Translated from German to English - www.onlinedoctranslator.com

operation manual

2023

Advanced servo motor controller for up to 7 engines

Motion4Sim

For AASD Series servo motors

3/10/2023

Table of contents

1 Introduction..... 3

2. Functions..... 4

2.1. Hardware features 4

2.2. Software features..... 5

2.3. Motion cueing software 6

3. Board and Connections 7

PCB version 1.04 ..... 8th

PCB Layout V1.04 - Pinout 10

PCB version 1.05 11

4. Hardware cabling..... 13

4.1. Servo Motor Amplifier Wiring AASD..... 13

4.2. Wiring servo 7 connection..... 15

4.3. Online switch and emergency stop switch wiring..... 16

4.4. Cabling of the position limit switches/limit switches..... 17

4.5. Connection of a servo brake..... 18

5. Settings of the servo amplifiers 19

6. Handset..... 20

7. Setting the controller / menu control 21

7.1. Filter Menu 23

7.1.1. Quick Guide to Adjusting the Spike Filter 24

7.2. Service menu..... 24

7.2.1. Repair – Moving without "Homeing" 25

7.3. actuators..... 26

7.4. SET UP..... 30

7.5. RIG 33

8. Geometry of the kinematics WIP..... 35

9. Linear and rotary actuators 39

9.1. Rotating actuators..... 39

9.2. Linear actuators..... 40

10. USB interface for motion data..... 41

11. Data transfer 42

11.1. direct mode..... 43

11.2. position mode 44

11.3. SRS mode–Sim Racing Studio 45

12.LED display 46

13.Factory Reset 47
14.Firmware update..... 48
14.1. Boot loader mode 49
14.2. Flash process..... 51
15. Dashboard app..... 52
15.1. Actuator Test Functions..... ..... .53
16. Serial commands..... 54

1 Introduction

The M4S-AASD15A servo controller enables fast and seamless transfer of motion information from the PC simulation to the servo motor. The data exchange takes place between the interface programs such as Simtools, FLyPT Mover, BFF, etc. and the simulation.

The interface software calculates the motion vectors and forwards the values to the M4S servo controller via the USB interface. This controls the servo motors and also offers the possibility to calculate and smooth motion cues yourself. The servo motor output stages are connected with DB25 cables.

All the parameters required to operate the many actuators and actuator types are freely adjustable and configurable. All common servomotors that support STEP/DIR can be used.

2. Features

The controller has been designed mainly for the AASD servo motor output stages. However, other servomotors can also be operated with it. It can be used to operate all servo motors whose amplifiers are operated via the Step/Dir function. Stepper motors that are equipped with Step / Dir also work.

The controller provides the following properties and functions:

2.1. hardware features

1. Control of 6 + 1 servo motors (optional)
2. Pulse speed up to 550kHz
3. Step/Dir Mode
4. Query and evaluation of torque (for calibration) and error status
5. Display of parameters via LC display 16x2, full menu
6. Operation via rotary encoder with push button
7. Switch for signal reception
8. Emergency stop button for the servo functions
9. Serial USB connection for data transfer
10. 32-bit processor
11. Storage of settings
12. Control of LEDs for status display (external handheld)
13. Housing (diy 3d print)
14. Controller can be installed 3m away (optional)
15. LC display and encoder can also be installed on the mainboard
16. automatic home calibration on power up or reconnection without limit switches
17. Limit switch optionally installable for home calibration
18. Protection from exceeding physical limits of the actuator
19. Optional Platform Health check, to ensure all actuators are active during gaming
20. Latency as low as 1ms for internal filtering
21. E-stop, Force Offline buttons and switches
22. USB power supply

2.2. sOftware functions

1. Software is menu driven and easy to use
2. All functions can be configured for your own requirements
3. Monitoring the status of the engines
4. Automatically calibrate the zero position of the servo motors
5. Calibration also possible via limit switches
6. Service function for repairing or checking the engines
7. Manual control of each motor individually for testing and maintenance
8. Adjustable direction for Inline or Foldback placement of servo motor
9. Adjustable screw lead Pitch advance freely adjustable per revolution
10. Adjustable stroke unlimited
11. Selection of belt reduction ratios or gear ratios freely adjustable
12. Rotary and linear actuators are supported
13. Scaling of the input signals (master gain)
14. Real exponential moving average filter for anti-vibration and smooth pulses on the actuator
15. Advanced fully adjustable Spike Filter to automatically eliminate jolts during crashes or unwanted motion cues
16. Offset for each motor (specifically for rotary actuators)
17. Variable parking position
18. Individually adjustable actuators (electronic translation, length of act., DOF system)
19. Speeds for calibration, slow speed and fast speed freely adjustable (max. 550 kHz)
20. Inverse kinematics with >1000 calculations per second (for linear and non-linear actuators)
21. Geometry for steward platforms with 6 DOF and 6 actuators individually adjustable
22. Geometry for 3 dof 4 dof and custom rigs is calculated
23. Evaluation of 24-bit input data (Simtools/Mover)
24. Calculation in 32/64 bit for maximum motion smoothness
25. 2 different operating modes : direct or inverse kinematics
26. Actuators can be operated in infinite loop (like YAW VR)
27. Support for belt systems
28. Hour meter

2.3. Motion cueing software

As an interface between the simulation / games or telemetry data supplier, a motion cueing interface is required to control the platform. This software receives the data, converts it as required, filters it and sends this data to the controller, which then calculates and converts the data/position of the motors.

