Adaptive Inner-Loop Rover Control

Nilesh Kulkarni & Corey Ippolito
QSS Group, Inc., NASA Ames Research Center, Moffett Field, CA
nilesh@email.arc.nasa.gov
cippolito@mail.arc.nasa.gov

Kalmanje Krishnakumar
NASA Ames Research Center, Moffett Field, CA
Kalmanje.Krishnakumar@nasa.gov

Khalid M. Al-Ali
Carnegie Melon University West Campus, NASA Ames Research Center, Moffett Field, CA
alali@cmu.edu

Abstract

Adaptive control technology is developed for the inner-loop speed and steering control of the MAX Rover. MAX, a CMU developed Rover, is a compact low-cost 4-wheel drive, 4-wheel steer (double Ackerman), with high-clearance agile durable chassis. It is outfitted with sensors and electronics that make it ideally suited for supporting research relevant to intelligent tele-operation, and as a low-cost autonomous robotic test bed and appliance. The control design consists of a feedback linearization based controller with a proportional-integral (PI) feedback that is augmented by an online adaptive neural network. The adaptation law has guaranteed stability properties for safe operation. The control design is retrofit in nature so that it fits below the outer-loop path planning algorithms. Successful hardware implementation of the controller is illustrated for several scenarios consisting of actuator failures and modeling errors in the nominal design.

1. Introduction

The Mars Rover Program illustrated the critical role of Rovers in exploration systems for assessing risks, gathering scientific information, and preparing for eventual human missions [1]. While significant attention has been given towards the development of their autonomous path planning and scheduling algorithms, the inner-loop control laws still consist of basic Proportional-Integral-Derivative (PID) algorithms. The PID control gains typically need to be hand-tuned for specific operating conditions. These include the Rover mass and inertia characteristics, along with the operating surface characteristics. Since the controller is static in design, performance is traded for robustness. In addition, explicit fault identification, classification, and isolation are needed for system failures. This is a feasible mode of design for specific operations involving a few Rovers. However, missions involving multiple Rovers, or missions desiring higher performance from a single Rover, dictate a clear need for more autonomy at the inner-loop controller level.

The Adaptive Control and Evolvable Systems (ACES) group at NASA Ames Research Center in collaboration with industry and academia has been actively developing adaptive control technologies for different aircraft platforms. The Georgia Tech/NASA Ames Intelligent-flight-control (IFC) program has been successfully implemented for controlling different flight platforms, ranging from small un-manned vehicles (UAVs) to fighter aircrafts to military and civilian transport planes [2-3]. The control architecture adapts for modeling uncertainties, as well as system and actuator failures. A recently concluded flight test on an F-15 research aircraft at the Dryden Flight Research Center has highlighted the success of this program. A critical feature of the IFC architecture has been its plug-n-play nature as illustrated by its implementation on different classes of aircraft

