

Cart Steering Subsystem



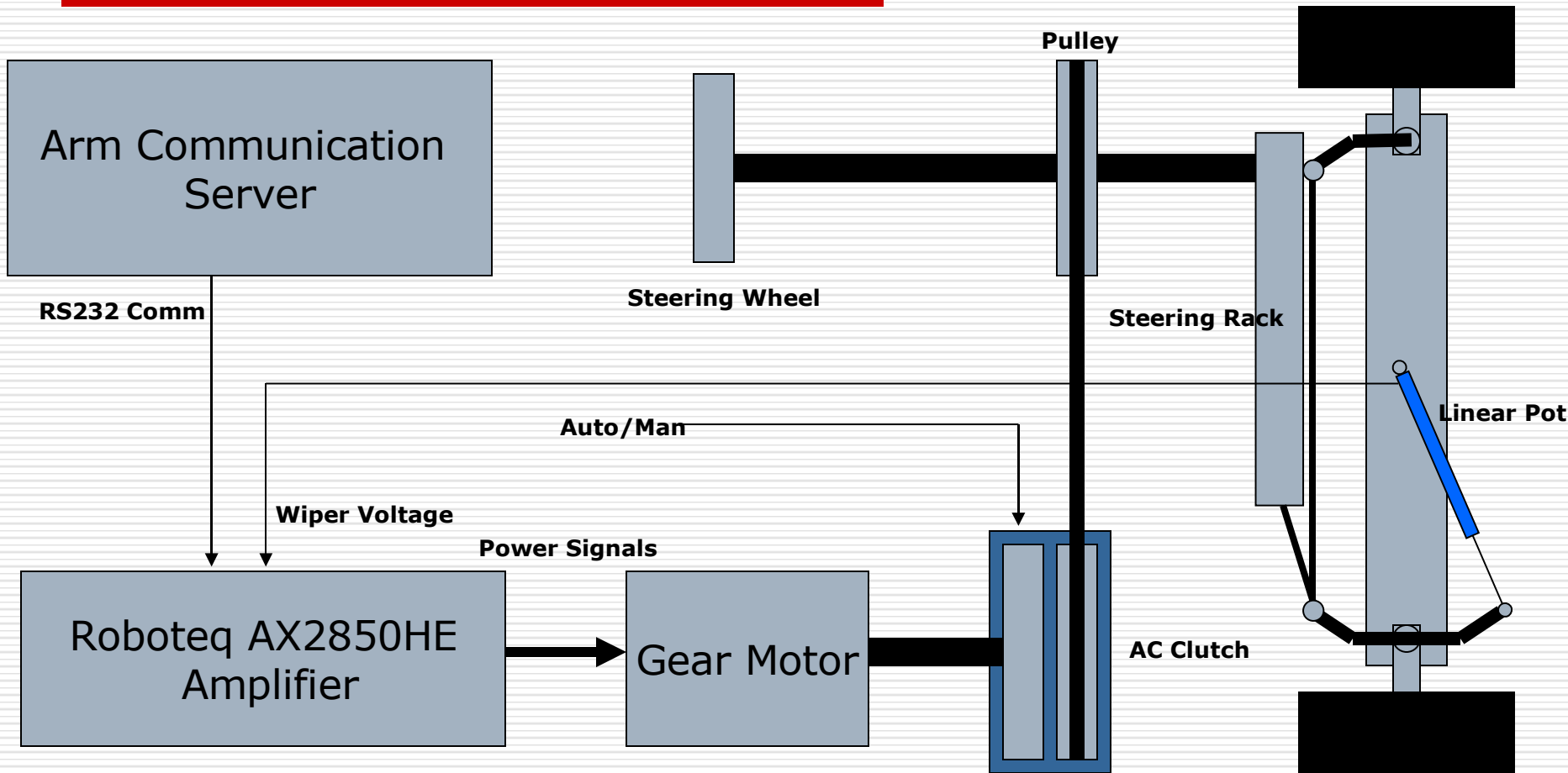
Overview

- System designed to translate commands in the form of $1/R$ ($R =$ turn radius in meters) into mechanical motion that executes the desired turn.
 - The ability to disable the system and allow manual steering control is a key design element.
-

