Robot Control Interface - Bug #1043
Modelisation of Wrench seems wrong

06/28/2012 11:15 AM - A. Tuleu

Status:	Feedback	**Start date:**	06/28/2012
Priority:	High	**Due date:**	
Assignee:	A. Tuleu	**% Done:**	80%
Category:	Modeling	**Estimated time:**	0.00 hour
Target version:	rci0.4		

Description

Modelisation of wrench seems wrong in RCI. Wrench store Torque as quaternion. However queternion are for storing rotation, that leaves in space SO (3). Rotation are not additive (you only combine them).

However a torque is a simply a linear vector living in space R^3 therefore, you should store them in this space, and not as quaternion. Wrench live in a more complex state, since they are (naively) defined by 9 dimension :

- a translational force
- a rotational force
- a point of origin

but actually these 3 values are bind together by the rule of transport.

Related issues:

Related to Robot Control Interface - Feature # 1715: Provide better interface...	**Resolved**	**01/06/2014**

Associated revisions

Revision 719 - 01/16/2014 07:02 PM - C. Emmerich

refs #1043, refs #1715 implemented better rci::Wrench interface

History

#1 - 06/28/2012 11:15 AM - A. Tuleu

- Description updated

#2 - 08/12/2013 02:08 PM - Anonymous

- Status changed from New to Feedback

- Target version set to rci0.4

#3 - 08/22/2013 01:36 PM - Anonymous

- Status changed from Feedback to Rejected

- Assignee set to Anonymous

Wrench represents force in free space, separated into its linear and angular parts. In this sense it is compatible with ROS, Orocos and rst.

#4 - 01/05/2014 05:58 PM - C. Emmerich

- Status changed from Rejected to Feedback

- Priority changed from Normal to High

I think, in some aspect Alexandre was right: Of course a wrench consists of a translational component (forces Fx, Fy, Fz) and a rotational component

(torques). But why do you store the torques of a wrench as quaternions or why don't you allow the creation of wrenches with 3 values? If the rci::Wrench type shall go in line with ROS, kdl and rst types, torques should be stored as or at least be constructable from a 3 dim vector or torque.

For instance, the KUKA LWR delivers as estimated external forces acting on the endeffector:

```
/** estimated TCP force/torque (N, Nm)
     KRL: $TORQUE_TCP_EST
    - reference frame: $STIFFNESS.TASKFRAME
    - Layout: Fx, Fy, Fz, Tz, Ty, Tx
    */
   fri_float_t estExtTcpFT[FRI_CART_VEC];
```

#5 - 01/06/2014 11:05 AM - Anonymous

- Category set to Modeling

- Assignee changed from Anonymous to C. Emmerich

Christian Emmerich wrote:

> *... why do you store the torques of a wrench as quaternions ...?*

This was just the basic decision in RCI to go with quaternions for rotational values as done for orientation/rotation as well. As long as this doesn't break anything, I would stick with this.

> *... why don't you allow the creation of wrenches with 3 values?*

This wasn't an active decision, but is just a missing feature. Would adding this feature meet your needs?

#6 - 01/06/2014 11:57 AM - C. Emmerich

- Assignee changed from C. Emmerich to Anonymous

Arne Nordmann wrote:

> *Christian Emmerich wrote:*
>
> > *... why do you store the torques of a wrench as quaternions ...?*
>
> *This was just the basic decision in RCI to go with quaternions for rotational values as done for orientation/rotation as well. As long as this doesn't break anything, I would stick with this.*

That's fine with me. Just two remarks: 1) rst goes with 3 dim values for torques in a wrench 2) Nm as unit in my opinion does not make sense with quaternions, i.e. you may offer a user-friendly interface for creating a torque-wrench from interpretable units/representations as you do e.g. for Poses.

> *... why don't you allow the creation of wrenches with 3 values?*

> *This wasn't an active decision, but is just a missing feature. Would adding this feature meet your needs?*

Yes, please :D I created an issue for this: #1715.

Then, from my point of view, this issue can be closed.

#7 - 01/09/2014 01:26 PM - A. Tuleu

Hi,

I still not agree with your decision. This is just a bad modelling / understanding what a torque and rotation are !

