-4} \).

In Fig. 3, 4, 5 the residuals of the nonconcurrent faults experiments are shown. In each experiment a fault was simulated (fault on the first actuator, on the second actuator and on the third actuator, respectively). It can be seen that on each experiment only one residual is greater than the threshold. Then, the faults are isolated approximately 0.3 seconds later. Note in Fig. 3, 4, 5 that after the faults appears, the other residuals increased gradually which is caused due to the change induce by the fault. Nonetheless, the fault diagnoser have been able to isolate the correct faulty actuators in the three cases. Moreover, when the fault is isolated, in order to avoid the residuals drift, the vehicle is immediately stopped to prevent an accident. Besides, the poles of the diagnoser are assigned to almost twice as faster as the poles of the linear model which satisfies a fast convergence of the designed fault diagnoser and at the same time the unmodelled dynamics of the system are avoided.

In Fig. 7, 8 and 9 are shown the residuals of two concurrent faults. The results are similar to the nonconcurrent fault experiments, but in this case, two residuals are greater than the threshold, since two faults occurs simultaneously.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_w )</td>
<td>0.04</td>
<td>m</td>
</tr>
<tr>
<td>( M )</td>
<td>4.5</td>
<td>kg</td>
</tr>
<tr>
<td>( R )</td>
<td>0.165</td>
<td>m</td>
</tr>
<tr>
<td>( I_o )</td>
<td>( 2.416 \times 10^{-4} )</td>
<td>kg.m²</td>
</tr>
<tr>
<td>( I_Q )</td>
<td>0.119025</td>
<td>kg.m²</td>
</tr>
<tr>
<td>( \beta )</td>
<td>0.1983</td>
<td></td>
</tr>
<tr>
<td>( K_r )</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>( \tau_1 )</td>
<td>-0.05</td>
<td>N.m</td>
</tr>
<tr>
<td>( \tau_2 )</td>
<td>-0.05</td>
<td>N.m</td>
</tr>
<tr>
<td>( \tau_3 )</td>
<td>0.1</td>
<td>N.m</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>0</td>
<td>rad</td>
</tr>
<tr>
<td>( x_4 )</td>
<td>0.02</td>
<td>m/s</td>
</tr>
<tr>
<td>( x_5 )</td>
<td>0</td>
<td>m/s</td>
</tr>
<tr>
<td>( x_6 )</td>
<td>0</td>
<td>rad/s</td>
</tr>
</tbody>
</table>

VI. CONCLUSIONS

The linear Massoumnia’s fault diagnoser is designed for a linearized dynamic model of a three-wheel omnidirectional mobile robot and it is tested on its nonlinear model. First it is verified that the Isolability Condition is satisfied for the linear model of the studied mobile robot, then it was possible to design a Massoumnia’s diagnoser. The simulation results show that the proposed diagnoser is able to detect and isolate both concurrent and nonconcurrent actuators faults, despite the nonlinearities induced by the faults. In order to prevent an accident, a simple control is proposed to stop the

![Fig. 3. The fault is simulated at the 25 seconds on the first actuator. Immediately after the fault occurred the blue residual associated with the fault on the first actuator is greater than threshold.](image)

![Fig. 4. Fault on the second actuator simulated at the 25 seconds of simulation. Immediately after the fault occurred the dark residual associated with the second actuator is greater than the threshold.](image)

![Fig. 5. Fault on the third actuator simulated at the 25 seconds of simulation. Immediately after the fault occurred the green residual associated with the third actuator is greater than the threshold.](image)
the first and third actuators are greater than the threshold.

Fig. 8. Concurrent faults on the first and third actuators at the 25 seconds of simulation. Immediately after the fault occurred the residual associated with the first and third actuators are greater than the threshold.

Fig. 9. Concurrent faults on the second and third actuators at the 25 seconds of simulation. Immediately after the fault occurred the residual associated with the second and third actuators are greater than the threshold.

