A Comparison of Forward and Inverse Simulation Methods for Fault Detection on a Rover

Stuart R.K. Shilliday[1], Euan W. McGookin[2], and Douglas G. Thomson[3]

Abstract— **Fault tolerant design is hugely important for autonomous mobile robots such as planetary exploration rovers (PERs), as they are required to be both robust and reliable in extremely harsh environments. One of the main principles of fault tolerance is the detection and diagnosis of any faults afflicting the system. A model-based fault detection procedure is presented using forward and inverse simulation methods. The results of each method are compared for faults in different system locations to display the differences and advantages of both methods. It is shown by this comparison that the methods complement each other and can be used concurrently to diagnose output and input faults.**

I. Introduction

Planetary Exploration Rovers (PERs) have become exceptionally useful and effective tools utilised by scientists to explore the surfaces of terrestrial bodies in the Solar System (e.g. moons, planets) [1]. Due to the hostile environments that these rovers operate in, their onboard systems must function autonomously and be highly reliable. To achieve these levels of reliability and autonomy, the rover systems need to be fault tolerant. One method of ensuring this is to design a subsystem that detects whenever a fault occurs at any point within the system, diagnose said fault, and then instruct the rover to take appropriate action to mitigate any adverse effects.

Common methods for detecting and isolating faults are based on mathematical models of the system that are used to predict its dynamic output response. This response is compared to the actual measured output response of the physical system. The quantified difference is called the residual [2] and in general large residual values indicate that a fault has occurred. However, for faults in the input actuators, the system dynamics can hide or alter their effect, which makes determining the exact origin of the faults difficult. In such cases an inverse simulation based approach could be used to estimate the input command applied to the actuators and then calculate an input residual for actuator fault detection and isolation (FDI) [3].

This paper compares the use of forward and inverse simulations for FDI relating to actuator and sensor faults. The application considered in this study is a wheeled rover so that parallels can be drawn with PERs and the benefits of such an FDI architecture demonstrated.

The structure of this paper is as follows. Section II gives an overview of the rover used in this investigation and briefly describes the various components of the mathematical model of this rover. Section III describes the nature of faults, where and how they have been implemented, and provides the results of the faults applied to various locations within the system. Section IV explains the residual calculation and inverse simulation processes, and displays the output and input residuals induced by the two faults. Section V provides a summary of the comparison between the two residual generation methods and Section VI states the conclusions.

II. Rover Modelling, Control, and Guidance

To study the usefulness of residuals for FDI systems onboard a rover, a representative mathematical model must first be established. In this particular case, the rover being modelled is a simple four-wheel drive rover with a fixed chassis and no suspension, that uses electric motors to drive the wheels and turns using a method called slip-steering (also known as skid or differential steering) [4]. This has been selected as it possesses a number of characteristics consistent with commonly used technologies on PERs [5].

To induce this rover to perform manoeuvres and provide a baseline against which the effects of the faults can be measured, a controller system is required. Closed-loop controllers [6] have been used to produce the desired response from the rover and a Line-of-Sight (LOS) guidance system [7] has been implemented to translate a set of waypoints into controller commands. Figure 1 shows a block diagram representing the guidance system, the controllers and the rover model. In this diagram, v represents the resultant velocity of the rover, ψ is the yaw (or heading) angle, V represents the voltages applied to the motors, τ represents the torques produced by the motors, x is the rover's state vector, and y is the vector of outputs measured by the sensors.

The rigid-body dynamics and kinematics of the rover are described by the Newton-Euler equations derived by Fossen [8] and McGookin [9].

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\boldsymbol{\nu}} \\ \dot{\boldsymbol{\eta}} \end{bmatrix} = \begin{bmatrix} -\mathbf{M^{-1}}\left[(\mathbf{C}+\mathbf{D})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) - \boldsymbol{\tau}\right] \\ J(\boldsymbol{\eta})\boldsymbol{\nu} \end{bmatrix} \\ = F(\mathbf{x}, \mathbf{u}) \tag{1}$$

In this equation, $\boldsymbol{\nu}$ is the vector of the accelerations relative to the body fixed axis of the rover; $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{D}$, are the mass/inertia, Coriolis, and damping matrices respectively; $\mathbf{g}$ is the gravitation matrix; and $\boldsymbol{\tau}$ is the input force/torque matrix. $\boldsymbol{\eta}$ is the inertially fixed position/orientation vector

[1] PhD Student, School of Engineering, Room 734, James Watt Building (South), University of Glasgow, Glasgow, Scotland, G12 8QQ, United Kingdom. *s.shilliday.1@research.gla.ac.uk*

[2] Senior Lecturer, School of Engineering, James Watt Building (South), University of Glasgow.