With the Motion 4 Sim Controller it is possible to transfer different data. If you have your own protocols, these can be implemented.

Currently supported motion cueing apps are:

Flypt mover	https://www.flyptmover.com/
Simtools	https://www.xsimulator.net/simtools-motion-simulator-software/
SimMotion	https://eu.sim-motion.com/software/
Sim racing studio	https://www.simracingstudio.com/ (currently no longer supported)
BFF motion software	http://bffsimulation.com/6DOF-Motion-Software.php
YAME	https://hexago-motion.com/downloads/

Direct access is also possible via a terminal program.

3. Board and connectors

There are several hardware versions that are compatible and run with the same firmware.

Version 1.04 with SAME70 xpld board and breakout board for the interfaces Fig.1.

Version 1.05 with ATSAME70 MCU with FTDI or CH340 USB USART Fig.2

Figure 1 M4S Controller V1.04 with display and encoder

Figure 2 M4S Controller v1.05 FTDI

There are also additional hardware options.

It is possible to carry out the control via a display and encoder directly on the board see Fig.1. Alternatively, the handheld device can be used for remote control. See fig.3

The handheld device is connected via a SUBD25 male/male cable. The cable can be up to 3m long.

Figure 1 Handheld device

There are 6 Sub D25 connectors for connecting 6 AASD servos as well as a Sub-D25 connector for connecting the handheld. Solder strips are provided for connecting Servo 7 and for integrating homing switches for each motor. There are also pins for connecting LEDs, an online switch and an emergency stop switch.

PCB version 1.04

There are a total of 2 Micro USB Connectors on the SAM E70 Board and one Micro USB Connector on the Motion4Sim Breakout Board. There is a reset button and a USr button hidden on the bottom board.

Overview of the connections:

Bossa Program Micro USB Port

Figure 4 v1.04 Micro Usb port for flashing firmware

PCB Layout V1.04 - Pinout

LEDS are switched with 3.3 volts without a series resistor. The homing switches switch their signal when in contact with GND. The emergency stop switch is "normally closed" and the online switch is "normally open". These settings are configurable in the software.

Figure 5 DSub25 handheld pinout

Figure 2 DSub25 servo connector pinout

PCB version 1.05

Overview of the connections.

The same connections are present as in the PCB version 1.04. There are 6 connections for servo motors, a connection for the handheld device, 2 micro USB connections, a reset and a USR button as well as various connections for LEDs, extra inputs and outputs as well as soldering pins for a 7th servo connection and its homing switches . To the left of the connection for the handheld there is an adjustable resistor with which you can adjust the brightness of the LCD display.

Figure 3 PCB v1.05 overview

Figure 4 Pinout Solder Blocks PCB v1.05 in detail

Version 1.05 with display and encoder is delivered with a screw terminal on the SW1 connections. Unfortunately, the labeling of version v1.05 is not implemented as in the visual graphics used here. Some resistors have to be removed for using the Servo 7 on some delivered boards. From the publication of these instructions, these are always already removed. These resistors were planned for functions that were not finally implemented in the firmware. If in doubt, contact support.

The pin assignment of the servo connections and the handheld are unchanged from the previous HW version. See fig.

4. Hardware wiring

4.1. Servo Motor Amplifier Wiring AASD

2.1.1 Servo driver wiring diagram

The mains voltage connections are sometimes different, please always consult the relevant data sheet from the manufacturer. Also see, for example:

> https://github.com/motion4sim/AASD15A-Servo-Controller-for-Motion-Simrigs/tree/master/Manuals

2.3 Standard connection

2.3.1 Position control wiring diagram

4.2. Wiring servo 7 connection

Wiring the pins for the 7th servo motor. It can either be installed directly on a DB25 cable or you can wire a female Sub DB25 connector for soldering

In addition, +5 volts and ground are required that can be used by the SW2.

PCB V1.04

PCB V1.05

For this version, all connections on the pin strips are included.

4.3. Wiring online switch and emergency stop switch

4.4. Cabling of the position limit switches/limit switches

The switches are connected via the corresponding pin strip. To trigger the switch, it must be connected to GND. On the board, the pins are just labeled "Calibrate". The assignment is as shown on the left from bottom to top.

