MRL Control JSON Protocol

Martin Jay McKee

Abstract

This document outlines the format of the JSON messages that are passed between the new rovers and the control system. Note, these formats are preliminary and are subject to change.

1 Frame Formats

IT WOULD BE GOOD TO ADD FRAMES TO HANDLE THE ROVER LOCATION SO THAT IT IS POSSIBLE TO DISPLAY A MAP ON THE CLIENT. WHILE SUCH A MAP IS A NICE FEATURE, HOWEVER, IT IS HIGHLY LIKELY TO DRIFT GIVEN NO ABSOLUTE POSITIONING SYSTEM AND, AS SUCH, IT SHOULD BE RESET FROM TIME TO TIME. THIS IS AN ADVANCED FEATURE AND NOT AT ALL IMPORTANT FOR A FIRST RUN OF THE SYSTEM. STILL, THIS CAN POTENTIALLY BE BASED, IN PART, UPON THE ROVER STATUS MESSAGES.

1.1 Error Format

An error may be sent as a subframe in status frames. It is never sent alone and, therefore, is missing some of the basic fields of the remaining frames. An error frame contains information to describe the error as well as to command a response to it.

Field	Type	Description
name	String	This field identifies the name of the error. It is upper-case and contains no w
level	String(enum)	This is a string from the list [Debug, Info, Warning, Error, Fatal].
message	String	This is a human-readable description of the error for display on the cli
action	List of Frames	This defines the actions to be taken by the client in response to the er

1.2 Basic Format

Each frame has a basic set of fields. These fields determine the type of the frame as well as give it a unique idenitifer.

Field	Type	Description
type	String	This field identifies the type of frame. It is a lower-case string without white
id	Unsigned Integer	This field is a unique id for the frame

1.3 Server Frames

These are frames that are sent out from the server to the client. TODO: ADD FRAMES WHICH CAN RETURN MORE ROVER STATUS INFORMATION. FOR INSTANCE, IT WOULD BE NICE TO HAVE THE ABILITY TO RETURN OBSTACLE DATA TO THE CONTROL SYSTEM.

1.3.1 Rover Config (rover)

This frame is required to correctly configure the control system. It contains information about the physical characteristics of the rover which are used to set distance and velocity limits within the interface. This frame will be sent when a connection is established or when the values need to be updated.

Field	Type	Description
wheelbase	Positive Float	Length between the center of the front and back wheels (m)
track	Positive Float	Distance between the center of the left and right wheels (m)
vlimit	Positive Float	Maximum velocity (m/s)
dlimit	Positive Float	Maximum distance to travel in a single command (m)
alimit	Positive Float	Maximum linear acceleration (m/s2)
wlimit	Positive Float	Maximum angular velocity (deg/s) ?? CALCULATED
wdotlimit	Positive Float	Maximum angular acceleration (deg/s2) ?? CALCULATED
pmin	Float [0, 100]	Percentage of minimum “power”
pmax	Float [0, 100]	Percentage of maximum ”power“

1.3.2 Client Config (client)

This frame is required to correctly configure the control system client. It contains information about the allowable control messages and control interface formats. This frame will be sent when a connection is established or when the values need to be updated.

Field	Type	Description
cid	Unsigned Integer	The client id value for the control client to use
pivotable	Boolean	Set to true if the rover supports in-place pivot
advancedmode	Boolean	Set to true if controlling the rover in advanced mode is supported

1.3.3 Command Status (cmdstatus)

This status frame returns the current progress of a command sent to the rover. This command is broadcast to all connected clients.

Field	Type	Description
src.type	String	This is the message type of the source command whose status is being retu
src.cid	Unsigned Integer	This is the client id of the source command’s originating client
src.id	Unsigned Integer	This is the id value of the source command whose status is being return
progress	Float [0, 100]	The approximate percentage of command completion
error	Dictionary/nil	Either empty (nil) or a dictionary describing the error

1.3.4 Rover Status (roverstatus)

This status returns the current state of the rover: orientation, position, system status, etc. Note that although the attitude and position will be returned in three-dimensional format, the initial implementation of the rover systems can be strictly two-dimensional. That would constrain the roll and pitch to zero degrees while constraining the z axis to 0. NOTE: IS IT BETTER TO RETURN ATTITUDE AS EULER ANGLES OR A QUATERNION???

Field	Type	Description
attitude	3d Float Vector	The rover angles, [roll, pitch, yaw] (deg)
position	3d Float Vector	The rover offset from initial, [x, y, z] (m)
traveled	Positive Float	Total distance traveled (m)
battery	Float [0, 100]	The approximate percentage of battery remaining
error	Dictionary/nil	Either empty (nil) or a dictionary describing the error

1.4 Client Frames

1.4.1 Set (set)

This frame allows the setting of values in the control system. It requires a user that is logged in as an administrator.

Field	Type	Description
cid	Unsigned Integer	The client id
name	String	Name of the value to be set
value	Variant	The value to set the variable to

NOTE: MORE DESCRIPTION OF THE AVAILABLE VARIABLES AND THEIR TYPES NEEDS TO BE INCLUDED. IT PROBABLY MAKES SENSE TO BE ABLE TO GET VALUES AS WELL AS TO LIST THE AVAILABLE VALUES. THE GET FRAME CAN SIMPLY REQUIRE A PATTERN (I.E. A REGEX) WHICH IS USED TO CREATE THE LIST OF VARIABLE VALUES TO RETURN. THIS IS USEFUL AS BOTH AN EXPLICIT GET FUNCTIONALITY AND AS A LIST OPERATION. BECAUSE THE OPERATIONS ARE ASYNCHRONOUS, IT COULD REQUIRE SOME INTERESTING PROGRAMMING ON THE CLIENT SIDE, HOWEVER, FOR OPERATIONS WHICH ARE SUPPOSED TO LOOK SYNCHRONOUS.

1.4.2 Move

This frame provides a single move command for a rover. The command can define either a move along a circular arc or an in-place rotation. Limits on both linear and angular accelerations are included as well as the total arc distance to be traveled, the target velocity and the angular displacement.

Field	**Type**	**Description**
cid	Unsigned Integer	The client id
v	Float	Linear velocity at which to travel
alimit	Positive Float	Linear acceleration limit
d	Positive Float	Linear distance to travel
theta	Float	Angular displacement of travel
wlimit	Positive Float	Angular velocity limit
wdotlimit	Positive Float	Angular acceleration limit