[3] Senior Lecturer, School of Engineering, James Watt Building (South), University of Glasgow.

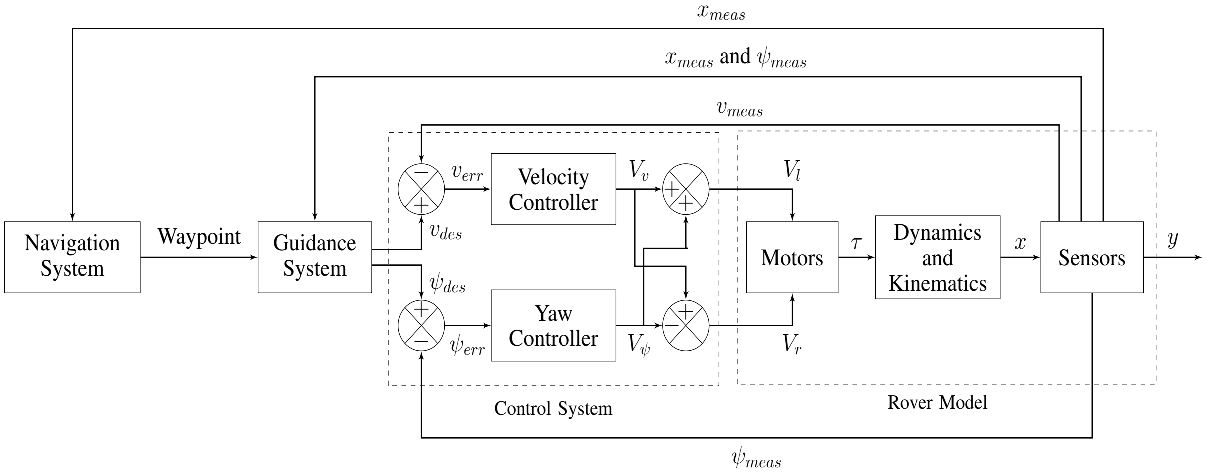


Fig. 1: Rover System Block Diagram

of the rover and J is the transformation matrix that relates $\dot{\boldsymbol{\eta}}$ to $\boldsymbol{\nu}$.

In this representation, the dynamics and kinematics of the rover are shown as a continuous-time state-space model, which means it can be simplified down to a non-linear function $F(\mathbf{x}, \mathbf{u})$. This equation is expanded further in [8] and [9], and is applied to the case of a rover in [3], which also provides the representative model for the motors.

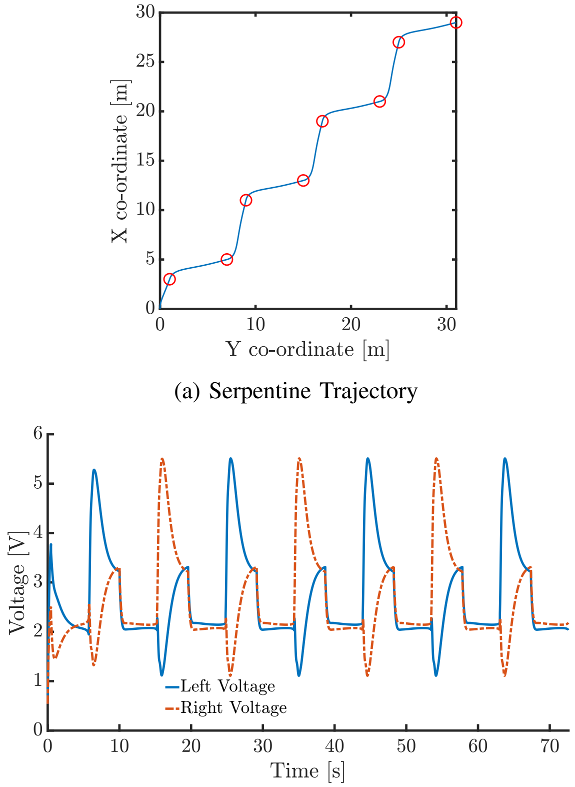
For the nominal case, the sensors are assumed to be ideal, where the state measurements, y, made by the sensors are identical to the state values, x. For the faulty sensor case, y is x plus the additive fault magnitude.

The waypoints selected to test the rover's performance are in a serpentine pattern, shown in Figure 2(a). The guidance system of the rover provides the two controllers with desired velocity and heading values. This pattern of waypoints is good for investigating the dynamics of the system because it ensures that both the velocity and the heading are almost constantly varying as the rover changes direction, accelerates towards each waypoint, and decelerates on approach.

