

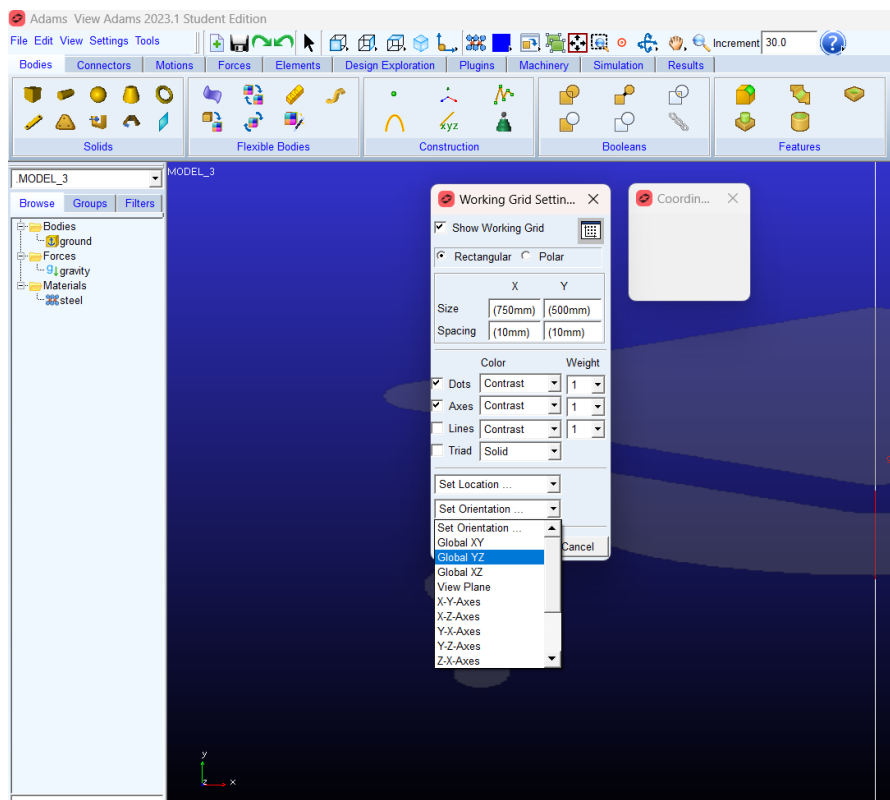
# TIROCINIO

## Braccio robotico

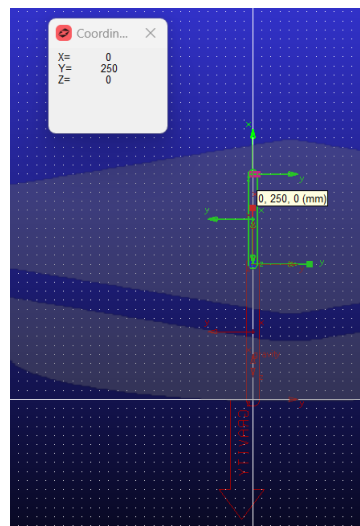
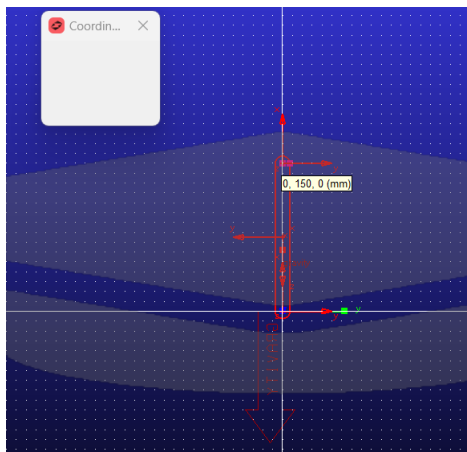
La lezione prevede la realizzazione su Adams di un braccio robotico che esegue come azioni il *pick-and-place*: prende un cubetto da una prima piattaforma e lo appoggia sulla seconda. Si vuole modellare la dinamica del braccio robotico sia con una coppia di controllo che per mezzo del *bushing*. Inoltre vi è da considerare la problematica del *contatto*: due corpi in Adams si compenetrano tra loro se non si specifica una forza di contatto tra loro; essa va impostata con dei parametri che si basano sulla legge di *Hertz* non lineare per il contratto e i parametri di attrito di *Coulomb*.

Si può usare la griglia a proprio vantaggio per la creazione dei links. Andando su in *Settings > Working grid* si osserva come di default la griglia possieda una dimensione di 750x500 ed uno *spacing* (distanza tra i punti) di 50x50. Con *Set Location* si modifica la posizione dell'origine, am è sconsigliato farlo; con *Set Orientation* si modifica il piano su cui deve giacere la griglia. Con *fn+f4* si apre la finestra che indica, se ci si muove con il cursore, le coordinate del punto in cui ci si trova.

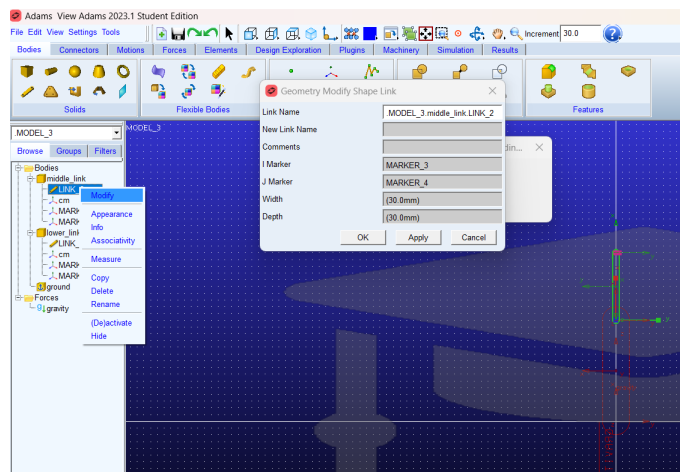
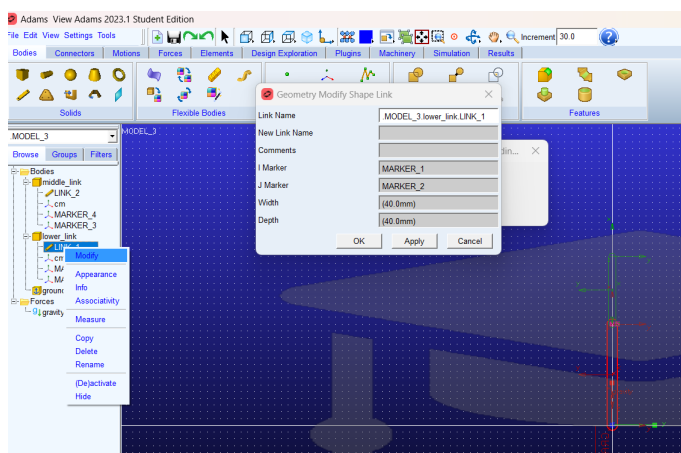
In particolare, si lasciano inalterate le dimensioni della griglia mentre si diminuisce lo *spacing* a 10x10; con *Set Orientation* si posiziona come piano della griglia il piano YZ.



