+getStateChains(out chn_q: cell, out chn_dq: cell, obj, chain_names: string[1..*])

+getStateJointIdx(out jnt_q: vector, out jnt_dq: vector, obj, joint_idx: integer[1..*])

+getStateParams(out stParams: wbmStateParams, obj, stChi: vector) +getStateParams(out stParams: wbmStateParams, obj, stChi: matrix) +getPositions(out vqT_b: vector, out q_j: vector, obj, stChi: vector) +getPositions(out vqT_b: matrix, out q_j: matrix, obj, stChi: matrix)

+getMixedVelocities(out v_b: vector, out dq_j: vector, obj, stChi: vector)

+getMixedVelocities(out v_b: matrix, out dq_: matrix, obj, stChi: matrix)
+getBaseVelocities(out v_b: vector, obj, stChi: vector)

+getPositionsData(out stmPos: matrix, obj, stmChi: matrix)

+getBaseVelocities(out v_b: matrix, obj, stChi: matrix)

+get.robot_body(out robot_body: wbmBody, obj)

+set.init_state(obj, stlnit: wbmStateParams)

+get.init_state(out stlnit: wbmStateParams, obj)

-initConfig(obj, robot_config: wbmBaseRobotConfig)

+get.robot_config(out robot_config: wbmBaseRobotConfig, obj)

+get.robot_params(out robot_params: wbmBaseRobotParams, obj)

-checkInitStateDimensions(out result: logical, obj, stInit: wbmStateParams)
-getLinkName(out lnk_name: string, obj, lnk_list: vector, idx: integer)

+get.stvChiInit(out stvChi: vector, obj)

+get.stvLen(out stvLen: integer, obj) +get.vqTInit(out vqT_b: vector, obj)

+get.stvqT(out vqT_b: vector, obj)

+dispConfig(obj, prec: integer)

+getStateJointNames(out jnt_q: vector, out jnt_dq: vector, obj, joint_names: string[1..*])

Tool at a spezified end-effector link, i.e.

0..1

+wbmBody(out obj: wbmBody, chain_names: string[1..*], chain_idx: matrix, joint_names: string[1..*], joint_idx: vector)

hand/finger, with an orientation and

translation relative to the link frame.

+chains.name: string[1..*] {readOnly}

+nChains: integer {readOnly} +joints.name: string[1..*] {readOnly}

+nJoints: integer {readOnly}

+joints.idx: integer[1..*] {readOnly}

+chains.start_idx: integer[1..*] {readOnly}

+chains.end_idx: integer[1..*] {readOnly}

+getChainTable(out chn_tbl: table, obj)

+getJointTable(out jnt tbl: table, obj)

+getChainIndices(out jnt_idx: vector, obj, chain_name: string)

+getJointNames(out jnt_names: string[1..*], obj, joint_idx: vector)

+getJointIndex(out jnt_idx: integer, obj, joint_name: string)

+getStateChains(out chn_q: cell, out chn_dq: cell, obj, chain_names: string[1..*], q_j: vector, dq_j: vector)

+getStateJointIdx(out jnt_q: vector, out jnt_dq: vector, obj, joint_idx: integer[1..*], q_j: vector, dq_j: vector)

+getStateJointNames(out jnt_q: vector, out jnt_dq: vector, obj, joint_names: string[1..*], q_j: vector, dq_j: vector)

-getJointValues(out jnt_q: vector, out jnt_dq: vector, obj, q_: vector, dq_j: vector, joint_idx: integer[1..*], len: integer)

«use»