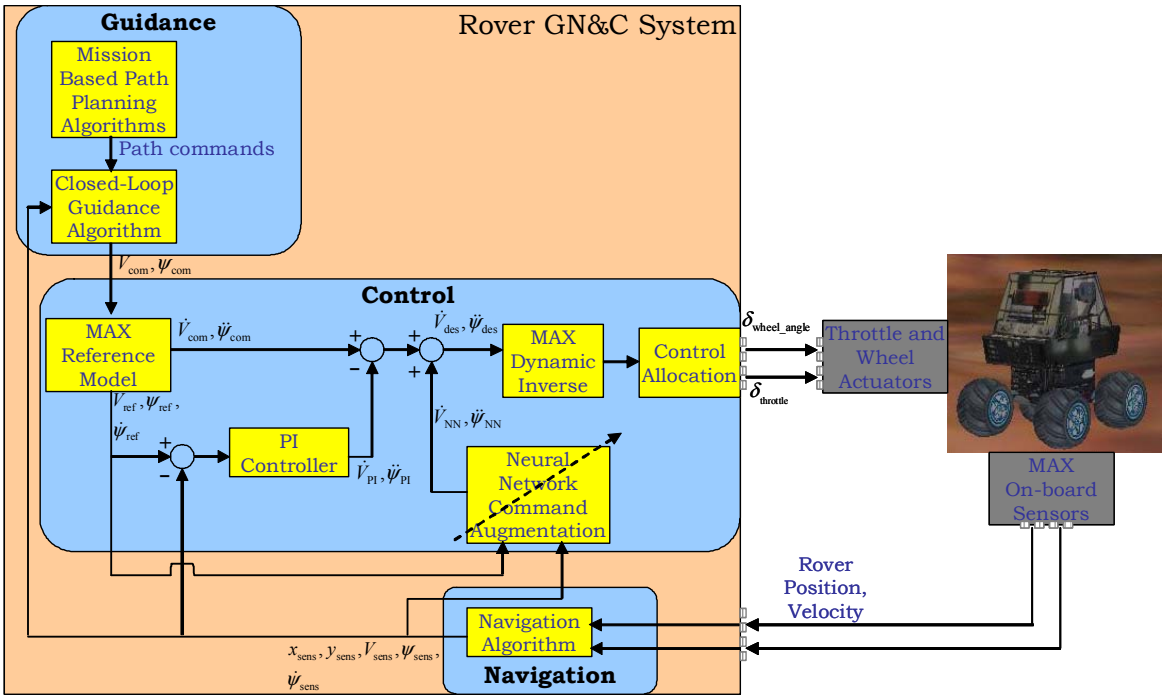

Figure 1. IFC-like Rover Guidance, Navigation, and Control Architecture

platforms. The goal of this research is to investigate the feasibility of application of the IFC architecture to space systems. Towards realizing this goal, the IFC architecture is implemented for the inner-loop adaptive control of the MAX Rover. Section 2 outlines the details of the implementation for the Rover system. Section 3 discusses Rover modeling for simulation and control system development. Section 4 presents the simulation results, and section 5, the hardware results. Finally section 6 summarizes the conclusions of this research effort and provides directions for future work.

2. Rover guidance, navigation and control architecture

Figure 1 outlines the Rover guidance, navigation, and control (GN&C) system. The guidance system consists of mission-specific path planning algorithms that compute way-points to generate the open-loop trajectory. The guidance system also includes a closed-loop algorithm, which, based on the current Rover position, provides velocity and heading commands to align the Rover with the open-loop trajectory. The inner-loop control system computes the Rover actuator (throttle and steering) settings to meet these velocity and heading commands.

The commanded values given by the guidance system do not take into account the dynamics of the Rover and can lead to actuator saturation and poor command following. The command values are therefore smoothed using a reference model. The velocity is smoothed by a first order reference model while the heading uses a second order reference model.

$$\frac{V_{\text{rover\_ref}}(s)}{V_{\text{rover\_com}}(s)} = \frac{1}{\tau_v s + 1} \tag{1}$$

$$\frac{\psi_{\text{rover\_ref}}(s)}{\psi_{\text{rover\_com}}(s)} = \frac{\omega_{n\psi}^2}{s^2 + 2\xi_\psi \omega_{n\psi} s + \omega_{n\psi}^2} \tag{2}$$

The velocity reference model provides reference velocity and reference acceleration. The second order heading reference model provides reference heading, reference yaw rate and reference yaw acceleration. These reference values along with the sensed states of the Rover are used in a proportional-integral (PI) feedback to compute the desired translational and yaw acceleration.

$$\begin{aligned} \dot{V}_{\text{rover\_des}} &= \dot{V}_{\text{rover\_ref}} + K_{\text{Pv}}\left(V_{\text{rover\_ref}} - V_{\text{rover\_sens}}\right) \\ &+ K_{\text{Iv}}\int\left(V_{\text{rover\_ref}} - V_{\text{rover\_sens}}\right)dt \end{aligned} \tag{3}$$

$$\begin{aligned}\dot{r}_{\text{rover\_des}} &= \dot{r}_{\text{rover\_ref}} + K_{\text{Pr}}\left(r_{\text{rover\_ref}} - r_{\text{rover\_sens}}\right) \\ &\quad + K_{\text{Ir}}\int\left(r_{\text{rover\_ref}} - r_{\text{rover\_sens}}\right)dt \\ &= \dot{r}_{\text{rover\_ref}} + K_{\text{Pr}}\left(r_{\text{rover\_ref}} - r_{\text{rover\_sens}}\right) \\ &\quad + K_{\text{Ir}}\left(\psi_{\text{rover\_ref}} - \psi_{\text{rover\_sens}}\right)\end{aligned} \tag{4}$$