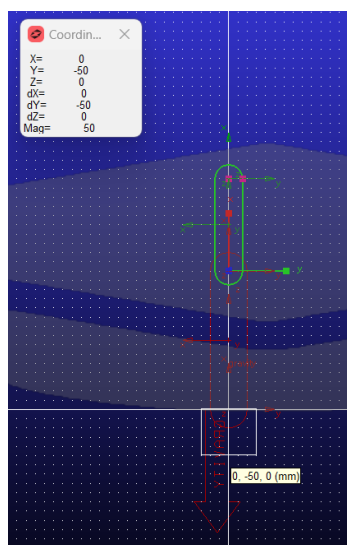
Fatto questo, si può iniziare ad inserire i vari *link* del braccio robotico. Il primo *link*, *lower link*, si estende da (0,0,0) a (0,150,0); il secondo, *middle link*, va da (0,150,0) a (0,250,0);



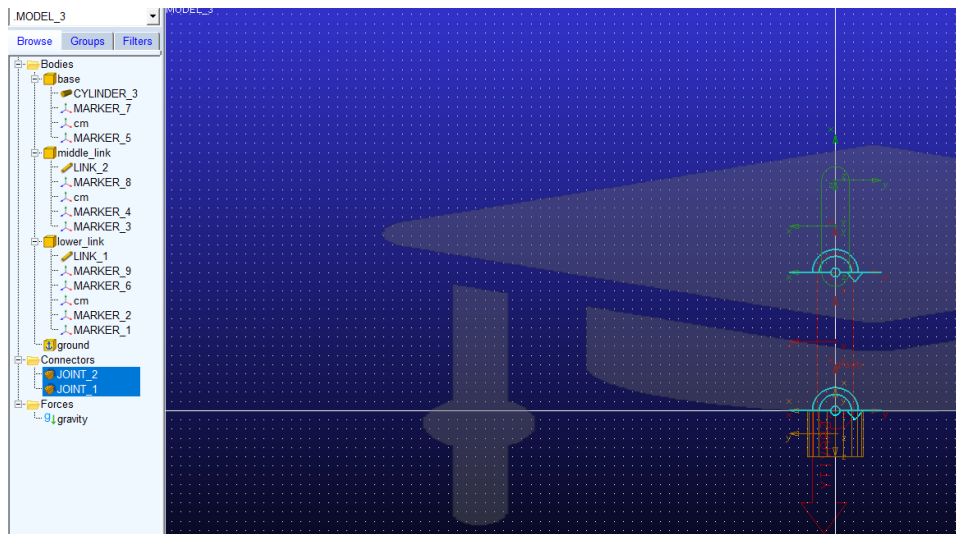
Prima di continuare ad aggiungere gli altri *link*, bisogna modificare le loro *depth* e *width*: il primo a 40 mentre il secondo a 30.



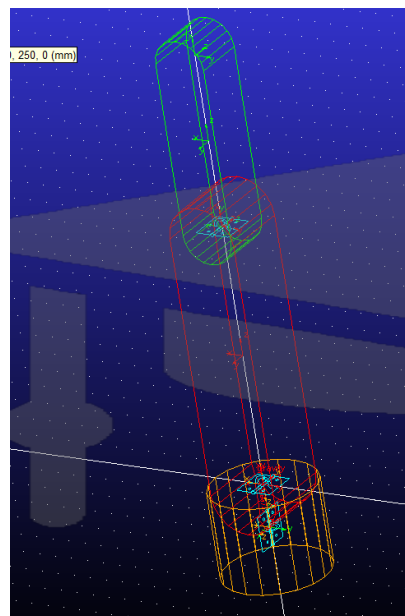
Si inserisce poi un *cilindro*, che si rinomina *base*, avente un raggio di 30 mm e con estensione da (0,0,0) a (0,-50,0).



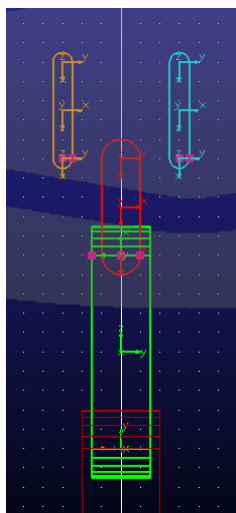
Bisogna creare adesso due *Revolute joints*, realizzate nell'ordine con cui vengono dichiarate, e cioè una tra *lower link* e base localizzata a  $(0,0,0)$  ed un'altra tra *middle link* e *lower link* localizzata a  $(0,150,0)$ . L'ordine con cui si clicca sulle parti è importante perché definisce la direzione positiva di rotazione.



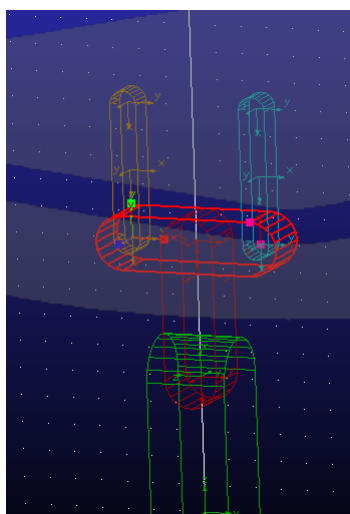
Per fare il *Revolute* tra la *base* e il *ground*, bisogna sapere che primo deve girare attorno ad  $y$  positivo. Si imposta quindi il *ground* con la *working grid* sul piano XZ e si inserisce la cerniera tra *ground* e *base* localizzata a  $(0,-25,0)$ , che corrisponde al *cm* della *base*. Poi si reimposta il *ground* sul piano YZ.



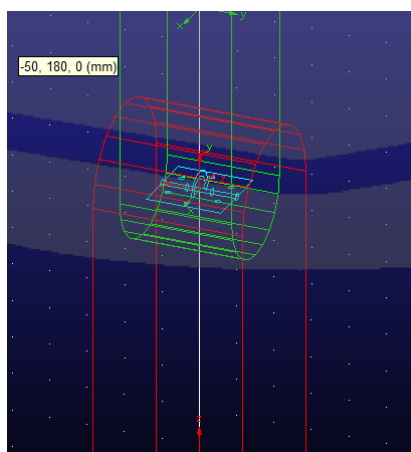
Si continua adesso ad inserire altri link. Il terzo è detto *manipulator base* e va da  $(0,300,0)$  a  $(0,250,0)$ ; il quarto è detto *gripper right* e va da  $(0,350,-30)$  a  $(0,300,-30)$ ; il quinto è detto *gripper left* e va da  $(0,350,30)$  a  $(0,300,30)$ .



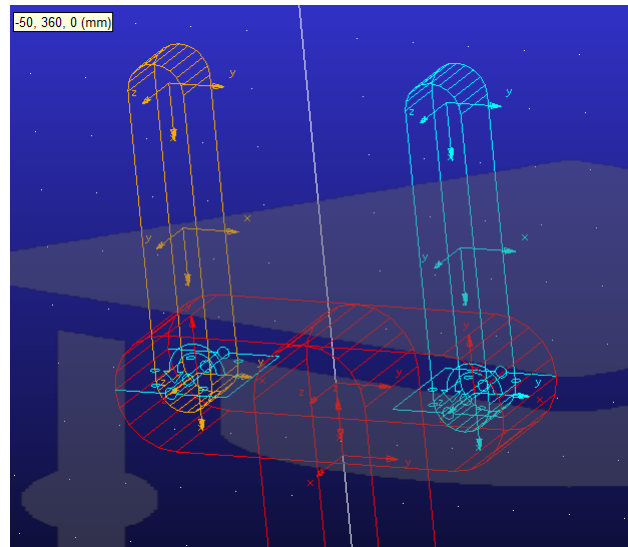
Adesso con l'icona *Move* in alto occorre spostare sia *gripper left* che *gripper right* di 5 mm. Una volta fatto questo, è necessario aggiungere un *link* non me nuova parte ma secondo l'opzione *Add to part*: si clicca prima il *manipulator base* poi i marker dei due *grippers*; infine le altre due dimensioni di questo *link* devono essere portate a 20 mm.



