Matlab functions for Scorbot TCP/IP Server

1. Session Commands – these start and stop a session. Starting a session starts a TCP/IP server program, connects Matlab to the server, initializes the Scorbot Dll, Defines a vector to receive points for the robot to move to, and ‘homes’ the robot. Ending a session closes all Dlls and closes the server program and window.
2. Start\_SB\_Session() no parameters, starts a TCP/IP server (location of server must be entered into the m-file), initializes the Scorbot DLL, defines a point vector (USNA1), and homes the robots.
3. End\_SB\_Session() no parameters, ends the TCP/IP server program, closes the Scorbot DLL and closes the console window.
4. SBInit() A housekeeping function that initializes the Scorbot Dll. This function is called in Start\_SB\_Session and is not normally called external to that routine.
5. Point Vector Commands – these commands determine robot position and allow the user to fill a vector of points for the robot to use in movement commands.
6. SBDefineVector ()– defines a vector named ‘USNA1’ to receive motion points (up to 1000 pts). This command is automatically called in Start\_SB\_Session and is not normally called external to that routine.
7. SBGetPositBSEPR ()- returns a vector the current position of the robot in joint angle coordinates. Angles are represented in 1/1000 of a degree.
8. SBGetPositXYXPR()- returns a vector of the current position of the robot in XYZ format. Positions are represented in micrometers.
9. SBAddPointBSEPR(N,Pt,Rel) – adds a point in joint angle format to the defined USNA1 vector. N is point reference number 1-1000, Pt is a vector of five doubles [B S E P R] giving joint angles in 1/1000 degree, Rel = 0 if the point is absolute or 1 if the point is relative to the current robot pose.
10. SBAddPointXYZPR(N,Pt,Rel) – adds a point in XYZPR format to the defined USNA1 vector. N is a point reference number 1-1000, Pt is a vector of five doubles [X Y Z P R] giving position in micrometers, Rel = 0 if the point is absolute, 1 if the point is relative to the current robot pose.
11. SBDeletePoint(N) – remove a point (N) from the defined point vector. If N= -1 all points

are removed.

1. SBGetJaw()- returns the current opening of the robot gripper in millimeters.
2. Motion Commands – these commands move the robot to points stored in the defined USNA1 point vector.
3. SBGripper(OC) – opens or closes the gripper. OC=1 to open, OC=0 to close.
4. SBMoveJoint(N) – moves by measuring joint angles to arrive at a viable trajectory. Move to point N in the USNA1 vector.
5. SBMoveLinear(N) – attempts to move in a straight line between the current position and position N in the USNA1 vector.
6. SBSetJaw(O) – moves the gripper to an opening size O in mm. Range is 0 to 70mm.
7. SBHome() – Sends the robot to it’s default ‘Home’ position. Returns when complete.
8. SBSpeed(P) – Sets robot motion speed relative to its maximum movement speed. P is in percent 0 to 100.
9. SBTime(T) – Attempts to set the amount of time (in msec) to elapse between two movements. T must be greater than zero. Will return an error if the time is too short.
10. SBStop() – Stops ongoing robot motion.
11. Status Related Commands
12. SBGetStatus() Returns a vector of 11 bytes in the following order: byte 1 is the ascii letter ‘S’ (83); byte 2 is a 1 if initialization is complete, otherwise a zero; byte 3 is a 1 if an error occurred during the last command, otherwise it is a zero; byte 4 is a 1 if homing has been completed, otherwise it is a zero; byte 5 is a 1 if motion has been completed, otherwise it is zero; byte 6 is a 1 if the robot is online, otherwise it is a zero; byte 7 is a 1 if the robot is in Teach mode, otherwise it is a zero; bytes 8 and 9 form a 16 bit short indicating the maximum point number in the USNA vector; bytes 10 and 11 form a 16 bit short indicating an error code if an error occurred.

Vector: [‘S’, InitDone, Error, HomeDone, MotionDone ,Online, TeachMode, MaxVecNum(2) ErrorNum(2)]

1. SBWaitStatus() requests a status vector, then waits to receive the vector from the robot controller.
2. SBIsInitDone() returns a 1/0 indicating if initialization is complete..
3. SBIsOnline() returns a 1/0 indicating if the robot controller is ‘on-line’.
4. SBIsTeachMode() returns a 1/0 indicating if the robot controller is in “teach mode”.
5. SBWaitMotionDone() returns when robot motion is reported complete by status. Use this for spacing between multiple commands. Does not return a value