The control voltages corresponding to this response are included in Figure 2(b). This graph shows the implementation of the slip-steering technique; different voltages are applied to the left- and right-hand motors of the rover, causing it to turn. The peaks and troughs where the voltages are in anti-phase are the instances where the rover is turning, and they converge together as the rover converges on the desired heading. The positive offset comes from the velocity controller constantly driving the rover forward and this gives the continuous trajectory response seen in Figure 2(a). These graphs provide a baseline for comparison between the fault-free and faulty rover scenarios.

III. Faults

Experimental and real-world experience has shown that faults are highly likely in mobile robots, particularly "field robots" [10]. In order to monitor for these faults and detect their presence, their nature and behaviour must be sufficiently understood, and to test the FDI procedures they must be modelled to a reasonable degree of accuracy.



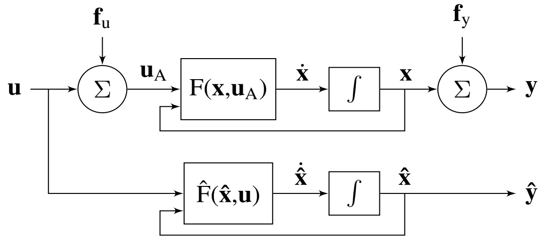
(a) Serpentine Trajectory

(b) Control Voltages

Fig. 2: Rover Serpentine Trajectory and Corresponding Control Voltages

Faults are, by definition, unintended digressions from the expected behaviour of a component or system [11], and as such they almost invariably result in adverse effects on the overall performance of the system.

Faults can occur at multiple points in the system. In the course of this study, two locations have been investigated; one at the voltage input to the motors, and one at the output of one of the sensors. These fault locations are shown in Figure 3 where the input fault is $\mathbf{f}_{\mathrm{u}}$, and the output fault is $\mathbf{f}_{\mathrm{y}}$. These faults are simply added to their respective signals, and are hence called additive faults, but only one fault is implemented at a time for this investigation.



$\mathbf{f}_{\mathrm{u}}$
$\mathbf{f}_{\mathrm{y}}$
$\mathbf{u}$
Σ
$\mathbf{u}_{\mathrm{A}}$
$\mathrm{F}(\mathbf{x},\mathbf{u}_{\mathrm{A}})$
$\dot{\mathbf{x}}$
$\int$
$\mathbf{x}$
Σ
$\mathbf{y}$
$\hat{\mathrm{F}}(\hat{\mathbf{x}},\mathbf{u})$
$\dot{\hat{\mathbf{x}}}$
$\int$
$\hat{\mathbf{x}}$
$\hat{\mathbf{y}}$

Fig. 3: Fault Location Block Diagram

For the purposes of this study, two mathematical models are used. One of these, designated "the model", describes the ideal, fault-free behaviour of the rover, $\hat{\mathrm{F}}(\hat{\mathbf{x}},\mathbf{u})$, and is representative of a mathematical model that would be used in parallel with a real-life rover. The other model used in this study is called "the system" and is representative of the rover itself, $\mathrm{F}(\mathbf{x},\mathbf{u}_{\mathrm{A}})$. F and $\hat{\mathrm{F}}$ are assumed to be identical, other than the inclusion of any faults.

$\mathbf{u}$ is the ideal control voltage input to the motors and $\mathbf{u}_{\mathrm{A}}$ is the actual voltage applied to the motors, including any input faults being implemented. $\dot{\mathbf{x}}$, $\mathbf{x}$, and $\mathbf{y}$ are the system state derivative, state, and output vectors respectively and $\dot{\hat{\mathbf{x}}}$, $\hat{\mathbf{x}}$, and $\hat{\mathbf{y}}$ are the same vectors as generated by the model.

The two faults tested in this study are both abrupt, persistent faults occurring 30 seconds into the simulation of the aforementioned serpentine trajectory. They each have a magnitude of -1 and the respective units are Volts for the input fault and metres for the sensor fault.

Despite the similarities in the characteristics of the two faults modelled, it will be seen that the different locations dramatically alter how they affect the rover's performance.

