

# ZIPload

From GridLAB-D Wiki

**TODO:** This page needs to be completed.

**ZIPload** - Residential ZIPload (explicit model)

## Synopsis

The ZIPload model is designed to provide a means to access the Residential\_enduse class at a fundamental level and from a "power engineer" perspective. It uses a classic ZIP load model (constant impedance, current, and power) where "base power" is specified, then the ZIP fractions and power factors are assigned. The ZIP model equations can be defined as:

$$P_i = \frac{|V_a^2|}{|V_n^2|} * |S_n| * Z_{\%} * \cos(Z_{\theta}) + \frac{|V_a|}{|V_n|} * |S_n| * I_{\%} * \cos(I_{\theta}) + |S_n| * P_{\%} * \cos(P_{\theta})$$

$$Q_i = \frac{|V_a^2|}{|V_n^2|} * |S_n| * Z_{\%} * \sin(Z_{\theta}) + \frac{|V_a|}{|V_n|} * |S_n| * I_{\%} * \sin(I_{\theta}) + |S_n| * P_{\%} * \sin(P_{\theta})$$

where:

- $P_i$ : Real power consumption of the ith load
- $Q_i$ : Reactive power consumption of the ith load
- $V_a$ : Actual terminal voltage
- $V_n$ : Nominal terminal voltage
- $S_n$ : Apparent Power consumption at nominal voltage
- $Z_{\%}$ : Percent of load that is constant impedance
- $I_{\%}$ : Percent of load that is constant current
- $P_{\%}$ : Percent of load that is constant power
- $Z_{\theta}$ : Phase angle of constant impedance fraction
- $I_{\theta}$ : Phase angle of constant current fraction
- $P_{\theta}$ : Phase angle of constant power fraction

In a time-variant load representation, the coefficients of the ZIP model,  $V_n$ ,  $S_n$ ,  $Z_{\%}$ ,  $I_{\%}$ ,  $P_{\%}$ ,  $Z_{\theta}$ ,  $I_{\theta}$ , and  $P_{\theta}$  remain constant, but the power consumption,  $P_i$  and  $Q_i$ , of the ith load varies with the actual terminal voltage,  $V_a$ . The ZIP model is similar to the polynomial representation used in many commercial software packages. In the polynomial representation of the ZIP load, the constant coefficient is roughly equivalent to the P fraction, the linear coefficient is roughly equivalent to the I fraction, and the quadratic coefficient is roughly equivalent to the Z fraction. The ZIP model only varies the power consumption as a function of actual terminal voltage,  $V_a$ . Note also that the sum of  $Z_{\%}$ ,  $I_{\%}$ , and  $P_{\%}$  must equal one.

Internally, the model performs all of the proper phase rotations and scaling for various voltage levels (e.g., the model would look the same for a house attached on Phase A versus Phase C). Currently, it is assumed that nominal voltage is either 120 or 240 V (this will be modified in future upgrades to residential\_enduse).

## Parameters

Property Name	Type	Unit	Description
base_power	double	kW	Base real power of the total load at nominal voltage.
power_pf	double	pu	Power factor for constant power portion of load.
current_pf	double	pu	Power factor for constant current portion of load.
impedance_pf	double	pu	Power factor for constant impedance portion of load.
actual_power	complex	kVA	[read only] Variable to monitor total power of load as a function of voltage.
heatgain_only	boolean	-	Toggles the zipload to generate heat only (no kW), is deactivated (false) by default
is_240	boolean	-	Toggles between a 120 V (unbalanced) vs. a 240 V (balanced) connection - true indicates it is a 240 V load.

*These variables are inherited from the enduse load structure.*

power_fraction	double	pu	The fraction of the load that is constant power.
current_fraction	double	pu	The fraction of the load that is constant current.
impedance_fraction	double	pu	The fraction of the load that is constant impedance.
heatgain_fraction	double	pu	The fraction of the total load (kW) that produces waste heat.

*These variables are used for creating cyclic load behavior.*

duty_cycle	double	pu	Fraction of time the device is in the <i>on</i> state.
period	double	hours	Time interval over which duty cycle is applied.
phase	double	pu	Indicates percentage of phase that the object is currently in; running period is assumed to be from 0.0 until percent of duty cycle. To create a distribution of devices, this variable should be randomized between 0 and 1.