Inductive proximity switches can also be connected. These only require an additional 5V voltage from the board.

eg

https://wiki.dfrobot.com/5V%20Metal%20Proximity%20Switch%20SKU%3A%20FIT0658

https://www.cytron.io/c-sensor/p-5vdc-inductive-proximity-sensor-npn-no

the brown cable is connected as a signal.

4.5. connecting a servo

A servo brake via the Extra1 pin of the be controlled in the same way LED solder bar.

The corresponding function must be assigned to Pin Extra01 in the software configuration.

The connection must be made via a relay. The pin only tolerates a current of 30 mA and 3.3 volts.

The servo brakes usually have an operating voltage of 24V, for which an additional power pack is required. The well-known Arduino relays can therefore also be used.

5. Settings of the servo amplifiers

The controller can be connected to all servos that have a DB25 connector. It is also possible to produce your own adapter cable. Ask about compatibility.

AASD-15A Servo Settings:

Push MOD until you see Pn000. This enters the parameter mode.

Change and check these settings on all motors:

FN9 = 230 (230Volts recommend to set this value) Check this value if you get errors

The comparison voltage is measured and stored with FN9.

Pn8 = 300
Pn9 = -300
Pn51 = 3000
Pn98 = 1-20 - Pulse Multiplier (electronics gear) different to Thanos and SFX for higher resolution

Typically on M4S you can set this to PN98 = 2

Pn109 = 1 - smoothing, 1=fixed smoothing, 2=s-shaped smoothing
Pn110 = 30 - smoothing filter time
Pn113 = 20 - Feedforward %
Pn114 = 10 - Feedforward Filter Time (ms)
Pn115 = 100 - Gain %
Pn24 = 100

Pn51 = 3000 Motorspeed (2500 or 3000 mainly)
Pn52 = 1 Sign Port 1 Servo enable
Pn60 = 2 signal port 1 servo ready
Pn61 = 6 signal port 2 servo treach

Pn62 = 4 signal port 3 servo preach

Please also read the manual of the AASD servo amplifiers

6. Handset

See also for more information

https://github.com/motion4sim/AASD15A-Servo-Controller-for-Motion-Simrigs/tree/master/handheld

and

https://www.thingiverse.com/thing:4641555

7. Controller setup / menu control

When you switch on the controller or supply it with USB power, the display activates and loads the last used settings from memory. It then waits for user input. If you press the encoder, the automatic calibration will start.

There is also the option to turn the encoder to the left. This enables you to access the service menu without performing the calibration. (2nd image) if you turn the encoder further to the left, there is still the option of performing a firmware reset. (Picture 3+4) Attention, all settings will be lost and standard settings will be loaded

Factory reset for rotating actuators

Factory reset for linear actuators

In the first case, you get to the main screen after pressing the encoder button. Here the status of the first 6 actuators is shown on the right side.

Possible actuator states:

X	is	Offline
P	is	Preparing (searching for the homeing
!	is	position) Moving to the offsets/safety parking
R	is	position Ready to use

The top line of the display shows the status of the rig system and which movements
the machine runs.

preparing..	calibrate and move to offset position Emergency
emergency..	stop button activated
offline..	Offline switch off, parking position
wait uart..	Movement to home position / in standby position waiting for serials Data
on-line..	Serial data is being received and movement is occurring

Functions of the encoder on the main screen:

The status of the master gain controller is displayed at the bottom left. (100%)
Turning the encoder to the left or right decreases or increases the amplification of the pulses in 10% increments. The gain has a range of 10% - 400%. The gain can also be changed in online mode.

If you press the encoder you activate the submenu structure. The menu structure is divided as follows.

1 Shutdown	- Park and shut down actuators	
2. Homebelt	- Tension the belt	
3. Homing	- automatic re-calibration	<- Jump to menu activation
4. Filters	- Output filters	
5. Services	- Menu for manual actuator movements	
6. Actuators	- Settings of the actuators	
7. Setup	- Controller settings	
8. Rig	- Settings for Stewart Geometry	
9. Kinematics	- Input filter for DOF data	
10. Reset	- Restart the controller	
11. Save and return	- Save current settings and close menu	

You can navigate through the menu structure by turning the encoder left or right. By pressing the encoder you activate the respective menu item.

7.1. Filter Menu

The following options can be set.

Filter enabled	:	Filter on or off
Filter Type	:	EMALPExponential Moving Average Low Pass Filter (like in Mover) EMALP & SPIKE
filter samples	:	Strength of the selected filter. The bigger the value the smoother the movement
Spike filter :		This filter is designed for excessively strong unwanted movements. This value affects the smoothing of the pulses when the spike filter has been activated by exceeding the limit. The strength increases exponentially (Filter ^4) the larger the window is exceeded. Video on how to set the spike filter : https://youtu.be/j6xVlIVM2L4
Spike Window:		when it is exceeded, the filter is activated. 100 roughly corresponds to the pulse interval of half of all possible pulses of the actuator. Try it out for an optimal setup. The lower the value, the sooner the filter is activated. At 1, all pulses are filtered. _bithalf *RIG.spike window *RIG.spike window * RIG.spikewindow / 10000000;
Sp.Filter smooth :		The spike filter operates dynamically and changes the smoothing accordingly. Therefore, another EMA filter for the dynamic change was built in, which is controlled by this value. Value small spike filter increases quickly and jerkily when triggered and also returns quickly and jerkily after the spike ends. Value large spike filter increases more gently and when triggered also returns to the initial filter value more smoothly.