Key Components

- Steering Actuator
 - Wheel Chair Gear Motor
 - Clutch
 - Automotive Air Conditioning System
 - Amplifier
 - Dual Channel 120 Amp Motor Controller
 - Linear Potentiometer
 - Provides a resistance that is proportional to the length of the device.
 - ARM Communication Server
 - Communicates with the Amplifier via RS232
-

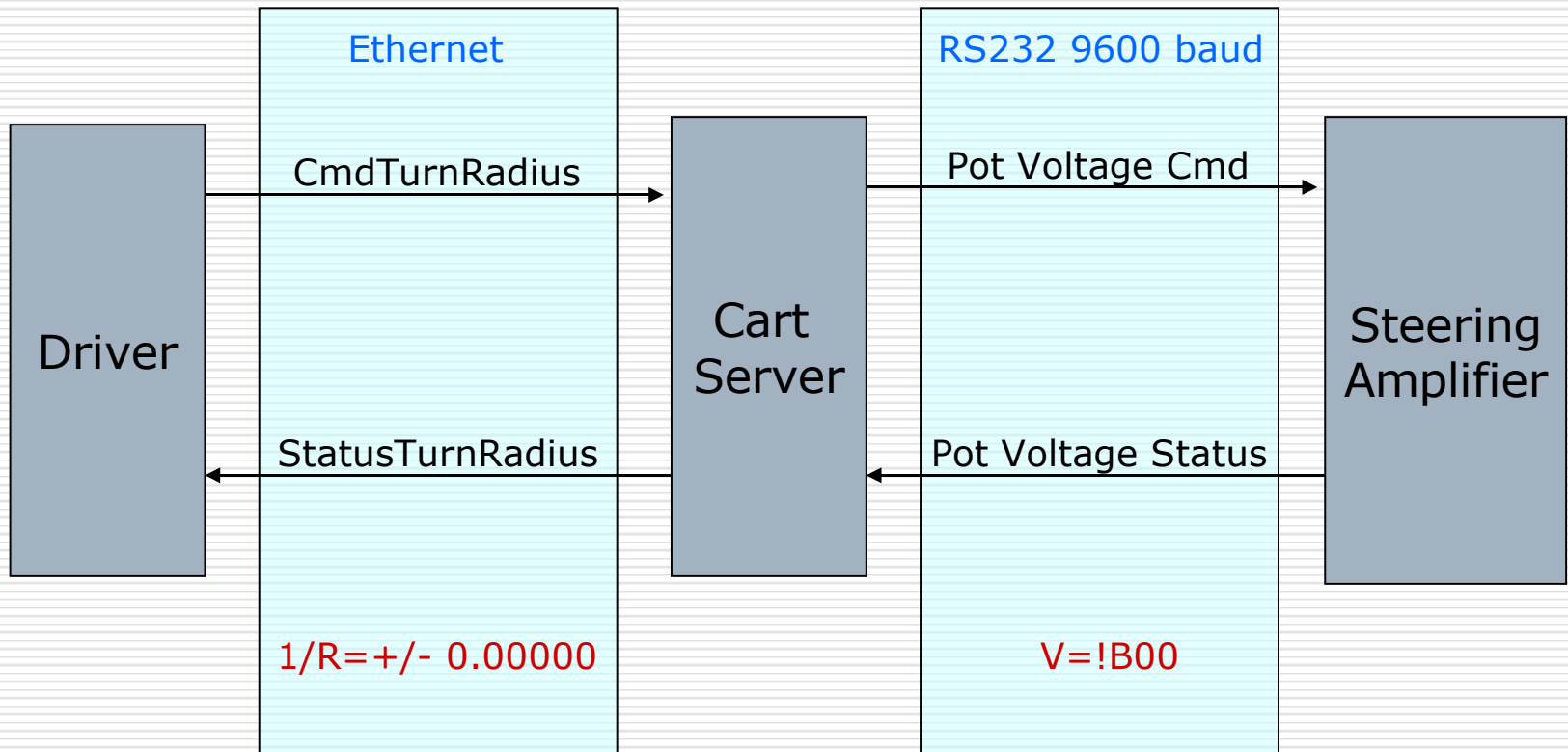
Component Block Diagram



Steering Drive Control

- Amplifier
 - 8-bit control value received from RS232 Link
 - Linear Pot Wiper Voltage Feedback is converted to an 8-bit result by an A/D converter in the amplifier.
 - A PID control loop in the amplifier is used to drive current through the motor such that the difference between the control and feedback values is forced toward 0.
 - Returns an 8-bit value corresponding to the current position of the steering system.
 - AC Clutch
 - Clutch is engaged by Auto/Man signal
 - Auto/Man signal is driven through interlock system. Software does not effect transitions from auto to manual mode.
-