These desired values of translational and yaw accelerations are used to invert the dynamical model of the Rover to compute the corresponding throttle and steering actuator values. For a Rover model given as

$$\begin{aligned}\begin{bmatrix}\dot{V}_{\text{rover}} \\ \dot{r}_{\text{rover}}\end{bmatrix} &= \mathbf{f}\left(V_{\text{rover}}, r_{\text{rover}}, \psi_{\text{rover}}\right) \\ &\quad + \mathbf{g}\left(V_{\text{rover}}, r_{\text{rover}}, \psi_{\text{rover}}\right)\begin{bmatrix}\delta_{\text{th}} \\ \delta_{\text{wh}}\end{bmatrix},\end{aligned} \tag{5}$$

the throttle and steering actuator settings are computed by equating the desired accelerations with the model.

$$\begin{bmatrix}\dot{V}_{\text{rover}} \\ \dot{r}_{\text{rover}}\end{bmatrix} = \begin{bmatrix}\dot{V}_{\text{rover\_des}} \\ \dot{r}_{\text{rover\_des}}\end{bmatrix} \tag{6}$$

$$\Rightarrow \begin{bmatrix}\delta_{\text{th}} \\ \delta_{\text{wh}}\end{bmatrix} = \mathbf{g}^{-1}\left(V_{\text{rover}}, r_{\text{rover}}, \psi_{\text{rover}}\right)\left\{\begin{bmatrix}\dot{V}_{\text{rover\_des}} \\ \dot{r}_{\text{rover\_des}}\end{bmatrix} - \mathbf{f}\left(V_{\text{rover}}, r_{\text{rover}}, \psi_{\text{rover}}\right)\right\} \tag{7}$$

Equation (6), and Eqs. (3-4) provide the corresponding error dynamics.

$$\begin{aligned}&\left(\dot{V}_{\text{rover\_sens}} - \dot{V}_{\text{rover\_ref}}\right) + K_{\text{Pv}}\left(V_{\text{rover\_sens}} - V_{\text{rover\_ref}}\right) \\ &+ K_{\text{Iv}}\int\left(V_{\text{rover\_sens}} - V_{\text{rover\_ref}}\right)dt = 0\end{aligned} \tag{8}$$

$$\begin{aligned}&\left(\dot{r}_{\text{rover\_sens}} - \dot{r}_{\text{rover\_ref}}\right) + K_{\text{Pr}}\left(r_{\text{rover\_sens}} - r_{\text{rover\_ref}}\right) \\ &+ K_{\text{Ir}}\int\left(r_{\text{rover\_ref}} - r_{\text{rover\_sens}}\right)dt = 0\end{aligned} \tag{9}$$

The choice of the proportional and integral gains provides the desired error transient characteristics.

2.1. Adaptive control formulation

The feedback linearization approach outlined in the previous sub-section assumes exact knowledge of the system dynamics. In cases of uncertainties in the dynamic modeling or changes in the dynamics during operation due to faults or degradation of the hardware, the controller needs to be adapted for providing the same level of performance.

The GN&C architecture outlined in figure 1 provides this adaptation by augmenting the desired acceleration signal, which goes into the dynamic inversion (Eq. 7). This augmenting signal is parameterized using an appropriate function-approximator. In this case a neural network is chosen to provide this augmentation signal due to its universal function approximating capability. The weights of the neural network are then adapted to minimize the transient error.

The desired acceleration with the neural network augmentation command is given as

$$\begin{aligned}\dot{V}_{\text{rover\_des}} &= \dot{V}_{\text{rover\_ref}} + K_{\text{Pv}}\left(V_{\text{rover\_ref}} - V_{\text{rover\_sens}}\right) \\ &\quad + K_{\text{Iv}}\int\left(V_{\text{rover\_ref}} - V_{\text{rover\_sens}}\right)dt - \dot{V}_{\text{NN}}\end{aligned} \tag{10}$$

$$\begin{aligned}\dot{r}_{\text{rover\_des}} &= \dot{r}_{\text{rover\_ref}} + K_{\text{Pr}}\left(r_{\text{rover\_ref}} - r_{\text{rover\_sens}}\right) \\ &\quad + K_{\text{Ir}}\int\left(r_{\text{rover\_ref}} - r_{\text{rover\_sens}}\right)dt - \dot{r}_{\text{NN}}\end{aligned} \tag{11}$$