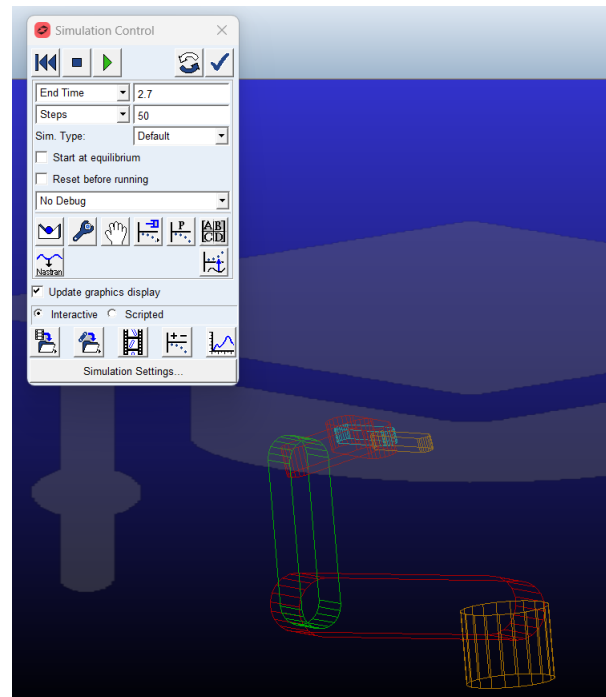
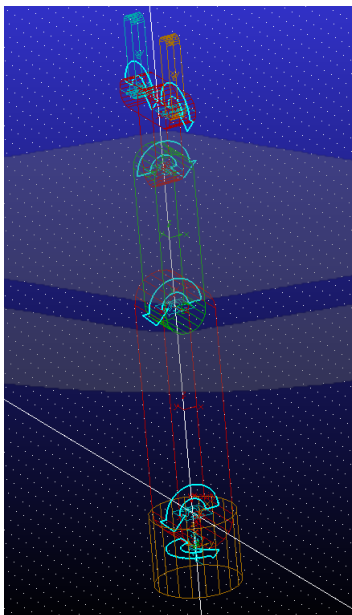
Poi ponendo il *ground* su *Global XY* si aggiunge una *Revolute Joint* tra il *manipulator base* e il *middle link* localizzato a (0,250,0).



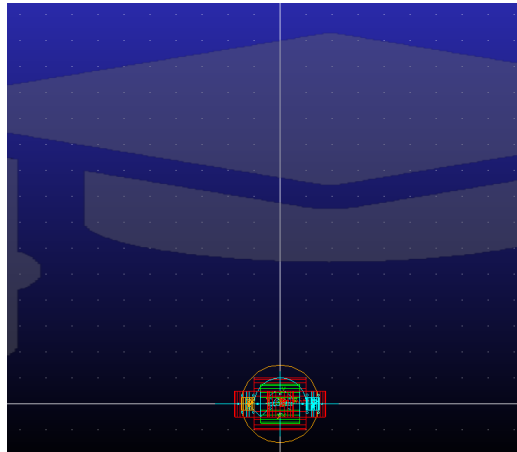
Infine, vanno aggiunte le ultime *Revolution joints*, di cui la prima tra *gripper left* e *manipulator base* e tra *gripper right* e *manipulator base* nei marker dei grippers.



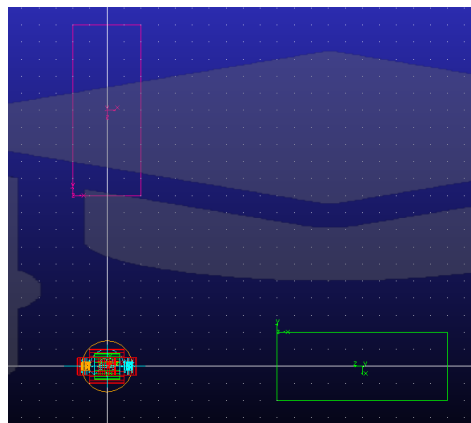
Successivamente vanno inserite le *motions*, una per ciascuna cerniera e ridenominate in modo tale da poter essere correlate alle rispettive cerniere. I versi delle motion (tutte a  $30^\circ/\text{s}$ ) vanno impostati in modo tale che facendo una *simulazione* il braccio robotico raggiunga la configurazione illustrata.



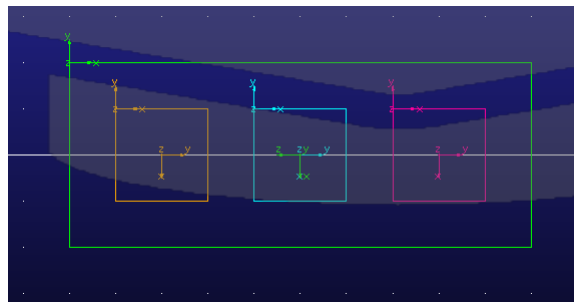
Adesso bisogna inserire le piattaforme con i blocchetti che il braccio robotico deve poter afferrare e trasportare. La griglia va posta su *Global XZ* con *spacing* 20x20; inoltre è consigliato porsi in vista *top*.



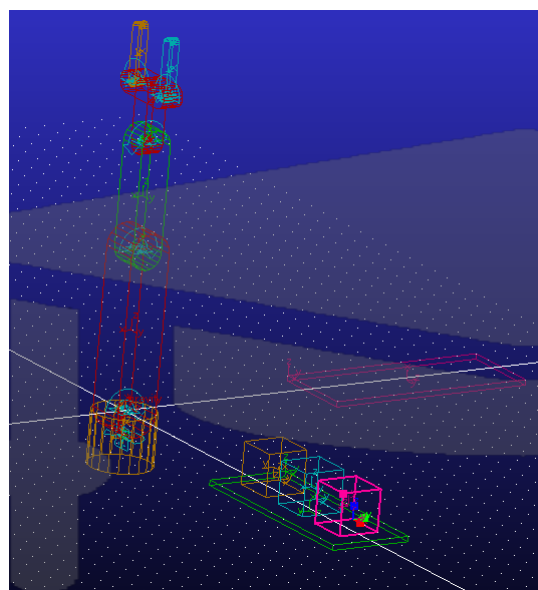
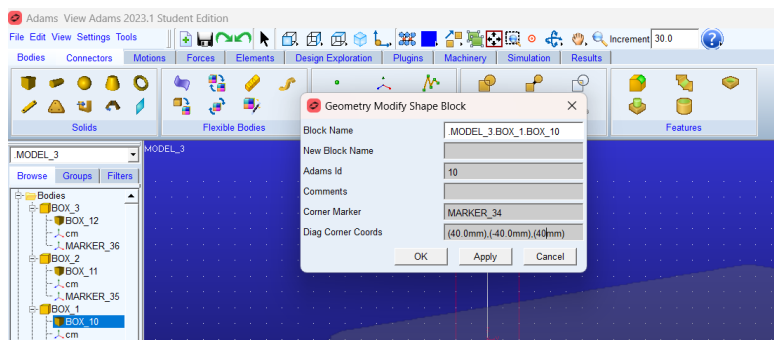
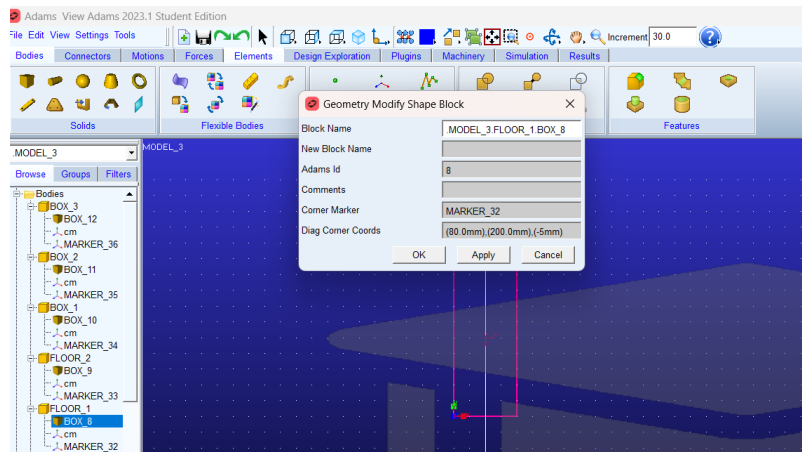
Ora vanno realizzate le piattaforme che devono essere create tramite il tipo di *body* chiamato *box*. La prima box va creata da  $(-40, 200, 0)$  fino a  $(40, 400, 0)$ . La seconda box da  $(200, 40, 0)$  a  $(400, -40, 0)$ .