Host/Cart Data Flow



Cart Server Data Flow



Lookup table

- Implemented as an array of structures
 - steering_table_element structure
 - amplifier_val (int)
 - turn_radius_inverse (int)
 - $1/R * 10000$
 - Binary search is used to select values within the table.
-

Table Lookup Examples

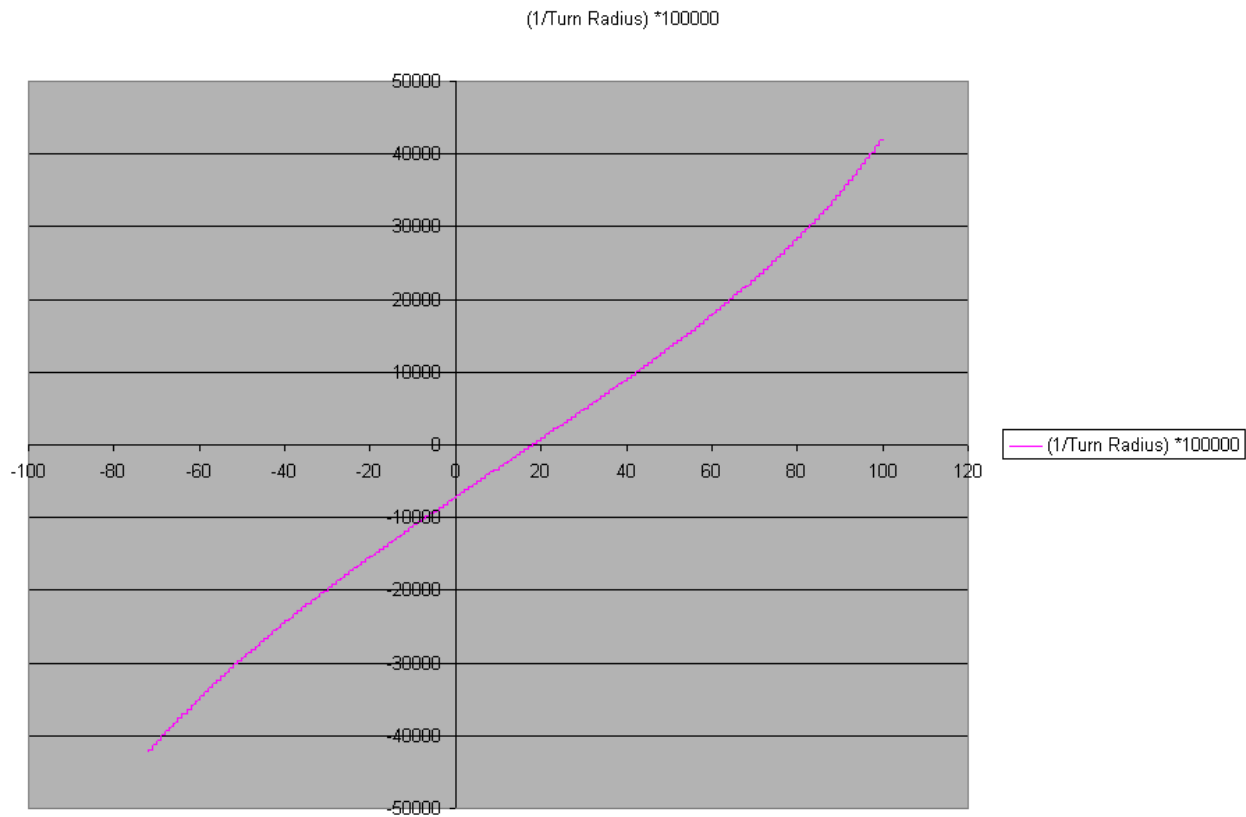
□ Lookup Table Struct Def.

```
typedef struct steering_table_element {  
    int amplifier_val;  
    int turn_radius_inverse;  
} steering_table_element_t;
```

□ Table Fragment

```
const steering_table_element_t steering_table[] = {  
    {-72,-42181},{-72,-42025},{-71,-41869},{-71,-41713},...
```

Amplifier Data vs 1/Turn Radius



Steering Thread State Machine