7.1.1. Quick guide to setting up the Spike Filter

1. Start with the default settings after a factory reset
2. Create a motion loop eg in Mover (Saw wave) with a high amplitude
3. or alternatively drive your car against a wall in a simulation, which triggers strong acceleration on the actuators.
4. Reduce the value for "Spike Window" until the spike clearly increases in strength and look for a tolerable value.
5. Slowly increase Spike Filter, Attenuation and Spike Window until you find your setup.
6. "Sp.Filter smooth" is for fine-tuning only and doesn't actually matter much.

Try it yourself here.

7.2. Service menu

This menu is intended for testing the actuators and for moving the actuators to special service positions. All actuators can be moved simultaneously or each one individually. After activating the respective sub-menu, the value of the actuator can be increased or decreased by turning the encoder to the right or left. The value is changed by the value stored in "Multiplier". This menu can also be reached directly when starting the controller without calibrating the actuators. This can be used, for example, to "jack up a Stewart platform" in the event of repairs. A calibration is carried out when the menu is exited.

This function can also be used to save the offset of the individual actuators and all actuators. Set "Yes" for Change Offset with the encoder and confirm with Push.

7.2.1. Repair - moving without "homeing"

For maintenance of the rig or an actuator it may be necessary to move one or all actuators without homeing.

After starting the controller, select and confirm with the "encoder".

Move the encoder left or right and select one or all actuators to move.

Caution after exiting the menu, a calibration is carried out.

If you press the emergency stop button, the servos are deactivated depending on the setting in the setup.

Note: Turning the encoder moves the actuator, always by exactly the number of pulses set in "Multi". Use a different "Multi" setting to move the actuator faster.

7.3. actuators

The parameters of the actuators used are set here. Save after changes and restart the controller so that the new values are calculated correctly.

The following values apply to all actuators, changes overwrite the individual values of all actuators.

RIG : Linear or rotary, custom

These values are used for the calculations of the inverse kinematics and for calculating the resolution of the actuator.

6dof Rotating Actuator

6dof linear actuator

SFX 100

SFX 150

PT100 4 act

PT100 4Act TL

PT100 4 Act TL SR

PT100 4 Act TL TL SR

PT150 4 act

PT150 4Act TL

PT150 4 Act TL SR

PT150 4 Act TL TL SR

SFX 150 3 Act 2 front 1 back

SFX 150 3 Act 2 back 1 front

Custom

encoder PPR : Number of encoder positions depending on

Servo motor, at AASD15 10000

electronics. Gear : For calculating the actuator resolution needed

PN(98) value from the servo output stage

These global properties overwrite the values of the individual actuators in the actuator sub-menus.

Gear box ratio : Ratio of gears used. Reduction only possible.

lead crew pitch : Pitch of the trapezoidal thread spindle 5, 10.25 ... in mm per revolution (only linear actuators)

actuator length : Length of the linear actuator in mm (only linear actuators)

Red.Range : "Range" for rotating actuators in degrees. Half a turn is 180 degrees

Lin Act. security : Safety distance to actuator limits in mm for linear actuators. Will be bilateral subtracted from the actuator length.

actuators : Sub-menu for actuator 1-8 Activate by press the encoder button

(sub-menu for actuator 1-8)

rotation	:	Direction of rotation of the actuator, CW vs CCW
Park enable	:	if you activate this option, the Actuator not parked
cal offset	:	Offset in pulses will be after calibration set. And as the lower zero of the actuator used. (plus parking position only with rotating actuators)
elec. Gear	:	see above
mechanical gear	:	Ratio of gears used. Reduction only possible.
encoder position	:	see above, encoder resolution
parking position	:	Offset from the lower zero point (calibration point) of the actuator as parking position. Offset is total for each Actuator individually calculates parking position + actuator cal. (rotary actuators only)
LS Length	:	Length of the linear actuator in mm (only linear actuators)
LS Pitch	:	Pitch of the trapezoidal thread spindle 1… 1000 … in mm per revolution (only linear actuators)
LS Safety	:	Safety distance to actuator limits in mm for linear actuators. Will be bilateral subtracted from the actuator length. Type. 2-5mm
red.Range	:	"Range" for rotating actuators in degrees. Half a turn is 180 degrees
Limitless	:	special function for e.g. 360° actuators for Endless YAW. This makes it possible via unscrew the actuator end without the

having to take a long way is best

watch the video on youtube

Infinite loop on actuators

https://www.youtube.com/watch?v=wvK86gH9gZg

belt function : the actuator is used as a belt tensioner.