A. Input Fault

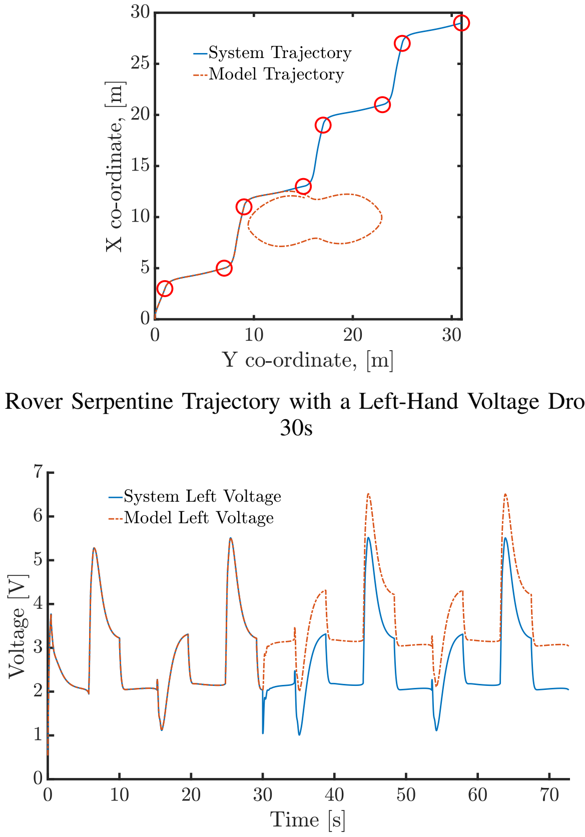
The first fault is applied to the voltage signal input to the rover's left-hand motors. The resultant trajectory of the rover and the applied voltages are shown in Figure 4.

In this figure it can be seen that, even in the presence of the fault, the system closely follows the desired trajectory. This is due to the effect of the closed loop controller compensating for the voltage drop. However, since the model does not have that same fault, the correction made by the controller forces the left-hand voltages consistently higher than required, and the model veers to the right. This overcorrection can be seen by the difference between the model and system left-hand voltages after 30s in Figure 4(b).

It can be noted from the data that the right-hand voltages are consistently identical to each other, and so have been omitted from Figure 4(b) for clarity. Also, it has been noted that, excepting an initial adjustment period of a couple of seconds after the appearance of the fault, the right-hand control voltages match those shown in Figure 2(b).

B. Output Fault

The second fault is again assumed to be a negative offset but is applied to the X-position sensor of the rover. The resulting trajectory and voltage values are shown in Figure 5. In this instance, the controller forces the sensed position to match the desired trajectory, whereas in reality the rover is consistently displaced 1m in the X-direction.



(a) Rover Serpentine Trajectory with a Left-Hand Voltage Drop at 30s

(b) Voltage Values showing a Left-Hand Voltage Drop at 30s

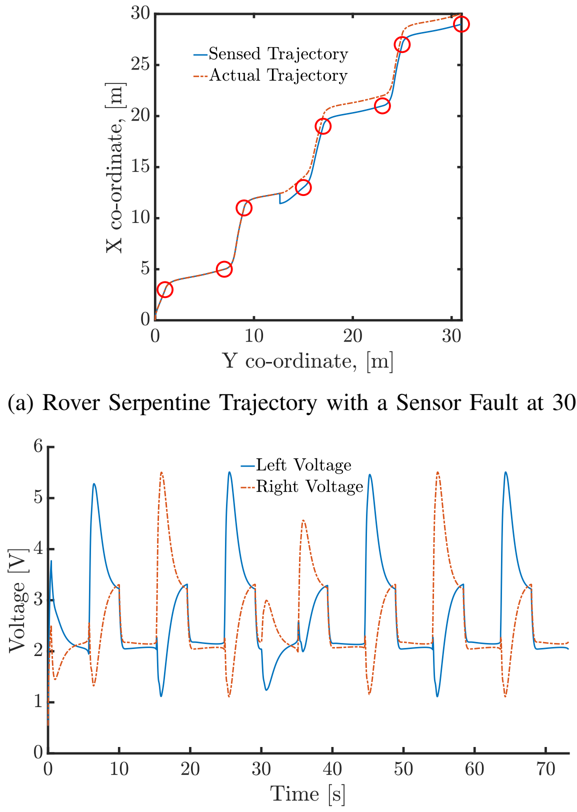
Fig. 4: Rover Serpentine Trajectory and Corresponding Motor Voltages with a Left-Hand Voltage Drop at 30s

The negative ramifications of this are obvious. If the rover travels along what it assumes to be a safe trajectory, unaware of the discrepancy between its sensed and actual positions then it runs the risk of encountering obstacles that would have otherwise been avoided.