A. Calculated matrices for isolating faults on the first actuator

The matrices $F_1$, $G_1$, $E_1$, $M_1$ and $H_1$ are calculated as in Section III, equation (6). These matrices are the following:

\[
F_1 = \begin{bmatrix}
-120 & 0 & 0 & 0 \\
0 & -139.127167 & 0 & 0 \\
0 & 0 & -177.381501 & 0 \\
0 & 0 & 0 & -120.542645
\end{bmatrix}
\]

\[
G_1 = \begin{bmatrix}
0 & 0 & 0 \\
-3.939056 & 0 & 0 \\
6.822646 & 0 & 0 \\
0.663475 & 0 & 0
\end{bmatrix}
\]

\[
M_1 = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
E_1 = \begin{bmatrix}
0 & -11.554199 & -25.515075 & 118.868268 \\
0 & 120.074381 & 85.914048 & 17.278043 \\
-100 & 0.341132 & -0.590858 & -0.057459 \\
0 & -24.855947 & 59.212032 & 4.381676 \\
0 & 43.770976 & -102.323045 & -5.737958 \\
-1 & 2.595529 & -7.266002 & -0.306294
\end{bmatrix}^T
\]

\[
H_1 = \begin{bmatrix}
0 & 0.083048 & 0.143843 & -0.986109 \\
0 & -0.863055 & -0.484346 & -0.143336 \\
1 & 0 & 0 & 0 \\
0 & 0.248239 & -0.429963 & -0.041812 \\
0 & -0.429963 & 0.744718 & 0.072421 \\
0 & -0.041812 & 0.072421 & 0.007043
\end{bmatrix}^T
\]
B. Calculated matrices for isolating faults on the second actuator

The matrices $F_2$, $G_2$, $E_2$, $M_2$ and $H_1$ are calculated as in Section III, equation (6). These matrices are the following:

\[
F_2 = \begin{bmatrix}
-100 & 0 & 0 & 0 \\
0 & -139.13206 & 0 & 0 \\
0 & 0 & -177.396181 & 0 \\
0 & 0 & 0 & -120.542784
\end{bmatrix}
\]

\[
G_2 = \begin{bmatrix}
0 & 0 & 0 \\
0 & -3.939056 & 0 \\
0 & -0.822646 & 0 \\
0 & 0.663475 & 0
\end{bmatrix}
\]

\[
M_2 = M_1
\]

\[
E_2 = \begin{bmatrix}
0 & -11.554605 & 25.517187 & 118.868405 \\
0 & 120.076804 & -85.921158 & 17.278062 \\
-100 & -0.341132 & -0.590858 & 0.057459 \\
0 & -24.857162 & -59.218343 & 4.381682 \\
0 & -42.046969 & -103.302669 & 6.024639 \\
-1 & -2.594909 & 7.265638 & -0.306156
\end{bmatrix}^T
\]

\[
H_2 = \begin{bmatrix}
0 & 0.083048 & -0.143843 & -0.986109 \\
0 & -0.863055 & 0.484346 & -0.143336 \\
1 & 0 & 0 & 0 \\
0 & 0.248239 & 0.429963 & -0.041812 \\
0 & 0.429963 & 0.744718 & -0.072421 \\
0 & -0.041812 & -0.072421 & 0.007043
\end{bmatrix}^T
\]

C. Calculated matrices for isolating faults on the third actuator

The matrices $F_3$, $G_3$, $E_3$, $M_3$ and $H_3$ are calculated as in Section III, equation (6). These matrices are the following:

\[
F_3 = \begin{bmatrix}
-100 & 0 & 0 & 0 \\
0 & -196.518455 & 0 & 0 \\
0 & 0 & -120 & 0 \\
0 & 0 & 0 & -120.542715
\end{bmatrix}
\]

\[
G_3 = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 7.878113 \\
0 & 0 & 0 \\
0 & 0 & 0.663475
\end{bmatrix}
\]

\[
M_3 = M_1
\]

\[
E_3 = \begin{bmatrix}
0 & -16.320416 & 17.261152 & 118.868337 \\
0 & -2.372247 & -118.752064 & 17.278052 \\
-100 & 0 & 0 & 0 \\
0 & -156.161748 & 0.143843 & -5.805029 \\
0 & -0.012071 & -0.989691 & 0.143336 \\
-1 & -9.898549 & 0 & -0.306225
\end{bmatrix}^T
\]

\[
H_3 = \begin{bmatrix}
0 & 0.083048 & -0.143843 & -0.986109 \\
0 & 0.012071 & 0.989601 & -0.143336 \\
1 & 0 & 0 & 0 \\
0 & 0.992957 & 0 & 0.083624 \\
0 & 0 & 0 & 0 \\
0 & 0.083624 & 0 & 0.007043
\end{bmatrix}
\]

\[
T
\]

\[