Here the servo is driven only a few pulses against the resistance, which builds up a moment and the pilot feels the pressure through the belt. The working direction is set with "Rotation". During homeing, the servo is calibrated for maximum torque. The belt then moves back by the number of pulses set in "Belt retreat". This is to represent the relaxed belt. The value should be adjusted by trial and error. The torque is set in PN8,PN9 and PN24 for AASD servomotors.

If the rig is switched "offline", the belt should be further relaxed and runs the pulses set in "Parkpos". If a loose belt is required, the number in "Parkpos" should be increased

Belt autohoming : the rig is switched "online".

the belt was homed at the same time. The driving speed of the servo motors is briefly switched to "home speed".

Belt retreat : Number of pulses to relax the belt

after home.

//End of Actuator Sub menu

homespeed : Pulse speed for calibration in Hz

park speed : Pulse speed for the slow ones

speed ,procedure from and to

Park/standby position

online speed	:	operating speed, should be at least 253 kHz with PN98=2
number actuators	:	Number of connected motors, Die Number is controlled, checked if "Actuator check" is activated.
Multi	:	The multiplier is used for value changes used where numbers in large amounts need to be changed. So that encoder cannot be rotated "1000" times set the multiplier to 1000 and rotates just one click.

ie with a multiplier =1, 1 is always added to/ subtracted from the value for each encoder detent. If the multiplier is 500, 500 is added or subtracted.

For offset and speeds, park pos etc.

7.4. SET UP

Emergency Stat	:	Indicates how the engines after activation of the emergency stop switch can be controlled. Pay careful attention to this setting. Your safety depends on it.
	"power off"	Servos are switched off (RIG falls down) Homing is performed after switching back on.
	hold power	Servos stay put but are energized, no movement. Position of the motion platform is maintained. Homing is performed after switching back on.
	"keep online"	Servos stay energized, no movement. Position of the motion platform is maintained. Homing will turn back after**not**carried out. The previous movement is recorded and continued. In online mode, a transition to the current movement is carried out. Attention in case of an error the homing is lost it can happen that the actuators drive towards their end and damage occurs.

Emergency NC : Normaly open or normaly closed , this is where you choose the switching status of the emergency stop switch.

actuator check : Checks all actuators when the servo motors are online are. If a fault occurs, there are no signals more sent to the engines. That is to be noted for this all servo connections one after the other are occupied and all adjusted engines also must be connected. This means, for example, connection 1-4.

If ports 1-3 and 5 are occupied, this results in one Error if actuator check is "enabled".

After pressing the encoder, in the event of a fault subsequently all engines that are online are calibrated.

Encoder direction : As noted, some encoder types provide the signals wrong. Should be an encoder wrong way around can switch the direction of rotation with this option be changed.

Extra01 : The function of the pin is set here. There are the following functions:

- no function never mind
- servo brake switches the servo brake
- send clock outputs a pulse every 1ms

Servo_enable : Changes the pin state from high to low for the servo enable signal.

master gain max	:	determines the upper value of the master gain.
online location	:	sets the standby position in percent. Ie if If you switch online, the system first moves to the standby position, usually the actuator center 50%, and then goes into motion via the transition. If you set the standby position to 0%, you drive directly via the transition into the motion.
transition speed	:	Transition mode speed. (Morphing into Motion)
critical Angle	:	here the angle is set, which is between Motor axis and connecting rod must not be fallen below (mechan. Limiting) Only for red. 6dof RIGs.
online time	:	Hour meter comes with factory reset Turned off.
Multi	:	see Multi above

7.5. RIG

Parameters for inverse kinematics. Please configure your rig with Mover to understand how inverse kinematics works. The value entries are based on Mover and can be adopted directly.

For rotary actuators and 6dof Stewart platforms

Source : FLypt "Mover" https://www.flyptmover.com/

For linear actuators

Source : FLypt "Mover" https://www.flyptmover.com/

List of Values:

Base L3 short	short side of lower platform in mm
Base L4 long	long side of the lower platform in mm
Rig L1 long	long side of the upper platform in mm
Rig L2 short	short side of the upper platform in mm
Crank/Range len	Crank arm length at red. platforms
	Length of the actuator from the middle position to
	Actuator stop in mm for linear actuators

