E S

Link ~~ends~~ ~~setup~~

to match everything else

To define an observation model, the various bodies, spacecraft, ground stations, etc. involved in the observation, and their role in the observation, must be defined. For a one-way range model, for instance, ~~a~~ the definition of a receiver and a transmitter is required. In Tudat, the transmitter and receiver are both referred to as 'link ends'. The full set of link ends in the observable, and their role in the observable, are stored in the `LinkDefinition`. Below, we describe the steps that are (or may be) required to set up this object.

maybe good to put more emphasis on SINGLE/PARTICULAR observable

good, maybe more frequent 'note' boxes'?

Note

A `LinkDefinition` object does *not* define the observation model itself, but only the various reference points (link ends) that are required for it. For instance, a one-way range, one-way Doppler and angular position observation may all use an identical `LinkDefinition` (containing a transmitter and a receiver).

Ground Station Creation

Often, you will need to define the positions of ground stations on celestial bodies to/from which observations are made. Note that in Tudat, a planetary lander is treated identically to a terrestrial ground station. The creation of a ground station is described in the environment setup.

Creating a Set of Link Ends

The creation of the link definition requires the definition of a set of link ends used for a given observable. These are stored in a dictionary as follows:

- The dictionary key denotes the role in the observation (e.g. receiver, transmitter, *etc.*), given by an entry from the `LinkEndType` enum. For each observation model in the API documentation, it is specified which link end typs are required.
- The dictionary value represents the identifier of the link end (spacecraft, ground station, *etc.*), as a `LinkEndId` object. To use a reference point on a body (for instance, a ground station on Earth), the `body_reference_point_link_end_id()` function can be used to create an object of this type. To use the origin (typically, but not necesarilly its center of mass) of a body as link end, use the `body_origin_link_end_id()` function. Although using a center of mass is unrealistic for data analysis, such a setup can often be useful for a simulated analysis. Example of defining link ends are given below:

WHY?! Wrap this in function, I just want to give the name

Link to comprehensive table / cheat sheat?

Each type of observable requires a specific combination of *types* of link ends. Below, a number of examples are given for one-, two- and three-way observables (see here for the distinction between two- and three-way observables when creating observation models):

```
one_way_link_ends = dict( );
one_way_link_ends[ transmitter ] =
estimation_setup.observation.body_reference_point_link_end_id( "Earth", "Graz"
);
one_way_link_ends[ receiver ] =
estimation_setup.observation.body_origin_link_end_id( "LRO" );
```

This defines a link for which the ground station termed Graz on the body called Earth acts as transmitter, and the body called LRO is used as the receiver (in this case placed at the body's center of mass).

An example of link-ends for a two-way link from Graz to LRO and back to Graz is:

```
two_way_link_ends = dict( );
two_way_link_ends[ transmitter ] =
estimation_setup.observation.body_reference_point_link_end_id( "Earth", "Graz"
);
two_way_link_ends[ reflector ] =
estimation_setup.observation.body_origin_link_end_id( "LRO" );
two_way_link_ends[ receiver ] =
estimation_setup.observation.body_reference_point_link_end_id( "Earth", "Graz"
);
```

Where the Graz station now acts as both transmitter and receiver. Similarly, the receiver may be different from the transmitter (in what is typically called a three-way observable in Deep Space tracking), so:

```
two_way_link_ends = dict( );
two_way_link_ends[ transmitter ] =
estimation_setup.observation.body_reference_point_link_end_id( "Earth", "Graz"
);
two_way_link_ends[ reflector ] =
estimation_setup.observation.body_origin_link_end_id( "LRO" );
two_way_link_ends[ receiver ] =
estimation_setup.observation.body_reference_point_link_end_id( "Earth",
"Matera" );
```

where the signal is transmitter by Graz station, retransmitter or reflected by LRO, and then received by the Matera station.

In addition to this manual creation of link ends, we also have a number of functions that allow you to generate a list of link ends for one- two- and three-way observables (`one_way_downlink_link_ends()` , `one_way_uplink_link_ends()` , `two_way_link_ends()` , `three_way_link_ends()`).

explain and put before the manual stuff

After the creation of the link ends dictionary, the `LinkDefinition` object can be created as:

```
two_way_link_ends = ..
two_way_link_definition = estimation_setup.link_definition( two_way_link_ends
)
```

where, for this basic example, the link definition is simply a wrapper class for the link ends.