WBM::WBM-Class

WBMBasic

-wb_params: wbmBasicModelParams

+WBMBasic(wbm_params: wbmBasicModelParams): WBMBasic
+WBMBasic(wbm_params: wbmBasicModelParams, init_wf_ctrl: string): WBMBasic
+copy(): WBMBasic
+initModel()
+initModel(urdf_robot_name: string)
+setWorldFrame(R_rootlnk_wf: matrix, p_rootlnk_wf: vector, g_wf: vector)
+setWorldLink(R_reflnk_wf: matrix, p_reflnk_wf: vector)
+setWorldLink(urdf_link_name: string, R_reflnk_wf: matrix, p_reflnk_wf: vector, g_wf: vector)
+setState(q_j: vector, dq_j: vector, v_b: vector)
+getState(): vector
+massMatrix(): matrix
+massMatrix(R_rootlnk_wf: matrix, p_rootlnk_wf: vector, q_j: vector): matrix
+getJointLimits(): vector
+jacobian(urdf_link_name: string): matrix
+jacobian(urdf_link_name: string, R_rootlnk_wf: matrix, p_rootlnk_wf: vector, q_j: vector): matrix
+dJdq(urdf_link_name: string): vector
+dJdq(urdf_link_name: string, R_rootlnk_wf: matrix, p_rootlnk_wf: vector, q_j: vector, dq_j: vector, v_b: vector): vector
+centrodialMomentum(): vector
+centrodialMomentum(R_rootlnk_wf: matrix, p_rootlnk_wf: vector, q_j: vector, dq_j: vector, v_b: vector): vector
+forwardKinematics(urdf_link_name: string): vector
+forwardKinematics(urdf_link_name: string, R_rootlnk_wf: matrix, p_rootlnk_wf: vector, q_j: vector): vector
+genBiasForces(): vector
+genBiasForces(R_rootlnk_wf: matrix, p_rootlnk_wf: vector, q_j: vector, dq_j: vector, v_b: vector): vector
+getWBMParams(): wbmBasicModelParams
+dispWBMParams(precision: integer)
-initWBM(wbm_params: wbmBasicModelParams, init_wf_ctrl: string)

«interface»
handle

+handle

«interface»
matlab.mixin.Copyable

+copy

Override the copy function to use it like as a copy constructor.

WBM basic parameters for any model of a floating base robot.

«dataType»
wbmBasicModelParams

+urdfRobotName: string
+urdfRefLinkName: string
+R_rootlnk_wf: matrix
+p_rootlnk_wf: vector
+R_reflnk_wf: matrix
+p_reflnk_wf: vector
+g_wf: vector

-wb_params

«dataType»
wbmState

+x_b: vector
+qt_b: vector
+q_j: vector
+dx_b: vector
+omega_b: vector
+dq_j: vector

State parameters of the robot.

«use»

+initState

Initial condition of any floating base robot.

WBM

-wb_config: wbmBasicRobotConfig

+WBM(wbm_params: wbmBasicModelParams, robot_config: wbmBasicRobotConfig): WBM
+WBM(wbm_params: wbmBasicModelParams, robot_config: wbmBasicRobotConfig, init_wf_ctrl: string): WBM
+copy(): WBM
+toODEinitCond(st: wbmState): vector
+getODEinitConditions(): vector
+forwardDynamics(t: integer, ctrlTrqs: array_of_vector, chi: vector): vector
+forwardDynamicsZeroExtForces(t: integer, ctrlTrqs: array_of_vector, chi: vector): vector
+visualizeForwardDynamics(xout: vector, time_span: double, params: struct)
+getWBMConfig(): wbmBasicRobotConfig
+dispWBMConfig(precision: integer)
-initConfig(robot_config: wbmBasicRobotConfig)

-wb_config

«dataType»
wbmBasicRobotConfig

+ndof: integer
+nCstrs: integer
+cstrLinkNames: stringlist
+dampCoeff: double
+initState: wbmState

Since every robot is different, you can specify the configuration parameters of your robot in your own data type.

General configuration structure of a humanoid robot.

«dataType»
wbm_YourSpecificRobotConfig

«dataType»
wbmHumanoidConfig

+pos_head: vector
+pos_torso: vector
+pos_leftArm: vector
+pos_leftHand: vector
+pos_leftLeg: vector
+pos_leftFeet: vector
+pos_rightArm: vector
+pos_rightHand: vector
+pos_rightLeg: vector
+pos_rightFeet: vector