Rod/Middle len	Length of connecting rod at red. platforms Length of the actuator in middle position at lin. actuators in mm
Middle Angle	only with rotating actuators , angle of the Crane arms in standby position of the actuator "determine by measuring"
Crank Angle 1	Angle of the motor axis of rotation in the base plane The angles are counterclockwise Measured. It is best understood if you paint it.
reversed 1	angle + 180 ° ,
Crank Angle 2	
reversed 2	
Crank Angle 3	
reversed 3	
Crank Angle 4	
reversed 4	
Crank Angle 5	
reversed 5	
Crank Angle 6	
reversed 6	
XY orientation	With this option you can set the axes in the XY Just twist, given in degrees
COR X/sway	X value for the center of rotation calculation in mm from Center
COR Y/surge	Y value for center of rotation calculation in mm from Center
COR Z/heave	Z value for center of rotation calculation in mm from Center

8. Geometry of the kinematics WIP

For calculations of the inverse kinematics of the Stewart platform, the geometric values of the platform are stored in this menu. These are queried as follows

Base platform: Length L3 and L4

The basis for this is a regular or irregular hexagon (hexagon)

The same geometric basis also applies to the upper platform (the movement platform), but here a scalene hexagon is almost always used as the geometric basis. Whose corners are rotated 60 degrees to the base.

Geometry overview:

Base L3 (shorter side), L4 (longer side)

Cockpit L1 (longer side) ,L2 (shorter side)

Uneven Base: Linear Hexpod

The total length of the actuator in the home position "Rod Length" is also required for the calculations for linear actuators. The value of the range of the actuator is taken from the "Actuator" menu. For rotating actuators, you also need the angles of the lever arms of the motors in relation to the X-axis according to the following sketch. These are stored in Crank Angel[1..6]. Also the length of the connecting linkage "Rod Length" and the length of the lever arms "Crank"

The angle of the lever arm in the home position is required for the geometric definition of the home position. This can be determined by calculation (18°-29°).

120°
240°
Motor6
Motor1
0°
Motor5
Motor2
0°
Motor4
Motor3
240°
120°
X Achse

ROD length
Crank

9. Linear and rotary actuators

9.1. Rotating actuators

MaxPulse = RIG.ppr *RIG.mechanics_GearRatio_Servo

/RIG.electric_GearRatio_Servo

/ (360 /RIG.range factor);

For example: (rotating)

AASD15A 10000ppr encoder

Gear box 1:50

Range 180°

PN98 = 2

= 10000* 50 / 2 / (360 / 180)

= 125000 positions per half round

Resolution depends on arm length

cal Offset:

The offset is defined and approached individually for each actuator. It indicates the number of pulses that are approached after homeing. This offset is used to compensate for possible geometric inaccuracies of the hard stop or the position limit switch

Parking position:

The parking position indicates the number of pulses that all actuators travel after calibration. This offset is used to uniformly define the position of the parking position. Calculation: see above 62500 pulses would be a 90 degree rotation, for example

PS: The actuators can be excellently calibrated with the help of the service menu.

Standby position : Calculated in pulses from half the range.

9.2. Linear actuators

MaxPulse =RIG.ppr *RIG.mechanics_GearRatio_Servo / RIG.electric_GearRatio_Servo * ((RIG.leadscrew_length -RIG.lin_act_safety * 2) /RIG.leadscrew_pitch

SFX 100 type

AASD15A 10000ppr encoder

Gear box 1:1

PN98 = 2

Lead crew = 5

Leadscrew length = 100mm

Linear Safety = 5mm

= 10000*1/2*((100-2*5)/5)

= 90000 positions / 100 mm

Resolution = 0.0011mm

10 USB interface for motion data

The controller contains an FTDI FT232 RL chip or Ch340 that installs via USB.

The drivers required for this are usually installed automatically under Windows. these can but also about:

FTDI:

https://www.ftdichip.com/Drivers/VCP.htm or

CH340:

https://www.makershop.de/ch340-341-usb-installieren/

be downloaded and installed.

11. data transmission

Data transfer is established via the standard USB interface. There are 2 modes to transfer the data

The data packet string now is 28 bytes long and includes additional spare motion data slots for up to 8axis

The serial interface can be configured as follows

data output	binary
speed	250000 bauds
data bits	8 bits
Stop Bits	1
Parity	None

11.1. direct fashion

This is the direct mode in this mode all actuators are controlled individually. The actuator position is sent bit-coded. The resolution is 24 bits, i.e. 3 bytes.

This means the following.

Value = 0x00000 (Decimal =0) Actuator all the way down

Value = 0x7FFFF (Decimal = 8388607) Actuator in the middle

Value = 0xFFFFFF (Decimal= 16777215) Actuator at top

This means that 16777215 positions can be approached per actuator.

The ID is byte values 0xFF + 0xFF Each Axis is 24 bit wide. (3byte) LF+CR is required in the end (0x0A + 0x0D)

ID AXIS1 AXIS2 AXIS3 AXIS4 AXIS5 AXIS6 AXIS7 AXIS8 LF/CR

data format

2 start bytes	<255><255>
24 data bytes	3 bytes (24bit) per channel/actuator 8 channels
2 stop bytes	<10><13>

Danger:

To avoid errors, the following information for movers/Simtools.