Rotation are member of the SO space. SO is not R^3 It is a **group** in linear algebra term using the the **composition** operator. It means you do not add elements of this group (the mathematical addition of element of SO does not give you an element of SO) . You can only combine them using **composition**. Now if you represent the element of SO using unit quaternion, it means that the only valid operation you can do on them is **multiplication**. Basically quaternions are just a mathematical artefact to help you numerically use element of SO. Here you could use angle axis, Rotation matrices as you wish. I agree quaternion is a valid choice to represent rotation, even if always (and will always be) questionnable. Providing simple interface to go from / to angle axis and matrices is a really wise decision, as you will never make everybody agree that quaternion are the best.

Now lets go again to torque. A torque is an element of R^3. YOu cannot mathematically compose torque. This is a clear non-sense. They lie in the euclidian space R^3. Therefore you cannot multiply them (there is no multiuply operation for a torque. you can use the cross product, but it does not mean anything physically to do the cross product of two torque). Representing them as quaternion and multiply these object, is in fact doing a composition of the two forces. BAsically it does not mean anything at all. because you do not appply torque 1, and apply torque 2 on the result of the application of torque 1. These are jsu additive elements. The only mathematically valid operation you can do on torque, that still give you a torque are multiplication by a scalar and addition with another torque.

Furthermore torque are quite complex object. Indeed in french they are called "couple" like beeing in couple. Why because the only way practical way to produce a pure torque at a given point is to appply a **couple** of linar counter-acting forces at two different poinst. In the middle of the two application point they will produce a pure couple, with no linear forces. Basically torque does not exist and are just mathematical object that tool to illustrate Archimede's cantilever effect. What I want to illustrate here is that a torque could not be not tied to a linear forces. They are always tied to such force. We always speak of the torque of a given (linear) force. Only in teh special case of a net torque (when the sum of the (linear) forces are zero), this torque will be invariant in space. otherwise we have to use the transport law (http://en.wikipedia.org/wiki/Torque see net versus torque section). Therefore they does not share the same algebra at all.

I still think using quaternion as opposed to just raw vector for torque is a **huge** mistake and a misunderstanding of kinematic and dynamic. You should definitively change this. Or at least I will for myself not use RCI anymore if such mistake want to be actively and knowingly kept in the code base.

#8 - 01/09/2014 01:26 PM - A. Tuleu

Alexandre Tuleu wrote:

> *Hi,*
>
> *I still not agree with your decision. This is just a bad modelling / understanding what a torque and rotation are !*
>
> *Rotation are member of the SO (3) space. SO (3) is not R^3 It is a **group** in linear algebra term using the the **composition** operator. It means you do not add elements of this group (the mathematical addition of element of SO does not give you an element of SO) . You can only combine them using **composition**. Now if you represent the element of SO (3) using unit quaternion, it means that the only valid operation you can do on them is **multiplication**. Basically quaternions are just a mathematical artefact to help you numerically use element of SO. Here you could use angle axis, Rotation matrices as you wish. I agree quaternion is a valid choice to represent rotation, even if always (and will always be) questionnable.*

> *Providing simple interface to go from / to angle axis and matrices is a really wise decision, as you will never make everybody agree that quaternion are the best.*
>
> *Now lets go again to torque. A torque is an element of R^3. YOu cannot mathematically compose torque. This is a clear non-sense. They lie in the euclidian space R^3. Therefore you cannot multiply them (there is no multiuply operation for a torque. you can use the cross product, but it does not mean anything physically to do the cross product of two torque). Representing them as quaternion and multiply these object, is in fact doing a composition of the two forces. BAsically it does not mean anything at all. because you do not appply torque 1, and apply torque 2 on the result of the application of torque 1. These are jsu additive elements. The only mathematically valid operation you can do on torque, that still give you a torque are multiplication by a scalar and addition with another torque.*
>
> *Furthermore torque are quite complex object. Indeed in french they are called "couple" like beeing in couple. Why because the only way practical way to produce a pure torque at a given point is to appply a* ***couple*** *of linar counter-acting forces at two different poinst. In the middle of the two application point they will produce a pure couple, with no linear forces. Basically torque does not exist and are just mathematical object that tool to illustrate Archimede's cantilever effect. What I want to illustrate here is that a torque could not be not tied to a linear forces. They are always tied to such force. We always speak of the torque of a given (linear) force. Only in teh special case of a net torque (when the sum of the (linear) forces are zero), this torque will be invariant in space. otherwise we have to use the transport law (http://en.wikipedia.org/wiki/Torque see net versus torque section). Therefore they does not share the same algebra at all.*
>
> *I still think using quaternion as opposed to just raw vector for torque is a* ***huge*** *mistake and a misunderstanding of kinematic and dynamic. You should definitively change this. Or at least I will for myself not use RCI anymore if such mistake want to be actively and knowingly kept in the code base.*