Let the nominal model of the Rover be given as

$$\begin{aligned}\begin{bmatrix}\dot{V}_{\text{rover}} \\ \dot{r}_{\text{rover}}\end{bmatrix}_{\text{model}} &= \hat{\mathbf{f}}\left(V_{\text{rover}}, r_{\text{rover}}, \psi_{\text{rover}}\right) \\ &\quad + \hat{\mathbf{g}}\left(V_{\text{rover}}, r_{\text{rover}}, \psi_{\text{rover}}\right)\begin{bmatrix}\delta_{\text{th}} \\ \delta_{\text{wh}}\end{bmatrix},\end{aligned} \tag{12}$$

and let the actual system dynamics be given by Eq. (5). The error in the system dynamics is represented as

$$\begin{aligned}\begin{bmatrix}\varepsilon_V \\ \varepsilon_r\end{bmatrix} &= \begin{bmatrix}\dot{V}_{\text{rover}} \\ \dot{r}_{\text{rover}}\end{bmatrix} - \begin{bmatrix}\dot{V}_{\text{rover}} \\ \dot{r}_{\text{rover}}\end{bmatrix}_{\text{model}} \\ &= \begin{bmatrix}\dot{V}_{\text{rover}} \\ \dot{r}_{\text{rover}}\end{bmatrix} - \left\{\begin{matrix}\hat{\mathbf{f}}\left(V_{\text{rover}}, r_{\text{rover}}, \psi_{\text{rover}}\right) \\ + \hat{\mathbf{g}}\left(V_{\text{rover}}, r_{\text{rover}}, \psi_{\text{rover}}\right)\begin{bmatrix}\delta_{\text{th}} \\ \delta_{\text{wh}}\end{bmatrix}\end{matrix}\right\}\end{aligned} \tag{13}$$

The control is computed by inverting the nominal model as in Eq. (7)

$$\begin{bmatrix}\delta_{\text{th}} \\ \delta_{\text{wh}}\end{bmatrix} = \hat{\mathbf{g}}^{-1}\left(V_{\text{rover}}, r_{\text{rover}}, \psi_{\text{rover}}\right)\left\{\begin{bmatrix}\dot{V}_{\text{rover\_des}} \\ \dot{r}_{\text{rover\_des}}\end{bmatrix} - \hat{\mathbf{f}}\left(V_{\text{rover}}, r_{\text{rover}}, \psi_{\text{rover}}\right)\right\} \tag{14}$$

Substituting Eq. (13) in Eq. (14) gives

$$\begin{bmatrix} \varepsilon_V \\ \varepsilon_r \end{bmatrix} = \begin{bmatrix} \dot{V}_{\text{rover}} \\ \dot{r}_{\text{rover}} \end{bmatrix} - \begin{bmatrix} \dot{V}_{\text{rover\_des}} \\ \dot{r}_{\text{rover\_des}} \end{bmatrix}, \tag{15}$$

which using Eqs. (10-11) gives

$$\begin{bmatrix} \varepsilon_V \\ \varepsilon_r \end{bmatrix} = \begin{bmatrix} \dot{V}_{\text{rover}} \\ \dot{r}_{\text{rover}} \end{bmatrix} - \begin{bmatrix} \dot{V}_{\text{rover\_ref}} + K_{\text{Pv}}\left(V_{\text{rover\_ref}} - V_{\text{rover\_sens}}\right) \\ +K_{\text{Iv}}\int\left(V_{\text{rover\_ref}} - V_{\text{rover\_sens}}\right)dt - \dot{V}_{\text{NN}} \\ \dot{r}_{\text{rover\_ref}} + K_{\text{Pr}}\left(r_{\text{rover\_ref}} - r_{\text{rover\_sens}}\right) \\ +K_{\text{Ir}}\int\left(r_{\text{rover\_ref}} - r_{\text{rover\_sens}}\right)dt - \dot{r}_{\text{NN}} \end{bmatrix}$$

$$\begin{bmatrix} \varepsilon_V - \dot{V}_{\text{NN}} \\ \varepsilon_r - \dot{r}_{\text{NN}} \end{bmatrix} = \begin{bmatrix} \left(\dot{V}_{\text{rover}} - \dot{V}_{\text{rover\_ref}}\right) + K_{\text{Pv}}\left(V_{\text{rover}} - V_{\text{rover\_ref}}\right) \\ +K_{\text{Iv}}\int\left(V_{\text{rover}} - V_{\text{rover\_ref}}\right)dt \\ \left(\dot{r}_{\text{rover}} - \dot{r}_{\text{rover\_ref}}\right) + K_{\text{Pr}}\left(r_{\text{rover}} - r_{\text{rover\_ref}}\right) \\ +K_{\text{Ir}}\int\left(r_{\text{rover}} - r_{\text{rover\_ref}}\right)dt \end{bmatrix} \tag{16}$$

Equation (16) implies that if the neural network augmentation provides the error between the nominal model and the actual system, the velocity and the yaw rate error dynamics will again match with the desired second order response characteristics.