Characters are entered for each unused actuator so that the string always has the same length (28 bytes).

For example, if you only use one <wind> actuator, 21 characters must be sent before/after

<255><255>aaabbbccc<wind>eeefffggghhh<10><13>

For the <Wind> actuator, 24 bits must be set in the Rig module or in the Direct module so that this value is also 3 bytes long.

11.2. position mode

In this mode, positions are sent to the controller in mm for the translational movements (Surge,Sway,Heave) and angles in degrees for the rotating axes (Yaw,Roll,Pitch)

Floating point numbers are used for this (from Mover 3.4.2) The so-called floats are 32 bits in size and are sent in 4 bytes. This makes the string 32 bytes long.

Sway Surge Heave as 4 byte float in mm

Yaw roll pitch as a 4-byte float in degrees

2 start bytes <252> <252>

8 x 4 byte floats

2 stop bytes <10><13>

Missing values have to be filled with 4 characters per axis, so that the string is always 36 bytes long.

Example for movers:

<252><252><PSway><PSurge><PHeave><PYaw><PRoll><PPitch>00000000<10><13>

The values of the axes are now 4 bytes (32 bits) wide, including the fill values for axes 7 and 8, so a total of 36 bytes are transferred.

Settings in the kinematics menu for scaling are no longer required.

11.3. SRS mode–Sim racing studio

Currently not supported.

There is another mode, this mode provides an interface to automatically connect to SRS - Sim Racing - Studio. It works “Plug&Play” The data transmitted by SRS is expanded from 16 bits to 24 bits.

Video explanation:

https://www.youtube.com/watch?v=BZgPRXhlQf8&t=15s

12. LED display

Meaning of the ads

Green LED shines	Controller standby position or online and receiving data
Red LED shines	Controller offline, homeing, parking position
Green blinks and red lights up	Movement to standby or park position
Red blinks on irregular	Spike filter active
Red flashes regularly	Emergency stop mode active

13. factory reset

Sometimes it is necessary to reset the data in the controller memory to its original state. To do this, use the following instructions

1. Reboot the controller
2. Wait until the message "press encoder for homeing..." appears on the display
3. Then turn the encoder to the left to open further menu items

4. Turn until "factory reset" appears in the display. Press the encoder to do this.

   From firmware 1.41 there are 2 options for a factory reset. The settings are adjusted as a basis for linear and rotary actuators.
   The controller restarts after execution.

5. The data has been reset. The controller restarts.

14 firmware upgrade

If necessary, regular firmware updates will be delivered. You can find them at

https://github.com/motion4sim/AASD15A-Servo-Controller-for-Motion-Simrigs/tree/master/firmware

Download and upload to the controller according to the separate instructions. It's very easy.

Video guide:

https://www.youtube.com/watch?v=sPiSqAtpf4k&t=181s

14.1. Bootloader mode

There are 2 ways to get into bootloader mode.

Version 1.

1. Connect the controller to the computer using the USB cable as shown in the picture above.
2. Press and hold the User button
3. Briefly press the reset button
4. A new USB device should appear in Device Manager and the controller is in bootloader mode.

Variant 2. With handheld

Enter Bootloader

1. **Operate Emergency Button**
   (if the button is normaly open relase it !)
2. **Switch Online Switch to off**
3. **Select Reset Option in the menu**
4. **execute the reset and hold the encoder button until reboot**
   (normaly red led goes off , display stays at reset)

5. **New usb port recognized**

The following steps are necessary with the handheld device

1. Press the emergency stop button
2. Turn the online switch to Off
3. Navigate to "Reset" in the menu
4. Press the encoder and keep it pressed (3 s) until a new USB device appears in the device manager. (see variant 1)
5. If that doesn't work, release the emergency stop button and repeat the procedure
6. The controller is in bootloader mode

14.2. flash process

1. Install the Dashboard app if you haven't already done so

2. Navigate to the "Firmware" tab

3. Download the latest firmwarehttps://github.com/motion4sim/AASD15A-Servo-Controller-for-Motion-Simrigs/tree/master/firmware

4. Select this file

5. Check the "encrypted" checkbox

6. Do not choose a com port. Make sure that unplugged all unnecessary USB serial devices and no Bluetooth serial is activated.

7. The Com port is determined automatically. Should it Problems give select the com port manually.

8. Press the Flash button

9. Confirm that the controller is in bootloader mode.

10. The firmware is transferred, the controller restarts.

15 Dashboard app

There is now a new smart remote app for easy editing of the controller configuration. With this, the most important data downloaded from the controller can be modified, saved and written back. It can also be used to test actuators and flash firmware. Also, a handheld device is emulated in software.

Download from Gitub:

https://github.com/motion4sim/AASD15A-Servo-Controller-for-Motion-Simrigs/tree/master/Dashboard

The app proves to be unstable on Windows 11 in some cases.