#9 - 01/09/2014 01:27 PM - A. Tuleu

Sorry I tried to edit my message. PLease read SO - parenthesis- 3 ~~parenthesis~~ instead of SO

Alexandre Tuleu wrote:

> *Hi,*
>
> *I still not agree with your decision. This is just a bad modelling / understanding what a torque and rotation are !*
>
> *Rotation are member of the SO space. SO is not R^3 It is a* ***group*** *in linear algebra term using the the* ***composition*** *operator. It means you do not add elements of this group (the mathematical addition of element of SO does not give you an element of SO) . You can only combine them using* ***composition****. Now if you represent the element of SO using unit quaternion, it means that the only valid operation you can do on them is* ***multiplication****. Basically quaternions are just a mathematical artefact to help you numerically use element of SO. Here you could use angle axis, Rotation matrices as you wish. I agree quaternion is a valid choice to represent rotation, even if always (and will always be) questionnable. Providing simple interface to go from / to angle axis and matrices is a really wise decision, as you will never make everybody agree that quaternion are the best.*
>
> *Now lets go again to torque. A torque is an element of R^3. YOu cannot mathematically compose torque. This is a clear non-sense. They lie in the euclidian space R^3. Therefore you cannot multiply them (there is no multiuply operation for a torque. you can use the cross product, but it does not mean anything physically to do the cross product of two torque). Representing them as quaternion and multiply these object, is in fact doing a composition of the two forces. BAsically it does not mean anything at all. because you do not appply torque 1, and apply torque 2 on the result of the application of torque 1. These are jsu additive elements. The only mathematically valid operation you can do on torque, that still give you a torque are multiplication by a scalar and addition with another torque.*
>
> *Furthermore torque are quite complex object. Indeed in french they are called "couple" like beeing in couple. Why because the only way practical way to produce a pure torque at a given point is to appply a* ***couple*** *of linar counter-acting forces at two different poinst. In the middle of the two application point they will produce a pure couple, with no linear forces. Basically torque does not exist and are just mathematical object that tool to illustrate Archimede's cantilever effect. What I want to illustrate here is that a torque could not be not tied to a linear forces. They are always tied to*

> *such force. We always speak of the torque of a given (linear) force. Only in teh special case of a net torque (when the sum of the (linear) forces are zero), this torque will be invariant in space. otherwise we have to use the transport law (http://en.wikipedia.org/wiki/Torque see net versus torque section). Therefore they does not share the same algebra at all.*
>
> *I still think using quaternion as opposed to just raw vector for torque is a **huge** mistake and a misunderstanding of kinematic and dynamic. You should definitively change this. Or at least I will for myself not use RCI anymore if such mistake want to be actively and knowingly kept in the code base.*

#10 - 01/20/2014 08:59 AM - C. Emmerich

- Assignee changed from Anonymous to A. Tuleu

- % Done changed from 0 to 80

Hey Alexandre,

I discussed this issue with Arne (unfortunately, we did not do this here, but in issue #1715) and we decided to represent a wrench consisting of 3 dim force F = (fx, fy, fz) component and 3 dim torque component T = (tx, ty, tz) representing the axis of the momentum.

I basically implemented this in r719 and r720, you may have a look if this fits your needs and close this issue or ,in case it does not, extend the interfaces according to your desires.