The neural network can be parameterized in any convenient form. For the radial basis function neural network, the augmenting acceleration signals are given as

$$\begin{bmatrix} \dot{V}_{\text{NN}} \\ \dot{r}_{\text{NN}} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_V{}^T \Phi_V\left(V_{\text{rover}}, r_{\text{rover}}, \psi_{\text{rover}}\right) \\ \mathbf{W}_r{}^T \Phi_r\left(V_{\text{rover}}, r_{\text{rover}}, \psi_{\text{rover}}\right) \end{bmatrix}, \tag{17}$$

where Φ_V and Φ_r represent the appropriately chosen basis functions. The weight vectors, $\mathbf{W}_V$ and $\mathbf{W}_r$, are adapted based on the Lyapunov analysis given in [3] that guarantees boundedness of the system error and the weight vectors.

$$\begin{aligned} \dot{\mathbf{W}}_V &= -\gamma_V \mathbf{e}_V \mathbf{P}_V \Phi_V \\ \dot{\mathbf{W}}_r &= -\gamma_r \mathbf{e}_r \mathbf{P}_r \Phi_r \end{aligned} \tag{18}$$

The coefficients, γ_V and γ_r, represent the learning rates in the weight update law, $\mathbf{P}_V$ and $\mathbf{P}_r$, the weighting coefficients in the chosen Lyapunov functions for each of the velocity and heading channels, and $\mathbf{e}_V$ and $\mathbf{e}_r$, the velocity and heading channel errors defined as

$$\begin{aligned} \mathbf{e}_V &= V_{\text{rover}} - V_{\text{rover\_ref}} \\ \mathbf{e}_r &= r_{\text{rover}} - r_{\text{rover\_ref}} \end{aligned} \tag{19}$$

This completes the adaptive control design for the Rover velocity and yaw rate channels. The choice of the radial basis functions used for designing the neural network dictates the quality of the adaptation and the error transients.

3. Rover modeling

The modeling of the MAX Rover was undertaken for supporting two key aspects of the control design. The control system uses dynamic inversion to compute the actuator commands. This requires a nominal model of the Rover as given by Eq. (12) that can be inverted quickly in real-time. The control system also needs to be tested on a high-fidelity simulation environment before hardware experimentation. The high-fidelity model needs to be sophisticated and incorporate Rover physical characteristics along with details of the operating surface. To satisfy these needs, a high fidelity model is developed for simulation purposes, and a comparative low fidelity model developed to be inverted in the control system.

3.1. High-fidelity Rover model

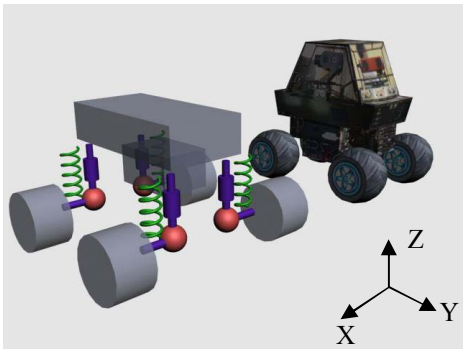

Figure 2. High-fidelity Rover model

Figure 2 illustrates modeling the MAX Rover using rigid bodies connected by springs and dampers. The Rover is built as a 5 body system consisting of a chassis and 4 wheels. The chassis is represented by a uniform mass rectangular box and the wheels by uniform mass cylinders. The 5 body motion is characterized by various constraints. The relative rotation of the chassis with respect to the wheels in the

local Z-axis is constrained by the maximum steering angle in both directions. Similarly the translation of the chassis relative to the wheels in its local Z-axis is constrained by the maximum suspension travel in both directions. The wheels are constrained to rotate along the local X-axis alone. The friction between the wheels and the ground is modeled by the pyramid approximation to the Coulomb friction model. This model is built and simulated using an indigenously developed physics engine that can also render the Rover operation graphically.