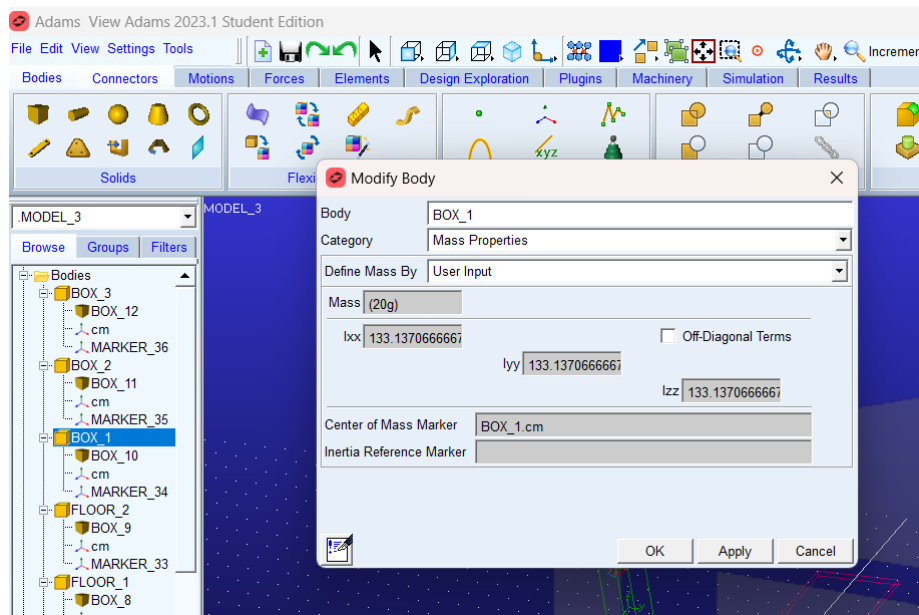
Ora vanno create altre tre box sulla seconda piattaforma. Queste vanno equidistanziate tra loro e opportunamente ridenominate rispettivamente come *BOX1*, *BOX2*, *BOX3*. Anche le piattaforme vanno ridenominate come *FLOOR1* e *FLOOR2*.



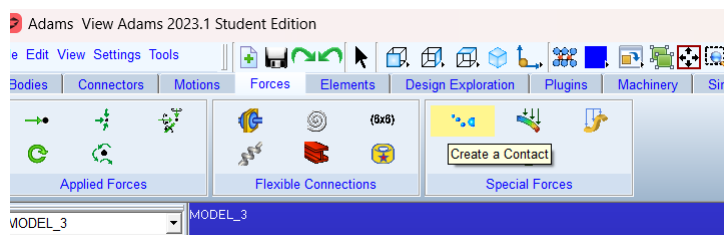
I vari *box* vanno cambiati di spessore lungo la z: in particolare, i due *floor* devono avere -5 mm, mentre i tre cubi 40 mm.



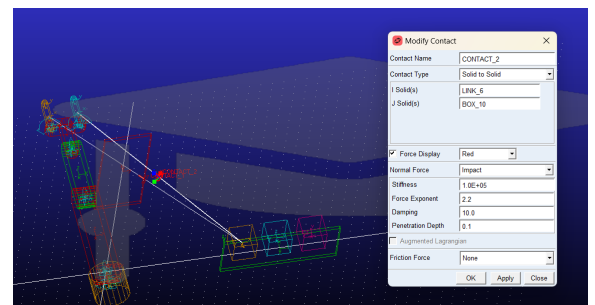
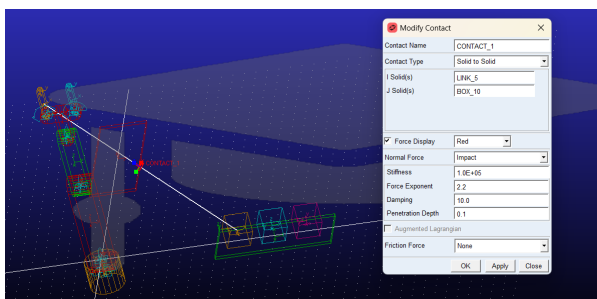
Ora, per assicurare la dinamica, bisogna cliccare *Modify* in riferimento alle tre piccole *box* e impostare la massa a 20 g.



Il passo successivo consiste nella definizione dei *contacts*. Anche se si andasse a definire la *motion* per modellare i grippers ad abbassarsi e ad afferrare i cubi, senza definire prima i contatti, questi verrebbero soltanto compenetrati e basta. Si va allora in *Forces > Contacts* e si crea un contatto (i contatti sono forze, cioè forze di contatto).



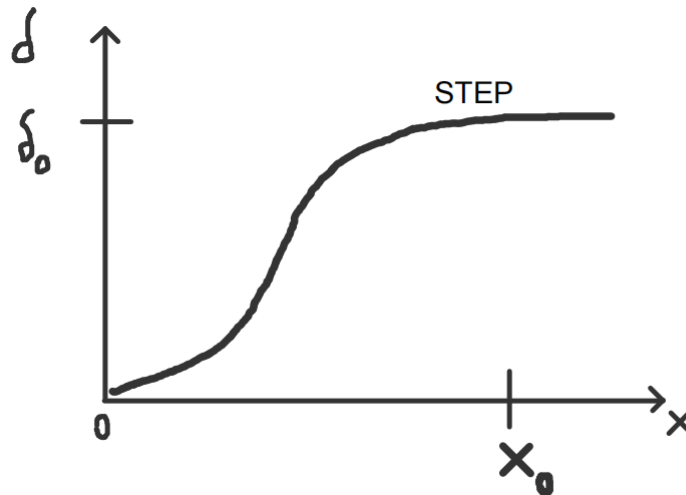
Si apre una finestra in cui come *I solid* va inserito uno dei due gripper (*gripper left* ad esempio) e come *J solid* la *BOX1* e si fa poi lo stesso tra *gripper right* e *BOX1*.





Per i contatti è opportuno andare anche a modificare i parametri di default.

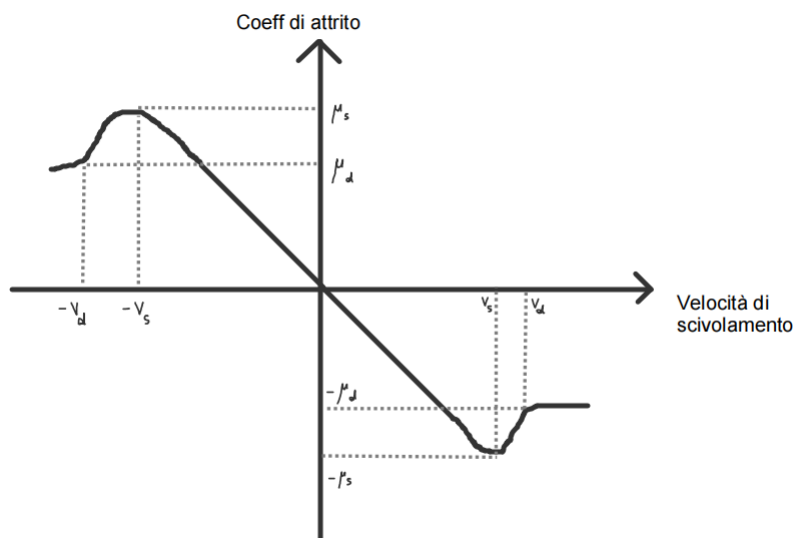
il *damping* serve a far sì che l'impatto non sia quello di un urto elastico ma che si abbia uno smorzamento tale da farlo avvenire più dolcemente. La *stiffness* indica la rigidità dell'impatto mentre l'esponente indica l'esponente di forza che permette di esprimere la non linearità; il valore di damping inserito non viene raggiunto immediatamente ma attraverso una legge step (gradino) altrimenti il solutore arriva alla instabilità (non conviene dare un valore secco altrimenti il solutore crush).