The effect of the controller relying on a faulty sensor to provide the feedback signal is also seen in Figure 5(b). Starting at 30s and ending approximately 45s after the start of the simulation there is a difference between the control voltage values in this instance and those shown in Figure 2(b). The controller deviates from its periodic behaviour to correct for the jump in the sensed position, and then returns to operate as though the fault is not there. This highlights the need for an FDI system to determine the presence, nature, and location of such a fault and allow the rover to compensate for it.

IV. Fault Detection using Output and Input Residuals

Residuals give a comparison between the expected behaviour of the system and the observed behaviour [2]. Therefore, if a fault occurs, any deviation from the desired performance of the system can be detected, and a diagnosis procedure implemented. The two types of residual used in



(a) Rover Serpentine Trajectory with a Sensor Fault at 30s

(b) Voltage Values with a Sensor Fault at 30s

Fig. 5: Rover Serpentine Trajectory and Corresponding Motor Voltages with a Sensor Fault at 30s

the course of this study are shown in the state-space model diagrams in Figure 6.

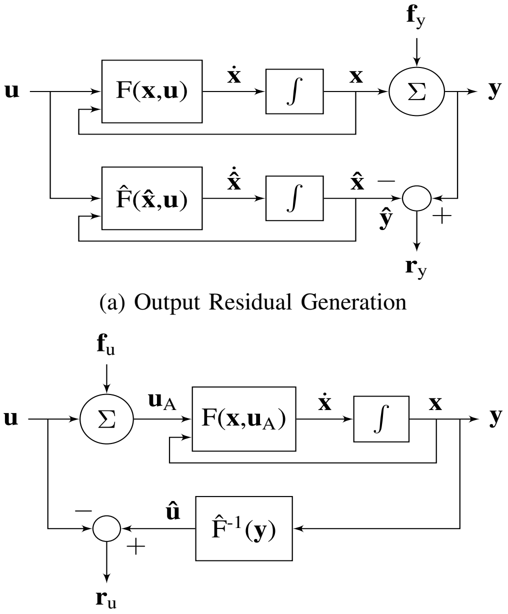
In this figure, $\hat{\mathbf{u}}$ is the input signal as estimated by $\hat{\mathrm{F}}^{-1}(\mathbf{y})$, the inverted mathematical model, and $\mathbf{r}_y$ and $\mathbf{r}_u$ are the output and input residuals respectively.

It can be seen that, using the calculations represented by Figures 6(a) and 6(b), $\mathbf{r}_y=\mathbf{y}-\hat{\mathbf{y}}=\mathbf{f}_y$ and $\mathbf{r}_u=\hat{\mathbf{u}}-\mathbf{u}=\mathbf{f}_u$ [12] meaning that the input and output residuals contain information about any faults occurring at their respective locations in the system.

A. *Output Residuals*

As has been previously stated, output residuals are defined as the difference between the output behaviour of the faulty system and a mathematical model of the fault-free scenario. Since multiple sensors would be implemented to measure the behaviour of the rover, there would be multiple residuals to choose from. It will be shown that monitoring each of these residuals would be useful for detecting and diagnosing faults in each of the sensors.

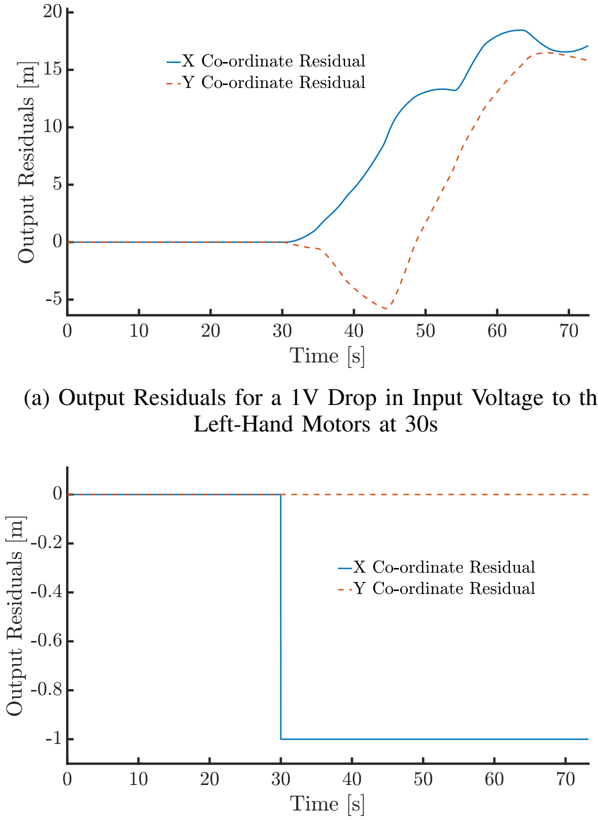
Figure 7(a) shows the X and Y co-ordinate residuals for the input voltage fault. Both plots show that the residuals are excited immediately after the injection of the fault, however they do not give much information about what kind of fault has occurred. The time-dependent behaviour, magnitude, and location of the fault cannot be determined from these graphs. It is clear that further analysis would be required to isolate faults at the input to the system.