Start the Dashboard App, it will automatically connect to the controller. If a handheld is installed, the app and handheld device can be operated in parallel.

Only the "emergency stop" switch of the handheld device has priority over the app.

Gain and filter can be easily operated via the app.

Functionality explained on Youtube.

https://www.youtube.com/watch?v=hUnLz29yCMw&t=469s

15.1. Actuator test functions

The “test page” enables the actuators to be tested. If the controller is "connected" you can switch to the "Test" page tab and press the "RUN" button. The app starts sending the set values for the actuators. The actuators should be calibrated and online.

The specified values for range are used for the "6dof mode float".

16 Serial Commands

Not only motion data as described in chapter 11 can be sent via a serial terminal. Additional control commands and data queries are described below. This makes it possible to control the controller.

Send:

Syntax:

1. Start bytes	2 bytes <f9><f9>	hex or <249><249> decimal	
2 Commands	2byte <0-ffff>	hexor	<0..65535>
3. value	2byte <0-ffff>	hex or	<0..65535>
4. Stop bytes	2byte <0a><0d> hex or <10><13>		

8 bytes in total

Reception:

Syntax:

1. Start bytes	2 bytes <f9><f9>	hex or <249><249> decimal	
2 Commands	2byte <0-ffff>	hexor	<0..65535>
3. value	2byte <0-ffff>	hex or	<0..65535>
4. Stop bytes	2byte <0a><0d> hex or <10><13>		

8 bytes in total

The command part always responds with the requested value.

commands

send	f9f9 0000 0000 0a0d	**(RIG Status)**
returns	f9f9 0000 XXXX 0a0d	
	xxxx is	

preparing	0x0000
parking	0x0001 0x0030
emergency	0x0002 0x0004
offline	0x0008 0x0010 0x0020
ready	0x0031
standby	0x0032
wait uart	0x0033 0x0034
on-line	0x0035

send	f9f9 0100 0000 0a0d	(portid)
returns	motionusbport	= f9f9010001000a0d
	native USB port	= f9f9010002000a0d

send	f9f9 0200 0000 0a0d	(fw_version)
returns	f9f9 0002 0002 0a0d	not implemented ready

send	f9f9 0300 0000 0a0d	(request bootloader mode)
returns	reboot to bootloader mode	

send	f9f9 0400 0000 0a0d	(enables softoverride for handheld app)
returns	f9f9 0004 0002 0a0d	

send	f9f9 0500 0000 0a0d (enable emergency)	
returns	f9f9 0005 00xx 0a0d	(emergent stat 0 or 1)

send	f9f9 0600 0000 0a0d enable online	
returns	f9f9 0006 00xx 0a0d	(online stat 0 or 1)

send	f9f9 0800 00xx 0a0d 0 = factory reset linear 1= rotary
returns	f9f9 0800 0001 0a0d reboots

send	f9f9 0900 0000 0a0d reboot
returns	

send	f9f9 1000 0000 0a0d	(request status actuators)
returns	f9f9 1000 00xx 0a0d	

xx is

0 ='x'; //not connected
1 ='C'; calibrate
2 ='P'; preparing
4 ='!'; moving
8th='R'; ready

send	f9f9 3000 0000 0a0d	request master gain
return	f9f9 0030 00XX 0a0d	XX 1-40*10 in % 10 is 100%

send	f9f9 3100 00XX 0a0d	set master gain 1-40)
return	f9f9 0031 0002 0a0d	is set

send	f9f9 4000 XXXX 0a0d	set filter to XX or read if xx=0
returns	f9f9 4000 XXXX 0a0d	filter size (emalp)

send	f9f9 4100 XXXX 0a0d	enable/disable filter XXXX = 1 or 0
returns	filterstat if XXXX=2	
send	f9f9 4200 XXXX 0a0d	enable/disable spikefilter XXXX = 1 or 0
returns	f9f9 0042 XXXX 0a0d	filterstat if XXXX=2
send	f9f9 4300 XXXX 0a0d	set spike filter to XXXX or read if XXXX=0
returns	f9f9 4300 XXXX 0a0d	returns spike filter size
send	f9f9 4400 XXXX 0a0d	set spikefilterwindows to XXXX or read if XXXX=0
returns	f9f9 0044 XXXX 0a0d	returns spike filter size
send	f9f9 5000 0000 0a0d	home
send	f9f9 0060 0000 0a0d	send display frame back,ask for details
	38 bytes f9f9 32 bytes Char of display 0a0d	
send	f9f9 0070 0000 0a0d	encoder button pressed
send	f9f9 0071 0000 0a0d	encoder left pressed
send	f9f9 0072 0000 0a0d	encoder right pressed
send	f9f9 0080 0000 0a0d	go online after homing without input

Thank you for using Motion4Sim

Have fun

Motion4SIM

Update follows.