$$F = k x^n + d \dot{x}$$

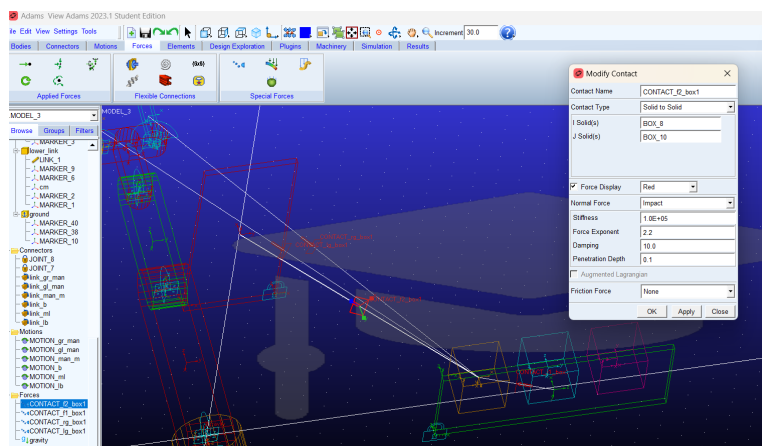
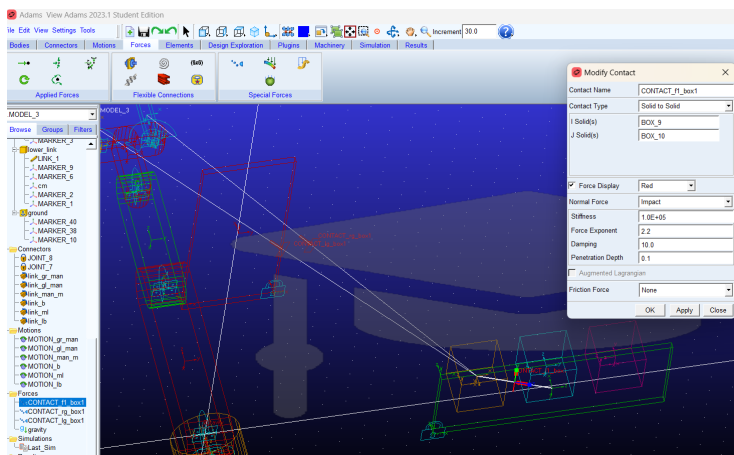
con:

- $x_0$  = profondità di penetrazione;
- $x$  = compenetrazione;
- $\dot{x}$  = velocità di compenetrazione;
- $k$  = stiffness [N/mm in mmKgs];
- $d_0$  = damping [Ns/mm] (in Adams si seleziona il valore massimo  $d_0$  che si raggiunge alla fine dello step,  $x = x_0$ );
- $n$  = force exponent

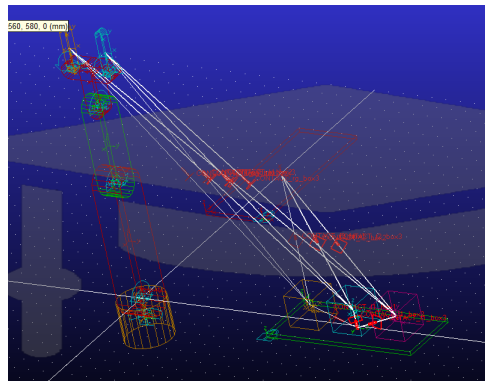


- $\mu_s$  = coefficiente di attrito statico ( $0 \leq \mu_s \leq 1$ );
- $\mu_d$  = coefficiente di attrito dinamico ( $0 \leq \mu_d \leq \mu_s$ );
- $v_s$  = stiction (quiete) transient velocity;
- $v_d$  = friction (moto) transient velocity;

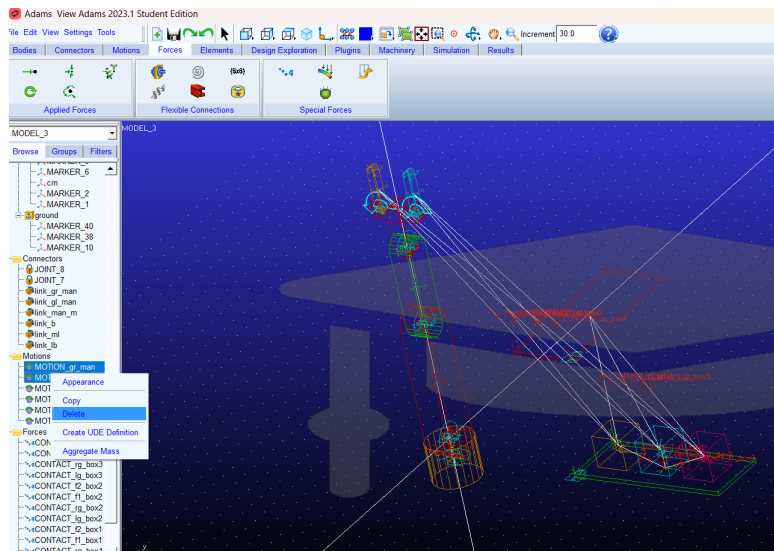
1.0, 0.7, 100, 0



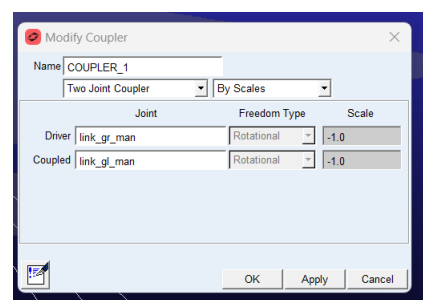
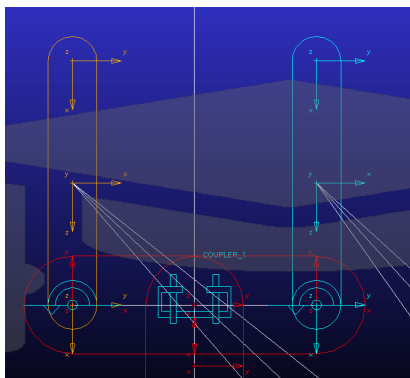
La stessa procedura di costruzione dei contatti tra *box* e *grippers* e tra *box* e *floors* viene eseguita anche per *BOX2* e *BOX3*.



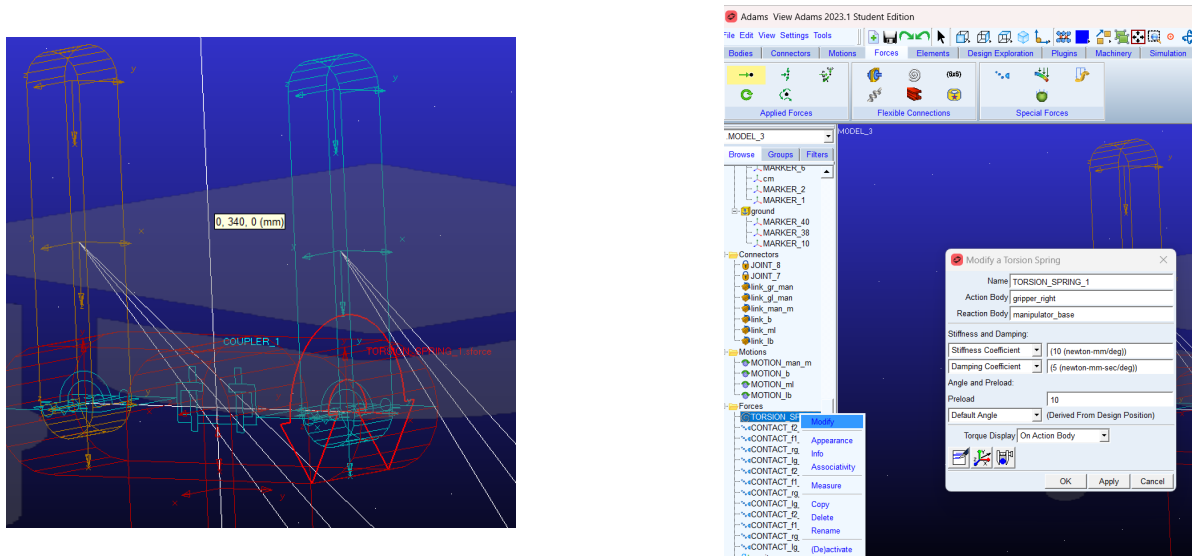
Adesso occorre accoppiare i movimenti dei due *grippers* mediante un *Joint Coupler*. Si pone la griglia sul piano YZ globale e si eliminano le *motions* relative ai *grippers*.