(a) Output Residual Generation

(b) Input Residual Generation

Fig. 6: System and Model Comparison to Generate Output and Input Residuals [12]



(a) Output Residuals for a 1V Drop in Input Voltage to the Left-Hand Motors at 30s

(b) Output Residuals for a -1m Additive Fault in the X Co-ordinate Sensor at 30s

Fig. 7: Output Residuals for Input and Output Faults

For the sensor fault the X co-ordinate residual shown in Figure 7(b) mimics the fault exactly, both in magnitude and time-dependent behaviour. It is also worth noting that the only output residual excited by the fault in the X position sensor is the X position residual. This is in contrast to the voltage fault, which propagates through the system, affects multiple states, and excites multiple residuals.

The imitation of the fault and uniqueness of excitation make sensor faults relatively simple, not only to detect, but also to diagnose; if only one output residual is excited, it can be assumed that the fault is in the sensor relating to that residual. For the input fault, the detection and diagnosis of the fault is less trivial, requiring further analysis.

B. *Inverse Simulation*

As shown in Figure 6(b), the calculation of input residuals requires the inversion of the mathematical model. There are multiple methods of achieving this [13][14][15], however the method used in this study is a numerical integration-based Inverse Simulation (InvSim) algorithm, known as the Generic Inverse Simulation Algorithm (GENISA) [16][17]. The basic procedure for this algorithm is shown in Figure 8.

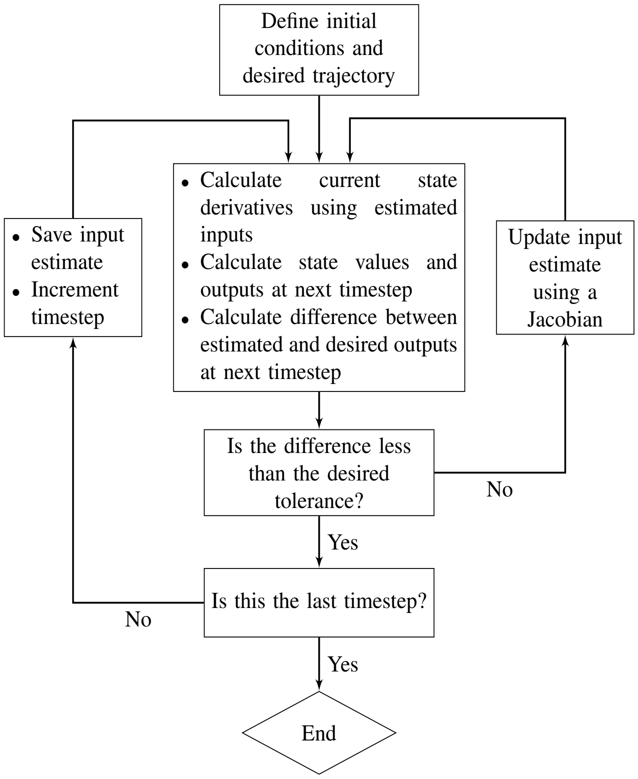

Define initial conditions and desired trajectory
• Calculate current state derivatives using estimated inputs
• Calculate state values and outputs at next timestep
• Calculate difference between estimated and desired outputs at next timestep
• Save input estimate
• Increment timestep
Update input estimate using a Jacobian
Is the difference less than the desired tolerance?
No
Yes
Is this the last timestep?
No
Yes
End