*These variables are used for various aggregate and demand response modes.*

demand_response_mode	boolean	-	Activates equilibrium dynamic representation of demand response.
number_of_devices	int16	-	[Used with <i>demand_response_mode</i> only.] Number of devices to model - base power is the total load of all devices.
thermostatic_control_range	int16	K	[Used with <i>demand_response_mode</i> only.] Range of the thermostat's control operation.
number_of_devices_off	double	-	[Used with <i>demand_response_mode</i> only.] Total number of devices that are off.
number_of_devices_on	double	-	[Used with <i>demand_response_mode</i> only.] Total number of devices that are on.
rate_of_cooling	double	K/h	[Used with <i>demand_response_mode</i> only.] rate at which devices cool down.
rate_of_heating	double	K/h	[Used with <i>demand_response_mode</i> only.] rate at which devices heat up.
temperature	int16	K	[Used with <i>demand_response_mode</i> only.] temperature of the device's controlled media (eg air temp or water temp).
phi	double	pu	[Used with <i>demand_response_mode</i> only.] duty cycle of the device(s).
demand_rate	double	1/h	[Used with <i>demand_response_mode</i> only.] consumer demand rate that prematurely turns on a device or population.
nominal_power	double	kW	[Used with <i>demand_response_mode</i> only.] the rated amount of power demanded by devices that are on.
recovery_duty_cycle	double	pu	[Used with cycling mode and passive_controller duty cycle mode.] Fraction of time in the on state, while in recovery interval.
multiplier	double	pu	[Used with cycling mode and passive_controller duty cycle mode.] This variable is used to modify the base power as a function of multiplier times base_power.

**TODO:** : Document cycling, demand response and aggregate modes.

## Example models

This model is representative of a simple ZIPload with scheduled load behavior.

```
object ZIPload {
  name house1_load;
  parent house1;
  base_power responsive_loads*1.06;
  heatgain_fraction 0.90;
  power_pf 1.0;
  current_pf 1.0;
  impedance_pf 1.0;
  impedance_fraction 0.20;
  current_fraction 0.40;
  power_fraction 0.40;
};
```

This model is representative of a ZIPload used in cycling mode to roughly represent a schedule pool pump (note, pool\_pump\_season is a schedule of 0 or 1).

```
object ZIPload {
  name house1_poolpump;
  parent house1;
  base_power pool_pump_season*1.44;
  duty_cycle 0.22;
  phase 0.26;
  period 4.96;
  heatgain_fraction 0.0;
  power_pf 1.0;
  current_pf 1.0;
  impedance_pf 1.0;
  impedance_fraction 0.20;
  current_fraction 0.40;
  power_fraction 0.40;
  is_240 TRUE;
};
```

This model is representative of a ZIPload with a passive controller used to implement the elasticity model out of the market module.

```
object ZIPload {
  name house1_load;
  parent house1;
  base_power responsive_loads*1.06;
  heatgain_fraction 0.90;
  power_pf 1.0;
  current_pf 1.0;
  impedance_pf 1.0;
  impedance_fraction 0.20;
  current_fraction 0.40;
  power_fraction 0.40;
  object passive_controller {
    period 900;
    control_mode ELASTICITY_MODEL;
    two_tier_cpp true;
    observation_object Market_1;
    observation_property past_market.clearing_price;
    state_property multiplier;
    linearize_elasticity true;
    price_offset 0.01;
    critical_day CPP_days_R1.value;
  };
};
```

```

        first_tier_hours 12;
        second_tier_hours 12;
        third_tier_hours 6;
        first_tier_price 0.060483;
        second_tier_price 0.120965;
        third_tier_price 0.604826;
        old_first_tier_price 0.124300;
        old_second_tier_price 0.124300;
        old_third_tier_price 0.124300;
        daily_elasticity daily_elasticity_wtech*1.1731;
        sub_elasticity_first_second -0.1783;
        sub_elasticity_first_third -0.2604;
    };
};

```

This model is representative of a "pool pump" or cycling model that is using a DR control from the market module.

```

object ZIPload {
    name house1_poolpump;
    parent house1;
    base_power pool_pump_season*1.44;
    duty_cycle 0.22;
    phase 0.26;
    period 4.96;
    heatgain_fraction 0.0;
    power_pf 1.0;
    current_pf 1.0;
    impedance_pf 1.0;
    impedance_fraction 0.20;
    current_fraction 0.40;
    power_fraction 0.40;
    is_240 TRUE;
    recovery_duty_cycle 0.27;
    object passive_controller {
        period 900;
        control_mode DUTYCYCLE;
        pool_pump_model true;
        observation_object Market_1;
        observation_property past_market.clearing_price;
        state_property override;
        base_duty_cycle 0.22;
        setpoint duty_cycle;
        first_tier_hours 12;
        second_tier_hours 12;
        third_tier_hours 6;
        first_tier_price 0.060483;
        second_tier_price 0.120965;
        third_tier_price 0.604826;
    };
};

```

**TODO:** : Examples for cycling, demand response and aggregate modes.

## See also

- Powerflow User Guide
- Residential module
  - User's Guide
  - Appliances

- house class – Single-family home model.
- residential\_enduse class – Abstract residential end-use class.
- occupantload – Residential occupants (sensible and latent heat).
- **ZIPload** – Generic constant impedance/current/power end-use load.
- Technical Documents
  - Requirements
  - Specifications
  - Developer notes
  - Technical support document
  - Validation

Retrieved from "<http://gridlab-d.sourceforge.net/wiki/index.php?title=ZIPload&oldid=6195>"

---

- This page was last modified on 19 March 2013, at 20:27.