3.2. Low-fidelity Rover model

The high-fidelity model described in the previous sub-section is difficult to be described in a closed-loop form, which can be inverted in real time. A comparatively lower fidelity model is therefore developed that captures the key physics of the system. The Rover model in a relative planar straight line mode of operation without hard cornering can be described using the following equations.

$$\begin{aligned}
\dot{x}_{\text{rover}} &= V_{\text{rover}} \cos\psi_{\text{rover}} \\
\dot{y}_{\text{rover}} &= V_{\text{rover}} \sin\psi_{\text{rover}} \\
\dot{V}_{\text{rover}} &= -K_{dv} V_{\text{rover}} + K_{tf}\delta_{\text{th_front}} + K_{tr}\delta_{\text{th_rear}} \\
\dot{\psi}_{\text{rover}} &= r_{\text{rover}} \\
\dot{r}_{\text{rover}} &= -K_{dr} r_{\text{rover}} + K_{wf} V_{\text{rover}}\delta_{\text{wh_front}} + K_{wr} V_{\text{rover}}\delta_{\text{wh_rear}}
\end{aligned} \tag{20-24}$$

The front and rear wheel steering and actuator settings are allocated equally thereby reducing the dimension of control inputs to 2.

$$\begin{aligned}
\delta_{\text{th_front}} &= \delta_{\text{th_rear}} = \delta_{\text{th}} \\
\delta_{\text{wh_front}} &= \delta_{\text{wh_rear}} = \delta_{\text{wh}}
\end{aligned} \tag{25-26}$$

Equations (22,24) and (25-26) are used in inverting the dynamics and computing the control inputs given the desired translational and yaw accelerations.

4. Simulation results

The Rover GN&C system was simulated using the high-fidelity model. Figure 3 outlines the reference path that needs to be followed by the Rover. The rectangular path with rounded corners is assigned velocity profiles as shown in the figure.

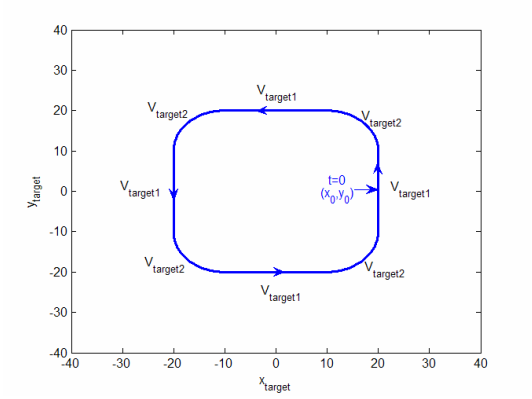

Figure 3. Reference path used for simulation studies

The control system is developed in the Simulink environment. A shared memory interface is set up to transfer control inputs from Simulink to the C++ based high fidelity simulation, and system state variables from the simulation back to the Simulink based control system.

4.1. Adapting for incorrect friction modeling

A common source of Rover modeling error Rover arises from its operating surface friction. Higher than assumed friction will lead to a lagging trajectory while lower than assumed friction will lead to excessive throttle and braking. Figure 4 compares the resulting path profile with and without adaptation for an incorrectly used value of friction coefficient in the dynamic inverse. It can be observed that since the assumed value of friction is higher than its actual value, the path without adaptation results in a forward and backward motion. The path with adaptation, however, results in a smooth capture and following of the reference path.

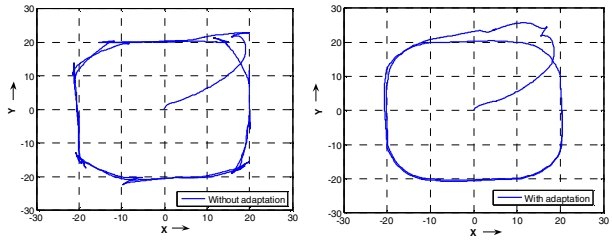

Figure 4. Rover path profile without and with adaptation

Figure 5 compares the Rover velocity tracking performance without and with adaptation. The substantial improvement in following the reference profiles is clearly seen. Similar comparison is

presented in figure 6 for the Rover yaw angle and yaw rate profiles without and with adaptation. Finally figure 7 compares the control inputs without and with adaptation.

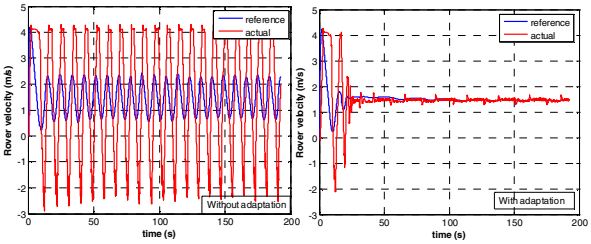

Figure 5. Rover velocity tracking without and with adaptation.