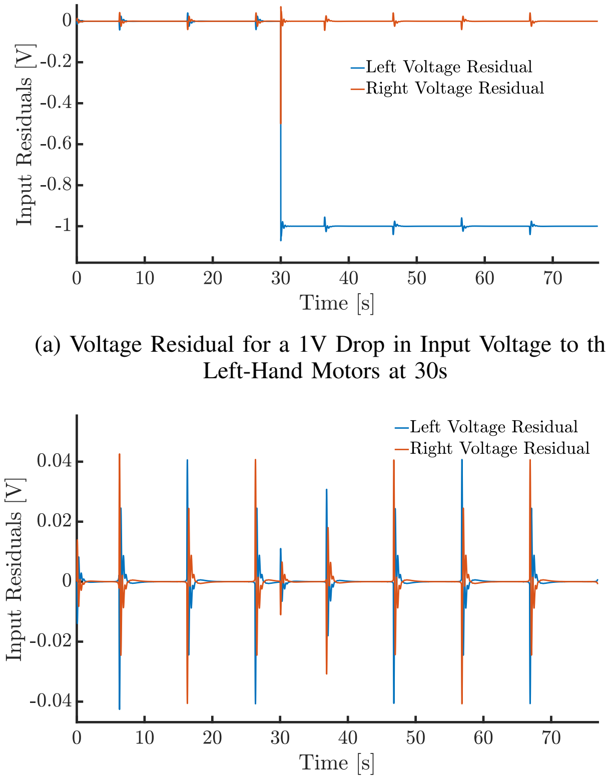
Fig. 8: GENISA Flow Diagram

After defining the initial conditions and the trajectory that the algorithm is aiming for, estimates for the input values are made and applied to the state-space equations of the system. This allows the state derivatives to be calculated, which can then be integrated numerically to provide the state values, a subset of which are chosen to be used as outputs from the system. These outputs are compared to the desired values determined by the predefined trajectory, and if the difference is greater than desired, the input estimates are corrected using a Jacobian matrix.

Once the input has been updated, the main steps are repeated until the calculated behaviour of the rover model is suitably accurate. When an accurate estimate of the required inputs has been made at the initial timestep, the algorithm saves the estimate and increments to the next timestep. This procedure is then repeated until the entire timeline of the trajectory has been covered.

C. *Input Residuals*

Using the GENISA, the rover model was inverted, to allow the input residuals to be calculated, and the faults described in Section III were applied to the simulated system. The input residuals for both fault scenarios are shown in Figure 9.



(a) Voltage Residual for a 1V Drop in Input Voltage to the Left-Hand Motors at 30s

(b) Voltage Residual for a -1m Additive Fault in the X Co-ordinate Sensor at 30s

Fig. 9: Voltage Residuals for Input and Output Faults

There are only two input residuals as there are only two control inputs for the rover modelled here: the left and right-hand motor voltages. In Figure 9(a), it can be seen that it is primarily the left-hand motor input residual that is affected by the presence of the voltage drop. Both residuals display small, oscillatory spikes which align with the moments where the rover begins to turn left or right and they both give a transient response at the instant of the fault occurrence. These artefacts are caused by the InvSim process failing to model rapid changes in the voltage instantaneously and their magnitude is related to both the simulation stepsize and the GENISA error threshold.

If these irregularities are ignored, the steady state values of the residuals can be used to evaluate the effect of the fault. As predicted, the left voltage residual matches the fault; it is persistent, abrupt, and stepwise, occurring at 30s and with a magnitude of -1V. The right voltage residual however remains at 0V. Once again, the uniqueness of the excited residual provides a means of isolating the fault to the input of the left-hand motors.

The X position sensor fault provides a different set of residual behaviours, shown in Figure 9(b). The small spikes corresponding to the rover's turning manoeuvres are once again evident and an even smaller disturbance is noted at 30s, the moment of fault injection. However, aside from this, there is no noticeable effect of the sensor fault on the input residuals. This is because the GENISA is driven by the measured accelerations of the rover, not the positions; a design choice made as it prevents instability in the algorithm [17]. Therefore these residuals are unsuitable for the detection of this sensor fault.

The presence of the oscillations does highlight the need for a detection threshold to prevent the FDI software from flagging a fault that was in fact simply noise from the InvSim procedure. This would also filter out any sensor noise introduced.

V. Comparison

The two types of residual tested in this work have proven to have different responses to faults at various locations in the system. For a fault affecting the input signal, it was shown that multiple output residuals were significantly excited, potentially allowing for the detection of the fault. However, since the relationship between the input voltages and the output residuals is not a simple one, and the dynamics of the system conceal the nature of the fault by dispersing the effect among multiple residuals, the determination of the type and location of the fault is not trivial. The input residuals give a much clearer picture. The time at which the input fault occurred, the nature of said fault, and the location can all be derived by noting that the input residual makes an abrupt jump to -1V at 30 seconds.

For a sensor fault, it was the output residuals that showed the clearer picture, with the residual corresponding to the faulty sensor replicating the behaviour of the fault exactly, while the other output residuals and the input residuals were left unaffected.

This matches the theory shown in Section IV, and shows that the behaviour of any faults and of the residuals calculated at the same location are equal. Hence, these residuals could be used to detect and characterise the faults as they occur.