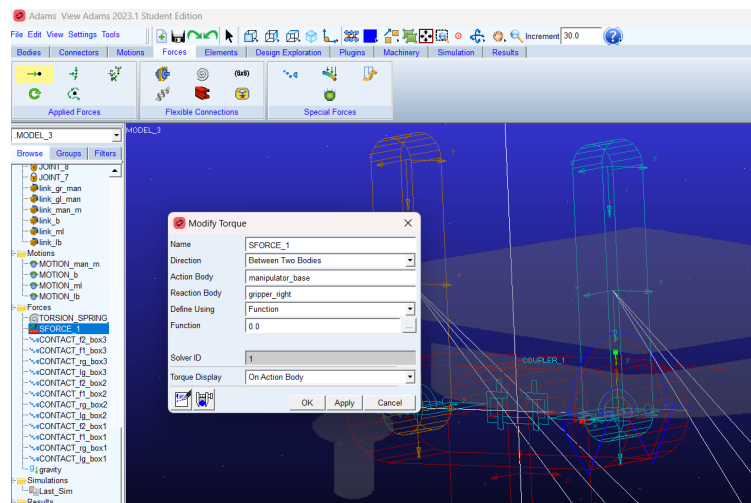
Si va poi su in *Connectors>Couplers* e si seleziona *Joint Coupler*. Bisogna cliccare prima sulla *Revolute joint* del *right gripper* e poi su quella sinistra in modo da definire una *motion coupler*. Una volta inserito va modificato inserendo *scale -1* anche al *joint coupled*. Praticamente assegnando una *motion* al movente anche l'altro si muoverà ugualmente in maniera coordinata.



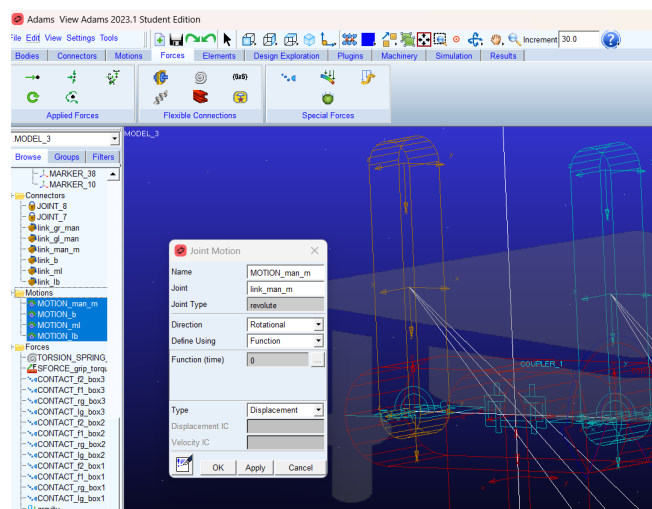
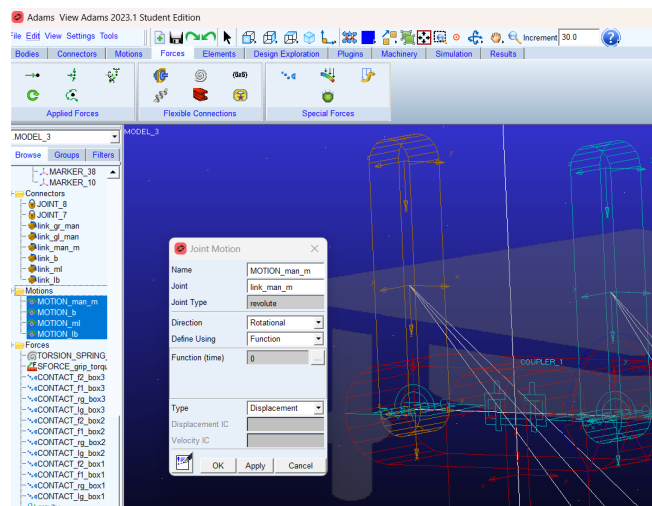
Il movimento di chiusura dei gripper non viene definito con una motion ma mediante una coppia (*torque*) a cui però si deve opporre una molla (*Rotational spring-damper*). La *spring* va inserita cliccando prima sul *gripper right* e poi sul *manipulator base* e localizzandola nel *Revolute Joint*. Modificare poi la *spring* in modo da avere *stiffness* 10, *damping* 5 e *preload* 10.



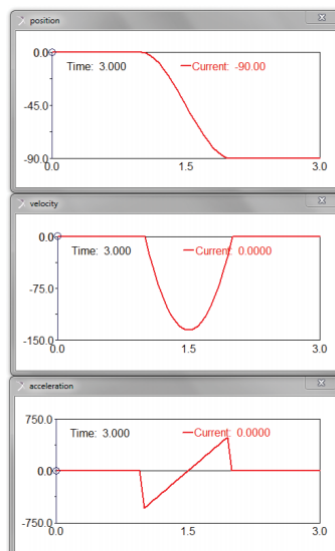
In seguito occorre creare la *Torque*. In teoria la coppia dovrebbe agire tra *gripper right* (su esso agisce l'azione) e *manipulator base* (su di esso la reazione). Usando come opzione *Space Fixed* (che reagisce sul telaio) può risultare vantaggioso. La *SFORCE\_grip\_torque* creata sarà poi modificata per reagire sul *manipulator base*. La sua *appearance* viene modificata in *blue* con *size* 11.



Occorre adesso modificare le altre *motions* e la *SFORCE\_grip\_torque*. Bisogna impostare a 0 tutte le *motions* mentre la funzione della *SFORCE\_grip\_torque* va impostata a 20. Si simuli poi per 3s con 30 steps.



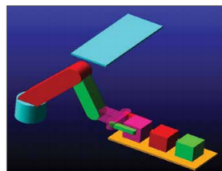
Adesso bisogna definire delle leggi *step* per le *motions*. *STEP* significa che la posizione è un gradino, la velocità presenta derivata crescente e poi decrescente.



Now, motions will be defined with step functions to bring into position to grip cube\_1.

- Define the following step functions. Be sure delete what is already in the function box and select apply in the motion modify window.
- Run a simulation for **1 second, 20 steps**.
- At the end of the simulation, the gripped should be positioned to grip the cube, as shown.

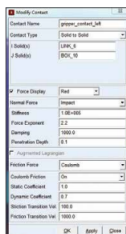
Motions	Function
Motor_1	step(time,0,0,1,-40d)
Motor_2	step(time,0,0,1,-110d)
Motor_3	step(time,0,0,1,60d)
Motor_4	0



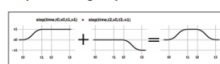
### Step 15. Grip the Cube

Next create a contact and friction force between the grippers and the cube.

- Create a solid to solid contact between **gripper\_left** and **cube\_1**.
- Change the **Static** and **Dynamic Coefficients** as shown.
- Repeat for **gripper\_right** and **cube\_1**.
- Rename each **gripper\_contact\_left/right**.  
Use a step function to set torque equal to 0 from 0 to 1-sec, allowing the spring the keep the grippers in the slightly open position, then apply 8000 N\*m when in position.
- Modify **grip\_torque's** function to be **-step(time,1,0,1,-8000)**
- Simulate for **1.1 seconds, 55 steps**. Confirm that the gripper makes contact with **cube\_1** as shown.