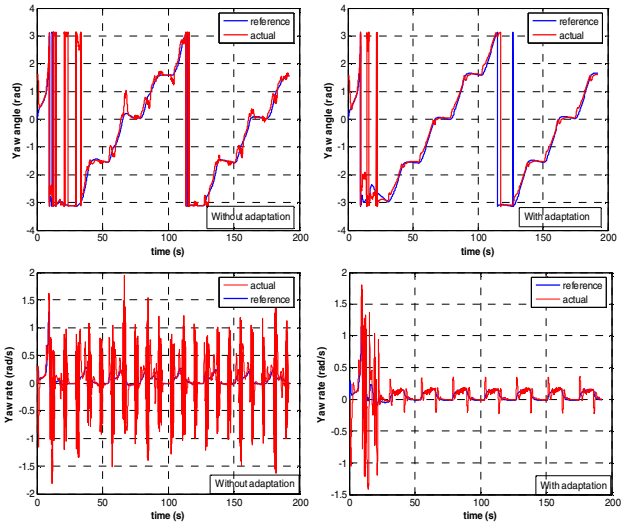

Figure 6. Rover yaw angle and yaw rate tracking without and with adaptation.

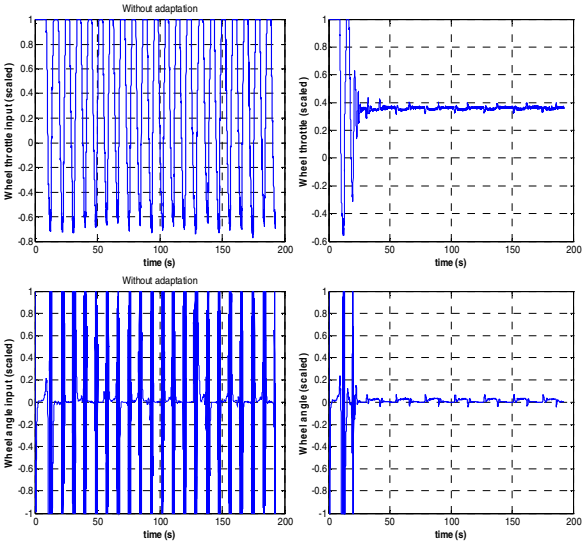

Figure 7. Rover wheel steering and throttle inputs without and with adaptation.

5. Hardware testing

Figure 8. The MAX Rover hardware platform

Figure 8 presents the MAX hardware platform. The Rover has an on-board CPU with the Windows 2000 operating system. The GN&C system is implemented using an indigenously developed modular Reflection architecture. This embedded software transports information between each of the software and hardware modules. The Rover state information is sensed using the on-board Crossbow NAV420 integrated navigation system and global positioning system (INS/GPS) unit. The control inputs are generated by the Matlab- Simulink code running on the Central Processing Unit (CPU) on the MAX, and are transported to the corresponding actuators by the Reflection architecture. A ground station is set-up with an 802.11g wireless connection between the ground station laptop and the Rover CPU. The ground station is used for remotely initiating and monitoring the performance of the hardware tests.

5.1. System Identification

Several tests were run to identify the coefficients of the assumed linear model of the Rover. These tests were separated in two sets: one for the Rover velocity channel, and the other for the Rover heading channel. Figure 9 illustrates the throttle command given to identify the velocity dynamics and the resulting velocity profile while the steering command is fixed to zero.

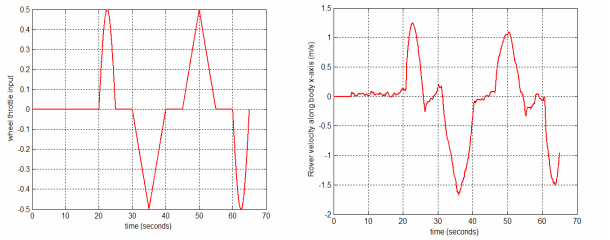

Figure 9. Identifying the Rover velocity dynamics

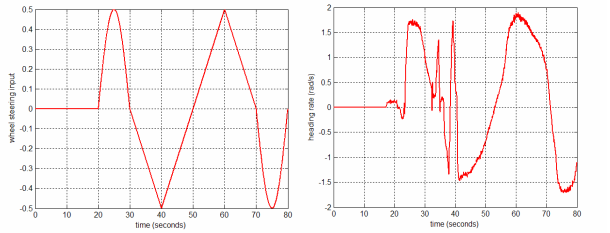

Figure 10. Identifying the Rover yaw rate dynamics

For identifying the coefficients of the heading dynamics, the throttle value was set at 0.35. Figure 10 illustrates the commanded wheel angle and the resulting heading rate and velocity profile. Choosing different starting and end points in the tests, the coefficients in Eqs. (22) and (24) for each of these time windows were identified using a least squares fit. The different coefficient sets thus obtained were averaged since a single set of coefficients is used in the dynamic inversion. This identified set of coefficients is computed to be