VI. Conclusions

It is difficult to overstate the importance of rover reliability to PER missions and it is clear that intelligently designed health management systems could be an invaluable way of increasing this reliability. Using the principles of fault detection and isolation, it has been shown that residual generation has significant potential to be used as a basis upon which such a health management system could be built.

Using a mathematical model inversion technique, it has been shown that residuals can be generated to provide information about different locations within the system. Using a traditional model, output residuals were generated, and InvSim allowed for the calculation of input residuals.

Upon the application of faults at the input and output of the rover, and the examination of the residuals for each case, it was found that the output residuals mimic any corresponding sensor faults and input residuals do the same for any additive faults in the input channels. By comparing the two residual generation techniques, it has been shown that they could be used in parallel to detect singular, additive faults in both the voltage inputs and the sensor outputs of the system.

Future work will be focused on how robust and sensitive the residual calculation method is when noise and disturbances are introduced into the system at various locations. More in-depth investigation will also be performed into the diagnosis and fault recovery procedures enabled by this methodology.

References

[1] A. Ellery. *Planetary Rovers: Robotic Exploration of the Solar System*. Springer; 2016.

[2] S. Simani, C. Fantuzzi, R.J. Patton. *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques*. Springer; 2006; DOI:10.1007/978-1-4471-3829-7.

[3] M.L. Ireland, K.J. Worrall, R. Mackenzie, T. Flessa, E.W. McGookin, D. Thomson. *A Comparison of Inverse Simulation-Based Fault Detection in a Simple Robotic Rover with a Traditional Model-Based Method*. World Academy of Science, Engineering and Technology, International Journal of Mechanical and Mechatronics Engineering; 2017; 11(3); pp. 607-615.

[4] G. Genta. *Conventional and Slip Steering for Multi-Wheel Planetary Rovers*. Atti dell'Accademia delle Scienze di Torino; 2010 March 10; 34.

[5] T. Flessa, E.W. McGookin, and D.G. Thomson. *Taxonomy, Systems Review and Performance Metrics of Planetary Exploration Rovers*. ICARCV'14, pp. 1554-1559, 2014 Dec 10-12.

[6] R.C. Dorf, R.H. Bishop. *Modern Control Systems. 12th Edition*. Chennai, India: Pearson; 2015; pp. 496-507

[7] T.I. Fossen, M. Brievik, R. Skjtne. *Line-of-Sight Path Following of Underactuated Marine Craft*. 6th IFAC MCMC Conference Proceedings; 2003; 36(21); pp. 211-216.

[8] T.I. Fossen. *Guidance and Control of Ocean Vehicles*. Wiley and Sons Ltd; 1994.

[9] E.W. McGookin. *Optimisation of Sliding Mode Controllers for Marine Applications: a Study of Methods and Implementation Issues*[dissertation]. Glasgow: University of Glasgow; 1997.

[10] J. Carlson, R. Murphy. *Reliability Analysis of Mobile Robots*. Proceedings of the 1003 IEEE International Conference on Robotics and Automation; 2003 September 14-19; pp. 274-281.

[11] R. Isermann. *Fault Diagnosis Systems –An Introduction from Fault Detection to Fault Tolerance*. 64283 Darmstadt, Germany: Springer; 2006.

[12] M.L. Ireland, R. Mackenzie, T. Flessa, K. Worrall, D. Thomson, E. McGookin. *Inverse Simulation as a Tool for Fault Detection and Isolation in Planetary Rovers*. ESA GNC 2017 May 29 - Jun 2.

[13] D.J. Murray-Smith. *Feedback methods for inverse simulation of dynamic models for engineering systems applications*. Mathematical and Computer Modelling of Dynamical Systems (2011); 17 (5); pp. 515-541. ISSN 1744-5051

[14] M.L. Ireland, T. Flessa, D. Thomson, E. McGookin. *A Comparison of Non-Linear Dynamic Inversion and Inverse Simulation*. Journal of Guidance, Control, and Dynamics; 2017 May 22; 40(12); pp. 3307-3312.

[15] D.J. Murray-Smith. *The inverse simulation approach: a focused review of methods and applications*. Mathematics and Computers in Simulation 53; 2000 July 28; pp. 239-247.

[16] R.A. Hess, C. Gao, S.H. Wang. *Generalized Technique for Inverse Simulation Applied to Aircraft Maneuvers*. AIAA J. Guidance Control and Dynamics. 1991; 14 (5); pp. 920-926.

[17] S. Rutherford, D.G. Thomson. *Improved Methodology for Inverse Simulation*. Aeronautical Journal; 1996 March; pp. 79-86.