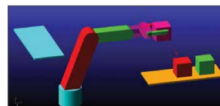
### Step 16. Adding Steps



- Add the following steps to the specified motion functions:

Motions	Step to Add
Motor_1	step(time,1.1D,2,20d)
Motor_2	step(time,1.1D,2,40d)
Motor_3	step(time,1.1D,2,-60d)

- Simulate for **2 seconds, 40 steps**.
- Verify that your simulation matches with the image



### Step 17. Finalize Definition of Motions and Torques

The chart below describes the necessary step values required to complete the entire operation. Try to figure out the missing values on your own. Recall that  $x$  values are the independent value (time) and the  $h$  values are the dependent variables (position/torque) and the step for the step function (with time defined as  $X$ ) is  $\text{step}(\text{time}, x_0, h_0, x_1, h_1)$

Measurement Description	Accuracy Score						Step5 Score	Step6 Score	Step7 Score	Step8 Score	Step9 Score	Step10 Score
	Step1 Metric	Step2 Metric	Step3 Metric	Step4 Metric	Step5 Metric	Step6 Metric						
Time Value	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
Memory Value	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0
Accuracy Value	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
Loss Value	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
Score Value	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
Step1 Value	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
Step2 Value	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
Step3 Value	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
Step4 Value	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
Step5 Value	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
Step6 Value	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
Step7 Value	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
Step8 Value	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
Step9 Value	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1
Step10 Value	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1

Add to motion definitions one step at a time and simulating to verify the model behaves as expected.

Note that all  $h_0$  values are zero and  $h_1$  always describes motion relative to the current position.

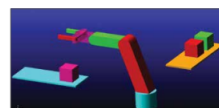
The necessary motion to place the cube on the platform and release it is essentially the reverse of picking it up. In other words, steps 3, 5, and 7 are very similar.

For example, the final function definition of `motor_2` is:

```
step(time,0,0,1,-110d) + step(time,1,1,0,2,40d) +
step(time,2,0,3,-40d) + step(time,4,1,0,5,40d)
```

Step 4 is simple a rotation of the base to bring the cube over platform. 2. Note that its x values overlap steps 2 & 5

Try to simulate the whole operation (0-4 seconds) with your values. If you are having trouble, continue to the next step.



### Step 18. Finalize Definition of Motions and Torques

Microcontroller	Actively flexed						
	Step1 Move into position	Step2 Grip the tube	Step 3 Lift	Step4 Move to platform	Step5 Lower into place	Step6 Release	Step7 Rotate manipulator
Time values	x1	x2	x3	x4	x5	x6	x7
Values	1	1.1	2	2	2.6	5.5	6.1
Min/Max/ Average values	50	50	50	50	50	50	50
maxima	0	400	N/A	0	200	N/A	0
minima	0	1100	N/A	0	400	N/A	0
maxima	0	400	N/A	0	200	N/A	0
minima	0	1100	N/A	0	400	N/A	0
maxima	N/A	N/A	N/A	N/A	0	1000	N/A
minima	N/A	N/A	N/A	N/A	0	1000	N/A
maxima	N/A	N/A	0	8000	N/A	0	4000
minima	N/A	N/A	0	8000	N/A	0	4000

Final list of motion/torque functions:

Motions	Function
Motor_1	step(time,0.0,1,-40d) + step(time,1.0,2.0,0d) + step(time,2.0,3.-20d) + step(time,3.0,4.0,20d)
Motor_2	step(time,0.0,1,-110d) + step(time,1.0,2.40d) + step(time,2.0,3.-40d) + step(time,3.0,4.0,40d)
Motor_3	step(time,0.0,1,60d) + step(time,1.0,2.-60d) + step(time,2.0,3.60d) + step(time,3.0,4.-60d)
Motor_4	step(time,1.5,0.2,5,90d)
gripper_torque	step(time,1.0,1.1,8000) + step(time,3.0,3.1,-8000)

### Step 19. Optional Tasks

Torque Demand

- Switch to **PostProcessor** and examine the results of the simulation.
- Look at torque demands (**Source:Objects >> motor\_xxx >> Element Torque >> Mag**), and gripper contact forces (**Source:Objects >> gripper\_contact\_[left/right] >> Element Torque >> Mag**).
- Note the sporadic spikes in torque required to maintain the smooth step motion.
- Switch back to **View** and increase the **contact damping** to **100** and re-simulate.
- How do the torque demands and contact forces look now?

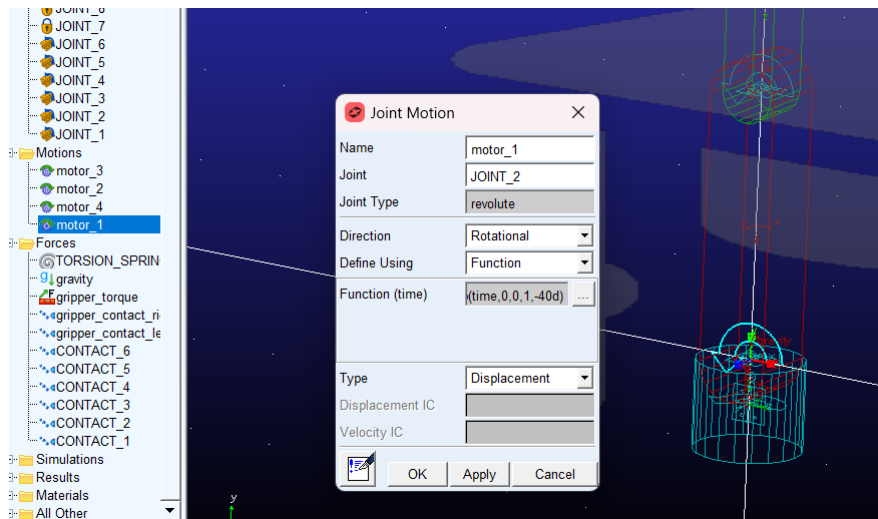
### Move the Remaining Cubes

For simplicity, the robot sits the block down in the same position on platform 2 as it was on platform 1.

- Use sketch paper to derive the necessary angles to sit the block near the far edge of platform 2, where cube 3 is on platform 1.
- Try to create the additional steps necessary to move the remaining blocks. Derive the necessary angles by hand or use trial and error to determine the correct values.

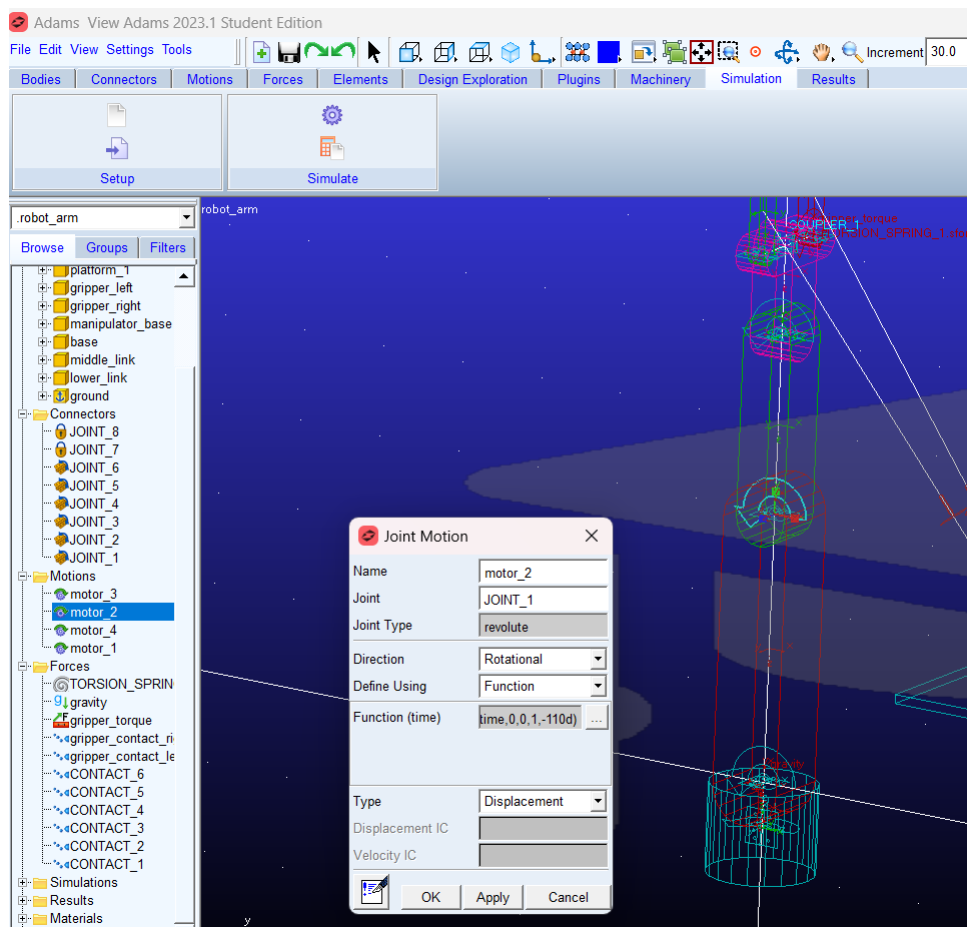
Per la *motion* (*motor 1*) tra *lower link* e *base* va inserita come legge step:

$$\text{step}(\text{time}, 0, 0, 1, -40d)$$



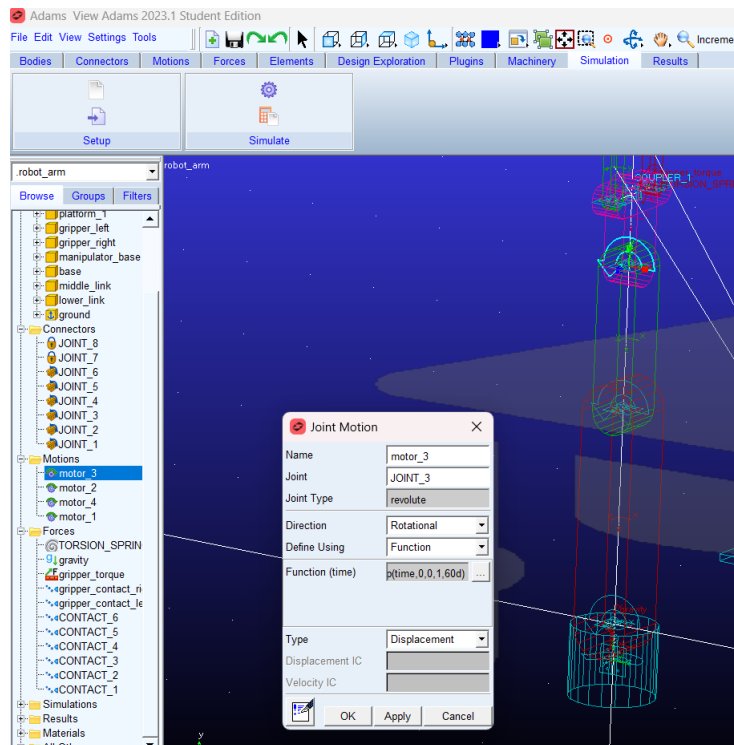
Per la *motion* (*motor 2*) tra *middle link* e *lower link* va inserita come legge step:

$$\text{step}(\text{time}, 0, 0, 1, -110d)$$



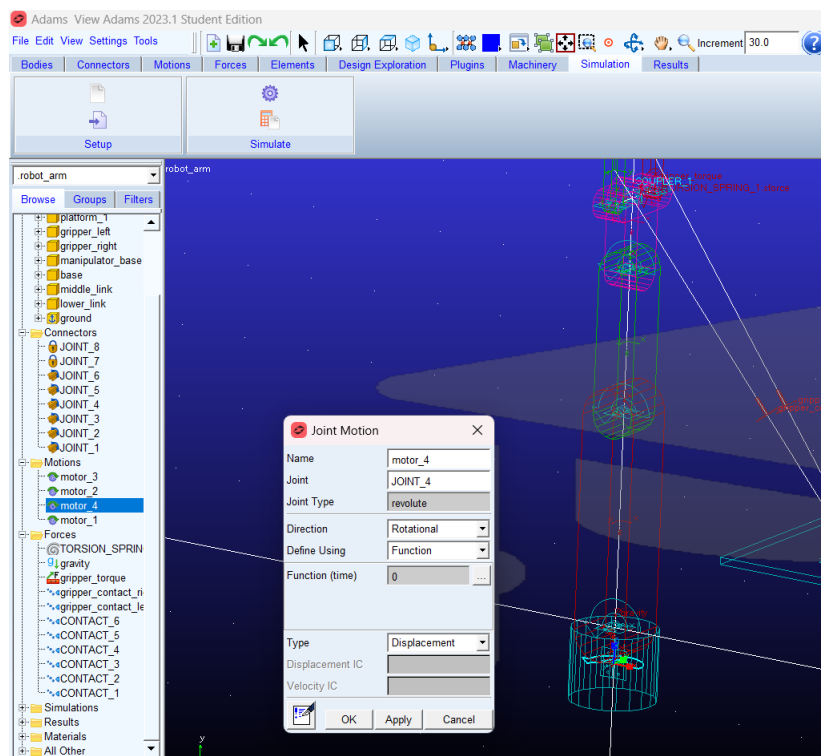
Per la *motion* (*motor 3*) tra *manipulator base* e *middle link* va inserita come legge step:

$$\text{step}(\text{time}, 0, 0, 1, 60d)$$



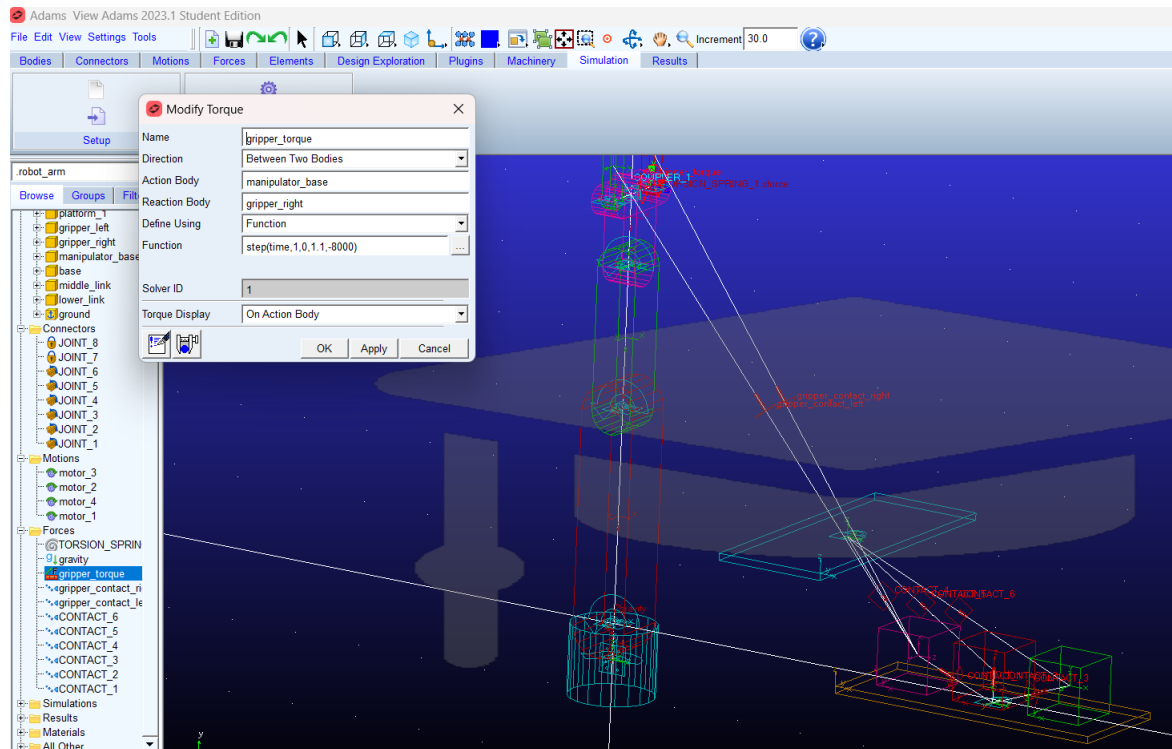
Per la *motion* (*motor 4*) nella base va inserita come *function*:

0





In seguito va imposta le legge step anche sulla coppia:



Di seguito viene illustrato in che modo vanno aggiunti i vari step per i motori e per la coppia in modo da avvicinarsi al BOX1, afferrarlo, sollevarsi, ruotare, abbassarsi, rilasciare e sollevarsi.