$$\begin{aligned} K_{dv} &= 0.8 \\ \left(K_{tf} + K_{tr}\right) &= 2.4 \\ K_{dr} &= 0.35 \\ \left(K_{wf} + K_{wr}\right) &= 1.6 \end{aligned} \tag{27}$$

The identified values for both the channels differed slightly depending on the particular choice of the selected data points. This variation was observed to be lesser on the heading channel than the velocity channel. This observation is attributed to two reasons. The velocity channel data is given by the INS unit that is rated for flight velocities and hence produces substantial estimation errors for the relatively low Rover velocity values. The second reason is that the velocity dynamics is nonlinear and the variation observed is related to the different values of the nominal velocity and heading rate.

Given the errors in the velocity channel, the adaptive control design is tested on the heading channel for the failed actuator scenario. The rear wheel steering is fixed at zero and the commanded trajectory is followed by the front wheel steering alone.

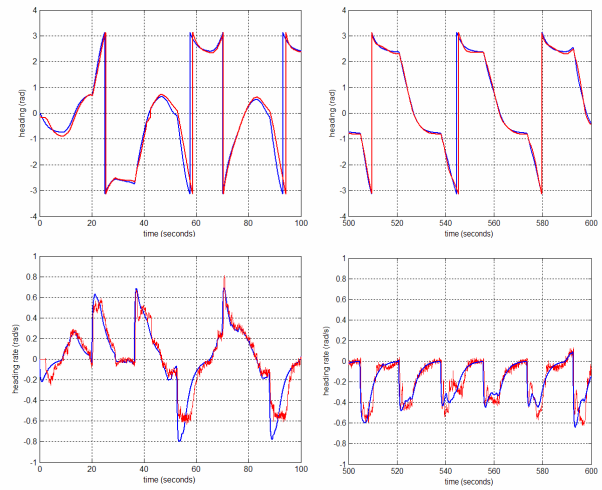

Figure 11. Heading and yaw rate tracking with adaptation (blue –reference, red –actual)

Figure 11 illustrates the behavior of the yaw angle tracking for the first 100 seconds and the last 100 seconds of the test to illustrate the effects of adaptation. Similarly, figure 12 illustrates the yaw rate tracking performance that is again presented for the first 100 seconds and the last 100 seconds of the test. It can be noted that without adaptation, the Rover yaw angle lags the desired value. Similarly the front wheel steering alone is unable to provide the desired yaw rates during the early phase of adaptation. Both these variables show substantial tracking improvement during the last 100 seconds thereby illustrating successful adaptation.

6. Conclusion and future work

In this research we initiated the development of the adaptive Rover-control technology based on the IFC legacy program. The adaptive control algorithm was developed and simulated for the Rover on a high-fidelity simulation along with hardware testing on the

MAX platform. The results suggest critical benefits of this adaptive control technology for handling modeling uncertainties as well as failed actuators. The hardware testing suggests a need for using nonlinear dynamics in the inversion. Existing literature in the vehicle dynamics field is being reviewed towards this effort [4]. The Rover velocity and position sensing is also being improved by incorporating individual wheel speed measurements and differential global positioning system (DGPS) measurements.

7. References

[1] http://marsrovers.jpl.nasa.gov

[2] Kaneshige, J and Gundy-Burlet, K, "Integrated Neural Flight and Propulsion Control System," AIAA-2001-4386, *Proceedings of the AIAA Guidance, Navigation and Conference*, August 2001.

[3] Rysdyk, R. T., and Calise, A. J., "Fault Tolerant Flight Control via Adaptive Neural Network Augmentation," AIAA 98-4483, *Proceedings of the AIAA Guidance, Navigation and Conference*, August 1998.

[4] Lim, E.H.M, and Hedrick, K., "Lateral and Longitudinal Vehicle Control Coupling for Automated Vehicle Operation, *Proceedings of the American Control Conference*, Volume 5, pp. 3676-3680, July 1999.